

Strumenti Open Source per l'ingegneria strutturale. Modellazione meccanica non lineare di edifici in muratura.

Michele BETTI, Gianni BARTOLI, Riccardo CORAZZI, Vladimir KOVAČEVIĆ

SOMMARIO

La memoria mostra l'impiego di codici Open Source, operanti in ambiente Linux, basati sulla tecnica degli elementi finiti affrontando nello specifico la modellazione di apparecchi murari. Dopo una breve descrizione dei codici adottati, e della legge non lineare di danno impiegata per la modellazione della risposta non lineare della muratura, vengono riportati e discussi due casi studio. I risultati ottenuti con il codice di calcolo Open Source sono confrontati con quelli ottenuti mediante modellazioni ad elementi finiti realizzate impiegando un codice di calcolo commerciale. I confronti mostrano come il codice Open Source, con un modello di danno i cui parametri siano opportunamente tarati su risultati di indagini sperimentali, sia adeguatamente capace di riprodurre il comportamento dei casi studio.

ABSTRACT

The paper shows the effectiveness of Open Source Finite Element Software through the discussion of the numerical modeling of masonry structures. After a short description of both the adopted code and the employed non-linear constitutive law, two case studies are analyzed. The results obtained with the Open Source code are compared with those obtained with a commercial one, and the comparison shows the ability and the effectiveness of the Open Source code (if the parameters needed to calibrate the non-linear constitutive law are correctly identified) to reproduce the non-linear structural behavior of the analyzed case studies.

PAROLE CHIAVE

Analisi non lineare, Codici FEM Open Source, Modello di danno scalare, Muratura.

INTRODUZIONE

Il 1991 vede ufficialmente la nascita di Linux, un kernel sviluppato da Linus Torvalds, allora giovane studente di informatica presso l'Università di Helsinki. Il nome in origine dato da Torvalds al suo codice era Freax, il termine Linux con il quale adesso è comunemente noto va invece attribuito ad Ari Lemmke, l'amministratore (allora assistente alla stessa università) che rese per primo disponibile Linux su Internet via FTP nel settembre del 1991 [1]. Da allora il kernel Linux è stato utilizzato come base per la realizzazione di molti progetti Open Source (sviluppando sistemi operativi con kernel Linux) e software sviluppato all'interno del progetto GNU. Grazie anche alle potenzialità di scambio e diffusione di

informazioni offerte dal *world wide web*, in questi ultimi anni la diffusione di sistemi operativi basati su kernel Linux (o GNU/Linux come forse sarebbe più proprio chiamarli) è cresciuta esponenzialmente, raggiungendo anche utenti tradizionalmente più orientati all'utilizzo di sistemi operativi standard. A questo ha, probabilmente, anche contribuito lo sviluppo, in ambiente GNU/Linux, di ambienti desktop come KDE e GNOME (interfacce grafiche molto simili a quelle di Microsoft Windows o di Mac OS) più vicine alle esigenze di utenti meno esperti.

La nascita del software libero propone un modello di sviluppo totalmente antitetico a quello del mondo commerciale. Raymond, in un suo saggio del 1997

[2], oggi generalmente considerato il manifesto del movimento Open Source, confronta e descrive i due stili di sviluppo coniando efficacemente le categorie di “modello cattedrale” e “modello bazaar”. Nel modello cattedrale, in voga in gran parte del mondo commerciale, il programma viene realizzato da un numero limitato di esperti che provvedono a scrivere il codice in quasi totale “isolamento”. Il progetto ha una suddivisione gerarchica molto stretta e ogni sviluppatore si preoccupa della sua piccola parte di codice. Ad esso si contrappone il modello bazaar, del mondo Linux, nel quale il codice sorgente della revisione in sviluppo è disponibile liberamente, gli utenti possono interagire con gli sviluppatori e modificare e integrare il codice. Lo sviluppo è decentralizzato e non esiste una rigida suddivisione dei compiti. Altrettanto efficacemente la differenza tra i due modelli può essere esemplificata dai termini *copyright* (diritti d’autore), per il mondo commerciale, e *copyleft* (permessi d’autore), per il mondo Open Source di cui la GNU General Public License creata da Richard Stallman rappresenta un primo esempio. La licenza GNU (GPL, *General Public License*), basata sui principi del *copyleft*, assicura quattro “libertà fondamentali” (così definite da Stallman): i) libertà di eseguire un programma per qualunque scopo; ii) libertà di studiare come funziona il programma e di adattarlo ai propri bisogni; iii) libertà di redistribuzione ad altri; iv) libertà di migliorare il programma e ridistribuire il miglioramento ad altri. Un programma è software libero se la licenza consente tutte queste libertà (ed ovviamente per poter avere queste libertà è necessario avere libero accesso al codice sorgente).

Nell’ambito del GNU Project (<http://www.gnu.org/>) sono stati, e vengono, scritti molti programmi Free Software. Free Software e Open Source usano lo stesso mezzo (libera circolazione del codice) per scopi diversi: la libera circolazione dell’informazione i primi e lo sviluppo di buon software i secondi. Per essere Open Source un software deve soddisfare i criteri: a) libera redistribuzione; b) disponibilità del codice sorgente; c) possibilità di creare opere derivate sotto la stessa licenza; d) integrità del codice sorgente dell’autore; e) nessuna discriminazione verso persone o gruppi; f) nessuna discriminazione verso campi di applicazione; g) i termini della licenza si applicano automaticamente a chi riceve il software; h) la licenza non deve essere specifica per un certo prodotto (insieme di programmi); i) la licenza non deve imporre restrizioni sul software distribuito assieme a software Open Source; l) la licenza deve essere indipendente dalla tecnologia. Sotto questa licenza, ed in parallelo ad ormai quanto disponibile per

molti altri settori, gli ultimi decenni hanno visto lo sviluppo e il rilascio di una serie di codici di calcolo (agli elementi finiti) di tipo Open Source anche nei tradizionali settori dell’ingegneria e della meccanica computazionale. Fra questi si può ricordare Elmer (<http://www.csc.fi>), OOFEM (<http://www.oofem.org>), OpenSees (<http://opensees.berkeley.edu/>) e Code Aster (<http://www.code-aster.org>), solo per citarne alcuni.

In questo quadro generale il lavoro intende mostrare alcune esperienze di modellazione numerica in campo non lineare svolte con uno di questi codici, il codice Code Aster, impiegato in combinazione con la piattaforma Salome-Meca. A tal fine nel seguito, dopo una breve descrizione dei due strumenti Open Source, si riportano risultati ottenuti nella modellazione di due casi studio di elementi in muratura. In particolare, dopo una descrizione, con maggiore dettaglio, del modello di danno adottato per il materiale muratura si svilupperà il caso della modellazione di una prova di compressione diagonale su un pannello in muratura, ed il caso di un’analisi di spinta su un edificio regolare in muratura. I risultati ottenuti con il codice Open Source verranno confrontati con i risultati di precedenti modellazioni agli elementi finiti realizzate con un codice commerciale.

Salome-Meca

Salome-Meca (<http://www.salome-platform.org/>) è una piattaforma, di uso generale, impiegata in ambiente Open Source come *pre e post processing* da diversi codici di modellazione numerica multifisica. La piattaforma è rilasciata sotto licenza GNU e dunque tanto il codice sorgente che gli esecutivi sono disponibili liberamente nella pagina ufficiale. La piattaforma dispone di un’interfaccia grafica (Fig. 1)

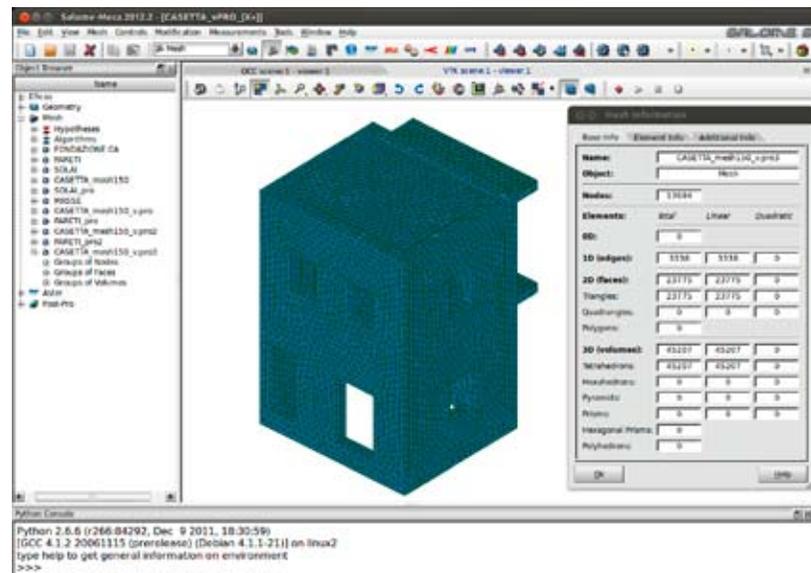


Fig.1. Screenshot dell’interfaccia grafica di Salome-Meca.

e di un'interfaccia testuale basata sul linguaggio python (<http://www.python.org/>). L'interfaccia grafica dispone di un CAD interno per la costruzione geometrica dei modelli (oltre a supportare file di importazione / esportazione con gli strumenti CAD tradizionali in diversi formati di interscambio), ed offre strumenti sufficientemente semplici per l'integrazione di nuove componenti (sistemi di calcolo numerico) organizzando inoltre in modo efficace l'accoppiamento di problemi fisici di diversa natura.

Code Aster

Code Aster (<http://www.code-aster.org>) è un software libero di simulazione basato sulla tecnica degli elementi finiti sviluppato principalmente dal dipartimento "Analyses Mécaniques et Acoustiques" del servizio R&D (Ricerca & Sviluppo) di EDF, *Électricité de France*. ASTER è l'acronimo di "Analyses des Structures et Thermo-mécanique pour des Études et des Recherches". La realizzazione del codice è iniziata nel 1989 per rispondere alle necessità interne di EDF, ed è stato reso disponibile sotto la licenza GNU GPL nel 2001. Nella versione in ambiente CAE-Linux (<http://www.caelinux.com>) Code Aster è direttamente integrato con la piattaforma Salome-Meca. Il codice è particolarmente robusto, contiene circa 1,500,000 linee di codice (scritte sia in Fortran sia in Python) ed è, come nella logica degli Open Source, in continuo sviluppo ed aggiornamento. Il codice, come detto, è stato originariamente implementato da EDF per rispondere alle proprie esigenze nel settore dell'industria nucleare e dunque risponde a richieste di affidabilità piuttosto severe. La parte di validazione è estremamente curata (con molti confronti con risultati sperimentali e *benchmarks* con altri codici), ed è stata svolta da organismi indipendenti da EDF. La documentazione consta di oltre 14,000 pagine (perlopiù in lingua francese).

Modello di danno isotropo lineare

Il codice di calcolo Code Aster ha una vasta libreria di modelli di comportamento non lineare. Nello specifico, proponendosi di affrontare la modellazione di elementi in muratura, il modello meccanico adottato per riprodurre il comportamento non lineare è il modello di danno scalare di Mazars [3] [4]. Esso è stato preferito ad altri modelli di leggi meccaniche per una serie di motivazioni. Da un lato, come si vede nel seguito, risulta sufficientemente semplice,

nel senso che il numero di parametri necessari alla sua definizione è sufficientemente ridotto ed inoltre i parametri richiesti sono di evidente significato fisico; dall'altro questo è un modello già implementato in altri codici di calcolo (ad esempio CAST3M, <http://www-cast3m.cea.fr/>) ed impiegato per l'analisi di apparecchi murari, e dunque la letteratura già presenta degli interessanti impieghi di questo modello per la modellazione di un materiale murario.

Il modello di Mazars [3], nella sua formulazione originale, è un modello omogeneo e isotropo a parametri scalari adoperato nella meccanica continua del danno, e proposto per l'analisi del calcestruzzo. Tipicamente la meccanica del danno si propone di modellare in modo continuo i processi di danneggiamento che intervengono in un mezzo a partire dal materiale integro fino ad arrivare alla formazione di fratture macroscopiche. Pensando, in modo specifico, alla trattazione proposta da Mazars si assuma un cilindro in condizioni di prova monoassiale avente sezione S e soggetto ad un'azione F . In condizioni integre la tensione nel cilindro risulta essere pari a F/S . Se il cilindro inizia a danneggiarsi solo una parte della sezione originaria, indicata con \hat{S} , contribuirà al trasferimento del carico. È possibile allora definire una variabile di danno D , una funzione continua nell'intervallo $DE[0, 1]$, tale che $\hat{S} = S \cdot (1-D)$. Così facendo il caso $D=0$ corrisponde al caso del materiale non danneggiato, il caso $D=1$ corrisponde al caso di materiale completamente danneggiato. Secondo questo schema la tensione netta nel provino cilindrico con modello di danno può essere espressa come segue:

$$\hat{\sigma} = \frac{F}{\hat{S}} = \frac{F}{S \cdot (1-D)} = \frac{\sigma}{(1-D)} \quad (1)$$

La (1), a stretto rigore (e per un materiale fragile o quasi-fragile), riporta la definizione di tensione netta nel caso di una sollecitazione di trazione. Nel caso di compressione l'area netta non è più $\hat{S} = S \cdot (1-D)$ in quanto, a causa della richiusura delle fessure prodotte dalla trazione, essa assume un valore compreso tra S e \hat{S} (per, appunto, la possibile richiusura delle fratture). Nel caso di compressione allora l'area netta può essere assunta come $\check{S} = S \cdot (1-t \cdot D)$ dove t è il parametro che tiene conto della richiusura delle fessure e dipende dal carico e, evidentemente, dal materiale. Così facendo in compressione può essere proposta un'espressione formalmente analoga alla (1):

$$\bar{\sigma} = \frac{F}{\bar{S}} = \frac{F}{S \cdot (1 - t \cdot D)} = \frac{\sigma}{(1 - t \cdot D)} \quad (2)$$

Implicitamente si assume che la parte reagente (cioè non danneggiata) della sezione sia ancora caratterizzata da un comportamento elastico lineare (a livello locale dunque), viceversa la parte danneggiata si assume non contribuisca al trasferimento dei carichi. Mantenendo per il provino dopo il danneggiamento la stessa definizione di deformazione formulata in assenza di danno si ha che il danno si manifesta attraverso una variazione di tensione. Questa corrisponde a quella nominale nel provino integro, ed a quella netta invece nel provino danneggiato. Così facendo la legge costitutiva del materiale, formalmente, assume la stessa espressione sia nel caso in assenza di danno che nel caso con danno. Sempre in riferimento al caso monoassiale la deformazione nel cilindro danneggiato è determinata dalla espressione seguente:

$$\hat{\varepsilon} = \frac{\hat{\sigma}}{E_0} = \frac{\sigma}{(1 - D) \cdot E_0} = \frac{\sigma}{\hat{E}} \quad (3)$$

dove E_0 indica il modulo di elasticità del materiale integro, mentre $\hat{E} = (1 - D) \cdot E_0$ rappresenta il cosiddetto modulo efficace. L'impostazione del problema secondo lo schema introdotto consente di continuare a trattare il comportamento del mezzo come un continuo, proponendo così un modello di danno isotropo con variabili di danno indipendenti in trazione e compressione le cui ipotesi di base possono essere così riassunte: a) le proprietà meccaniche del mezzo continuo si modificano dopo il raggiungimento di una soglia di danneggiamento; b) il mezzo ha un differente comportamento in trazione rispetto alla compressione; c) si sviluppano deformazioni permanenti.

A stretto rigore, stante la validità dell'ipotesi di isotropia, esso non è pienamente applicabile a costruzioni in muratura (che, data la particolare tessitura che le caratterizza, dovrebbero essere più efficacemente rappresentate da leggi di comportamento ortotropo, se non anisotrope). Va comunque osservato che tale assunzione, seppure non rispettosa dell'effettivo comportamento materiale, viene comunque ritenuta accettabile.

Nella sua forma generale, la legge di danno di Mazars [3] può essere espressa dalla seguente relazione:

$$\varepsilon_{ij} = \frac{1 + \nu_0}{(1 - D)E_0} \sigma_{ij} - \frac{\nu_0}{(1 - D)E_0} [\sigma_{kk} \delta_{ij}] \quad (4)$$

dove E_0 e ν_0 indicano, rispettivamente, il modulo di elasticità e il coefficiente di Poisson del materiale integro, D la variabile scalare di danno precedentemente introdotta, ε_{ij} e σ_{ij} indicano, rispettivamente, le componenti dei tensori di deformazione infinitesimo e di tensione di Cauchy, mentre δ_{ij} rappresenta il simbolo di Kronecker. La variabile di danno D è funzione dello stato di deformazione presente nel continuo, e l'evoluzione del danno è controllata attraverso una deformazione equivalente così definita:

$$\hat{\varepsilon} = \sqrt{\sum_{i=1}^3 ((\varepsilon_i)_+)^2} \quad (5)$$

dove ε_i indica la i -esima componente principale di deformazione e $(...)_+$ sono le parentesi di Macaulay:

$$(\varepsilon_i)_+ = \frac{|\varepsilon_i| + \varepsilon_i}{2} \quad (6)$$

La funzione di attivazione del danno viene definita come segue:

$$f(\varepsilon, D) = \varepsilon - \kappa \quad (7)$$

dove κ rappresenta la soglia d'innescio del danno (una variabile collegata alla storia di danneggiamento). Il valore iniziale di tale soglia (di prima insorgenza del danno) può essere messo in relazione con la tensione di picco a trazione del materiale, indicata con f_t , tramite la seguente: $\kappa_0 = \varepsilon_{D0} = f_t / E_0$.

Il danno in sostanza si attiva quando la deformazione equivalente raggiunge la soglia κ .

In queste condizioni $f(\varepsilon, D) = \varepsilon - \kappa = 0$ quindi $D = D(\kappa)$ e $\kappa = \varepsilon$.

Dopo l'attivazione del primo danno, con il crescere dei carichi esterni la soglia di attivazione si muove assumendo κ il valore raggiunto dalla deformazione equivalente.

Il modello di danno di Mazars, come visto, ha un differente comportamento in trazione (t) rispetto alla compressione (c), e questo è descritto dalla variabile di danno che viene espressa nella seguente forma:

$$D = \alpha_c D_c + \alpha_t D_t \quad (8)$$

dove α_c e α_t rappresentano dei pesi, definiti in funzio-

ne delle componenti speciali di deformazione, e sono definiti dalla seguente:

$$\alpha_c = \sum_{i=1}^a \left(\frac{(\varepsilon_i^c)(\varepsilon_i)_+}{\hat{\varepsilon}^2} \right)^\beta$$

$$\alpha_t = \sum_{i=1}^a \left(\frac{(\varepsilon_i^t)(\varepsilon_i)_+}{\hat{\varepsilon}^2} \right)^\beta$$
(9)

Il termine β presente nella (9) rappresenta un coefficiente di accoppiamento del danno. I due termini D_c e D_t sono invece le variabili di danno (rispettivamente a compressione ed a trazione) che, in forma integrale, assumono la seguente espressione:

$$D_c = 1 - \frac{\kappa_0(1 - A_c)}{\kappa} - \frac{A_c}{e^{[B_c(\kappa - \kappa_0)]}}$$

$$D_t = 1 - \frac{\kappa_0(1 - A_t)}{\kappa} - \frac{A_t}{e^{[B_t(\kappa - \kappa_0)]}}$$
(10)

con A_c, A_t, B_c e B_t dei parametri di definizione (della forma) delle curve. La (8) è la formulazione originariamente proposta da Mazars [3]; nel codice di calcolo Code Aster essa assume una forma lievemente più semplice [4] che è la seguente:

$$D = (1 - \alpha_t)^\beta D_c + \alpha_t^\beta D_t$$
(11)

dove il significato dei simboli è lo stesso riportato in precedenza.

È opportuno precisare che nel codice di calcolo Code Aster il modello di danno appena descritto è presente in due versioni, una locale (per la quale la tensione in un punto dipende solo dalla deformazione nello stesso punto) ed una non locale (la tensione in un punto dipende oltre che dalla deformazione di quel punto, da una deformazione media definita sull'intorno del punto stesso). Nel caso della versione locale, in termini assolutamente generali, si ha che la soluzione ottenuta dipende dalla discretizzazione in elementi finiti e occorre dunque preliminarmente effettuare delle attente analisi parametriche al fine di stimare questa dipendenza. Nel caso non locale (espresso utilizzando un tensore delle deformazioni non locale) questa dipendenza viene eliminata introducendo la dipendenza tra la tensione in un punto e la deformazione media su un volume rappresentativo

del materiale centrato nel punto stesso.

Il modello di Mazars ha il particolare pregio di essere di facile implementazione (anche i parametri necessari alla sua definizione nel seguito sono di facile determinazione sperimentale) e sufficientemente robusto sotto il profilo computazionale. La definizione completa del modello meccanico di Mazars necessita di 8 parametri, elencati in Tab. 1 i valori numerici riportati in tabella si riferiscono ad un materiale come il calcestruzzo).

Tabella 1. Parametri della legge costitutiva di Mazars

Simbolo	Significato	Calcestruzzo
E_0 [N/mm ²]	modulo di Young (materiale integro)	30 - 40 x 10 ³
ν_0 [-]	coefficiente di Poisson (materiale integro)	0.2
κ_0 [-]	soglia di innesco del danno	1 x 10 ⁻⁴
A_c [-]	coefficiente di forma (asintoto a compressione)	0
B_c [-]	coefficiente di forma (picco a compressione)	1 - 2 x 10 ³
A_t [-]	coefficiente di forma (asintoto a trazione)	0
B_t [-]	coefficiente di forma (picco a trazione)	1 - 5 x 10 ³
β [-]	coefficiente di accoppiamento del danno a compressione e trazione	1.00 - 1.05

Di questi i primi tre sono ricavabili, nel caso tipico del calcestruzzo ad esempio, tramite una semplice prova monoassiale. I successivi quattro (i coefficienti di forma A e B) sono dei parametri necessari a definire, rispettivamente, l'asintoto e il picco del ramo di interesse della curva (trazione o compressione). Il parametro β , il coefficiente di accoppiamento del danno, ha un'importanza piuttosto rilevante nella risposta materiale e regola in sostanza l'accoppiamento del danno a trazione e a compressione. A titolo di esempio (e per i valori che tipicamente si può assegnare ad un calcestruzzo) le figure seguenti illustrano la variabilità delle curve tensione-deformazione (esprese per il caso monoassiale, e concentrando l'attenzione sul ramo in trazione) assumendo diversi valori ai coefficienti di forma. La Fig. 2 mostra la dipendenza del ramo a trazione nella fase post-picco dal parametro A_t ; la Fig. 3 mostra sempre la dipendenza del ramo a trazione nella fase post-picco rispetto però al pa-

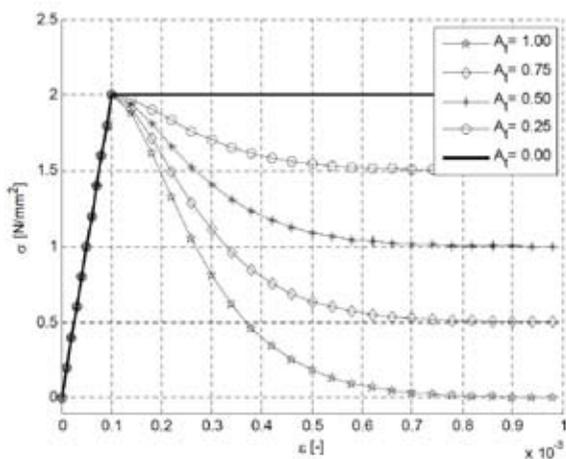


Fig. 2. Dipendenza del ramo a trazione della legge di Mazars (diagramma monoassiale σ - ε) dal parametro A_t (fissati $E_0=2 \cdot 10^4$ N/mm², $\nu_0=0.2$, $B=10^4$).

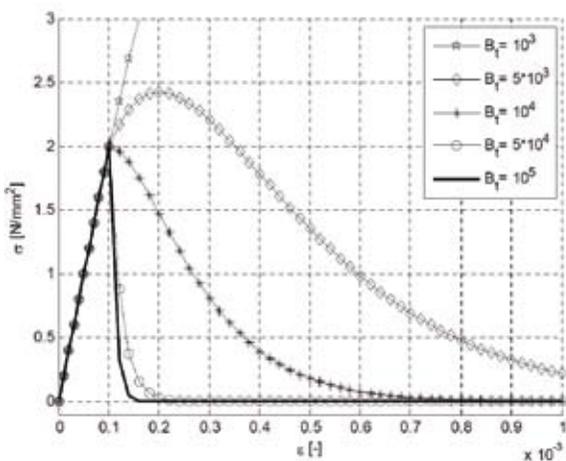


Fig. 3. Dipendenza del ramo a trazione della legge di Mazars (diagramma monoassiale σ - ε) dal parametro B_t (fissati $E_0=2 \cdot 10^4$ N/mm², $\nu_0=0.2$, $A=1$).

Casi di studio

I due casi studio analizzati prendono spunto da precedenti attività svolte nell'ambito del progetto TREMA [5] (acronimo per Tecnologie per la Riduzione degli Effetti sismici sui Manufatti Architettonici in muratura e cemento armato). Il primo caso corrisponde alla modellazione di una prova di compressione diagonale su un pannello in muratura, mentre il secondo corrisponde alla modellazione di un edificio regolare in muratura sviluppando delle analisi statiche non lineari di *pushover*. Essi risultano interessanti in quanto consentono un confronto con i risultati di precedenti modellazioni agli elementi finiti (realizzate con un codice commerciale) ([6] [7]).

Modellazione di una prova di compressione diagonale

Il primo caso studio analizzato è relativo alla modella-

zione di una prova di compressione diagonale effettuata su un muretto quadrato di lato 90 cm e spessore 25 cm realizzato in pietrame caotico di tufo e malta bastarda. Lo schema della prova, riportato in Fig. 4, ha previsto prima un'applicazione di un carico di compressione q per una lunghezza di circa 42 cm su due lati opposti dei pannelli (pari a 0.10 N/mm²). Successivamente è stato applicato un carico diagonale P_d , fino al raggiungimento del collasso, con una procedura sperimentale in controllo di spostamento. Il caso studio analizzato è interessante in quanto è caratterizzato da un comportamento fortemente non lineare (si analizza il pannello fino al collasso) e rappresenta quindi un efficace banco di prova del codice. I risultati numerici ottenuti con il codice Open Source sono confrontati con risultati sperimentali, e con quelli ottenuti da una precedente modellazione numerica eseguita con il co-

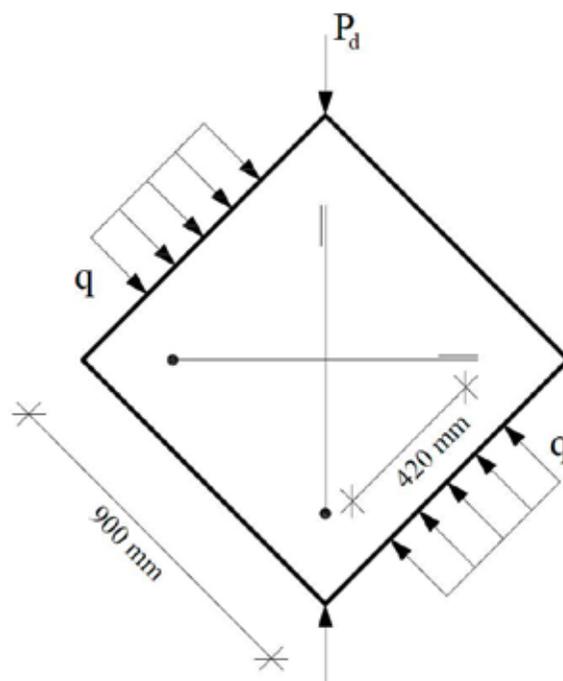


Fig. 4. Schema della prova di compressione diagonale.

La modellazione con il codice di calcolo ANSYS (<http://www.ansys.com/>) è stata effettuata utilizzando elementi solidi isoparametrici ad otto nodi (con otto punti di integrazione di Gauss). L'elemento finito utilizzato prevede la possibilità di adottare un modello anelastico a fessurazione diffusa. Il modello costitutivo (oltre ai parametri elastici, modulo di Young E_w e coefficiente di Poisson ν) richiede la definizione dei domini di plasticità di Drucker-Prager e di rottura di Willam-Warnke [6]. I parametri elastici identificati sono i seguenti: $E_w = 700$ N/mm² e $\nu = 0.25$, mentre i parametri necessari alla definizione dei domini di plasticità e di rottura sono riportati in Tab. 2.

Tabella 2. Parametri del modello non lineare di ANSYS: f_{wc} (resistenza a compressione); f_{wt} (resistenza a trazione); β_c (coefficiente di trasmissione del taglio per fessure richiuse); β_t (coefficiente di trasmissione del taglio per fessure aperte); c (coesione); δ (dilatanza); φ (angolo di attrito).

Parametro	Valore	Parametro	Valore
c	0.07 N/mm ²	f_{wc}	2.50 N/mm ²
δ	20°	f_{wt}	0.065 N/mm ²
φ	40°	β_c	0.75
		β_t	0.15

La Fig. 5 riporta una vista del modello numerico.

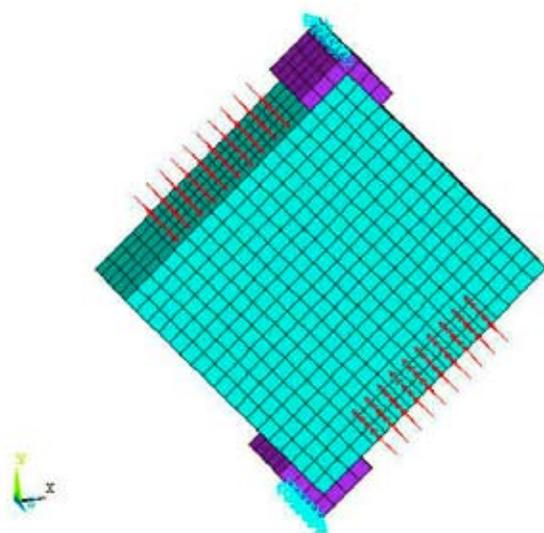


Fig. 5. Modello agli elementi finiti (ANSYS).

Il comportamento meccanico non lineare della muratura, riprodotto combinando il criterio di plasticizzazione di Drucker-Prager con il dominio di fessurazione di Willam-Warnke, corrisponde, nel campo delle trazioni ed in regime di sforzo biasiale trazione-compressione con compressioni medio-basse, ad un materiale con comportamento elasto-fragile, mentre in regime di sforzo biasiale trazione-compressione con compressioni elevate, ed in regime di compressione biassiale, ad un comportamento elasto-plastico. Il materiale è dunque modellato come un mezzo isotropo capace di deformarsi plasticamente, di fessurarsi per trazione e schiacciarsi per compressione.

La modellazione con il codice Open Source è stata eseguita operando preliminarmente la ricostruzione della geometria del pannello direttamente tramite il CAD integrato nella piattaforma Salomè-Meca. Oltre al pannello sono state riprodotte le scarpe di carico (per la successiva riproduzione delle condizioni di

vincolo sperimentali) e le piastre laterali (mediante le quali è stata applicata la precompressione iniziale).

La Fig. 6a mostra la ricostruzione geometrica. La successiva discretizzazione in elementi finiti è stata effettuata adottando l'algoritmo netgen (un generatore automatico di discretizzazioni tetraedriche solide sviluppato alla Johannes Kepler University di Linz e liberamente disponibile secondo licenza GNU, <http://www.hpem.jku.at/netgen/>). Il modello così realizzato contiene 13,053 elementi solidi e 59,199 nodi. La discretizzazione adottata è riportata in Fig. 6b. Avendo adottato un modello locale di danno, dei test parametrici preliminari sono stati svolti per verificare la dipendenza della soluzione dalla discretizzazione verificando così che la discretizzazione adottata conduca a risultati stabili.

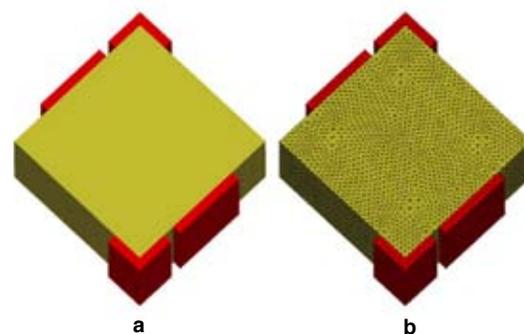


Fig. 6. Modello agli elementi finiti (Code Aster): a) rappresentazione geometrica; b) discretizzazione in elementi finiti.

Il modello di comportamento meccanico impiegato è quello con danno scalare di Mazars descritto nel paragrafo precedente, nel quale i parametri sono stati stimati mediante taratura della risposta numerica sui risultati sperimentali della prova. Sono state inoltre condotte delle analisi di sensibilità della risposta numerica al variare dei parametri di resistenza in trazione ed in compressione del materiale (entro un certo intervallo).

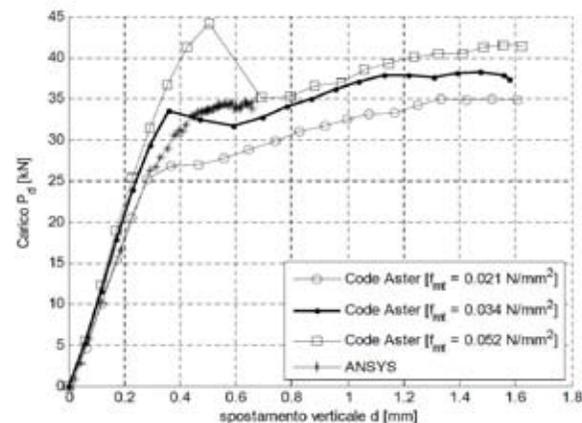


Fig. 7. Variabilità delle curve "carico applicato - spostamento della diagonale compressa" del pannello al variare della resistenza a trazione del materiale (assumendo una resistenza a compressione $f_{mc} = -1.821$ MPa).

La Fig. 7 mostra la variabilità della risposta (curva “carico applicato – spostamento della diagonale compressa del pannello”) in funzione della resistenza a trazione tenendo ferma la resistenza a compressione. La Fig. 8 mostra invece tale variabilità in funzione della resistenza a compressione (tenendo ferma la

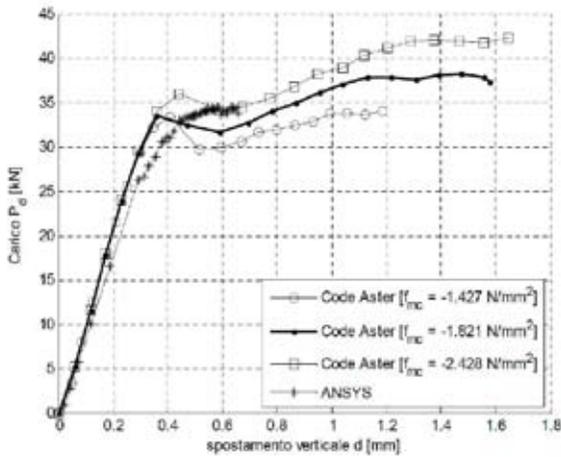


Fig. 8. Variabilità delle curve “carico applicato – spostamento della diagonale compressa” del pannello al variare della resistenza a compressione del materiale (assumendo una resistenza a trazione $f_{mt} = 0.034$ Mpa).

Come è ragionevole attendersi, anche in funzione del tipo di comportamento fisico modellato, il parametro che maggiormente influenza la risposta del modello è la resistenza a trazione. I parametri identificati, al termine del processo di taratura, del modello di danno scalare sono riportati in Tab. 3 e la Fig. 9 riporta la forma del legame costitutivo nel caso monoassiale. La Fig. 10 riporta il confronto (in termini sempre di carico diagonale in funzione dello spostamento verticale lungo la diagonale compressa) fra la curva sperimentale, la curva numerica ottenuta con la modellazione con ANSYS e la curva ricavata con la modellazione con Code Aster. Si vede come entrambe le modellazioni consentano di riprodurre in modo sufficientemente accurato sia il carico diagonale massimo (valore numerico 34.6 kN) che la pendenza “secante” del primo tratto della prova, prima della fessurazione.

Il modello numerico realizzato con ANSYS, una volta raggiunto il carico massimo, non riesce, per motivi di convergenza, a seguire il ramo softening evidenziato nella sperimentazione.

La Fig. 11 mostra il quadro fessurativo a collasso ottenuto con ANSYS, che è in accordo con la rottura per fessurazione diagonale. La Fig. 12 riporta l’evoluzione del danno ottenuta con Code Aster nei diversi passi di carico. La configurazione di danno al termine dell’analisi è in buon accordo con il quadro fessurativo ottenuto con il codice ANSYS e con la prova sperimentale.

Tabella 3. Parametri che regolano la legge costitutiva di Mazars per la muratura analizzata.

Simbolo	Significato	Pannello murario
E_0 [N/mm ²]	modulo di Young (materiale integro)	700
ν_0 [-]	coefficiente di Poisson (materiale integro)	0.1
κ_0 [-]	soglia di innesco del danno	10^{-5}
A_c [-]	coefficiente di forma (asintoto a compressione)	1
B_c [-]	coefficiente di forma (picco a compressione)	1000
A_t [-]	coefficiente di forma (asintoto a trazione)	1
B_t [-]	coefficiente di forma (picco a trazione)	7500
β [-]	coefficiente di accoppiamento del danno a compressione e trazione	1.02

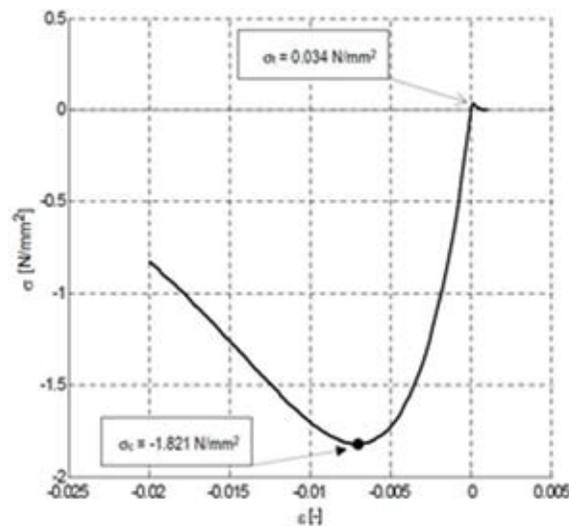


Fig. 9. Legame costitutivo (monoassiale) del modello di Mazars.

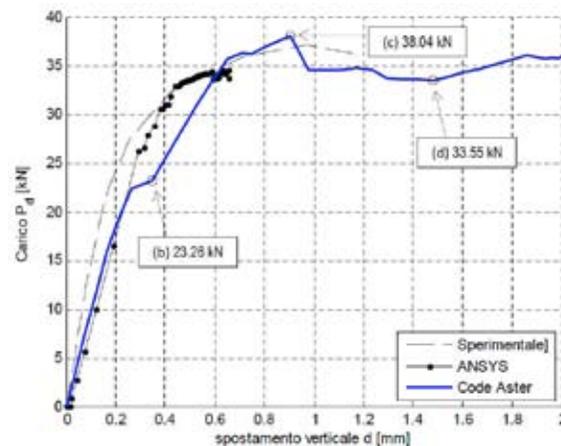


Fig. 10. Confronti curve di “carico-spostamento della diagonale compressa”.

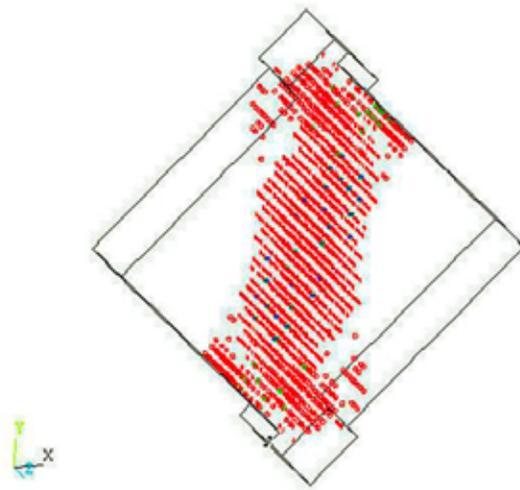


Fig. 11. Quadro fessurativo a collasso (ANSYS).

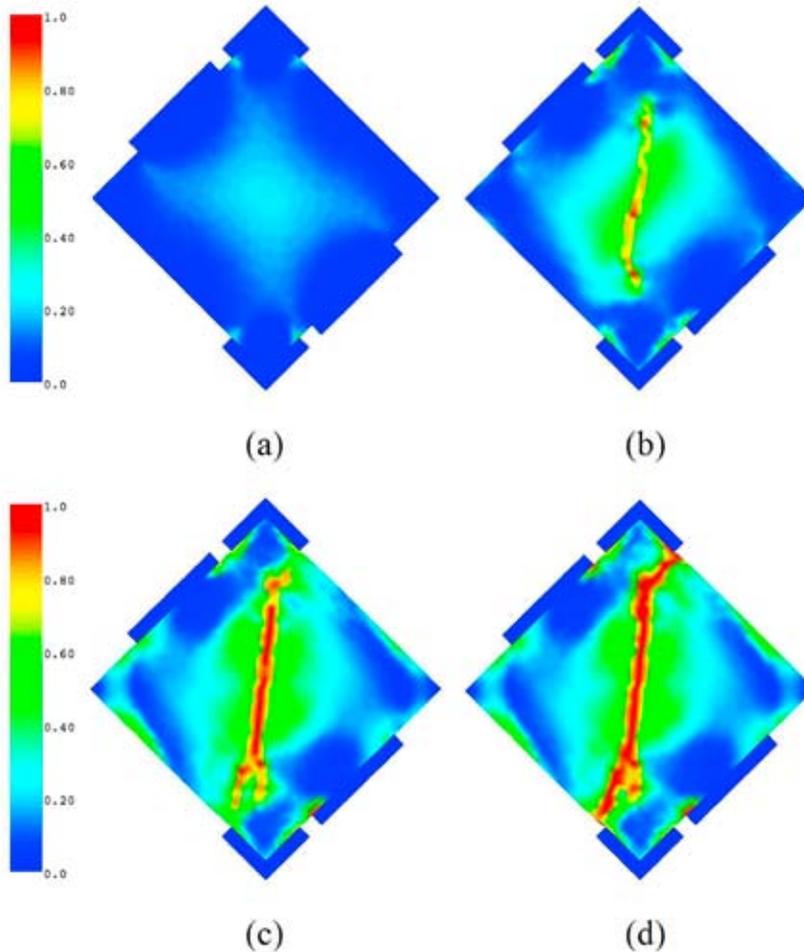


Fig. 12. Estensione del danno in alcune fasi dell'analisi: (a) al termine della fase di precompressione laterale; (b) al verificarsi del primo danneggiamento, Carico = 23.26kN; (c) in corrispondenza del picco di resistenza a trazione, Carico = 38.04kN; (c) in prossimità del termine dell'analisi, Carico = 33.55kN (in blu D= 0, materiale integro; in rosso D=1, materiale danneggiato).

Al fine di mostrare la dipendenza dei risultati (estensione del danneggiamento) ottenuti con il codice Open Source dal parametro β (il coefficiente di accoppiamento del danno a trazione e compressione) sono state svolte ulteriori analisi dove, rispetto al valore identificato ($\beta = 1.04$), si sono assunti anche i valori $\beta = 1.00$ (prova M00) e $\beta = 1.07$ (prova M02).

La Fig. 13 mostra il quadro di danneggiamento che si ottiene al termine delle analisi adottando un valore di β pari a 1.00 (Fig. 13a) e pari a 1.07 (Fig. 13b). Si nota che tale coefficiente riveste un ruolo determinante, e discriminante, nei risultati dell'analisi. Modeste variazioni percentuali sul valore di tale parametro si ripercuotono con variazioni sensibili sia sull'estensione del danno (giustificabili come un migliore/peggiore comportamento a taglio del materiale, dovute all'effetto d'attrito/ingranamento) sia sulla determinazione del valore di collasso e della duttilità (Fig. 14).

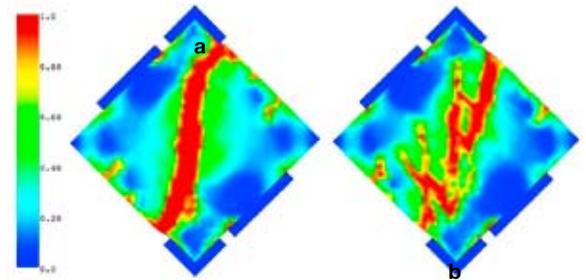


Fig. 13. Quadro di danneggiamento finale ottenuto adottando un valore di β pari a 1.00 (a) e pari a 1.07 (b) (in blu D= 0, materiale integro; in rosso D=1, materiale danneggiato).

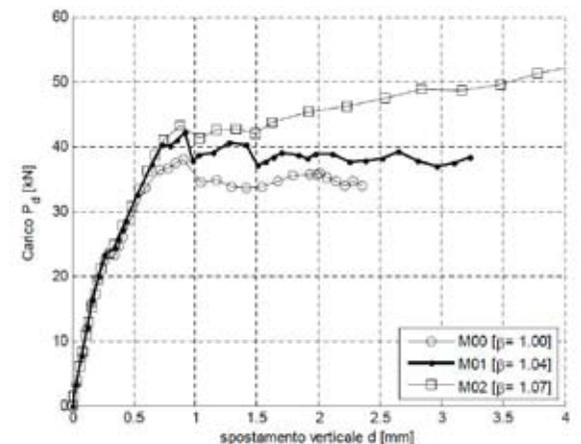


Fig. 14. Curve carico-spostamento in funzione parametro β (si nota l'aumento di duttilità e di resistenza del provino all'aumentare di β).

Modellazione di edificio regolare in muratura

Il secondo caso analizzato è il prototipo di un edificio in muratura (pietrame di tufo e malta bastarda) di due piani testato su tavola vibrante presso il centro di ricerca ENEA Casaccia di Roma (Fig. 15). Il prototipo, realizzato in scala 1:1.5, ha dimensioni in pianta di 3.5 m×3 m ed altezze di interpiano (misurate all'estradosso del solaio) di 2.2 m. Esso è fissato alla base su un cordolo in c.a. del peso complessivo di 2.88 t, mentre il peso totale della muratura risulta di circa 17.3 t. Su ciascun impalcato, realizzato con travi e tavolati in legno, è disposta una massa aggiuntiva di 2.5 t ed il peso totale di ciascun edificio, comprensivo dei solai e delle masse aggiunte, risulta pertanto pari a 25.6 t.

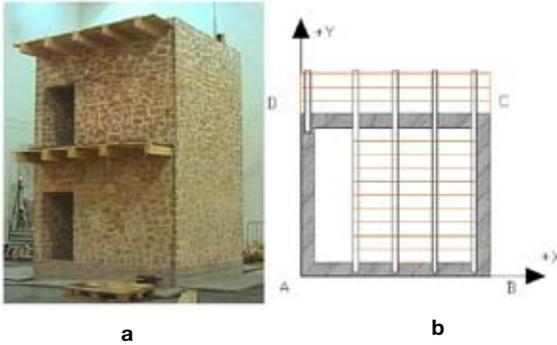


Fig. 15. (a) Vista del prototipo in muratura del progetto TREMA; (b) Pianta tipo.

Il modello realizzato con il codice ANSYS è stato costruito adottando elementi finiti tridimensionali *Solid65* (gli stessi adottati per la modellazione del caso studio precedente) sia per la modellazione delle pareti in muratura, sia per modellare la base in c.a.; le travi in legno di sostegno degli assiti dei due piani sono state modellate mediante gli elementi monodimensionali *Beam44*, mentre per il tavolato dell'assito si è ricorso agli elementi piani-*Shell63*. Ulteriori elementi *Mass21* sono stati utilizzati per le masse aggiunte dei solai. Globalmente il modello comprende 7,234 elementi ed 11,081 nodi. I nodi alla base della struttura sono stati vincolati, ed i gradi di libertà totali sono 30,933. Per i pesi specifici dei materiali si sono assunti i valori $\gamma_w = 14.18 \text{ kN/m}^3$ per la muratura di tufo e $\gamma_l = 5.88 \text{ kN/m}^3$ per il legno dei solai. La Fig. 16 riporta una vista tridimensionale del modello.

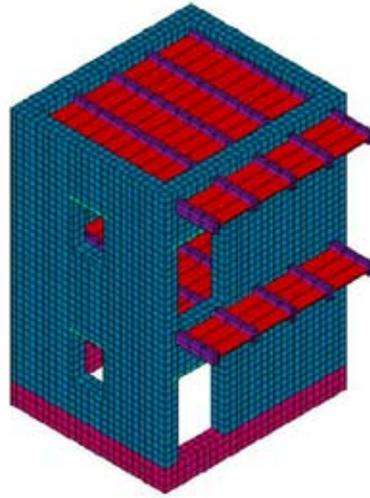


Fig. 16. Discretizzazione in elementi finiti del modello ANSYS.

Nel caso del codice di calcolo Code Aster per la discretizzazione agli elementi finiti si è adottato ancora l'algoritmo netgen, basato su elementi tridimensionali tetraedrici a 4 nodi, con lato di lunghezza di 150 mm. La maglia così caratterizzata consta di 13,694 nodi, 3,338 elementi 1D (*edges*), 23,775 elementi 2D (*faces*) e 45,207 elementi 3D (*volumes*). La Fig. 17a mostra la ricostruzione a volumi del prototipo, la Fig. 17b invece riporta la discretizzazione adottata.

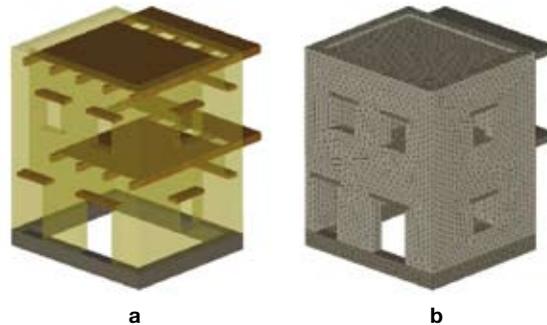


Fig. 17. Modello agli elementi finiti (Code Aster): a) rappresentazione geometrica; b) discretizzazione in elementi finiti.

Per ciascuno dei due codici i modelli meccanici ed i valori numerici dei rispettivi parametri sono quelli descritti in riferimento al precedente caso di studio. Il prototipo, presso il centro di ricerca CNR - ENEA di Casaccia, è stato sottoposto ad una serie di prove dinamiche su tavola vibrante applicando alla tavola un'azione sismica progressivamente crescente fino al raggiungimento del collasso. Nel seguito l'analisi a collasso verrà

eseguita tramite delle analisi *pushover* applicando ai due modelli delle azioni orizzontali crescenti in modo monotono (nelle due direzioni principali in pianta dell'edificio) con intensità proporzionale alle masse (distribuzione uniforme). Nel caso del codice ANSYS l'algoritmo usato nella soluzione è il metodo di Newton-Raphson, con matrice di rigidità aggiornata al passo. Nel caso del codice Code Aster è stato adottato un metodo in controllo di carico con lunghezza d'arco.

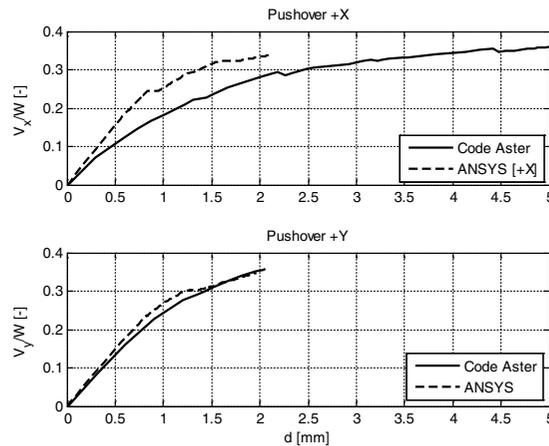


Fig. 18. Confronto delle curve di capacità.

La Fig. 18 confronta le curve di *pushover* ricavate per due distribuzioni di carico uniforme (una in direzione +X e l'altra in direzione +Y) ottenute con i due codici. È possibile osservare che mentre per il caso in direzione +Y si ha un sostanziale accordo (tanto in termini taglio ultimo alla base che di spostamento ultimo del punto di controllo, assunto come media degli spostamenti dell'ultimo livello), la direzione di carico +X mostra invece sostanzialmente lo stesso taglio ultimo a collasso (adimensionalizzato rispetto al peso totale dell'edificio), ma una differente duttilità, molto più marcata nel caso del codice Code Aster così come si registra anche una più rapida perdita di rigidità. In entrambi i codici non risulta possibile cogliere il ramo *softening* della curva di capacità.

Per quanto riguarda il quadro di danneggiamento riportato sulle pareti del prototipo, le Figg. 19 e 20 descrivono il risultato ottenuto nel caso della *pushover* uniforme con carico agente in direzione +X per le due pareti disposte parallelamente all'azione orizzontale imposta nell'analisi. È possibile verificare il buon accordo fra le zone nelle quali viene a svilupparsi il danno (Code Aster) con quelle dove si sviluppano le fessure (ANSYS).

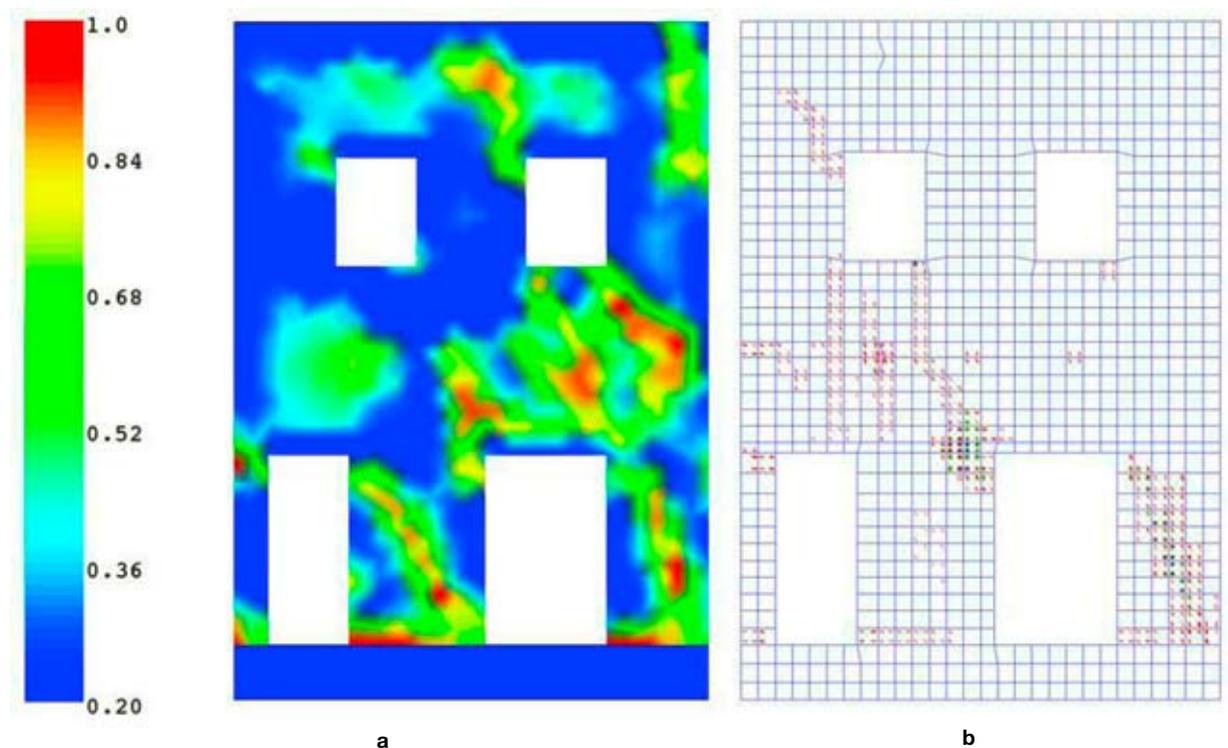


Fig. 19. Livello di danno per *pushover* uniforme direzione +X (parete AB, si veda Fig. 15b): quadro di danno Code Aster; b) *Crushing e Cracking*, ANSYS.

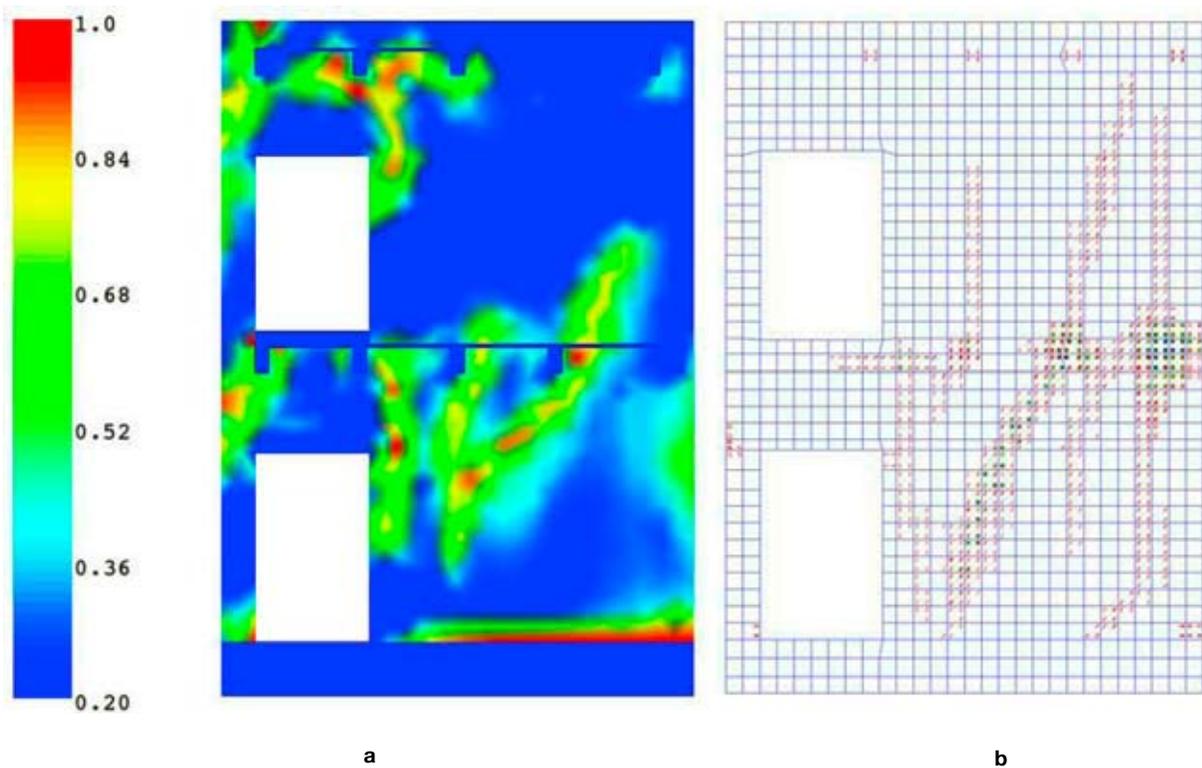


Fig. 20. Livello di danno per *pushover* uniforme direzione +X (parete CD, si veda Fig. 15b): quadro di danno Code Aster; b) *Crushing e Cracking*, ANSYS.

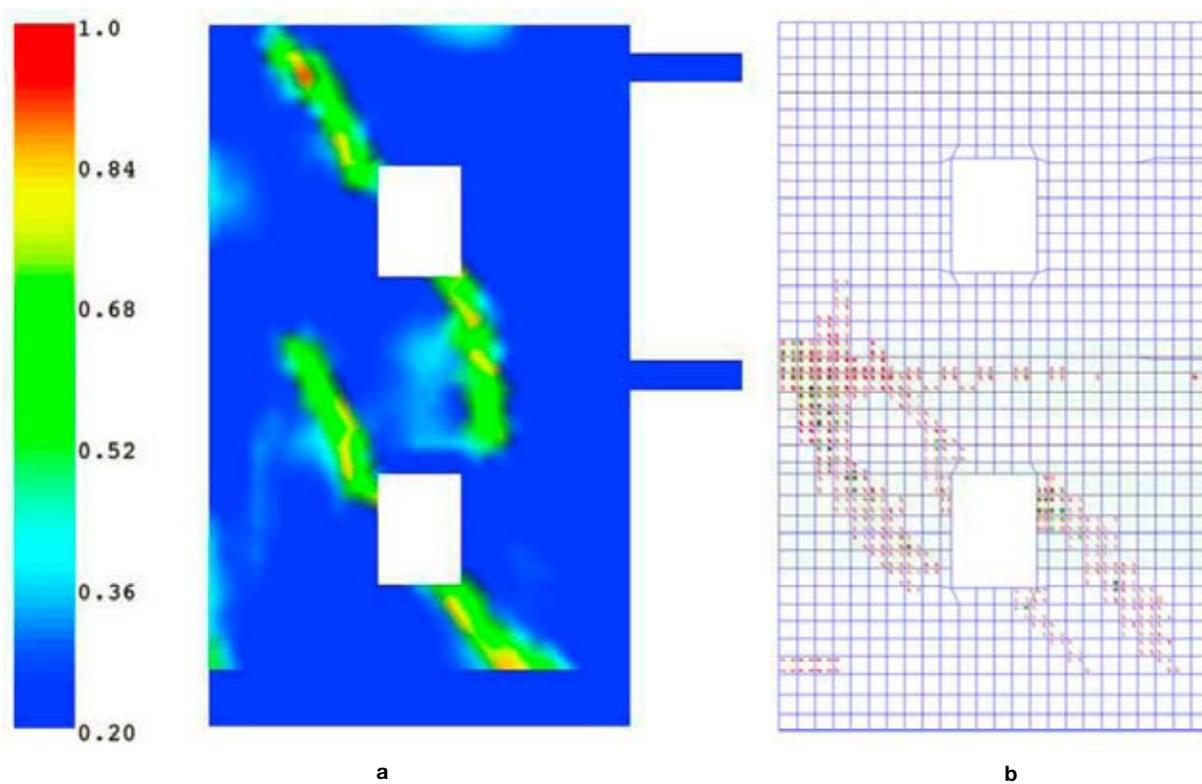


Fig. 21. Livello di danno per *pushover* uniforme direzione +Y (parete BC, si veda Fig. 15b): quadro di danno Code Aster; b) *Crushing e Cracking*, ANSYS.

La Fig. 21 riporta infine, nel caso di *pushover* in direzione +Y, il confronto tra il quadro di danno sulla parete BC (di veda Fig. 15b) ottenuto con Code Aster. È possibile osservare lo sviluppo di un meccanismo di una rottura per taglio per fessurazione diagonale della parete. Nel caso del codice ANSYS questo sembra interessare principalmente il primo livello del prototipo, nel caso del codice Code Aster è possibile invece osservare il coinvolgimento dell'intera parete. In generale comunque, all'interno delle incertezze di carattere epistemico, si osserva un buon allineamento tra i risultati offerti tra i due codici sia per quanto riguarda le curve di capacità che per quanto riguarda la restituzione del quadro di danneggiamento.

Conclusioni

La memoria ha mostrato l'impiego di un codice di calcolo agli elementi finiti nel settore dell'ingegneria tramite la modellazione e l'analisi di due casi studio relativi ad edifici e/o elementi in muratura. Il codice Open Source, operante in ambiente Linux, è stato utilizzato in combinazione con una piattaforma di uso generale che ne costituisce un pre e un postprocessor. Tale piattaforma è comprensiva di un modulo geometrico, un modulo di definizione della discretizzazione in elementi finiti ed un post processor grafico dei risultati. In essa è integrato il solutore non lineare. I due casi studio sviluppati (il modello di una prova di compressione diagonale, e delle analisi di spinta su un edificio regolare in muratura realizzato con gli stessi materiali oggetto della prova di compressione), hanno consentito di verificare sia la versatilità dello strumento Open Source, sia la sua robustezza nel trattare problemi caratterizzati da forti non linearità. I confronti fra i risultati ottenuti con i codici Open Source e quelli ottenuti con un codice commerciale mostrano come gli strumenti Open Source possano essere considerati una valida ed attraente alternativa ai tradizionali codici di calcolo. A fronte di, evidentemente, maggiori richieste nei confronti dell'utente (anche in termini di nozioni, oramai divenute di base, nella meccanica computazionale), essi risultano particolarmente attraenti anche per le possibilità di intervenire sul codice.

Ringraziamenti

L'autore Riccardo Corazzi ringrazia la società p.èn.lab Srl (www.penlab.it) presso la quale ha svolto il proprio tirocinio durante l'anno conclusivo dei propri studi magistrali approfondendo l'utilizzo del codice Opensource

Riferimenti Bibliografici

- [1] Torvalds L., Diamond D. (2011). *Rivoluzionario per caso. Come ho creato Linux (solo per divertirmi)*. Garzanti. ISBN 88-11-73896-2.
- [2] Raymond E.S. (1997). *The Cathedral and the Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary*. <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>.
- [3] Mazars J.M. (1984). *Application de la Mécanique de l'endommagement au comportement non linéaire et à la rupture du béton de structure*. Thèse de Doctorat d'état, Univ. Paris VI (France).
- [4] Mazars J.M., Pijaudier-Cabot G. (1989). Continuum Damage Theory. Application to Concrete. *Journal of Engineering Mechanics*, 115(2), pp. 345-365.
- [5] Progetto TRE.RE.M. (2006). Sito internet: <http://www.unibas.it/trerem/TREREMDW/>.
- [6] Betti M., Galano L., Vignoli A. (2007). Analisi a collasso di edifici in muratura: un modello dinamico tridimensionale per il progetto TRE.RE.M., *Atti del XII Convegno ANIDIS. L'ingegneria sismica in Italia ANIDIS XII*. Pisa, 10-14 Giugno 2007, paper n°403.
- [7] Corazzi R., Kovačević V. (2012). *Alle radici del palazzo del marchese: analisi e interpretazione dei dissesti statici del Palazzo Bourbon del Monte a Piancastagnaio (SI)*. Tesi di Laurea Magistrale in Ingegneria Civile, Facoltà di Ingegneria, Firenze.

Michele BETTI (mbetti@dicea.unifi.it), nato nel 1971 a Firenze, si è laureato in Ingegneria Civile nel 1996 presso la Facoltà d'Ingegneria dell'Università di Firenze. Nel 2001 consegue il titolo di dottore di Ricerca in Ingegneria delle strutture. Si interessa di progettazione strutturale occupandosi delle tematiche della vulnerabilità sismica di edifici monumentali (in muratura) e di identificazione strutturale.

Gianni BARTOLI (gianni.bartoli@unifi.it), è nato nel 1962 a Firenze e si è laureato in Ingegneria Civile nel 1988 presso la Facoltà d'Ingegneria dell'Università di Firenze. Attualmente è Professore Associato di Tecnica delle Costruzioni presso la Facoltà di Ingegneria dell'Università degli Studi di Firenze, dove principalmente si occupa di argomenti di ricerca collegati all'ingegneria del vento ed alla risposta di edifici monumentali.

Riccardo CORAZZI (riccardocorazzi@gmail.com), nato nel 1987 a Montevarchi, conclude il corso di studi magistrale in ingegneria civile nel 2012 presso la Facoltà d'Ingegneria dell'Ateneo Fiorentino. Dopo una prima esperienza con la società di ingegneria G.P.A. Ingegneria s.r.l, incentrata sulla valutazione e mitigazione del rischio sismico su strutture industriali esistenti, attualmente è libero professionista occupandosi principalmente di interventi sul patrimonio edilizio esistente. In parallelo si occupa di gestione e commercializzazione di impianti di cogenerazione a biomassa

Vladimir KOVAČEVIĆ (v.c.kovacevic@gmail.com), nato nel 1982 a Tomislavgrad (Bosnia ed Erzegovina), dopo aver conseguito il titolo triennale nel 2007, si laurea in Ingegneria Civile nel 2012 presso la Facoltà d'Ingegneria dell'Ateneo Fiorentino. Attualmente collabora con la A&I di Firenze, occupandosi di progettazione strutturale.