



DESENVOLVIMENTO DE UMA CLASSE DE BASE DE OBJETOS PARA ELEMENTOS FINITOS POSICIONAIS

Marcio Takashi Uyeno (PIC / Uem), Wilson Wesley Wutzow (Orientador), e-mail: wwwutzow@uem.br

Universidade Estadual de Maringá / Centro de Tecnologia / Maringá, PR.

Área: Engenharia Civil. Subárea: Estruturas

Palavras-chave: MEF, C++, Python.

Resumo:

O uso de ferramentas numéricas computacionais destinadas à previsão do comportamento de elementos estruturais sob diferentes condições de solicitação, vem sendo, ao longo dos últimos anos, cada vez mais utilizadas na área de projetos estruturais e estudo por pesquisadores no ambiente acadêmico. Isto se deve ao fato destas ferramentas proporcionarem uma ampla gama de análises possíveis de serem efetuadas, as quais podem incluir problemas de grande complexidade. O presente trabalho consistiu na elaboração de um pré-processador para um programa computacional de elementos finitos posicionais, bem como a passagem de tal programa da linguagem computacional C++ para *Python 2.7*, afim avaliar a viabilidade, quanto ao tempo de execução e qualidade dos resultados, da utilização do *Python* para elaboração de ferramentas numéricas computacionais para o método dos elementos finitos (MEF). Os resultados obtidos permitiram uma análise qualitativa dos resultados em escalas de cores através do *ParaView* além de permitir a comparação no tempo de processamento de dados entre o programa em C++ e *Python*, sendo que o primeiro apresentou maior velocidade neste quesito.

Introdução

A utilização de métodos computacionais na análise estrutural é uma atividade relativamente antiga e remonta os anos 1960. Tal como ocorreu com a mecânica dos fluidos o Método das Diferenças Finitas teve grande impulso no início dos desenvolvimentos computacionais, porém com o tempo, o MEF se destacou como alternativa viável e mais versátil para a solução de problemas de análise estrutural (ZIENKIEWICZ E TAYLOR, 2000). O estudo de MEF e suas potencialidades estão em constante



desenvolvimento, sendo que atualmente, o MEF isoparamétricos de alta ordem também estão sendo utilizados na análise de placas e cascas com auxílio de métodos de integração numérica, neste quesito pode-se referir a (CODA e PACCOLA, 2007). (BONET ET AL., 2000) e (CODA, 2003) introduzem a formulação posicional baseada no MEF para análise não linear geométrica de sólidos e estruturas. Esta utiliza posições finais e inclinações finais de vetores inicialmente normais como parâmetros nodais ao invés de deslocamentos e rotações, sendo bastante robusta e de simples implementação.

Materiais e métodos

Neste projeto, utilizou-se as linguagens computacionais *C++* e *Python 2.7*, e também suas respectivas bibliotecas disponíveis para *Linux* e *Windows*, respectivamente. Sendo que o desenvolvimento do programa em *Python* utilizou-se a plataforma *Eclipse*. Para a geração de insumos para o pré-processador, utilizou-se o programa *ANSYS*, para visualização dos resultados gerados pelo programa de MEF posicional, utilizou-se o programa *ParaView*.

O método utilizado seguiu o seguinte sequenciamento: transcrição do programa em *C++* para *Python 2.7*; processamento de um exemplo pré-existente em ambos os programas para verificação da confiabilidade dos resultados; geração de dados através do *ANSYS* para analisar seu padrão de saída de dados; elaboração do pré-processador baseado no padrão de saída do *ANSYS*; processamento de dados, após o pré-processamento, em ambos programas (em *C++* e *Python*) para comparação de resultados e tempo de execução; análise dos resultados utilizando o *ParaView* e comparação com os resultados do *ANSYS*.

Resultados e Discussão

Através do pré-processamento dos dados gerados pelo programa desenvolvido neste projeto, obteve-se uma maior malha para análise da estrutura de uma viga engastada na esquerda e carga aplicada na extremidade direita e ao mesmo tempo possibilitando uma análise qualitativa (através das escalas de cores) dos resultados de deslocamento vertical e deformação longitudinal da viga. As Figuras 01 e 03 referem-se a resultados gerados pelo *ANSYS* (antes do pré-processamento), enquanto as Figuras 02 e 04 referem-se a resultados visualizados no *ParaView* após o pré-processamento de dados, sendo os dados qualitativamente semelhantes, como se pode observar nas imagens, mas com uma mudança mais suave entre as cores devido à nova malha gerada.



Os resultados obtidos após o pré-processamento, tanto no programa em C++ quanto no programa em *Python*, foram os mostrados nas Figuras 02 e 04, entretanto, no que se refere à tempo de processamento o programa em C++ mostrou um desempenho muito maior através de um tempo de processamento muito inferior ao outro.

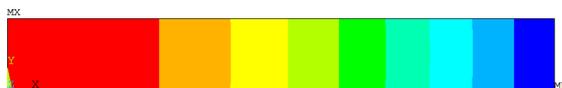


Figura 01 –Deslocamento na direção vertical (Eixo Y), para análise do ANSYS.



Figura 02 – Deslocamento na direção vertical (Eixo Y), para análise após o pré-processamento e processamento através do programa que utiliza MEF posicional.

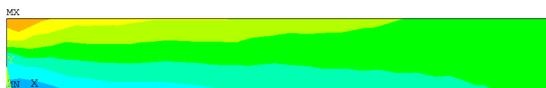


Figura 03 – Deformação Longitudinal no eixo da barra (Eixo X), para análise do ANSYS.



Figura 04 – Deformação Longitudinal no eixo da barra (Eixo X), para análise após o pré-processamento e processamento através do programa que utiliza MEF posicional.

Conclusões

Após o desenvolvimento do pré-processador, da criação de classes para o método de elementos finitos posicionais e passagem do programa, desenvolvido inicialmente em C++ para o método apresentado, para *Python* 2.7, pôde-se observar que os resultados foram satisfatórios para uma análise qualitativa, através dos resultados obtidos visualizados através do *Paraview*. Quantitativamente falando, os dados não puderam ser comparados com os resultados do ANSYS, por se tratarem de grandezas diferentes, ou seja, enquanto o ANSYS resultou em deslocamentos e deformações no estado indeformado da viga (sem uma variação da seção ao longo do tempo), o programa de MEF posicional resultou em deslocamentos e deformações no estado deformado da viga. Com relação ao tempo de processamento dos dados dos programas de MEF posicional em C++ e *Python*, pôde-se concluir, que neste primeiro momento, a programação em *Python* foi inviável para a elaboração do programa



estrutural referido por levar horas para o processamento de um exemplo que em C++ levava poucos minutos, no entanto, o presente estudo por si só não é suficiente para descartar o *Python* para o desenvolvimento de programas estruturais, pois são necessários estudos mais aprofundados na linguagem computacional para elaboração de programas mais robustos e também a verificação da diferença de capacidade de processamento entre as plataformas *Linux* e *Windows*. Sendo assim, os resultados obtidos foram satisfatórios para esta pesquisa inicial, e que servirão de base e impulso para estudos futuros neste tema.

Agradecimentos

Agradeço ao meu professor orientador Wilson Wesley Wutzow pela orientação, paciência e ensinamentos.

À Universidade Estadual de Maringá através de seu corpo docente e infraestrutura.

Aos meus pais que sempre me apoiaram em minhas decisões e sempre presaram pelo melhor para os meus estudos.

Referências

BONET, J. et al. Finite element analysis of air supported membrane structures. **Computer Methods In Applied Mechanics and Engineering**, v. 190, n. 5, p. 579-595, 2000.

CODA, H. B. **Análise não linear geométrica de sólidos e estruturas: Uma formulação posicional baseada no MEF**. 2003. 168 (Tese para concurso de professor titular). SET, USP, São Carlos.

CODA, H. B.; PACCOLA, R. R. An alternative positional FEM formulation for geometrically non-linear analysis of shells: Curved triangular isoparametric elements. **Computational Mechanics**, v. 40, n. 1, p. 185-200, Jun 2007.

CODA, H. B.; PACCOLA, R. R. Improved finite element for 3D laminate frame analysis including warping for any cross-section. **Applied Mathematical Modelling**, v. 34, n. 4, p. 1107-1137, 2010.

ZIENKIEWICZ, O. C.; TAYLOR, R. L. **The Finite Element Method, v1: The Basis**. Butterworth-heinemann Linacre house, 2000. 689