



**SOFTING SRL**

## **Nòlian**

### **Informazioni per la valutazione dell'affidabilità dei codici di calcolo**

#### **Identificazione del codice di calcolo**

Produttore:	Softing srl
Distributore esclusivo:	Softing srl
Denominazione commerciale:	Nòlian (denominazione commerciale registrata)
Identificatore di build:	Ogni versione è identificata da un numero a due cifre che ne indica la versione principale (EWS) e soprattutto da un numero univoco di build a quattro cifre che identifica con precisione una specifica compilazione.
Finalità del codice:	Analisi tramite il metodo degli elementi finiti
Lingua usata nella interfaccia e nella manualistica:	Italiano
Manualistica:	Digitale. Comprende anche manuali di validazione e teorici.
Registrazioni:	Nòlian è registrato presso il Registro Pubblico Speciale per i programmi per Elaboratore in data 14/07/2000 al progressivo 001629, ordinativo D002017, e con la precedente denominazione di MacSap è registrato in data 23/11/97 al progressivo 000222, ordinativo D000264.
Data di prima immissione in commercio:	Ottobre 1984.
Metodo di commercializzazione:	Cessione di licenza d'uso non esclusiva
Modalità di aggiornamento del codice:	Automatica tramite accesso a internet, su contratto
Identificazione univoca del codice:	Ogni copia del programma è identificata da un numero di licenza univoco tale da identificare il licenziatario.
Sistema di protezione:	Il codice dispone di un sistema hardware di protezione contro l'uso e la duplicazione abusiva.
Marchi commerciali registrati:	Softing®, il logo Softing, Nòlian®, il logo Nòlian®, MacSap®, MacBeam®, CADSap®, EasyWall®, EasySteel®, EasyBeam®, EasyFrame®, EasyWorld®, HyperGuide®, SapScript®, FreeLit®, inMod®, sono marchi registrati di Softing s.r.l.

#### **Cronistoria del prodotto**

La Softing è una srl costituitasi nel 1983 per la realizzazione di codici di calcolo in ambiente tecnico scientifico. Nòlian è stato progettato nel 1983 su commessa della Apple Computer Spa, che desiderava avere un prodotto per ingegneria che aderisse ai nuovi concetti di interfaccia grafica introdotti dalla Apple stessa. Nòlian (all'epoca MacSap) è stato da questa collaudato in fase di commessa e successivamente (1986) sottoposto a validazione da parte del Politecnico di Milano su



casi prova prodotti dallo stesso. La prima stesura del codice era in linguaggio Pascal. Completamente riscritto in C++ nel 1992 e portato su Windows. Dal 2004 non è più disponibile per Apple Macintosh.

### **Modelli matematici**

Nòlian è un codice di calcolo che impiega la tecnica degli elementi finiti per la soluzione di problemi nel campo dell'analisi tensionale (stress analysis). Questa tecnica è nota e ampiamente consolidata, documentata ormai da più di cinquant'anni di studi e ricerche. Quindi estremamente affidabile. Gli elementi finiti di Nòlian sono tutti progettati dalla Softing stessa che si è avvalsa anche di prestigiose collaborazioni. Gli elementi guscio e brick a rigidità completa (6 gdl per nodo), ad esempio, sono stati progettati dal Prof. Mario Cannarozzi, attuale vice Rettore della Facoltà di Ingegneria di Modena. Tutti gli elementi sono stati verificati con i principali casi prova disponibili in letteratura (si veda anche nel seguito) e hanno passato i patch test.

I solutori lineari fattorizzano la matrice di rigidità con varie tecniche numeriche. L'ampiezza di semibanda banda è ottimizzata automaticamente. Il più raffinato e attuale metodo di fattorizzazione di Nòlian segue la tecnica per matrici sparse che consente la gestione di strutture di gradi dimensioni con tempi di calcolo molto ridotti. Come riferimento, un sistema a 541.800 gradi di libertà (piastra 300x300 elementi), impiegando un processore Pentium 4 a 2.4 GHz con 1.5 GB di RAM, è stato fattorizzato in 2' 27". Questo metodo è disponibile in Nòlian sia nella modalità in-core che out-of-core.

L'analisi modale è condotta con il "subspace iteration method" su matrice dinamica sia piena che diagonale (modello delle masse sia "lumped" che consistente). Anche questo metodo è implementato sia con tecnologia skyline che sparse.

### **Implementazione dei modelli matematici**

L'implementazione è in C++, linguaggio a oggetti, strutturato in modo da riflettere con cura la struttura del codice a elementi finiti che ben si presta a questa strutturazione. Oltre ai metodi standard di debug, di profiling e gli altri strumenti prettamente informatici, il codice viene controllato costantemente durante lo sviluppo secondo le specifiche di progetto o di manutenzione delle singole unità. Si procede poi ai controlli di insieme e quindi alla validazione (vedi sotto). Softing usa un sistema CVS (Concurrent Version System) per il controllo delle fasi interne di sviluppo al fine di assicurare la massima qualità e produttività nel lavoro dei gruppi di sviluppo.

### **Criteri di assegnazione del modello di calcolo**

Il modello di calcolo è costruito in Nòlian tramite elementi della interfaccia utente. Non vi è pertanto una costruzione automatica tramite un modellatore solido e un mesher a monte. Ciò elimina qualsiasi sovrapposizione di errori imputabili ai pre-processor (modellatore e mesher). L'utente ha quindi il pieno controllo sulla mesh a elementi fini che viene costruita e gestita nello spazio tridimensionale tramite funzioni CAD standard. L'utente costruisce la geometria della mesh e, sempre con funzioni di interfaccia grafica, attribuisce alle entità geometriche le caratteristiche strutturali. Questo metodo assicura la massima trasparenza nella creazione, gestione, controllo del modello di calcolo eliminando i problemi connessi alla mancata trasparenza del modello stesso.



### **Criteri di interfaccia utente**

L'interfaccia grafica utente (GUI) di Nòlian nasce su Apple Lisa nel 1983 e quindi secondo i dettami Apple, che inventò tale modo di impiegare il calcolatore elettronico. Quindi Softing ha vissuto e partecipato alla nascita e allo sviluppo di questi criteri. Non si deve confondere l'insieme di metafore che consentono di interagire con il programma (GUI) con il modo di operare di un programma. Quindi una immissione dati, ad esempio per oggetti strutturali, non è una "interfaccia" ma un modus operandi del programma. Qui ci si riferisce al significato tecnico corretto del termine. Per questo motivo l'interfaccia di Nòlian è estremamente ergonomica e soprattutto chiara e trasparente. Non indulge in comandi che possano sembrare "attraenti" ma con poco significato operativo. Le funzioni associate ai comandi hanno tutte un'auto-diagnostica e sono tutte abilitate o disabilitate, a livello anche di interfaccia, secondo il contesto operativo. In genere, per le associazioni, si usa la tecnica di "verbo-predicato" perché più sicura e intuitiva. Quindi le possibilità di un uso accidentale scorretto dei comandi di interfaccia è molto limitato. Questi controlli vengono progettati insieme ai comandi stessi che ne fanno uso. Questa metodologia è estesa ed è la base della diagnostica nelle assegnazioni (vedi sopra: "Criteri di assegnazione del modello di calcolo").

### **Metodi di diagnostica sulla soluzione**

Metodi di diagnostica sono applicati sia per evidenziare la qualità della soluzione sia per filtrare a monte le eventuali scorrettezze del modello. Ad esempio il fattore di forma, di planarità, la coincidenza, la sconnessione degli elementi finiti è inibita dalla interfaccia grafica stessa (vedi) che non consente l'accettazione di elementi finiti non idonei.

Oltre ai criteri di diagnostica sulla plausibilità delle assegnazioni, vi è un'ampia possibilità di rappresentazioni grafiche sia generali che tematiche (ad esempio per intervalli di valori, per caratteristiche specifiche) che consentono un efficace e immediato controllo visivo sulle assegnazioni fatte dall'utente.

Durante l'elaborazione, gli algoritmi sono tutti "autoprotetti" nel senso che vi è un rigoroso controllo sia su dati anomali che possano falsarne il comportamento che su dati non plausibili per il modello matematico implementato. Inoltre eventuali limiti propri dell'algoritmo vengono accuratamente segnalati. L'utente può essere "deluso" nello scoprire che l'algoritmo non è in grado di risolvere un problema ma tale evenienza viene sempre verificata prima della elaborazione (o monitorata durante, secondo i casi) in modo che l'utente non possa mai ricevere risposte falsate da un algoritmo i cui limiti di applicabilità siano violati senza controllo. Oltre alla protezione degli algoritmi, vengono controllate tutte le quantità caratteristiche per un buon esito della elaborazione. A esempio, nella fattorizzazione monitorare la "decadenza di precisione" può essere più significativo di controllare solo la condizione di singolarità in quanto avvengono arrotondamenti e troncamenti che non consentono verifiche assolute in aritmetica finita. Queste valutazioni hanno spesso una interpretazione fisica precisa (a esempio una matrice singolare indica una labilità della struttura) e vengono sempre esposte all'operatore attraverso un apposito messaggio (avviso) di errore. I messaggi di errore di questo tipo non solo evitano soluzioni non corrette ma guidano l'utente verso la eliminazione dell'errore commesso.

### **Metodi di esposizione dei risultati**



I risultati in Nòlian sono esposti con metodo prioritario a “dialogo” intendendosi la possibilità di scegliere tematicamente il tipo di dato da consultare e poi “cliccando” l’oggetto che si desidera consultare per conoscere i valori del parametro sotto esame. Questo metodo consente un immediato controllo sui valori e sui punti critici senza richiedere la consultazione di stampe voluminose di difficile gestione. Inoltre Nòlian è dotato di potenti rappresentazioni grafiche, anche tra loro combinabili, che consentono un’immediata conoscenza del comportamento strutturale sia locale che globale. Infine si ha una produzione della descrizione del modello completa sia su file di testo che direttamente su stampante. La descrizione completa segue l’organizzazione tipica dei programmi a elementi finiti. E’ in forma sintetica tabellare e riporta una descrizione completa e ripetibile della struttura e dei risultati ottenuti in modo che l’analisi possa essere eventualmente ripetuta con altri programmi. Nòlian non produce “relazioni di calcolo”, che non è compito di programma di analisi strutturale, ma solo i dati suddetti che possono eventualmente essere allegati alla relazione di calcolo. Inoltre Nòlian ha la possibilità di copiare, registrare, stampare immagini sintetiche dei dati e dei risultati che possono essere impiegate nelle relazioni di calcolo per una documentazione sintetica del comportamento strutturale. Tra queste citiamo gli schemi del modello di calcolo dotati eventualmente della rappresentazione delle condizioni al contorno e delle azioni esterne agenti, e le mappe a colori di tutti i risultati tipici: dagli spostamenti agli sforzi. Si possono anche avere rappresentazioni personalizzate dall’utente tramite espressioni algebriche che possono documentare in modo sintetico presso che qualsiasi aspetto della struttura.

### **La relazione di calcolo**

Come meglio esposto sopra, al paragrafo “Metodi di esposizione dei risultati”, Nòlian non produce “relazioni di calcolo” essendo questo compito specifico del progettista. Nòlian produce elaborati completi e “ripetibili” (nel senso che le informazioni hanno completezza assicurata per la ripetizione del calcolo con altri programmi). Tali elaborati possono essere usati come allegati alla relazione di calcolo sia nella forma completa sia tramite estratti o immagini sintetiche che possono essere prodotti con Nòlian.

### **Criteri di validazione e controlli di qualità**

Il software non può essere “certificato” e cioè dichiarato rispondente a dei dettami normativi di costruzione bensì “validato” a fronte di casi prova. I casi prova sono generalmente le specifiche della commessa. Nel caso del software “pacchettizzato”, come questo, e non “custom”, in assenza di specifiche di validazione si impiegano dei casi prova forniti dalla letteratura internazionale. Si noti che i casi prova devono essere progettati con molta esperienza in quanto devono poter mettere in luce gli effettivi punti deboli del codice (stressing). Oltre alle verifiche tramite i casi prova, le verifiche vanno fatte sul comportamento globale del programma. La Softing ha un sistema di “bug tracking” che segue la fase di sviluppo (o aggiornamento e manutenzione), rilascio, uso presso l’utente. Questo rilevamento costante dà luogo a delle curve di difettosità che vengono costantemente tenute sotto controllo. Infatti la qualità del software si consegue in modo dinamico più che statico. Per questo motivo la Softing consiglia ai suoi clienti un costante aggiornamento del software e asserisce essere sempre maggiormente affidabile il software in corso più che le versioni obsolete. Gli utenti (si veda anche nel seguito) che hanno un contratto di manutenzione sono sia avvisati di eventuali malfunzionamenti riscontrati che messi in grado in tempi brevissimi di adeguare il loro codice tramite delle patch disponibili in automatico su Internet. Il metodo di bug-tracking della Softing è stato certificato ISO-9002. La Softing non ha ritenuto opportuno adeguare questa certificazione alla Vision 2000 ritenendo che gli espletamenti prettamente burocratici



imposti da queste normative comportasse un inutile aggravio di costi in quanto il metodo di controllo della qualità adottato da Softing è praticato a prescindere da certificazioni esterne di stampo burocratico.

### **Casi prova**

Per la validazione di Nòlian sono stati usati moltissimi casi prova reperibili in letteratura. I principali di questi sono dovuti al NAFEMS (National Agency for Finite Element Standard, della quale Softing è membro). Nòlian ha due manuali di validazione (uno per i casi lineari e uno per quelli non lineari) dove sono riportati gli esiti dei principali di questi raffronti con i casi prova. Si faccia riferimento a tali manuali per questo aspetto. L'elenco di alcuni casi prova è riportato in Appendice 1.

### **Criteri riassuntivi di qualità**

La produzione del software è ormai una tecnologia consolidata e confluita nella disciplina scientifica della Ingegneria del Software. La Softing è una azienda che produce software e lo fa dal 1983. Il software non può essere privo di malfunzionamenti. La curva di difettosità scende con il tempo ma ha andamento asintotico a valore non nullo. Il software più usato e collaudato è quello più sicuro a patto che vi siano, come nella Softing, strutture e metodologie atte a migliorare il software in base al monitoraggio di funzionamento nel tempo. La softing soprattutto impiega solo algoritmi assolutamente adeguati allo scopo che ci si prefigge di raggiungere, che ha come "filosofia" aziendale quella di dichiarare i limiti degli algoritmi che usa e di "filtrarne" l'uso via software. L'implementazione degli algoritmi avviene secondo metodi sperimentati e controllati che prevedono casi-priva progettati sia per il test dei componenti software che per i test dell'insieme. La Softing non usa algoritmi semplificati (nonostante essi spesso si avvicinano di più alle conoscenze professionali dell'utilizzatore e quindi siano commercialmente più "appetibili") in quanto essi rischiano di non avere sufficienti controlli in un sistema automatizzato. Oltre a questa rigorosa impostazione di partenza, la Softing segue nel tempo con rigore la difettosità del software rimuovendo puntigliosamente i malfunzionamenti ed, eventualmente, aggiornando costantemente algoritmi che abbiano trovato nuove soluzioni nella ricerca con ciò seguendo lo sviluppo sia dell'informatica che dei metodi matematici applicati all'ingegneria. La Softing offre ai propri utenti la possibilità di avere codici sempre aggiornati. Ciò tramite delle patch disponibili su internet che auto-aggiornano il codice. Le patch, essendo di ridotte dimensioni, non costringono a scaricare un'intera copia dell'eseguibile e di reinstallarla, ma la "riparano" automaticamente il codice eseguibile. In questo modo l'utente ha la garanzia che i malfunzionamenti eventualmente individuati siano subito rimossi anche nel suoi codice e ha la garanzia di avere i benefici della qualità di un software che cresce sempre nel tempo.

---

## **Appendici**

### **Appendice 1: Elenco di alcuni casi prova**



Elenco esemplificativo di 14 casi prova, tutti passati da Nòlian, dettagliatamente illustrati nel manuale di validazione di Nòlian per analisi lineare. Per l'analisi non lineare si consulti direttamente il manuale di validazione. L'elenco è per tipologia di elementi finiti e tipologia di problemi.

#### Elementi Trave: Analisi dinamica

1. Autovalori di un sistema di travi. Lo scopo di questo test è quello di verificare la qualità dei risultati in analisi dinamica di elementi trave. Si verifica il comportamento per un accoppiamento di caratteristiche estensionali e flessionali e la valutazione di autovalori coincidenti. Il test è proposto da: NAFEMS "The standard Nafems Benchmarks", test FV2.

2. Analisi dinamica di telaio. Questo test è uno dei pochi test in campo edile. Fu commissionato dalla Softing al Politecnico di Milano nel 1986 per validare Nòlian (allora MacSap). Contiene elementi tipici dei modelli edili: impalcati rigidi (master slave) ed elementi infinitamente rigidi (Rigel). Il risultati per la validazione, riportati in seguito, furono ottenuti con SAP IV.

#### Elementi Guscio: Patch test

3. Patch test per la membrana. E' un test importante. Di importanza anche storica perché è un test classico per elementi finiti. Se un elemento finto non passa i patch test, e questo è uno dei più impegnativi, non può essere impiegato nell'uso pratico. Il test è proposto in: R.H. MacNeal, R.L. Harder, "A proposed standard set of problems to test finite element accuracy" in Finite Element Analysis and Design, 1 (1985) 3-20, North Holland.

#### Elementi Guscio: Statica lineare

4. Torsione di mensola con sezione a "Z". Lo scopo del test è quello di verificare il comportamento di elementi guscio soggetti a un regime di sforzo complesso. Infatti viene trattata una mensola a "Z" soggetta a torsione. Il test è proposto in : NAFEMS : "The standard NAFEMS benchmarks", test LE5.

5. Piastra sottoposta a pressione. Il test riguarda una piastra appoggiata uniformemente caricata. Lo scopo del test è quello di verificare il comportamento degli elementi nel caso abbiano forma distorta. Il test è proposto in: NAFEMS " The standard Nafems Benchmarks", test LE6

6. Mensola con forza concentrata in estremità. Questo è un test basato su un elemento strutturale molto semplice, quindi adatto a confronti di immediata comprensione. L'importanza del test è nei parametri molto critici e nel ridotto numero di elementi impiegato che aiutano a valutare la "velocità di convergenza" degli elementi. Il test è proposto in: R.H. MacNeal, R.L. harder, "A proposed standard set of problems to test finite element accuracy" in Finite Element Analysis and Design, 1 (1985) 3-20, North Holland.

I valori di riferimento sono anche facilmente ricavabili da una formulazione analitica.

7. Membrana ellittica. Si tratta di una membrana ellittica con un foro ellittico. Viene modellato solo un quarto dell'ellisse. Lo scopo di questo test è quello di testare la capacità dei programmi di rappresentare adeguatamente forme curve e carichi applicati a lati curvi e inoltre di valutare gli sforzi nei punti di massima concentrazione. Il test è uno dei primi proposti dal NAFEMS e è dettagliatamente spiegato in Il test è proposto da: NAFEMS, "Beckground to Benchmarks" pag 5 e segg.

#### Elementi Guscio: Dinamica



8. Autovalori di piastra quadrata appoggiata. Scopo di questo test è verificare il comportamento in analisi dinamica degli elementi guscio. Generalmente un buon comportamento nella estrazione degli autovalori (direttamente correlati ai periodi) è indice anche di una buona affidabilità in analisi statica. Il test è proposto in: NAFEMS, "Selected Benchmark for Natural Frequency Analysis", Test 13.

9. Autovalori di mensola rastremata. Questo test verifica il comportamento membranale (a taglio) degli elementi piani soprattutto con una mesh irregolare. Il Test è proposto in: NAFEMS "The standard Nafems Benchmarks", test FV32

Elementi a deformazione piana e assialsimmetrici: statica lineare

10. Tubo di lunghezza indefinita. Modellazione di un cilindro di lunghezza indefinita tramite elementi a deformazione piana. Il test è proposto in: E. Hinton, D.R.J. Owen, Finite Element in Plasticity, Pineridge Press, 1980.

11. Elemento di lunghezza infinita con nucleo riscaldato. Lo scopo del test è quello di verificare il comportamento termo-elastico. Qui sono impiegati elementi a deformazione piana per modellare un elemento di lunghezza infinita, quale un conduttore. Il nucleo centrale presenta una variazione di temperatura rispetto agli elementi esterni. Tale variazione di temperatura causa una tensione negli elementi di contorno. Il test è proposto da: NAFEMS "The standard Nafems Benchmarks", test T1.

Elementi Brick: Statica Lineare

12. Piastra spessa sottoposta a pressione. Test su elementi brick di forma non regolare. Il test è proposto da: NAFEMS, "Beckground to Benchmarks" pag 77

13. Colonna spessa sottoposta a variazione di temperatura. Uso degli elementi brick per variazione di temperatura. Valutazione del comportamento per un campo di deformazione costante. Soluzione analitica.

Elementi Brick: Dinamica

14. Autovalori di mensola di forte spessore. Uso degli elementi brick per modelli di elementi di forte spessore. Il test è proposto NAFEMS "Selected BechMark for Natural Frequency Analysis", Test 51. e in NAFEMS "Free Vibration Benchmarks" vol. 3 pag. 451.