

Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006. ISBN 85-7515-371-4

APLICAÇÃO GRÁFICA INTERATIVA PARA ENSINO DO MÉTODO DOS ELEMENTOS FINITOS

Renata Nicoliello Moreira – renatanm@fea.fumec.br
Faculdade de Engenharia e Arquitetura da FUMEC
Rua Cobre 200.
30310190 – Belo Horizonte - MG
Roque Luiz Pitangueira – roque@dees.ufmg.br
UFMG - Universidade Federal de Minas Gerais, Departamento de Engenharia de Estruturas
Av. do Contorno, 842, 2º andar.
30110060 – Belo Horizonte - MG

Resumo: O artigo apresenta uma expansão do INSANE (INteractive Structural ANalysis Environment): um sistema computacional para modelos discretos de análise estrutural do método de elementos finitos (MEF). A expansão de que trata este artigo refere-se ao desenvolvimento de uma interface gráfica para auxiliar o ensino do MEF para engenharia de estruturas. As características que tornam o sistema amigável a mudanças e escalável em complexidade, bem como os recursos tecnológicos que o dotam destas características são apresentados. O uso do computador como ferramenta favorável ao processo de aprendizagem é discutido, tanto do ponto de vista histórico quanto do ponto de vista do próprio INSANE, enfatizando as possibilidades que o INSANE oferece para este processo. Dentre estas possibilidades, apontam-se aquelas que permitem expor o núcleo numérico do sistema computacional para o usuário, de modo que o mesmo possa visualizar informações referentes à resolução de modelos do MEF, interagir com o sistema e interferir no processo. Partindo-se desta discussão e de sugestões disponíveis na literatura para caracterização de etapas de solução de problemas através do MEF, apresenta-se a análise orientada a objetos para a interface gráfica. O projeto orientado a objetos da expansão é então apresentado, com base no padrão Model-View-Controller, onde o modelo é separado de sua representação, já adotado na implementação do sistema. Finalmente, os recursos disponibilizados na nova interface são apresentados através de exemplos e as possibilidades de enriquecimento do processo de aprendizagem são, então, discutidas.

Palavras-chave: Método de Elementos Finitos, Computação Gráfica, Programação Orientada a Objetos, Ensino de Engenharia.

1. INTRODUÇÃO

O impacto das novas tecnologias no ensino de engenharia é inquestionável, fazendo com que os profissionais desta área estejam atentos aos rumos desta transformação. A revolução proporcionada pela facilidade de acesso à informação, além de impulsionar o mercado científico e tecnológico, se faz presente nos métodos pedagógicos e educacionais, podendo ser usada em favor do processo de aprendizado.

O uso de softwares e recursos computacionais diversos na prática de engenharia não é novidade. Entretanto, ainda é preciso a utilização destes recursos para transformar metodologias de ensino, auxiliando professor e aluno no processo de aprendizagem.

O grande desenvolvimento do Método dos Elementos Finitos (MEF) nas últimas décadas e a facilidade de acesso ao computador alterou a prática da análise estrutural, tornando possível o estudo de estruturas de geometria arbitrária, constituídas por múltiplos materiais e sujeitas a qualquer tipo de carregamento.

O ensino do MEF, por sua vez, pode se tornar bastante árido, à medida que exige do aluno conceitos físicos e matemáticos e abstrações aprofundados, para formulação dos modelos. Para facilitar o aprendizado do MEF, é fundamental dispor do computador como uma ferramenta favorável ao processo. Assim, em lugar do uso tradicional, em que as soluções são obtidas automaticamente a partir da inserção dos dados iniciais do problema, propõe-se aqui, um sistema computacional, em que a solução aconteça passo a passo, e o aluno possa interagir e, interferindo no processo, aprender a metodologia.

Este artigo apresenta uma expansão do INSANE (Interactive Structural Analysis Environment): um sistema computacional para modelos discretos de análise estrutural do método de elementos finitos (MEF). Esta expansão refere-se ao desenvolvimento de uma interface gráfica para auxiliar o ensino do MEF para engenharia de estruturas.

2. O PROJETO INSANE

A pesquisa na área de métodos numéricos e computacionais para os modelos discretos de análise estrutural procura um aprimoramento das hipóteses simplificadoras dos mesmos. Isto é feito de forma a ampliar complexidades a partir dos conceitos já consolidados. Entretanto, observa-se sempre um recomeço do processo ao se recriarem as ferramentas relativas às tecnologias dominadas. Um exemplo ilustrativo deste fato é a (re) implementação computacional de algoritmos de solução de sistemas de equações algébricas lineares, toda vez que os mesmos são usados como parte do processo de aprimoramento de determinado modelo discreto.

Ao longo do tempo, algumas iniciativas de desenvolvimento de software pela comunidade acadêmica resultaram em produtos dependentes de sistema operacional, pouco amigáveis, escritos em linguagens de programação não apropriadas, de expansão, distribuição e manutenção difíceis, desenvolvidos por equipes fechadas, com documentação deficiente, entre outras limitações. Tais fracassos podem ser creditados a falta de disposição da comunidade em se apropriar das tecnologias emergentes ou mesmo a inexistência das mesmas.

Esta constatação confronta-se com o surgimento e aprimoramento de soluções tecnológicas para desenvolvimento de software, como programação orientada a objetos (FONSECA e PITANGUEIRA, 2004), linguagem Java (SANTOS, 2003), XML (eXtensible Markup Language) (TIDWELL, 2001), padrões de projeto de software (GAMMA et al., 1995), entre outras. O domínio destes recursos e a aplicação dos mesmos no aprimoramento progressivo dos modelos, sem ter que recomeçar o processo a cada novo aperfeicoamento.

requer um ambiente computacional segmentado, amigável a mudanças e escalável em complexidade.

O projeto INSANE ("INteractive Structural ANalysis Environment") objetiva desenvolver um sistema computacional com estas características. Para isto, o ambiente possui três grandes segmentos: pré-processador, processador e pós-processador (SORIANO e LIMA, 1999).

Os pré e pós-processadores são aplicações gráficas interativas, implementadas na linguagem Java, que disponibilizam recursos diversos para diferentes modelos discretos. O processador é uma aplicação, também implementada em Java, que representa o núcleo numérico do sistema. Este núcleo é responsável pela obtenção dos resultados de diferentes modelos discretos de análise estrutural. A persistência dos dados compartilhados pelas três aplicações é alcançada através de uma interface baseada em arquivo(s) XML e/ou objetos Java.

Cada um dos três segmentos da aplicação é implementado segundo o paradigma de programação orientada a objetos (POO), adotando-se uma arquitetura em camadas e padrões de projeto de software apropriados.

2.1 Arquitetura em Camadas e Padrões de Projeto de Software

A figura 1 mostra a combinação da arquitetura em camadas e padrões de projeto de software adotada para o INSANE (FONSECA e PITANGUEIRA, 2004). Como pode ser visto na figura, a versão atual do sistema possui quatro camadas lógicas e duas camadas físicas. Três das camadas lógicas do sistema foram definidas utilizando-se o padrão de projeto de software denominado *Modelo-Vista-Controlador* (MVC) (GAMMA et al., 1995). Este padrão é bastante apropriado uma vez que preconiza a separação do processamento da informação de sua representação gráfica, facilitando assim os trabalhos de expansão e manutenção da aplicação. A quarta camada lógica é a camada de persistência.

Em termos físicos, o INSANE possui atualmente somente duas camadas: uma aplicação carregada na memória do computador (compreendendo as camadas lógicas Modelo, Vista e Controlador) e arquivos textos e/ou binários persistidos em disco.

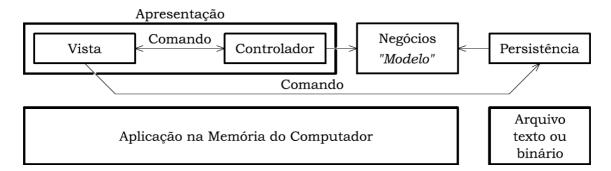


Figura 1 – Arquitetura em Camadas e Padrões de projetos adotados no INSANE.

O inter-relacionamento entre as camadas é conseguido, principalmente, através da implementação do padrão de projeto de software denominado Comando. A figura 2 exemplifica este relacionamento para o caso da tarefa de adição de uma entidade geométrica ao modelo corrente e sua visualização. Como pode ser visto nesta figura, o fluxo de informações para realização de tal tarefa ocorre em quatro etapas. Na primeira etapa, o objeto Comando, responsável pela tarefa, aciona o Controlador ativo informando a requisição. A seguir (2), o Controlador cria o objeto correspondente à entidade geométrica e o adiciona ao Modelo pertinente. Na etapa 3, o Controlador cria objetos de desenho

representativos dos objetos do Modelo. Finalmente, na etapa 4, os objetos de desenho pertencentes ao Controlador são apresentados na área de desenho da Vista.

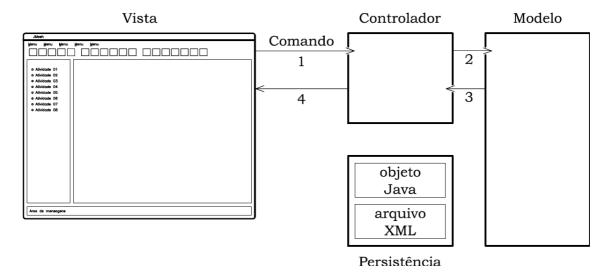


Figura 2 – Relacionamento entre camadas do INSANE, para adição de uma entidade geométrica.

3. SOLUÇÃO INTERATIVA PARA O INSANE

A análise estrutural via MEF pressupõe a divisão do domínio de estudo em sub-domínios interconectados, denominados elementos finitos. Na formulação do MEF em deslocamentos, cada um dos elementos finitos tem uma função de aproximação de deslocamentos a ele associada. Nestes elementos também é necessário estabelecer hipóteses relativas ao regime de deformações (relações DeformaçõesXDeslocamentos) e ao comportamento do material (relações TensõesXDeformações). Utilizando princípios físicos apropriados, estas três hipóteses são consideradas na obtenção das equações de equilíbrio nodais do elemento, gerando assim, sua matriz de rigidez.

Os vários elementos de uma discretização do MEF são interconectados entre si através de nós. A imposição das condições de equilíbrio a cada um dos nós da malha, considerando as equações de equilíbrio do elemento e a compatibilidade de deslocamentos nos nós, permite obter-se o sistema de equações de equilíbrio do modelo. Este sistema, inicialmente singular, pode ser resolvido através da imposição das condições de contorno.

Obtidos os deslocamentos nodais, pode-se reutilizar as três hipóteses fundamentais de qualquer um dos elementos finitos, para a obtenção de grandezas internas (deslocamentos, deformações e tensões).

Este processo está ilustrado na figura 3, conforme sugestão de LOGAN (2001). As etapas mostradas na figura 3 podem ser agrupadas em três blocos, como normalmente encontrado em um programa de análise via MEF (SORIANO e LIMA, 1999). (Pré-processador (Etapa 1), Processador (Etapas 2 a 7), Pós-processador (Etapa 8)). As etapas 1 e 8, caracterizadas pelo pré-processador e pós-processador, respectivamente, são partes do programa em que o usuário interfere diretamente, tanto nos programas didáticos, como nos comerciais. Nas etapas de 2 a 7, geralmente, não ocorre interferência do usuário. O processamento dos dados acontece de maneira automática, sem que o usuário tenha acesso às etapas da solução. As classes, que foram concebidas na expansão do programa discutida neste artigo, estão relacionadas com as possíveis interações descritas na figura 3.

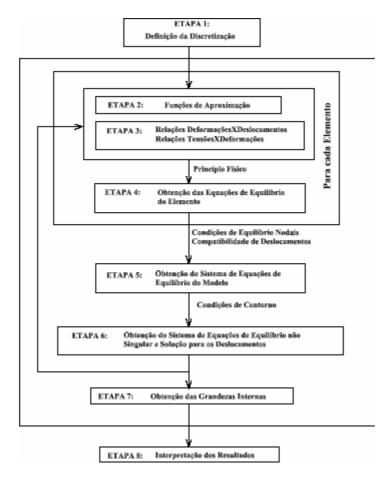


Figura 3 – Etapas de análise através do MEF.

4. EXPANSÃO JÁ IMPLEMENTADA

4.1 Modelos de Barras

A versão atual do sistema disponibiliza recursos que facilitam a compreensão do processamento de uma malha composta por elementos de barra.

Na figura 4 definiu-se um modelo de pórtico plano para exemplificar o funcionamento da solução interativa. Definido o modelo da análise e todos os atributos necessários para a solução via MEF, inicia-se a solução interativa, em que o usuário acompanha passo a passo o processamento do sistema. Na figura 4 pode-se observar a primeira opção em que o usuário pode interferir no processo. Nesta etapa é feita a numeração das equações do modelo, a partir das restrições nodais da estrutura. Esta numeração pode ser feita de maneira automática ou o usuário pode selecionar a ordem em que a numeração deve ser executada.

Outro recurso, também já implementado na solução interativa, é a visualização da matriz de rigidez de cada elemento do modelo. Este recurso permite tanto a consulta à matriz completa, sem a imposição das condições de contorno do problema, quanto à matriz reduzida. A figura 5 mostra a consulta à matriz de rigidez reduzida, em que são eliminados os valores referentes aos graus de liberdade restritos. O programa também permite a visualização das matrizes de força completa e reduzida dos elementos.

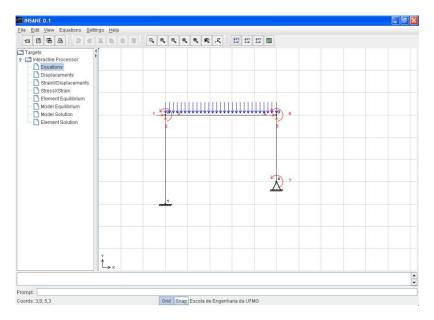


Figura 4 – Numeração das equações.

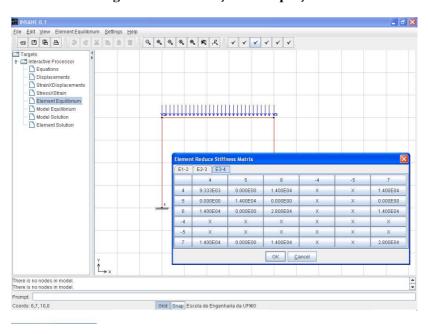


Figura 5 – Matriz de rigidez reduzida de cada elemento do modelo.

A partir do equilíbrio de cada elemento, é gerado o equilíbrio do modelo, este também pode ser consultado, sendo mostrada a contribuição de cada elemento do problema, à rigidez total do modelo. A figura 6 mostra a matriz de rigidez completa do modelo e a visualização da contribuição de cada elemento para formar um dos termos da mesma. A matriz de rigidez reduzida também pode ser consultada. Para esta opção também é possível visualizar as contribuições de cada elemento. O mesmo pode ser aplicado à consulta das matrizes de força completa e reduzida.

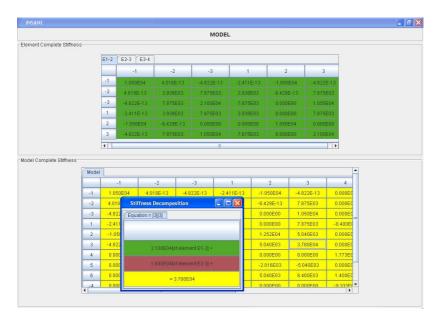


Figura 6 – Matriz de rigidez completa do modelo.

A partir da montagem do equilíbrio do modelo, pode-se solucionar o sistema, para os deslocamentos nodais desconhecidos. Estes podem ser consultados, através da seleção de cada nó do modelo, como ilustra a figura 7.

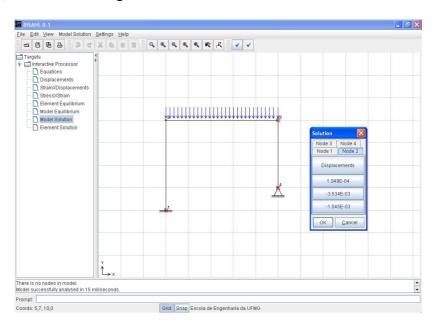


Figura 7 – Solução do Sistema de Equações do modelo.

Uma vez que os deslocamentos nodais são obtidos, as grandezas referentes aos elementos também podem ser calculadas. A solução para cada elemento encontra-se em fase de implementação.

4.2 Modelos de Elementos Planos

Para modelos de elementos planos, o programa contempla a análise de problemas de Estado Plano de Tensão, Estado Plano de Deformação e Axissimétricos. Para estes casos, são disponibilizados todos os recursos descritos acima para elementos de barras: visualização das

equações do modelo, com eventuais intervenções na ordem desta numeração (figura 8), equilíbrio de cada elemento e do modelo. Além dos mesmos recursos disponíveis para os elementos de barra, no caso de problemas planos, o programa possibilita outras interações.

A matriz das funções de forma para cada ponto de Gauss de cada elemento pode ser consultada, como mostra a figura 9.

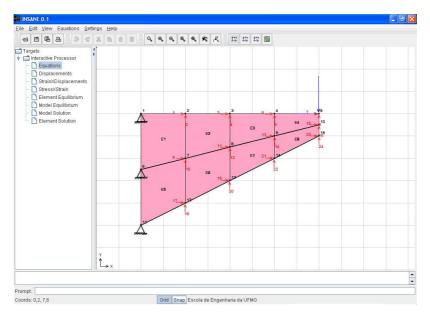


Figura 8 – Numeração das equações para elementos planos.

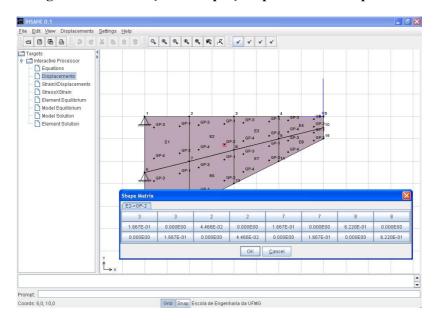


Figura 9 – Matriz das funções de forma para elementos planos.

As componentes de deformação de cada elemento plano podem ser calculadas, a partir do produto da matriz B, que contém as derivadas das funções de forma dos elementos, e dos deslocamentos nodais. O programa possibilita a consulta à matriz B de cada elemento, para cada ponto de Gauss selecionado, como mostra a figura 10.

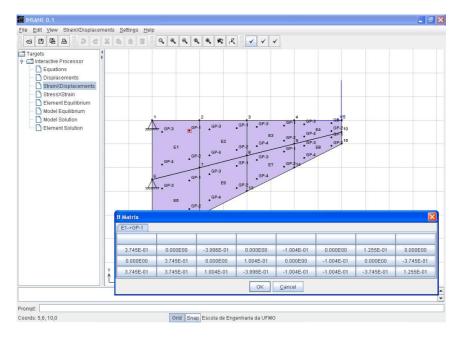


Figura 10 – Matriz das derivadas das funções de forma para elementos planos.

Adotando-se hipóteses relativas ao comportamento do material, as tensões podem ser calculadas através do produto da matriz D, que contém combinações de propriedades do material e do vetor de deformações do elemento. A figura 11 mostra a consulta à matriz D.

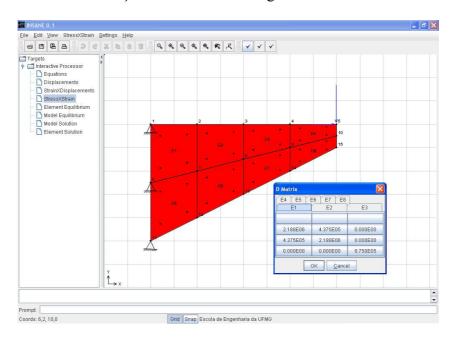


Figura 11 – Matriz das propriedades do material para elementos planos.

Combinando-se as relações acima citadas, e informações do carregamento, obtém-se o equilíbrio de cada elemento, que combinados levam ao equilíbrio do modelo. Estas etapas já foram descritas e ilustradas para elementos de barra, e também são possíveis para elementos planos. A solução do problema também pode ser consultada para elementos planos, através da visualização dos deslocamentos de cada nó do modelo (figura 7).

5. CONSIDERÇÕES FINAIS

O estágio atual de evolução da computação gráfica permite dar saltos qualitativos relevantes no ensino do MEF. É isto que o trabalho aqui apresentado tenta fazer. Motivados pela necessidade de tornar menos árido o ensino do MEF, a ferramenta de análise estrutural mais utilizada na atualidade, os autores projetaram um sistema computacional com este propósito.

A implementação tem como base o INSANE, um sistema computacional para desenvolvimento de modelos do MEF. A expansão do INSANE tratada neste artigo ainda se encontra em fase de implementação. Outros recursos serão disponibilizados para a solução interativa

Espera-se que o programa aqui apresentado diminua as barreiras existentes entre professor e aluno, facilitando o aprendizado dos diversos conceitos do Método dos Elementos Finitos, evitando que os complicadores matemáticos inerentes ao método interfiram negativamente no processo. No próximo semestre letivo, o programa será utilizado como ferramenta auxiliar ao ensino presencial do MEF, tanto para alunos de graduação como de pós-graduação, quando a influência do mesmo no processo poderá ser verificada.

Deseja-se também que o INSANE, base do programa aqui apresentado, seja fomentador do desenvolvimento de novos modelos discretos, evitando o recomeço do processo de implementação e permitindo maior agilidade e criatividade da pesquisa na área.

Agradecimentos

Os autores agradecem ao apoio financeiro em forma de bolsa de mestrado dado pelo Departamento de Engenharia de Estruturas da UFMG, através do CNPq.

REFERÊNCIAS BIBLIOGRÁFICAS

CAMARÃO, C.; FIGUEIREDO, L. Programação de Computadores em Java. LTC, 2003.

FONSECA, F. T.; PITANGUEIRA, R. L.. Um programa gráfico interativo para modelos estruturais de barras. In: *CILAMCE*, Outubro 2004.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design Patterns - Element of Reusable of Object Oriented Software.1995.

GONÇALVES, M. A. B. **Geração de malhas bidimensionais de elementos finitos baseada em mapeamentos transfinitos.** 2004. Dissertação (Mestrado em Engenharia) - Universidade Federal de Minas Gerais, Belo Horizonte.

GOODRICH M. T.; TAMASSIA, R. Estruturas de Dados e Algoritmos em Java, Bookman.

LARMAN, C. Utilizando UML e Padões: Uma Introdução à Análise a ao Projeto Orientado a Objetos. 2000.

LOGAN, D. A First Course in the Finite Element Method. Boston, second edition, 2001.

LOZANO F. Entenda a tecnologia XML. Revista do Linux, dezembro 2003.

SANTOS, R. Introdução à Programação Orientada a Objetos Usando Java. Rio de Janeiro, 2003.

SORIANO, H. L.; LIMA, S. S. **Método de Elementos Finitos em Análise de Estruturas.** UFRJ, 1999.

TIDWELL, D. Mastering XML Transformations. XSLT, 2001.

GRAPHICAL INTERACTIVE COMPUTATIONAL SYSTEM FOR FINITE ELEMENT METHOD TEACHING

Abstract: The article presents an expansion of INSANE (INteractive Structural ANalysis Environment): a computational system for finite element method (FEM) structural analysis discrete models. The expansion that it deals with this article mentions the development of a graphical interface to assist the FEM teaching for structural engineering. The characteristics that become the system friendly to changes, as well as the technological resources that endow it with these characteristics are presented. The use of the computer as a tool to the learning process is argued, as much of the historical point of view how much of the INSANE point of view, emphasizing the possibilities that the INSANE offers for this process. Amongst these possibilities, are pointed those that allow to display the numerical nucleus of the computational system for the user, in a such way that it can visualize the informations about the resolution of FEM models, interact with the system and intervene with the process. Starting from this discussion and available suggestions in literature for characterization of stages of solution of problems through the FEM, it is presented the object-oriented analysis for the graphical interface. The object-oriented design of the expansion is then presented, on the basis of the Model-View-Controller pattern, where the model is separate of its representation, already adopted in the system implementation. Finally, the resources of the new graphical interface are presented through examples and the possibilities of enrichment of the learning process are, then, argued.

Key-words: Finite Element Method, Computer Graphics, Object-Oriented Programming, Engineering Education.