

# Pos3D: Um pós-processador genérico para modelos 3D de elementos finitos

MARCELO TILIO M. CARVALHO<sup>1</sup>  
LUIZ FERNANDO MARTHA<sup>1,2</sup>  
WALDEMAR CELES FILHO<sup>1</sup>

PUC-Rio – Pontifícia Universidade Católica do Rio de Janeiro

<sup>1</sup>TeCGraf – Grupo de Tecnologia em Computação Gráfica

<sup>2</sup>Departamento de Engenharia Civil

Rua Marquês de São Vicente 225, Gávea

22453-900 Rio de Janeiro, RJ, Brasil

tilio,lfm,celes@tecgraf.puc-rio.br

**Abstract.** Pos3D is a interactive graphical post processor for unstructured tri-dimensional meshes, such as finite element models. The program is generic, efficient and independent of any particular analysis code. The first version of Pos3D was developed in 1990 and it has been widely used in several universities all over the country since then. In 1996 we started to develop the new version as part of the co-operation project with Brazilian Oil Company Research Center, CENPES. In this article, we describe the architecture of the new version, its data structure, and its key features.

## 1 Introdução

Existe uma grande variedade de problemas complexos em diversos campos como, engenharia, matemática, física, química, entre outros, que, por não possuírem soluções analíticas, são resolvidos através de simulações numéricas. Uma simulação numérica é uma tentativa de se reproduzir um evento físico no computador. Uma simulação envolve diversas atividades que podem ser, simplificada-mente, agrupadas nas etapas de preparação do modelo, análise numérica (solução das equações diferenciais que descrevem o evento físico) e interpretação e visualização dos resultados.

Um dos métodos numéricos mais utilizado neste tipo de simulação é o método dos elementos finitos [Zienkiewicz 1989]. O método consiste em dividir o domínio do modelo em subdomínios de formas simples (quadriláteros, triângulos, tetraedros, cubos, etc.), nos quais pode-se aproximar seu comportamento através de funções conhecidas. Estes subdomínios são chamados de elementos finitos e os vértices destes elementos são chamado de nós. A forma e o tipo dos elementos depende do tipo de problema que se deseja resolver. A coleção destes subdomínios é chamada de malha de elementos finitos (Fig. 1).

Os resultados provenientes de uma análise feita por elementos finitos são, muitas vezes, complexos e difíceis de serem interpretados e podem representar um grande volume de dados. Deste modo, a utilização de ferramentas gráficas para pós-processamento de modelos de elementos finitos se tornou fundamental em simulações computacionais, pois oferecem recursos para se visu-

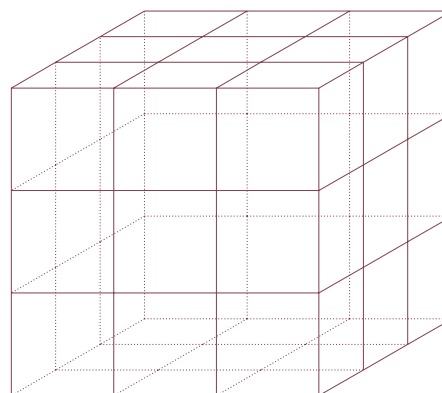


Figure 1: Exemplo de uma malha de elementos finitos composta de cubos.

alizar um resultado de diversas formas diferentes, o que ajuda na sua interpretação.

As informações normalmente disponíveis para visualização consistem da geometria do modelo, definida em termos da malha utilizada na análise, dos atributos associados a esta malha (cargas, material, etc.) e dos resultados da análise, que são representados também em relação aos nós e elementos da malha.

Um pós-processador deve ser genérico, o que implica em ser independente do programa utilizado na análise numérica, representar qualquer tipo de elemento finito e qualquer tipo de resultado. Os resultados são, normalmente, de 3 tipos: i) resultados escalares, como temperatura; ii) resultados vetoriais, como deslocamen-

tos ou velocidades; iii) resultados tensoriais, como campo de tensões.

As tarefas realizadas em um pós-processamento são baseadas, predominantemente, na interação com o usuário e em uma manipulação intensa da estrutura de dados que representa o modelo. Desta forma, um pós-processador deve possuir estruturas de dados que possibilitem a implementação de algoritmos eficientes.

Nos programas de elementos finitos convencionais utiliza-se uma estrutura de dados, que consiste de uma tabela de conectividade dos elementos, indicando a posição relativa dos elementos com respeito aos nós da malha e de uma tabela com as coordenadas dos vértices, que constituem as informações geométricas disponíveis. Esta estrutura é inadequada para as tarefas de pós-processamento, pois não existem praticamente informações topológicas, necessárias em vários algoritmos de visualização e geração de imagens realistas.

A utilização de uma primitiva que represente explicitamente as superfícies melhora bastante esta situação. Sendo assim, poderia-se utilizar uma simples lista de faces, que se adequaria para a visualização do modelo, mas esta estrutura não satisfaz a necessidade de obtenção das relações de adjacência topológicas necessárias no pós-processamento. Com o intuito de solucionar estes problemas, foi concebida uma estrutura de dados baseada na fronteira dos elementos [Celes 1991], representada pela primitiva face. Esta estrutura de dados constitui a parte central do Pos3D, um pós-processador para elementos finitos tri-dimensionais. Descreve-se a seguir suas principais características.

## 2 Pos3D

O Pos3D é um pós-processador gráfico interativo para modelos tridimensionais de elementos finitos que trata de modelos sólidos ou de superfícies. O programa faz o pós-processamento de modelos com qualquer tipo de elemento finito e considera qualquer tipo de análise. A descrição da malha e dos resultados é feita através de um arquivo padrão, chamado de arquivo neutro. Este arquivo tem um formato extensível e o programa pode, facilmente, incorporar novos elementos, implementando apenas as funções que diferenciam o novo elemento dos já implementados.

O Pos3D vem sendo largamente utilizado desde a sua primeira versão, desenvolvida em 1990 [Celes 1990] na plataforma DOS. Esta versão utilizava um sistema gráfico bi-dimensional, de modo que todos os algoritmos para geração de imagens realistas (linhas escondidas, superfícies escondidas, sombreado, etc.), além dos algoritmos de projeção do modelo 3D para a tela, tinham que ser implementados pelo próprio programa. A partir de 1996, iniciou-se o desenvolvimento da nova versão, que faz parte do convênio entre o TeCGraf e o Centro

de Pesquisa da PETROBRAS. Nesta nova versão, foram retirados todos os algoritmos de geração de imagens realistas e de manipulação do modelo. Estas funcionalidades são realizadas por bibliotecas de uso geral, desenvolvidas no TeCGraf, que têm como característica fundamental o fato de serem portáteis, no sentido que permitem às aplicações "rodarem" nas plataformas DOS, Unix, Windows e Macintosh. Deste modo, o novo código do Pos3D se resume a um interpretador do arquivo neutro e a módulos que gerenciam a estrutura de dados.

Com a utilização de bibliotecas de uso geral, a arquitetura atual do Pos3D pode ser dividida em três partes (Fig. 2): i) bibliotecas básicas, responsáveis pela interface com o usuário, desenho e manipulação do modelo; ii) módulos que implementam a estrutura de dados, responsáveis pela representação do modelo de uma forma eficiente e; iii) bibliotecas específicas, responsáveis pelos diversos algoritmos de visualização.

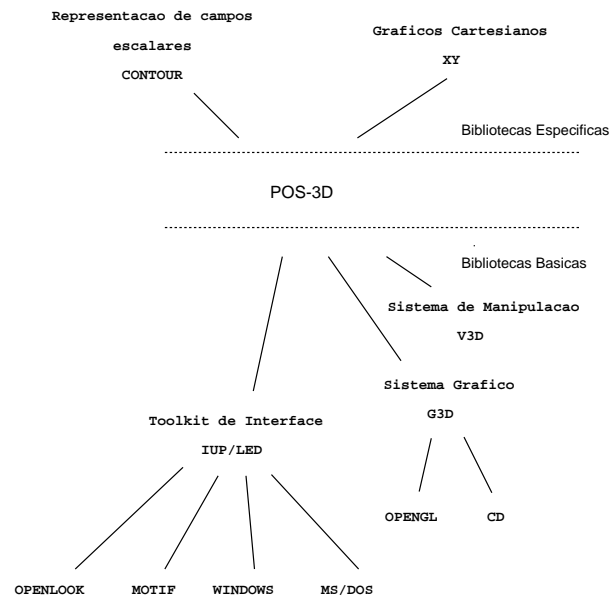


Figure 2: Arquitetura do Pos3D

### 2.1 Bibliotecas básicas

A interface com o usuário do Pos3D foi desenvolvida utilizando-se o *toolkit* IUP/LED [Levy 1996]. Este *toolkit* possui uma linguagem declarativa que permite a criação e especificação do *layout* dos objetos de interface, através de uma sintaxe bastante simples. O uso desta linguagem possibilita que alterações sejam feitas facilmente, sem necessidade de se "recompilar" o código do programa.

Todos os desenhos do Pos3D são feitos através do G3D [Tecgraf 1997], uma biblioteca gráfica que cria uma camada de abstração para a exibição de primitivas gráficas tridimensionais e bidimensionais. Esta biblioteca

foi projetada para ser utilizada juntamente com o *toolkit* IUP/LED e sobre sistemas gráficos como o OpenGL [Neider93], CD (sistema gráfico projetado no TeCGraf [Tecgraf 1997]), entre outros. A ligação com o IUP/LED se faz através do elemento de interface *canvas*. O G3D utiliza um modelo de câmera para a representação do modelo projetado na tela.

Além do desenho de primitivas, o G3D oferece recursos para a definição do modelo de iluminação, onde as cores são automaticamente calculadas, baseadas em fontes de luz e, na geometria e propriedades dos materiais do modelo. Estes dados são fornecidos como entrada para as equações do modelo de iluminação, que são baseadas no modelo definido pelo OpenGL. Este modelo permite ao usuário controlar o grau de realismo do modelo geométrico exibido.

Para facilitar a implementação de aplicações clientes, que tenham somente valores das cores de suas entidades como atributos de visualização, o G3D implementa um modelo simples de iluminação. Este modelo, analisando apenas a cor corrente, a normal à face e a direção de visualização, simula uma única luz presa à câmera (posição referente ao observador do modelo), permitindo que o objeto possa ser visualizado de qualquer posição. Atualmente, no Pos3D, somente está implementado este modelo simples,

Para a manipulação do modelo na tela (translação, rotação, escala, etc.) o Pos3d utiliza uma terceira biblioteca, V3D [Castier 1994], que gerencia um modelo de câmera, baseado no modelo implementado pelo OpenGL.

O modelo de câmera proporciona uma maneira natural para se controlar a visualização de um ambiente ou de um objeto tridimensional. No V3D, o objeto é manipulado em relação a eixos cujas direções são paralelas à tela, ou seja, eixo horizontal, vertical e perpendicular (apontando para fora da tela). Deste modo, o movimento horizontal do *mouse* na tela é associado a uma rotação da câmera em torno do eixo vertical, e o movimento vertical com uma rotação em torno do eixo horizontal. Um movimento do *mouse* para a esquerda resulta em uma rotação do objeto na tela no mesmo sentido, o mesmo acontecendo para a direita, e para os movimentos verticais. O centro de rotação é o centro da caixa envolvente do objeto. O usuário pode, no entanto, redefinir interativamente este ponto. Esta biblioteca oferece, ainda, as funções normalmente associadas à visualização de objetos gráficos, como escala, *zoom*, etc.

## 2.2 Estrutura de dados

Conforme mencionado anteriormente, um pós-processador requer uma estrutura de dados que forneça, de maneira eficiente, informações sobre a topologia da malha. Além disso, a estrutura de dados deve utilizar uma primitiva que represente explicitamente as superfícies do modelo.

Um outro aspecto relevante diz respeito às propriedades de uma malha de elementos finitos.

Em uma malha, cada elemento representa uma sub-região fechada na qual sua fronteira é contínua. A interseção de 2 elementos consiste de um vértice, uma aresta ou uma face e a união de todos os elementos é aproximadamente igual ao domínio do modelo. Isto significa que não é possível existir elementos finitos delimitados por mais de uma fronteira, ou seja, um elemento finito não pode ser oco, e as faces que compõem a fronteira dos elementos finitos não podem ser furadas.

De acordo com estas considerações, idealizou-se uma estrutura de dados [Celes 1991], inspirada nas estruturas topológicas utilizadas na área de modelagem geométrica [Weiler 1986], no sentido de atender às necessidades de um pós-processador de maneira eficiente, com o menor uso possível de memória. A estrutura de dados é baseada na fronteira dos elementos finitos, representada pela lista de faces que os delimitam. A partir desta lista de faces de cada elemento, geram-se duas listas globais, uma com as faces externas e outra com as faces internas do modelo.

A lista de faces externas permite a representação da superfície externa do modelo e é utilizada nos algoritmos de *rendering*. A lista de faces internas permite a representação de toda a malha, necessária para a visualização de resultados no seu interior. Optou-se por não representar explicitamente as arestas do modelo, sendo as faces definidas por uma sequência de vértices. Esta opção permite uma economia razoável do espaço de memória necessário, sem afetar em demasiado a eficiência das operações de pós-processamento. As restrições apresentadas por uma malha de elementos finitos, mencionadas acima, dispensam a representação explícita de entidades topológicas equivalentes aos elementos de cascas e ciclo.

A maneira como as listas globais de faces são obtidas constitui um dos pontos principais da estrutura de dados. A partir da incidência de cada elemento finito, lida do arquivo neutro, gera-se um conjunto de faces orientadas que representam a fronteira do elemento. Este procedimento equivale a uma "explosão" do elemento, sendo que as faces guardam uma orientação de modo que sua normal aponta para o lado externo do elemento (Fig. 3).

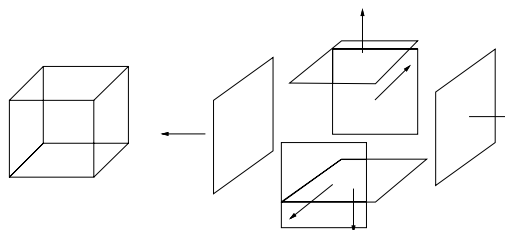


Figure 3: Explosão dos elementos.

Executando-se este procedimento para todos os elementos da malha obtém-se uma lista global de faces do modelo, a partir da qual são extraídas as faces externas e internas. Uma face é interna quando é definida por dois elementos. Para identificação destas faces basta verificar a existência de faces geometricamente equivalentes, isto é, possuidoras do mesmo conjunto de vértices, porém com incidência inversa, já que cada uma das faces origina-se da "explosão" de elementos distintos (Fig. 4). É esta estratégia que permite ao Pos3D tratar qualquer tipo de elemento finito, pois basta se escrever as funções que forneçam as faces que delimitam o elemento.

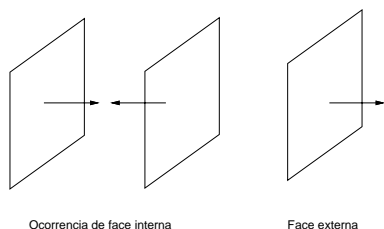


Figure 4: Definição de face interna e externa.

Com o intuito de permitir a visualização de resultados no interior da malha, a estrutura de dados implementada suporta a realização de cortes no modelo original. Quando um corte é efetuado, novas faces são criadas, automaticamente, para restaurar a consistência dos elementos cortados. As listas de faces são atualizadas, de modo a representar o modelo cortado (Fig. 5).

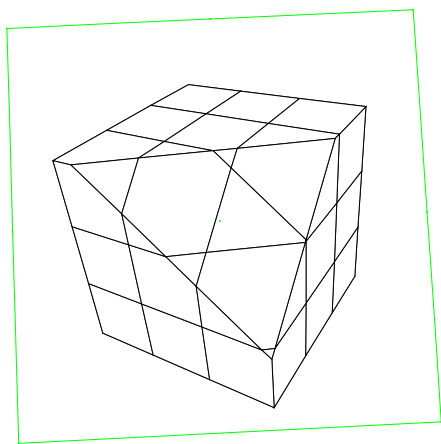


Figure 5: Definição de um corte no modelo.

### 2.3 Bibliotecas Específicas

Os diferentes tipos de visualização disponíveis no Pos3D são implementados através de bibliotecas de uso geral, do mesmo modo que as bibliotecas básicas. Na versão atual do Pos3D existem 2 bibliotecas, uma para representação

gráfica de valores escalares e outra para criação de gráficos cartesianos.

A biblioteca para criação de gráficos, chamada XY, disponibiliza diversas funcionalidades (formato, legenda, estilos das curvas, etc.) para o traçado de gráficos cartesianos.

A biblioteca para representação dos resultados escalares, chamada CONTOUR [Martha 1996], implementa técnicas de contorno baseada em isovalores [Gallagher 1995]. Esta biblioteca disponibiliza quatro tipos de visualização diferentes. Na superfície do modelo, pode-se obter isolinhas (Fig. 6), que são linhas com um valor constante ao longo de uma superfície, ou isofaixas (Fig. 7), que correspondem às regiões entre duas isolinhas. Estas duas formas de representação, no entanto, não permitem a inspeção de resultados no interior do modelo, fundamental em determinados tipos de simulação. Para estes casos pode-se obter as isosuperfícies do modelo (Fig. 8), que representam, análogo às isolinhas, valores constantes dentro de um volume, ou isovolumes (Fig. 9), que correspondem às regiões entre duas isosuperfícies e limitada pelo contorno do modelo.

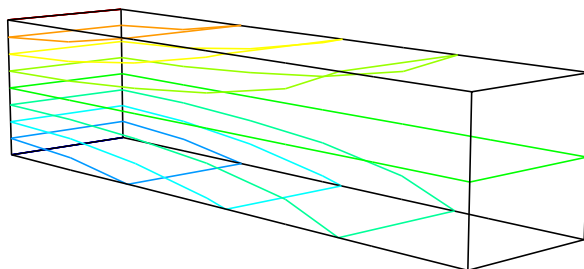


Figure 6: Contorno de IsoLinhas de uma viga.

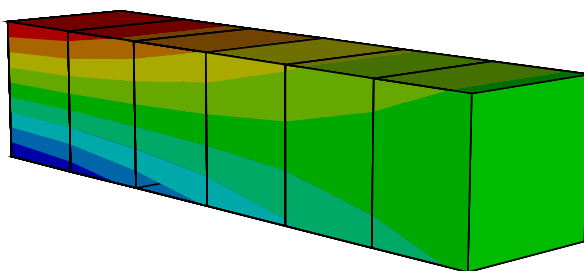


Figure 7: Contorno de IsoFaixas de uma viga.

### 2.4 Funcionalidade

O Pos3D permite obter informações qualitativas e quantitativas da malha de elementos finitos e dos resultados associados a esta malha. Os resultados podem estar associados aos nós ou aos elementos da malha. Quando associados aos nós, os resultados são contínuos ao longo da malha. No caso de serem associados aos elementos, os

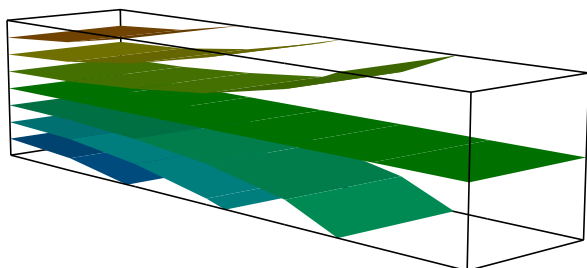


Figure 8: Contorno de IsoSuperfícies de uma viga.

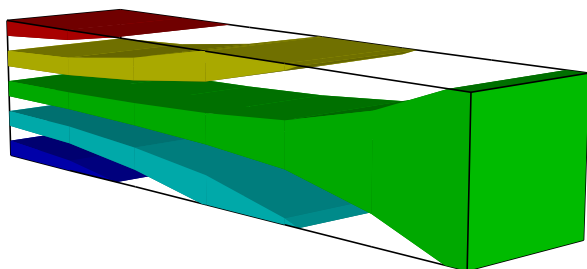


Figure 9: Contorno de IsoVolumes de uma viga.

resultados são locais e descontínuos e podem ser fornecidos tanto nos nós do elemento, quanto em pontos interiores, chamados de pontos de Gauss. Neste caso, o Pos3D faz uma extrapolação para os nós do elemento. Os resultados associados aos elementos podem ser vistos de forma suavizada ou descontínua.

As informações qualitativas da malha podem ser obtidas através de funções que desenharam a malha de diversas formas: i) malha toda, transparente ou de forma opaca (remoção das linhas escondidas); ii) somente as faces externas da malha; iii) somente as arestas do contorno da malha; iv) malha externa com efeito de iluminação. Podem ser obtidas, também, informações quantitativas, que são fornecidas através de funções de consulta que permitem ao usuário obter informações referentes a um nó ou elemento selecionado (incidência, material, coordenadas etc.), além do número total de elementos e nós.

Em relação aos resultados, uma análise qualitativa pode ser feita através da representação gráfica dos mesmos. Resultados escalares são mostrados através de uma gradação de cores, onde cada cor está associada a uma escala que contém valores do campo escalar. Resultados vetoriais, como os deslocamentos de uma estrutura ou seus modos de vibração, são representados através do desenho da configuração deformada do modelo. Resultados tensoriais, como as tensões em uma estrutura, são de difícil visualização e, por isso, são representados através de suas componentes, como se fossem resultados escalares.

### 3 Conclusões

O Pos3D representa uma ferramenta poderosa para pós-processamento de modelos de elementos finitos, tendo como principal característica a simplicidade de utilização. Sua estrutura de dados permite a representação de qualquer tipo de elemento finito e a estratégia de se utilizar um arquivo padrão para entrada dos dados faz com que o programa seja independente de qualquer código específico de análise numérica.

Em relação a arquitetura do Pos3D, pode-se citar algumas vantagens devido à utilização de bibliotecas de uso geral do TeCGraf. Estas bibliotecas possibilitam uma redução no tamanho do código dos programas, assim como aumentam o nível de reuso de *software*. Isto facilita a manutenção do programa e a incorporação de novas funcionalidades.

### 4 Agradecimentos

Este trabalho faz parte de um projeto de pesquisa do TeCGraf, Grupo de Tecnologia em Computação Gráfica da PUC-Rio, com o CENPES/Petrobras, responsável por seu financiamento. O primeiro autor gostaria de agradecer à FAPERJ, pelo financiamento de sua bolsa de estudos. Este trabalho apresenta resultados obtidos no âmbito do projeto PROTEM/GEOTEC, financiado pelo CNPq.

### 5 Referências

- B. Castier, L. F. Martha, M. Gattass, "Uma taxonomia para Manipulação Interativa e Visualização de Objetos 3D" *Anais do VII Sibgrapi, Curitiba, Brasil* (1994) 149–156.
- W. Celes Filho "Um Pós processador genérico de elementos finitos sólidos baseado na representação da fronteira dos elementos" *Tese de Mestrado, Departamento de Engenharia Civil, Puc-Rio* (1990).
- W. Celes Filho, L. F. Martha, M. Gattass, "Uma estrutura de dados eficiente para pós-processamento de análise de elementos finitos sólidos" *Anais do XI Congresso Brasileiro de Engenharia Mecânica, São Paulo, ABCM* (1991) 105–108.
- R. S. Gallagher, *Computer Visualization – Graphics Techniques for Scientific and Engineering Analysis*, CRC Press, 1995.
- C. H. Levy, L. H. de Figueiredo, M. Gattass, C. J. P. de Lucena, D. D. Cowan, "TUP/LED: a portable user interface development tool" *Software: Practice & Experience* (1996), vol.26 - 7, 737–762.
- L. F. Martha, M. Tilio M. Carvalho, R. B. Seixas, "Volume Contouring of Generic Unstructured Meshes" *Anais do IX Sibgrapi, Caxambu, M.G.* (1996) 211–218. URLib rep.: <dpi.inpe.br/ambro/1998/04.24.10.35>.
- J. Neider et al, *OpenGL Programming Guide: the Official Guide Learning OpenGL, release 1*, Addison-Wesley

Publishing Company, 1993.

TeCGraf, "Manuais de Referência: <http://www.tecgraf.puc-rio.br/manuais>" 1997

O. C. Zienkiewicz, R. L. Taylor. *The Finite Element Method, Volume 1: Basic Formulation and Linear Problems*, Forth Edition, McGraw-Hill, 1989.

K. Weiler, "Topological Structure for Geometric Modeling", *Ph.D. Thesis*, Rensselaer Polytechnic Institute, Troy, New York, 1986.