Mecánica Computacional Vol. XXI, pp. 3110–3120
S. R. Idelsohn, V. E. Sonzogni and A. Cardona (Eds.)
Santa Fe–Paraná, Argentina, October 2002

PRÉ E PÓS PROCESSADOR PARA UTILIZAÇÃO EM PROGRAMAS DE ELEMENTOS FINITOS .

José G. M. Soler[I], José P. B. M. de Almeida[II], Paulo de M. Pimenta[III]

[I]PUC-MINAS *Campus* Poços de Caldas
Av. Padre Francis Cletus Cox 1661, CEP 37701-056
e-mail: jgmsoler@yahoo.com

[II]IST – Instituto Superior Técnico – Lisboa
Av. Rovisco Pais 1049-001, Lisboa, Portugal
Universidade Técnica de Lisboa
e-mail: moitinho@civil.ist.utl.pt

[III]Universidade de São Paulo
Escola Politécnica da Universidade de São Paulo
e-mail: ppimenta@usp.br

Palavras-chave: pré-processador, pós-processador, elementos finitos, mayavi, Python.

Resumo: *Em uma análise utilizando qualquer tipo de elemento finito, existem sempre partes comuns de entrada e saída de dados, em que são fornecidas a localização dos pontos e as incidências dos elementos e onde são obtidos os deslocamentos e os esforços.*
A sua visualização permite validar os dados de entrada e pode ajudar a verificar e interpretar os resultados produzidos pela análise.
Pensando nisso, apresentamos neste trabalho como utilizar um software de domínio público, escrito em PYTHON [8], que permite conferir os dados de entrada, visualizando os elementos, e os dados de saída, sendo ainda que é possível por meio de recursos implementados, girar a estrutura, ampliar, cortar, deslocar de acordo com os valores encontrados na análise, selecionar pontos, células, enfim, obter tudo aquilo que é necessário para uma boa interpretação da estrutura, utilizando apenas um arquivo texto cujo formato será apresentado no final deste trabalho.
Espera-se com isso contribuir para uma melhor e mais fácil utilização e interpretação das capacidades dos programas de elementos finitos por parte de seus programadores e usuários.

1 INTRODUÇÃO

Neste trabalho será apresentado uma abordagem que permite de uma forma simples, realizar uma verificação dos dados de entrada de programas que se utilizem do método dos elementos finitos, e dos resultados encontrados numa análise, por meio de sua visualização em um ambiente gráfico.

Os dados a fornecer ao software de domínio público, devem ser guardados em um arquivo tipo texto, facilmente obtido por qualquer tipo de linguagem de programação.

A apresentação dos resultados e dos recursos disponíveis tem por base um programa para a análise não-linear física e geométrica de pórticos espaciais de concreto armado, de acordo com Soler, J.G.M.[5,6].

Alguns módulos e filtros do software tiveram que sofrer pequenas alterações para que o mesmo pudesse ser adaptado às nossas expectativas.

O software pode ser obtido no endereço http://mayavi.sourceforge.net/[4] e os módulos e filtros modificados ou adicionados no endereço eletrônico do pesquisador, indicado no início do trabalho.

2 A CONCEPÇÃO DO TRABALHO

Logicamente a análise de uma estrutura requer sempre uma boa teoria e a sua boa aplicação, contudo nunca se deve esquecer que uma boa apresentação dos resultados encontrados é fundamental.

Pensando nisso, resolveu-se adaptar um software de domínio público que pudesse auxiliar na visualização dos resultados encontrados e dos dados gerados. Optou-se pelo Mayavi[7].

Como já se dispunha de uma ambiente praticamente desenvolvido, resolveu-se então estudá-lo e observar quais seriam as modificações que precisariam ser introduzidas para que o mesmo viesse a ser utilizado para atingir os nossos objetivos.

3 ARQUIVO BASE

Em geral tem-se as coordenadas nodais, que logicamente em uma análise de pórtico plano possuem duas ordenadas e no caso de um pórtico espacial possuem três ordenadas.

Nesse caso, como já referido anteriormente, se trata da apresentação de um exemplo de pórtico espacial.

Inicialmente deve-se identificar o arquivo como sendo um arquivo relacionado a esse software. Para que isso seja possível é necessário que na primeira linha do arquivo, começando na primeira coluna (observe bem esse ítem, pois no caso de um programa em FORTRAN[1], deverá ser uma escrita formatada, pois se não for ele escreverá na segunda coluna, porque reserva a primeira coluna para controle da impressora), esteja escrito: #vtk DataFile Version 3.0, onde além de identificar o arquivo, também identificamos a versão que estamos utilizando.

Na próxima linha pode-se atribuir um título a esse arquivo que poderá ter um máximo de 256 caracteres. Na terceira linha deve-se identificar se os dados serão do tipo ASCII ou Binary. Na Quarta linha deve-se identificar a topologia e geometria, que no nosso exemplo será do tipo UNSTRUCTURED_GRID. O software suporta cinco tipos de combinação de topologia e de geometria que são: "STRUCTURED_POINTS, STRUCTURED_GRID, UNSTRUCTURED_GRID, POLYDATA e RECTILINEAR_GRID".

Caso a estrutura possua uma simetria e os elementos apresentem espaçamentos iguais, pode-se utilizar o tipo STRUCTURED_POINTS. No caso de espaçamentos diferentes, pode-se utilizar o STRUCTURED_GRID.

Na frente da topologia/geometria deve constar o comando DATASET.

No exemplo tem-se um arquivo com a seguinte estrutura inicial:

#vtk DataFile Version 3.0
Pórtico espacial
ASCII
DATASET UNSTRUCTURED_GRID.

A seguir será fornecido o número de pontos, com a palavra POINTS seguida do número de pontos e nesse caso a declaração float.

No exemplo ter-se-á a seguinte linha no arquivo:

POINTS 44 float **.**

A seguir são fornecidas as ordenadas dos pontos.

Após os pontos, e aqui devemos tomar um cuidado especial, pois o primeiro ponto sempre será designado pelo número 0 (zero), devemos indicar a quantidade de células e o número dos pontos que irão compor essas células.

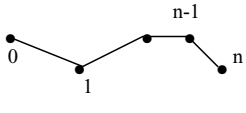
Tem-se vários tipos de células disponíveis, portanto faz-se necessário indicar o tipo de célula que se pretende utilizar. Pode-se ter num mesmo arquivo vários tipos de células.

Os tipos de células são as seguintes:

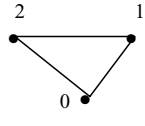
VTK_VERTEX (=1)

VTK_POLY_VERTEX (=2)

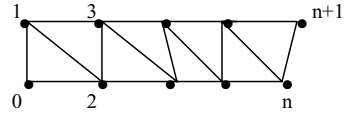
VTK_LINE (=3)



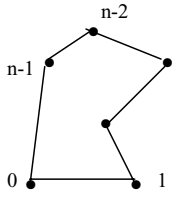
VTK_POLY_LINE (=4)



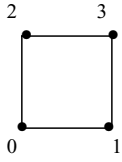
VTK_TRIANGLE (=5)



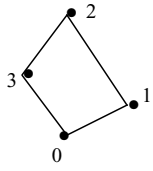
VTK_TRIANGLE_STRIP (=6)



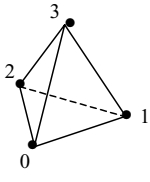
VTK_POLYGON (=7)



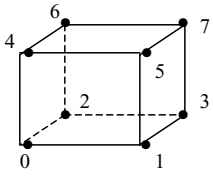
VTK_PIXEL (=8)



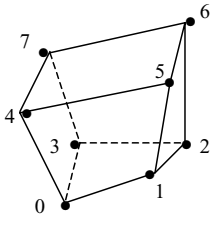
VTK_QUAD (=9)



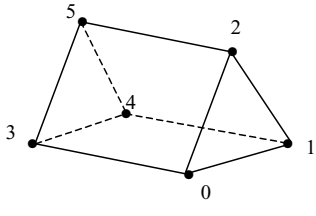
VTK_TETRA (=10)



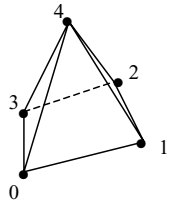
VTK_VOXEL (=11)



VTK_HEXAHEDRON (=12)



VTK_WEDGE (=13)



VTK_PYRAMID (=14).

Como se pode em uma mesma estrutura, ter vários tipos de células, há necessidade da declaração do tipo de célula, e a seguir a quantidade de pontos que estas células irão gerar.

Neste exemplo somente se utilizará células do tipo VTK_VOXEL (=11) que possuem 8 pontos. Na análise que será realizada, se utilizará a discretização da seção transversal em várias camadas ao longo de suas duas dimensões, o que facilita a obtenção dos pontos da seção transversal. No caso da integração por pontos de Lobato de acordo com Pereira, E.M.B.R[2,3], isso deve ser feito à parte.

No caso onde se quer observar a representação dos eixos das peças, sem as dimensões de suas seções, pode-se utilizar a célula do tipo 3, ou seja, a VTK_LINE.

Percebe-se que o número de células é função da discretização da seção transversal do elemento. Sejam as figuras abaixo:

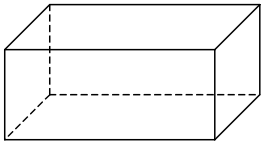


Figura 1

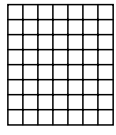


Figura 2

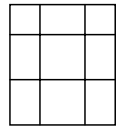


Figura 3

onde a Figura 1 seria o elemento a ser discretizado. Se a seção desse elemento fosse discretizada como mostra a Figura 2 ter-se-iam 56 células. Já no caso da discretização ser realizada de acordo com a Figura 3, ter-se-iam 9 células. Todas as células nesse caso formadas por 8 pontos.

A declaração para identificação das células seria:

*CELLS nc 9*nc,*

onde nc é o número total de células de nossa estrutura e 9*nc o número total de declarações da quantidade de pontos da célula e os números das ordenadas que formam essa célula.

Como vamos utilizar uma célula de oito pontos, temos que realizar a seguinte declaração:

8 1 0 3 2 5 4 7 6,

que informa que a célula é formada por oito pontos, cujos vértices correspondem aos pontos *1,0,3,2,5,4,7 e 6.*

A seguir devemos informar o tipo de célula. O comando para essa declaração é *CELL_TYPES nc.*

Colocamos então o número do tipo de célula, que no nosso exemplo é o número 11. Esse número é colocado nc vezes no arquivo.

Com isso já temos nossa estrutura definida e determinada. No caso de não se ter a seção transversal definida pode-se adotar uma seção qualquer, somente para efeito de visualização inicial do problema.

Percebe-se que todos os dados que foram fornecidos até o momento, são dados que sempre encontramos nos programas de análise estrutural pelo método dos elementos finitos, ou por métodos semelhantes.

Pode-se agora com esse arquivo realizar uma verificação dos dados de entrada. Para isso é necessário que se ative o software mayavi[7] e se abra o arquivo onde os dados foram escritos. Logo a seguir, com o comando *Visualize/Modules/SurfaceMap* se visualizará a estrutura e poderá ser conferido se realmente os dados introduzidos estão corretos.

Pode-se com o botão esquerdo do mouse girar a estrutura e com o botão direito, ampliar ou reduzir a estrutura. Também com a opção Set Representation to Wireframe , pode-se verificar como ficou representada a discretização de cada seção transversal em qualquer elemento (Figura 7). Pode-se ainda tornar a estrutura opaca (Figura 5), pois pretende-se criar um módulo para a visualização do carregamento e assim evitar a sobreposição de imagens com a mesma intensidade de coloração. Você pode reduzir a intensidade da cor na estrutura e colocar o carregamento na estrutura com uma maior intensidade de cor, ficando assim mais fácil de se visualizar o carregamento.

No caso do pós processamento, em geral, temos os deslocamentos e os esforços nos elementos. Percebe-se que no caso dos deslocamentos, temos uma relação com os pontos e no caso dos esforços temos uma relação com as células.

Por meio da criação de uma palette ou com a utilização de uma palette padrão, pode-se colorir as células de acordo com os valores encontrados para os esforços e podemos movimentar nossa estrutura aplicando os deslocamentos encontrados aos pontos de nossa estrutura (Figura 6). Para que isso seja possível é necessário que as declarações sejam feitas no arquivo base.

No caso dos deslocamentos pode-se declarar os mesmos como um vetor e nesse caso nossa declaração será do tipo POINT_DATA, como já referido anteriormente. A seguir em outra linha definimos o vetor, com a declaração VECTORS deslocamentos float. A seguir será escrito os valores dos deslocamentos nos nós. Com isso pode-se verificar como a estrutura se desloca e ainda tem-se a possibilidade de ampliar o deslocamento para observar o comportamento global da estrutura deformada. Para que isso seja possível basta acionar o comando *Visualize/Filters/WarpVector*. Se você precisar ampliar a deformada, basta aumentar o fator de escala por meio do comando *Set Scale Factor*.

No caso dos esforços solicitantes, estes podem ser declarados como um tensor ou como vários escalares, separando cada um dos esforços. No exemplo preferiu-se declará-los como escalares, assim cria-se automaticamente uma caixa de diálogo com o nome de cada esforço, podendo ser mais facilmente visualizado.

Pode-se ainda por meio do filtro Threshold verificar quais são as células que possuem os esforços positivos e quais as que possuem os esforços negativos, fazendo com que somente as células positivas fiquem aparentes e com isso perceba-se onde temos, por exemplo, tração e compressão em nossa estrutura (Figura 9). A armadura pode também ser representada (Figura10). O carregamento pode ser representado na forma de seta ou ainda na forma de cone (Figura11).

4 TRABALHO FUTURO

Pretende-se criar ainda dois módulos, onde se terá a visualização dos esforços internos resistentes em cada célula e poderá ser solicitado a informação dos valores relacionados a cadas célula. A estrutura já está discretizada para essa finalidade e o aproveitamento do módulo que torna a estrutura opaca será aproveitado para uma melhor visualização dos esforços internos resistentes.

5 CONCLUSÃO

Com a utilização desse software de domínio público, percebe-se que a apresentação dos resultados, bem como a visualização da estrutura a ser analisada, ficam mais simples de serem observadas por pessoas que não tenham uma familiaridade com o método dos elementos finitos, sendo que sua utilização é bastante simples, uma vez que a base da estrutura de dados utilizada se encontra disponível nos programas de análise estrutural.

6 AGRADECIMENTOS

À CAPES pelo apoio recebido por meio do financiamento deste projeto (BEX2248/01-8)

Ao IST (Instituto Superior Técnico de Lisboa), à USP(Universidade de São Paulo) e à PUC-MINAS (Pontifícia Universidade Católica de Minas Gerais – *Campus* Poços de Caldas), pela oportunidade do desenvolvimento deste trabalho.

7 REFERÊNCIAS BIBLIOGRÁFICAS

[1] Ellis, T. M. R., *Fortran 77 Programming, second edition, (with an introduction to the fortran 90 standard)*, Addilson Wesley,1990.

[2] Pereira, E.M.B.R, *Análise fisicamente não linear de pórticos tridimensionais de betão armado*, Dissertação de Mestrado, Universidade Técnica de Lisboa, 1989.

[3] Pereira, E.M.B.R, *Elementos finitos de tensão (Aplicação à análise elástica de estruturas)*, Tese de Doutorado, Universidade Técnica de Lisboa, 1993.

[4] Schroeder, W.; Martin,K.; Lorensen, B.; *The visualization Toolkit, An Object-Oriented Appoach to 3D Graphics,* Prentice Hall, second edition, 646 p . , 1997.

[5] Soler, J.G.M, *Análise não-linear de pórticos planos de concreto armado*, Dissertação de Mestrado, EPUSP/USP,1989.

[6] Soler, J.G.M, *Análise não-linear de pórticos espaciais de concreto armado*, Tese de Doutoramento, EPUSP/USP,1995.

[7] Ramachandran, P. , *MayaVi Users Guide*, 2001

[8] Jr. Drake, F.L: Rossum, G. , *Python Reference Manual, Release 2.1.2*, 2002.

8 ANEXO

8.1 Arquivo texto base

```
# vtk DataFile Version 3.0
vtk output
ASCII
DATASET UNSTRUCTURED_GRID
POINTS        416 float
      -20.000000    0.000000E+00      -20.000000
      -20.000000    0.000000E+00       20.000000
       20.000000    0.000000E+00      -20.000000
        (Demais Pontos)
CELLS          52          468
            8           1           0           3           2           5
            4           7           6
            8           9           8          11          10          13
           12          15          14
            8          17          16          19          18          21
           20          23          22
            8          25          24          27          26          29
           28          31          30
          (demais células)
CELL_TYPES         52
 11
 11
 11
 (deve-se colocar 52 vezes o número 11)
CELL_DATA          52
SCALARS cortante1 float
LOOKUP_TABLE default
       24.960000
       24.850000
       24.850000
    1.346000E-01
    1.717000E-02
   -5.907000E-03
     (demais valores da cortante1 nas células)
SCALARS cortante2 float
LOOKUP_TABLE default
   -2.982000E-02
   -2.972000E-02
   -2.975000E-02
       29.550000
       29.590000
       29.550000
     (demais valores da cortante2 nas células)
SCALARS normal float
LOOKUP_TABLE default
       32.520000
       32.610000
       32.610000
```

```
      -20.700000
      -17.460000
      -20.780000
       (demais valores da normal nas células)
SCALARS momento1 float
LOOKUP_TABLE default
        5.136000
    9.162000E-01
       -2.016000
    -4295.000000
       63.770000
     4423.000000
      -11.680000
       -3.957000
        7.653000
   (demais valores dos momentos nas células)
SCALARS momento2 float
LOOKUP_TABLE default
     3642.000000
     1150.000000
    -1335.000000
      613.000000
      -12.450000
      627.900000
     (demais valores do momento2 nas células)
SCALARS momento3 float
LOOKUP_TABLE default
      338.100000
      338.100000
      338.100000
      -39.110000
      -44.020000
      -39.570000
     (demais valores do momento3 nas células)
POINT_DATA        416
VECTORS deslocamentos float
    0.000000E+00    0.000000E+00    0.000000E+00
    0.000000E+00    0.000000E+00    0.000000E+00
    0.000000E+00    0.000000E+00    0.000000E+00
    0.000000E+00    0.000000E+00    0.000000E+00
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        2.282000   -3.046000E-03    7.424000E-01
        6.903000   -7.752000E-03    6.790000E-01
        6.903000   -7.752000E-03    6.790000E-01
        6.903000   -7.752000E-03    6.790000E-01
       (demais valores dos deslocamentos nos nós)
```

8.2 Visualizações da estrutura

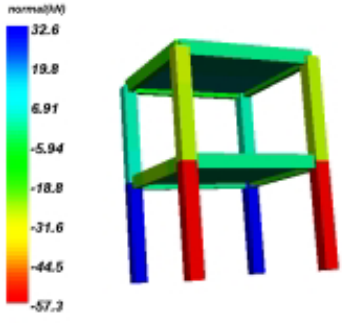


Figura 4
Estrutura vista com o esforço normal

Figura 5
Estrutura com 20% de sua coloração

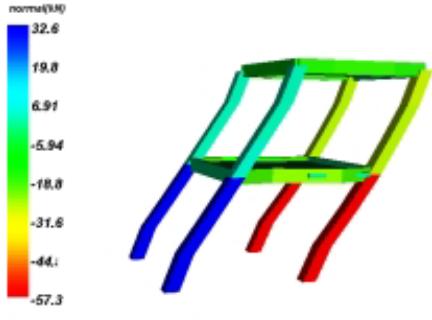


Figura 6
Estrutura deformada

Figura 7
Estrutura em sua forma discretizada

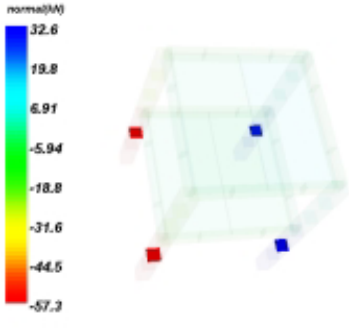


Figura 8
Realce de uma seção da estrutura

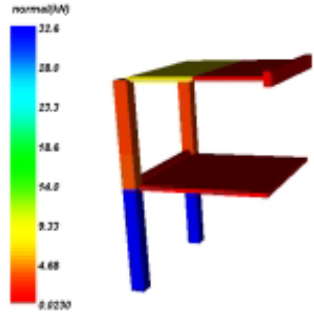


Figura 9
Representação das células de esforço normal positivo

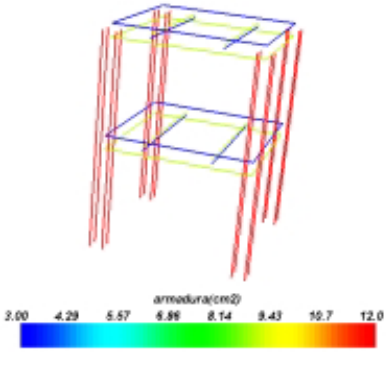


Figura 10
Representação da armadura

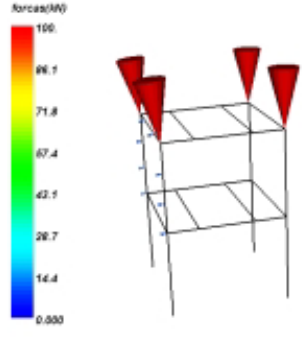


Figura 11
Representação do carregamento na forma de cone