

POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'informazione
Corso di Laurea Magistrale in Ingegneria Nucleare
Dipartimento di Energia



**MODI ACUSTICI IN MEZZI STRATIFICATI
ANISOTROPI: SIMMETRIE CUBICA ED ESAGONALE**

Relatore: Prof. Marco G. BEGHI

**Tesi di Laurea di:
Davide TADDEI
Matr. 786560**

Anno Accademico 2013-2014

Alla mia famiglia

É nella crisi che emerge il meglio di ognuno, perchè senza crisi tutti i venti sono solo lievi brezze. Chi supera la crisi supera sé stesso senza essere superato.

Albert Einstein

Ringraziamenti

Ringrazio il prof. Beghi per la disponibilità e la gentilezza che mi ha mostrato durante tutto il tempo in cui mi ha seguito nel mio lavoro di tesi. Un sentito ringraziamento inoltre va a coloro che mi hanno permesso di raggiungere questo importante traguardo: i miei genitori, mio fratello Giordano, mia sorella Serena e in particolar modo il mio gemello Fabrizio, che più di tutti mi ha aiutato in questi anni universitari.

Sommario

Lo studio e la fabbricazione di materiali innovativi sta acquisendo una importanza sempre crescente in ambito energetico. La ricerca di prestazioni migliori ha portato con sè la necessità di avere a disposizione materiali con ben determinate caratteristiche. In tale contesto si è sviluppata la tecnologia del film sottile (thin film). La caratterizzazione del film, ovvero l'indagare le proprietà morfologiche, strutturali ed elettroniche viene fatta mediante l'uso di varie tecniche tra cui la spettroscopia vibrazionale Brillouin di superficie. I risultati vengono comparati con quelli di un modello teorico, nel quale sono state usate proprietà elastiche note. Quest'ultime vengono modificate in maniera tale da minimizzare l'errore tra i risultati sperimentali e quelli teorici. In questo modo si ricavano le proprietà elastiche del film sottile prodotto. Quindi molta importanza ricade sull'accuratezza del modello teorico utilizzato. Scopo di questa tesi è la creazione di un programma di calcolo di significativa generalità, scritto in MATLAB, che permetta di trovare le onde elastiche in mezzi stratificati, considerando oltre alla simmetria isotropa, anche quelle cubica ed esagonale. Esso è costruito per trattare sistemi multistrato ed il sistema monolayer risulta esserne un caso particolare. Non c'è alcun vincolo sul numero massimo degli strati, così come sul loro spessore. Si può fare qualsiasi tipo di combinazione tra le simmetrie considerate e gli studi possono essere effettuati in qualsiasi direzione.

Parole chiave: onde elastiche nei solidi, simmetria cubica, simmetria esagonale, MATLAB, film sottili, sistemi multistrato.

Abstract

The study and manufacture of innovative materials is gaining increasing importance in the energy field. The search for better performance has brought with it the need to have available materials with well-defined characteristics. In this context it has developed the technology of thin film. The characterization of the film, that is, to investigate the morphological, structural and electronic equipment is made by the use of various techniques including vibrational spectroscopy surface Brillouin. The results are compared with those of a theoretical model, in which the elastic properties are known. The latter are modified in such a way as to minimize the error between the experimental results and the theoretical ones. In this way we obtain the elastic properties of the thin film. So much importance falls on the accuracy of the theoretical model used. The purpose of this thesis is the creation of a calculation program of significant generality, written in MATLAB, which allows to find the elastic waves in stratified media, whereas in addition to the symmetry isotropic, even the cubic and hexagonal. It is constructed to treat multi-layer systems and the system being mono-layer is a particular case. There is no constraint on the maximum number of layers, as well as on their thickness. You can do any combination of the symmetries considered and studies can be performed in any direction.

Keywords: elastic waves in solids, cubic symmetry, hexagonal symmetry, MATLAB, thin films, multi-layer systems.

Indice

Sommario	III
Abstract	V
1 Introduzione	1
2 Teoria dell'elasticità	5
2.1 Introduzione	5
2.2 Deformazione e Sforzi	5
2.3 Il tensore delle costanti elastiche	8
2.4 La notazione matriciale (o di Voigt)	10
2.5 Simmetrie cristalline	12
3 Onde elastiche	17
3.1 Introduzione	17
3.2 Le onde di volume	17
3.3 Le onde superficiali	19
3.3.1 Il mezzo semi-infinito	20
3.3.2 Sistemi multistrati	25
3.3.3 Alcune considerazioni	27
4 Il programma	29
4.1 Il processo di calcolo	29
4.2 Procedura di investigazione	34
4.3 Verifiche sul Programma	35

5	Risultati	43
5.1	Radici dell'equazione secolare	43
5.2	Effetti dell'esagonalità del film	47
5.3	Simmetria cubica	55
5.4	Conclusioni	57
A	Codice	61
A.1	Prima Parte	61
A.2	Programmi Accessori	92
A.3	Seconda Parte	93
B	Manuale d'uso	95
	Bibliografia	99

Elenco delle figure

1.1	Immagine SEM di un film colonnare di TiO_2 , prodotto alla pressione di 10 Pa. Immagine tratta da [5].	3
2.1	Sistema esagonale compatto	14
2.2	Sistema cubico a corpo centrato	15
4.1	Schema logico del programma	30
4.2	Convenzione assunta sulla numerazione degli strati e sulla loro delimitazione	31
4.3	Verifica isotropia nel piano: caso isotropo.	37
4.4	Verifica isotropia nel piano: caso esagonale.	38
4.5	Verifica monolayer-bilayer equivalenti: caso isotropo.	39
4.6	Verifica monolayer-bilayer equivalenti: caso esagonale.	40
4.7	Variazione in numero delle onde di slab per nickel cubico.	41
5.1	Onda di Rayleigh per il mezzo isotropo "simil-silicio"	49
5.2	Onda Anisotropia	50
5.3	Pulsazione delle onde in funzione del rapporto $An(C_{11})$	51
5.4	Spostamenti dell'onda 1 per $An(C_{11}) = 0.4$	52
5.5	Spostamenti dell'onda 2 per $An(C_{11}) = 0.4$	53
5.6	Spostamenti dell'onda 3 per $An(C_{11}) = 0.4$	54
5.7	Determinante della matrice delle condizioni al contorno per spessori del film: a) e c) 100 nm; b) e d) 500nm.	55
5.8	Determinante della matrice delle condizioni al contorno per nickel cubico. (I)	56

5.9	Determinante della matrice delle condizioni al contorno per nickel cubico. (II)	57
B.1	Schermata tipo per un monolayer isotropo	96

Elenco delle tabelle

2.1	Costanti elastiche indipendenti per alcuni elementi	15
5.1	Andamento delle α_j in funzione della pulsazione ω : simmetria isotropa.	44
5.2	Andamento delle α_j in funzione della pulsazione ω : simmetria esagonale.	45
5.3	Andamento delle α_j in funzione della pulsazione ω : simmetria cubica.	46

Capitolo 1

Introduzione

Lo studio e la fabbricazione di materiali innovativi sta acquisendo una importanza sempre crescente in ambito energetico. La ricerca di prestazioni migliori ha portato con sé la necessità di avere a disposizione materiali con ben determinate caratteristiche. Spesso, però, esse non sono riscontrabili in un unico materiale. In tale contesto si è sviluppata la tecnologia del film sottile (thin film), con la quale si riesce a combinare diverse proprietà dei materiali. Essa consiste nella deposizione di un sottile strato di un dato elemento o composto, di cui si vogliono sfruttare determinate proprietà fisiche, sopra un substrato di spessore maggiore, del quale si vogliono usare altre proprietà, tra cui quelle meccaniche e strutturali. Lo spessore del film è al più di qualche micron, oltre questo valore si parla di film spesso (thick film). Sono stati messi a punto diversi metodi per la produzione dei film sottili tra cui la Pulsed Laser Deposition (PLD), evaporazione termica e sputtering ionico, che consentono di ottenere il film in condizioni controllate di pressione e temperatura. Da queste ultime dipende la struttura del film, dalla quale a loro volta dipendono molte proprietà fisiche. La fase successiva consiste nella caratterizzazione del film, ovvero nell'indagare le proprietà morfologiche, strutturali ed elettroniche. Ciò viene fatto mediante l'uso di varie tecniche tra cui la spettroscopia vibrazionale Brillouin di superficie. Essa è una tecnica di misura non distruttiva che utilizza lo scattering di luce per sondare le proprietà delle onde acustiche di superficie (SAWs) [19], [11]. Questa tecnica non utilizza l'eccitazione delle onde di superficie, ma si basa sulle loro spontanee fluttuazioni termiche. Il vettore d'onda delle SAWs

che si vogliono analizzare è determinato dalle condizioni sperimentali e con l'utilizzo di luce visibile la lunghezza d'onda delle onde da esplorare è dell'ordine di mezzo micron o meno. Le velocità tipiche delle SAWs di tali lunghezze d'onda corrispondono a frequenze che vanno da pochi GHz fino a 30 GHz. In un mezzo continuo elastico i gradi di libertà vibrazionali corrispondono a modi acustici di qualsiasi lunghezza d'onda. Il rumore termico è perciò dato dalla sovrapposizione di tutti i modi acustici. Nella spettroscopia vibrazionale Brillouin di superficie la geometria di scattering seleziona un particolare vettore d'onda ed il rumore termico viene sondato in corrispondenza di tale valore. Ciò viene fatto illuminando la superficie con un raggio laser di bassa potenza, in modo da avere un riscaldamento minimo. Dallo spettro della luce scatterata anelasticamente e dalla geometria, si può ricavare la relazione di dispersione delle onde acustiche di superficie. Questi risultati vengono poi comparati con quelli di un modello teorico, nel quale sono state usate proprietà elastiche note [12], [17]. Quest'ultime vengono modificate in maniera tale da minimizzare l'errore tra i risultati sperimentali e quelli teorici. In questo modo si ricavano le proprietà elastiche del film sottile prodotto. Ulteriori informazioni riguardo al modo in cui si calcolano le costanti elastiche utilizzando la spettroscopia Brillouin si trovano in [3], [2], [4], [15], [16] e [9]. Quindi molta importanza ricade sull'accuratezza del modello teorico utilizzato. Spesso, però, viene usato in prima approssimazione un modello omogeneo isotropo anche quando il film presenta altre simmetrie, causando la presenza di un errore intrinseco nei valori delle proprietà elastiche estrapolate. Esistono infatti diversi tipi di film. Possono essere monocristallini, in cui il reticolo cristallino si estende su tutto lo strato. Esso quindi possiede le proprietà del reticolo, in relazione alla risposta a sollecitazioni esterne, ed è quindi in generale anisotropo, ovvero la risposta cambia col variare della direzione. Poi ci sono film policristallini, composti da un insieme di piccoli cristalli, detti grani, i quali conservano singolarmente le caratteristiche del proprio reticolo. Le proprietà complessive del film, però, dipendono da come questi grani sono disposti all'interno di esso. Generalmente essi assumono una disposizione del tutto casuale ed in questo caso la risposta complessiva del film è isotropa, se vista su una scala di grandezza maggiore delle dimensioni dei grani. In alcuni casi, invece, la

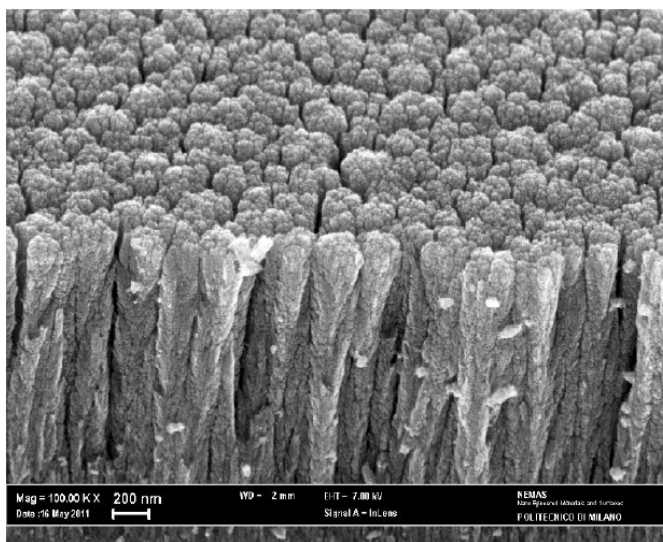


Fig. 1.1: Immagine SEM di un film colonnare di TiO_2 , prodotto alla pressione di 10 Pa. Immagine tratta da [5].

loro disposizione presenta delle direzioni preferenziali, dette texture (tessiture), che rendono il comportamento del film anisotropo lungo quelle direzioni. Esistono due tipi di texture nei film sottili: quella fibrosa, nella quale l'orientamento di un certo piano cristallino è preferenzialmente parallelo al piano del substrato, e la texture biassiale, in cui l'orientamento dei cristalliti è fissato anche in un piano parallelo alla superficie del campione. Vi sono poi i film composti da materiali amorfi, che sono per loro natura isotropi. Infine esistono i film nanostrutturati, dove i grani hanno dimensioni nanometriche e, se hanno direzioni preferenziali, comportano una risposta anisotropa. Essa non è quindi dovuta alla disposizione degli atomi, come accade nei cristalli, ma alla disposizione dei piccoli grani, perciò ad una scala mesoscopica, cioè intermedia tra quella atomica e quella macroscopica. Un esempio di film nanostrutturati sono quelli colonnari (Fig.1.1), nei quali la struttura è composta da un insieme di colonne. Essi non sono intrinsecamente omogenei, ma se vengono studiati con una lunghezza d'onda dell'ordine del mezzo micron, come avviene nella spettroscopia Brillouin, possono essere considerati omogenei in quanto la struttura colonnare non viene "vista". Tuttavia c'è una differenza nella risposta elastica tra la direzione parallela al piano su cui sono disposte le colonne e quella perpendicolare ad esso, ovvero nella direzione di crescita delle colonne. Infatti la disposizione delle colonne è del tutto casuale e quindi nel piano la risposta è isotropa, ma è diversa da quella perpendicolare al piano. Si può capire

questo intuitivamente pensando che è più facile "compattare fra di loro le colonne" rispetto a comprimerle nella loro direzione di crescita. Quindi, se visti in una scala di grandezza maggiore delle dimensioni delle colonne in modo da poter considerare il mezzo omogeneo, tali film presentano una simmetria esagonale.

Si è visto che esistono diversi tipi di film e che l'uso di un modello omogeneo ed isotropo risulta corretto solamente per alcuni di essi. In tutti gli altri casi esso comporta un errore intrinseco nella determinazione delle proprietà elastiche del film. Scopo di questa tesi è la creazione di un programma di calcolo scritto in MATLAB che permetta di trovare le onde elastiche, in particolare di superficie, in mezzi stratificati, considerando oltre alla simmetria isotropa, anche quelle cubica ed esagonale. La simmetria cubica è tipica della maggior parte degli elementi metallici e di alcuni composti, mentre quella esagonale è propria di alcuni elementi come lo zinco ed il cadmio, nonché dei film colonnari precedentemente citati. In tutti i casi si considera il mezzo omogeneo. Di seguito la tesi è così strutturata:

- Nel CAPITOLO 2 viene introdotta la teoria dell'elasticità, necessaria per trattare le onde elastiche.
- Il CAPITOLO 3 è dedicato alle onde elastiche, in particolare a quelle di superficie. In esso viene dettagliatamente trattato il metodo delle onde parziali.
- Nel CAPITOLO 4 viene presentato il programma di calcolo creato, basato sulla teoria presentata nei capitoli precedenti.
- Il CAPITOLO 5 è dedicato a dei risultati ottenuti utilizzando il programma per le tre simmetrie considerate: isotropo, cubico ed esagonale.

Capitolo 2

Teoria dell'elasticità

2.1 Introduzione

In questo capitolo viene introdotta la teoria dell'elasticità. Si definiscono i tensori degli sforzi e delle deformazioni, arrivando a trovare l'equazione di equilibrio statico per i corpi solidi. Questa è il punto di partenza per trovare l'equazione di equilibrio dinamico, che è la base per la trattazione delle onde elastiche. Infine viene introdotto il concetto di tensore delle costanti elastiche e come questo si riduce in determinate simmetrie cristalline.

2.2 Deformazione e Sforzi

I corpi solidi se sottoposti a forze si deformano in una certa misura, cioè cambiano di forma e volume. Ogni punto del corpo è identificato, in un sistema di coordinate, dal suo raggio vettore \mathbf{r} , le cui componenti sono $x_1 = x, x_2 = y, x_3 = z$. Nella deformazione del corpo, in generale, tutti i suoi punti si spostano. Considerandone uno qualunque, se \mathbf{r} è il raggio vettore prima della deformazione, a seguito di questa esso diviene \mathbf{r}' , di componenti x'_1, x'_2, x'_3 . Allora lo spostamento di tale punto è rappresentato dal vettore $\mathbf{r}' - \mathbf{r}$, designato con \mathbf{u} :

$$u_i = x'_i - x_i, i = 1, 2, 3. \quad (2.1)$$

Tale vettore si definisce *vettore spostamento*. Nel corso della deformazione di un corpo variano le mutue distanze tra i suoi punti. Consideriamone due infinitamente

vicini. Se dx_i indicano le componenti del vettore differenza tra i corrispettivi raggi vettori prima della deformazione, queste si trasformano in $dx'_i = dx_i + du_i$. Il quadrato della distanza prima della deformazione è $dl^2 = dx_i^2$, dopo diventa $dl'^2 = dx_i'^2 = (dx_i + du_i)^2$, dove si è scritto facendo uso della convenzione usuale secondo cui si omette il simbolo di somma, intendendo che indici ripetuti sottintendono una somma sul loro campo di variabilità (da 1 a 3). Sostituendo $du_i = \frac{\delta x_i}{\delta x_k} dx_k$ nell'espressione di dl'^2 , con qualche manipolazione matematica, si introduce

$$\varepsilon_{ik} = \frac{1}{2} \left(\frac{\delta u_i}{\delta x_k} + \frac{\delta u_k}{\delta x_i} + \frac{\delta u_l}{\delta x_i} \frac{\delta u_l}{\delta x_k} \right)$$

Questo si definisce *tensore di deformazione*, è simmetrico e determina la variazione dell'elemento di lunghezza e della sua orientazione a seguito della deformazione del corpo. Nella quasi totalità dei casi che si considerano le deformazioni dei corpi sono piccole. Ciò significa che la variazione delle distanze dei punti nel corpo è piccola raffrontata alle distanze stesse. In altri termini, le dilatazioni relative sono infinitesimi di ordine superiore. Quindi nell'espressione di ε_{ik} si può trascurare l'ultimo termine in quanto infinitesimo del secondo ordine, ottenendo il tensore di deformazione per piccole deformazioni:

$$\varepsilon_{ik} = \frac{1}{2} \left(\frac{\delta u_i}{\delta x_k} + \frac{\delta u_k}{\delta x_i} \right) \quad (2.2)$$

Si consideri ora la situazione in cui un corpo solido in equilibrio è soggetto a forze esterne. Il sistema di forze che complessivamente si instaurano è duplice: le forze di volume e le forze di superficie. Le *forze di volume* sono dipendenti unicamente dai campi esterni che agiscono sul solido e vengono descritte dal vettore $\mathbf{b}(x)$ che rappresenta la loro densità, così che la forza totale $d\mathbf{F}_V$ applicata ad un piccolo volume dV centrato sul punto x sia $d\mathbf{F}_V = \mathbf{b}(x)dV$. Il vettore $\mathbf{b}(x)$ rappresenta quindi una forza per unità di volume. Il caso tipico è quello delle forze di gravità che sono proporzionali alla massa del volume di solido su cui agiscono. In questo caso possiamo scrivere $d\mathbf{F}_V = \mathbf{g}dm$, dove \mathbf{g} è l'accelerazione di gravità, mentre dm è la massa locale del volume dV . Prendendo poi $\rho = \frac{dm}{dV}$ come densità di massa, allora si ha che $\mathbf{b}(x) = \rho\mathbf{g}$. Le *forze di superficie* sono dovute alle azioni di forza che ciascuna porzione di solido subisce dalla parte rimanente del mezzo continuo. Se il corpo è indeformato tali forze sono nulle. Ma se il corpo è deformato, esso si

porta fuori dal suo stato di equilibrio iniziale e nascono forze, dette *sforzi interni*, che tendono a riportare il corpo nel suo stato di equilibrio. Gli sforzi interni sono dovuti a forze molecolari, cioè forze di interazione tra molecole. Un fatto essenziale nell'elasticità è che le forze molecolari possiedono un "raggio d'azione" trascurabile, estendendo la loro influenza su lunghezze dell'ordine della distanza tra molecole vicine. Nella teoria dell'elasticità intervengono soltanto lunghezze grandi rispetto alle distanze tra molecole, pertanto si considera come nullo il "raggio d'azione" delle forze molecolari. Ciò equivale a dire, in un certo senso, che le forze che provocano gli sforzi interni sono forze a "corto raggio", esercitando in ogni punto la loro azione solo sui punti vicini. Ne consegue che le forze, con le quali una parte qualunque del corpo è sollecitata dalle parti contigue, agiscono solo direttamente attraverso la superficie di tale parte.

La convenzione universalmente accettata prevede che, definito il versore normale uscente \mathbf{n} all'elemento di superficie dS , una forza di superficie sia negativa se orientata oppostamente ad \mathbf{n} , positiva invece se concorde. Dunque le forze di compressione sono negative, mentre quelle di trazione positive. Si introduce inoltre la seguente notazione per la forza di superficie infinitesima $d\mathbf{F}_S$ agente sull'elemento dS

$$d\mathbf{F}_S = \mathbf{f}dS$$

dove \mathbf{f} assume il significato fisico di una forza per unità di superficie. Il teorema di Cauchy dimostra analiticamente che esiste (ed è unico) un tensore $\hat{\sigma}$ di rango 3x3 tale per cui

$$\mathbf{f} = \hat{\sigma}\mathbf{n}.$$

Grazie a questo teorema si può scrivere la forza di superficie infinitesima come

$$d\mathbf{F}_S = \hat{\sigma}\mathbf{n}dS$$

, ovvero

$$d\mathbf{F}_{S,i} = \sigma_{ik}n_k dS \longrightarrow \frac{d\mathbf{F}_{S,i}}{dS} = \sigma_{ik}n_k. \quad (2.3)$$

σ_{ik} viene detto *tensore degli sforzi*, mentre $\sigma_{ik}dS_k$ è la componente della forza che agisce sull'elemento di superficie dS . Scegliendo gli elementi di superficie nei piani xy , yz , xz , si trova che la componente σ_{ik} del tensore degli sforzi è l' i -esima

componente della forza agente sull'unità di superficie perpendicolare all'asse x_k . Dal calcolo del momento delle forze che agiscono su un volume del corpo si ottiene che dev'essere $\sigma_{ik} - \sigma_{ki} = 0$, cioè

$$\sigma_{ik} = \sigma_{ki}$$

ovvero che il tensore degli sforzi dev'essere simmetrico.

All'equilibrio le forze degli sforzi interni debbono compensarsi mutuamente in ogni elemento di volume, cioè deve aversi $F_i = 0$. In questo modo le equazioni di equilibrio di un corpo deformato si scrivono

$$\frac{\delta\sigma_{ik}}{\delta x_k} = 0. \quad (2.4)$$

Nel campo di gravità dovrà annullarsi la somma $\mathbf{F} + \rho g$ delle forze degli sforzi interni e della forza di gravità ρg agente sull'unità di volume (ρ è la densità, g è l'accelerazione di gravità). Le equazioni dell'equilibrio in questo caso divengono [14]

$$\frac{\delta\sigma_{ik}}{\delta x_k} + \rho g_i = 0. \quad (2.5)$$

In questo paragrafo si sono ricavati i tensori di sforzi e deformazioni i quali contengono ciascuno 9 componenti, ma essendo entrambi simmetrici quelle indipendenti si riducono per ciascun tensore a 6. Essi assumono la seguente forma

$$\hat{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{31} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{23} & \sigma_{33} \end{bmatrix} \quad \hat{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{31} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix}$$

2.3 Il tensore delle costanti elastiche

Si è visto che quando un corpo è sottoposto a forze esterne esso si deforma ed a seguito di tale evento si generano all'interno del corpo sforzi che tendono a riportarlo alla condizione di equilibrio iniziale. Esiste quindi un legame diretto tra sforzi e deformazioni che viene formalmente espresso attraverso l'*equazione costitutiva*

$$\sigma_{ij} = f(\varepsilon_{ij}).$$

Essa deve riflettere il comportamento fisico (cioè sperimentalmente osservabile) del solido e non si può ricavare dalla struttura formale della meccanica dei solidi, ma

deve essere assunta a priori. Ogni risultato del continuo, dunque, è profondamente legato alla specifica equazione costitutiva che è stata adottata sulla base della conoscenza fenomenologica del mezzo fisico o sulla base della convenienza formale (o numerica). Per molti materiali, specialmente se sottoposti a piccole deformazioni, tale relazione risulta lineare. Questi mezzi e le corrispondenti equazioni costitutive vengono detti lineari. La nostra trattazione rientra in questa categoria e l'equazione costitutiva nell'approssimazione *elastica lineare* assume la seguente forma

$$\sigma_{ij} = C_{ijkh}\varepsilon_{kh} \quad (i, j, k, h = 1, 2, 3) \quad (2.6)$$

dove le C_{ijkh} sono opportune costanti. L'Eq. (2.6) è di validità generale, comprendendo cioè ogni possibile caso di simmetria cristallina e/o anisotropia. Questa equazione è una generalizzazione della famosa legge di Hooke $F = kx$. Il tensore \hat{C} è noto come *tensore elastico* o *delle costanti elastiche* e poichè gli indici i, j, k, h assumono i valori 1, 2, 3 esso ha $3^4 = 81$ componenti. Il numero effettivo delle sue componenti indipendenti è ridotto notevolmente da relazioni matematiche universalmente valide:

- La simmetria del tensore degli sforzi impone che

$$C_{ijkh} = C_{jikh}$$

- La simmetria del tensore delle deformazioni impone che

$$C_{ijkh} = C_{ijhk}$$

- Argomentazioni termodinamiche impongono che

$$C_{ijkh} = C_{khij}$$

Grazie a queste simmetrie il numero delle costanti indipendenti si riduce a 21, nel caso più generale. Un'ulteriore riduzione del numero di costanti indipendenti è imposta dalla simmetria cristallina del solido considerato.[6]

È opportuno fare una precisazione sulla natura dei tensori. Essi rappresentano una "quantità fisica" che ha una propria esistenza ed è indipendente dalla particolare

scelta del sistema di riferimento che facciamo. Quando cambiamo gli assi la quantità fisica non cambia, ma solo il nostro modo di rappresentarla. Ciò non è vero per una qualsiasi matrice, la quale, nonostante si presenti, analogamente ai tensori, come un array di numeri, esiste ed è definita solamente in concomitanza di un set di assi. Per essa non si può parlare di trasformazione rispetto ad altri assi di riferimento, poichè sarebbe del tutto privo di significato.[18]

2.4 La notazione matriciale (o di Voigt)

Le tre relazioni di simmetria discusse nel paragrafo precedente suggeriscono di utilizzare una notazione semplificata, detta di Voigt. Anzichè rappresentare i tensori $\hat{\varepsilon}$ e $\hat{\sigma}$ tramite le corrispondenti matrici $\{\varepsilon_{ij}\}$ e $\{\sigma_{ij}\}$, è conveniente utilizzare dei vettori colonna i cui elementi siano tutte e sole le sei componenti indipendenti della deformazione e dello sforzo. Per formalizzare questa convenienza bisogna innanzitutto ricordare che si è sempre indicato le direzioni cartesiane (x, y, z) con gli indici $(1, 2, 3)$. Le sei componenti del tensore degli sforzi vengono arrangiate in un unico vettore colonna nel modo seguente:

$$\bar{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{zx} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{bmatrix}$$

Analogamente si procede per il tensore delle deformazioni aggiungendo però un fattore 2 alle componenti yz, zx, xy :

$$\bar{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{zx} \\ 2\varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \\ 2\varepsilon_{12} \end{bmatrix}$$

I doppi indici vengono abbreviati in uno solo variabile da 1 a 6 in accordo con lo schema seguente

notazione tensoriale	11	22	33	23,32	31,13	12,21
notazione matriciale	1	2	3	4	5	6

Così facendo i due vettori diventano:

$$\bar{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} \quad \bar{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ 2\varepsilon_4 \\ 2\varepsilon_5 \\ 2\varepsilon_6 \end{bmatrix}$$

Similmente anche le coppie di indici i, j e k, h del tensore delle costanti elastiche, le quali corrispondono rispettivamente agli indici del tensore degli sforzi e a quello delle deformazioni, vengono ridotte ad una soltanto ottenendo la corrispondente matrice delle costanti elastiche:

$$\bar{\mathbf{C}} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix}$$

Per la simmetria dovuta ad argomentazioni termodinamiche anche questa matrice è simmetrica. Con questa notazione l'Eq. (2.6) diventa

$$\sigma_i = C_{ij}\varepsilon_j \quad (i, j = 1, 2, \dots, 6). \quad (2.7)$$

Bisogna, purtroppo, segnalare una possibile sorgente di confusione legata ad una mancanza di universalità nell'adozione della notazione compatta: mentre i primi tre elementi del vettore colonna delle deformazioni sono ovunque definiti come precedentemente esposto, si trovano invece ordinamenti differenti per gli ultimi

tre elementi. Inoltre, altre volte il vettore delle deformazioni in notazione di Voigt viene definito senza l'introduzione negli ultimi tre termini del fattore 2. Tuttavia in questo modo tali coefficienti finiscono nella matrice delle costanti elastiche affinché l'Eq. (2.7) rimanga in tale forma.

È immediato rendersi conto che le componenti indipendenti di $\bar{\mathbf{C}}$ sono 21. Infatti una matrice simmetrica $n \times n$ ha $n(n+1)/2$ elementi indipendenti, come risulta evidente da un conteggio elementare. Ponendo $n = 6$ si ottengono subito 21 elementi per i corrispondenti tensori elastici.

Con questa notazione il tensore $\hat{\mathbf{C}}$ è estremamente utile per definire il tipo di anisotropia presente nel mezzo e, quindi, per identificare il tipo di simmetria cristallina del solido che si sta considerando. [6], [18]

2.5 Simmetrie cristalline

La presenza di simmetria nel cristallo riduce ulteriormente il numero di costanti indipendenti. Un caso di grande importanza (sia sotto il profilo concettuale, sia sotto il profilo applicativo) è quello di un mezzo *continuo, omogeneo ed isotropo*. Per omogeneità si intende che il comportamento meccanico del mezzo è identico in tutti i suoi punti: formalmente cioè equivale a dire che il tensore elastico $\hat{\mathbf{C}}$ non dipende dalle coordinate \mathbf{r} del punto considerato. Nella nostra trattazione rimarremo sempre nel caso di omogeneità. D'altro canto, per isotropia si intende che le proprietà meccaniche non dipendono dalla direzione considerata: formalmente ciò equivale a dire che tali proprietà sono invarianti per rotazioni. Un esempio di mezzo elastico, omogeneo ed isotropo è quello di un materiale che, a parità di trazione, si allunga della stessa quantità indipendentemente dalla direzione lungo la quale viene applicato lo sforzo ed indipendentemente dal punto ove viene applicato lo stesso. Un mezzo che gode delle proprietà di omogeneità, isotropia e linearità (elasticità) viene detto *mezzo normale*. [6]

Nel caso isotropo il numero di costanti indipendenti si riduce a 2 soltanto, generalmente identificate in C_{11} e C_{12} oppure C_{11} e C_{44} . La matrice delle costanti

elastiche assume la seguente forma

$$\bar{C}_{isotropo} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(C_{11} - C_{12}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(C_{11} - C_{12}) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(C_{11} - C_{12}) \end{bmatrix}. \quad (2.8)$$

Il sistema isotropo è, ovviamente, quello con la maggiore simmetria e, perciò, il minor numero di costanti indipendenti. Per tutti gli altri casi questo numero cresce parallelamente al diminuire delle simmetrie (assi, piani, punti) e quindi al crescere dell'anisotropia, come riportato nella seguente tabella per i principali sistemi cristallini.

triclino	21
monoclino	13
ortorombico	9
tetragonale	6
romboedrico	6
esagonale	5
cubico	3

In particolare tratteremo oltre al caso isotropo, quello esagonale e quello cubico.

Per il sistema esagonale, raffigurato in figura 2.1, il numero delle costanti indipendenti diventa 5: C_{11} , C_{12} , C_{13} , C_{33} e C_{44} . La matrice delle costanti elastiche assume la forma di seguito indicato.

$$\bar{C}_{esagonale} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(C_{11} - C_{12}) \end{bmatrix}. \quad (2.9)$$

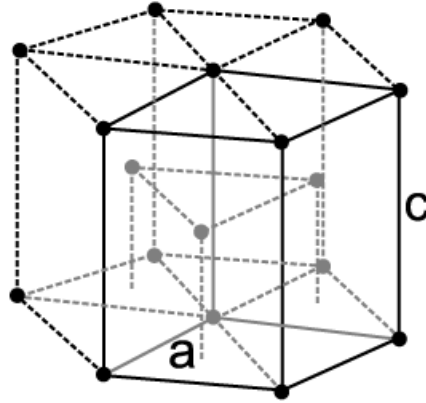


Fig. 2.1: Sistema esagonale compatto

Esistono delle restrizioni sui valori che tali costanti possono assumere, dovute al fatto che l'energia di deformazione del cristallo dev'essere positiva. Per il sistema esagonale esse sono:

$$C_{44} > 0, \quad C_{11} > |C_{12}|, \quad (C_{11} + C_{12})C_{33} > 2C_{13}^2. \quad (2.10)$$

Per il sistema cubico, raffigurato in figura 2.2, il numero delle costanti indipendenti è invece 3: C_{11} , C_{12} e C_{44} . La matrice delle costanti elastiche assume la forma di seguito indicato.

$$\bar{C}_{cubico} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix}. \quad (2.11)$$

Anche per esso ci sono restrizioni dovute all'energia di deformazione che in questo caso diventano:

$$C_{44} > 0, \quad C_{11} > |C_{12}|, \quad C_{11} + 2C_{12} > 0. \quad (2.12)$$

Nella tabella 2.1 vengono indicati i valori delle costanti elastiche indipendenti per alcuni materiali.[10],[18]

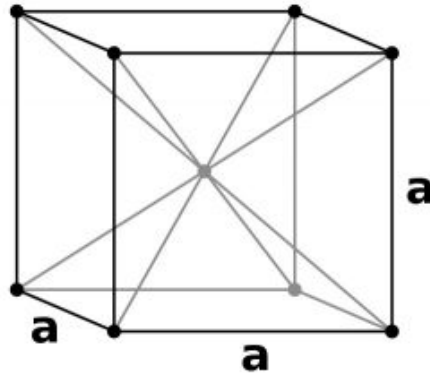


Fig. 2.2: Sistema cubico a corpo centrato

Tabella 2.1: Costanti elastiche indipendenti per alcuni elementi

Cristallo	Simmetria	Costanti elastiche $\times 10^{11}$ (N/m^2)				
		C_{11}	C_{12}	C_{44}	C_{33}	C_{13}
Sodio	cubica	0,074	0,062	0,042	/	/
Piombo	cubica	0,466	0,392	0,1443	/	/
Rame	cubica	1,684	1,214	0,752	/	/
Argento	cubica	1,240	0,934	0,4605	/	/
Oro	cubica	2,02	1,697	0,4603	/	/
Nickel	cubica	2,61	1,510	1,3090	/	/
Germanio	cubica	1,293	0,485	0,6747	/	/
Silicio	cubica	1,657	0,639	0,7991	/	/
Alluminio	cubica	1,073	0,609	0,2830	/	/
Tungsteno	cubica	5,13	2,06	1,5273	/	/
Molibdeno	cubica	4,60	1,760	1,1005	/	/
Zinco	esagonale	1,6355	0,2656	0,3788	0,6293	0,5167
Cadmio	esagonale	1,2059	0,4812	0,1852	0,5133	0,4420

Capitolo 3

Onde elastiche

3.1 Introduzione

In questo capitolo vengono introdotte le equazioni di equilibrio dinamico e viene spiegato come da esse si può ottenere il vettore spostamento relativo alle onde elastiche. Dopo un breve accenno alle onde di volume, viene dettagliatamente analizzato un metodo utile per determinare le onde di superficie, detto *metodo delle onde parziali*.

3.2 Le onde di volume

Nel capitolo precedente si è ricavata l'equazione (2.5), che è l'equazione di equilibrio statico di un corpo solido immerso nel campo di gravità. Nell'equilibrio dinamico la somma dei contributi dovuti agli sforzi interni e alle forze di volume non è nulla, ma bisogna uguagliarla al termine accelerativo $\rho \ddot{u}_i$, dove i due punti indicano la derivata seconda rispetto al tempo e ρ è la densità del mezzo. Così facendo l'equazione diventa:

$$\rho \frac{\delta^2 u_i}{\delta t^2} = \frac{\delta \sigma_{ij}}{\delta x_j} + \rho g_i \quad (i, j = 1, 2, 3). \quad (3.1)$$

Sostituendo in essa l'equazione costitutiva (2.6), che lega gli sforzi alle deformazioni, e la definizione del tensore di deformazione per piccole deformazioni (2.2) e

trascurando la forza peso si ottiene

$$\rho \frac{\delta^2 u_i}{\delta t^2} = C_{ijkl} \frac{\delta^2 u_k}{\delta x_j \delta x_l} \quad (i, j, k, l = 1, 2, 3) \quad (3.2)$$

che è l'equazione delle onde per un mezzo elastico ed omogeneo.

Se il mezzo è infinito in tutte le direzioni, allora le soluzioni più semplici dell'equazione delle onde sono onde piane e monocromatiche definite dalla parte reale di

$$u_i = A_i \exp[ik(l_j x_j - vt)]. \quad (3.3)$$

Ogni qualsiasi altra onda elastica può essere scritta come combinazione lineare di esse. La velocità di fase dell'onda è v misurata lungo la direzione del vettore d'onda \mathbf{k} , i cui coseni direttori sono gli l_j . Le onde piane di questa forma vengono dette onde di volume (in inglese *bulk waves*).

La soluzione per la velocità di fase e i corrispondenti autovettori A_i dello spostamento nel mezzo infinito (bulk-wave problem) può essere determinata esplicitamente sostituendo l'onda piana di Eq.(3.3) nell'equazione (3.2). Questa sostituzione fornisce il seguente set omogeneo di equazioni

$$(\Gamma_{ik} - \delta_{ik} \rho v^2) A_i = 0$$

dove $\Gamma_{ik} = l_j l_l C_{ijkl}$ e δ_{ik} è la delta di Kronecker. Ponendo il determinante dei coefficienti uguale a zero,

$$|\Gamma_{ik} - \delta_{ik} \rho v^2| = 0$$

si produce un'equazione cubica in v^2 (equazione di Christoffel), dove le tre radici sono il quadrato delle velocità per le tre onde di volume. Gli autovettori delle tre soluzioni formano una triade mutuamente perpendicolare, essendo un vettore spostamento quasi-longitudinale e gli altri due quasi-trasversi. Per propagazioni lungo direzioni ad alta simmetria gli autovettori sono rispettivamente paralleli e perpendicolari al vettore d'onda e tutte e tre le onde diventano modi puri con il flusso di energia anch'esso parallelo al vettore d'onda, ma per la propagazione in una generica direzione la velocità di gruppo non è parallela al vettore d'onda. [10]

Il caso con maggior simmetria in assoluto è per definizione il caso isotropo, dove, come si è già visto precedentemente, il numero delle costanti indipendenti del tensore

elastico si riduce a 2 e di seguito verranno identificate in C_{11} e C_{12} . Risolvendo in questa geometria il sistema precedente si ricava che le onde elastiche in un mezzo infinito ed isotropo possono propagarsi in due differenti modi: il modo-P, che è il modo longitudinale, e il modo-S, che identifica i due modi trasversali, in questo caso degeneri. Quando un'onda elastica propaga come un'onda P vengono generati solo sforzi normali (di compressione o di dilatazione) nel solido e la velocità dell'onda è $c_p = \sqrt{\frac{C_{11}}{\rho}}$. Se invece l'onda elastica propaga come un'onda S, allora compaiono solo sforzi di taglio nel solido e la velocità dell'onda è $c_p = \sqrt{\frac{C_{11}-C_{12}}{2\rho}} = \sqrt{\frac{C_{44}}{\rho}}$. [13]

Le onde longitudinali risultano essere più veloci di quelle trasverse e per questo le prime vengono dette P (*primarie*) e le altre S (*secondarie*). Le onde P sono, fra le onde generate da un terremoto, le più veloci, e, dunque, le prime avvertite da una stazione sismica. Le onde S invece vengono sempre avvertite in un secondo tempo: esse raggiungono velocità che si aggirano solitamente intorno al 60-70% della velocità delle onde P. [6]

3.3 Le onde superficiali

Fino ad ora si è trattato di onde elastiche in grado di propagarsi in un mezzo che è infinito in tutte le direzioni e per questo chiamate onde di volume. Tuttavia ogni solido possiede delle superfici che ne delimitano le dimensioni. Se si introduce nel mezzo infinito trattato precedentemente una superficie infinita che lo taglia in due parti e se ne considera una soltanto si ottiene un mezzo semi-infinito. Applicando sulla superficie la condizione di *superficie libera*, cioè l'annullarsi degli sforzi in essa, si è ottenuto che sono possibili altri tipi di onde, dette *onde di superficie* ed il primo a proporre l'esistenza fu Rayleigh nel 1887. Queste onde sono modi di propagazione dell'energia elastica lungo la superficie libera di un mezzo semi-infinito nei quali l'ampiezza dello spostamento dell'onda di propagazione decade in maniera esponenziale con la profondità al di sotto della superficie, così che essenzialmente tutta la densità di energia è concentrata in una distanza dell'ordine di una lunghezza d'onda dalla superficie libera. [10]

3.3.1 Il mezzo semi-infinito

Consideriamo un sistema di riferimento cartesiano ortogonale (x_1, x_2, x_3) , dove gli assi x_1 e x_2 giacciono sulla superficie libera con una orientazione conveniente in base al tipo di problema, mentre l'asse x_3 , perpendicolare alla superficie, è rivolto verso il mezzo semi-infinito, il quale giace quindi nel semispazio con $x_3 \geq 0$. Le soluzioni che stiamo cercando sono onde che viaggiano parallele alla superficie libera, quindi con vettore d'onda \mathbf{k} avente due componenti (k_1, k_2) . Il vettore d'onda è un dato iniziale del problema, cioè cercheremo soluzioni lungo una determinata direzione nel piano x_1x_2 con un dato valore del modulo di \mathbf{k} , fissando i valori (k_1, k_2) . Le incognite da trovare per determinare completamente l'onda sono la dipendenza di quest'ultima dall'asse x_3 , racchiusa nel parametro α , uguale per tutti e tre gli spostamenti u_1, u_2, u_3 , e le tre ampiezze di quest'ultimi A_1, A_2, A_3 . L'onda di superficie è perciò data dalla combinazione lineare di termini aventi la seguente forma:

$$u_i = A_i \exp[i(k_1x_1 + k_2x_2 - \omega t)] \exp(-\alpha x_3) \quad (i = 1, 2, 3) \quad (3.4)$$

Sostituendo questo tipo di soluzione nel set di equazioni (3.2) si ottiene il sistema

$$\begin{bmatrix} \Gamma_1 & \Gamma_2 & \Gamma_3 \\ \Gamma_4 & \Gamma_5 & \Gamma_6 \\ \Gamma_7 & \Gamma_8 & \Gamma_9 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = 0 \quad (3.5)$$

dove in notazione matriciale i Γ_l assumono la seguente forma:

$$\begin{aligned} \Gamma_1 &= C_{55}\alpha^2 - C_{11}k_1^2 - C_{66}k_2^2 + \rho\omega^2 - C_{15}\alpha k_1 i - C_{51}\alpha k_1 i - C_{56}\alpha k_2 i - C_{65}\alpha k_2 i \\ &\quad - C_{16}k_1 k_2 - C_{61}k_1 k_2; \end{aligned}$$

$$\begin{aligned} \Gamma_2 &= C_{54}\alpha^2 - C_{16}k_1^2 - C_{62}k_2^2 - C_{14}\alpha k_1 i - C_{52}\alpha k_2 i - C_{56}\alpha k_1 i - C_{64}\alpha k_2 i - C_{12}k_1 k_2 \\ &\quad - C_{66}k_1 k_2; \end{aligned}$$

$$\begin{aligned} \Gamma_3 &= C_{53}\alpha^2 - C_{15}k_1^2 - C_{64}k_2^2 - C_{13}\alpha k_1 i - C_{54}\alpha k_2 i - C_{55}\alpha k_1 i - C_{63}\alpha k_2 i - C_{14}k_1 k_2 \\ &\quad - C_{65}k_1 k_2; \end{aligned}$$

$$\Gamma_4 = C_{45}\alpha^2 - C_{26}k_2^2 - C_{61}k_1^2 - C_{25}\alpha k_2 i - C_{41}\alpha k_1 i - C_{46}\alpha k_2 i - C_{65}\alpha k_1 i - C_{21}k_1 k_2 - C_{66}k_1 k_2;$$

$$\Gamma_5 = C_{44}\alpha^2 - C_{22}k_2^2 - C_{66}k_1^2 + \rho\omega^2 - C_{24}\alpha k_2 i - C_{42}\alpha k_2 i - C_{46}\alpha k_1 i - C_{64}\alpha k_1 i - C_{26}k_1 k_2 - C_{62}k_1 k_2;$$

$$\Gamma_6 = C_{43}\alpha^2 - C_{24}k_2^2 - C_{65}k_1^2 - C_{23}\alpha k_2 i - C_{44}\alpha k_2 i - C_{45}\alpha k_1 i - C_{63}\alpha k_1 i - C_{25}k_1 k_2 - C_{64}k_1 k_2;$$

$$\Gamma_7 = C_{35}\alpha^2 - C_{46}k_2^2 - C_{51}k_1^2 - C_{31}\alpha k_1 i - C_{36}\alpha k_2 i - C_{45}\alpha k_2 i - C_{55}\alpha k_1 i - C_{41}k_1 k_2 - C_{56}k_1 k_2;$$

$$\Gamma_8 = C_{34}\alpha^2 - C_{42}k_2^2 - C_{56}k_1^2 - C_{32}\alpha k_2 i - C_{36}\alpha k_1 i - C_{44}\alpha k_2 i - C_{54}\alpha k_1 i - C_{46}k_1 k_2 - C_{52}k_1 k_2;$$

$$\Gamma_9 = C_{33}\alpha^2 - C_{44}k_2^2 - C_{55}k_1^2 + \rho\omega^2 - C_{34}\alpha k_2 i - C_{35}\alpha k_1 i - C_{43}\alpha k_2 i - C_{53}\alpha k_1 i - C_{45}k_1 k_2 - C_{54}k_1 k_2;$$

Affinchè il sistema omogeneo abbia soluzione diversa dalla soluzione banale $[A_i] = 0$, bisogna che il determinante della matrice associata al sistema, d'ora in poi chiamata $\bar{\Gamma}$, sia uguale a zero. Quindi ponendo

$$\det(\bar{\Gamma}) = 0, \quad (3.6)$$

si ottiene un'equazione di sesto grado in α , detta *equazione secolare*, con la pulsazione ω come parametro, poichè k_1 e k_2 sono posti come dato iniziale, così come la densità ρ e le costanti elastiche che dipendono dal tipo di mezzo considerato. Dal momento che le componenti del vettore d'onda, la densità e le costanti elastiche sono quantità reali, l'equazione secolare è un'equazione razionale a coefficienti reali di sesto grado e quindi ammette 6 radici che possono essere reali o complesse coniugate. Per queste soluzioni le tre equazioni del sistema (3.5) diventano dipendenti, quindi si può prendere un'ampiezza A_i come riferimento ed esprimere le altre due in funzione di essa. Per esempio prendiamo A_3 come riferimento ed utilizziamo le prime due equazioni (la scelta è arbitraria, si possono utilizzare anche la prima e la terza o la seconda e la terza), allora il sistema diventa

$$\begin{bmatrix} \Gamma_1 & \Gamma_2 \\ \Gamma_4 & \Gamma_5 \end{bmatrix} \begin{bmatrix} \frac{A_1}{A_3} \\ \frac{A_2}{A_3} \end{bmatrix} = \begin{bmatrix} \Gamma_3 \\ \Gamma_6 \end{bmatrix}$$

dal quale si ricavano i seguenti due rapporti:

$$\eta_1 = \frac{A_1}{A_3} = \frac{\Gamma_3 - \Gamma_2}{\Gamma_1 - \Gamma_2}, \quad \eta_2 = \frac{A_2}{A_3} = \frac{\Gamma_6 - \Gamma_4}{\Gamma_5 - \Gamma_4}. \quad (3.7)$$

Tali rapporti dipendono dalle radici α_j ($j = 1; \dots, 6$) e ovviamente dal parametro ω . La scelta dell'ampiezza di riferimento è, in generale, del tutto arbitraria, tuttavia potrebbero capitare dei casi in cui uno dei due rapporti diverga per particolari valori delle α_j . In tal caso bisogna prendere una delle altre due ampiezze come riferimento, verificando nuovamente che non diverga per tutte le α_j . Si sono ottenute quindi le radici dell'equazione secolare α_j , a ognuna delle quali sono associati i due specifici rapporti η_{1j} e η_{2j} . Per ogni esponente α_j si possono esprimere gli spostamenti u_{1j} , u_{2j} e u_{3j} solo in funzione delle ampiezze di riferimento A_{3j} utilizzando i rapporti trovati precedentemente. La soluzione finale che stiamo cercando è data quindi dalla loro combinazione lineare, cioè assume la seguente forma:

$$\bar{\mathbf{u}} = \begin{cases} u_1 = \sum_{j=1}^6 A_{3j} \eta_{1j} \exp[i(k_1 x_1 + k_2 x_2 - \omega t)] \exp(-\alpha_j x_3) \\ u_2 = \sum_{j=1}^6 A_{3j} \eta_{2j} \exp[i(k_1 x_1 + k_2 x_2 - \omega t)] \exp(-\alpha_j x_3) \\ u_3 = \sum_{j=1}^6 A_{3j} \exp[i(k_1 x_1 + k_2 x_2 - \omega t)] \exp(-\alpha_j x_3) \end{cases} \quad (3.8)$$

Questa procedura è del tutto generale, in quanto non si è fatta alcuna ipotesi sulla configurazione del mezzo per ricavare le soluzioni precedenti. Tuttavia ciò va effettuato per trovare le ampiezze A_{3j} , delineando quindi in modo univoco il vettore spostamento, e si attua applicando le condizioni al contorno.

Ad inizio paragrafo si è detto che verrà considerato un mezzo semi-infinito. Avendo posto il sistema di riferimento in maniera tale che l'origine fosse sulla superficie libera e l'asse x_3 diretto verso il mezzo, quindi con valori positivi all'interno di esso, è necessario fare delle considerazioni sui valori che possono assumere le α_j . Infatti con la scelta fatta la variabile x_3 all'interno del mezzo assume valori positivi via via crescenti allontanandosi dalla superficie. Essendo stata la dipendenza da essa degli spostamenti definita come $\exp(-\alpha_j x_3)$, quando le α_j hanno parte reale negativa tali esponenziali divergono per $x_3 \rightarrow +\infty$, facendo divergere gli spostamenti. Ovviamente in questo caso non si avrebbe una soluzione fisica accettabile e perciò tali radici dell'equazione secolare vanno scartate. D'altra parte si è sottolineato

precedentemente che tali radici possono essere solamente reali o complesse coniugate. In particolare anche quando esse risultassero reali, nei casi presi in considerazione (isotropo, cubico, esagonale), esse sono a coppie, una positiva e una negativa con modulo uguale, poichè l'equazione secolare è del tipo $a\alpha^6 + b\alpha^4 + c\alpha^2 + d = 0$ quando k_1 e k_2 sono entrambi diversi da zero, oppure del tipo $a\alpha^4 + b\alpha^2 + c = 0$ quando k_1 o k_2 sono nulli. Ne consegue quindi che delle sei radici dell'Eq.(3.6) tre hanno parte reale positiva e tre negativa. Ci sarebbe anche il caso in cui la parte reale di una coppia o più di esse sia nulla, ma tale situazione, come si vedrà più avanti, non si verifica per i casi di interesse. Scartando quelle con parte reale negativa, la soluzione generale degli spostamenti per un mezzo semi-infinito si ridimensiona come segue:

$$\bar{\mathbf{u}} = \begin{cases} u_1 = \sum_{j=1}^3 A_{3j}\eta_{1j}\exp[i(k_1x_1 + k_2x_2 - \omega t)]\exp(-\alpha_jx_3) \\ u_2 = \sum_{j=1}^3 A_{3j}\eta_{2j}\exp[i(k_1x_1 + k_2x_2 - \omega t)]\exp(-\alpha_jx_3) \\ u_3 = \sum_{j=1}^3 A_{3j}\exp[i(k_1x_1 + k_2x_2 - \omega t)]\exp(-\alpha_jx_3) \end{cases} \quad (3.9)$$

dove anche il numero delle ampiezze A_{3j} si è ridotto a 3. A questo punto vanno applicate le condizioni al contorno.

La condizione di superficie libera implica che gli sforzi su di essa si annullino. Per come è stato configurato il problema tale superficie risulta normale all'asse x_3 , perciò le componenti del tensore delle costanti elastiche che vanno annullate sono σ_{i3} con $i = 1, 2, 3$ poichè il secondo indice indica la normale al piano su cui agiscono gli sforzi, mentre il primo ne indica la componente. In notazione matriciale esse diventano:

$$\sigma_{13} \longrightarrow \sigma_5, \quad \sigma_{23} \longrightarrow \sigma_4, \quad \sigma_{33} \longrightarrow \sigma_3$$

e il loro legame con gli spostamenti u_i è il seguente

$$\begin{aligned}
\sigma_5 &= C_{51} \frac{\delta u_1}{\delta x_1} + C_{52} \frac{\delta u_2}{\delta x_2} + C_{53} \frac{\delta u_3}{\delta x_3} + C_{54} \left(\frac{\delta u_2}{\delta x_3} + \frac{\delta u_3}{\delta x_2} \right) + C_{55} \left(\frac{\delta u_1}{\delta x_3} + \frac{\delta u_3}{\delta x_1} \right) \\
&\quad + C_{56} \left(\frac{\delta u_1}{\delta x_2} + \frac{\delta u_2}{\delta x_1} \right); \\
\sigma_4 &= C_{41} \frac{\delta u_1}{\delta x_1} + C_{42} \frac{\delta u_2}{\delta x_2} + C_{43} \frac{\delta u_3}{\delta x_3} + C_{44} \left(\frac{\delta u_2}{\delta x_3} + \frac{\delta u_3}{\delta x_2} \right) + C_{45} \left(\frac{\delta u_1}{\delta x_3} + \frac{\delta u_3}{\delta x_1} \right) \\
&\quad + C_{46} \left(\frac{\delta u_1}{\delta x_2} + \frac{\delta u_2}{\delta x_1} \right); \\
\sigma_3 &= C_{31} \frac{\delta u_1}{\delta x_1} + C_{32} \frac{\delta u_2}{\delta x_2} + C_{33} \frac{\delta u_3}{\delta x_3} + C_{34} \left(\frac{\delta u_2}{\delta x_3} + \frac{\delta u_3}{\delta x_2} \right) + C_{35} \left(\frac{\delta u_1}{\delta x_3} + \frac{\delta u_3}{\delta x_1} \right) \\
&\quad + C_{36} \left(\frac{\delta u_1}{\delta x_2} + \frac{\delta u_2}{\delta x_1} \right);
\end{aligned} \tag{3.10}$$

Il set di condizioni al contorno è quindi:

$$\begin{cases} \sigma_5(x_3 = 0) = 0 \\ \sigma_4(x_3 = 0) = 0 \\ \sigma_3(x_3 = 0) = 0 \end{cases} \tag{3.11}$$

Sostituendovi le espressioni (3.10) e poi quelle degli spostamenti (3.9) si ottiene un sistema omogeneo di tre equazioni lineari nelle tre incognite A_{3j} . Tale sistema contiene come parametro la pulsazione ω , poichè le α_j e i rapporti η_{1j} e η_{2j} dipendevano da essa. Nuovamente, affinchè il sistema abbia soluzione diversa da quella banale $[A_{3j}] = 0$, bisogna che il determinante della matrice associata ad essa sia nullo. Chiamando quindi tale matrice B , ponendo

$$\det(B) = 0 \tag{3.12}$$

si ottiene un'equazione nell'incognità w . Le soluzioni reali, se esistono, corrispondono alle pulsazioni delle onde di superficie cercate. Tuttavia in generale non si riesce a ricavarle analiticamente e bisogna utilizzare metodi numerici. A questo punto per trovare le ampiezze bisogna ripertere il procedimento già adottato per il sistema (3.5): sostituire la pulsazione trovata come soluzione ω_{sol} nella matrice B ottenendo una matrice numerica, prendere una delle ampiezze A_{3j} come riferimento e trovare le altre in funzione di essa. Poi sostituirla (la pulsazione) nelle espressioni delle α_j e dei rapporti η_{1j} e η_{2j} e sostituire il tutto nelle (3.9) determinando così gli spostamenti u_1 , u_2 e u_3 per l'onda di superficie con pulsazione w_{sol} , tutti definiti a

meno di una stessa costante di proporzionalità, che è l'ampiezza A_{3j} presa come riferimento.

3.3.2 Sistemi multistrati

Il procedimento visto nella sezione precedente vale anche quando il mezzo abbia una dimensione finita. In tal caso bisogna considerare tutte e sei le radici α_j dell'equazione secolare e quindi utilizzare gli spostamenti nella forma (3.8). Le incognite da trovare sono le 6 ampiezze A_j e le 6 equazioni necessarie si ottengono sempre applicando le condizioni al contorno. Infatti ora le superfici libere sono due: una che delimita il mezzo superiormente (quella che esiste anche nel caso semi-infinito) ed una che lo delimita inferiormente, entrambe parallele fra di loro e perpendicolari all'asse x_3 . Se indichiamo con x_{3max} la posizione della superficie libera inferiore rispetto all'asse x_3 , il sistema delle condizioni al contorno diventa il seguente:

$$\left\{ \begin{array}{l} \sigma_5(x_3 = 0) = 0 \\ \sigma_4(x_3 = 0) = 0 \\ \sigma_3(x_3 = 0) = 0 \\ \sigma_5(x_3 = x_{3max}) = 0 \\ \sigma_4(x_3 = x_{3max}) = 0 \\ \sigma_3(x_3 = x_{3max}) = 0 \end{array} \right. \quad (3.13)$$

Si può facilmente estendere al caso di un sistema multistrati. In questa situazione bisogna eseguire la procedura di calcolo delle α_j e dei rapporti η_{1j} e η_{2j} fino ad ottenere gli spostamenti in forma di combinazione lineare dati dalle (3.8) per ogni strato, nello stesso modo visto prima. Si hanno quindi 6 ampiezze A_j incognite per ogni strato. Se L è il numero totale di strati (layer), allora il numero complessivo di ampiezze incognite è $6L$. Come condizioni al contorno si hanno ancora quelle di superficie libera sulle superfici superiore ed inferiore del sistema multistrato considerato, in più bisogna imporre la continuità degli sforzi e degli spostamenti all'interfaccia tra due strati. Quindi si ottengono 3 condizioni per gli sforzi ($\sigma_5, \sigma_4, \sigma_3$) e 3 per gli spostamenti (u_1, u_2, u_3) per ogni interfaccia. Per ogni configurazione a L

strati ci sono $L - 1$ interfacce, quindi si ottengono $6(L - 1)$ equazioni. Riassumendo abbiamo: 3 equazioni per la superficie libera superiore, 3 per quella inferiore e $6(L - 1)$ per le interfacce, quindi in totale $3 + 3 + 6(L - 1) = 3 + 3 + 6L - 6 = 6L$ equazioni. Perciò si è ottenuto un set completo di equazioni omogenee. Se l è un indice numerico che identifica lo strato, x_{3min_l} e x_{3max_l} sono le coordinate che delimitano lo strato l -esimo rispetto all'asse x_3 , u_{1_l} , u_{2_l} e u_{3_l} sono gli spostamenti di quest'ultimo nella forma data dalle (3.8) e σ_{5_l} , σ_{4_l} e σ_{3_l} sono gli sforzi su un piano perpendicolare all'asse x_3 ad essi associati, il sistema può essere espresso come segue:

$$\left\{ \begin{array}{l} \sigma_{5_1}(x_3 = x_{3min_1}) = 0 \\ \sigma_{4_1}(x_3 = x_{3min_1}) = 0 \\ \sigma_{3_1}(x_3 = x_{3min_1}) = 0 \\ \sigma_{5_l}(x_3 = x_{3max_l}) = \sigma_{5_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ \sigma_{4_l}(x_3 = x_{3max_l}) = \sigma_{4_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ \sigma_{3_l}(x_3 = x_{3max_l}) = \sigma_{3_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ u_{1_l}(x_3 = x_{3max_l}) = u_{1_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ u_{2_l}(x_3 = x_{3max_l}) = u_{2_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ u_{3_l}(x_3 = x_{3max_l}) = u_{3_{(l+1)}}(x_3 = x_{3min_{(l+1)}}) \\ \sigma_{5_L}(x_3 = x_{3max_L}) = 0 \\ \sigma_{4_L}(x_3 = x_{3max_L}) = 0 \\ \sigma_{3_L}(x_3 = x_{3max_L}) = 0 \end{array} \right. \quad l = 1, \dots, L - 1; \quad (3.14)$$

Tale sistema è omogeneo nelle incognite A_{j_l} . Il procedimento è del tutto identico a quello trattato nel caso del mezzo semi-infinito. Perchè abbia soluzione diversa dalla soluzione nulla occorre che il determinante della matrice associata ad esso sia nullo. Si ottiene un'equazione dove l'incognita è la pulsazione ω ; le soluzioni reali, se esistono, corrispondono alle pulsazioni delle onde elastiche che soddisfano le condizioni al contorno. Si risostituisce la soluzione ω_{sol} all'interno del sistema (3.14), si prende una delle $6L$ ampiezze A_{j_l} come riferimento e si ricavano le altre in funzione di essa. Infine si risostituisce il tutto nelle espressioni degli spostamenti

trovando quindi gli andamenti degli stessi all'interno di ogni singolo strato, tutti definiti a meno di una medesima costante di riferimento.

3.3.3 Alcune considerazioni

Il set di equazioni (3.14) relativo alla configurazione multistrati, può essere applicato con piccole modifiche anche nel caso in cui l'ultimo strato, il substrato, sia semi-infinito. Infatti basta scartare le tre α_j negative del substrato, in quanto, come detto, causano il divergere degli spostamenti, per il solo substrato ed eliminare le ultime tre equazioni inerenti alla condizione di superficie libera sulla superficie inferiore, in quanto essa in questo caso non esiste. Si ottiene così un sistema $(6L - 3) \times (6L - 3)$ e poi si procede nello stesso modo.

Altri due casi da sottolineare sono quelli, speculari, in cui k_1 o k_2 sono nulli, cioè il vettore d'onda \mathbf{k} è rispettivamente parallelo all'asse x_1 o all'asse x_2 . Prendiamo la situazione in cui sia $k_2 = 0$, cioè si pone il vettore d'onda lungo l'asse x_1 . Così facendo, se tale direzione è di sufficiente simmetria, si ottiene che nelle equazioni di equilibrio dinamico (3.2) le due relative ad u_1 e u_3 si disaccoppiano da quella relativa ad u_2 , ovvero si separano i modi nel piano sagittale, in questo caso il piano (x_1, x_3) , da quelli perpendicolari ad esso. Il nostro studio tratterà solamente quelli sagittali, poichè sono quelli di maggiore interesse. Il sistema si riduce a due equazioni nelle due incognite A_1 e A_3 e l'equazione secolare diventa di quarto grado. Quindi si hanno 4 radici invece che 6. Per il resto il procedimento rimane lo stesso. Condizione sufficiente per il disaccoppiamento delle equazioni è che il piano sagittale sia piano di simmetria speculare, requisito verificato per le simmetrie prese in esame. Il caso con $k_1 = 0$ è speculare, poichè in tale situazione sono le equazioni inerenti ad u_2 e u_3 a disaccoppiarsi da quella relativa ad u_1 . Si ottiene sempre un sistema di equazioni 2×2 dove stavolta le incognite sono A_2 e A_3 e l'equazione secolare è ancora di quarto grado.

Capitolo 4

Il programma

4.1 Il processo di calcolo

In questo capitolo viene presentato il programma creato in linguaggio MATLAB per trovare le onde elastiche, basato sulla teoria vista nei precedenti capitoli. In particolare segue il procedimento esposto nella sezione dedicata alle onde di superficie, mettendosi nel caso generico del sistema multistrato. Si è visto che tale procedura porta alla composizione della matrice delle condizioni al contorno, dalla quale si calcola il determinante, che è funzione della pulsazione ω , ponendolo poi uguale a zero. Le radici reali di tale equazione sono le pulsazioni delle onde elastiche che soddisfano le condizioni al contorno. Per trovare poi le ampiezze di tali onde, bisogna sostituire la pulsazione corrispondente nella matrice e da questa ricavarle prendendone una come riferimento. Come già indicato nel capitolo precedente, le ampiezze delle diverse componenti vengono determinate a meno di una ampiezza complessiva arbitraria. Tuttavia, come già evidenziato, il determinante della matrice delle condizioni al contorno è in generale una funzione molto complessa della pulsazione ω per cui il calcolo analitico degli zeri è impraticabile. Ciò ha portato alla necessità di ricorrere a metodi numerici ed è per questo motivo che il programma è composto da due parti: la prima è uno script chiamato *'Tesi'* che porta alla composizione della matrice delle condizioni al contorno; la seconda, chiamata *'tesi2'*, è scritta in formato *'function'* e ha come uscita gli spostamenti u_1 , u_2 e u_3 per una specifica onda elastica. Interposti tra le due parti ci sono dei programmi

accessori: il file chiamato *'printdetlog'* serve per plottare in un grafico il modulo del determinante della matrice delle condizioni al contorno in funzione della pulsazione ω in scala semi-logaritmica; il file *'trovaminimo'* serve per trovare il valore di ω che minimizza tale funzione all'interno di un intervallo imposto. Nello schema a blocchi in figura 4.1 viene rappresentato il procedimento logico-concettuale con cui è stato costruito il programma. La prima parte del programma prende in ingresso i dati

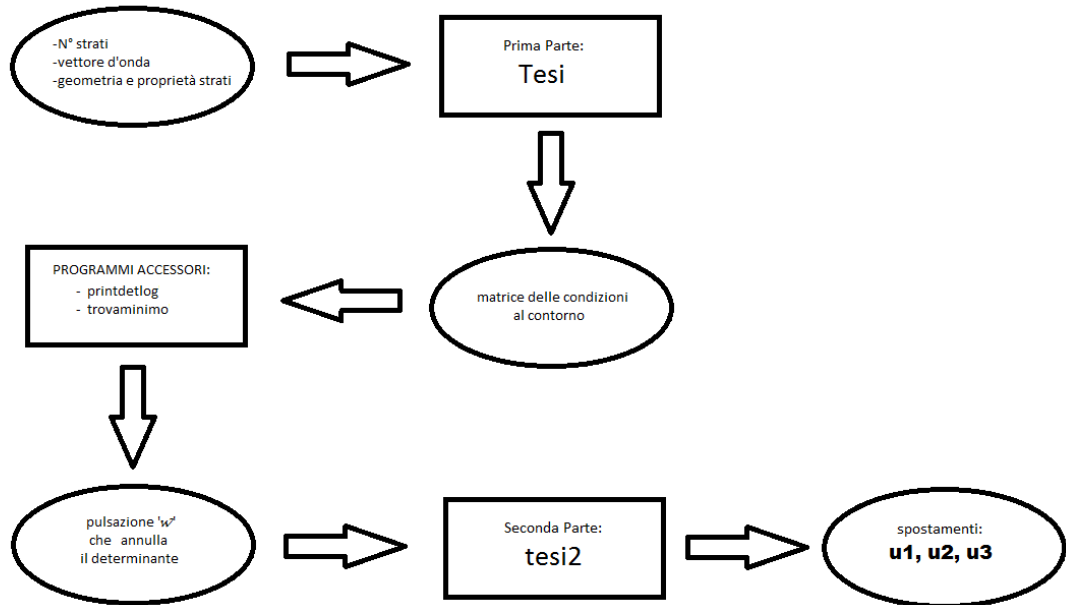


Fig. 4.1: Schema logico del programma

relativi al problema e da essa si ottengono due matrici delle condizioni al contorno: una relativa al caso in cui il substrato è un mezzo finito ed una al caso di in cui è semi-infinito. Queste (una o l'altra) sono usate attraverso i programmi accessori per ricavare la pulsazione ω dell'onda elastica, la quale a sua volta è utilizzata dalla seconda parte del programma per calcolare gli spostamenti u_1 , u_2 e u_3 .

Il programma è stato creato per trattare sistemi multistrati e non esiste alcun tipo di vincolo sul numero degli strati; il sistema monolayer (singolo strato) è incluso come caso particolare. Il sistema di riferimento alla base del processo di calcolo è definito in modo che: gli assi x_1 e x_2 giacciono su una superficie parallela a quelle che delimitano gli strati; l'asse x_3 è di conseguenza perpendicolare ad esse ed è quello che descrive le proprietà delle onde con la profondità. Tuttavia il verso dell'asse x_3 ,

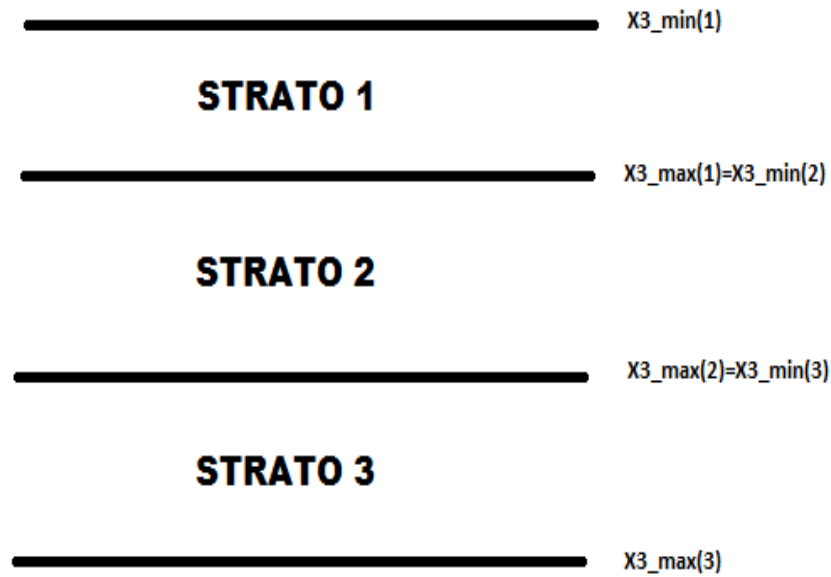


Fig. 4.2: Convenzione assunta sulla numerazione degli strati e sulla loro delimitazione

così come la scelta sulla posizione dell'origine degli assi sono arbitrio dell'utente. Gli strati sono numerati in maniera crescente dalla superficie esterna ($x_3(1)$) verso la profondità del substrato, come in figura 4.2. Sono prese in considerazione tre tipi di simmetria: *isotropa*, *cubica* ed *esagonale*. Per ognuna di esse il calcolo è differenziato in tre casi in base alla direzione del vettore d'onda nel piano (x_1, x_2) : due, simili, sono più semplici e specifici e corrispondono rispettivamente a $k_1 = 0$ e $k_2 = 0$, il terzo è quello più generale con $k_1 \neq 0$ e $k_2 \neq 0$. I primi due sono più semplici poichè in tali condizioni, per le simmetrie considerate, avviene il disaccoppiamento dei modi sagittali da quelli nella direzione trasversale al piano sagittale, con conseguente notevole semplificazione delle equazioni. Esistono già molti programmi in grado di calcolare le onde elastiche in queste situazioni ed anche presso il nostro Dipartimento ne sono disponibili. Non sono invece disponibili presso il nostro Dipartimento programmi che considerano il caso generale, più complesso, in cui non c'è disaccoppiamento. Ciò non è un problema nello studio di sistemi isotropi ed esagonali, in quanto essi hanno come propria caratteristica l'isotropia nel piano (x_1, x_2) , ovvero in tale piano le onde elastiche si propagano nello stesso modo per qualsiasi direzione. Diventa invece di cruciale importanza nel trattare la

simmetria cubica, poichè in tale situazione il disaccoppiamento dei modi avviene solo in pochissime direzioni e un software che tratta solo questo caso risulta perciò assai limitato.

Una volta avviato, lo script `Tesi.m` chiede per prima cosa il numero di strati complessivo del problema e le componenti del vettore d'onda k_1 e k_2 . Ricevuti i primi input si avvia un procedimento identico per ogni strato. Vengono chiesti i valori di x_3 che delimitano superiormente ($x_{3_{min}}$) ed inferiormente ($x_{3_{max}}$) lo strato se è il primo, mentre soltanto il limite inferiore per gli altri poichè quello superiore viene automaticamente impostato uguale a quello inferiore dello strato precedente. Si definiscono gli spostamenti secondo le 3.4 ed il vettore deformazione in notazione matriciale. Poi viene chiesto in quale tipo di geometria ci si vuole mettere. Per ogni strato può essere scelta una qualsiasi delle tre simmetrie considerate, quindi si può fare qualsiasi tipo di combinazione. Il programma chiede i valori delle corrispondenti costanti elastiche indipendenti: C_{11} e C_{12} per l'isotropo, C_{11} , C_{12} e C_{44} per il cubico, C_{11} , C_{12} , C_{13} , C_{33} e C_{44} per l'esagonale. La matrice viene composta in automatico ed esiste anche un controllo sui vincoli termodinamici; nel caso in cui non fossero rispettati esce un messaggio di errore ed il programma si interrompe. Successivamente si calcola il vettore degli sforzi usando la 2.7 e si compongono le tre equazioni delle onde elastiche 3.2. Nei casi in cui avviene il disaccoppiamento dei modi, il sistema 3.5 diventa di due equazioni in due incognite e quindi si ha un solo rapporto η_1 , mentre le α_j ricavate dall'equazione secolare sono 4. Il caso generale è quello generico spiegato nel paragrafo dedicato alle onde di superficie: le radici dell'equazione secolare (α_j) sono 6 e ci sono due rapporti η_1 e η_2 . Questi ultimi dipendono da quale ampiezza è presa come riferimento, cioè se A_1 , A_2 o A_3 , e sono funzioni delle α_j . Fissata l'ampiezza di riferimento, i rapporti η_1 e η_2 sono validi sempre tranne quando l'ampiezza scelta si annulla in corrispondenza di una o più α_j , causando il divergere dei suddetti rapporti. Il programma ha in sé una parte di codice che verifica il non divergere di essi. Ciò viene fatto considerando i denominatori dei rapporti, sostituendoci a turno tutte le α_j trovate e verificando che non si annullino. Nel qual caso i rapporti vengono modificati prendendo un'altra ampiezza come riferimento; poi viene nuovamente effettuato

il processo di verifica sulla non divergenza. Definita così la forma generale degli spostamenti come combinazione lineare di sei contributi secondo le 3.8, si calcolano le grandezze necessarie per comporre la matrice delle condizioni al contorno cioè: $\sigma_5, \sigma_4, \sigma_3, u_1, u_2$ e u_3 calcolate in $x_{3_{min}}$ e $x_{3_{max}}$. Queste sono poste in una matrice dove l'indice della colonna indica il numero dello strato. Ciò è fatto per ogni strato. Alla fine si compone la matrice delle condizioni al contorno dove bisogna considerare tutti i casi possibili: $k_1 = 0, k_2 = 0, k_1 \neq 0$ e $k_2 \neq 0$ e per ognuno di essi, per ogni strato, se si è stato preso come riferimento A_1 o A_2 o A_3 . Tale matrice è inerente al caso di substrato finito. Quella relativa al caso di substrato semi-infinito è ottenuto eliminando le colonne corrispondenti alle α_j del substrato con parte reale negativa e le righe corrispondenti alle condizioni di superficie libera della superficie inferiore di esso.

Alla fine della procedura di calcolo della Prima Parte del programma si ottengono quindi le matrici citate e per poter utilizzare le parti successive bisogna salvare il 'workspace' ottenuto. È importante sottolineare che esse hanno come parametro la pulsazione ω . Quindi una volta definito il problema con tutti i suoi dati, i calcoli per arrivare ad ottenere la matrice delle condizioni al contorno (sia per il mezzo finito che semi-infinito) vengono effettuati una sola volta. Poi, tramite l'utilizzo del programma accessorio '*printdetlog*', si può separatamente ottenere l'andamento del determinante rispetto alla pulsazione ω , con la possibilità di fissare a piacimento l'intervallo di interesse per quest'ultima e la risoluzione. Si può quindi procedere iniziando con un intervallo ampio (non sapendo dove si trovano le pulsazioni delle onde elastiche cercate) ed una bassa risoluzione per contenere i tempi di calcolo. Identificate le pulsazioni delle onde, si può procedere restringendo l'intervallo ed aumentando la risoluzione per uno studio più dettagliato. Successivamente con l'utilizzo del secondo programma accessorio, cioè '*trovaminimo*', si ottiene il valore esatto della pulsazione dell'onda elastica, a meno di una incertezza fissabile, all'interno dell'intervallo definito. Per il corretto funzionamento del programma, l'intervallo deve essere ristretto in maniera tale che al suo interno ci sia la pulsazione di una sola onda elastica. Trovato il valore di quest'ultima, esso viene utilizzato dalla Seconda Parte del programma, il file '*tesi2.m*', insieme al 'workspace' ottenuto

dalla Prima Parte, per ottenere gli spostamenti u_1 , u_2 e u_3 dell'onda elastica.

4.2 Procedura di investigazione

Il programma è stato creato per poter trattare le simmetrie isotropa, cubica ed esagonale sia singolarmente, sia combinate attraverso una configurazione multistrato, dove ogni strato presenta una propria geometria tra le precedenti. Quindi è possibile studiare le caratteristiche dei modi elastici propri di ognuna delle suddette simmetrie e successivamente vedere come cambiano nelle configurazioni multistrato, per esempio nella più semplice costituita da un film depositato sopra un substrato. Le possibilità di investigazione sono molteplici ed in questa tesi ne verranno presentate alcune. Inizialmente cominceremo con lo studio delle radici α_j dell'equazione secolare al variare della pulsazione ω per le tre simmetrie. Successivamente tratteremo i sistemi bi-layer, cioè a due strati, in particolare costituiti da un substrato su cui è posto un film sottile. Lo scopo è quello di investigare gli effetti che comporta la presenza di un film a simmetria esagonale sopra un substrato sulle caratteristiche delle onde di superficie (una trattazione approfondita sulle onde elastiche in film supportati si trova in [1] e [20]). Per far ciò si è scelto un substrato isotropo per eliminare ogni tipo di influenza dovuta alla presenza di anisotropia in quest'ultimo. Lo studio prende come riferimento la situazione in cui il film presenta proprietà identiche al substrato. Poi vengono variate le costanti elastiche del film in maniera tale da introdurre una anisotropia inizialmente leggera e renderla poi via via crescente, andando a studiare come cambiano i modi durante questo processo.

Il materiale preso come riferimento è il silicio, poichè esso è molto usato nelle attività di ricerca ed anche in molti processi industriali. Infatti essendo un semiconduttore intrinseco è usato in molte apparecchiature elettroniche, nei transistor ed anche nei pannelli solari o celle solari. Il silicio è altresì il tipico materiale con cui sono costruiti i MEMS. Esso presenta simmetria cubica, la quale, come visto, ha tre costanti indipendenti, che noi identifichiamo in C_{11} , C_{12} e C_{44} e per il silicio valgono: $C_{11} = 166 \text{ GPa}$, $C_{12} = 64 \text{ GPa}$ e $C_{44} = 79.6 \text{ GPa}$; la densità invece vale $\rho = 2329 \frac{\text{kg}}{\text{m}^3}$. Come detto sopra, ai fini del nostro studio è utile prendere come

riferimento un materiale isotropo. Vogliamo pertanto definire un materiale fittizio isotropo che abbia proprietà analoghe al silicio. Nella simmetria isotropa le costanti indipendenti sono due; rispetto al caso cubico la terza costante è ottenuta attraverso una precisa relazione con le altre due. La condizione che le lega è

$$C_{11} = C_{12} + 2C_{44}.$$

Per ottenere quelle di questo materiale fittizio isotropo, che chiameremo "simil-silicio", si può procedere in più modi: il primo è diminuire C_{12} , in modo che venga rispettata la precedente condizione, fino ad un valore pari a $C_{12} = 5.9 \text{ GPa}$. Tuttavia ciò avrebbe prodotto un rapporto di Poisson troppo basso, pari a circa 0,03, riscontrabile solo in materiali molto particolari e lontano dai valori tipici dei materiali più utilizzati. Quindi si è scelto di alzare C_{11} , mantenendo fisse C_{12} e C_{44} . In tal modo si arriva ad avere $C_{11} = 223.2 \text{ GPa}$, con un rapporto di Poisson pari a 0,22, un valore dello stesso ordine di quelli dei materiali ceramici.

Vengono quindi presentati alcuni risultati inerenti la simmetria cubica, in particolare viene testata la parte di codice dedicata al caso generale con $k_1 \neq 0$ e $k_2 \neq 0$. In questo caso verrà messa in evidenza l'anisotropia nel piano (x_1, x_2) . Come detto sopra, prendendo come riferimento questo "simil-silicio" isotropo, sono stati poi definiti materiali di anisotropia crescente, secondo le simmetrie esagonale e cubica.

4.3 Verifiche sul Programma

Prima di iniziare i calcoli si sono operate alcune prove sul programma per testarne il corretto funzionamento:

1. verifica dell'isotropia nel piano (x_1, x_2) per le simmetrie isotropa ed esagonale;
2. verifica dell'equivalenza tra un monolayer ed un bilayer dove i due strati hanno le stesse proprietà e lo spessore totale è uguale a quello del monolayer;
3. verifica del cambiamento in numero delle onde di slab al variare dello spessore;

1) La prima verifica è stata quella sull'isotropia nel piano x_1x_2 per i casi di un monolayer isotropo ed esagonale. Si è scelto per entrambi i casi lo spessore di $5 \mu m$ e densità pari a $2329 \frac{kg}{m^3}$, come quella del silicio. Tuttavia nell'operare la verifica si è voluto testare il programma in un campo più ampio, riguardo alle costanti elastiche, rispetto ai valori esposti nel paragrafo precedente, che saranno poi quelli utilizzati nelle indagini.

Per il caso isotropo si sono scelti i seguenti valori delle costanti elastiche indipendenti: $C_{11} = 261 GPa$ e $C_{12} = 151 GPa$. Come si vede, rispetto al "simil-silicio" risultano maggiori entrambe le costanti elastiche indipendenti, quindi il materiale risulterà più veloce. Tuttavia il rapporto di Poisson è pari a 0,31, quindi nel range della maggior parte dei materiali metallici.

Per la simmetria esagonale si è fatta la seguente scelta: rispetto al caso cubico ha 2 costanti elastiche indipendenti in più, in totale 5; C_{12} e C_{44} sono state mantenute come nel silicio, C_{11} è stata posta pari a $200 GPa$, un valore intermedio tra quello del silicio cubico e quello del 'simil-silicio' isotropo, quindi il materiale in questione è leggermente più lento; C_{33} , che nei casi isotropo e cubico è uguale a C_{11} , è stata diminuita a $150 GPa$; infine bisognava modificare C_{13} rispetto a C_{12} per ottenere una simmetria esagonale ben definita e si è scelto di aumentarla per compensare la diminuzione di quelle viste precedentemente, portandola a $C_{13} = 75 GPa$. Quindi riassumendo per le verifiche nella simmetria esagonale sono state usate le seguenti costanti elastiche indipendenti: $C_{11} = 200 GPa$, $C_{12} = 64 GPa$, $C_{13} = 75 GPa$, $C_{33} = 150 GPa$ e $C_{44} = 79.6 GPa$. Si è operato prendendo il seguente valore del modulo del vettore d'onda: $|\mathbf{k}| = 2 * \frac{2\pi}{514.5 * 10^{-9}} * \sin(30^\circ) m^{-1}$, mantenuto costante, variandone le componenti (k_1, k_2) , cioè l'angolo nel piano x_1x_2 , prendendo come angolo di riferimento quello rispetto all'asse x_1 . Si sono operati i calcoli con la Prima Parte del programma, poi si è utilizzato il programma accessorio 'printdetlog' per ottenere il grafico del determinante della matrice delle condizioni al contorno in funzione della pulsazione ω , per il caso di mezzo finito. Per entrambi i casi sono state operate le prove a 0, 30, 45 e 60 gradi. Nel mezzo isotropo, come si può notare in Figura 4.3, per tutte le prove si è ottenuta lo stesso valore della pulsazione ω dell'onda di superficie a circa $55 \frac{Grad}{s}$, identificabile nei grafici dal

piccolo avvallamento in corrispondenza di tale valore, dove la profondità di esso è data dalla risoluzione con cui i grafici sono costruiti. Si nota anche che a partire da circa $60 \frac{\text{Grad}}{\text{s}}$ sono presenti degli avvallamenti regolari: essi corrispondono alle onde di slab ed infatti compaiono solamente dopo la pulsazione corrispondente alla frequenza dell'onda trasversa (soglia trasversa) pari a $59.35 \frac{\text{Grad}}{\text{s}}$. Anche nel caso

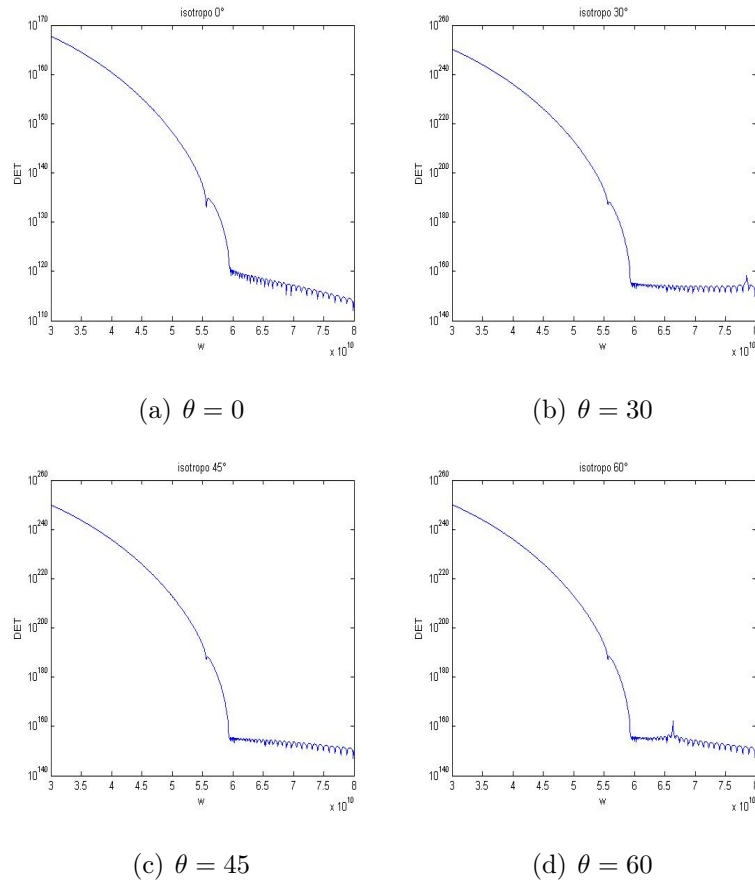


Fig. 4.3: Verifica isotropia nel piano: caso isotropo.

esagonale risulta soddisfatta l'isotropia nel piano x_1x_2 , come si vede in Figura 4.4 . In questo caso la pulsazione dell'onda di superficie si trova a circa $61 \frac{\text{Grad}}{\text{s}}$. Pure per il caso esagonale si osserva l'apparire delle onde di slab oltre la soglia trasversa che cade a circa $71 \frac{\text{Grad}}{\text{s}}$. Bisogna sottolineare però la presenza del breve tratto tra $69.5 \frac{\text{Grad}}{\text{s}}$ e $71 \frac{\text{Grad}}{\text{s}}$, non presente per l'isotropo; le variazioni del determinante in quella zona hanno solo il significato di 'rumore'. Questa zona è una caratteristica propria della simmetria esagonale, come vedremo più avanti.

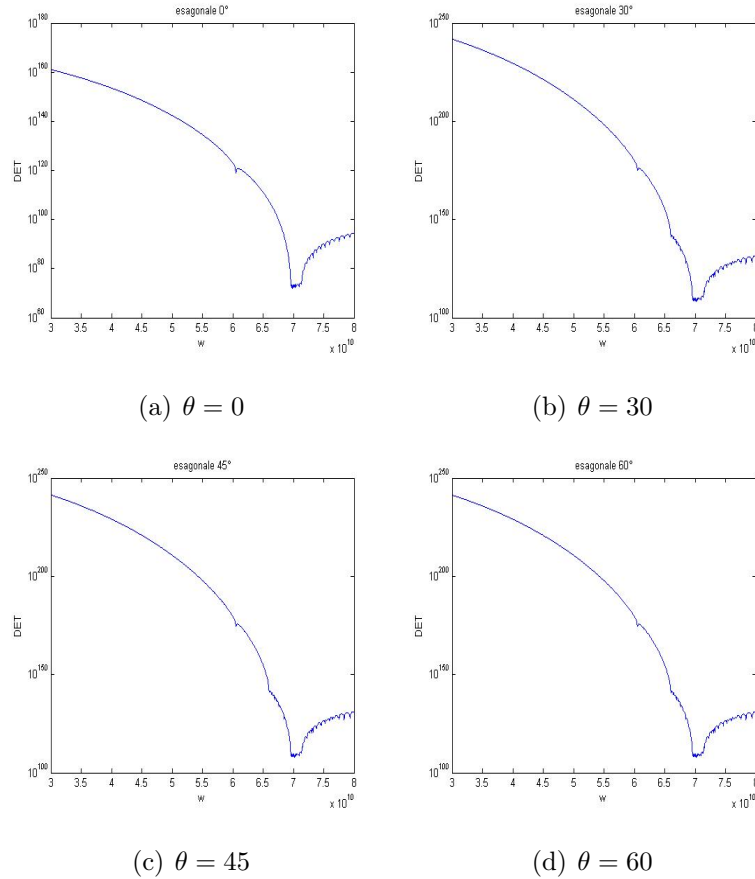


Fig. 4.4: Verifica isotropia nel piano: caso esagonale.

2) Un'altra verifica effettuata è stata quella di testare la concordanza di risultati tra monolayer e bilayer equivalente. Infatti se si prende un monolayer di spessore \mathbf{S} con proprietà definite, esso risulta del tutto equivalente a due strati perfettamente adiacenti con le medesime proprietà la cui somma degli spessori risulta pari a $S_1 + S_2 = \mathbf{S}$. Infatti in tale situazione la superficie di separazione tra i due strati risulta del tutto fittizia. Ciò è stato fatto sia per il caso isotropo sia per il caso esagonale come riportato nelle Figure 4.5 e 4.6.

3) Infine l'ultima verifica riguarda le onde di slab; come è noto il loro numero cambia al variare dello spessore, in particolare cresce al crescere di esso. Come materiale è stato usato il nickel, avente simmetria cubica, sia per testare il programma con valori diversi da quelli già usati, sia perchè il nickel è un materiale già molto studiato e si trovano in letteratura molti risultati riguardo ad esso, per esempio in [10] e [18]. Esso presenta le seguenti proprietà: $C_{11} = 261 \text{ GPa}$, $C_{12} = 151 \text{ GPa}$

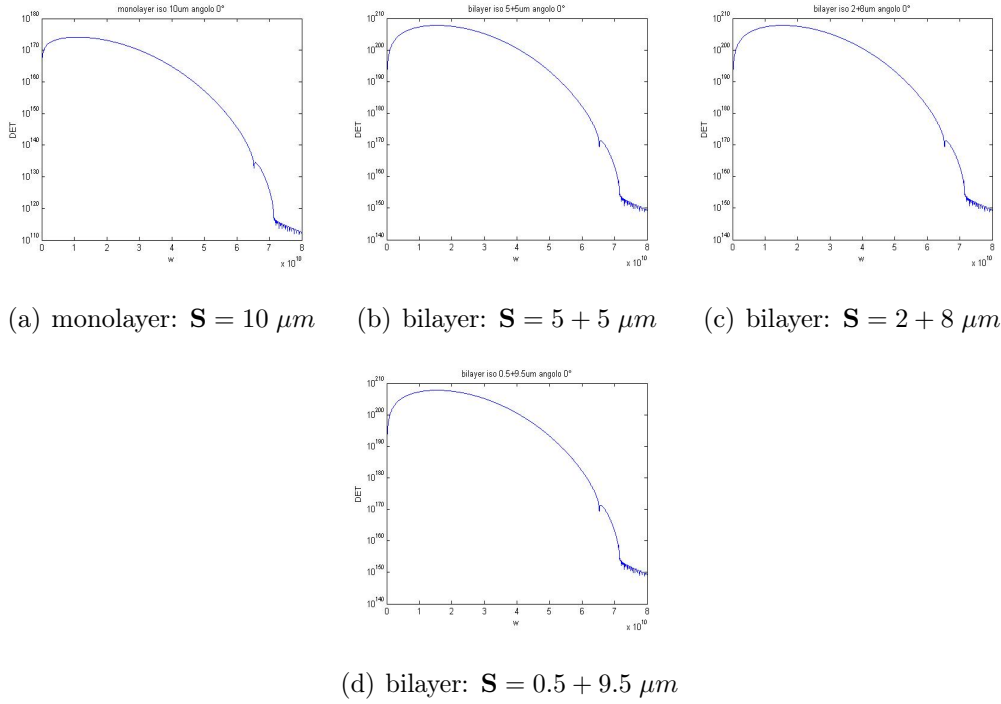


Fig. 4.5: Verifica monolayer-bilayer equivalenti: caso isotropo.

e $C_{44} = 130.9 \text{ GPa}$ e densità pari a $\rho = 8908 \frac{\text{kg}}{\text{m}^3}$. In Figura 4.7 è chiaramente visibile come col crescere dello spessore le onde di slab, presenti oltre circa $40 \frac{\text{Grad}}{\text{s}}$, diventino sempre più fitte.

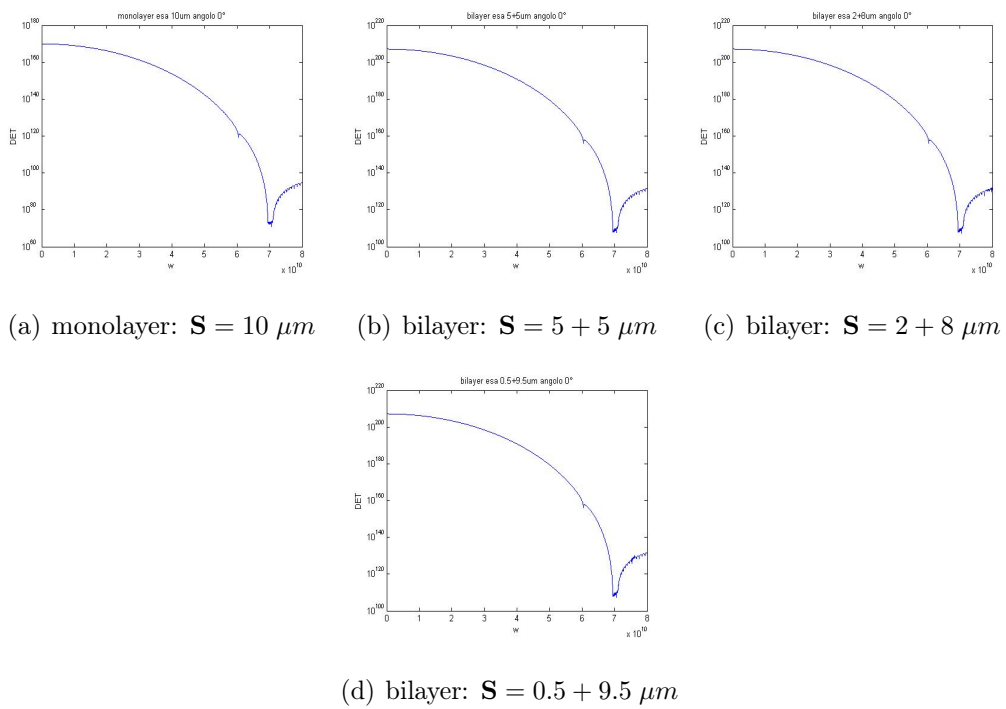


Fig. 4.6: Verifica monolayer-bilayer equivalenti: caso esagonale.

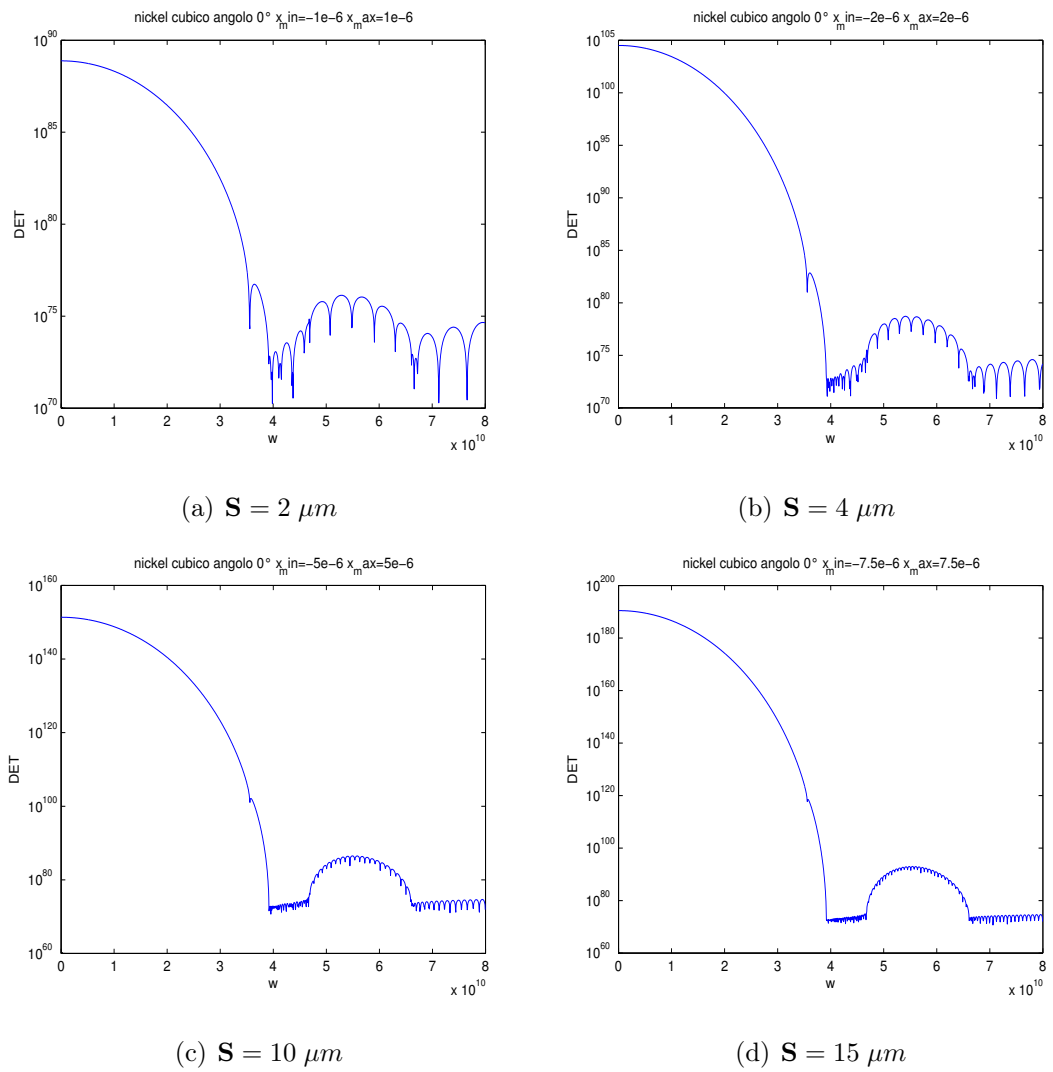


Fig. 4.7: Variazione in numero delle onde di slab per nickel cubico.

Capitolo 5

Risultati

5.1 Radici dell'equazione secolare

Una prima analisi resa possibile dal programma è stata quella di studiare la natura delle α_j al variare della pulsazione ω per le geometrie prese in esame. Infatti dalla Prima Parte del programma si ottengono le espressioni delle α_j come funzioni di ω . Esse dipendono unicamente dalle proprietà del mezzo (densità e costanti elastiche) e dal vettore d'onda \mathbf{k} , poichè sono calcolate dal sistema 3.5, prima che vengano imposte le condizioni al contorno. In tutte le prove effettuate si è scelto il vettore d'onda parallelo all'asse x_1 con modulo $|\mathbf{k}| = 2 * \frac{2\pi}{514.5*10^{-9}} * \sin(30^\circ) m^{-1}$.

Nello spettro delle pulsazioni ω , o se si preferisce delle frequenze ($\omega = 2\pi f$), ci sono quattro zone riguardanti i valori della quattro α_j inerenti ai modi sagittali. La prima si estende da zero fino alla prima soglia trasversa, cioè alla pulsazione corrispondente all'onda trasversa più lenta, ed è quella propria delle onde di superficie. La seconda zona è compresa tra le due soglie trasverse e non esiste per il mezzo isotropo, poichè per esso le due onde trasverse sono degeneri. La terza è compresa tra la soglia trasversa maggiore, corrispondente all'onda trasversa più veloce, e quella longitudinale, corrispondente all'onda longitudinale. Infine la quarta zona si estende oltre la soglia longitudinale. Ci aspettiamo comportamenti diversi delle α_j all'interno delle diverse zone e per le diverse simmetrie.

Per il caso *isotropo* si è usato come materiale il "simil-silicio", le cui proprietà sono state spiegate nel paragrafo 4.2. In Tabella 5.1 sono elencate le α_j (sono 4

Tabella 5.1: Andamento delle α_j in funzione della pulsazione ω : simmetria isotropa.

$x10^7$	Pulsazione $\omega \times 10^9$ (rad/s)				
	10	50	71	72	100
α_1	1,2169	1,1093	0,9825	0,9749 + 0,0000i	0,6693 + 0,0000i
α_2	1,2092	0,8717	0,1282	0,0000 + 0,1593i	0,0000 + 1,1977i
α_3	-1,2169	-1,1093	-0,9825	-0,9749 + 0,0000i	-0,6693 + 0,0000i
α_4	-1,2092	-0,8717	-0,1282	0,0000 - 0,1593i	0,0000 - 1,1977i

$x10^7$	Pulsazione $\omega \times 10^9$ (rad/s)			
	115	119	120	150
α_1	0,3338 + 0,0000i	0,1172 + 0,0000i	0,0000 + 0,1058i	0,0000 + 0,9254i
α_2	0,0000 + 1,5421i	0,0000 + 1,6285i	0,0000 + 1,6498i	0,0000 + 2,2565i
α_3	-0,3338 + 0,0000i	-0,1172 + 0,0000i	0,0000 - 0,1058i	0,0000 - 0,9254i
α_4	0,0000 - 1,5421i	0,0000 - 1,6285i	0,0000 - 1,6498i	0,0000 - 2,2565i

poichè siamo nel caso di $k_2 = 0$ essendo il vettore \mathbf{k} parallelo all'asse x_1) per alcuni valori della pulsazione ω . Si nota che fino alla soglia trasversa ($\sqrt{\frac{C_{44}}{\rho}} * |\mathbf{k}|$), pari in questo caso a $71.4 \frac{Grad}{s}$, esse sono reali ed il loro modulo ha andamento decrescente. Superato tale valore le due con modulo minore diventano immaginarie pure e cominciano a crescere in modulo, mentre le altre continuano nel trend precedente. Raggiunta la soglia longitudinale ($\sqrt{\frac{C_{11}}{\rho}} * |\mathbf{k}|$), pari a $119.5 \frac{Grad}{s}$, anche queste ultime diventano immaginarie pure ed iniziano a crescere in modulo al crescere della pulsazione.

Per il caso *esagonale* si sono scelti i seguenti valori delle costanti indipendenti: $C_{11} = 200 \text{ GPa}$, $C_{12} = 64 \text{ GPa}$, $C_{13} = 75 \text{ GPa}$, $C_{33} = 150 \text{ GPa}$ e $C_{44} = 79.6 \text{ GPa}$, scelte con i criteri spiegati nel paragrafo 4.2. Come densità si è utilizzata quella del silicio. Dalla Tabella 5.2 si vede che inizialmente le α_j sono complesse con parte reale e immaginaria entrambe diverse da zero. Al crescere della pulsazione il loro modulo diminuisce fino a circa $70 \frac{Grad}{s}$, dove diventano tutte immaginarie pure. Tra questo valore e la soglia trasversa isotropa ($\sqrt{\frac{C_{44}}{\rho}} * |\mathbf{k}|$) la coppia con modulo minore

Tabella 5.2: Andamento delle α_j in funzione della pulsazione ω : simmetria esagonale.

$x10^7$	Pulsazione $\omega \times 10^9$ (rad/s)			
	50	69	70	71
α_1	0,8123 + 0,6656i	0,1747 + 0,5657i	0,0000 + 0,3766i	0,0000 + 0,1714i
α_2	0,8123 - 0,6656i	0,1747 - 0,5657i	0,0000 + 0,7067i	0,0000 + 0,8216i
α_3	-0,8123 - 0,6656i	-0,1747 - 0,5657i	0,0000 - 0,3766i	0,0000 - 0,1714i
α_4	-0,8123 + 0,6656i	-0,1747 + 0,5657i	0,0000 - 0,7067i	0,0000 - 0,8216i

$x10^7$	Pulsazione $\omega \times 10^9$ (rad/s)			
	72	100	113	114
α_1	0,1931 + 0,0000i	0,4473 + 0,0000i	0,0564 + 0,0000i	0,0000 + 0,1263i
α_2	0,0000 + 0,8976i	0,0000 + 1,7679i	0,0000 + 2,0417i	0,0000 + 2,0618i
α_3	-0,1931 + 0,0000i	-0,4473 + 0,0000i	-0,0564 + 0,0000i	0,0000 - 0,1263i
α_4	0,0000 - 0,8976i	0,0000 - 1,7679i	0,0000 - 2,0417i	0,0000 - 2,0618i

decesce, mentre l'altra cresce. Superata la soglia trasversa quella decrescente diventa reale e comincia a crescere in modulo fino ad un certo punto e poi decresce, l'altra continua il suo trend. Raggiunta la soglia longitudinale isotropa ($\sqrt{\frac{C_{11}}{\rho}} * |\mathbf{k}|$), la coppia reale ritorna complessa pura come l'altra coppia, poi entrambe crescono in modulo al crescere della pulsazione. Quindi l'andamento generale è analogo a quello isotropo con la differenza di quel breve tratto precedente alla soglia trasversa isotropa dove sono tutte immaginarie pure. Questa zona è quella compresa tra le due soglie trasverse, che nel caso isotropo sono degeneri e quindi tale intervallo collassa, mentre nel caso esagonale, venendo meno la degenerazione, risulta visibile. In questo caso una soglia trasversa e quella longitudinale sono ancora date dalle stesse relazioni della simmetria isotropa.

Per il caso *cubico* questa prima serie di risultati sono stati ottenuti ponendo il vettore d'onda lungo l'asse x_1 , quindi per la direzione [100]. Per le simmetrie precedenti non era necessario precisarlo, essendo entrambe isotrope nel piano (x_1, x_2) , mentre in questo caso tale peculiarità viene persa. Come materiale è stato

Tabella 5.3: Andamento delle α_j in funzione della pulsazione ω : simmetria cubica.

$\times 10^7$	Pulsazione $\omega \times 10^9$ (rad/s)			
	20	39	40	46
α_1	$0,7769 + 0,8256i$	$0,0875 + 0,8114i$	$0,0000 + 0,6160i$	$0,0000 + 0,1447i$
α_2	$0,7769 - 0,8256i$	$0,0875 - 0,8114i$	$0,0000 + 1,0014i$	$0,0000 + 1,3743i$
α_3	$-0,7769 - 0,8256i$	$-0,0875 - 0,8114i$	$0,0000 - 0,6160i$	$0,0000 - 0,1447i$
α_4	$-0,7769 + 0,8256i$	$-0,0875 + 0,8114i$	$0,0000 - 1,0014i$	$0,0000 - 1,3743i$

$\times 10^7$	Pulsazione $\omega \times 10^9$ (rad/s)			
	47	66	67	90
α_1	$0,0661 + 0,0000i$	$0,0405 + 0,0000i$	$0,0000 + 0,1214i$	$0,0000 + 0,8360i$
α_2	$0,0000 + 1,4174i$	$0,0000 + 2,0494i$	$0,0000 + 2,0784i$	$0,0000 + 2,7065i$
α_3	$-0,0661 + 0,0000i$	$-0,0405 + 0,0000i$	$0,0000 - 0,1214i$	$0,0000 - 0,8360i$
α_4	$0,0000 - 1,4174i$	$0,0000 - 2,0494i$	$0,0000 - 2,0784i$	$0,0000 - 2,7065i$

scelto il nickel, di cui si richiamano le proprietà: $C_{11} = 261 \text{ GPa}$, $C_{12} = 151 \text{ GPa}$, $C_{44} = 130.9 \text{ GPa}$ e $\rho = 8908 \frac{\text{kg}}{\text{m}^3}$. I risultati ottenuti sono riportati in Tabella 5.3. Come nel mezzo esagonale le α_j inizialmente sono complesse con modulo decrescente al crescere di ω . Oltrepassato il valore di circa $40 \frac{\text{Grad}}{\text{s}}$ entrambe le coppie diventano immaginarie pure fino alla soglia trasversa isotropa pari $46.8 \frac{\text{Grad}}{\text{s}}$, una crescente in modulo, l'altra decrescente. Questi due valori corrispondono alle due soglie trasverse per la simmetria cubica lungo la direzione [100]. Tra i $46.8 \frac{\text{Grad}}{\text{s}}$ e la soglia longitudinale, che in questo caso coincide con quella isotropa pari a $66.10 \frac{\text{Grad}}{\text{s}}$, la coppia con modulo minore diventa reale, mentre l'altra continua con l'andamento già descritto. Oltre la soglia longitudinale ritornano tutte immaginarie pure e crescono in modulo con ω . Si è quindi riscontrato un andamento complessivo simile a quello del mezzo esagonale.

Ricapitolando, esistono quattro zone nello spettro della pulsazione ω riguardanti i valori della quattro α_j inerenti ai modi sagittali. La prima si estende da zero fino alla prima soglia trasversa ed è quella propria delle onde di superficie. In essa le α_j

sono a coppie reali per il mezzo isotropo e a coppie complesse coniugate per i mezzi esagonale e cubico e decrescono in modulo al crescere della frequenza. La seconda zona è compresa tra le due soglie trasverse e non esiste per il mezzo isotropo, poichè per esso le due onde trasverse sono degeneri. In questo intervallo, sia per il caso esagonale che cubico, entrambe le coppie sono immaginarie pure, una crescente in modulo, l'altra decrescente. Nella terza zona, compresa tra la soglia trasversa maggiore e quella longitudinale, per tutti e tre i casi una coppia è reale mentre l'altra immaginaria pura. Infine nella quarta zona che si estende oltre la soglia longitudinale entrambe le coppie sono immaginarie pure con modulo crescente con la frequenza. Quanto trovato dimostra anche perchè il metodo delle onde parziali su cui si basa il programma riesce a trovare solamente le onde di superficie più lente delle onde trasverse e non i modi più veloci definiti *pseudo-modi*, poichè in grado di accoppiarsi con le onde di volume. Infatti quando la parte reale delle α_j è nulla rimane solamente una dipendenza sinusoidale con la profondità (x_3) e quindi le uniche onde che possono soddisfare le condizioni al contorno sono le onde di volume per il caso di mezzo finito, mentre nessuna per il caso di mezzo semi-infinito.

5.2 Effetti dell'esagonalità del film

Lo studio si è concentrato in particolare sugli effetti che ha un film di simmetria esagonale posto sopra un substrato isotropo, in condizioni di perfetta aderenza. Come materiale è stato scelto il "simil-silicio", presentato nel paragrafo precedente, come spessore del substrato $10 \mu m$. Quest'ultimo valore assicura che le onde di superficie sulle due facce (superiore ed inferiore) non interagiscano tra di loro, poichè l'entità dello spessore è tale che esse si smorzano prima di riuscire ad incontrarsi. Così nello studio si può indifferentemente considerare la matrice delle condizioni al contorno collegata al caso di substrato finito o quella del substrato semi-infinito. La differenza sarà che nel primo caso troveremo anche le onde di volume, in particolare di slab, mentre nel secondo verranno "selezionate" solo le onde di superficie, le quali sono il soggetto della ricerca in corso. Quindi, per semplicità di analisi, utilizzeremo la matrice inerente al substrato semi-infinito. Si è adottata la seguente strategia:

inizialmente il film è stato posto con proprietà e geometria identiche al substrato, quindi anch'esso isotropo, poi si sono variate alcune costanti elastiche in modo da "accendere" una lieve anisotropia e renderla via via sempre crescente, osservando come cambiano i modi originari e se e quando ne nascono di nuovi. In particolare per far ciò è necessario porre le seguenti osservazioni. Nella simmetria isotropa il numero di costanti indipendenti sono 2, mentre in quella esagonale 5. Quindi ci sono 3 gradi di libertà di differenza. Identificando le due costanti indipendenti comuni in C_{12} e C_{44} , questi gradi di libertà di differenza sono dati dai valori che assumono C_{11} , C_{33} e C_{13} . Infatti nel caso isotropo esse assumono valori ben precisi, in particolare $C_{11_{iso}} = C_{12} + 2C_{44}$, $C_{33_{iso}} = C_{11_{iso}}$ e $C_{13_{iso}} = C_{12}$, mentre nel caso esagonale possono assumere qualsiasi valore entro i limiti termodinamici. Quindi si possono identificare i seguenti *rapporti di anisotropia* per descrivere i tre gradi di libertà:

$$\begin{aligned} An(C_{11}) &= \frac{C_{11}}{C_{11_{iso}}} = \frac{C_{11}}{C_{12} + 2C_{44}}; \\ An(C_{33}) &= \frac{C_{33}}{C_{11}}; \\ An(C_{13}) &= \frac{C_{13}}{C_{12}}; \end{aligned} \quad (5.1)$$

Tali rapporti possono assumere qualsiasi valore purchè siano rispettati i vincoli termodinamici, che per la simmetria esagonale sono:

$$C_{44} > 0, \quad C_{11} > |C_{12}|, \quad (C_{11} + C_{12})C_{33} > 2C_{13}^2.$$

Lo spessore del film è stato scelto in maniera tale che lo stesso abbia una influenza rilevante all'interno dei modi di superficie di partenza. Poichè inizialmente consideriamo il film uguale al substrato, che è isotropo, i modi di superficie in questione sono le onde di Rayleigh per il mezzo isotropo, già ampiamente studiate. Esse presentano due componenti, una longitudinale ed una trasversale, in quadratura di fase tra di loro. Lo spostamento delle particelle ad ogni profondità è perciò ellittico, con il piano dell'ellisse coincidente con il piano sagittale e l'asse maggiore perpendicolare alla superficie libera. Entrambe le componenti decadono entro uno spazio pari a circa 2.5 lunghezze d'onda. Nelle prove effettuate si è usato il vettore d'onda già utilizzato precedentemente con modulo pari $|\mathbf{k}| = 2 * \frac{2\pi}{514.5 * 10^{-9}} * \sin(30^\circ) \text{ m}^{-1}$

e direzione posta per semplicità lungo l'asse x_1 , a cui corrisponde una lunghezza d'onda ($|\mathbf{k}| = \frac{2\pi}{\lambda}$) $\lambda = 514.5 \text{ nm}$, quindi circa mezzo micrometro. Perciò l'onda decade entro circa $1.5 \mu\text{m}$. Per questo motivo come spessore del film è stato scelto il valore di $0.5 \mu\text{m}$. In Figura 5.1 è rappresentato l'andamento delle due componenti degli spostamenti dell'onda di Rayleigh in funzione della profondità x_3 e il determinante della matrice delle condizioni al contorno in funzione di ω . Essa è stata trovata ponendo le proprietà del film identiche a quelle del substrato, quindi è come se fosse un mezzo unico. Si sono differenziate mediante l'uso dei colori le parti all'interno del film e quelle all'interno del substrato sia per la componente longitudinale u_1 , che per quella trasversale u_3 . In verde è riportato il cambiamento di fase relativa tra le due componenti. La linea nera indica la fase nulla. Il valore superiore della linea verde corrisponde alla fase $\frac{\pi}{2}$, quello inferiore alla fase $-\frac{\pi}{2}$ e la simmetria rispetto alla linea nera sottolinea il cambiamento di segno nella fase; ciò è dovuto al cambiamento di segno di una delle componenti, in questo caso u_1 . Quindi in prossimità della superficie libera lo spostamento delle particelle è di tipo ellittico con un dato verso di rotazione, ad una certa profondità esso diventa lineare lungo x_3 e corrisponde al punto in cui si annulla u_1 , poi torna ellittico ma con verso di rotazione invertito. In Figura, invece, è rappresentato l'andamento

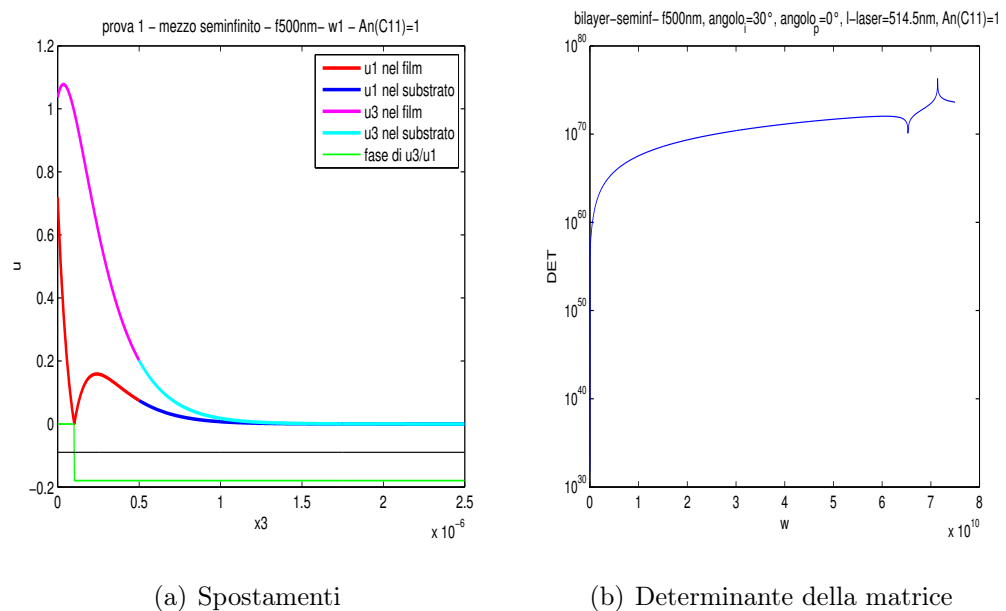


Fig. 5.1: Onda di Rayleigh per il mezzo isotropo "simil-silicio"

del determinante della matrice delle condizioni al contorno, nel caso di substrato semi-infinito, per la situazione precedente. Da essa si vede che la pulsazione dell'onda precedentemente descritta è compresa tra $60 \frac{\text{Grad}}{s}$ e $70 \frac{\text{Grad}}{s}$, in particolare è pari a $65.32 \frac{\text{Grad}}{s}$. Il picco presente a circa $71 \frac{\text{Grad}}{s}$ corrisponde invece alla soglia trasversa. Quanto descritto è la condizione di riferimento. A questo punto si è "accesa" l'anisotropia nel film mantenendo i due rapporti $An(C_{33})$ e $An(C_{13})$ unitari e diminuendo gradualmente $An(C_{11})$. Quindi rispetto al caso isotropo vengono cambiate contemporaneamente C_{11} , C_{22} e C_{33} , le quali si mantengono uguali fra di loro, e C_{66} , poichè nel mezzo esagonale mantiene la relazione $C_{66} = \frac{1}{2}(C_{11} - C_{12})$, mentre C_{55} rimane uguale a C_{44} , il cui valore è fisso, così come permane l'uguaglianza tra C_{13} e C_{12} . Si è ottenuto che basta rompere l'isotropia per far nascere un nuovo tipo di onda all'interfaccia tra il film ed il substrato, come illustrato in Figura 5.2 per $An(C_{11}) = 0.99$. L'onda è stata chiamata Onda "Anisotropia" per sottolineare il fatto che nasce a seguito della rottura dell'isotropia. Inizialmente essa accelera al diminuire di $An(C_{11})$ fino a diventare più veloce dell'onda trasversa e quindi scomparire. Ricompare però per $An(C_{11}) = 0.8$ e da qui in poi comincia a rallentare. Essa in realtà è stata ottenuta a seguito di un duplice effetto: la

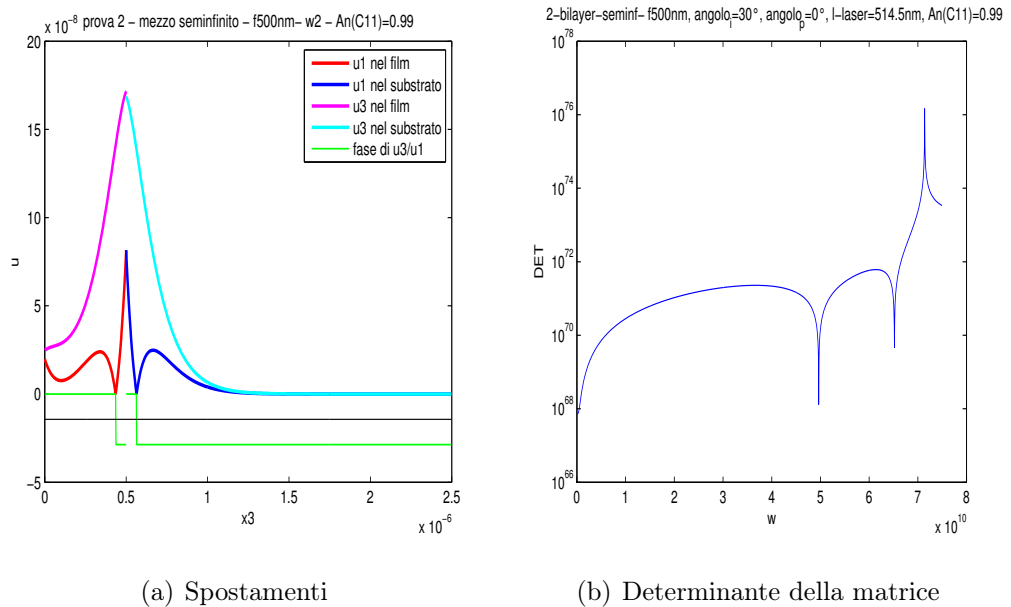


Fig. 5.2: Onda Anisotropia

diminuzione di C_{11} rispetto al substrato e la rottura dell'isotropia dovuta al diverso

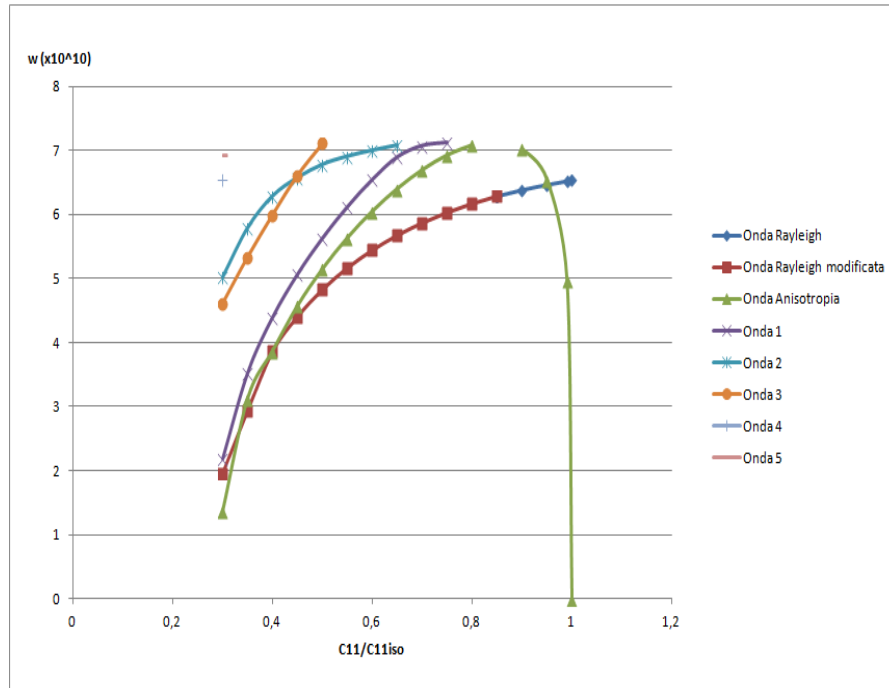
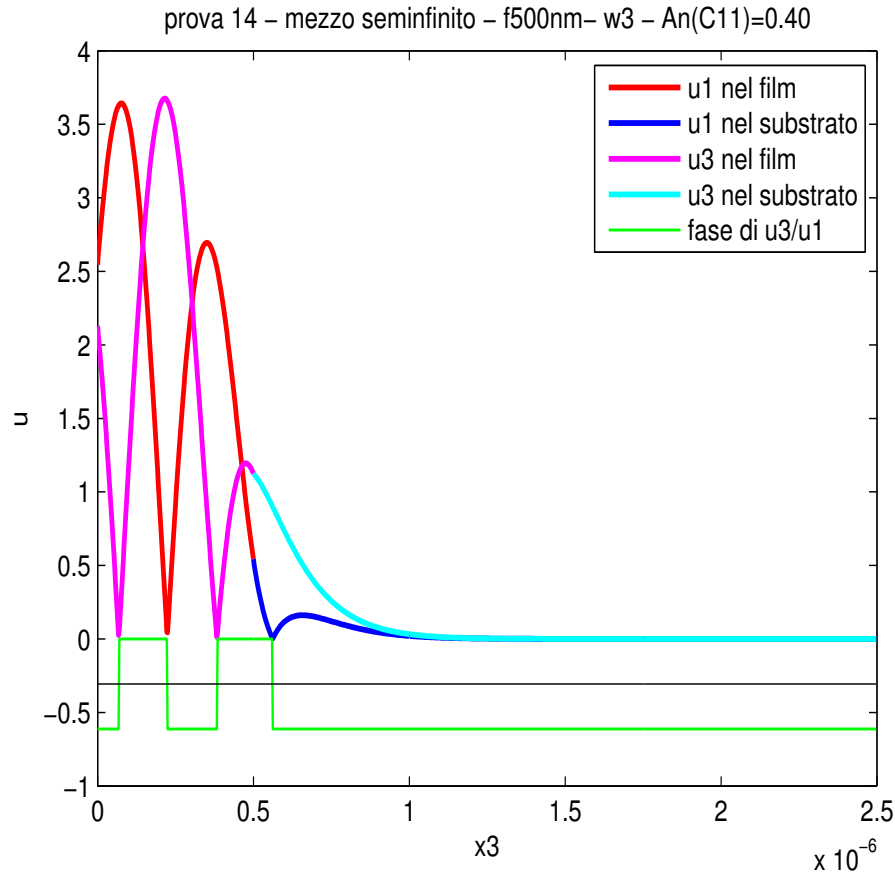


Fig. 5.3: Pulsazione delle onde in funzione del rapporto $An(C_{11})$

valore assunto da C_{66} rispetto a C_{44} . Si è quindi fatta un'ulteriore prova ponendo il film isotropo, ma con la stessa C_{11} , ottenendo solamente l'onda di Rayleigh. Ciò porterebbe a pensare che l'Onda "Anisotropia" sia appunto dovuta alla rottura dell'isotropia del film rispetto al substrato, piuttosto che alla diminuzione di C_{11} . L'onda di Rayleigh rimane, anche se rallentata e distorta. Ulteriori diminuzioni del rapporto $An(C_{11})$ hanno portato ad un continuo rallentamento dei modi sopra descritti ed anche alla loro distorsione. Ad un certo punto, però compaiono nuove onde di superficie: una per $An(C_{11}) = 0.75$ ("Onda 1"), una per $An(C_{11}) = 0.65$ ("Onda 2"), una per $An(C_{11}) = 0.5$ ("Onda 3") ed altre due per $An(C_{11}) = 0.3$ ("Onda 4" e "Onda 5"). Il limite termodinamico è a circa $An(C_{11}) = 0.29$. Nel grafico in Figura 5.3 sono riportate le pulsazioni di tutte le onde trovate in funzione del rapporto $An(C_{11})$. Da sottolineare che oltre i $71 \frac{Grad}{s}$ non esistono onde di superficie, in quanto tale valore corrisponde alla soglia trasversa del substrato. Le onde dalla 1 alla 5 compaiono quando le onde di volume del film diventano più lente di quella trasversa del substrato, così da potersi accoppiare con l'onda di superficie di quest'ultimo e dar luogo alle suddette onde. Ciò è visibile nelle Figure 5.4, 5.5, 5.6 dove sono raffigurate le onde 1, 2 e 3 per $An(C_{11}) = 0.4$. Per provare

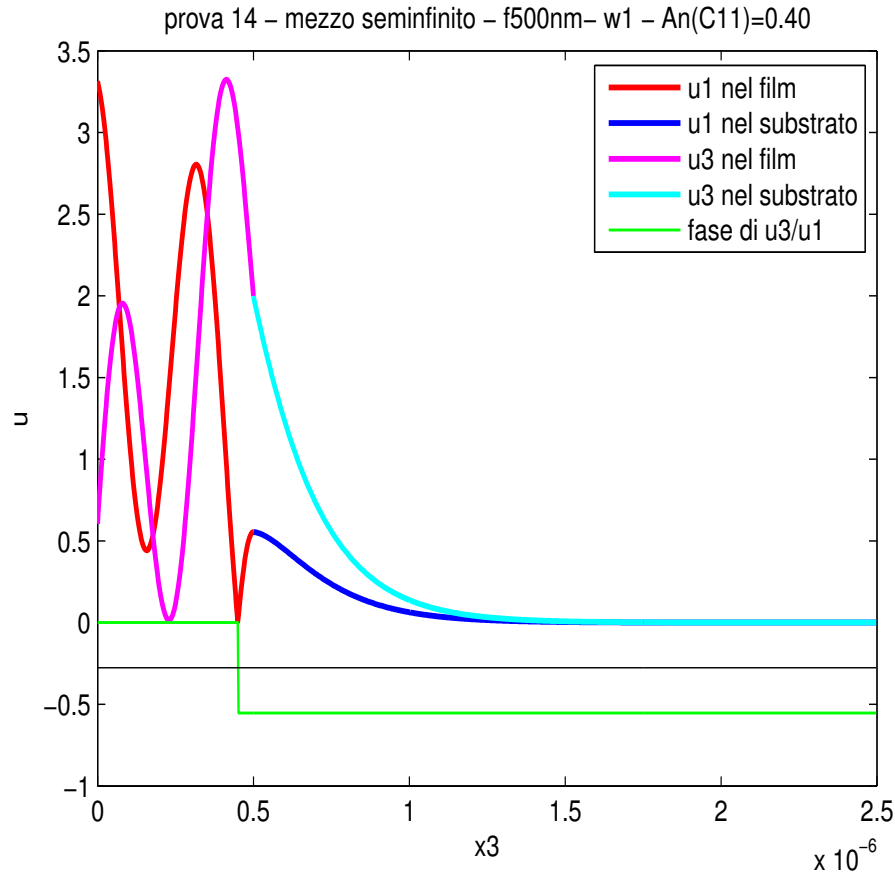


(a) Onda 1

Fig. 5.4: Spostamenti dell'onda 1 per $An(C_{11}) = 0.4$

questa ipotesi, sono state effettuate alcune simulazioni con lo spessore di 100 nm . Si è ottenuto che la fase iniziale è la medesima fino a $An(C_{11}) = 0.8$, quando ricompare l'onda "Anisotropia". Poi si ha sempre la comparsa di altre onde, ma in numero minore, come mostrato in Figura 5.7 per $An(C_{11}) = 0.8$ e $An(C_{11}) = 0.3$ per entrambi gli spessori. Questo è un altro punto a favore dell'ipotesi che tali onde siano, all'interno del film, onde di volume di slab. Infatti tali onde sono in numero maggiore per spessori maggiori. Onde di questo tipo, in film lento su substrato veloce, per la simmetria isotropa sono già state studiate in precedenza e sono dette di Sezawa, dal nome di colui che per primo le studiò.

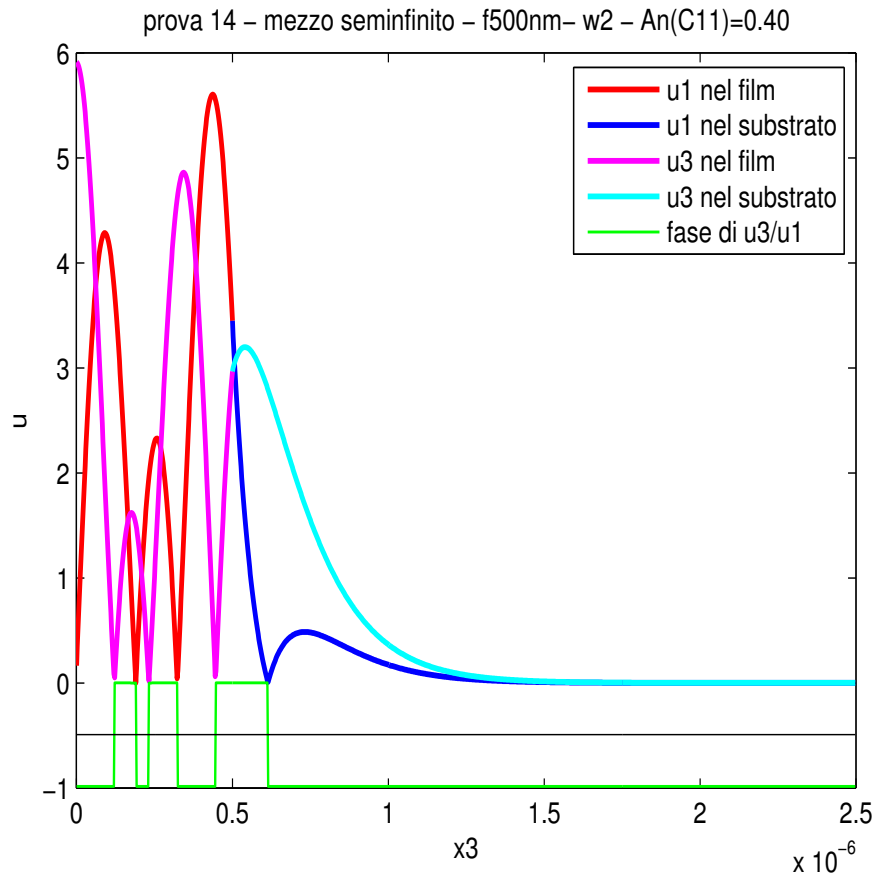
Dal grafico in Figura 5.3 è possibile osservare l'effetto complessivo che ha la variazione del rapporto $An(C_{11})$. Un suo aumento causa l'accelerazione dei modi e, viceversa, una sua diminuzione il loro rallentamento. Tuttavia essi ne risentono



(a) Onda 2

Fig. 5.5: Spostamenti dell'onda 2 per $An(C_{11}) = 0.4$

in maniera differente, poichè si osserva che alcuni modi rallentano in maniera più marcata fino a diventare più lenti di altri di cui prima erano più veloci. La sola rottura dell'isotropia causa il nascere di un'onda all'interfaccia film-substrato che accelera al crescere dell'anisotropia, fino a diventare più veloce dell'onda trasversa del substrato. Inoltre si vede che ad un certo punto l'onda di Rayleigh viene definita modificata, questo per sottolineare l'entità rilevante della distorsione. Infatti si è definito l'onda come di Rayleigh quando, pur leggermente distorta, presenta le caratteristiche riportate in Figura 5.1(a), cioè una componente (u_3) che decade senza cambiar fase, ed una (u_1) la cui fase si inverte. Con l'aggettivo "modificata" si intende che la distorsione è tale che le suddette proprietà sono cambiate (per esempio anche u_3 ad un certo punto inverte la propria fase). Un aumento oltre l'unità di $An(C_{11})$ non comporta il nascere di alcuna nuova onda, ma solamente



(a) Onda 3

Fig. 5.6: Spostamenti dell'onda 3 per $An(C_{11}) = 0.4$

l'accelerazione di quella di Rayleigh ed una sua forte distorsione. Noti questi risultati, si è variato anche il rapporto $An(C_{33})$ per alcuni valori fissi di $An(C_{11})$. Il risultato è un comportamento analogo a quello visto precedentemente, ma molto meno marcato. Se si cambiano contemporaneamente i due rapporti, gli effetti si sommano.

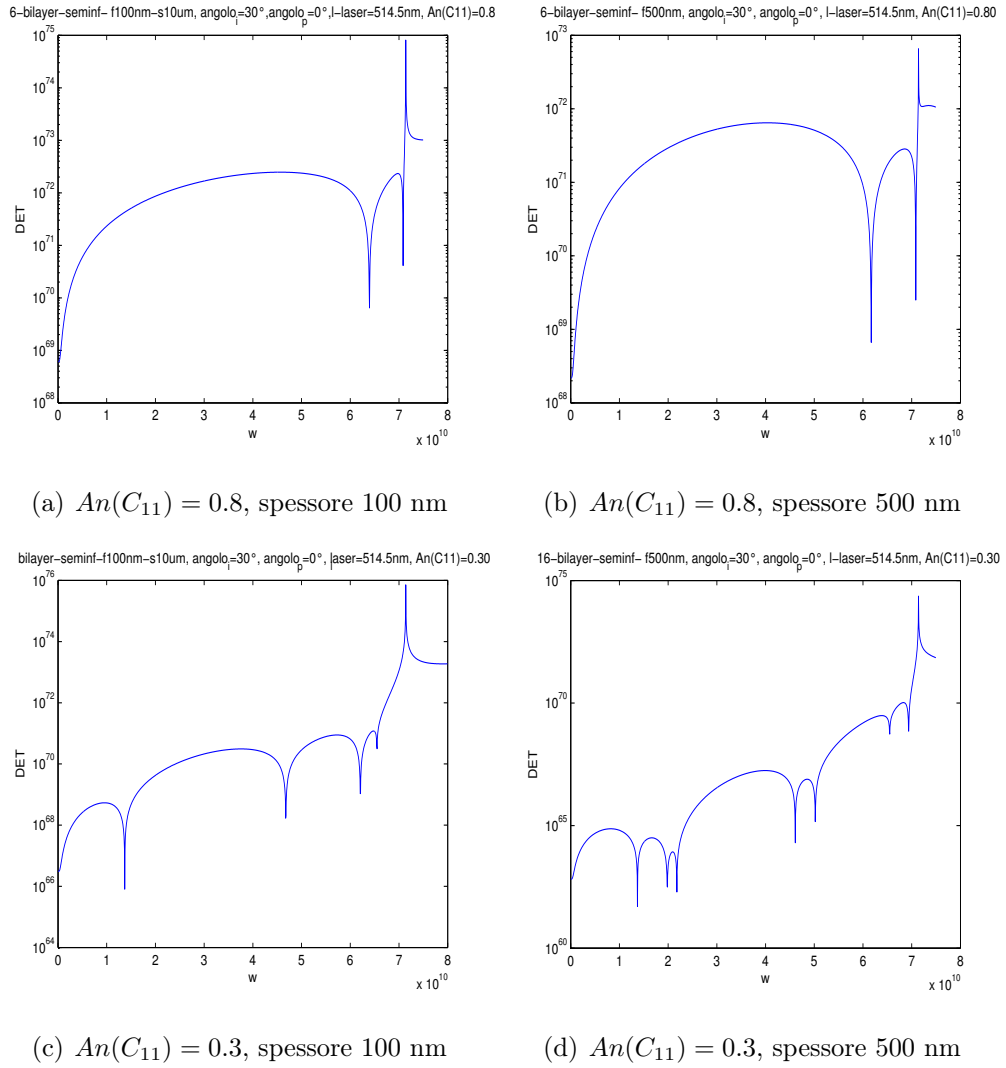


Fig. 5.7: Determinante della matrice delle condizioni al contorno per spessori del film: a) e c) 100 nm; b) e d) 500nm.

5.3 Simmetria cubica

Il programma, come si è visto, è in grado di trattare anche la simmetria cubica. In particolare riesce a cogliere l'anisotropia nel piano (x_1, x_2) . Infatti in questo caso non esiste più la condizione di isotropia come per l'isotropo e l'esagonale. Tuttavia un cubo presenta come piano di simmetria il piano $(\bar{1}, 1, 0)$, ovvero a 45 gradi tra gli assi x_1 e x_2 . Ciò significa che la direzione ad un angolo θ è equivalente alla direzione $90 - \theta$. Si è verificato ciò prendendo come materiale il nickel, le cui costanti elastiche indipendenti sono $C_{11} = 261 \text{ GPa}$, $C_{12} = 151.1 \text{ GPa}$ e $C_{44} = 130.9 \text{ GPa}$, mentre la densità vale $8908 \frac{\text{kg}}{\text{m}^3}$. Come spessore è stato preso $10 \mu\text{m}$. Dalle Figure

5.8 e 5.9 si vede che tale simmetria è soddisfatta. Studi inerenti alle onde di

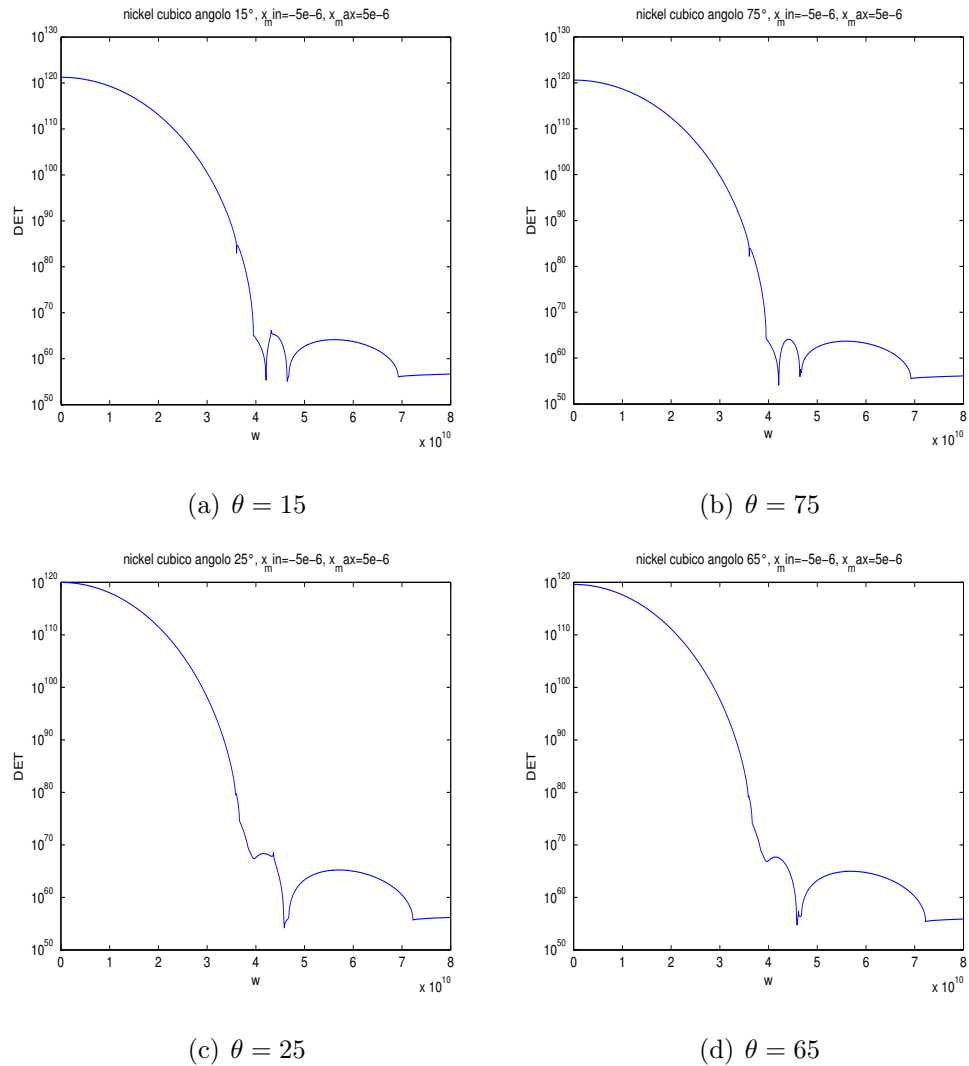


Fig. 5.8: Determinante della matrice delle condizioni al contorno per nickel cubico. (I)

superficie in simmetria cubica si trovano in [7] e [8]. Per concludere è necessario fare le seguenti considerazioni. Spesso, nel compiere le simulazioni, ci si potrebbe trovare in condizioni numeriche critiche, ovvero nell'avere la matrice delle condizioni al contorno malcondizionata. In molti casi, tuttavia il calcolatore riesce comunque a ricavare gli spostamenti. A volte invece il malcondizionamento è talmente elevato che non è possibile nemmeno ottenere il determinante in funzione di ω . Questo è quanto è successo con il nickel. È bastato però esprimere C_{44} , anziché con il valore numerico diretto 130.9, in funzione di C_{11} e C_{12} , attraverso l'espressione $\frac{2.38}{2}(C_{11} - C_{12})$, per rimuovere il problema. Numericamente i due valori differiscono

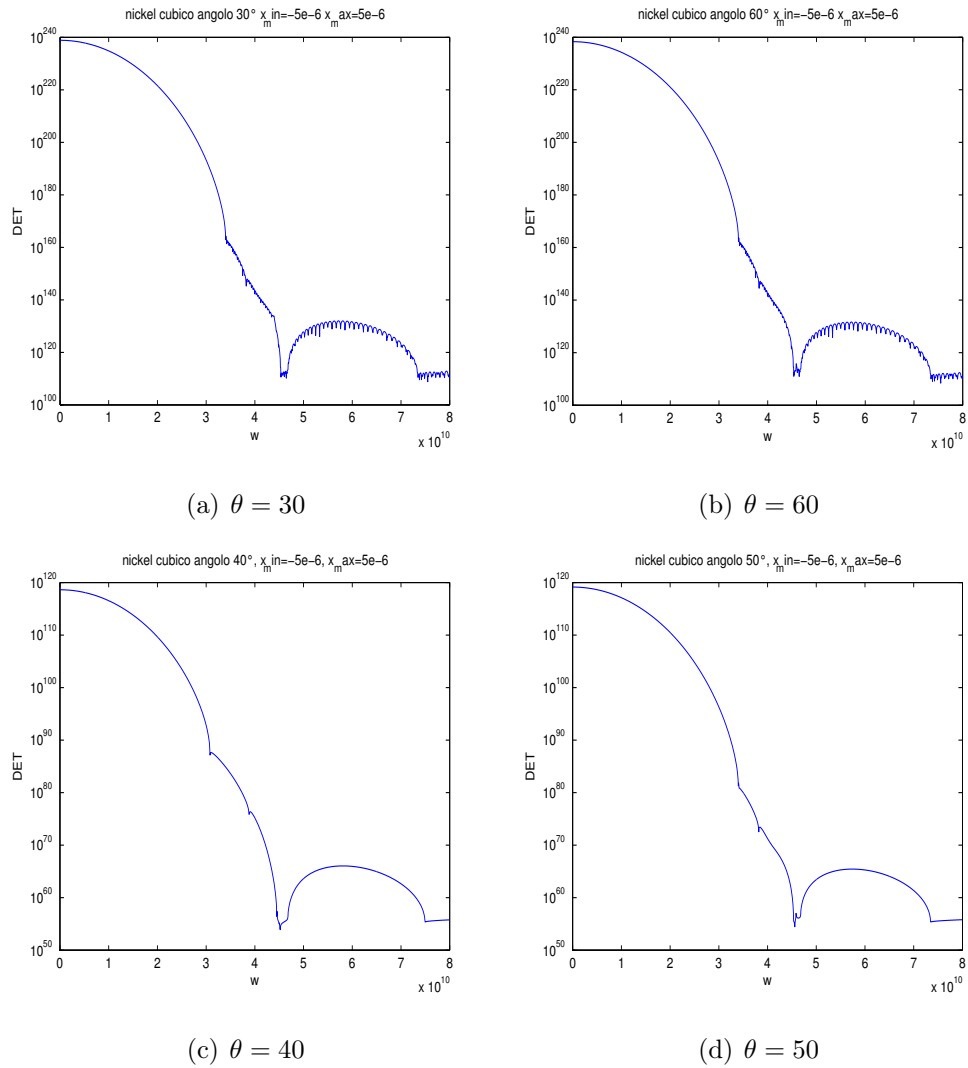


Fig. 5.9: Determinante della matrice delle condizioni al contorno per nickel cubico. (II)

solamente di errore dell'ordine di 10^{-5} , ma ciò è stato sufficiente per permetterci di ottenere i risultati esposti. Nei casi in cui nemmeno questa manipolazione sia efficace, pur essendo corrette nella forma le matrici delle condizioni al contorno ottenute dalla Prima Parte del programma, una loro risoluzione richiede l'utilizzo di metodi propri per le matrici malcondizionate, non inseriti nella Seconda Parte.

5.4 Conclusioni

Il processo di caratterizzazione delle proprietà elastiche di un materiale avviene attraverso il confronto tra dati sperimentali, ad esempio ottenuti attraverso la

spettroscopia Brillouin di superficie, e dati analitici ottenuti da un opportuno software. Al fine di minimizzare l'errore nella determinazione di tali costanti è necessario avere un programma di calcolo basato su un modello che approcci al meglio la struttura fisica del materiale. Spesso si trattano sistemi multistrati, in particolare costituiti da un film sottile depositato sopra un substrato di spessore molto maggiore. I programmi di calcolo generalmente utilizzati sono basati su un modello che considera il substrato avente simmetria isotropa. Ciò non sempre corrisponde alla realtà fisica del campione studiato, comportando degli errori nella determinazione delle proprietà elastiche. Inoltre presso il nostro Dipartimento sono presenti programmi che trattano solamente i casi semplificati in cui i modi sagittali si disaccoppiano da quelli nella direzione perpendicolare al piano sagittale (ciò causa una notevole semplificazione delle equazioni delle onde). Questo non comporta un problema nel trattare le simmetrie isotropa ed esagonale, poichè presentano come propria caratteristica l'isotropia nel piano (x_1, x_2) . Tuttavia risulta di cruciale importanza nello studiare la simmetria cubica, poichè per essa il disaccoppiamento avviene solamente in un numero limitato di direzioni.

Si è quindi creato un programma di calcolo di significativa generalità, nel quale sono trattate le simmetrie isotropa, cubica ed esagonale. Esso è costruito per trattare sistemi multistrato ed il sistema monolayer risulta esserne un caso particolare. Non c'è alcun vincolo sul numero massimo degli strati, così come sul loro spessore (gli unici vincoli sono pratici di natura computazionale, poichè al crescere degli strati aumentano le tempistiche di calcolo). Si può fare qualsiasi tipo di combinazione tra le simmetrie considerate e gli studi possono essere effettuati in qualsiasi direzione.

Sono state effettuate delle prove per testare il corretto funzionamento del programma.

Si sono poi effettuati alcuni studi volti a mostrare le capacità del programma di calcolo. Inizialmente si è cercato di delineare alcune caratteristiche proprie di ogni simmetria considerata studiando il comportamento delle α_j al variare della pulsazione ω . Successivamente si è passato allo studio di film sottili a simmetria esagonale posti sopra un substrato isotropo. Lo scopo era quello di valutare gli effetti della simmetria esagonale del film sulle onde di superficie del substrato

isotropo. Per far ciò si è preso come riferimento il sistema composto dal film con proprietà identiche al substrato, quindi isotropo. Si è scelto un opportuno modo di sviluppare anisotropia esagonale crescente, partendo dalla condizione di riferimento. Durante tale procedimento si sono osservati gli effetti dell'anisotropia sulle onde di superficie. Infine si è mostrata la potenzialità del programma nello studio della simmetria cubica in qualunque direzione nel piano (x_1, x_2) .

Appendice A

Codice

A.1 Prima Parte

```
Tesim x
1 %Calcolo delle funzioni spostamento
2
3 %definizione delle variabili e delle funzioni spostamento
4 - syms x1 x2 x3 alfa b c t w q_1 q_2 A1 A2 A3 rho
5 - syms u1(x1,x2,x3,t) u2(x1,x2,x3,t) u3(x1,x2,x3,t)
6 - syms A1_1 A1_2 A1_3 A1_4 A1_5 A1_6
7 - syms A2_1 A2_2 A2_3 A2_4 A2_5 A2_6
8 - syms A3_1 A3_2 A3_3 A3_4 A3_5 A3_6
9 - syms x_max x_min
10
11
12 strati=input('Inserire il numero di strati: ');
13 q_1=input('Digitare il valore del vettore d onda k1: ');
14 q_2=input('Digitare il valore del vettore d onda k2: ');
15 if strati==0
16     error('Valore non accettabile');
17 end
18
19
20 X_MIN=sym('X_MIN',[strati 1]);
21 X_MAX=sym('X_MAX',[strati 1]);
22 RHO=sym('RHO',[strati 1]);
23
24 ETA=sym('ETA',[2 strati]);
25 rif=zeros(1,strati);
26
27 if q_2==0||q_1==0
28     BCMatr=sym('BCMatr',[8 strati]);
29     FinM=sym('FinM',[4*strati (4*strati)]);
```

```

30 -     ALFA=sym('ALFA',[4 strati]);
31 -     ir2_min=1;
32 -
33 -     for index1=1:(4*strati)
34 -     for index2=1:(4*strati)
35 -         FinM(index1,index2)=0;
36 -     end
37 - end
38 -
39 - else
40 -     BCMatr=sym('BCMatr', [12 strati]);
41 -     FinM=sym('FinM',[(6*strati) (6*strati)]);
42 -     ALFA=sym('ALFA',[6 strati]);
43 -     ir_min=1;
44 -     for index1=1:(6*strati)
45 -     for index2=1:(6*strati)
46 -         FinM(index1,index2)=0;
47 -     end
48 - end
49 -
50 - end
51 -
52 -
53 - for l=1:strati
54 -
55 -     disp(' ')
56 -     Layer=sprintf('Layer n° %d',l);
57 -     disp(Layer)
58 -     disp(' ')
59 -
60 -     if l==1
61 -         x_min=input('Digitare il valore di x_min: ');
62 -         x_max=input('Digitare il valore di x_max: ');
63 -         rho=input('Digitare il valore della densità: ');
64 -         RHO(l)=rho;
65 -         X_MIN(l)=x_min;
66 -         X_MAX(l)=x_max;
67 -     else
68 -         x_min=X_MAX(l-1); rs=double(x_min);
69 -         spes=sprintf('Il valore di x_min è %d',rs);
70 -         disp(spes)
71 -         x_max=input('Digitare il valore di x_max: ');
72 -         rho=input('Digitare il valore della densità: ');
73 -         RHO(l)=rho;
74 -         X_MIN(l)=x_min;
75 -         X_MAX(l)=x_max;
76 -
77 -     end
78 -
79 -     %definizione degli spostamenti
80 -
81 -     u1(x1,x2,x3,t)=exp(i1*(q_1*x1+q_2*x2-w*t))*A1*exp(-alfa*x3);
82 -
83 -
84 -     u2(x1,x2,x3,t)=exp(i1*(q_1*x1+q_2*x2-w*t))*A2*exp(-alfa*x3);
85 -
86 -
87 -     u3(x1,x2,x3,t)=exp(i1*(q_1*x1+q_2*x2-w*t))*A3*exp(-alfa*x3);
88 -
89 -

```



```

90
91 %definizione del vettore deformazione (6X1) (notazione matriciale)
92 E=[diff(u1,x1); diff(u2,x2); diff(u3,x3); diff(u2,x3)+diff(u3,x2); diff(u1,x3)+diff(u3,x1); diff(u1,x2)+diff(u2,x1)];
93
94 %definizione della matrice (6X6) delle costanti elastiche
95 C=zeros(6);
96
97 %scelta della geometria
98 disp(' ')
99 disp('Scegliere la geometria')
100 n=input('Digitare 1 per isotropo, 2 per cubico, 3 per esagonale: ');
101 disp(' ')
102
103 %isotropo
104 if n==1
105     disp('Isotropo');
106     C(1,1)=input('Inserire un valore per C11: ');
107     C(2,2)=C(1,1); C(3,3)=C(1,1);
108     C(1,2)=input('Inserire un valore per C12: ');
109     C(1,3)=C(1,2); C(2,3)=C(1,2);
110     C(3,1)=C(1,3); C(2,1)=C(1,2); C(3,2)=C(2,3);
111     C(4,4)=0.5*(C(1,1)-C(1,2))*0.9999999999; C(5,5)=C(4,4); C(6,6)=C(4,4);
112
113 %cubico
114 elseif n==2
115     disp('Cubico')
116     C(1,1)=input('Inserire un valore per C11: ');
117     C(2,2)=C(1,1); C(3,3)=C(1,1);
118     C(1,2)=input('Inserire un valore per C12: ');
119     C(1,3)=C(1,2); C(2,3)=C(1,2);
120     C(3,1)=C(1,3); C(2,1)=C(1,2); C(3,2)=C(2,3);
121     C(4,4)=input('Inserire un valore per C44: ');
122     C(5,5)=C(4,4); C(6,6)=C(4,4);
123     if C(4,4)<0
124         error('Error: C44 must be >0');
125     end
126     if C(1,1)<abs(C(1,2))
127         error('Error: C11 must be >|C12|');
128     end
129     if C(1,1)+2*C(1,2)<0
130         error('Error: C11+2*C12 must be > 0');
131     end
132
133 %esagonale
134 elseif n==3
135     disp('Esagonale');
136     C(1,1)=input('Inserire un valore per C11: ');
137     C(2,2)=C(1,1);
138     C(1,2)=input('Inserire un valore per C12: ');
139     C(2,1)=C(1,2);
140     C(1,3)=input('Inserire un valore per C13: ');
141     C(2,3)=C(1,3); C(3,1)=C(1,3); C(3,2)=C(2,3);
142     C(3,3)=input('Inserire un valore per C33: ');
143     C(4,4)=input('Inserire un valore per C44: ');
144     C(5,5)=C(4,4);
145     C(6,6)=0.5*(C(1,1)-C(1,2))*0.9999999999;
146     if C(4,4)<0
147         error('Error: C44 must be >0');
148     end
149     if C(1,1)<abs(C(1,2))

```



```

210 - if q_2==0
211
212     %trasformazione da sistema a matrice e soluzioni
213     [B]=equationsToMatrix([eq1==0, eq3==0],[A1,A3]);
214     sol_alfa=solve(det(B)==0, alfa, 'IgnoreAnalyticConstraints', true);
215     sol_alfa=simplify(sol_alfa);
216     ALFA(:,1)=sol_alfa;
217
218
219
220     %espressione dei rapporti tra le ampiezze
221     amp=B*[A1;A3];
222     amp1=[0 1]*amp;
223     sol_A3=solve(amp1==0,A3);
224     etal=simplifyFraction(sol_A3/A1); %etal=A3/A1
225     %A2=0
226     zz=simplify(w-w); %lo uso per porre A2=0 in simbolico
227
228     rif(1)=1;
229
230     %verifico la non singularità di etal per le soluzioni di alfa
231     [num1,den1]=numden(etal);
232     J=zeros(1,4);
233     for index=1:4
234         test=subs(den1, alfa, sol_alfa(index));
235         J(1,index)=isequaln(test,0);
236     end
237     if sum(J)==0
238         riferimento='Il riferimento è A1'; %etal=A3/A1
239         rif(1)=1;
240     else
241         etal=num1/den1;
242         etal=simplifyFraction(1/etal); %etal=A1/A3
243         riferimento='Il riferimento è A3';
244         rif(1)=3;
245     end
246
247
248     ETA(1,1)=etal;
249     ETA(2,1)=NaN;
250     disp(riferimento)
251
252     if rif(1)==1
253
254         %esprimo sigma5 e sigma3 solo in funzione di A1 usando etal
255         BCS1=subs(S5,A2,zz);
256         BCS1=subs(BCS1,A3,(A1*etal));
257         BCS3=subs(S3,A2,zz);
258         BCS3=subs(BCS3,A3,(A1*etal));
259         BCS1=simplifyFraction(BCS1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
260         BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
261
262         %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
263         BCS1_1=subs(BCS1,A1,A1_1);
264         BCS1_1=subs(BCS1_1, alfa, sol_alfa(1));
265         BCS1_2=subs(BCS1,A1,A1_2);
266         BCS1_2=subs(BCS1_2, alfa, sol_alfa(2));
267         BCS1_3=subs(BCS1,A1,A1_3);
268         BCS1_3=subs(BCS1_3, alfa, sol_alfa(3));
269         BCS1_4=subs(BCS1,A1,A1_4);

```

```

270 - BCS1_4=subs(BCS1_4,alfa,sol_alfa(4));
271
272
273 - BCS3_1=subs(BCS3,A1,A1_1);
274 - BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
275 - BCS3_2=subs(BCS3,A1,A1_2);
276 - BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
277 - BCS3_3=subs(BCS3,A1,A1_3);
278 - BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
279 - BCS3_4=subs(BCS3,A1,A1_4);
280 - BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
281
282
283 %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
284 - eq_BCS1=BCS1_1+BCS1_2+BCS1_3+BCS1_4; %sigma5
285 - eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4; %sigma3
286
287 %sostituisco il valore di x3 agli estremi
288 - eq_BCS1_x_min=subs(eq_BCS1,x3,x_min);
289 - eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
290 - eq_BCS1_x_max=subs(eq_BCS1,x3,x_max);
291 - eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
292
293 %esprimo gli spostamenti solo in funzione di A1 usando etal
294 - BCU1=u1;
295 - BCU3=subs(u3,A3,(A1*etal));
296 - BCU1=simplifyFraction(BCU1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
297 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
298
299 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
300 - BCU1_1=subs(BCU1,A1,A1_1);
301 - BCU1_1=subs(BCU1_1,alfa,sol_alfa(1));
302 - BCU1_2=subs(BCU1,A1,A1_2);
303 - BCU1_2=subs(BCU1_2,alfa,sol_alfa(2));
304 - BCU1_3=subs(BCU1,A1,A1_3);
305 - BCU1_3=subs(BCU1_3,alfa,sol_alfa(3));
306 - BCU1_4=subs(BCU1,A1,A1_4);
307 - BCU1_4=subs(BCU1_4,alfa,sol_alfa(4));
308
309
310 - BCU3_1=subs(BCU3,A1,A1_1);
311 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
312 - BCU3_2=subs(BCU3,A1,A1_2);
313 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
314 - BCU3_3=subs(BCU3,A1,A1_3);
315 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
316 - BCU3_4=subs(BCU3,A1,A1_4);
317 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
318
319 %ottengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
320 - eq_BCU1=BCU1_1+BCU1_2+BCU1_3+BCU1_4; %u1
321 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4; %u3
322
323 %sostituisco il valore di x3 agli estremi
324 - eq_BCU1_x_min=subs(eq_BCU1,x3,x_min);
325 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
326 - eq_BCU1_x_max=subs(eq_BCU1,x3,x_max);
327 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
328
329 %Metto tutto in una matrice dove la colonna indica lo strato

```

```

330 - BCMatr(1,1)=eq_BCS1_x_min; %sigma5 in x_min
331 - BCMatr(2,1)=eq_BCS3_x_min; %sigma3 in x_min
332 - BCMatr(3,1)=eq_BCS1_x_max; %sigma5 in x_max
333 - BCMatr(4,1)=eq_BCS3_x_max; %sigma3 in x_max
334 - BCMatr(5,1)=eq_BCU1_x_min; %u1 in x_min
335 - BCMatr(6,1)=eq_BCU3_x_min; %u3 in x_min
336 - BCMatr(7,1)=eq_BCU1_x_max; %u1 in x_max
337 - BCMatr(8,1)=eq_BCU3_x_max; %u3 in x_max
338
339 - elseif rif(1)==3
340
341 %esprimo sigma5 e sigma3 solo in funzione di A3 usando etal
342 BCS1=subs(S5,A1,(A3*etal));
343 BCS1=subs(BCS1,A2,zz);
344 BCS3=subs(S3,A1,(A3*etal));
345 BCS3=subs(BCS3,A2,zz);
346 BCS1=simplifyFraction(BCS1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
347 BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
348
349 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
350 BCS1_1=subs(BCS1,A3,A3_1);
351 BCS1_1=subs(BCS1_1,alfa,sol_alfa(1));
352 BCS1_2=subs(BCS1,A3,A3_2);
353 BCS1_2=subs(BCS1_2,alfa,sol_alfa(2));
354 BCS1_3=subs(BCS1,A3,A3_3);
355 BCS1_3=subs(BCS1_3,alfa,sol_alfa(3));
356 BCS1_4=subs(BCS1,A3,A3_4);
357 BCS1_4=subs(BCS1_4,alfa,sol_alfa(4));
358
359 BCS3_1=subs(BCS3,A3,A3_1);
360 BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
361 BCS3_2=subs(BCS3,A3,A3_2);
362 BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
363 BCS3_3=subs(BCS3,A3,A3_3);
364 BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
365 BCS3_4=subs(BCS3,A3,A3_4);
366 BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
367
368 %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
369 eq_BCS1=BCS1_1+BCS1_2+BCS1_3+BCS1_4; %sigma5
370 eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4; %sigma3
371
372 %sostituisco il valore di x3 agli estremi
373 eq_BCS1_x_min=subs(eq_BCS1,x3,x_min);
374 eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
375 eq_BCS1_x_max=subs(eq_BCS1,x3,x_max);
376 eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
377
378 %esprimo gli spostamenti solo in funzione di A3 usando etal e eta2
379 BCU1=subs(u1,A1,(A3*etal));
380 BCU3=u3;
381 BCU1=simplifyFraction(BCU1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
382 BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
383
384 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
385 BCU1_1=subs(BCU1,A3,A3_1);
386 BCU1_1=subs(BCU1_1,alfa,sol_alfa(1));
387 BCU1_2=subs(BCU1,A3,A3_2);
388 BCU1_2=subs(BCU1_2,alfa,sol_alfa(2));
389 BCU1_3=subs(BCU1,A3,A3_3);

```

```

390 - BCU1_3=subs(BCU1_3,alfa,sol_alfa(3));
391 - BCU1_4=subs(BCU1,A3,A3_4);
392 - BCU1_4=subs(BCU1_4,alfa,sol_alfa(4));
393
394
395 - BCU3_1=subs(BCU3,A3,A3_1);
396 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
397 - BCU3_2=subs(BCU3,A3,A3_2);
398 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
399 - BCU3_3=subs(BCU3,A3,A3_3);
400 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
401 - BCU3_4=subs(BCU3,A3,A3_4);
402 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
403
404 %ottengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
405 - eq_BCU1=BCU1_1+BCU1_2+BCU1_3+BCU1_4; %u1
406 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4; %u3
407
408 %sostituisco il valore di x3 agli estremi
409 - eq_BCU1_x_min=subs(eq_BCU1,x3,x_min);
410 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
411 - eq_BCU1_x_max=subs(eq_BCU1,x3,x_max);
412 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
413
414 %Metto tutto in una matrice dove la colonna indica lo strato
415 - BCMatr(1,1)=eq_BCU1_x_min; %sigma5 in x_min
416 - BCMatr(2,1)=eq_BCU3_x_min; %sigma3 in x_min
417 - BCMatr(3,1)=eq_BCU1_x_max; %sigma5 in x_max
418 - BCMatr(4,1)=eq_BCU3_x_max; %sigma3 in x_max
419 - BCMatr(5,1)=eq_BCU1_x_min; %u1 in x_min
420 - BCMatr(6,1)=eq_BCU3_x_min; %u3 in x_min
421 - BCMatr(7,1)=eq_BCU1_x_max; %u1 in x_max
422 - BCMatr(8,1)=eq_BCU3_x_max; %u3 in x_max
423
424 - end
425
426 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% caso in cui k1=0
427 - elseif q_1==0
428
429 - %trasformazione da sistema a matrice e soluzioni
430 - [B]=equationsToMatrix([eq2==0, eq3==0],[A2,A3]);
431 - sol_alfa=solve(det(B)==0,alfa,'IgnoreAnalyticConstraints',true);
432 - sol_alfa=simplify(sol_alfa);
433 - ALFA(:,1)=sol_alfa;
434
435 - %espressione dei rapporti tra le ampiezze
436 - amp=B*[A2;A3];
437 - amp1=[0 1]*amp;
438 - sol_A3=solve(amp1==0,A3);
439 - etal=simplifyFraction(sol_A3/A2); %etal=A3/A2
440 - %A1=0
441 - zz=simplify(w-w); %lo uso per porre A2=0 in simbolico
442
443 - rif(1)=2;
444
445 - %verifico la non singolarità di etal per le soluzioni di alfa
446 - [num1,den1]=numden(etal);
447 - J=zeros(1,4);
448 - for index=1:4
449 -     test=subs(den1,alfa,sol_alfa(index));

```

```

450 -         J(1,index)=isequaln(test,0);
451 -     end
452 -     if sum(J)==0
453 -         riferimento='Il riferimento è A2';
454 -         rif(1)=2;
455 -     else
456 -         etai=num1/den1;
457 -         etai=simplifyFraction(1/eta1); %eta1=A2/A3
458 -         riferimento='Il riferimento è A3';
459 -         rif(1)=3;
460 -     end
461
462
463 -     ETA(1,1)=eta1;
464 -     ETA(2,1)=NaN;
465 -     disp(riferimento)
466
467 -     if rif(1)==2
468
469 -         %esprimo sigma4 e sigma3 solo in funzione di A2 usando etai
470 -         BCS2=subs(S4,A1,zz);
471 -         BCS2=subs(BCS2,A3,(A2*eta1));
472 -         BCS3=subs(S3,A1,zz);
473 -         BCS3=subs(BCS3,A3,(A2*eta1));
474 -         BCS2=simplifyFraction(BCS2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
475 -         BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
476
477 -         %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
478 -         BCS2_1=subs(BCS2,A2,A2_1);
479 -         BCS2_1=subs(BCS2_1,alfa,sol_alfa(1));
480 -         BCS2_2=subs(BCS2,A2,A2_2);
481 -         BCS2_2=subs(BCS2_2,alfa,sol_alfa(2));
482 -         BCS2_3=subs(BCS2,A2,A2_3);
483 -         BCS2_3=subs(BCS2_3,alfa,sol_alfa(3));
484 -         BCS2_4=subs(BCS2,A2,A2_4);
485 -         BCS2_4=subs(BCS2_4,alfa,sol_alfa(4));
486
487
488 -         BCS3_1=subs(BCS3,A2,A2_1);
489 -         BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
490 -         BCS3_2=subs(BCS3,A2,A2_2);
491 -         BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
492 -         BCS3_3=subs(BCS3,A2,A2_3);
493 -         BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
494 -         BCS3_4=subs(BCS3,A2,A2_4);
495 -         BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
496
497
498 -         %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
499 -         eq_BCS2=BCS2_1+BCS2_2+BCS2_3+BCS2_4; %sigma4
500 -         eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4; %sigma3
501
502 -         %sostituisco il valore di x3 agli estremi
503 -         eq_BCS2_x_min=subs(eq_BCS2,x3,x_min);
504 -         eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
505 -         eq_BCS2_x_max=subs(eq_BCS2,x3,x_max);
506 -         eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
507
508 -         %esprimo gli spostamenti solo in funzione di A2 usando etai
509 -         BCU2=u2;

```

```

510 - BCU3=subs(u3,A3, (A2*etal));
511 - BCU2=simplifyFraction(BCU2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
512 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
513
514 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
515 - BCU2_1=subs(BCU2,A2,A2_1);
516 - BCU2_1=subs(BCU2_1,alfa,sol_alfa(1));
517 - BCU2_2=subs(BCU2,A2,A2_2);
518 - BCU2_2=subs(BCU2_2,alfa,sol_alfa(2));
519 - BCU2_3=subs(BCU2,A2,A2_3);
520 - BCU2_3=subs(BCU2_3,alfa,sol_alfa(3));
521 - BCU2_4=subs(BCU2,A2,A2_4);
522 - BCU2_4=subs(BCU2_4,alfa,sol_alfa(4));
523
524
525 - BCU3_1=subs(BCU3,A2,A2_1);
526 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
527 - BCU3_2=subs(BCU3,A2,A2_2);
528 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
529 - BCU3_3=subs(BCU3,A2,A2_3);
530 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
531 - BCU3_4=subs(BCU3,A2,A2_4);
532 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
533
534 %ottengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
535 - eq_BCU2=BCU2_1+BCU2_2+BCU2_3+BCU2_4; %u2
536 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4; %u3
537
538 %sostituisco il valore di x3 agli estremi
539 - eq_BCU2_x_min=subs(eq_BCU2,x3,x_min);
540 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
541 - eq_BCU2_x_max=subs(eq_BCU2,x3,x_max);
542 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
543
544 %Metto tutto in una matrice dove la colonna indica lo strato
545 - BCMatr(1,1)=eq_BCS2_x_min; %sigma4 in x_min
546 - BCMatr(2,1)=eq_BCS3_x_min; %sigma3 in x_min
547 - BCMatr(3,1)=eq_BCS2_x_max; %sigma4 in x_max
548 - BCMatr(4,1)=eq_BCS3_x_max; %sigma3 in x_max
549 - BCMatr(5,1)=eq_BCU2_x_min; %u2 in x_min
550 - BCMatr(6,1)=eq_BCU3_x_min; %u3 in x_min
551 - BCMatr(7,1)=eq_BCU2_x_max; %u2 in x_max
552 - BCMatr(8,1)=eq_BCU3_x_max; %u3 in x_max
553
554 - elseif rif(1)==3
555
556 %esprimo sigma4 e sigma3 solo in funzione di A3 usando etal
557 - BCS2=subs(S4,A2, (A3*etal));
558 - BCS2=subs(BCS2,A1,zz);
559 - BCS3=subs(S3,A2, (A3*etal));
560 - BCS3=subs(BCS3,A1,zz);
561 - BCS2=simplifyFraction(BCS2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
562 - BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
563
564 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
565 - BCS2_1=subs(BCS2,A3,A3_1);
566 - BCS2_1=subs(BCS2_1,alfa,sol_alfa(1));
567 - BCS2_2=subs(BCS2,A3,A3_2);
568 - BCS2_2=subs(BCS2_2,alfa,sol_alfa(2));
569 - BCS2_3=subs(BCS2,A3,A3_3);

```



```

570 - BCS2_3=subs(BCS2_3,alfa,sol_alfa(3));
571 - BCS2_4=subs(BCS2_4,A3,A3_4);
572 - BCS2_4=subs(BCS2_4,alfa,sol_alfa(4));
573
574 - BCS3_1=subs(BCS3,A3,A3_1);
575 - BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
576 - BCS3_2=subs(BCS3,A3,A3_2);
577 - BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
578 - BCS3_3=subs(BCS3,A3,A3_3);
579 - BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
580 - BCS3_4=subs(BCS3,A3,A3_4);
581 - BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
582
583 %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
584 - eq_BCS2=BCS2_1+BCS2_2+BCS2_3+BCS2_4; %sigma5
585 - eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4; %sigma3
586
587 %sostituisco il valore di x3 agli estremi
588 - eq_BCS2_x_min=subs(eq_BCS2,x3,x_min);
589 - eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
590 - eq_BCS2_x_max=subs(eq_BCS2,x3,x_max);
591 - eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
592
593 %esprimo gli spostamenti solo in funzione di A3 usando eta1 e eta2
594 - BCU2=subs(u2,A2,(A3*eta1));
595 - BCU3=u3;
596 - BCU2=simplifyFraction(BCU2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
597 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
598
599 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
600 - BCU2_1=subs(BCU2,A3,A3_1);
601 - BCU2_1=subs(BCU2_1,alfa,sol_alfa(1));
602 - BCU2_2=subs(BCU2,A3,A3_2);
603 - BCU2_2=subs(BCU2_2,alfa,sol_alfa(2));
604 - BCU2_3=subs(BCU2,A3,A3_3);
605 - BCU2_3=subs(BCU2_3,alfa,sol_alfa(3));
606 - BCU2_4=subs(BCU2,A3,A3_4);
607 - BCU2_4=subs(BCU2_4,alfa,sol_alfa(4));
608
609
610 - BCU3_1=subs(BCU3,A3,A3_1);
611 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
612 - BCU3_2=subs(BCU3,A3,A3_2);
613 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
614 - BCU3_3=subs(BCU3,A3,A3_3);
615 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
616 - BCU3_4=subs(BCU3,A3,A3_4);
617 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
618
619 %ottengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
620 - eq_BCU2=BCU2_1+BCU2_2+BCU2_3+BCU2_4; %u1
621 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4; %u3
622
623 %sostituisco il valore di x3 agli estremi
624 - eq_BCU2_x_min=subs(eq_BCU2,x3,x_min);
625 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
626 - eq_BCU2_x_max=subs(eq_BCU2,x3,x_max);
627 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
628
629 %Metto tutto in una matrice dove la colonna indica lo strato

```

```

630 -     BCMatr(1,1)=eq_BCS2_x_min; %sigma5 in x_min
631 -     BCMatr(2,1)=eq_BCS3_x_min; %sigma3 in x_min
632 -     BCMatr(3,1)=eq_BCS2_x_max; %sigma5 in x_max
633 -     BCMatr(4,1)=eq_BCS3_x_max; %sigma3 in x_max
634 -     BCMatr(5,1)=eq_BCU2_x_min; %u1 in x_min
635 -     BCMatr(6,1)=eq_BCU3_x_min; %u3 in x_min
636 -     BCMatr(7,1)=eq_BCU2_x_max; %u1 in x_max
637 -     BCMatr(8,1)=eq_BCU3_x_max; %u3 in x_max
638 -
639 -     end
640 -
641 - else
642 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% caso generale K1 e K2 diversi da 0
643 -
644 - %trasformazione da sistema a matrice e soluzioni
645 - [B]=equationsToMatrix([eq1==0, eq2==0, eq3==0],[A1,A2,A3]);
646 - sol_alfa=solve(det(B)==0, alfa, 'IgnoreAnalyticConstraints', true);
647 - sol_alfa=simplify(sol_alfa);
648 - ALFA(:,1)=sol_alfa;
649 -
650 - %espressione dei rapporti tra le ampiezze
651 - amp=B*[A1;A2;A3];
652 - amp1=[0 1 0]*amp;
653 - amp2=[0 0 1]*amp;
654 - sol_A1=solve(amp1==0,A1);
655 - amp2=simplify(subs(amp2,A1,sol_A1));
656 - sol_A3=solve(amp2==0,A3);
657 - eta2=simplifyFraction(sol_A3/A2); %eta2=A3/A2
658 - eta1=simplifyFraction(subs(sol_A1,A3,sol_A3)/A2); %eta1=A1/A2
659 -
660 - rif(1)=2;
661 -
662 - %verifico la non singolarità di eta1 e eta2 per le soluzioni di alfa
663 - [num1,den1]=numden(eta1);
664 - J=zeros(1,6);
665 - for index=1:6
666 -     test=subs(den1, alfa, sol_alfa(index));
667 -     J(1,index)=isequaln(test,0);
668 - end
669 - if sum(J)==0
670 -     [num2,den2]=numden(eta2);
671 -     J=zeros(1,6);
672 -     for index=1:6
673 -         test=subs(den2, alfa, sol_alfa(index));
674 -         J(1,index)=isequaln(test,0);
675 -     end
676 -     if sum(J)==0
677 -         riferimento='Il riferimento è A2';
678 -         rif(1)=2;
679 -     else
680 -         eta2=num2/den2;
681 -         eta2=simplifyFraction(1/eta2); %eta2=A2/A3
682 -         eta1=simplifyFraction(eta1/eta2); %eta1=A1/A3
683 -         riferimento='Il riferimento è A3';
684 -         rif(1)=3;
685 -     end
686 - else
687 -     eta1=num1/den1;
688 -     eta1=simplifyFraction(1/eta1); %eta1=A2/A1
689 -     eta2=simplifyFraction(eta2/eta1); %eta2=A3/A1

```

```

690 -     riferimento='Il riferimento è A1';
691 -     rif(1)=1;
692 - end
693
694 - ETA(1,1)=eta1;
695 - ETA(2,1)=eta2;
696 - disp(riferimento)
697
698 %condizioni al contorno
699
700 - if rif(1)==2
701
702 %esprimo sigma5, sigma4 e sigma3 solo in funzione di A2 usando eta1 e eta2
703 - BCS1=subs(S5,A1,(A2*eta1));
704 - BCS1=subs(BCS1,A3,(A2*eta2));
705 - BCS2=subs(S4,A1,(A2*eta1));
706 - BCS2=subs(BCS2,A3,(A2*eta2));
707 - BCS3=subs(S3,A1,(A2*eta1));
708 - BCS3=subs(BCS3,A3,(A2*eta2));
709 - BCS1=simplifyFraction(BCS1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
710 - BCS2=simplifyFraction(BCS2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
711 - BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
712
713 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
714 - BCS1_1=subs(BCS1,A2,A2_1);
715 - BCS1_1=subs(BCS1_1,alfa,sol_alfa(1));
716 - BCS1_2=subs(BCS1,A2,A2_2);
717 - BCS1_2=subs(BCS1_2,alfa,sol_alfa(2));
718 - BCS1_3=subs(BCS1,A2,A2_3);
719 - BCS1_3=subs(BCS1_3,alfa,sol_alfa(3));
720 - BCS1_4=subs(BCS1,A2,A2_4);
721 - BCS1_4=subs(BCS1_4,alfa,sol_alfa(4));
722 - BCS1_5=subs(BCS1,A2,A2_5);
723 - BCS1_5=subs(BCS1_5,alfa,sol_alfa(5));
724 - BCS1_6=subs(BCS1,A2,A2_6);
725 - BCS1_6=subs(BCS1_6,alfa,sol_alfa(6));
726
727 - BCS2_1=subs(BCS2,A2,A2_1);
728 - BCS2_1=subs(BCS2_1,alfa,sol_alfa(1));
729 - BCS2_2=subs(BCS2,A2,A2_2);
730 - BCS2_2=subs(BCS2_2,alfa,sol_alfa(2));
731 - BCS2_3=subs(BCS2,A2,A2_3);
732 - BCS2_3=subs(BCS2_3,alfa,sol_alfa(3));
733 - BCS2_4=subs(BCS2,A2,A2_4);
734 - BCS2_4=subs(BCS2_4,alfa,sol_alfa(4));
735 - BCS2_5=subs(BCS2,A2,A2_5);
736 - BCS2_5=subs(BCS2_5,alfa,sol_alfa(5));
737 - BCS2_6=subs(BCS2,A2,A2_6);
738 - BCS2_6=subs(BCS2_6,alfa,sol_alfa(6));
739
740 - BCS3_1=subs(BCS3,A2,A2_1);
741 - BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
742 - BCS3_2=subs(BCS3,A2,A2_2);
743 - BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
744 - BCS3_3=subs(BCS3,A2,A2_3);
745 - BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
746 - BCS3_4=subs(BCS3,A2,A2_4);
747 - BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
748 - BCS3_5=subs(BCS3,A2,A2_5);
749 - BCS3_5=subs(BCS3_5,alfa,sol_alfa(5));

```

```

750 - BCS3_6=subs(BCS3,A2,A2_6);
751 - BCS3_6=subs(BCS3_6,alfa,sol_alfa(6));
752
753 %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
754 - eq_BCS1=BCS1_1+BCS1_2+BCS1_3+BCS1_4+BCS1_5+BCS1_6; %sigma5
755 - eq_BCS2=BCS2_1+BCS2_2+BCS2_3+BCS2_4+BCS2_5+BCS2_6; %sigma4
756 - eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4+BCS3_5+BCS3_6; %sigma3
757
758 %sostituisco il valore di x3 agli estremi
759 - eq_BCS1_x_min=subs(eq_BCS1,x3,x_min);
760 - eq_BCS2_x_min=subs(eq_BCS2,x3,x_min);
761 - eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
762 - eq_BCS1_x_max=subs(eq_BCS1,x3,x_max);
763 - eq_BCS2_x_max=subs(eq_BCS2,x3,x_max);
764 - eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
765
766 %esprimo gli spostamenti solo in funzione di A2 usando etal e eta2
767 - BCU1=subs(u1,A1,(A2*etal));
768 - BCU2=u2;
769 - BCU3=subs(u3,A3,(A2*eta2));
770 - BCU1=simplifyFraction(BCU1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
771 - BCU2=simplifyFraction(BCU2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
772 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
773
774 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
775 - BCU1_1=subs(BCU1,A2,A2_1);
776 - BCU1_1=subs(BCU1_1,alfa,sol_alfa(1));
777 - BCU1_2=subs(BCU1,A2,A2_2);
778 - BCU1_2=subs(BCU1_2,alfa,sol_alfa(2));
779 - BCU1_3=subs(BCU1,A2,A2_3);
780 - BCU1_3=subs(BCU1_3,alfa,sol_alfa(3));
781 - BCU1_4=subs(BCU1,A2,A2_4);
782 - BCU1_4=subs(BCU1_4,alfa,sol_alfa(4));
783 - BCU1_5=subs(BCU1,A2,A2_5);
784 - BCU1_5=subs(BCU1_5,alfa,sol_alfa(5));
785 - BCU1_6=subs(BCU1,A2,A2_6);
786 - BCU1_6=subs(BCU1_6,alfa,sol_alfa(6));
787
788 - BCU2_1=subs(BCU2,A2,A2_1);
789 - BCU2_1=subs(BCU2_1,alfa,sol_alfa(1));
790 - BCU2_2=subs(BCU2,A2,A2_2);
791 - BCU2_2=subs(BCU2_2,alfa,sol_alfa(2));
792 - BCU2_3=subs(BCU2,A2,A2_3);
793 - BCU2_3=subs(BCU2_3,alfa,sol_alfa(3));
794 - BCU2_4=subs(BCU2,A2,A2_4);
795 - BCU2_4=subs(BCU2_4,alfa,sol_alfa(4));
796 - BCU2_5=subs(BCU2,A2,A2_5);
797 - BCU2_5=subs(BCU2_5,alfa,sol_alfa(5));
798 - BCU2_6=subs(BCU2,A2,A2_6);
799 - BCU2_6=subs(BCU2_6,alfa,sol_alfa(6));
800
801 - BCU3_1=subs(BCU3,A2,A2_1);
802 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
803 - BCU3_2=subs(BCU3,A2,A2_2);
804 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
805 - BCU3_3=subs(BCU3,A2,A2_3);
806 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
807 - BCU3_4=subs(BCU3,A2,A2_4);
808 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
809 - BCU3_5=subs(BCU3,A2,A2_5);

```

```

810 - BCU3_5=subs(BCU3_5,alfa,sol_alfa(5));
811 - BCU3_6=subs(BCU3,A2,A2_6);
812 - BCU3_6=subs(BCU3_6,alfa,sol_alfa(6));
813
814 %tottingo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
815 - eq_BCU1=BCU1_1+BCU1_2+BCU1_3+BCU1_4+BCU1_5+BCU1_6; %u1
816 - eq_BCU2=BCU2_1+BCU2_2+BCU2_3+BCU2_4+BCU2_5+BCU2_6; %u2
817 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4+BCU3_5+BCU3_6; %u3
818
819 %sostituisco il valore di x3 agli estremi
820 - eq_BCU1_x_min=subs(eq_BCU1,x3,x_min);
821 - eq_BCU2_x_min=subs(eq_BCU2,x3,x_min);
822 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
823 - eq_BCU1_x_max=subs(eq_BCU1,x3,x_max);
824 - eq_BCU2_x_max=subs(eq_BCU2,x3,x_max);
825 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
826
827 %Metto tutto in una matrice dove la colonna indica lo strato
828 - BCMatr(1,1)=eq_BCS1_x_min; %sigma5 in x_min
829 - BCMatr(2,1)=eq_BCS2_x_min; %sigma4 in x_min
830 - BCMatr(3,1)=eq_BCS3_x_min; %sigma3 in x_min
831 - BCMatr(4,1)=eq_BCS1_x_max; %sigma5 in x_max
832 - BCMatr(5,1)=eq_BCS2_x_max; %sigma4 in x_max
833 - BCMatr(6,1)=eq_BCS3_x_max; %sigma3 in x_max
834 - BCMatr(7,1)=eq_BCU1_x_min; %u1 in x_min
835 - BCMatr(8,1)=eq_BCU2_x_min; %u2 in x_min
836 - BCMatr(9,1)=eq_BCU3_x_min; %u3 in x_min
837 - BCMatr(10,1)=eq_BCU1_x_max; %u1 in x_max
838 - BCMatr(11,1)=eq_BCU2_x_max; %u2 in x_max
839 - BCMatr(12,1)=eq_BCU3_x_max; %u3 in x_max
840
841 - elseif rif(1)==1
842
843 %esprimo sigma5,sigma4 e sigma3 solo in funzione di A1 usando eta1 e eta2
844 - BCS1=subs(S5,A2,(A1*eta1));
845 - BCS1=subs(BCS1,A3,(A1*eta2));
846 - BCS2=subs(S4,A2,(A1*eta1));
847 - BCS2=subs(BCS2,A3,(A1*eta2));
848 - BCS3=subs(S3,A2,(A1*eta1));
849 - BCS3=subs(BCS3,A3,(A1*eta2));
850 - BCS1=simplifyFraction(BCS1/(exp(li*(q_1*x1+q_2*x2-w*t))));
851 - BCS2=simplifyFraction(BCS2/(exp(li*(q_1*x1+q_2*x2-w*t))));
852 - BCS3=simplifyFraction(BCS3/(exp(li*(q_1*x1+q_2*x2-w*t))));
853
854 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
855 - BCS1_1=subs(BCS1,A1,A1_1);
856 - BCS1_1=subs(BCS1_1,alfa,sol_alfa(1));
857 - BCS1_2=subs(BCS1,A1,A1_2);
858 - BCS1_2=subs(BCS1_2,alfa,sol_alfa(2));
859 - BCS1_3=subs(BCS1,A1,A1_3);
860 - BCS1_3=subs(BCS1_3,alfa,sol_alfa(3));
861 - BCS1_4=subs(BCS1,A1,A1_4);
862 - BCS1_4=subs(BCS1_4,alfa,sol_alfa(4));
863 - BCS1_5=subs(BCS1,A1,A1_5);
864 - BCS1_5=subs(BCS1_5,alfa,sol_alfa(5));
865 - BCS1_6=subs(BCS1,A1,A1_6);
866 - BCS1_6=subs(BCS1_6,alfa,sol_alfa(6));
867
868 - BCS2_1=subs(BCS2,A1,A1_1);
869 - BCS2_1=subs(BCS2_1,alfa,sol_alfa(1));

```

```

870 - BCS2_2=subs(BCS2,A1,A1_2);
871 - BCS2_2=subs(BCS2_2,alfa,sol_alfa(2));
872 - BCS2_3=subs(BCS2,A1,A1_3);
873 - BCS2_3=subs(BCS2_3,alfa,sol_alfa(3));
874 - BCS2_4=subs(BCS2,A1,A1_4);
875 - BCS2_4=subs(BCS2_4,alfa,sol_alfa(4));
876 - BCS2_5=subs(BCS2,A1,A1_5);
877 - BCS2_5=subs(BCS2_5,alfa,sol_alfa(5));
878 - BCS2_6=subs(BCS2,A1,A1_6);
879 - BCS2_6=subs(BCS2_6,alfa,sol_alfa(6));
880
881 - BCS3_1=subs(BCS3,A1,A1_1);
882 - BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
883 - BCS3_2=subs(BCS3,A1,A1_2);
884 - BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
885 - BCS3_3=subs(BCS3,A1,A1_3);
886 - BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
887 - BCS3_4=subs(BCS3,A1,A1_4);
888 - BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
889 - BCS3_5=subs(BCS3,A1,A1_5);
890 - BCS3_5=subs(BCS3_5,alfa,sol_alfa(5));
891 - BCS3_6=subs(BCS3,A1,A1_6);
892 - BCS3_6=subs(BCS3_6,alfa,sol_alfa(6));
893
894 %totengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
895 - eq_BCS1=BCS1_1+BCS1_2+BCS1_3+BCS1_4+BCS1_5+BCS1_6; %sigma5
896 - eq_BCS2=BCS2_1+BCS2_2+BCS2_3+BCS2_4+BCS2_5+BCS2_6; %sigma4
897 - eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4+BCS3_5+BCS3_6; %sigma3
898
899 %sostituisco il valore di x3 agli estremi
900 - eq_BCS1_x_min=subs(eq_BCS1,x3,x_min);
901 - eq_BCS2_x_min=subs(eq_BCS2,x3,x_min);
902 - eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
903 - eq_BCS1_x_max=subs(eq_BCS1,x3,x_max);
904 - eq_BCS2_x_max=subs(eq_BCS2,x3,x_max);
905 - eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
906
907 %esprimo gli spostamenti solo in funzione di A1 usando eta1 e eta2
908 - BCU1=u1;
909 - BCU2=subs(u2,A2,(A1*eta1));
910 - BCU3=subs(u3,A3,(A1*eta2));
911 - BCU1=simplifyFraction(BCU1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
912 - BCU2=simplifyFraction(BCU2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
913 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
914
915 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
916 - BCU1_1=subs(BCU1,A1,A1_1);
917 - BCU1_1=subs(BCU1_1,alfa,sol_alfa(1));
918 - BCU1_2=subs(BCU1,A1,A1_2);
919 - BCU1_2=subs(BCU1_2,alfa,sol_alfa(2));
920 - BCU1_3=subs(BCU1,A1,A1_3);
921 - BCU1_3=subs(BCU1_3,alfa,sol_alfa(3));
922 - BCU1_4=subs(BCU1,A1,A1_4);
923 - BCU1_4=subs(BCU1_4,alfa,sol_alfa(4));
924 - BCU1_5=subs(BCU1,A1,A1_5);
925 - BCU1_5=subs(BCU1_5,alfa,sol_alfa(5));
926 - BCU1_6=subs(BCU1,A1,A1_6);
927 - BCU1_6=subs(BCU1_6,alfa,sol_alfa(6));
928
929 - BCU2_1=subs(BCU2,A1,A1_1);

```

```

930 - BCU2_1=subs(BCU2_1,alfa,sol_alfa(1));
931 - BCU2_2=subs(BCU2,A1,A1_2);
932 - BCU2_2=subs(BCU2_2,alfa,sol_alfa(2));
933 - BCU2_3=subs(BCU2,A1,A1_3);
934 - BCU2_3=subs(BCU2_3,alfa,sol_alfa(3));
935 - BCU2_4=subs(BCU2,A1,A1_4);
936 - BCU2_4=subs(BCU2_4,alfa,sol_alfa(4));
937 - BCU2_5=subs(BCU2,A1,A1_5);
938 - BCU2_5=subs(BCU2_5,alfa,sol_alfa(5));
939 - BCU2_6=subs(BCU2,A1,A1_6);
940 - BCU2_6=subs(BCU2_6,alfa,sol_alfa(6));
941
942 - BCU3_1=subs(BCU3,A1,A1_1);
943 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
944 - BCU3_2=subs(BCU3,A1,A1_2);
945 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
946 - BCU3_3=subs(BCU3,A1,A1_3);
947 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
948 - BCU3_4=subs(BCU3,A1,A1_4);
949 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
950 - BCU3_5=subs(BCU3,A1,A1_5);
951 - BCU3_5=subs(BCU3_5,alfa,sol_alfa(5));
952 - BCU3_6=subs(BCU3,A1,A1_6);
953 - BCU3_6=subs(BCU3_6,alfa,sol_alfa(6));
954
955 %totengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
956 - eq_BCU1=BCU1_1+BCU1_2+BCU1_3+BCU1_4+BCU1_5+BCU1_6; %u1
957 - eq_BCU2=BCU2_1+BCU2_2+BCU2_3+BCU2_4+BCU2_5+BCU2_6; %u2
958 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4+BCU3_5+BCU3_6; %u3
959
960 %sostituisco il valore di x3 agli estremi
961 - eq_BCU1_x_min=subs(eq_BCU1,x3,x_min);
962 - eq_BCU2_x_min=subs(eq_BCU2,x3,x_min);
963 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
964 - eq_BCU1_x_max=subs(eq_BCU1,x3,x_max);
965 - eq_BCU2_x_max=subs(eq_BCU2,x3,x_max);
966 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
967
968 %Metto tutto in una matrice dove la colonna indica lo strato
969 - BCMatr(1,1)=eq_BCS1_x_min; %sigma5 in x_min
970 - BCMatr(2,1)=eq_BCS2_x_min; %sigma4 in x_min
971 - BCMatr(3,1)=eq_BCS3_x_min; %sigma3 in x_min
972 - BCMatr(4,1)=eq_BCS1_x_max; %sigma5 in x_max
973 - BCMatr(5,1)=eq_BCS2_x_max; %sigma4 in x_max
974 - BCMatr(6,1)=eq_BCS3_x_max; %sigma3 in x_max
975 - BCMatr(7,1)=eq_BCU1_x_min; %u1 in x_min
976 - BCMatr(8,1)=eq_BCU2_x_min; %u2 in x_min
977 - BCMatr(9,1)=eq_BCU3_x_min; %u3 in x_min
978 - BCMatr(10,1)=eq_BCU1_x_max; %u1 in x_max
979 - BCMatr(11,1)=eq_BCU2_x_max; %u2 in x_max
980 - BCMatr(12,1)=eq_BCU3_x_max; %u3 in x_max
981
982 - elseif rif(1)==3
983
984 %esprimo sigma5,sigma4 e sigma3 solo in funzione di A3 usando eta1 e eta2
985 - BCS1=subs(S5,A1,(A3*eta1));
986 - BCS1=subs(BCS1,A2,(A3*eta2));
987 - BCS2=subs(S4,A1,(A3*eta1));
988 - BCS2=subs(BCS2,A2,(A3*eta2));
989 - BCS3=subs(S3,A1,(A3*eta1));

```

```

990 - BCS3=subs(BCS3,A2,(A3*eta2));
991 - BCS1=simplifyFraction(BCS1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
992 - BCS2=simplifyFraction(BCS2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
993 - BCS3=simplifyFraction(BCS3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
994
995 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
996 - BCS1_1=subs(BCS1,A3,A3_1);
997 - BCS1_1=subs(BCS1_1,alfa,sol_alfa(1));
998 - BCS1_2=subs(BCS1,A3,A3_2);
999 - BCS1_2=subs(BCS1_2,alfa,sol_alfa(2));
1000 - BCS1_3=subs(BCS1,A3,A3_3);
1001 - BCS1_3=subs(BCS1_3,alfa,sol_alfa(3));
1002 - BCS1_4=subs(BCS1,A3,A3_4);
1003 - BCS1_4=subs(BCS1_4,alfa,sol_alfa(4));
1004 - BCS1_5=subs(BCS1,A3,A3_5);
1005 - BCS1_5=subs(BCS1_5,alfa,sol_alfa(5));
1006 - BCS1_6=subs(BCS1,A3,A3_6);
1007 - BCS1_6=subs(BCS1_6,alfa,sol_alfa(6));
1008
1009 - BCS2_1=subs(BCS2,A3,A3_1);
1010 - BCS2_1=subs(BCS2_1,alfa,sol_alfa(1));
1011 - BCS2_2=subs(BCS2,A3,A3_2);
1012 - BCS2_2=subs(BCS2_2,alfa,sol_alfa(2));
1013 - BCS2_3=subs(BCS2,A3,A3_3);
1014 - BCS2_3=subs(BCS2_3,alfa,sol_alfa(3));
1015 - BCS2_4=subs(BCS2,A3,A3_4);
1016 - BCS2_4=subs(BCS2_4,alfa,sol_alfa(4));
1017 - BCS2_5=subs(BCS2,A3,A3_5);
1018 - BCS2_5=subs(BCS2_5,alfa,sol_alfa(5));
1019 - BCS2_6=subs(BCS2,A3,A3_6);
1020 - BCS2_6=subs(BCS2_6,alfa,sol_alfa(6));
1021
1022 - BCS3_1=subs(BCS3,A3,A3_1);
1023 - BCS3_1=subs(BCS3_1,alfa,sol_alfa(1));
1024 - BCS3_2=subs(BCS3,A3,A3_2);
1025 - BCS3_2=subs(BCS3_2,alfa,sol_alfa(2));
1026 - BCS3_3=subs(BCS3,A3,A3_3);
1027 - BCS3_3=subs(BCS3_3,alfa,sol_alfa(3));
1028 - BCS3_4=subs(BCS3,A3,A3_4);
1029 - BCS3_4=subs(BCS3_4,alfa,sol_alfa(4));
1030 - BCS3_5=subs(BCS3,A3,A3_5);
1031 - BCS3_5=subs(BCS3_5,alfa,sol_alfa(5));
1032 - BCS3_6=subs(BCS3,A3,A3_6);
1033 - BCS3_6=subs(BCS3_6,alfa,sol_alfa(6));
1034
1035 %ottengo le condizioni al contorno degli sforzi come sommatoria rispetto a tutte le soluzioni di alfa
1036 - eq_BCS1=BCS1_1+BCS1_2+BCS1_3+BCS1_4+BCS1_5+BCS1_6; %sigma5
1037 - eq_BCS2=BCS2_1+BCS2_2+BCS2_3+BCS2_4+BCS2_5+BCS2_6; %sigma4
1038 - eq_BCS3=BCS3_1+BCS3_2+BCS3_3+BCS3_4+BCS3_5+BCS3_6; %sigma3
1039
1040 %sostituisco il valore di x3 agli estremi
1041 - eq_BCS1_x_min=subs(eq_BCS1,x3,x_min);
1042 - eq_BCS2_x_min=subs(eq_BCS2,x3,x_min);
1043 - eq_BCS3_x_min=subs(eq_BCS3,x3,x_min);
1044 - eq_BCS1_x_max=subs(eq_BCS1,x3,x_max);
1045 - eq_BCS2_x_max=subs(eq_BCS2,x3,x_max);
1046 - eq_BCS3_x_max=subs(eq_BCS3,x3,x_max);
1047
1048 %esprimo gli spostamenti solo in funzione di A3 usando etal e eta2
1049 - BCUI=subs(u1,A1,(A3*etal));

```



```

1050 - BCU2=subs(u2,A2, (A3*eta2));
1051 - BCU3=u3;
1052 - BCU1=simplifyFraction(BCU1/(exp(1i*(q_1*x1+q_2*x2-w*t))));
1053 - BCU2=simplifyFraction(BCU2/(exp(1i*(q_1*x1+q_2*x2-w*t))));
1054 - BCU3=simplifyFraction(BCU3/(exp(1i*(q_1*x1+q_2*x2-w*t))));
1055
1056 %sostituisco tutte le soluzioni di alfa nelle condizioni al contorno
1057 - BCU1_1=subs(BCU1,A3,A3_1);
1058 - BCU1_1=subs(BCU1_1,alfa,sol_alfa(1));
1059 - BCU1_2=subs(BCU1,A3,A3_2);
1060 - BCU1_2=subs(BCU1_2,alfa,sol_alfa(2));
1061 - BCU1_3=subs(BCU1,A3,A3_3);
1062 - BCU1_3=subs(BCU1_3,alfa,sol_alfa(3));
1063 - BCU1_4=subs(BCU1,A3,A3_4);
1064 - BCU1_4=subs(BCU1_4,alfa,sol_alfa(4));
1065 - BCU1_5=subs(BCU1,A3,A3_5);
1066 - BCU1_5=subs(BCU1_5,alfa,sol_alfa(5));
1067 - BCU1_6=subs(BCU1,A3,A3_6);
1068 - BCU1_6=subs(BCU1_6,alfa,sol_alfa(6));
1069
1070 - BCU2_1=subs(BCU2,A3,A3_1);
1071 - BCU2_1=subs(BCU2_1,alfa,sol_alfa(1));
1072 - BCU2_2=subs(BCU2,A3,A3_2);
1073 - BCU2_2=subs(BCU2_2,alfa,sol_alfa(2));
1074 - BCU2_3=subs(BCU2,A3,A3_3);
1075 - BCU2_3=subs(BCU2_3,alfa,sol_alfa(3));
1076 - BCU2_4=subs(BCU2,A3,A3_4);
1077 - BCU2_4=subs(BCU2_4,alfa,sol_alfa(4));
1078 - BCU2_5=subs(BCU2,A3,A3_5);
1079 - BCU2_5=subs(BCU2_5,alfa,sol_alfa(5));
1080 - BCU2_6=subs(BCU2,A3,A3_6);
1081 - BCU2_6=subs(BCU2_6,alfa,sol_alfa(6));
1082
1083 - BCU3_1=subs(BCU3,A3,A3_1);
1084 - BCU3_1=subs(BCU3_1,alfa,sol_alfa(1));
1085 - BCU3_2=subs(BCU3,A3,A3_2);
1086 - BCU3_2=subs(BCU3_2,alfa,sol_alfa(2));
1087 - BCU3_3=subs(BCU3,A3,A3_3);
1088 - BCU3_3=subs(BCU3_3,alfa,sol_alfa(3));
1089 - BCU3_4=subs(BCU3,A3,A3_4);
1090 - BCU3_4=subs(BCU3_4,alfa,sol_alfa(4));
1091 - BCU3_5=subs(BCU3,A3,A3_5);
1092 - BCU3_5=subs(BCU3_5,alfa,sol_alfa(5));
1093 - BCU3_6=subs(BCU3,A3,A3_6);
1094 - BCU3_6=subs(BCU3_6,alfa,sol_alfa(6));
1095
1096 %ottengo le condizioni al contorno degli spostamenti come sommatoria rispetto a tutte le soluzioni di alfa
1097 - eq_BCU1=BCU1_1+BCU1_2+BCU1_3+BCU1_4+BCU1_5+BCU1_6; %u1
1098 - eq_BCU2=BCU2_1+BCU2_2+BCU2_3+BCU2_4+BCU2_5+BCU2_6; %u2
1099 - eq_BCU3=BCU3_1+BCU3_2+BCU3_3+BCU3_4+BCU3_5+BCU3_6; %u3
1100
1101 %sostituisco il valore di x3 agli estremi
1102 - eq_BCU1_x_min=subs(eq_BCU1,x3,x_min);
1103 - eq_BCU2_x_min=subs(eq_BCU2,x3,x_min);
1104 - eq_BCU3_x_min=subs(eq_BCU3,x3,x_min);
1105 - eq_BCU1_x_max=subs(eq_BCU1,x3,x_max);
1106 - eq_BCU2_x_max=subs(eq_BCU2,x3,x_max);
1107 - eq_BCU3_x_max=subs(eq_BCU3,x3,x_max);
1108
1109 %Metto tutto in una matrice dove la colonna indica lo strato

```

```

1110 - BCMatr(1,1)=eq_BCS1_x_min; %sigma5 in x_min
1111 - BCMatr(2,1)=eq_BCS2_x_min; %sigma4 in x_min
1112 - BCMatr(3,1)=eq_BCS3_x_min; %sigma3 in x_min
1113 - BCMatr(4,1)=eq_BCS1_x_max; %sigma5 in x_max
1114 - BCMatr(5,1)=eq_BCS2_x_max; %sigma4 in x_max
1115 - BCMatr(6,1)=eq_BCS3_x_max; %sigma3 in x_max
1116 - BCMatr(7,1)=eq_BCU1_x_min; %u1 in x_min
1117 - BCMatr(8,1)=eq_BCU2_x_min; %u2 in x_min
1118 - BCMatr(9,1)=eq_BCU3_x_min; %u3 in x_min
1119 - BCMatr(10,1)=eq_BCU1_x_max; %u1 in x_max
1120 - BCMatr(11,1)=eq_BCU2_x_max; %u2 in x_max
1121 - BCMatr(12,1)=eq_BCU3_x_max; %u3 in x_max
1122
1123 - end
1124
1125 - end
1126 - if q_2==0||q_1==0
1127 -     syms zk;
1128 -     for id1=1:8
1129 -         for id2=1:strati
1130 -             if BCMatr(id1,id2)==0
1131 -                 BCMatr(id1,id2)=zk;
1132 -             end
1133 -         end
1134 -     end
1135 - end
1136
1137 - %%%%%%%%%%%%%%%%%%%%%%%%%composizione della matrice finale delle condizioni al contorno
1138
1139 - %%%%% caso in cui k2=0 o k1=0
1140 - if q_2==0||q_1==0
1141
1142
1143 - if strati==1
1144 -     if rif(1)==3
1145 -         [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1146 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1147 -         ir2_min=ir2_min+1;
1148 -         [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1149 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1150 -         ir2_min=ir2_min+1;
1151 -         [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1152 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1153 -         ir2_min=ir2_min+1;
1154 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1155 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1156
1157 -     elseif rif(1)==1
1158 -         [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1159 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1160 -         ir2_min=ir2_min+1;
1161 -         [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1162 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1163 -         ir2_min=ir2_min+1;
1164 -         [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1165 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1166 -         ir2_min=ir2_min+1;
1167 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1168 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1169

```

```

1170 -     elseif rif(l)==2
1171 -         [matr1]=equationsToMatrix(BCMatr(1,l)==0,[A2_1,A2_2,A2_3,A2_4]);
1172 -         FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1173 -         ir2_min=ir2_min+1;
1174 -         [matr1]=equationsToMatrix(BCMatr(2,l)==0,[A2_1,A2_2,A2_3,A2_4]);
1175 -         FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1176 -         ir2_min=ir2_min+1;
1177 -         [matr1]=equationsToMatrix(BCMatr(3,l)==0,[A2_1,A2_2,A2_3,A2_4]);
1178 -         FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1179 -         ir2_min=ir2_min+1;
1180 -         [matr1]=equationsToMatrix(BCMatr(4,l)==0,[A2_1,A2_2,A2_3,A2_4]);
1181 -         FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1182 -
1183 -         end
1184 -
1185 -     else
1186 -
1187 -     if l==1
1188 -         if rif(l)==3
1189 -             [matr1]=equationsToMatrix(BCMatr(1,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1190 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1191 -             ir2_min=ir2_min+1;
1192 -             [matr1]=equationsToMatrix(BCMatr(2,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1193 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1194 -             ir2_min=ir2_min+1;
1195 -             [matr1]=equationsToMatrix(BCMatr(3,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1196 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1197 -             ir2_min=ir2_min+1;
1198 -             [matr1]=equationsToMatrix(BCMatr(4,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1199 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1200 -             ir2_min=ir2_min+1;
1201 -             [matr1]=equationsToMatrix(BCMatr(7,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1202 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1203 -             ir2_min=ir2_min+1;
1204 -             [matr1]=equationsToMatrix(BCMatr(8,l)==0,[A3_1,A3_2,A3_3,A3_4]);
1205 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1206 -
1207 -
1208 -         elseif rif(l)==1
1209 -             [matr1]=equationsToMatrix(BCMatr(1,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1210 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1211 -             ir2_min=ir2_min+1;
1212 -             [matr1]=equationsToMatrix(BCMatr(2,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1213 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1214 -             ir2_min=ir2_min+1;
1215 -             [matr1]=equationsToMatrix(BCMatr(3,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1216 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1217 -             ir2_min=ir2_min+1;
1218 -             [matr1]=equationsToMatrix(BCMatr(4,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1219 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1220 -             ir2_min=ir2_min+1;
1221 -             [matr1]=equationsToMatrix(BCMatr(7,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1222 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1223 -             ir2_min=ir2_min+1;
1224 -             [matr1]=equationsToMatrix(BCMatr(8,l)==0,[A1_1,A1_2,A1_3,A1_4]);
1225 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;
1226 -
1227 -         elseif rif(l)==2
1228 -             [matr1]=equationsToMatrix(BCMatr(1,l)==0,[A2_1,A2_2,A2_3,A2_4]);
1229 -             FinM(ir2_min,(1+4*(l-1)):1*4)=matr1;

```

```

1230 -     ir2_min=ir2_min+1;
1231 -     [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A2_1,A2_2,A2_3,A2_4]);
1232 -     FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1233 -     ir2_min=ir2_min+1;
1234 -     [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A2_1,A2_2,A2_3,A2_4]);
1235 -     FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1236 -     ir2_min=ir2_min+1;
1237 -     [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A2_1,A2_2,A2_3,A2_4]);
1238 -     FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1239 -     ir2_min=ir2_min+1;
1240 -     [matr1]=equationsToMatrix(BCMatr(7,1)==0,[A2_1,A2_2,A2_3,A2_4]);
1241 -     FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1242 -     ir2_min=ir2_min+1;
1243 -     [matr1]=equationsToMatrix(BCMatr(8,1)==0,[A2_1,A2_2,A2_3,A2_4]);
1244 -     FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1245 -
1246 -     end
1247 -
1248 - elseif l==strati
1249 -     ir2_min=ir2_min-3;
1250 -     if rif(1)==3
1251 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1252 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1253 -         ir2_min=ir2_min+1;
1254 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1255 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1256 -         ir2_min=ir2_min+1;
1257 -         [matr1]=equationsToMatrix(-BCMatr(5,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1258 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1259 -         ir2_min=ir2_min+1;
1260 -         [matr1]=equationsToMatrix(-BCMatr(6,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1261 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1262 -         ir2_min=ir2_min+1;
1263 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1264 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1265 -         ir2_min=ir2_min+1;
1266 -         [matr1]=equationsToMatrix(-BCMatr(4,1)==0,[A3_1,A3_2,A3_3,A3_4]);
1267 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1268 -
1269 -     elseif rif(1)==1
1270 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1271 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1272 -         ir2_min=ir2_min+1;
1273 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1274 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1275 -         ir2_min=ir2_min+1;
1276 -         [matr1]=equationsToMatrix(-BCMatr(5,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1277 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1278 -         ir2_min=ir2_min+1;
1279 -         [matr1]=equationsToMatrix(-BCMatr(6,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1280 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1281 -         ir2_min=ir2_min+1;
1282 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1283 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1284 -         ir2_min=ir2_min+1;
1285 -         [matr1]=equationsToMatrix(-BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4]);
1286 -         FinM(ir2_min,(1+4*(1-1)):1*4)=matr1;
1287 -
1288 -     elseif rif(1)==2
1289 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A2_1,A2_2,A2_3,A2_4]);

```

```

1290 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1291 - ir2_min=ir2_min+1;
1292 - [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1293 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1294 - ir2_min=ir2_min+1;
1295 - [matr1]=equationsToMatrix(-BCMatr(5,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1296 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1297 - ir2_min=ir2_min+1;
1298 - [matr1]=equationsToMatrix(-BCMatr(6,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1299 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1300 - ir2_min=ir2_min+1;
1301 - [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1302 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1303 - ir2_min=ir2_min+1;
1304 - [matr1]=equationsToMatrix(-BCMatr(4,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1305 - FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1306
1307 -     end
1308
1309 - else
1310 -     ir2_min=ir2_min-3;
1311 -     if rif(1)==3
1312 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1313 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1314 -         ir2_min=ir2_min+1;
1315 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1316 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1317 -         ir2_min=ir2_min+1;
1318 -         [matr1]=equationsToMatrix(-BCMatr(5,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1319 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1320 -         ir2_min=ir2_min+1;
1321 -         [matr1]=equationsToMatrix(-BCMatr(6,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1322 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1323 -         ir2_min=ir2_min+1;
1324 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1325 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1326 -         ir2_min=ir2_min+1;
1327 -         [matr1]=equationsToMatrix(-BCMatr(4,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1328 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1329 -         ir2_min=ir2_min+1;
1330 -         [matr1]=equationsToMatrix(BCMatr(7,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1331 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1332 -         ir2_min=ir2_min+1;
1333 -         [matr1]=equationsToMatrix(BCMatr(8,1)==0, [A3_1,A3_2,A3_3,A3_4]);
1334 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1335
1336 -     elseif rif(1)==1
1337 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1338 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1339 -         ir2_min=ir2_min+1;
1340 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1341 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1342 -         ir2_min=ir2_min+1;
1343 -         [matr1]=equationsToMatrix(-BCMatr(5,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1344 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1345 -         ir2_min=ir2_min+1;
1346 -         [matr1]=equationsToMatrix(-BCMatr(6,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1347 -         FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1348 -         ir2_min=ir2_min+1;
1349 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A1_1,A1_2,A1_3,A1_4]);

```

```

1350 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1351 -     ir2_min=ir2_min+1;
1352 -     [matr1]=equationsToMatrix(-BCMatr(4,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1353 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1354 -     ir2_min=ir2_min+1;
1355 -     [matr1]=equationsToMatrix(BCMatr(7,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1356 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1357 -     ir2_min=ir2_min+1;
1358 -     [matr1]=equationsToMatrix(BCMatr(8,1)==0, [A1_1,A1_2,A1_3,A1_4]);
1359 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1360
1361 -     elseif rif(1)==2
1362 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1363 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1364 -     ir2_min=ir2_min+1;
1365 -     [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1366 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1367 -     ir2_min=ir2_min+1;
1368 -     [matr1]=equationsToMatrix(-BCMatr(5,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1369 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1370 -     ir2_min=ir2_min+1;
1371 -     [matr1]=equationsToMatrix(-BCMatr(6,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1372 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1373 -     ir2_min=ir2_min+1;
1374 -     [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1375 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1376 -     ir2_min=ir2_min+1;
1377 -     [matr1]=equationsToMatrix(-BCMatr(4,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1378 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1379 -     ir2_min=ir2_min+1;
1380 -     [matr1]=equationsToMatrix(BCMatr(7,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1381 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1382 -     ir2_min=ir2_min+1;
1383 -     [matr1]=equationsToMatrix(BCMatr(8,1)==0, [A2_1,A2_2,A2_3,A2_4]);
1384 -     FinM(ir2_min, (1+4*(1-1)):1*4)=matr1;
1385
1386 -     end
1387
1388 - end
1389
1390 - end
1391
1392 - else
1393
1394 -     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%2% caso generale
1395
1396 -     if strati==1
1397 -         if rif(1)==3
1398 -             [matr1]=equationsToMatrix(BCMatr(1,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1399 -             FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1400 -             ir_min=ir_min+1;
1401 -             [matr1]=equationsToMatrix(BCMatr(2,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1402 -             FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1403 -             ir_min=ir_min+1;
1404 -             [matr1]=equationsToMatrix(BCMatr(3,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1405 -             FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1406 -             ir_min=ir_min+1;
1407 -             [matr1]=equationsToMatrix(BCMatr(4,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1408 -             FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1409 -             ir_min=ir_min+1;

```

```

1410 - [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1411 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1412 - ir_min=ir_min+1;
1413 - [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1414 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1415
1416
1417 - elseif rif(1)==2
1418 - [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1419 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1420 - ir_min=ir_min+1;
1421 - [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1422 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1423 - ir_min=ir_min+1;
1424 - [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1425 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1426 - ir_min=ir_min+1;
1427 - [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1428 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1429 - ir_min=ir_min+1;
1430 - [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1431 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1432 - ir_min=ir_min+1;
1433 - [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1434 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1435
1436
1437 - elseif rif(1)==1
1438 - [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1439 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1440 - ir_min=ir_min+1;
1441 - [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1442 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1443 - ir_min=ir_min+1;
1444 - [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1445 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1446 - ir_min=ir_min+1;
1447 - [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1448 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1449 - ir_min=ir_min+1;
1450 - [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1451 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1452 - ir_min=ir_min+1;
1453 - [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1454 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1455
1456 - end
1457
1458 - else
1459
1460 - if l==1
1461 - if rif(1)==3
1462 - [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1463 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1464 - ir_min=ir_min+1;
1465 - [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1466 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1467 - ir_min=ir_min+1;
1468 - [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1469 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;

```

```

1470 -     ir_min=ir_min+1;
1471 -     [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1472 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1473 -     ir_min=ir_min+1;
1474 -     [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1475 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1476 -     ir_min=ir_min+1;
1477 -     [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1478 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1479 -     ir_min=ir_min+1;
1480 -     [matr1]=equationsToMatrix(BCMatr(10,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1481 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1482 -     ir_min=ir_min+1;
1483 -     [matr1]=equationsToMatrix(BCMatr(11,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1484 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1485 -     ir_min=ir_min+1;
1486 -     [matr1]=equationsToMatrix(BCMatr(12,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1487 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1488
1489
1490 -     elseif rif(1)==2
1491 -         [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1492 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1493 -         ir_min=ir_min+1;
1494 -         [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1495 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1496 -         ir_min=ir_min+1;
1497 -         [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1498 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1499 -         ir_min=ir_min+1;
1500 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1501 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1502 -         ir_min=ir_min+1;
1503 -         [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1504 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1505 -         ir_min=ir_min+1;
1506 -         [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1507 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1508 -         ir_min=ir_min+1;
1509 -         [matr1]=equationsToMatrix(BCMatr(10,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1510 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1511 -         ir_min=ir_min+1;
1512 -         [matr1]=equationsToMatrix(BCMatr(11,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1513 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1514 -         ir_min=ir_min+1;
1515 -         [matr1]=equationsToMatrix(BCMatr(12,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1516 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1517
1518
1519 -     elseif rif(1)==1
1520 -         [matr1]=equationsToMatrix(BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1521 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1522 -         ir_min=ir_min+1;
1523 -         [matr1]=equationsToMatrix(BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1524 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1525 -         ir_min=ir_min+1;
1526 -         [matr1]=equationsToMatrix(BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1527 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1528 -         ir_min=ir_min+1;
1529 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);

```



```

1530 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1531 -     ir_min=ir_min+1;
1532 -     [matr1]=equationsToMatrix(BCMatr(5,1)==0, [A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1533 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1534 -     ir_min=ir_min+1;
1535 -     [matr1]=equationsToMatrix(BCMatr(6,1)==0, [A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1536 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1537 -     ir_min=ir_min+1;
1538 -     [matr1]=equationsToMatrix(BCMatr(10,1)==0, [A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1539 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1540 -     ir_min=ir_min+1;
1541 -     [matr1]=equationsToMatrix(BCMatr(11,1)==0, [A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1542 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1543 -     ir_min=ir_min+1;
1544 -     [matr1]=equationsToMatrix(BCMatr(12,1)==0, [A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1545 -     FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1546 -
1547 -     end
1548 -
1549 - elseif l==strati
1550 -     ir_min=ir_min-5;
1551 -     if rif(1)==3
1552 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1553 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1554 -         ir_min=ir_min+1;
1555 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1556 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1557 -         ir_min=ir_min+1;
1558 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1559 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1560 -         ir_min=ir_min+1;
1561 -         [matr1]=equationsToMatrix(-BCMatr(7,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1562 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1563 -         ir_min=ir_min+1;
1564 -         [matr1]=equationsToMatrix(-BCMatr(8,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1565 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1566 -         ir_min=ir_min+1;
1567 -         [matr1]=equationsToMatrix(-BCMatr(9,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1568 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1569 -         ir_min=ir_min+1;
1570 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1571 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1572 -         ir_min=ir_min+1;
1573 -         [matr1]=equationsToMatrix(BCMatr(5,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1574 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1575 -         ir_min=ir_min+1;
1576 -         [matr1]=equationsToMatrix(BCMatr(6,1)==0, [A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1577 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1578 -
1579 -
1580 -     elseif rif(1)==2
1581 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0, [A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1582 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1583 -         ir_min=ir_min+1;
1584 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0, [A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1585 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1586 -         ir_min=ir_min+1;
1587 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0, [A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1588 -         FinM(ir_min, (1+6*(1-1)):1*6)=matr1;
1589 -         ir_min=ir_min+1;

```

```

1590 - [matr1]=equationsToMatrix(-BCMatr(7,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1591 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1592 - ir_min=ir_min+1;
1593 - [matr1]=equationsToMatrix(-BCMatr(8,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1594 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1595 - ir_min=ir_min+1;
1596 - [matr1]=equationsToMatrix(-BCMatr(9,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1597 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1598 - ir_min=ir_min+1;
1599 - [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1600 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1601 - ir_min=ir_min+1;
1602 - [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1603 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1604 - ir_min=ir_min+1;
1605 - [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1606 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1607
1608
1609 - elseif rif(1)==1
1610 -     [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1611 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1612 - ir_min=ir_min+1;
1613 - [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1614 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1615 - ir_min=ir_min+1;
1616 - [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1617 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1618 - ir_min=ir_min+1;
1619 - [matr1]=equationsToMatrix(-BCMatr(7,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1620 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1621 - ir_min=ir_min+1;
1622 - [matr1]=equationsToMatrix(-BCMatr(8,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1623 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1624 - ir_min=ir_min+1;
1625 - [matr1]=equationsToMatrix(-BCMatr(9,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1626 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1627 - ir_min=ir_min+1;
1628 - [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1629 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1630 - ir_min=ir_min+1;
1631 - [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1632 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1633 - ir_min=ir_min+1;
1634 - [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1635 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1636
1637 - end
1638
1639 - else
1640 -     ir_min=ir_min-5;
1641 -     if rif(1)==3
1642 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1643 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1644 -         ir_min=ir_min+1;
1645 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1646 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1647 -         ir_min=ir_min+1;
1648 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1649 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;

```

```

1650 -     ir_min=ir_min+1;
1651 -     [matr1]=equationsToMatrix(-BCMatr(7,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1652 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1653 -     ir_min=ir_min+1;
1654 -     [matr1]=equationsToMatrix(-BCMatr(8,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1655 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1656 -     ir_min=ir_min+1;
1657 -     [matr1]=equationsToMatrix(-BCMatr(9,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1658 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1659 -     ir_min=ir_min+1;
1660 -     [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1661 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1662 -     ir_min=ir_min+1;
1663 -     [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1664 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1665 -     ir_min=ir_min+1;
1666 -     [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1667 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1668 -     ir_min=ir_min+1;
1669 -     [matr1]=equationsToMatrix(BCMatr(10,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1670 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1671 -     ir_min=ir_min+1;
1672 -     [matr1]=equationsToMatrix(BCMatr(11,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1673 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1674 -     ir_min=ir_min+1;
1675 -     [matr1]=equationsToMatrix(BCMatr(12,1)==0,[A3_1,A3_2,A3_3,A3_4,A3_5,A3_6]);
1676 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1677 -
1678 -
1679 -     elseif rif(1)==2
1680 -         [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1681 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1682 -         ir_min=ir_min+1;
1683 -         [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1684 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1685 -         ir_min=ir_min+1;
1686 -         [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1687 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1688 -         ir_min=ir_min+1;
1689 -         [matr1]=equationsToMatrix(-BCMatr(7,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1690 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1691 -         ir_min=ir_min+1;
1692 -         [matr1]=equationsToMatrix(-BCMatr(8,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1693 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1694 -         ir_min=ir_min+1;
1695 -         [matr1]=equationsToMatrix(-BCMatr(9,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1696 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1697 -         ir_min=ir_min+1;
1698 -         [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1699 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1700 -         ir_min=ir_min+1;
1701 -         [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1702 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1703 -         ir_min=ir_min+1;
1704 -         [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1705 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1706 -         ir_min=ir_min+1;
1707 -         [matr1]=equationsToMatrix(BCMatr(10,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1708 -         FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1709 -         ir_min=ir_min+1;

```

```

1710 - [matr1]=equationsToMatrix(BCMatr(11,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1711 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1712 - ir_min=ir_min+1;
1713 - [matr1]=equationsToMatrix(BCMatr(12,1)==0,[A2_1,A2_2,A2_3,A2_4,A2_5,A2_6]);
1714 - FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1715
1716
1717 - elseif rif(1)==1
1718 -     [matr1]=equationsToMatrix(-BCMatr(1,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1719 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1720 -     ir_min=ir_min+1;
1721 -     [matr1]=equationsToMatrix(-BCMatr(2,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1722 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1723 -     ir_min=ir_min+1;
1724 -     [matr1]=equationsToMatrix(-BCMatr(3,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1725 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1726 -     ir_min=ir_min+1;
1727 -     [matr1]=equationsToMatrix(-BCMatr(7,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1728 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1729 -     ir_min=ir_min+1;
1730 -     [matr1]=equationsToMatrix(-BCMatr(8,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1731 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1732 -     ir_min=ir_min+1;
1733 -     [matr1]=equationsToMatrix(-BCMatr(9,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1734 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1735 -     ir_min=ir_min+1;
1736 -     [matr1]=equationsToMatrix(BCMatr(4,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1737 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1738 -     ir_min=ir_min+1;
1739 -     [matr1]=equationsToMatrix(BCMatr(5,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1740 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1741 -     ir_min=ir_min+1;
1742 -     [matr1]=equationsToMatrix(BCMatr(6,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1743 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1744 -     ir_min=ir_min+1;
1745 -     [matr1]=equationsToMatrix(BCMatr(10,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1746 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1747 -     ir_min=ir_min+1;
1748 -     [matr1]=equationsToMatrix(BCMatr(11,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1749 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1750 -     ir_min=ir_min+1;
1751 -     [matr1]=equationsToMatrix(BCMatr(12,1)==0,[A1_1,A1_2,A1_3,A1_4,A1_5,A1_6]);
1752 -     FinM(ir_min,(1+6*(1-1)):1*6)=matr1;
1753
1754 -     end
1755
1756
1757 - end
1758
1759 - end
1760
1761 - end
1762
1763 - if q_2==0||q_1==0
1764 -     FinM=subs(FinM,zz,0);
1765 -     end
1766
1767 - end
1768
1769 - MM=FinM;

```

```

1770 - p_alf=real(double(subs(ALFA(:,strati),w,20e9)));
1771 - wer=(p_alf<0);
1772 - if sum(wer)==2||sum(wer)==3
1773 -     if q_1==0||q_2==0
1774 -         iji=4;
1775 -     else
1776 -         iji=6;
1777 -     end
1778
1779 - klm=sym('klm',[iji*strati iji]);
1780 - ff=double(subs(ALFA(:,strati),w,20e9));
1781 - klc=zeros(iji,1);
1782 - klc(1)=ff(1);
1783 - for xz=1:iji-1
1784 -     if real(klc(xz))<real(ff(xz+1))
1785 -         klc(xz+1)=klc(xz);
1786 -         klc(xz)=ff(xz+1);
1787 -     elseif real(klc(xz))==real(ff(xz+1)) && imag(klc(xz))<imag(ff(xz+1))
1788 -         klc(xz+1)=klc(xz);
1789 -         klc(xz)=ff(xz+1);
1790 -     else
1791 -         klc(xz+1)=ff(xz+1);
1792 -     end
1793 - end
1794 - vbn=ALFA(:,strati);
1795 - for ii1=1:iji
1796 -     for ii2=1:iji
1797 -         if ff(ii1)==double(subs(klc(ii2),w,20e9))
1798 -             klm(:,ii2)=FinM(:,iji*(strati-1)+ii1);
1799 -             ALFA(ii2,strati)=vbn(ii1,1);
1800 -             ALFA(ii2,strati)=vbn(ii1,1);
1801 -         end
1802 -     end
1803 - end
1804 - FinM(:,iji*(strati-1)+1:iji*strati)=klm;
1805 - MM=FinM;
1806 - if q_1==0||q_2==0
1807 -     FinM=FinM(1:iji*strati-2,1:iji*strati-2);
1808 - else
1809 -     FinM=FinM(1:iji*strati-3,1:iji*strati-3);
1810 - end
1811 - disp(' ')
1812 - disp('matrici MM per mezzo finito e FinM per mezzo semi-infinito pronte all uso')
1813 - disp(' ')
1814
1815 - else
1816 -     disp(' ')
1817 -     disp(p_alf)
1818 -     disp(' ')
1819 -     disp('il mezzo semi infinito non può essere usato')
1820 -     disp('matrice MM per mezzo finito pronta all uso')
1821 - end
1822
1823

```

A.2 Programmi Accessori

```

printdetlog.m x
1  function printdetlog(file,wmin,wmax,step,ms)
2  - load(file)
3  - if ms==2
4  -     matrix=FinM;
5  - else
6  -     matrix=MM;
7  - end
8  - f=1;
9  - clear DET W
10 - numero=(wmax-wmin)/step+1;
11 - DET=zeros(numero,1);
12 - W=zeros(numero,1);
13 - for count=wmin:step:wmax
14 -     prova=subs(matrix,w,count);
15 -     DET(f)=abs(det(double(prova)));
16 -     W(f)=count;
17 -     f=f+1;
18 - end
19 - figure
20 - semilogy(W,DET)
21 - xlabel('w')
22 - ylabel('DET')
23 - title('Determinante della matrice delle condizioni al contorno')
24 - end
25

trovaminimo.m x
1  function [soluz,n_it]=trovaminimo(file,amin,bmax,toll,max_it,msinf)
2  - load(file) %msinf=1 per mezzo finito, =2 per semi-infinito
3  - if msinf==2
4  -     matrix=FinM;
5  - else
6  -     matrix=MM;
7  - end
8  - n_it=0;
9  - soluz=NaN;
10 - j=1;
11 - h=toll;
12 - while (j<=max_it)
13 -     n_it=j;
14 -     if abs(amin-bmax)<=toll
15 -         soluz=amin;
16 -         break;
17 -     else
18 -         soluz=(amin+bmax)/2;
19 -         ff=abs(det(double(subs(matrix,w,soluz))));
20 -         ffh=abs(det(double(subs(matrix,w,soluz+h))));
21 -         fdf=(ffh-ff)/h;
22 -         if fdf<=0
23 -             amin=soluz;
24 -         else
25 -             bmax=soluz;
26 -         end
27 -         j=j+1;
28 -     end
29 - end
30 - end

```

A.3 Seconda Parte

```

tes2.m x
1 function [u1,u2,u3]=tesi2(w1,file,msinf) %per mezzo semi-infinito msinf=2 altrimenti msinf=1
2 load(file)
3 if msinf==2
4     matrix=FinM;
5 else
6     matrix=MM;
7 end
8
9 %sostituisco la soluzione nella matrice
10 M1=double(subs(matrix,w,w1));
11 [sz1,sz2]=size(M1);
12 alfw1=double(subs(ALFA,w,w1));
13 etaw1=subs(ETA,w,w1);
14 %calcolo tutte le ampiezze rapportate alla prima del primo strato
15 %la prima è posta uguale ad 1
16 new_F=M1(1:sz1-1,2:sz2);
17 term_noto=-M1(1:sz1-1,1);
18 [Ampsol]=linsolve(new_F,term_noto);
19 if msinf==2
20     Ampsol=[1;Ampsol;0;0];
21 else
22     Ampsol=[1;Ampsol];
23 end
24 %le dispongo in una matrice dove l'indice della colonna indica lo strato
25 if q_2==0
26     amp=zeros(4,strati);
27     coefu1=zeros(4,strati);
28     coefu2=zeros(4,strati);
29     coefu3=zeros(4,strati);
30 for l=1:strati
31     amp(:,l)=Ampsol((1+4*(l-1)):4*l);
32     if rif(l)==3
33         A1et=double(subs(etaw1(1,1),alfa,alfw1(:,l)));
34         coefu1(:,l)=amp(:,l).*A1et;
35         coefu3(:,l)=amp(:,l);
36     elseif rif(l)==1
37         coefu1(:,l)=amp(:,l);
38         A1et=double(subs(etaw1(1,1),alfa,alfw1(:,l)));
39         coefu3(:,l)=amp(:,l).*A1et;
40     end
41     if subs(eq2,w,w1)==0
42         coefu2(:,l)=1;
43     else
44         coefu2(:,l)=0;
45     end
46 end
47 end
48
49 elseif q_1==0
50     amp=zeros(4,strati);
51     coefu1=zeros(4,strati);
52     coefu2=zeros(4,strati);
53     coefu3=zeros(4,strati);
54 for l=1:strati
55     amp(:,l)=Ampsol((1+4*(l-1)):4*l);
56     if rif(l)==3
57         A1et=double(subs(etaw1(1,1),alfa,alfw1(:,l)));
58         coefu2(:,l)=amp(:,l).*A1et;
59         coefu3(:,l)=amp(:,l);

```

```

60 -     elseif rif(1)==2
61 -         A1et=double(subs(etaw1(1,1), alfa, alfw1(:,1)));
62 -         coefu2(:,1)=amp(:,1);
63 -         coefu3(:,1)=amp(:,1).*A1et;
64 -
65 -     end
66 -     if subs(eq1,w,w1)==0
67 -         coefu1(:,1)=1;
68 -     else
69 -         coefu1(:,1)=0;
70 -     end
71 -
72 - end
73 -
74 - else
75 -     amp=zeros(6,strati);
76 -     coefu1=zeros(6,strati);
77 -     coefu2=zeros(6,strati);
78 -     coefu3=zeros(6,strati);
79 -     for l=1:strati
80 -         amp(:,l)=Ampsol((1+6*(1-1)):6*l);
81 -         if rif(1)==3
82 -             A1et=double(subs(etaw1(1,1), alfa, alfw1(:,1)));
83 -             coefu1(:,l)=amp(:,l).*A1et;
84 -             A2et=double(subs(etaw1(2,1), alfa, alfw1(:,1)));
85 -             coefu2(:,l)=amp(:,l).*A2et;
86 -             coefu3(:,l)=amp(:,l);
87 -         elseif rif(1)==2
88 -             A1et=double(subs(etaw1(1,1), alfa, alfw1(:,1)));
89 -             coefu1(:,l)=amp(:,l).*A1et;
90 -             coefu2(:,l)=amp(:,l);
91 -             A2et=double(subs(etaw1(2,1), alfa, alfw1(:,1)));
92 -             coefu3(:,l)=amp(:,l).*A2et;
93 -         elseif rif(1)==1
94 -             coefu1(:,l)=amp(:,l);
95 -             A1et=double(subs(etaw1(1,1), alfa, alfw1(:,1)));
96 -             coefu2(:,l)=amp(:,l).*A1et;
97 -             A2et=double(subs(etaw1(2,1), alfa, alfw1(:,1)));
98 -             coefu3(:,l)=amp(:,l).*A2et;
99 -         end
100 -
101 -     end
102 - end
103 -
104 -     esp=exp(-alfw1*x3);
105 -     u1=sum(coefu1.*esp);
106 -     u2=sum(coefu2.*esp);
107 -     u3=sum(coefu3.*esp);
108 -
109 -     u1=u1*exp(1i*(q_1*x1+q_2*x2-w1*t));
110 -     u2=u2*exp(1i*(q_1*x1+q_2*x2-w1*t));
111 -     u3=u3*exp(1i*(q_1*x1+q_2*x2-w1*t));
112 -
113 - end

```


Appendice B

Manuale d'uso

Una volta avviato, lo script `Tesi.m` chiede per prima cosa il numero di strati complessivo del problema e le componenti del vettore d'onda k_1 e k_2 . Ricevuti i primi input si avvia un procedimento identico per ogni strato. Vengono chiesti i valori di x_3 che delimitano superiormente ($x_{3_{min}}$) ed inferiormente ($x_{3_{max}}$) lo strato se è il primo, mentre soltanto il limite inferiore per gli altri poichè quello superiore viene automaticamente impostato uguale a quello inferiore dello strato precedente. Si definiscono gli spostamenti ed il vettore deformazione in notazione matriciale. Poi viene chiesto in quale tipo di geometria ci si vuole mettere; i casi previsti sono tre: *isotropo*, *cubico*, *esagonale*. Per essi si chiedono i valori delle corrispondenti costanti elastiche indipendenti: C_{11} e C_{12} per l'isotropo, C_{11} , C_{12} e C_{44} per il cubico, C_{11} , C_{12} , C_{13} , C_{33} e C_{44} per l'esagonale. La matrice viene composta in automatico ed esiste anche un controllo sui vincoli termodinamici; in tal caso esce un messaggio di errore ed il programma si interrompe. Alla fine della procedura di calcolo si ottengono le matrici delle condizioni al contorno, salvate sotto i nomi di 'MM' per il substrato finito e 'FinM' per il substrato semi-infinito. Nei programmi accessori e nella Seconda Parte si potrà scegliere se utilizzare l'una o l'altra semplicemente impostando un indice che vale '1' per il primo caso e '2' per il secondo. Per poter utilizzare le parti successive bisogna salvare il 'workspace' ottenuto con questa prima parte. In esso, oltre alle suddette matrici, sono presenti tutti gli altri dati del problema, salvati in apposite matrici dove l'indice della colonna coincide con il numero dello strato; per esempio le densità sono salvate nella variabile 'RHO',

```

>> Tesi
Inserire il numero di strati: 1
Digitare il valore del vettore d onda k1: (2)*cos(0/180*pi)*2*pi/514.5e-9*sin(30/180*pi)
Digitare il valore del vettore d onda k2: (2)*sin(0/180*pi)*2*pi/514.5e-9*sin(30/180*pi)

Layer n° 1

Digitare il valore di x_min: 0
Digitare il valore di x_max: 5e-6
Digitare il valore della densità: 2329

Scegliere la geometria
Digitare 1 per isotropo, 2 per cubico, 3 per esagonale: 1

Isotropo
Inserire un valore per C11: 223e9
Inserire un valore per C12: 64e9

C =

1.0e+11 *

    2.2300    0.6400    0.6400         0         0         0
    0.6400    2.2300    0.6400         0         0         0
    0.6400    0.6400    2.2300         0         0         0
         0         0         0    0.7950         0         0
         0         0         0         0    0.7950         0
         0         0         0         0         0    0.7950

Il riferimento è A1

matrici MM per mezzo finito e FinM per mezzo semi-infinito pronte all uso

fx >>

```

Fig. B.1: Schermata tipo per un monolayer isotropo

le α_j in 'ALFA' mentre le matrici delle costanti elastiche in 'C1', 'C2', ... e così via. In figura B.1 è mostrata una tipica schermata per il caso di un monolayer in geometria isotropa.

Il programma chiamato 'printdetlog' è definito come una funzione di MATLAB ed è stato creato per rappresentare in un grafico l'andamento del modulo del determinante di una delle matrici delle condizioni al contorno in funzione della pulsazione ω . Esso prende in ingresso 5 input. Il primo è il nome del file del 'workspace' ottenuto dal file 'Tesi.m'; il secondo ed il terzo sono i limiti dell'intervallo dei valori di ω di interesse; il quarto è lo step tra due punti di campionamento, ovvero la risoluzione grafica rispetto alla pulsazione; il quinto è l'indice per decidere se utilizzare la matrice per il mezzo finito o per quello semi-infinito ed assume rispettivamente i valori '1' o '2'. In uscita compare il grafico corrispondente del modulo del determinante della matrice in scala semi-logaritmica.

Il programma 'trovaminimo', anch'esso in forma di funzione, serve per trovare il valore di ω , all'interno di un intervallo, che minimizza il logaritmo del modulo del determinante, cioè il valore che lo annulla a meno di una risoluzione imposta. Esso prende in ingresso 6 input: il nome del file del 'workspace' ottenuto dal file 'Tesi.m'; i limiti dell'intervallo della pulsazione w in cui trovare il minimo; la tolleranza sull'errore; il numero massimo di interazioni ammissibili e l'indice per decidere se mettersi nel caso finito e semi-infinito. In uscita ha due elementi: il valore di w cercato ed il numero di iterazioni fatte. Tale pulsazione sarà poi usata nella seconda parte del programma per trovare gli spostamenti.

Il file 'tesi2' è una funzione che ha come uscita i tre spostamenti u_1 , u_2 e u_3 . Prende in ingresso 3 input: la pulsazione w dell'onda di cui si vuole calcolare gli spostamenti trovata con i programmi accessori; il file del 'workspace' del calcolo effettuato con la Prima Parte; l'indice per decidere se usare la matrice del caso di substrato finito o quella di substrato semi-infinito. Lo script carica i dati contenuti nel file e sostituisce la pulsazione w data in ingresso nella matrice corrispondente al caso scelto e nelle espressioni delle α_j e dei rapporti η_1 e η_2 . Calcola dalla matrice le ampiezze incognite prendendone come riferimento una e poi ricostruisce le funzioni degli spostamenti che pone in uscita.

Bibliografia

- [1] A. Akjouj et al. «Resonant Guide Elastic Waves in an Adsorbed Slab: Theoretical Analysis of the Density of States». In: *J. Phys.: Condens. Matter* 6 (1994), pp. 1089–1098.
- [2] V. V. Aleksandrov et al. «Anisotropy of Rayleigh Wave Velocity of $\text{TmBa}_2\text{C}_3\text{O}_{7-y}$ Single Crystals». In: *Solid State Commun.* 74 (1990), pp. 749–752.
- [3] V. V. Aleksandrov et al. «Elastic Coefficients of KTiOPO_4 RbTiOPO_4 TlTiOPO_4 Crystals Determined from Mandelstamm-Brillouin Light Scattering Spectra». In: *Solid State Commun.* 69 (1989), pp. 877–881.
- [4] M. Boekholt et al. «Determination of the Sound Velocities and the Complete Set of Elastic Constants for $\text{Bi}_2\text{Sr}_2\text{CaCu}_2\text{O}_{8+\delta}$ Single Crystals Using Brillouin Light Scattering». In: *Physica C* 179 (1991). costanti elastiche di superconduttori HTC bulk da misure di onde bulk e SAW, pp. 101–106.
- [5] Alessandro Cattivelli. «Onde Acustiche di Superficie in Film Supportati di Simmetria Esagonale». Tesi di Laurea. Politecnico di Milano.
- [6] L. Colombo e S. Giordano. *Introduzione alla teoria della elasticità*. Springer, 2007.
- [7] O. Hardouin Duparc, E. Sanz-Velasco e V. R. Velasco. «Elastic Surface in Crystals with Overlayers: Cubic Symmetry». In: *Phys. Rev. B* 30 (1984), pp. 2042–2048.
- [8] V. R. Velasco E. Sanz-Velasco O. Hardouin-Duparc. «Acoustic Surface Waves in Cubic Crystals with Overlayers». In: *Surface Science* 126 (1983), pp. 202–207.

- [9] A. G. Every. «General Closed-Form Expressions for Acoustic Waves in Elastically Anisotropic Solids». In: *Phys. Rev. B* 22 (1980), pp. 1746–1760.
- [10] G. W. Farnell. «Properties of Elastic Surface Waves». In: *Physical Acoustics*. 1970.
- [11] J.P. Jones. «Wave Propagation in a Two-Layered Medium». In: *Transact. of the ASME – J. Appl. Mech.* (Giu. 1964). onde parziali per un free standing bistrato, pp. 213–222.
- [12] J. O. Kim et al. «Elastic constants of single crystal transition-metal nitride films measured by line-focus acoustic microscopy». In: *J. Appl. Phys.* 72 (1992). costanti elastiche di film monocristallini. c’e’ bene la procedura del problema inverso, e la trattazione con le onde parziali del substrato cubico + film cubico, pp. 1805–1811.
- [13] T. Kundu. «Capitolo 1, Mechanics of Elastic Waves and Ultrasonic NDE». In: *Ultrasonic nondestructive evaluation*. CRC Press, 2004.
- [14] L. D. Landau e E. M. Lifshits. *Teoria dell’elasticità*. Editori Riuniti, 1979.
- [15] L.Dobrzynski e A.A. Maradudin. «boh». In: *Phys. Rev. B* 14 (1976). referenza non verificata, calcolo di velocita di Rayleigh per semiinfinito esagonale con asse ortogonale alla superficie (isotropo nel piano della superficie), p. 2200.
- [16] L.Dobrzynski e A.A. Maradudin. «boh». In: *Phys. Rev. B* 15 (1977). referenza non verificata, tensore di Green (da cui velocita di Rayleigh) per semiinfinito esagonale con asse ortogonale alla superficie (isotropo nel piano della superficie), p. 2432.
- [17] S. Makarov, E. Chilla e H. J. Frölich. «Determination of elastic constants of thin films from phase velocity dispersion of different surface acoustic wave modes». In: *J. Appl. Phys.* 78 (1995), pp. 5028–5034.
- [18] J. F. Nye. *Physical Properties of Crystals*. Oxford at the Clarendon Press, 1972.

-
- [19] X.Jiang, J. Fassbender e B. Hillebrands. «Elastic Constants of WC-a-C:H Composite Films Studied by Brillouin Spectroscopy». In: *Phys. Rev. B* 49 (1994), pp. 13815–13818.
- [20] P.-C. Xu e S. K. Datta. «Guided Waves in a Bonded Plate: A Parametric Study». In: *J. Appl. Phys.* 67 (1990), pp. 6779–6786.