

Manual de referência

SIGMA 2D

Sistema Integrado de Geotecnia para Múltiplas Análises



Tecgraf
PUC-RIO

Tecnologia em
Computação Gráfica



Informações de copyright

SIGMA2D © 2013

Este manual é para uso exclusivo da Petrobras e do instituto Tecgraf, podendo conter informações confidenciais e/ou legalmente privilegiadas. Não pode ser parcial ou totalmente reproduzido sem o consentimento da Petrobras e do instituto Tecgraf. Qualquer divulgação ou uso não autorizado deste manual é expressamente proibido.

ÍNDICE

1. CONSIDERAÇÕES GERAIS.....	7
2. INTERFACE	8
2.1 - <i>Command Line</i>	8
2.2 - Comandos utilizando o Mouse como meio de entrada.....	15
3. COMANDOS DO SIGMA2D	16
3.1 – Comandos para criação de primitivas.....	16
3.1.1 – Criação de linha (comando <i>LINE</i>)	16
3.1.2 – Criação de linhas poligonais (Comando <i>POLYLINE</i>)	17
3.1.3 – Criação de arco de circunferência (Comando <i>ARC</i>)	18
3.1.4 – Criação de arco de elipse (Comando <i>ELIPSEARC</i>)	19
3.1.5 – Criação de círculo (comando <i>CIRCLE</i>)	21
3.1.6 – Criação de elipse (comando <i>ELIPSE</i>)	22
3.1.7 – Criação de bezier (comando <i>BEZIER</i>)	24
3.1.8 – Criação de spline (por pontos de interpolação, comando <i>SPLINE</i>).....	25
3.1.9 – Criação de spline (por parâmetros geométricos da curva, comando <i>GEOMSPLINE</i>)	26
3.1.10 – Criação de vértice (Comando <i>VERTEX</i>)	26
3.2 – Comandos de transformações afins.....	28
3.2.1 – Translação (Comando <i>MOVE</i>)	28
3.2.2 – Rotação (Comando <i>ROTATE</i>)	29
3.2.3 – Escala (Comando <i>SCALE</i>).....	30
3.2.4 – Espelhamento (Comando <i>MIRROR</i>)	32
3.2.5 – Repetição da última transformação afim (Comando <i>REP</i>)	33
3.2.6 – Permanência da entidade original numa transformação afim (Comando <i>LEAVEORIG</i>)	34
3.3 – Comandos de seleção e remoção de entidades e captura de pontos.....	36
3.3.1 – Pick (Comando <i>PICK</i>).....	36
3.3.2 – Pick por cerca (Comando <i>FENCEPICK</i>).....	36
3.3.3 – Pick por cerca poligonal (Comando <i>POLYFENCE</i>).....	37
3.3.4 – Seleção de vértices (Comando <i>SELVERTEX</i>)	39
3.3.5 – Retirada de seleção de vértice (Comando <i>UNSELVERTEX</i>)	40
3.3.6 – Seleção de curvas (Comando <i>SELCURVE</i>)	41
3.3.7 – Retirada de seleção de curva	42
3.3.8 – Seleção de faces.....	42
3.3.9 – Retirada de seleção de face	43
3.3.10 – Retirada de seleção de todas as entidades.....	43
3.3.11 – Seleção de entidades por atributo.....	44
3.3.12 – Remoção de entidades ou de atributos.....	46
3.4 – Comando de ajuste de tolerância geométrica do modelo.....	48
3.5 – Comandos de subdivisão de curvas	49
3.5.1 – Subdivisão de curva por número de subdivisões.....	50
3.5.2 – Subdivisão de curva por tamanho	51
3.5.3 – Subdivisão de curva por comprimento mínimo de curva	52
3.5.4 – Subdivisão de curva por comprimento relativo de curva	53
3.5.5 – Subdivisão de curva por comprimento da menor curva selecionada	54

3.6 – Comandos de geração e remoção de malhas	54
3.6.1 – Geração de malha bilinear quadrática.....	55
3.6.2 – Geração de malha bilinear triangular.....	56
3.6.3 – Geração de malha trilinear.....	57
3.6.4 – Geração de malha por triangulação.....	58
3.6.5 – Geração de malha quadrilateral.....	59
3.6.6 – Geração de malha de transição.....	60
3.6.7 – Remoção de malha.....	61
3.7 – Comandos de manipulação das características das entidades	62
3.7.1 – Tipo de curva.....	62
3.7.2 – Transformação de face em orifício.....	64
3.7.3 – Transformação de orifício em face.....	65
3.7.4 – Definir lado de uma curva de interface a partir de uma face adjacente.....	65
3.7.5 – Definir lado de uma curva de interface adjacente ao exterior.....	66
3.8 - Comandos de visualização	68
3.8.1 – Visualização por arame (Wireframe).....	68
3.8.2 – Visualização por material.....	68
3.8.3 – Visualização de malhas.....	69
3.8.4 – Visualização de subdivisão de malhas.....	69
3.8.5 – Visualização de subdivisão de curvas.....	69
3.8.6 – Visualização de entidades por material.....	70
3.8.7 – Redesenho.....	70
3.8.8 – Ajuste do modelo à janela de visualização.....	70
3.8.9 – Aumento de zoom.....	71
3.8.10 – Diminuição de zoom.....	71
3.8.11 – Último zoom.....	71
3.8.12 – Próximo zoom.....	72
3.8.13 – Zoom por cerca.....	72
3.8.14 – Visualização de grade.....	73
3.8.15 – Atração para a grade.....	73
3.8.16 – Visualização de eixos coordenados.....	74
3.8.17 – Largura de visualização.....	74
3.8.18 – Altura de visualização.....	75
3.8.19 – Passo da grade (distância entre pontos consecutivos da grade).....	75
3.8.20 – Fator de distorção (Y/X).....	75
3.8.21 – Fator de escala.....	76
3.8.22 – Atribuição de cores do modelo.....	76
3.9 – Gerenciamento de grupos	78
3.9.1 – Criando um grupo.....	78
3.9.2 – Adicionar entidades ao grupo.....	79
3.9.3 – Remover entidades de um grupo.....	80
3.9.4 – Deletar grupos.....	82
3.9.5 – Renomear grupos.....	82
3.9.6 – Informações sobre grupos.....	83
3.9.7 – Selecionar/Deselecionar grupos.....	84
3.9.8 – Mostrar/Ocultar grupos.....	85
3.10 – Comandos de início e fim de bloco para serem interpretados em linguagem LUA	87
3.10.1 – Início de bloco de comandos em LUA.....	87
3.10.2 – Fim de bloco de comandos em LUA.....	87
3.11 – Manipulação de atributos	88
3.11.1 – Associação de um atributo às entidades selecionadas.....	88
3.11.2 – Remoção de um atributo das entidades selecionadas.....	89

3.11.3 – Cópia de um atributo	89
3.11.4 – Consulta aos parâmetros de um atributo	89
3.11.5 – Deletar um atributo	90
3.12 – Criação e modificação de atributos	91
3.12.1– Funções.....	92
3.12.1.1 – Função de tempo	92
3.12.1.2 – Função de espaço.....	94
3.12.2 – Materiais	94
3.12.2.1 – Materiais de sólidos	94
3.12.2.2 – Materiais de interface.....	103
3.12.3 – Propriedades.....	110
3.12.3.1 – Propriedades de sólidos	110
3.12.3.2 – Propriedades de interface.....	112
3.12.4 – Fronteiras.....	115
3.12.4.1 – Escavação	115
3.12.4.2 – Reconstrução	116
3.12.4.3 – Restrições.....	118
3.12.5 – Cargas	121
3.12.5.1 – Geostática	121
3.12.5.3 – Cargas genéricas	123
4 – IMPORTAÇÃO / EXPORTAÇÃO DE ARQUIVOS.....	142
4.1 – Abrir arquivos	142
4.1.1 – Abrir arquivo geométrico.....	142
4.1.2 – Abrir arquivo binário.....	143
4.1.3 – Abrir arquivo neutro	143
4.2 – Importação.....	144
4.2.1 – Templates	144
4.2.2 – Atributos	146
4.2.3 – Primitivas	146
4.2.4 – Arquivo dxf.....	147
4.2.5 – Arquivo de linha de comando.....	148
4.2.6 – Arquivo UTM.....	149
4.2.7 – Arquivo GoCAD	149
4.2.8 – Arquivo IGES	149
4.3 – Exportação	149
4.3.1 – Atributos	150
4.3.2 – Primitivas	150
4.3.3 – Regiões.....	150
4.3.4 – Arquivo de linha de comando.....	150
4.3.5 – Arquivo IGES	150
5 – PARAMETRIZAÇÃO.....	151
5.1 – Parametrização no SIGMA.....	153
5.1.1 – Comandos em linguagem LUA	155
5.1.2 – Chamadas de comandos para execução de tarefas de dentro de um bloco de comandos.....	161
5.1.3 – Uso de funções e variáveis definidas em LUA pelo usuário fora do bloco de comandos.....	162
5.1.4 – Atalhos para <i>strings</i> comumente usadas.....	163
6 – CONSIDERAÇÕES GERAIS.....	167

6.1 - Help	167
6.2 - Dynamic Help	168
6.3 - Sigma2D na Internet	169
6.4 - Informações	169

1. CONSIDERAÇÕES GERAIS

O **SIGMA2D**, Sistema Integrado de Geotecnia para Múltiplas Análises em 2 dimensões, é um programa que foi desenvolvido para realizar simulações numéricas de problemas bidimensionais de geotecnia, utilizando para tal, o Método dos Elementos Finitos (MEF). O **SIGMA2D** pode simular, por exemplo, a escavação de poços para prospecção de petróleo e integridade estrutural de seus revestimentos. Este sistema integra, em um único ambiente, os módulos de pré-processamento Mtool (sistema gráfico interativo e configurável para geração, edição e manipulação de malhas de elementos finitos bidimensionais) e pós-processamento Mview. Esta integração é responsável pela geração do modelo numérico para análise pelo MEF e visualização dos resultados. Este programa foi desenvolvido pelo grupo MVGEO (modelagem e visualização geométrica) do TecGraf/PUC-Rio. A análise numérica é feita através dos programas para problemas de geotecnia Aeepecd (análises numéricas elasto-plásticas, com não-linearidade física, de modelos de estado plano de tensão, estado plano de deformação e aximétricos) e Anvec (análises numéricas quasi-estáticas, com não-linearidade física, de modelos de estado plano de tensão, estado plano de deformação e aximétricos), ambos desenvolvidos no Centro de Pesquisa da PETROBRAS (CENPES/DIPREX/SEDEM). Este sistema é extensível e configurável, permitindo que novas funcionalidades sejam facilmente implementadas.

Este sistema foi desenvolvido utilizando-se o sistema de interface com o usuário IupLua que permite a obtenção de uma interface atrativa, configurável e eficiente para execução das tarefas. Sob o sistema de interface e o sistema para geração de malhas propriamente dito, encontra-se o sistema gráfico Cd que garante a portabilidade para diversos ambientes. Outra ferramenta importante utilizada no desenvolvimento do **SIGMA2D** é a biblioteca de funções, HED, que manipula estruturas de dados topológicas para o gerenciamento de subdivisões planares. Além disso, foi utilizado um sistema configurável para o gerenciamento de atributos, ESAM, escrito em uma linguagem de configuração, Lua, que permite ao usuário definir seus próprios atributos em tempo de execução, ou seja, sem precisar recompilar e religar o sistema.

2. INTERFACE

2.1 - Command Line

Para facilitar a modelagem e permitir que o conjunto de tarefas executadas pelo usuário ao longo da mesma seja armazenado sequencialmente e de forma simples e didática, foi criada a opção de *linha de comando* dentro do programa **SIGMA2D**.

A *linha de comando* possui três funções básicas:

- 1) Permitir ao usuário executar os comandos do programa por meio de um conjunto lógico de operações digitadas via teclado, como por exemplo utilizar o comando *line* para inserir uma linha.
- 2) Permitir que o usuário tenha acesso imediato a todas as tarefas realizadas durante a modelagem, tendo sido essas tarefas realizadas via *linha de comando* ou através da interface do programa.
- 3) Armazenar num arquivo de backup (se o usuário desejar) todo o conjunto de tarefas executadas pelo usuário durante a modelagem (esta opção está ativa por padrão).

Não há dúvida quanto à utilidade dessa ferramenta. Dentre as suas principais vantagens, podem-se citar:

- 1) Mais rapidez e simplicidade na modelagem, para um usuário já acostumado com a sintaxe dos principais comandos que irá utilizar na construção de um modelo. A linha de comando permite que o usuário não precise alternar entre o uso do teclado e do mouse constantemente, podendo executar comandos apenas através do teclado.
- 2) Acesso direto a etapas de modelagem já realizadas, podendo repetir processos inteiros sem nenhuma dificuldade ou perda de tempo.
- 3) Importação de arquivos de backup com todas as etapas realizadas até o momento de ocorrência de um eventual erro do programa.
- 4) Edição do arquivo de backup, podendo-se adaptar um modelo, modificando-se os parâmetros passados para as tarefas, de forma a criar um novo modelo baseado num modelo mais antigo.

Para habilitar a linha de comando no programa **SIGMA2D**, deve-se acessar o submenu do item “*Edit*”, no menu principal do programa. Em seguida, deve-se clicar no item “*Command Line*” (ver Fig. 1).

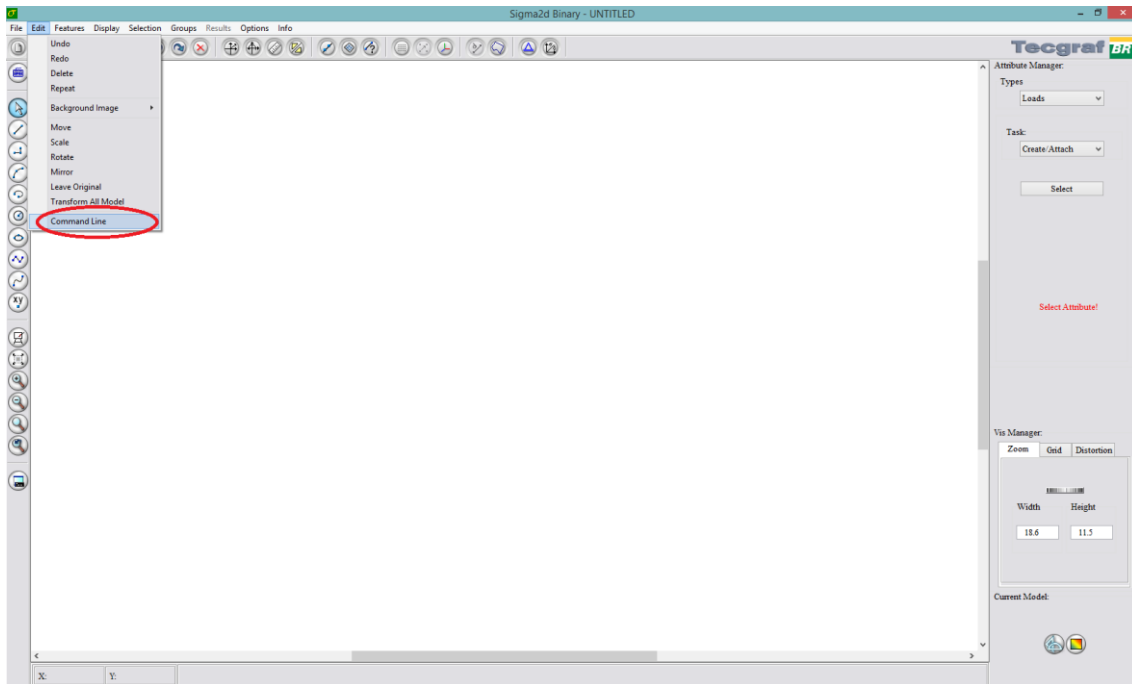


Figura 1 - Habilitação da linha de comando do programa SIGMA2D.

Em seguida, duas novas janelas surgirão na parte inferior do canvas do **SIGMA2D**, uma intitulada “*Command Line*” (**CMD**) e a outra, “*Task History*” (**TH**) (ver Figura 2).

A janela **CMD** é onde o usuário pode digitar os comandos necessários para a execução de uma tarefa. Ela é, portanto, editável a qualquer instante durante a modelagem.

A janela **TH** é onde vai ser armazenado o histórico das tarefas realizadas pelo usuário durante a modelagem. Essa janela não é editável em nenhum momento. Contudo, um ou mais comandos já executados e presentes nessa janela podem ser copiados para serem reproduzidos em outra janela (um arquivo ou a própria **CMD**).

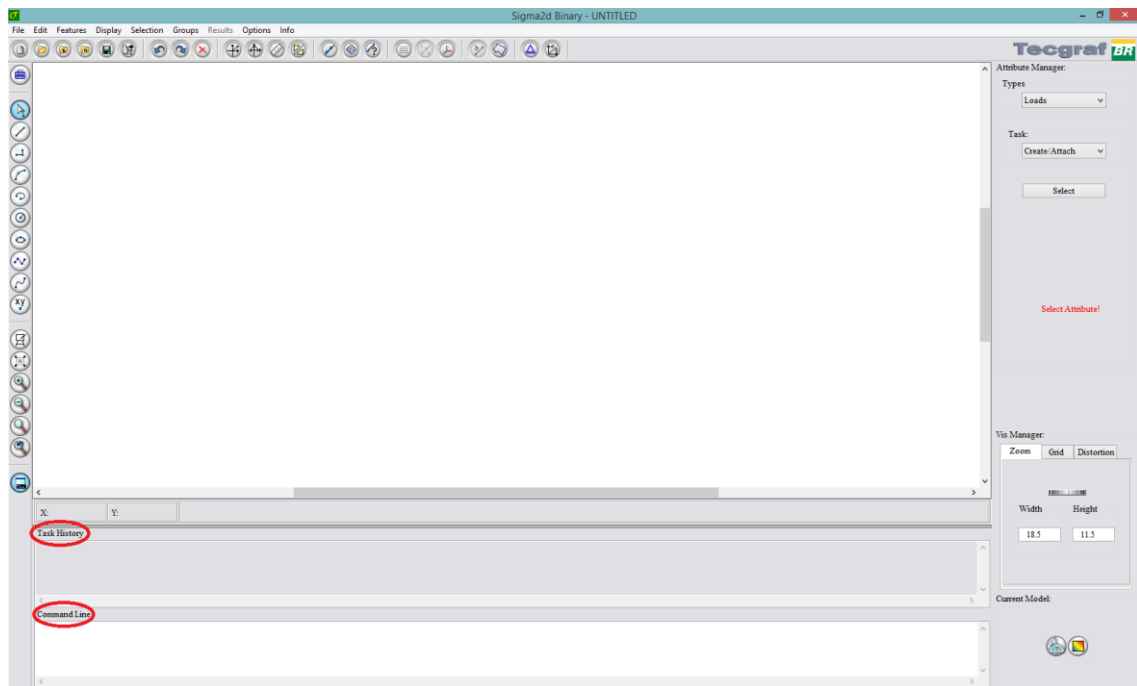


Figura 2 – Janelas “Command Line” (CMD) e “Task History” (TH).

A relação entre o tamanho dessas janelas e o tamanho do Canvas (área de desenho) pode ser alterada, permitindo uma melhor visualização dos comandos digitados ou já executados. Para isso, basta arrastar com o mouse a barra de cor cinza localizada imediatamente acima da janela **TH**, como pode ser visto na Figura 3.

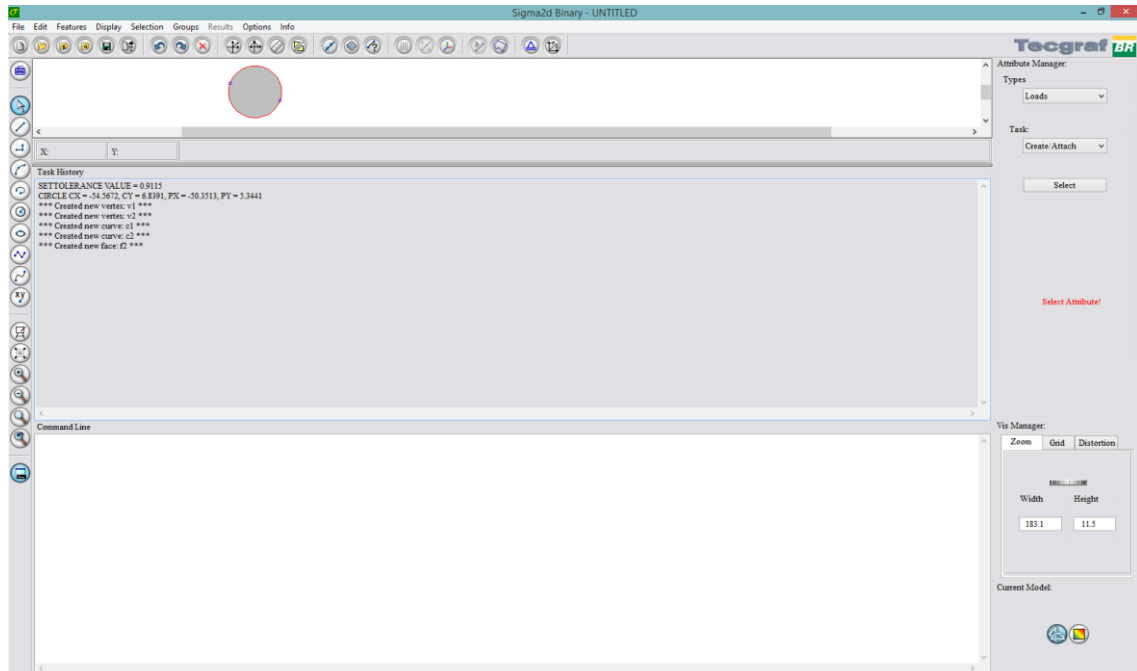


Figura 3 – Alteração da relação entre o tamanho das janelas CMD e TH e o Canvas do programa.

As tarefas executadas pelo usuário durante a modelagem são armazenadas em um arquivo de *backup*. Por padrão, este arquivo é atualizado a cada dez tarefas executadas. O usuário pode alterar esta configuração, definindo o número de tarefas

que desejar (este número pode ser inclusive um, situação em que o arquivo de *backup* é atualizado a cada tarefa que o usuário executa) ou pode também optar por não salvar nenhum arquivo de *backup*. O conteúdo do arquivo de *backup* é idêntico ao que aparece na janela **TH**. O nome do arquivo de *backup* é o nome do arquivo **mtl** corrente, porém com uma extensão própria, denominada **cml**.

Para alterar esta configuração, basta acessar o submenu do item “*Options*” do menu principal do programa, e clicar no item “*Command Line File*”. (Figura 4 e Figura 5)

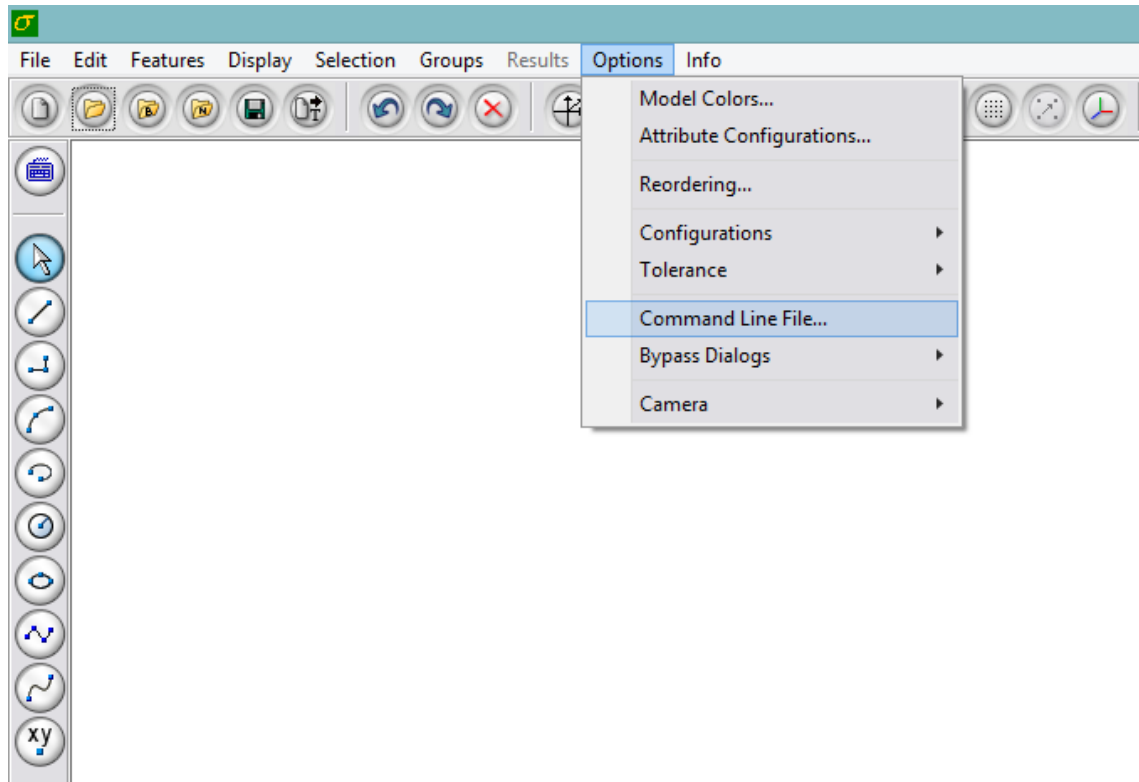


Figura 4 – Acesso às configurações relativas ao arquivo de backup.

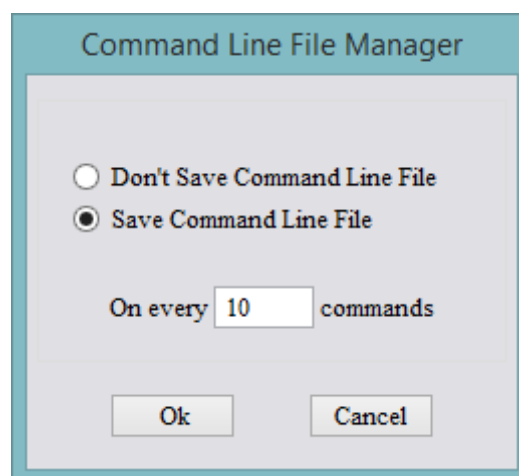


Figura 5 – Janela de configuração do arquivo de backup.

2.1.1 – Sintaxe

Para uma melhor compreensão da sintaxe adotada no **SIGMA2D**, este manual utiliza a seguinte definição:

< > - representa um tipo de informação necessária para a execução do comando.

() - representa um tipo de informação que pode ser omitida, não comprometendo a execução do comando.

[] - representa um tipo de informação que pode ser omitida apenas em alguns casos, dependendo do comando.

Esses caracteres não fazem parte dos comandos do **SIGMA2D** e são apenas utilizados para definição ao longo deste manual.

Todos os comandos que podem ser executados via linha de comando seguem uma sintaxe global comum (obs: alguns comandos do **SIGMA2D** não podem ser executados pela linha de comando):

<LABEL> [PARAMETERS]

A primeira palavra a ser digitada é o "*LABEL*" que identifica a tarefa desejada.

A segunda, caso seja necessário, são os parâmetros que devem ser informados para que a tarefa possa ser completada com êxito. Em alguns casos, o próprio "*LABEL*" da tarefa já é suficiente, com isso não há a necessidade de se escrever nenhum parâmetro.

Normalmente, como será visto na próxima seção, as tarefas possuem mais de um "*LABEL*" que faz com que ela seja executada. Trata-se, na verdade, de um "*LABEL*" mais próximo do nome real da tarefa e de outro(s) que serve(m) como atalho(s). Desta forma, com poucos caracteres o usuário pode definir qual é a tarefa que deseja executar.

Os parâmetros relativos a cada tarefa representam o conjunto de valores inteiros, reais, caracteres ou palavras que definem completamente a tarefa em questão. A maneira como esses parâmetros devem ser passados também obedece a uma sintaxe própria:

*[paramlabel_1 =] <paramvalue_1> [, paramlabel_2 =] <paramvalue_2> ...
[paramlabel_n =] <paramvalue_n> ([paramlabel_n+1 =] paramvalue_n+1) ...
(,[paramlabel_p =] paramvalue_p)*

Isto significa que, para as tarefas que requerem parâmetros para serem completadas com êxito, necessariamente deve-se digitar os valores dos parâmetros essenciais, separados por vírgulas. Existe a possibilidade de se passar o "*LABEL*" relativo a cada parâmetro, seguido de um símbolo "=", antes de se passar o valor do parâmetro em si. Para as tarefas em que a ordem dos parâmetros é fixa, estes *labels* são desnecessários. Contudo, em várias tarefas, estes *labels* tornam-se necessários para identificar qual parâmetro está sendo passado. Isso será analisado mais detalhadamente na próxima seção, quando a sintaxe própria de cada comando será descrita.

Além do conjunto de parâmetros essenciais para a definição da tarefa, existem aqueles que podem ou não ser definidos. Se não forem definidos, não comprometerão a

execução da tarefa (aqueles que estão entre parênteses no diagrama acima). Isso porque esses parâmetros, quando não definidos, assumem valores *default*. Isso também será analisado caso a caso na próxima seção.

Tanto o "*LABEL*" de uma tarefa como seus parâmetros podem ser escritos com letras maiúsculas ou minúsculas, podendo-se inclusive mesclar caracteres maiúsculos e minúsculos. Os valores dos parâmetros serão analisados de acordo com o caso. De forma geral, o valor de um parâmetro representado por uma string (cadeia de caracteres), deve ser escrito entre aspas e da forma correta, a não ser que esteja sendo referenciado através de uma variável definida em LUA, como será visto mais adiante.

A tecla *ENTER* é a chave para a finalização de uma tarefa. Quando esta tecla é digitada, a *string* (cadeia de caracteres) contendo todas as informações digitadas pelo usuário desde a última vez em que a tecla *ENTER* foi acionada é passada para as funções internas que irão analisá-la, verificarão a sua validade e executarão a tarefa correspondente.

Deve-se notar que devido ao fato de a tecla *ENTER* ser a chave para a finalização de um comando, cada um deve ter suas informações digitadas numa mesma linha da **CMD**. Com isso, cada linha escrita na **CMD**, na **TH** e no arquivo de backup, representam uma tarefa executada pelo usuário.

Porém, isto não significa que um conjunto de tarefas não possa ser copiado do arquivo de backup, da **TH** ou de outra fonte diretamente para a **CMD** para ser executado em bloco. Se um bloco de comandos é copiado para a **CMD** e em seguida é digitada a tecla *ENTER*, o programa irá interpretar cada linha separadamente como um novo comando. A exceção a esta regra é o caso de comandos programáveis em linguagem LUA, que serão vistos mais adiante.

As teclas *UP* e *DOWN* (setas para cima e para baixo, respectivamente) permitem que o usuário navegue pelas tarefas que já foram executadas e que estão escritas na **TH**. Essa facilidade é bastante útil na execução repetida de tarefas, onde apenas o valor de um ou mais parâmetros precisa ser alterado.

Apesar de o usuário poder utilizar caracteres de atalho para a execução de uma tarefa, e de poder omitir os *labels* de alguns parâmetros em alguns casos, bem como os próprios parâmetros, quando a tarefa é interpretada internamente pelo programa, a *string* que a define é enviada para a **TH** em sua forma padrão, ou formal, com o nome mais completo da tarefa, com eventuais *labels* de parâmetros e seus valores padrões, mesmo os mesmos tenham sido omitidos.

2.1.2 – Erros de sintaxe

Existem basicamente três tipos de erro que o usuário pode cometer ao utilizar a linha de comando para executar tarefas:

- 1) **Label de tarefa inválido** : significa que o usuário digitou um label inválido, não reconhecido pelo programa como identificador de nenhuma tarefa. Neste caso, a

string digitada pelo usuário é passada para a **TH** inalterada, o programa escreve automaticamente na linha seguinte a mensagem padrão:

“*** Invalid task label! ***”

- 2) **Parâmetros de tarefa inválidos** : significa que o usuário digitou um *LABEL* de tarefa válido, mas houve algum problema com os parâmetros passados. Pode ser que tenha faltado um ou mais parâmetros, ou que seus valores estejam errados. Neste caso, a *string* digitada pelo usuário é passada para a **TH** inalterada, o programa escreve automaticamente na linha seguinte a mensagem padrão:

“*** Invalid task parameters! ***”

- 3) **Bloco de comandos em LUA inválido**: significa que o usuário digitou, dentro do bloco de comandos específico para ser interpretado pela linguagem LUA, comandos inválidos relativamente à sintaxe própria da linguagem. Neste caso, o bloco de comandos digitado pelo usuário é passado para a **TH** inalterada, o programa escreve automaticamente na linha seguinte a mensagem padrão:

“*** Invalid block of LUA commands! ***”

2.2 - Comandos utilizando o Mouse como meio de entrada

Para facilitar a modelagem e possibilitar uma maior interação entre o usuário e o **SIGMA2D**, permite-se que muitas funções do programa sejam ativadas através da utilização do mouse e de botões com ícones e menus dentro de *frames*.

Não há dúvida quanto à utilidade dessa ferramenta, e sua principal vantagem reside no fato de possuir maior rapidez e simplicidade na modelagem, mesmo para um usuário que não esteja acostumado com o uso do programa, pois a maioria dos ícones dos botões são auto-explicativos e mesmo quando houver dúvida sobre sua funcionalidade o seu *tipe* elucidará tal dúvida.

A habilitação do uso do mouse é automática, pois ela é um comando padrão (*default*) do **SIGMA2D**. Para se ratificar que a utilização do mouse está corrente basta se verificar se botão que contenha a imagem do mesmo esteja selecionado, como mostra a Figura 6.

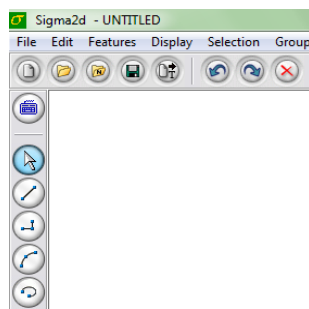


Figura 6 – Botão do mouse selecionado.

A execução de tais comandos é feita clicando-se no botão esquerdo do mouse sobre o botão que tenha o ícone referente ao comando desejado.

Ao se passar para a área de desenho (*canvas*), as funções podem ser executadas diretamente pelo mouse. É importante ressaltar que as coordenadas dos vértices não são controladas explicitamente pelo usuário, a única forma de se ter o controle total é usar o comando *grid* juntamente com o comando *snap*, que serão detalhados mais adiante.

3.COMANDOS DO SIGMA2D

O **SIGMA2D** integra, em um único ambiente, os módulos de pré-processamento **Mtool** (sistema gráfico interativo e configurável para geração, edição e manipulação de malhas de elementos finitos bidimensionais) e pós-processamento **Mview**, responsáveis pela geração do modelo numérico para análise pelo **MEF** e visualização de resultados, com os programas de análise numérica.

Peculiaridades de cada módulo serão explicadas como por exemplo: os comandos relativos à criação/remoção de entidades geométricas, transformações geométricas, visualização, seleção simples e múltipla, geração/remoção de malhas e manipulação de atributos. Ao longo desta seção, serão analisados todos os comandos aceitos minuciosamente.

Eles serão definidos da seguinte forma, 1º será definido qual o comando, depois como executá-lo via *Command Line* e em seguida usando o *mouse* e a interface gráfica. Lembrando que quando tiver apenas a explicação de uso através do *mouse*, o comando só pode ser executado usando comandos de *mouse*.

3.1 – Comandos para criação de primitivas

3.1.1 – Criação de linha (comando *LINE*)

Via *Command Line*

As linhas podem ser criadas passando-se as coordenadas dos pontos inicial e final.

$\langle L \text{ ou } LINE \rangle \langle x1, y1, x2, y2 \rangle$

$x1$ – abscissa do primeiro ponto que define o segmento.

$y1$ – ordenada do primeiro ponto que define o segmento.

$x2$ – abscissa do segundo ponto que define o segmento.

$y2$ – ordenada do segundo ponto que define o segmento.

Outra opção para criação de linhas é a passagem do nome de dois vértices já existentes.

$\langle VL \text{ ou } VLINE \rangle \langle vertex1, vertex2 \rangle$

$vertex1$ – primeiro vértice (já existente no modelo) que define o segmento. O nome deste vértice deve ser o nome (*label*) de um dos vértices já criados no modelo. Nomes de vértices são sempre formados pela letra “v” minúscula e por um número inteiro (Ex: “v1”). O nome do vértice deve estar entre aspas.

$vertex2$ – segundo vértice (já existente no modelo) que define o segmento. Valem as mesmas observações feitas acima.

Via Mouse

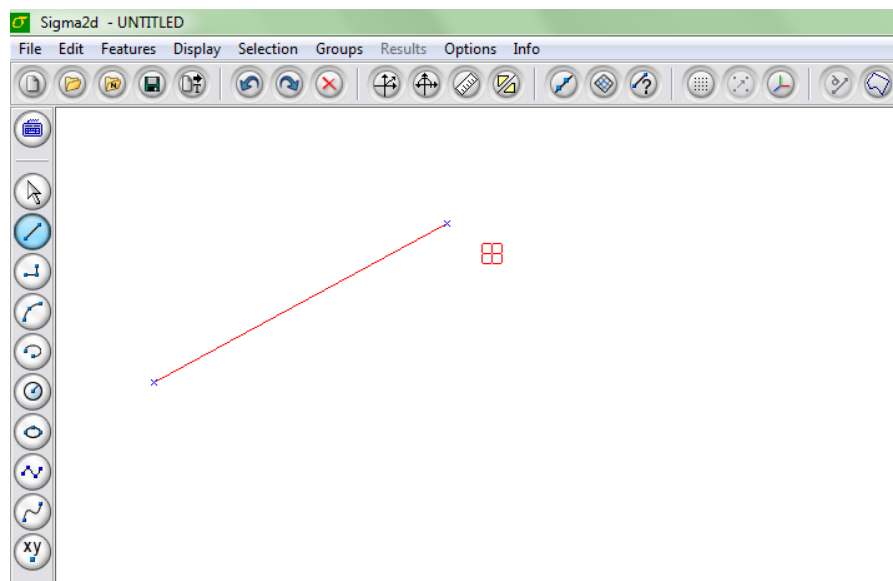


Figura 7 - Execução do comando *Line*

A Figura 7 ilustra a execução do comando *Line*, mostrando o segmento de reta em vermelho e seus dois vértices, marcados com 'x' e de cor azul. Este comando é acessado através da barra de comandos da esquerda, o comando *Line* é o que está selecionado em azul. Para criar a linha usando o mouse, o usuário deve clicar com o botão esquerdo no ponto inicial e depois no ponto final.

3.1.2 – Criação de linhas poligonais (Comando *POLYLINE*)

Via Command Line

$\langle PL \text{ ou } POLYLINE \rangle \langle n, x_1, y_1, x_2, y_2, \dots, x_n, y_n \rangle$

n – número de pontos que definem a linha poligonal.

x_1 – abscissa do primeiro ponto que define a linha poligonal.

y_1 – ordenada do primeiro ponto que define a linha poligonal.

x_2 – abscissa do segundo ponto que define a linha poligonal.

y_2 – ordenada do segundo ponto que define a linha poligonal.

.

.

.

x_n – abscissa do último (n-ésimo) ponto que define a linha poligonal.

y_n – ordenada do último (n-ésimo) ponto que define a linha poligonal.

Via Mouse

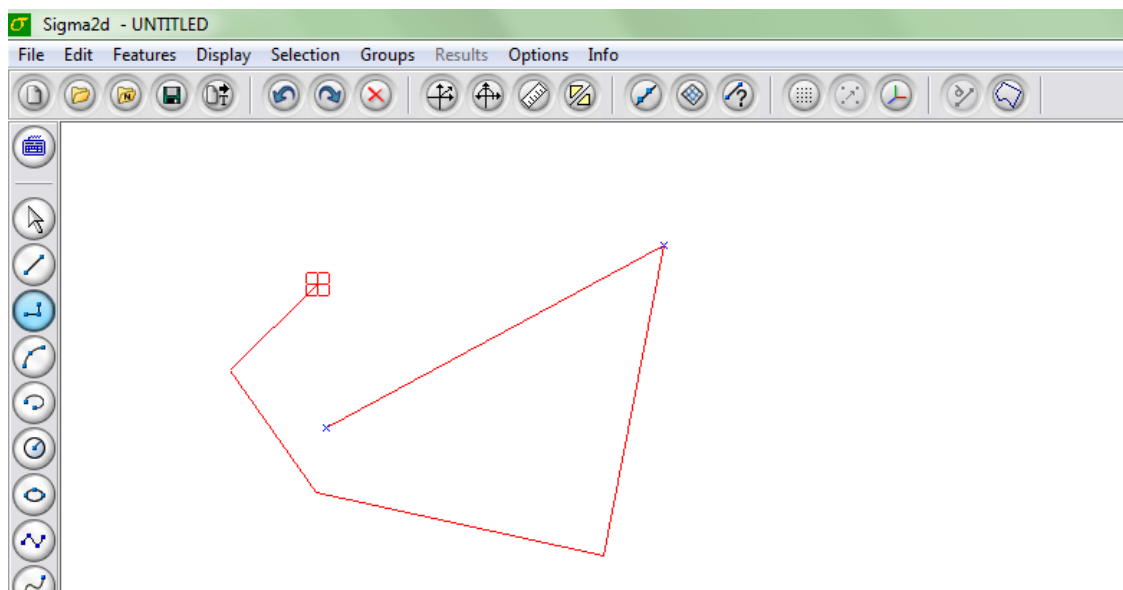


Figura 8 - Execução do comando *Polyline*.

A Figura 8 ilustra a execução do comando *Polyline*, mostrando os segmentos de reta em vermelho e seus vértices inicial e final, marcados com 'x' e de cor azul. O processo de inserção deste comando é similar ao do comando *Line*.

3.1.3 – Criação de arco de circunferência (Comando ARC)

Via Command Line

Existem duas formas de se criar um arco de circunferência através da linha de comando do programa:

- 1) Passando-se as coordenadas (x e y) do ponto que define o centro da circunferência, o raio da mesma e o ângulo inicial e final do arco (os ângulos são em graus, medidos no sentido trigonométrico (anti-horário) a partir da horizontal).

$$\langle A \text{ ou } ARC \rangle \langle cx, cy, r, angi, angf \rangle$$

cx – abscissa do centro da circunferência que contém o arco.

cy – ordenada do centro da circunferência que contém o arco.

r – raio da circunferência que contém o arco.

angi – ângulo inicial do arco (medido a partir da horizontal no sentido trigonométrico).

angf – ângulo final do arco (medido a partir da horizontal no sentido trigonométrico).

- 2) Passando-se as coordenadas (x e y) do ponto que define o centro da circunferência e as coordenadas (x e y) dos dois pontos que definem o início e o fim do arco, nesta ordem (o arco é sempre criado no sentido trigonométrico, do ponto inicial ao ponto final).

$\langle A \text{ ou } ARC \rangle \langle cx, cy, p1x, p1y, p2x, p2y \rangle$

cx e cy – idem ao descrito anteriormente.

$p1x$ – abscissa do ponto inicial do arco.

$p1y$ – ordenada do ponto inicial do arco.

$p2x$ – abscissa do ponto final do arco.

$p2y$ – ordenada do ponto final do arco.

Nesta segunda forma de criar um arco de circunferência, o ponto que representa o centro da circunferência, na verdade, serve para indicar apenas a concavidade do arco, pois o centro da circunferência é recalculado automaticamente, baseado nos dois pontos pelos quais o arco deve passar. Ou seja, não é necessário o usuário saber precisamente a posição do centro da circunferência no plano, apenas sua posição relativa em relação aos dois pontos passados como parâmetro.

Via Mouse

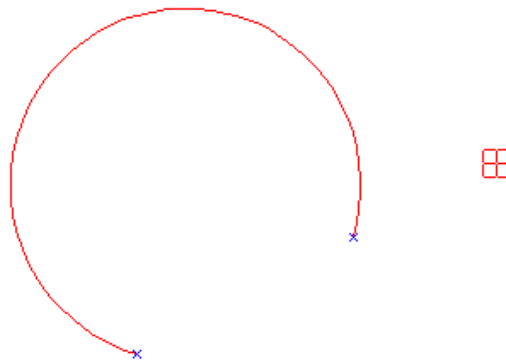


Figura 9 - Execução do comando Arc.

A Figura 9 ilustra a execução do comando *Arc*, mostrando o arco de circunferência em vermelho e seus vértices inicial e final, marcados com 'x' e de cor azul. A criação de um arco através do *mouse* é feita em 3 passos, primeiro clica-se com o botão esquerdo onde deseja o centro do arco, depois clica-se no ponto inicial e depois no ponto final. Lembrando-se que, como através dos comandos da *Command Line*, o arco é criado em sentido anti-horário.

3.1.4 – Criação de arco de elipse (Comando ELIPSEARC)

Via Command Line

Existem duas formas de se criar um arco de elipse através da linha de comando do programa:

- 1) Passando-se as coordenadas (x e y) do ponto que define o centro da elipse, os valores dos semi-eixos da elipse, o ângulo em graus (medido no sentido trigonométrico) entre o primeiro semi-eixo passado como parâmetro e a horizontal, e os ângulos inicial e final do arco de elipse (em graus).

<ELIPSEARC ou EARC ou EA> <cx, cy, a, b, ang, angi, angf>

cx – abscissa do centro da elipse.

cy – ordenada do centro da elipse.

a – primeiro semi-eixo da elipse (é indiferente se ele é maior ou menor que o outro).

b – segundo semi-eixo da elipse.

ang – ângulo (medido no sentido trigonométrico) entre o primeiro semi-eixo e a horizontal.

angi – ângulo inicial do arco (medido a partir da horizontal no sentido trigonométrico).

angf – ângulo final do arco (medido a partir da horizontal no sentido trigonométrico).

- 2) Passando-se as coordenadas (x e y) do ponto que define o centro da elipse, os valores dos semi-eixos da elipse, o ângulo em graus (medido no sentido trigonométrico) entre o primeiro semi-eixo passado como parâmetro e a horizontal, e as coordenadas dos dois pontos que definem o início e o fim do arco, nesta ordem (o arco é sempre criado no sentido trigonométrico, do ponto inicial ao ponto final).

<ELIPSEARC ou EARC ou EA> <cx, cy, a, b, ang, p1x, p1y, p2x, p2y>

cx, cy, a, b e ang – idem ao descrito anteriormente.

p1x – abscissa do ponto inicial do arco.

p1y – ordenada do ponto inicial do arco.

p2x – abscissa do ponto final do arco.

p2y – ordenada do ponto final do arco.

Neste caso, os pontos que são passados como parâmetro como ponto inicial e final do arco não precisam necessariamente localizar-se exatamente sobre a elipse de origem. Dado um ponto qualquer, é feita a interseção entre a elipse e a reta que contém este ponto e o centro da elipse. Este é o ponto que realmente será computado.

Via Mouse

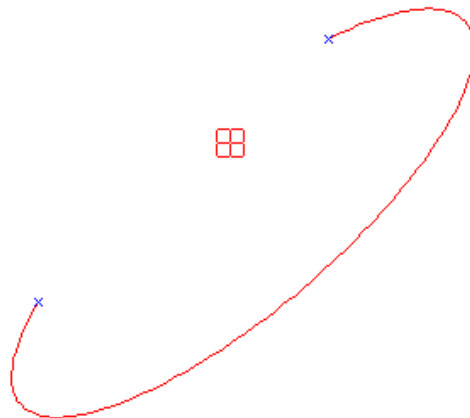


Figura 10 – Execução do comando *Elipse Arc*.

A Figura 10 ilustra a execução do comando *Elipse Arc*, mostrando o arco de elipse em vermelho e seus vértices inicial e final, marcados com 'x' e de cor azul. A criação de arcos de elipse através do *mouse* se dá em 4 passos:

1. Clica-se com o botão esquerdo no ponto que será o centro do arco,
2. Clica-se em um ponto com a distância desejada para ser o primeiro semi-eixo,
3. Aparecerá na tela uma elipse, o usuário então irá clicar em um ponto que definirá o segundo semi-eixo.
4. o usuário irá clicar no ponto inicial e no ponto final do arco, lembrando que ele é criado no sentido trigonométrico.

3.1.5 – Criação de círculo (comando *CIRCLE*)

Via Command Line

Existem duas formas de se criar um círculo através da linha de comando do programa:

- 1) Passando-se as coordenadas (x e y) do ponto que define o centro do círculo e o raio do círculo.

`<CIRCLE ou C> <cx, cy, r>`

cx – abscissa do centro do círculo.

cy – ordenada do centro do círculo.

r – raio do círculo.

- 2) Passando-se as coordenadas (x e y) do ponto que define o centro do círculo e um ponto qualquer sobre a circunferência que define a fronteira do círculo.

$\langle \text{CIRCLE ou } C \rangle \langle cx, cy, p1x, p1y \rangle$

cx e cy – idem ao descrito anteriormente.

$p1x$ – abscissa de um ponto qualquer sobre a circunferência que limita o círculo.

$p1y$ – ordenada de um ponto qualquer sobre a circunferência que limita o círculo.

Via Mouse

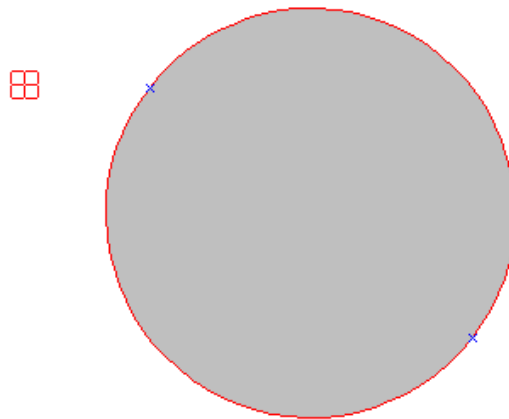


Figura 11 – Resultado da execução do comando *Circle*.

A Figura 11 ilustra a execução do comando *Circle*, mostrando o círculo em vermelho e seus vértices inicial e final, marcados com 'x' e de cor azul. O círculo é criado clicando-se com o botão esquerdo no ponto que será o centro do círculo e depois clicando-se no ponto que definirá o tamanho do raio do círculo. É importante ressaltar que devido ao círculo formar uma região fechada, a parte que se encontra em seu interior é reconhecida como uma nova face topológica.

3.1.6 – Criação de elipse (comando *ELIPSE*)

Via Command Line

Existem duas formas de se criar uma elipse através da linha de comando do programa:

- 1) Passando-se as coordenadas (x e y) do ponto que define o centro da elipse, os valores dos dois semi-eixos da elipse e o ângulo que o primeiro semi-eixo faz com a horizontal.

$\langle \text{ELIPSE ou } E \rangle \langle cx, cy, a, b, ang \rangle$

cx – abscissa do centro da elipse.

cy – ordenada do centro da elipse.

a – primeiro semi-eixo da elipse (é indiferente se ele é maior ou menor que o outro).

b – segundo semi-eixo da elipse.

ang – ângulo (medido no sentido trigonométrico) entre o primeiro semi-eixo e a horizontal

- 2) Passando-se as coordenadas (x e y) do ponto que define o centro da elipse e as coordenadas (x e y) de dois pontos. O primeiro ponto irá definir o comprimento do primeiro semi-eixo da elipse, pela distância entre ele e o centro da elipse, além do ângulo que esse semi-eixo faz com a horizontal (ângulo entre o segmento de reta que une esse ponto ao centro da elipse e a horizontal trigonométrica). O segundo ponto irá definir o comprimento do segundo semi-eixo da elipse, pela distância entre ele e o centro da elipse.

$\langle ELIPSE \text{ ou } E \rangle \langle cx, cy, p1x, p1y, p2x, p2y \rangle$

cx e *cy* – idem ao descrito anteriormente.

p1x – abscissa de um ponto que vai definir o comprimento do primeiro semi-eixo e o ângulo que este irá fazer com a horizontal.

p1y – ordenada de um ponto que vai definir o comprimento do primeiro semi-eixo e o ângulo que este irá fazer com a horizontal.

p2x – abscissa de um ponto que vai definir o comprimento do segundo semi-eixo.

p2y – ordenada de um ponto que vai definir o comprimento do segundo semi-eixo.

Via Mouse

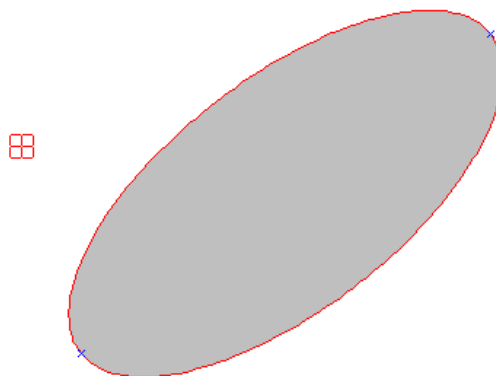


Figura 12 – Execução do comando *Ellipse*.

A Figura 12 ilustra a execução do comando *Ellipse*, mostrando a elipse em vermelho e os vértices inicial e final, marcados com 'x' de cor azul. A elipse é criada clicando-se no ponto que será o centro trigonométrico dela, depois clica-se no ponto que definirá o raio do primeiro semi-eixo, lembrando que esse primeiro ponto, como através do *command line* também define o ângulo do primeiro semi-eixo com a horizontal. Finalmente, clica-se no ponto que definirá o comprimento do segundo semi-eixo. Também é importante ressaltar que devido à elipse formar uma região fechada, a parte que se encontra em seu interior é reconhecida como uma nova face topológica (área cinza da figura).

3.1.7 – Criação de bezier (comando BEZIER)

Via Command Line

Existe apenas uma forma de se criar linhas *bezier* através do *command line*, essa forma é descrita abaixo:

$\langle \text{BEZIER ou BEZ ou B} \rangle \langle p1x, p1y, p2x, p2y, p3x, p3y, p4x, p4y \rangle$

$p1x$ – abscissa do primeiro ponto de controle da curva.

$p1y$ – ordenada do primeiro ponto de controle da curva.

$p2x$ – abscissa do segundo ponto de controle da curva.

$p2y$ – ordenada do segundo ponto de controle da curva.

$p3x$ – abscissa do terceiro ponto de controle da curva.

$p3y$ – ordenada do terceiro ponto de controle da curva.

$p4x$ – abscissa do quarto ponto de controle da curva.

$p4y$ – ordenada do quarto ponto de controle da curva.

Via Mouse

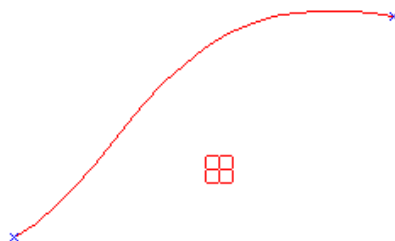


Figura 13 - Execução da função *Bezier*.

A Figura 13 ilustra a forma final da *bezier* após clicar-se no botão **direito** do mouse, onde a *bezier* está destacada de vermelho e os vértices inicial e final estão destacados com 'x' e de cor azul. A linha *bezier* é criada via mouse da seguinte maneira:

- 1) clica-se com o botão **esquerdo** no ponto inicial;
- 2) clica-se com o botão **esquerdo** no ponto final;
- 3) Aparecerá na tela uma reta, com 4 vertices, todos em vermelho e reposicionáveis. O usuário poderá posicionar todos esses pontos até criar a linha com o formato desejado. O comando só se encerra e cria definitivamente a linha quando o usuário clicar em qualquer lugar do canvas com o botão **direito**.

3.1.8 – Criação de spline (por pontos de interpolação, comando SPLINE)

Via Command Line

Existe apenas uma forma de se criar linhas *spline* através do *command line*, essa forma é descrita abaixo:

$\langle \text{SPLINE ou SPL ou S} \rangle \langle n, p1x, p1y, p2x, p2y, \dots, pnx, pny \rangle (, itanx, itany, ftanx, ftany)$

n – número de pontos de interpolação (aqui o usuário determinará quantos pontos irá passar para o programa)

$p1x$ – abscissa do primeiro ponto de interpolação.

$p1y$ – ordenada do primeiro ponto de interpolação.

$p2x$ – abscissa do segundo ponto de interpolação.

$p2y$ – ordenada do segundo ponto de interpolação.

.

.

.

pnx – abscissa do último (n-ésimo) ponto de interpolação.

pny – ordenada do último (n-ésimo) ponto de interpolação.

$itanx$ – componente horizontal do vetor tangente à curva no ponto inicial (opcional).

$itany$ – componente vertical do vetor tangente à curva no ponto inicial (opcional).

$ftanx$ – componente horizontal do vetor tangente à curva no ponto final (opcional).

$ftany$ – componente vertical do vetor tangente à curva no ponto final (opcional).

OBS: Os vetores tangentes à curva em suas extremidades devem ser omitidos por completo ou terem todos os seus valores determinados. Ou seja, não é possível passar como parâmetro as componentes do vetor tangente à curva em seu ponto inicial e não passar as componentes do vetor tangente à curva no seu ponto final.

Via Mouse

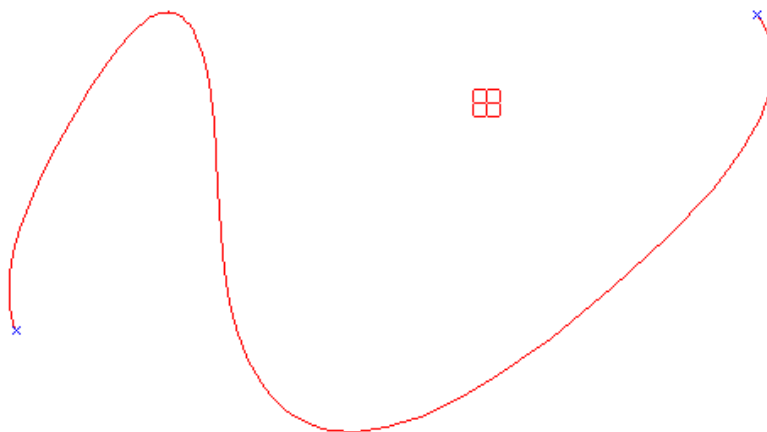


Figura 14 - Finalização do comando *Spline*

A Figura 14 ilustra a forma final da *spline* com três trechos criada, após se clicar no botão **direito** do mouse. A *spline* está destacada de vermelho e os vértices inicial e final estão destacados com 'x' e de cor azul. A *spline* tem seus trechos criados com cliques do botão **esquerdo** até o usuário clicar com o botão **direito**, que encerra o comando e finaliza a *spline*.

3.1.9 – Criação de spline (por parâmetros geométricos da curva, comando GEOMSPLINE)

Os parâmetros geométricos que definem uma curva do tipo *spline* são: o grau da curva, o número de nós do vetor de nós, o vetor de nós, o número de pontos de controle e, os pontos de controle. Para uma melhor compreensão desses parâmetros, recomenda-se a leitura de bibliografia especializada relacionada ao assunto.

Via Command Line

<GEOMSPLINE ou GSPL ou GS> <dg, nu, u1, ..., unu, np, p1x, p1y, ..., pnp_x, pnp_y>

dg – grau da curva (por default, quando uma spline é criada através da interface do programa, o grau da curva é sempre igual a 2 (*spline* cúbica)).

nu – número de nós do vetor de nós

u1 – primeiro nó do vetor de nós

.

unu – último (nu-ésimo) nó do vetor de nós.

np – número de pontos de controle da curva.

p1x – abscissa do primeiro ponto de controle da curva.

p1y – ordenada do primeiro ponto de controle da curva.

.

pnp_x – abscissa do último (np-ésimo) ponto de controle da curva.

pnp_y – ordenada do último (np-ésimo) ponto de controle da curva.

Uma condição básica para que a curva seja corretamente representada é a satisfação da seguinte equação:

$$nu = np + dg + 1$$

Via Mouse

O comando *geomspline* **não** possui comando interativo *via mouse*.

3.1.10 – Criação de vértice (Comando VERTEX)

Este comando serve para o usuário criar vértices no modelo. Eles podem ser criados "soltos", para apenas depois serem definidas linhas que passem por eles, ou podem servir apenas como orientação para o usuário criar seu modelo.

Via Command Line

$\langle VERTEX \text{ ou } V \rangle \langle px, py \rangle (, tol)$

px – abscissa do vértice a ser criado.

py – ordenada do vértice a ser criado.

tol – tolerância geométrica na criação do vértice, para fins de atração (opcional).

Via Mouse



Figura 15 - Criação de dois vértices.

A Figura 15 ilustra dois vértices criados, em azul, o primeiro vértice (V1) e em vermelho o último vértice criado (V2). Os vértices são criados *via mouse* clicando-se sobre o *canvas*, com o botão **esquerdo** do *mouse* no local desejado.

3.2 – Comandos de transformações afins

As transformações afins são aplicadas às entidades já previamente selecionadas. Os exemplos mostrados neste manual para a utilização do *mouse* serão feitos sobre uma primitiva do tipo *line*. Em uso normal, o usuário pode aplicar essas transformações em qualquer entidade presente no *canvas*, ou em certos casos, no modelo inteiro.

3.2.1 – Translação (Comando MOVE)

O comando de translação serve para mover qualquer entidade do modelo para algum outro lugar desejado, como por exemplo, centralizar no eixo "x" uma linha criada em coordenadas negativas.

Via Command Line

$\langle \text{MOVE ou MV} \rangle \langle dx, dy \rangle$

dx – componente horizontal do vetor deslocamento.

dy – componente vertical do vetor deslocamento.

Via Mouse

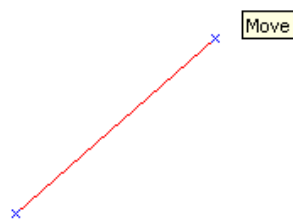


Figura 16 - Posição inicial do segmento de reta.

A Figura 16 ilustra a posição inicial do segmento de reta que será transladado através do comando *Move*, já a Figura 17, mostra a entidade após a translação.

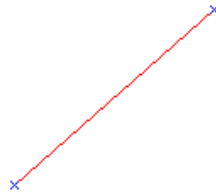


Figura 17 - Posição final do segmento de reta.

A Figura 17 ilustra a finalização do comando *Move*. O comando *via mouse* é executado nas seguintes etapas: Primeiro seleciona-se o comando na guia superior; depois o usuário irá selecionar as entidades que deseja mover, que no caso deste exemplo, é o segmento de reta em vermelho (a cor vermelha indica que a reta está selecionada); a última parte do comando é definir a translação desejada, onde o usuário irá clicar com o botão **esquerdo** no ponto inicial e novamente com o botão **esquerdo** no ponto final, lembrando que a translação que será feita, respeitará tanto a distância quanto o ângulo definido por esse seguimento de reta.

3.2.2 – Rotação (Comando ROTATE)

O comando de rotação serve para rotacionar as entidades selecionadas ao redor de um ponto selecionado pelo usuário.

Via Command Line

<ROTATE ou RT> <ang, cx, cy>

cx – abscissa do centro de rotação.

cy – ordenada do centro de rotação.

ang – ângulo de rotação (em graus).

Via Mouse

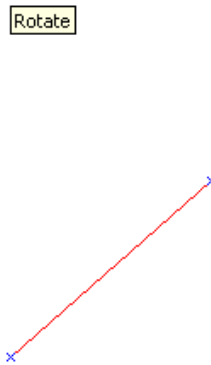


Figura 18 - Posição inicial do segmento de reta.

A Figura 18 ilustra a posição inicial do segmento de reta que será rotacionado através do comando *Rotate*.



Figura 19 - Posição final do segmento de reta.

A Figura 19 ilustra a finalização do comando *Rotate*. Este comando via mouse é executado da seguinte maneira:

- 1) Seleciona-se as entidades que se deseja mover e seleciona-se o comando na guia de acesso rápido.
- 2) Clica-se com o botão direito no centro de rotação desejado.

3.2.3 – Escala (Comando SCALE)

O comando de escala serve para rotacionar as entidades selecionadas ao redor de um ponto selecionado pelo usuário

Via Command Line

<SCALE ou SC> <cx, cy, sx, sy>

cx – abscissa do centro de escala.

cy – ordenada do centro de escala.

sx – fator de escala no eixo “x”.

sy – fator de escala no eixo “y”.

Via Mouse

Após ativar o comando e selecionar a primitiva que será modificada pelo mesmo é necessário se desenhar um segmento de reta sobre o qual se determinará o fator de escala, que será usado para modificar o tamanho da primitiva selecionada.

Scale



Figura 20 - Tamanho inicial do segmento de reta.

A Figura 20 acima ilustra o tamanho inicial do segmento de reta que será alterado através do comando *Scale*.



Figura 21 - Tamanho final do segmento de reta.



A Figura 21 acima ilustra a finalização do comando *Scale* através do clique no botão esquerdo do mouse juntamente com o tamanho final do segmento de reta.

3.2.4 – Espelhamento (Comando *MIRROR*)

Via Command Line

$\langle \text{MIRROR ou MR} \rangle \langle p1x, p1y, p2x, p2y \rangle$

$p1x$ – abscissa do primeiro ponto do segmento que define a reta de espelhamento.

$p1y$ – ordenada do primeiro ponto do segmento que define a reta de espelhamento.

$p2x$ – abscissa do segundo ponto do segmento que define a reta de espelhamento.

$p2y$ – ordenada do segundo ponto do segmento que define a reta de espelhamento.

Via Mouse

Mirror

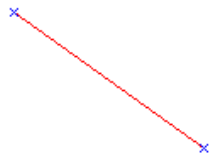


Figura 22 - Posição inicial do segmento de reta.

A Figura 22 ilustra a posição inicial do segmento de reta que será refletido “espelhado” através do comando *Mirror*.

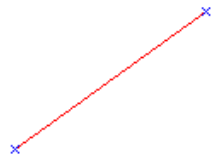


Figura 23 - Posição final do segmento de reta.

A Figura 23 ilustra a finalização do comando *Mirror* através do clique no botão esquerdo do mouse juntamente com a posição final do segmento de reta.

3.2.5 – Repetição da última transformação afim (Comando *REP*)

Via Command Line

<REPEAT ou REP>

Via Mouse

Para exemplificar o comando *Repeat* via *mouse*, usou-se, primeiramente, o comando *Scale*, e após isto, o programa ativa automaticamente a opção do comando *Repeat*, como ilustrado na figura abaixo, desta forma, o usuário estará modificando mais uma vez o tamanho do segmento de reta, ou seja, repetindo o ultimo comando executado.

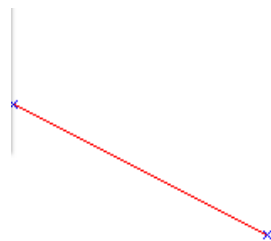


Figura 24 - Tamanho inicial do segmento de reta.

A Figura 24 acima ilustra a seleção do comando *Repeat* através do clique do botão esquerdo do *mouse*, assim como o tamanho inicial do segmento de reta.



Figura 25 - Tamanho final do segmento de reta.

A Figura 25 ilustra o tamanho final do segmento de reta após a utilização do comando *Repeat* (repetição do comando *Scale*).

3.2.6 – Permanência da entidade original numa transformação afim (Comando *LEAVEORIG*)

Via Command Line

<*LEAVEORIG* ou *LO*>

Via Mouse

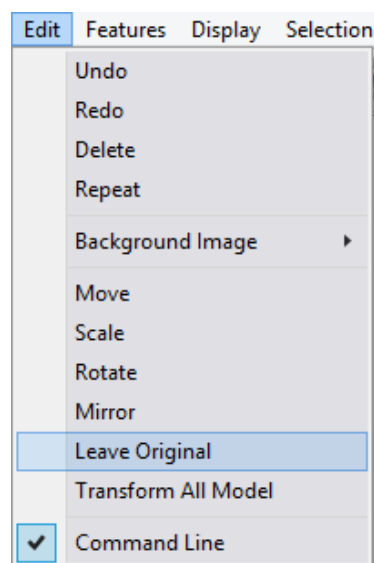


Figura 26 - Seleção do comando Leave Original.

A Figura 26 ilustra a seleção do comando *Leave Orig*.

Cada vez que esse comando é executado, a ferramenta de manter a entidade original é ligada ou desligada, alternadamente (a opção *default* é desligada).

3.3 – Comandos de seleção e remoção de entidades e captura de pontos

3.3.1 – Pick (Comando PICK)

Via Command Line

$\langle PICK \text{ ou } P \rangle \langle px, py, add \rangle (, tol)$

px – abscissa do ponto no plano onde se deseja dar o *pick*.

py – ordenada do ponto no plano onde se deseja dar o *pick*.

add – *flag* para indicar se, no caso do ponto onde o *pick* está sendo efetuado coincidir com um vértice, um ponto interno de uma aresta ou de uma face, esta entidade deve ser selecionada juntamente com outras entidades que já estejam eventualmente selecionadas ($add = 1$), ou se a seleção de quaisquer outras entidades deve ser removida antes de se selecionar a entidade em questão ($add = 0$). Se $add = 1$, este comando só será válido se as entidades previamente selecionadas e a entidade capturada pelo *pick* forem do mesmo tipo. No caso do *pick* ser efetuado num ponto qualquer do plano onde não haja entidades geométricas, todas as eventuais entidades selecionadas deixam de sê-lo se $add = 0$, ou nada acontece se $add = 1$.

tol – tolerância geométrica do modelo, para fins de atração do *pick* (opcional).

Via Mouse

Para usar o comando *Pick* através do uso do mouse basta sobrepor o mesmo sobre a entidade que se deseja selecionar e clicar com o botão esquerdo.

3.3.2 – Pick por cerca (Comando FENCEPICK)

O *pick* por cerca é realizado selecionando-se uma área retangular que englobe todas as entidades que se deseja selecionar, objetivando-se facilitar o entendimento foram usadas três figuras geométricas: um círculo, um quadrilátero e um triângulo. É importante enfatizar que para usar este comando é necessário que o botão *Select* esteja ativado.

Via Command Line

$\langle FENCEPICK \text{ ou } FPICK \text{ ou } FP \rangle \langle xmin, xmax, ymin, ymax, add, op \rangle$

$xmin$ – abscissa mínima dos pontos do retângulo envolvente.

$xmax$ – abscissa máxima dos pontos do retângulo envolvente.

$ymin$ – ordenada mínima dos pontos do retângulo envolvente.

$ymax$ – ordenada máxima dos pontos do retângulo envolvente.

add – *flag* idêntico ao descrito para o *pick* comum.

op – tipo das entidades a serem selecionadas (1 – face, 2 – curva, 3 – vértice).

Via Mouse

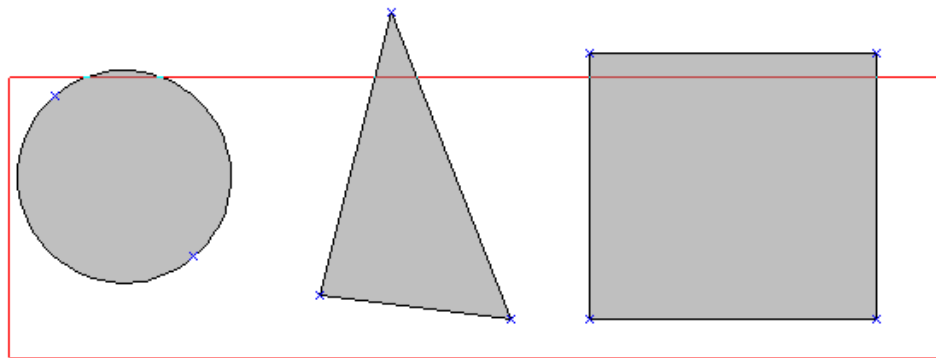


Figura 27 - Seleção das entidades através de retângulo.

A Figura 27 ilustra o retângulo, em vermelho, que engloba todas as figuras, cujas entidades se deseja selecionar, de tal forma, que somente as entidades que estejam totalmente englobadas possam ser selecionadas através do uso do mouse.

3.3.3 – Pick por cerca poligonal (Comando *POLYFENCE*)

O *pick por cerca* é realizado selecionando-se uma área poligonal que englobe todas as entidades que se deseja selecionar. Para facilitar o entendimento, foram usadas quatro figuras geométricas: um círculo, um quadrilátero, uma elipse e um triângulo. Neste caso, para demonstrar a utilidade de tal comando, serão selecionadas entidades de apenas três figuras, de forma que não se selecione nenhuma entidade do círculo, mostrando a eficiência do uso do comando *Pick por cerca poligonal*.

Via Command Line

$\langle \text{POLYFENCE ou PF} \rangle \langle n, p1x, p1y, \dots, pnx, pny, op \rangle$

n – número de vértices do polígono envolvente

$p1x$ – abscissa do primeiro vértice do polígono envolvente

$p1y$ – ordenada do primeiro vértice do polígono envolvente

.

.

pnx – abscissa do último (n -ésimo) vértice do polígono envolvente

pny – ordenada do último (n -ésimo) vértice do polígono envolvente

op - tipo das entidades a serem selecionadas (1 – face, 2 – curva, 3 – vértice).

Via Mouse

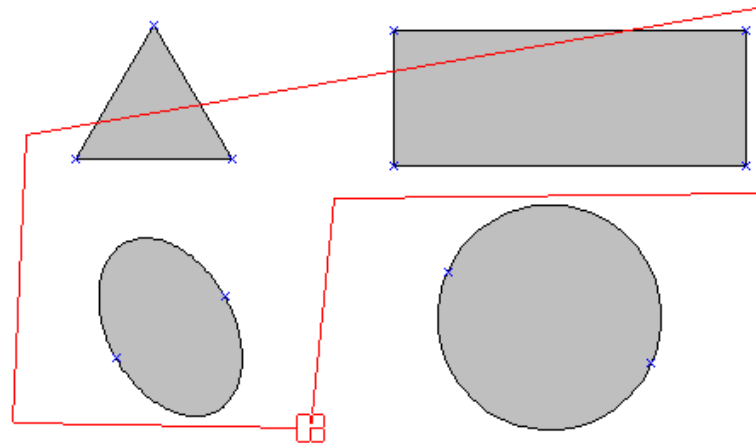


Figura 28 - Seleção das entidades através de polígono.

A Figura 28 ilustra o polígono, em vermelho, que engloba as entidades que se deseja selecionar de tal forma que somente as entidades que estejam totalmente englobadas possam ser selecionadas; bem como o botão que deve ser selecionado para ativar o comando Polygonal Fence.

O comando *Polygonal Fence* pode ser ativado também, através da barra de menu, como mostra a figura abaixo.

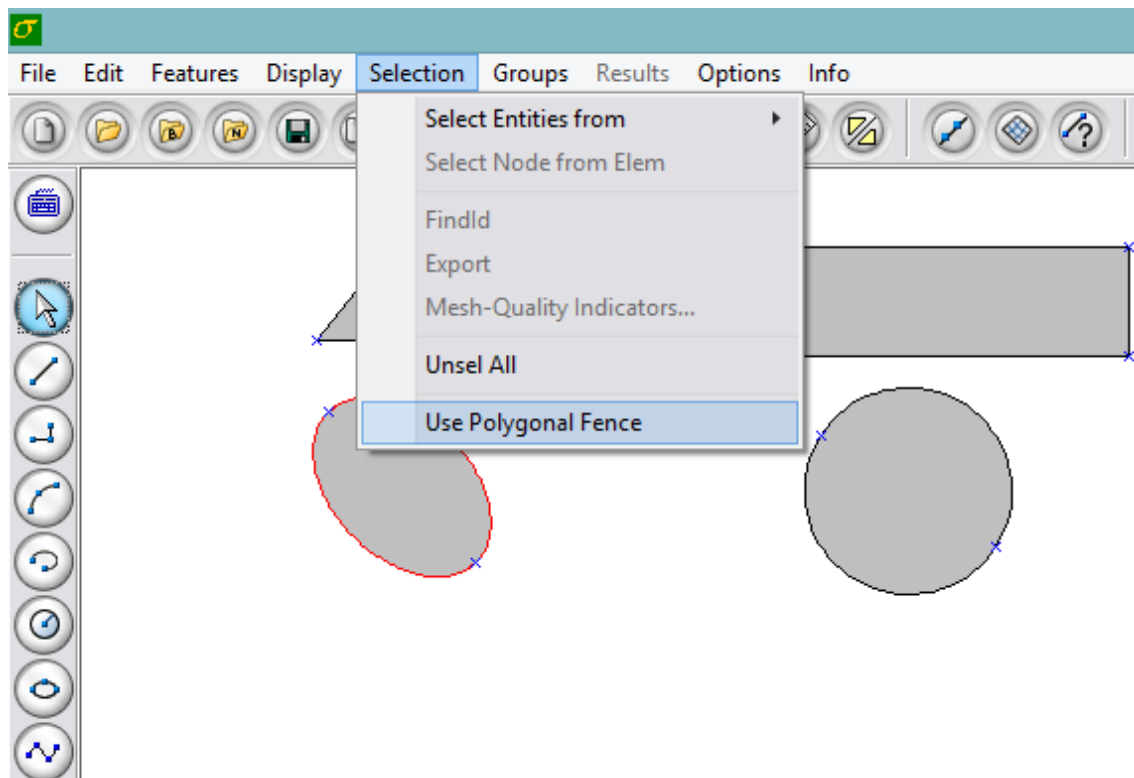


Figura 29 - Ativando o comando Polygonal Fence através da barra de menu.

Após se usar o comando *Pick por cerca poligonal* ou *Pick por cerca* será aberta uma janela para que se possa escolher que tipo de entidade se deseja selecionar. É importante ressaltar também, que nestes dois comandos não é necessário a seleção de toda a figura para que se selecione uma determinada entidade da mesma.

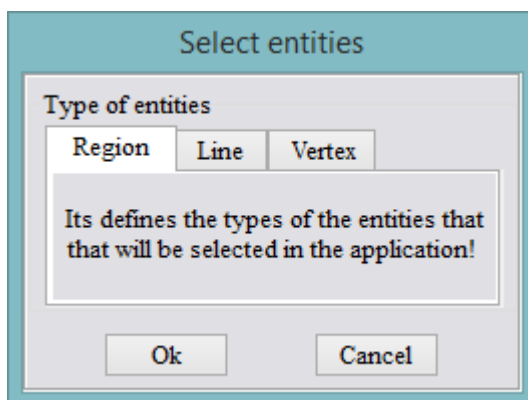


Figura 30 - Janela com as opções das entidades que podem ser selecionadas.

3.3.4 – Seleção de vértices (Comando SELVERTEX)

Via Command Line

$\langle \text{SELVERTEX ou SELV} \rangle \langle v_1, v_2, \dots, v_n, \text{add} \rangle$

v_1 – primeiro vértice a ser selecionado.

v_2 – segundo vértice a ser selecionado.

.

.

.

v_n – último (n-ésimo) vértice a ser selecionado.

add – flag idêntico ao descrito para o *pick* comum, com a ressalva de que o tipo de entidade a ser selecionado é, obviamente, um vértice.

Os nomes dos vértices devem ser *strings* (cadeias de caracteres) escritas entre aspas, sempre iniciadas pela letra “v” e seguidas de um número inteiro. Estes nomes são os próprios nomes dos vértices atribuídos pelo programa.

Via Mouse

Para a seleção de entidades, do tipo vértice, é necessário escolher a opção *Vertex* na janela da Figura 30

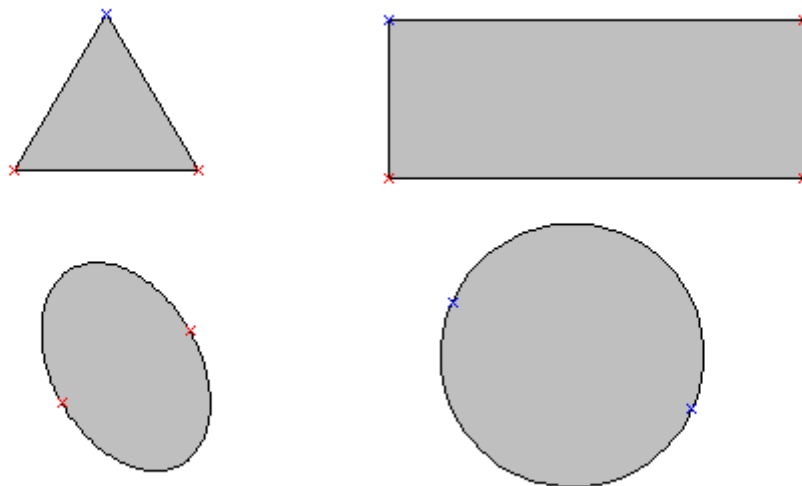


Figura 31 - Seleção de entidades do tipo Vertex.

A Figura 31 ilustra o resultado da finalização do comando Polygonal Fence na Figura 28, onde apenas os vértices que se encontravam dentro do polígono foram destacados de vermelho.

3.3.5 – Retirada de seleção de vértice (Comando *UNSELVERTEX*)

Via Command Line

`<UNSELVERTEX ou UNSELV> <v>`

v – vértice cuja seleção deve ser retirada.

As observações relativas ao nome desse vértice são as mesmas feitas no caso de seleção de vértices. Apenas um vértice pode ter sua seleção retirada de cada vez, assim como é feito por meio da interface do programa.

Via Mouse

Caso haja alguma entidade do tipo *Vertex* selecionada e caso se deseje retirar sua seleção é necessário pressionar a tecla *shift* do teclado, sobrepor o mouse sobre a entidade desejada e clicar no seu botão esquerdo. Esse comando irá deselegionar os vertices individualmente. Pode-se deselegionar todos os vertices desejados sem a necessidade de se soltar a tecla *shift*.

3.3.6 – Seleção de curvas (Comando SELCURVE)

Via Command Line

$\langle \text{SELCURVE ou SELC} \rangle \langle c_1, c_2, \dots, c_n, \text{add} \rangle$

c_1 – primeira curva a ser selecionada.

c_2 – segunda curva a ser selecionada.

.

.

.

c_n – última (n-ésima) curva a ser selecionada

add – flag idêntico ao descrito para o *pick* comum, com a ressalva de que o tipo de entidade a ser selecionado é, obviamente, curva.

Os nomes das curvas devem ser strings (cadeias de caracteres) escritas entre aspas, sempre iniciadas pela letra “c” e seguidas de um número inteiro. Estes nomes são os próprios nomes das curvas atribuídos pelo programa.

Via Mouse

Para a seleção de entidades, do tipo curva, é necessário escolher a opção *Line* na janela da Figura 30.

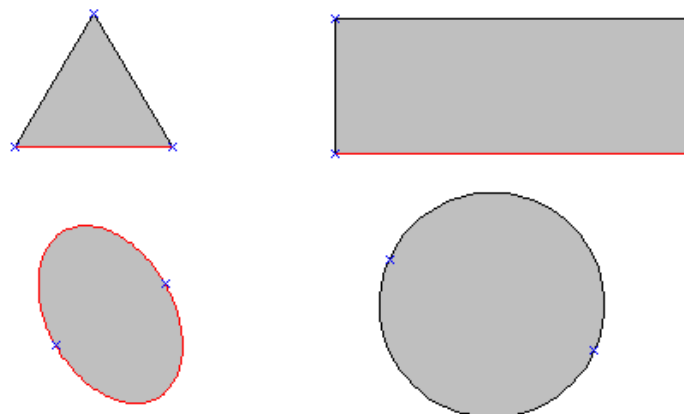


Figura 32 - Seleção de entidades do tipo Line.

A Figura 32 acima ilustra o resultado da finalização do comando Polygonal Fence na Figura 28, onde apenas as linhas que se encontravam dentro do polígono foram destacadas em vermelho.

3.3.7 – Retirada de seleção de curva

Command Line

<UNSELCURVE ou UNSELC> <c>

c – curva cuja seleção deve ser retirada.

As observações relativas ao nome dessa curva são as mesmas feitas no caso de seleção de curvas. Apenas uma curva pode ter sua seleção retirada de cada vez, assim como é feito por meio da interface do programa.

Via Mouse

Caso haja alguma entidade do tipo *Line* selecionada e caso se deseje retirar sua seleção é necessário pressionar a tecla *shift* do teclado, sobrepor o mouse sobre a entidade e clicar no seu botão esquerdo.

3.3.8 – Seleção de faces

Command Line

<SELFACE ou SELF> <f1, f2, ..., fn, add>

f1 – primeira face a ser selecionada.

f2 – segunda face a ser selecionada.

.

.

.

fn – última (n-ésima) face a ser selecionada

add – flag idêntico ao descrito para o *pick* comum, com a ressalva de que o tipo de entidade a ser selecionado é, obviamente, face.

Os nomes das faces devem ser strings (cadeias de caracteres) escritas entre aspas, sempre iniciadas pela letra “F” e seguidas de um número inteiro. Estes nomes são os próprios nomes das faces atribuídos pelo programa.

Via Mouse

Para a seleção de entidades, do tipo superfície, é necessário escolher a opção *Region* na janela da Figura 30.

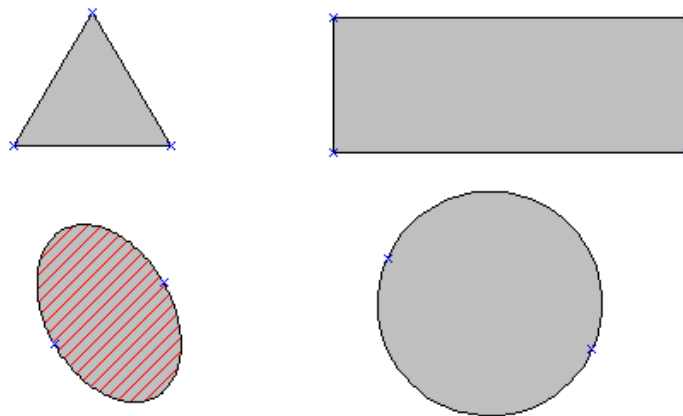


Figura 33 - Seleção de entidades do tipo Region.

A Figura 33 acima ilustra o resultado da finalização do comando Polygonal Fence na Figura 28, onde apenas as regiões que se encontravam dentro do polígono foram destacadas em vermelho.

3.3.9 – Retirada de seleção de face

Command Line

<UNSELFACE ou UNSELF> <f>

f – face cuja seleção deve ser retirada.

As observações relativas ao nome dessa face são as mesmas feitas no caso de seleção de faces. Apenas uma face pode ter sua seleção retirada de cada vez, assim como é feito por meio da interface do programa.

Via Mouse

Caso haja alguma entidade do tipo *Region* selecionada e caso se deseje retirar sua seleção é necessário pressionar a tecla *shift* do teclado, sobrepor o mouse sobre a entidade e clicar no seu botão esquerdo.

3.3.10 – Retirada de seleção de todas as entidades

Command Line

<UNSELALL ou UA>

Via Mouse

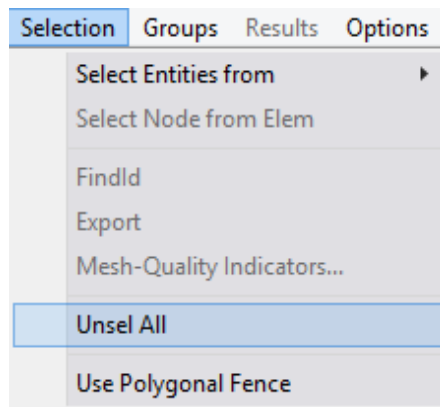


Figura 34 - Item (Unsel All) que retira a seleção de todas as entidades.

A Figura 34 acima ilustra o item que deve ser selecionado para que se retire a seleção de todas as entidades selecionadas, caso haja alguma.

3.3.11 – Seleção de entidades por atributo

Command Line

<SELENTFROM ou SEF> <class, label>

class – nome da classe do atributo. Ex: “Distributed Load”. Esse nome deve estar entre aspas, e deve ser escrito da forma correta, com as iniciais das palavras em maiúsculas e as outras letras minúsculas.

label – nome do atributo daquela classe cujas entidades que o agregam devem ser selecionadas. Esse nome também deve estar entre aspas e deve ser escrito da forma correta, como foi criado.

Via Mouse

Para exemplificar a seleção de entidade através do atributo, criou-se três tipos de materiais: o material do tipo 1 (amarelo), do tipo 2 (vermelho) e do tipo 3 (azul). Ao se clicar no item *Properties* irá se abrir uma janela, Figura 36, na qual estarão todas as propriedades criadas.

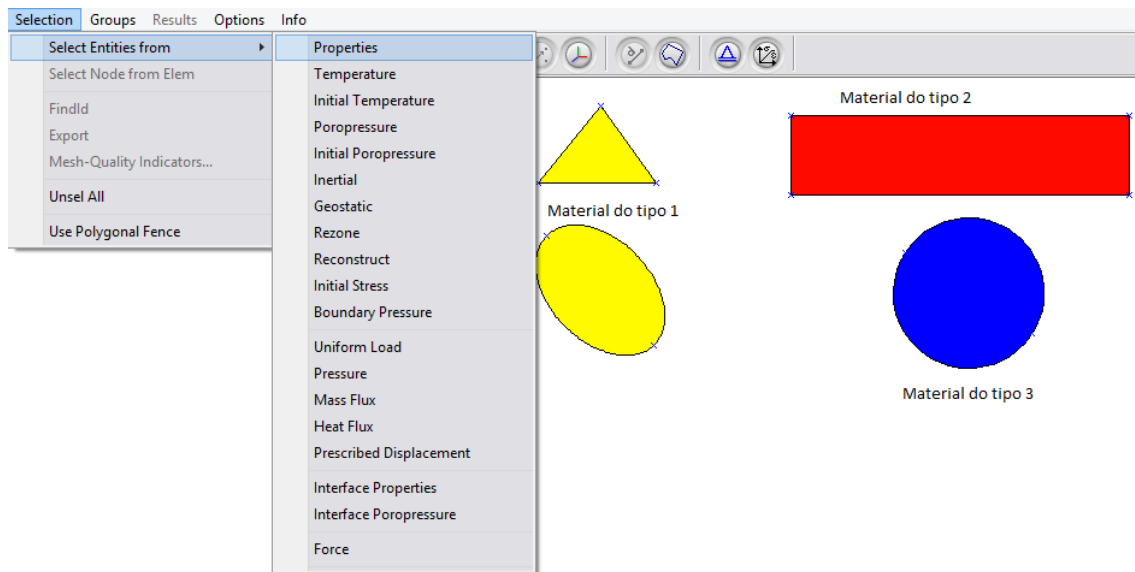


Figura 35 - Item (Properties) que seleciona entidades a partir de atributos.

A Figura 35 acima ilustra o item que deve ser selecionado para que se selecione uma entidade a partir de um atributo.

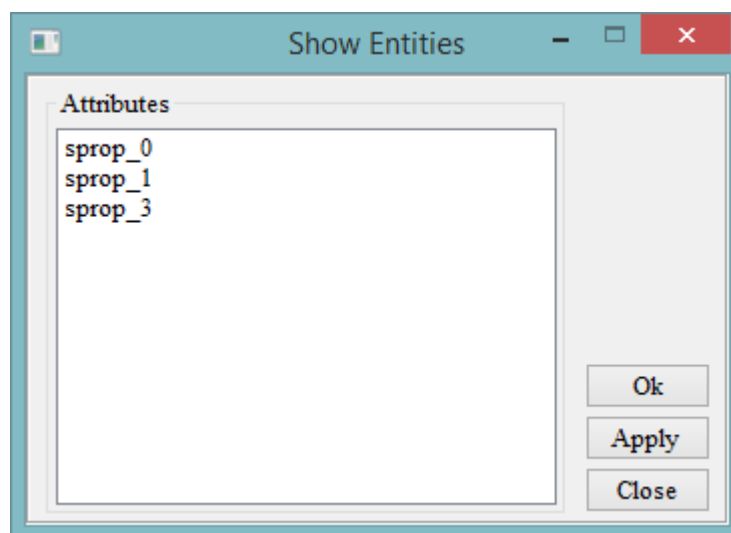


Figura 36 - Janela com as opções dos atributos.

A Figura 36 acima ilustra as opções de atributos criados.

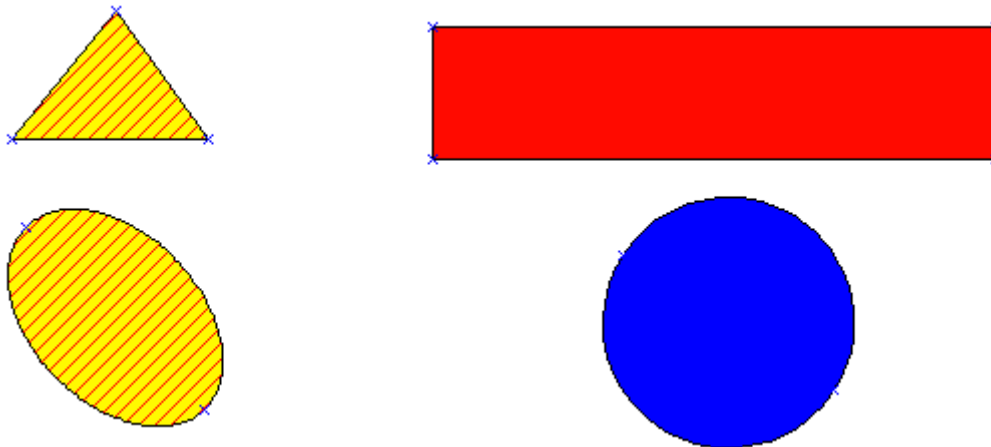


Figura 37 - Seleção de entidades através de atributos.

A Figura 37 acima ilustra a seleção das entidades, cujos os atributos associados são do tipo 1 (*sprop_0*), o que já foi especificado na janela da Figura 36.

3.3.12 – Remoção de entidades ou de atributos

Este comando serve tanto para remover entidades selecionadas como para remover atributos já criados. No primeiro caso, apenas o LABEL do comando é necessário. No segundo caso, o nome do atributo deve ser passado como parâmetro.

Command Line

<DELETE ou DEL ou D> [label]

label – nome do atributo que deve ser removido (apenas no caso de remoção de atributos).

Via Mouse

A remoção de entidades através do mouse é feita pela seleção da mesma e pelo clique com o botão esquerdo do mouse no botão cujo *tipo* é *Delete Entity*.



Figura 38 - Botão que remove entidades.

A Figura 38 acima ilustra o botão que deve ser selecionado para se remover um determinado tipo de entidade (*Region, Vertex, Line*).

A remoção de atributos através do mouse é feita através do preenchimento de campos englobados pelo seguinte frame.

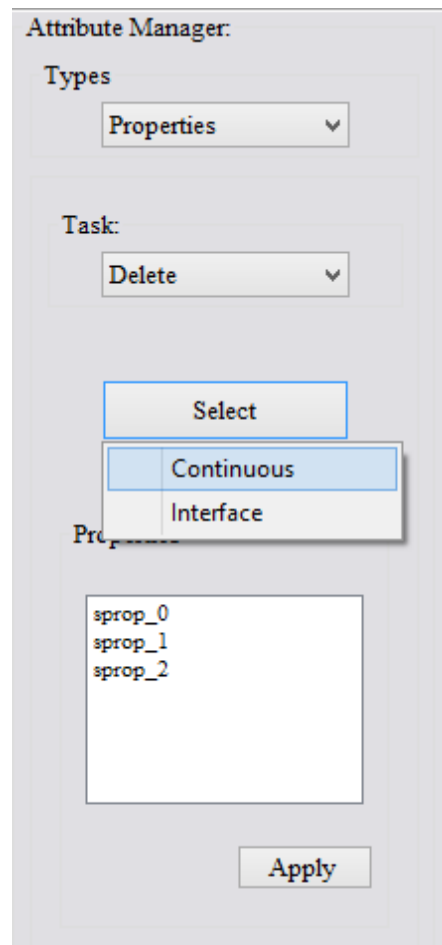


Figura 39 - Configuração dos campos do frame Attribute Manager para remover entidades.

A Figura 39 acima ilustra a configuração do frame *Attribute Manager* para remover atributos do tipo *Continuous*. Cada campo deste frame será explicado em outra ocasião.

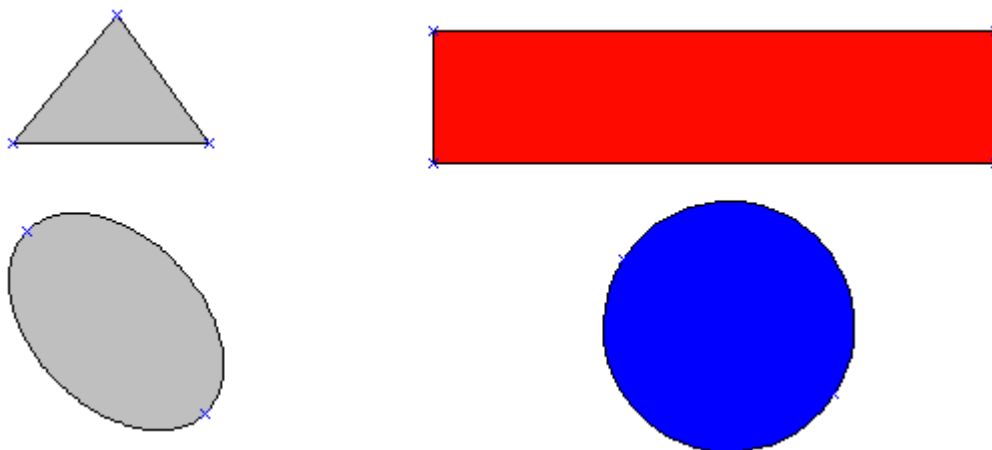


Figura 40 - Resultado após a remoção do atributo (*sprop_0*).

A Figura 40 acima ilustra o resultado após se remover o atributo *sprop_0* como mostra a Figura 39.

3.4 – Comando de ajuste de tolerância geométrica do modelo

Assim que o usuário aciona o comando para abrir a linha de comando do programa SIGMA, pode-se notar que a primeira tarefa que aparece na **TH** independentemente do modelo é o comando de ajuste de tolerância. Isso porque essa tarefa não é armazenada na **TH** apenas quando o usuário a digita, mas também toda vez que o programa tem de alterar internamente essa tolerância. Isso acontece normalmente quando algum ajuste é feito na área de visualização do modelo (*canvas*), como por exemplo quando o usuário aumenta ou diminui o *zoom*, quando é aplicado um fator de distorção, quando se aumenta a largura ou a altura de visualização, etc.

O motivo pelo qual esse comando é armazenado na **TH** e conseqüentemente no arquivo de backup gerado pelo programa é que, se eventualmente o arquivo de backup for importado posteriormente, o modelo será re-gerado exatamente da forma como foi criado, sem que se corra o risco de haver algum problema de atração ou seleção de entidades devido a um valor inadequado de tolerância. É importante ressaltar que atração por entidades segue uma hierarquia: primeiro, entidades do tipo *Vertex*, depois entidades do tipo *Line*, ou seja, caso a distância entre o mouse e um vértice seja menor que a distância entre o mouse e uma aresta e ambas sejam menor que a tolerância, o mouse será atraído para o vértice.

O usuário pode, manualmente, decidir qual deve ser o valor da tolerância geométrica usado pelo programa para a(s) próxima(s) tarefa(s) que deseja executar, através do seguinte comando:

Command Line

`<SETOLERANCE ou SETTOL> <tol>`

tol – valor numérico da tolerância geométrica

Via Mouse

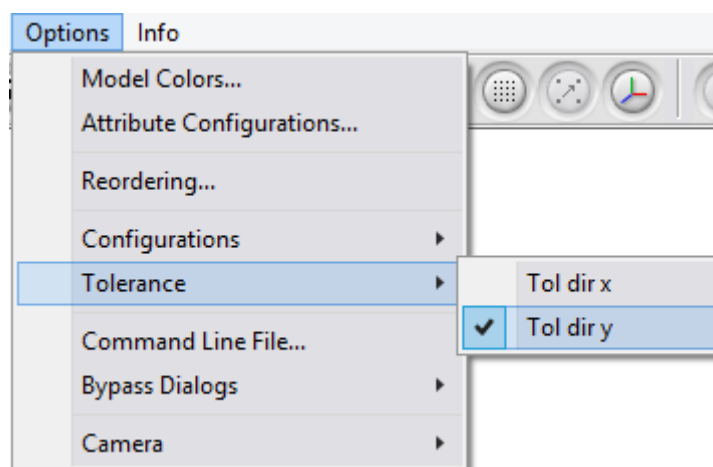


Figura 41 - Item para ajustar a tolerância.

A Figura 41 acima ilustra os itens que servem para se ajustar a tolerância.

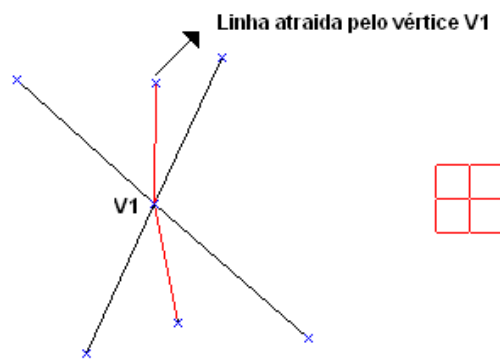


Figura 42 - Entidade do tipo *Line* atraída pela entidade do tipo *Vertex*.

A Figura 42 acima ilustra um exemplo que envolve a questão de tolerância, pois a linha destacada em vermelho foi desenhada, originalmente, reta e próxima ao vértice V1, porém devido ao fato da menor distância entre o vértice e a linha ser menor que a tolerância, a linha foi atraída pelo vértice. A figura também mostra o tamanho do cursor do mouse, que é proporcional a tolerância. Pode-se alterar a tolerância para valores pré-definidos pelo programa pressionando-se a tecla *ctrl* do teclado juntamente com o botão do meio do mouse.

Esse comando não garante que essa será a tolerância usada para o resto da modelagem. Se algum ajuste for feito na área de visualização do modelo, a tolerância será novamente recalculada.

3.5 – Comandos de subdivisão de curvas

Os comandos de subdivisão de curvas são habilitados caso haja alguma curva selecionada. O item que deve ser selecionado está ilustrado na figura abaixo. Para exemplificar os comandos de subdivisão de curvas, usou-se um retângulo, onde através deste serão feitas as subdivisões.

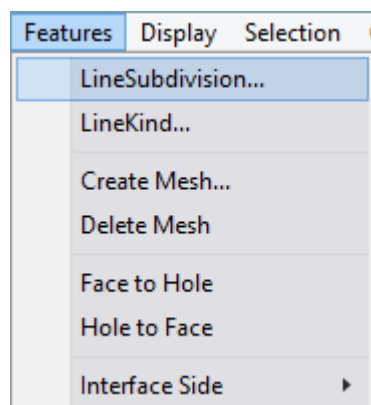


Figura 43 - Item para ajustar a tolerância.

A Figura 43 acima ilustra o item que deve ser selecionado para ativar o comando *Line Subdivision* para que se possa dividir entidades do tipo *Line*. Após este item ser selecionado será aberta uma janela com todas as opções do comando *Line Subdivision*.

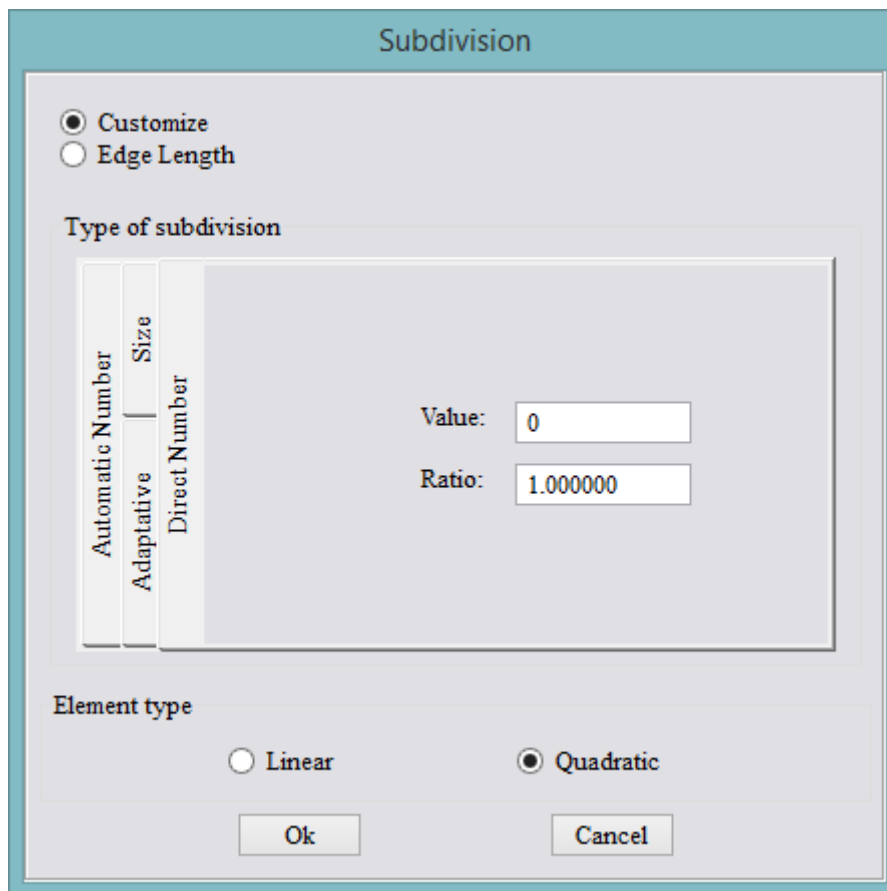


Figura 44 - Janela com todas as opções de subdivisão de linhas.

A Figura 44 acima ilustra a janela aberta após se selecionar o comando *Line Subdivision* da Figura 43, os campos preenchidos na figura são padronizados (*default*), porém eles podem ser adaptados de acordo com a necessidade.

3.5.1 – Subdivisão de curva por número de subdivisões

As entidades do tipo *Line* serão divididas de acordo com o número de subdivisões determinado pelo usuário.

Command Line

$\langle SDVNUM \text{ ou } SNUM \rangle \langle nsdv, ratio, sdvtype \rangle$

nsdv – número de subdivisões da curva.

ratio – razão entre o tamanho das subdivisões.

sdvtype – tipo de subdivisão da curva (0 – LINEAR, 1 – QUADRÁTICA).

Via Mouse

Neste modo, a divisão será feita de acordo com o número de trechos especificado no campo *value*.

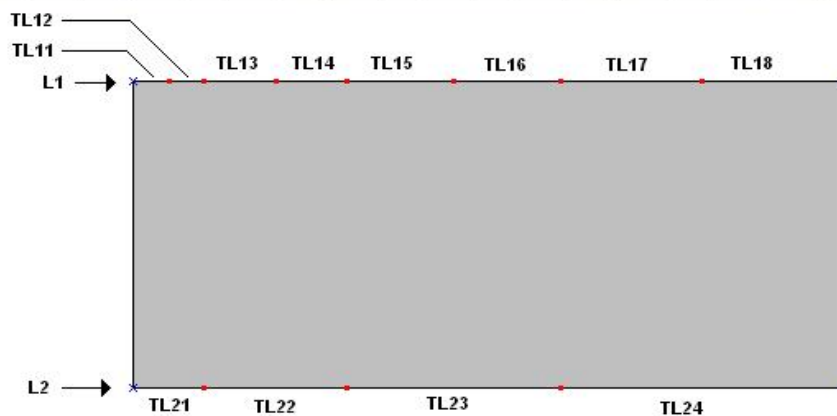


Figura 45 - Resultado da subdivisão de duas entidades do tipo linha.

L1 e *L2* são respectivamente duas entidades do tipo linha. *TLij*, representam trechos das entidades do tipo linha, onde *i* é referente a linha e *j* é referente ao trecho.

A Figura 45 acima ilustra o resultado após se selecionar as entidades *L1* e *L2*. Para a entidade *L1* os campos foram preenchidos da seguinte forma: togle *Customize* selecionado, *Number* selecionado, *value* = 4, *ratio* = 0,25 e *Quadratic* selecionado. Já para a entidade *L2* os campos foram preenchidos da seguinte forma: togle *Customize* selecionado, *Number* selecionado, *value* = 4, *ratio* = 0,25 e *Linear* selecionado, ou seja, o número de trechos de *L1* é o dobro de *L2*, com a razão entre os trechos, em ambas as linhas, diminuindo pelo fator de 0,25. Caso *ratio* fosse igual a 1, os trechos de cada linha seriam do mesmo tamanho.

3.5.2 – Subdivisão de curva por tamanho

As entidades do tipo *Line* serão divididas especificando-se o tamanho dos trechos. Caso o tamanho do trecho seja maior que o tamanho da entidade, esta não será dividida.

Command Line

<SDVSIZ ou *SSIZ*> *<size, sdvtype>*

size – tamanho mínimo da subdivisão. No caso deste tamanho ser maior que o comprimento total da curva, esta não é subdividida.

sdvtype – idem ao caso anterior.

Via Mouse

Para se selecionar este tipo de subdivisão tem que se clicar no fichário *Size* da Figura 44. Posteriormente, determina-se a distância entre as subdivisões (tamanho de cada trecho).

3.5.3 – Subdivisão de curva por comprimento mínimo de curva

Cria-se um comprimento mínimo de uma curva de referência (hipotética). Nenhuma curva selecionada será subdividida com um número de subdivisões menor que o dessa curva de referência. O número de subdivisões de uma curva qualquer selecionada será a potência de dois imediatamente menor que a razão entre o comprimento dessa curva e o comprimento da curva de referência, multiplicado pelo número de subdivisões da curva de referência.

Command Line

$\langle SDVMIN \text{ ou } SMIN \rangle \langle smin, nsdv, sdvtype \rangle$

smin – comprimento mínimo de uma curva de referência (hipotética).

nsdv – número de subdivisões da curva de referência.

sdvtype – idem ao caso anterior.

Via Mouse

Para se ativar este comando através do mouse, seleciona-se o toggle *Edge Length* na Figura 44 e posteriormente o toggle *Define smallest edge size* da Figura 46 abaixo.

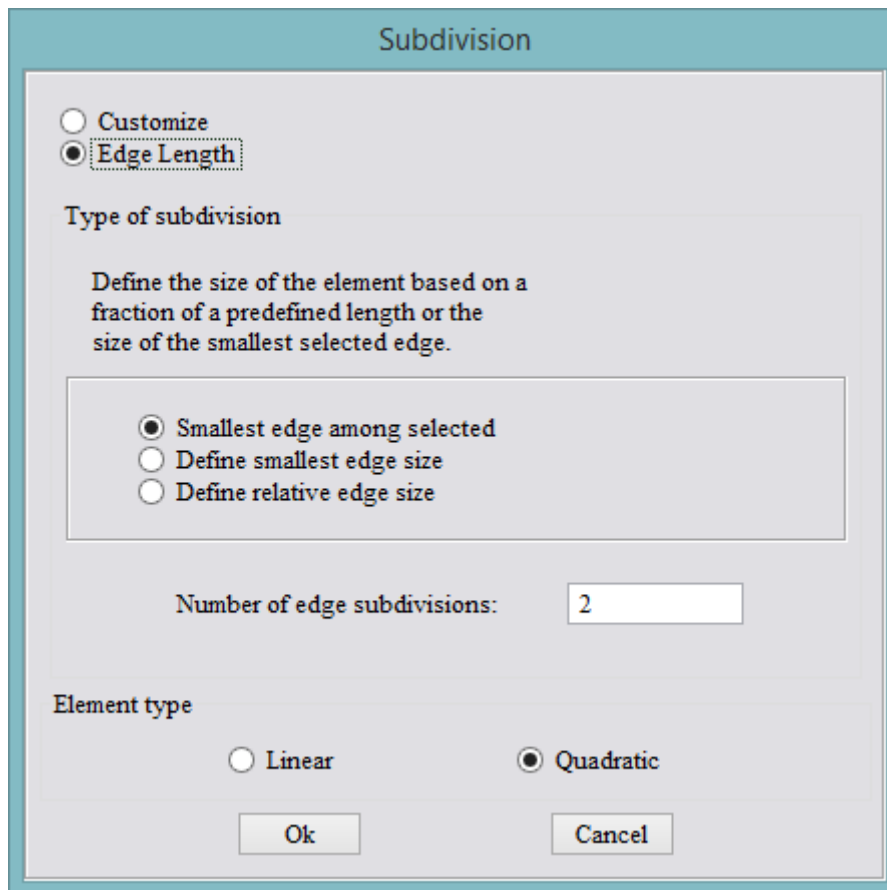


Figura 46 - Janela aberta ao se seleccionar o toggle *Edge Length*.

A Figura 46 acima ilustra a janela que é aberta ao se seleccionar o toggle *Edge Length*, onde se selecciona o toggle *Define smallest edge size* e se determina o comprimento mínimo no campo *Minimum length*.

3.5.4 – Subdivisão de curva por comprimento relativo de curva

Cria-se um comprimento relativo de uma curva de referência (hipotética). Se uma curva qualquer seleccionada tiver um comprimento maior que o da curva de referência, o número de subdivisões será calculado como no caso acima. Se o comprimento da curva seleccionada for menor que o da curva de referência, o seu número de subdivisões será igual à potência de dois imediatamente maior que a razão entre o comprimento dessa curva e o comprimento da curva de referência, multiplicado pelo número de subdivisões da curva de referência. Se este número for fracionário, ele é truncado (arredondado para baixo).

Command Line

$\langle SDVREL \text{ ou } SREL \rangle \langle srel, nsdv, sdvtype \rangle$

srel – comprimento relativo de uma curva de referência (hipotética).

nsdv – número de subdivisões da curva de referência.

sdvtype – idem ao caso anterior.

Via Mouse

Para se ativar este comando através do mouse seleciona-se o togle *Edge Length* na Figura 44, posteriormente o togle *Define relative edge size* da Figura 46 e preenche-se o campo *Minimum length*.

3.5.5 – Subdivisão de curva por comprimento da menor curva selecionada

Command Line

`<SDVSEL ou SSEL> <nsdv, sdvtype>`

nsdv – número de subdivisões da curva selecionada que possui o menor comprimento. As outras curvas selecionadas serão subdivididas conforme as mesmas regras ditadas para o caso anterior.

sdvtype – idem ao caso anterior.

Via Mouse

Para se ativar este comando através do mouse seleciona-se o togle *Edge Length* na Figura 44, posteriormente o togle *Smallest edge among select* da Figura 46.

3.6 – Comandos de geração e remoção de malhas

Para se gerar uma malha é necessário selecionar uma entidade do tipo *Region* e o seguinte item.

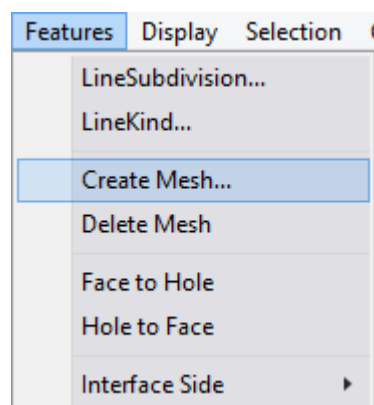


Figura 47 – Item responsável pela criação da malha.

A Figura 47 acima ilustra o item responsável para o início dos requisitos para a geração da malha. Após selecionar o item *Create Mesh* será aberta uma janela que contém os tipos de elementos que podem ser gerados para cada malha.

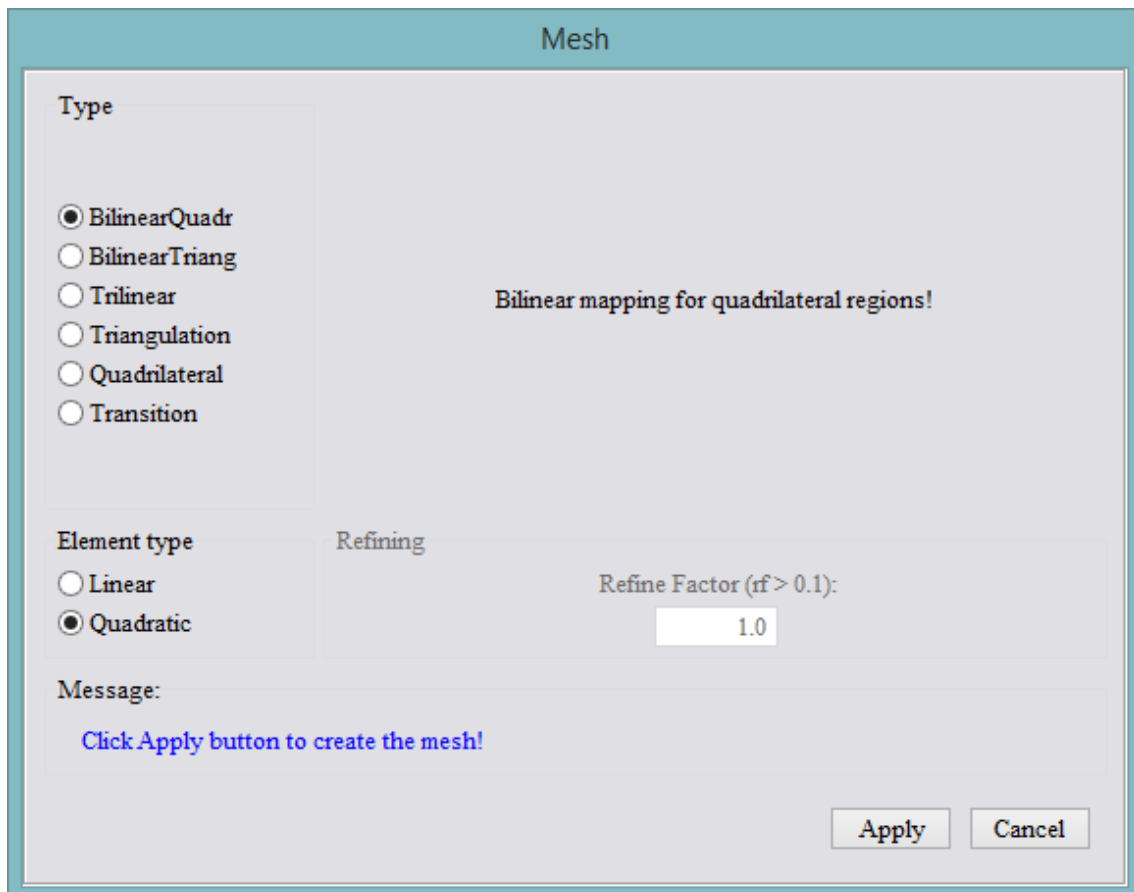


Figura 48 - Janela com os tipos de elementos finitos.

A Figura 48 acima ilustra a janela, que é aberta após se selecionar o item *Create Mesh* na Figura 47, contendo os tipos de elementos finitos que podem ser usados na malha.

3.6.1 – Geração de malha bilinear quadrática

Command Line

<BILQUAD ou BQ> <sdvtype>

sdvtype – tipo de elemento (0 – LINEAR, 1 – QUADRÁTICO).

Via Mouse

Seleciona-se o togle *BilinearQuadr* na janela da Figura 48.

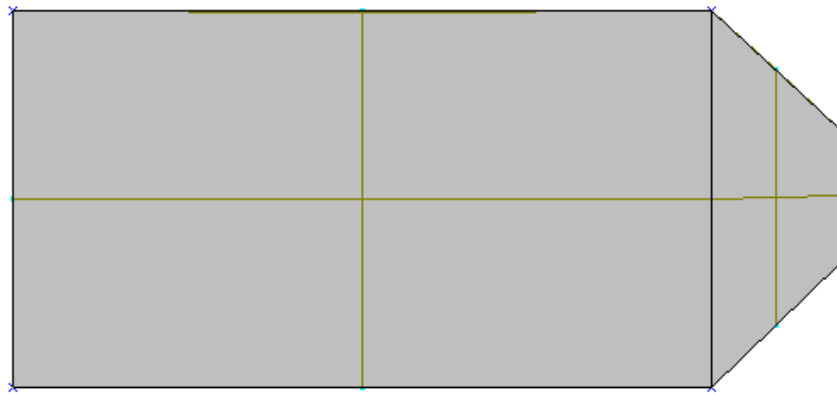


Figura 49 - Malha com elementos do tipo BilinearQuadr.

A Figura 49 acima ilustra uma malha com elementos do tipo *BilinearQuadr*.

3.6.2 – Geração de malha bilinear triangular

Command Line

<BILTRI ou BT> <method, sdvtype>

method – método de geração da malha:

- 1 – LEFT CROSS.
- 2 – RIGHT CROSS.
- 3 – BEST CROSS.
- 4 – UNION JACK.

sdvtype – idem ao caso anterior.

Via Mouse

Seleciona-se o togle *BilinearTriang* na janela da Figura 48.

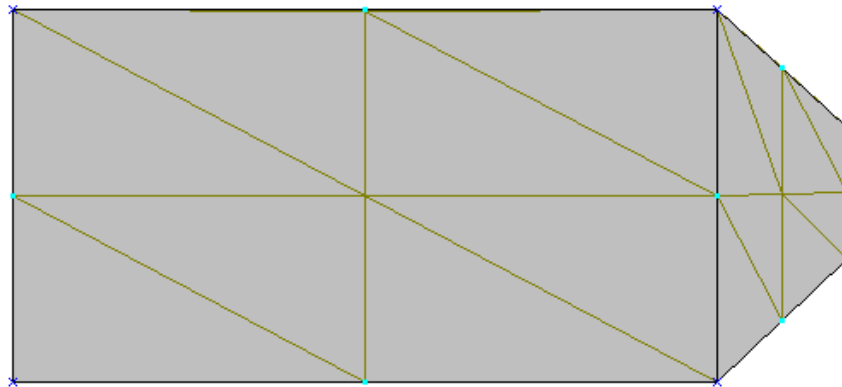


Figura 50 - Malha com elementos do tipo BilinearTriang.

A Figura 50 acima ilustra uma malha com elementos do tipo *BilinearTriang*.

3.6.3 – Geração de malha trilinear

Este tipo de malha só pode ser gerado para modelos triangulares por isso foram intruzidas mais cinco entidades do tipo *Line*. É importante ressaltar que é necessário que os lados do triângulo tenham o mesmo número de trechos.

Command Line

`<TRILINEAR ou TRIL ou TL> <sdvtype>`

sdvtype – idem ao caso anterior.

Via Mouse

Seleciona-se o toggle *Trilinear* na janela da Figura 48. No exemplo da figura abaixo teve uma região em que não foi possível gerar uma malha *Trilinear* justamente porque os seus lados não têm o mesmo número de trechos.

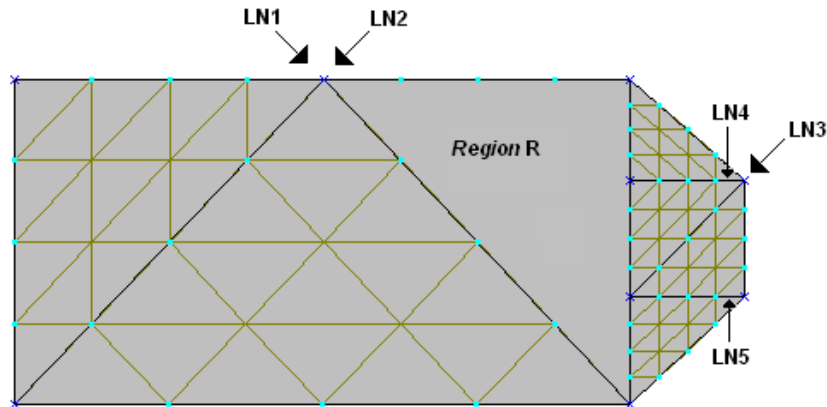


Figura 51 - Malha com elementos do tipo Trilinear.

A Figura 51 acima ilustra uma malha com elementos do tipo *Trilinear*, onde *LN* são as novas entidades do tipo *Line* que foram introduzidas para poder gerar a malha do tipo *Trilinear* e a entidade *R* do tipo *Region* em que não se pôde gerar a malha.

3.6.4 – Geração de malha por triangulação

Command Line

<TRIANGULATION ou TRIANG ou TRI> <method, sdvtype, reffactor>

method – método de geração da malha:

1 – BOUNDARY CONTRACTION.

2 – BOUNDARY CONTRACTION COMBINED WITH QUADTREE.

3 – BOUNDARY CONTRACTION WITH INTERNAL POINTS GENERATION (QUADTREE).

4 - BOUNDARY CONTRACTION WITH INTERNAL POINTS GENERATION (GRID). 5 – NO INTERNAL POINTS.

sdvtype – idem ao caso anterior.

reffactor – fator de refinamento.

Via Mouse

Seleciona-se o toggle *Triangulation* na janela da Figura 48.

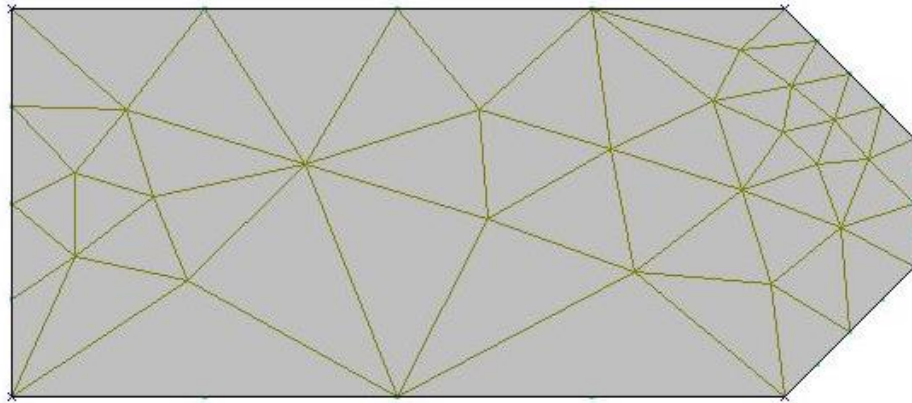


Figura 52 - Malha com elementos do tipo Triangulation.

A Figura 52 acima ilustra uma malha com elementos do tipo *Triangulation*.

3.6.5 – Geração de malha quadrilateral

Command Line

<QUADRILATERAL ou QUAD ou Q> <method, sdvtype, reffactor>

method – método de geração da malha:

- 1 – QUADTREE WITH BOUNDARY CONTRACTION (STANDARD).
- 2 – QUADTREE WITH BOUNDARY CONTRACTION (REFINED).
- 3 – TRIANGULATION TO QUADRILATERAL SEAM (VERY SLOW).

sdvtype – idem ao caso anterior.

reffactor – fator de refinamento.

Via Mouse

Seleciona-se o toggle *Quadrilateral* na janela da Figura 48.

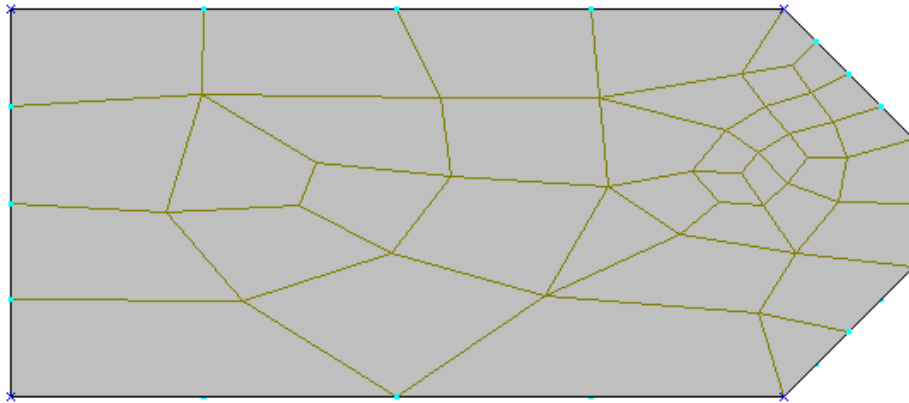


Figura 53 - Malha com elementos do tipo Quadrilateral.

A Figura 53 acima ilustra uma malha com elementos do tipo *Quadrilateral*, é importante ressaltar que esse tipo de elemento só pode ser gerado se as entidades do tipo *Line* forem divididas em um número par de trechos.

3.6.6 – Geração de malha de transição

Command Line

<TRANSITION ou TRANS ou TR> <pattern, sdvtype>

pattern – número inteiro, de 1 a 5, correspondente ao tipo de *pattern* desejado.

sdvtype – idem ao caso anterior.

Via Mouse

Seleciona-se o toggle *Transition* na janela da Figura 48.

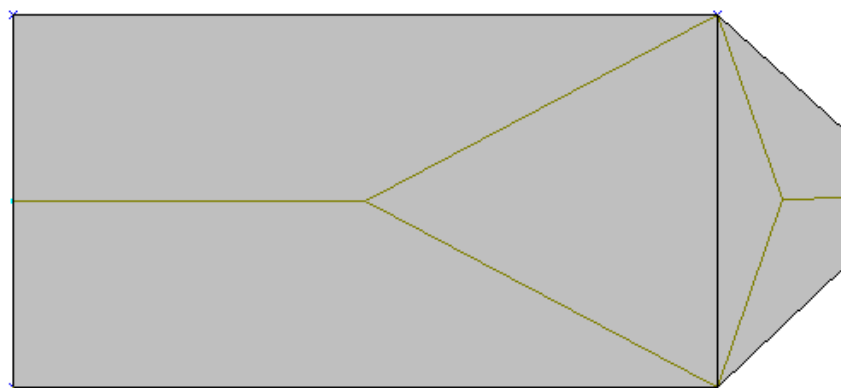


Figura 54 - Malha com elementos do tipo Transition.

A Figura 54 acima ilustra uma malha com elementos do tipo *Transition*. É importante ressaltar que esse tipo de elemento só pode ser gerado se uma entidade do tipo *Line* for dividida em duas e a outra, oposta a esta, não for dividida.

3.6.7 – Remoção de malha

Command Line

<DELETEMESH ou DELMESH ou DM>

Via Mouse

Para se usar este comando é necessário que se tenha selecionado a malha e clicar no seguinte item da barra de menu.

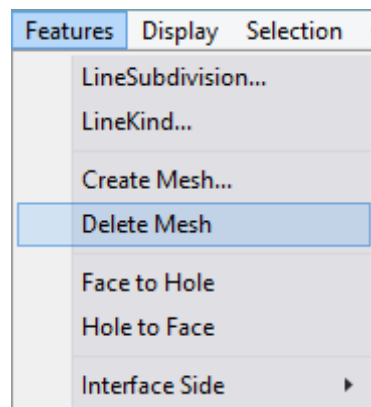


Figura 55 - Item que remove a malha.

A Figura 55 acima ilustra o item que deve ser acionado para se remover a malha.

3.7 – Comandos de manipulação das características das entidades

3.7.1 – Tipo de curva

Este comando serve para definir o tipo de curva relacionado a entidade do tipo *Line*, há três possibilidades: ser apenas uma curva geométrica feita para o desenho do modelo (*Geometry Line*), uma curva que separa uma parte do modelo do seu contorno infinito (*Infinite Boundary*) ou uma curva que é a interface do elemento (*Interface Element*).

Command Line

<LINEKIND ou LK> <kind>

kind – tipo da curva:

“GEOMLINE” ou “GEO”

“INFBOUND” ou “INF”

“INTERFELEM” ou “INT”

Via Mouse

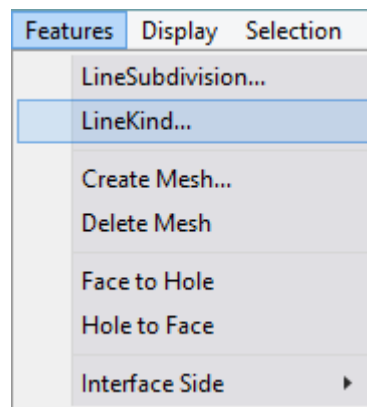


Figura 56 - Item que define o tipo de curva.

A Figura 56 acima ilustra o item *LineKind* que deve ser acionado para definir o tipo de curva. Após se clicar neste item será aberta uma janela (ver Figura 57) com as três opções de curva.

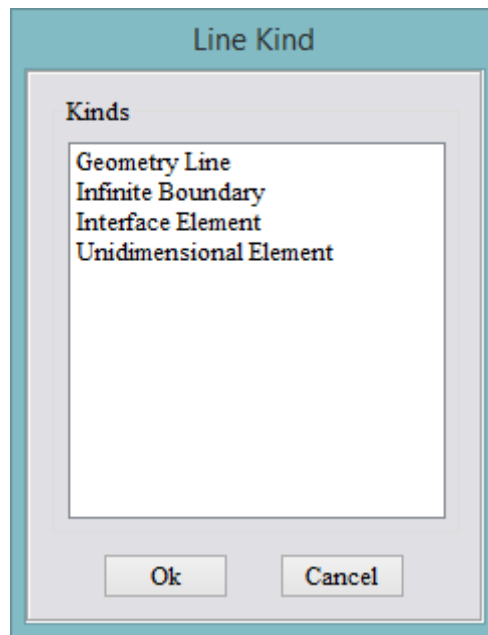


Figura 57 - Janela com os três tipos de curvas.

A Figura 57 acima ilustra a janela que é aberta após se selecionar o item *LineKind* da Figura 56 com suas quatro opções.

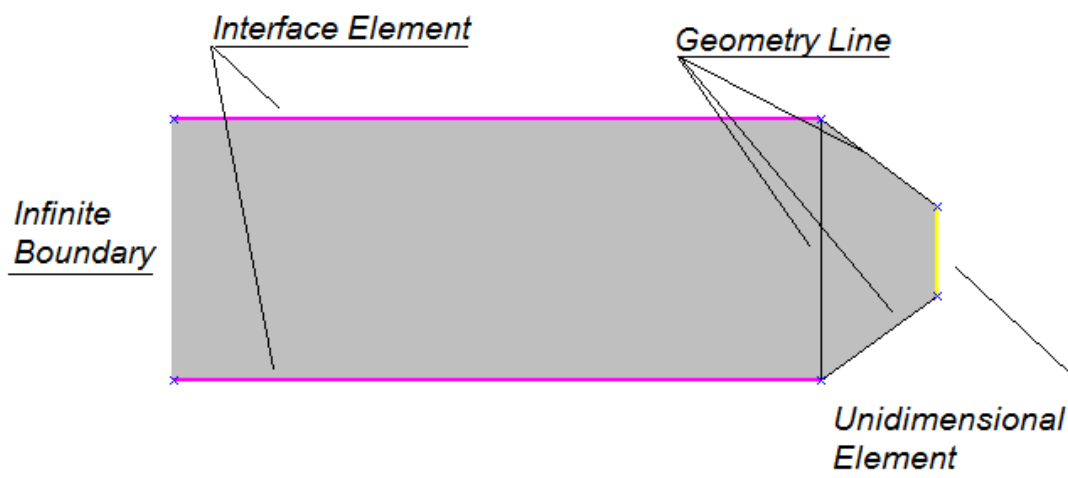


Figura 58 - Malha com os tipos de curvas definidos.

A Figura 58 acima ilustra os quatro tipos de curvas. Em rosa estão as curvas do tipo *InterfaceElement*, de preto estão as *Geometry Line*, de cinza as *Infinite Boundary* e de amarelo a *Unidimensional Element*.

3.7.2 – Transformação de face em orifício

Command Line

<FACETHOLE ou F2H>

Via Mouse

Selecione-se o item *Face to Hole* da figura abaixo.

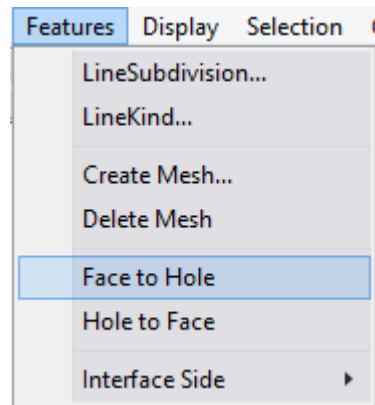


Figura 59 - Transformação de face em orifício.

A Figura 59 acima ilustra o item que deve ser selecionado para se transformar uma entidade do tipo *Region* em *Hole*.

Para exemplificar este comando adicionou-se um círculo no modelo da Figura 58, mas o comando também é válido para qualquer outra entidade geométrica do tipo *Region*.

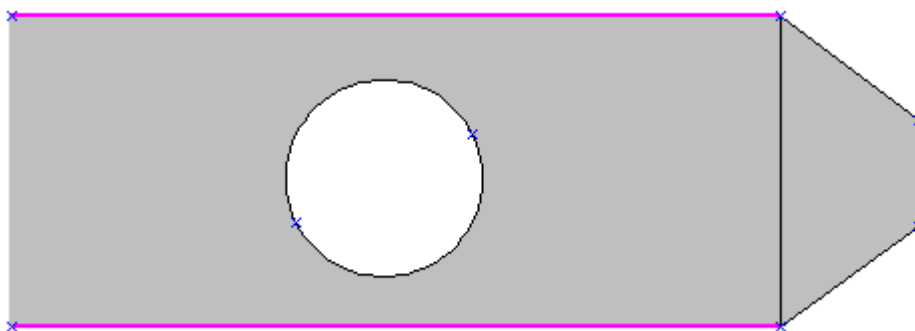


Figura 60 - Transformação de face em orifício.

A Figura 60 acima ilustra a transformação do círculo adicionado no modelo da Figura 58 em orifício.

3.7.3 – Transformação de orifício em face

Command Line

<HOLETOFACE ou H2F>

Via Mouse

Seleciona-se na Figura 59 o item *Hole to Face*.

Para exemplificar este comando adicionou-se um círculo no modelo da Figura 58.

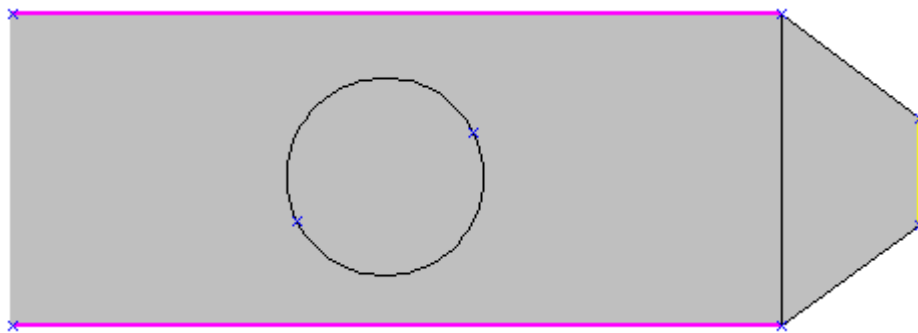


Figura 61 - Transformação de orifício em face.

A Figura 61 acima ilustra a transformação de *Hole* para *Face*.

3.7.4 – Definir lado de uma curva de interface a partir de uma face adjacente

Command Line

<FROMFACE ou FF>

Via Mouse

Seleciona-se o item *Interface Side From Face* da figura abaixo.

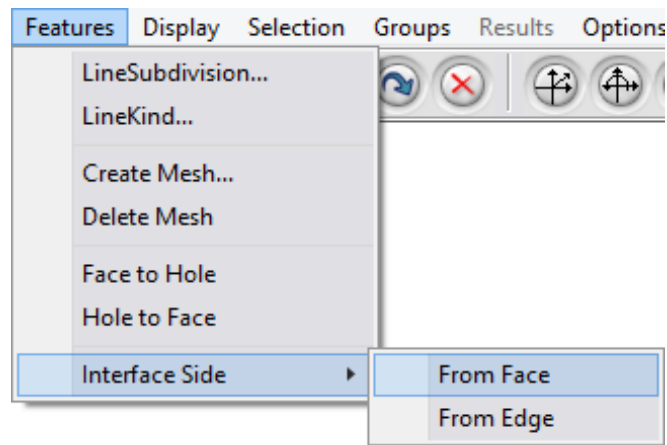


Figura 62 - Ativar o comando Interface Side From Face.

A Figura 62 acima ilustra como ativar o comando *Interface Side From Face*.

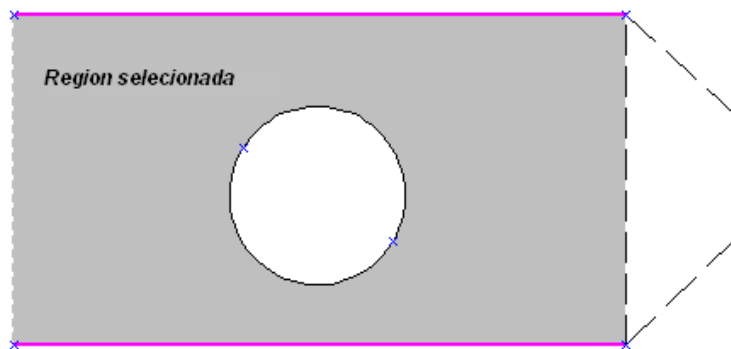


Figura 63 - Resultado após se ativar o comando Interface Side From Face.

A Figura 63 acima ilustra o resultado após se ativar o comando *Interface Side From Face*, tendo-se selecionado a entidade do tipo *Region* mostrada na figura.

Cada vez que esse comando é acionado, este módulo é ligado ou desligado, alternadamente.

3.7.5 – Definir lado de uma curva de interface adjacente ao exterior

Command Line

<FROMEDGE ou FE>

Via Mouse

Seleciona-se o item *Interface Side From Edge* da Figura 62.

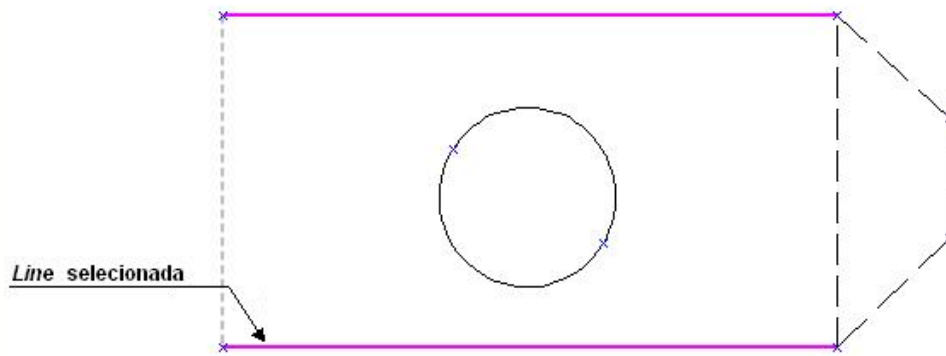


Figura 64 - Resultado após se ativar o comando *Interface Side From Edge*.

A Figura 64 acima ilustra o resultado após se ativar o comando *Interface Side From Edge*, tendo-se selecionado a entidade do tipo *Line* mostrada na figura.

Cada vez que esse comando é acionado, este módulo é ligado ou desligado, alternadamente.

3.8 - Comandos de visualização

Todos os comandos de visualização são acionados a partir de cada item da figura abaixo, presentes no menu *Display*.

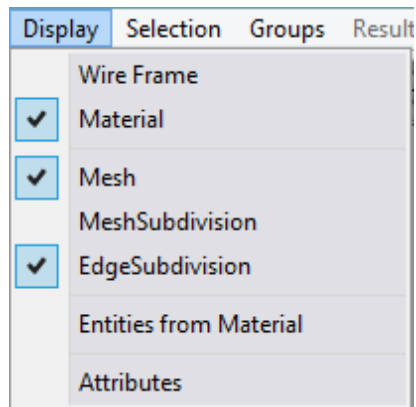


Figura 65 - Submenu com todos os itens dos comandos de visualização.

A Figura 65 acima ilustra o submenu com todos os itens do comando de visualização.

3.8.1 – Visualização por arame (Wireframe)

Command Line

<WIREFRAME ou WF>

Via Mouse

Seleciona-se o item *Wire Frame* na Figura 65.

3.8.2 – Visualização por material

Command Line

<MATERIAL ou MAT>

Via Mouse

Seleciona-se o item *Material* na Figura 65.

3.8.3 – Visualização de malhas

Command Line

<MESH ou M>

Via Mouse

Selecione-se o item *Mesh* na Figura 65.

Cada vez que esse comando é acionado, o módulo de visualização de malhas é ligado ou desligado, alternadamente.

3.8.4 – Visualização de subdivisão de malhas

Command Line

<MESHSDV ou MS>

Via Mouse

Selecione-se o item *MeshSubdivision* na Figura 65.

Cada vez que esse comando é acionado, o módulo de visualização de subdivisão de malhas é ligado ou desligado, alternadamente.

3.8.5 – Visualização de subdivisão de curvas

Command Line

<CURVESDV ou CS>

Via Mouse

Selecione-se o item *EdgeSubdivision* na Figura 65.

Cada vez que esse comando é acionado, o módulo de visualização de subdivisão de curvas é ligado ou desligado, alternadamente.

3.8.6 – Visualização de entidades por material

Command Line

<ENTFROMMAT ou EFM> <prop1, prop2, ..., propn>

Via Mouse

Seleciona-se o item *Entities from Material* na Figura 65.

prop1 – nome do primeiro atributo do tipo “Properties” (propriedades do material) cujas entidades que o agregam devem ser visualizadas

prop2 – nome do segundo atributo do tipo “Properties” (propriedades do material) cujas entidades que o agregam devem ser visualizadas

.
.
.

propn - nome do último (n-ésimo) atributo do tipo “Properties” (propriedades do material) cujas entidades que o agregam devem ser visualizadas

Existe também a opção de se visualizar as entidades que agregam TODOS os tipos de material sem ter de listar o nome de todos eles, usando-se a expressão “ALL” no lugar dos nomes dos atributos. A opção de não visualizar nenhuma entidade que possui algum tipo de material associado também pode ser executada usando-se a expressão “NONE”.

3.8.7 – Redesenho

Command Line

<REDRAW ou RD>

Via Mouse



Figura 66 - Botão do comando *Redraw*.

A Figura 66 acima ilustra o botão referente ao comando *Redraw*.

3.8.8 – Ajuste do modelo à janela de visualização

Command Line

<FIT>

Via Mouse



Figura 67 - Botão do comando *Fit*.

A Figura 67 acima ilustra o botão referente ao comando *Fit*.

3.8.9 – Aumento de zoom

Command Line

<ZOOMIN ou Z+>

Via Mouse



Figura 68 - Botão do comando *Zoomin*.

A figura acima ilustra o botão referente ao comando *Zoomin*.

3.8.10 – Diminuição de zoom

Command Line

<ZOOMOUT ou Z->

Via Mouse



Figura 69 - Botão do comando *ZoomOut*.

A Figura 69 acima ilustra o botão referente ao comando *ZoomOut*.

3.8.11 – Último zoom

Command Line

<ZOOMLAST ou ZL>

Via Mouse



Figura 70 - Botão do comando *ZoomLast*.

A Figura 70 acima ilustra o botão referente ao comando *ZoomLast*.

3.8.12 – Próximo zoom

Este comando equivale ao comando de aumento de zoom, só que o usuário define o ponto que ficará no centro da janela de visualização.

Command Line

$\langle ZOOMNEXT \text{ ou } ZN \rangle \langle px, py \rangle$

px – abscissa do ponto central da janela de visualização.

py – ordenada do ponto central da janela de visualização.

Via Mouse

Este comando é acionado após se clicar no botão da figura abaixo e der um clique com o botão esquerdo do mouse no canvas.



Figura 71 - Botão do comando *ZoomNext*.

A Figura 71 acima ilustra o botão referente ao comando *ZoomNext*.

3.8.13 – Zoom por cerca

Command Line

$\langle ZOOMFENCE \text{ ou } ZF \rangle \langle xmin, ymin, xmax, ymax \rangle$

$xmin$ – abscissa mínima do retângulo que limita a região de visualização que será ampliada, ajustando-se à janela de visualização.

$ymin$ – ordenada mínima do retângulo que limita a região de visualização que será ampliada, ajustando-se à janela de visualização.

$xmax$ – abscissa máxima do retângulo que limita a região de visualização que será ampliada, ajustando-se à janela de visualização.

$ymax$ – ordenada máxima do retângulo que limita a região de visualização que será ampliada, ajustando-se à janela de visualização.

Via Mouse

Este comando é ativado também através do botão da Figura 71. Posteriormente, seleciona-se uma área, através do mouse, dentro do canvas. A diferença entre o comando *ZoomNext* e o comando *ZoomFence* é que neste se determina uma área e no outro basta um clique com o botão esquerdo do mouse dentro do canvas

3.8.14 – Visualização de grade

Command Line

<GRID ou G>

Via Mouse



Figura 72 - Botão do comando *Grid*.

A Figura 72 acima ilustra o botão referente ao comando *Grid*.

Cada vez que esse comando é acionado, o módulo de visualização de grade é ligado ou desligado, alternadamente.

3.8.15 – Atração para a grade

Command Line

<SNAP ou SN>

Via Mouse



Figura 73 - Botão do comando *Snap*.

A Figura 73 acima ilustra o botão referente ao comando *Snap*.

Cada vez que esse comando é acionado, o módulo de visualização de atração para grade é ligado ou desligado, alternadamente.

3.8.16 – Visualização de eixos coordenados

Command Line

<AXIS ou AX>

Via Mouse



Figura 74 - Botão do comando *Axis*.

A Figura 74 acima ilustra o botão referente ao comando *Axis*.

Cada vez que esse comando é acionado, o módulo de visualização de eixos é ligado ou desligado, alternadamente.

3.8.17 – Largura de visualização

Command Line

<WIDTH ou W> <width>

width – valor da largura da janela de visualização

Via Mouse

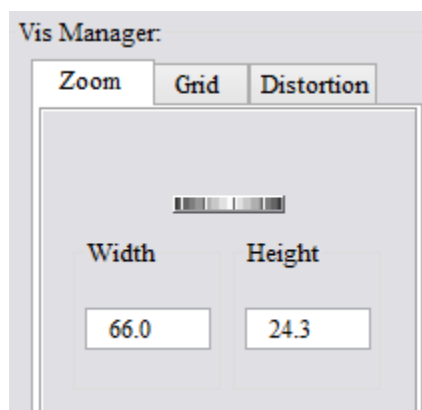


Figura 75 - Campos do frame *Vis Manager*.

A Figura 75 acima ilustra os campos do frame *Vis Manager*, onde pode se ajustar o valor de *Width* preenchendo-se o campo correspondente a tal comando na figura.

3.8.18 – Altura de visualização

Command Line

<HEIGHT ou H> <height>

height – valor da altura da janela de visualização

Via Mouse

Ajusta-se o valor de *Height* preenchendo-se o campo correspondente a tal comando na Figura 75.

3.8.19 – Passo da grade (distância entre pontos consecutivos da grade)

Command Line

<GRIDSTEP ou GST> <step>

step – valor do passo da grade

Via Mouse

Ajusta-se o valor de *Grid* preenchendo-se o campo correspondente a tal comando na Figura 75.

3.8.20 – Fator de distorção (Y/X)

Command Line

<DISTFAC ou DF> <factor>

factor – valor do fator de distorção (y/x)

Via Mouse

Ajusta-se o valor de *Dist(Y/X)* preenchendo-se o campo correspondente a tal comando na Figura 75.

3.8.21 – Fator de escala

Command Line

$\langle SCFAC \text{ ou } SCF \rangle \langle factor \rangle$

factor – valor do fator de escala

Via Mouse

Este comando é ativado pelo clique do mouse e de seu arrasto para direita ou para esquerda sobre o potenciômetro que está dentro do frame da Figura 75. Se o arrasto do mouse for para esquerda se estará diminuindo a escala para o lado contrário se estará aumentando a escala.



Figura 76 - Potenciômetro para ajustar o fator de escala.

A Figura 76 acima ilustra o potenciômetro que está dentro do frame da Figura 75 responsável pelo ajuste do comando do *Fator de escala*.

3.8.22 – Atribuição de cores do modelo

Command Line

$\langle SETCOLOR \text{ ou } COLOR \text{ ou } COL \rangle \langle field, r, g, b \rangle$

field – tipo de campo do modelo cuja cor deve ser alterada (os nomes desses campos devem ser escritos entre aspas e corretamente, da mesma forma como estão escritos no diálogo que é aberto na interface do programa quando se vai em “Options” e “Model Colors”).

r – valor entre 0 e 255 representando a intensidade da cor básica vermelho (*red*).

g – valor entre 0 e 255 representando a intensidade da cor básica verde (*green*).

b – valor entre 0 e 255 representando a intensidade da cor básica azul (*blue*).

Via Mouse

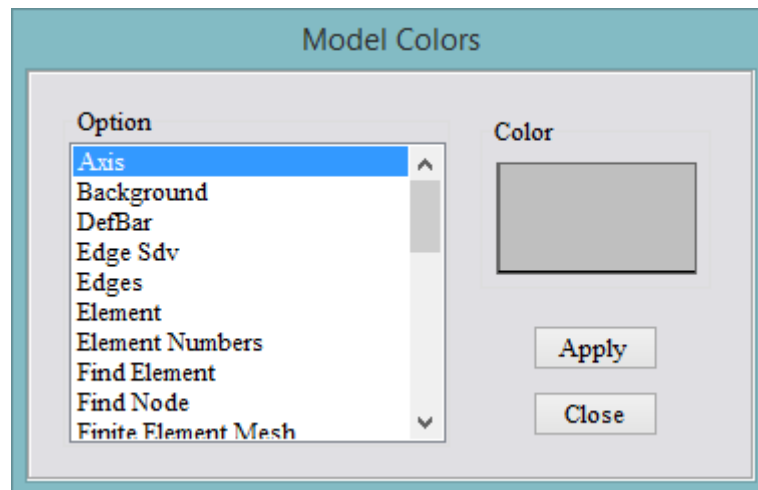


Figura 77 - Janela com os ajustes de cores.

A Figura 77 acima ilustra a janela que ajusta as cores dos comandos do programa e que é aberta após se selecionar o item *Model Colors* do menu *Options*.

3.9 – Gerenciamento de grupos

O Sigma2D fornece um meio de realizar uma certa operação uma só vez de modo que afete diversas entidades de um mesmo grupo. Esta funcionalidade recebe justamente esse nome: grupo. Grupos são formados por entidades de um mesmo tipo, logo, há três tipos de grupos: grupos formados por entidades do tipo vértice, curva e região.

3.9.1 – Criando um grupo

O primeiro passo para utilizar esta funcionalidade é definir entidades que pertencerão ao grupo. Isto inclui definir o tipo de grupo a ser criado, o seu nome e as entidades que o compõem.

Command Line

<SETGROUP ou SG> <Name, entity type, entities>

NAME - nome do grupo a ser definido,

ENTITY TYPE - tipo de entidade (0 - vertex, 1 - curvas, 2 - faces),

ENTITIES - números das entidades a serem adicionadas ao grupo.

Labels não são necessárias, mas a definição deve ser feita na ordem indicada.

Exemplos (exemplos a seguir levam em consideração a criação de duas regiões circulares):

Ex1: (criar dois círculos) sg Name = "Teste123", ET = "2", EN1 = "2", EN2 = "3"

Ex2: (criar duas retas) SETGROUP "teste", "1", "1", "2"

Via Mouse

Para utilizar esta ferramenta via mouse, acesse o menu Groups e clique no item "Set Group...". O diálogo de criação de grupos aparecerá (Figura 78). Caso as entidades que se deseja adicionar ao grupo já estejam selecionadas, basta definir o nome do grupo e aplicar (o toggle "Add selected entities" estará ativo como default). Caso contrário, selecione o toggle "Choose entities from list", e uma lista de entidades aparecerá, de acordo com o tipo de entidades desejado. Por fim, clique em Apply caso queira definir mais algum grupo ou Ok caso contrário.

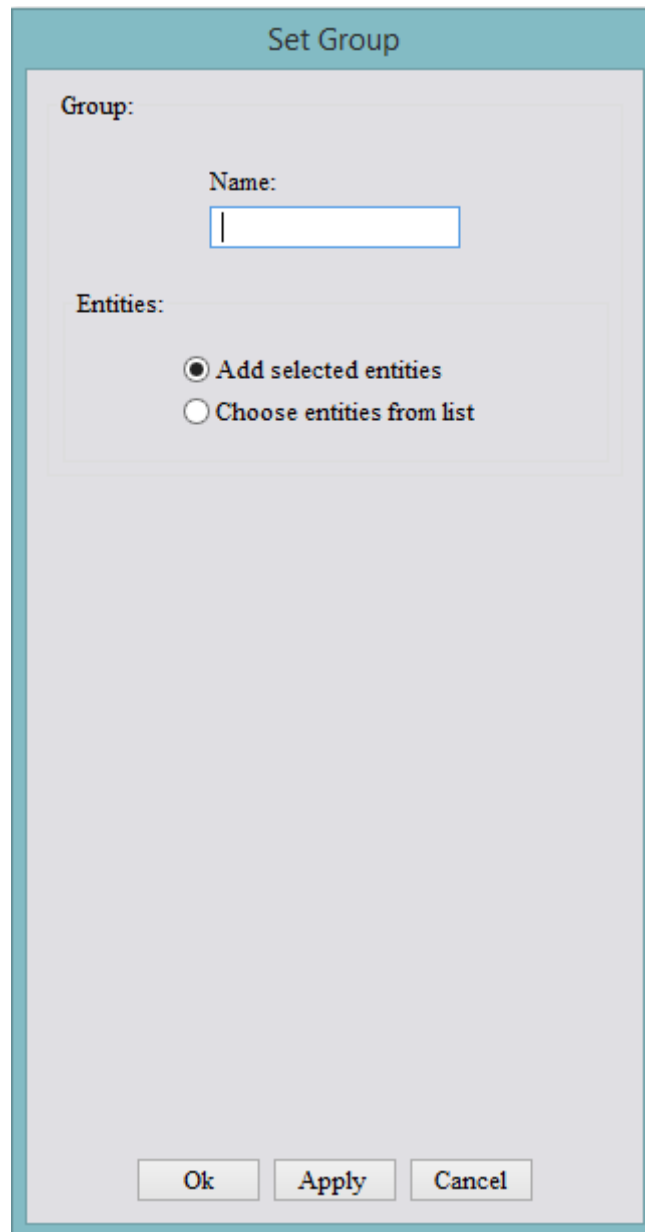


Figura 78 - Diálogo de criação de grupos

3.9.2 – Adicionar entidades ao grupo

Com este comando, é possível adicionar entidades à grupos já existentes, sem que seja necessário recriar o grupo.

Command Line

<ADDGROUP ou AG> <Name, entities>

NAME - nome do grupo para o qual se quer adicionar entidades,
ENTITIES - números das entidades a serem adicionadas ao grupo.

Labels não são necessárias, mas a definição deve ser feita na ordem indicada.

Exemplos (os exemplos a seguir se baseiam nos grupos criados nos exemplos do item 3.9.1):

Ex1: ADDGROUP Name = "Teste123", EN = "4"

Ex2: ag "teste", "3"

Via Mouse

Este comando pode ser executado pela interface através do item Add to Group... do menu Groups. O diálogo de adição de entidades à grupos aparecerá (Figura 79), e seu funcionamento é similar ao descrito no item 3.9.1.

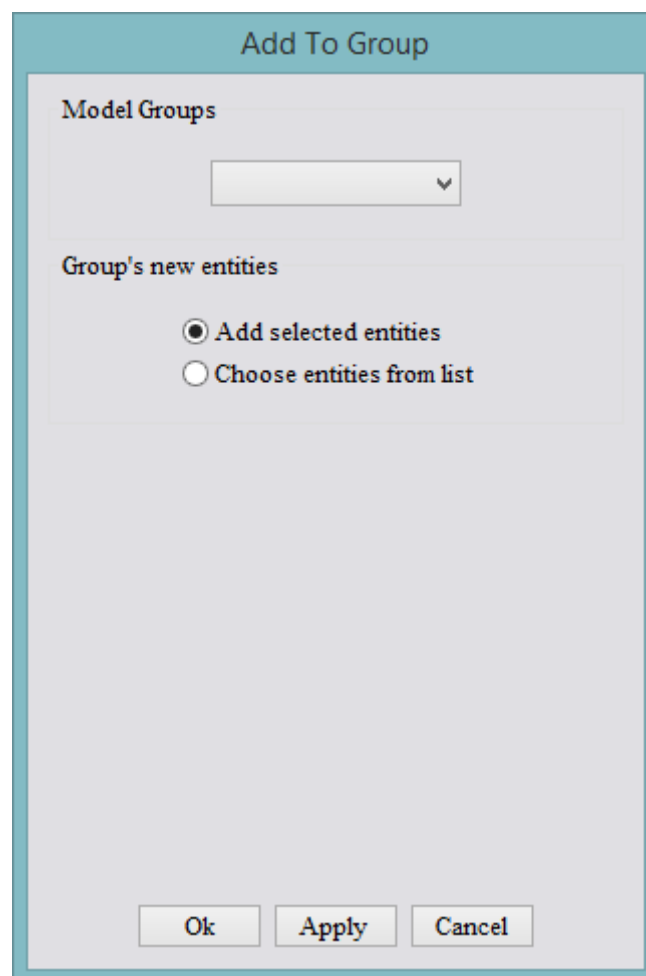


Figura 79 - Diálogo de adição de entidades à grupos

3.9.3 – Remover entidades de um grupo

Assim como é possível adicionar entidades de um grupo existente, pode-se também remover.

Command Line

<REMGROUP ou RG> <Name, entities>

NAME - nome do grupo do qual se quer remover entidades,

ENTITIES - números das entidades a serem removidas do grupo.

Labels não são necessárias, mas a definição deve ser feita na ordem indicada.

Exemplos: (os exemplos a seguir se baseiam nos grupos criados nos exemplos do item 3.9.1):

Ex1: REMGROUP name = "Teste123", EN = "3"

Ex2: rg "teste", "2"

Via Mouse

Este comando pode ser executado pela interface através do item Remove from Group... do menu Groups. O diálogo de remoção de entidades de grupos aparecerá (Figura 80), e seu funcionamento é similar ao descrito no item 3.9.1.

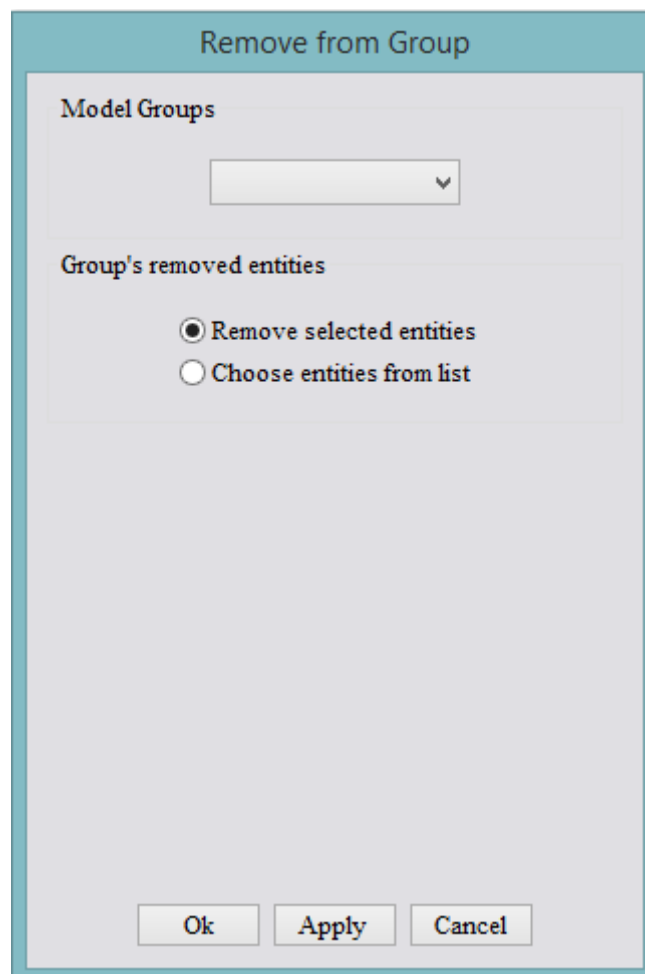


Figura 80 - Diálogo de remoção de entidades de grupos.

3.9.4 – Deletar grupos

Caso um grupo criado não seja mais necessário, ele pode ser deletado a qualquer momento através deste comando.

Command Line

<DELGROUP ou DG><Name>

NAME - nome do grupo a ser deletado.

Label do nome não é necessária.

Exemplos (os exemplos a seguir são baseados nos grupos criados no item 3.9.1):

Ex1: Delgroup Name = "Teste123"

Ex2: dg "teste"

Via Mouse

Este comando pode ser executado pela interface através do item Delete Group... do menu Groups. O diálogo de deleção de grupos aparecerá (Figura 81). Selecione o grupo que se deseja deletar e clique em Ok ou Apply.

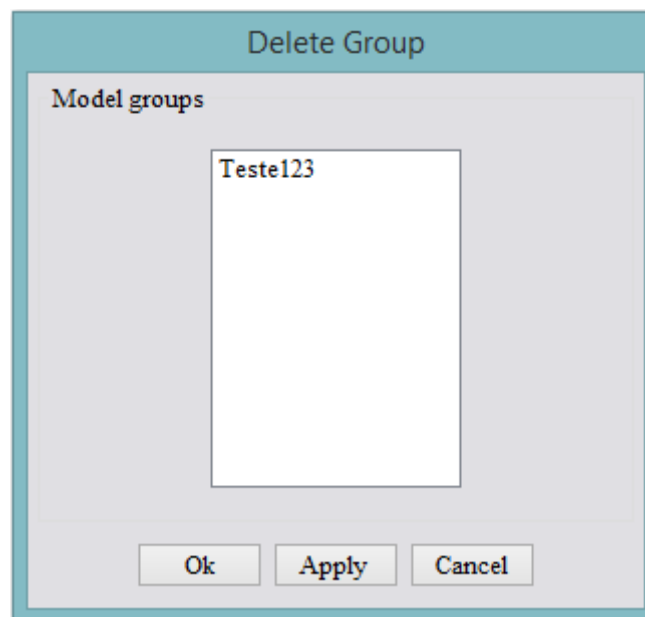


Figura 81 - Diálogo de deleção de grupos.

3.9.5 – Renomear grupos

Após criar um grupo, pode ser necessário renomeá-lo, o que pode ser feito através deste comando.

Command Line

<RENGROUP ou RENGR><OldName, NewName>

OLDNAME - nome atual do grupo a ser renomeado.

NEWNAME - nome que vai substituir o nome atual do grupo

Labels não são necessárias.

Exemplos (os exemplos a seguir são baseados nos grupos criados no item 3.9.1):

Ex1: `rengroup oldname = "Teste123", newname = "testenovo"`

Ex2: `reng "teste", "newteste"`

Via Mouse

Através da interface, este comando pode ser executado clicando-se no item Rename Group... do menu Groups. O diálogo de renomeação de grupos aparecerá (Figura 82), no qual se deve selecionar o grupo desejado e inserir seu novo nome.

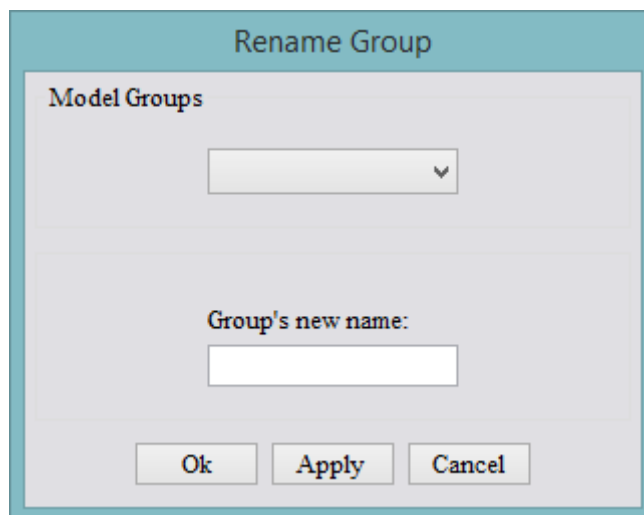


Figura 82 - Diálogo de renomeação de grupos.

3.9.6 – Informações sobre grupos

Informações como nome de um grupo, tipo de entidades que compõem e número de entidades que pertencem à um certo grupo podem ser obtidas através deste comando.

Command Line

<INQGR ou IG> <Name>

NAME - nome do grupo a ser inspecionado.

Label do nome não é necessária.

Exemplos (os exemplos a seguir são baseados nos grupos criados nos exemplos do item 3.9.1):

Ex1: ig name = "Teste123"

Ex2: inqgr "teste"

Via Mouse

Através da interface, este comando pode ser executado clicando-se no item Inquire Group... do menu Groups. O diálogo de informações dos grupos aparecerá (Figura 83), e, ao selecionar o grupo desejado, as informações aparecerão.

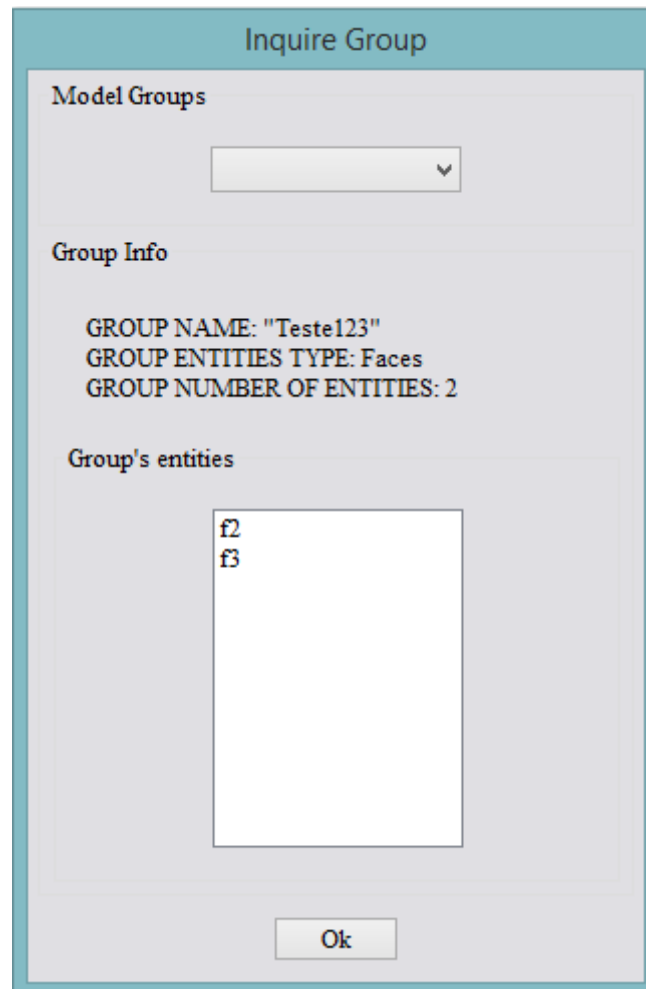


Figura 83 - Diálogo de informações sobre grupos.

3.9.7 – Selecionar/Deselecionar grupos

Para aplicar atributos ou realizar transformações nas entidades de um certo grupo, pode-se selecionar todas as entidades deste grupo através deste comando.

Via Mouse

Clique no item Select/Unselect Groups... do menu Groups. O diálogo de seleção de grupos aparecerá (Figura 84), com algumas opções de ações, inclusive deselegionar grupos atualmente selecionados.

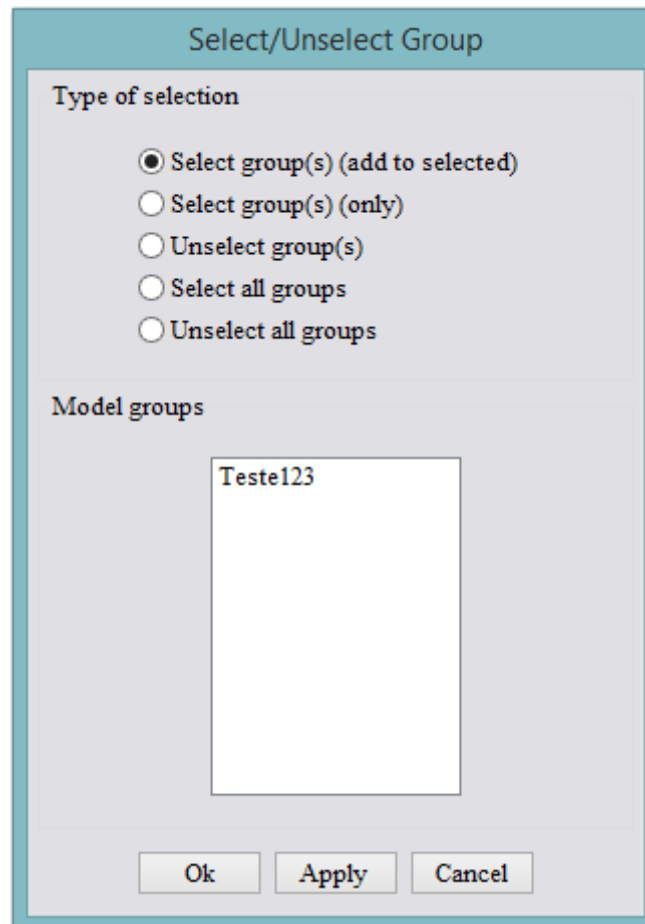


Figura 84 - Diálogo de seleção/deseleção de grupos.

3.9.8 – Mostrar/Ocultar grupos

Eventualmente pode ser interessante que certas entidades não sejam exibidas na tela. Para isso, pode-se utilizar este comando, que “esconde”, ou deixa de desenhar os grupos selecionados.

Command Line

`<HIDEGROUP><Name>` ou `<SHOWGROUP><Name>`

NAME - nome do grupo a ser utilizado,

Label não é necessárias.

Exemplos (os exemplos a seguir se baseiam nos grupos criados nos exemplos do item 3.9.1):

Ex1: `hidegroup "Teste123"`

Ex2: `hidegroup "teste"`

Para que as entidades apareçam novamente, basta substituir "hidegroup" por "showgroup"

3.10 – Comandos de início e fim de bloco para serem interpretados em linguagem LUA

Os comandos a seguir devem ser chamados toda vez que o usuário deseja executar um bloco de comandos programáveis em linguagem LUA. Dentro desse bloco, o usuário pode criar e atribuir valores para variáveis, criar funções, utilizar laços, condicionais e outras facilidades presentes na maioria das linguagens de programação. Os detalhes inerentes à sintaxe da linguagem LUA serão expostos no próximo capítulo.

3.10.1 – Início de bloco de comandos em LUA

<BEGINBLOCK ou BB>

3.10.2 – Fim de bloco de comandos em LUA

<ENDBLOCK ou EB>

Após a última linha de comandos em linguagem LUA, o usuário deve digitar a tecla ENTER antes de acionar esse comando de fim de bloco.

3.11 – Manipulação de atributos

Os *labels* (nomes) dos atributos, bem como quaisquer parâmetros vinculados a eles que devam ser expressos por meio de *strings* (cadeias de caracteres), devem sempre ser escritos entre aspas. Ressalva-se aqui o fato de que dois ou mais atributos não podem possuir o mesmo nome, mesmo pertencendo a classes diferentes de atributos.

Na manipulação dos atributos via mouse tem que se manipular os campos do frame *Task* do frame *Attribute Manager* (lado direito do canvas).

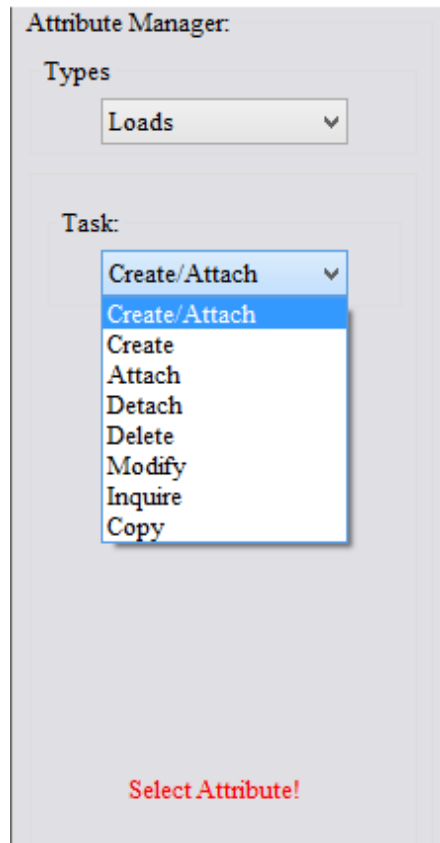


Figura 85 - Opções do frame *Task*.

A Figura 85 acima ilustra as opções do frame *Task*.

3.11.1 – Associação de um atributo às entidades selecionadas

Command Line

<ATTACH ou AT> <label>

label – nome do atributo.

Via Mouse

Para se ativar este comando através do uso do mouse, seleciona-se a opção *Attach* no campo do frame *Task* da Figura 85.

3.11.2 – Remoção de um atributo das entidades selecionadas

Command Line

<DETACH ou DT> <label>

label – nome do atributo.

Via Mouse

Para se ativar este comando através do uso do mouse, seleciona-se a opção *Detach* no campo do frame *Task* da Figura 85.

3.11.3 – Cópia de um atributo

Command Line

<COPY ou CP> <label, newlabel>

label – nome atual do atributo.

newlabel – novo nome do atributo.

Via Mouse

Para se ativar este comando através do uso do mouse, seleciona-se a opção *Copy* no campo do frame *Task* da Figura 85.

3.11.4 – Consulta aos parâmetros de um atributo

Command Line

<INQUIRE ou IN> <label>

label – nome do atributo.

Via Mouse

Para se ativar este comando através do uso do mouse, seleciona-se a opção *Inquire* no campo do frame *Task* da Figura 85.

3.11.5 – Deletar um atributo

Command Line

<DELETE ou DEL><label>

Label – nome do atributo

Via Mouse

Para deletar um atributo utilizando o mouse, basta selecionar a opção *Delete* no campo do frame *Task* da Figura 85.

3.12 – Criação e modificação de atributos

A criação e a modificação de atributos são realizadas de forma semelhante, usando-se o mesmo LABEL para atributos do mesmo tipo. Esse LABEL indica o tipo de atributo que será criado ou modificado e, normalmente, como será visto a seguir, é representado por uma palavra muito semelhante ou mesmo igual ao nome da classe de atributos correspondente.

A diferença entre a criação e a modificação é que, na criação, todos os parâmetros referentes a cada atributo devem ter seus valores definidos, ou então, assumirão valores *default* (valores padrão) caso não sejam explicitamente definidos. Os valores *default* dependem, obviamente, do tipo de atributo considerado e serão discriminados caso a caso. Na modificação, apenas os parâmetros que se deseja modificar precisam ter seus novos valores definidos. Os parâmetros que não forem explicitamente modificados, continuarão com seus valores anteriores. Por esse mesmo motivo, na modificação é estritamente necessário indicar os *labels* dos parâmetros que se desejam modificar.

Se o usuário deseja que o *label* (nome) de um atributo seja modificado, em qualquer caso, basta usar a sintaxe:

<LABEL ou LB = > <newlabel>
onde *newlabel* representa o novo nome dado ao atributo.

Deve-se atentar para o fato de que a modificação de um ou mais parâmetros pode ocasionar, necessariamente, a modificação de outros parâmetros. Pode-se citar como exemplo o caso de atributos que possuam como parâmetro pontos no plano (par ordenado (x, y)). Além dos pontos em si, o número de pontos também é um parâmetro desses tipos de atributos. Logo, se um atributo desse tipo possui n pontos, e se o usuário deseja modificar o número de pontos para $n+p$, sendo $p \geq 1$, os pontos $(x_{n+1}, y_{n+1}), \dots, (x_{n+p}, y_{n+p})$ devem ser definidos.

Na criação de atributos, a ordem dos parâmetros pode ser importante ou não. Em muitos casos, contudo, os parâmetros podem ser definidos em qualquer ordem, desde que o *label* do parâmetro em questão seja explicitado. Isto será analisado detalhadamente para cada tipo de atributo.

Após o LABEL principal do comando, que, como dito acima, indica o tipo de atributo a ser tratado, em qualquer caso deve-se digitar o *label* do atributo que está sendo criado ou modificado, ou seja, o seu nome. A seguir vêm os parâmetros que serão definidos ou modificados.

Estas funções estão presentes no frame *Types* do frame *Attribute Manager* (lado direito do canvas).

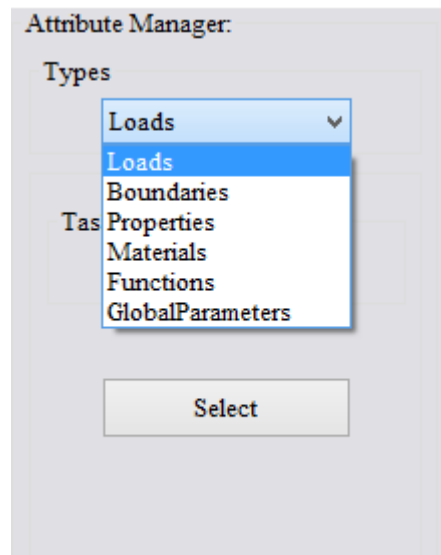


Figura 86 - Opções do frame *Types*.

A Figura 86 acima ilustra as opções do frame *Types*.

3.12.1– Funções

3.12.1.1 – Função de tempo

Command Line

$\langle \text{TIMEFUNC ou TF} \rangle \langle \text{label} \rangle (, np, inc1, val1, \dots, incn, valn)$

label – nome do atributo

np – número de pontos

inc1 – abscissa (incremento) do primeiro ponto

val1 – ordenada (valor) do primeiro ponto

.

.

.

incn – abscissa (incremento) do último (n-ésimo) ponto

valn – ordenada (valor) do último (n-ésimo) ponto

Via Mouse

Seleciona-se a opção *Functions* no campo *Types* da Figura 86, posteriormente serão disponibilizados dois botões: *Time* e *Spatial*.

Clica-se no botão *Time* que estará abaixo do frame *Task*. Após se selecionar esta opção será aberta uma janela (Figura 87).

Os parâmetros não precisam ter seus *labels* especificados, desde que estejam na ordem exposta acima e que nenhum parâmetro seja omitido.

Ex: tf “tf_1”, 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0

Se o usuário desejar atribuir valores aos parâmetros fora da ordem descrita acima, necessariamente os *labels* de todos os parâmetros devem ser especificados. Os nomes aceitos para o *label* de cada parâmetro são:

Número de pontos – NP

Primeiro incremento (abscissa) – P1X

Primeiro valor (ordenada) – P1Y

.

.

.

N-ésimo incremento (abscissa) – PNX

N-ésimo valor (ordenada) - PNY

Ex: tf “tf_1”, p2x = 4.5, p1y = -0.5, p1x = 8.56, p2y = -5.6, np = 2

Se desejar, o usuário pode criar uma função de tempo com os valores *default* para os parâmetros desse tipo de atributo. São eles (obviamente, exceto o *label*):

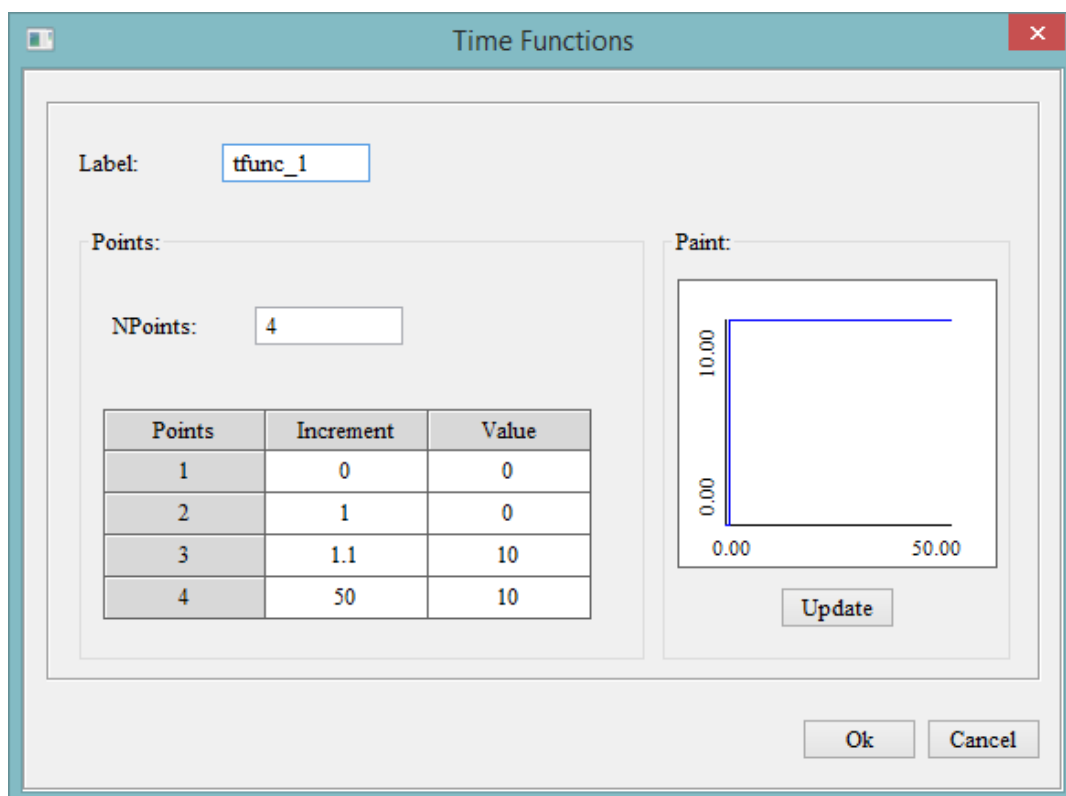


Figura 87 - Valores *default* dos parâmetros de uma função *Time*.

Ex: tf “tf_1”

3.12.1.2 – Função de espaço

Command Line

<SPATFUNC ou SF> <label> <function>

label – nome do atributo.

function – função ($f(x, y)$).

Via Mouse

Clica-se no botão *Spatial* que estará abaixo do frame *Task*. Após se selecionar esta opção será aberta uma janela (Figura 88).

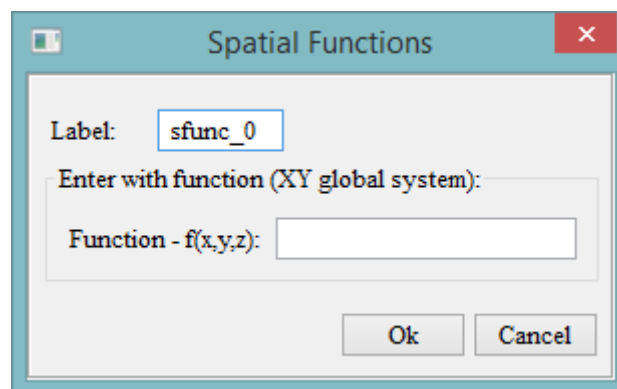


Figura 88 - Janela de ajuste da função *Spatial*.

A Figura 88 acima ilustra a janela aberta após se clicar no botão *Spatial*.

A função $f(x, y)$ pode ser um polinômio de qualquer grau nas variáveis x e y , ou ainda uma função trigonométrica, exponencial ou logarítmica nessas variáveis. Uma combinação de todos esses tipos de função também é válida. A função deve ser escrita entre aspas.

O *label* do parâmetro que se refere à função não precisa ser especificado, contudo, se o for, os nomes aceitos são: F ou FUNCTION.

Ex: sf “sf_1”, “ $\log(x^2 - 3 * y) + \sin(\cos(-8.3 * 1/x))$ ”

3.12.2 – Materiais

3.12.2.1 – Materiais de sólidos

Command Line

<MATERIAL ou MAT> <label> (constitutive, const_params, common_fields)

constitutive – modelo constitutivo.

Valores possíveis para o *label* desse parâmetro:
CONST

Valores possíveis para o parâmetro:
“ELASTIC” ou “EL”
“ELASTO-PLASTIC” ou “ELP”
“VISCO-ELASTO” ou “VEL”

Se esse parâmetro for especificado, seu *label* não precisa estar presente. Além disso, esse parâmetro deve ser sempre o primeiro da lista de parâmetros, caso seja especificado. Se ele não for especificado, assume-se o seu valor *default*, que é “ELASTIC”.

Após a definição do modelo constitutivo, os parâmetros inerentes a cada tipo de modelo devem ser especificados, necessariamente (*const_params*).

No caso de um modelo elástico (“ELASTIC”), não há nenhum parâmetro específico desse tipo de modelo que deva ser especificado. Contudo, nos outros dois casos, existem parâmetros que devem ser definidos (ou, dependendo do caso, se não forem definidos assumem seus valores *default*).

- Modelo Elasto-Plástico (“ELASTO-PLASTIC”):

Os parâmetros descritos a seguir podem ser definidos em qualquer ordem.

1) *envelope* – envelope.

Valores possíveis para o *label* desse parâmetro:
ENVELOPE ou ENV – o *label* desse parâmetro é sempre necessário.

Valores possíveis para o parâmetro:
“ON”
“OFF”

Se este parâmetro não for definido, assumirá seu valor *default*, que é “ON”.

2) *yield criteria* – critério de escoamento

Valores possíveis para o *label* desse parâmetro:
YIELD ou Y – o *label* desse parâmetro não é necessário

Valores possíveis para o parâmetro:
“MOHR COULOMB DRAINED” ou “MCD”
“MOHR COULOMB UNDRAINED” ou “MCU”
“MOHR COULOMB MOD.” ou “MCM”
“VON MISES IDEAL” ou “VMI”
“VON MISES ISOTROPIC” ou “VMISO”

Se o critério de escoamento não for especificado, assume-se o seu valor *default*, que é “MOHR COULOMB DRAINED”. Os valores *default* dos parâmetros inerentes a este tipo de critério são mostrados abaixo.

Se o critério de escoamento for especificado, os parâmetros relativos ao tipo de critério escolhido também devem ser definidos, imediatamente a seguir.

Mohr Coulomb Drained e Mohr Coulomb Mod. :

Valores possíveis para o *label* do parâmetro:

C – o *label* desse parâmetro é sempre necessário.

Valores possíveis para o *label* do parâmetro:

Fi – o *label* desse parâmetro é sempre necessário.

No caso do Mohr Coulomb Drained e Mohr Coulomb Mod, qualquer um desses parâmetros pode ser omitido, assumindo seu valor *default*.

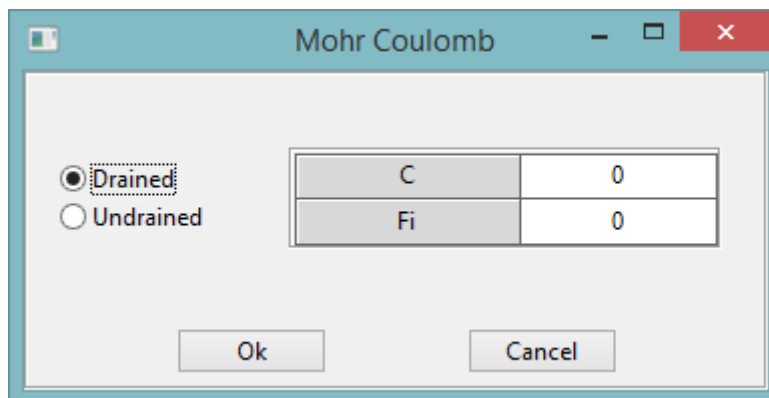


Figura 89 - Valores *default* para um material com critério de escoamento do tipo *Mohr Coulomb Drained*.

Mohr Coulomb Undrained :

Valores possíveis para o *label* do parâmetro:

E/SU – o *label* desse parâmetro é sempre necessário.

Valores possíveis para o *label* do parâmetro:

SU0– o *label* desse parâmetro é sempre necessário.

Valores possíveis para o *label* do parâmetro:

SU/Z– o *label* desse parâmetro é sempre necessário

Qualquer um desses parâmetros pode ser omitido, assumindo seu valor *default*:

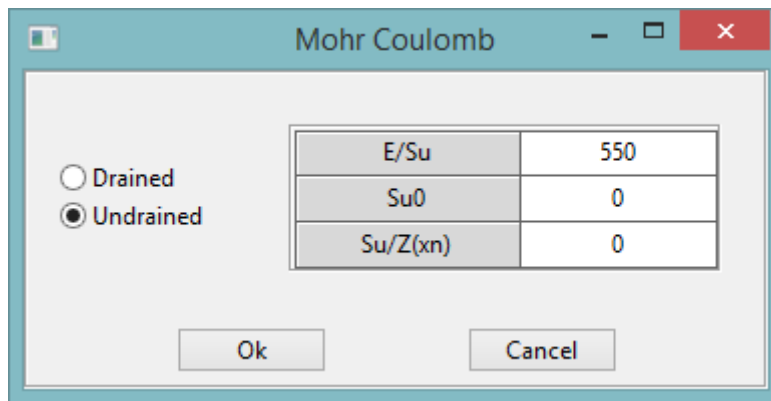


Figura 90 - Valores *default* para um material com critério de escoamento do tipo *Mohr Coulomb Undrained*.

Von Mises Ideal (Elastic-Plastic Perfect):

Valores possíveis para o *label* do parâmetro:

STRESS ou S – o *label* desse parâmetro é sempre necessário

Esse parâmetro pode ser omitido, assumindo seu valor *default*:

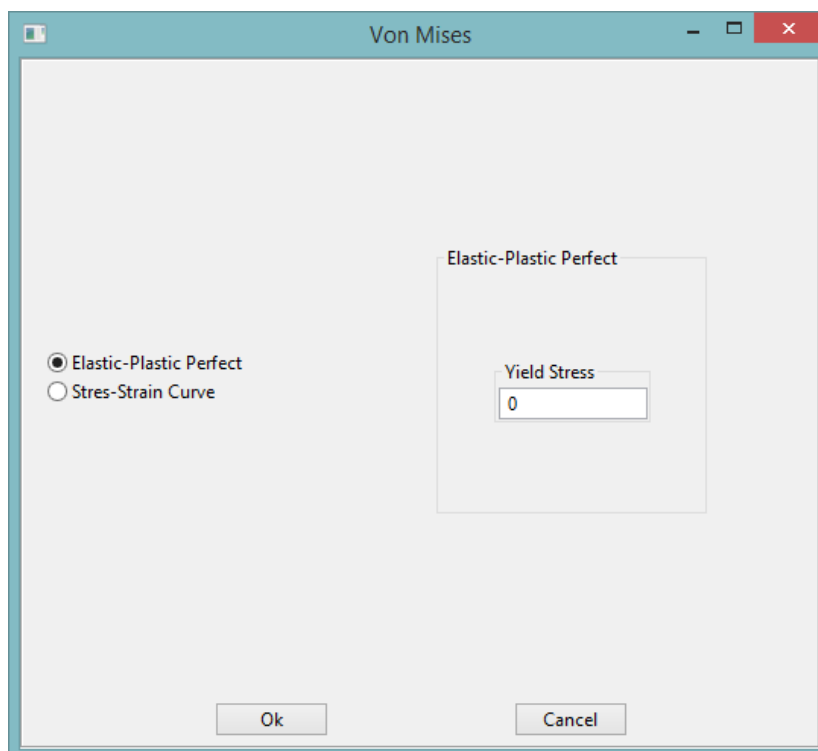


Figura 91 - Valor *default* para um material com critério de escoamento do tipo *Von Mises Ideal (Elastic-Plastic Perfect)*.

Von Mises Isotropic (Stress-Strain Curve):

Valores possíveis para o *label* do parâmetro:

NP – o *label* desse parâmetro é sempre necessário

Pontos:

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

PIX

PIY

.

.

.

PNX

PNY

Os pontos devem ser definidos sempre nessa ordem (abscissa do primeiro ponto, ordenada do primeiro ponto, abscissa do segundo ponto, ordenada do segundo ponto, etc...).

- Modelo Visco-Elástico (“VISCO-ELASTO”)

Os parâmetros desse tipo de modelo que são comuns ao Modelo Elasto-Plástico podem ser definidos da mesma forma como descrito acima. Além desses parâmetros, um Modelo Visco-Elástico também comporta um parâmetro que define a sua fluência. Esse parâmetro pode ser definido antes ou depois dos parâmetros descritos acima, contudo necessariamente deve ser definido.

Valores possíveis para o *label* do parâmetro:

CREEP – o *label* desse parâmetro não é necessário

Valores possíveis para o parâmetro:

“POTENTIAL” ou “POT”

“ADDITIVE” ou “ADD”

“SELECTIVE” ou “SEL”

Os parâmetros relativos a cada tipo de fluência também devem ser definidos, imediatamente a seguir.

Potencial :

Valores possíveis para os *labels* dos parâmetros (todos necessários):

AA

ALPHA ou A

BETA ou B

GAMA ou G

APOWER ou AP

ACONST ou AC

FRAC ou F

Todos esses parâmetros devem ser definidos.

Aditivo e Seletivo :

Valores possíveis para os *labels* dos parâmetros (todos necessários):

SIGCRT ou SIG

VDEF CRT ou VDEF

XN1CRT ou XN1
 XN2CRT ou XN2
 FRAC ou F

Todos esses parâmetros devem ser definidos.

common_fields – campos comuns a qualquer tipo de material. São eles: E, Ni, Mass Density, Thrust, Thermal, Poropressure e Color. A única exceção são os materiais com critério de escoamento do tipo Mohr Coulomb Undrained, que não possuem E e Ni definidos (se o usuário definir valores para esses parâmetros nesse caso, o comando não será validado). Nenhum desses parâmetros é necessário. Todos possuem valores *default* que serão atribuídos se o usuário não discriminar valores para esses parâmetros.

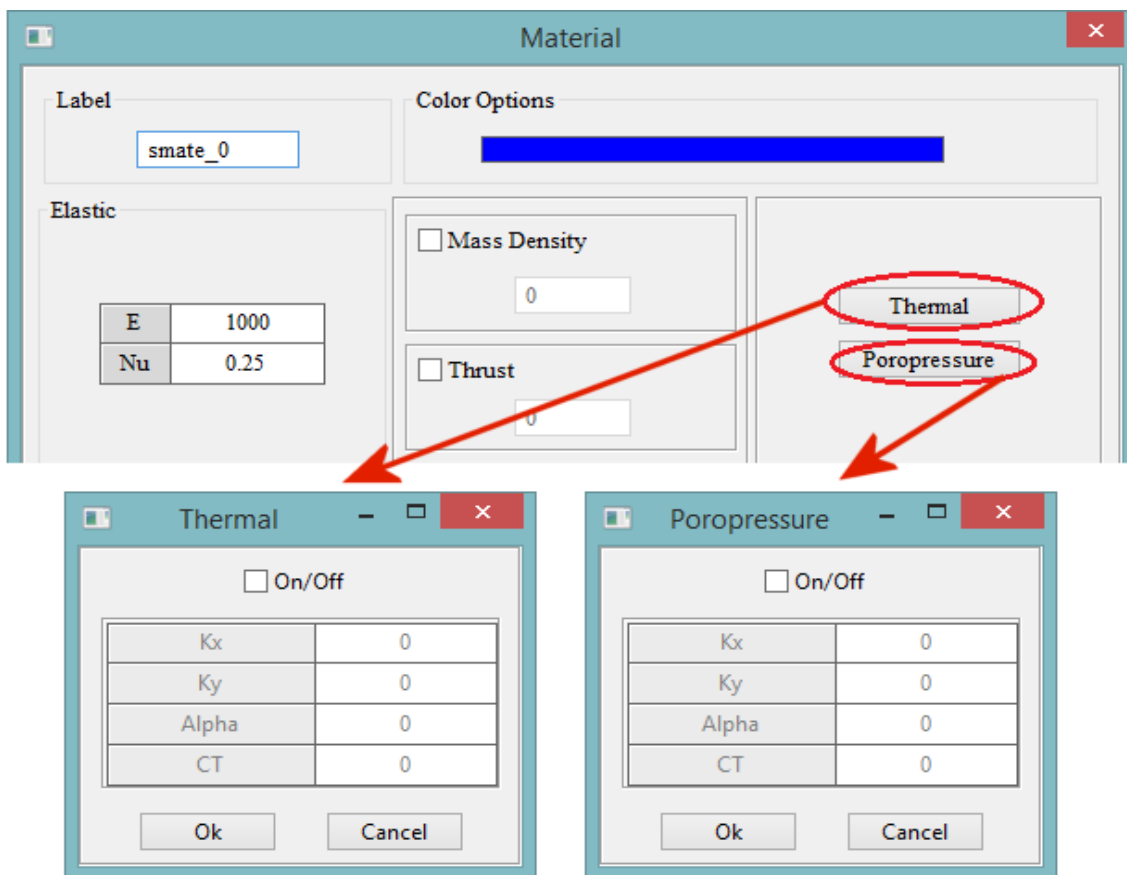


Figura 92 - Valores *default* para os campos comuns a todos os tipos de material (exceto o label)

Os parâmetros a seguir podem ser definidos em qualquer ordem:

Valores possíveis para os *labels* dos parâmetros (todos necessários):

E
 NI
 MASS DENSITY ou MD
 THRUST ou THR
 THERMAL ou THE
 POROPRESSURE ou PP
 COLOR ou COL

Valores possíveis para os parâmetros:

E – número real maior que zero

NI – número real maior ou igual a zero

MASS DENSITY e THRUST – se o usuário deseja que a opção de definir algum desses parâmetros esteja ativa, basta atribuir um valor real maior ou igual a zero ao parâmetro desejado. Se deseja que essa opção esteja inativa, deve atribuir o valor “OFF” para esse parâmetro.

THERMAL – se o usuário deseja que a opção este parâmetro esteja ativo, deve atribuir o valor “ON” para ele. Se deseja que esteja inativo, deve atribuir o valor “OFF”.

Se a opção THERMAL estiver habilitada, o usuário pode definir valores reais para os parâmetros inerentes a essa opção.

Valores possíveis para os *labels* dos parâmetros (todos necessários):

KXT

KYT

ALFT

CT

Contudo, a atribuição de valores para esses parâmetros não é necessária. Se eles não forem definidos, assumirão seus valores *default*.

POROPRESSURE – se o usuário deseja que este parâmetro esteja ativo, deve atribuir o valor “ON” para ele. Se deseja que esteja inativo, deve atribuir o valor “OFF”.

Se a opção POROPRESSURE estiver habilitada, o usuário pode definir valores reais para os parâmetros inerentes a essa opção.

Valores possíveis para os *labels* dos parâmetros (todos necessários):

KXP

KYP

ALFP

CP

Contudo, a atribuição de valores para esses parâmetros não é necessária. Se eles não forem definidos, assumirão seus valores *default*.

COLOR – para definição de cor do material via linha de comandos, existem apenas 23 opções de cores, que são combinações básicas do modelo RGB. A cor *default* é branca. Uma lista é disponibilizada a seguir:

“WHITE” ou “W”

“GRAY” ou “GY”

“RED” ou “R”

“SOFTBLUE” ou “SB”

“BLUE” ou “B”

"DARKBLUE" ou "DB"
"DARKGREEN" ou "DG"
"GREEN" ou "G"
"SOFTGREEN" ou "SG"
"YELLOW" ou "Y"
"SOFTORANGE" ou "SO"
"ORANGE" ou "O"
"DARKORANGE" ou "DO"
"SOFTPURPLE" ou "SP"
"PURPLE" ou "P"
"DARKPURPLE" ou "DP"
"PURPLEBROWN" ou "PB"
"DARKBROWN" ou "DB"
"BROWN" ou "B"
"SOFTBROWN" ou "SB"
"YELLOWBROWN" ou "YB"
"GREENBROWN" ou "GB"

Para mais possibilidades de cores, é preciso utilizar a interface. O diálogo da figura abaixo ilustra o modo de seleção da cor desejada:

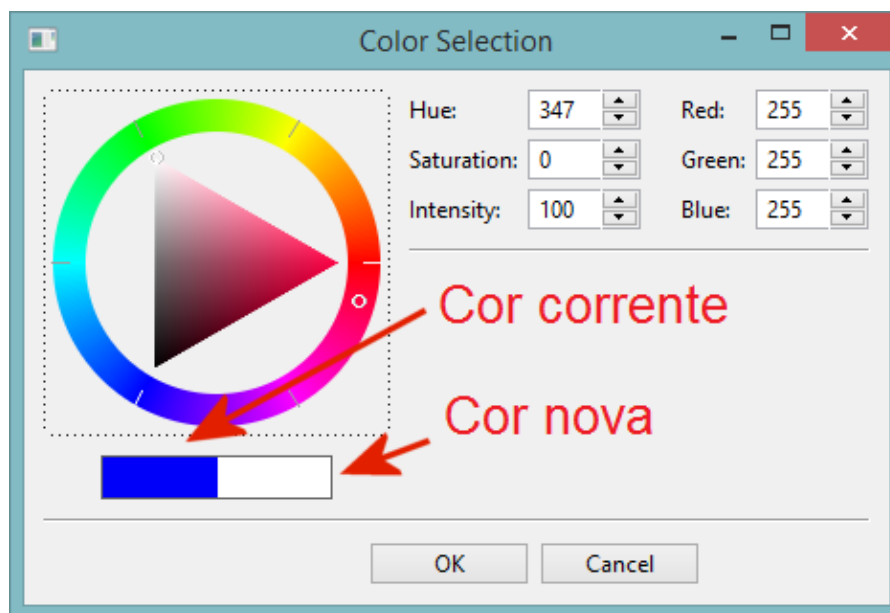


Figura 93 - Diálogo de seleção de cores

Este diálogo é aberto clicando-se na barra de cor existente no diálogo de criação de material, como pode ser visto na Figura 92.

Se apenas o *label* do material for especificado, será criado um material com todos os valores *default* para seus parâmetros.

Via Mouse

Selecione-se a opção *Materials* no campo *Types* da Figura 86, posteriormente serão disponibilizados dois botões: *Solid* e *Interface*. É importante ressaltar que este comando só é válido para entidades do tipo *Region*.

Clica-se no botão *Solid* que estará abaixo do frame *Task* e uma janela será aberta (Figura 94).

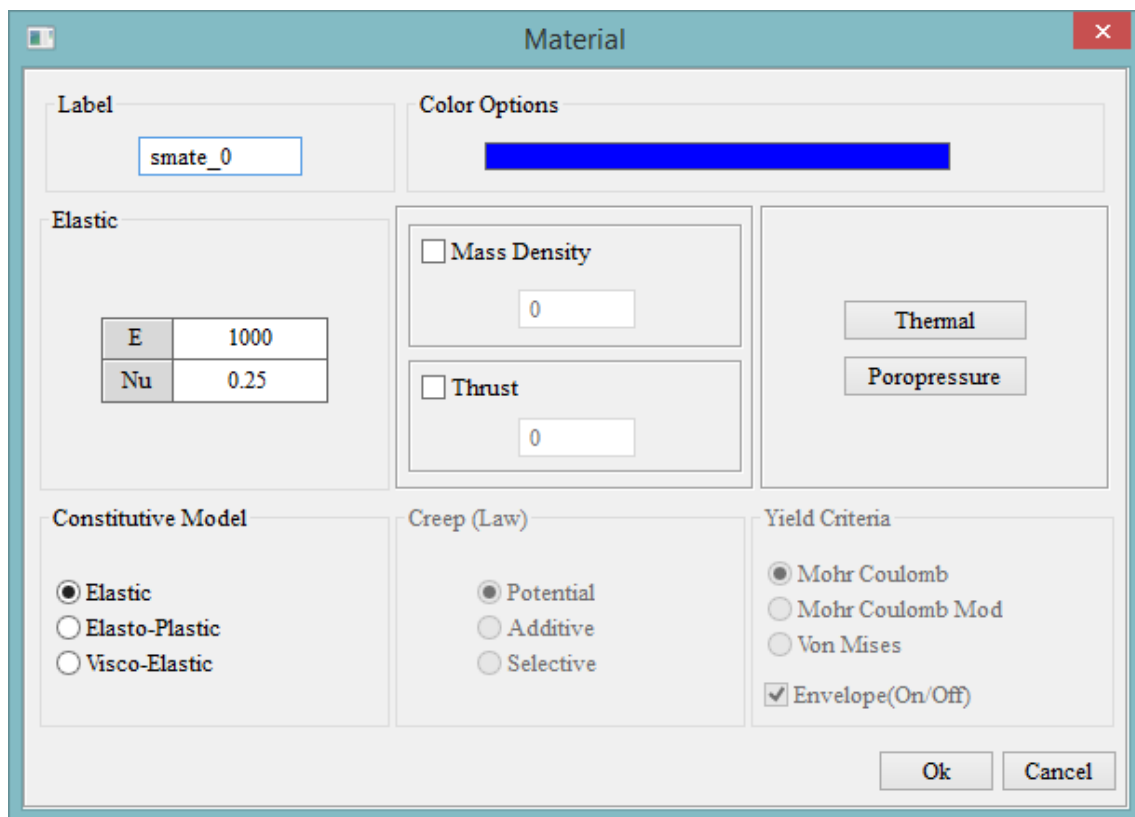


Figura 94 - Valores *default* para os parâmetros de um material

Exemplos:

Ex1: mat "m1"

Ex2: material "m2", e = 2300, ni = 0.45, col = "blue"

Ex3: mat "m3", "elp", "mcd", c = 12, fi = 1, env = "off", md = 12.3

Ex4: MAT "m4", "vel", "add", sig = 1, vdef = 2, xn1 = 3, xn2 = 4, f = 5

Ex5: mat "m5", "elp", the = "on", kyt = 12, pp = "on", alfp = 23

3.12.2.2 – Materiais de interface

Command Line

<INTERFACEMATERIAL ou INTERFMAT ou IMAT> <label> (, *params*)

label – nome do atributo

params – parâmetros inerentes a um material de interface.

Valores possíveis para o *label* do parâmetro (necessário):

- THRUST ou THR

Valores possíveis para o parâmetro:

- “ON” ou “OFF”

Valores possíveis para o *label* do parâmetro:

- XKO

Este parâmetro pode ser definido logo após o *label* do atributo, sem a necessidade de se discriminar o seu *label*, bastando escrever o seu valor real. Isso fará com que, automaticamente, o valor do parâmetro THRUST seja definido como “ON”. Contudo, se este parâmetro não for o primeiro a ser definido, o seu *label* passa a ser necessário.

Valores possíveis para o *label* do parâmetro (pode ou não ser necessário):

- NCONST

Valores possíveis para o parâmetro:

- “HIPERBOLIC” ou “HIP” ou “H”
- “EXPONENTIAL” ou “EXP” ou “E”
- “LINEAR” ou “LIN” ou “L”
- “PERFECT PLASTIC” ou “PERFPLAST” ou “PP”
- “MULTILINEAR” ou “MULTILIN” ou “ML”

Apenas no caso de uma Lei Constitutiva do tipo “PERFECT PLASTIC” o *label* do parâmetro (NCONST) se torna necessário. Isso porque a Lei Constitutiva da Tensão Tangencial também pode ser do mesmo tipo.

Cada uma dessas Leis Constitutivas possui alguns parâmetros inerentes, que podem precisar ou não ser definidos. Se não forem, assumirão seus valores *default*.

Hiperbólica :

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- DEFMC ou DEF
- ALPHA ou A
- XA

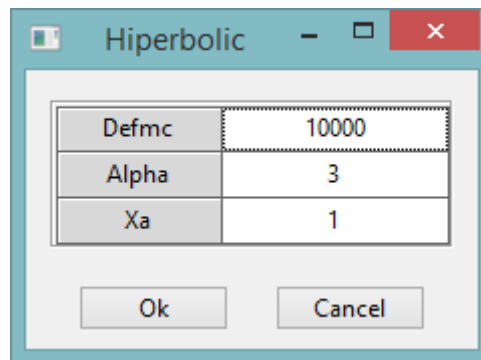


Figura 95 - Valores *default* para os parâmetros de um material de interface com Lei Constitutiva Hiperbólica para a Tensão Normal

Exponencial :

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- DEFMC ou DEF
- ALPHA ou A
- SEXP ou S

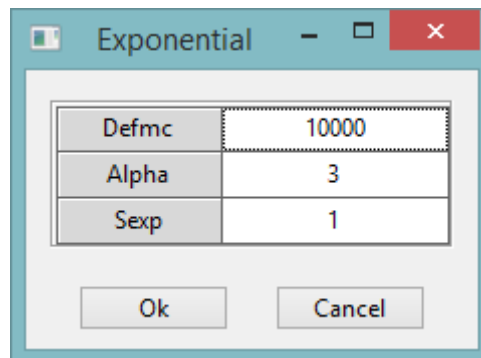


Figura 96 - Valores *default* para os parâmetros de um material de interface com Lei Constitutiva Exponencial para a Tensão Normal

Linear :

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- KN
- DEFMC ou DEF

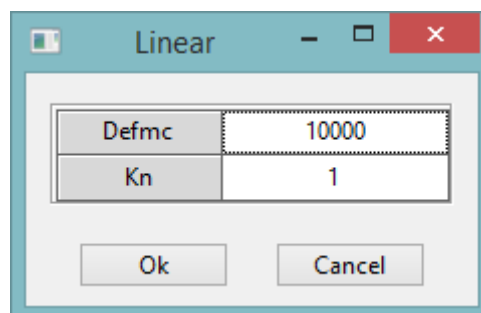


Figura 97 - Valores *default* para os parâmetros de um material de interface com Lei Constitutiva Linear para a Tensão Normal

Perfeitamente Plástica :

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- DEFMC ou DEF
- KN
- SIGMAX ou SX

Valores possíveis para o *label* do parâmetro (sempre necessário):

- CUTOFF ou CUT

Valores possíveis para o parâmetro:

- “ON” ou “OFF”

Valores possíveis para o *label* do parâmetro (sempre necessário):

- CUTTYPE ou CUTTP

Valores possíveis para o parâmetro:

- “TENSION” ou “T”
- “GEOSTATIC” ou “G”

Valores possíveis para o *label* do parâmetro (sempre necessário):

- TENSION ou T

Este parâmetro só pode ser definido se o CUTOFF estiver habilitado (“ON”) e se o CUTTYPE estiver definido como “TENSION”.

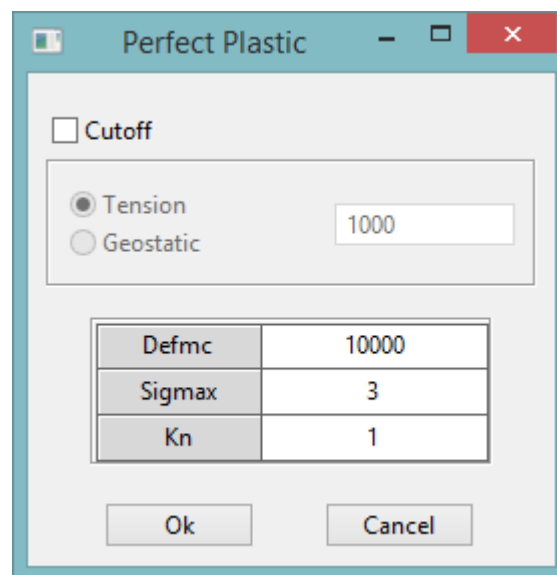


Figura 98 - Valores *default* para os parâmetros de um material de interface com Lei Constitutiva Perfeitamente Plástica para a Tensão Normal

Multilinear :

Valores possíveis para o *label* do parâmetro (sempre necessário):

- CUTOFF ou CUT

Valores possíveis para o parâmetro:

- “ON” ou “OFF”

Valores possíveis para o *label* do parâmetro (sempre necessário):

TENSION ou T

Este parâmetro só pode ser definido se o CUTOFF estiver habilitado (“ON”).

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- MNP – número de pontos (maior que zero)
- DSN1 – abscissa do primeiro ponto
- TSN1 – ordenada do primeiro ponto
- .
- .
- .
- DSNn – abscissa do último (n-ésimo) ponto
- TSNn – ordenada do último (n-ésimo) ponto

Se a Lei Constitutiva da Tensão Normal for Multilinear, os parâmetros relativos ao número de pontos e aos valores das abscissas e ordenadas dos pontos são estritamente necessários. O parâmetro CUTOFF, se não for definido, assume o valor *default* “OFF”.

Valores possíveis para o *label* do parâmetro (pode ou não ser necessário):

- TCONST

Valores possíveis para o parâmetro:

- “KINEMATING HARDENING” ou “KINEHARD” ou “KH”
- “PERFECT PLASTIC” ou “PERFPLAST” ou “PP”

Apenas no caso de uma Lei Constitutiva do tipo “PERFECT PLASTIC” o *label* do parâmetro (TCONST) se torna necessário. Isso porque a Lei Constitutiva da Tensão Normal também pode ser do mesmo tipo.

Cada uma dessas Leis Constitutivas possui alguns parâmetros inerentes, que podem precisar ou não ser definidos. Se não forem, assumirão seus valores *default*.

Perfeitamente Plástica :

Valores possíveis para o *label* do parâmetro (sempre necessário):

- YIELD ou Y

Valores possíveis para o parâmetro:

- “MOHR COULOMB” ou “MC”
- “JAEGER” ou “J”

Valores possíveis para o *label* do parâmetro (sempre necessário):

- UPDATE ou UPD – Atualizar a Tensão Tangencial Máxima

Valores possíveis para o parâmetro:

- “ON”
- “OFF”

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- KT
- C ou CRES

- FI ou FIRES

Valores possíveis para o *label* do parâmetro (sempre necessário):

- RESIDUAL ou RES

Valores possíveis para o parâmetro:

- “ON”
- “OFF”

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- BO
- XM

Esses parâmetros só poderão ser definidos se o parâmetro RESIDUAL estiver habilitado (“ON”). Se o parâmetro RESIDUAL estiver habilitado e esses parâmetros não forem definidos, assumirão seus valores *default* (Fig.16).

Se um ou mais dos parâmetros relativos a essa Lei Constitutiva forem omitidos, assumirão seus valores *default*:

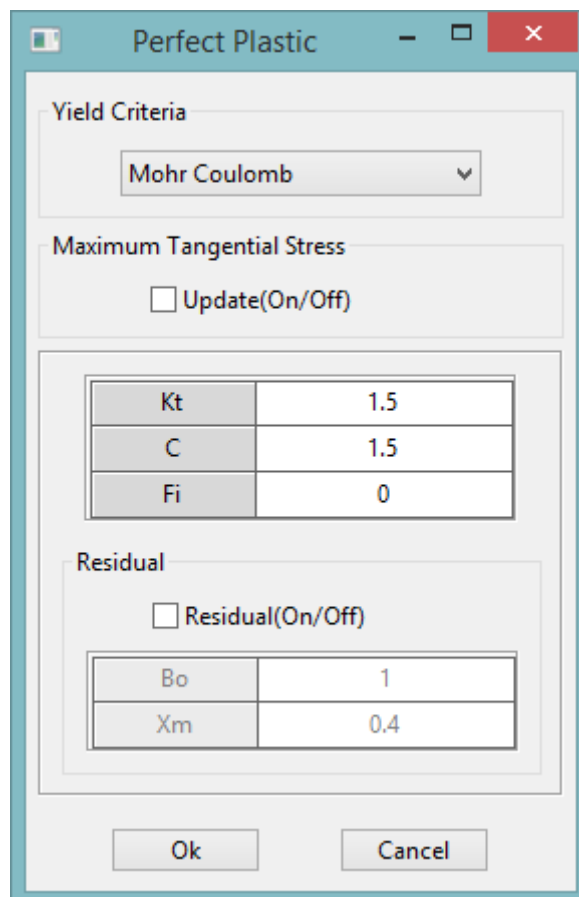


Figura 99 - Valores default para os parâmetros de um material de interface com Lei Constitutiva Perfeitamente Plástica para a Tensão Tangencial

Kinemating Hardening :

Valores possíveis para os *labels* dos parâmetros (sempre necessários):

- KNP – número de pontos (maior que zero)
- DST1 – abscissa do primeiro ponto
- TST1 – ordenada do primeiro ponto
- .
- .
- .
- DSTn – abscissa do último (n-ésimo) ponto
- TSTn – ordenada do último (n-ésimo) ponto

Se a Lei Constitutiva da Tensão Tangencial for do tipo Kinemating Hardening, os parâmetros relativos ao número de pontos e aos valores das abscissas e ordenadas dos pontos são estritamente necessários.

Se nenhum parâmetro do material de interface for definido após o seu *label*, será criado um material de interface com todos os valores *default* para os seus parâmetros.

Via Mouse

Clica-se no botão *Interface* que estará abaixo do frame *Task*, após a seleção da opção *Materials* no campo *Types*, e será aberta uma janela (Figura 100).

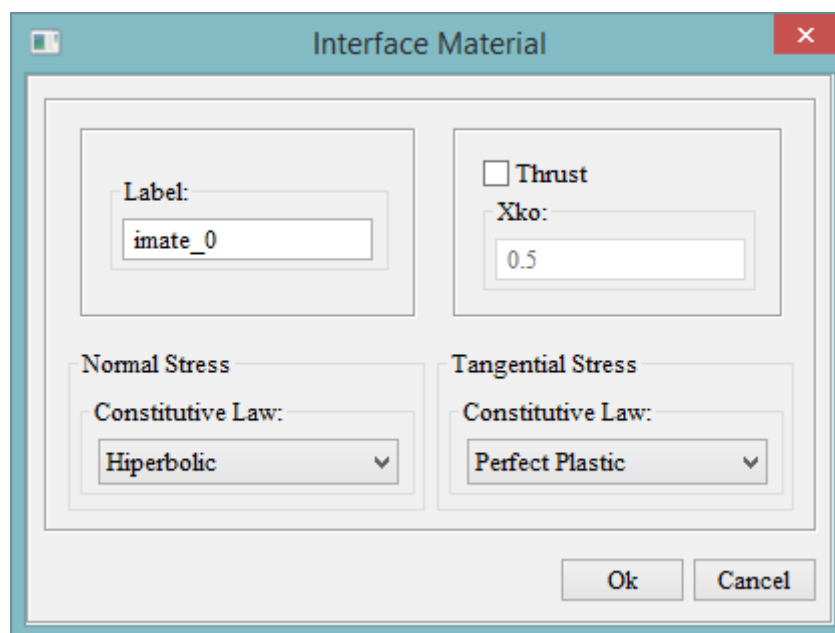


Figura 100 - Material de interface *default*.

Exemplos:

Ex1: imat "im1"

Ex2: interfmat "im2", 32

Ex3: IMAT "im3", thr = "on", "kh", knp = 2, dst1 = 0.0, tst1 = 0.0, dst2 = 3.4, tst2 = -4.5

Ex4: interfacematerial "im4", 12, nconst = "pp", cut = "on", cuttp = "g", def = 13000

Ex5: imat "im5", "l", tconst = "pp", y = "j", c = 1.4, res = "on"

3.12.3 – Propriedades

3.12.3.1 – Propriedades de sólidos

Command Line

`<PROPERTIES ou PR> <label> (, thickness) <, material> (, integration, model)`

Se os parâmetros desse tipo de atributo forem todos definidos na ordem acima (sem omissão de nenhum deles), os seus *labels* são desnecessários. Caso o usuário deseje omitir um ou mais parâmetros (que assumirão seus valores *default*), ou deseje trocar a ordem dos parâmetros, os *labels* dos mesmos passam a ser necessários. O único parâmetro que não pode ser omitido é o material associado ao atributo.

thickness – espessura.

Valores possíveis para o *label* do parâmetro:

THICKNESS ou THICK ou T

material – material associado à propriedade. Esse parâmetro deve ser definido passando-se a *string* (cadeia de caracteres) que define o nome do atributo do tipo “Material” que se deseja associar. Esse material já deve ter sido criado anteriormente.

Valores possíveis para o *label* do parâmetro:

MATERIAL ou MAT ou M

integration – tipo de integração

Valores possíveis para o *label* do parâmetro:

INTEGRATION ou INT ou I

Valores possíveis para o parâmetro:

“REDUCED” ou “RED” ou “R”

“COMPLETE” ou “COMP” ou “C”

model – tipo de modelo.

Valores possíveis para o *label* do parâmetro:

“AXISSIMETRIC” ou “AX”

“PLANE STRESS” ou “PSE”

“PLANE STRAIN” ou “PSA”

Se o usuário deseja passar como parâmetro apenas o material associado ao atributo, os outros parâmetros assumirão seus valores *default* (exceto o *label*).

Exemplos:

Ex1: `pr "p1", 1.2, "mat1", "c", "psa"`

Ex2: `properties "p2", mat = "mat2"`

Ex3: `PR "p3", model = "ax", mat = "im3", t = 2.3`

Via Mouse

Selecione-se a opção *Properties* no campo *Types*, posteriormente será disponibilizado o botão *Select*, clicando-se neste botão se terá duas opções: *Solid* e *Interface*, de acordo com a Figura 101.

Para se ativar as propriedades dos sólidos tem que se selecionar uma entidade do tipo *Region* e clicar no botão *Select* da Figura 101, selecionando a opção *Continuous* o que abrirá uma janela (Figura 102).

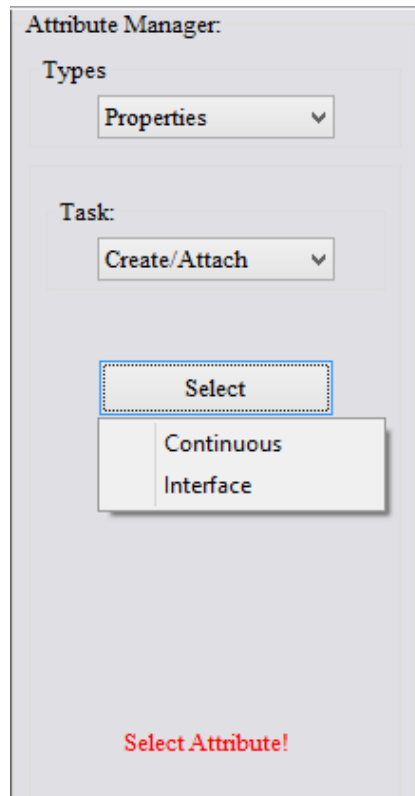


Figura 101 - Clique do botão *Select*.

A Figura 101 acima ilustra as opções disponibilizadas após se clicar no botão *Select*.

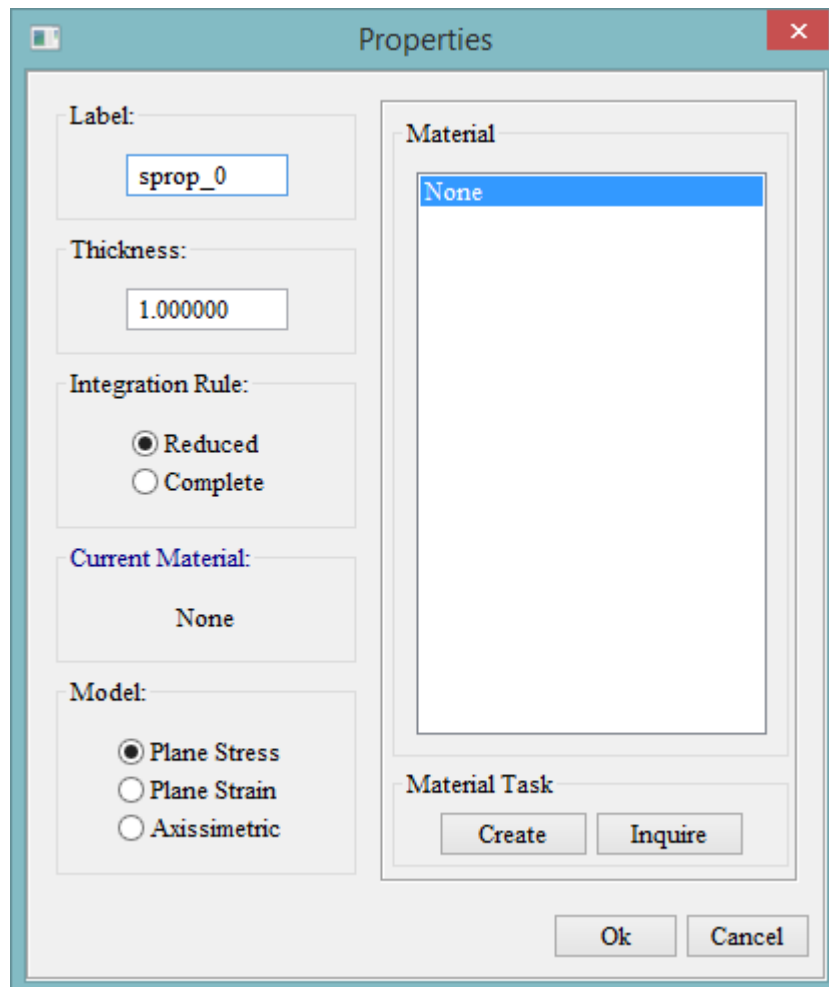


Figura 102 - Valores *default* para os parâmetros de um atributo do tipo “Properties”.

3.12.3.2 – Propriedades de interface

Command Line

`<INTERFPROPERTIES ou IPR> <label> (, model, integration, thickness, gap, hook)
<, interfmaterial>`

Se os parâmetros desse tipo de atributo forem todos definidos na ordem acima (sem omissão de nenhum deles), os seus *labels* são desnecessários. Caso o usuário deseje omitir um ou mais parâmetros (que assumirão seus valores *default*), ou deseje trocar a ordem dos parâmetros, os *labels* dos mesmos passam a ser necessários. O único parâmetro que não pode ser omitido é o material de interface associado ao atributo.

Os parâmetros desse tipo de atributo são semelhantes aos parâmetros do atributo definido anteriormente (“Properties”). Os únicos parâmetros adicionais são:

gap

Valores possíveis para o *label* do parâmetro:

GAP ou G

hook

Valores possíveis para o *label* do parâmetro:
HOOK ou H

A única ressalva a ser feita é o fato de que se o tipo de modelo for axissimétrico, o parâmetro relativo à espessura do material (THICKNESS) não deve ser definido. Dessa forma, se os parâmetros forem passados na ordem (sem necessidade de escrever os seus *labels*), basta ignorar este parâmetro.

Se o usuário deseja passar como parâmetro apenas o material de interface associado ao atributo, os outros parâmetros assumirão seus valores *default* (exceto o *label*).

Via Mouse

Seleciona-se uma entidade do tipo *Line* do tipo *Interface Element*, clica-se no botão *Select*, selecionando a opção *Interface* da Figura 101 o que abrirá uma janela (Figura 103).

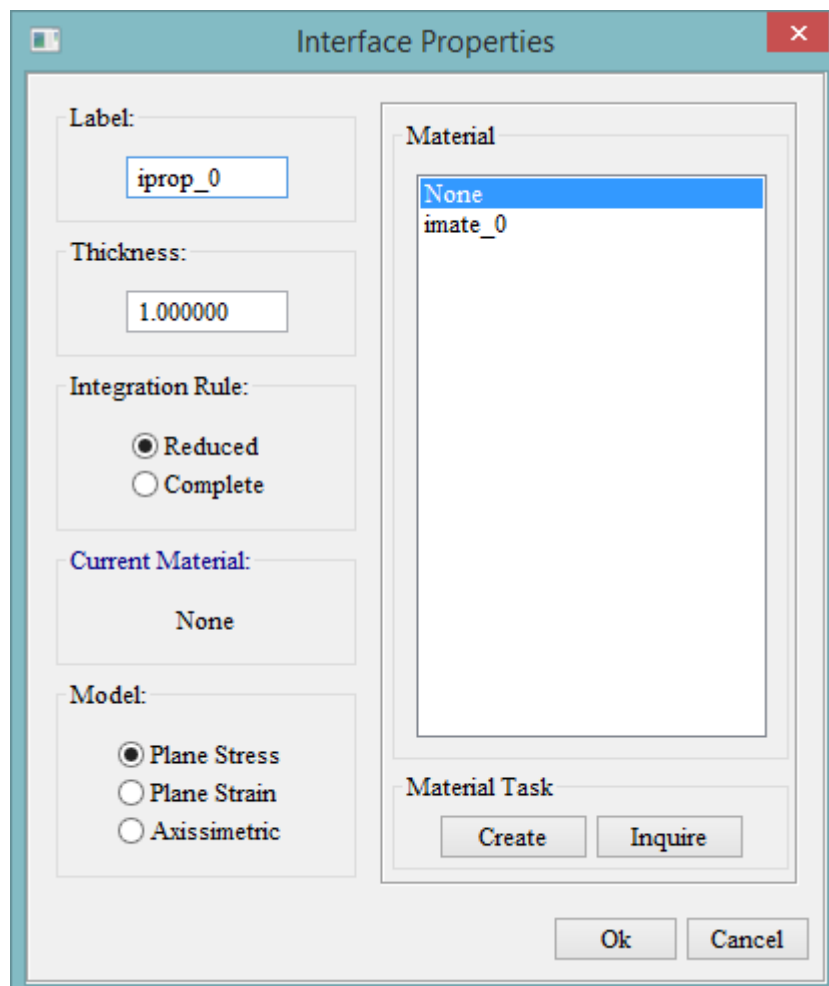


Figura 103 - Valores *default* para os parâmetros de um atributo do tipo “Interface Properties”

Exemplos:

Ex1: ipr "ip1", "ax", 0.3, 1.2, "im1"

Ex2: interfproperties "ip2", "pse", 1.2, 0.3, 0.5, "im2"

Ex3: IPR "ip3", mat = "im3"

Ex4: ipr "ip4", gap = 0.3, t = 1.3, mat = "im4"

3.12.4 – Fronteiras

3.12.4.1 – Escavação

Command Line

<REZONE ou RZ> <time, step, mudweight, face1, face2, ..., facen>

Os parâmetros devem sempre ser passados nessa ordem. Em compensação, todos os *labels* são desnecessários.

time – tempo (número real maior que zero).

Valores possíveis para o *label* do parâmetro:
TIME ou T

step – número do passo (número inteiro maior que zero).

Valores possíveis para o *label* do parâmetro:
STEP ou S

mudweight – peso do fluido de perfuração (número real maior ou igual a zero).

Valores possíveis para o *label* do parâmetro:
MUDWEIGHT ou MW

face1 – nome da primeira face sobre a qual o atributo deve incidir

face2 – nome da segunda face sobre a qual o atributo deve incidir

.

facen – nome da última (n-ésima) face sobre a qual o atributo deve incidir

Vale ressaltar que um comando deste tipo cria apenas UM PASSO de um atributo desse tipo. Para que se criem vários passos para um mesmo valor do parâmetro tempo, devem-se acionar várias vezes esse comando, trocando-se o parâmetro relativo ao passo.

Exemplos:

Ex1: rz 3, 1, 12.4, "f2", "f3"

Ex2: REZONE 3, 2, 12.4, "f4"

Ex3: RZ 3, 3, 12.4, "f5", "f6"

Via Mouse

Selecione-se a opção *Boundaries* no campo *Types*, posteriormente serão disponibilizados três botões: *Rezone*, *Reconstruct* e *Constraint*.

Para se ativar o comando *Rezone* basta se clicar no botão correspondente a função na Figura 104.

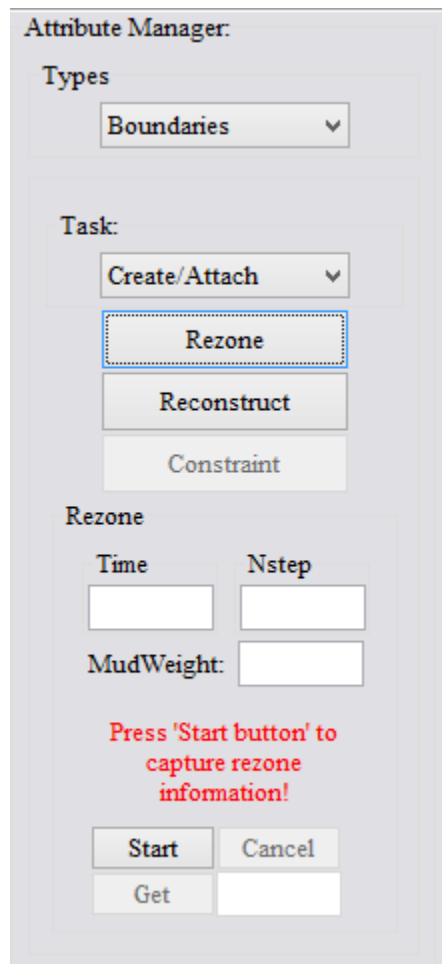


Figura 104 - Interior do frame *Attribute Manager* após se ativar o comando *Rezone*.

A Figura 104 acima ilustra o interior do frame *Attribute Manager* após se clicar no botão *Rezone*.

3.12.4.2 – Reconstrução

Command Line

<RECONSTRUCT ou RC> <time, step, material, face1, face2, ..., facen>

Os parâmetros devem sempre ser passados nessa ordem. Em compensação, todos os *labels* são desnecessários. Esse atributo funciona exatamente como o atributo anterior (REZONE), exceto pelo fato de que, ao invés de se passar um parâmetro relativo ao peso

do fluido de perfuração (*mudweight*), deve-se passar um parâmetro relativo a um material (nome de um atributo do tipo “Material” já criado previamente).

Exemplos:

Ex1: rc 1.3, 1, "mat1", "f7"

Ex2: reconstruct 1.3, 2, "mat2", "f10", "f5"

Ex3: RC 1.3, 3, "mat3", "f7"

Via Mouse

Seleciona-se a opção *Boundaries* no campo *Types* e clica-se no botão *Reconstruct*: (Figura 105)

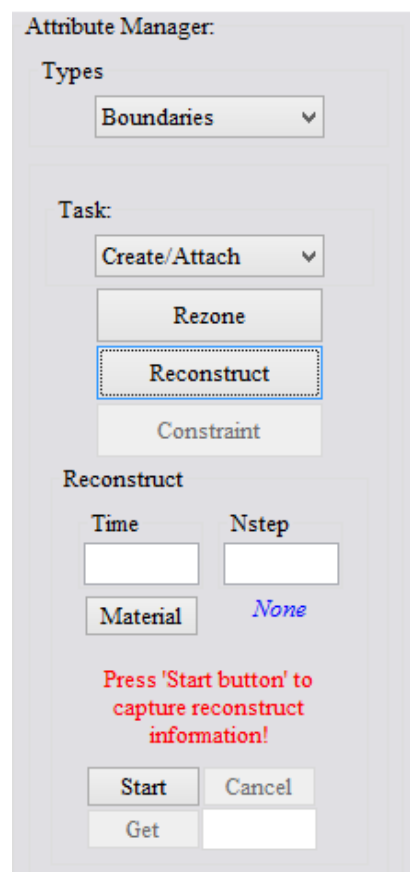


Figura 105 - Interior do frame *Attribute Manager* após se ativar o comando *Reconstruct*.

A Figura 105 acima ilustra o interior do frame *Attribute Manager* após se clicar no botão *Reconstruct*.

3.12.4.3 – Restrições

Command Line

Os atributos relacionados a restrições não precisam ser criados, nem podem ser modificados. Eles são fixos e podem apenas ser agregados às entidades do modelo. Logo, funcionam exatamente como um outro atributo qualquer que se deseje associar a uma ou mais entidades do modelo já previamente selecionadas. Os nomes desses atributos são:

“Free” – entidade fica livre de restrições

“Const_x” – entidade possui deslocamento horizontal restrito

“Const_y” – entidade possui deslocamento vertical restrito

“Const_xy” – entidade possui deslocamento horizontal e vertical restritos

Exemplos:

Ex1: at "Const_x"

Ex2: ATTACH "Const_xy"

Via Mouse

Seleciona-se a opção *Boundaries* no campo *Types* e clica-se no botão *Constraint* (Figura 106).

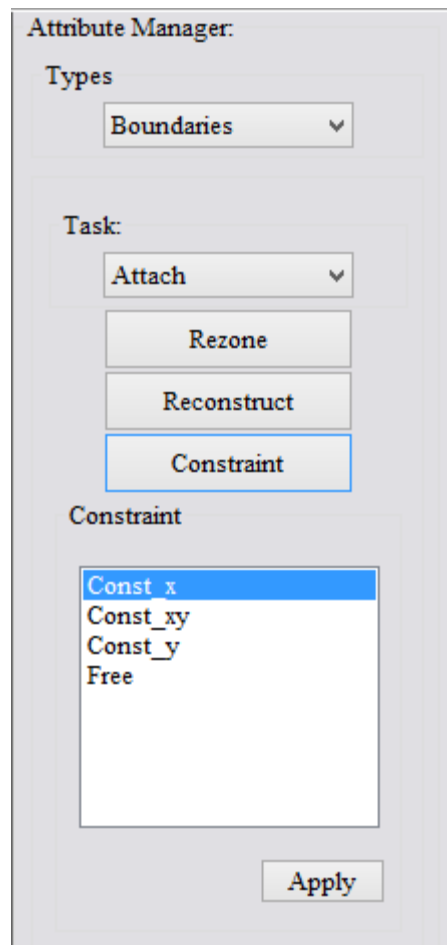


Figura 106 - Interior do frame *Attribute Manager* após se ativar o comando *Constraint*.

A Figura 106 acima ilustra o interior do frame *Attribute Manager* após se clicar no botão *Constraint*, as entidades do tipo *Vertex* e *Line* podem ser classificadas com uma das quatro opções: *Free*, *Const_x*, *Const_y* ou *Const_xy*.

A Figura 107 abaixo ilustra duas entidades do tipo *Line* e uma entidade do tipo *Vertex* onde se usou o comando *Constraint*. Em cada entidade usou-se uma restrição diferente.

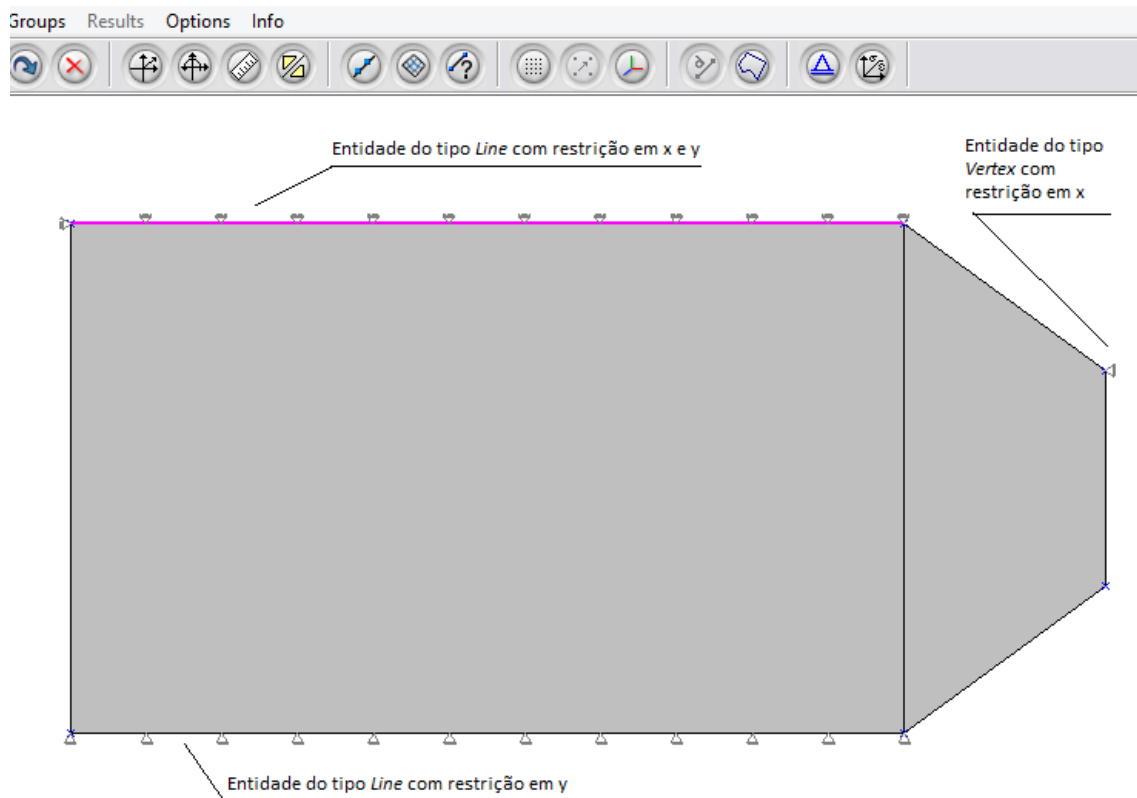


Figura 107 - Resultado após se aplicar o comando *Constraint*.

A Figura 107 acima ilustra o resultado após se aplicar o comando *Constraint* em três entidades: em uma entidade do tipo *Line* usou-se a opção *Const_y*, na outra entidade do tipo *Line* usou-se a opção *Const_xy* e na entidade do tipo *Vertex* usou-se a opção *Const_x*.

3.12.5 – Cargas

3.12.5.1 – Geostática

Command Line

<GEOSTATIC ou GEO> (*plane, mudweight*) (*np, x1, y1, ..., xn, yn*)
 ou
 <GEOSTATIC ou GEO> (*plane, mudweight*) (*depth*)

O uso de cada uma das sintaxes acima irá depender do tipo de plano (*plane*) definido.

Valores possíveis para o *label* do parâmetro:

PLANE ou PL

Valores possíveis para o parâmetro:

“VERTICAL” OU “VERT” ou “V”

“HORIZONTAL” ou “HORIZ” ou “H”

Se o plano for vertical, usa-se a primeira sintaxe. Se o plano for horizontal, usa-se a segunda sintaxe.

Se os parâmetros desse tipo de atributo forem todos definidos na ordem acima (sem omissão de nenhum deles), os seus *labels* são desnecessários. Caso o usuário deseje omitir um ou mais parâmetros (que assumirão seus valores *default*), ou deseje trocar a ordem dos parâmetros, os *labels* dos mesmos passam a ser necessários.

Valores possíveis para o *label* do parâmetro:

MUDWEIGHT ou MW

Valores possíveis para o *label* do parâmetro:

NP – número de pontos (número inteiro maior que zero)

Valores possíveis para os *labels* dos parâmetros:

X1 – abscissa do primeiro ponto

Y1 – ordenada do primeiro ponto

.

.

.

XN – abscissa do último (n-ésimo) ponto

YN – ordenada do último (n-ésimo) ponto

Valores possíveis para o *label* do parâmetro:

DEPTH ou D

Se o usuário não definir um ou mais desses parâmetros, eles assumirão seus valores *default* (exceto o *label*).

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*. Caso a entidade selecionada seja do tipo *Region* ao se clicar neste botão se terá oito opções de carga: *Inertial*, *Geostatic*, *Temperature*, *Poropressure*, *Initial Stress*, *Boundary Pressure*, *Initial Temperature* e *Initial Poropressure* de acordo com a Figura 108.

Para se ativar cargas geostáticas tem que se selecionar uma entidade do tipo *Region* e clicar no botão *Select* da Figura 108. Após se optar pela opção *Geostatic* será aberta uma janela (Figura 109).

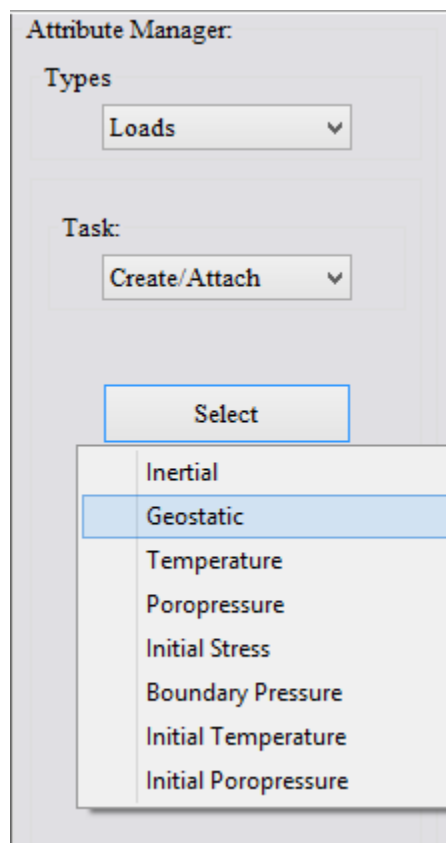


Figura 108 - Opções de cargas para entidades do tipo *Region*.

A Figura 108 acima ilustra as opções de cargas para entidades do tipo *Region*.

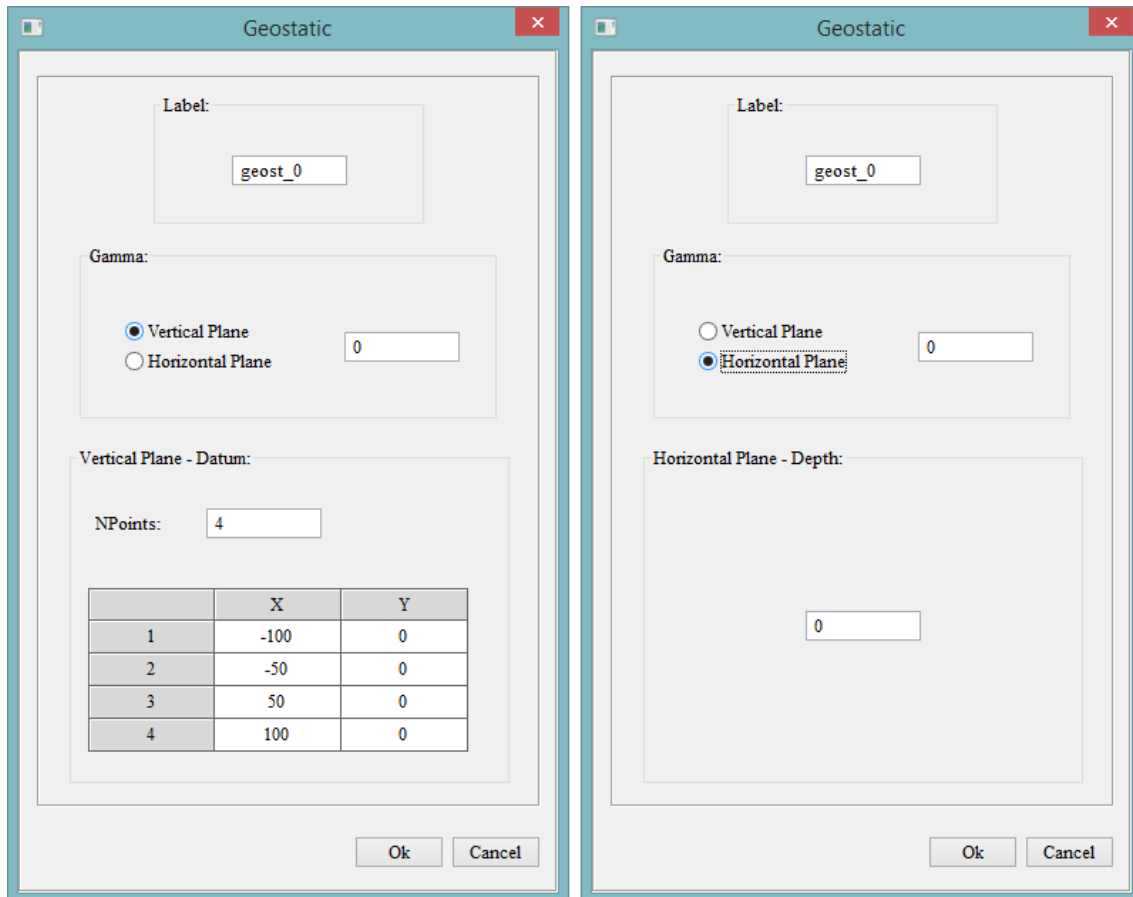


Figura 109 - Valores *default* para os parâmetros de um atributo de Carga Geostática (com plano vertical e horizontal).

3.12.5.3 – Cargas genéricas

Cargas genéricas são atributos de carga cujos parâmetros podem ser generalizados da seguinte forma:

- Um ou mais valores reais
- Uma ou duas funções de tempo (quando são duas, significa que há uma para cada eixo – horizontal (x) e vertical (y)).
- Zero ou uma função de espaço

Command Line

$\langle GENATTR \rangle \langle label \rangle (val1, \dots, valn) [timefuncs] (spatfunc)$

A função de espaço, para os atributos que a possuem, pode ser definida, mas nunca é necessária. As funções de tempo, independentemente do fato de serem únicas ou uma para cada eixo, podem ou não ser necessárias, dependendo do tipo de atributo.

Cada tipo de carga genérica possui *labels* específicos para os valores reais que agrega, que serão vistos de acordo com o caso. Contudo, as funções de tempo e espaço possuem *labels* iguais para todos os atributos desse tipo, que são:

- Funções de tempo

Valores possíveis para o *label* do parâmetro:

- 1) Quando a função de tempo é única
TIME FUNCTION ou TIMEFUNC ou TFUNC ou TF
- 2) Quando há uma função de tempo para cada eixo
TIME FUNCTION X ou TIMEFUNCX ou TFUNCX ou TFX
TIME FUNCTION Y ou TIMEFUNCY ou TFUNCY ou TFY

- Função de espaço

Valores possíveis para o *label* do parâmetro:

SPATIALFUNCTION ou SPATFUNC ou SFUNC ou SF

Se o usuário definir todos os parâmetros de algum atributo desse tipo em ordem, não há a necessidade de se utilizar os *labels* dos parâmetros. Deve-se atentar para o fato de que, se uma função de tempo não é necessária e o usuário não deseja associar nenhuma função desse tipo ao atributo, deve-se usar o valor “NONE”. Se o atributo possui como um de seus parâmetros uma função de espaço e o usuário não deseja associar nenhuma função desse tipo ao atributo, pode simplesmente não digitar nada ou digitar o valor “NONE” para esse parâmetro.

No entanto, se quiser omitir um ou mais parâmetros ou se desejar alterar a ordem dos parâmetros, necessariamente deve utilizar todos os *labels* dos parâmetros que definir. A ordem dos valores reais inerentes a cada atributo será vista adiante.

Apresenta-se a seguir uma lista com os tipos de atributos classificados como cargas genéricas. São mostrados os *labels* possíveis para cada atributo, os *labels* possíveis para os valores reais que aquele atributo agrega, a ordem em que estes valores reais devem ser escritos para que não haja a necessidade de se utilizar os seus *labels*, o número de funções de tempo e de espaço de cada um e a necessidade ou não de se definir a(s) função(ões) de tempo que ele eventualmente agregue.

Os valores *default* para os parâmetros representados por valores reais de todos os atributos genéricos são sempre iguais a 0.0 (zero), exceto para o parâmetro GY do atributo INERTIAL, que possui como valor *default* -9.81 (valor de *g* – gravidade ao nível do mar).

➤ FORÇA

Command Line

Valores possíveis para o *label* desse parâmetro:

FORCE ou F

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

PX

PY

Número de funções de tempo que agrega: 2 (necessárias)

Número de funções de espaço que agrega: 0

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 FORCE "force1", px = 10, py = 1.25, tfuncx = "tf_1", tfuncy = "tf_1"

Ex2: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 FORCE "force2", 10, 5, "tf_1", "tf_1"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*. Caso a entidade selecionada seja do tipo *Vertex* ao se clicar neste botão se terá seis opções de carga: *Force*, *Temperature*, *Poropressure*, *Initial Temperature*, *Initial Poropressure* e *Prescribed Displacement* de acordo com a Figura 110.

Para se ativar cargas do tipo *Force* tem que se selecionar uma entidade do tipo *Vertex* e clicar no botão *Select* da Figura 110. Após se optar pela opção *Force* será aberta uma janela (Figura 111).

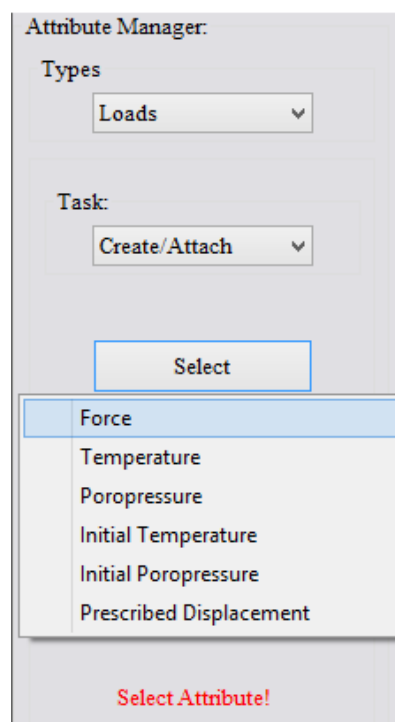


Figura 110 - Opções de cargas para entidades do tipo *Vertex*.

A Figura 110 acima ilustra as opções de cargas para entidades do tipo *Vertex*.

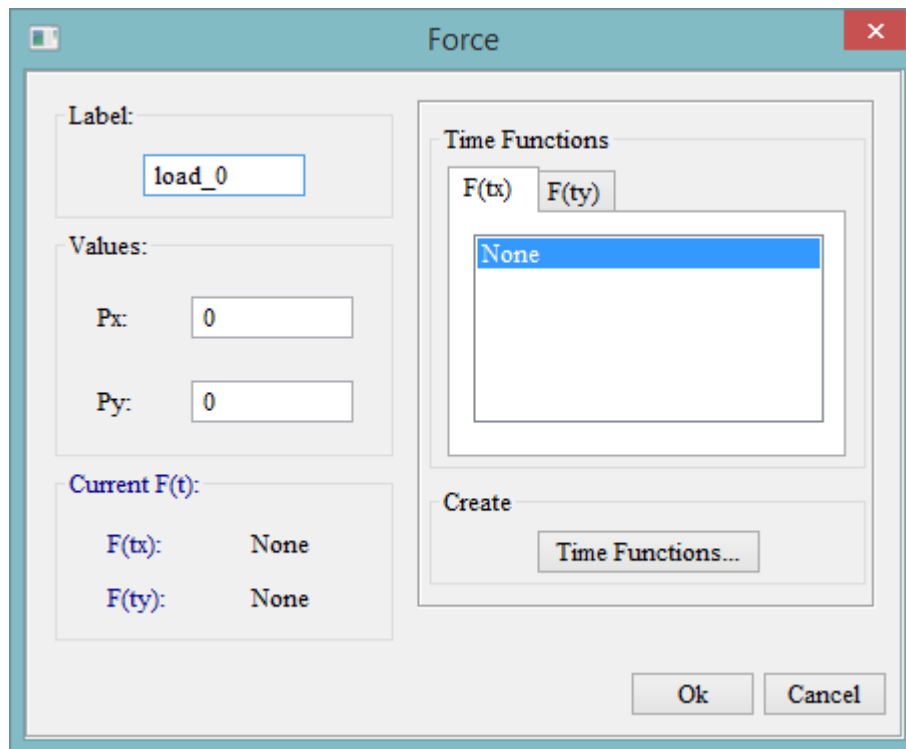


Figura 111 - Janela aberta após seleção carga do tipo *Force*.

A Figura 111 acima ilustra a janela aberta após se selecionar a carga do tipo *Force*.

➤ TEMPERATURA

Command Line

Valores possíveis para o *label* desse parâmetro:
TEMPERATURE ou T

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):
VALUE ou VAL ou V

Número de funções de tempo que agrega: 1 (necessária)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
t "temp1", 130.4, "tf_1", "none"

Ex2: sf "sf_2", 4
t "temp2", value = 100, tfunc = "tf_1", spatfunc = "sf_2"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, para qualquer tipo de entidade a opção *Temperature* será disponibilizada.

Como exemplo, usou-se uma entidade selecionada do tipo *Vertex*. Para ativar cargas do tipo *Temperature*, tem que se clicar no botão *Select* da Figura 110 e escolher a opção *Temperature*, o que abrirá uma janela (Figura 112).

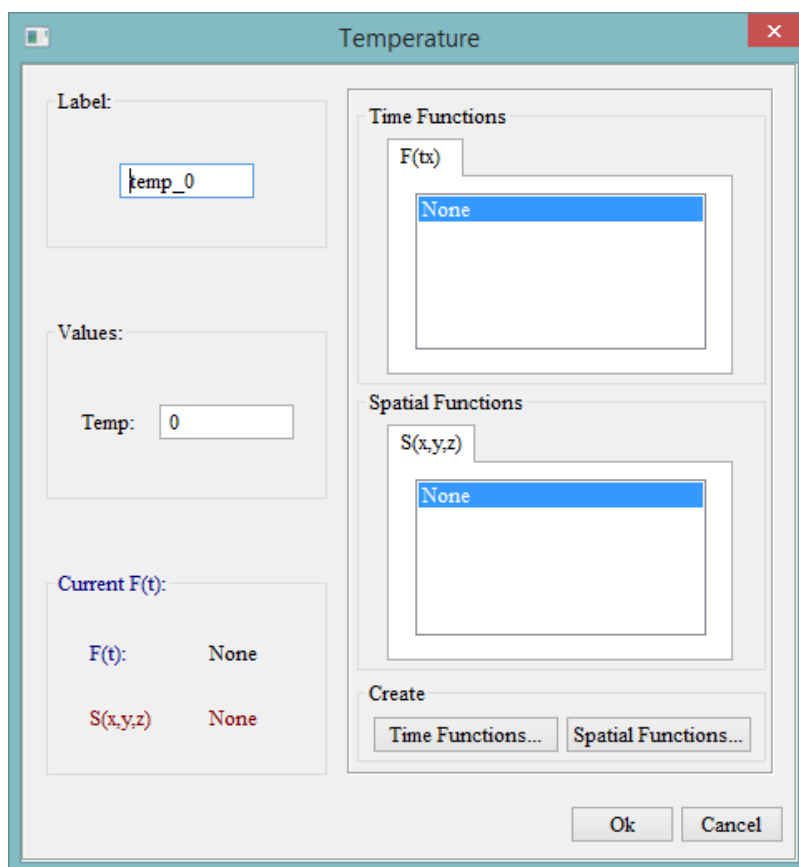


Figura 112 - Janela aberta após se selecionar carga do tipo *Temperature*.

A Figura 112 acima ilustra a janela aberta após se selecionar a carga do tipo *Temperature*.

➤ POROPRESSÃO

Command Line

Valores possíveis para o *label* desse parâmetro:
POROPRESSURE ou PP

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):
VALUE ou VAL ou V

Número de funções de tempo que agrega: 1 (necessária)

Número de funções de espaço que agrega: 1

Ex1: `tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0`
 `sf "sf_2", 4`
 `poropressure "pp1", value = 12, tfunc = "tf_1", spatfunc = "sf_2"`

Ex2: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 poropressure "pp2", 10, tf = "tf_1", sf = "none"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, para qualquer tipo de entidade a opção *Poropressure* será disponibilizada.

Como exemplo usou-se uma entidade selecionada do tipo *Vertex*. Para ativar cargas do tipo *Poropressure* tem que se clicar no botão *Select* da Figura 110 e escolher a opção *Poropressure*, o que abrirá uma janela (Figura 113).

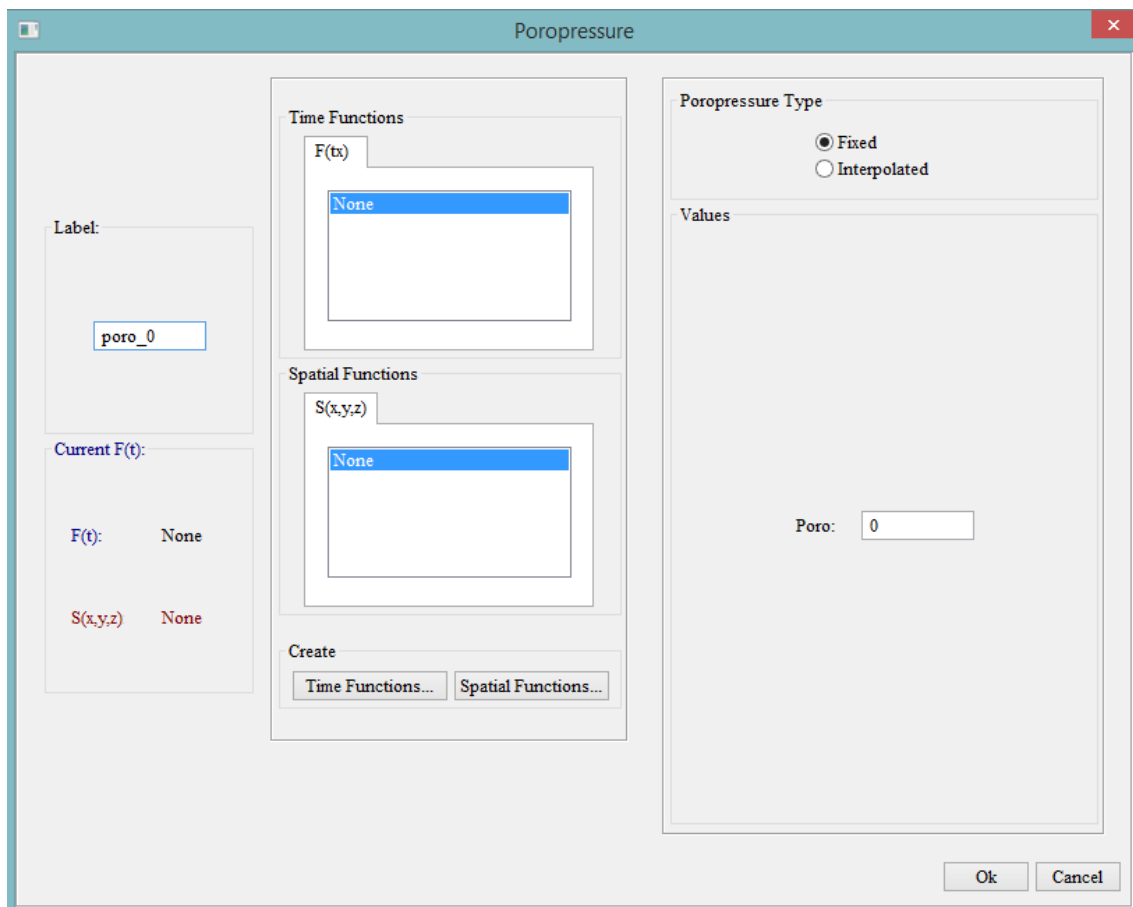


Figura 113 - Janela aberta após se selecionar carga do tipo *Poropressure*.

A Figura 113 acima ilustra a janela aberta após se selecionar a carga do tipo *Poropressure*.

➤ DESLOCAMENTO PRESCRITO

Command Line

Valores possíveis para o *label* desse parâmetro:
 DISPLACEMENT ou DISP

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

PRESC_X ou PRESCX ou PX

PRESC_Y ou PRESCY ou PY

Número de funções de tempo que agrega: 2 (não são necessárias)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 displacement "d1", 10, 10, tfuncx = "tf_1", tfuncy = "tf_1", spatfunc =
 "none"

Ex2: sf "sf_2", 4
 displacement "d2", px = 1, py = 2, tfx = "tf_1", tfy = "tf_1", sf = "sf_2"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, para as entidades do tipo *Vertex* e *Line* a opção *Prescribed Displacement* será disponibilizada.

Como exemplo usou-se uma entidade selecionada do tipo *Vertex*. Para ativar cargas do tipo *Prescribed Displacement* tem que se clicar no botão *Select* da Figura 110 e escolher a opção *Prescribed Displacement*, o que abrirá uma janela (Figura 114).

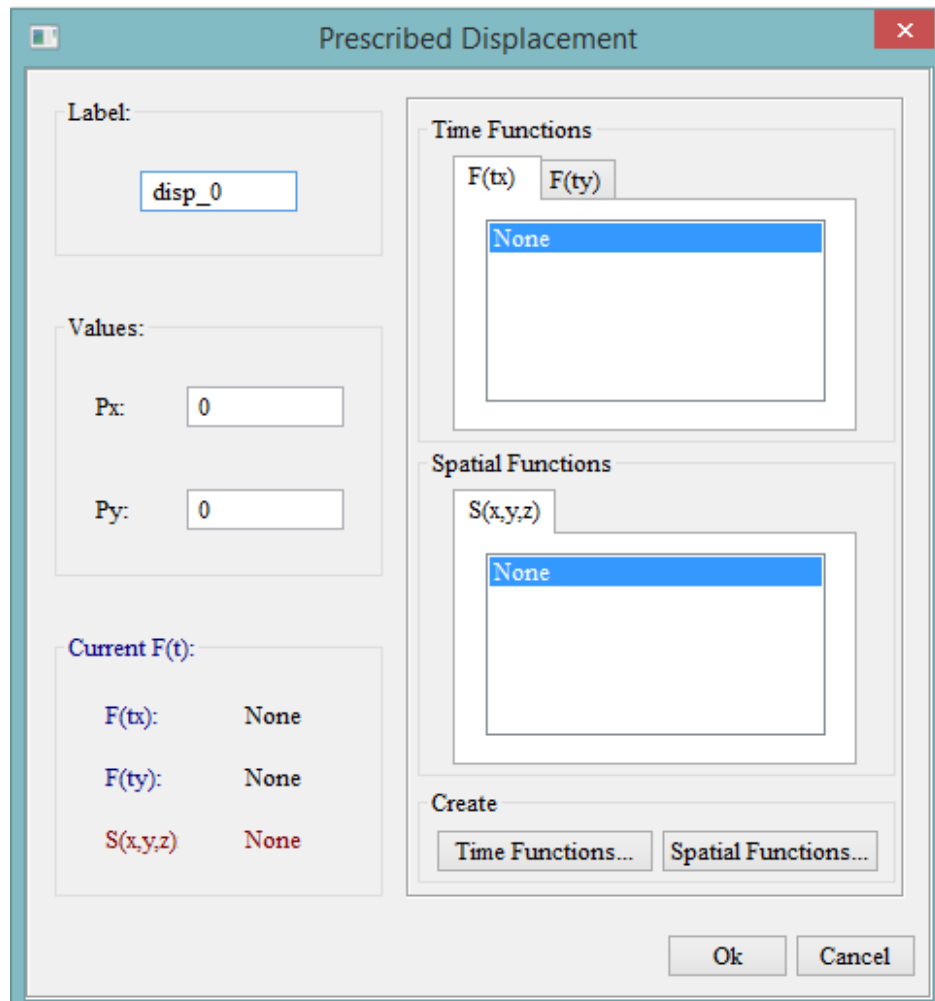


Figura 114 - Janela aberta após se selecionar carga do tipo *Prescribed Displacement*.

A Figura 114 acima ilustra a janela aberta após se selecionar a carga do tipo *Prescribed Displacement*.

➤ TEMPERATURA INICIAL

Command Line

Valores possíveis para o *label* desse parâmetro:
INITIALTEMP ou IT

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

VALUE ou VAL ou V

Número de funções de tempo que agrega: 0

Número de funções de espaço que agrega: 1

Ex1: sf "sf_2", 4
it "it1", value = 12, spatfunc = "sf_2"

Ex2: IT "it2", sf = "sf_2", v = 12.4

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, para qualquer tipo de entidade a opção *Initial Temperature* será disponibilizada.

Como exemplo usou-se uma entidade selecionada do tipo *Vertex*. Para ativar cargas do tipo *Initial Temperature* tem que se clicar no botão *Select* da Figura 110 e escolher a opção *Initial Temperature*, o que abrirá uma janela (Figura 115).

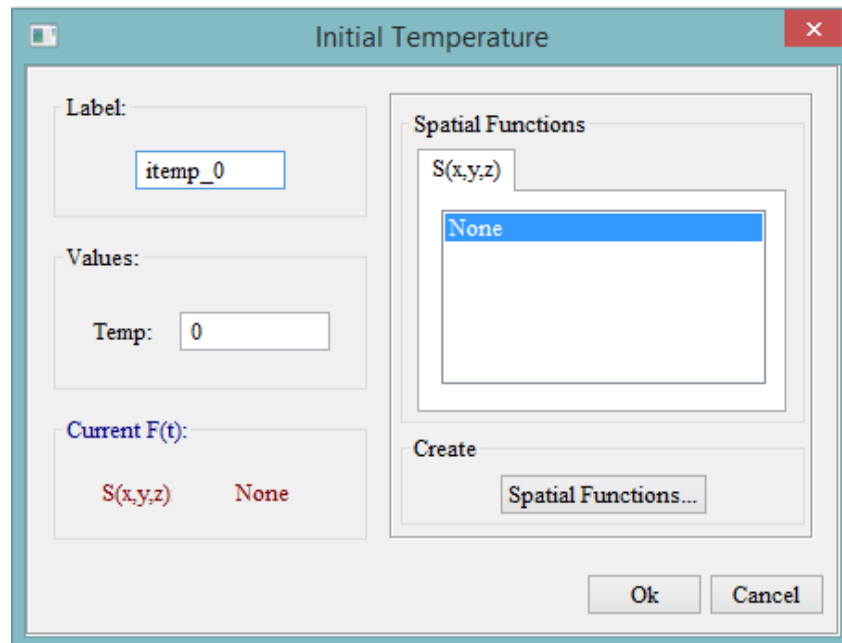


Figura 115 - Janela aberta após se selecionar carga do tipo *Initial Temperature*.

A Figura 115 acima ilustra a janela aberta após se selecionar a carga do tipo *Initial Temperature*.

➤ POROPRESSÃO INICIAL

Command Line

Valores possíveis para o *label* desse parâmetro:
INITIALPORO ou IP

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):
VALUE ou VAL ou V

Número de funções de tempo que agrega: 0

Número de funções de espaço que agrega: 1

Ex1: initialporo "ip1", 1.2, "none"

Ex2: sf "sf_2", 4

ip "ip2", sf = "sf_2", value = 12

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, para qualquer tipo de entidade a opção *Initial Poropressure* será disponibilizada.

Como exemplo usou-se uma entidade selecionada do tipo *Vertex*. Para ativar cargas do tipo *Initial Poropressure* tem que se clicar no botão *Select* da Figura 110 e escolher a opção *Initial Poropressure*, o que abrirá uma janela (Figura 116).

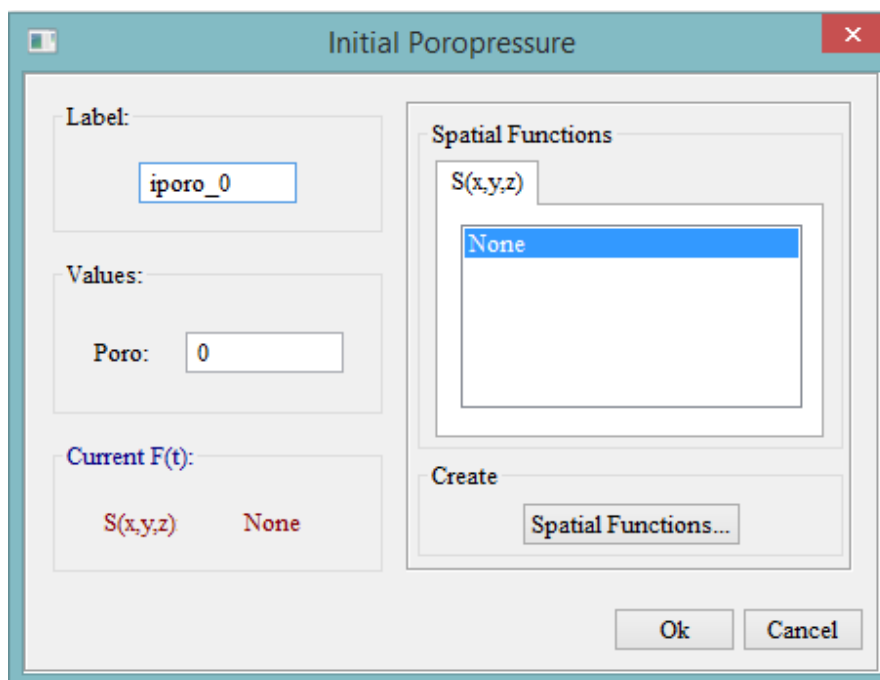


Figura 116 - Janela aberta após se selecionar carga do tipo *Initial Poropressure*.

A Figura 116 acima ilustra a janela aberta após se selecionar a carga do tipo *Initial Poropressure*.

➤ PRESSÃO

Command Line

Valores possíveis para o *label* desse parâmetro:
PRESSURE ou PRESS

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

VALUE ou VAL ou V

Número de funções de tempo que agrega: 2 (necessárias)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0

sf "sf_2", 4

pressure "Pressure1", value = 10, tfx = "tf_1", tfy = "tf_1", spatfunc = "sf_2"

Ex2: PRESS "press3", tfy = "tf_1", tfx = "tf_1", spatfunc = "sf_2", v = 120

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*. Caso a entidade selecionada seja do tipo *Line*, ao se clicar neste botão se terá nove opções de carga: *Pressure*, *Mass Flux*, *Heat Flux*, *Temperature*, *Poropressure*, *Uniform Load*, *Initial Temperature*, *Initial Poropressure* e *Prescribed Displacement*, de acordo com a Figura 117.

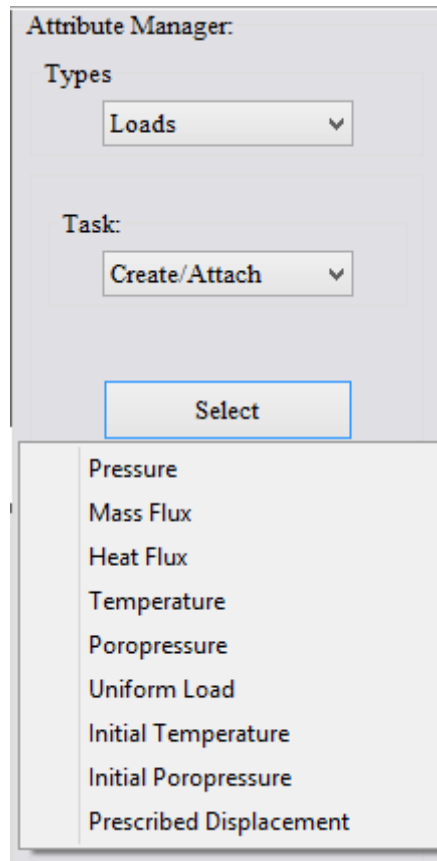


Figura 117 – Opções de carga para entidades do tipo *Line*.

Ao se clicar no botão a opção *Pressure* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

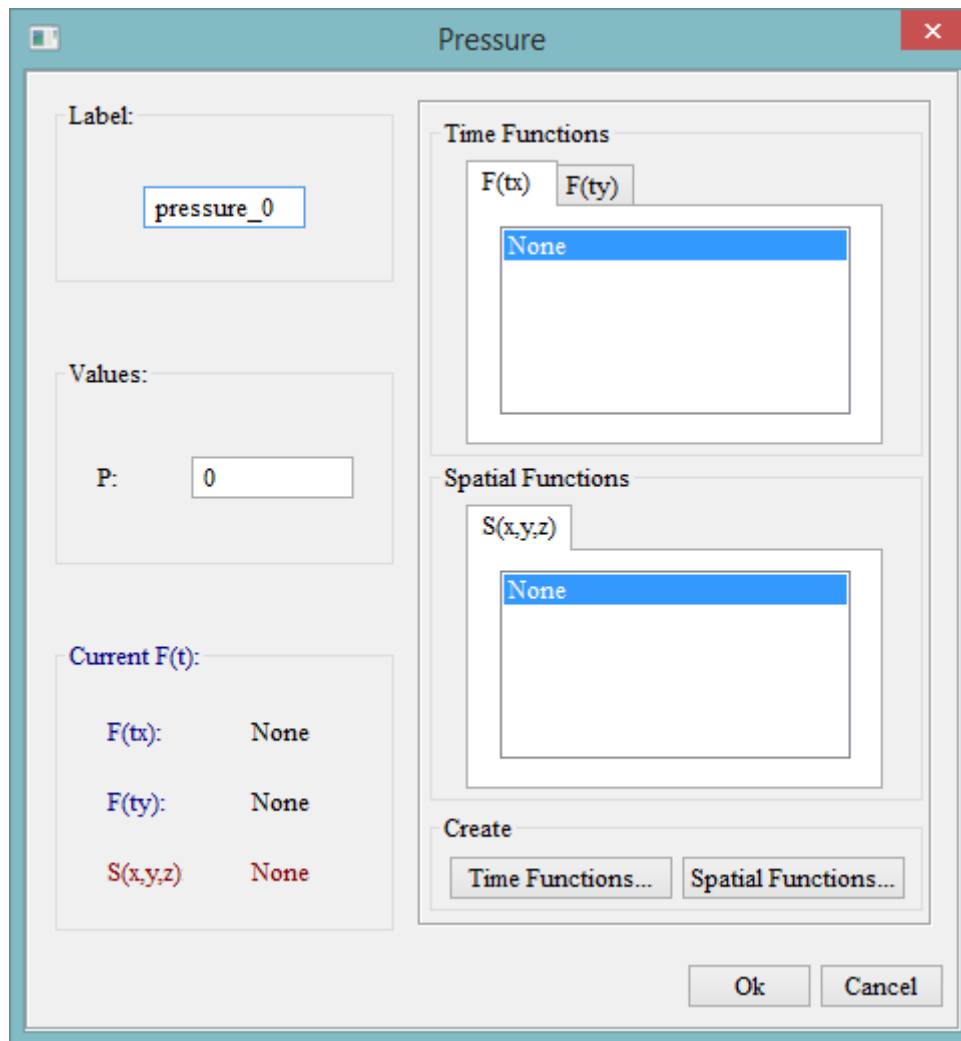


Figura 118 - Janela aberta após seleção de carga do tipo *Pressure*.

➤ FLUXO DE CALOR

Command Line

Valores possíveis para o *label* desse parâmetro:
HEATFLUX ou HF

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):
VALUE ou VAL ou V

Número de funções de tempo que agrega: 1 (necessária)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 sf "sf_2", 4
 heatflux "heat1", value = 10, tf = "tf_1", spatfunc = "sf_2"

Ex2: hf "heat2", val = 12, tf = "tf_1", sf = "sf_2"

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Line* (Figura 117). Ao se clicar no botão a opção *Heat Flux* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

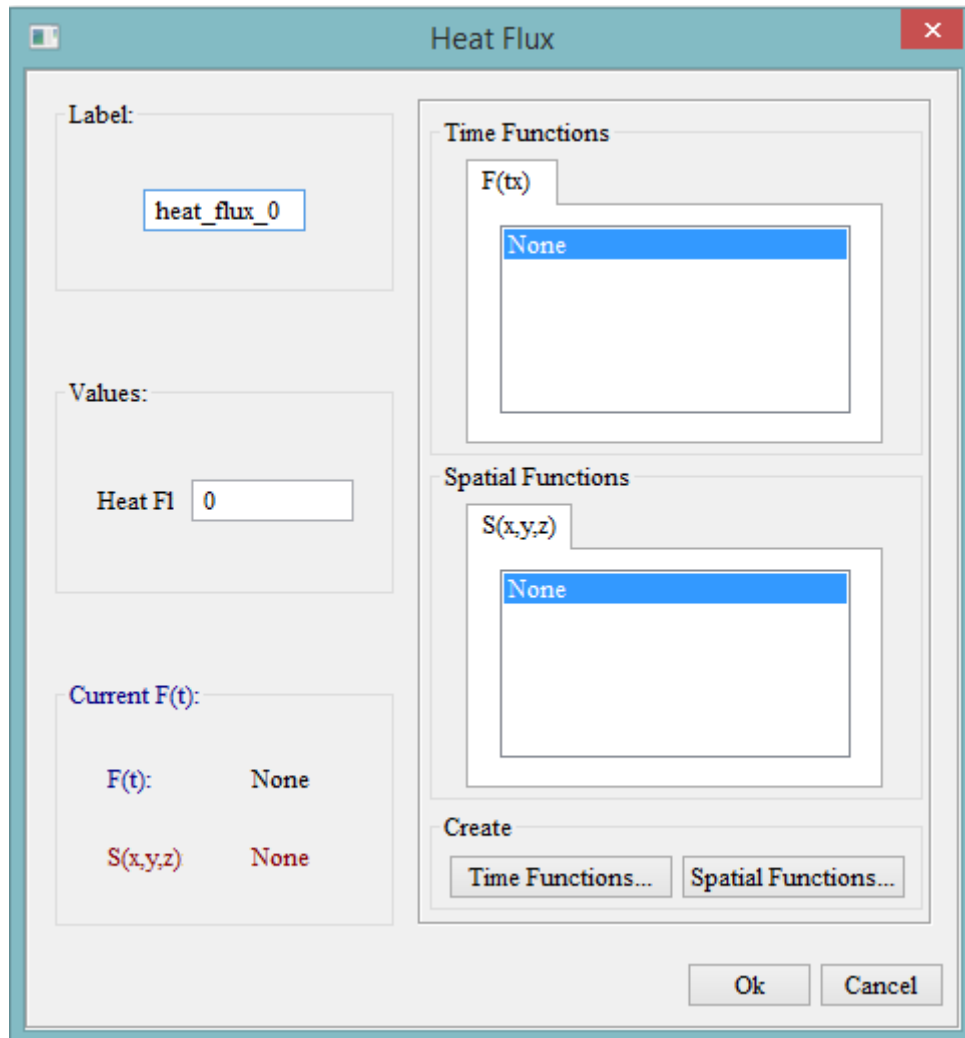


Figura 119 - Janela aberta após se selecionar carga do tipo *Heat Flux*.

➤ FLUXO DE MASSA

Command Line

Valores possíveis para o *label* desse parâmetro:
 MASSFLUX ou MF

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):
 VALUE ou VAL ou V

Número de funções de tempo que agrega: 1 (necessária)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
massflux "mf1", 4.3, "tf_1", "none"

Ex2: sf "sf_2", 4
MF "mf2", sf = "sf_2", tf = "tf_1", value = 3.4

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Line* (Figura 117). Ao se clicar no botão a opção *Mass Flux* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

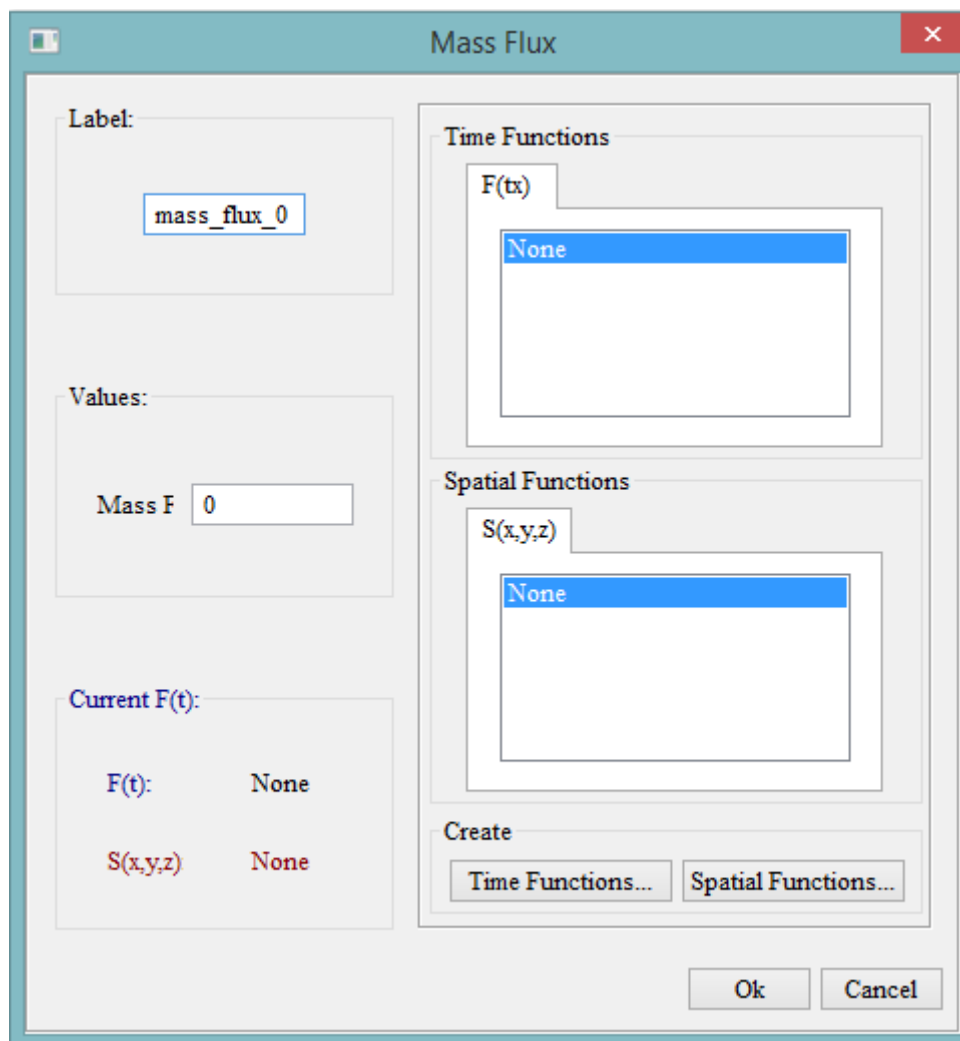


Figura 120 - Janela aberta após se selecionar carga do tipo *Mass Flux*.

➤ CARGA DISTRIBUÍDA

Command Line

Valores possíveis para o *label* desse parâmetro:
DISTRIBUTEDLOAD ou DL

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

PX

PY

Número de funções de tempo que agrega: 2 (necessárias)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0

sf "sf_2", 4

dl "dl1", px = 12, py = -13.45, tfx = "tf_1", tfy = "tf_1", spatfunc = "sf_2"

Ex2: dl "dl2", px = 1, py = -1, tfx = "tf_1", tfy = "tf_1", sf = "sf_2"

Via Mouse

Selecione-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Line* (Figura 117). Ao se clicar no botão a opção *Uniform Load* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

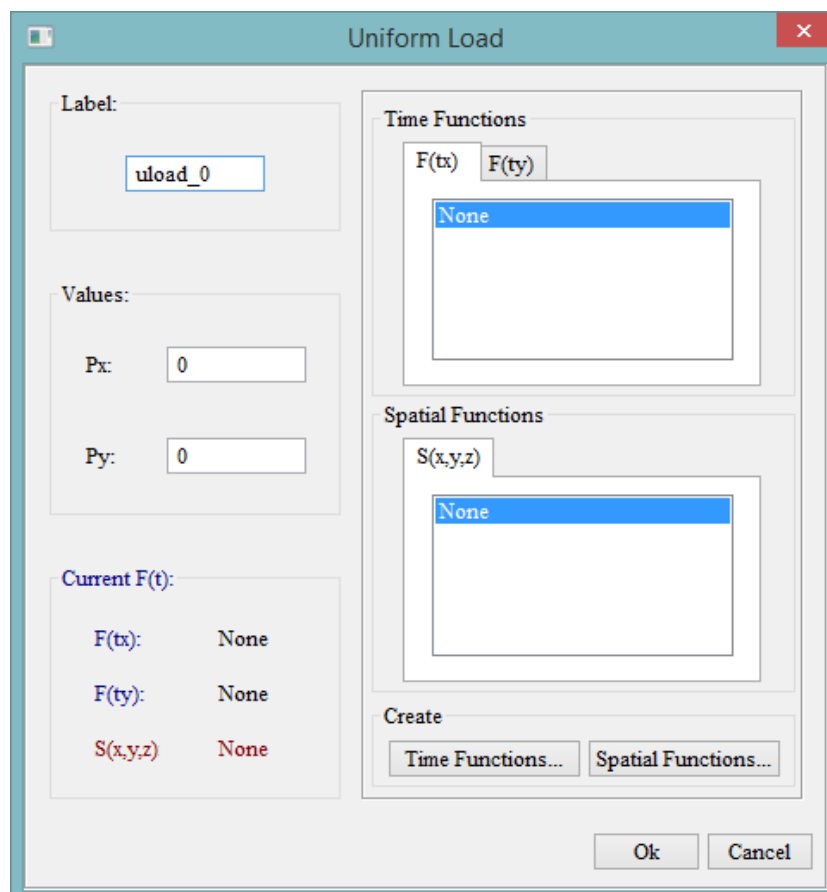


Figura 121 - Janela aberta após se selecionar carga do tipo *Uniform Load*.

➤ INERCIAL

Command Line

Valores possíveis para o *label* desse parâmetro:
INERTIAL ou I

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

GX
GY

Número de funções de tempo que agrega: 2 (necessárias)

Número de funções de espaço que agrega: 0

Valor *default* para o parâmetro GY: -9.81

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 inertial "in1", 1.2, 2.3, "tf_1", "tf_1"

Ex2: i "in2", gx = 1.2, gy = 2.3, tfx = "tf_1", tfy = "tf_1"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Region* (Figura 108). Ao se clicar no botão a opção *Inertial* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

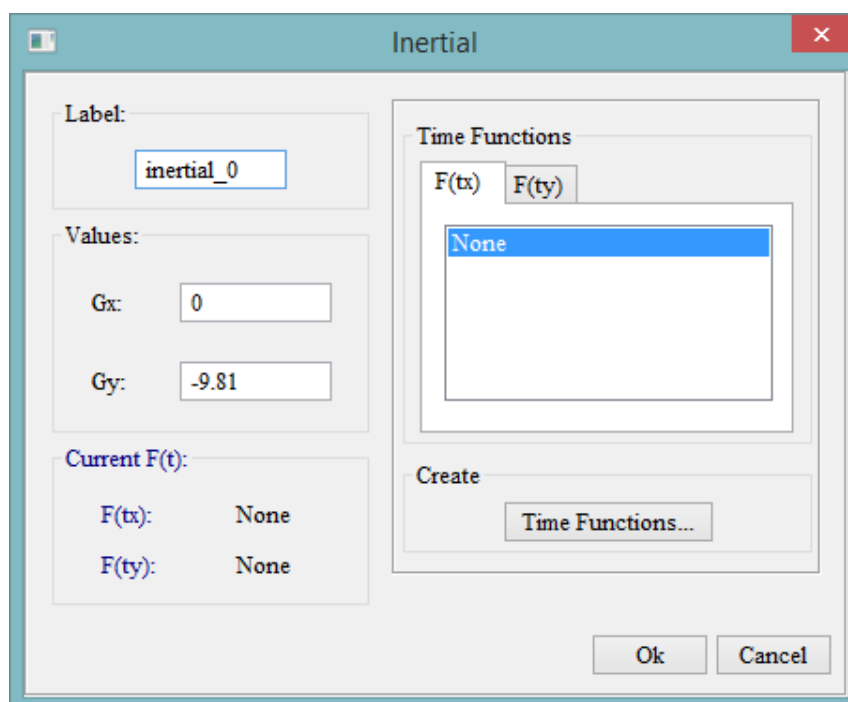


Figura 122 - Janela aberta após se selecionar carga do tipo *Inertial*.



➤ TENSÃO INICIAL

Command Line

Valores possíveis para o *label* desse parâmetro:
INITIALSTRESS ou IS

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

SX
SY
SZ
SXY

Número de funções de tempo que agrega: 1 (necessária)

Número de funções de espaço que agrega: 0

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 is "is1", 1.2,3.4,0.0,3.56, "tf_1"

Ex2: is "is5", sx = 1, sy = 2, sz = 3, sxy = 0, tfunc = "tf_1"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Region* (Figura 108), ao se clicar no botão a opção *Initial Stress* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

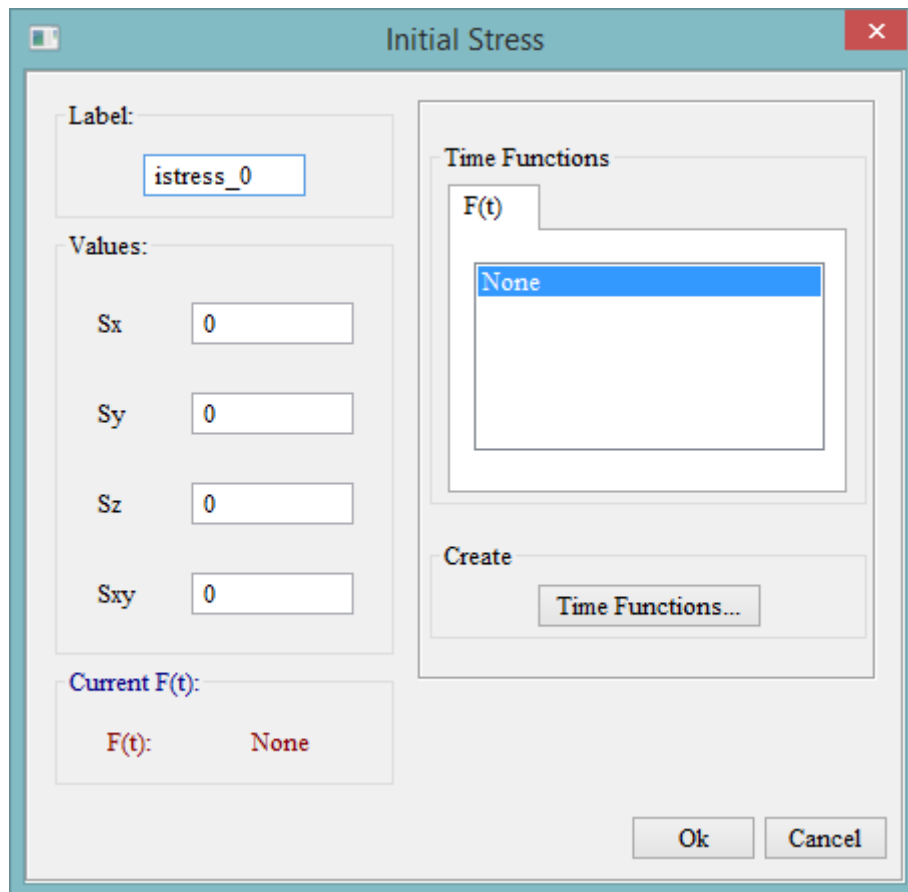


Figura 123 - Janela aberta após se selecionar carga do tipo *Initial Stress*.

➤ PRESSÃO DE FRONTEIRA

Command Line

Valores possíveis para a *label* desse parâmetro:

BOUNDPRESSURE ou BPRESS

Valores possíveis para os *labels* dos parâmetros representados por valores reais (esses parâmetros já estão colocados em ordem):

VALUE ou VAL ou V

Número de funções de tempo que agrega: 2 (necessárias)

Número de funções de espaço que agrega: 1

Ex1: tf "tf_1", 3, 0.0, 1.2, -4.5, 5.3, 9, 0.0
 boundpressure "bp1", 13.56, "tf_1", "tf_1", "NONE"

Ex2: sf "sf_2", 4
 BPRESS "bp3", tfy = "tf_1", v = 4.5, tfx = "tf_1", spatfunc = "sf_2"

Via Mouse

Seleciona-se a opção *Loads* no campo *Types*, posteriormente será disponibilizado o botão *Select*, caso a entidade selecionada seja do tipo *Region* (Figura 108), ao se clicar no botão

a opção *Boundary Pressure* será disponibilizada, clicando-se nesta opção será aberta a seguinte janela:

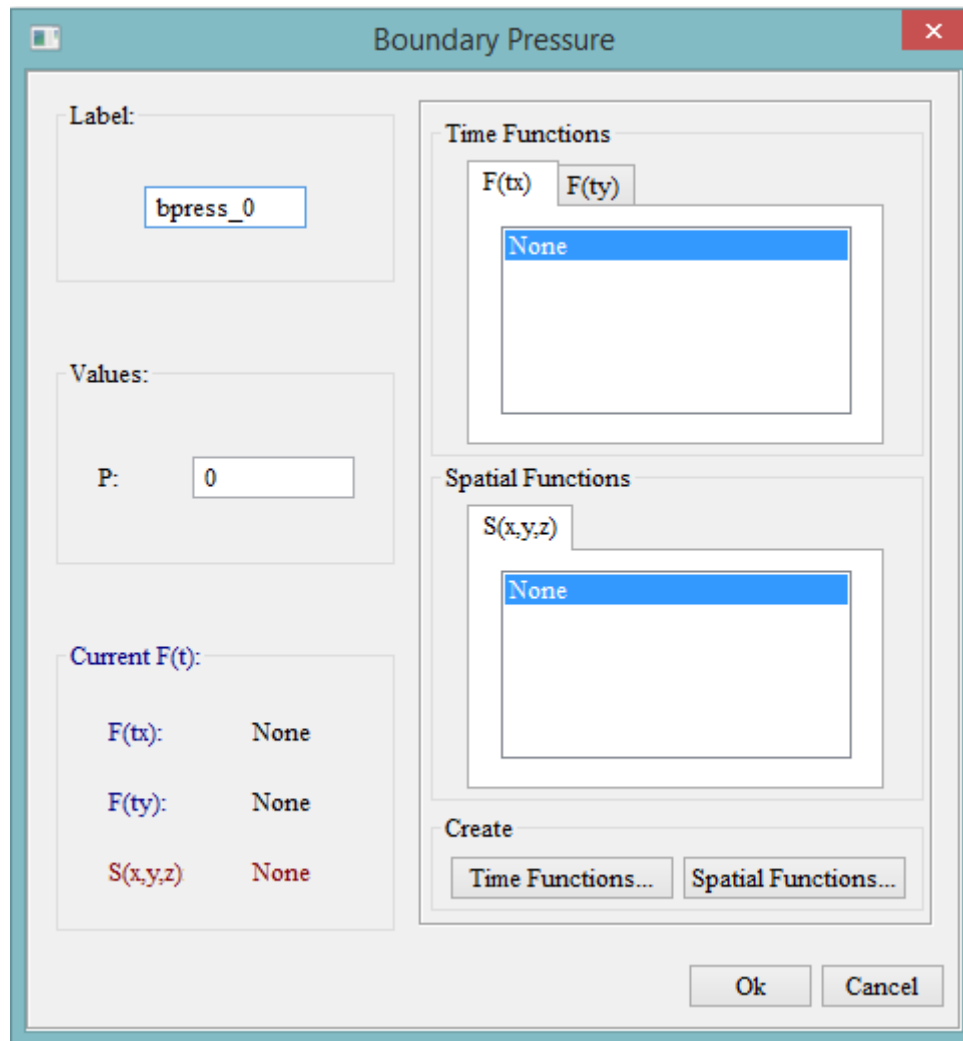


Figura 124 - Janela aberta após se selecionar carga do tipo *Boundary Pressure*.

4 – IMPORTAÇÃO / EXPORTAÇÃO DE ARQUIVOS

Todos os arquivos gerados durante o processo de modelagem no Sigma2D podem ser reutilizados posteriormente através das ferramentas de importação e exportação de arquivos. Isto é bastante positivo do ponto de vista da diminuição do tempo gasto na modelagem, principalmente na geração das geometrias e criação de atributos. Também, tais ferramentas podem permitir uma comunicação com outros softwares de modelagem e CAD, o que é bastante interessante.

O Sigma2D gera, em ambiente de geometria (pré-processamento), 4 arquivos diferentes durante uma modelagem, são eles: .cml, .mtl, .mgr e .top. Ao abrir um modelo já existente, o arquivo selecionado será o .mtl, mas é necessário que todos os quatro arquivos estejam no mesmo diretório para correta leitura.

4.1 – Abrir arquivos

A interface do Sigma2D apresenta três botões destinados a abertura de arquivos, indicados a seguir:

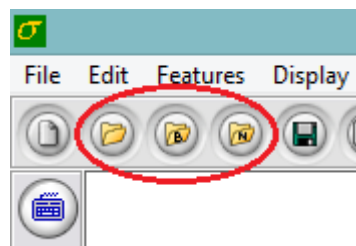


Figura 125 - Abrir arquivos geométricos, binários ou neutros

4.1.1 – Abrir arquivo geométrico

Pode-se abrir arquivos com extensão .mtl através do botão de abertura de arquivos geométricos (botão da esquerda na Figura 125) ou do item Geometric Model... do sub-item Open do menu File. Ambos levam ao diálogo de abertura de arquivos, com filtro para extensão .mtl. Ao abrir um arquivo desse tipo, a tela é imediatamente atualizada e exibe o modelo criado, assim como seus atributos.

A Figura 126 ilustra a tela do programa após ser aberto o arquivo ModeloSimples.mtl, referente ao tutorial de geração modelo de placa simples.

Ao abrir um arquivo de geometria (.mtl), é importante que os arquivos de extensão .cml, .top e .mgr estejam na mesma pasta, pois estes arquivos também são lidos para gerar o modelo na tela do Sigma2D.

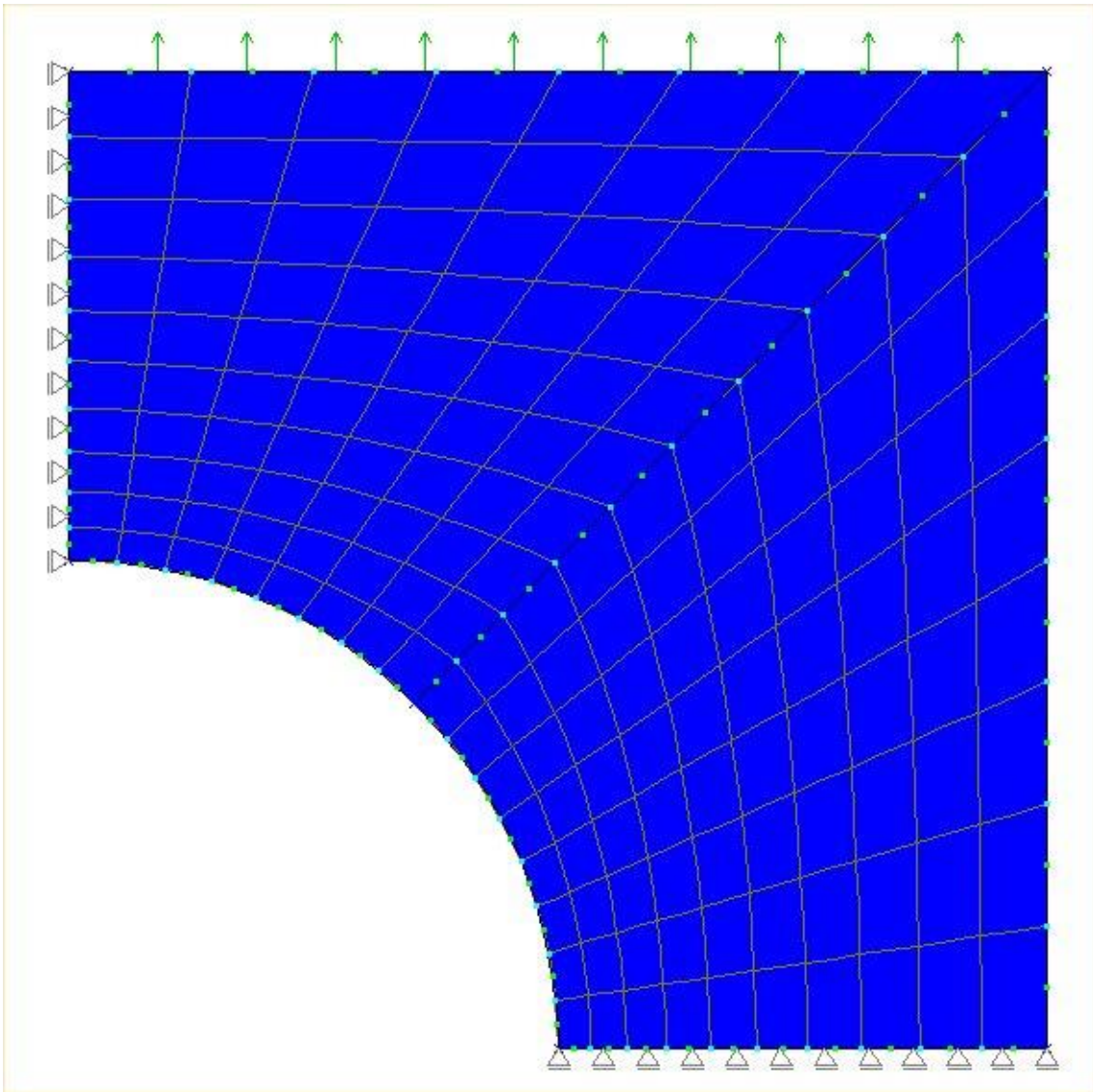


Figura 126 - Modelo gerado na tela após abrir um arquivo geométrico (.mtl)

4.1.2 – Abrir arquivo binário

A abertura de arquivos binários (arquivos com extensão .p2b) leva o usuário ao ambiente de malha do Sigma2D. Este arquivo contém resultados de análise e, ao abri-lo, tais resultados são imediatamente exibidos na tela.

4.1.3 – Abrir arquivo neutro

Arquivos neutros possuem duas extensões: .pos e .nf. Abrir qualquer um destes, faz com que o Sigma2D gere um arquivo binário (.p2b), que tem as funcionalidades descritas no item anterior. Portanto, caso o usuário decida abrir um arquivo neutro em uma pasta que já exista um arquivo binário, o Sigma2D o avisará da existência de tal arquivo e pedirá confirmação da ação (Figura 127).

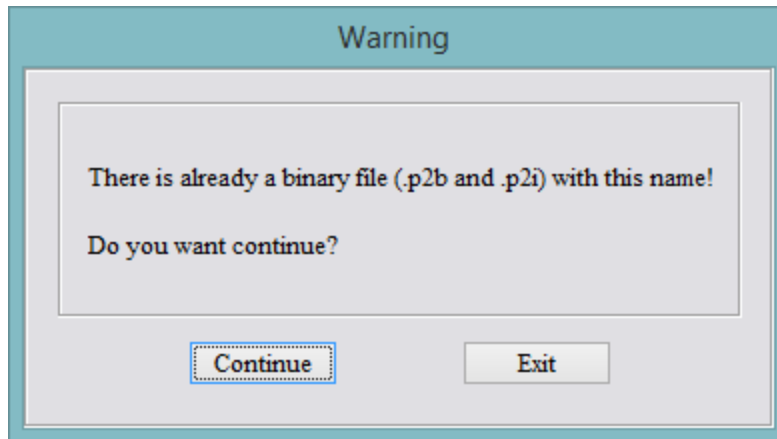


Figura 127 - Geração de arquivo binário a partir de arquivo neutro.

4.2 – Importação

A seguir são descritas as funcionalidades disponíveis de importação de diferentes tipos de arquivos, ilustradas pela Figura 128 abaixo:

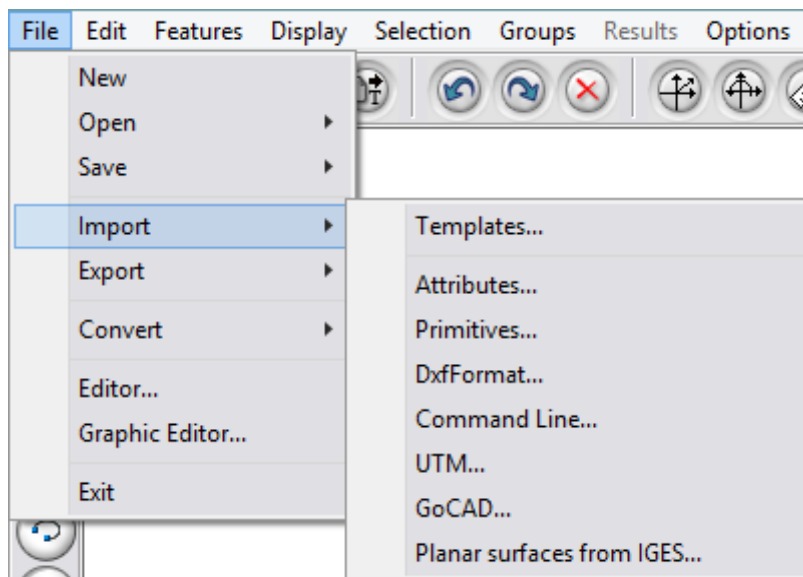


Figura 128 – Importações que podem ser feitas pelo Sigma2D

4.2.1 – Templates

O Sigma2D possui templates para geração de modelos parametrizados, que têm como finalidade a redução do tempo gasto pelo usuário na criação dos mesmos. Os modelos de templates disponíveis pelo Sigma2D são os seguintes:



Figura 129 – Templates disponíveis no Sigma2D

Para utilizar os templates, deve-se clicar no item Templates do menu da Figura 128, fazendo com que a Figura 129 apareça. Em seguida, o usuário será guiado através da interface para introduzir os dados de entrada que serão lidos pelo programa para gerar o modelo desejado. A figura a seguir mostra a janela de fornecimento de dados para criação de um modelo de poço com caverna (axissimétrico).

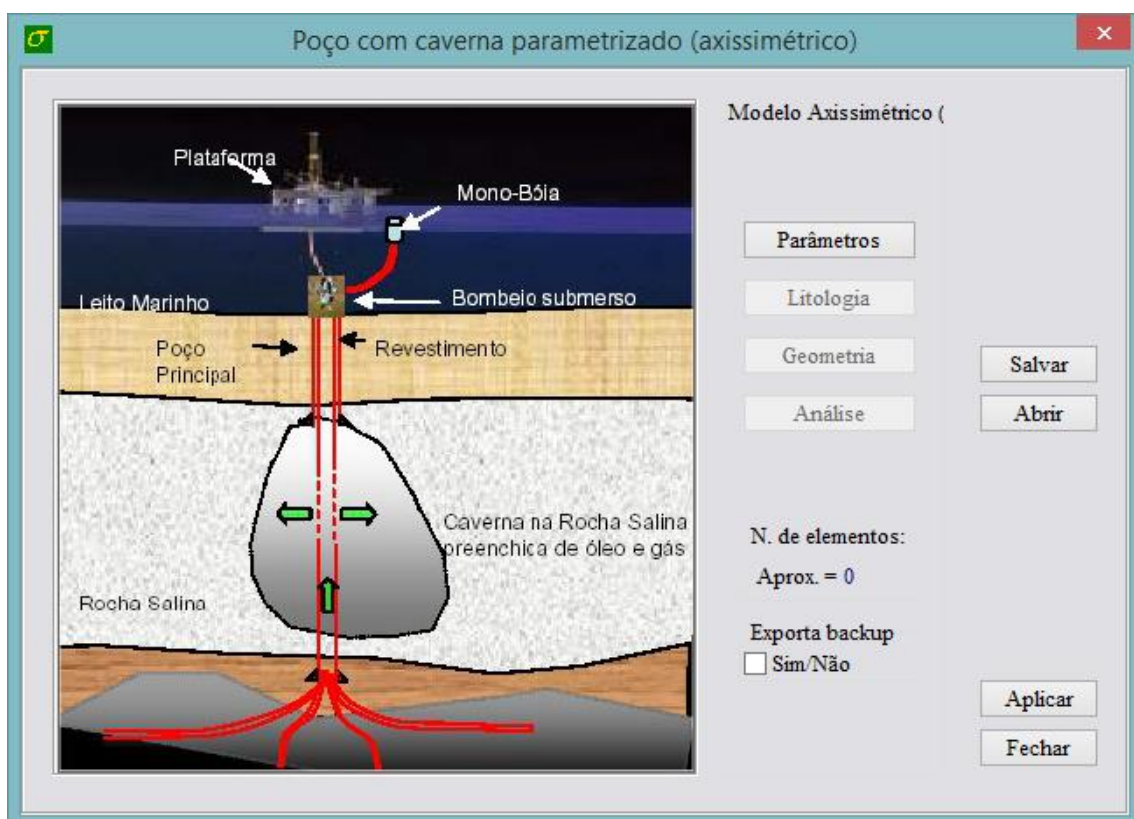


Figura 130 – Diálogo do template de poço com caverna axissimétrico.

Alternativamente, os diálogos de template podem ser acessados através do botão de importação de templates, presente na interface do Sigma2D, mostrado a seguir:



Figura 131 - Botão de importação de templates do Sigma2D

Os diálogos fornecem valores default, que podem e devem ser alterados pelo usuário. Para fins de visualização, o resultado da utilização dos valores default na criação de um modelo de poço com caverna axissimétrico é disposto a seguir:

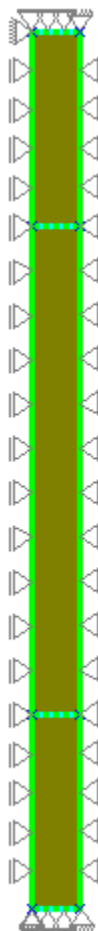


Figura 132 - Modelo de poço com caverna (axissimétrico) com valores default.

4.2.2 – Atributos

A importação de atributos disponibiliza para uma nova modelagem todos os tipos de atributos (carregamentos, materiais, funções, ...) já criados em modelagens anteriores. O Sigma2D tem uma extensão específica para arquivos deste tipo, a .ATT. A importação deve ser feita clicando no item Attributes... da Figura 128. Após a importação, os atributos estarão automaticamente disponíveis para uso no gerenciador de atributos do Sigma2D.

4.2.3 – Primitivas

É possível importar primitivas geométricas (vértices, curvas e regiões) já criadas em modelagens anteriores. Observe que importar para a tela corrente entidades já criadas

anteriormente pode causar conflitos entre as entidades já existentes na tela e as importadas. Por isso, no momento da importação, o usuário deve impor a tolerância geométrica desejada, ou manter a atual. Logo em seguida, as entidades importadas são desenhadas no canvas. A extensão adotada pelo Sigma2D para arquivos de primitivas geométricas é a .PRI. Para importar primitivas geométricas, basta clicar no item Primitivas... da Figura 128.

4.2.4 – Arquivo dxf

Arquivos com extensão .DXF (Drawing Exchange Format) são arquivos gerados por modelos de CAD, como, por exemplo, o AutoCad. O arquivo principal do AutoCad tem extensão .DWG, porém, alternativamente, é possível salvar arquivos com extensão .DXF. O Sigma2D lê esse arquivo e desenha na tela o modelo geométrico. A seguir, observa-se um modelo de placa simples na interface do AutoCad e, posteriormente, na interface do Sigma2D:

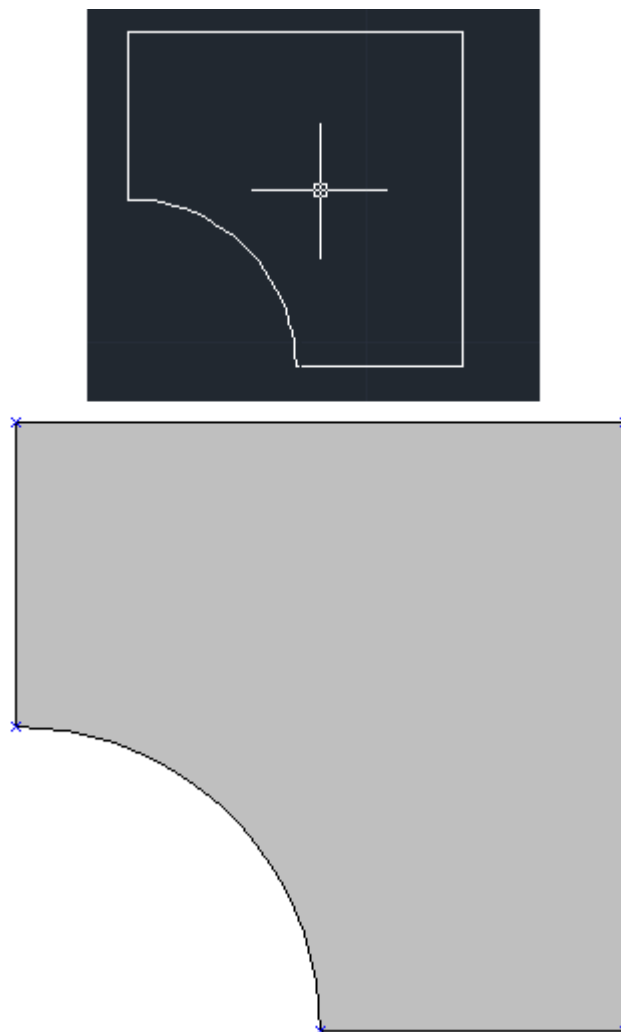


Figura 133 - Modelo de placa importado através de arquivo .DXF.

É interessante notar que as coordenadas dos elementos são mantidas, logo, é preciso ter cautela ao importar um arquivo .DXF se deseja-se utilizar elementos deste arquivo juntamente com elementos já existentes em uma modelagem corrente.

A importação de arquivos .DXF deve ser feita clicando-se no ítem DxfFormat do menu da Figura 128.

4.2.5 – Arquivo de linha de comando

Como já foi dito, o arquivo de backup salva os comandos executados pelo usuário ao longo da modelagem, seja através da interface ou através da linha de comando do programa.

Os comandos são gravados no arquivo de backup exatamente da forma como aparecem no **TH**. Cada comando, por conseguinte, ocupa uma única linha do arquivo.

O arquivo de backup recebe o mesmo nome do arquivo corrente, porém sua extensão é do tipo **cml**. Após o término da modelagem, o usuário pode renomear esse arquivo, se desejar.

O usuário pode editar um arquivo de backup existente, alterando valores de parâmetros, ordem dos comandos, inserindo ou retirando comandos. Um arquivo de backup totalmente novo também pode ser editado para ser interpretado pelo programa. Isso permite ao usuário criar um modelo completo apenas editando um arquivo do tipo **cml**, seguindo as regras vistas anteriormente para cada tipo de comando.

São permitidas linhas em branco entre duas linhas contendo comandos consecutivos, bem como a inserção de comentários ao longo do arquivo. Os comentários podem ser inseridos iniciando uma linha com dois caracteres do tipo asterisco (**). Cada linha contendo um comentário deve conter esses dois caracteres em seu início.

Não é permitido quebrar um comando em duas ou mais linhas. Um mesmo comando deve conter o seu LABEL e os seus parâmetros numa mesma linha do arquivo, seguindo a metodologia adotada para a modelagem via linha de comando.

Os comandos que caracterizam início e fim de bloco de comandos em linguagem LUA (BEGINBLOCK ou BB / ENDBLOCK ou EB) devem estar isolados nas respectivas linhas do arquivo onde se localizam. Os comandos em linguagem LUA devem ser digitados em linhas inseridas entre as que contêm esses dois comandos.

Exemplo:

```
linha 1      BEGINBLOCK
linhas 2...n-1 (comandos em LUA)
linha n      ENDBLOCK
```

Um arquivo de backup pode ser importado através do item “Command Line...” do submenu “Import” do menu “File”.

A interpretação de comandos importados de um arquivo de backup é feita de forma exatamente idêntica à interpretação de comandos via linha de comando. Ou seja, cada linha do arquivo contendo um comando é interpretada, primeiramente verificando-se o LABEL da tarefa, e, caso este seja validado, interpretando-se a seguir os seus parâmetros (exceto para o caso de blocos de comandos em LUA, que será tratado no próximo capítulo). Desta forma, o usuário pode usar os LABELS completos ou simplificados dos comandos, omitir *labels* de parâmetros que não sejam necessários, etc.

4.2.6 – Arquivo UTM

A importação de arquivos UTM pode ser feita clicando-se no ítem UTM... do menu da Figura 128.

4.2.7 – Arquivo GoCAD

O Sigma2D também lê arquivos gerados pelo software GOCAD. A importação segue a mesma idéia descrita no ítem 4.2.4, porém, destaca-se a necessidade de que o arquivo GOCAD esteja acompanhado de um arquivo .PL, que contém informações de linhas de inflexão que determinam as interseções dos painéis. A importação é feita clicando-se no ítem GoCAD... do menu da Figura 128.

4.2.8 – Arquivo IGES

Arquivos IGES (Initial Graphics Exchange Specification) são arquivos neutros padronizados internacionalmente, que levam informações de modelos 2D e 3D para diferentes softwares CAD. Mais detalhes e especificações sobre este tipo de arquivo podem ser encontrados na literatura. A importação deve ser feita clicando-se no ítem Planar surfaces from IGES... do menu da Figura 128

4.3 – Exportação

A exportação de arquivos segue, basicamente, os mesmo princípios descritos no ítem 4.2. A diferença é que para que os arquivos sejam importados, é necessário que, em algum momento anterior, eles tenham sido exportados. Este ítem descreve a exportação dos arquivos disponíveis para serem exportados, como ilustra a figura a seguir:

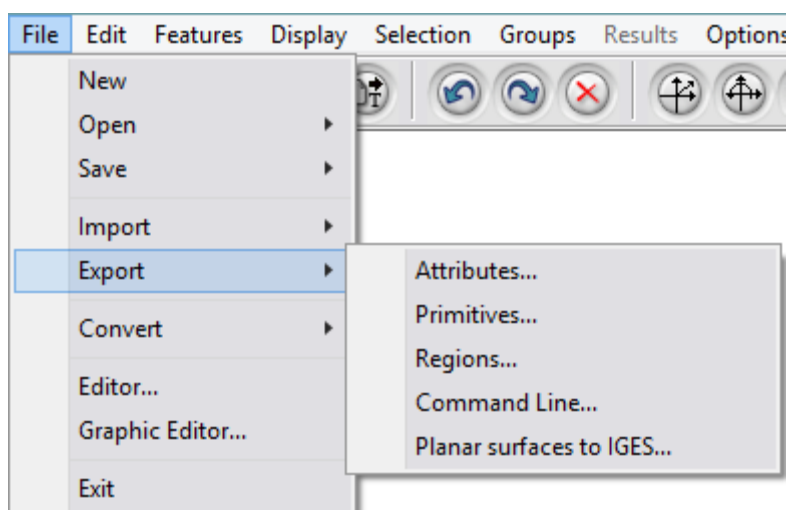


Figura 134 – Arquivos que podem ser exportados pelo Sigma2D.

4.3.1 – Atributos

Ao fim de uma modelagem, pode-se exportar os atributos criados para posterior reutilização, como descrito no ítem 4.2.2. Para isto, basta clicar no ítem Attributes... do menu da Figura 134.

4.3.2 – Primitivas

Ao fim de uma modelagem, pode-se exportar as primitivas geométricas criadas para posterior reutilização, como descrito no ítem 4.2.3. Para isto, basta clicar no ítem Primitives... do menu da Figura 134.

4.3.3 – Regiões

A exportação de regiões é feita para arquivos com extensão .REG, que podem ser editados posteriormente de forma manual, se necessário. A exportação pode ser feita clicando-se no ítem Regions... do menu da Figura 134.

4.3.4 – Arquivo de linha de comando

A exportação de arquivos de linha de comando segue todo o modelo descrito no ítem 4.2.5, e deve ser feita clicando-se no ítem Command Line... da Figura 134.

4.3.5 – Arquivo IGES

O Sigma2D também é capaz de exportar arquivos IGES, descritos no ítem 4.2.8. Para fazer esta exportação, basta clicar no ítem Planar surfaces to IGES... do menu da Figura 134.

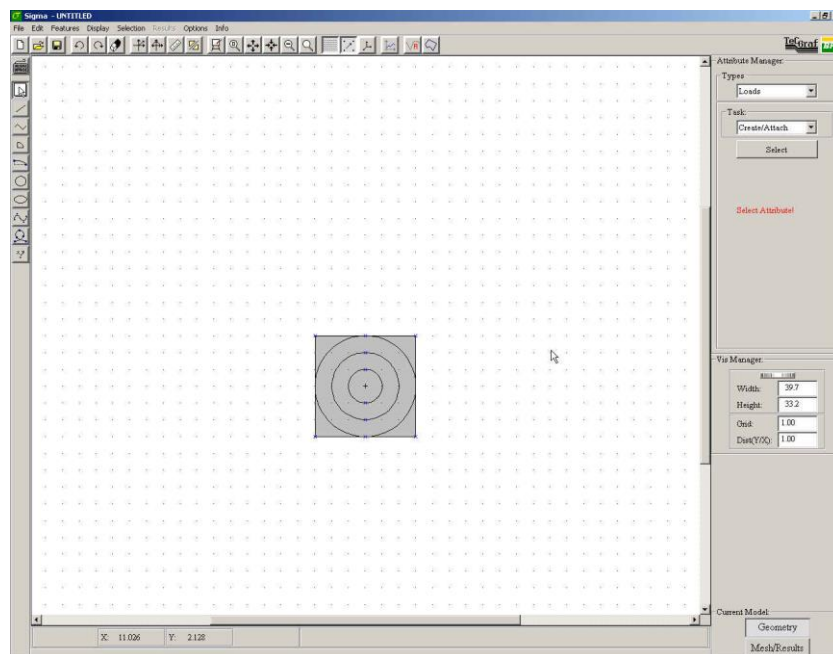
5 – PARAMETRIZAÇÃO

Parametrizar, no presente contexto, significa tornar algo dependente de parâmetros, ou seja, fazer com que o resultado de um procedimento dependa dos valores assumidos por certas variáveis que são necessárias para a completa definição do problema.

Todos os comandos que foram apresentados no último capítulo dependem do valor de parâmetros. Dependendo dos valores que esses parâmetros assumirem, podem-se criar modelos completamente distintos.

Uma dos problemas que surge com frequência no desenvolvimento de modelos complexos, em qualquer ambiente de modelagem, é a necessidade de se criar modelos semelhantes a outros já existentes sem que se precise efetuar todos ou grande parte dos passos de modelagem novamente.

Modelos topologicamente idênticos, por exemplo, contudo geometricamente distintos, não deveriam exigir do usuário esforço dobrado para serem gerados.



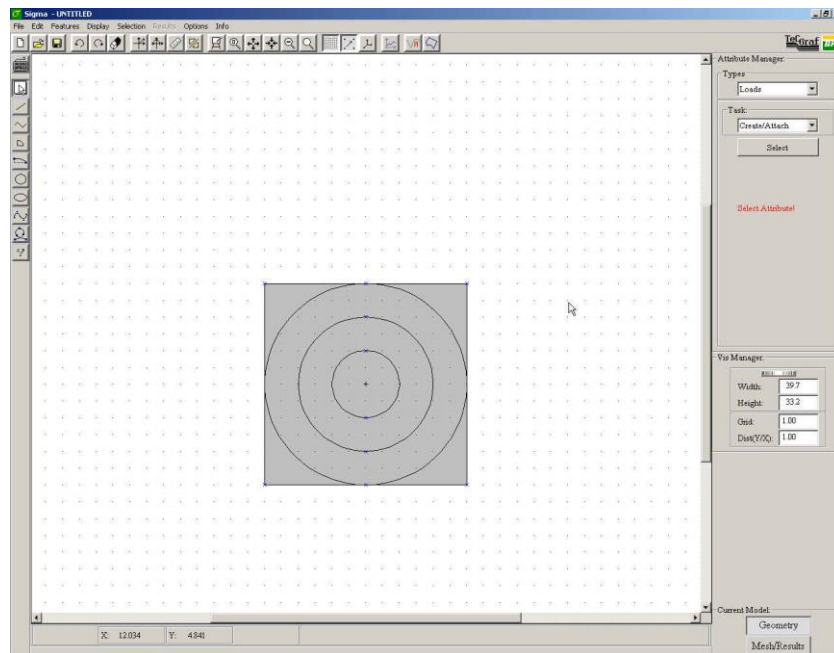


Figura 135 - Dois modelos topologicamente idênticos, mas geometricamente distintos.

Outro exemplo seria o de um modelo com diversos atributos de um ou mais tipos criados com determinados valores para os seus parâmetros, valores estes determinados por meio de equações. Pode-se desejar criar um modelo geométrico idêntico, contudo definindo valores diferentes para os parâmetros dos atributos. Sem a ferramenta de parametrização, ao usuário só restaria a opção de renomear o modelo corrente, recalcular os valores dos parâmetros baseando-se nas equações que os definem e modificar esses valores um a um, para conseguir o modelo desejado.

A parametrização funciona, então, como uma generalização, permitindo que o usuário possa tornar os valores de certos parâmetros dependentes de variáveis a serem definidas.

Ex: O modelo da Figura 135, bem simples, foi gerado criando-se 3 círculos concêntricos (centro em (0,0, 0,0)), e em seguida um quadrado circunscrito ao círculo de maior raio. Este quadrado foi gerado criando-se quatro segmentos de reta de mesmo tamanho, dois horizontais e dois verticais, que tangenciam o círculo de maior raio. No primeiro caso, os círculos têm raio igual a 1, 2 e 3, e cada segmento de reta tem tamanho igual a 6. No segundo caso, os círculos têm raio 2, 4 e 6, e cada segmento de reta tem tamanho igual a 12. Poder-se-ia generalizar esses dois modelos da seguinte forma, por exemplo:

Primeiro círculo: centro = (cx, cy), raio = r
 Segundo círculo: centro = (cx, cy), raio = 2*r
 Terceiro círculo: centro = (cx, cy), raio = 3*r
 $p1 = (cx - r, cy - r)$
 $p2 = (cx + r, cy - r)$
 $p3 = (cx + r, cy + r)$
 $p4 = (cx - r, cy + r)$
 Primeiro segmento: $p1 \rightarrow p2$
 Segundo segmento: $p2 \rightarrow p3$
 Terceiro segmento: $p3 \rightarrow p4$

Quarto segmento: $p4 \rightarrow p1$

Para os dois modelos, bastaria definir $cx = cy = 0.0$. Para o primeiro modelo, $r = 1$. Para o segundo modelo, $r = 2$.

5.1 – Parametrização no SIGMA

No programa SIGMA, a parametrização é feita através da declaração e definição de variáveis e funções em linguagem LUA.

Como foi descrito no capítulo anterior, o usuário possui duas formas de criar comandos em LUA: via linha de comando do programa (durante a modelagem – parametrização DINÂMICA) ou através da edição de um arquivo de backup criado por ele ou gerado pelo programa (antes ou após a modelagem – parametrização ESTÁTICA).

Independentemente da forma que o usuário escolher para criar os comandos em LUA que deseja, deve-se sempre ter em mente que quaisquer comandos desse tipo devem estar agrupados entre os comandos de inicialização de bloco de comandos em LUA (BEGINBLOCK ou BB) e de finalização de bloco de comandos em LUA (ENDBLOCK ou EB).

A vantagem da parametrização dinâmica é a possibilidade de o usuário decidir quais são as variáveis do modelo que deseja parametrizar DURANTE o processo de modelagem. Obviamente, se um comando qualquer é digitado de forma que seus parâmetros sejam dependentes de variáveis declaradas e definidas anteriormente, uma posterior mudança no valor dessas variáveis não alterará o resultado prévio da tarefa concluída pela execução do comando.

Novamente tome-se como exemplo os modelos da Figura 135 e a parametrização sugerida para esses modelos. Se, no início da modelagem, o parâmetro “r” fosse definido como sendo igual a 1, o modelo gerado seria o primeiro (supondo $cx = cy = 0.0$). Se, após a criação dos círculos e dos segmentos de reta com base no parâmetro “r”, o valor deste fosse alterado para 2, por exemplo, o modelo já existente não iria ser modificado. No entanto, qualquer comando que utilizasse o parâmetro “r” após essa modificação, estaria dependente do seu novo valor.

Nesse exemplo, se o usuário quisesse gerar o segundo modelo da figura, ele deveria modificar o valor de “r” no arquivo de backup gerado pelo programa para o primeiro modelo. Poderia também salvar o modelo com outro nome (para não perder o primeiro modelo), alterar o valor de “r” dinamicamente, remover todas as entidades presentes no modelo e repetir todos os comandos efetuados relativos à criação das entidades, simplesmente copiando e colando a seqüência de comandos adequada guardada no **TH**.

Deve-se ter muito cuidado quando um modelo geométrico é gerado de tal forma que as entidades sejam criadas com base em parâmetros. Se o objetivo do usuário é gerar modelos semelhantes posteriormente, apenas alterando os valores dos parâmetros, ele deve ter em mente que qualquer alteração na topologia do modelo irá comprometer o modelo como um todo. Não só um modelo diferente será gerado, como a seqüência de

comandos que eventualmente será lida no arquivo de backup pode desviar o rumo da modelagem.

Tome-se como exemplo o modelo da Figura 136. Foi criado um segmento de reta unindo-se os pontos $(-7.0, 4.0)$ e $(7.0, 4.0)$. Em seguida foi criado um círculo com centro em $(0.0, 0.0)$, cujo raio é igual ao valor de um parâmetro pré-definido “r”. No modelo dessa figura, definiu-se “r” como sendo igual a 3. Foram então selecionados as curvas “c2” e “c3” (curvas que delimitam o círculo) e aplicada uma subdivisão dos mesmos. Eles foram divididos em 10 partes iguais (SDVNUM 10, 1.0). Em seguida selecionou-se a face “f2” (círculo) e aplicou-se uma malha triangular com elementos lineares pelo processo “Boundary Contraction” (TRIAGULATION 1, 0).

Supondo-se que o usuário salve este modelo, edite o arquivo de backup e altere o valor do parâmetro “r” para 5. Nesse caso, a topologia do modelo estaria sendo modificada. Isso porque o segmento de reta, que no modelo anterior nem tocava o círculo, passaria a ser secante ao mesmo. Quando a seqüência de comandos do arquivo de backup fosse lida e interpretada pelo programa, o modelo gerado seria aquele da Figura 137. As duas interseções que não existiam antes e que passaram a existir fariam não só com que a geometria e a topologia do modelo mudassem, mas também com que as entidades mudassem de nome. Os segmentos “c2” e “c3”, nesse caso, passaram a ser aqueles que estão com as subdivisões mostradas na figura. Quando a malha de triângulos fosse aplicada à face “f2”, o resultado seria esse que pode ser visualizado na figura.

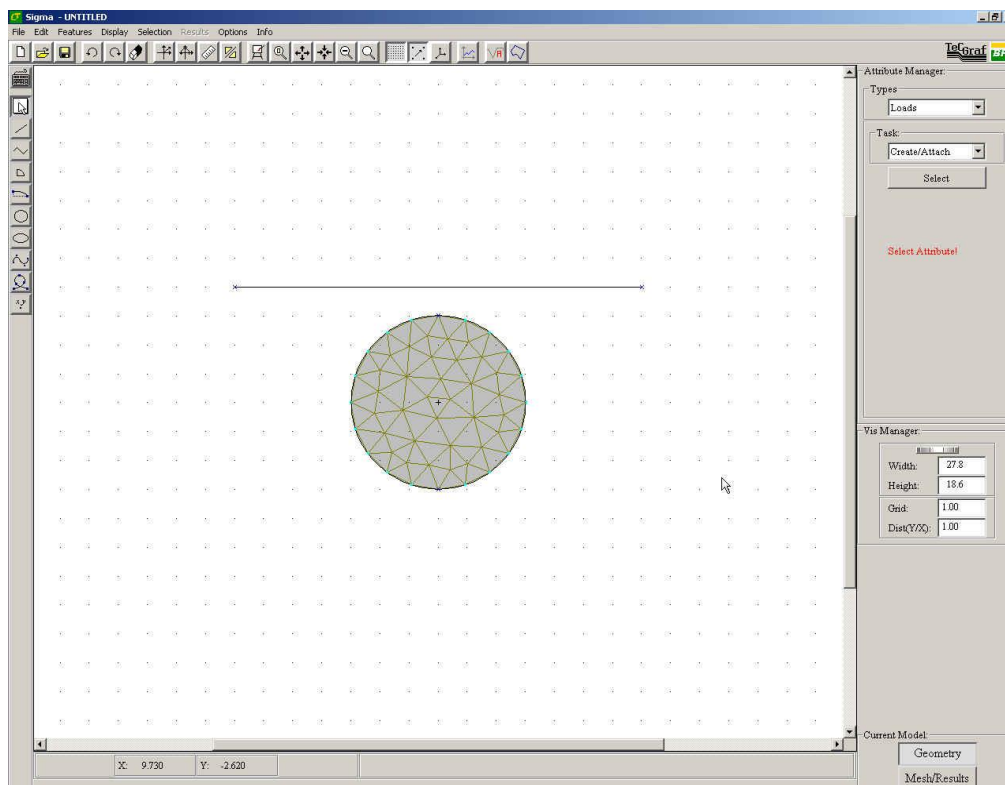


Figura 136 - Modelo gerado usando-se parâmetro.

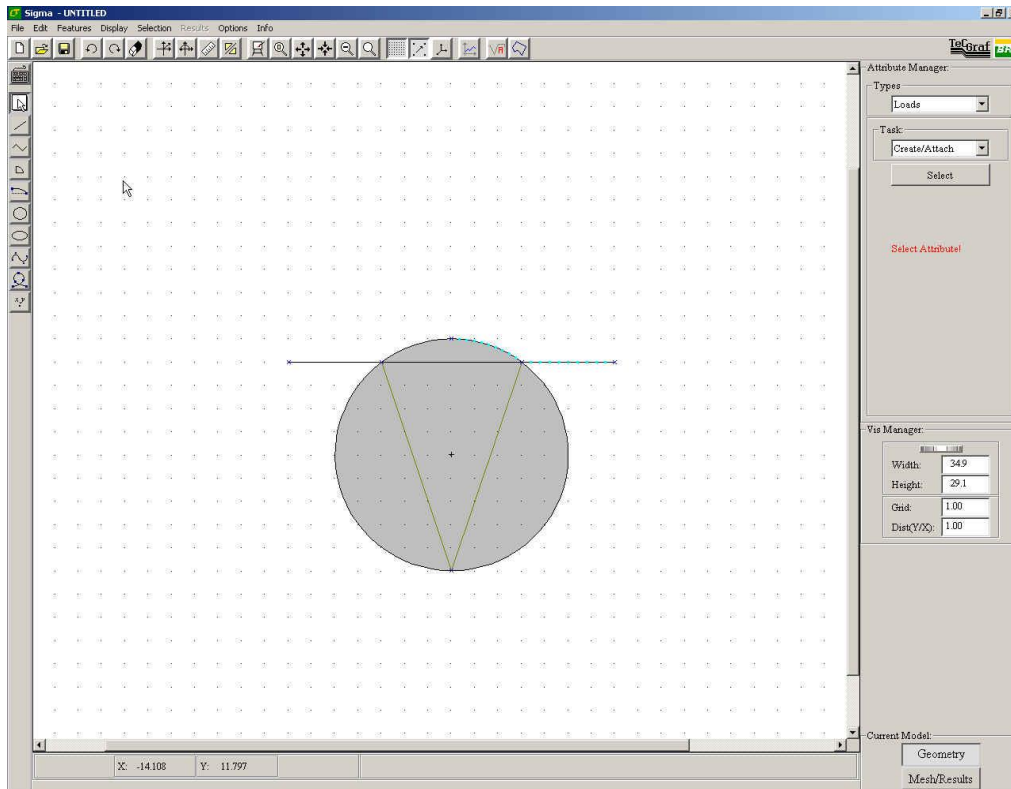


Figura 137 - Alteração errada do valor do parâmetro ocasionando um modelo diferente do anterior.

5.1.1 – Comandos em linguagem LUA

A linguagem LUA foi escolhida para servir como suporte para a parametrização no programa SIGMA pelos seguintes motivos:

- Trata-se de uma linguagem extremamente simples e didática para ser aprendida
- Não requer que o usuário lide com alocação e liberação de memória, nem tenha de conhecer os tipos de variáveis aceitos pelo compilador
- A manipulação de *strings* (cadeias de caracteres) é muito fácil e simples
- Possui uma enorme flexibilidade em relação à criação e manipulação de tabelas, vetores e matrizes
- Possui todos os comandos básicos de qualquer linguagem de programação (laços, condicionais, funções, declaração e definição de variáveis)

A sintaxe completa da linguagem e todos os detalhes inerentes a ela podem ser vislumbrados no manual e no livro disponíveis online no site www.lua.org. Atenta-se para o fato de que a versão da linguagem LUA utilizada no programa SIGMA é a versão 3.2.

A seguir são apresentados os comandos mais básicos da linguagem, que devem ser os mais utilizados pelos usuários do SIGMA:

- Declaração e definição de variáveis

Em LUA, as variáveis não precisam ser declaradas. Elas podem ser definidas a qualquer momento. Não há também a necessidade de se especificar o tipo de variável que está sendo definida. Essa identificação é feita internamente. Basicamente, para os fins a que essa ferramenta se destina no presente caso, apenas quatro tipos de variáveis são necessárias: “number” (número – pode ser inteiro ou real), “string” (caracter ou cadeia de caracteres), “function” (funções) e “table” (tabela). As variáveis do tipo “function” serão tratadas posteriormente.

As variáveis não possuem tipos fixos. Se, por exemplo, uma variável foi definida inicialmente atribuindo-se um valor numérico à mesma, não significa que posteriormente não se possa atribuir uma *string* a essa mesma variável.

O sinal de igual (=) atribui o valor à sua direita à variável localizada à sua esquerda. Pode-se dar um nome qualquer, formado por letras, dígitos e traços baixos (*underscore* → `_`) às variáveis. Não são válidos nomes começados por dígitos. Devem-se evitar nomes começados por traços baixos, bem como nomes com todas as letras maiúsculas, pois podem ser nomes de variáveis internas do programa SIGMA. Nomes iniciados pela palavra “sigma” ou pela palavra “mtool” também devem ser evitados.

Várias variáveis podem ser definidas simultaneamente. Isso pode ser feito de duas formas:

- a) Se todas as variáveis irão receber o mesmo valor, podem-se escrever os nomes de todas elas intercalados por sinais de igual e, após o nome da última variável, um sinal de igual seguido do valor que se quer atribuir.
- b) Se as variáveis irão receber valores distintos, podem-se escrever seus nomes intercalados por vírgulas e, após o nome da última variável, os valores que se deseja atribuir a cada uma delas, respectivamente, separados por vírgulas.

A linguagem LUA incorpora uma maneira prática de se lidar com variáveis cujo valor deve ser diferente de qualquer outro valor. Trata-se de um valor simbólico, desprezível, que não será usado em nenhum tipo de operação e serve apenas para indicar que a variável ainda não foi definida. A este valor associa-se o nome **nil**.

Variáveis do tipo “table” podem ser vetores, matrizes, registros, conjuntos, etc. Elas podem ser indexadas não somente com números, mas com quaisquer valores (exceto **nil**). LUA usa o nome do campo como índice. A linguagem permite que se indexe uma variável desse tipo de duas formas: *var.fieldname* ou *var[“fieldname”]*. Tabelas precisam ser criadas antes de terem seus campos definidos. A criação de uma tabela é feita atribuindo-se um sinal de “abre chaves” e outro de “fecha chaves” ({ }) ao nome da tabela.

Exemplos:

a = 1

b = "teste"

a = b = c = d = 3.1

a, b, c, d = 2.3, -0.45, "sigma", nil

t = { }

t[1] = 1.2

t["material"] = "mat1"

t.color = "white"

t.newt = { }

t.newt.value = 8

t.newt[3] = 12

- Operações com variáveis numéricas

Variáveis numéricas podem combinar-se através de operações matemáticas. São válidas as seguintes operações (serão listadas as mais usuais, para a lista completa ver manual online):

+ - * / ^

e (potência de 10)

log log10

sin cos tan asin acos atan

atan2 (com dois parâmetros)

max min sqrt deg rad

mod (resto da divisão inteira)

abs (valor absoluto – módulo)

floor (truncar)

Exemplos:

a = 2.0

b = 3.0

c = 2 * a - 4 * b / 3 + 5 -- o valor atribuído a **c** será 5

d = atan2(1, 0) -- o valor atribuído a **d** será 90 (sempre em graus)

f = floor(4.3) -- o valor atribuído a **f** será 4

- Expressões

Expressões em LUA podem ser avaliadas com o auxílio dos seguintes sinais:

> >= < <=

== (igual)

~= (diferente)

not (negação)

and (e)

or (ou)

O valor de retorno de uma expressão pode ser "1" se a expressão for verdadeira ou "nil" se ela for falsa (cuidado, **nil** não é a mesma coisa que 0!).

Exemplos:

a = 1.0

```
b = 1.2
a < b -- verdadeiro
a + b == 2.2 -- verdadeiro
a ~= b -- verdadeiro
a - b > a + b -- falso
a and b -- verdadeiro
a and nil -- falso
a or nil -- verdadeiro
(not a) or (not b) -- falso
```

- Laços

Existem duas formas de se fazer um laço em LUA:

```
while (expression) do (block) end
repeat (block) until (expression)
```

As duas formas são semelhantes. A única diferença é que, no primeiro caso, a expressão é avaliada ANTES da primeira iteração. No segundo caso, a expressão é avaliada APÓS a primeira iteração.

O uso de parênteses para confinar a expressão que deve ser avaliada é facultativo.

Exemplos:

```
i = 1
a = 2.0
while i < 3 do
  a = a ^ 2
  i = i + 1
end
```

```
a = 1
b = 30
repeat
  a = a * 3
until (a > b)
```

- Condicional

Uma condicional é uma maneira de se avaliar o valor de uma expressão e definir o que deve ser feito dependendo do valor que essa expressão assumir.

Em LUA, isso pode ser feito da seguinte forma:

```
if (expression1) then (block1) elseif (expression2) then (block2) elseif (expression3)
then (block3) ... else (blockn) end
```

O uso dos parênteses para confinar as expressões é facultativo.

Exemplos:

```
a = 3
```

```
b = 4
if a < b then
  a = b
elseif a == b then
  a = -b
else
  a = b + 10
end
```

- Funções

Funções são procedimentos que podem receber e retornar um ou mais valores. Uma função em LUA é chamada através do comando **function**.

Se o usuário deseja que a função receba valores como parâmetro, basta enumerar o conjunto de valores que devem ser passados para a função numa ordem determinada, separados por vírgulas e entre parênteses, na chamada da função. Se o usuário deseja que a função retorne um ou mais valores, basta usar o comando **return** e em seguida enumerar os valores que devem ser retornados, separados por vírgulas.

Na definição de uma função, deve-se usar o comando **function** seguido do nome que o usuário deseja dar à função. Em seguida, deve-se abrir parênteses, enumerar nomes para as variáveis locais que receberão os valores que serão passados como parâmetros quando a função for chamada, e fechar parênteses. Se a função não necessita de parâmetros, basta abrir e fechar parênteses. Após esse procedimento, deve-se digitar o corpo da função e, quando terminar, deve-se usar o comando **end**.

Eventuais variáveis locais que precisem ser definidas no corpo da função para auxiliarem nas operações realizadas pela mesma devem ser explicitamente declaradas como locais. Uma variável pode ser declarada como local usando-se o comando **local** antes do nome da variável. Isso é necessário pois a linguagem LUA considera como variáveis globais todas as aquelas que não são declaradas como locais. Isso pode ser perigoso caso o usuário defina uma variável auxiliar no corpo de uma função com o mesmo nome de uma variável global definida anteriormente. Basicamente, para os fins a que essa ferramenta se destina no programa SIGMA, devem ser globais todas as variáveis que representarem parâmetros que podem ter seus valores alterados posteriormente (durante ou após a modelagem, via arquivo de backup).

Se, por exemplo, uma função retorna 2 valores e, no entanto, três variáveis estiverem recebendo os valores de retorno da função, a terceira variável receberá o valor **nil**.

Exemplos:

```
function dobro(value)
  return value * 2
end
a = 3.0
b = dobro(a) -- será atribuído o valor 6.0 a b
```

```
function posicao(s0, v0, t, a)
  local pos -- essa variável só existirá dentro do corpo dessa função
```

```

pos = s0 + v0*t + 0.5 * a * t ^ 2
return pos
end

```

```

function cinematic(s0, v0, t, a)
local pos
local vel
pos = s0 + v0*t + 0.5 * a * t ^ 2
vel = v0 + a * t
return pos, vel
end

```

```

function fatorial(n)
if n < 0 then return nil
elseif n == 0 or n == 1 then return 1
else return n * fatorial(n-1) -- função recursiva
end
end

```

```

function f ( )
if a <= 0 then -- a é uma variável global, nesse caso
return a
else
return -a
end
end

```

- Comentários

Em LUA, comentários podem ser feitos ao longo do código da forma como foi visto nos exemplos anteriores, ou seja, colocando-se dois sinais de menos (--) antes de cada comentário. Um comentário está limitado a uma única linha de código. Se o usuário deseja fazer comentários em várias linhas consecutivas, deve usar o sinal "--" em cada linha.

- Manipulação de *strings*

Strings, ou cadeias de caracteres, são muito simples de serem manipuladas em LUA. Uma *string* é sempre representada entre aspas, mesmo que contenha um único caracter. Algumas funções úteis na manipulação de *strings* em LUA são:

1) **strlen(str)**

Retorna um valor numérico inteiro que representa o tamanho da cadeia *str*, ou seja, o número de caracteres que ela contém.

Ex: `a = strlen("SIGMA")` -- *a* recebe o valor 5

2) **strsub(str, i, j)**

Retorna uma sub-cadeia da cadeia *str*, que se inicia no caracter *i* e vai até o caracter *j*. Se *i* ou *j* forem negativos, eles são substituídos pelo tamanho da *string* menos o valor absoluto de *i* ou *j* mais um. Ou seja, é como se a *string* fosse lida ao contrário, com os valores absolutos de *i* ou *j*.

Ex:

```
str1 = "Linguagem LUA"
str2 = strsub(str1, 4, -2)
-- O valor de str2 será "guagem LU"
```

3) **strlower(str)**

Retorna uma *string* igual à *str*, porém com todos os caracteres convertidos para minúsculas.

Ex: str1 = strlower("SIGMA") -- *str1* receberá "sigma"

4) **strupper(str)**

Retorna uma *string* igual à *str*, porém com todos os caracteres convertidos para maiúsculas.

Ex: str1 = strupper("sigma") -- *str1* receberá "SIGMA"

5) Duas ou mais *strings* podem ser concatenadas usando o comando `..`.

Ex:

```
str1 = "Programa"
str2 = "SIGMA"
str = str1.. " " ..str2 -- str receberá "Programa SIGMA"
```

6) **tonumber(str)**

Converte uma *string* em um número. Se a *string str* representar um número, retorna o valor desse número. Se a *string str* não for um número, retorna **nil**.

7) **tostring(num)**

Converte um número em uma *string*. Retorna uma *string* contendo o número *num*.

- Tratamento de erros

Se eventualmente o usuário digitar algum comando inválido, após a finalização do bloco de comandos em LUA (através do comando ENDBLOCK ou EB), será disparada uma mensagem de erro que será exibida no **TH**, como já foi visto.

```
*** Invalid block of LUA commands! ***
```

5.1.2 – Chamadas de comandos para execução de tarefas de dentro de um bloco de comandos

Em alguns casos, pode ser necessário chamar o comando responsável pela execução de uma tarefa de dentro de um bloco de comandos em LUA. Isso não pode ser feito simplesmente digitando-se a sintaxe própria do comando (LABEL + parâmetros), como é feito normalmente na **CMD**. Isso porque o que é escrito dentro de um bloco de comandos em LUA é interpretado pela linguagem. Como a sintaxe dos comandos apresentados foi criada independentemente da linguagem LUA, de forma a ser o mais didática possível para o usuário, deve haver alguma maneira de indicar para o

interpretador da linguagem que o que será escrito a seguir é um código a ser interpretado pelo programa SIGMA. Ou seja, um comando escrito dentro de um bloco de comandos em LUA funciona exatamente como um bloco que deve ser interpretado por outra fonte, exatamente como é feito com o próprio bloco escrito em LUA na **CMD**. Seria um bloco dentro de outro bloco.

Para manter a simplicidade e a didática adotadas até agora, foram criadas duas palavras-chave que indicam o início e o fim de um comando: **tcall** e **tend**. A letra “t”, antes das palavras “call” e “end”, representa a inicial palavra “task” (tarefa). Logo, *tcall* e *tend* significariam “*task call*” (chamada de uma tarefa) e “*task end*” (finalização da tarefa). Vale ressaltar que apenas UMA tarefa deve ser chamada entre essas duas palavras. Ou seja, para que várias tarefas sejam chamadas, deve-se usar esse par de palavras tantas vezes quantas forem as tarefas, delimitando o início e o fim de cada uma delas.

Exemplo:

```
BEGINBLOCK
i = 1
while i <= 5 do
  tcall line 0, i, i + 1, 0 tend
  matname = "mat"..tostring(i)
  tcall mat matname tend
  i = i+ 1
end
ENDBLOCK
```

5.1.3 – Uso de funções e variáveis definidas em LUA pelo usuário fora do bloco de comandos

Se o usuário definiu funções ou variáveis dentro de um bloco de comandos em LUA e deseja utilizá-las como parâmetros de uma tarefa digitada fora do bloco de comandos em LUA, basta proceder normalmente, digitando o nome da variável ou da função (com os devidos parâmetros) no lugar dos parâmetros da tarefa que deseja parametrizar.

Ex:

```
BEGINBLOCK
r = 2.0
cx = 0.0
cy = 0.0
function max_x (cx, r)
  return cx+r
end
function min_x (cx, r)
  return cx-r
end
ENDBLOCK
circle cx, cy, r
line max_x(cx, cy, r), -10, max_x(cx, cy, r), 10
```

line min_x(cx, cy, r), -10, min_x(cx, cy, r), 10

5.1.4 – Atalhos para *strings* comumente usadas

Existem algumas palavras-chave que serão constantemente usadas em alguns comandos que exigem que *strings* sejam passadas como parâmetros. Por esse motivo, para evitar o uso das aspas toda vez que uma dessas palavras tiver de ser digitada, foram criadas variáveis internas em LUA que podem ser usadas no lugar dessas palavras. Basicamente, essas variáveis em LUA possuem um nome muito semelhante ao da palavra que estão substituindo. Usando-se o nome dessas variáveis, torna-se desnecessário o uso das aspas.

Switch de toggles:

on = "ON"
off = "OFF"
yes = "YES"
no = "NO"

Valores genéricos para indicar quantidades de atributos:

all = "ALL"
none = "NONE"

Nomes de atributos:

force = "Force"
temperature = "Temperature"
poropressure = "Poropressure"
displacement = "Displacement"
initialtemperature = "Initial Temperature"
initialporopressure = "Initial Poropressure"
pressure = "Pressure"
heatflux = "Heat Flux"
massflux = "Mass Flux"
pipepressure = "Pipe Pressure"
distributedload = "Distributed Load"
inertial = "Inertial"
geostatic = "Geostatic"
initialstress = "Initial Stress"
boundarypressure = "Boundary Pressure"
rezone = "Rezone"
reconstruct = "Reconstruct"
constraint = "Constraint"
material = "Material"
interfacematerial = "InterfaceMaterial"
properties = "Properties"
interfaceproperties = "Interface Properties"

timefunction = “Time Functions”
spatialfunction = “Spatial Functions”

Valores de parâmetros de materiais sólidos:

elastic = “ELASTIC”
elastoplastic = “ELASTO-PLASTIC”
viscoelasto = “VISCO-ELASTO”

mohrcouldrained = “MOHR COULOMB DRAINED”
mohrcoulundrained = “MOHR COULOMB UNDRAINED”
mohrcoulmod = “MOHR COULOMB MOD.”
vonmisesideal = “VON MISES IDEAL”
vonmisesisotropic = “VON MISES ISOTROPIC”

white = “WHITE”
gray = “GRAY”
red = “RED”
softblue = “SOFTBLUE”
blue = “BLUE”
darkblue = “DARKBLUE”
darkgreen = “DARKGREEN”
green = “GREEN”
softgreen = “SOFTGREEN”
yellow = “YELLOW”
softorange = “SOFTORANGE”
orange = “ORANGE”
darkorange = “DARKORANGE”
softpurple = “SOFTPURPLE”
purple = “PURPLE”
darkpurple = “DARKPURPLE”
purplebrown = “PURPLEBROWN”
darkbrown = “DARKBROWN”
brown = “BROWN”
softbrown = “SOFTBROWN”
yellowbrown = “YELLOWBROWN”
greenbrown = “GREENBROWN”
anidrita = “ANIDRITA”
vulcano = “VULCANO”
siltito = “SILTITO”
peloide = “PELOIDE”
oopel = “OOPEL”
margafol = “MARGAFOL”
intra = “INTRA”
igneas = “IGNEAS”
folhelho = “FOLHELHO”
diabasio = “DIABASIO”
calcário = “CALCARIO”
basalto = “BASALTO”
areia = “AREIA”

pelintra = "PELINTRA"

Valores de parâmetros de materiais de interface:

hiperbolic = "HIPERBOLIC"

exponential = "EXPONENTIAL"

linear = "LINEAR"

perfectplastic = "PERFECT PLASTIC"

multilinear = "MULTILINEAR"

kinematinghardening = "KINEMATING HARDENING"

tension = "TENSION"

geostatic = "GEOSTATIC"

mohrcoulomb = "MOHR COULOMB"

jaeger = "JAEGER"

Valores de parâmetros de propriedades de sólidos:

reduced = "REDUCED"

complete = "COMPLETE"

axissymmetric = "AXISSIMETRIC"

planestress = "PLANE STRESS"

planestrain = "PLANE STRAIN"

Valores de parâmetros de restrições:

constx = "Const_x"

consty = "Const_y"

constxy = "Const_xy"

Parâmetros de subdivisão de linhas:

geomline = "GEOMLINE"

infbound = "INFBOUND"

interfelem = "INTERFELEM"

Cores do Modelo:

grid = "Grid"

regionmeshes = "Region Meshes"

axis = "Axis"

gausspoints = "Gauss Points"

submiddle = "submiddle"

tensorsigma3 = "Tensor Sigma 3"

findelement = "Find Element"

edgesdv = "Edge Sdv"

infiniteelm = "Infinite Elm"

meshboundary = "Mesh Boundary"
selectcolors = "Select Colors"
nodenumbers = "Node Numbers"
defbar = "Def Bar"
vertexs = "Vertexs"
fractureelm = "Fracture Elm"
nodes = "Nodes"
interfaceelm = "Interface Elm"
undeformedmesh = "Undeformed Mesh"
element = "Element"
regions = "Regions"
mesh = "Mesh"
listof = "List of"
tensorsigma1 = "Tensor Sigma 1"
elementnumbers = "Element Numbers"
background = "Background"
edges = "Edges"
findnode = "Find Node"

6 – Considerações Gerais

Além deste manual de consulta, o Sigma2D conta com algumas ferramentas para auxílio ao usuário, seja de dúvidas pontuais ou mesmo na criação de um modelo e geração de resultados para análise.

Tais ferramentas se encontram no item Info da barra de menus, mostrado a seguir:

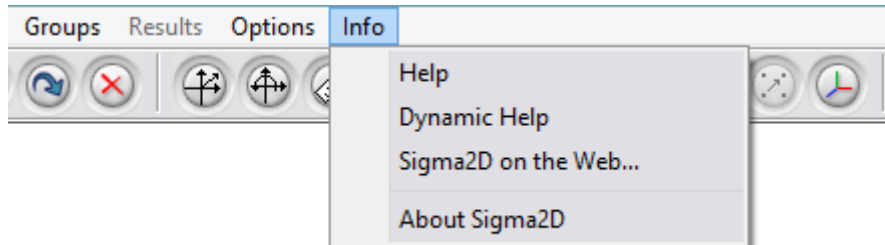


Figura 138 – Items do menu Info.

6.1 - Help

As descrições sobre funcionalidades feitas neste manual foram levadas para arquivos com extensão .html, para que pudessem ser abertos através da interface do Sigma2D. Isto pode ser feito através do item Help da Figura 138, que abrirá um link no navegador do usuário contendo diversas informações. Estas informações estão divididas em seções, como pode ser visto na figura abaixo, para que o usuário possa navegar com facilidade e buscar com eficiência a informação desejada.

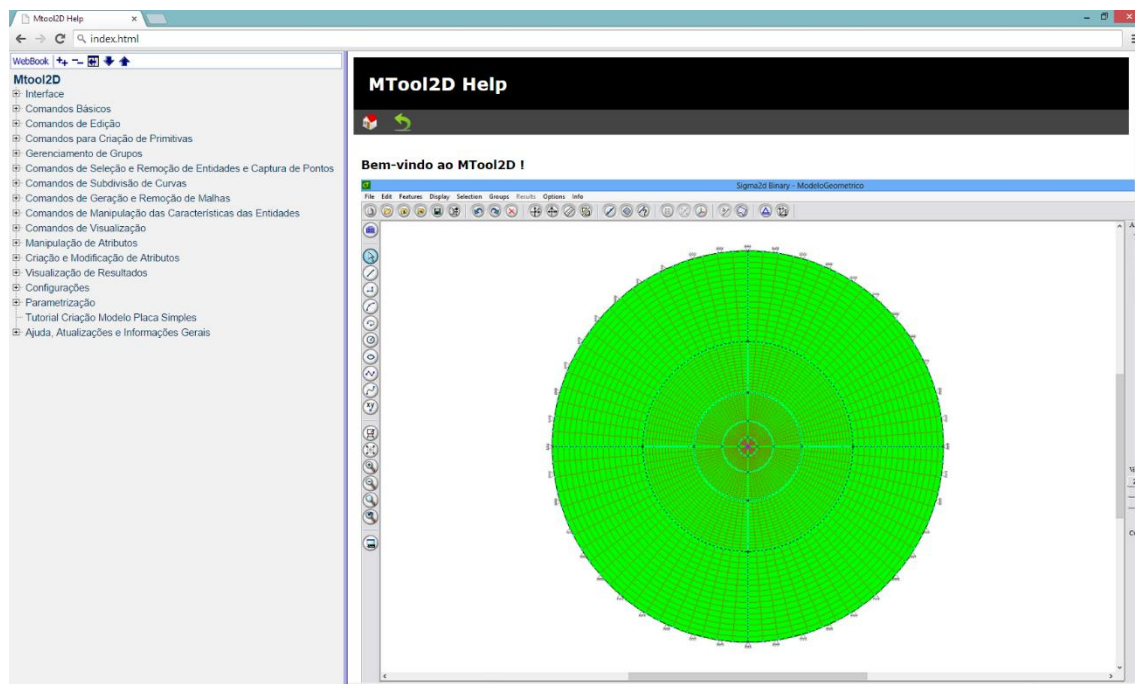


Figura 139 – Ajuda “estática” do Sigma2D.

6.2 - Dynamic Help

Alternativamente, o usuário pode utilizar a funcionalidade do Dynamic Help, que cria um diálogo na interface do Sigma2D. Este diálogo é atualizado de acordo com o elemento da interface sobre o qual o mouse é posicionado (sem clicar). O diálogo apresentará a descrição de uso e da funcionalidade desejada, sendo automaticamente atualizado caso o mouse mude de posição e passe por outro elemento da interface. A imagem a seguir ilustra melhor o funcionamento, com o mouse posicionado sobre o botão Delete:

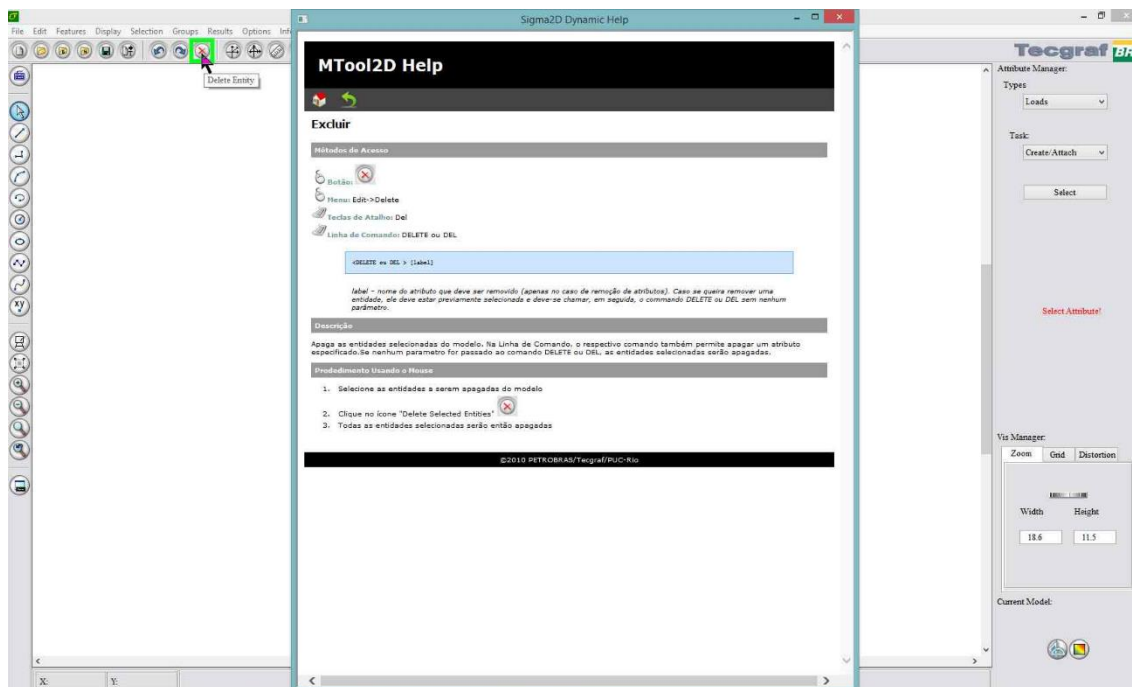


Figura 140 – Ajuda Dinâmica do Sigma2D.

Ítems de menu também funcionam desta mesma forma, porém os menus precisam ser abertos antes. A janela da Ajuda Dinâmica sempre se mantém fixa à frente da janela principal do programa. Apesar disso, botões e ítems de menu podem ser clicados mesmo quando a janela da Ajuda Dinâmica estiver aberta. Isso permite ao usuário ficar recebendo dicas/informações sobre as ferramentas do programa enquanto utiliza o próprio programa. Além disso, a janela da Ajuda Dinâmica pode ser redimensionada ou reposicionada conforme o usuário queira, para facilitar a exploração da interface gráfica do programa. Ela também pode ser minimizada e restaurada. Para ler melhor o texto da Ajuda Dinâmica, sua janela também pode ser maximizada.

Os itens carregados no diálogo do Dynamic Help são os mesmo arquivos existentes no Help “estático”, a diferença é que a busca fica concentrada na interface do Sigma2D e evita que o usuário tenha que fazer qualquer tipo de pesquisa, agilizando o processo de aprendizagem.

6.3 - Sigma2D na Internet

Encontra-se também no menu info, através do ítem Sigma2D on the Web..., um link para a página online do grupo MVGEO, onde usuários cadastrados poderão fazer download das últimas versões e atualizações do Sigma2D.

6.4 - Informações

Por fim, o ítem About Sigma2D traz ao usuário informações sobre a versão e revisão do Sigma2D instalado em sua máquina, como mostra o diálogo da Figura 141. Na aba Mesh deste mesmo diálogo, são dispostas informações sobre a malha de elementos finitos (caso haja uma) criada na modelagem. Observando que essas informações só serão geradas, e consequentemente dispostas ao usuário, após a conversão do modelo geométrico para malha de elementos finitos, através do botão Mesh Model & Results.

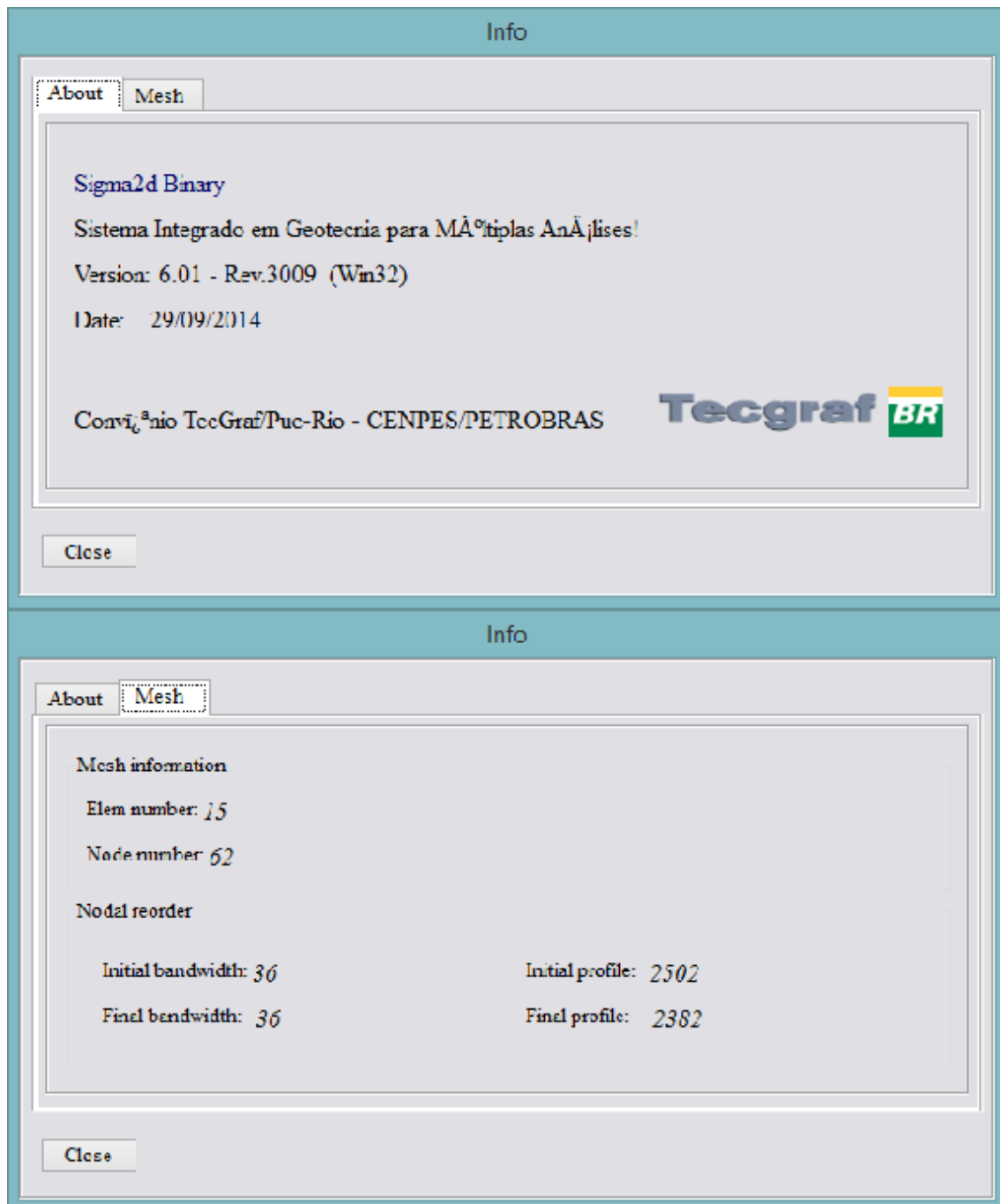


Figura 141 – Diálogo de informações do Sigma2D.