

C.T.A. Collegio dei Tecnici dell'Acciaio

GIORNATE ITALIANE DELLA COSTRUZIONE IN ACCIAIO

**SOFTWARE ORIENTATI ALLA COMUNICABILITA'  
ED AL CONTROLLO DEI DATI NELL'INGEGNERIA DELLE STRUTTURE**

**ing. Paolo Rugarli - Castalia s.r.l., Novara**

Le recenti acquisizioni nel campo dell'hardware hanno reso evidente la necessità di adeguare i programmi di calcolo agli elementi finiti, anch'essi per la verità abbastanza recenti, alle nuove tecnologie disponibili.

Pur dando per scontato ciò che scontato non è, vale a dire un impiego diffuso del metodo degli elementi finiti, v'è molto da fare per rimanere al passo coi tempi: generazioni di macchine sostituiscono le precedenti ogni due anni. Gli elementi finiti sono ormai ben studiati e ampiamente affidabili, eppure la loro diffusione non è ancora sufficiente. Di ciò noi crediamo sia responsabile principalmente il software.

In questa breve nota si vogliono richiamare quegli aspetti sui quali si deve intervenire (e lo si sta già facendo), per rendere agevole l'impiego del software per l'ingegneria strutturale, così da estendere l'uso del metodo degli elementi finiti, che, se accoppiato a strumenti efficienti, può consentire soluzioni economiche anche nel campo della ingegneria civile.

Per questo fine è essenziale far sì che il software sia semplice e rapido da usare, consentendo all'analista di avere tutte le informazioni che vuole (e solo quelle) in un tempo brevissimo. Oggi capita spesso che si impieghi più tempo per trovare un errore nel modello, anche banale, come una sconnessione non voluta o un vincolo mal posto, che per risolvere un problema da cinquemila gradi di libertà: è assurdo. Ci sono alcune cose ormai urgenti da realizzare, e tutti i codici agli elementi finiti dovrebbero adeguarsi, ivi inclusi i più potenti, sotto il profilo della classe di problemi affrontabili, "mostri" creati negli anni settanta che si trovano oggi parecchio indietro nella maneggevolezza e semplicità d'uso.

L'impiego della grafica.

Grazie alle interfacce grafiche dell'ultima generazione è possibile ormai fare del tutto a meno delle famigerate numerazioni, che hanno costretto per anni gli analisti a classificare i nodi e gli elementi in base al numero,

al fine di individuarli: attualmente il generico elemento viene estratto indicandolo con un sistema di puntamento automatico quale il mouse o la tavoletta grafica, e ciò consente una facilità ed immediatezza paragonabili a quelle di una normale conversazione.

La creazione della mesh e la sua numerazione sono procedure ormai automatiche: l'analista introduce le coordinate di alcuni punti notevoli e sfrutta delle primitive grafiche per descrivere il proprio modello nelle sue linee generali; in un secondo tempo, per via automatica, suddivide il dominio così individuato creando il modello discretizzato.

Ma non è solo sotto il profilo della generazione delle incidenze e dei nodi (coordinate e numerazione) che la macchina può essere sfruttata: ancora oggi molti importanti codici di calcolo agli elementi finiti non consentono il puntamento col mouse (o la tavoletta grafica), e per conoscere le coordinate, gli spostamenti o le forze in un nodo, occorre saperne il numero.

Ciò è ormai obsoleto, poichè una trattazione grafica e colloquiale del problema consente di avere queste notizie puntando col mouse il nodo desiderato e premendo un tasto. L'operazione è concettualmente semplice e risparmia al progettista il fastidio di cercare su tabulati o nei file di input le notizie per lui utili, lasciandogli più tempo per fare le cose realmente importanti, quelle che la macchina non può in alcun modo fare (né mai crediamo potrà). Che il computer debba sapere "che numero ha l'elemento che descrive la colonna 1-h tra le quote 6.25 e 10.25" è comprensibile, ma al progettista non interessa: indichi "quella" colonna col mouse, il programma farà il resto.

Per accedere graficamente alle informazioni è indispensabile gestire le operazioni di zoom, di rotazione, di cambiamento di vista del modello, in modo efficiente. Non è abbastanza rapido dare 6 numeri che definiscono gli estremi di una regione per poi vedere ciò che in essa è contenuto: la regione si può definire più celermente ed in modo più preciso premendo due volte col mouse. Infatti, le coordinate in sè non sono rilevanti: l'essenziale è definire il particolare voluto. Non ci si deve più preoccupare di trovare le coordinate giuste per la regione, quelle entro le quali il particolare cercato è ben contenuto: è il programma che deve saperle trovare. È snervante dover battezzare la regione, trovarne gli estremi, eppoi vederla, scoprendo d'aver magari sbagliato un estremo, ragion per cui occorre ricominciare: ciò che si voleva era un ingrandimento di una zona del modello, subito, per verificare che in quella zona tutto fosse a posto, e con due pressioni di tasto del mouse il problema è risolto.

Accedendo alle informazioni in modo grafico si può controllare molto più rapidamente il modello, sia in fase di input che di output: sembra in effetti

di dire cose ovvie, ma ovvie non sono se a tutt'oggi la maggioranza dei codici ad elementi finiti non fa queste cose; ovvio non è poichè molti sviluppatori software del settore non reputano sufficientemente seri questi problemi, che sono invece capitali per consentire l'impiego del metodo come uno strumento, e non come il fine del proprio lavoro.

#### Ricerca e limitazione degli errori.

Chi ha lavorato con gli elementi finiti sa che buona parte del lavoro consiste nella ricerca degli errori inseriti accidentalmente nel file di input. Gli errori possono essere fatti per vari motivi: esistono errori di introduzione dei dati; errori nella descrizione fisica della struttura da parte del modello (errori di modellazione); esistono infine, in particolare, errori nella generazione dei carichi. Ad esempio, l'introduzione erronea di un codice di vincolo, oppure la corrispondenza scorretta tra proprietà sezionali e assi di riferimento locali, sono tipici errori di modellazione.

I più recenti codici di calcolo dovrebbero essere progettati non solo per fornire sofisticazioni di calcolo crescenti (elementi finiti più precisi o potenti, leggi costitutive elasto-plastiche ecc.), ma anche per rendere meno probabili gli errori.

Gli errori dovuti alla introduzione dei dati possono essere resi meno probabili riducendo drasticamente il numero dei dati da inserire manualmente. Ciò si fa avvicinando al ragionamento umano il "ragionamento" del software, e non viceversa, come è avvenuto finora. Un esempio può chiarire questo concetto: se l'utente desidera attribuire ad un certo elemento finito le proprietà di una IPE300, usando il software tradizionale egli deve svolgere le seguenti operazioni: a) cercare nei suoi schizzi a mano, nei disegni del plotter oppure nel file di input, che numero ha l'asta alla quale intende attribuire le caratteristiche di una IPE300; b) cercare nel profilario i 6 numeri che individuano, ai fini del metodo, il profilo tipo IPE300; c) scrivere in una certa linea questi sei numeri, in un'unità coerente con quella adottata nel modello (ciò implica in generale sei somme o differenze d'esponente); d) stare attento all'ordine con cui introduce i dati, poichè a tale ordine è associata l'orientazione dell'asta nello spazio; e) assicurarsi che le convenzioni adottate per orientare l'asta siano state rispettate.

Non sorprende che molti trovino tutto ciò troppo macchinoso e lontano dal problema originario, che è attribuire una sezione ad un'asta. Chi si occupa prevalentemente di elementi finiti ha sviluppato una serie di tecniche che rendono un pò più rapida questa trafila, ma non si può chiedere a tutti di fare altrettanto, e non si può portare la dimestichezza del fiaccolaio come un motivo sufficiente per conservare l'uso dei lampioni a gas.

Un software della nuova generazione potrebbe comportarsi invece così: con il mouse viene indicata l'asta che si vuole sia un' IPE300; premendo un tasto si accede per via grafica ad un archivio profili e si dice al programma la parola chiave "IPE300"; immediatamente a schermo compare il disegno del profilo IPE300 e il disegno della struttura nella vista corrente, con l'asta selezionata chiaramente visibile, insieme alla sua orientazione (rappresentata, quest'ultima, da una freccia visibile a schermo); premendo una terza volta col mouse si dice al programma che il profilo mostrato è associato all'asta prescelta, disponendo il profilo in modo tale che la freccia sull'elemento selezionato coincida con l'analogica freccia che si vede disegnata sull'IPE300. Se si desidera che tra le due frecce vi sia un angolo, all'esplicita domanda del programma si risponde introducendo tale angolo in gradi. In tutto si sono introdotti solo i seguenti dati: "IPE300" e per esempio, "90°". In tal modo il problema dell'orientazione è notevolmente semplificato: occorre solo far coincidere due frecce.

E' poi il programma che deve leggere le inerzie della IPE300 nel suo database e convertirle nella unità prescelta; è ancora il programma che deve calcolare le snellezze e le deve mostrare in modo chiaro, sotto forma di mappe a colori o indici scritti sull'asta a schermo. Può essere ancora il programma ad assistere nella scelta del profilo, perchè deve essere possibile far sì che esso trovi un insieme di profili soddisfacenti alcuni requisiti, non ultimo la reperibilità, in modo del tutto automatico: se si vogliono snellezze massime inferiori a 100 e minime superiori a 40, ma profili europei del tipo HEA, HEB o HEM, deve trovare la macchina quali profili soddisfino questi requisiti, non chi la usa.

Curare questi aspetti di interfacciamento consente di ridurre drasticamente il volume di informazioni "brute" da fornire, rendendo al tempo stesso più piacevole il lavoro e meno probabili gli errori di introduzione dei dati.

Anche gli errori di modellazione possono essere resi meno probabili, consentendo che le scelte fatte in termini di vincoli e svincoli trovino una immediata e chiara rappresentazione grafica: non più la ricerca di serie di zeri e di uni, ma la rappresentazione grafica del vincolo, univoca. Nel caso delle convenzioni usate per orientare le aste, s'è già visto che esistono metodi atti a rendere intuitivo e, si direbbe, "palpabile", ciò che si sta facendo: il principio generale dovrebbe essere quello di far vedere a chi usa il software le conseguenze delle sue scelte. Non si può chiedere a tutti la dimestichezza che consente di vedere un "1" in cinquantesima colonna là dove dovrebbe esserci, manuale alla mano, uno "0".

Anche eventuali labilità potrebbero essere, almeno in parte, rese evidenti per via grafica: basterebbe aggiungere al criptico messaggio del file di output, "check dof 4357", ovvero "verificare il grado di libertà numero 4357", un opportuno simbolo luminoso sul nodo a cui è associato il "dof" in questione: il più delle volte, noto il nodo, è richiamata alla mente la dimenticanza che ha generato il problema.

Nel caso delle azioni gli errori di modellazione derivano in genere dall'errata attribuzione delle forze alle aste (scambiando i numeri), oppure dall'errata risultante (per esempio, errori nell'unità di misura).

La gestione dell'unità di misura è automatizzabile e deve essere automatizzata, poichè cambiare unità di misura nel corso del lavoro comporta esclusivamente qualche moltiplicazione. Cambiare unità di misura a seconda delle esigenze che si prospettano via via, vuol dire parlare alla macchina, e recepire i suoi messaggi, in una lingua a noi più chiara, e per lei del tutto equivalente: se si è abituati ad usare le tonnellate e non ci si ritrova coi kN, si possono usare le tonnellate, sarà poi la macchina a scrivere la relazione di calcolo in kN. Le unità che sono congeniali per i carichi, sono molto scomode se si ragiona, per esempio, sui momenti di inerzia: si preferiranno altre unità, in tal caso, e la macchina deve consentire la sostituzione immediata.

L'applicazione di forze a nodi o aste sbagliate è resa improbabile dal fatto che si può procedere graficamente: valgono qui le considerazioni già fatte sulla rapidità di puntamento tramite mouse o tavoletta grafica.

L'introduzione di moduli sbagliati è resa meno probabile scalando le frecce che a schermo rappresentano le azioni in funzione del loro modulo, e consentendo di sapere in ogni momento, nell'unità di misura che l'utente vuole, quanto vale la singola azione o la risultante di gruppi di azioni selezionati per mezzo del mouse. "Che carico c'è sulla copertura" è una domanda legittima: per rispondere deve poter bastare premere due o tre volte in modo opportuno un tasto. Sarebbe assurdo non solo sommare i numeri che rappresentano le singole forze, ma anche scrivere un programmino di utilità per farlo: il software di cui si sta parlando è il "programmino".

Software con queste caratteristiche possono rendere assai meno probabili gli errori: l'introduzione stessa dei dati diviene estremamente limitata: se un carico è definito a regione, basta la sua intensità e la descrizione della regione; il calcolo delle aree di influenza dei pilastri o dei nodi di facciata può e deve essere automatizzato, rendendo più agevole l'impiego del software e meno probabili errori di calcolo banali e fuorvianti. Si tenga presente che poichè l'utente non si cura della numerazione nodale, i nodi adiacenti ai fini del dominio di influenza non vanno trovati sulla base del

loro numero, ma sulla base delle proprietà geometriche del luogo a cui appartengono. L'attribuzione dei carichi sulla base dei domini di influenza può anche richiedere l'elaborazione di complicati algoritmi, capaci di catalogare la topologia della superficie prescelta sì da scovare i tutti i nodi ad essa appartenenti, e il loro dominio di influenza: non sono problemi sempre banali, l'esperto di programmazione può trovare lavoro avvincente ed al tempo stesso importante, perché svolgerlo significa semplificare notevolmente l'impiego del metodo degli elementi finiti.

#### Modellazione dei carichi.

Un altro settore promettente e pressoché del tutto inesplorato è quello della modellazione dei carichi, intesa come programmazione delle normative che fissano le modalità di calcolo delle azioni di progetto.

Gli indirizzi normativi più recenti tendono a complicare e specializzare questa parte del lavoro, ragion per cui sembra ormai matura l'istituzione di un insieme di procedure automatiche capaci di risparmiare al progettista i calcoli più ripetitivi. Si prenda, ad esempio, il vento: è possibile definire topologicamente l'esterno della struttura, associandolo a classi predefinite, e creare parametricamente interi insiemi di forze equivalenti. Generate le direzioni del vento, e definite col mouse le superfici esposte ad esso, nota l'ubicazione geografica del sito sono sufficienti poche altre informazioni per avere i carichi nodali. Lo sforzo da fare, in questo come in casi simili, è quello di trovare concetti sufficientemente flessibili e generali da poter costituire la base per l'implementazione. Si richiede uno sforzo di fantasia, in somma. Nel caso delle superfici, per esempio, si dovrebbe far sì da definirle semplicemente (col mouse), stabilendone gli attributi utili ai calcoli - permeabile, liscia, scabra -, in modo rapido ed efficiente.

Nel caso della neve la definizione della superficie è bastevole, insieme al dato geografico, per risolvere la maggioranza dei casi. Le procedure di cui si tratta si limiterebbero a domandare la regione geografica, generando i carichi in modo automatico.

Nel caso del sisma, l'esecuzione automatica di un'analisi modale con  $m$  modi, ove  $m$  è tale da garantire masse partecipanti percentualmente superiori ai valori di soglia previsti dalle norme, accoppiata con spettri di risposta di normativa, è tecnicamente fattibile: la generazione delle masse può essere svolta automaticamente, perchè se il programma capisce che, per esempio, il caso di carico 2 è di tipo "vento", sa anche che a quel carico non è associata massa; ed i calcoli sui coefficienti di partecipazione atti a verificare la percentuale di massa oscillante sono facilmente programmabili.

Si tenga presente che gli eurocodici prevedono obbligatoriamente le analisi dinamiche a spettro di risposta per classi sempre più vaste di strutture, e che non tutti i progettisti hanno nel proprio bagaglio di studi aggiornate nozioni di dinamica e di elementi finiti. Tanto vero che la necessità di fare calcoli seri sul tema è tutt'altro che soddisfatta.

Anche nel campo dell'interazione suolo struttura, la modellazione del terreno per mezzo di molle equivalenti può essere concepita in modo automatico: nota la forma e la dimensione della fondazione, ed il modulo G del terreno, esistono formule chiuse ed abachi capaci di fornire il valore delle molle equivalenti.

Accoppiato alle routine di cui s'è detto sopra, un software di questo tipo potrebbe consentire di ridurre ulteriormente i dati in input, facendo eseguire analisi più raffinate ad un insieme più numeroso di utenti, in modo complessivamente più rapido. Se si riuscisse a calcolare le azioni statisticamente più probabili sulla base dei riferimenti normativi, in modo sostanzialmente automatico, il progettista potrebbe finalmente soffermarsi sul problema che naturalmente gli dovrebbe essere più vicino: come concepire la struttura per far sì che resista alle azioni in modo ottimale.

#### Produzione di disegni esecutivi.

Si vuole ora accennare alla possibilità di realizzare ambienti di lavoro capaci di gestire problemi molto diversi, ma egualmente importanti per il progettista: mentre da un lato sono indispensabili i calcoli agli elementi finiti e l'esecuzione di verifiche in modo automatico, dall'altro sono indispensabili mezzi atti a produrre rapidamente i disegni. Le due necessità trovano oggi risposte separate, ottenute grazie a strumenti diversi. La ragione è nel fatto che solo da pochissimi anni sono disponibili mezzi adeguati; e nella differenza di formazione che separa gli esperti dei due settori: difficile essere al tempo stesso esperti di elementi finiti e di problemi di disegno tecnico, quelli cioè che un disegnatore deve trovarsi a risolvere quotidianamente. La soluzione al problema non può quindi che venire dal collegamento dei vari programmi che si occupano di elementi finiti, con i programmi che si occupano di disegno.

Esiste oggi la possibilità di consentire al progettista ed ai suoi collaboratori l'impiego di un unico grande strumento, che prenda i risultati delle analisi agli elementi finiti, con tutte le informazioni sulle aste e la loro orientazione, come dati di input per il successivo modulo orientato al disegno. Questo strumento dovrà consentire di rieseguire le analisi numeriche per tener conto di modifiche successivamente adottate, consentendo così ai due "mondi" un collegamento continuo.

La necessità di distruggere la Babele dei diversi linguaggi e delle diverse convenzioni è impellente: i vari codici agli elementi finiti devono dialogare tra loro, e devono dialogare con i programmi di CAD. L'utente dotato del programma "alfa" che si trovi a dover risolvere un problema che solo "beta" risolve, deve poter trasferire in modo automatico i suoi dati ed il suo lavoro al programma "beta". In questo modo il servizio reso a chi usa i programmi sarà davvero valido, e le punte di efficienza di un programma saranno disponibili a tutti gli altri.

#### Conclusione.

I quattro aspetti sopra richiamati non esauriscono l'insieme di necessità oggi esistenti, ma descrivono in modo sufficiente cosa si intenda per software orientati alla comunicabilità ed al controllo dei dati, nel campo della ingegneria strutturale. Forse, oltre alla precisione e la versatilità, non è da trascurare la maneggevolezza, se si vuole davvero che il metodo degli elementi finiti divenga patrimonio comune, piuttosto che palestra per iniziati.

Non ci paiono lontani i tempi in cui si potranno fare (pre e postprocessing inclusi) 5 analisi in un'ora, variando parametricamente le dimensioni, i tipi sezionali, le tipologie costruttive, ottenendo poi una stima ragionevole dei costi, e l'insieme dei disegni esecutivi, in output: la progettazione di software con queste capacità è attualmente il nostro obiettivo e la nostra speranza.