



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**

**UNIVERSITA' DEGLI STUDI DI PADOVA
Dipartimento di Ingegneria Industriale DII**

Corso di Laurea Magistrale in Ingegneria Aerospaziale

**Propagazione di cricche e impatti
con la teoria della Peridinamica in Abaqus**

Relatore

Prof. Ugo Galvanetto

Correlatori

Prof. Mirco Zaccariotto

Teo Mudric

Laureando: Matteo Garelli

Matricola n. 1041163

Anno Accademico 2014/2015

Sommario

Introduzione	5
Indice delle figure	7
Indice delle tabelle	8
1. Peridinamica.....	9
1.1. Introduzione e principali campi di utilizzo	9
1.2. Modellazione peridinamica del continuo	10
1.3. Modello costitutivo	14
1.3.1. Riferimenti per il calcolo dell'energia di frattura	18
1.4. Peridinamica 2D.....	19
1.5. Forze di corto raggio.....	21
1.6. Strategie di discretizzazione.....	22
1.7. Normalizzazione di forza alle superfici	25
1.8. Implementazione con gli elementi finiti	26
1.9. Coefficiente correttivo del volume	29
1.10. Coefficiente di Poisson nella teoria della peridinamica bond-based	31
1.11. Materiale Microelastico PMB (Prototype Microelastic Brittle material)	34
2. Propagazione delle cricche in piastre di piccolo spessore	37
Modalità di applicazione del carico	41
2.1. Tipologia di convergenza n. 1: m convergenza.....	43
2.1.1. Piastra in Duran 50 Glass con $\delta=0.002\text{m}$, $m=2$ e $\Delta x=0.001\text{m}$	45
2.1.2. Piastra in Duran 50 Glass con $\delta=0,002\text{ m}$, $m=4$ e $\Delta x=0,0005\text{m}$	48
2.1.3. Piastra in Duran 50 Glass con $\delta=0,002\text{m}$, $m=6$ e $\Delta x=0,00033\text{m}$	50
2.1.4. Piastra in Duran 50 Glass con $\delta=0.003\text{m}$, $m=3$ e $\Delta x=0.001\text{m}$	53
2.1.5. Piastra in Duran 50 Glass con $\delta=0.003\text{m}$, $m=6$ e $\Delta x=0.0005\text{m}$	55
2.2. Tipologia di convergenza n. 2: δ convergenza	57
2.2.1. Piastra in Duran 50 Glass con $\delta=0,004\text{m}$, $m=4$ e $\Delta x=0,001\text{m}$	59
2.2.2. Piastra in Duran 50 Glass con $\delta=0,002\text{m}$, $m=4$ e $\Delta x=0,0005\text{m}$	61
2.2.3. Piastra in Duran 50 Glass con $\delta=0.001\text{m}$, $m=4$ e $\Delta x=0.00025\text{m}$	63
2.3. Confronto fra le distanze delle biforcazioni.....	66
2.4. Piastra con cricca ramificata	68
2.4.1. Piastra in Soda Lime Glass con $\delta=0.001\text{m}$, $m=4$, $\Delta x=0.00025\text{m}$	69
3. Simulazione con Abaqus dell'esperimento di Kalthoff - Winkler.....	73
3.1. Simulazioni dell'esperimento di Kalthoff – Winkler	75
4. Simulazione con Abaqus di un impatto fra un corpo rigido modellato con gli elementi finiti e una piastra modellata con la teoria della Peridinamica	85
4.1. Impatto con velocità di 1 m/s	88

4.2.	Impatto con velocità di 5 m/s	90
4.3.	Impatto con velocità di 10 m/s	92
4.4.	Impatto con velocità di 20 m/s	93
4.5.	Impatto con velocità di 30 m/s	95
4.6.	Confronto fra impatti a velocità elevate: impatti con velocità di 50 m/s, di 75 m/s, di 100 m/s	97
	Conclusioni e sviluppi futuri	101
	Appendice A – Programmi in Matlab	103
	Programma in Matlab per la creazione dei file inp per Abaqus	103
	Sezione per la creazione dei nodi.....	103
	Sezione per la creazione degli elementi bond	104
	Sezione per la creazione della cricca	105
	Sezione per la creazione degli elset	107
	Sezione per la creazione dei file.....	112
	Programma in Matlab per l’elaborazione dei risultati di Abaqus e la creazione delle figure dei risultati.....	118
	Appendice B – Programmi in Visual Studio	123
	Vumat per la simulazione dell’esperimento di Kalthoff - Winkler.....	123
	Vumat per la simulazione dell’impatto	124
	Bibliografia	127

Introduzione

In questo elaborato si vuole analizzare la Teoria della Peridinamica applicata allo studio delle cricche all'interno di un provino. Questa teoria è stata oggetto di approfondimenti in questi ultimi anni in quanto la Teoria del Continuo se utilizzata per lo studio della frattura e la propagazione di cricche non riesce a prevederne il percorso. Sono disponibili criteri o semplificazioni esterne per quest'ultima ma non raggiungono il livello di precisione e dettaglio della teoria della Peridinamica.

Le equazioni del moto della Peridinamica sono definite anche in casi di superfici con cricche, di parti vuote ed in generale in zone di discontinuità. Nella Teoria del Continuo invece le derivate che appaiono nelle equazioni differenziali alle derivate parziali non sono definite in corrispondenza di queste tipologie di discontinuità.

Nota la geometria del modello, utilizzando le equazioni del moto della Peridinamica è possibile sviluppare delle analogie con gli elementi finiti che si basano invece sulla Teoria del Continuo. Nasce quindi la possibilità di simulare il comportamento dei *bond* (legami) della Peridinamica con degli elementi monodimensionali chiamati *truss* già presenti all'interno dei programmi software, come Abaqus, che utilizzano gli elementi finiti.

In questa tesi si vuole dimostrare come, utilizzando il software "Abaqus", sia possibile inserendo la geometria del modello, i materiali e le condizioni al contorno, avere direttamente il risultato finale con evidenziate le zone di rottura.

Partendo dalle equazioni della peridinamica, nel caso in cui il *bond* ha un comportamento prettamente elastico esso può essere simulato per mezzo di una rigidità. Le equazioni che descrivono il comportamento elastico del *bond* sono uguali a quelle della molla nella meccanica classica ponendo la condizione che l'allungamento fra le due estremità del *bond* non può superare un determinato valore massimo.

I dati attualmente disponibili quali massa, modulo elastico, energia di frattura sono idonei ad essere inseriti direttamente nelle proprietà degli elementi finiti. Partendo da questi dati attraverso passaggi matematici è possibile arrivare ai valori da assegnare ai nodi e ai *bond*. Tali valori verranno successivamente inseriti nel programma per definire le caratteristiche dei nodi e dei *bond* stessi.

Quando vi è la rottura di un provino, se il materiale non subisce trasformazioni fisiche o chimiche, la massa totale dei singoli pezzi risultanti deve essere uguale alla massa di partenza.

La zona di rottura è l'area in cui i *bond* presenti hanno subito un allungamento maggiore di quello massimo e quindi si sono rotti. Essi vengono eliminati durante la simulazione; la massa del provino deve allora essere assegnata ai nodi, i quali subiscono solamente spostamenti ma non vengono mai cancellati in tutta la simulazione. Dalle caratteristiche sopra evidenziate l'elemento finito denominato *truss* di Abaqus è quello che più si avvicina al comportamento del *bond*, in quanto si rompe sia in trazione che in compressione.

Creato il modello in Abaqus si confrontano i risultati ottenuti con quelli disponibili in letteratura scientifica. Simulando il comportamento del materiale di un provino sottoposto a trazione si osserva che il percorso risultante della cricca all'interno di esso è lo stesso degli studi di riferimento. Dal confronto dei risultati è stato così possibile validare il modello creato in Abaqus per la situazione di trazione.

Un'ulteriore conferma della correttezza del modello in "Abaqus", si è avuta simulando l'esperimento di Kalthoff – Winkler, i cui risultati trovano corrispondenza con quelli di riferimento.

Con il modello validato si è poi analizzato il comportamento di un corpo rigido che impatta su una piastra di piccolo spessore. Questo caso è interessante in quanto utilizzando solamente i comandi di Abaqus per contatti ed impatti è stato possibile fare interagire un corpo modellato con gli elementi finiti ed una piastra modellata con nodi e *bond*.

Indice delle figure

FIGURA 1-1 - INTERAZIONE FRA PARTICELLA E BOND	11
FIGURA 1-2 - GRAFICO S-F CON L'ANDAMENTO DI μ	15
FIGURA 1-3 - RIFERIMENTI PER IL CALCOLO DELL'ENERGIA DI FRATTURA	19
FIGURA 1-4 - CONSTANT (SINISTRA) E CONICAL (DESTRA) MICRO-MODULUS FUNCTIONS	19
FIGURA 1-5 COEFFICIENTE CORRETTIVO BETA DEL VOLUME	30
FIGURA 1-6 VALORI DI BETA CON M=2 E M=3	31
FIGURA 1-7 VALORI DI BETA CON M=4 E M=6	31
FIGURA 1-8 RIFERIMENTI PER CAMBIO DI VARIABILE	32
FIGURA 1-9 CURVA FORZA PERIDINAMICA - ALLUNGAMENTO DEL MATERIALE PMB	35
FIGURA 2-1 - GEOMETRIA DELLA PIASTRA	37
FIGURA 2-2 - TABELLA DEI MATERIALI UTILIZZATI	38
FIGURA 2-3 - TIPOLOGIE DI CONVERGENZA: M-CONVERGENZA δ -CONVERGENZA	39
FIGURA 2-4 - PIASTRA CON IL CARICO APPLICATO AI BORDI	42
FIGURA 2-5 - PARTICOLARE INGRANDITO DELL'ANGOLO IN ALTO A SINISTRA DELLA FIGURA 13	42
FIGURA 2-6 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=2 $\Delta x=0,001$ M	47
FIGURA 2-7 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=2 $\Delta x=0,001$ M	47
FIGURA 2-8 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=2 $\Delta x=0,001$ M	48
FIGURA 2-9 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	49
FIGURA 2-10 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	50
FIGURA 2-11 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	50
FIGURA 2-12 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=6 E $\Delta x=0,000333$ M	52
FIGURA 2-13 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=6 E $\Delta x=0,000333$ M	52
FIGURA 2-14 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=3 E $\Delta x=0,001$ M	54
FIGURA 2-15 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=3 E $\Delta x=0,001$ M	54
FIGURA 2-16 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=3 E $\Delta x=0,001$ M	54
FIGURA 2-17 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=6 E $\Delta x=0,0005$ M	56
FIGURA 2-18 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=6 E $\Delta x=0,0005$ M	56
FIGURA 2-19 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,003$ M M=6 E $\Delta x=0,0005$ M	57
FIGURA 2-20 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,004$ M M=4 E $\Delta x=0,001$ M	60
FIGURA 2-21 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,004$ M M=4 E $\Delta x=0,001$ M	60
FIGURA 2-22 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,004$ M M=4 E $\Delta x=0,001$ M	61
FIGURA 2-23 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	62
FIGURA 2-24 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	63
FIGURA 2-25 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,002$ M M=4 E $\Delta x=0,0005$ M	63
FIGURA 2-26 - PIASTRA IN DURAN 50 GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	65
FIGURA 2-27 - (IMMAGINE MATLAB) PIASTRA IN DURAN 50 GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	65
FIGURA 2-28 - (IMMAGINE ABAQUS) PIASTRA IN DURAN 50 GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	65
FIGURA 2-29 - DETTAGLIO DELLA BIFORCAZIONE DELLA PIASTRA CON $\delta=0,002$ M, M=2, $\Delta x=0,001$ M (§ 2.1.1)	67
FIGURA 2-30 - DETTAGLIO DELLA BIFORCAZIONE DELLA PIASTRA CON $\delta=0,002$ M, M=4, $\Delta x=0,0005$ M (§ 2.1.2)	68
FIGURA 2-31 - DETTAGLIO DELLA BIFORCAZIONE DELLA PIASTRA CON $\delta=0,002$ M, M=6, $\Delta x=0,00033$ M (§ 2.1.3)	68
FIGURA 2-32 - PIASTRA IN SODA LIME GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	71
FIGURA 2-33 - (IMMAGINE MATLAB) PIASTRA IN SODA LIME GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	71
FIGURA 2-34 - (IMMAGINE ABAQUS) PIASTRA IN SODA LIME GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M	71
FIGURA 2-35 - (IMMAGINE ABAQUS) PIASTRA IN SODA LIME GLASS CON $\delta=0,001$ M M=4 E $\Delta x=0,00025$ M A SIMULAZIONE TERMINATA	72
FIGURA 3-1 - SCHEMA DELL'ESPERIMENTO DI KALTHOFF – WINKLER	74
FIGURA 3-2 - GEOMETRIA DEL MODELLO	74
FIGURA 3-3 - A - RISULTATI DELL'ARTICOLO A 63 μ S B- RISULTATI ELABORATI DA MATLAB A 63 μ S	77
FIGURA 3-4 - RISULTATI DELLA SIMULAZIONE DELL'ESPERIMENTO DI KALTHOFF – WINKLER GENERATI DA ABAQUS A 63 μ S	78

FIGURA 3-5 - A - RISULTATI DELL'ARTICOLO A 90 μ s	B- RISULTATI ELABORATI DA MATLAB A 90 μ s	78
FIGURA 3-6 - RISULTATI DELLA SIMULAZIONE DELL'ESPERIMENTO DI KALTHOFF – WINKLER GENERATI DA ABAQUS A 90 μ s		79
FIGURA 3-7 – A - RISULTATI DEL LIBRO	B- RISULTATI GENERATI DA MATLAB	82
FIGURA 3-8 - RISULTATI DELLA SIMULAZIONE DELL'ESPERIMENTO	FIGURA 3-9 - RISULTATI CON EMU	82
FIGURA 4-1 - ESTREMITÀ SINISTRA CON I NODI A CUI È APPLICATO L'INCASTRO		87
FIGURA 4-2 - MODELLO IN ABAQUS		88
FIGURA 4-3 - GRAFICO DELLA VELOCITÀ DEL PROIETTILE IN FUNZIONE DEL TEMPO A 1M/s		89
FIGURA 4-4 - POSIZIONE DEL <i>BOND 154</i>		90
FIGURA 4-5 - IMMAGINE RISULTANTE DI ABAQUS PER L'IMPATTO A 5 M/s A 0.0005 s		91
FIGURA 4-6 - - IMMAGINE DEL DANNO DA MATLAB PER L'IMPATTO A 5 M/s A 0.0005 s		91
FIGURA 4-7 - PARTICOLARE DELL'IMPATTO A 5 M/s		92
FIGURA 4-8 - GRAFICO DELLA VELOCITÀ DEL PROIETTILE IN FUNZIONE DEL TEMPO A 5 M/s		92
FIGURA 4-9 - IMMAGINE RISULTANTE DI ABAQUS PER L'IMPATTO A 10 M/s A 12,5 μ s		93
FIGURA 4-10 - IMMAGINE DEL DANNO DA MATLAB PER L'IMPATTO A 10 M/s A 12,5 μ s		93
FIGURA 4-11 - GRAFICO DELLA VELOCITÀ DEL PROIETTILE IN FUNZIONE DEL TEMPO A 10 M/s		93
FIGURA 4-12 - IMMAGINE RISULTANTE DI ABAQUS PER L'IMPATTO A 20 M/s A 12,5 μ s		94
FIGURA 4-13 - IMMAGINE DEL DANNO DA MATLAB PER L'IMPATTO A 20 M/s A 12,5 μ s		94
FIGURA 4-14 - GRAFICO DELLA VELOCITÀ DEL PROIETTILE IN FUNZIONE DEL TEMPO A 20 M/s A 12,5 μ s		94
FIGURA 4-15- IMMAGINE RISULTANTE DI ABAQUS PER L'IMPATTO A 30 M/s A 12,5 μ s		96
FIGURA 4-16 - IMMAGINE DEL DANNO DA MATLAB PER L'IMPATTO A 30 M/s A 12,5 μ s		96
FIGURA 4-17 - GRAFICO DELLA VELOCITÀ DEL PROIETTILE IN FUNZIONE DEL TEMPO A 30 M/s		96

Indice delle tabelle

TABELLA 2-1 - TABELLA M CONVERGENZA CON $\delta=0,002M$	44
TABELLA 2-2- TABELLA M CONVERGENZA CON $\delta=0,003M$	44
TABELLA 2-3 - TABELLA δ CONVERGENZA CON $\delta=0,002M$	58
TABELLA 2-4 - TABELLA DATI RAMIFICAZIONE. SONO RIPORTATE LE DISTANZE	67
TABELLA 4-1 - TABELLA IMMAGINI RELATIVE ALLA PROPAGAZIONE DELLE CRICCHE A 20 M/s	95
TABELLA 4-2 - TABELLA RISULTATI DI ABAQUS PER LE VELOCITÀ DI 50 M/s, 75 M/s, 100 M/s	98
TABELLA 4-3 - TABELLA RISULTATI DI MATLAB PER LE VELOCITÀ DI 50 M/s, 75 M/s, 100 M/s	98
TABELLA 4-4 - TABELLA CON I GRAFICI DELLE VELOCITÀ DEL PROIETTILE PER LE VELOCITÀ DI 50 M/s, 75 M/s, 100M/s	99
TABELLA 4-5 - TABELLA ASSORBIMENTO ENERGIA CINETICA	100

1. Peridinamica

1.1. Introduzione e principali campi di utilizzo

La peridinamica è una teoria riguardante la meccanica dei solidi e si basa sull'interazione diretta di punti materiali disposti ad una distanza finita. Dovendo essere applicata a corpi solidi a questi punti vengono associate delle celle cubiche in grado di interagire fra di loro. I problemi che riguardano la propagazione di cricche sono molto difficili da trattare con la teoria classica del continuo perché le equazioni differenziali alle derivate parziali risultanti da quest'ultima sono di difficile soluzione in presenza delle discontinuità. Le derivate che appaiono in queste equazioni non sono infatti definite nelle estremità delle cricche e sulle loro superfici. I metodi di calcolo con la teoria locale del continuo richiedono informazioni supplementari come i punti di partenza e la velocità di propagazione della discontinuità non sempre disponibili fra i dati iniziali.

In questi ultimi anni si è ulteriormente sviluppata la teoria della peridinamica passando dall'iniziale teoria bond-based alla teoria state-based, rimuovendo la condizione restrittiva del coefficiente di Poisson fisso ed introducendo ulteriori analogie con la teoria del continuo tramite dei concetti di deformazione e tensione che mettono così in relazione le due tipologie di risultati.

In questo capitolo viene brevemente illustrata la teoria bond-based facendo riferimento a vari testi ed articoli¹ citati nella bibliografia prendendo da ognuno le parti di interesse per lo sviluppo della seguente tesi ed integrandole fra di loro.

¹ Siling S.A., Askary E. *A meshfree method based on the peridynamic model of solid mechanics*, Elsevier, 83 (2005) 1526-1535, p. 1526-1535, 21/03/2005

Ha Y. D., Bobaru F., *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

Kebing Yu. *Enhanced Integration methods for the peridynamic theory*, Kansas State University, 2011

Macek M. W., Siling S.A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, 23/10/2007

Scudellaro L., *Nonlocal theories for the study of continuum: a finite elements method implementation*, Università degli studi di Padova, 2012

Madenci E., Erkan Oterkus E., *Peridynamic theory and its applications*, Springer, DOI10.1007/978-1-4614-8465-3, 2014

1.2. Modellazione peridinamica del continuo

La peridinamica può essere paragonata alla dinamica molecolare dove le particelle in cui è stato suddiviso il materiale continuo, subiscono una determinata accelerazione.

L'accelerazione lungo x nella configurazione di riferimento al momento t è data da:

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \int_{H_x} f(\mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t), \mathbf{x}' - \mathbf{x}) dV_{x'} + \mathbf{b}(\mathbf{x}, t) \quad (1.1)^2$$

con:

H_x : intorno di \mathbf{x} \mathbf{u} : spostamento \mathbf{b} : campo di densità di forza del corpo

ρ : densità di massa nella configurazione di riferimento

f : pairwise force function il cui valore è pari al vettore di forza (per unità di volume al quadrato) che la particella x' esercita sulla particella x . La posizione relativa tra queste due particelle nella configurazione di riferimento è denominata ξ :

$$\xi = \mathbf{x}' - \mathbf{x} \quad \text{e } \boldsymbol{\eta} \text{ rappresenta gli spostamenti: } \boldsymbol{\eta} = \mathbf{u}(\mathbf{x}', t) - \mathbf{u}(\mathbf{x}, t) \quad (1.2)$$

Il vettore della posizione relativa in un determinato istante fra le due particelle è dato quindi da $\xi + \boldsymbol{\eta}$.

L'interazione fisica fra la particella x' e la particella x è chiamato *bond* (legame) che se è di tipo elastico è chiamato *spring* (molla).

Questo concetto che identifica un *bond* con una distanza di valore noto è la principale differenza fra la peridinamica e la teoria classica. Mentre quest'ultima è basata sull'idea delle forze di contatto; nella peridinamica le particelle interagiscono sia con le particelle con cui sono in contatto sia con quelle con cui non lo sono. Nessun *bond* relativo ad una data particella può essere maggiore di una determinata lunghezza chiamata orizzonte di quella data particella³. Si può dimostrare che nella peridinamica il momento lineare (quantità di moto) e angolare si conservano perché le forze dei *bond* sono uguali ed opposte e dirette lungo un vettore. Tale

² Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p. 1526-1535, 21/03/2005

³ Vedi nota 2 p.10

vettore connette, nella posizione attuale, due particelle interagenti nello stesso modo fra di loro. L'orizzonte viene in genere indicato con δ ⁴:

$$|\xi| > \delta \Rightarrow f(\eta, \xi) = \mathbf{0} \quad \forall \eta, \xi \quad (1.3)$$

La particella x quindi non può avere interazioni con particelle che stanno oltre l'orizzonte. Come indicato nella Figura 1-1 l'orizzonte è rappresentato da δ e la zona di interazione della particella è rappresentata da H_x che ha forma sferica nel caso tridimensionale ed è chiamata sfera di influenza.

La funzione di forma per essere valida ed utilizzabile deve sottostare alle seguenti proprietà⁵:

$$f(-\eta, -\xi) = -f(\eta, \xi) \quad \forall \eta, \xi \quad \text{per avere la conservazione della quantità di moto} \quad (1.4)$$

$$(\xi + \eta) \times f(\eta, \xi) = \mathbf{0} \quad \forall \eta, \xi \quad \text{per avere la conservazione del momento angolare} \quad (1.5)$$

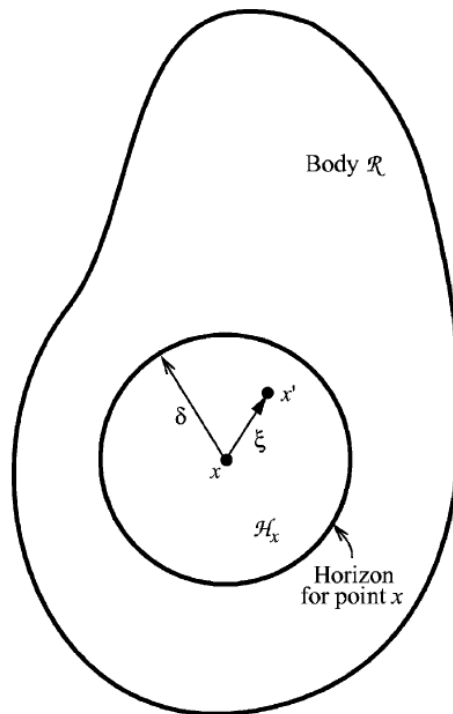


Figura 1-1 - Interazione fra particella e bond

⁴ Vedi nota 2 p.10

⁵ Vedi nota precedente

Un materiale è detto micro elastico se la pairwise force function è derivabile da un micropotenziale scalare w ⁶:

$$f(\boldsymbol{\eta}, \boldsymbol{\xi}) = \frac{\partial w}{\partial \boldsymbol{\eta}}(\boldsymbol{\eta}, \boldsymbol{\xi}) \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad (1.6)$$

Il micropotenziale è l'energia di ogni singolo *bond*, e dimensionalmente è un'energia per unità di volume al quadrato. L'energia per unità di volume è data quindi da:

$$W = \frac{1}{2} \int_{Hx} w(\boldsymbol{\eta}, \boldsymbol{\xi}) dV_{\boldsymbol{\xi}} \quad (1.7)$$

Il fattore $\frac{1}{2}$ è dovuto al fatto che il *bond* possiede solo metà dell'energia di legame. Se il corpo è composto da un materiale microelastico il lavoro svolto dalle forze esterne "viene memorizzato" dal componente variando in maniera recuperabile la sua forma come avviene nella teoria della meccanica classica (Teoria dell'elasticità). Si può inoltre dimostrare che il micropotenziale dipende dallo spostamento $\boldsymbol{\eta}$ in relazione alla distanza scalare fra i punti deformati⁷:

$$\hat{w}(y, \boldsymbol{\xi}) = w(\boldsymbol{\eta}, \boldsymbol{\xi}) \quad \forall \boldsymbol{\eta}, \boldsymbol{\xi} \quad y = |\boldsymbol{\eta} + \boldsymbol{\xi}| \quad (1.8)$$

L'interazione fra due punti di materiale microelastico può essere pensata come proprietà di una molla elastica. Le proprietà della molla possono dipendere in maniera separata anche dal vettore $\boldsymbol{\xi}$, e quindi eventuali anisotropie in materiali microelastici possono essere introdotte modificando questo vettore. Se si prende ad esempio un materiale composito formato da fibre tutte parallele ad un vettore \mathbf{g} e rinforzo (matrice) si ha⁸:

$$\hat{w}(y, \boldsymbol{\xi}) = \beta^2 \hat{w}_{fibra}(y, |\boldsymbol{\xi}|) + (1 - \beta^2) \hat{w}_{matrice}(y, |\boldsymbol{\xi}|)$$

$$\beta = \mathbf{g} \cdot \boldsymbol{\xi} / |\boldsymbol{\xi}|$$

Da cui si vede la dipendenza dell'energia dalla direzione del *bond*.

⁶ Vedi nota 2 p.10

⁷ Vedi nota precedente

⁸ Vedi nota precedente

Dalle equazioni (1.6) e (1.8) e differenziando la seconda rispetto a η si ottiene:

$$f(\eta, \xi) = \frac{\xi + \eta}{|\xi + \eta|} f(|\xi + \eta|, \xi) \quad \forall \xi, \eta \quad (1.9)$$

con f definito da:

$$f(y, \xi) = \frac{\partial \hat{w}}{\partial y}(y, \xi) \quad \forall y, \xi \quad (1.10)$$

Ciò soddisfa le equazioni (1.4) e (1.5):

$$\hat{w}(y, -\xi) = \hat{w}(y, \xi) \quad \forall y, \xi \quad (1.11)$$

Le relazioni formate dalla equazione (1.1) e dalla equazione (1.9) contengono i parametri per far sì che si possa costruire un modello di un materiale non lineare microelastico con la teoria della peridinamica.

Il problema della rotazione rigida non viene preso in considerazione in quanto y è invariante rispetto alla rotazione del corpo⁹.

Una versione linearizzata della teoria del materiale microelastico consente di porre:

$$f(\eta, \xi) = \mathbf{C}(\xi)\eta \quad \forall \eta, \xi \quad (1.12)$$

con \mathbf{C} chiamata funzione micromodulo del materiale e che è un tensore del secondo ordine¹⁰:

$$\mathbf{C}(\xi) = \frac{\partial f}{\partial \eta}(\mathbf{0}, \xi) \quad \forall \xi \quad (1.13)$$

$$\mathbf{C}(-\xi) = \mathbf{C}(\xi) \quad \forall \xi \quad (1.14)$$

⁹ Vedi nota 2 p.10

¹⁰ Vedi nota precedente

1.3. Modello costitutivo

Per descrivere una frattura spontanea bisogna introdurre il concetto di limite di rottura per il materiale non lineare microelastico. Si assume che f dipenda solamente dal *bond stretch*:

$$s = \frac{|\xi + \eta| - |\xi|}{|\xi|} = \frac{y - |\xi|}{|\xi|} \quad (1.15)$$

Con questa notazione l'allungamento " s " rispetto alla configurazione iniziale è positivo quando il *bond* è in tensione e c' è un aumento di lunghezza, negativo se vi è un accorciamento dovuto a compressione. Se il materiale è isotropo non vi è alcuna dipendenza di f dalla direzione di ξ . Il modo più immediato per introdurre il concetto di limite di rottura è ipotizzare che il *bond* (modellizzato come una molla) si rompa qualora subisca un allungamento " s " maggiore a un limite prestabilito. Quando il *bond* si rompe non deve poter ritornare integro in nessun modo ma deve rimanere rotto per tutta la durata della simulazione.

Si consideri ora un "Prototype Microelastic Brittle material-PMB (vedere anche §1.10) il quale è definito come¹¹:

$$f(y(t), \xi) = g(s(t, \xi))\mu(t, \xi) \quad (1.16)$$

Con g che è una funzione lineare a valori scalari data da:

$$g(s) = c s \quad \forall s \quad (1.17)$$

Si ha quindi g costante e μ una funzione scalare dipendente dalla storia pregressa che può assumere solamente valore unitario oppure nullo¹²:

$$\mu(t, \xi) = \begin{cases} 1 & \text{se } s(t', \xi) < s_0 \text{ per ogni } 0 \leq t' \leq t \\ 0 & \text{per gli altri casi} \end{cases} \quad (1.18)$$

¹¹ Vedi nota 2 p.10

¹² Vedi nota precedente

$\mu=0$ significa che il *bond* è irrimediabilmente rotto e $\mu=1$ indica che è integro e la sua lunghezza dipende dal carico che gli è stato applicato.

Il carico critico del *bond*, s_0 , viene assunto costante (Figura 1-2¹³). Se il materiale inizialmente è isotropo quando si rompono dei *bond* la condizione di isotropia viene persa e nel passaggio di calcolo successivo il materiale risulta complessivamente anisotropo.

Inserendo la condizione di rottura nell'equazione del materiale del *bond* (1.16) si crea una condizione univoca di danno locale definita come equazione del danno¹⁴, essa rappresenta un indice con cui indicare il numero dei *bond* rotti sul totale dei *bond* presenti in una porzione di materiale:

$$\varphi(\mathbf{x}, t) = 1 - \frac{\int_{Hx} \mu(\mathbf{x}, t, \xi) dV_{\xi}}{\int_{Hx} dV_{\xi}} \quad (1.19)^{15}$$

In cui \mathbf{x} è incluso come argomento di μ . φ rappresenta il danno ed è compreso fra $0 \leq \varphi \leq 1$. Il valore zero rappresenta lo stato del materiale iniziale isotropo in cui tutti i *bond* sono presenti, mentre lo stato 1 rappresenta lo stato in cui il nodo si trova quando è stato completamente disconnesso dai nodi con cui inizialmente era collegato. Se i *bond* sono rotti non sostengono più nessun carico creando un indebolimento localizzato del materiale; il carico infatti non varia e continua ad essere applicato ai *bond*

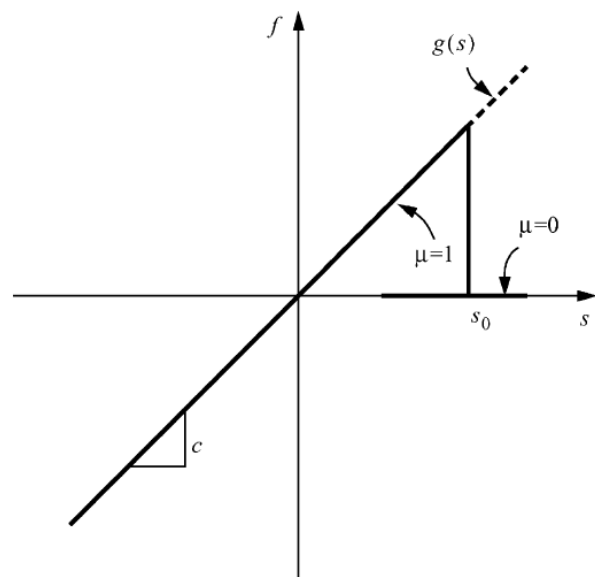


Figura 1-2 - Grafico s-f con l'andamento di μ

ancora integri. Questi ultimi sono soggetti ad una maggiore sollecitazione in quanto lo stesso carico viene ora suddiviso su un numero minore di *bond* rispetto al momento iniziale. I legami rimasti dovendo sopportare ora un carico maggiore hanno maggiore probabilità di raggiungere

¹³ Vedi nota 2 p.10

¹⁴ Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p.1528, 21/03/2005

¹⁵ Equazione del danno

la condizione di rottura e quindi di rompersi a loro volta. Da un singolo *bond* rotto si creano le condizioni affinché altri *bond* cedano e questo crea un effetto a cascata, ben visibile nelle simulazioni, che consente alla frattura di avanzare all'interno del corpo.

I due parametri che governano questo meccanismo sono la costante della molla del *bond* (*spring constant*) indicata con la lettera "c" e il valore critico del *bond* (*critical bond stretch*)¹⁶ indicato con "s₀". Si consideri ora un corpo omogeneo ed isotropo con s costante per ogni ξ e con η=s ξ, se si assume ξ=|ξ| e η=|η| si ha η=s ξ e dall'equazione del micropotenziale (1.6) si ha f=c s = c η/ξ da cui w= cη²/2ξ=c s² ξ/2 e dall'equazione (1.7) si ha:

$$W = \frac{1}{2} \int_{Hx} w dV_{\xi} = \frac{1}{2} \int_0^{\delta} \left(\frac{c s^2 \xi}{2} \right) 4 \pi \xi^2 d\xi = \frac{\pi c s^2 \delta^4}{4} \quad (1.20)$$

E' necessario fare ciò per poter eguagliare l'equazione (1.20) alla densità di energia di deformazione per lo stesso materiale e, tenendo presente che la deformazione calcolata è la stessa con la teoria classica dell'elasticità (W=9 k s²/2), è possibile successivamente ricavare la costante della molla (*spring*) per un materiale PMB che risulta essere di¹⁷:

$$c = \frac{18 k}{\pi \delta^4} \quad (1.21)^{18}$$

Il carico critico per avere la rottura dei *bond*, s₀, può essere considerato come una quantità misurabile che interagisce con una superficie interna piana di frattura di un corpo omogeneo sufficientemente grande. Per avere la separazione completa del corpo in due metà è necessario che la superficie di frattura si estenda fra due estremi opposti del corpo e che quindi tutti i *bond* dei nodi del corpo che attraversano quella superficie siano stati rotti. Il lavoro necessario per rompere un singolo *bond* viene indicato con w₀(ξ) ed è dato da¹⁹:

$$w_0(\xi) = \int_0^{s_0} g(s)(\xi ds) \quad \text{con } \xi = |\xi| \text{ ed inoltre } d\eta = \xi ds \quad (1.22)$$

¹⁶ Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p.1529, 21/03/2005

¹⁷ Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p.1529, 21/03/2005

¹⁸ Spring Constant

¹⁹ Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p.1529, 21/03/2005

Nel caso in esame di materiale PMB si ha che $w_0(\xi) = c s_0^2 \xi/2$. Viene quindi definito il lavoro (energia) necessario per rompere tutti i *bond* per unità di area (dell'area della superficie di frattura):

$$G_0 = \int_0^\delta \int_0^{2\pi} \int_z^\delta \int_0^{\cos^{-1} z/\xi} \left(\frac{c s_0^2 \xi}{2} \right) \xi^2 \sin\phi \, d\phi \, d\xi \, d\theta \, dz \quad (1.23)^{20}$$

Da cui una volta risolti gli integrali (vedi § 1.3.1) ed avendo G_0 si ottiene il valore numerico dell'energia per unità di superficie (della superficie di frattura) necessaria per avere la completa separazione del corpo in due metà.

$$G_0 = \frac{\pi c s_0^2 \delta^5}{10} \quad (1.24)$$

Poiché per i materiali PMB la quantità di energia è misurabile, dalla equazione (1.24) è possibile ricavare s_0 per avere la completa separazione ma con l'ipotesi di assenza di altri meccanismi dissipativi.

$$s_0 = \sqrt{\frac{10 G_0}{\pi c \delta^5}} = \sqrt{\frac{5 G_0}{9 k \delta}} \quad (1.25)^{21}$$

Per molti materiali le ipotesi di PMB sono limitative ed è poco accurato ritenere i *bond* indipendenti dalle condizioni degli altri *bond*. Perciò vengono introdotti dei coefficienti correttivi e delle relazioni che mettono in relazione i *bond* interessati dalla frattura con lo stato dei *bond* del resto del corpo. In un materiale microplastico invece le forze dei *bond* diventano²²:

$$f(s, \xi, t) = \begin{cases} c(s - \bar{s}(t)) & \text{se } |\xi| \leq \delta \\ 0 & \text{negli altri casi} \end{cases} \quad (1.26)$$

$$\bar{s}(0) = 0 \quad \dot{\bar{s}} = \begin{cases} \dot{\bar{s}} & \text{se } |s - \bar{s}| \leq s_Y \\ 0 & \text{negli altri casi} \end{cases} \quad (1.27)$$

²⁰ Equazione dell'energia di frattura nel caso tridimensionale

²¹ Allungamento critico nel caso tridimensionale, vedere nota 19

²² Macek M. W., Siling S.A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, p. 1170, 23/10/2007

A livello dei *bond* il materiale ha caratteristiche elastiche unite a caratteristiche di perfetta plasticità. A livello macroscopico il materiale si indurisce ed i *bond* non cedono tutti istantaneamente o alla stessa deformazione.

Le proprietà a livello dei *bond* non devono quindi essere in contrasto con le proprietà a livello macroscopico. Per un materiale micro fragile il cedimento dei *bond* è legato alla resistenza massima ingegneristica (*stress*, σ_{ult}) a cui tutti i *bond* cedono se viene raggiunta. Dalla (1.25) si ricava:

$$\sigma_{ult} \approx \int_0^\delta \int_0^{2\pi} \int_z^\delta \int_0^{\cos^{-1} z/\xi} f_Y \xi^2 \cos\phi \sin\phi \, d\phi \, d\xi \, d\theta \, dz = \frac{\pi f_Y \delta^4}{6} \quad (1.28)$$

Con f_y valore di snervamento del *bond*

$$f_y = c s_Y$$

(1.29)²³

L'equazione (1.28) risulta approssimata in quanto viene calcolata sulla geometria indeformata anziché su quella deformata. Dalle equazioni (1.21), (1.28) e (1.29) si ricava la (1.30) che mette in relazione lo snervamento del *bond* con la teoria del continuo convenzionale²⁴:

$$s_Y \approx \frac{\sigma_{ult}}{3k} \quad (1.30)$$

1.3.1. Riferimenti per il calcolo dell'energia di frattura

Per calcolare l'energia di frattura G_0 dalla (1.23) si ipotizza di prendere un punto A lungo la linea tratteggiata $0 \leq z \leq \delta$.

Il lavoro necessario per rompere il *bond* che collega A e B (con B qualsiasi punto appartenente alla parte di calotta sferica opposta alla superficie di frattura rispetto ad A) è dato dall'equazione (1.22) per i materiali PMB ($w_0(\xi)=c s_0^2 \xi/2$) risolta utilizzando le coordinate sferiche centrate in A. Vedere figura 1-3²⁵ per i riferimenti utilizzati nell'equazione (1.23).

²³ f_y , valore di snervamento del bond

²⁴ Vedi nota 22

²⁵ Siling S.A., Askary E. *A meshfree method based on the peridynamic model of solid mechanics*, Elsevier, 83 (2005) 1526-1535, p. 1529, 21/03/2005

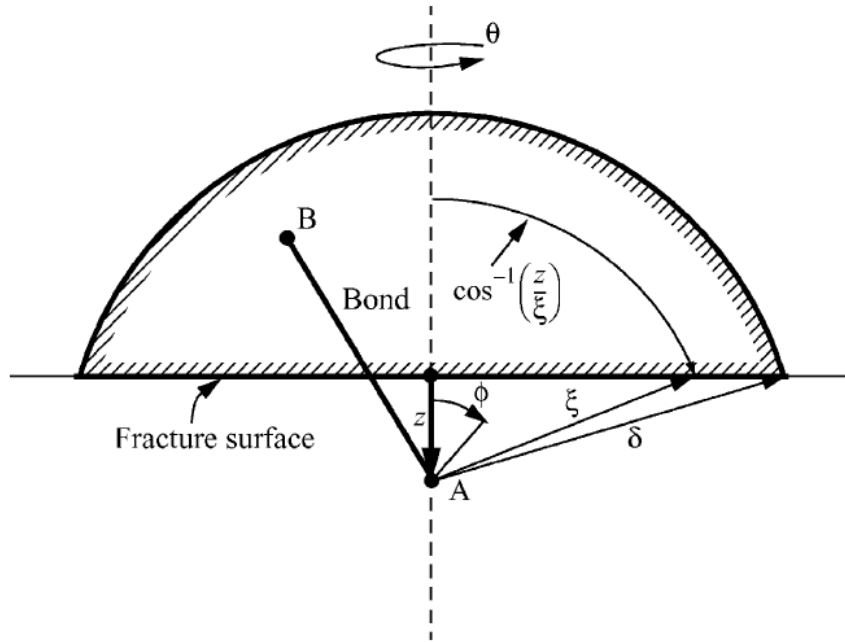


Figura 1-3 - Riferimenti per il calcolo dell'energia di frattura

1.4.Peridinamica 2D

In questo elaborato verrà preso in esame solo il caso della peridinamica bidimensionale. Come spiegato nell'articolo²⁶ esistono due tipi di micro-modulus function: la constant micro-modulus function e la conical micro-modulus functions. Negli esempi riportati nei capitoli successivi è stata utilizzata la prima tipologia di funzione.

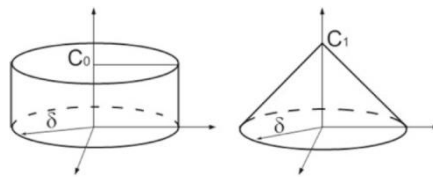


Figura 1-4 - Constant (sinistra) e Conical (destra) micro-modulus functions

Il micro-modulo con la constant micro-modulus function in 2D in condizioni di plane stress risulta essere:

$$c = c_0 = \frac{6E}{\pi \delta^3 (1 - \nu)}$$

(1.31)²⁷

²⁶ Ha Y. D., Bobaru F., *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 233, 20/1/2010

²⁷ Constant micro-modulus function

Nel caso invece di conical micro-modulus function si ha:

$$c = c_1 \left(1 - \frac{\xi}{\delta}\right) = \frac{24 E}{\pi \delta^3 (1 - \nu)} \left(1 - \frac{\xi}{\delta}\right) \quad (1.32)^{28}$$

Il valore del modulo di Poisson da inserire è di 1/3 ed è fisso per teoria della peridinamica bond-based quando si utilizzano materiali isotropi, lineari e microelastici. Nel § 1.10 è stato illustrato il perché di questo valore. Nei capitoli successivi e nelle simulazioni viene sempre utilizzato il valore di 1/3. La condizione che permette di stabilire se un *bond* è integro oppure rotto è chiamata “critical relative elongation” (allungamento critico relativo). Appena le forze applicate ad un *bond* esprimono come risultato un allungamento maggiore di quello critico, esso si rompe e su di esso non agiscono più forze le quali vengono quindi ripartite fra i *bond* limitrofi. Nell’articolo²⁹ viene esposto come è calcolata l’energia di frattura (G_0) che è presente nella formula dell’allungamento critico (s_0).

L’equazione (1.33) rappresenta s_0 per constant micro-modulus-function³⁰:

$$s_0 = \sqrt{\frac{4 \pi G_0}{9 E \delta}} \quad (1.33)^{31}$$

Formula di s_0 con conical micro-modulus function:

$$s_0 = \sqrt{\frac{5 \pi G_0}{9 E \delta}} \quad (1.34)^{32}$$

L’equazione (1.33) e l’equazione (1.34) sono molto importanti perché partendo dalle caratteristiche macroscopiche del materiale (densità ed energia di frattura) e dal valore dell’orizzonte (scelto dall’utente), sono in grado di definire in maniera univoca la condizione a cui un *bond* si rompe in base all’allungamento critico.

²⁸ Conical micro-modulus function

²⁹ Vedi nota 26 p.19

³⁰ Vedi nota precedente

³¹ Allungamento critico con constant micro-modulus function

³² Allungamento critico con conical micro-modulus function

1.5. Forze di corto raggio

Nella teoria della peridinamica le particelle influiscono reciprocamente attraverso i *bond* ma i nodi che sono ad una distanza superiore dell'orizzonte rispetto ad un altro nodo non interagiscono (1.1). È possibile però, dopo delle trasformazioni, che i nodi si trovino ad una distanza minore dell'orizzonte fino al contatto. Prima di arrivare al contatto i due nodi si respingono con una forza proporzionale alla distanza fra di essi. Per modellizzare ciò con la peridinamica vengono introdotte le forze di corto raggio, le quali non dipendono dalla distanza iniziale delle particelle ma dalla distanza a cui si trovano nella posizione corrente deformata³³:

$$f_s(\mathbf{y}', \mathbf{y}) = \frac{\mathbf{y}' - \mathbf{y}}{|\mathbf{y}' - \mathbf{y}|} \min\left\{0, c_s \left(\frac{|\mathbf{y}' - \mathbf{y}|}{2 r_s} - 1 \right)\right\}$$

(1.35)

Con c_s e r_s costanti positive e \mathbf{y} e \mathbf{y}' le posizioni deformate di \mathbf{x} e \mathbf{x}'

$$\mathbf{y} = \mathbf{x} + \mathbf{u}$$

$$\mathbf{y}' = \mathbf{x}' + \mathbf{u}'$$

(1.36)

(1.37)

Le forze di corto raggio intervengono solamente quando le due particelle sono più vicine di $2 r_s$, r_s è chiamato raggio del nodo. Le forze di corto raggio sono di compressione e quindi repulsive in accordo con (1.31). Esse tendono quindi ad ostacolare l'accorciamento del *bond*. Il valore di c_s è proporzionale a c (costante della molla – *constant spring*) per i materiali che variano proporzionalmente, e dipende dalla rigidità a cui si vuole arrivare con le forze di contatto. Un valore molto usato per i materiali che hanno proprietà elastiche è il seguente³⁴:

$$c_s = 15 c$$

(1.38)

³³ Richard W. Macek, Stewart A. Siling. *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, 23/10/2007

³⁴ Vedi nota precedente

1.6.Strategie di discretizzazione

Il corpo viene discretizzato in nodi che formano la griglia iniziale. Questa modellazione è “meshfree” (senza *mesh*) nel senso che non ci sono elementi finiti o connessioni geometriche come nella FEA. Viene utilizzata la teoria linearizzata della peridinamica per un corpo omogeneo. Dalla equazione (1.1) del moto si ricava la seguente sommatoria finita³⁵:

$$\rho \ddot{\mathbf{u}}_i^n = \sum_p \mathbf{f}(\mathbf{u}_p^n - \mathbf{u}_i^n, \mathbf{x}_p - \mathbf{x}_i) V_p + \mathbf{b}_i^n \quad (1.39)$$

\mathbf{f} deriva dalla equazione (1.9) ed “n” è il numero del passo temporale (*time step number*). Il pedice indica il numero del nodo:

$$\mathbf{u}_i^n = \mathbf{u}(\mathbf{x}_i, t^n) \quad (1.40)$$

V_p è il volume del nodo p. Siccome è necessario che le dimensioni delle quantità in esame siano corrette per l’espressione tridimensionale (1.1), nel caso monodimensionale si pone costante l’area della sezione della trave che modella il *bond* chiamandola A, si ha quindi:

$$V_p = A \Delta x \quad \text{con } \Delta x \text{ che è la spaziatura della griglia ed è costante} \quad (1.41)$$

La sommatoria (1.35) è riferita a tutti i nodi p che rispettano la seguente condizione:

$$|\mathbf{x}_p - \mathbf{x}_i| \leq \delta \quad (1.42)$$

La forma discretizzata della peridinamica linearizzata risulta quindi:

$$\rho \ddot{\mathbf{u}}_i^n = \sum_p \mathbf{C}(\mathbf{x}_p - \mathbf{x}_i)(\mathbf{u}_p^n - \mathbf{u}_i^n) V_p + \mathbf{b}_i^n \quad (1.43)$$

C è definito dalla (1.13).

³⁵ Siling S.A., Askary E. A meshfree method based on the peridynamic model of solid mechanics, Elsevier, 83 (2005) 1526-1535, p. 1530, 21/03/2005

L'accelerazione viene calcolata mediante una formula esplicita alle differenze centrali:

$$\ddot{\mathbf{u}}_i^n = \frac{\mathbf{u}_i^{n+1} - 2\mathbf{u}_i^n + \mathbf{u}_i^{n-1}}{\Delta t^2} \quad (1.44)$$

Con un time step ipotizzato costante Δt .

L'errore di troncamento associato al modello discretizzato può essere analizzato in una dimensione, assumendo il corpo omogeneo³⁶:

$$\rho \ddot{u}_i^n = \sum_p C(x_p - x_i)(u_p^n - u_i^n)V_p + b_i^n \quad (1.45)$$

Nell'equazione precedente il micro-modulus C è una funzione scalare.

Con queste ipotesi è possibile ricavare l'equazione del moto:

$$\rho \ddot{u}(x_i) = \int_{-\delta}^{\delta} C(\xi)(u(x_i + \xi) - u(x_i))A d\xi + b(x_i) \quad (1.46)$$

Che può essere scritta anche come³⁷:

$$\rho \ddot{u}(x_i) = A \sum_p \int_{-\Delta x/2}^{\Delta x/2} C(x_p - x_i + \xi)(u(x_p + \xi) - u(x_i))d\xi + b(x_i) \quad (1.47)$$

Sia C che " u " sono funzioni continue e differenziabili, è quindi possibile espandere C nell'intorno di $x_p - x_i$ e " u " nell'intorno di x_p utilizzando la serie di Taylor fino al secondo termine:

$$C(x_p - x_i + \xi) = C(x_p - x_i) + C'(x_p - x_i)\xi + O(\xi^2) \quad (1.48)$$

$$u(x_p + \xi) = u(x_p) + u'(x_p)\xi + O(\xi^2) \quad (1.49)$$

³⁶ Vedi nota 35 p.22

³⁷ Siling S.A., Askary E. *A meshfree method based on the peridynamic model of solid mechanics*, Elsevier, 83 (2005) 1526-1535, p. 1531, 21/03/2005

Combinando le equazioni (1.42), (1.43) e (1.44) ed integrando si ha³⁸:

$$\rho \ddot{u}(x_i) = \sum_p C(x_p - x_i) (u(x_p) - u(x_i)) V_p + b(x_i) + O(\Delta x^2) \quad (1.50)$$

Quando ci sono discontinuità in C e in u , l'ipotesi di continuità e derivabilità non è più valida e l'errore in queste condizioni diventa di $O(\Delta x) + O(\Delta t^2)$. Un condizione di stabilità utilizzata è la stabilità di Von Neumann:

$$u_i^n = \zeta^n \exp(\kappa i \sqrt{-1}) \quad (1.51)$$

Con κ numero reale positivo e ζ un numero complesso. Bisogna quindi trovare la condizione di Δt per cui $|\zeta| \leq 1$ per ogni κ . Questa condizione è necessaria per non avere divergenza. Con $q = p - i$ e $C_q = C(x_p - x_i)$ si ha³⁹:

$$\frac{\rho}{\Delta t^2} (\zeta - 2 + \zeta^{-1}) = \sum_{q=-\infty}^{\infty} A C_q (\exp(\kappa q \sqrt{-1}) - 1) \Delta x = \sum_{q=1}^{\infty} 2 A C_q (\cos \kappa q - 1) \Delta x \quad (1.52)$$

$$M_\kappa = \sum_{q=1}^{\infty} A C_q (1 - \cos \kappa q) \Delta x \quad (1.53)$$

Dalle equazioni (1.48) e (1.49) si ha:

$$\zeta = 1 - \frac{M_\kappa \Delta t^2}{\rho} \pm \sqrt{\left(\left(1 - \frac{M_\kappa \Delta t^2}{\rho} \right)^2 - 1 \right)} \quad (1.54)$$

Dalla condizione $|\zeta| \leq 1$ si ha:

$$\Delta t < \sqrt{\frac{2\rho}{M_{kappa}}} \quad \forall \kappa \quad (1.55)$$

³⁸ Vedi nota 37 p.23

³⁹ Vedi nota precedente

Per essere sicuri che κ possa assumere qualsiasi valore, dalla equazione precedente si ha⁴⁰:

$$M_{\kappa} \leq 2 A \Delta x \sum_{q=1}^{\infty} C_q \quad (1.56)$$

E quindi affinché l'equazione (1.50) sia soddisfatta si deve avere:

$$\Delta t < \sqrt{\frac{\rho}{A \Delta x \sum_{q=1}^{\infty} C_q}} \quad (1.57)$$

Nella forma più utilizzata⁴¹:

$$\Delta t < \sqrt{\frac{2 \rho}{\sum_p V_p C_{ip}}} \quad (1.58)$$

Con $C_{ip} = C(x_p - x_i)$, inoltre questa forma non richiede che i nodi siano numerati sequenzialmente ed è applicabile per ogni numero di dimensioni con $C_{ip} = |C(x_p - x_i)|$.

Nel caso di materiali non lineari la condizione di stabilità deve derivare da (è sempre comunque consigliato applicare anche un fattore di sicurezza):

$$C_{ip} = \left| \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}} \right| \quad (1.59)$$

1.7. Normalizzazione di forza alle superfici

Come dimostrato dall'equazione (1.21) la costante della molla può essere ottenuta, per un materiale reale, dal modulo di massa. L'assunzione precedente afferma che all'interno del raggio dell'orizzonte il materiale è sempre lo stesso ma questo non è sempre vero se le altre particelle sono a contatto con materiali diversi o se si è in prossimità dei bordi liberi. Se si usa l'assunzione che c sia costante non si modellizzano correttamente le caratteristiche e le proprietà del materiale. Per tenere conto di questo effetto in prossimità delle superfici libere e

⁴⁰ Vedi nota 37 p.23

⁴¹ Vedi nota precedente

delle interfacce con altri materiali viene usata una correzione detta normalizzazione di forza (*force normalization*), così facendo se \mathbf{X} è un punto del corpo con le caratteristiche sopra descritte il suo tensore $\mathbf{P}(\mathbf{X})$ è definito come⁴²:

$$\mathbf{P}(\mathbf{X}) = \frac{\partial}{\partial \mathbf{U}} \int_{Hx} \mathbf{f}(\mathbf{U}, \mathbf{X} - \mathbf{x}') dV_{x'} \quad (1.60)$$

Si ha quindi che $-\mathbf{P}(\mathbf{X}) \Delta \mathbf{u}$ eguaglia la forza di richiamo (per unità di volume) che la particella X ha su di sé se è stata spostata di un piccolo vettore $\Delta \mathbf{u}$ con il resto dei punti che rimangono fissi. Si può dimostrare che \mathbf{P} è un tensore simmetrico ed anche positivo per molti materiali. Quindi \mathbf{P} ha tre autovalori positivi $\{p^1, p^2, p^3\}$, con p^1 maggiore rispetto agli altri. Ora si costruisce \mathbf{P}_∞ che è l'analogo tensore di \mathbf{P} ottenuto però per un grande corpo omogeneo, a differenza di $\mathbf{P}(\mathbf{X})$ che era stato ottenuto per particolari posizioni di \mathbf{X} . Gli autovalori di \mathbf{P}_∞ sono allora $\{p^1_\infty, p^2_\infty, p^3_\infty\}$ con p^1_∞ il maggiore dei tre. Da questi due tensori si calcola c_∞ dall'equazione 21. Facendo così si può modificare la costante della molla $c(\mathbf{X})$ nelle situazioni vicino alla superficie e nelle zone di interferenza (ossia nelle zone vicino al punto \mathbf{X}).

$$c(\mathbf{X}) = \frac{c_\infty p^1_\infty}{p^1(\mathbf{X})} \quad (1.61)$$

In generale $c(\mathbf{X}) \geq c_\infty$

Si ha quindi che la rigidità locale scala in rapporto agli autovalori del tensore \mathbf{P} . Questo metodo è stato sviluppato per consentire di avere una risposta corretta con strutture fini e che con strutture che contengono superfici libere o vincolate con altri materiali.

1.8. Implementazione con gli elementi finiti

L'interazione che intercorre fra le particelle attraverso i *bond* rappresenta la forza fisica fra le coppie di particelle, mentre l'insieme dei *bond* crea una rete che collega fra loro le particelle. Se la rete dei *bond* è lagrangiana (ossia i *bond* sono definiti nel riferimento della configurazione indeformabile) allora le connessioni non cambiano durante le deformazioni. Per poter incrementare questa teoria nei software di calcolo è necessario creare una *mesh* di tutti *truss*

⁴² Macek M. W., Siling S.A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, 23/10/2007

con appropriate caratteristiche di rigidezza i quali rappresentano i *bond*. Dall'equazione (1.35) moltiplicando per il volume V_i si ottiene un'equazione utilizzabile per gli elementi finiti⁴³:

$$V_i \rho \ddot{\mathbf{u}}_i^n = \sum_p \mathbf{f}(\mathbf{u}_p^n - \mathbf{u}_i^n, \mathbf{x}_p - \mathbf{x}_i) V_p V_i + \mathbf{b}_i^n V_i \quad (1.62)$$

e raccogliendo ed evidenziando i termini si ha:

$$\mathbf{M} \ddot{\mathbf{u}}^n + \mathbf{F}_T^n = \mathbf{F}_e^n \quad (1.63)$$

Con \mathbf{M} : Matrice delle masse concentrate i cui termini diagonali sono $V_i \rho$

\mathbf{F}_e^n : vettore delle forze esterne i cui componenti sono $\mathbf{b}_i^n V_i$

\mathbf{F}_T^n : vettore delle forze interne e i cui componenti sono $\sum_p \mathbf{f}(\mathbf{u}_p^n - \mathbf{u}_i^n, \mathbf{x}_p - \mathbf{x}_i) V_p V_i$ che è la somma di tutte le forze delle travi connesse al nodo i .

Quindi il processo di implementazione della peridinamica con il codice agli elementi finiti si riduce ad una fase di *preprocessing* in cui si crea la rete di *truss* con le caratteristiche e proprietà ricavate dalla teoria della peridinamica.

Sebbene non vi siano problemi dal punto di vista tecnico per generare una rete con elementi *truss* per una qualsiasi figura complessa, questo modo di operare non è sempre conveniente. Infatti anche per pezzi di piccole dimensioni si crea un elevato numero di *bond*. Ad esempio nel caso tridimensionale per $\delta=3\Delta x$ si creano approssimativamente 114 travi per ogni nodo interno, creando milioni di elementi anche per oggetti di piccole dimensioni. Per avere tempi di calcolo ragionevoli è necessario modellizzare solo le parti di maggiore interesse con la peridinamica lasciando, se possibile, gli elementi finiti con la teoria del continuo nelle parti meno importanti. La peridinamica viene quindi utilizzata in quelle aree, che saranno chiamate subregioni, in cui avvengono fratture o cricche in quanto molto più efficace rispetto all'analisi agli elementi finiti. Occorre però porre attenzione all'accoppiamento fra i nodi della peridinamica ed i nodi dell'analisi agli elementi finiti. I bordi in peridinamica non sono strettamente definiti, i dati iniziali quali spostamenti o carichi non vengono applicati alla superficie bensì ad un volume

⁴³ Macek M. W., Siling S.A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, p. 1172, 23/10/2007

finito. L'accoppiamento deve quindi essere effettuato in una regione che si espande per qualche spessore dalla superficie esterna (quale, ad esempio, il raggio dell'orizzonte).

Viene quindi introdotto un vincolo di spostamento fra il "confine" della peridinamica ed il confine della *mesh* FEA.

Se p rappresenta i nodi con una distanza pari all'orizzonte (confine peridinamica) e h rappresenta i nodi o gli elementi relativi alla FEA classica con cui i primi sono in relazione, allora si ha il seguente vincolo:

$$\mathbf{u}_p = \mathbf{N}(a, b, c) \mathbf{u}_h \quad (1.64)$$

con (a, b, c) che sono le coordinate parametriche dei nodi p all'interno o estrapolati dagli elementi h ed \mathbf{N} è la matrice di forma per gli elementi h . Il vincolo definito da quest'ultima equazione rende rispettata la continuità fra la parte modellata con peridinamica e la parte modellata con elementi finiti.

Viene quindi creata una zona in cui le due teorie si sovrappongono, generalmente di spessore δ , le travi vengono quindi incorporate negli elementi continui. Affinché questa regione non diventi troppo rigida la densità e il modulo elastico degli elementi continui che vengono ospitati nella zona in comune con la peridinamica hanno valori molto piccoli.

La massa dei nodi e la rigidezza dei *bond* sono derivate dalla teoria e la massa concentrata dei nodi è definita come ρV_i mentre la massa degli elementi trave è nulla.

Dalle equazioni (1.58) e (1.59) possibile definire delle nuove proprietà per l'area e il modulo elastico delle travi di questa zona in comune⁴⁴:

$$A = (V_i V_p)^{1/3} \quad E = c(V_i V_p)^{2/3} \quad (1.65)$$

$$(1.66)$$

e nel caso di una *mesh* rettangolare uniforme si ha:

$$A = \Delta x^2 \quad E = c \Delta x^4 \quad (1.67)$$

$$(1.68)$$

⁴⁴ Macek M. W., Siling S.A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, p. 1173, 23/10/2007

Inoltre il legame s del *bond* è identico alla deformazione ingegneristica della trave, così la frattura della trave si ha per una determinata deformazione⁴⁵:

$$\epsilon_f = s_0 \quad (1.69)$$

Per materiali elasto-plastici con modulo di Poisson di $\frac{1}{4}$ l'equazione (1.30) descrive, in funzione del modulo elastico E e dello sforzo massimo di tensione σ_{ult} , la deformazione della trave:

$$s_Y \approx \frac{\sigma_{ult}}{2E} \quad (1.70)$$

Per l'implementazione agli elementi finiti si pone $\epsilon_f = s_Y$ con s_Y definito dalla precedente equazione.

1.9. Coefficiente correttivo del volume

La teoria prevede che tutti i nodi che sono all'interno della sfera di influenza di raggio pari all'orizzonte δ interagiscano con le equazioni della peridinamica. Avendo però discretizzato il volume in celle queste, pur avendo il nodo centrale della cella al di fuori dell'orizzonte, possono avere una frazione interna a δ che, se trascurata comporterebbe una errata valutazione dei contributi volumetrici. È possibile minimizzare questo errore creando dei *bond* con caratteristiche specifiche che collegano la cella di riferimento con la cella esterna al *bond* ma con le caratteristiche sopra descritte. Viene quindi creata una funzione β che calcola quanto il nodo famiglia interagisce con il nodo centrale. In Matlab è stato creato un orizzonte di ricerca maggiore rispetto all'orizzonte teorico con lo scopo di cercare questi nodi famiglia che interagiscono parzialmente. Il valore β risultante viene poi moltiplicato per il valore teorico della resistenza dei *bond* interni alla sfera di influenza⁴⁶.

⁴⁵ Vedi nota 44 p.28

⁴⁶ Scudellaro L., *Nonlocal theories for the study of continuum: a finite elements method implementation*, Università degli studi di Padova, 2012

Nel caso tridimensionale si ha:

$$\beta = \begin{cases} 1 & \text{per } |\xi| \leq \delta - 0.5\Delta x \\ \frac{\delta + 0.5\Delta x - |\xi|}{\Delta x} & \text{per } \delta - 0.5\Delta x < |\xi| \leq \delta + 0.5\Delta x \\ 0 & \text{negli altri casi} \end{cases} \quad (1.71)$$

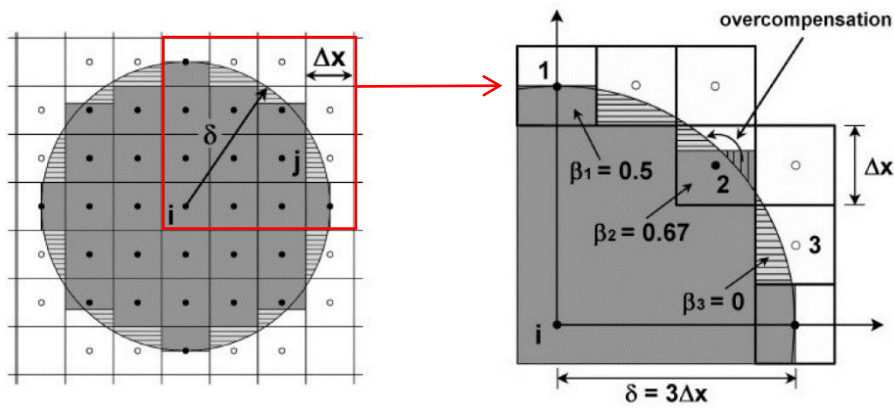
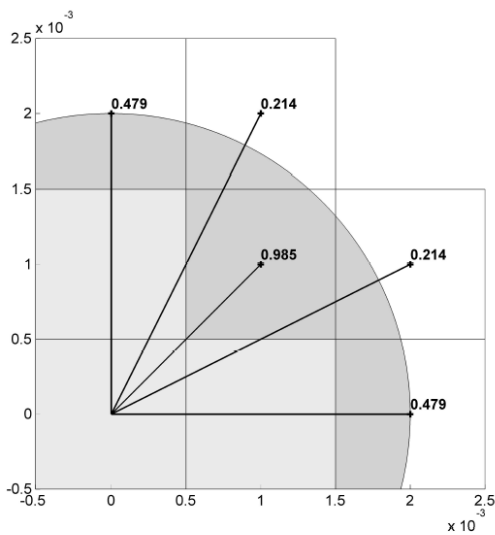
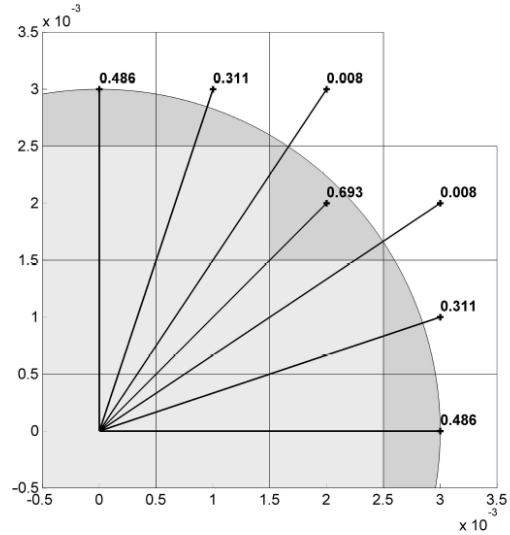


Figura 1-5 Coefficiente correttivo beta del volume

Dalla figura 1-5 si può vedere come le celle che possiedono il nodo famiglia all'esterno dell'orizzonte non possono essere totalmente trascurate e vanno prese in considerazione in base a (1.71). Questa procedura è anche in grado di rilevare e assegnare il giusto peso a celle aventi il nodo famiglia all'interno dell'orizzonte ma con il volume non interamente contenuto all'interno di questo. Il numero dei valori di beta aumenta all'aumentare di m. Nelle figure 1-6 e 1-7 sono riportati alcuni esempi di come i valori di beta e si nota come questi siano influenzati dal valore del passo.

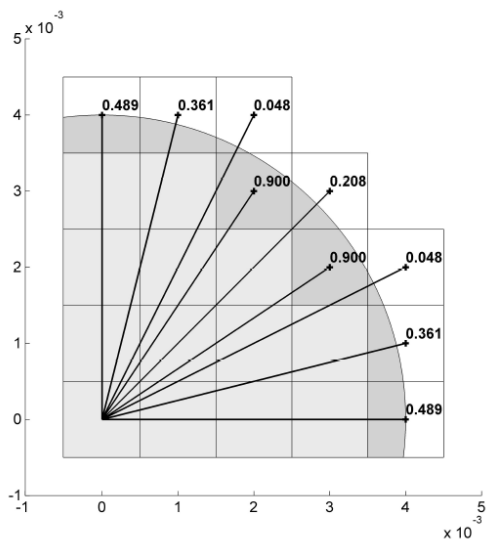


m=2

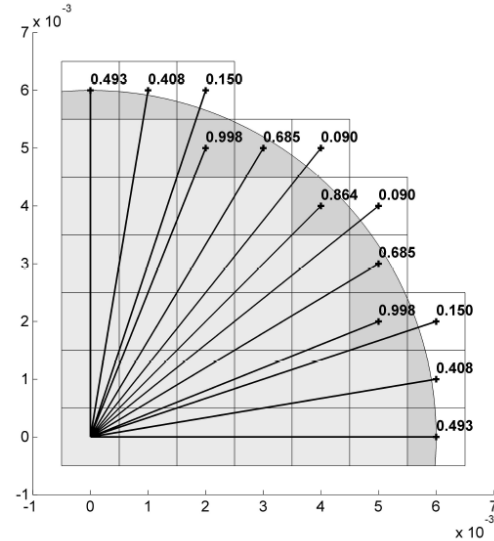


m=3

Figura 1-6 Valori di beta con m=2 e m=3



m=4



m=6

Figura 1-7 Valori di beta con m=4 e m=6

1.10. Coefficiente di Poisson nella teoria della peridinamica bond-based

Il principale svantaggio della peridinamica bond-based è che il coefficiente di Poisson è fisso ed è indipendente dal materiale utilizzato. Questo non crea problemi se il materiale ha un valore vicino ad $1/4$ nel caso tridimensionale o $1/3$ nel caso bidimensionale. Questa limitazione non c'è nella peridinamica stat-10-based che però non viene trattata in questo elaborato. Il fatto di avere il coefficiente di Poisson fisso deriva dal fatto che viene confrontato il tensore dello stress della densità di forza con il tensore degli stress classico di un materiale micro elastico della

teoria *bond*-based della peridinamica. Data una base ortonormale $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, si assume di avere un corpo infinito e microelastico lineare nella configurazione di riferimento e che sia sottoposto ad una deformazione omogenea data d:

$$u_1 = c_{11}x_1 \quad u_2 = u_3 = 0$$

Da cui dall'equazione 2 si ha:

$$\eta_1 = c_{11}\xi_1 \quad \eta_2 = \eta_3 = 0$$

Dalla formula della pairwise force function si ottiene:

$$\begin{aligned} \mathbf{f}(\boldsymbol{\eta}, \boldsymbol{\xi}) = \mathbf{C}(\boldsymbol{\xi})\boldsymbol{\eta} + \mathbf{f}(\mathbf{0}, \boldsymbol{\eta}) &= \lambda(\boldsymbol{\xi}) \begin{bmatrix} \xi_1^2 & \xi_1\xi_2 & \xi_1\xi_3 \\ \xi_2\xi_1 & \xi_2^2 & \xi_2\xi_3 \\ \xi_3\xi_1 & \xi_3\xi_2 & \xi_3^2 \end{bmatrix} \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{Bmatrix} + F(\mathbf{0}, \boldsymbol{\xi}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{Bmatrix} \\ &= \begin{Bmatrix} c_{11}[\lambda(\boldsymbol{\xi})\xi_1^3 + F(\mathbf{0}, \boldsymbol{\xi})\xi_1] \\ c_{11}\lambda(\boldsymbol{\xi})\xi_1^2\xi_2 \\ c_{11}\lambda(\boldsymbol{\xi})\xi_1^2\xi_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \end{aligned}$$

(1.72)

Le nove componenti del tensore degli stress all'origine sono:

$$\sigma_{ij} = \tau_j(\mathbf{e}_i) = \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_i dV_{x'} d\hat{l} \quad \text{per } i, j = 1, 2, 3$$

(1.73)

Per calcolare la prima componente del tensore degli stress ci si pone in un sistema di coordinate sferiche eseguendo un cambio di variabili:

$$\xi_1 = |\boldsymbol{\xi}| \cos \theta$$

$$\xi_2 = |\boldsymbol{\xi}| \sin \theta \cos \phi$$

$$\xi_3 = |\boldsymbol{\xi}| \sin \theta \sin \phi$$

$$r = |\boldsymbol{\xi}|$$

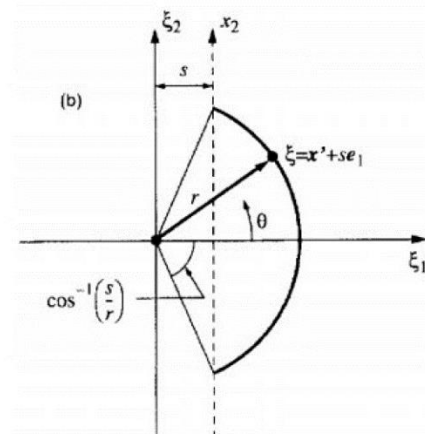


Figura 1-8 Riferimenti per cambio di variabile

Combinando le equazioni (1.67) e (1.68) e (1.69) si ricava⁴⁷:

$$\begin{aligned}
 \sigma_{11} &= \tau_1(\mathbf{e}_1) = \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_1 dV_{x'} d\hat{l} \\
 &= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}\frac{s}{r}} \int_0^{2\pi} [\lambda(r)(r \cos\theta)^3 + F(0,r)(r \cos\theta)] r^2 \sin\theta \, d\phi \, d\theta \, ds \, dr \\
 &= c_{11} \left[\frac{2\pi}{5} \int_0^\infty \lambda(r) r^6 \, dr + \frac{2\pi}{3} \int_0^\infty F(0,r) r^4 \, dr \right] = c_{11}(\Lambda + \Psi)
 \end{aligned}
 \tag{1.74}$$

$$\begin{aligned}
 \sigma_{12} &= \tau_2(\mathbf{e}_1) = \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_2 dV_{x'} d\hat{l} = \\
 &= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}\frac{s}{r}} \int_0^{2\pi} [\lambda(r)(r \cos\theta)^2 r \sin\theta \cos\phi] r^2 \sin\theta \, d\phi \, d\theta \, ds \, dr = 0
 \end{aligned}
 \tag{1.75}$$

$$\begin{aligned}
 \sigma_{13} &= \tau_3(\mathbf{e}_1) = \int_{\mathcal{L}} \int_{\mathcal{R}^+} f_3 dV_{x'} d\hat{l} = \\
 &= c_{11} \int_0^\infty \int_0^r \int_0^{\cos^{-1}\frac{s}{r}} \int_0^{2\pi} [\lambda(r)(r \cos\theta)^2 r \sin\theta \sin\phi] r^2 \sin\theta \, d\phi \, d\theta \, ds \, dr = 0
 \end{aligned}
 \tag{1.76}$$

$$\Lambda = \frac{2\pi}{5} \int_0^\infty \lambda(r) r^6 \, dr
 \tag{1.77}$$

$\Psi = 0$ in quanto l'equazione a esso riferito rappresenta la configurazione di riferimento priva di stress. Con lo stesso metodo si calcolano le altre sei componenti. Per \mathbf{e}_2 il cambio di variabile è il seguente:

$$\xi_1 = |\xi| \sin\theta \sin\phi$$

$$\xi_2 = |\xi| \cos\theta$$

$$\xi_3 = |\xi| \sin\theta \cos\phi$$

$$r = |\xi|$$

⁴⁷ Kebing Yu. *Enhanced Integration methods for the perydynamic theory*, Kansas State University, 2011

Il tensore degli stress diventa quindi:

$$[\boldsymbol{\sigma}] = c_{11} \begin{bmatrix} \Lambda & 0 & 0 \\ 0 & \frac{\Lambda}{3} & 0 \\ 0 & 0 & \frac{\Lambda}{3} \end{bmatrix} \quad \text{Dalla teoria della Peridinamica}^{48}$$

$$[\boldsymbol{\sigma}] = c_{11} \begin{bmatrix} \lambda + 2\mu & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \quad \text{Dalla teoria classica dell'elasticità}^{49}$$

Con

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad \mu = \frac{E}{2(1 + \nu)}$$

(1.78)

(1.79)

(1.80)

(1.81)

Si ha quindi il seguente sistema:

$$\begin{cases} \Lambda = \lambda + 2\mu \\ \frac{\Lambda}{3} = \lambda \end{cases}$$

Da cui si ha:

$$\mu = \frac{\Lambda}{3} \quad \nu = \frac{1}{4} \quad E = \frac{5\Lambda}{9}$$

E quindi il coefficiente di Poisson nel caso tridimensionale è pari a $1/4$ ⁵⁰.

1.11. Materiale Microelastico PMB (Prototype Microelastic Brittle material)

E' un materiale che viene utilizzato nella peridinamica bond-base le cui caratteristiche lo rendono molto simile alla molla della teoria della meccanica classica, un *bond* con questo materiale ha le seguenti proprietà⁵¹:

- L'allungamento del *bond* è proporzionale alla forza applicata ad esso e alla sua rigidezza
- Se il *bond* è sottoposto a trazione e viene superato il valore di s_0 , detto allungamento critico, il *bond* si rompe definitivamente e non può più ritornare integro anche se la forza applicata dovesse diminuire. Il modello è quindi dipendente dal tempo e dalla storia che precede il punto di inizio delle simulazioni. I *bond* rotti non sopportano più alcun carico che va quindi a gravare sui *bond* ancora integri.

⁴⁸ Vedi nota 46 p.29

⁴⁹ Vedi nota precedente

⁵⁰ Vedi nota precedente

⁵¹ Kebing Yu. *Enhanced Integration methods for the perydinamic theory*, Kansas State University, 2011

- Un *bond* è sottoposto a compressione non si rompe mai

In figura 1-9 è rappresentato il grafico allungamento - forza di un materiale PMB⁵².

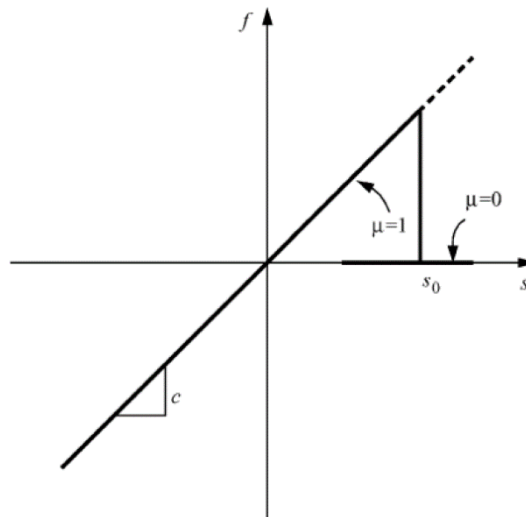


Figura 1-9 Curva forza peridinamica - allungamento del materiale PMB

⁵² Yu K., *Enhanced Integration methods for the perydynamic theory*, Kansas State University, 2011

Tovo G., *Studio della teoria peridinamica per sistemi assialsimmetrici: sviluppo ed implementazione*, Università degli studi di Padova, 2012

2. Propagazione delle cricche in piastre di piccolo spessore

In questo capitolo è stato preso in esame l'articolo di Youn Doh Ha e Florin Bobaru "*Studies of Dynamic crack propagation and crack branching with peridynamics*"⁵³ in cui viene descritto il comportamento di una piastra con geometria definita sottoposta a trazione. I due autori espongono la teoria della peridinamica e mettono in evidenza i pregi di quest'ultima per lo studio della propagazione delle cricche al suo interno. Viene quindi esaminata una piastra con un cricca a metà altezza e di lunghezza metà del provino (le dimensioni sono indicate nella figura 2-1). Su ogni lato è applicato una tensione di 12 MPa senza ulteriori vincoli esterni. Il problema viene risolto solo nel caso piano in quanto si trascura la dimensione relativa alla profondità perché ipotizzata così elevata tale da giungere a questa approssimazione.

I materiali utilizzati sono di due tipologie di vetro con i relativi dati specificati nella figura 2-2. Il primo, *Duran 50 glass*, presenta un modulo elastico minore rispetto al secondo, *Soda Lime Glass*, mentre il secondo presenta una energia di frattura minore rispetto al primo. Le caratteristiche appena evidenziate fanno sì che il percorso con cui si propaga il danno nella piastra sia diverso. Come si vedrà dalle simulazioni nel caso del *Duran 50 Glass* la rottura è una cricca continua con biforcazione iniziale, invece nel caso del *Soda Lime Glass* vi sono diverse ramificazioni. I due esempi verranno trattati rispettivamente nei § 2.1 e 2.4.

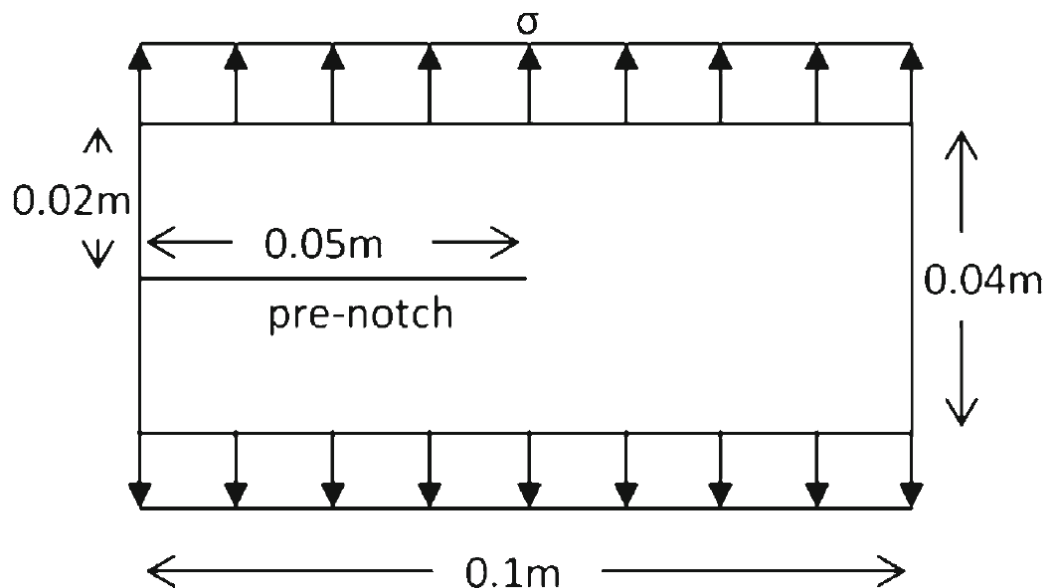


Figura 2-1 - Geometria della piastra

⁵³ Ha Y. D., Bobaru F.. *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

	Density (ρ) (kg/m ³)	Young's modulus (E) (GPa)	Poisson ratio (ν)	Fracture energy (G_0) (J/m ²)
Duran 50 glass	2,235	65	0.2	204
Soda-lime glass	2,440	72	0.22	135

Figura 2-2 - Tabella dei materiali utilizzati

La teoria della peridinamica, come spiegato nel capitolo precedente, discretizza il materiale in nodi collegati fra loro da elementi rettilinei detti *bond*. La scelta di come creare la rete che collega il nodo di riferimento ai nodi famiglia che gli stanno attorno spetta all'utente. L'ideale sarebbe collegare il maggior numero possibile di nodi famiglia al nodo di riferimento, ma questo comporterebbe risorse di calcolo molto elevate.

Nelle simulazioni effettuate si vede, ad esempio, che confrontando due strutture con lo stesso numero di nodi ma aventi diversi valori di m^{54} (numero di passi fra due nodi), e quindi con orizzonte che aumenta (§ 2.1.1. e § 2.1.4), il numero dei *bond* è raddoppiato. Gli autori utilizzano un "codice colori" per rappresentare intuitivamente il livello di danno della piastra calcolato con l'equazione di danno (1.19). In questo codice vengono associati i colori in base a valori ben precisi di danno, con il blu il danno è nullo, fino ad arrivare con varie gradazioni al rosso in cui vi è la rottura di almeno il 45% dei *bond* di quell'area. Se si immagina una superficie di taglio che passa tra due nodi, si ha che tutti i *bond* che collegano i due nodi e attraversano la superficie risultano rotti. Quanto appena descritto implica che entrambi i nodi hanno i *bond* dalla parte opposta del nodo rispetto alla cricca che sono integri. Sono inoltre integri anche i *bond* paralleli alla cricca. Questo fa sì che per avere la creazione di una zona di frattura sia sufficiente che poco meno della metà dei *bond* siano rotti. I pochi *bond* che dovessero rimanere collegati nella zona di frattura sono al limite del valore di allungamento e non è sicuro considerarli ancora in grado di tenere unite le due parti che si stanno dividendo.

Sono di seguito state ricreate le stesse situazioni descritte nell'articolo con il programma Abaqus, per la parte di elaborazione numerica, e in un secondo tempo attraverso Matlab sono state create delle immagini con il medesimo codice colori. L'elaborazione grafica eseguita dal secondo programma è stata effettuata introducendo i file di report generati da Abaqus al termine delle simulazioni. In questo modo si possono confrontare le immagini e notare che le differenze sono praticamente nulle tranne in alcuni casi in cui la simulazione in Abaqus ha dato

⁵⁴ Si prenda come esempio il caso 2.1.1 con $m=2$ e $\Delta x=0.001$ m e il caso 2.1.4 con $m=3$ e $\Delta x=0.001$ m

risultati lievemente diversi dalle figure di riferimento ma, come verrà spiegato nei singoli casi, sono da ritenersi migliori quelle risultanti da Abaqus.

Per valutare come cambiavano i risultati al variare del passo e dell'orizzonte si è seguito il criterio della "m convergenza" e della "δ convergenza". Nel caso della m convergenza è stato lasciato inalterato il valore dell'orizzonte e si è aumentato in ogni simulazione il valore di m. Così facendo il passo fra i nodi si riduce ed aumentano il numero di nodi della piastra ed il numero di nodi collegati a ciascun nodo di riferimento. Come si vedrà dai risultati delle simulazioni il livello di dettaglio aumenta con l'aumentare di m ed anche le figure risultano via via sempre più definite. Nel caso della "δ convergenza" la variabile che rimane inalterata è il valore di m e quindi ogni singolo nodo di riferimento ha sempre lo stesso numero di nodi a se collegati. Diminuisce perciò il valore dell'orizzonte assieme al valore del passo e di conseguenza aumenta il numero dei nodi complessivi.

Nella figura 2-3⁵⁵ sono illustrati graficamente i due criteri.

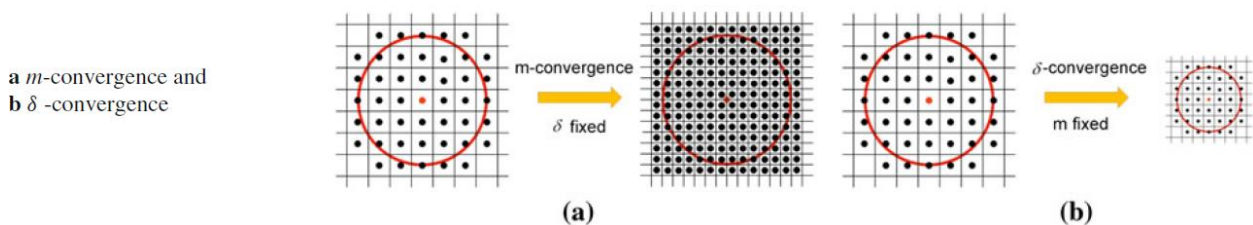


Figura 2-3 - Tipologie di convergenza: m-convergenza δ-convergenza

Il valore dell'orizzonte utilizzato realmente da Matlab per la ricerca dei nodi che sono collegati al nodo di riferimento, per poi successivamente creare il *bond*, è stato inserito aumentato di mezzo passo. Questo è stato eseguito sia per motivi puramente di approssimazione del programma sia per poter individuare i nodi che facevano parte dei volumi parziali all'interno dell'orizzonte, come spiegato nel § 1.9.

⁵⁵ Vedi nota 53 p.37

Il materiale della piastra utilizzato per le simulazioni delle due tipologie di convergenza è il Duran 50 glass che ha le seguenti caratteristiche:

E (Modulo di Young): $6.50 \cdot 10^{10}$ Pa

ν (Modulo di Poisson): 0.33

Densità: 2235 kg/m^3

G_0 (Fracture Energy): 204 J/m^2

Non essendo la piastra composta da elementi finiti, ma bensì da nodi e *bond*, per modellizzare la massa della piastra si è associato ad ogni nodo un valore di massa tale per cui la massa totale sia la somma della massa dei singoli nodi.

Per i *bond* essendo fisicamente degli elementi trave ad un grado di libertà, il programma Abaqus richiede che venga associata obbligatoriamente una massa anche a loro, la quale è calcolata internamente attraverso l'inserimento di una densità. Viene quindi assegnata ad essi una densità fittizia di $1 \cdot 10^{-18} \text{ kg/m}^3$. In questo modo, anche con milioni di elementi, la massa risultante è irrilevante e molti ordini di grandezza inferiore rispetto alla massa dell'intera struttura. Un altro vantaggio di fare le scelte sopra descritte è che durante il propagarsi della cricca i *bond* si rompono mentre i nodi si spostano senza venire mai eliminati durante l'intera durata della simulazione. La perdita di massa della struttura quindi è dovuta solo all'eliminazione dei *bond*, che avviene una volta che questi superano il proprio allungamento critico. Tale perdita di massa non influisce sui risultati perché la densità di questi è bassissima. Questa è una differenza notevole rispetto alle analisi con gli elementi finiti in cui una volta eliminato l'elemento dalla simulazione, perché considerato rotto, si porta con se la sua massa. La perdita della massa del modello dovuta ad un singolo elemento finito non può essere considerata ininfluenza. Tale perdita risulta dannosa ai fini dei risultati della simulazione.

I dati del materiale inseriti in Abaqus sono quindi associati direttamente al *bond* e i valori sono calcolati con le formule del capitolo 1.

Per stabilire quando un *bond* sia rotto o integro è stato utilizzato il criterio del danno di Johnson Cook con la specifica del *type energy*. Il criterio di Johnson Cook è un criterio già implementato all'interno di Abaqus. Essendo una piastra sottoposta solo a trazione e considerando il materiale perfettamente elastico, il programma elimina il *bond* selezionato non appena questo

raggiunge un determinato livello di energia plastica che nel nostro caso è stato posto uguale a zero. Così facendo quando il *bond* supera il valore di snervamento viene creata energia plastica, e perciò il criterio appena inserito lo elimina. Viceversa quando esso rimane in campo elastico non subisce danni e si deforma con la legge lineare elastica della meccanica classica. Inserendo quindi il modulo elastico e il valore di snervamento calcolati con la teoria della peridinamica è possibile eseguire la simulazione in Abaqus senza dover fare approssimazioni o supposizioni.

Nell'articolo⁵⁶ vengono anche specificati i valori dei tempi e degli intervalli di integrazione:

- Durata simulazione: $5,00 \cdot 10^{-5}$ s
- Step Temporale simulazione: $2,50 \cdot 10^{-8}$ s
- Valore del temporale a cui si visualizzano i risultati: $4,60 \cdot 10^{-5}$ s

Per creare la struttura dei nodi e dei *bond* non è possibile disegnarli fisicamente nel file “.cae” e quindi il programma Matlab, descritto in appendice A, crea i file con estensione “.inp” che verranno successivamente importati in Abaqus. Tali file contengono la lista sia dei nodi che dei *bond* con le loro rispettive proprietà. Per creare la cricca iniziale che è una zona priva di *bond* si eliminano quelli che la attraversano.

Al termine delle varie elaborazioni si riscontra come l'andamento del danno delle simulazioni sia pressoché uguale all'andamento del danno dell'articolo. Tutti i calcoli sono stati effettuati utilizzando le formule del caso *plane stress* descritto nel capitolo 1.

Modalità di applicazione del carico

Ipotizzando di applicare tutto il carico sui nodi del contorno, i *bond* ad essi collegati si sarebbero rotti, creando di fatto una striscia di nodi strappati dalla struttura senza che questa influisse sul resto piastra. Gli autori dell'articolo⁵⁷ per risolvere questo problema hanno creato dei *bond* indistruttibili sul contorno in modo da trasmettere direttamente il carico all'interno della struttura. Nelle simulazioni che sono state effettuate in Abaqus, invece che creare questo gruppo di *bond*, si è diviso il carico su più file di nodi così da avere sui *bond* di contorno una tensione minore a quella di rottura.

⁵⁶ Vedi nota 52 p.35

⁵⁷ Vedi nota precedente

In quest'ultimo caso i *bond* di contorno sono identici a quelli della struttura e di conseguenza avere applicato il carico su più file di essi non ha influito sulla simulazione perché queste sono molto ravvicinate come si può vedere dalla figura 2-5. Il numero di file di nodi è stato sempre posto uguale ad "m" in modo da non avere questo dato variabile fra le varie simulazioni. Le zone di contorno modellate con la peridinamica sono più deboli rispetto a quelle interne alla struttura in quanto i nodi in queste posizioni non hanno *bond* al di là della zona di bordo e quindi sono meno legati al resto della struttura. Prendendo in esame i nodi situati poche file all'interno della struttura questo problema non è più rilevante in quanto essi hanno attorno altri nodi a cui sono collegati con *bond*.

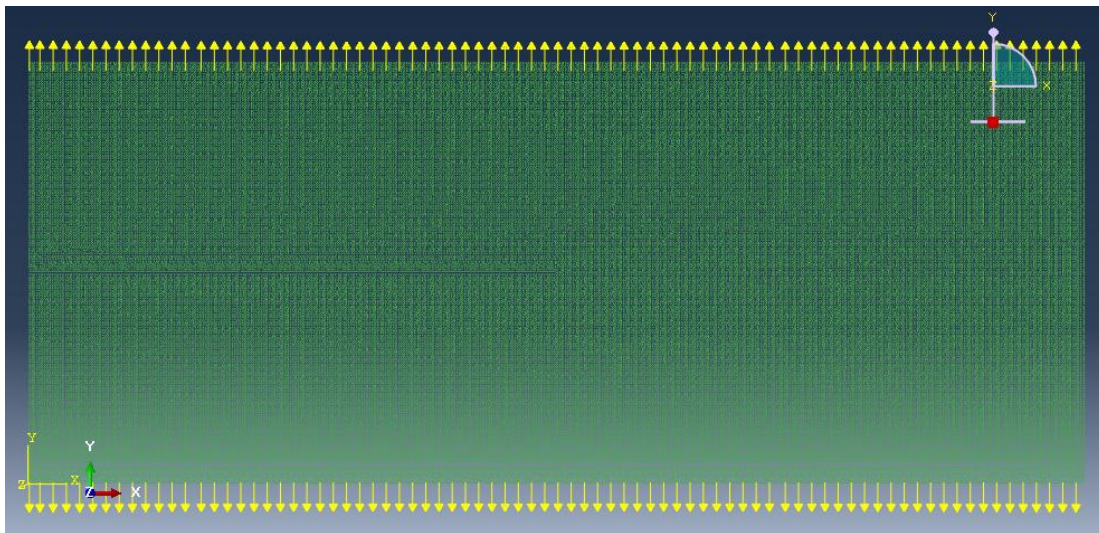


Figura 2-4 - Piastra con il carico applicato ai bordi

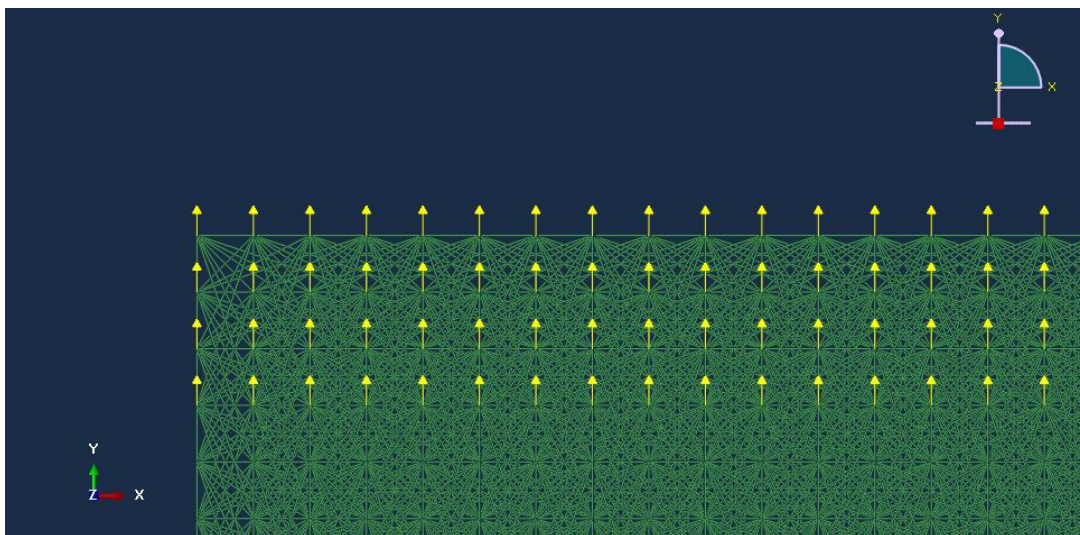


Figura 2-5 - Particolare ingrandito dell'angolo in alto a sinistra della figura 13

2.1. Tipologia di convergenza n. 1: m convergenza

In tutte le simulazioni il passo lungo la direzione dell'asse delle ascisse e lungo la direzione dell'asse delle ordinate è stato posto uguale. L'articolo⁵⁸ prende in esame la piastra in *Duran 50 glass* modellata tenendo costante il valore dell'orizzonte e variando di volta in volta il valore di m . Viene preso in considerazione come ultimo caso per questo δ il caso della piastra con $\delta=0,002m$, $m=8$ e $\Delta x=0,00025m$.

Per motivi di risorse di calcolo visto l'elevato numero di elementi che ne sarebbe risultato, si è provveduto ad analizzare un esempio intermedio con $\delta=0,002m$, $m=6$ e passo= $0,000333$. Nonostante il passo con $m=6$ non fosse un numero finito e non fosse multiplo della lunghezza della base della piastra, i risultati ottenuti sono stati quasi identici al caso con $m=8$. La cosa evidente è che fra i tre casi la differenza è minima con però un livello di complessità e necessità di risorse di calcolo via via crescente man mano che il valore di m cresce, e di conseguenza anche il numero di *bond*. Avendo analizzato i tre i casi è quindi ragionevole ritenere il caso con $m=4$ idoneo a rappresentare un andamento chiaro del percorso della cricca.

Nella tabella 2-1 sono stati messi a confronto i risultati.

Sono state effettuate anche simulazioni con $\delta=0,003 m$. A parità di passo all'aumentare dell'orizzonte aumenta il numero dei nodi che interagiscono con il nodo di riferimento.

⁵⁸ Vedi nota 52 p.35

Metodo m convergenza: $\delta=0,002m$

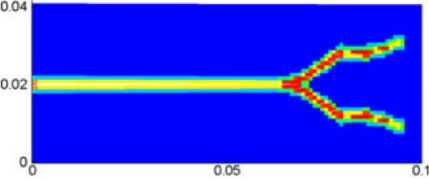
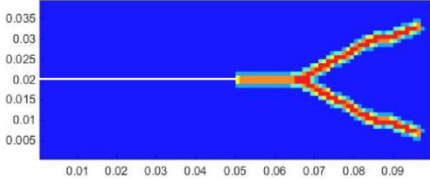
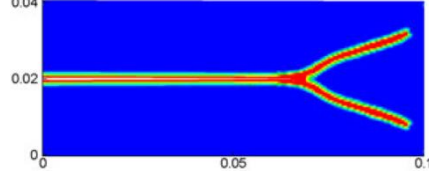
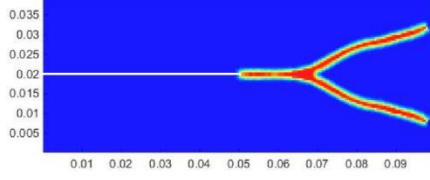
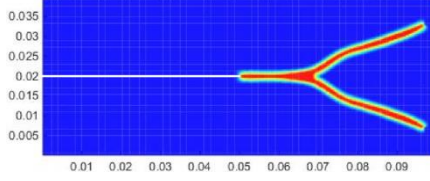
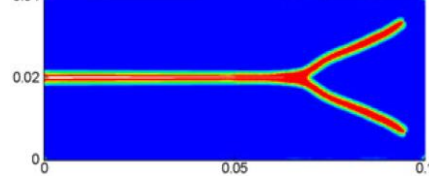
Paragrafo	δ (m)	m	Δx (m)	Immagine di riferimento ⁵⁹	Immagine risultante ⁶⁰
2.1.1. Piastra in Duran	0,002	2	0,001		
2.1.2. Piastra in Duran	0,002	4	0,0005		
2.1.3. Piastra in Duran	0,002	6	0,00033	Caso non presente nell'articolo[2]	
Piastra in Duran	0,002	8	0,00025		Caso non analizzato in Abaqus

Tabella 2-1 - Tabella m convergenza con $\delta=0,002m$

Metodo m convergenza: $\delta=0,003$

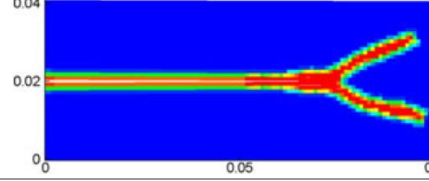
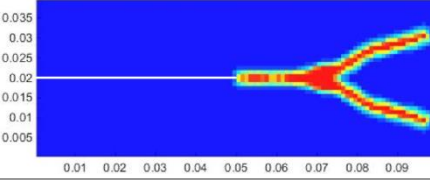
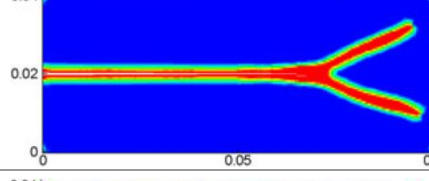
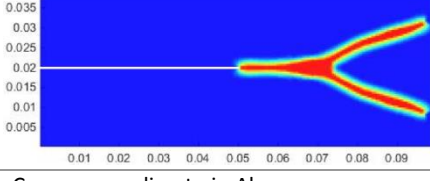
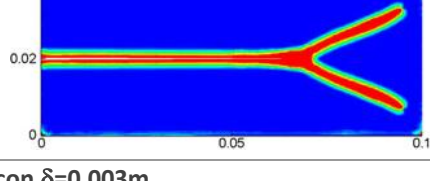
Paragrafo	δ (m)	m	Δx	Immagine di riferimento ⁶¹	Immagine risultante ⁶²
2.1.4. Piastra in Duran	0,003	3	0,001		
2.1.5. Piastra in Duran	0,003	6	0,0005		
Piastra in Duran	0,003	12	0,00025		Caso non analizzato in Abaqus

Tabella 2-2- Tabella m convergenza con $\delta=0,003m$

⁵⁹ Immagine tratta da: Ha Y. D., Bobaru F. *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

⁶⁰ Si intende l'immagine elaborata con il programma Matlab dagli spostamenti e dagli stati dei bond presenti nei file report generati da Abaqus

⁶¹ Vedi nota 59

⁶² Vedi nota 60

Confrontando il caso della δ convergenza⁶³ con $m=4$ e $m=6$ anche dal punto di vista grafico non vi sono sostanziali differenze. Nel primo caso però il numero nodi ed elementi sono rispettivamente 16000 e 368876 mentre nel secondo caso sono 36000 e 1934954 e si evidenzia un notevole aumento dei tempi di calcolo. Da ciò si evince che il caso con $m=8$, vista anche la figura di riferimento dell'articolo, non avrebbe apportato alcuna sostanziale informazione in più sull'andamento della cricca rispetto ai due casi esaminati. Fra le figure di riferimento⁶⁴ e quelle risultanti si nota che le prime hanno una leggera asimmetria (soprattutto il caso $m=4$ e $m=8$) rispetto alla direzione della cricca; un'imprecisione che nelle simulazioni in Abaqus non è presente.

Una situazione analoga avviene anche nella convergenza con $\delta=0.003$ m; tra il caso con $m=6$ e il caso con $m=12$ non vi sono sostanziali differenze. Il caso con $m=12$ non è stato simulato sia per il motivo appena scritto sia perché è gravoso dal punto di vista computazione per i calcolatori a disposizione.

Per non appesantire il programma in Matlab nell'elaborazione della figura la cricca iniziale, presente come dato del problema, non è stata modellizzata come zona di danno come invece è stato fatto nell'articolo. Questa differenza influisce solo nei colori della figura elaborata con Matlab, la cricca iniziale viene quindi visualizzata come linea bianca. Sotto sono riportate le piastre, descritte singolarmente in ogni paragrafo, con il confronto fra l'immagine di riferimento, l'immagine elaborata da Matlab e l'immagine risultante che dà il programma Abaqus.

2.1.1. Piastra in Duran 50 Glass con $\delta=0.002$ m, $m=2$ e $\Delta x=0.001$ m

Nonostante questa sia la simulazione meno accurata fra quelle effettuate si nota sia nell'articolo⁶⁵ sia in Abaqus che la cricca non è rettilinea ma ha un lieve cambio di direzione a circa 0.075 m dall'estremità sinistra. Questo cambio di direzione scompare praticamente nelle successive figure con "m" maggiori nelle figure di riferimento⁶⁶ ma rimane presente nelle altre simulazioni in Abaqus diventando invece in quest'ultimo più evidente.

⁶³ $\delta=0.002$ m, vedi tabella 2-1

⁶⁴ Ha Y. D., Bobaru F., *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

⁶⁵ Vedi nota precedente

⁶⁶ Vedi nota precedente

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 2

Passo in direzione x: 0,001 m

Passo in direzione y: 0,001 m

Valore dell'orizzonte (δ): 0,00202 m

L'orizzonte è stato aumentato di mezzo passo in modo da evitare errori di approssimazione e poter individuare i nodi non compresi nell'orizzonte nominale, che possiedono quindi volumi all'interno della sfera di influenza del nodo di riferimento.

Numero totale di nodi: 4000

Numero totale di elementi: 23052

$$c = 2,26 \cdot 10^{19}$$

$$s_0 = 1,473 \cdot 10^{-03} \text{ m}$$

I vari *bond* hanno valori di aree differenti in modo da poter simulare l'influenza di un volume di materiale parzialmente compreso nell'orizzonte. Se un nodo è fuori dall'orizzonte ma una porzione di volume ad esso associata è all'interno della sfera di influenza del nodo di riferimento, è necessario collegarlo con un *bond* di rigidezza differente a quella nominale. La rigidezza nominale viene calcolata moltiplicando l'area dei *bond* interni alla sfera di influenza per un valore correttivo chiamato beta, trattato nel § 1.9, in base al valore del volume contenuto.

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,52 e 1,00

Valore della sezione del bond non modificata da beta: $1,00 \cdot 10^{-4} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta: $5,2 \cdot 10^{-5} \text{ m}^2$ e $1,00 \cdot 10^{-4} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $2,235 \cdot 10^{-3} \text{ kg}$

Valore di Yield stress: $3,328 \cdot 10^8$ Pa

Valore del modulo di elasticità del bond: $2,26 \cdot 10^{11}$ Pa

Valore della densità del *bond*: $1,00 \cdot 10^{-18}$ kg/m³

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,20 \cdot 10^7$ Pa

Numero di righe di nodi a cui applicare la forza equivalente: 2

Valore della forza equivalente da applicare al singolo nodo: $6,00 \cdot 10^3$ N

Risultati:

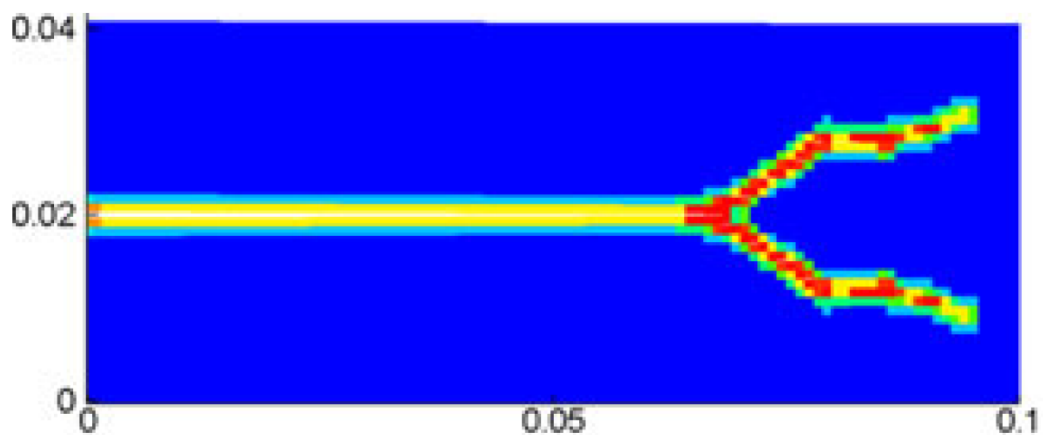


Figura 2-6 - Piastra In Duran 50 Glass con $\delta=0,002$ m $m=2$ $\Delta x=0,001$ m⁶⁷

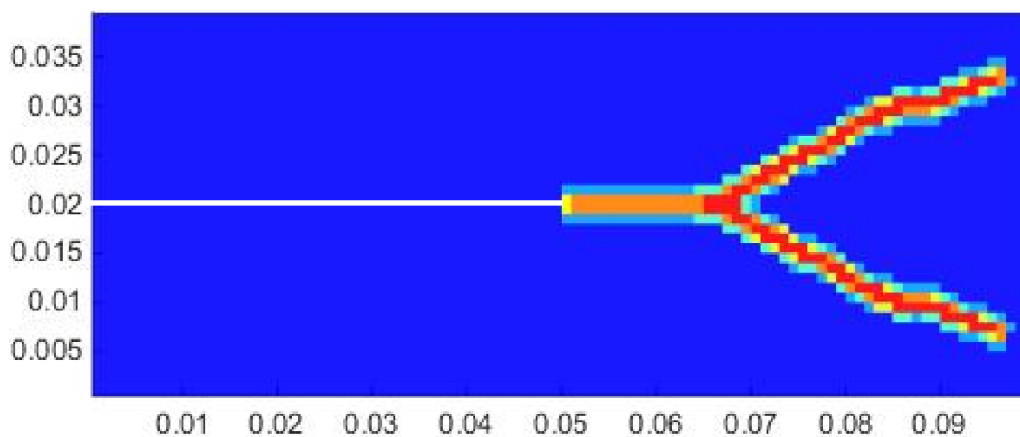


Figura 2-7 - (Immagine Matlab) Piastra In Duran 50 Glass con $\delta=0,002$ m $m=2$ $\Delta x=0,001$ m

⁶⁷ Vedi tabella 2-1

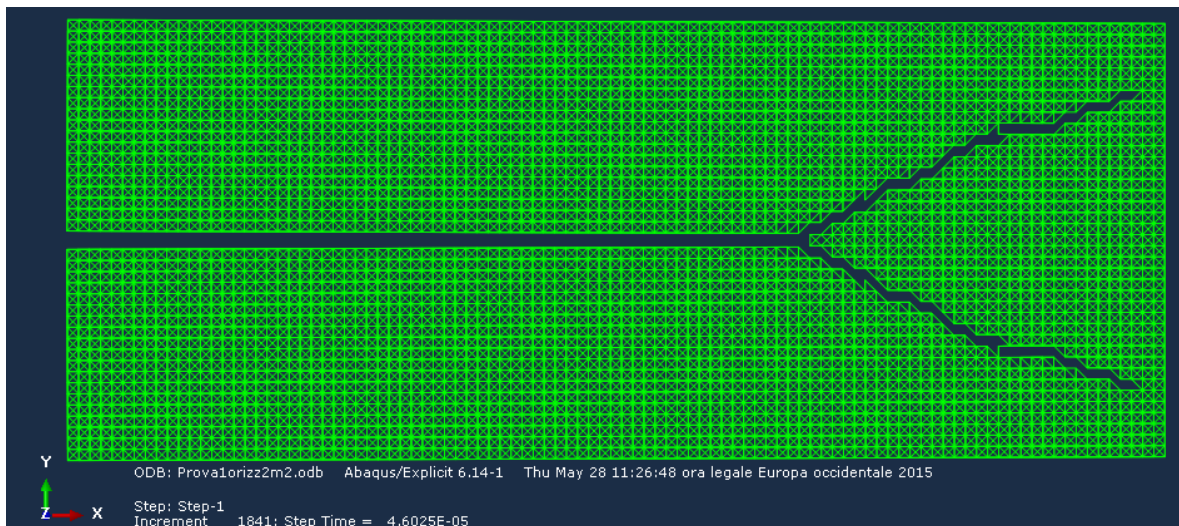


Figura 2-8 - (Immagine Abaqus) Piastra In Duran 50 Glass con $\delta=0,002$ m $m=2$ $\Delta x=0,001$ m

Essendoci molti *bond* non interi in prossimità dei bordi della cricca, la figura di Matlab risulta con una cricca estesa in larghezza non facendo visualizzare bene il cambio di direzione, che invece è più chiaro nella figura 2-8 risultante da Abaqus.

2.1.2. Piastra in Duran 50 Glass con $\delta=0,002$ m, $m=4$ e $\Delta x=0,0005$ m

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 4

Passo in direzione x: 0,0005 m

Passo in direzione y: 0,0005 m

Valore dell'orizzonte: 0,00202 m

Numero totale di nodi: 16000

Numero totale di elementi: 368876

$$c=2,26 \cdot 10^{19}$$

$$s_0= 1,473 \cdot 10^{-3} \text{ m}$$

Dati inseriti e valori calcolati dal programma Matlab relativi al file main per Abagus:

Valori di beta: 0,54 0,93 1,00

Valore di della sezione del *bond* non modificata da beta: $3,969 \cdot 10^{-5} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta: $2,14 \cdot 10^{-5} \text{ m}^2$ $3,71 \cdot 10^{-5} \text{ m}^2$ $3,97 \cdot 10^{-5} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $5,59 \cdot 10^{-4} \text{ kg}$

Valore di Yield stress: $5,24 \cdot 10^7 \text{ Pa}$

Valore del modulo di elasticità del *bond*: $3,556 \cdot 10^{10} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,20 \cdot 10^7 \text{ Pa}$

Numero di righe di nodi a cui applicare la forza equivalente: 4

Valore della forza da applicare al singolo nodo: $1,50 \cdot 10^3 \text{ N}$

Risultati:

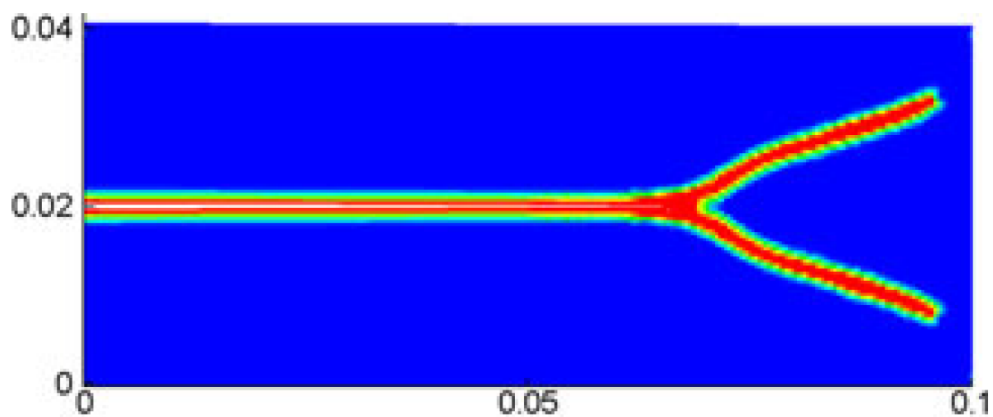


Figura 2-9 - Piastra in Duran 50 Glass con $\delta=0,002 \text{ m}$ $m=4$ e $\Delta x=0,0005 \text{ m}$ ⁶⁸

⁶⁸ Vedi tabella 2-1

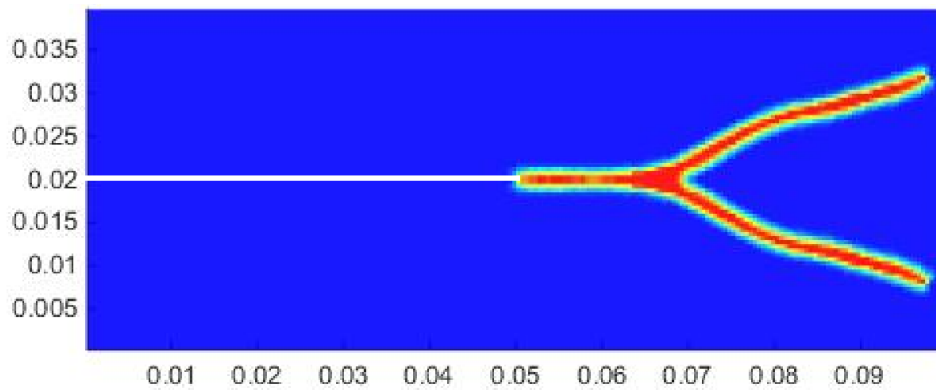


Figura 2-10 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0,002$ m=4 e $\Delta x=0,0005$ m

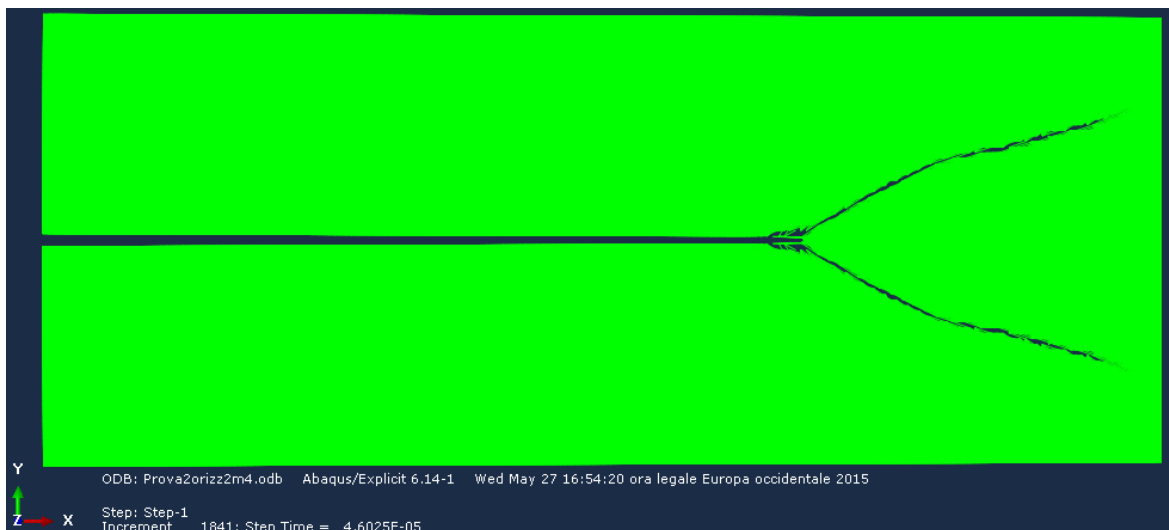


Figura 2-11 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0,002$ m m=4 e $\Delta x=0,0005$ m

2.1.3. Piastra in Duran 50 Glass con $\delta=0,002$ m, m=6 e $\Delta x=0,00033$ m

Questo è il caso intermedio fra il caso m=4 ed m=8 non presente nell'articolo. Guardando le immagini risultanti da Abaqus si nota che man mano che diminuisce il passo la cricca diventa sempre più un fenomeno circoscritto e risulta di conseguenza più sottile.

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 6

Passo in direzione x: 0,000333 m

Passo in direzione y: 0,000333 m

Valore dell'orizzonte: 0,00202 m

Numero totale di nodi: 36000

Numero totale di elementi: 1934954

$c = 2,26 \cdot 10^{19}$

$s_0 = 1,473 \cdot 10^{-3}$ m

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abagus:

Valori di beta: 0,56 0,73 0,903 1,00

Valore di della sezione del *bond* non modificata da beta: $2,31 \cdot 10^{-5}$ m²

Valori delle sezioni in ordine crescente di valore di beta:

$1,294 \cdot 10^{-5}$ m² $1,685 \cdot 10^{-5}$ m² $2,087 \cdot 10^{-5}$ m² $2,311 \cdot 10^{-5}$ m²

Massa applicata ad ogni singolo nodo: $2,483 \cdot 10^{-4}$ kg

Valore di Yield stress: $1,778 \cdot 10^7$ Pa

Valore del modulo di elasticità del *bond*: $1,207 \cdot 10^{10}$ Pa

Valore della densità del *bond*: $1,00 \cdot 10^{-18}$ kg

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,20 \cdot 10^7$ Pa

Numero di righe di nodi a cui applicare la forza equivalente: 6

Valore della forza da applicare al singolo nodo: $6,67 \cdot 10^2$ N

Risultati:

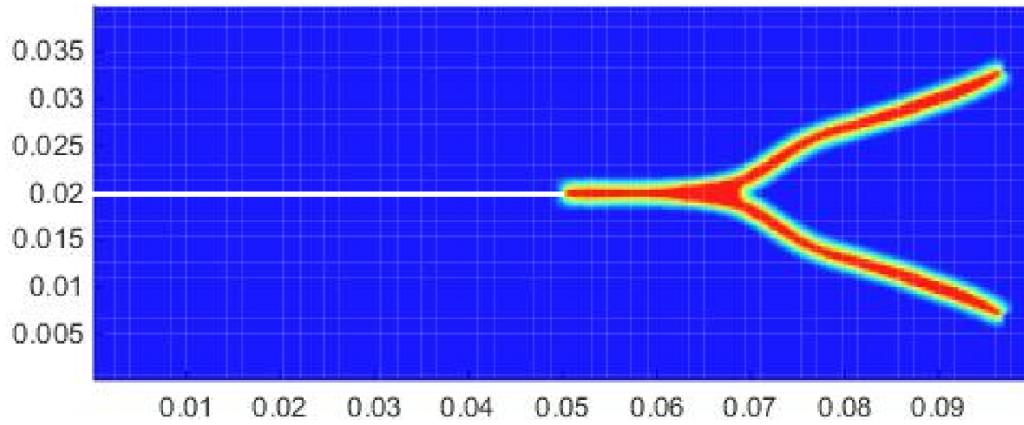


Figura 2-12 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0.002$ m $m=6$ e $\Delta x=0.000333$ m

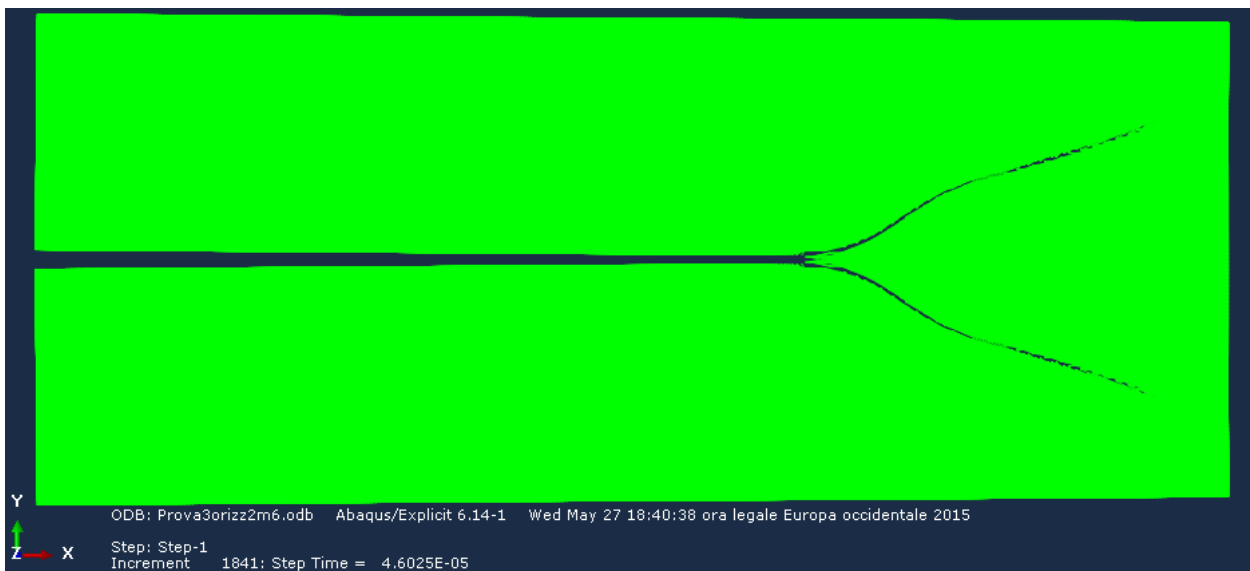


Figura 2-13 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0.002$ m $m=6$ e $\Delta x=0.000333$ m

2.1.4. Piastra in Duran 50 Glass con $\delta=0.003\text{m}$, $m=3$ e $\Delta x=0.001\text{m}$

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m : 3

Passo in direzione x : 0,001 m

Passo in direzione y : 0,001 m

Valore dell'orizzonte: 0,00303 m

Numero totale di nodi: 4000

Numero totale di elementi: 52605

$c= 6,69 \cdot 10^{18}$

$s_0= 1,2026 \cdot 10^{-3}$ m

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di β : 0,53 0,702 1,00

Valore di della sezione del *bond* non modificata da β : $1,00 \cdot 10^{-4} \text{m}^2$

Valori delle sezioni in ordine crescente di valore di β : $5,3 \cdot 10^{-5} \text{m}^2$ $7,016 \cdot 10^{-5} \text{m}^2$ $1,00 \cdot 10^{-4} \text{m}^2$

Massa applicata ad ogni singolo nodo: $2,235 \cdot 10^{-3}$ kg

Valore di Yield stress: $8,05 \cdot 10^7$ Pa

Valore del modulo di elasticità del *bond*: $6,69 \cdot 10^{10}$ Pa

Valore della densità del *bond*: $1,00 \cdot 10^{-18}$ kg

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,20 \cdot 10^7$ Pa

Numero di righe di nodi a cui applicare la forza equivalente: 3

Valore della forza da applicare al singolo nodo: $4,00 \cdot 10^3$ N

Risultati:

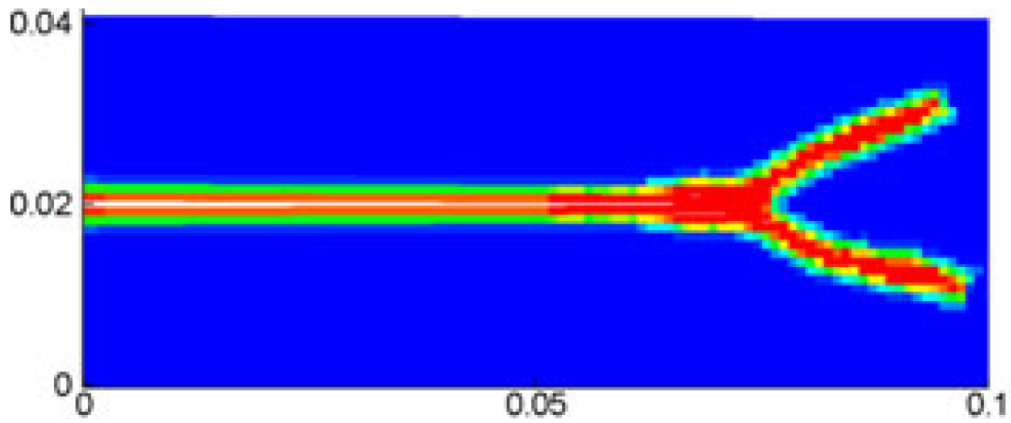


Figura 2-14 - Piastra in Duran 50 Glass con $\delta=0,003$ m $m=3$ e $\Delta x=0,001$ m ⁶⁹

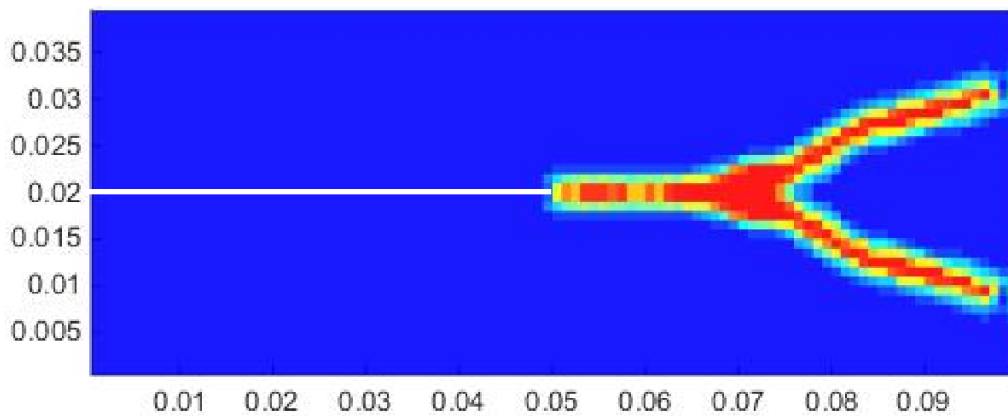


Figura 2-15 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0,003$ m $m=3$ e $\Delta x=0,001$ m

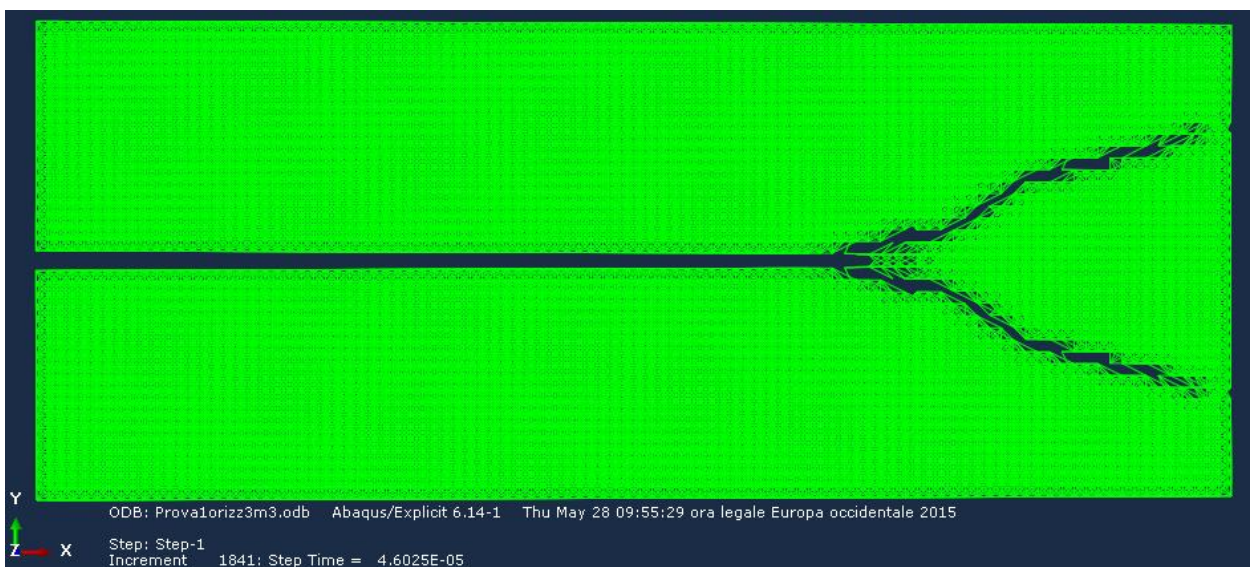


Figura 2-16 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0.003$ m $m=3$ e $\Delta x=0.001$ m

⁶⁹ Vedi tabella 2-2

2.1.5. Piastra in Duran 50 Glass con $\delta=0.003\text{m}$, $m=6$ e $\Delta x=0.0005\text{m}$

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m : 6

Passo in direzione x : 0,0005 m

Passo in direzione y : 0,0005 m

Valore dell'orizzonte: 0,00303 m

Numero totale di nodi: 16000

Numero totale di elementi: 842124

$c=6,69 \cdot 10^{18}$

$s_0=1,2026 \cdot 10^{-3}$ m

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di β : 0,56 0,73 0,903 1,00

Valore di della sezione del *bond* non modificata da β : $3,97 \cdot 10^{-5}$ m²

Valori delle sezioni in ordine crescente di valore di β :

$2,22 \cdot 10^{-5}$ m² ; $2,89 \cdot 10^{-5}$ m² ; $3,58 \cdot 10^{-5}$ m² ; $3,97 \cdot 10^{-5}$ m²

Massa applicata ad ogni singolo nodo: $5,59 \cdot 10^{-4}$ kg

Valore di Yield stress: $1,27 \cdot 10^7$ Pa

Valore del modulo di elasticità del *bond*: $1,054 \cdot 10^{10}$ Pa

Valore della densità del *bond*: $1,00 \cdot 10^{-18}$ kg/m³

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,20 \cdot 10^7$ Pa

Numero di righe di nodi a cui applicare la forza equivalente: 6

Valore della forza da applicare al singolo nodo: $1,00 \cdot 10^3$ N

Risultati:

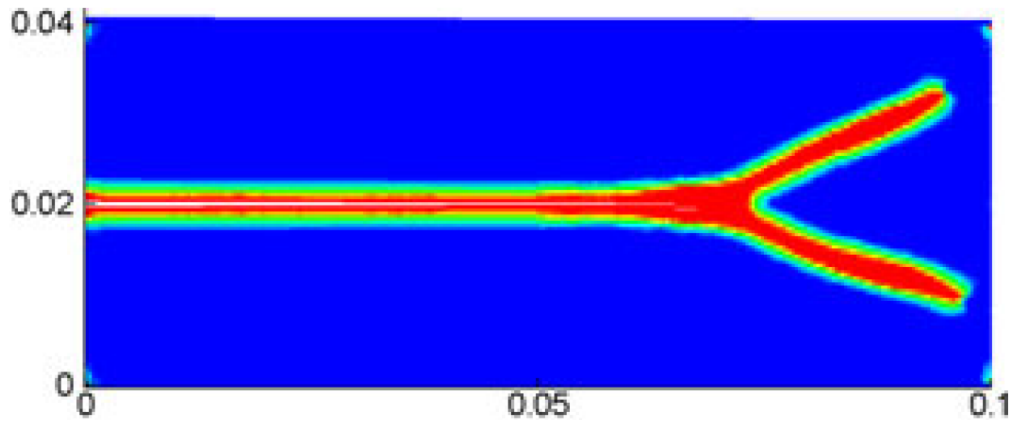


Figura 2-17 - Piastra in Duran 50 Glass con $\delta=0.003$ m $m=6$ e $\Delta x=0.0005$ m ⁷⁰

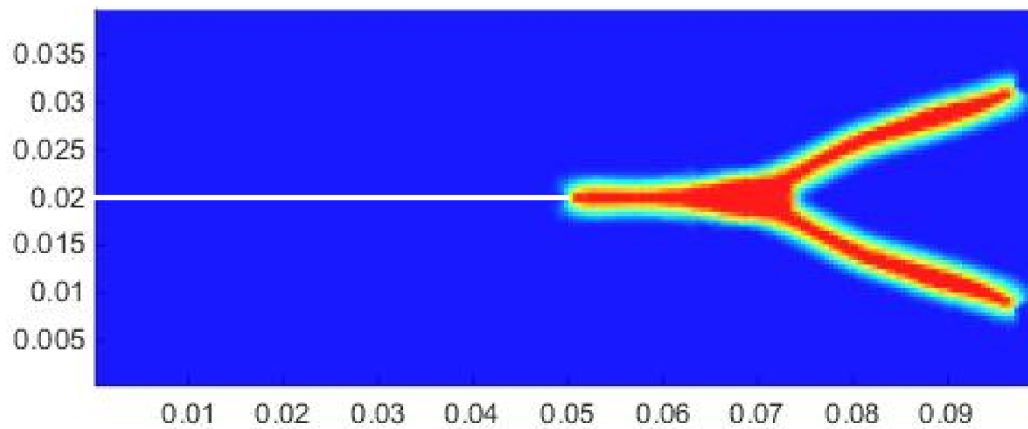


Figura 2-18 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0.003$ m $m=6$ e $\Delta x=0.0005$ m

⁷⁰ Vedi tabella 2-2

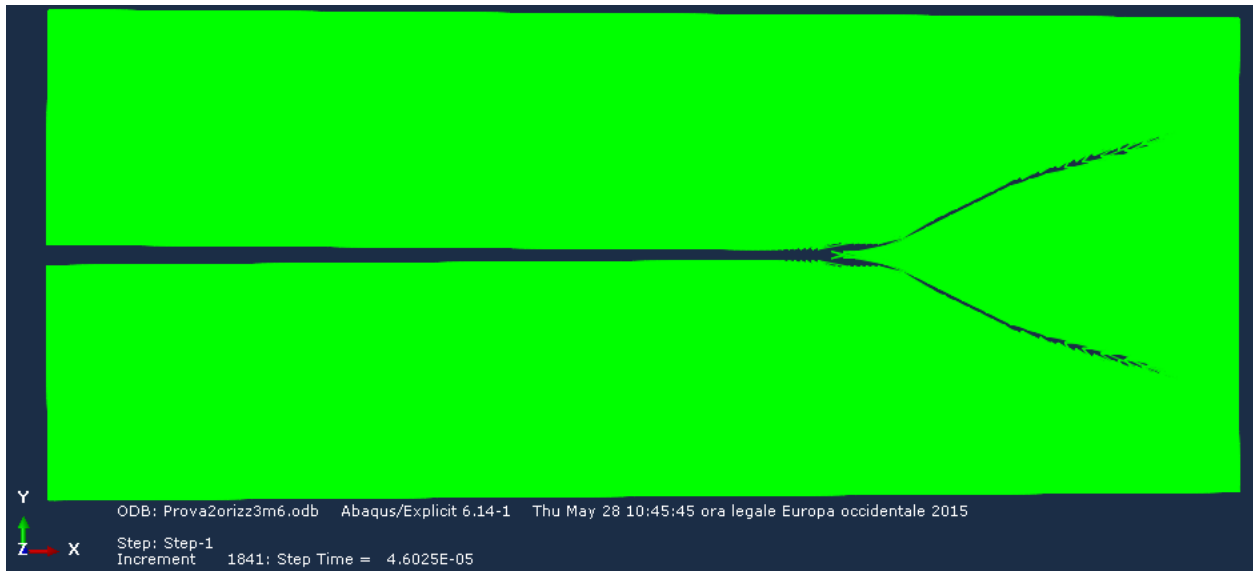


Figura 2-19 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0.003$ m $m=6$ e $\Delta x=0.0005$ m

2.2. Tipologia di convergenza n. 2: δ convergenza

Utilizzando sempre la stessa geometria e lo stesso materiale delle simulazioni con la m convergenza si è analizzato l'andamento della cricca con la δ convergenza, lasciando quindi inalterato il valore di m . Facendo così il numero di nodi presenti all'interno dell'orizzonte non varia da simulazione a simulazione, ma l'orizzonte man mano si rimpicciolisce. Così facendo la regione della sfera di influenza diminuisce. Come per i casi precedenti dell'altra modalità di convergenza, anche qui i risultati ottenuti con Abaqus sono coerenti con quelli dell'articolo.

Nei due casi con $\Delta x=0,0005$ m e $\Delta x=0,00025$ m la leggera asimmetria presente nei risultati dell'articolo⁷¹ non è presente nell'immagine risultante (come verrà evidenziato in § 2.2.1, § 2.2.2 e § 2.2.3). Quello che si nota man mano che diminuisce il passo e l'orizzonte è che la parte della piastra con danno maggiore è sempre più ristretta. Questo dimostra che man mano che l'orizzonte diminuisce, l'influenza di un nodo sui nodi circostanti diminuisce, e la rottura di un numero ristretto di *bond* è in grado di isolare la parte di piastra vicina alla frattura da ulteriori allargamenti della cricca.

Dai risultati sotto esposti e la durata della simulazione con $\Delta x=0,00025$ m non si è provveduto a riprodurre anche quella con $\Delta x=0,000125$ m in quanto le informazioni che questa avrebbe aggiunto non sarebbero state rilevanti vista l'immagine di output della pubblicazione. Il

⁷¹ Ha Y. D., Bobaru F. *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

materiale fin qui impiegato avendo una notevole energia di frattura non ha attivato quei meccanismi di ramificazione delle cricche tipici dei vetri ma l'andamento della cricca è stato regolare e senza biforcazioni oltre a quella principale iniziale. Si vedono delle variazioni modeste di percorso che non hanno comunque innescato tali meccanismi di ramificazione. Le considerazioni fatte per la convergenza sono le medesime per quanto riguarda il valore dell'orizzonte e il valore delle sezioni dei *bond*. Nella tabella sottostante sono state messe a confronto tutte le immagini delle simulazioni, mentre in § 2.2.1., § 2.2.2., § 2.2.3. sono state descritte le singole simulazioni con i rispettivi dati e valori ottenuti.

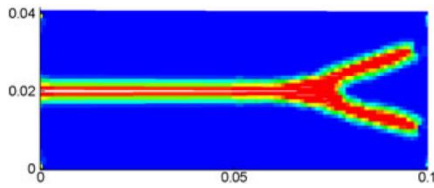
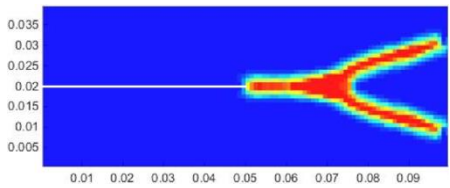
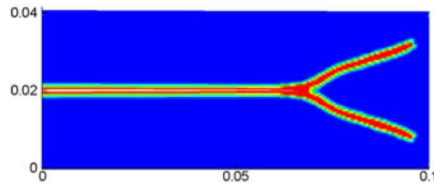
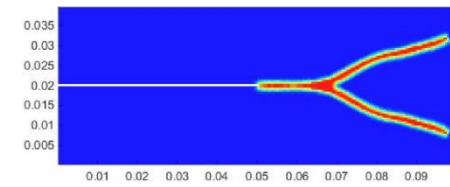
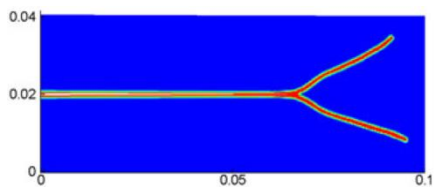
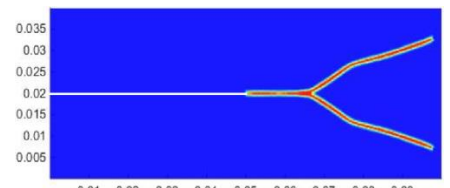
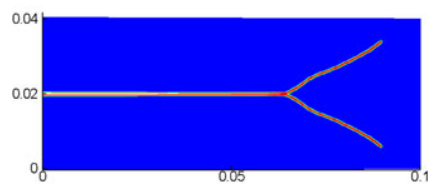
Paragrafo	δ (m)	m	Δx	Immagine di riferimento ⁷²	Immagine risultante ⁷³
2.1.1. Piastra in Duran	0,004	4	0,0001		
2.1.2. Piastra in Duran	0,002	4	0,0005		
2.1.3. Piastra in Duran	0,001	4	0,00025		
Piastra in Duran	0,0005	4	0,000125		Caso non analizzato in Abaqus

Tabella 2-3 - Tabella δ convergenza con $\delta=0,002m$

⁷² Immagine tratta da: Ha Y. D., Bobaru F.. Studies of dynamic crack propagation and crack branching with peridynamics, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

⁷³ Si intende l'immagine elaborata con il programma Matlab dagli spostamenti e dagli stati dei bond presenti nei file report generati da Abaqus

2.2.1. Piastra in Duran 50 Glass con $\delta=0,004\text{m}$, $m=4$ e $\Delta x=0,001\text{m}$

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 4

Passo in direzione x: 0,001 m

Passo in direzione y: 0,001 m

Valore dell'orizzonte: 0,00404 m

Numero totale di nodi: 4000

Numero totale di elementi: 88476

$c= 2,82 \cdot 10^{18}$

$s_0= 1,041 \cdot 10^{-3} \text{ m}$

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,54 ; 0,93 ; 1,00

Valore della sezione del *bond* non modificata da beta: $1,00 \cdot 10^{-4} \text{ m}^2$

Valori delle sezioni in ordine di valore di beta: $5.4 \cdot 10^{-5} \text{ m}^2$; $9.34 \cdot 10^{-5} \text{ m}^2$; $1.00 \cdot 10^{-4} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $2,235 \cdot 10^{-3} \text{ kg}$

Valore di Yield stress: $2,94 \cdot 10^7 \text{ Pa}$

Valore del modulo di elasticità del *bond*: $2,82 \cdot 10^{10} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $1,200000 \cdot 10^7 \text{ Pa}$

Numero di righe di nodi a cui applicare la forza equivalente: 4

Valore della forza da applicare al singolo nodo: $3,00 \cdot 10^3$ N

Risultati:

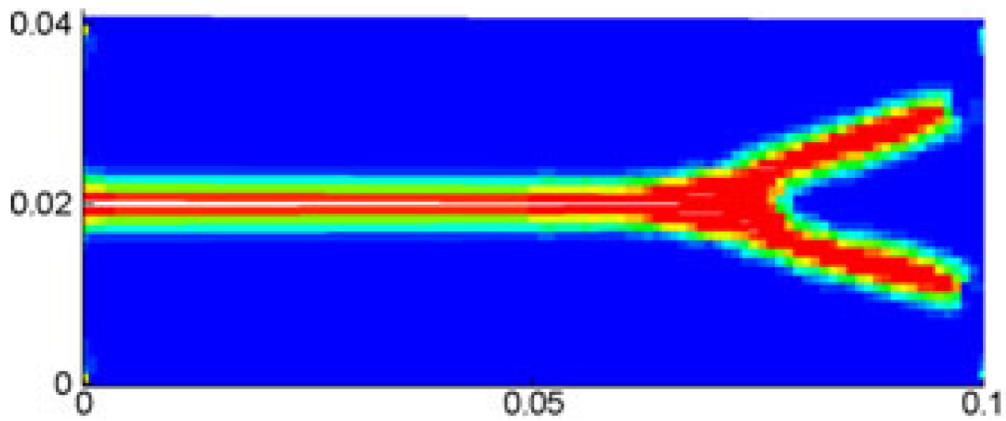


Figura 2-20 - Piastra in Duran 50 Glass con $\delta=0,004$ m $m=4$ e $\Delta x=0,001$ m⁷⁴

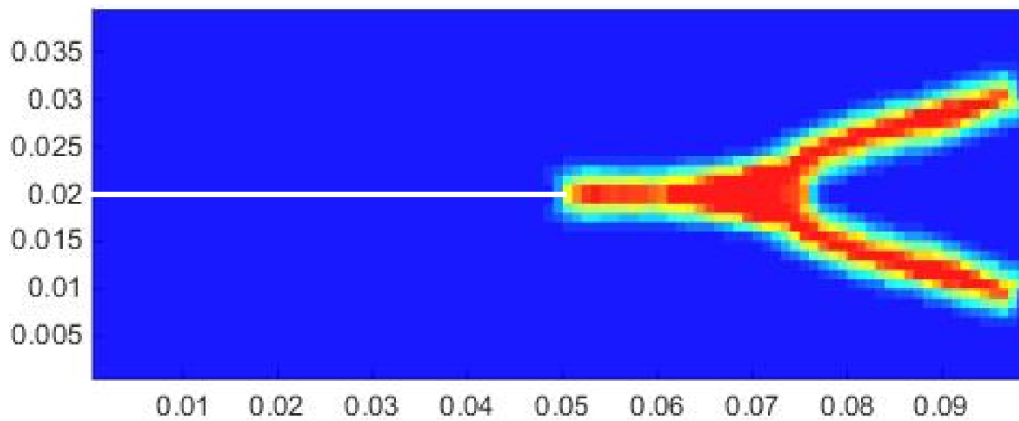


Figura 2-21 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0,004$ m $m=4$ e $\Delta x=0,001$ m

⁷⁴ Vedi tabella 2-3

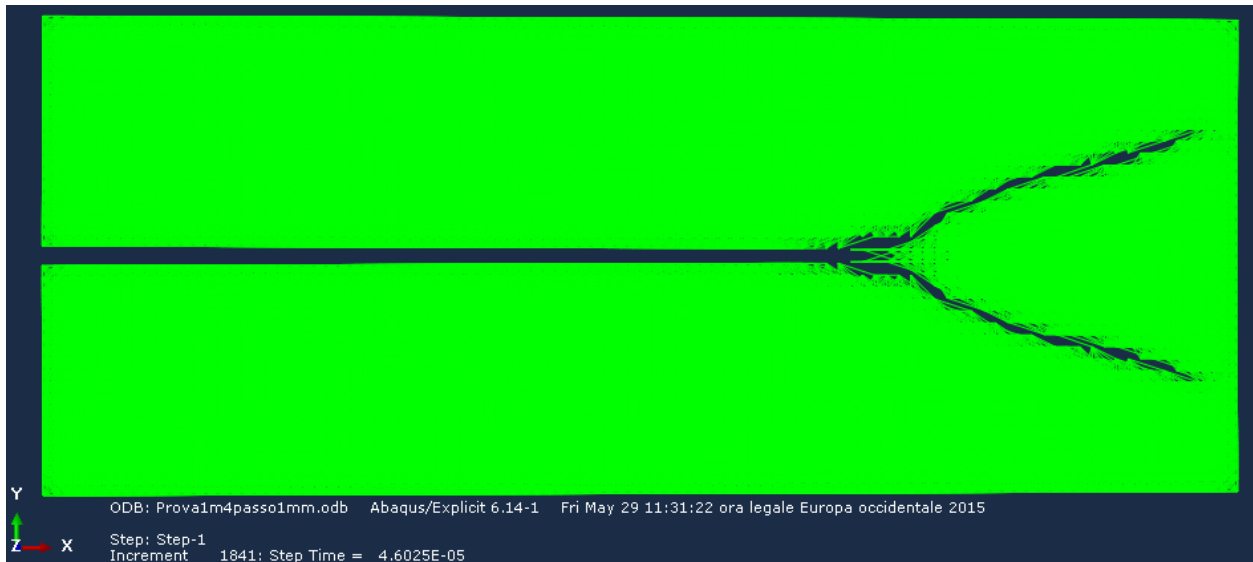


Figura 2-22 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0,004$ m $m=4$ e $\Delta x=0,001$ m

2.2.2. Piastra in Duran 50 Glass con $\delta=0,002$ m, $m=4$ e $\Delta x=0,0005$ m

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m : 4

Passo in direzione x : 0,0005 m

Passo in direzione y : 0,0005 m

Valore dell'orizzonte: 0,00202 m

Numero totale di nodi: 16000

Numero totale di elementi: 368876

$c=2,26 \cdot 10^{19}$

$s_0= 1,47 \cdot 10^{-3}$ m

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,54 0,93 1,00

Valore di della sezione del *bond* non modificata da beta: $3,97 \cdot 10^{-5} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta:

$2,14 \cdot 10^{-5} \text{ m}^2$; $3,71 \cdot 10^{-5} \text{ m}^2$; $3,97 \cdot 10^{-5} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $5,59 \cdot 10^{-4} \text{ kg}$

Valore di Yield stress: $5,24 \cdot 10^7 \text{ Pa}$

Valore del modulo di elasticità del *bond*: $3,56 \cdot 10^{10} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale (Pa): $1,20 \cdot 10^7 \text{ Pa}$

Numero di righe di nodi a cui applicare la forza equivalente: 4

Valore della forza da applicare al singolo nodo: $1,50 \cdot 10^3 \text{ N}$

Risultati:

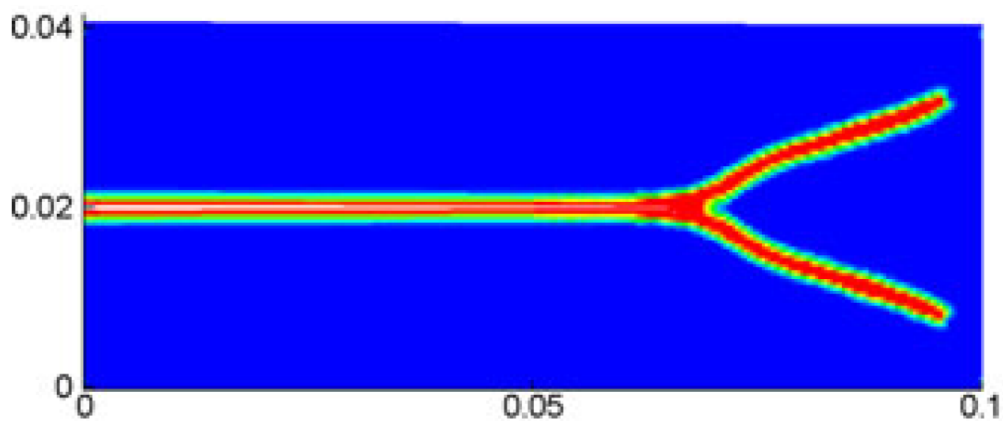


Figura 2-23 - Piastra in Duran 50 Glass con $\delta=0.002\text{m}$ $m=4$ e $\Delta x=0.0005\text{m}$ ⁷⁵

⁷⁵ Vedi tabella 2-3

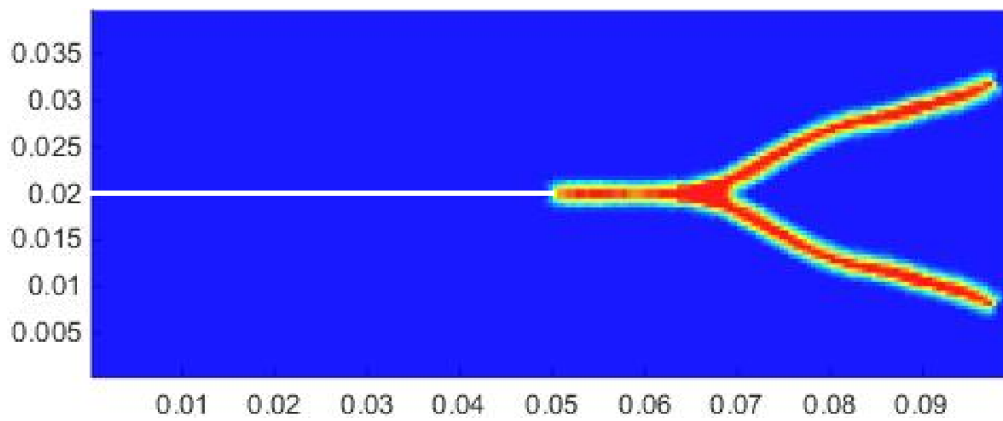


Figura 2-24 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0.002$ m $m=4$ e $\Delta x=0.0005$ m

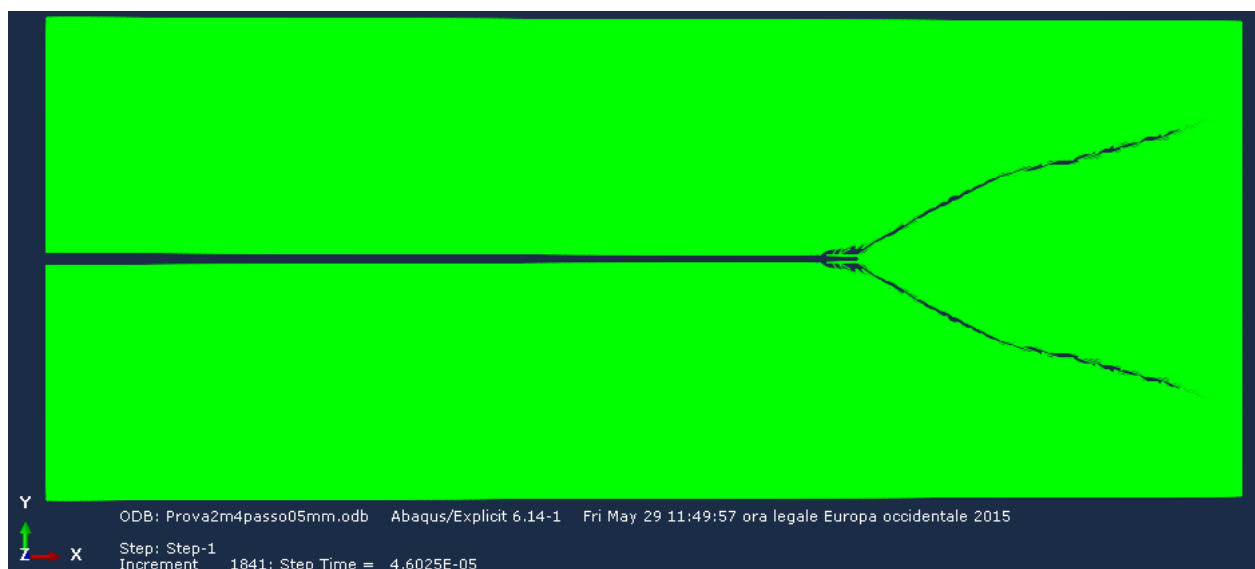


Figura 2-25 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0.002$ m $m=4$ e $\Delta x=0.0005$ m

2.2.3. Piastra in Duran 50 Glass con $\delta=0.001$ m, $m=4$ e $\Delta x=0.00025$ m

Dati:

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m : 4

Passo in direzione x : 0,00025 m

Passo in direzione y : 0,00025 m

Valore dell'orizzonte: 0,00101 m

Numero totale di nodi: 64000

Numero totale di elementi: 1505676

$$c=1,81 \cdot 10^{20}$$

$$s_0=2,083 \cdot 10^{-3} \text{ m}$$

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,54 ; 0,93 ; 1,00

Valore di della sezione del *bond* non modificata da beta: $1,57 \cdot 10^{-5} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta:

$$8,504 \cdot 10^{-6} \text{ m}^2 ; 1,472 \cdot 10^{-5} \text{ m}^2 ; 1,575 \cdot 10^{-5} \text{ m}^2$$

Massa applicata ad ogni singolo nodo: $1,397 \cdot 10^{-4} \text{ kg}$

Valore di Yield stress: $9,33 \cdot 10^7 \text{ Pa}$

Valore del modulo di elasticità del *bond*: $4,48 \cdot 10^{10} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale (Pa): $1,20 \cdot 10^7 \text{ Pa}$

Numero di righe di nodi a cui applicare la forza equivalente: 4

Valore della forza da applicare al singolo nodo: $7,50 \cdot 10^2 \text{ N}$

Risultati:

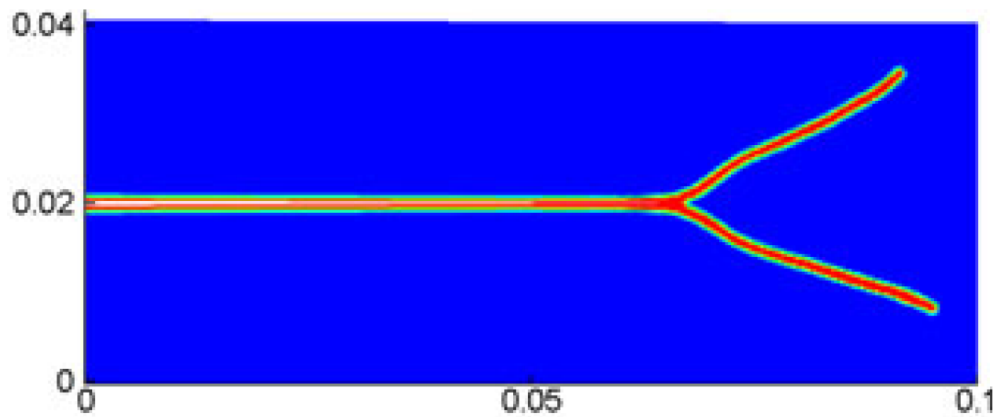


Figura 2-26 - Piastra in Duran 50 Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m ⁷⁶

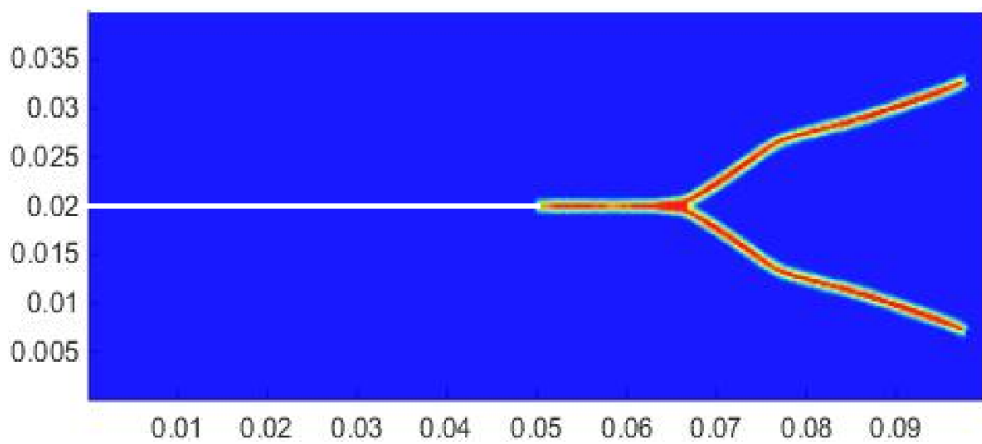


Figura 2-27 - (Immagine Matlab) Piastra in Duran 50 Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m

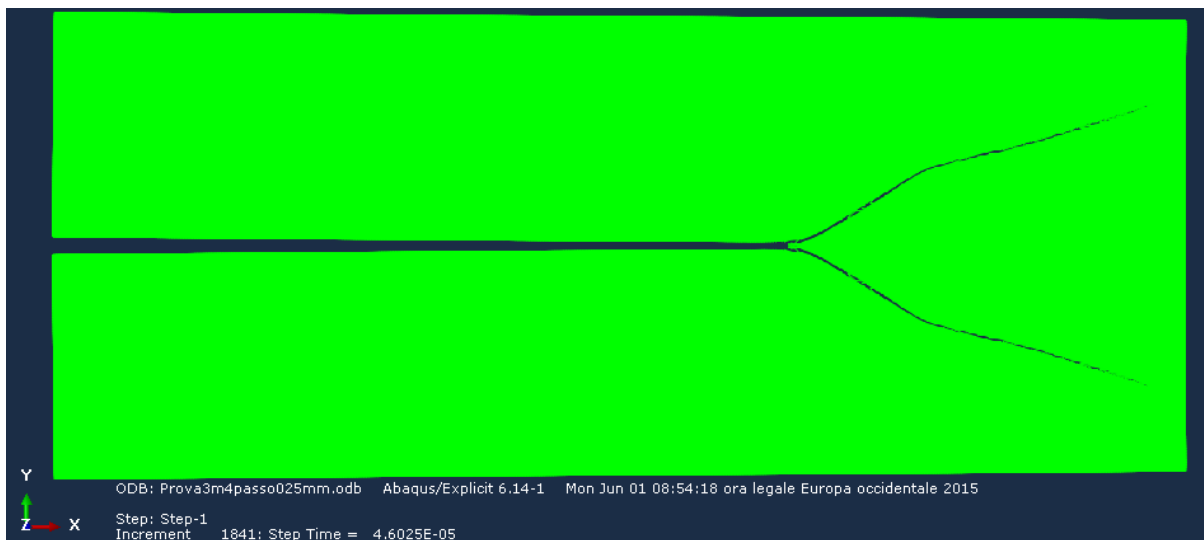


Figura 2-28 - (Immagine Abaqus) Piastra in Duran 50 Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m

⁷⁶ Vedi tabella 2-3

2.3. Confronto fra le distanze delle biforcazioni

Nella pubblicazione viene evidenziato il momento dell'inizio della biforcazione della cricca che non avviene immediatamente all'applicazione del carico. Inizialmente si ha un tratto rettilineo di frattura che successivamente si separa in due parti ad una determinata distanza e ad uno specifico valore temporale. La sua biforcazione risultante ha inizio alla distanza di circa 0,071 m dalla parete verticale sinistra e dopo 25 μ s dall'applicazione della sollecitazione. Per avere la completa corrispondenza con le immagini delle tabelle⁷⁷ sono stati utilizzati gli stessi valori di *time step* e durata delle simulazioni di riferimento. Ciò comporta che non è possibile avere l'istante preciso di 25 μ s (a meno che non si ripetano tutte le simulazioni con un altro *time step*) e quindi è stato considerato il valore di 26 μ s. Si può notare che l'approssimazione sopra descritta non ha influito in maniera rilevante e che tutte le simulazioni in Abaqus tendono ad anticipare e mai a ritardare la creazione della biforcazione.

⁷⁷ Vedi sopra tabella 2-1, 2-2, 2-3

Tabella 2-4 - Tabella dati ramificazione. Sono riportate le distanze della ramificazione e l'errore relativo percentuale

Paragrafo	Caratteristiche della piastra	Distanza risultante in Abaqus [m]	Distanza dell'articolo [m]	Errore percentuale [%]	Numero figura
2.1.1	Piastra in Duran 50 Glass $\delta=0,002\text{m}$, $m=2$, $\Delta x=0,001\text{m}$	0,0675	0,071	4,93	Figura 36
2.1.2	Piastra in Duran 50 Glass $\delta=0,002\text{m}$, $m=4$, $\Delta x=0,0005\text{m}$	0,06525	0,071	8,1	Figura 37
2.1.3	Piastra in Duran 50 Glass $\delta=0,002\text{m}$, $m=6$, $\Delta x=0,000333\text{m}$	0,0642	0,071	9,6	Figura38
2.1.4	Piastra in Duran 50 Glass $\delta=0,003\text{m}$, $m=3$, $\Delta x=0,001\text{m}$	0,0675	0,071	4,93	
2.1.5	Piastra in Duran 50 Glass $\delta=0,003\text{m}$, $m=6$, $\Delta x=0,0005\text{m}$	0,06525	0,71	8,1	
2.2.1	Piastra in Duran 50 Glass $\delta=0,004\text{m}$, $m=4$, $\Delta x=0,001\text{m}$	0,0675	0,071	4,93	
2.2.2	Piastra in Duran 50 Glass $\delta=0,002\text{m}$, $m=4$, $\Delta x=0,0005\text{m}$	0,06525	0,71	8,1	
2.2.3	Piastra in Duran 50 Glass $\delta=0,002\text{m}$, $m=4$, $\Delta x=0,00025\text{m}$	0,064875	0,071	8,6	

Sono di seguito riportate le immagini con il dettaglio della piastra in cui è stato rilevato il dato della distanza di biforcazione.

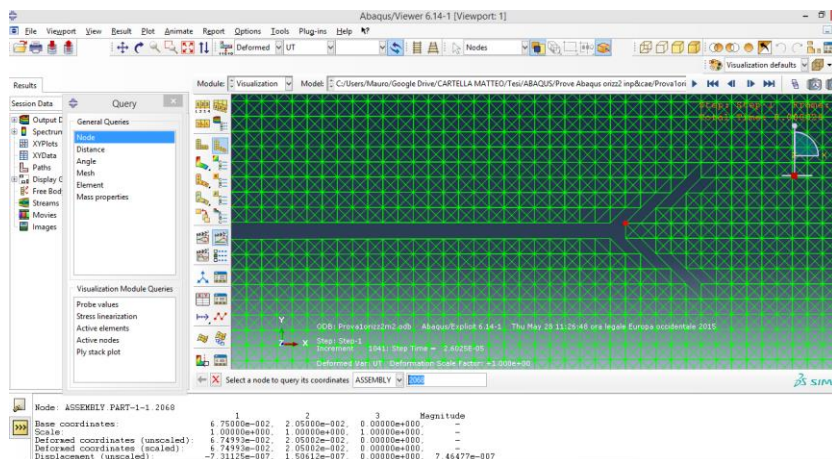


Figura 2-29 - Dettaglio della biforcazione della piastra con $\delta=0,002\text{ m}$, $m=2$, $\Delta x=0,001\text{ m}$ (§ 2.1.1)

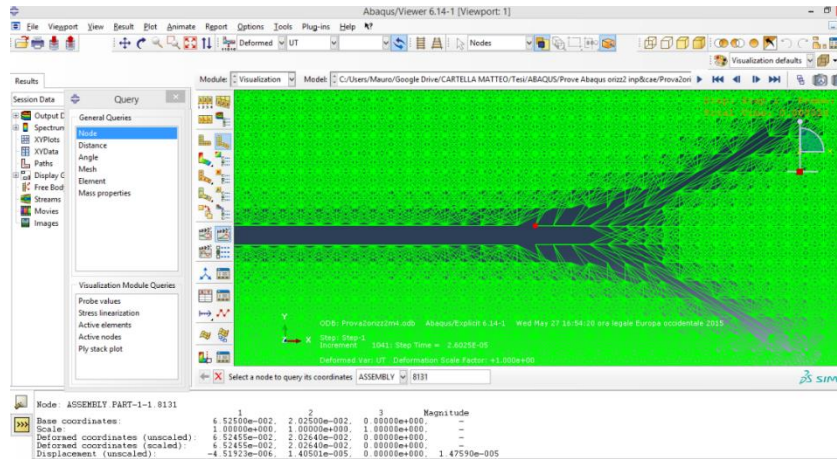


Figura 2-30 - Dettaglio della biforcazione della piastra con $\delta=0,002$ m, $m=4$, $\Delta x=0,0005$ m (§ 2.1.2)

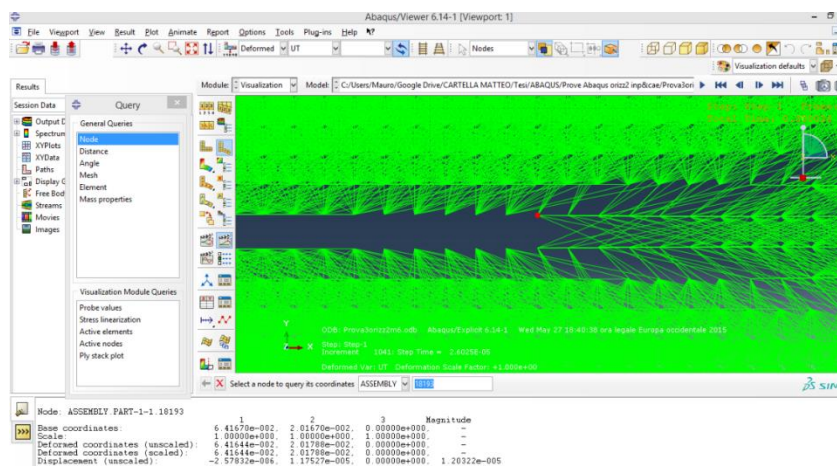


Figura 2-31 - Dettaglio della biforcazione della piastra con $\delta=0,002$ m, $m=6$, $\Delta x=0,00033$ m (§ 2.1.3)

2.4. Piastra con cricca ramificata

Viene preso ora in esame il caso in cui la cricca non ha un andamento lineare ma si divide in vari rami. È stata riprodotta la simulazione della piastra avente come caratteristiche geometriche le medesime di quella in *Duran 50 Glass*, ma costituita da *Soda Lime Glass*.

In questo caso la cricca che si forma non è una semplice biforcazione, ma si creano delle diramazioni. Queste ultime si formano perché il materiale *Soda Lime Glass* ha un modulo elastico di poco superiore rispetto a quello del *Duran 50 Glass*, ed è quindi più duro; di contro ha un'energia di frattura molto inferiore (quasi la metà) e questo rende più probabile la frattura dei *bond* in direzioni diverse da quella principale. Il materiale risulta perciò più fragile e la minore energia necessaria affinché un *bond* si rompa fa sì che vi siano più *bond* rotti in zone

differenti dalla cricca principale e di conseguenza più strade possibili al propagarsi della stessa. Nel § 2.4.1 sono riportati i dati e le figure risultanti. La figura 2-33 è il risultato a $28\mu\text{s}$ mentre la figura 2-35 è il risultato finale dato dalla simulazione in Abaqus.

Dal confronto fra la figura 2-32 e la figura 2-33 si nota che nonostante entrambe non siano perfettamente simmetriche rispetto alla cricca iniziale quest'ultima ha un'asimmetria minore rispetto alla figura 2-32. Questa piccola asimmetria, non riscontrata nelle simulazioni fino a questo punto effettuate, è da attribuire all'elevato numero di elementi e quindi alle semplificazioni effettuate da Abaqus. Il programma elabora internamente la matrice con i gradi di libertà che viene risolta con procedure matematiche che non tengono in considerazione eventuali simmetrie della struttura. Il problema della simmetria è noto ma non influisce in maniera significativa negli altri ambiti, in quanto il numero degli elementi raramente raggiunge cifre superiori al milione come avviene nel caso in esame.

2.4.1. Piastra in Soda Lime Glass con $\delta=0.001\text{m}$, $m=4$, $\Delta x=0.00025\text{m}$

Dati:

Dati del materiale:

Nome del materiale: Soda Lime Glass

E (Modulo di Young): $7,20 \cdot 10^{10}$ Pa

ν (Modulo di Poisson): 0,33

Densità: 2440 kg/m^3

G_0 (Fracture Energy): 135 J/m^2

Dati relativi ai tempi della simulazione in Abaqus:

Durata simulazione: $5,00 \cdot 10^{-5}$ s

Step Temporale simulazione $2,50 \cdot 10^{-8}$ s

Valore del temporale a cui si vogliono vedere i risultati: $2,80 \cdot 10^{-5}$ s

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 4

Passo in direzione x: 0,00025m

Passo in direzione y: 0,00025m

Valore dell'orizzonte: 0,00101 m

Numero totale di nodi: 64000

Numero totale di elementi: 1505676

$c=2,002 \cdot 10^{20}$

$s_0= 1,61 \cdot 10^{-3} \text{ m}$

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,54 ; 0,93 ; 1,00

Valore di della sezione del *bond* non modificata da beta: $1,575 \cdot 10^{-5} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta:

$8,504 \cdot 10^{-6} \text{ m}^2$; $1,472 \cdot 10^{-5} \text{ m}^2$; $1,575 \cdot 10^{-5} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $1,525 \cdot 10^{-4} \text{ kg}$

Valore di Yield stress: $7,9945 \cdot 10^7 \text{ Pa}$

Valore del modulo di elasticità del *bond*: $4,9656 \cdot 10^{10} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti e valori calcolati dal programma Matlab relativi alle condizioni al contorno:

Pressione applicata su ogni bordo orizzontale: $2,40 \cdot 10^7 \text{ Pa}$

Numero di righe di nodi a cui applicare la forza equivalente: 4

Valore della forza da applicare al singolo nodo: $1,50 \cdot 10^3 \text{ N}$

Risultati:

2.4.1.1. Risultati a 28 μ s

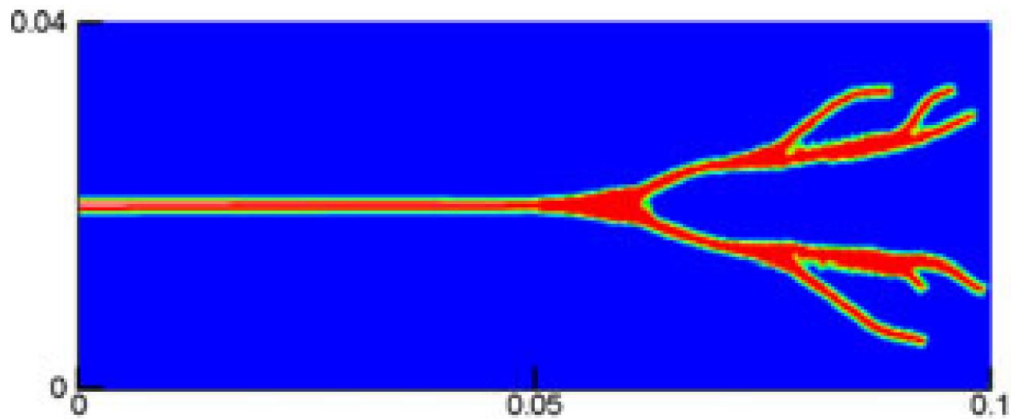


Figura 2-32 – Piastra in Soda Lime Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m⁷⁸

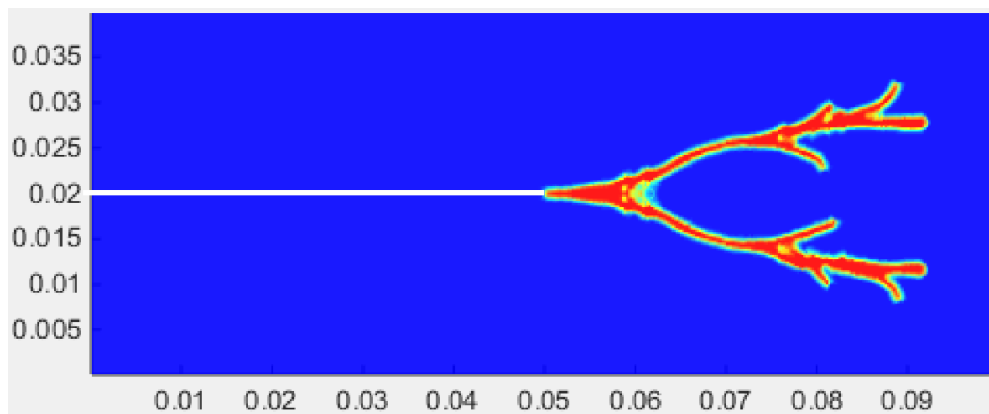


Figura 2-33 - (Immagine Matlab) Piastra in Soda Lime Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m

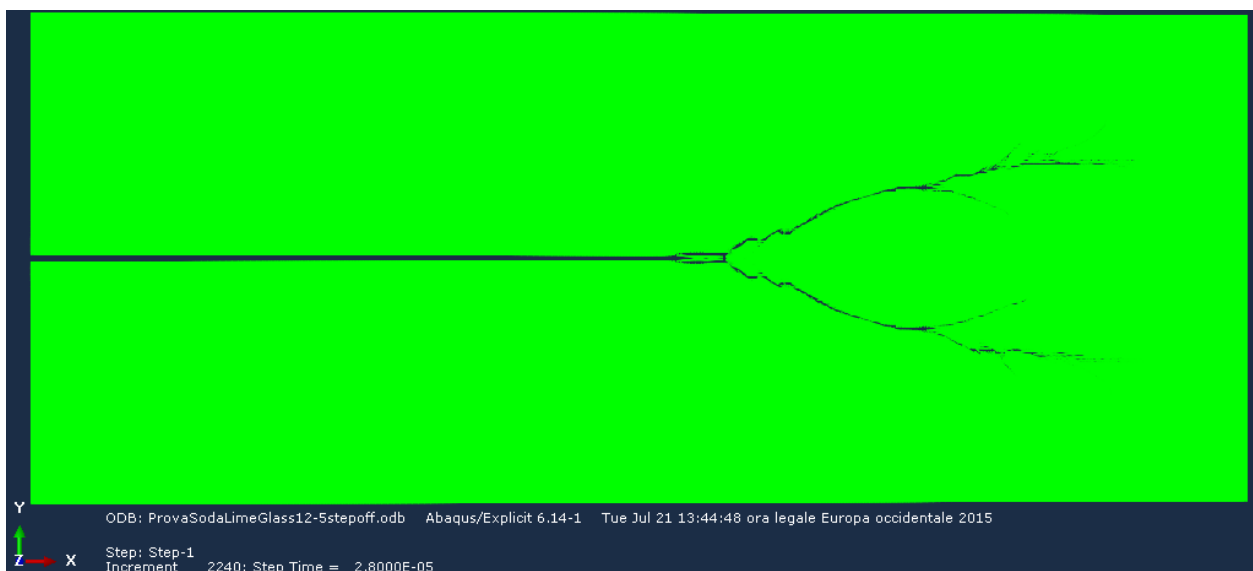


Figura 2-34 - (Immagine Abaqus) Piastra in Soda Lime Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m

⁷⁸ Immagine tratta da: Ha Y. D., Bobaru F. *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010

2.4.1.2. Risultati a 50 μ s

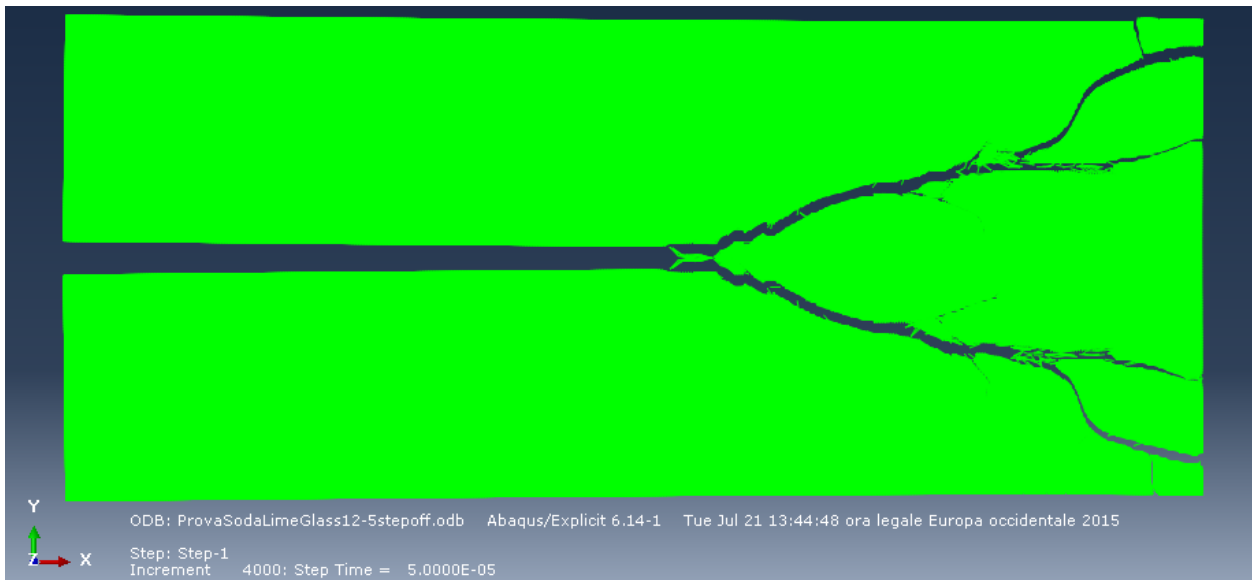


Figura 2-35 - (Immagine Abaqus) Piastra in Soda Lime Glass con $\delta=0,001$ m $m=4$ e $\Delta x=0,00025$ m a simulazione terminata

3. Simulazione con Abaqus dell'esperimento di Kalthoff - Winkler

In questo capitolo è stato simulato l'esperimento di Kalthoff – Winkler⁷⁹, che consiste nel far impattare a velocità controllata un proiettile contro una piastra in cui sono state introdotte due cricche iniziali. Esso viene utilizzato per testare la correttezza di un metodo di simulazione per la previsione dell'andamento delle cricche. Se in *input* si inseriscono i dati iniziali di velocità e le caratteristiche del materiale, il programma deve essere in grado di riprodurre l'andamento della cricca come nel provino. Quando ciò avviene la procedura è corretta.

La differenza tra queste simulazioni e le precedenti è che mentre prima vi era solamente trazione nei *bond*, qui vi è sia trazione che compressione. Dalla teoria della peridinamica (§ 1.11) un *bond* con materiale microelastico può rompersi in trazione ed in compressione, ma nel caso in esame il valore dell'allungamento critico in trazione viene effettivamente raggiunto, mentre in compressione questo è talmente elevato che non ha significato pratico prenderlo in considerazione. Si è quindi provveduto a creare un sottoprogramma denominato *Vumat*⁸⁰ che modifica le caratteristiche del materiale lineare elastico di Abaqus. Questo consente di avere un materiale elastico in grado di rompersi solo a trazione mentre viene inibita la rottura in compressione.

In Abaqus non è presente questo tipo di materiale e questo criterio di rottura e, quindi, nel programma, dove è presente la sezione che descrive il materiale, viene compilata una tabella con i valori di quest'ultimo. In fase di compilazione, verranno inviati alla *Vumat* per essere elaborati e da questa rinvii al programma. La *Vumat* non va in alcun modo ad intaccare la struttura interna di calcolo di Abaqus, ma prendendo i valori di stress e di deformazione si individua se il *bond* è rotto oppure integro. Si utilizzano perciò tutte le potenzialità del programma senza influire sull'andamento dei risultati.

⁷⁹ J.F. Kalthoff, S. Winkler. Failure mode transition at high rates of shear loading, ISBN 3-88355-125-2, 1988

⁸⁰ Vedi Appendice B

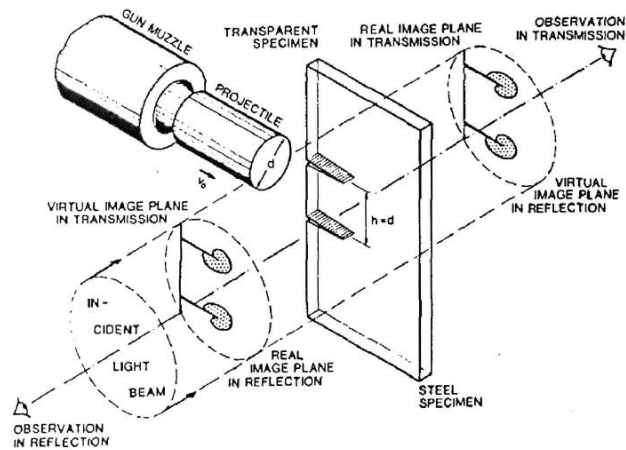


Figura 3-1 - Schema dell'esperimento di Kalthoff – Winkler

Dalla figura 3-1 si nota che l'impatto avviene su tutta la superficie compresa fra le due scanalature in modo uniforme. Ciò permette di semplificare in fase di simulazione il proiettile. Non essendo interessante simulare il proiettile, lo si modella assegnando al bordo dove avrebbe dovuto impattare una determinata velocità. Ciò semplifica in maniera notevole il carico di lavoro al programma non pregiudicando ciò che accade nella piastra.

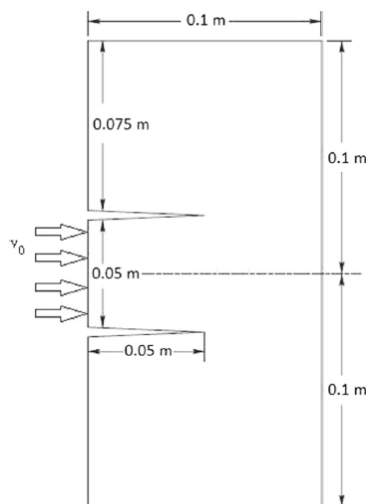


Figura 3-2 – Geometria del modello

Al termine della simulazione il risultato deve essere due cricche con rispettivamente angolazione rispetto al piano orizzontale di 70° verso l'alto e di 70° verso il basso. Queste partono dalle estremità delle due scanalature e fuoriescono con un angolo retto dalla superficie superiore ed inferiore della piastra.

3.1. Simulazioni dell'esperimento di Kalthoff – Winkler

Si vuole simulare l'esperimento di Kalthoff-Winkler con due diversi valori di allungamento critico dei *bond*.

Come primo caso di allungamento critico viene riproposta con Abaqus la simulazione esposta in "Crack propagation with adaptive grid refinement in 2D peridynamics"⁸¹ simulata interamente con Matlab. La geometria del provino simulato è quella della figura 46 con una velocità pari a 16,5 m/s pari a circa la metà della velocità di impatto reale.

Dati del materiale:

Nome del materiale: Stell 18Ni1900

E (Modulo di Young): $1,90 \cdot 10^{11}$ Pa

ν (Modulo di Poisson): 0,333

Densità: 8000 kg/m³

G_0 (Fracture Energy): 22170 J/m²

La velocità non è stata applicata ad una singola fila di nodi bensì a 3 file (3 è anche il valore di m), in quanto la zona di bordo del provino essendo simulata con *bond* non indistruttibili, risulta con caratteristiche leggermente diverse rispetto al resto della piastra. Questa semplificazione non influisce sui risultati della simulazione in quanto la zona è molto ridotta rispetto all'estensione della piastra. Siccome i nodi di bordo non hanno collegati a sé lo stesso numero di *bond* dei nodi presenti all'interno della struttura si evince che la zona di bordo è meno resistente di quella interna.

Di seguito sono riportati i dati relativi alla simulazione:

Dati inseriti in Abaqus relativi alle condizioni al contorno:

Velocità applicata su ogni nodo del bordo compreso fra le due cricche: 16,5 m/s

⁸¹Dipasquale D., Zaccariotto M., Galvanetto U. *Crack propagation with adaptive grid refinement in 2D peridynamics*, Springer, DOI10.1007/s10704-014-9970-4, pg 1-22, 13/08/2014

Numero di righe di nodi a cui è applicata la velocità: 3

Dati relativi ai tempi della simulazione in Abaqus:

Durata simulazione: $9,00 \cdot 10^{-5}$ s

Step Temporale simulazione $7,00 \cdot 10^{-8}$ s

I valori inseriti sono gli stessi dell'articolo così da avere completa corrispondenza nei risultati. Sotto sono messe a confronto le figure ottenute con l'elaborazione grafica di Matlab dai risultati di Abaqus con le immagini presenti nell'articolo⁸².

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 3

Passo in direzione x: 0,001 m

Passo in direzione y: 0,001 m

Valore dell'orizzonte: 0,00303

Numero totale di nodi: 20000

Numero totale di elementi: 272830

$c = 1,957 \cdot 10^{19}$

$s_0 = 7,332 \cdot 10^{-3}$ m

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0.53 ; 0.702 ; 1.00

Valore di della sezione del *bond* non modificata da beta: $1.00 \cdot 10^{-4}$ m²

Valori delle sezioni in ordine crescente di valore di beta:

⁸² Dipasquale D., Zaccariotto M., Galvanetto U. Crack propagation with adaptive grid refinement in 2D peridynamics, Springer, DOI10.1007/s10704-014-9970-4, pg 1-22, 13/08/2014

$5,3 \cdot 10^{-5} \text{ m}^2$; $7,02 \cdot 10^{-5} \text{ m}^2$; $1,00 \cdot 10^{-4} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $8,00 \cdot 10^{-3} \text{ kg}$

Valore del modulo di elasticità del *bond*: $1,96 \cdot 10^{11} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Nell'articolo è mostrato l'intero evolversi della simulazione facendo vedere molti più *time step* dei due presenti di seguito. La propagazione della cricca con Abaqus è la stessa dell'articolo ma si è scelto di far vedere solo due momenti della simulazione particolarmente importanti, ossia quando si forma la cricca falsa a destra a $63 \mu\text{s}$ e lo stato finale a $90 \mu\text{s}$.

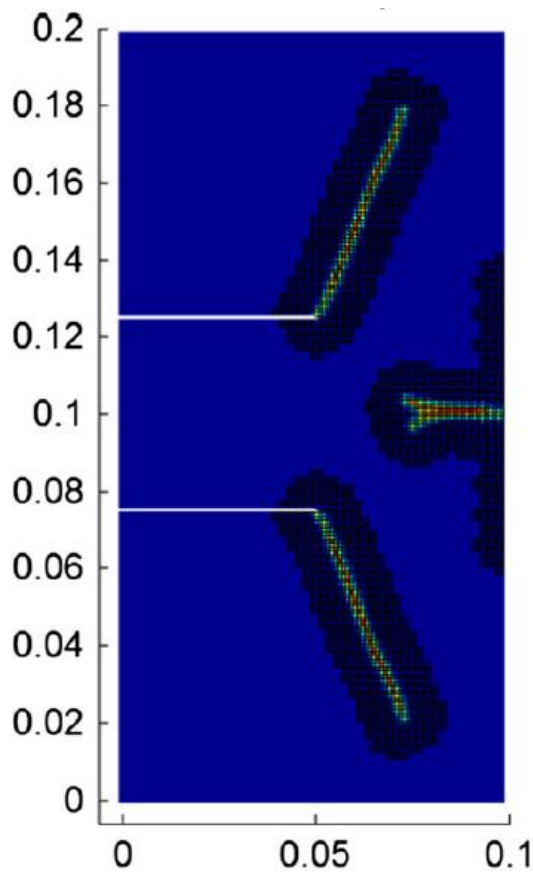
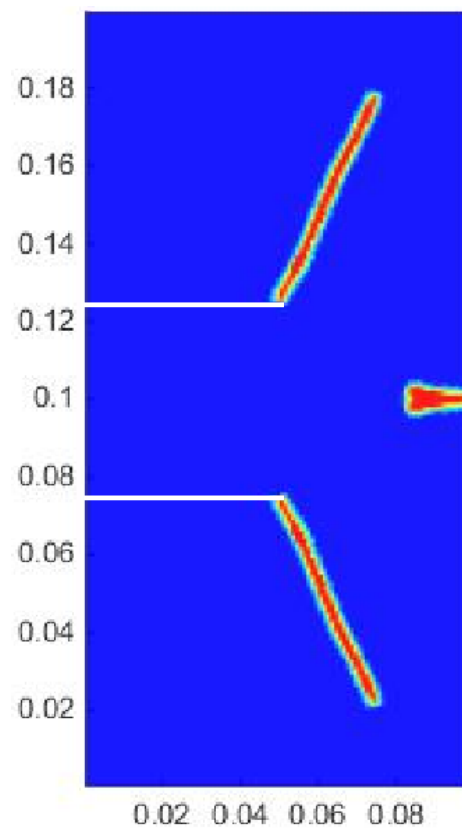


Figura 3-3 - A - Risultati dell'articolo a $63 \mu\text{s}$



B- Risultati elaborati da Matlab a $63 \mu\text{s}$

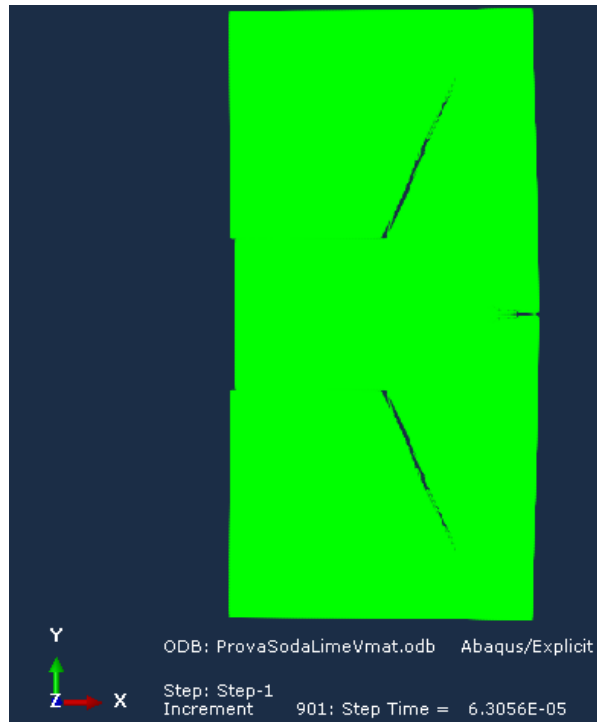


Figura 3-4 - Risultati della simulazione dell'esperimento di Kalthoff – Winkler generati da Abaqus a $63\mu\text{s}$

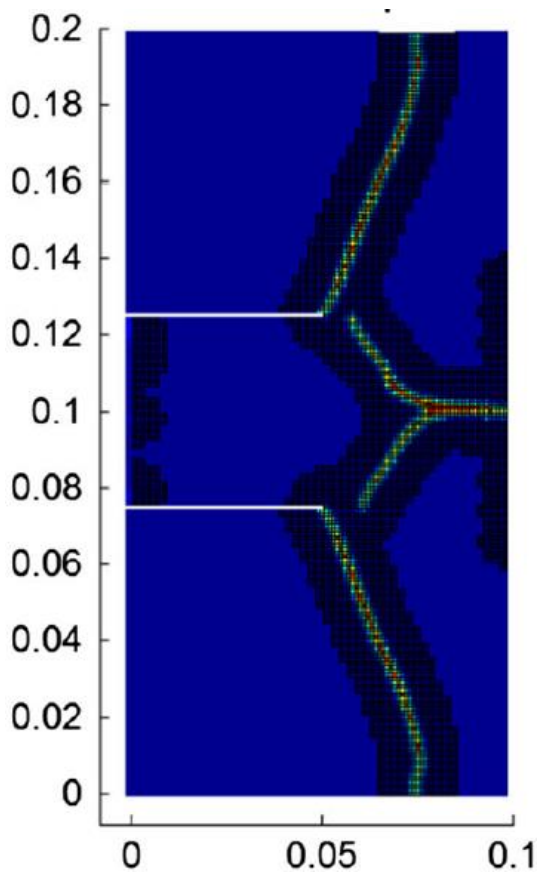
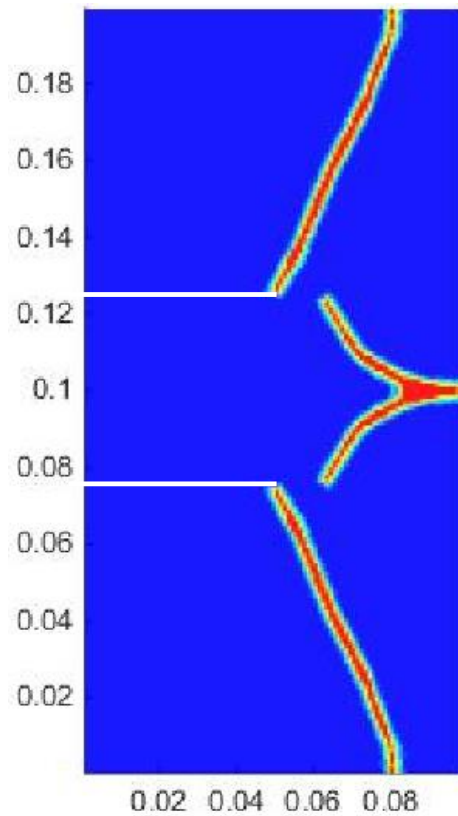


Figura 3-5 - A - Risultati dell'articolo a $90\mu\text{s}$



B- Risultati elaborati da Matlab a $90\mu\text{s}$

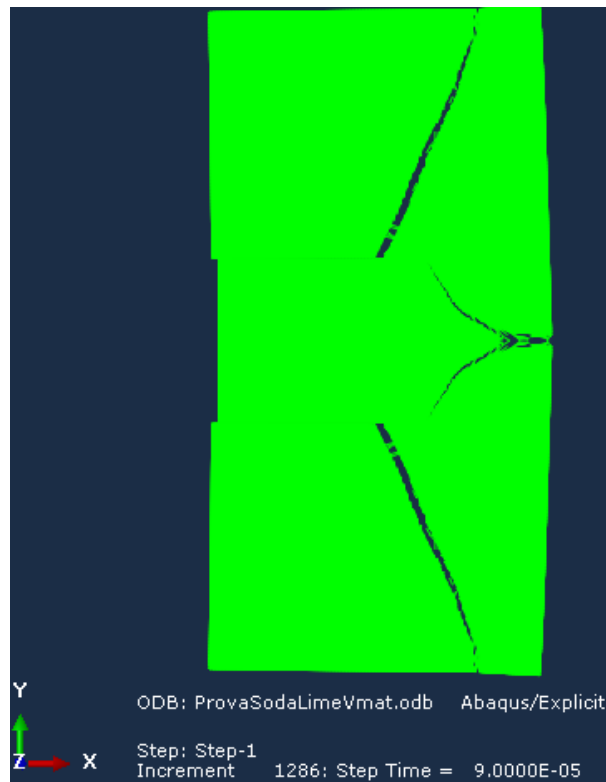


Figura 3-6 - Risultati della simulazione dell'esperimento di Kalthoff – Winkler generati da Abaqus a 90 μ s

Dal confronto fra le figure (figura 3-3 e figura 3-5) si vede immediatamente che i risultati delle simulazioni in Abaqus sono identici a quelli descritti nell'articolo. Le due cricche che partono dalle scanalature hanno angolazione di +68° quella superiore e, -68° quella inferiore rispetto al piano orizzontale, questo è compatibile con il risultato sperimentale di 70°. Si nota però una cricca che parte dalla superficie destra della piastra e che poi si biforca, essa non è presente nell'esperimento di Kalthoff – Winkler reale. Questa apparente incoerenza con l'esperimento reale è dovuta alle formule della peridinamica per il calcolo di s_0 (allungamento critico, valore oltre il quale i *bond* si rompono) che risulta essere di $7,332 \cdot 10^{-3}$ m. Tale valore risulta essere troppo basso e fa sì che si attivano sistemi di cricche non reali.

Nella simulazione del secondo caso di allungamento critico⁸³ quest'ultimo valore viene posto uguale a 0,01 m e la cricca destra non si presenta più. A parte questa differenza dalle immagini si può vedere che viene simulata correttamente la cricca sia per il fatto che l'inclinazione di questa è di circa 70° sia per il fatto che anche con Abaqus il "percorso di uscita" è perpendicolare alla superficie dei bordi superiore e inferiore nel tratto finale.

⁸³ Vedi Madenci E., Erkan Oterkus E. *Peridynamic theory and its applications*, Springer, DOI10.1007/978-1-4614-8465-3, 2014

La simulazione del libro “*Peridynamic theory and its applications*” viene effettuata con un software chiamato EMU⁸⁴. I risultati ottenuti con Abaqus sono identici a quelli ottenuti con quest’ultimo.

Sotto sono riportati i dati della simulazione:

Dati:

Dati del materiale:

Nome del materiale: Stell 18Ni1900

E (Modulo di Young): $1,91 \cdot 10^{11}$ Pa

ν (Modulo di Poisson): 0,33

Densità: 8000 kg/m^3

G_0 (Fracture Energy): 22170 J/m^2

Dati inseriti in Abaqus relativi alle condizioni al contorno:

Velocità applicata su ogni nodo del bordo compreso fra le due cricche: 16,5 m/s

Numero di righe di nodi a cui è applicata la velocità: 3

Dati relativi ai tempi della simulazione in Abaqus:

Durata simulazione: $1,1745 \cdot 10^{-7}$ secondi

Step Temporale simulazione $8,70 \cdot 10^{-8}$ secondi

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m : 3

Passo in direzione x: 0,001 m

Passo in direzione y: 0,001m

⁸⁴ Software che utilizza la teoria della Peridinamica

Numero totale di nodi: 20000

Numero totale di elementi: 349650

$c: 1,97 \cdot 10^{19}$

$s_0: 0,01 \text{ m}$

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,368 ; 0,53 ; 0,702 ; 1,00

Valore di della sezione del *bond* non modificata da beta: $1,00 \cdot 10^{-4} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta:

$3,68 \cdot 10^{-5} \text{ m}^2$; $5,3 \cdot 10^{-5} \text{ m}^2$; $7,02 \cdot 10^{-5} \text{ m}^2$; $1,00 \cdot 10^{-4} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $8,00 \cdot 10^{-3} \text{ kg}$

Valore del modulo di elasticità del *bond*: $1,97 \cdot 10^{11} \text{ Pa}$

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

Dati inseriti in Abaqus relativi alle condizioni al contorno:

Velocità applicata su ogni nodo del bordo compreso fra le due cricche: 16,5 m/s

Numero di righe di nodi a cui è applicata la velocità: 3

Dati relativi ai tempi della simulazione in Abaqus:

Durata simulazione: $1.1745 \cdot 10^{-7} \text{ s}$

Step Temporale simulazione $8.70 \cdot 10^{-8} \text{ s}$

Risultati:

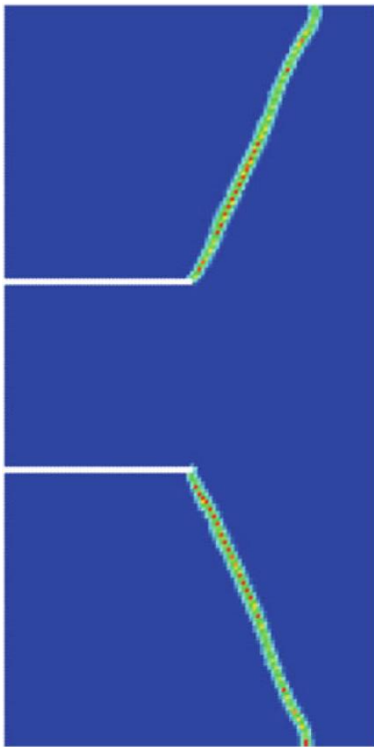
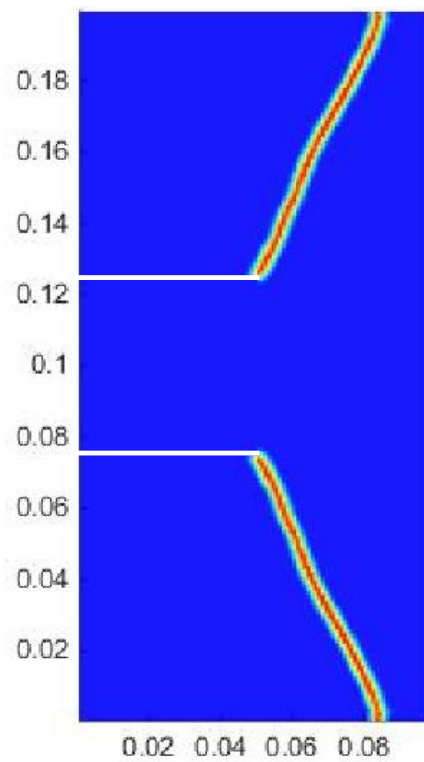


Figura 3-7 – A - Risultati del libro



B- Risultati generati da Matlab

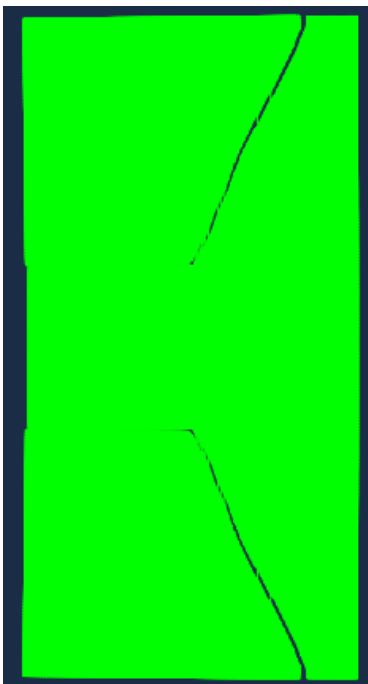


Figura 3-8 - Risultati della simulazione dell'esperimento di Kalthoff – Winkler generati da Abaqus

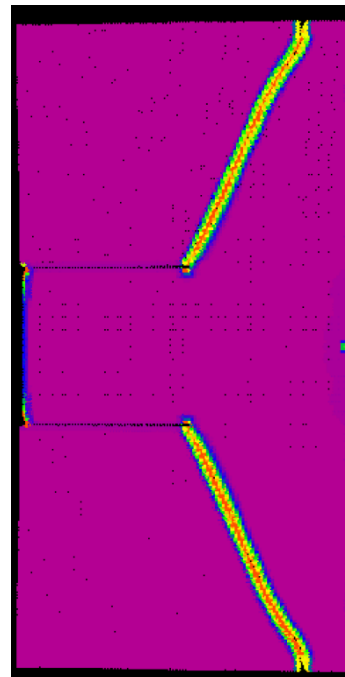


Figura 3-9 - Risultati con EMU⁸⁵

⁸⁵ Siling S.A., *Peridynamic modeling of the Kalthoff-Winkler experiment*, 9232, pg1-8, 2/1/2002

Dalle figure si nota che Abaqus ha simulato, senza inserire altro che i valori del materiale e della velocità, il corretto andamento delle cricche sia per quanto riguarda l'inclinazione (di +68° e -68°) sia per quanto riguarda l'angolo di uscita. Nell'articolo "*Peridynamic modeling of the Kalthoff-Winkler experiment*" Siling S.A.⁸⁶ viene simulato l'esperimento di Kalthoff-Winkler con il software EMU con implementata la Teoria della Peridinamica. Il risultato ottenuto con quest'ultimo programma è illustrato in figura 3-9 e non ci sono differenze con i risultati ottenuti con Abaqus.

⁸⁶ Vedi nota 85 p.82

4. Simulazione con Abaqus di un impatto fra un corpo rigido modellato con gli elementi finiti e una piastra modellata con la teoria della Peridinamica

Dopo aver validato il modello e il sottoprogramma Vumat⁸⁷ con le simulazioni precedenti, si procede ora ad analizzare il caso di un cilindro infinitamente lungo che impatta su una piastra infinitamente lunga anch'essa. Dalla geometria sopra descritta si può semplificare l'analisi considerando una sezione infinitamente sottile della geometria e di conseguenza utilizzare le formule del caso piano. La convenienza ad utilizzare Abaqus per questo tipo di analisi deriva dal fatto che il programma ha già al suo interno le funzioni che gestiscono il contatto fra i corpi. Come nel caso dell'esperimento di Kalthoff-Winkler, Abaqus non contiene di *default* il materiale dei *bond* e quindi è stato scritto un nuovo sottoprogramma Vumat per questa situazione. Rispetto alla simulazione dell'esperimento di Kalthoff-Winkler, in cui le deformazioni in compressione erano presenti ma non rilevanti, negli impatti reali le deformazioni in compressione sono elevate e di conseguenza le forze di corto raggio, derivate dalla teoria della Peridinamica⁸⁸, influiscono sulla lunghezza finale del *bond*. Il nuovo Vumat è quindi uguale al precedente come struttura e come modo di interagire con Abaqus, ma possiede al suo interno la formula delle forze di corto raggio. Questa influenza il valore della lunghezza del *bond* e quindi la distanza fra i nodi.

Il modello è costituito da una piastra e un cilindro rigido, a cui viene applicata una velocità iniziale di volta in volta differente così da visualizzare i vari tipi di rottura della piastra che ne derivano. La piastra è in Soda Lime Glass e sono stati utilizzati gli stessi valori del §2.4. Non è stata fatta alcuna semplificazione se non quelle già discusse nelle simulazione precedenti che non influiscono sui risultati finali. I percorsi che le fratture seguono sono direttamente calcolati da Abaqus senza aver posto inizialmente alcuna condizione di partenza o cricca iniziale. Sotto sono indicati i dati del cilindro e della piastra. Sono state prese in esame le seguenti velocità iniziali del cilindro: 1m/s, 5m/s, 10 m/s, 20 m/s, 30 m/s, 50 m/s, 75 m/s, 100m/s.

⁸⁷ Vedi Appendice B

⁸⁸ Vedi §1.5

-Dati del cilindro:

Densità: 7500 kg/m^3

Corpo infinitamente rigido

-Dati della piastra:

Dati del materiale:

Nome del materiale: Soda Lime Glass

E (Modulo di Young): $7,20 \cdot 10^{10} \text{ Pa}$

ν (Modulo di Poisson): 0,33

Densità: 2440 kg/m^3

G_0 (Fracture Energy): 135 J/m^2

Dati inseriti e valori calcolati dal programma Matlab relativi alla peridinamica:

Valore di m: 4

Passo in direzione x: 0,0006 m

Passo in direzione y: 0,0006 m

Valore dell'orizzonte: 0,002424 m

Numero totale di nodi della piastra: 16000

Numero totale di elementi della piastra: 366454

$c = 1,45 \cdot 10^{19}$

$s_0 = 1,04 \cdot 10^{-3} \text{ m}$

Dati inseriti e valori calcolati dal programma Matlab relativi al *file main* per Abaqus:

Valori di beta: 0,54 ; 0,93 ; 1,00

Valore di della sezione del *bond* non modificata da beta: $5,06 \cdot 10^{-5} \text{ m}^2$

Valori delle sezioni in ordine crescente di valore di beta:

$2,73 \cdot 10^{-5} \text{ m}^2$; $4,73 \cdot 10^{-5} \text{ m}^2$; $5,06 \cdot 10^{-5} \text{ m}^2$

Massa applicata ad ogni singolo nodo: $8,78 \cdot 10^{-4} \text{ kg}$

Valore del modulo di elasticità del *bond*: $3,71 \cdot 10^{10} \text{ Pa}$

Valore di c_s (per le forze di corto raggio) = $5,56 \cdot 10^{11} \text{ Pa}$

($c_s = 15 E_{bond}$)

Valore di d_s (per le forze di corto raggio) = $0,00054 \text{ m}$

($d_s = 0,9 \Delta x$)

Valore della densità del *bond*: $1,00 \cdot 10^{-18} \text{ kg/m}^3$

La geometria del modello è mostrata in figura 4-2, come si può vedere la velocità iniziale è applicata al centro del cilindro e alle estremità inferiori della piastra sono assegnati i vincoli. E' stato deciso di vincolare i lati inferiori della piastra con un incastro. Non potendo applicare l'incastro al singolo nodo, in quanto connesso a troppi pochi nodi e quindi non rappresentativo di un incastro reale, si è applicato questo vincolo ad un "quadrato" di nodi con il lato pari a due volte il valore di m . Così facendo vengono considerati anche i nodi distanti due volte m dal bordo che sono completi di tutti i *bond*. Il fatto che i nodi del bordo non posseggono tanti *bond* quanti quelli all'interno rende il bordo meno resistente. In figura 4-1 è mostrata l'estremità sinistra della piastra con applicati gli incastri.

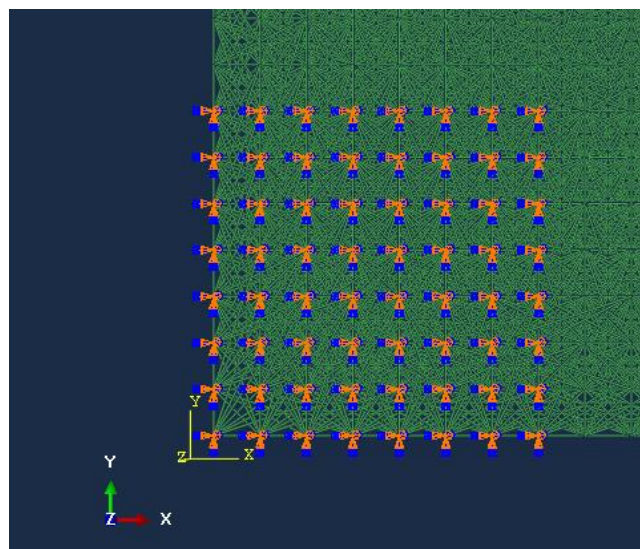


Figura 4-1 - Estremità sinistra con i nodi a cui è applicato l'incastro

In figura 4-2 è mostrato il modello con estensione “.cae” di Abaqus utilizzato per tutte le simulazioni a cui di volta in volta viene cambiata la velocità iniziale del cilindro.

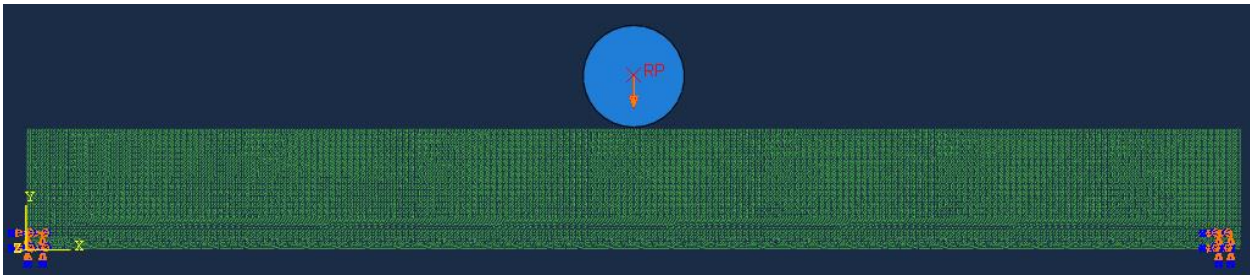


Figura 4-2 - Modello in Abaqus

Il contatto tra il cilindro impattante⁸⁹ e la piastra è stato realizzato mettendo in relazione, con i comandi di Abaqus, la superficie del proiettile⁹⁰ con i nodi della piastra. I *bond* non hanno alcuna condizione di contatto né fra di loro né con il proiettile. Il programma è stato previsto per visualizzare le cricche e le fratture a seguito di un impatto, ciò che succede negli istanti successivi non è realistico in quanto manca la condizione di contatto fra nodo e nodo e fra *bond* e *bond*. Queste condizioni di contatto avrebbero la funzione di modellizzare i vari pezzi di piastra che collidono fra loro a seguito dell'urto. Questo non è rilevante in questo elaborato in quanto non si vuole conoscere la traiettoria dei detriti ma solo individuare le zone di rottura e il loro propagarsi.

4.1. Impatto con velocità di 1 m/s

In questo paragrafo viene preso in considerazione l'impatto del cilindro con una velocità iniziale di 1 m/s (3,6 km/h). L'impatto non provoca danni se non due *bond* rotti, il *bond* numero 154 e il *bond* numero 9370 sui quasi trecentosettantamila *bond* totali. Questi due *bond* sono posizionati tra due nodi simmetrici fra di loro⁹¹ e collegano entrambi un nodo a cui è applicato un incastro. La rottura di questi due *bond* è stata rilevata esaminando il *report* dello stato generato da Abaqus. La rottura di solo questi due *bond* e il cambio del verso della velocità della sfera vista dai risultati e dal grafico delle velocità (figura 4-3) indicano che l'impatto è stato

⁸⁹ Denominato successivamente anche proiettile

⁹⁰ La superficie del proiettile è stata modellizzata come una circonferenza come rappresentato in figura 4-2.

⁹¹ Il primo nodo è a sinistra e il secondo a destra dell'asse di simmetria verticale passante per il punto medio della piastra.

quasi esclusivamente di tipo elastico e che il cilindro è rimbalzato senza provocare danni alla struttura se non quelli già descritti. La velocità iniziale della sfera è di 1 m/s verso il basso e quella finale di 0,594 m/s verso l'alto.

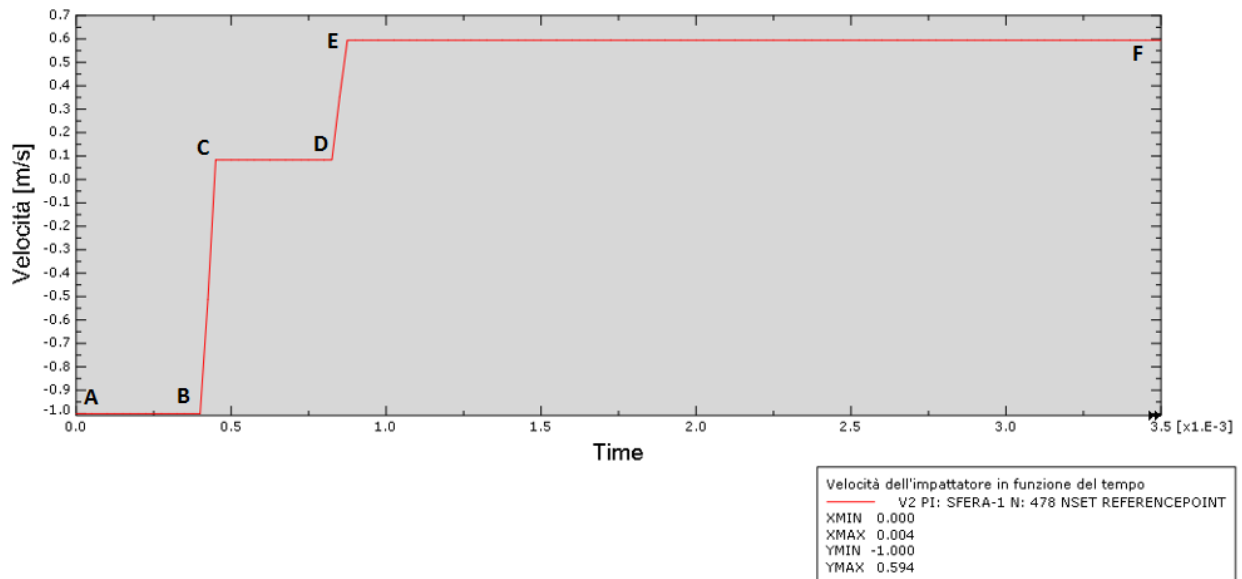


Figura 4-3 - Grafico della velocità del proiettile in funzione del tempo a 1m/s

In figura 4-4 è stata riportata la schermata in cui si vede la posizione del *bond 154* che è distante dalla zona di impatto e riguarda il collegamento fra un nodo della piastra e un nodo della stessa a cui è associato l'incastro. La posizione del *bond 9370* è speculare a quella del *bond 154*.

Dal grafico della velocità di figura 4-3, dopo la parte in cui il proiettile ha subito una decelerazione (tratto B-C), si può notare un tratto a quasi 0 m/s di velocità (tratto C-D), e successivamente un nuovo tratto di accelerazione (tratto D-E) per poi infine stabilizzarsi (tratto E-F). Il tratto C-D è dovuto al fatto che la piastra ha assorbito parte dell'energia cinetica del proiettile trasformandola in energia elastica mediante la propria deformazione. L'accelerazione riscontrata nel tratto D-E, successivo al tratto C-D, è dovuta al ritorno elastico del materiale. Questo ritorno è molto più rapido della velocità del cilindro dovuta al solo impatto, di conseguenza si verifica un secondo urto. Questo secondo impatto, realizzato ora dalla piastra nei confronti del cilindro, imprime un'accelerazione al proiettile che termina quando la piastra ritorna alla posizione iniziale. Il ritorno alla posizione iniziale ha consentito alla piastra di

dissipare tutta l'energia elastica. Finito questa fase transitoria il proiettile non essendo soggetto a nessuna forza mantiene la sua corsa a velocità costante.

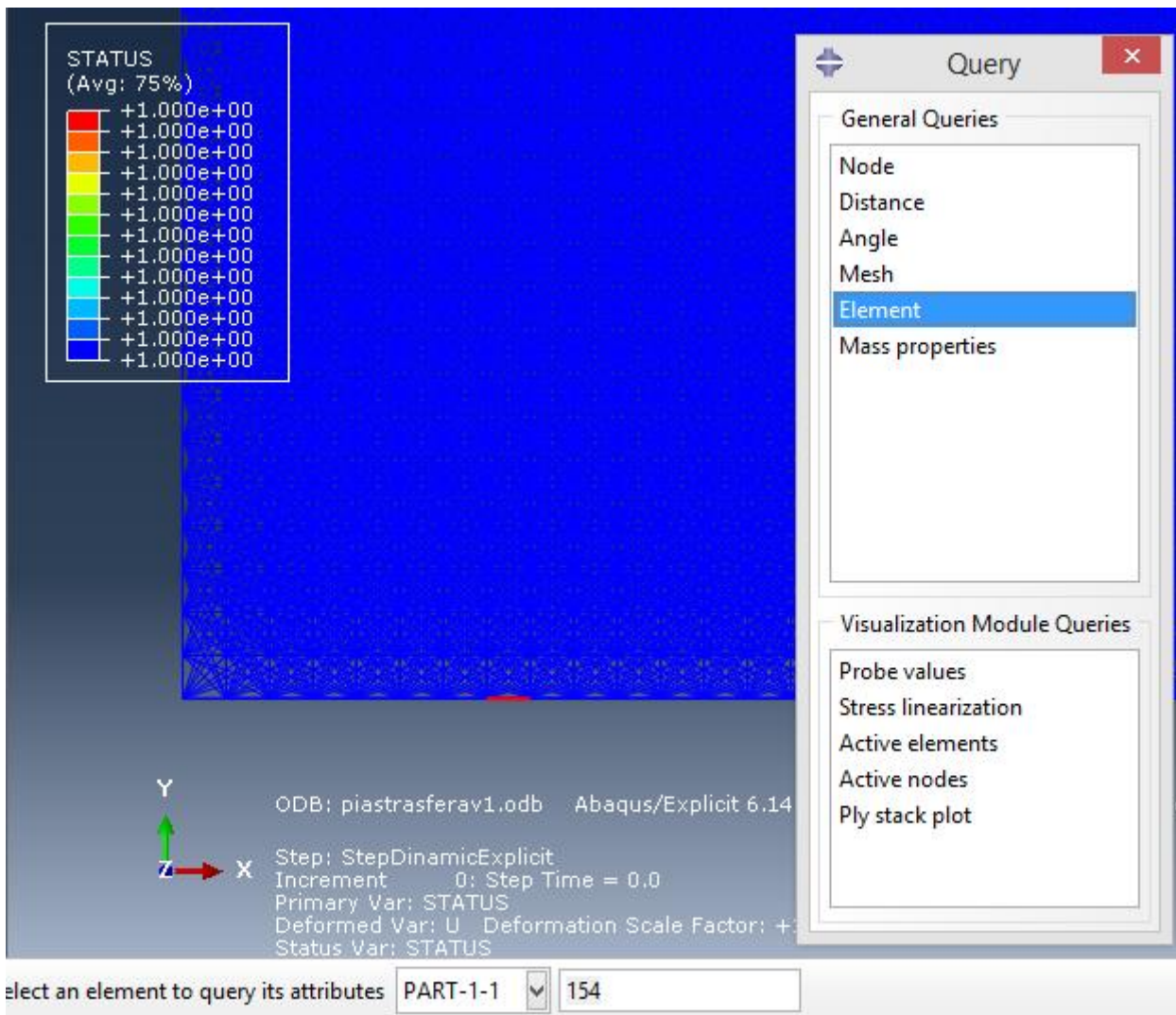


Figura 4-4 - Posizione del *bond* 154

4.2. Impatto con velocità di 5 m/s

Nell'impatto a 5 m/s (18 km/h) si ha una piccola frattura a metà della piastra lungo tutto lo spessore ed altre fratture nelle vicinanze dei vincoli di incastro. Il cilindro dopo l'impatto rimbalza con una velocità di 0,436 m/s verso l'alto, di molto inferiore a quella iniziale. Il meccanismo di frattura ha assorbito parte dell'energia del cilindro facendolo arrestare e ripartire nel verso opposto. Oltre all'immagine risultante da Abaqus è stata creata con Matlab un'immagine in cui viene visualizzato il danno che la piastra ha riportato con il medesimo

codice colori utilizzato in precedenza⁹². Si può vedere la completa corrispondenza fra le zone prive di *bond* di Abaqus (zone di frattura) e le zone rosse di Matlab in cui la concentrazione di *bond* rotti è superiore al 60%. Nel caso di zone in cui i *bond* avessero subito una rottura ma non in numero tale da creare una frattura, l'immagine Abaqus non le avrebbe messe in evidenza. Nell'immagine risultante quindi sarebbero stati presenti solo i *bond* integri, Matlab invece avrebbe assegnato a tali zone un colore diverso dal blu (anche se non rosso) e di conseguenza sarebbero state facilmente individuate.

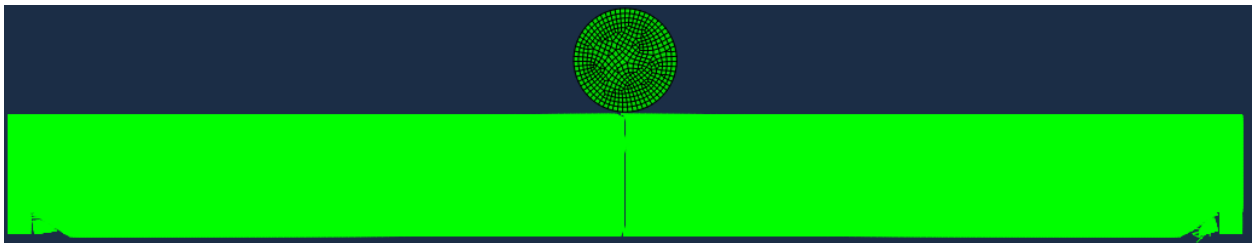


Figura 4-5 - Immagine risultante di Abaqus per l'impatto a 5 m/s a 0.0005 s

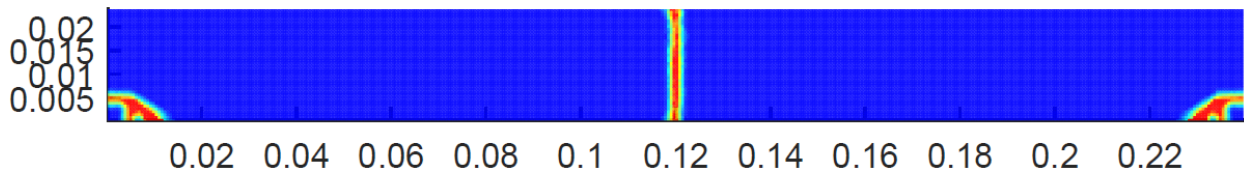


Figura 4-6 - - Immagine del danno da Matlab per l'impatto a 5 m/s a 0.0005 s

In figura 4-7 è mostrato un ingrandimento della frattura dopo l'impatto.

⁹² Vedi immagini Capitolo 2

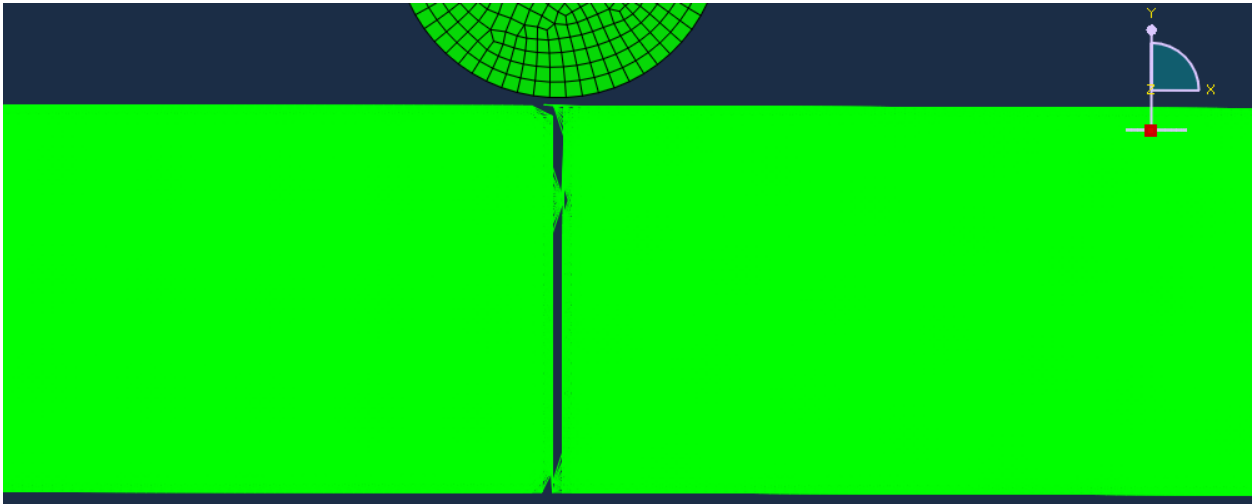


Figura 4-7 - Particolare dell'impatto a 5 m/s

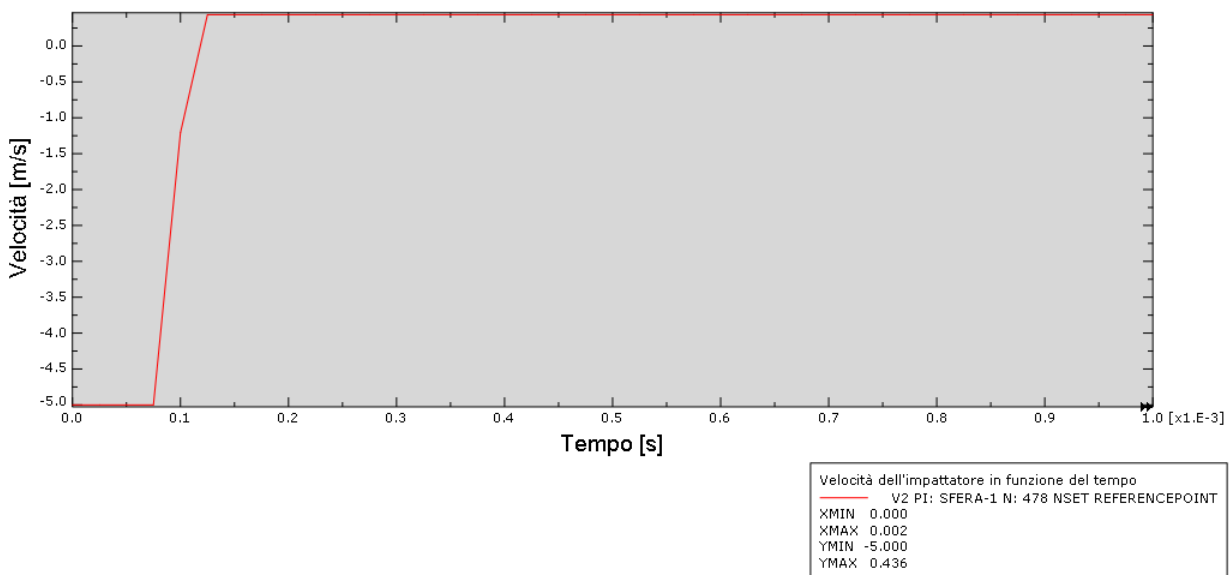


Figura 4-8 - Grafico della velocità del proiettile in funzione del tempo a 5 m/s

4.3. Impatto con velocità di 10 m/s

L'impatto a 10 m/s (36 km/h) a differenza di quello a 5 m/s crea una serie di cricche ramificate. Dall'immagine si nota che sono simmetriche rispetto all'asse verticale passante per il punto medio della piastra. Il cilindro dopo l'impatto ha una velocità di 0,344 m/s verso l'alto, perciò c'è stato un piccolo rimbalzo. L'immagine elaborata da Matlab (figura 4-10) mette in evidenza in maniera più chiara il percorso delle cricche rispetto all'immagine di Abaqus (figura 4-9).

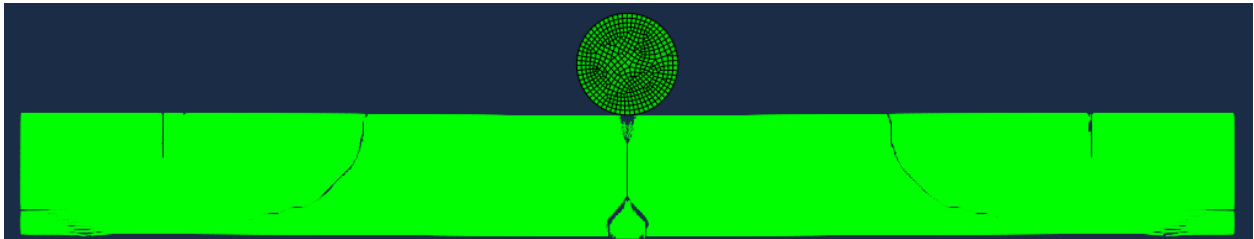


Figura 4-9 - Immagine risultante di Abaqus per l'impatto a 10 m/s a 12,5 μ s

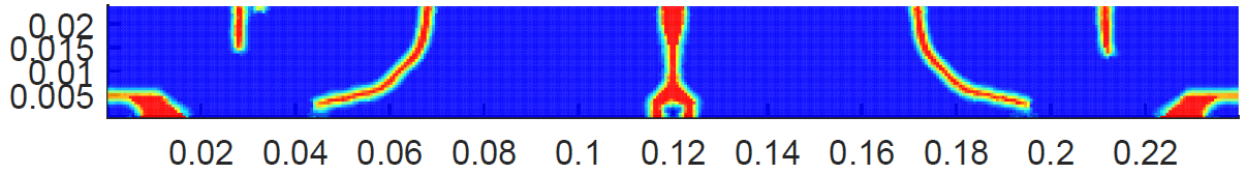


Figura 4-10 - Immagine del danno da Matlab per l'impatto a 10 m/s a 12,5 μ s

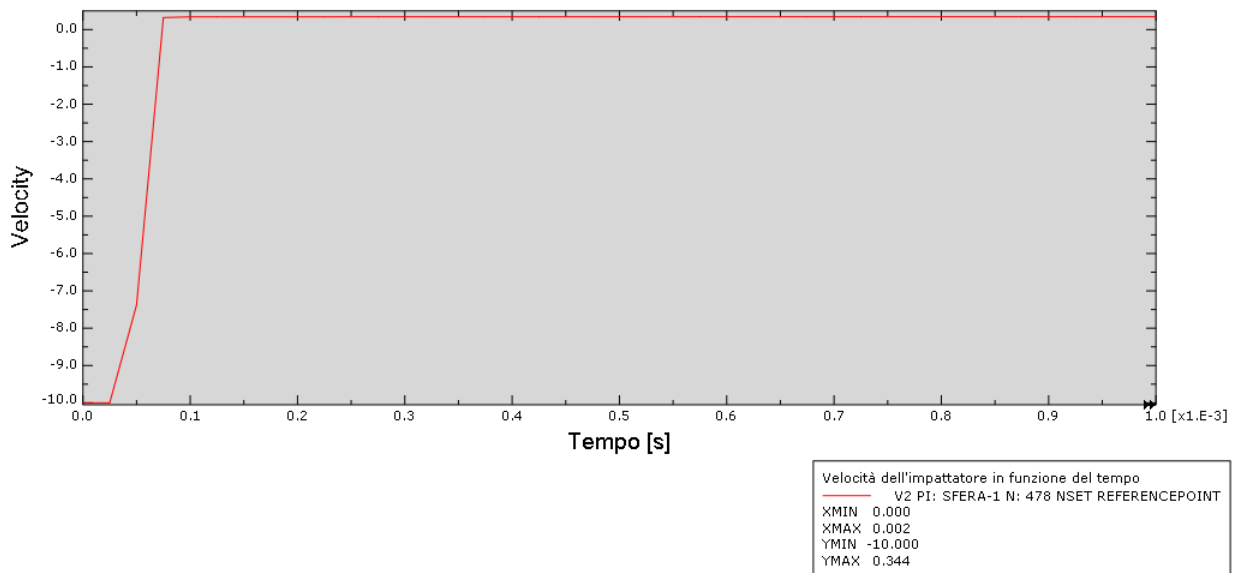


Figura 4-11 - Grafico della velocità del proiettile in funzione del tempo a 10 m/s

4.4. Impatto con velocità di 20 m/s

A 20 m/s (72 km/h) il percorso delle cricche diventa più complesso e ramificato rispetto al precedente ma segue uno schema analogo al caso di 10 m/s con la partenza delle cricche che tendono a muoversi verso i lati verticali. La frattura centrale è molto più complessa e ramificata. In questo caso il livello di dettaglio di Abaqus è maggiore rispetto a quello di Matlab

il quale non fa vedere le cricche più piccole. In Matlab queste cricche minori se si trovano in prossimità delle cricche principali vengono inglobate con quest'ultime. Il proiettile in questo caso non rimbalza ma prosegue la sua corsa nello stesso verso della velocità iniziale anche se con un valore nettamente inferiore ai 20 m/s (1.723 m/s).

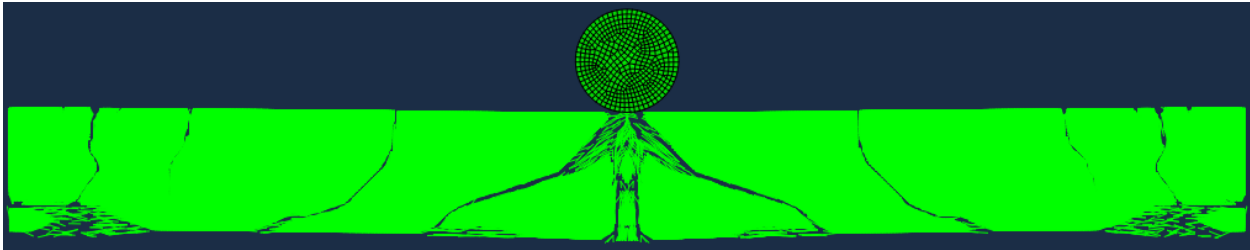


Figura 4-12 - Immagine risultante di Abaqus per l'impatto a 20 m/s a 12,5 μ s

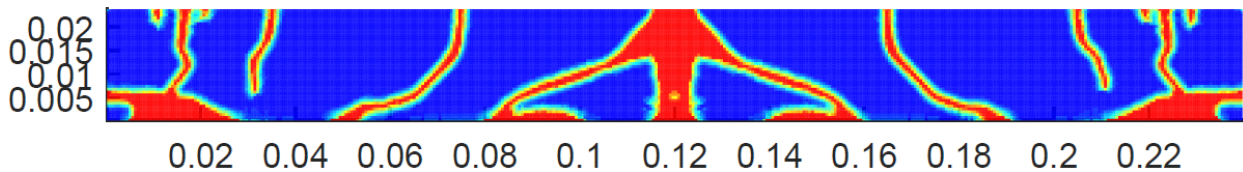


Figura 4-13 - Immagine del danno da Matlab per l'impatto a 20 m/s a 12,5 μ s

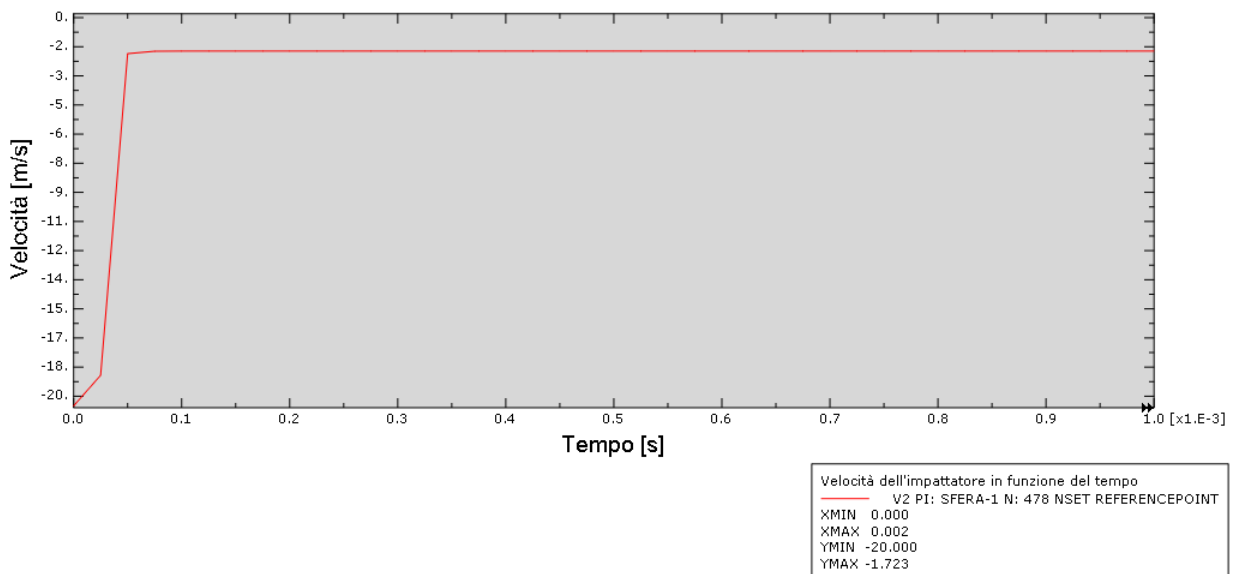


Figura 4-14 - Grafico della velocità del proiettile in funzione del tempo a 20 m/s a 12,5 μ s

In tabella 4-1 sono raffigurati in successione i vari step temporali della simulazione, così da poter mettere in evidenza la creazione, l'avanzamento e lo sviluppo delle cricche.

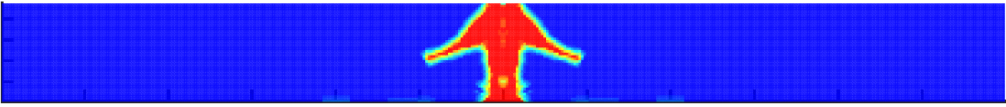
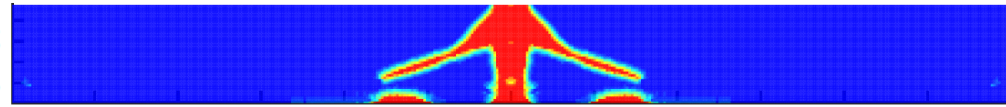
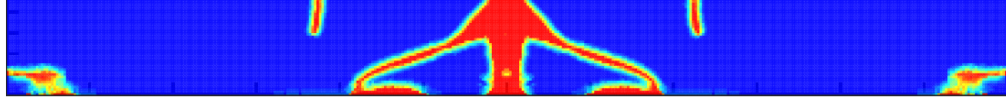
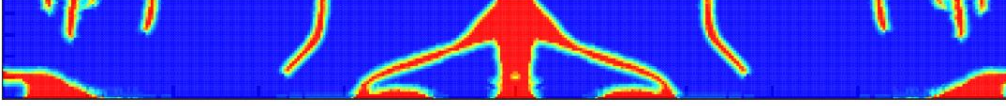
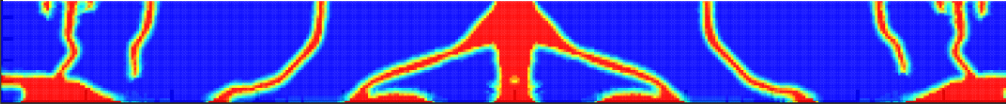
Step temporale [10 ⁻⁴ s]	Immagine del danno in Matlab con velocità del proiettile di 20 m/s	
4	0.02 0.015 0.01 0.005	 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22
5	0.02 0.015 0.01 0.005	 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22
6	0.02 0.015 0.01 0.005	 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22
9	0.02 0.015 0.01 0.005	 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22
11	0.02 0.015 0.01 0.005	 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2 0.22

Tabella 4-1 - Tabella immagini relative alla propagazione delle cricche a 20 m/s

4.5. Impatto con velocità di 30 m/s

Aumentando la velocità iniziale del proiettile a 30 m/s (108 km/h) questo non rimbalza ma prosegue con una velocità di 2.245 m/s, è stato quindi notevolmente rallentato ma non fermato. Si notano danni evidenti soprattutto nelle vicinanze dei vincoli e nella zona dell'impatto.

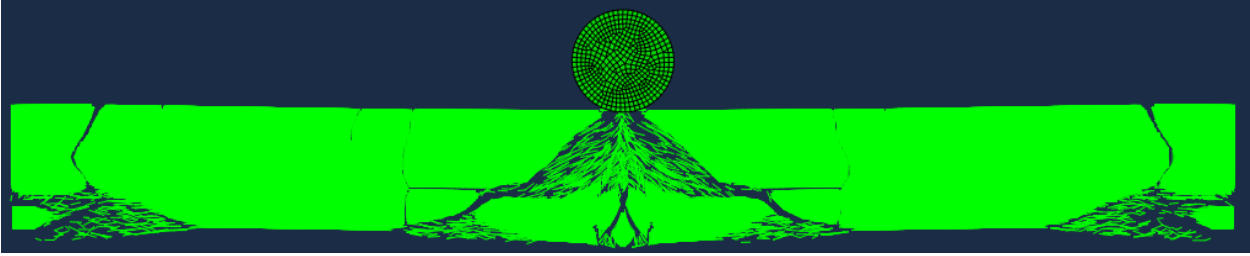


Figura 4-15- Immagine risultante di Abaqus per l'impatto a 30 m/s a 12,5 μ s

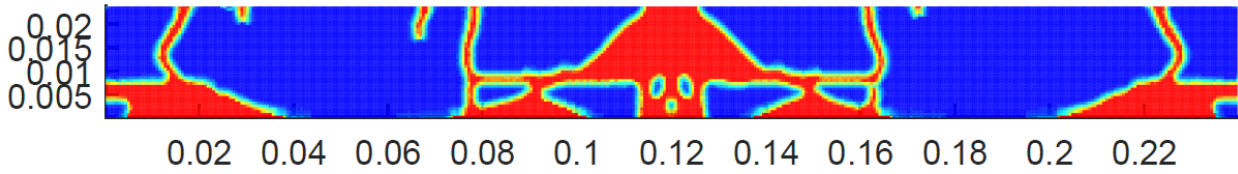


Figura 4-16 - Immagine del danno da Matlab per l'impatto a 30 m/s a 12,5 μ s

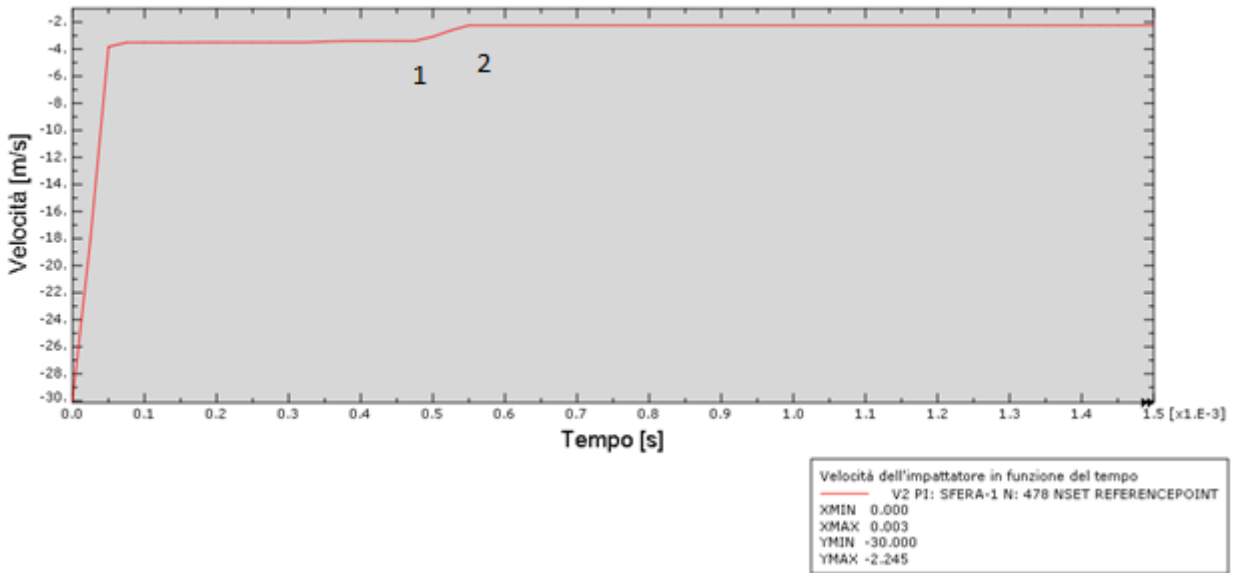


Figura 4-17 - Grafico della velocità del proiettile in funzione del tempo a 30 m/s

Dal grafico delle velocità di figura 4-17 si nota un lieve cambio di velocità nell'intorno dei 0.0005 s (evidenziati con i punti 1 e 2 nella figura 4-17). Se si analizzano i risultati della simulazione di Abaqus si osserva che a quei valori di tempo avviene un impatto secondario. Un pezzo della piastra che aveva già subito un impatto ne subisce un altro e si frattura nuovamente. In questo secondo impatto il proiettile rallenta lievemente la sua velocità. Nelle figure 4-18, 4-19, 4-20 viene riportata la sequenza temporale del secondo impatto.

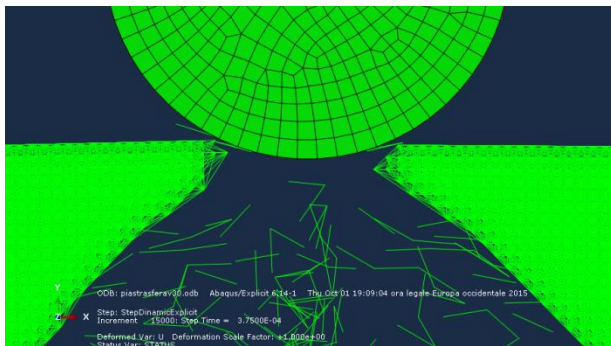


Figura 4-18 - Particolare dell'impatto a 30 m/s a 37,5 μ s

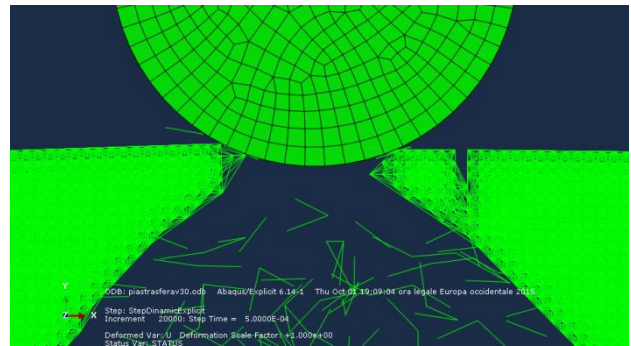


Figura 4-19 - Particolare dell'impatto a 30 m/s a 50 μ s

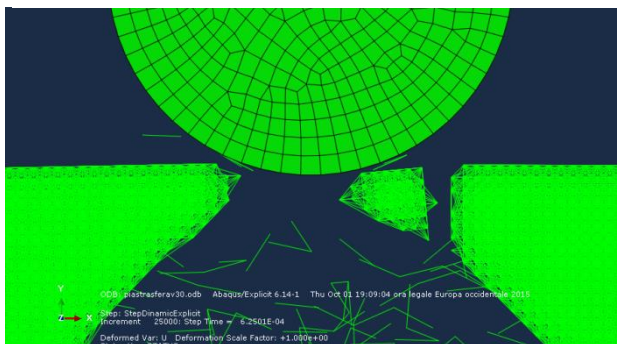


Figura 4-20 - Particolare dell'impatto a 30 m/s a 62,5 μ s

4.6. Confronto fra impatti a velocità elevate: impatti con velocità di 50 m/s, di 75 m/s, di 100 m/s

Aumentando la velocità oltre i 30 m/s il fenomeno delle cricche diventa sempre meno visibile fino ad arrivare al caso massimo qui in esame di 100 m/s in cui la piastra ha pochissime cricche ma si è rotta in tanti piccoli pezzi visibili in Abaqus. In Matlab invece la figura è quasi esclusivamente rossa e quindi con almeno il 60 % dei *bond* rotti. Nelle tabelle 4-2, 4-3, 4-4 sono riportate le figure elaborate da Abaqus, da Matlab e i grafici delle velocità del proiettile nei vari casi.

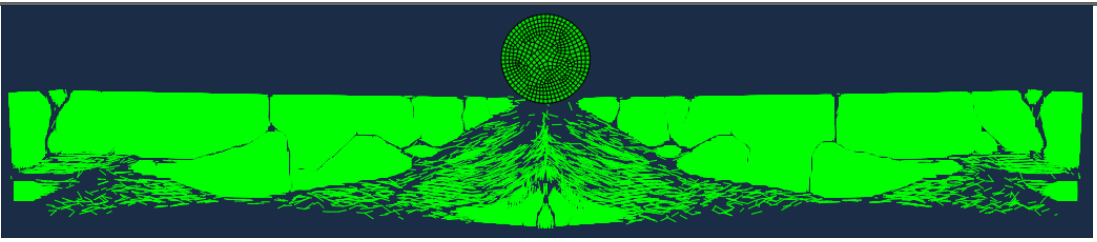
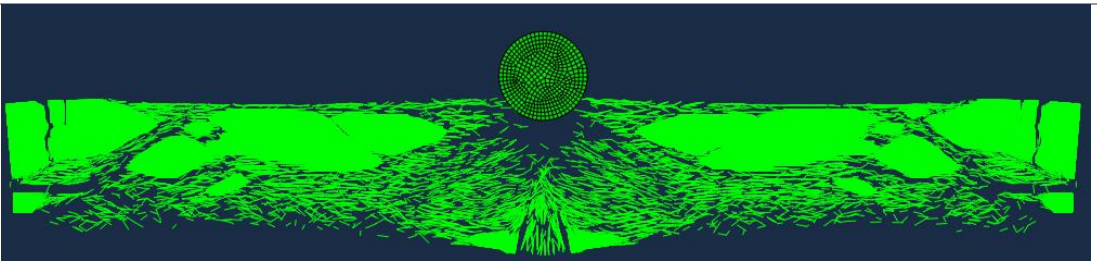
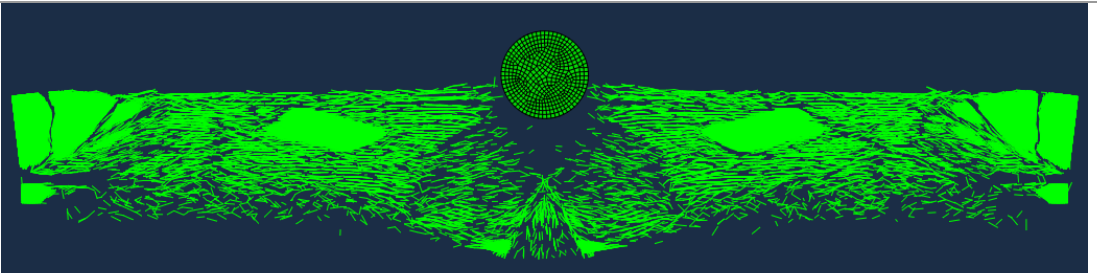
Velocità [m/s]	Velocità [km/h]	Immagine da Abaqus
50	180	
75	270	
100	360	

Tabella 4-2 - Tabella risultati di Abaqus per le velocità di 50 m/s, 75 m/s, 100 m/s

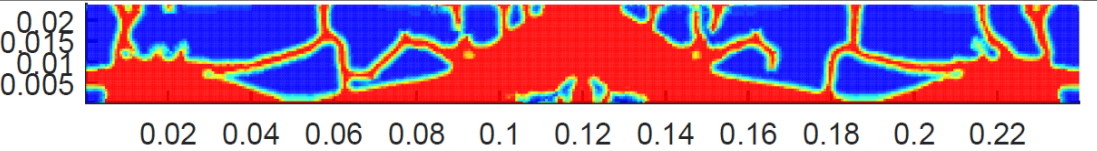
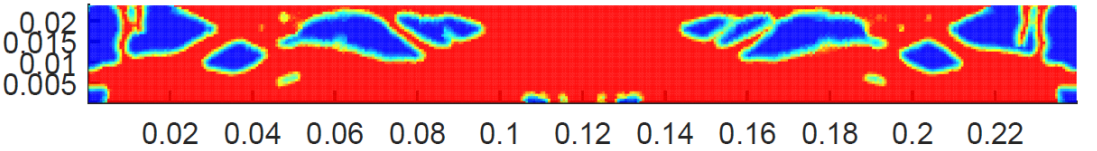
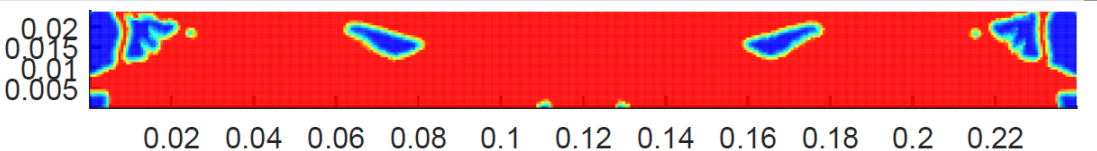
Velocità [m/s]	Velocità [km/h]	Immagine risultante ⁹³
50	180	
75	270	
100	360	

Tabella 4-3 - Tabella risultati di Matlab per le velocità di 50 m/s, 75 m/s, 100 m/s

⁹³ Con Immagine risultante si intende l'immagine elaborata con il programma Matlab dagli spostamenti e dagli stati dei bond presenti nei file report generati da Abaqus

Velocità [m/s]	Velocità [km/h]	Grafici delle velocità
50	180	<p>Velocità dell'impattatore in funzione del tempo V2 PI: SFERA-1 N: 478 NSET REFERENCEPOINT XMIN 0.000 XMAX 0.001 YMIN -50.000 YMAX -7.601</p>
75	270	<p>Velocità dell'impattatore in funzione del tempo V2 PI: SFERA-1 N: 478 NSET REFERENCEPOINT XMIN 0.000 XMAX 0.001 YMIN -75.000 YMAX -17.407</p>
100	360	<p>Velocità dell'impattatore in funzione del tempo V2 PI: SFERA-1 N: 478 NSET REFERENCEPOINT XMIN 0.000 XMAX 0.001 YMIN -100.000 YMAX -27.564</p>

Tabella 4-4 - Tabella con i grafici delle velocità del proiettile per le velocità di 50 m/s, 75 m/s, 100m/s

Velocità iniziale [m/s]	Velocità residua [m/s]	Energia cinetica assorbita dalla piastra [J]
20	1.723	198,52
30	2.245	447,48
50	7.601	1221,11
75	17.407	2660,99
100	27.564	4620,11

Tabella 4-5 - Tabella assorbimento energia cinetica

Se si fa la differenza fra energia cinetica iniziale e finale del proiettile (che nel caso in esame è rigido e quindi non ha dissipato energia con altri fenomeni che non siano la variazione di velocità), il valore di energia cinetica assorbito dalla piastra è di volta in volta maggiore. Come riportato in tabella 4-5. Si può dedurre che il maggior numero di cricche e fratture presenti negli ultimi casi aumenta la capacità del materiale di dissipare l'energia del proiettile. La piastra, nonostante possa rompersi attraverso cricche, possiede comunque un valore massimo di energia che può dissipare, che nei casi esaminati non è stato raggiunto.

Conclusioni e sviluppi futuri

Si è studiato il fenomeno della propagazione delle cricche applicando la teoria della Peridinamica ed effettuando le simulazioni con il programma Abaqus.

Abaqus è molto utilizzato in ambito accademico e non per simulazioni dinamiche con gli elementi finiti e tutt'ora continuamente aggiornato. Lo stesso si può dire di Matlab per suoi campi di applicazione.

L'interfaccia grafica che Abaqus usa per gestire le superfici, i punti in contatto fra loro e gli impatti è molto intuitiva dal punto di vista dell'utilizzatore il quale non è obbligato ad inserire limitazioni o ipotesi che possono compromettere i risultati della simulazione.

L'utilizzo di Matlab per la creazione degli elementi e dei nodi è solamente di supporto pratico per l'utente, il quale inserendo i dati della piastra ha in output i file da importare in Abaqus. Tali file non condizionano le procedure di calcolo di Abaqus ma contengono solamente il disegno del provino.

L'implementazione con le due subroutine Vumat ha consentito di risolvere il problema della non rottura in compressione dei *bond*. Esse hanno la funzione di modificare il materiale di default di Abaqus inibendo la rottura in compressione. La Vumat utilizzata per gli impatti tiene anche in considerazione le forze di corto raggio che sono proprie della Peridinamica, permettendo di modellare urti fra frammenti, o fenomeni di contatto fra le superfici di una cricca.

Diventa preferibile utilizzare l'approccio con Abaqus (integrato con le Vumat se vi è compressione) qualora la struttura da analizzare sia composta da varie parti e per alcune di queste (perché più deboli o con imperfezioni) si possa prevedere che abbiano maggiore probabilità di rottura rispetto alle altre. Si possono quindi modellare le "parti deboli" con la Peridinamica così da vedere l'evolversi di un'eventuale rottura. Le restanti parti, che si ipotizza non saranno soggette a rottura, possono invece essere modellate con gli elementi finiti sempre in Abaqus. Questo permette di ridurre il tempo totale di simulazione rispetto al caso in cui tutte le parti siano rappresentate con l'approccio peridinamico per mezzo di nodi e *bond*.

Il materiale inserito è unico in tutta la struttura delle piastre, ma si potrebbe implementare il programma per creare zone delle piastre con altra tipologia di materiale stabilendo però le caratteristiche dei *bond* nelle zone di confine.

E' inoltre possibile partendo dal programma base Abaqus di impatto cilindro/piastra creare altre forme ed altre caratteristiche del proiettile. In questa sperimentazione è stato considerato un proiettile rigido per semplificare la simulazione, ma nel programma Abaqus il proiettile è descritto in un set esterno alla struttura e quindi facilmente modificabile.

Con questo elaborato si è voluto dimostrare come sia possibile utilizzare un programma FEM per studiare la propagazione di cricche in un solido impiegando la teoria Perydinamics, nonostante questo approccio non sia ancora direttamente supportato dal solutore Abaqus.

Programma in Matlab per la creazione dei file inp per Abaqus

Non potendo definire manualmente tutti i *bond* è stato creato un programma in Matlab che date in ingresso le caratteristiche geometriche della piastra, i carichi applicati ed il materiale crea il *file main.inp* e i suoi file collegati che importati in Abaqus permettono di avere già la geometria della piastra. Sotto vengono riportate le principali sezioni del programma con spiegata la funzione.

Sezione per la creazione dei nodi

```
%-----  
%Ciclo While per la creazione della lista dei nodi  
%-----  
  
fprintf('Inizio creazione file nodi');  
fprintf('\n');  
  
id_nodo=0;      %numero identificativo nodo  
nododirx=0;    %avanzamento nodo direzione x  
nododiry=0;    %avanzamento nodo direzione y  
nododirt=0;    %avanzamento nodo direzione t  
  
B=[0,0,0,0];   %matrice con numero del nodo e coordinate del nodo  
  
%Parte relativa alla visualizzazione e salvataggio dati  
%per aprire il file di testo  
nodedefinition='nodedefinition';  
nome_file_da_inserire=horzcat(link_cartelle,link_nome_prova,nodedefinition,li  
nk_estensione_inp);  
ftx=fopen(nome_file_da_inserire,'w');  
  
if nt_elem==1  
while nododiry<(b-passoy)  
    if nododiry==0  
        nododiry=nododiry+passoy/2;  
    else  
        nododiry=nododiry+passoy;  
    end  
    nododirx=0;  
    while nododirx<(a-passox)  
        id_nodo=id_nodo+1;  
        if nododirx==0  
            nododirx=nododirx+passox/2;  
        else  
            nododirx=nododirx+passox;  
        end  
        fprintf(ftx,'%10i,          %12f,          %12f,          %12f  
\n',id_nodo,nododirx,nododiry,nododirt); %per creare il file  
  
        A=[id_nodo,nododirx,nododiry,nododirt];  
        if B==0
```

```

        B=A;
    else
        B=[B;A];
    end

    end

end

else
    fprintf('Sistema tridimensionale')
end

fclose(ftx); % per chiudere il file e salvarlo

```

Sezione per la creazione degli elementi *bond*

```

%-----
%Ciclo for per la creazione della lista degli elementi
%-----

mat_size1=((2*m+1)^2)*id_nodo; % dimensione verticale
massima della matrice
mat_size2=4; % dimensione orizzontale
massima della matrice
matrice_elem1=zeros(mat_size1,mat_size2); % matrice ausiliaria con
elementi doppi con dimensioni massime

%ciclo for per creare i bond---sono doppi
for i=1:id_nodo
    %calcolo della lunghezza dell'elemento
    dist_confr=((B(:,colonna_x)-B(i,colonna_x)).^2+(B(:,colonna_y)-
B(i,colonna_y)).^2).^(0.5);
    %condizione per avere la distanza corretta
    II=find((dist_confr<h)&(dist_confr~=0));
    matrice_elem1(id_elemento:id_elemento+length(II)-
1,:)=[[id_elemento:1:id_elemento+length(II)-
1]',i*ones(length(II),1),II,dist_confr(II)]; %formata da numero elemento,nodo
iniziale,nodofinale e distanza
    id_elemento=id_elemento+length(II);
end
matrice_elem=zeros(id_elemento,4); %creazione di una matrice
dove inserire gli elementi doppi
matrice_elem=matrice_elem1(1:id_elemento,1:4); % matrice con elementi
doppi senza gli zeri
matrice_elem1=[]; % distruzione
matrice_elem1 per occupare meno spazio

matrice_elem_fin1=zeros(id_elemento,4); % matrice ausiliaria con
elementi singoli con dimensioni massime

%ciclo for per eliminare i bond doppi
for i=1:id_elemento
    i;
    elem_aus2=matrice_elem(i,2);
    elem_aus3=matrice_elem(i,3);
    elem_aus4=matrice_elem(i,4);

    if elem_aus2<elem_aus3
        id_elemento_singoli=id_elemento_singoli+1;
    end
end

```



```

matrice_elem_fin1(id_elemento_singoli,:)= [id_elemento_singoli,elem_aus2,elem_
aus3,elem_aus4];
    end
end
matrice_elem_fin=matrice_elem_fin1(1:id_elemento_singoli,1:4);    %matrice con
tutti i bond presi singolarmente

```

Sezione per la creazione della cricca

In questa parte del programma vengono eliminati tutti i *bond* che attraversano la cricca iniziale in modo da creare una zona priva di elementi che simula la presenza dell'intaglio.

```

%-----
%CRICCA
%-----

fprintf('Inizio creazione cricca o cricche');
fprintf('\n');

%Creazione delle coordinate con i dati della cricca
x_iniz_cricca=zeros(1,id_numero_di_cricche);
y_iniz_cricca=ac;

x_fin_cricca=lc;
y_fin_cricca=ac;

indice_riferimento_cricca=0; %indice per il ciclo for creazione cricche

for i=1:id_numero_di_cricche

    indice_riferimento_cricca=indice_riferimento_cricca+1;

    %Creazione dell equazione implicita della cricca
    %estremi cricca

    p_iniz_cric=[x_iniz_cricca(1,indice_riferimento_cricca),y_iniz_cricca(1,indic
e_riferimento_cricca)];    %Punto iniziale cricca

    p_fin_cric=[x_fin_cricca(1,indice_riferimento_cricca),y_fin_cricca(1,indice_r
iferimento_cricca)];    %Punto finale cricca

    a_c=(y_iniz_cricca(1,indice_riferimento_cricca)-
y_fin_cricca(1,indice_riferimento_cricca));    % a b c retta implicita della
cricca
    b_c=(x_fin_cricca(1,indice_riferimento_cricca)-
x_iniz_cricca(1,indice_riferimento_cricca));

    c_c=y_iniz_cricca(1,indice_riferimento_cricca)*(x_iniz_cricca(1,indice_riferi
mento_cricca)-
x_fin_cricca(1,indice_riferimento_cricca))+x_iniz_cricca(1,indice_riferimento
_cricca)*(y_fin_cricca(1,indice_riferimento_cricca)-
y_iniz_cricca(1,indice_riferimento_cricca));

    %Azzeramento contatori per ciclo for
    id_mat_cricca=0;

    x0_1=0;    %nodo partenza
    y0_1=0;

```

```

x0_2=0;          %nodo arrivo
y0_2=0;

distanza_cricca1=0; %distanza nodo partenza dalla cricca
distanza_cricca2=0; %distanza nodo arrivo dalla cricca

matr_aus2=[0,0,0,0]; %matrice ausiliaria

coeff_angolare_retta_elem=0; %coefficiente angolare della retta dell
elemento
%se la retta dell'elemento e parallela alla cricca non bisogna fare il
%confronto perche e sufficiente la condizione sulle x dell or, quella del
%prodotto delle distanze 0*0uguale0 e quindi il programma controlla le x
mat_elem_con_cricca=zeros(numero_elementi_senza_cricca,10); %matrice
ausiliaria per la creazione della matrice con gli elementi con cricca
%Ciclo while per la creazione della matrice elementi senza gli elementi
%appartenti alla cricca
for i=1:numero_elementi_senza_cricca
    i;

    elem_aus1=matrice_elem_cood(i,1); %numero elemento
matrice_elem_cood

    elem_aus2=matrice_elem_cood(i,2); %numero nodo partenza
    elem_aus3=matrice_elem_cood(i,3); %numero nodo arrivo

    elem_aus4=matrice_elem_cood(i,4); %x nodo partenza
    elem_aus5=matrice_elem_cood(i,5); %y nodo partenza

    elem_aus7=matrice_elem_cood(i,7); %x nodo arrivo
    elem_aus8=matrice_elem_cood(i,8); %y nodo arrivo

    elem_aus15=matrice_elem_cood(i,10); %lunghezza dell elemento

    %Coordinate temporanee nodo di partenza
    x0_1=elem_aus4; %nodo partenza
    y0_1=elem_aus5;

    %Coordinate temporanee nodo di arrivo
    x0_2=elem_aus7; %nodo arrivo
    y0_2=elem_aus8;

    %distanza nodo partenza dalla cricca
    distanza_cricca1=((a_c*x0_1+b_c*y0_1+c_c)/((a_c^2+b_c^2)^(0.5)));
    %distanza nodo arrivo dalla cricca
    distanza_cricca2=((a_c*x0_2+b_c*y0_2+c_c)/((a_c^2+b_c^2)^(0.5)));

    %Equazione implicita della retta dell elemento
    a_el=(y0_1-y0_2); % a b c retta implicita dell elemento
    b_el=(x0_2-x0_1);
    c_el=y0_1*(x0_1-x0_2)+x0_1*(y0_2-y0_1);

    %Calcolo coeff retta elemento
    coeff_angolare_retta_elem=(a_el)/(b_el);

    %controllo del punto di intersezione con la cricca

```

```

        if
            (((x0_1<=x_fin_cricca(1,indice_riferimento_cricca))&&(x0_2>=x_fin_cricca(1,indice_riferimento_cricca)))||((x0_1>=x_fin_cricca(1,indice_riferimento_cricca))&&(x0_2<=x_fin_cricca(1,indice_riferimento_cricca))))

                if (not(coeff_angolare_retta_elem==0))           %se la retta dell
elemento non è parallela alla cricca
                    Q=[a_c,b_c;a_el,b_el];
                    T=[-c_c;-c_el];
                    X=Q^(-1)*T;
                    %coordinate punto di intersezione fra retta elemento e retta
cricca
                    elem_aus10=X(1,1);           %Coordinata x punto intersezione
                    elem_aus11=X(2,1);           %Coordinata y punto intersezione
                end

            end

            %Condizione per poter scrivere l elemento nella matrice elementi con
            %cricca (la condizione deve essere vera)

            if
                (((distanza_cricca1)*(distanza_cricca2))>0)||((elem_aus4>x_fin_cricca(1,indice_riferimento_cricca))&&(elem_aus7>x_fin_cricca(1,indice_riferimento_cricca))||((elem_aus10>x_fin_cricca(1,indice_riferimento_cricca))))

                matr_aus2=[elem_aus1,elem_aus2,elem_aus3,elem_aus4,elem_aus5,elem_aus6,elem_aus7,elem_aus8,elem_aus9,elem_aus15];

                id_mat_cricca=id_mat_cricca+1;
                mat_elem_con_cricca1(id_mat_cricca,:)=matr_aus2;
            end

            elem_aus10=0;           %per uscire dalla condizione if nel ciclo
successivo

            end
            mat_elem_con_cricca=zeros(id_mat_cricca,10);
            mat_elem_con_cricca=mat_elem_con_cricca1(1:id_mat_cricca,1:10);
            matrice_elem_cood=mat_elem_con_cricca;
            numero_elementi_senza_cricca=id_mat_cricca;

        end
        fprintf('Fine creazione cricca o cricche');
        fprintf('\n');
    
```

Sezione per la creazione degli elset

Ad ogni elemento a seconda del nodo famiglia a cui è collegato è stata assegnata una sezione specifica che ne caratterizza la rigidità complessiva dell'elemento e quindi la sua influenza nei confronti degli altri *bond*. Ogni gruppo di *bond* che hanno la stessa sezione è stato chiamato elset.

```

%-----
%ELSET
%-----

fprintf('Inizio creazione file elset');
fprintf('\n');

epsilon=0;          %distanza reale ---dist confr
%Numero di elementi a cui calcolare il beta
numero_elementi_con_beta=id_mat_cricca;          %numero
elementi mat_elem_con_cricca
matrice_elem_distanze_fin=mat_elem_con_cricca;

%calcolo di beta
limite1=h1-passox/2;
limite2=h1+passox/2;
beta=0;
mat_elem_beta=zeros(numero_elementi_con_beta,2);

%ciclo for per creare i beta---valori sono numerici e approssimati
for riga=1:numero_elementi_con_beta

    epsilon=matrice_elem_distanze_fin(riga,10);

    if(abs(epsilon))<=limite1
        beta=1;
        mat_elem_beta_au=[matrice_elem_distanze_fin(riga,1),beta];

    elseif ((abs(epsilon))>limite1)&&((abs(epsilon))<=limite2)
        beta=(h1+0.5*passox-(abs(epsilon)))/(passox);
        mat_elem_beta_au=[matrice_elem_distanze_fin(riga,1),beta];
    else
        beta=0;
        mat_elem_beta_au=[matrice_elem_distanze_fin(riga,1),beta];
    end

    mat_elem_beta(riga,:)=mat_elem_beta_au;
end

%viene ordinata la matrice con numero dell elemento e beta calcolato a lui
associato
mat_elem_beta_ordinata = sortrows (mat_elem_beta,2); % Per ordinare la
matrice in base alla 2 colonna lasciando invariate le righe

%Ciclo per individuare il numero dei beta ed individuare i beta di
%riferimento
riga=0;
id_beta=0;
num_beta=0;

mat_elementi_beta_rif=zeros(numero_elementi_con_beta,2);
mat_beta_rif=zeros(numero_elementi_con_beta,1);

matrice_beta1=zeros(1,10000);          %matrice ausiliaria che contiene i
valori di riferimento di beta

%ciclo for per eliminare i valori simili di beta e metterli uguale a quello
di riferimento
for riga=1:numero_elementi_con_beta

```

```

elemento0=mat_elem_beta_ordinata(riga,1);      %numero dell elemento
elemento1=mat_elem_beta_ordinata(riga,2);      %beta calcolato

if ((elemento1>num_beta)&&(((elemento1)/(num_beta))>1.001))
    %si entra in questo if per fare la matrice dei beta i riferimento
    %in questo passaggio non si sa ancora quanti siano i beta di
    %riferimento, le condizioni per entrare devono essere che:
    %-il valore di beta calcolato sia maggiore del precedente beta di
    %riferimento
    %condizione logica end
    %-il valore di beta calcolato deve essere molto più grande del
    %valore beta di riferimento

    id_beta=id_beta+1; %numero di beta --- calcola il numero di beta
    num_beta=elemento1; %valore di beta --- viene cambiato il valore beta
di riferimento

    matrice_beta1(1,id_beta)=num_beta;

end %fine condizione per creare i beta di riferimento

%creazione della matrice con numero elemento e beta di riferimento
mat_aus_beta1=[elemento0,num_beta];

    mat_elementi_beta_rif(riga,:)=mat_aus_beta1; %matrice con gli elementi
ed i relativi beta di riferimento
    mat_beta_rif(riga,1)=mat_aus_beta1(1,2); %matrice con solo i numeri
beta ma nell'ordine di mat_elementi_beta_rif
end %dal beta minore al beta
maggiore

matrice_beta=zeros(1,id_beta);
matrice_beta=matrice_beta1(1,1:id_beta);

%-----

%numero effettivo
numero_beta=id_beta; %solo per scollegare le parti

numero_elset_tot=numero_beta; %numero totale complessivo di elset
contatore_elset=0; %variabile controllo numero di elset del ciclo
for
    contatore_elset_aus=0; %variabile controllo numero elset interna al
ciclo

    pos_iniz_elset=1; %posizione iniziale array elset
    pos_fin_elset=0; %posizione finale array elset

    array_elset1=zeros(numero_elementi_con_beta,1); %matrice zero con numero
massimo di elementi in un array

%creazione file con elset

for contatore_elset=1:numero_elset_tot

    contatore_elset_aus=contatore_elset_aus+1; %è uguale a
contatore_elset ma così non influisce sul ciclo for

```

```

posizioni_valori_elset=find((mat_beta_rif==matrice_beta(1,contatore_elset_au
))); %crea una matrice con le posizioni degli elementi

lunghezza_posizioni=length(posizioni_valori_elset); %numero di elementi

pos_fin_elset=pos_iniz_elset+lunghezza_posizioni-1; %i numeri degli
elementi sono in ordine crescente di elset

array_elset1=mat_elementi_beta_rif(pos_iniz_elset:pos_fin_elset,1);
%valori di un determinato elset

array_elset=zeros(lunghezza_posizioni,1); %creazione di un
array con gli elementi dell elset
array_elset=array_elset1(1:lunghezza_posizioni,1);

pos_iniz_elset=pos_fin_elset+1; %posizione iniziale del primo elemento
dell elset successivo

array_elset=array_elset'; %gli elementi vengono messi in riga
array_elset=sort(array_elset); %gli elementi vengono ordinati

%Parte creazione file elset
vettore=array_elset; %per scorporare le due procedure di
creazione e visualizzazione
array_elset=0;%azzeramento array elset per non sovrascrivere al file
vecchio

if vettore==0
    break;
end

lunghezza=length (vettore);
righe=lunghezza/12;
righe=fix(righe);

resto_righe=lunghezza-(righe*12);
righe_aus=righe*12;

vettore_aus=zeros(1,righe_aus);

for i=1:righe_aus
    vettore_aus(1,i)=vettore(1,i);
end

i=righe_aus+1;
id=0;
id1=0;

vettore_aus1=zeros(1,(lunghezza-i));

for id=i:1:lunghezza,
    id1=id1+1;
    vettore_aus1(1,id1)=vettore(1,id);

```

```

end

lung_vett_au1=id;

vettore_au=reshape(vettore_au,12,[]);
vettore_au=vettore_au';

%Parte relativa alla visualizzazione e salvataggio dati
%per aprire il file di testo

numeroelset=contatore_elset;

matrice_numeri_ascii=[0,0,0,0];

if numeroelset<=9999
    numerodecimale3=fix                (numeroelset/1000);
%prendo solo le migliaia
    numerodecimale2=fix((numeroelset-numerodecimale3*1000)/100);
%prendo solo le centinaia
    numerodecimale1=fix((numeroelset-numerodecimale3*1000-
numerodecimale2*100)/10);           %prendo solo le decine
    numerodecimale0=fix(numeroelset-numerodecimale3*1000-
numerodecimale2*100-numerodecimale1*10); %prendo solo le unita

matrice_numeri_decimali=[numerodecimale3;numerodecimale2;numerodecimale1;nume
rodecimale0];

    matrice_numeri_ascii=zeros(4,1);

    for riga=1:4 %Indice matrice con i numeri posizionali degli elset

        numerodecimale=matrice_numeri_decimali(riga,1);
        numeroaggiuntivo=48; %partenza della tabella ascii .... e lo
zero che e 48
        numero=numeroaggiuntivo+numerodecimale;
        numerobinario=dec2bin(numero);
        numeroascii=char(bin2dec(numerobinario));

        matrice_numeri_ascii(riga,1)=numeroascii;

    end

else
    numeroelset=0;
    fprintf('Il numero di elset è troppo elevato ed è stato posto uguale
a zero--risultati non attendibili');
end

%creazione file elset
elset='elset';
nome_file_da_inserire=horzcat(link_cartelle,link_nome_prova,elset);
percorsol=nome_file_da_inserire;

percorsol2=matrice_numeri_ascii(1,1);
percorsol3=matrice_numeri_ascii(2,1);
percorsol4=matrice_numeri_ascii(3,1);
percorsol5=matrice_numeri_ascii(4,1);
percorsol6='.inp';

```

```

percorso=horzcat(percorso1,percorso2,percorso3,percorso4,percorso5,percorso6)
;

ftxl=fopen(percorso, 'w');

for indicel=1:righe

    elem_aus1=vettore_aus(indicel,1);
    elem_aus2=vettore_aus(indicel,2);
    elem_aus3=vettore_aus(indicel,3);
    elem_aus4=vettore_aus(indicel,4);
    elem_aus5=vettore_aus(indicel,5);
    elem_aus6=vettore_aus(indicel,6);
    elem_aus7=vettore_aus(indicel,7);
    elem_aus8=vettore_aus(indicel,8);
    elem_aus9=vettore_aus(indicel,9);
    elem_aus10=vettore_aus(indicel,10);
    elem_aus11=vettore_aus(indicel,11);
    elem_aus12=vettore_aus(indicel,12);

fprintf(ftxl, '%8i,%8i,%8i,%8i,%8i,%8i,%8i,%8i,%8i,%8i,%8i,%8i\n',elem_aus1,el
em_aus2,elem_aus3,elem_aus4,elem_aus5,elem_aus6,elem_aus7,elem_aus8,elem_aus9
,elem_aus10,elem_aus11,elem_aus12);    %per creare il file
end

if (not(resto_righe==0))
    fprintf(ftxl, '%8i,',vettore_aus1);        %per creare il file
end
fclose(ftxl); % per chiudere il file e salvarlo

end

fprintf('Fine creazione file elset');
fprintf('\n');

```

Sezione per la creazione dei file

L'ultima parte del programma è dedicata alla creazione del *file main.inp* , del file output e del file delle caratteristiche geometriche.

```

%-----
%           FILE MAIN
%-----

fprintf('Inizio creazione file Main');
fprintf('\n');

nome_file_main=horzcat(link_cartelle,link_nome_prova,link_nome_file_main,link
_estensione_file_main);

%Apertura file MAIN
ftxl=fopen(nome_file_main, 'w');

```



```

fprintf(ftxl, '*HEADING');
fprintf(ftxl, '\n');
fprintf(ftxl, 'Peridynamic model with FEA using truss elements. ');
fprintf(ftxl, '\n');
fprintf(ftxl, '*PREPRINT, ECHO=YES, MODEL=NO, HISTORY=YES');
fprintf(ftxl, '\n');
fprintf(ftxl, '*****');
fprintf(ftxl, '\n');
fprintf(ftxl, '** Model definition **');
fprintf(ftxl, '\n');
fprintf(ftxl, '*****');
fprintf(ftxl, '\n');

%Inserimento file nodi
visualizz_parte_abaqus='*NODE, nset=all_nodes, input=';
nodedefinition='nodedefinition';
nome_file_da_inserire=horzcat(link_nome_prova,nodedefinition,link_estensione_
inp);
visulizz_riga=horzcat(visualizz_parte_abaqus,nome_file_da_inserire);
fprintf(ftxl,visulizz_riga);

fprintf(ftxl, '\n');

%Inserimento file elementi
visualizz_parte_abaqus='*ELEMENT, type=T2D2, elset=AllTrussElements, input=';
elementconnectivity='elementconnectivity';
nome_file_da_inserire=horzcat(link_nome_prova,elementconnectivity,link_estens
ione_inp);
visulizz_riga=horzcat(visualizz_parte_abaqus,nome_file_da_inserire);
fprintf(ftxl,visulizz_riga);

%Inserimento elset

fprintf(ftxl, '\n');

id_elset=0;

for i=1:numero_elset_tot

    id_elset=id_elset+1;

    fprintf(ftxl, '*ELSET, elset=Elements_Material_%i \n', id_elset);

    %Parte relativa alla visualizzazione e salvataggio dati
    %per aprire il file di testo

    numeroelset=id_elset;

    matrice_numeri_ascii=[0,0,0,0];

    if numeroelset<=9999
        numerodecimale3=fix((numeroelset-1)/1000);
    %prendo solo le migliaia
        numerodecimale2=fix((numeroelset-numerodecimale3*1000)/100);
    %prendo solo le centinaia
        numerodecimale1=fix((numeroelset-numerodecimale3*1000-
numerodecimale2*100)/10);
        %prendo solo le decine

```

```

        numerodecimale0=fix(numeroelset-numerodecimale3*1000-
numerodecimale2*100-numerodecimale1*10);    %prendo solo le unita

matrice_numeri_decimali=[numerodecimale3;numerodecimale2;numerodecimale1;nume
rodecimale0];

        matrice_numeri_ascii=zeros(4,1);

        for riga=1:4 %Indice matrice con i numeri posizionali degli elset

            numerodecimale=matrice_numeri_decimali(riga,1);
            numeroaggiuntivo=48;    %partenza della tabella ascii .... e lo
zero che e 48
            numero=numeroaggiuntivo+numerodecimale;
            numerobinario=dec2bin(numero);
            numeroascii=char(bin2dec(numerobinario));

            matrice_numeri_ascii(riga,1)=numeroascii;

        end

    else
        numeroelset=0;
        fprintf('Il numero di elset è troppo elevato ed è stato posto uguale
a zero--risultati non attendibili');
    end

    visualizz_parte_abaqus='*include, input=';
    elset='elset';
    nome_file_da_inserire=horzcat(link_nome_prova,elset);
    visulizz_riga=horzcat(visualizz_parte_abaqus,nome_file_da_inserire);

    percorsol=visulizz_riga;

    percorso2=matrice_numeri_ascii(1,1);
    percorso3=matrice_numeri_ascii(2,1);
    percorso4=matrice_numeri_ascii(3,1);
    percorso5=matrice_numeri_ascii(4,1);
    percorso6='.inp';

    percorso=horzcat(percorsol,percorso2,percorso3,percorso4,percorso5,percorso6)
;

    fprintf(ftxl,percorso);

    fprintf(ftxl,'\n');

end

%Inserimento file mass e valore massa nodi

visualizz_parte_abaqus='*ELEMENT, type=MASS, elset=mass_elements, input=';
elementimass='elementimass';
nome_file_da_inserire=horzcat(link_nome_prova,elementimass,link_estensione_in
p);
visulizz_riga=horzcat(visualizz_parte_abaqus,nome_file_da_inserire);
fprintf(ftxl,visulizz_riga);

fprintf(ftxl,'\n');

```

```

fprintf(ftx1, '*MASS, elset=mass_elements');
fprintf(ftx1, '\n');
fprintf(ftx1, ' %1.10e, ', massa_nodo);
fprintf(ftx1, '\n');
fprintf(ftx1, '**');
fprintf(ftx1, '\n');
fprintf(ftx1, '** Section definition');
fprintf(ftx1, '\n');
fprintf(ftx1, '**');
fprintf(ftx1, '\n');
id_elset=0;

for i=1:numero_elset_tot
    id_elset=id_elset+1;

    fprintf(ftx1, '*SOLID          SECTION,          elset=Elements_Material_%i,
material=material_1 ', id_elset);
    fprintf(ftx1, '\n');
    fprintf(ftx1, ' %1.10e, ', matrice_sezioni(1, id_elset));

    fprintf(ftx1, '\n');
end

%Inserimento valori materiale densità e plasticità dei bond

fprintf(ftx1, '**');
fprintf(ftx1, '\n');
fprintf(ftx1, '** Materials');
fprintf(ftx1, '\n');
fprintf(ftx1, '**');
fprintf(ftx1, '\n');
fprintf(ftx1, '*MATERIAL, name=material_1');
fprintf(ftx1, '\n');
fprintf(ftx1, '*ELASTIC');
fprintf(ftx1, '\n');
fprintf(ftx1, ' %1.8e, %1.5e', Ebond, Poisson);
fprintf(ftx1, '\n');
fprintf(ftx1, '*DENSITY');
fprintf(ftx1, '\n');
fprintf(ftx1, ' %1.8e, ', densita_bond);
fprintf(ftx1, '\n');
fprintf(ftx1, '*PLASTIC');
fprintf(ftx1, '\n');
fprintf(ftx1, ' %1.8e, 0.0', yield_stress);
fprintf(ftx1, '\n');
fprintf(ftx1, '*DAMAGE INITIATION, criterion=johnson cook');
fprintf(ftx1, '\n');
fprintf(ftx1, '0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0');
fprintf(ftx1, '\n');
fprintf(ftx1, '*DAMAGE EVOLUTION, type=energy');
fprintf(ftx1, '\n');
fprintf(ftx1, '0.0, ');
fprintf(ftx1, '\n');
fprintf(ftx1, '**-----');
fprintf(ftx1, '\n');

%Chiusura file MAIN
fclose(ftx1);

fprintf('Fine creazione file Main');
fprintf('\n');

```

```

fprintf('\n');
%-----
%Inizio creazione output_file
%-----

fprintf('Creare in Abaqus i file report con i seguenti
nomi:\nabaqus_status.rpt\nabaqus_displacement.rpt\n\n');

link_file_filename=link_nome_prova;

nome_file_output='output_filenames.txt';
filename=horzcat(link_cartelle,nome_file_output);
ftx3=fopen(filename,'w');
fprintf(ftx3,'*-----*\n');
fprintf(ftx3,'* The output file names for damage plotting\n');
fprintf(ftx3,'* must occur in the following order\n');
fprintf(ftx3,'*-----*\n');
fprintf(ftx3,'*\n');
fprintf(ftx3,'* Name of the file containing the elements id number and
connectivity:\n');
fprintf(ftx3,'* Nome del file contenente gli elementi:\n');

%Scrittura nome file 1 elenco elementi
nome_da_visualizzare=horzcat(link_file_filename,elementconnectivity,link_este
nsione_inp);

fprintf(ftx3,nome_da_visualizzare);

fprintf(ftx3,'\n');
fprintf(ftx3,'*\n');
fprintf(ftx3,'* Name of the file containing the element id and element status
(abaqus report file):\n');
fprintf(ftx3,'* Nome del file Status creato da Abaqus:\n');

%Scrittura nome file 2 abaqus status
fprintf(ftx3,'abaqus_status.rpt');

fprintf(ftx3,'\n');
fprintf(ftx3,'*\n');
fprintf(ftx3,'* Name of the file containing the nodes id number and
coordinates at the beginning of the simulation:\n');
fprintf(ftx3,'* Nome del file contenete i nodi:\n');

%Scrittura nome file 3 elenco nodi
nome_da_visualizzare=horzcat(link_file_filename,nodedefinition,link_estensio
ne_inp);

fprintf(ftx3,nome_da_visualizzare);

fprintf(ftx3,'\n');
fprintf(ftx3,'*\n');
fprintf(ftx3,'* Name of the file containing the nodes id number and the node
displacement at a specific instant (abaqus report file):\n');
fprintf(ftx3,'* Nome del file Displacement creato da Abaqus:\n');

%Scrittura nome file 4 abaqus displacement

```

```

fprintf(ftx3, 'abaqus_displacement.rpt');

fprintf(ftx3, '\n');
fprintf(ftx3, '*\n');
fprintf(ftx3, '* Name of the file containing the geometry input (used for dx,
dy and dz):\n');

%Scrittura nome file 5 dxdydz
dxdydz='dxdydz';
nome_da_visualizzare=horzcat(link_file_filename,dxdydz,link_estensione_inp);

fprintf(ftx3,nome_da_visualizzare);

nome_file_dxdydz=nome_da_visualizzare;

fprintf(ftx3, '\n');

fclose(ftx3);

%-----
%Fine creazione output_file
%-----

%-----
%Inizio creazione file dxdydz
%-----

link_file_dxdydz=horzcat(link_cartelle,nome_file_dxdydz);

ftx4=fopen(link_file_dxdydz, 'w');

fprintf(ftx4, ' Lengths of the volume edges:\n');
fprintf(ftx4, ' dx =%11.5f\n',passox);
fprintf(ftx4, ' dy =%11.5f\n',passoy);
fprintf(ftx4, ' dz =%11.5f\n',passot);
fclose(ftx4);

%-----
%Fine creazione file dxdydz
%-----

```

Programma in Matlab per l'elaborazione dei risultati di Abaqus e la creazione delle figure dei risultati

Per creare le figure dei risultati del capitolo 2 è stato creato un programma in Matlab che riceve in ingresso i seguenti file:

- file .inp che contiene l'elenco degli elementi di Abaqus
- file report status creato da Abaqus in cui ci sono tutti i *bond* con il loro relativo stato (ossia se sono integri o rotti)
- file .inp con la lista dei nodi
- file report displacement creato da Abaqus in cui vi sono gli spostamenti dei nodi
- file .inp con le caratteristiche geometriche della piastra

e a seconda del livello di danno che risulta dall'equazione 19 associa una scala di colori che va dal blu per tutti i *bond* integri a rosso con tutti i *bond* rotti.

```
clear all
clc
fprintf('Salvare tutti e 5 i file nella stessa cartella\n');
fprintf('\n');
frase_visualizzata= 'Inserire il percorso dove salvare i file mettere "\\\"
alla fine\n(Premere invio per il valore di default\nC:-Users-mauro-Google
Driv\10-CARTELLA MATTEO-Tesi-MATLAB-Salvataggi matlab file txt- \nPremere "1"
per salvare nella cartella in cui è salvato questo file matlab ma eliminare
manualmente l''1 dal file):\n';
str = input(frase_visualizzata,'s');
if isempty(str)
    str = 'C:\Users\mauro\Google Drive\CARTELLA MATTEO\Tesi\MATLAB\Salvataggi
matlab file txt\';
end

link_cartelle=str;          %percorso salvataggio file
disp(link_cartelle)
fprintf('\n')

%Inserimento valore scalefactor
frase_visualizzata= 'Inserire il valore scalefactor\n(Premere invio per il
valore di default 0):\n';
str=input(frase_visualizzata);
if isempty(str)
    str =0;
end
scalefactor=str;          %valore scalefactor
disp(scalefactor)
fprintf('\n');

%Inserimento valore scala colori
frase_visualizzata= 'Inserire il valore scala colori del danno\n(Premere
invio per il valore di default 90):\n';
str=input(frase_visualizzata);
if isempty(str)
    str =90;
```

```

end
scala_colori=str;          %valore scalefactor
disp(scala_colori)
fprintf('\n');

%-----
%Inizio programma
%-----

nome_file_output='output_filenames.txt';
filename=horzcat(link_cartelle,nome_file_output);

% Reading a file that contains the names of the files necessary as input
% for this script file. The appearance order of the file names in the file
% 'filename' must correspond to the order of appearance in this script
% file.
fid1=fopen(filename,'r');
input_files=textscan(fid1,'%s','CommentStyle','*');
fclose(fid1);
% Reading the file containing the elements id number and connectivity.
fprintf('Inizio operazioni con file n1\n');
input_file1=horzcat(link_cartelle,input_files{1}{1});
fid1=fopen(input_file1,'r');
all_elements=textscan(fid1,'%d %d %d','Delimiter',' ');
fclose(fid1);
all_elements=[all_elements{1}, all_elements{2}, all_elements{3}];
all_elements=sortrows(all_elements,1);
fprintf('Fine operazioni con file n1\n');
% Reading the file containing the elements id number and element status
% (abaqus report file). The lines that shouldn't be read by textscan as
% input CANNOT begin with a number.
fprintf('Inizio operazioni con file n2\n');
input_file2=horzcat(link_cartelle,input_files{1}{2});
fid1=fopen(input_file2,'r');
element_status=[];
while ~feof(fid1)
    PR=textscan(fid1,'%d                                     %d');
    %textscan reads attempts to read the file according to formatspec, when/if it
    %fails it stops
    fgetl(fid1);
    %fgetl skips the text lines which PR=textscan should not read, those text
    %lines cannot begin with a number. fegtl reads the entire line in the file
    %from the position where textscan stopped, and moves to the position at the
    %beginning of the next line from where PR=textscan will resume reading in the
    %next 'while' iteration
    el_stat_part=[PR{1},PR{2}];
    %Converting the cell PR into a matrix
    element_status=[element_status;el_stat_part];
    %Concatenating the values read in PR to the values read in the previous
    %iteration
end
fclose(fid1);
element_status=sortrows(element_status,1);
%Sorting the elements according to the first column
if size(element_status,1)~=size(all_elements,1)
    disp('ERROR: different size of arrays 'all_elements' and
'element_status')
end
% Creating an array 'unbroken_el' that contains only the unbroken elements.

```

```

num_unbroken_el=sum(element_status(:,2));
unbroken_el=zeros(num_unbroken_el,4);
for i=1:size(element_status,1)
    if all_elements(i,1)==element_status(i,1)
        if element_status(i,2)==1
            unbroken_el(i,1)=all_elements(i,1);
            unbroken_el(i,2)=all_elements(i,2);
            unbroken_el(i,3)=all_elements(i,3);
            unbroken_el(i,4)=element_status(i,2);
        end
    else
        disp('ERROR: the elements in row %i of arrays 'all_elements' and
',i)
        disp('          'element_status' are not equal')
    end
end

fprintf('Fine operazioni con file n2\n');
% Reading the file containing the nodes id number and coordinates.
fprintf('Inizio operazioni con file n3\n');
input_file3=horzcat(link_cartelle,input_files{1}{3});
fidl=fopen(input_file3,'r');
nodes=textscan(fidl,'%d %f %f %f','Delimiter',' ');
fclose(fidl);
nodes_id=[nodes{1}];
%Nodes id number
nodes_xyz_initial=[nodes{2},nodes{3},nodes{4}];
%Initial position coordinates of the nodes
% Initializing the matrix that contains the number of elements connected to
% each node.
node_elems_num=zeros(size(nodes_id,1),3);
%'node_elems_num' contains the node id in the first row. In the second row is
the number of elements connected to the node at the beginning of the
simulation. The third row contains the number of elements connected to the
node at the moment that we want to plot.
node_elems_num(:,1)=nodes_id(:,1);
%Findig the number of elements connected to each node at the beginning of
%the simulation.
for i=1:size(node_elems_num,1)
    node1=find(all_elements(:,2)==node_elems_num(i,1));
    node2=find(all_elements(:,3)==node_elems_num(i,1));
    node_n_of_el=numel(node1)+numel(node2);
    node_elems_num(i,2)=node_n_of_el;
end

% Findig the number of elements connected to each node at the instan we are
% analysing.
for i=1:size(node_elems_num,1)
    node1=find(unbroken_el(:,2)==node_elems_num(i,1));
    node2=find(unbroken_el(:,3)==node_elems_num(i,1));
    node_n_of_el=numel(node1)+numel(node2);
    node_elems_num(i,3)=node_n_of_el;
end
fprintf('Fine operazioni con file n3\n');
% Reading the displacement of the nodes at the instant we are interested.
% The file containing the displacements is generated with abaqus viewer.
fprintf('Inizio operazioni con file n4\n');
input_file4=horzcat(link_cartelle,input_files{1}{4});
fidl=fopen(input_file4,'r');
node_displacement=[];
while ~feof(fidl)
    part_n_disp=[];

```



```

part_n_disp=fscanf(fid1,'%d %g %g',[3 Inf]);
fgetl(fid1);
part_n_disp=part_n_disp';
node_displacement=[node_displacement;part_n_disp];
end
fclose(fid1);
node_displacement=sortrows(node_displacement,1);
if size(nodes_id,1)~=size(node_displacement,1)
    disp('ERROR: different size of arrays 'nodes' and
'node_displacement')
end
% Calculating the damage at each node
damage(:,1)=node_elems_num(:,1);
damage(:,2)=1-(node_elems_num(:,3)./node_elems_num(:,2));
% Node position contains the nodes id number and their current position.
% The displacement of the node can be scaled by a scale factor given as
% input.
node_position=zeros(size(nodes_id,1),4);
node_position(:,1)=nodes_id(:,1);
node_position(:,2)=nodes_xyz_initial(:,1)+scalefactor*node_displacement(:,2);
node_position(:,3)=nodes_xyz_initial(:,2)+scalefactor*node_displacement(:,3);
node_position(:,4)=nodes_xyz_initial(:,3);
fprintf('Fine operazioni con file n4\n');
% Reading the values of dx, dy and dz.
fprintf('Inizio operazioni con file n5\n');
input_file5=horzcat(link_cartelle,input_files{1}{5});
fid1=fopen(input_file5,'r');
dxdydz=textscan(fid1,'%*s %*s %f',3,'HeaderLines',1);
dxdydz={dxdydz{1}};
dx=dxdydz(1);
dy=dxdydz(2);
% Plotting the damage at each node
figure
face_matrix=[1 2 3 4];
for i=1:size(nodes_id,1)
    vertex_1=[node_position(i,2);node_position(i,3)]+[-dx/2;-dy/2];
    vertex_2=[node_position(i,2);node_position(i,3)]+[dx/2;-dy/2];
    vertex_3=[node_position(i,2);node_position(i,3)]+[dx/2;dy/2];
    vertex_4=[node_position(i,2);node_position(i,3)]+[-dx/2;dy/2];
    %index=round((scala_colori-1)*(damage(i,2))+1);
    index=round((scala_colori-1)*((damage(i,2)-
min(damage(:,2)))/(max(damage(:,2))-min(damage(:,2))))+1);

```

```

colore = [ 0          0      1.0000
          0      0.0625    1.0000
          0      0.1250    1.0000
          0      0.1875    1.0000
          0      0.2500    1.0000
          0      0.3125    1.0000
          0      0.3750    1.0000
          0      0.4375    1.0000
          0      0.5000    1.0000
          0      0.5625    1.0000
          0      0.6250    1.0000
          0      0.6875    1.0000
          0      0.7500    1.0000
          0      0.8125    1.0000
          0      0.8750    1.0000
          0      0.9375    1.0000
          0      1.0000    1.0000
0.0625    1.0000    0.9375

```

```

0.1250    1.0000    0.8750
0.1875    1.0000    0.8125
0.2500    1.0000    0.7500
0.3125    1.0000    0.6875
0.3750    1.0000    0.6250
0.4375    1.0000    0.5625
0.5000    1.0000    0.5000
0.5625    1.0000    0.4375
0.6250    1.0000    0.3750
0.6875    1.0000    0.3125
0.7500    1.0000    0.2500
0.8125    1.0000    0.1875
0.8750    1.0000    0.1250
0.9375    1.0000    0.0625
1.0000    1.0000         0
1.0000    0.9375         0
1.0000    0.8750         0
1.0000    0.8125         0
1.0000    0.7500         0
1.0000    0.6875         0
1.0000    0.6250         0
1.0000    0.5625         0
1.0000    0.5000         0
1.0000    0.4375         0
1.0000    0.3750         0
1.0000    0.3125         0
1.0000    0.2500         0
1.0000    0.1875         0
1.0000    0.1250         0
1.0000    0.0625         0
1.0000         0         0];
vertices_matrix=[vertex_1';vertex_2';vertex_3';vertex_4'];
if index <=scala_colori && index >=1
    color=ind2rgb(index,colore);
else
    color=ind2rgb(1,colore);
end
p=patch('Vertices',vertices_matrix,'Faces',face_matrix,'FaceColor',color);
set(p,'EdgeColor','none','FaceAlpha',0.9)

end
fprintf('Fine operazioni con file n5\n');
% Figure properties
max_x_plot=max(node_position(:,2));
min_x_plot=min(node_position(:,2));
max_y_plot=max(node_position(:,3));
min_y_plot=min(node_position(:,3));
%colorbar;
axis equal
axis([min_x_plot max_x_plot min_y_plot max_y_plot]);
%axis equal tight
fprintf('Attendere creazione figura\n');

```

Appendice B – Programmi in Visual Studio

Fra i materiali che Abaqus mette a disposizione per gli utenti non è presente un materiale che abbia le caratteristiche del materiale microelastico BMP. Il programma consente però di creare esternamente ad Abaqus un materiale con le caratteristiche desiderate dall'utente creando un file denominato Vumat. Sotto sono stati riportati i due programmi Vumat che sono stati creati per poter simulare l'esperimento di Kalthoff – Winkler e l'impatto.

Vumat per la simulazione dell'esperimento di Kalthoff - Winkler

Il programma qui descritto è una Vumat che consente di simulare un materiale perfettamente elastico sia in compressione che in trazione con la frattura ad una determinata soglia decisa dall'utente ma presente solo in trazione. I dati inseriti dall'utente sono il modulo elastico del *bond*, il modulo di Poisson e la lunghezza critica di rottura del *bond*.

```
subroutine vumat(  
c Read only (unmodifiable) variables-  
1 nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,  
2 stepTime, totalTime, dt, cmname, coordMp, charLength,  
3 props, density, strainInc, relSpinInc,  
4 tempOld, stretchOld, defgradOld, fieldOld,  
5 stressOld, stateOld, enerInternOld, enerInelasOld,  
6 tempNew, stretchNew, defgradNew, fieldNew,  
c write only (modifiable) variables -  
7 stressNew, stateNew, enerInternNew, enerInelasNew)  
c  
  include 'vaba_param.inc'  
c  
  dimension coordMp(nblock,*), charLength(nblock), props(nprops),  
1 density(nblock), strainInc(nblock,ndir+nshr),  
2 relSpinInc(nblock,nshr), tempOld(nblock),  
3 stretchOld(nblock,ndir+nshr),  
4 defgradOld(nblock,ndir+nshr+nshr),  
5 fieldOld(nblock,nfieldv), stressOld(nblock,ndir+nshr),  
6 stateOld(nblock,nstatev), enerInternOld(nblock),  
7 enerInelasOld(nblock), tempNew(nblock),  
8 stretchNew(nblock,ndir+nshr),  
9 defgradNew(nblock,ndir+nshr+nshr),  
1 fieldNew(nblock,nfieldv),  
2 stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),  
3 enerInternNew(nblock), enerInelasNew(nblock)  
c  
  character*80 cmname  
c  
  parameter (zero=0.,one=1.,two=2.)  
c  
c Modulo elastico  
  e=props(1)  
c Modulo di Poisson  
  v=props(2)  
c so  
  s0=props(3)  
c Calcolo delle deformazioni
```

```

Do i=1,nblock
  stressNew(i,1)=stressOld(i,1)+strainInc(i,1)*e
  stateNew(i,2)=stressNew(i,1)/e
end do
Do i=1,nblock
  if (stateNew(i,2).GE.s0) then
    stateNew(i,1)=0.
  else
    stateNew(i,1)=1.
  endif
end do
return
end

```

Vumat per la simulazione dell'impatto

Questa Vumat è stata utilizzata per le simulazioni del capitolo 4 riguardanti l'impatto fra un cilindro rigido e una piastra modellata con la teoria della peridinamica. I dati in ingresso alla Vumat, inseriti dall'utente, sono le caratteristiche del *bond*: il modulo elastico, il modulo di Poisson, la lunghezza critica, *cs* e *ds* che sono dati dalla formula per il calcolo delle forze di corto raggio e la distanza fra i nodi. Il problema dell'impatto riguarda, oltre che la parte di trazione del grafico tensione-deformazione, anche il campo della compressione del materiale e, a differenza del caso di Kalthoff-Winkler, l'accorciamento del *bond* è molto rilevante rispetto alla lunghezza iniziale quindi le forze di corto raggio non possono più essere trascurate. La seguente Vumat dà come risultato finale lo stress e lo stato del *bond*, che verrà poi analizzato da Abaqus, ma tenendo conto anche delle formule del § 1.5 Forze di corto raggio.

```

subroutine vumat(
c Read only (unmodifiable) variables-
1 nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
2 stepTime, totalTime, dt, cmname, coordMp, charLength,
3 props, density, strainInc, relSpinInc,
4 tempOld, stretchOld, defgradOld, fieldOld,
5 stressOld, stateOld, enerInternOld, enerInelasOld,
6 tempNew, stretchNew, defgradNew, fieldNew,
c write only (modifiable) variables -
7 stressNew, stateNew, enerInternNew, enerInelasNew)
c
  include 'vaba_param.inc'
C
  dimension coordMp(nblock,*), charLength(nblock), props(nprops),
1 density(nblock), strainInc(nblock,ndir+nshr),
2 relSpinInc(nblock,nshr), tempOld(nblock),
3 stretchOld(nblock,ndir+nshr),
4 defgradOld(nblock,ndir+nshr+nshr),
5 fieldOld(nblock,nfieldv), stressOld(nblock,ndir+nshr),
6 stateOld(nblock,nstatev), enerInternOld(nblock),
7 enerInelasOld(nblock), tempNew(nblock),
8 stretchNew(nblock,ndir+nshr),
9 defgradNew(nblock,ndir+nshr+nshr),
1 fieldNew(nblock,nfieldv),
2 stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
3 enerInternNew(nblock), enerInelasNew(nblock)
c
  character*80 cmname
c

```

```

    parameter (zero=0.,one=1.,two=2.)
c
c Modulo elastico
  e=props(1)
c Modulo di Poisson
  v=props(2)
c s0
  s0=props(3)
c ds
  ds=props(4)
c cs
  cs=props(5)
c dx
  dx=props(6)
c Calcolo delle deformazioni
  Do i=1,nblock
    stateNew(i,2)=stateOld(i,2)+strainInc(i,1)
  end do
  Do i=1,nblock
    if (charLength(i).GE.ds) then
      stressNew(i,1)=stateNew(i,2)*e
    else
      stressNew(i,1)=stateNew(i,2)*e
      + (cs*((charLength(i)/ds)-1.0))
    endif
  end do
  Do i=1,nblock
    if (stateNew(i,2).GE.s0) then
      stateNew(i,1)=0.
    else
      stateNew(i,1)=1.
    endif
  end do
  return
end

```


Bibliografia

- [1] Siling S.A., Askary E., *A meshfree method based on the peridynamic model of solid mechanics*, Elsevier, 83 (2005) 1526-1535, p. 1526-1535, 21/03/2005
- [2] Ha Y.D., Bobaru F., *Studies of dynamic crack propagation and crack branching with peridynamics*, Springer, DOI10.1007/s10704-010-9442-4, p. 229-244, 20/1/2010
- [3] Yu K., *Enhanced Integration methods for the peridynamic theory*, Kansas State University, 2011
- [4] Oterkus E., *Peridynamic theory for modelling three dimensional damage growth in metallic and composite structures*, University of Arizona, 2010
- [5] Dipasquale D., Zaccariotto M., Galvanetto U., *Crack propagation with adaptive grid refinement in 2D peridynamics*, Springer, DOI10.1007/s10704-014-9970-4, pg 1-22, 13/08/2014
- [6] Macek R. W., Siling S. A., *Peridynamics via finite element analysis*, Elsevier, (2007)1169-1178, 23/10/2007
- [7] Scudellaro L., *Nonlocal theories for the study of continuum: a finite elements method implementation*, Università degli studi di Padova, 2012
- [8] Siling S.A., *Peridynamic modeling of the Kalthoff-Winkler experiment*, 9232, pg1-8, 2/1/2002
- [9] Kalthoff J.F., Winkler S., *Failure mode transition at high rates of shear loading*, ISBN 3-88355-125-2, 1988
- [10] Madenci E., Erkan Oterkus E., *Peridynamic theory and its applications*, Springer, DOI10.1007/978-1-4614-8465-3, 2014
- [11] Lee J., W. Liu, Hong J. W., *Impact fracture analysis enhanced by contact of peridynamic and finite element formulations*, Elsevier, International journal of impact engineering, (2015)1-12, pg. 1-12, 2015
- [12] Tovo G., *Studio della teoria peridinamica per sistemi assialsimmetrici: sviluppo ed implementazione*, Università degli studi di Padova, 2012

Siti web:

<http://it.mathworks.com/> (Matlab - ultimo accesso 1/10/2015)

<http://www.3ds.com/it> (Abaqus - ultimo accesso 1/10/2015)