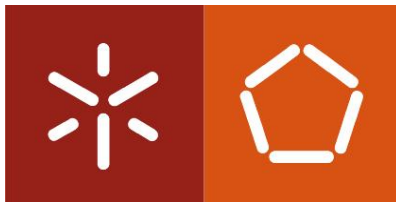


**Universidade do Minho**  
Escola de Engenharia

Carlos Filipe Carvalho Veloso nº48778

**Consolidação de solos recorrendo a  
ferramentas computacionais**





**Universidade do Minho**  
Escola de Engenharia

Carlos Filipe Carvalho Veloso nº48778

**Consolidação de solos recorrendo a  
ferramentas computacionais**

Dissertação de Mestrado  
Mestrado Integrado em Engenharia Civil

Trabalho efetuado sob a orientação de  
Professor Doutor Francisco Martins  
Professor Doutor Rui Pereira



## **AGRADECIMENTOS**

Ao Professor Doutor Francisco Martins pela ajuda no significado das principais variáveis contidas no código do programa de cálculo automático existente na Universidade do Minho, pela ajuda em todas as dúvidas que foram surgindo, pela disponibilidade constante quer pessoalmente quer por email e pelas sugestões dadas durante a elaboração desta dissertação.

Ao Professor Doutor Rui Pereira pela ajuda nos aspetos básicos da linguagem Fortran em que foi escrito o código do programa, pela ajuda nas dúvidas que foram surgindo durante a criação das interfaces em MatLab, pela disponibilidade constante quer pessoalmente quer por email e pelas sugestões dadas durante a elaboração desta dissertação.

À minha mãe, irmãos, avó e familiares mais próximos pelo carinho, apoio e incentivo demonstrados ao longo de toda a minha vida.

Aos meus amigos por todos bons momentos que passei com eles.



## **RESUMO**

Nesta dissertação foi realizado o melhoramento de um programa de cálculo automático sobre consolidação, existente na Universidade do Minho, tornando-o mais amigável para o utilizador, principalmente para fins pedagógicos.

Para esse efeito, foram criadas duas interfaces gráficas em MatLab, sendo uma delas para a introdução de dados e a outra para a saída de resultados. A primeira permite quer uma introdução básica de dados, onde muitos dos parâmetros de entrada são assumidos por defeito, quer uma introdução completa de dados, onde o utilizador terá de introduzir todos os dados. A segunda permite que os resultados obtidos possam ser visualizados através de gráficos, sendo muitos deles apresentados de uma forma dinâmica, com os parâmetros de saída a variar ao longo do tempo. Com os melhoramentos efetuados passaram a existir múltiplas opções relacionadas com a saída dos resultados.

Nesta dissertação são ainda apresentados dois exemplos de aplicação, sendo um de consolidação unidimensional e outro de consolidação bidimensional. Nesses exemplos são mostrados os valores usados na interface de introdução de dados bem como vários menus com opções de saída dos resultados.

**Palavras-chave:** programa de cálculo automático, consolidação, pré-processamento, pós-processamento, interfaces gráficas.





## **ABSTRACT**

In this dissertation it was performed the improvement of an automatic calculation program about consolidation, that was developed many years ago at University of Minho, making it more friendly to the user, mainly for pedagogical purposes.

For this purpose, two graphical interfaces were created in MatLab, one being for data input and another for output of results. The first allows both a basic introduction of data, where many of the input parameters are assumed by default, and a complete introduction of data where the user has to enter all the data. The second allows that the results obtained may be displayed using graphics, many of which are presented in a dynamic manner, with the output parameters varying over time. With the new improvements there are multiple options related to the results output.

This dissertation also presents two application examples, one about one-dimensional consolidation and other about two-dimensional consolidation. Several menus used in data input and multiple output options of the results are shown in these examples.

**Keywords:** automatic calculation program, consolidation, preprocessing, postprocessing, graphical interfaces.



# ÍNDICE

1. INTRODUÇÃO.....	1
1.1. Generalidades .....	1
1.2. Objetivos.....	1
1.3. Organização e conteúdo .....	2
2. ESTADO DE ARTE RELACIONADO COM A CONSOLIDAÇÃO .....	3
2.1. Considerações gerais .....	3
2.2. Teorias existentes .....	4
2.2.1. Teoria unidimensional de Terzaghi .....	4
2.2.2. Teoria de Terzaghi-Rendulic .....	5
2.2.3. Teoria de Biot .....	6
2.3. Soluções analíticas recentes e métodos numéricos.....	6
3. PROGRAMA DE CÁLCULO AUTOMÁTICO EXISTENTE NA UNIVERSIDADE DO MINHO.....	11
3.1. Formulação .....	11
3.2. Descrição sucinta do programa.....	12
3.3. Validação do programa.....	16
4. DESENVOLVIMENTO DE INTERFACES PARA O PROGRAMA.....	19
4.1. Interface de introdução de dados .....	20
4.1.1. Interface de introdução de dados básica .....	20
4.1.2. Interface de introdução de dados avançada .....	30
4.2. Interface de saída de resultados .....	38
5. EXEMPLOS DE APLICAÇÃO .....	43
5.1. Exemplo de consolidação unidimensional .....	43
5.2. Exemplo de consolidação bidimensional .....	56
6. CONCLUSÕES E ORIENTAÇÕES PARA FUTUROS DESENVOLVIMENTOS .....	75
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	79
ANEXOS .....	A1
A1. CÓDIGOS EM MATLAB DAS INTERFACES .....	A3
A1.1. Interface de introdução de dados .....	A4
A1.1.1. Interface de introdução de dados básica .....	A4

A1.1.2. Interface de introdução de dados avançada.....	A24
A1.2. Interface de saída de resultados.....	A53

## ÍNDICE DE FIGURAS

Figura 1: Diagrama de sequência do programa CONSOL.....	15
Figura 2: Malha de elementos finitos do exemplo usado (Martins, 1993).....	16
Figura 3: Gráfico variação de $r_p/p$ com o fator tempo $T'$ num ponto situado no eixo de simetria ( $x/a=0m$ ) e $y/a=1m$ . ....	17
Figura 4: Gráfico variação de $r_p/p$ com $x/a$ em pontos situados numa linha correspondente a $y/a=0,5m$ para $T'=0,214$ .....	18
Figura 5: Gráfico variação de $r_p/p$ com $x/a$ em pontos situados numa linha correspondente a $y/a=0,5m$ para $T'=0,857$ .....	18
Figura 6: Menu principal do programa.....	19
Figura 7: Primeiros parâmetros da interface de introdução de dados básica.....	20
Figura 8: Malha usada no caso da consolidação unidimensional.....	21
Figura 9: Malha usada no caso da consolidação bidimensional.....	21
Figura 10: Parâmetros de controlo da interface de introdução de dados básica.....	22
Figura 11: Parâmetros dos elementos da interface de introdução de dados básica. ....	23
Figura 12: Parâmetros dos nós restringidos.....	24
Figura 13: Parâmetros dos nós onde se prescreve a pressão neutra. ....	24
Figura 14: Outros parâmetros da interface de introdução de dados básica. ....	25
Figura 15: Parâmetros dos materiais. ....	26
Figura 16: Parâmetros do tempo da interface de introdução de dados básica. ....	26
Figura 17: Parâmetros das cargas da interface de introdução de dados básica. ....	27
Figura 18: Parâmetros das forças de gravidade.....	28
Figura 19: Parâmetros da tensão inicial.....	28
Figura 20: Parâmetros do coeficiente de impulso em repouso.....	29
Figura 21: Parâmetros das cargas distribuídas da interface de introdução de dados básica.....	30
Figura 22: Parâmetros de controlo da interface de introdução de dados avançada. ....	31
Figura 23: Parâmetros dos elementos da interface de introdução de dados avançada. ....	32
Figura 24: Parâmetros das coordenadas dos nós da interface de introdução de dados avançada. ....	32
Figura 25: Outros parâmetros da interface de introdução de dados avançada. ....	33
Figura 26: Parâmetros do tempo e tensões da interface de introdução de dados avançada. ....	34

Figura 27: Parâmetros das cargas da interface de introdução de dados avançada.....	35
Figura 28: Parâmetros das cargas pontuais da interface de introdução de dados avançada. ...	35
Figura 29: Parâmetros das cargas distribuídas da interface de introdução de dados avançada. .....	37
Figura 30: Representação do problema em análise (exemplo de consolidação unidimensional). .....	45
Figura 31: Representação da malha com a numeração dos nós (exemplo de consolidação unidimensional).....	45
Figura 32: Coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo (exemplo de consolidação unidimensional).....	46
Figura 33: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão vertical efetiva e total (exemplo de consolidação unidimensional). .	47
Figura 34: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total nos nós pertencentes à linha horizontal (exemplo de consolidação unidimensional).....	48
Figura 35: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva nos nós pertencentes à linha horizontal (exemplo de consolidação unidimensional).....	48
Figura 36: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento vertical (exemplo de consolidação unidimensional).....	49
Figura 37: Exemplo da opção "Malha (deslocamento total)" (exemplo de consolidação unidimensional).....	50
Figura 38: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento total (exemplo de consolidação unidimensional).....	51
Figura 39: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação unidimensional). ....	51
Figura 40: Exemplo dos resultados da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação unidimensional).....	52
Figura 41: Exemplo da opção "Deslocamento vertical vs tempo" (exemplo de consolidação unidimensional).....	53
Figura 42: Exemplo da opção "Deslocamento vertical vs profundidade" (exemplo de consolidação unidimensional).....	54
Figura 43: Exemplo da opção "Pressão neutra vs tempo" (exemplo de consolidação unidimensional).....	54

Figura 44: Exemplo da opção "Pressão neutra vs profundidade" (exemplo de consolidação unidimensional). .....	55
Figura 45: Representação do problema em análise (exemplo de consolidação bidimensional). .....	58
Figura 46: Representação da malha com a numeração dos nós (exemplo de consolidação bidimensional). .....	58
Figura 47: Coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo (exemplo de consolidação bidimensional). .....	59
Figura 48: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão vertical efetiva e total (exemplo de consolidação bidimensional). ....	60
Figura 49: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão horizontal efetiva e total (exemplo de consolidação bidimensional). .....	60
Figura 50: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento vertical nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional). .....	61
Figura 51: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional). .....	62
Figura 52: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento vertical em todos os nós (exemplo de consolidação bidimensional). .....	62
Figura 53: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total em todos os nós (exemplo de consolidação bidimensional). .....	63
Figura 54: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional). .....	63
Figura 55: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva em todos os nós (exemplo de consolidação bidimensional). .....	64
Figura 56: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento vertical (exemplo de consolidação bidimensional). .....	64
Figura 57: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento horizontal (exemplo de consolidação bidimensional). .....	65
Figura 58: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão vertical efetiva (exemplo de consolidação bidimensional). .....	65

Figura 59: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão horizontal efetiva (exemplo de consolidação bidimensional).....	66
Figura 60: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão de corte ( $\tau_{xy}$ ) (exemplo de consolidação bidimensional).....	66
Figura 61: Exemplo da opção "Deslocamento total e altura da coluna de água" (exemplo de consolidação bidimensional).....	67
Figura 62: Exemplo da opção "Malha (deslocamento total)" (exemplo de consolidação bidimensional).....	68
Figura 63: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento vertical (exemplo de consolidação bidimensional).....	68
Figura 64: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento horizontal (exemplo de consolidação bidimensional).....	69
Figura 65: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento total (exemplo de consolidação bidimensional).....	69
Figura 66: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação bidimensional).....	70
Figura 67: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão horizontal efetiva (exemplo de consolidação bidimensional).....	70
Figura 68: Exemplo dos resultados da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação bidimensional).....	71
Figura 69: Exemplo da opção "Deslocamento vertical vs tempo" (exemplo de consolidação bidimensional).....	72
Figura 70: Exemplo da opção "Deslocamento vertical vs distância ao eixo y" (exemplo de consolidação bidimensional).....	73
Figura 71: Exemplo da opção "Pressão neutra vs tempo" (exemplo de consolidação bidimensional).....	73
Figura 72: Exemplo da opção "Pressão neutra vs distância ao eixo y" (exemplo de consolidação bidimensional).....	74



## SIMBOLOGIA E ABREVIATURAS

a	Largura ou metade da largura da faixa da carga distribuída
AALFA	Valor de alfa, $\alpha$
ALOAD1	Subrotina principal onde é feita a montagem do vetor solicitação e dos caudais
ASCOU1	Subrotina auxiliar onde é feita a montagem das matrizes de acoplamento
ASEST1	Subrotina auxiliar onde é feita a montagem das matrizes de rigidez
ASFLO1	Subrotina auxiliar onde é feita a montagem das matrizes de fluxo
B	Matriz de deformação-deslocamento
BBANDA	Subrotina principal onde é resolvido o sistema de equações
BMATPS	Subrotina auxiliar que calcula a matriz de deformação-deslocamento [B]
BPMTPS	Subrotina auxiliar que calcula a matriz [Bp]
BVMTPS	Subrotina auxiliar que calcula a matriz [Bv]
C	Matriz de acoplamento
CHECK1	Subrotina auxiliar destinada a detetar os erros resultantes de uma má definição dos dados
CHECK2	Subrotina auxiliar destinada a detetar os erros resultantes de uma má definição dos dados
CONSOL	Programa de cálculo automático existente na Universidade do Minho e nomenclatura dada à rotina principal do programa
COORD	Coordenadas x e y dos nós IPOIN
COUPPS	Subrotina principal que calcula as matrizes de acoplamento dos elementos
D	Matriz de elasticidade
DBE	Subrotina auxiliar que efetua o produto da matriz [D] com a matriz [B]
DIRVEC	Subrotina principal onde é construído o vetor do lado direito
E	Módulo de elasticidade
ELFLOW	Subrotina principal que calcula o vetor de escoamento de cada elemento
FLOWPS	Subrotina principal que calcula as matrizes de fluxo dos elementos
G	Módulo de elasticidade transversal
GAMAW	Peso específico da água, $\gamma_w$ em $\text{kN/m}^3$

GAUSSQ	Subrotina principal que estabelece as posições dos pontos de Gauss e os fatores de peso a usar na integração numérica
GRAVY	Constante de gravidade definida como um múltiplo da aceleração gravitacional, g
H	Matriz de fluxo
HALIN	Subrotina principal onde é feito o produto da matriz [H] por alfa
IEDGE	Parâmetro para indicação da existência de cargas distribuídas
IFPRE	Condições de restrição do nó restringido no deslocamento segundo os eixos x e y
IGRAV	Parâmetro designativo das forças de gravidade
IGUAL	Subrotina principal onde é feita a colocação da matriz global em GBMAT1
IK	Parâmetro do coeficiente de impulso em repouso
INCRT	Valor do incremento de tempo pretendido, em horas
INIST	Parâmetro para indicar se é pretendido considerar a tensão inicial do solo
INITI1	Subrotina principal onde é feita a inicialização dos deslocamentos e pressões
INPUT	Subrotina principal onde é feita a leitura da maior parte dos dados
IPL0D	Parâmetro para indicação da existência de cargas pontuais
IPOIN	Número dos nós dos vértices dos elementos para serem introduzidas as coordenadas x e y
ISOUT	Parâmetro para indicar se é pretendido o cálculo de tensões no incremento de tempo em causa
JACOB2	Subrotina auxiliar que calcula as coordenadas cartesianas dos pontos de gauss, a matriz jacobiana e o seu determinante, a inversa da matriz jacobiana e as derivadas cartesianas das funções de forma
k	Coeficiente de permeabilidade
K	Matriz de rigidez
K0	Valor do coeficiente de impulso em repouso
K <sub>0</sub>	Coeficiente de impulso em repouso
KBP	Subrotina auxiliar que multiplica a matriz permeabilidade pela matriz [Bp]
LINCA	Parâmetro da pressão prescrita
LN0DS	Número dos nós pertencentes ao elemento
LOADPS	Subrotina principal onde é feita a leitura do tipo de solicitação bem como do seu valor e calculadas, para cada elemento, as forças nodais equivalentes, formando o vetor de cargas aplicadas

LODPT	Número do nó onde é aplicada a carga pontual
MAPFUN	Subrotina auxiliar que calcula as funções de "mapping" e as suas derivadas nos pontos de gauss dos elementos
MATNO	Número da propriedade material do elemento
MBANDA	Subrotina principal onde é calculada a semi-banda
MNODE	Número de nós por elemento
MODPS	Subrotina auxiliar que calcula a matriz de elasticidade [D] para problemas bidimensionais
$m_v$	Coefficiente de compressibilidade volumétrica
NEASS	Número do elemento em relação ao qual o lado carregado está associado
NEDGE	Número de lados dos elementos onde estão aplicadas cargas distribuídas
NELEM	Número de elementos
NGAUS	Número de pontos de Gauss em cada direção
NINCR	Número de incrementos de tempo
NMATS	Número de materiais diferentes
NNODE	Número de nós que pertencem ao elemento
NODEXY	Subrotina auxiliar que calcula as coordenadas dos nós situados a meio dos lados
NOFIX	Número do nó restringido
NOPRF	Número do nó onde se prescreve a pressão neutra
NOPRS	Números dos nós que formam o lado carregado
NPOIN	Número de nós
NPRFX	Número de nós onde se prescreve a pressão neutra
NTYPE	Tipo de problema
NVFIX	Número de nós onde há graus de liberdade restringidos
OUTPU1	Subrotina principal onde é feita a saída dos resultados
PERMB	Valor do coeficiente da permeabilidade $k$ ou da matriz (matriz 2 por 2) do coeficiente de permeabilidade $k$ , em m/h
POINT	Valor das componentes da carga pontual nas direções $x$ e $y$
PREPR	Valor prescrito da pressão neutra
PRESC	Valor prescrito das componentes $x$ e $y$ do deslocamento nodal
PRESS(1)	Valor da componente normal da carga distribuída
PRESS(2)	Valor da componente tangencial da carga distribuída
PROPS1	Valor do módulo de elasticidade, $E$ em $\text{kN/m}^2$

PROPS2	Valor do coeficiente de Poisson, $\nu$
PROPS3	Valor da espessura do material, $t$
PROPS4	Valor da densidade do material, $\rho$ em $\text{kg/m}^3$
RESMO1	Subrotina auxiliar onde são introduzidas as restrições de alisamento
RESMO2	Subrotina auxiliar onde são introduzidas as restrições de alisamento
RESTR1	Subrotina principal onde são colocadas as restrições nos vetores globais
$r_p$	Pressão neutra
$R_p$	Vetor dos fluxos nodais aplicados
RPIM	Método de interpolação de pontos radial
$R_u$	Vetor das forças nodais equivalentes
$r_u$	Vetor dos deslocamentos nodais
SFR2	Subrotina auxiliar que calcula as funções de forma e as suas derivadas nos pontos de gauss dos elementos
SMOOTH	Subrotina principal de alisamento
STIFPS	Subrotina principal que calcula as matrizes de rigidez dos vários elementos
STREPS	Subrotina principal onde é feito o cálculo das tensões em todos os elementos
SURCO	Valor da coordenada $y$ da superfície do terreno
$T'$	Fator tempo
TELEM	Tipo de elementos
THETA	Ângulo entre a direção da gravidade e o eixo $y$ medido a partir deste eixo, no sentido direto ou retrógrado, de forma a obter o menor ângulo
$\alpha$	Parâmetro de interpolação na integração em ordem ao tempo, Alfa
$\gamma_w$	Peso específico da água
$\Delta t$	Incremento de tempo
$\nu$	Coefficiente de Poisson

# **1. INTRODUÇÃO**

## **1.1. Generalidades**

Em várias áreas, assim como em Engenharia Civil, o uso de programas de cálculo automático facilita a análise de problemas complexos. Entre esses problemas inclui-se o fenómeno da consolidação de solos.

Tendo isso em conta, é sugerido como objetivo desta Dissertação o melhoramento de um programa de cálculo automático sobre consolidação que foi desenvolvido no antigo Centro de Informática da Universidade do Minho. O programa existente necessita que seja criado um ficheiro de dados devidamente organizado para que os parâmetros de entrada possam ser lidos convenientemente. Quanto aos resultados, estes são impressos num ficheiro que pode ser aberto com o bloco de notas. Por isso, torna-se difícil usar o programa devido à inexistência de interfaces amigáveis de introdução de dados e de saída de resultados. Daí a necessidade de melhorar o programa através da criação de interfaces de pré-processamento e do pós-processamento com o desenvolvimento de uma componente gráfica atrativa.

No final, espera-se que o programa existente se torne mais amigável para o utilizador.

## **1.2. Objetivos**

Esta Dissertação tem como objetivo global o melhoramento de um programa de cálculo automático existente que foi escrito em Fortran, com a finalidade de o tornar mais amigável para fins pedagógicos. Mais especificamente, pretende-se criar um programa em MatLab para melhorar o pré-processamento e o pós-processamento do programa existente.

O resultado esperado é que o programa existente se torne mais amigável para o utilizador, nomeadamente na simplificação do processo de introdução de dados e na melhoria substancial da observação dos resultados fornecidos pelo programa facilitando a sua interpretação, sendo criadas saídas gráficas para esse efeito.

### **1.3. Organização e conteúdo**

Esta dissertação encontra-se organizada em vários capítulos, cujos conteúdos são a seguir descritos. A seguir a este primeiro capítulo introdutório, no capítulo 2 é feita uma revisão do estado de arte, resultante da pesquisa que foi feita sobre consolidação de solos. Nele são tecidas considerações gerais sobre o fenómeno da consolidação, é feita uma ligeira descrição das teorias existentes e são feitas referências a alguns trabalhos que foram feitos nos últimos anos relacionados com este tema.

O capítulo 3 debruça-se sobre o programa de cálculo automático existente na Universidade do Minho, sendo feita uma apresentação sumária da formulação do problema da consolidação e as equações usada, uma breve descrição do programa usado com a apresentação do respetivo organigrama e a validação do programa.

O capítulo 4 está relacionado com o desenvolvimento das interfaces para o programa de cálculo automático existente na Universidade do Minho, nomeadamente as interfaces de introdução de dados básica e avançada e a interface de saída de resultados, sendo explicado o funcionamento destas.

No capítulo 5 são mostrados dois exemplos de aplicação, sendo um de consolidação unidimensional e outro de consolidação bidimensional, sendo mostrados os valores usados nos parâmetros da interface de introdução de dados básica e os resultados que podem ser vistos na interface de saída de resultados.

No capítulo 6 são apresentadas as conclusões finais e feitas sugestões para futuro desenvolvimento.

Em anexo podem ser vistos os códigos em MatLab das interfaces que foram realizadas para o programa de cálculo automático existente na Universidade do Minho.

## **2. ESTADO DE ARTE RELACIONADO COM A CONSOLIDAÇÃO**

### **2.1. Considerações gerais**

Consolidação é um fenômeno que consiste na redução gradual do índice de vazios do solo por expulsão do fluido intersticial e transferência da pressão do fluido para o esqueleto sólido, a uma velocidade dependente da permeabilidade do solo. A drenagem da água dos poros pode ser originada, por exemplo, pela aplicação de sobrecargas, pelo peso próprio das camadas sobrejacentes, por rebaixamentos de níveis freáticos e por bombagens subterrâneas.

A consolidação pode ser dividida em consolidação primária e consolidação secundária. A consolidação primária é o processo pelo qual se produz assentamento do solo ao longo do tempo, em resultado da dissipação do excesso da pressão intersticial gerado devido ao carregamento. Após a dissipação do excesso da pressão intersticial, verifica-se que o solo continua a deformar-se com uma tensão efetiva constante, correspondendo esta fase à consolidação secundária. Esta fase ocorre devido à existência de excessos de pressão neutra e de gradientes hidráulicos reduzidos.

Apesar do fenômeno de consolidação se tratar de um processo tridimensional, existem casos com interesse prático em que a aplicação de um modelo unidimensional constitui uma simplificação aceitável, como por exemplo, o caso da aplicação de uma sobrecarga uniforme numa superfície com dimensões muito superiores à espessura da camada de solo sujeito à consolidação.

Por outro lado, os modelos bidimensionais são modelos mais complexos do que os modelos unidimensionais mas permitem um estudo mais elaborado dos processos de consolidação.

## 2.2. Teorias existentes

### 2.2.1. Teoria unidimensional de Terzaghi

A primeira Teoria da Consolidação consistente foi desenvolvida por Karl Terzaghi (1943), reconhecido internacionalmente como o fundador da Mecânica dos solos, que estabeleceu o Princípio da Tensão Efetiva que permitiu compreender a transmissão das cargas através do esqueleto sólido de partícula a partícula do solo.

A teoria do Princípio da Tensão Efetiva divide a tensão total em duas parcelas designadas por tensão efetiva e pressão na água dos poros ou pressão neutra. A tensão efetiva corresponde à parte da tensão que é suportada pelo esqueleto formado pelas partículas sólidas do solo, enquanto que a pressão neutra é a parte da tensão suportada pela água do solo. Num solo saturado sujeito a uma solicitação, parte dessa solicitação vai para as partículas do solo e a outra parte vai para a fase líquida do solo. Como a deformabilidade do solo é superior à deformabilidade da água contida nos poros do solo, a solicitação vai ser totalmente suportada pela água no instante de aplicação da carga. À medida que a água vai sendo expulsa do solo, a carga vai sendo transferida da água para as partículas do solo, acabando por ser apenas suportada pela fase sólida do solo após um tempo suficientemente elevado.

Esta teoria de Terzaghi foi desenvolvida considerando o seguinte conjunto de hipóteses básicas (Terzaghi,1943):

- 1) O solo é considerado homogéneo;
- 2) O solo é considerado completamente saturado;
- 3) É desprezável a compressibilidade da água e das partículas do solo;
- 4) Os efeitos, fenómenos e o seu curso, que se verificam em massas de dimensões infinitesimais, são extrapoláveis para massas de dimensões representativas de um maciço real;
- 5) As deformações e o fluxo processam-se só na direção vertical;
- 6) É válida a lei de Darcy para todos os gradientes hidráulicos;
- 7) São considerados constantes os coeficientes de permeabilidade,  $k$ , e de compressibilidade volumétrica,  $m_v$ ;
- 8) Existe uma única relação, independente do tempo, entre o índice de vazios e a tensão efetiva (não existe fluência do esqueleto sólido);



9) As deformações são pequenas;

10) Em qualquer secção horizontal e em qualquer instante o estado de tensão e deformação são uniformes.

Como é perceptível, as hipóteses assumidas por Terzaghi são bastante discutíveis, nomeadamente os pontos 5 a 8. Desde logo porque a consolidação não é um processo unidimensional mas sim tridimensional e para baixos gradientes hidráulicos existem desvios em relação à lei de Darcy. Para além disso, o coeficiente de permeabilidade,  $k$ , decresce à medida que o índice de vazios decresce durante a consolidação e o coeficiente de compressibilidade volumétrica,  $m_v$ , também decresce durante a consolidação uma vez que a relação entre o índice de vazios e a tensão efetiva não é linear. Contudo, para pequenos incrementos de tempo, é razoável considerar a constância dos coeficientes de permeabilidade e de compressibilidade volumétrica. Finalmente, a relação entre o índice de vazios e a tensão efetiva não é independente do tempo.

### **2.2.2. Teoria de Terzaghi-Rendulic**

As hipóteses simplificadas assumidas pela teoria da consolidação unidimensional de Terzaghi, nomeadamente a constância de tensões totais com o tempo e a compressão e fluxo unidimensionais, conduzem a erros nos resultados como seja a sobrestimação do tempo de consolidação (Schiffman et al., 1969) principalmente nos casos em que a argila está sobreconsolidada. Rendulic (1936), reconhecendo a semelhança do fenómeno da consolidação com o fenómeno da condução de calor, desenvolveu uma teoria de consolidação mais evoluída do que a teoria unidimensional de Terzaghi designada de Terzaghi-Rendulic. Esta teoria tem como base o conhecimento da lei de distribuição das tensões totais, em qualquer ponto e instante, originada pela aplicação de uma solicitação exterior. Essa lei é necessária para a resolução da equação diferencial obtida por esta teoria.

Por isso, há quem considere esta teoria como pseudo-tridimensional (Schiffman et al., 1969). Apesar disto, fornece resultados com suficiente aproximação para efeitos de utilização em problemas de engenharia (Correia, 1980).

### **2.2.3. Teoria de Biot**

Tanto a teoria de Terzaghi como a de Rendulic não consideram o processo de deformação acoplado ao processo de dissipação das pressões na água dos poros, tendo apenas como incógnita, nas equações diferenciais, o excesso de pressão na água dos poros. A primeira teoria de consolidação multidimensional que considera a inter-relação entre os fenómenos de deformação do esqueleto sólido e de escoamento na água dos poros foi desenvolvida por Biot (1941).

Esta teoria assume que:

- O solo é um material isotrópico em que existe reversibilidade nas relações tensões-deformações, sendo essas relações lineares;
- As deformações são pequenas;
- A água contida nos poros é incompressível;
- É válida a lei de Darcy para o escoamento da água.

### **2.3. Soluções analíticas recentes e métodos numéricos**

Com o desenvolvimento dos computadores e a evolução das técnicas numéricas tornou-se mais simples a análise de casos mais complicados ligados ao fenómeno da consolidação. Pela facilidade e versatilidade no seu uso, destaca-se o método dos elementos finitos que permite estudar a resposta do solo desde o início do carregamento até ao colapso, podendo ser atribuída ao solo uma dada lei de comportamento.

Tal método implica uma divisão do domínio que se pretende utilizar em subdomínios, designados por elementos, que se ligam entre si em pontos chamados nós. Assim, as soluções são formuladas para cada elemento e a seguir são combinadas para obter a solução para o domínio completo.

No entanto, a utilização de elementos finitos revela limitações, nomeadamente na análise de problemas em meios infinitos ou semi-infinitos, pois impõe a consideração de domínios espacialmente finitos. Para contornar estas limitações são associados aos elementos finitos outros tipos de elementos denominados infinitos.

Apesar de ter uma grande aplicabilidade e ser considerado robusto, o método dos elementos finitos ainda não consegue resolver eficientemente alguns problemas específicos. Tendo em conta essas limitações, têm surgido novos métodos como por exemplo os métodos sem malha. Os métodos sem malha, tal como o método de elementos finitos, são métodos de aproximação numérica para resolução numérica de equações diferenciais. Os métodos sem malha fazem a discretização do domínio por nós colocados arbitrariamente sem interligação entre estes, ou seja, sem a existência explícita de uma malha estruturada de elementos. De seguida serão mencionados, com uma breve descrição, alguns trabalhos acerca da consolidação de solos.

A primeira formulação para a consolidação acoplada foi feita por Sandhu e Wilson (1969) que se basearam no princípio variacional de Gurtin. A seguir, Christian e Boehmer (1970) desenvolveram um procedimento alternativo para o estado plano de deformação. Ghaboussi e Wilson (1973) também formularam o escoamento dum meio poroso e elástico a partir dum princípio variacional do tipo Gurtin mas incluíram nas relações constitutivas a compressibilidade do fluido e das partículas sólidas. Booker e Small (1975) apresentam um funcional que representa o integral de uma função contendo as transformadas de Laplace das deformações, dos excessos de pressões neutras, da deformação volumétrica, das tensões impostas nas fronteiras e das componentes do vetor deslocamento. A minimização desse funcional conduz a uma equação que invertida dá a solução em termos de excessos de pressões neutras e das componentes do vetor deslocamento.

Krause (1978) obteve um conjunto de equações diferenciais lineares para a consolidação baseada no Princípio dos trabalhos virtuais. No mesmo ano, Verruijt (1978) apresenta equações diferenciais para a consolidação de um meio poroso elástico, linear, isotrópico contendo um fluido linearmente compressível, apresentando também um funcional a partir do qual obtém as equações de elementos finitos para o caso do estado plano de deformação. Booker (1973), Kanok-Nukulchai e Suaris (1982) usam uma solução semi-analítica baseada na teoria de Biot em que a discretização espacial do meio poroso elástico linear é feita usando elementos finitos e a dependência do tempo é removida usando a transformação de Laplace. Correia (1982) implementou a teoria de Biot através de elementos finitos baseando-se no método de Galerkin para a discretização espacial e num esquema de integração no tempo baseado no método dos resíduos pesados. Ding e Naylor (1989) usaram o Princípio dos Trabalhos Virtuais para obterem a equação de equilíbrio e o Princípio do Trabalho

Complementar para a equação de continuidade. O programa de cálculo automático CONSOL proposto nesta dissertação para melhoramento é desenvolvido com base neste último trabalho.

Na sua Tese de Mestrado, Suoangkat (1994), propõe uma formulação de elementos finitos para análise não-linear da consolidação baseada na interpolação deslocamento-pressão, conhecida por dar ótimas aproximações na análise de meios incompressíveis. Na Tese de Doutorado de Perrone (1998) é desenvolvido um programa para análise unidimensional da consolidação e fluência de solos argilosos contendo multicamadas, usando um modelo elasto-visco-plástico. Xie et al. (1999) apresentaram uma solução analítica para a consolidação unidimensional de solos com duas camadas e com limites parcialmente drenados.

NG e Small (2000) apresentam uma variação da teoria de Biot para análise da consolidação acoplada de solos não-saturados usando o método dos elementos finitos. Mira et al. (2003) referiram que certos modelos de elementos finitos que consideram a interação entre o esqueleto do solo e o fluido contido nos poros estão sujeitos a instabilidades numéricas por desprezarem a aceleração do fluido em relação ao esqueleto do solo. Por isso, apresentam formulações estáveis através de técnicas de estabilização numérica. Xie e Leo (2004) desenvolveram soluções analíticas para consolidação unidimensional de argilas saturadas e homogêneas sujeitas a grandes tensões. Zhu e Yin (2004) apresentaram uma nova solução analítica para a análise de consolidação de solos com um dreno vertical quando aplicada uma carga que aumenta linearmente com o tempo até estabilizar e considerando a influência da zona envolvente perturbada pela instalação do dreno.

Chen et al. (2005a) apresentam uma solução analítica exata para a consolidação acoplada axissimétrica de solos saturados semi-infinitos e transversalmente isotrópicos submetidos a uma carga uniforme circular na superfície. No mesmo ano, Chen et al. (2005b) fizeram uma análise não linear da consolidação unidimensional de terrenos com vários estratos com limites parcialmente drenados usando o método da quadratura diferencial, tendo tido em conta o efeito da distribuição das tensões efetivas iniciais. Na sua Tese de Mestrado, Moreira (2005) apresenta um modelo de elementos finitos baseado no método dos resíduos pesados adotando a formulação de Galerkin, para análise da consolidação acoplada de solos com condições de simetria axial.

Rujikiatkamjorn e Indraratna (2006) apresentaram um modelo de elementos finitos para análise tridimensional da consolidação de solos moles acelerada através de drenos verticais pré-fabricados. Geng et al. (2006) desenvolveram um método semi-analítico para análise não-linear da consolidação unidimensional de solos com compressibilidade e permeabilidade variáveis submetidos a cargas cíclicas. Hassanen (2007) adaptou um método que combina as vantagens do método dos elementos finitos e do método dos elementos de fronteira, considerado uma poderosa ferramenta para analisar os problemas de interação solo-estrutura, de modo a incluir a consolidação acoplada de Biot para análise bidimensional de solos totalmente saturados.

Wang et al. (2008) exploram a aplicabilidade dum método sem malha denominado de método de interpolação de pontos radial (RPIM) para simulação numérica da consolidação de solos saturados considerando a sua anisotropia. Xie et al. (2008) apresentam soluções analíticas para o problema de consolidação unidimensional levando em consideração as propriedades reológicas do solo argiloso sob o efeito de diferentes tipos de carregamento. Rado et al. (2009) apresentam um modelo numérico para a análise isotérmica da consolidação acoplada de solos não saturados com base em combinações de estados de tensão diferentes. Menéndez et al. (2009) apresentam um modelo, usando o método dos elementos finitos, para a consolidação de solos elásticos saturados com fluido incompressível. Mais tarde, Menéndez et al. (2010) apresentaram uma análise não linear da consolidação de solos elásticos saturados com fluido incompressível e permeabilidade variável usando o método dos elementos finitos.

Qin et al. (2010) desenvolveram uma solução analítica para a consolidação unidimensional em solos não-saturados com confinamento lateral e carga vertical a variar exponencialmente com o tempo. Ferronato, Castelletto e Gambolati (2010) desenvolveram um modelo tridimensional de consolidação acoplada usando elementos finitos mistos, com o objetivo de atenuar as oscilações numéricas existentes nas pressões dos poros verificadas na interface entre materiais com permeabilidades diferentes. Xie et al. (2010) desenvolveram soluções analíticas para estudo da consolidação unidimensional de solos argilosos considerando uma carga dependente do tempo e os desvios do escoamento da água em relação à lei de Darcy que demonstram a existência de um gradiente limite em alguns solos finos com baixa permeabilidade.

Mais recentemente, Walker (2011) apresentou um novo método para a análise unidimensional, bidimensional e tridimensional da consolidação de solos com drenos verticais ou horizontais. Também Zhang e Tao (2011) introduziram uma nova forma de prever a consolidação do solo a partir da sua microestrutura, baseada numa espécie de função de distribuição da densidade dos poros, estabelecendo a relação entre as deformações macroscópica e microscópica. Para verificar este novo modelo foram feitos ensaios em argilas moles usando um microscópico eletrónico de varrimento.

### 3. PROGRAMA DE CÁLCULO AUTOMÁTICO EXISTENTE NA UNIVERSIDADE DO MINHO

O programa de cálculo automático, do qual se pretende melhorar o pré-processamento e pós-processamento gráfico, permite o cálculo de problemas elásticos quer considerando só o estado de tensão e deformação do meio em análise, quer considerando o problema da consolidação acoplada em que são analisados simultaneamente o processo de deformação e a dissipação da água dos poros.

#### 3.1. Formulação

O programa de cálculo automático usado baseou-se na formulação apresentada por Ding e Naylor (1989) que deduziram o seguinte sistema de equações:

$$\begin{bmatrix} K & C \\ C^T & -H \cdot \alpha \cdot \Delta t \end{bmatrix} \begin{Bmatrix} r_u(t + \Delta t) \\ r_p(t + \Delta t) \end{Bmatrix} = \begin{bmatrix} 0 & 0 \\ C^T & H(1 - \alpha)\Delta t \end{bmatrix} \begin{Bmatrix} r_u(t) \\ r_p(t) \end{Bmatrix} + \begin{Bmatrix} R_u \\ R_p \end{Bmatrix} \quad (1)$$

sendo  $K$  a matriz de rigidez da fase sólida,  $C$  a matriz de acoplamento,  $H$  a matriz de fluxo,  $\Delta t$  o incremento de tempo,  $r_u$  o vetor dos deslocamentos nodais,  $r_p$  o vetor das pressões neutras nodais,  $R_u$  o vetor das forças nodais equivalentes,  $R_p$  o vetor dos fluxos nodais aplicados e  $\alpha$  o parâmetro de interpolação na integração em ordem ao tempo.

Nos problemas em que se considera apenas o estado de tensão e deformação, a Equação 1 toma a seguinte forma:

$$K \cdot r_u = R_u \quad (2)$$

### 3.2. Descrição sucinta do programa

O programa é constituído por uma rotina principal denominada CONSOL que controla a ordem de chamada de várias subrotinas principais que executam os passos básicos e que controlam a chamada de subrotinas auxiliares que executam operações secundárias.

A primeira subrotina principal a ser solicitada é a subrotina INPUT. Nesta subrotina é pedido o nome do ficheiro de dados e o nome do ficheiro no qual serão escritos os resultados. É também feita a leitura da maior parte dos dados do problema, nomeadamente os dados respeitantes à geometria do problema, às condições de fronteira e às propriedades materiais. Esta subrotina usa como subrotinas auxiliares as subrotinas CHECK1, CHECK2 e NODEXY. As subrotinas CHECK1 e CHECK2 destinam-se a detetar os erros resultantes de uma má definição dos dados. Quando é detetado, pelo menos, um erro nessas subrotinas, um diagnóstico dos erros ocorridos é escrito no ficheiro de resultados escolhido anteriormente. A subrotina NODEXY calcula as coordenadas dos nós situados a meio dos lados.

A seguir à subrotina INPUT é chamada a subrotina GAUSSQ que estabelece as posições dos pontos de Gauss e os fatores de peso a usar na integração numérica. O programa está restringido às regras de integração de 2x2 e 3x3 pontos. Após a subrotina GAUSSQ, é calculada a semi-banda por meio da subrotina MBANDA. Terminado o cálculo da semi-banda, o programa principal chama a subrotina STIFPS. Esta subrotina calcula as matrizes de rigidez dos vários elementos. Ela comanda a chamada das seguintes subrotinas:

- MODPS: calcula a matriz de elasticidade [D] para problemas bidimensionais;
- SFR2: calcula as funções de forma e as suas derivadas nos pontos de gauss dos elementos;
- MAPFUN: calcula as funções de "mapping" e as suas derivadas nos pontos de gauss dos elementos;
- JACOB2: calcula as coordenadas cartesianas dos pontos de gauss, a matriz jacobiana e o seu determinante, a inversa da matriz jacobiana e as derivadas cartesianas das funções de forma;
- BMATPS: calcula a matriz de deformação-deslocamento [B];
- DBE: efetua o produto da matriz [D] com a matriz [B];
- AEST1: efetua a assemblagem das matrizes de rigidez.



De volta à rotina principal, é chamada a subrotina COUPPS que calcula as matrizes de acoplamento dos elementos. Esta subrotina usa como subrotinas auxiliares as subrotinas SFR2, MAPFUN, JACOB2, que já foram descritas acima, e também as subrotinas:

- BVMTPS: calcula a matriz  $[B_v]$ ;
- ASCOU1: faz a montagem das matrizes de acoplamento.

Depois da subrotina COUPPS, é chamada a subrotina FLOWPS que calcula as matrizes de fluxo dos elementos. Esta subrotina usa como subrotinas auxiliares as subrotinas SFR2, MAPFUN, JACOB2 que já foram descritas acima, e também as subrotinas:

- BPMTPS: calcula a matriz  $[B_p]$ ;
- KBP: multiplica a matriz permeabilidade pela matriz  $[B_p]$ ;
- ASFLO1: Faz a montagem das matrizes de fluxo.

De volta ao programa principal, é chamada a subrotina LOADPS. Nela é feita a leitura do tipo de solicitação bem como do seu valor e calculadas, para cada elemento, as forças nodais equivalentes de acordo com o tipo de solicitação aplicada, formando então o vetor de cargas aplicadas. As solicitações incluem cargas concentradas nos nós, cargas distribuídas nos lados dos elementos, ação da gravidade e tensões iniciais. A subrotina LOADPS chama as subrotinas auxiliares SFR2, MAPFUN, JACOB2 e BMATPS, que já foram descritas acima.

Concluída a subrotina LOADPS, é chamada a subrotina ELFLOW que calcula o vetor de escoamento de cada elemento. Esta subrotina usa como subrotinas auxiliares as subrotinas SFR2, MAPFUN, JACOB2 e BPMTPS, que já foram descritas acima.

Terminada a subrotina ELFLOW, é feita a montagem do vetor solicitação e dos caudais através da subrotina ALOAD1. Depois é feita a inicialização dos deslocamentos e pressões através da subrotina INITI1. Após isso, é chamada a subrotina IGUAL para colocação da matriz global em GBMAT1, a subrotina HALIN para fazer o produto da matriz  $[H]$  por alfa, e a subrotina de alisamento (smoothing na terminologia anglo-saxónica) SMOOTH. Havendo smoothing, são introduzidas as restrições através das subrotinas RESMO1 e RESMO2.

Concluído isto, é construído o vetor do lado direito através da subrotina DIRVEC e colocadas as restrições nos vetores globais através da subrotina RESTR1. Depois é resolvido o sistema de equações com a subrotina BBANDA.

Por fim é chamada a subrotina OUTPU1 para saída dos resultados e a subrotina STREPS para cálculo das tensões em todos os elementos, caso isso seja pretendido. De seguida é apresentado um diagrama de sequência do programa CONSOL (Figura.1).

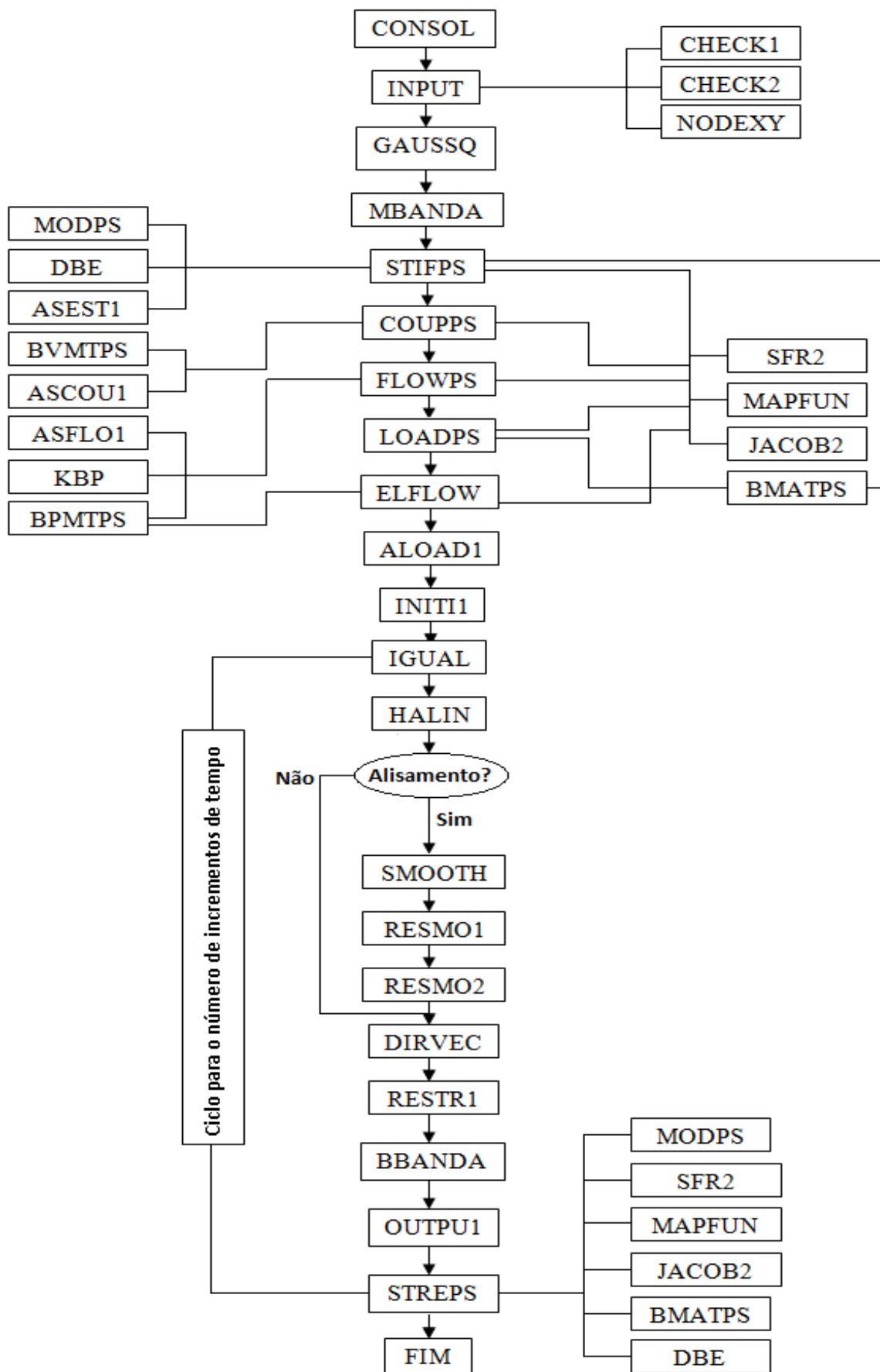


Figura 1: Diagrama de seqüência do programa CONSOL.

### 3.3. Validação do programa

Dado o programa de cálculo automático não ser usado há algum tempo e, uma vez que não se tinha a certeza que a versão usada era a versão correta, resolveu-se efetuar a validação da versão usada recorrendo a um exemplo apresentado por Martins (1993).

Para isso, foi utilizado o exemplo de consolidação bidimensional e comparados os resultados obtidos com os resultados da versão usada do programa.

O exemplo de consolidação bidimensional considera o solo elástico linear e isotrópico, sendo usados elementos finitos de 8 nós, correspondentes a uma interpolação quadrática, quer para as pressões neutras quer para os deslocamentos (elementos do tipo 8-8). Nele é considerado um meio semi-infinito submetido a uma carga distribuída de  $1\text{kN/m}^2$  aplicada à superfície numa faixa infinita de 2 metros de largura.

Na malha utilizada, que é constituída por 29 elementos, é aproveitada a simetria e a truncagem é feita a 8m de profundidade e a 10m do eixo de simetria como mostra a Figura 2.

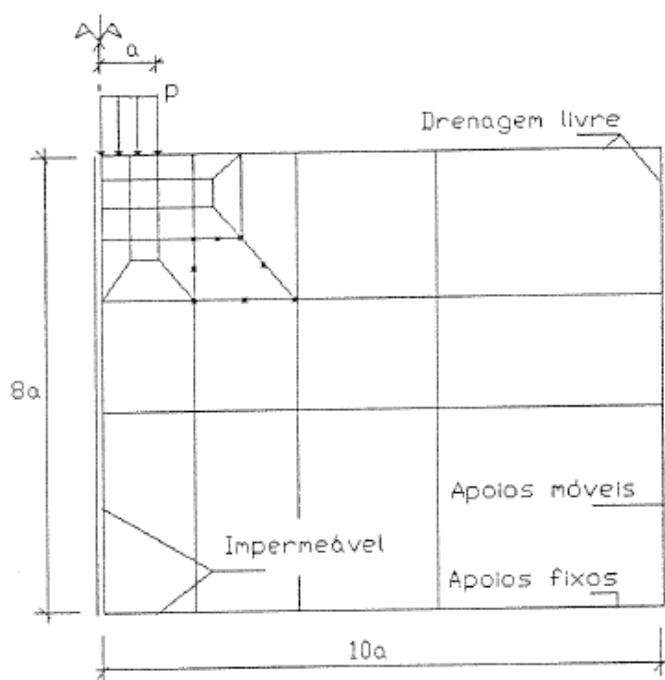


Figura 2: Malha de elementos finitos do exemplo usado (Martins, 1993).

Neste exemplo são usados os seguintes parâmetros:

- Módulo de elasticidade  $E = 15000 \text{ kN/m}^2$ ;
- Coeficiente de Poisson  $\nu = 0,0$ ;
- Coeficiente de permeabilidade  $k = 0,7 \times 10^{-3} \text{ m/h}$ ;
- Peso específico da água  $\gamma_w = 9,8 \text{ kN/m}^3$ ;
- Metade da largura da faixa carregada  $a = 1 \text{ m}$ ;
- $\alpha = 0,666667$ ;
- Módulo de elasticidade transversal  $G$  dado por:

$$G = \frac{E}{2(1 + \nu)} \quad (3)$$

- Fator tempo  $T'$  dado por:

$$T' = \frac{2 \times G \times k}{\gamma_w \times a^2} \times t \quad (4)$$

Na Figura 3 está representada a variação de  $r_p/p$  com o fator tempo  $T'$  num ponto situado no eixo de simetria ( $x/a=0\text{m}$ ) e  $y/a=1\text{m}$ . Nessa mesma figura está também representada a solução analítica e é possível observar que os valores obtidos se aproximam bastante dessa solução.

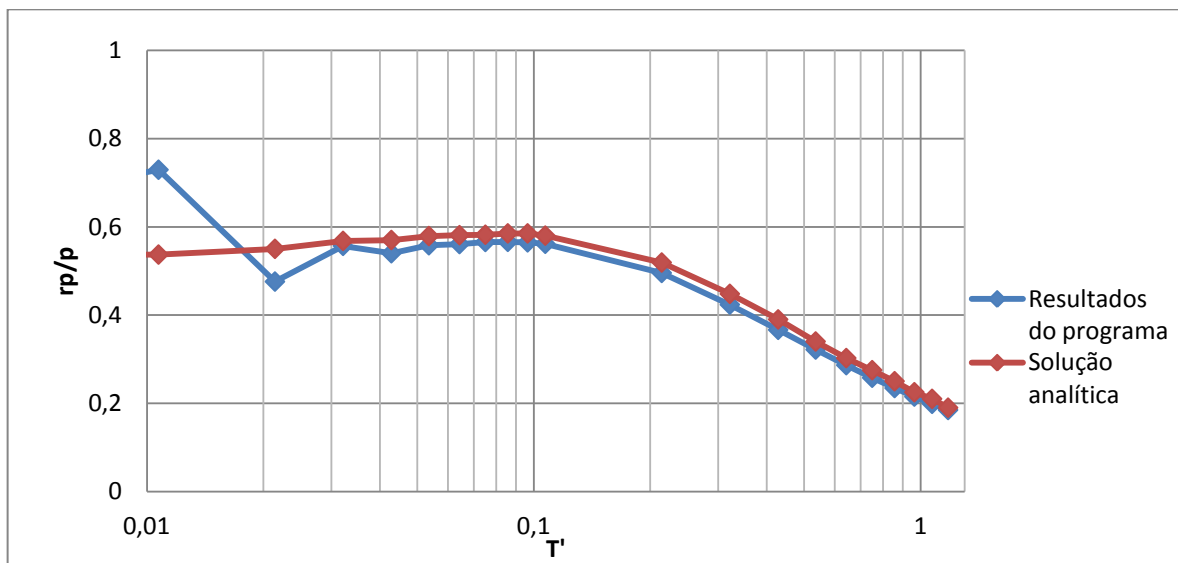


Figura 3: Gráfico variação de  $r_p/p$  com o fator tempo  $T'$  num ponto situado no eixo de simetria ( $x/a=0\text{m}$ ) e  $y/a=1\text{m}$ .

Na Figura 4 é apresentada a variação de  $r_p/p$  com  $x/a$  em pontos situados numa linha correspondente a  $y/a=0,5m$  para  $T'=0,214$ . É possível observar que os resultados obtidos através do programa se situam próximos da solução analítica que é também apresentada na figura.

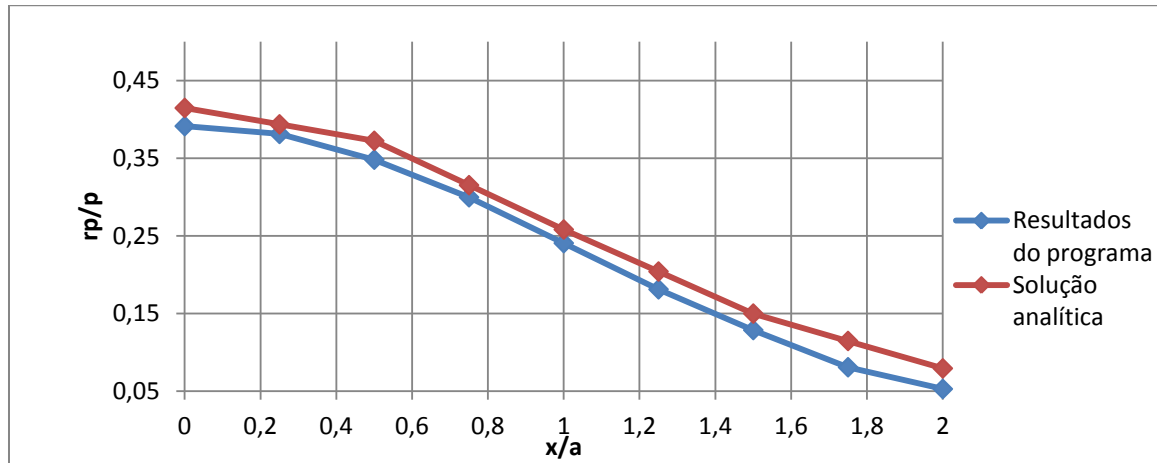


Figura 4: Gráfico variação de  $r_p/p$  com  $x/a$  em pontos situados numa linha correspondente a  $y/a=0,5m$  para  $T'=0,214$ .

Na Figura 5 é apresentada a variação de  $r_p/p$  com  $x/a$  em pontos situados numa linha correspondente a  $y/a=0,5m$  para  $T'=0,857$ . Da sua análise é possível constatar a proximidade entre a solução analítica e a solução obtida com o programa. Após a análise dos resultados obtidos com este exemplo, que são semelhantes aos obtidos por Martins (1993), verifica-se que o programa está a funcionar convenientemente.

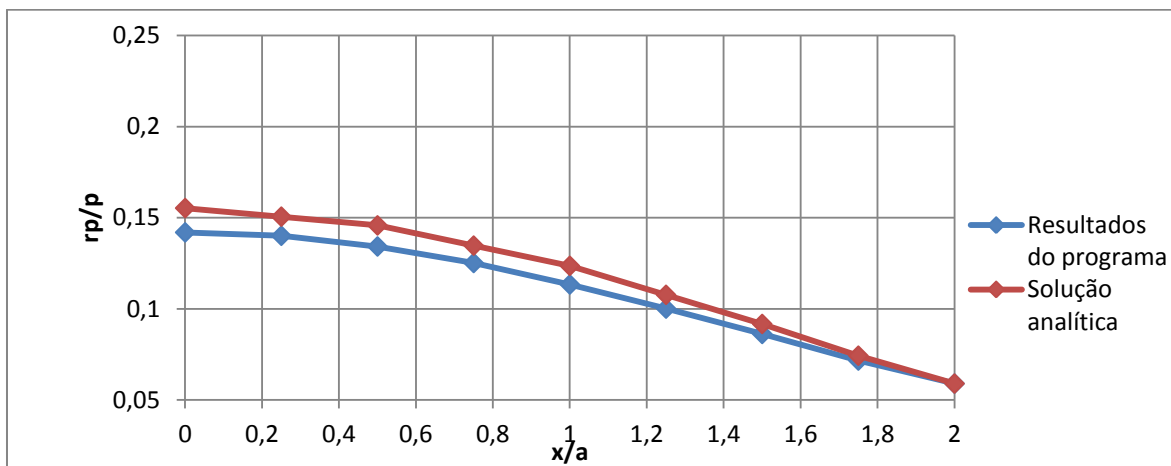


Figura 5: Gráfico variação de  $r_p/p$  com  $x/a$  em pontos situados numa linha correspondente a  $y/a=0,5m$  para  $T'=0,857$ .

#### **4. DESENVOLVIMENTO DE INTERFACES PARA O PROGRAMA**

Com o objetivo de tornar o programa CONSOL amigável para o utilizador, foram criadas interfaces gráficas usando o software MatLab, interfaces essas que são apresentadas neste capítulo e cujos códigos podem ser vistos em anexo. Quando executado o programa, é apresentado o menu principal, no qual o utilizador pode escolher o que pretende efetuar.

Nesse menu principal (Figura 6), o utilizador pode escolher entre criar um ficheiro de dados, que o programa de cálculo automático existente na Universidade do Minho necessita que esteja previamente criado, criar um ficheiro de resultados e também ver os resultados desse ficheiro graficamente.

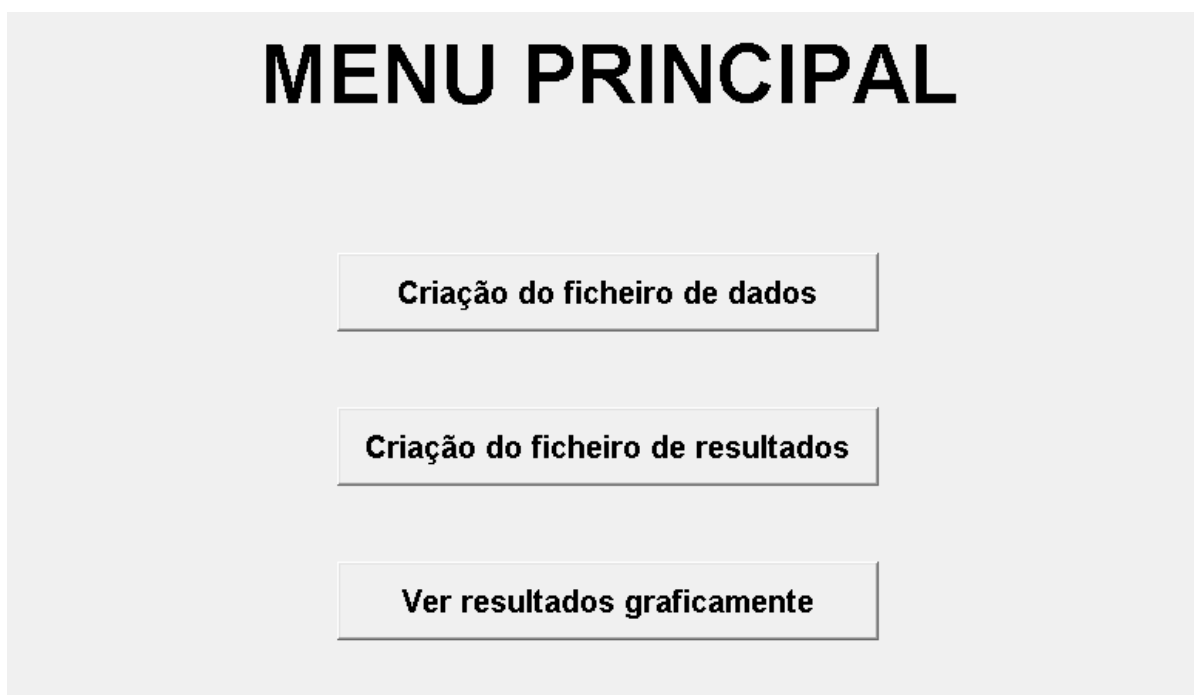


Figura 6: Menu principal do programa.

## 4.1. Interface de introdução de dados

Escolhendo criar um ficheiro de dados, o utilizador pode optar entre uma introdução de dados básica ou uma introdução de dados avançada. Na introdução de dados básica o utilizador só pode escolher entre dois tipos de malha e alguns parâmetros estão definidos por defeito, enquanto que na introdução de dados avançada o utilizador pode usar qualquer tipo de malha e não estão definidos por defeito os parâmetros que estão definidos por defeito na introdução de dados básica. De referir que em ambas as interfaces de introdução de dados o separador decimal é o ponto (.) em vez da vírgula (,) e que no caso dos números do tipo  $1,5 \times 10^{-6}$  são indicados da forma 1.5E-6.

### 4.1.1. Interface de introdução de dados básica

Se o utilizador optar pela introdução de dados básica é-lhe pedido que escolha entre consolidação unidimensional ou consolidação bidimensional e que introduza o valor de "a" da carga distribuída, em metros, como mostra a Figura 7. Clicando no botão "Consolidação unidimensional" ou no botão "Consolidação bidimensional" surge uma imagem do tipo de malha que é usado em cada caso, sendo possível ver com maior pormenor o tipo de malha usado em cada caso e o sistema de eixos adotado (Figuras 8 e 9).

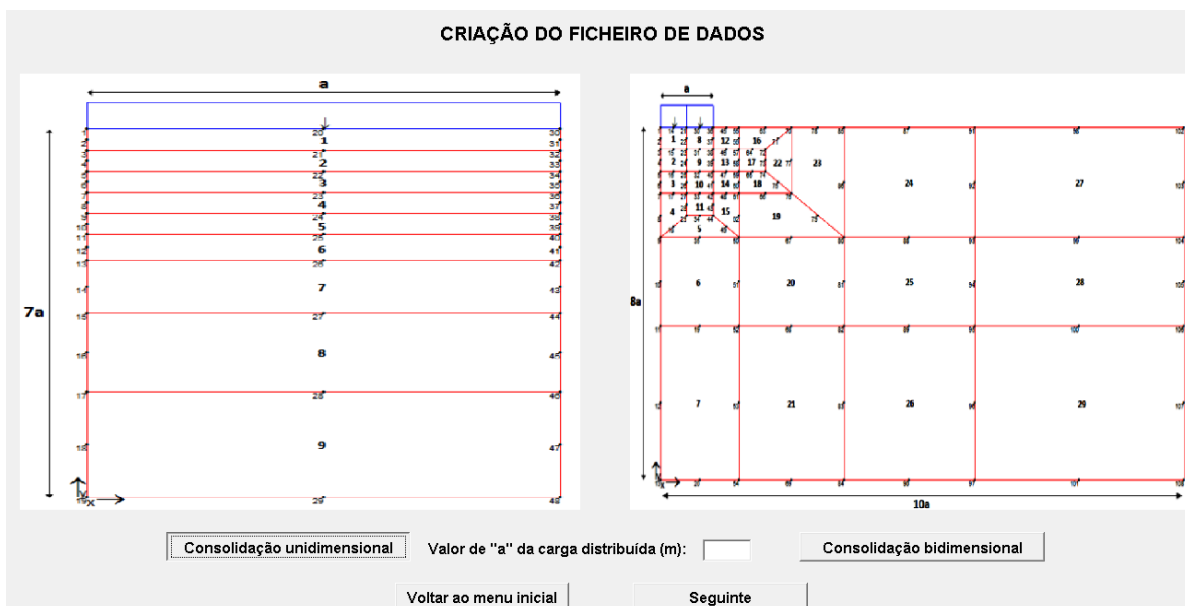


Figura 7: Primeiros parâmetros da interface de introdução de dados básica.



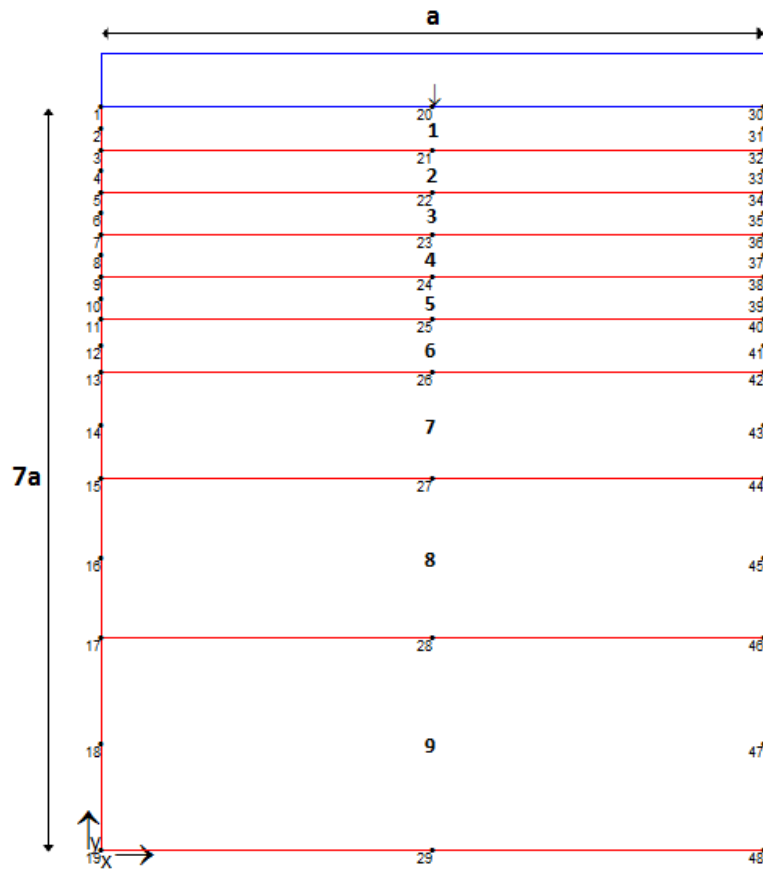


Figura 8: Malha usada no caso da consolidação unidimensional.

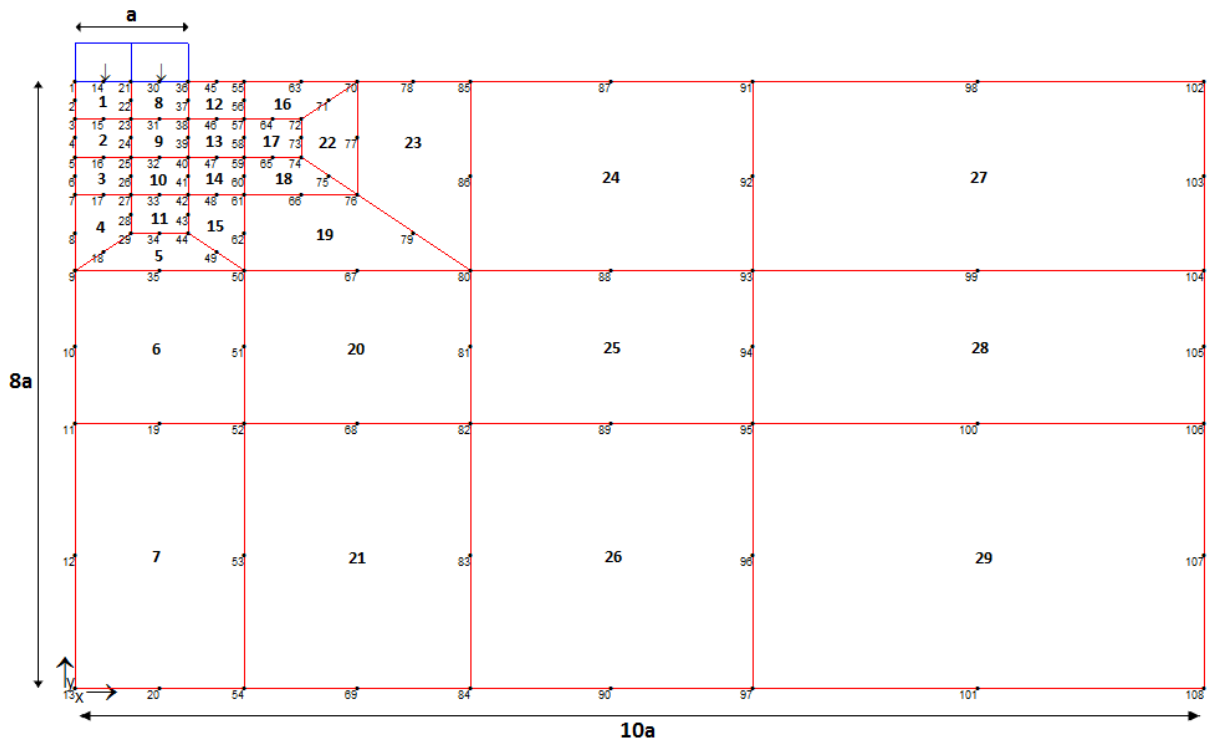


Figura 9: Malha usada no caso da consolidação bidimensional.

Clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros de controlo (Figura 10):

- Número de nós onde há graus de liberdade restringidos ("NVFIX");
- Número de nós onde se prescreve a pressão neutra ("NPRFX");
- Número de materiais diferentes ("NMATS");
- Número de pontos de Gauss em cada direção ("NGAUS");
- Tipo de elementos ("TELEM");
- Número de incrementos de tempo ("NINCR");
- Parâmetro da pressão prescrita ("LINCA").

De referir que alguns parâmetros estão definidos por defeito, nomeadamente o tipo de problema como sendo de deformação plana ("NTYPE=2") e os elementos como sendo elementos finitos "serendipity" de 8 nós ("MNODE=8").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros de controlo**

Número total de nós onde há graus de liberdade restringidos:	<input type="text"/>	Tipo de elementos: 0-elementos tendo todos os seus nós com deslocamentos segundo x e y e ainda a pressão neutra (elementos 8-8); 1-elementos 8-8 com alisamento.	<input type="text"/>
Número de nós onde se prescreve a pressão neutra (faz parte das condições de fronteira):	<input type="text"/>	Número de incrementos de tempo:	<input type="text"/>
Número de materiais diferentes:	<input type="text"/>	Parâmetro da pressão prescrita: 0-na primeira iteração introduz a pressão prescrita no vetor global 1-considera por defeito essas pressões iguais a zero	<input type="text"/>
Número de pontos de Gauss em cada uma das direções: (2 ou 3 pontos)	<input type="text"/>		

1

Figura 10: Parâmetros de controlo da interface de introdução de dados básica.

Após terem sido introduzidos todos os valores dos parâmetros de controlo, se o número de materiais diferentes for maior do que um ("NMATS>1") clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros dos elementos (Figura 11). Nesta fase, é pedido ao utilizador que introduza, para cada elemento, o número da propriedade material do elemento ("MATNO"), sendo apresentado o número de nós que

pertencem ao elemento ("NNODE") e o número dos nós pertencentes ao elemento ("LNODS"). Clicando no botão "Ver malha" surge uma imagem da malha em uso (Figuras 8 e 9) podendo-se assim ver com maior pormenor a numeração dos nós e o sistema de eixos adotado.

The screenshot shows a software interface titled "CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros dos elementos". It contains several input fields: "Número do elemento:" with the value "1"; "Número de nós que pertencem ao elemento:" with the value "8"; "Número dos nós pertencentes ao elemento:" with the values "1 2 3 21 32 31 30 20"; and "Número da propriedade material do elemento:" which is empty. Below these fields is a button labeled "Elemento seguinte". To the right, there is a diagram of a rectangular mesh with nodes numbered 1 to 20. The width is labeled 'a' and the height is labeled '7a'. A coordinate system (x, y) is shown at the bottom left. At the bottom of the interface are three buttons: "Voltar ao menu inicial", "Anterior", and "Seguinte". A "Ver malha" button is located below the diagram.

Figura 11: Parâmetros dos elementos da interface de introdução de dados básica.

Tendo sido introduzidos todos os parâmetros dos elementos, clicando no botão "Seguinte" surgem os parâmetros dos nós restringidos (Figura 12), sendo necessário introduzir, para cada nó restringido, os seguintes parâmetros:

- O número do nó restringido ("NOFIX");
- As condições de restrição do nó restringido no deslocamento segundo os eixos x e y ("IFPRE");
- O valor prescrito das componentes x e y do deslocamento nodal, em metros ("PRESC").

Clicando no botão "Ver malha" surge uma imagem da malha em uso (Figuras 8 e 9) podendo-se assim ver com maior pormenor a numeração dos nós e o sistema de eixos adotado.

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros dos nós restringidos**

Número do nó restringido:

Condição de restrição no deslocamento segundo x:  
 0-deslocamento não restringido;  
 1-deslocamento restringido.

Condição de restrição no deslocamento segundo y:  
 0-deslocamento não restringido;  
 1-deslocamento restringido.

Valor prescrito da componente x do deslocamento nodal (m):

Valor prescrito da componente y do deslocamento nodal (m):

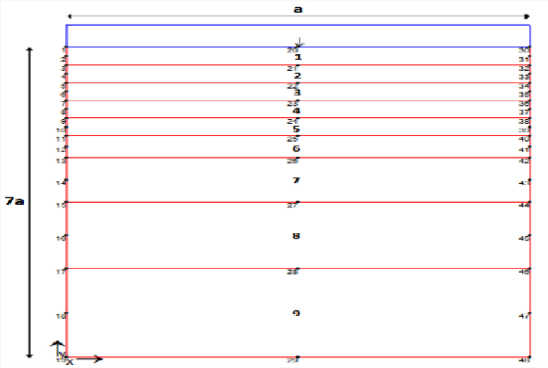


Figura 12: Parâmetros dos nós restringidos.

Após terem sido introduzidos todos os valores dos parâmetros dos nós restringidos, clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros dos nós onde se prescreve a pressão neutra (Figura 13), sendo necessário introduzir, para cada nó, o número do nó ("NOPRF") e o valor prescrito da pressão neutra, em kPa ("PREPR"). Clicando no botão "Ver malha" surge uma imagem da malha em uso (Figuras 8 e 9) podendo-se assim ver com maior pormenor a numeração dos nós e o sistema de eixos adotado.

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros dos nós onde se prescreve a pressão neutra**

Número do nó:

Valor prescrito da pressão neutra (kPa):

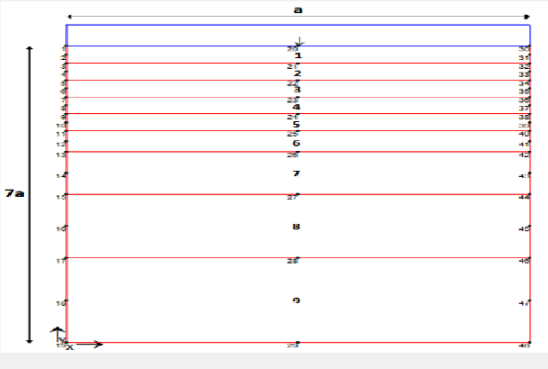


Figura 13: Parâmetros dos nós onde se prescreve a pressão neutra.

Terminada a introdução de todos os valores dos parâmetros dos nós onde se prescreve a pressão neutra, clicando no botão "Seguinte" é pedido ao utilizador que introduza o seguinte conjunto de parâmetros (Figura 14):

- O valor de alfa,  $\alpha$  entre 0,5 e 1,0 ("AALFA");
- O peso específico da água,  $\gamma_w$  em  $\text{kN/m}^3$  ("GAMAW");
- O valor do coeficiente da permeabilidade,  $k$  em  $\text{m/h}$  ("PERMB").

The image shows a software interface titled "CRIAÇÃO DO FICHEIRO DE DADOS". It contains three input fields with labels: "Alfa (valor entre 0.5 e 1.0):", "Peso específico da água, em  $\text{kN/m}^3$ :", and "Coeficiente de permeabilidade,  $k$  em  $\text{m/h}$ :". At the bottom, there are three buttons: "Voltar ao menu inicial", "Anterior", and "Seguinte".

Figura 14: Outros parâmetros da interface de introdução de dados básica.

Após a introdução de todos os valores dos parâmetros referidos anteriormente, clicando no botão "Seguinte" surgem os parâmetros dos materiais (Figura 15), sendo necessário introduzir, para cada tipo de material diferente, os seguintes parâmetros:

- O valor do módulo de elasticidade,  $E$  em  $\text{kN/m}^2$  ("PROPS1");
- O valor do coeficiente de Poisson,  $\nu$  ("PROPS2");
- O valor da espessura do material,  $t$  em metros ("PROPS3");
- O valor da densidade do material,  $\rho$  em  $\text{kg/m}^3$  ("PROPS4").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros dos materiais**

Número de identificação do material:

Módulo de elasticidade, E em kN/m<sup>2</sup>:

Coefficiente de Poisson:

Espessura do material, t (m):

Densidade do material, em kg/m<sup>3</sup>:

Figura 15: Parâmetros dos materiais.

Tendo sido introduzidos todos os valores dos parâmetros dos materiais, clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros do tempo (Figura 16), sendo necessário introduzir, para cada incremento de tempo o valor do incremento de tempo pretendido, em horas ("INCRT"). De referir que o parâmetro para indicar se é pretendido o cálculo de tensões no incremento de tempo em causa ("ISOUT") está definido por defeito como sendo para efetuar o cálculo de tensões ("ISOUT=1").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros do tempo**

Valor do incremento de tempo, em horas:

Figura 16: Parâmetros do tempo da interface de introdução de dados básica.

Após a introdução de todos os parâmetros do tempo, clicando no botão "Seguinte" surgem os parâmetros das cargas (Figura 17):

- Indicação do parâmetro designativo das forças de gravidade ("IGRAV");
- Indicação se é pretendido considerar a tensão inicial do solo ("INIST").

De referir que alguns parâmetros estão definidos por defeito, nomeadamente a não existência de cargas pontuais ("IPL0D=0") e a existência de cargas distribuídas ("IEDGE=1").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das cargas**

Parâmetro designativo das forças de gravidade:   
0-não há forças de gravidade;  
1-há forças de gravidade.

Parâmetro designativo da tensão inicial:   
0-não há tensão inicial;  
1-há tensão inicial.

Figura 17: Parâmetros das cargas da interface de introdução de dados básica.

Após a introdução de todos os valores dos parâmetros das cargas, clicando no botão "Seguinte" surgem, caso o parâmetro designativo das forças de gravidade seja igual a um ("IGRAV=1"), os parâmetros das forças de gravidade (Figura 18):

- O ângulo entre a direção da gravidade e o eixo y medido a partir deste eixo, no sentido direto ou retrógrado, de forma a obter o menor ângulo ("THETA");
- A constante de gravidade definida como um múltiplo da aceleração gravitacional, g ("GRAVY").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das forças de gravidade**

Ângulo entre a direção da gravidade e o eixo dos yy medido a partir deste eixo, no sentido direto ou retrógrado, de forma a obter o menor ângulo:

Constante de gravidade definida como um múltiplo da aceleração gravitacional g:

Figura 18: Parâmetros das forças de gravidade.

Tendo sido introduzidos todos os valores dos parâmetros das forças de gravidade, clicando no botão "Seguinte" surgem, caso tenha sido indicado que se pretende considerar a tensão inicial do solo ("INIST=1"), os parâmetros da tensão inicial (Figura 19):

- O valor da coordenada y da superfície do terreno, em metros ("SURCO");
- A constante de gravidade definida como um múltiplo da aceleração gravitacional, g ("GRAVY").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros da tensão inicial**

Coordenada y da superfície do terreno (m):

Constante de gravidade definida como um múltiplo da aceleração gravitacional g:

Figura 19: Parâmetros da tensão inicial.



Após terem sido introduzidos todos os valores dos parâmetros da tensão inicial, clicando no botão "Seguinte" é ainda necessário introduzir os valores dos parâmetros do coeficiente de impulso em repouso (Figura 20):

- Parâmetro do coeficiente de impulso em repouso ("IK");
- Se o parâmetro do coeficiente de impulso em repouso for igual a zero ("IK=0"), é necessário introduzir o valor do coeficiente de impulso em repouso ("K0"). Caso o parâmetro do coeficiente de impulso em repouso seja igual a um ("IK=1"), o coeficiente de impulso em repouso é calculado através da Equação 5:

$$K_0 = \frac{\nu}{1 - \nu} \quad (5)$$

sendo  $\nu$  o coeficiente de Poisson.

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros do coeficiente de impulso em repouso**

Parâmetro do coeficiente de impulso em repouso:

0- para introduzir o valor do coeficiente de impulso em repouso;  
1- coeficiente de impulso em repouso calculado pela razão:  
coeficiente de poisson/(1-coeficiente de poisson)

Valor do coeficiente de impulso em repouso:

Figura 20: Parâmetros do coeficiente de impulso em repouso.

Para finalizar a introdução dos valores dos parâmetros necessários para a criação do ficheiro de dados, tendo sido introduzidos todos os valores dos parâmetros do coeficiente de impulso em repouso e clicando no botão "Seguinte" surge o último conjunto de parâmetros, denominados de parâmetros das cargas distribuídas (Figura 21), sendo necessário introduzir o valor da componente normal ("PRESS(1)") e da componente tangencial ("PRESS(2)") da

carga distribuída, tendo em conta o sistema de eixos representado nas Figuras 8 e 9 para a definição do sentido da carga.

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das cargas distribuídas**

Ter em atenção o sistema de eixos usado para a definição do sentido da carga.

Valor da componente normal da carga distribuída, em kN/m<sup>2</sup>:

Valor da componente tangencial da carga distribuída, em kN/m<sup>2</sup>:

Figura 21: Parâmetros das cargas distribuídas da interface de introdução de dados básica.

#### 4.1.2. Interface de introdução de dados avançada

Se optar pela introdução de dados avançada surge a interface de introdução de dados avançada onde é pedido ao utilizador que introduza o valor de um conjunto de parâmetros. De referir que a numeração dos nós e dos elementos devem seguir uma ordem específica como mostram as Figuras 8 e 9, isto é, devem ser numerados de cima para baixo e da esquerda para a direita.

O primeiro conjunto de parâmetros a introduzir são os parâmetros de controlo (Figura 22):

- Número de nós ("NPOIN");
- Número de elementos ("NELEM");
- Número de nós onde há graus de liberdade restringidos ("NVFIX");
- Número de nós onde se prescreve a pressão neutra ("NPRFX");
- Tipo de problema ("NTYPE");
- Número de nós por elemento ("MNODE");

- Número de materiais diferentes ("NMATS");
- Número de pontos de Gauss em cada direção ("NGAUS");
- Tipo de elementos ("TELEM");
- Número de incrementos de tempo ("NINCR");
- Parâmetro da pressão prescrita ("LINCA").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros de controlo**

Número total de nós: <input style="width: 50px;" type="text"/>	Número de materiais diferentes: <input style="width: 50px;" type="text"/>
Número total de elementos: <input style="width: 50px;" type="text"/>	Número de pontos de Gauss em cada uma das direções: (2 ou 3 pontos) <input style="width: 50px;" type="text"/>
Número total de nós onde há graus de liberdade restringidos: <input style="width: 50px;" type="text"/>	Tipo de elementos: 0-elementos tendo todos os seus nós com deslocamentos segundo x e y e ainda a pressão neutra (elementos 8-8); 1-elementos 8-8 com alisamento. <input style="width: 50px;" type="text"/>
Número de nós onde se prescreve a pressão neutra (faz parte das condições de fronteira): <input style="width: 50px;" type="text"/>	Número de incrementos de tempo: <input style="width: 50px;" type="text"/>
Tipo de problema: 1-tensão plana; 2-deformação plana. <input style="width: 50px;" type="text"/>	Parâmetro da pressão prescrita: 0-na primeira iteração introduz a pressão prescrita no vetor global 1-considera por defeito essas pressões iguais a zero <input style="width: 50px;" type="text"/>
Número máximo de nós por elemento: 4-elementos finitos lagrangianos de 4 nós; 8-elementos finitos "serendipity" de 8 nós. <input style="width: 50px;" type="text"/>	<input style="width: 50px;" type="text"/>
<input type="button" value="Voltar ao menu inicial"/>	<input type="button" value="Seguinte"/>

Figura 22: Parâmetros de controlo da interface de introdução de dados avançada.

Após terem sido introduzidos todos os valores dos parâmetros de controlo, clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros dos elementos (Figura 23). Nesta fase, é pedido ao utilizador que introduza, para cada elemento, o número dos nós que pertencem ao elemento em causa ("LNODS"), introduzindo os números separados entre eles por espaço e segundo uma ordem específica, de uma forma geral começando pelo nó do elemento mais acima e mais à esquerda e introduzindo os outros nós do elemento no sentido anti-horário. Usando como exemplo a Figura 9, no caso do elemento número 1 a ordem seria "1 2 3 15 23 22 21 14". No entanto, no caso do elemento número 5 a ordem passa a ser "9 35 50 49 44 34 29 18" e no caso do elemento número 22 passa a ser "70 71 72 73 74 75 76 77". Caso o número de materiais diferentes seja maior do que um ("NMATS>1") é pedido também ao utilizador que introduza, para cada elemento, o número da propriedade material do elemento ("MATNO").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros dos elementos**

Número do elemento:

Número da propriedade material do elemento:

Número de nós que pertencem ao elemento:

Número dos nós pertencentes ao elemento:  
(introduzir número dos nós separados por espaço)

Figura 23: Parâmetros dos elementos da interface de introdução de dados avançada.

Após a introdução de todos os valores dos parâmetros dos elementos, clicando no botão "Seguinte" é pedido ao utilizador que introduza as coordenadas dos nós (Figura 24), sendo necessário apenas que sejam introduzidas as coordenadas dos nós ("IPOIN") dos vértices dos elementos. De referir que as coordenadas x e y ("COORD") dos nós devem ser valores positivos, ou seja, devem ser medidos a partir da posição dos eixos como está indicado nas Figuras 8 e 9.

**CRIAÇÃO DO FICHEIRO DE DADOS - Coordenadas dos nós**

Inserir apenas os dados dos nós localizados nos cantos dos elementos

Número do nó:

Coordenada x do nó (m):

Coordenada y do nó (m):

Figura 24: Parâmetros das coordenadas dos nós da interface de introdução de dados avançada.

Tendo sido introduzidas as coordenadas de todos os nós dos vértices dos elementos, clicando no botão "Seguinte" surgem os parâmetros dos nós restringidos (Figura 12), que são os mesmos que os parâmetros dos nós restringidos da interface de introdução de dados básica. Após terem sido introduzidos todos os valores dos parâmetros dos nós restringidos, clicando no botão "Seguinte" surge o próximo conjunto de parâmetros a introduzir, denominados de parâmetros dos nós onde se prescreve a pressão neutra (Figura 13), que são os mesmos que os parâmetros dos nós onde se prescreve a pressão neutra da interface de introdução de dados básica. Terminada a introdução de todos os valores dos parâmetros dos nós onde se prescreve a pressão neutra, clicando no botão "Seguinte" é pedido ao utilizador que introduza o seguinte conjunto de parâmetros (Figura 25):

- O valor de alfa,  $\alpha$  entre 0,5 e 1,0 ("AALFA");
- O peso específico da água,  $\gamma_w$  em  $\text{kN/m}^3$  ("GAMAW");
- O valor da matriz (matriz 2 por 2) do coeficiente de permeabilidade,  $k$  em  $\text{m/h}$  ("PERMB"), sendo introduzidos os quatro valores separados entre si por espaço com seguinte ordem "PERMB(1,1) PERMB(1,2) PERMB(2,1) PERMB(2,2)".

The screenshot shows a software interface titled "CRIAÇÃO DO FICHEIRO DE DADOS". It contains three input fields with corresponding labels: "Alfa (valor entre 0.5 e 1.0):", "Peso específico da água, em  $\text{kN/m}^3$ :", and "Matriz do coeficiente de permeabilidade,  $k$  em  $\text{m/h}$ : (introduzir os 4 valores separados por espaço pela ordem (1,1) (1,2) (2,1) (2,2) (matriz 2x2))". At the bottom, there are three buttons: "Voltar ao menu inicial", "Anterior", and "Seguinte".

Figura 25: Outros parâmetros da interface de introdução de dados avançada.

Após a introdução de todos os valores dos parâmetros referidos anteriormente, clicando no botão "Seguinte" surge o menu para introdução dos valores dos parâmetros dos materiais, que é igual ao da interface de introdução de dados básica (Figura 15). Tendo sido introduzidos todos os valores dos parâmetros dos materiais, clicando no botão "Seguinte" surge um menu com o próximo conjunto de parâmetros a introduzir, denominados de parâmetros do tempo e tensões (Figura 26), sendo necessário introduzir, para cada incremento de tempo:

- O valor do incremento de tempo pretendido ("INCRT");
- Indicar se é pretendido o cálculo de tensões no incremento de tempo em causa ("ISOUT").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros do tempo e tensões**

Valor do incremento de tempo, em horas:

Cálculo de tensões neste incremento de tempo:

0-não;  
1-sim.

Figura 26: Parâmetros do tempo e tensões da interface de introdução de dados avançada.

Após a introdução de todos os parâmetros do tempo e tensões, clicando no botão "Seguinte" surgem os parâmetros das cargas (Figura 27):

- Indicação da existência de cargas pontuais ("IPL0D"), sendo que em caso afirmativo é necessário introduzir o número de nós onde estão aplicadas cargas pontuais;
- Indicação do parâmetro designativo das forças de gravidade ("IGRAV");
- Indicação da existência de cargas distribuídas ("IEDGE"), sendo que em caso afirmativo é necessário introduzir o número de lados dos elementos onde estão aplicadas cargas distribuídas ("NEDGE");
- Indicação se é pretendido considerar a tensão inicial do solo ("INIST").

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das cargas**

Parâmetro designativo das cargas pontuais: 0-não há cargas pontuais; 1-há cargas pontuais.	<input type="text" value="1"/>	Parâmetro designativo de cargas distribuídas: 0-não há cargas distribuídas; 1-há cargas distribuídas.	<input type="text" value="1"/>
Parâmetro designativo das forças de gravidade: 0-não há forças de gravidade; 1-há forças de gravidade.	<input type="text"/>	Parâmetro designativo da tensão inicial: 0-não há tensão inicial; 1-há tensão inicial.	<input type="text"/>
Número de nós onde estão aplicadas cargas pontuais:	<input type="text"/>	Número de lados onde estão aplicadas cargas distribuídas:	<input type="text"/>

Figura 27: Parâmetros das cargas da interface de introdução de dados avançada.

Terminada a introdução de todos os valores dos parâmetros das cargas, clicando no botão "Seguinte" surgem, caso tenha sido dada a indicação de que existem cargas pontuais ("IPL0D=1"), os parâmetros das cargas pontuais (Figura 28), sendo necessário introduzir, para cada nó onde esteja aplicada uma carga pontual, os seguintes parâmetros:

- O número do nó onde é aplicada a carga pontual ("LODPT");
- O valor das componentes da carga nas direções x e y ("POINT"), tendo em conta o sistema de eixos representado nas Figuras 8 e 9 para a definição do sentido da carga.

**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das cargas pontuais**

Ter em atenção o sistema de eixos usado para a definição do sentido da carga.

Carga pontual:

Número do nó:

Componente da carga na direção x (kN/m):

Componente da carga na direção y (kN/m):

Figura 28: Parâmetros das cargas pontuais da interface de introdução de dados avançada.

Após a introdução de todos os valores dos parâmetros das cargas pontuais, clicando no botão "Seguinte" surge, caso o parâmetro designativo das forças de gravidade seja igual a um ("IGRAV=1"), os parâmetros das forças de gravidade (Figura 18) que são os mesmos que os parâmetros das forças de gravidade da interface de introdução de dados básica. Tendo sido introduzidos todos os valores dos parâmetros das forças de gravidade, clicando no botão "Seguinte" surgem, caso tenha sido indicado que se pretende considerar a tensão inicial do solo ("INIST=1"), os parâmetros da tensão inicial (Figura 19) que são os mesmos que os parâmetros da tensão inicial da interface de introdução de dados básica. Após terem sido introduzidos todos os valores dos parâmetros da tensão inicial, clicando no botão "Seguinte" é ainda necessário introduzir os valores dos parâmetros do coeficiente de impulso em repouso (Figura 20) que são os mesmos que os parâmetros do coeficiente de impulso em repouso da interface de introdução de dados básica.

Para finalizar a introdução dos valores dos parâmetros necessários para a criação do ficheiro de dados, tendo sido introduzidos todos os valores dos parâmetros do coeficiente de impulso em repouso e clicando no botão "Seguinte" surge, caso tenha sido dada a indicação de que existem cargas distribuídas ("IEDGE=1"), o último conjunto de parâmetros denominados de parâmetros das cargas distribuídas (Figura 29), sendo necessário introduzir, para cada lado dos elementos onde esteja aplicada uma carga distribuída, os seguintes parâmetros:

- O número do elemento em relação ao qual o lado carregado está associado ("NEASS");
- Os números dos nós que formam o lado carregado ("NOPRS"). Caso o número máximo de nós por elemento seja igual a quatro (elementos finitos lagrangianos de quatro nós: "MNODE=4") o número de nós que formam o lado carregado será dois ([nó(1) nó(2)]). Caso o número máximo de nós por elemento seja igual a oito (elementos finitos "serendipity" de oito nós: "MNODE=8") o número de nós que formam o lado carregado será três ([nó(1) nó(2) nó(3)]). Os números dos nós são introduzidos em sentido direto e separados entre si por espaço, isto é, usando como o exemplo a Figura 9, para o elemento número 1, seria "21 14 1";
- O valor da componente normal ("PRESS(i,1)") e da componente tangencial ("PRESS(i,2)") da carga distribuída em cada nó(i) que forma o lado carregado em causa, tendo em conta o sistema de eixos representado nas Figuras 8 e 9 para a definição do sentido da carga.



**CRIAÇÃO DO FICHEIRO DE DADOS - Parâmetros das cargas distribuídas**

Ter em atenção o sistema de eixos usado para a definição do sentido da carga.

Carga distribuída:	<input type="text" value="1"/>	Valor da componente normal da carga distribuída no nó(2), em kN/m <sup>2</sup> :	<input type="text"/>
Número do elemento em relação ao qual o lado carregado está associado:	<input type="text"/>	Valor da componente tangencial da carga distribuída no nó(2), em kN/m <sup>2</sup> :	<input type="text"/>
Lista dos números, em sentido direto, dos nós que formam o lado carregado [nó(1) nó(2) nó(3)]:	<input type="text"/>	Valor da componente normal da carga distribuída no nó(3), em kN/m <sup>2</sup> :	<input type="text"/>
Valor da componente normal da carga distribuída no nó(1), em kN/m <sup>2</sup> :	<input type="text"/>	Valor da componente tangencial da carga distribuída no nó(3), em kN/m <sup>2</sup> :	<input type="text"/>
Valor da componente tangencial da carga distribuída no nó(1), em kN/m <sup>2</sup> :	<input type="text"/>		

Figura 29: Parâmetros das cargas distribuídas da interface de introdução de dados avançada.

Tanto na interface de introdução de dados básica como na interface de introdução de dados avançada, com todos os valores dos parâmetros referidos anteriormente introduzidos, procede-se à criação do ficheiro de dados clicando no botão "Concluído" e definindo o nome que se pretende dar ao ficheiro, existindo a partir de ambas as interfaces de introdução de dados a opção de criar o ficheiro de resultados e também de ver os resultados desse ficheiro graficamente. De referir que, a qualquer instante da introdução dos valores dos parâmetros necessários para a criação do ficheiro de dados, é dada a hipótese ao utilizador de retroceder e alterar os valores anteriores introduzidos clicando no botão "Anterior" e também de voltar ao menu principal do programa clicando no botão "Voltar ao menu inicial".

Voltando às opções disponíveis no menu principal do programa, o utilizador tem ainda a opção de criar o ficheiro de resultados (Figura 6) correspondente a um dado ficheiro de dados que já tenha sido criado anteriormente e para o qual não tenha sido criado o respetivo ficheiro de resultados. Esta opção do menu principal permite ainda ver os resultados desse ficheiro graficamente.

## 4.2. Interface de saída de resultados

Escolhendo a opção do menu principal do programa de ver os resultados graficamente (Figura 6), é pedido ao utilizador que escolha o ficheiro de resultados do qual pretende ver os resultados graficamente e posteriormente surge a interface de saída de resultados com uma representação do problema em análise (Figuras 30 e 45). De referir que, a qualquer instante, o utilizador pode ver a representação da malha com a numeração dos nós clicando no botão "Malha" (Figuras 31 e 46). Para além disso, o utilizador pode ver também as coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo clicando no botão "Dados" (Figuras 32 e 47).

Nas opções de representação do gráfico situado na parte superior do menu, que a partir de agora será designado por gráfico1, escolhendo a opção "Deslocamento vertical e altura da coluna de água" no menu pop-up 1 e deixando a opção do menu pop-up 2 por defeito, é mostrado no gráfico 1 uma representação dinâmica, ao longo do tempo, do deslocamento vertical de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador e também a altura da coluna de água num tubo colocado à profundidade do referido nó, considerando o nível freático à superfície. De referir que nesta representação não são mostrados valores reais mas sim mostrados usando uma escala. Ao mesmo tempo no gráfico situado na parte inferior do menu, a partir de agora designado por gráfico 2, é mostrada uma representação dinâmica, ao longo do tempo, da tensão efetiva e total e da pressão neutra no nó definido. De referir que nesta representação são mostrados valores reais e não mostrados usando uma escala. Nas opções do menu pop-up 1 do gráfico 2 o utilizador pode escolher entre representar as tensões verticais (Figuras 33 e 48) ou as tensões horizontais (Figura 49) no nó por si definido.

O utilizador pode ainda adicionar outras representações às referidas anteriormente, nomeadamente vetores do deslocamento, vetores da tensão efetiva, isolinhas do deslocamento e isolinhas da tensão efetiva que podem ser escolhidas no menu pop-up 2 do gráfico 1. No caso dos vetores do deslocamento, as opções disponíveis no menu pop-up 3 do gráfico 1 são entre vertical (Figuras 50 e 52), horizontal e total (Figuras 34, 51 e 53). Para o caso dos vetores da tensão efetiva, as opções disponíveis no menu pop-up 3 do gráfico 1 são entre vertical (Figuras 35, 54 e 55) e horizontal. De referir que nestas representações de vetores não são mostrados valores reais mas sim mostrados usando uma escala. No caso das isolinhas do

deslocamento, as opções disponíveis no menu pop-up 3 do gráfico 1 são entre vertical (Figuras 36 e 56) e horizontal (Figura 57). Por fim, no caso das isolinhas da tensão efetiva, as opções disponíveis no menu pop-up 3 do gráfico 1 são entre vertical (Figura 58), horizontal (Figura 59) e de corte ( $\tau_{xy}$ ) (Figura 60). De referir que nestas representações de isolinhas são mostrados valores reais e não mostrados usando uma escala.

A escolha da opção "Deslocamento total e altura da coluna de água" no menu pop-up 1 do gráfico 1 permite uma representação semelhante à escolha da opção "Deslocamento vertical e altura da coluna de água", diferindo apenas desta no facto de que passa a ser o deslocamento total de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador (Figura 61).

De referir que nas duas opções atrás referidas do menu pop-up 1 do gráfico 1, no caso de consolidação bidimensional, o utilizador pode escolher entre usar o parâmetro "tempo" ou um "fator tempo T" (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso de consolidação unidimensional, o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas.

A opção "Malha (deslocamento total)" do menu pop-up 1 do gráfico 1 mostra uma representação dinâmica, ao longo do tempo, do deslocamento total de todos os nós da malha (Figuras 37 e 62). O utilizador pode ainda adicionar outras representações à referida anteriormente, nomeadamente vetores do deslocamento e vetores da tensão efetiva que podem ser escolhidas no menu pop-up 2 do gráfico 1, sendo que as opções disponíveis no menu pop-up 3 do gráfico 1 são, no caso dos vetores do deslocamento, entre vertical (Figura 63), horizontal (Figura 64) e total (Figuras 38 e 65) e, no caso dos vetores da tensão efetiva, entre vertical (Figuras 39 e 66) e horizontal (Figura 67).

De referir que nas três opções atrás referidas do menu pop-up 1 do gráfico 1 é possível, a qualquer instante, parar a representação dinâmica clicando no botão "Pausa" e também ver os valores de um conjunto de parâmetros clicando no botão "Resultados" (Figuras 40 e 68).

No menu pop-up 1 do gráfico 1, a opção "Deslocamento vertical vs tempo" mostra um gráfico do deslocamento vertical do nó definido pelo utilizador ao longo do tempo (Figuras 41 e 69). Nesta opção, no caso de consolidação bidimensional, o utilizador pode escolher entre usar o

parâmetro "tempo" ou um "fator tempo T" (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso de consolidação unidimensional, o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas.

No caso de consolidação unidimensional, escolhendo a opção "Deslocamento vertical vs profundidade" no menu pop-up 1 do gráfico 1 é mostrado um gráfico do deslocamento vertical de um conjunto de nós distribuídos numa linha vertical com a mesma abcissa do nó definido pelo utilizador versus a profundidade de cada nó, para o tempo definido (Figura 42). No caso de consolidação bidimensional, escolhendo a opção "Deslocamento vertical vs distância ao eixo y" no menu pop-up 1 do gráfico 1, é mostrado um gráfico do deslocamento vertical de um conjunto de nós distribuídos numa linha horizontal com a mesma ordenada do nó definido pelo utilizador versus a distância de cada nó ao eixo y, para o tempo definido (Figura 70).

Nas opções de representação do gráfico 1, escolhendo a opção "Pressão neutra vs tempo" no menu pop-up 1, é mostrado um gráfico da pressão neutra do nó definido pelo utilizador ao longo do tempo (Figuras 43 e 71). Nesta opção, no caso de consolidação bidimensional, o utilizador pode escolher entre usar o parâmetro "tempo" ou um "fator tempo T" (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso de consolidação unidimensional, o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso do eixo das ordenadas, tanto para o caso de consolidação bidimensional como para o caso de consolidação unidimensional, o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ).

Por fim, no caso de consolidação unidimensional, escolhendo a opção "Pressão neutra vs profundidade" no menu pop-up 1 do gráfico 1, é mostrado um gráfico da pressão neutra de um conjunto de nós distribuídos numa linha vertical com a mesma abcissa do nó definido pelo utilizador versus a profundidade de cada nó, para o tempo definido (Figura 44). Nesta opção, para o eixo das ordenadas, o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ). No caso de consolidação bidimensional, escolhendo a opção "Pressão neutra vs distância ao eixo y" no menu pop-up 1 do gráfico 1, é mostrado um gráfico da pressão neutra de um conjunto de nós distribuídos numa linha horizontal com a mesma ordenada do nó definido pelo utilizador versus a

distância de cada nó ao eixo y, para o tempo definido (Figura 72). Nesta opção o utilizador pode escolher entre representar, no eixo das abcissas, a distância ao eixo y ( $x$ ) ou a razão entre a distância ao eixo y e o tamanho da carga distribuída ( $x/a$ ). No caso do eixo das ordenadas o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ).

De referir que, a qualquer instante, o utilizador pode voltar ao menu principal do programa clicando no botão "Voltar ao menu inicial".



## 5. EXEMPLOS DE APLICAÇÃO

Neste capítulo serão apresentados dois exemplos, dizendo um respeito à consolidação unidimensional (Figura 8) e outro à consolidação bidimensional (Figura 9). São mostrados os valores usados nos parâmetros da interface de introdução de dados básica e os resultados que podem ser vistos na interface de saída de resultados.

### 5.1. Exemplo de consolidação unidimensional

Este exemplo de consolidação unidimensional foi baseado no caso estudado por Reed (1984) e por Kanok-Nukulchai e Suaris (1982). De notar que os autores não apresentam unidades para os parâmetros de entrada. No entanto, o programa permite a introdução de qualquer sistema coerente de unidades, apesar de serem indicadas nos menus as unidades a introduzir. Pelo valor usado para o peso específico da água presume-se que os autores usaram toneladas em vez de kN. Apesar de não ser demonstrado nesta dissertação, observou-se que os resultados obtidos através do programa são semelhantes aos resultados dos autores referidos.

Na interface de introdução de dados básica, é escolhida a opção consolidação unidimensional e definido um valor de "a" da carga distribuída igual a 1 metro (Figura 7). Nos parâmetros de controlo (Figura 10) são definidos os seguintes valores:

- 39 nós onde há graus de liberdade restringidos ("NVFIX=39");
- 3 nós onde se prescreve a pressão neutra ("NPRFX=3");
- 1 tipo de material ("NMATS=1");
- 2 pontos de Gauss em cada direção ("NGAUS=2");
- Tipo de elementos tendo todos os seus nós com deslocamentos segundo x e y e ainda a pressão neutra ("TELEM=0");
- 21 incrementos de tempo ("NINCR=21");
- Parâmetro da pressão prescrita igual a zero ("LINCA=0").

Nos parâmetros dos nós restringidos (Figura 12) são definidos os nós laterais (nós número 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46 e 47) como estando apenas impedidos de se deslocarem na horizontal

(segundo x) e são definidos os nós da base (nós número 19, 29 e 48) como estando impedidos de se deslocarem na horizontal e na vertical (segundo x e y), perfazendo assim os 39 nós que foram definidos como tendo graus de liberdade restringidos. Para todos os 39 nós, é ainda definido o valor prescrito da componente x e da componente y do deslocamento nodal como sendo igual a zero metros.

Para os parâmetros dos nós onde se prescreve a pressão neutra (Figura 13) são definidos os 3 nós à superfície (nós número 1, 20 e 30) com o valor prescrito da pressão neutra igual a zero kPa, perfazendo assim os 3 nós onde se prescreve a pressão neutra. No próximo conjunto de parâmetros (Figura 14) são definidos os seguintes valores:

- Alfa,  $\alpha$ , igual a 1 ("AALFA=1");
- Peso específico da água,  $\gamma_w$ , igual a 1 ("GAMAW=1");
- Coeficiente da permeabilidade, k, igual a  $4,0 \times 10^{-6}$  ("PERMB=4.0E-6").

Para os parâmetros dos materiais (Figura 15) são definidos os seguintes valores:

- Módulo de elasticidade, E, igual a 6000 ("PROPS1=6000");
- Coeficiente de Poisson,  $\nu$ , igual a 0,4 ("PROPS2=0.4");
- Espessura do material, t, igual a 1 ("PROPS3=1");
- Densidade do material,  $\rho$ , igual a 0 ("PROPS4=0").

Nos parâmetros do tempo (Figura 16) é definido o primeiro incremento de tempo como sendo igual a  $1,0 \times 10^{-6}$ , seguido de dez incrementos de tempo com o valor de 0,01 e por fim mais dez incrementos de tempo com o valor de 0,1 perfazendo assim os 21 incrementos de tempo com o total de tempo aproximado de 1,1.

Para os parâmetros das cargas (Figura 17) é definido como não havendo forças de gravidade ("IGRAV=0") e tensão inicial ("INIST=0"). Nos parâmetros das cargas distribuídas (Figura 21) é definida a componente normal da carga distribuída com o valor de  $-1 \text{ kN/m}^2$  ("PRESS(1)=-1") e a componente tangencial da carga distribuída com o valor de  $0 \text{ kN/m}^2$  ("PRESS(2)=0"), terminando assim a introdução dos valores dos parâmetros da interface de introdução de dados básica necessários para a criação do ficheiro de dados.



Na interface de saída de resultados, inicialmente surge a representação do problema em análise como mostra a Figura 30.

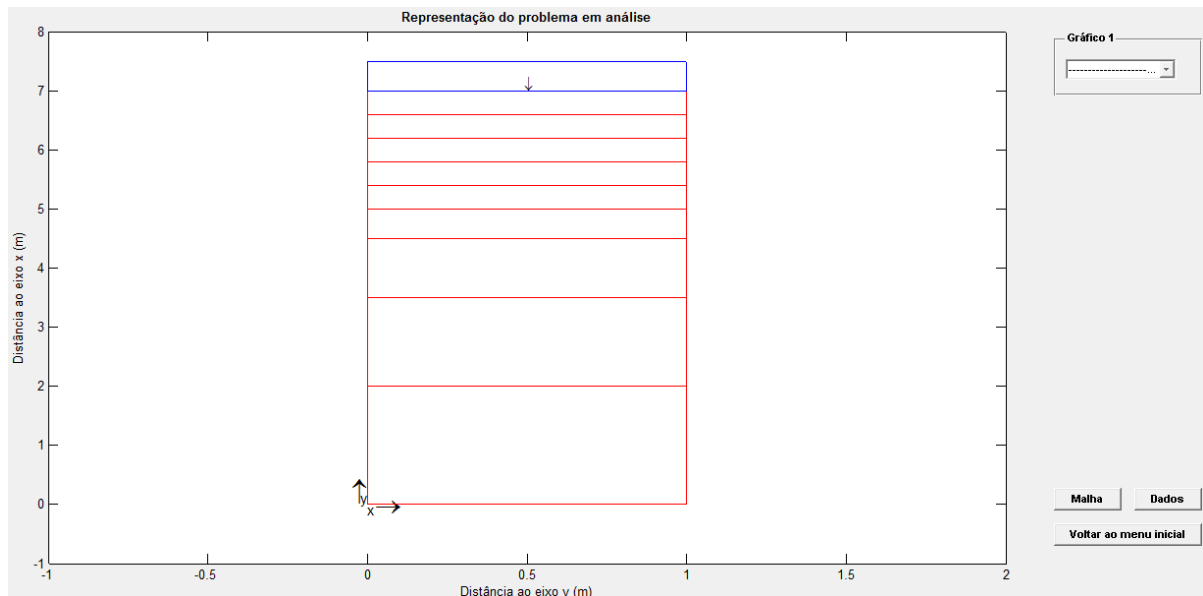


Figura 30: Representação do problema em análise (exemplo de consolidação unidimensional).

Clicando no botão "Malha" surge a representação da malha com a numeração dos nós como mostra a Figura 31.

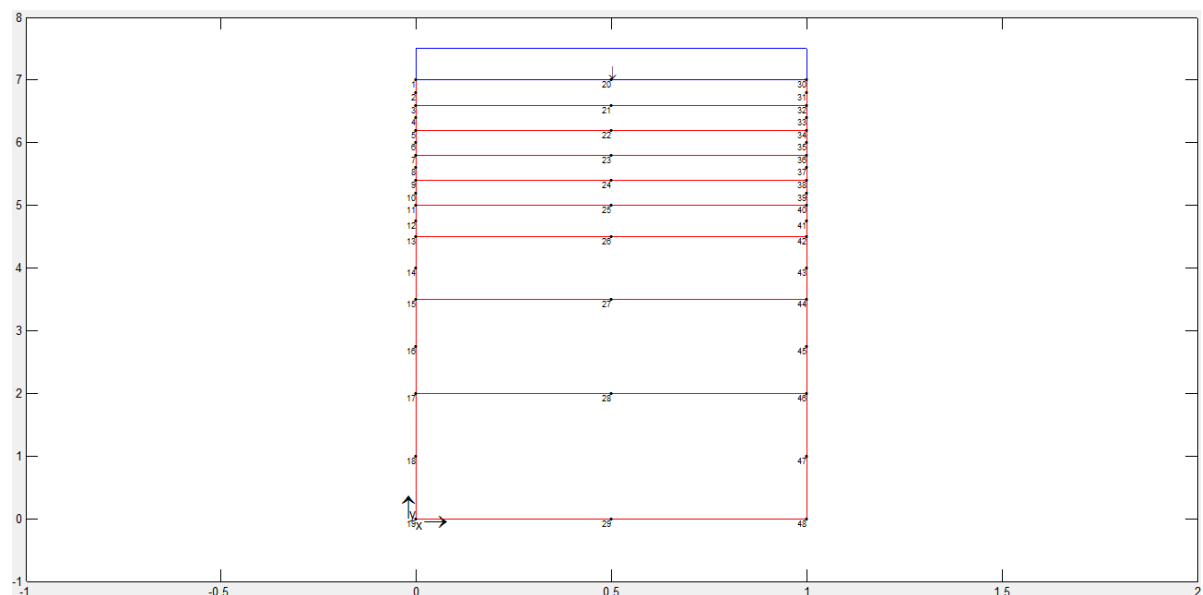


Figura 31: Representação da malha com a numeração dos nós (exemplo de consolidação unidimensional).

Clicando no botão "Dados" o utilizador pode ver as coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo como mostra a Figura 32.

Número do nó	X	Y
1	1.0	0.0
2	2.0	0.0
3	3.0	0.0
4	4.0	0.0
5	5.0	0.0
6	6.0	0.0
7	7.0	0.0
8	8.0	0.0
9	9.0	0.0
10	10.0	0.0
11	11.0	0.0
12	12.0	0.0
13	13.0	0.0
14	14.0	0.0
15	15.0	0.0
16	16.0	0.0
17	17.0	0.0
18	18.0	0.0
19	19.0	0.0
20	20.0	0.5
21	21.0	0.5
22	22.0	0.5
23	23.0	0.5
24	24.0	0.5
25	25.0	0.5
26	26.0	0.5
27	27.0	0.5
28	28.0	0.5
29	29.0	0.5
30	30.0	1.0
31	31.0	1.0
32	32.0	1.0
33	33.0	1.0
34	34.0	1.0
35	35.0	1.0

Número do incremento de tempo	Tempo
1	1.0E-6
2	0.01
3	0.02
4	0.03
5	0.04
6	0.05
7	0.06
8	0.07
9	0.08
10	0.09
11	0.1
12	0.2
13	0.3
14	0.4
15	0.5
16	0.6
17	0.7
18	0.8
19	0.9
20	1.0
21	1.1

Figura 32: Coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo (exemplo de consolidação unidimensional).

De seguida serão mostrados alguns exemplos dos resultados que se podem ver graficamente na interface de saída de resultados. Escolhendo a opção "Deslocamento vertical e altura da coluna de água" nas opções do gráfico1 e deixando as outras opções por defeito, é mostrado no gráfico 1 uma representação dinâmica, ao longo do tempo, do deslocamento vertical de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador e também a altura da coluna de água num tubo colocado à profundidade do nó definido, considerando o nível freático à superfície. Ao mesmo tempo no gráfico 2 é mostrado uma representação dinâmica, ao longo do tempo, da tensão vertical efetiva e total e da pressão neutra no nó definido pelo utilizador como mostra a Figura 33, onde foi definido o nó número 21 (Figura 8).

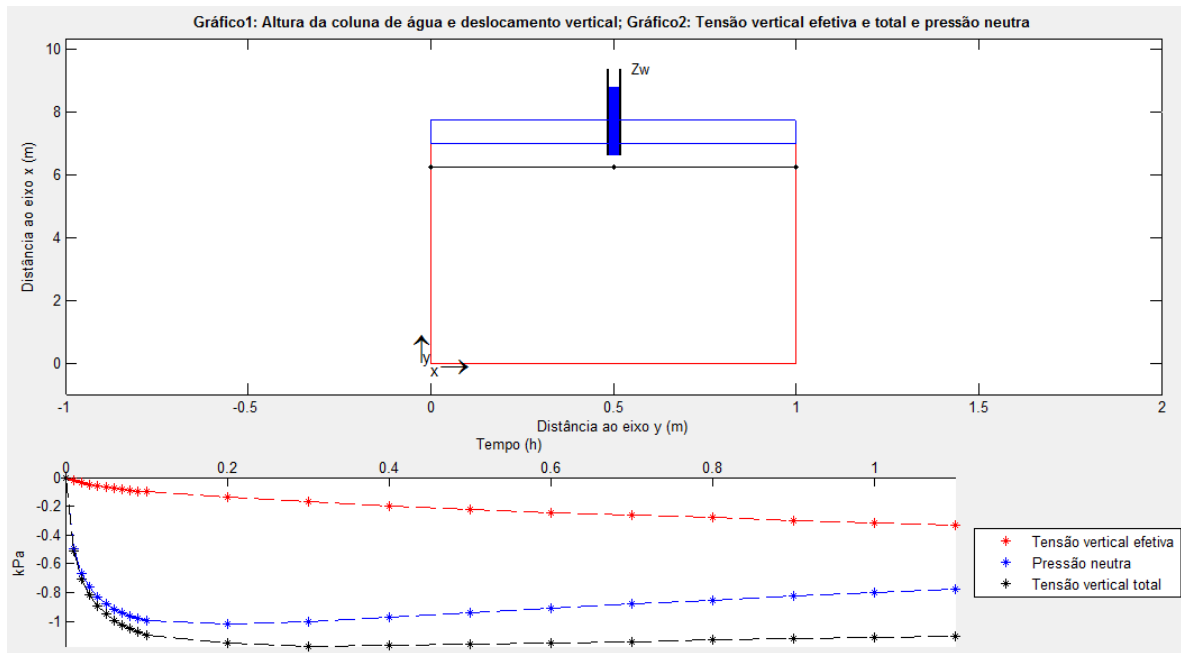


Figura 33: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão vertical efetiva e total (exemplo de consolidação unidimensional).

Nas opções do gráfico 2 o utilizador pode escolher representar a tensão horizontal efetiva e total em vez da tensão vertical efetiva e total.

O utilizador pode ainda adicionar outras representações às referidas anteriormente, nomeadamente vetores do deslocamento, vetores da tensão efetiva, isolinhas do deslocamento e isolinhas da tensão efetiva. No caso dos vetores do deslocamento, as opções disponíveis são entre vertical, horizontal e total sendo mostrado na Figura 34 um exemplo com vetores do deslocamento total nos nós pertencentes à linha horizontal, sendo que neste exemplo foi definido o nó número 20 (Figura 8) e como a pressão neutra à superfície é nula significa que a tensão efetiva e total têm o mesmo valor e são representadas sobrepostas no gráfico 2.

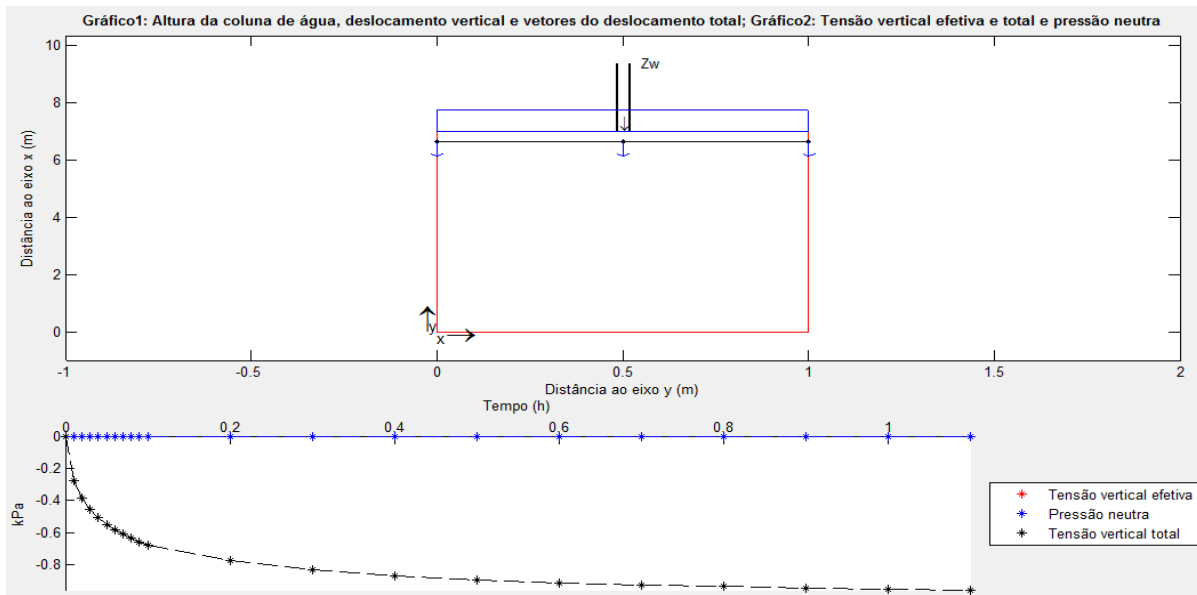


Figura 34: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total nos nós pertencentes à linha horizontal (exemplo de consolidação unidimensional).

No caso dos vetores da tensão efetiva, as opções disponíveis são entre vertical e horizontal sendo mostrado na Figura 35 um exemplo com vetores da tensão vertical efetiva nos nós pertencentes à linha horizontal, sendo que neste exemplo foi definido o nó número 20 (Figura 8).

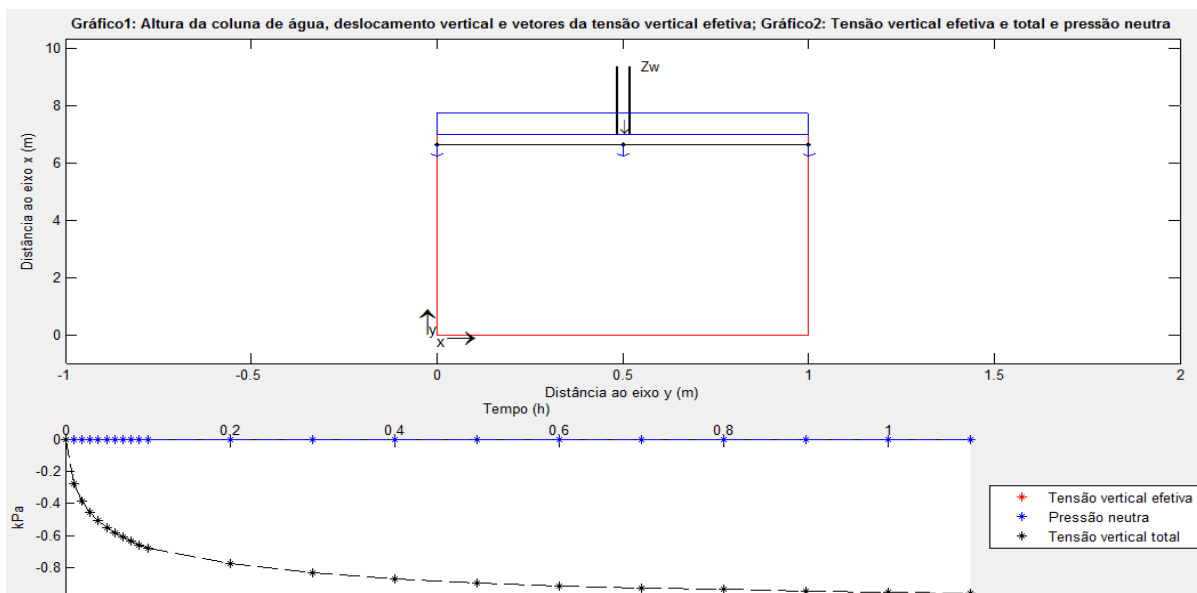


Figura 35: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva nos nós pertencentes à linha horizontal (exemplo de consolidação unidimensional).

No caso das isolinhas do deslocamento, as opções disponíveis são entre vertical e horizontal sendo mostrado na Figura 36 um exemplo com isolinhas do deslocamento vertical, sendo que neste exemplo foi definido o nó número 20 (Figura 8).

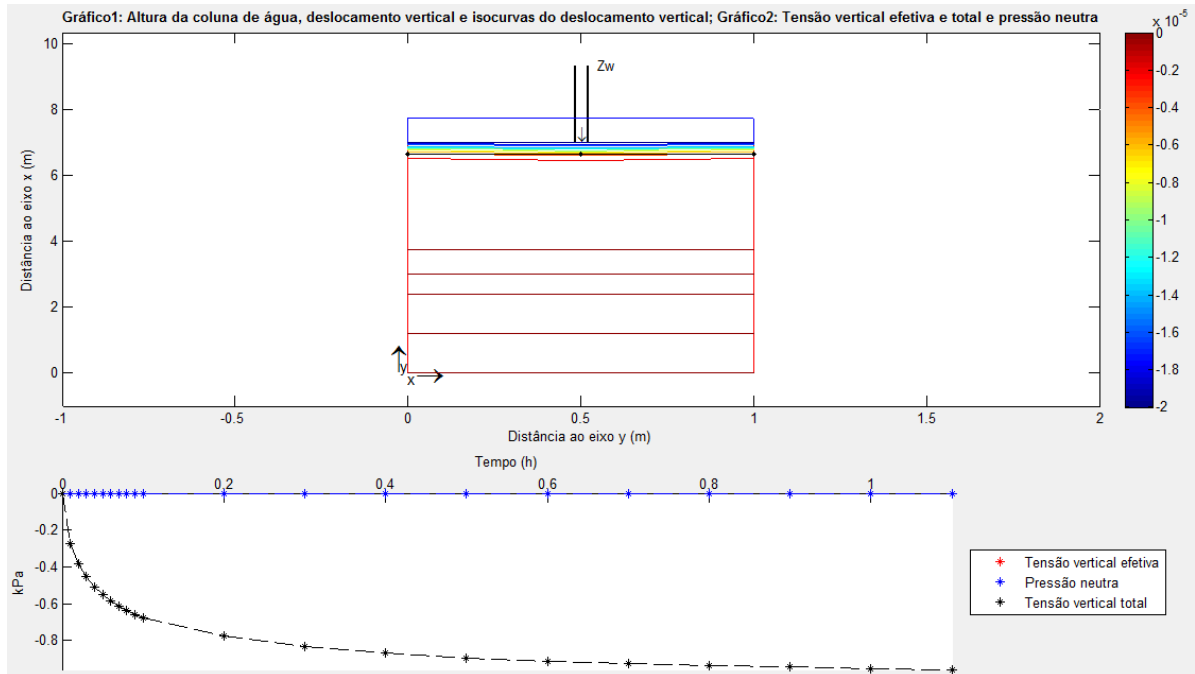


Figura 36: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento vertical (exemplo de consolidação unidimensional).

Por fim, no caso das isolinhas da tensão efetiva, as opções disponíveis são entre vertical, horizontal e de corte ( $\tau_{xy}$ ). Escolhendo a opção "Deslocamento total e altura da coluna de água" nas opções do gráfico 1 é representado o mesmo que escolhendo a opção "Deslocamento vertical e altura da coluna de água", diferindo apenas deste no facto que é representado o deslocamento total de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador. De referir que nas duas opções do gráfico 1 referidas anteriormente o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas do gráfico 2.

A opção "Malha (deslocamento total)" das opções do gráfico 1 mostra uma representação dinâmica, ao longo do tempo, do deslocamento total de todos os nós da malha como mostra a Figura 37.

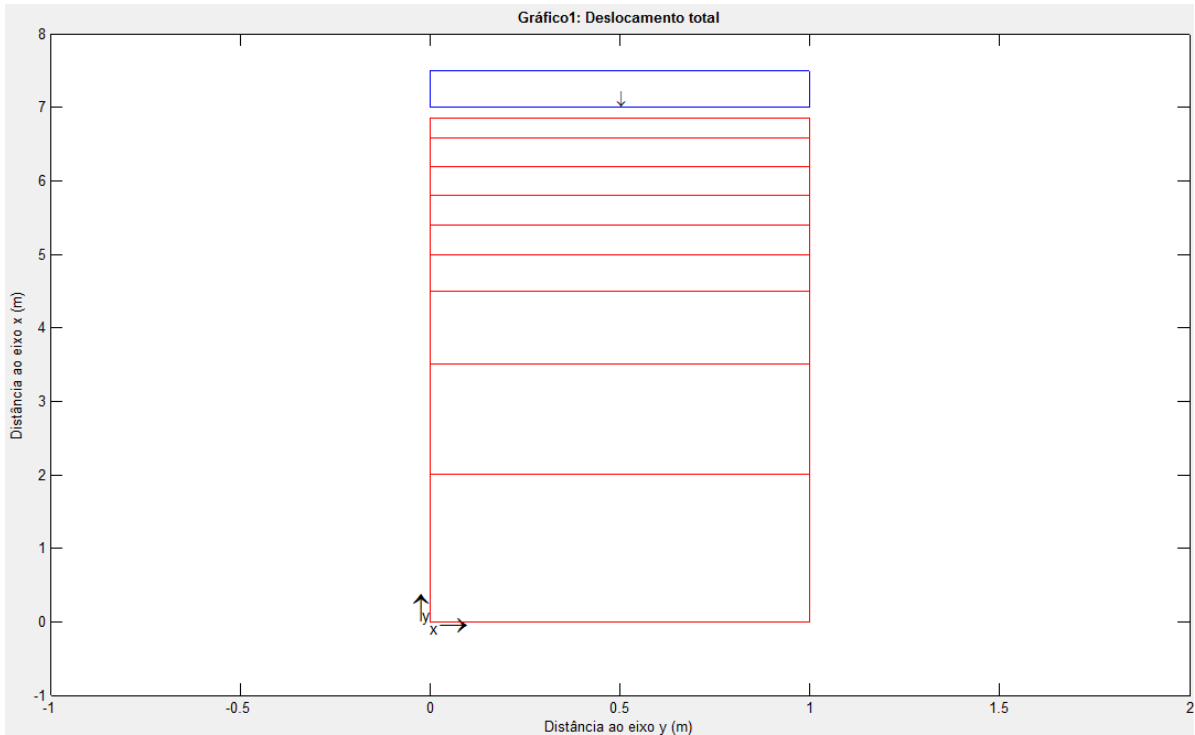


Figura 37: Exemplo da opção "Malha (deslocamento total)" (exemplo de consolidação unidimensional).

Analisando a Figura 37 parece que só existem deslocamentos à superfície, dado ser bem visível um espaço entre a carga e a malha. No entanto, também existem deslocamentos nos outros pontos da malha que só podem ser vistos com a representação dinâmica que se visualiza no computador.

O utilizador pode ainda adicionar outras representações à referida anteriormente, nomeadamente vetores do deslocamento e vetores da tensão efetiva. No caso dos vetores do deslocamento, as opções disponíveis são entre vertical, horizontal e total sendo mostrado na Figura 38 um exemplo com vetores do deslocamento total.

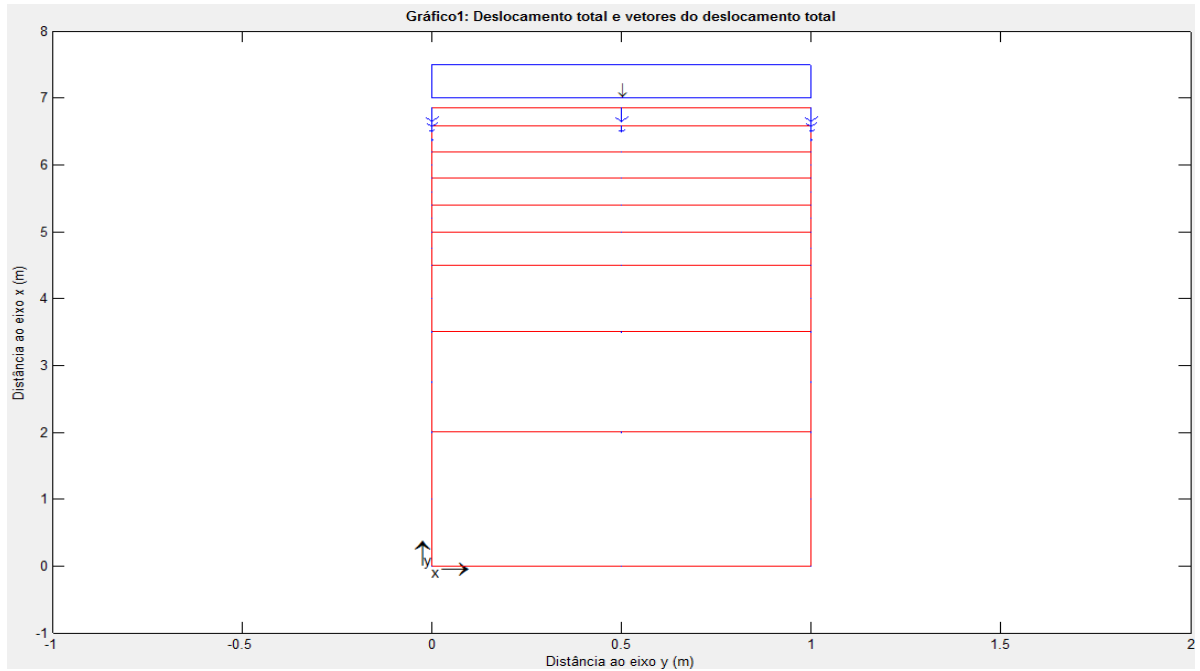


Figura 38: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento total (exemplo de consolidação unidimensional).

No caso dos vetores da tensão efetiva, as opções disponíveis são entre vertical e horizontal sendo mostrado na Figura 39 um exemplo com vetores da tensão vertical efetiva.

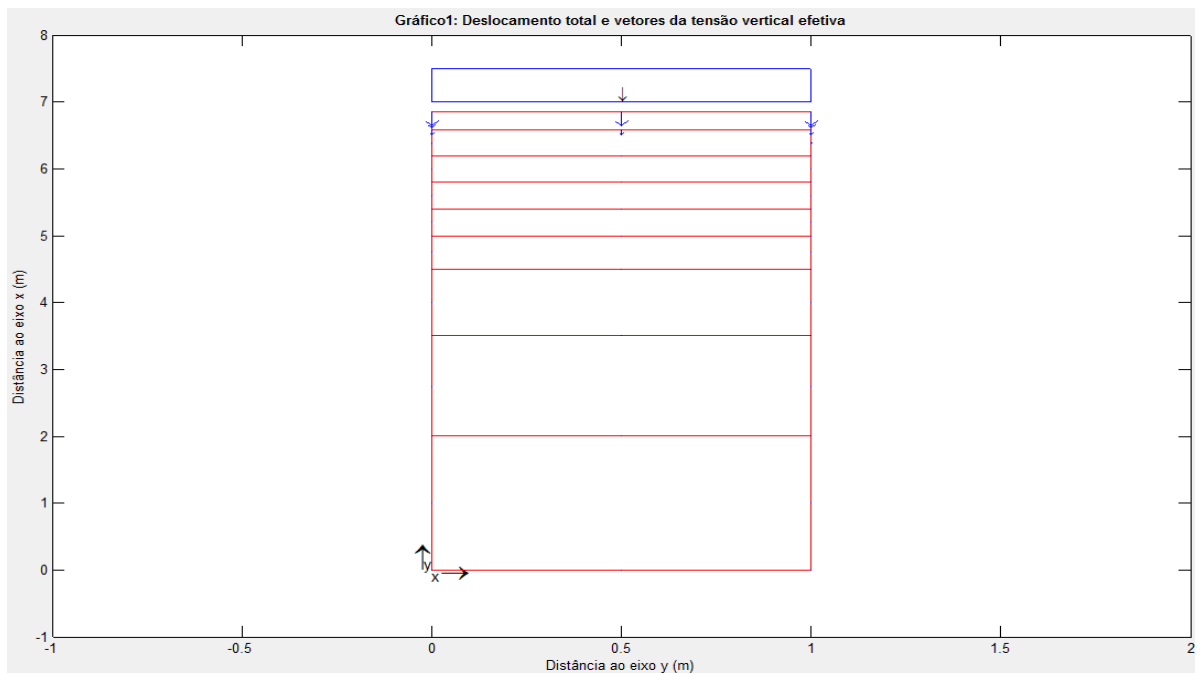


Figura 39: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação unidimensional).

De referir que nas três opções do gráfico 1 referidas anteriormente é possível, a qualquer instante, parar a representação dinâmica clicando no botão "Pausa" e também ver os valores de um conjunto de parâmetros clicando no botão "Resultados" como mostra a Figura 40.

	Número do nó	Deslocamento vertical	Deslocamento horizontal	Tensão vertical efetiva
1	1.0	-2.0659E-5	0.0	-0.9562051082251575
2	2.0	-8.80342E-6	0.0	-0.5679800000000004
3	3.0	-2.98855E-6	0.0	-0.32797100000000423
4	4.0	-8.30529E-7	0.0	-0.0879620000000001
5	5.0	-2.51904E-7	0.0	-0.04769545000000064
6	6.0	-6.49246E-8	0.0	-0.007428900000000016
7	7.0	-2.07775E-8	0.0	-0.0040205750000001...
8	8.0	-4.46118E-9	0.0	-6.122500000000017E-4
9	9.0	-1.7304E-9	0.0	-3.3213390000000069E-4
10	10.0	-3.48503E-10	0.0	-5.201779999999888E-5
11	11.0	-1.12067E-10	0.0	-3.0424300879752894...
12	12.0	-1.96813E-11	0.0	-2.810110000000003E-6
13	13.0	-2.78566E-12	0.0	-1.8697289895415836...
14	14.0	-6.63678E-13	0.0	-4.47965E-8
15	15.0	6.98513E-13	0.0	-2.4178062301731572...
16	16.0	-3.65606E-13	0.0	2.319999999999997E-9
17	17.0	4.2785E-13	0.0	3.1686611916763095E-9
18	18.0	-1.14472E-13	0.0	2.750449999999985E-9
19	19.0	0.0	0.0	-5.693349792171816E-9
20	20.0	-2.0659E-5	4.28266E-20	-0.9562051082251575
21	21.0	-2.98855E-6	-6.64742E-20	-0.3279710000000007
22	22.0	-2.51904E-7	2.34345E-20	-0.04769544999999997
23	23.0	-2.07775E-8	7.8243E-20	-0.004020575000000012
24	24.0	-1.7304E-9	-1.30083E-20	-3.3213390000000034E-4
25	25.0	-1.12067E-10	1.94589E-20	-3.0424300879752894...
26	26.0	-2.78566E-12	-2.03202E-20	-1.8697289895415836...
27	27.0	6.98513E-13	-7.66247E-21	-2.417806230173152E-8
28	28.0	4.2785E-13	-7.97862E-22	3.168661191676301E-9
29	29.0	0.0	0.0	-5.693349792171815E-9
30	30.0	-2.0659E-5	0.0	-0.9562051082251575
31	31.0	-8.80342E-6	0.0	-0.5679800000000004
32	32.0	-2.98855E-6	0.0	-0.3279709999999998
33	33.0	-8.30529E-7	0.0	-0.0879620000000001
34	34.0	-2.51904E-7	0.0	-0.04769544999999953
35	35.0	-6.49246E-8	0.0	-0.007428900000000016

Figura 40: Exemplo dos resultados da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação unidimensional).



Nas opções do gráfico 1, a opção "Deslocamento vertical vs tempo" mostra um gráfico do deslocamento vertical do nó definido pelo utilizador ao longo do tempo como mostra a Figura 41, onde foi definido o nó número 20 (Figura 8).

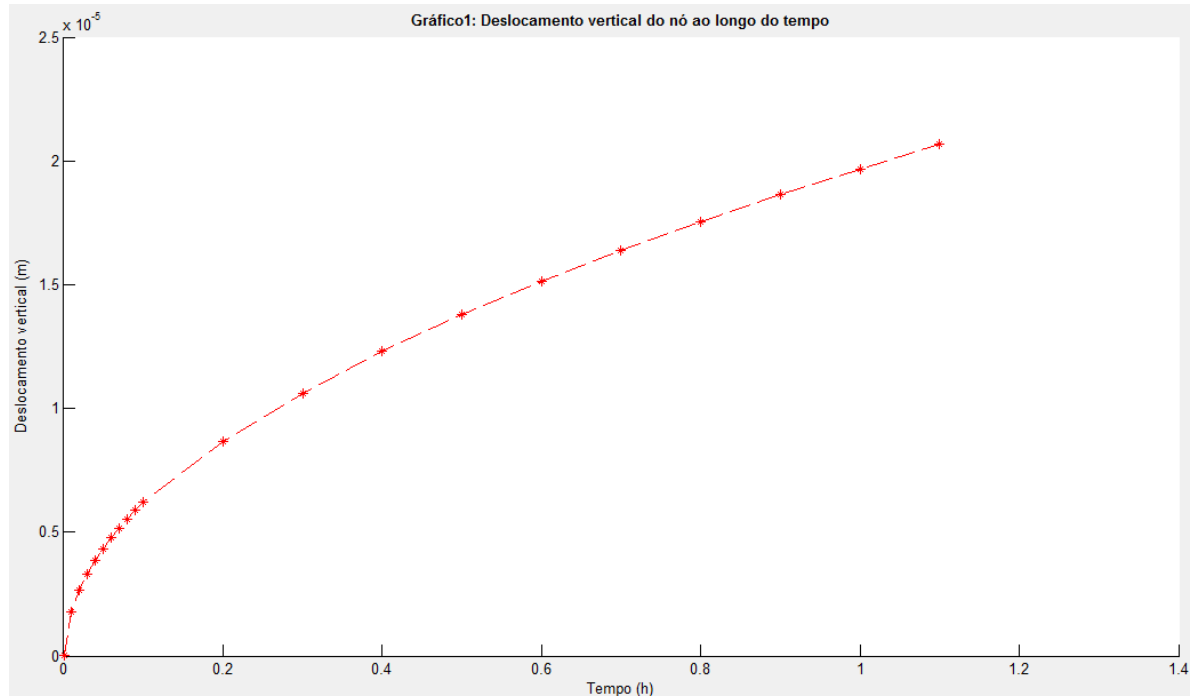


Figura 41: Exemplo da opção "Deslocamento vertical vs tempo" (exemplo de consolidação unidimensional).

Nesta opção o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas. Escolhendo a opção "Deslocamento vertical vs profundidade" nas opções do gráfico 1 é mostrado um gráfico do deslocamento vertical de um conjunto de nós distribuídos numa linha vertical com a mesma abcissa do nó definido pelo utilizador versus a profundidade de cada nó, para o tempo definido como mostra a Figura 42, onde foi definido o nó número 20 (Figura 8) e o incremento de tempo número 21 (Figura 32) que corresponde a um tempo de 1,1.

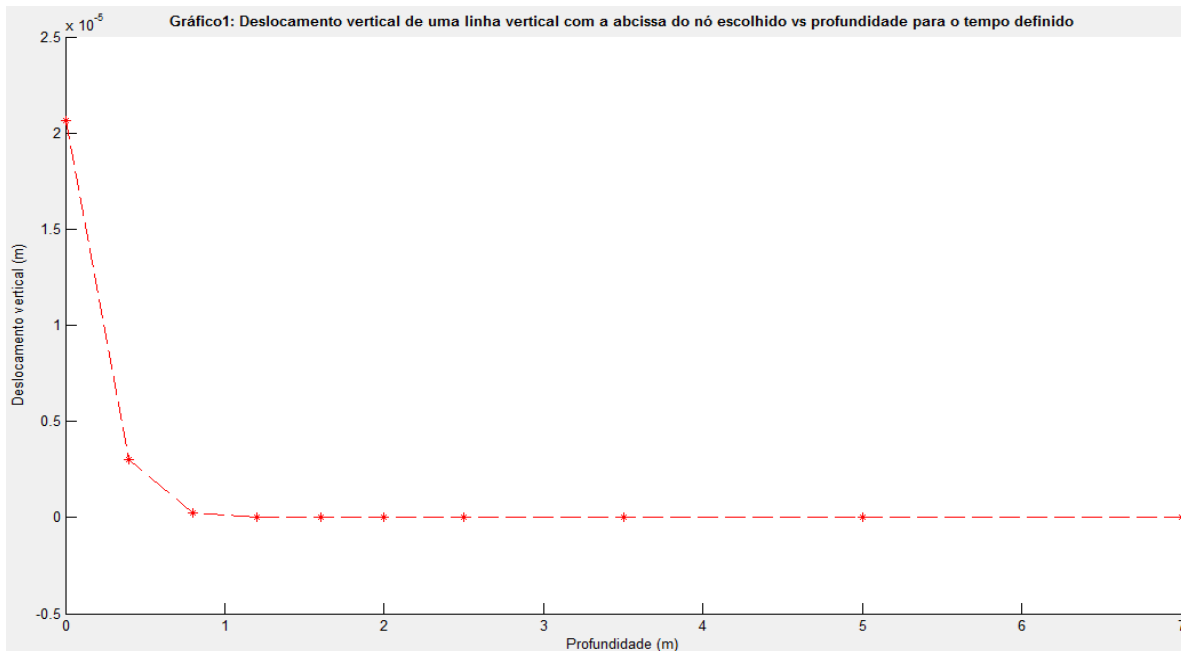


Figura 42: Exemplo da opção "Deslocamento vertical vs profundidade" (exemplo de consolidação unidimensional).

Nas opções do gráfico 1, escolhendo a opção "Pressão neutra vs tempo" é mostrado um gráfico da pressão neutra do nó definido pelo utilizador ao longo do tempo como mostra a Figura 43, onde foi definido o nó número 22 (Figura 8).

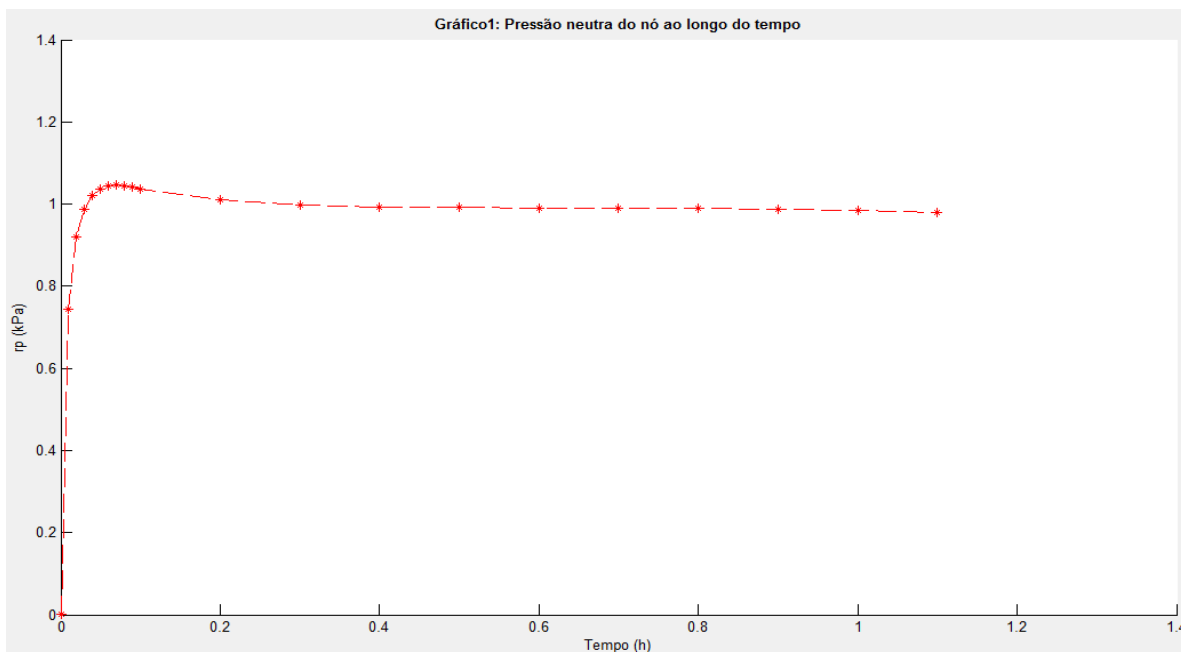


Figura 43: Exemplo da opção "Pressão neutra vs tempo" (exemplo de consolidação unidimensional).

Nesta opção o utilizador pode escolher entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso do eixo das ordenadas o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ). Por fim, escolhendo a opção "Pressão neutra vs profundidade" nas opções do gráfico 1 é mostrado um gráfico da pressão neutra de um conjunto de nós distribuídos numa linha vertical com a mesma abcissa do nó definido versus a profundidade de cada nó, para o tempo definido como mostra a Figura 44, onde foi definido o nó número 20 (Figura 8) e o incremento de tempo número 16 (Figura 32) que corresponde a um tempo de 0,6.

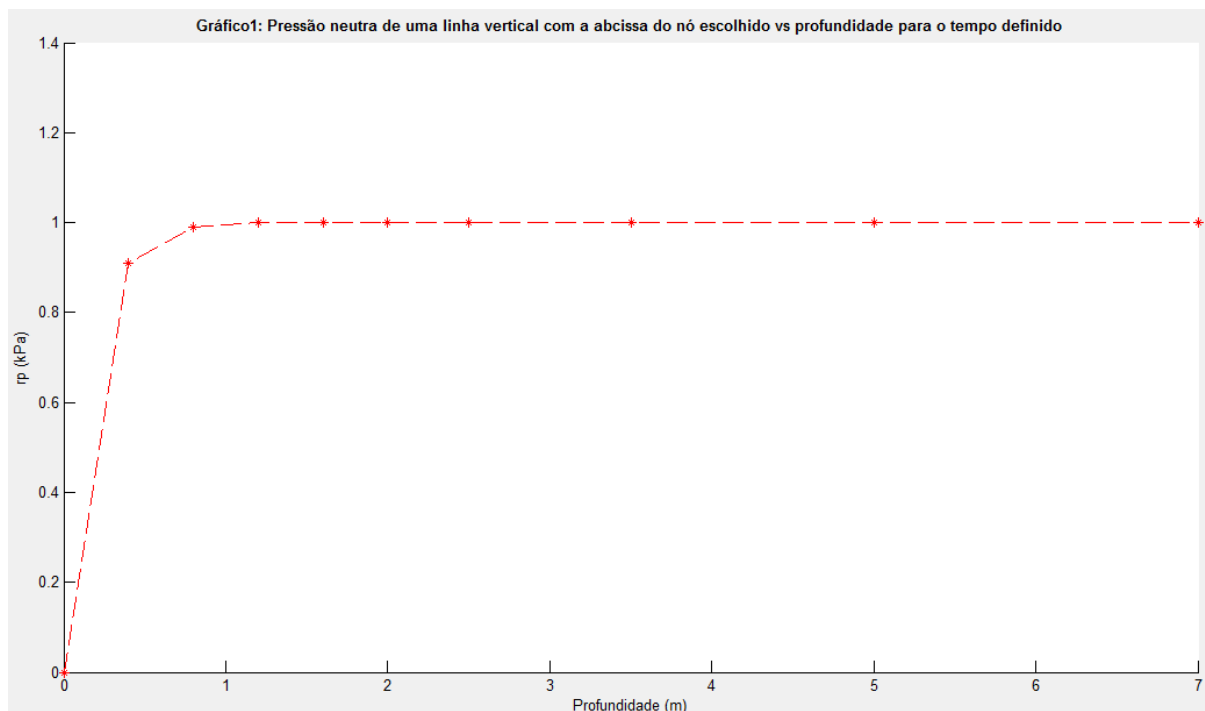


Figura 44: Exemplo da opção "Pressão neutra vs profundidade" (exemplo de consolidação unidimensional).

Nesta opção, para o eixo das ordenadas, o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ).

## 5.2. Exemplo de consolidação bidimensional

Este exemplo de consolidação bidimensional é o mesmo que foi usado no capítulo 4 da validação do programa onde se observou que os resultados obtidos através do programa são semelhantes aos da solução analítica.

Na interface de introdução de dados básica, é escolhida a opção consolidação bidimensional e definido um valor de "a", correspondente à largura de metade da faixa carregada, igual a 1 metro (Figura 7). São definidos os seguintes valores para os parâmetros de controlo (Figura 10):

- 27 nós onde há graus de liberdade restringidos ("NVFIX=27");
- 21 nós onde se prescreve a pressão neutra ("NPRFX=21");
- 1 tipo de material ("NMATS=1");
- 2 pontos de Gauss em cada direção ("NGAUS=2");
- Tipo de elementos tendo todos os seus nós com deslocamentos segundo x e y e ainda a pressão neutra ("TELEM=0");
- 21 incrementos de tempo ("NINCR=21");
- Parâmetro da pressão prescrita igual a zero ("LINCA=0").

Nos parâmetros dos nós restringidos (Figura 12) são definidos os nós laterais (nós número 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 102, 103, 104, 105, 106 e 107) como estando apenas impedidos de se deslocarem horizontalmente e são definidos os nós da base (nós número 13, 20, 54, 69, 84, 90, 97, 101 e 108) como estando impedidos de se deslocarem na horizontal e na vertical, perfazendo assim os 27 nós que foram definidos como tendo graus de liberdade restringidos. Para todos os 27 nós, é ainda definido o valor prescrito da componente x e da componente y do deslocamento nodal como sendo igual a zero metros.

Para os parâmetros dos nós onde se prescreve a pressão neutra (Figura 13) são definidos os nós da superfície e os do lado direito (nós número 1, 14, 21, 30, 36, 45, 55, 63, 70, 78, 85, 87, 91, 98, 102, 103, 104, 105, 106, 107 e 108) com o valor prescrito da pressão neutra igual a zero kPa, perfazendo assim os 21 nós onde se prescreve a pressão neutra.

No conjunto seguinte de parâmetros (Figura 14) são definidos os seguintes valores:

- Alfa,  $\alpha$ , igual a 0,666667 ("AALFA=0.666667");
- Peso específico da água,  $\gamma_w$ , igual a 9,8 kN/m<sup>3</sup> ("GAMAW=9.8");
- Coeficiente da permeabilidade, k, igual a 7,0x10<sup>-4</sup> m/h ("PERMB=7.0E-4").

Para os parâmetros dos materiais (Figura 15) são definidos os seguintes valores:

- Módulo de elasticidade, E, igual a 15000 kN/m<sup>2</sup> ("PROPS1=15000");
- Coeficiente de Poisson,  $\nu$ , igual a 0 ("PROPS2=0");
- Espessura do material, t, igual a 1 m ("PROPS3=1");
- Densidade do material,  $\rho$ , igual a 0 kg/m<sup>3</sup> ("PROPS4=0").

Nos parâmetros do tempo (Figura 16) é definido o primeiro incremento de tempo como sendo igual a 1,0x10<sup>-6</sup>, seguido de dez incrementos de tempo com o valor de 0,01 e por fim mais dez incrementos de tempo com o valor de 0,1 perfazendo assim os 21 incrementos de tempo com o total de tempo aproximado de 1,1 horas.

Para os parâmetros das cargas (Figura 17) são consideradas nulas quer as forças de gravidade ("IGRAV=0") quer as tensões iniciais ("INIST=0"). Nos parâmetros das cargas distribuídas (Figura 21) é definida a componente normal da carga distribuída com o valor de -1 kN/m<sup>2</sup> ("PRESS(1)=-1") e a componente tangencial da carga distribuída com o valor de 0 kN/m<sup>2</sup> ("PRESS(2)=0"), terminando assim a introdução dos valores dos parâmetros da interface de introdução de dados básica necessários para a criação do ficheiro de dados.

Na interface de saída de resultados, é apresentada inicialmente a representação do problema em análise como mostra a Figura 45.

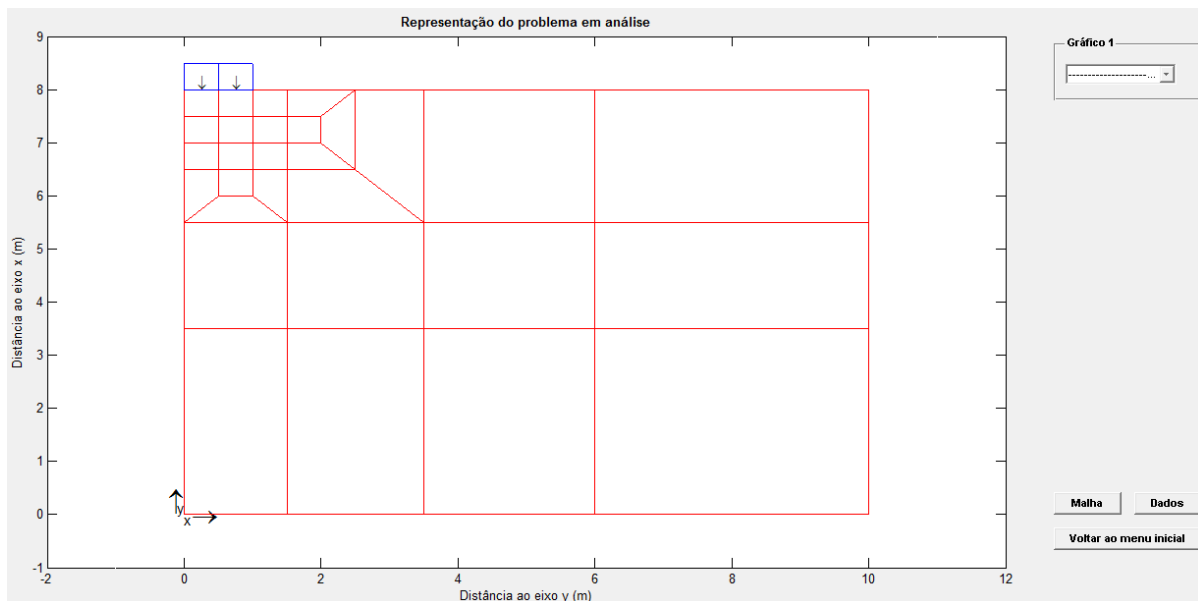


Figura 45: Representação do problema em análise (exemplo de consolidação bidimensional).

Clicando no botão "Malha" surge a representação da malha com a numeração dos nós como mostra a Figura 46.

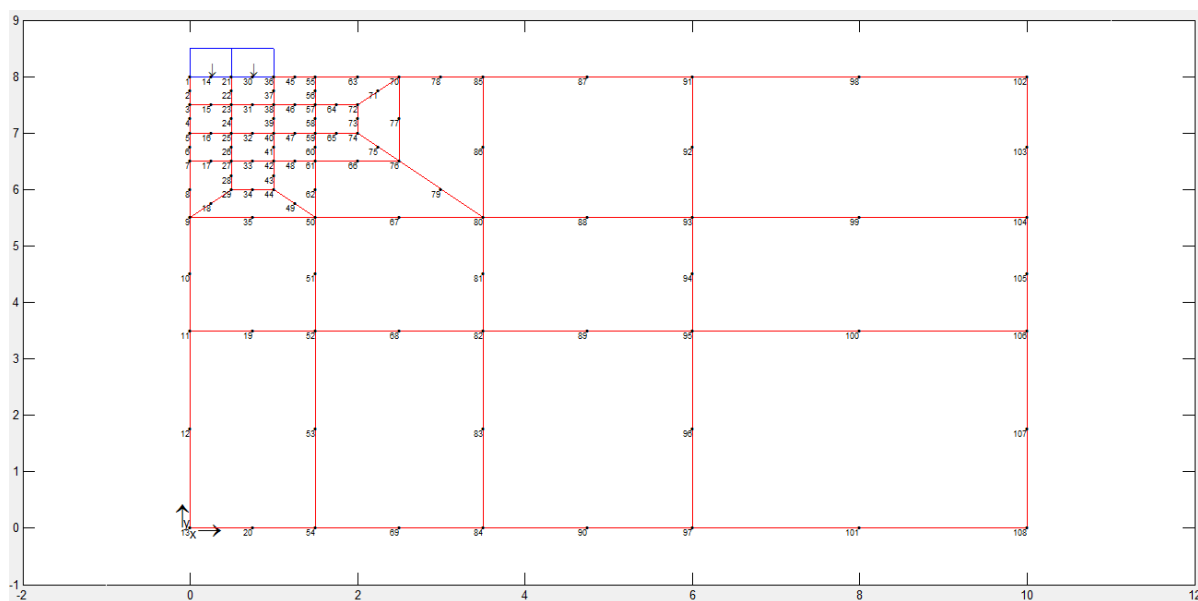


Figura 46: Representação da malha com a numeração dos nós (exemplo de consolidação bidimensional).

Clicando no botão "Dados" o utilizador pode ver as coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo como mostra a Figura 47.

	Número do nó	X	Y
1	1.0	0.0	8.0
2	2.0	0.0	7.75
3	3.0	0.0	7.5
4	4.0	0.0	7.25
5	5.0	0.0	7.0
6	6.0	0.0	6.75
7	7.0	0.0	6.5
8	8.0	0.0	6.0
9	9.0	0.0	5.5
10	10.0	0.0	4.5
11	11.0	0.0	3.5
12	12.0	0.0	1.75
13	13.0	0.0	0.0
14	14.0	0.25	8.0
15	15.0	0.25	7.5
16	16.0	0.25	7.0
17	17.0	0.25	6.5
18	18.0	0.25	5.75
19	19.0	0.75	3.5
20	20.0	0.75	0.0
21	21.0	0.5	8.0
22	22.0	0.5	7.75
23	23.0	0.5	7.5
24	24.0	0.5	7.25
25	25.0	0.5	7.0
26	26.0	0.5	6.75
27	27.0	0.5	6.5
28	28.0	0.5	6.25
29	29.0	0.5	6.0
30	30.0	0.75	8.0
31	31.0	0.75	7.5
32	32.0	0.75	7.0
33	33.0	0.75	6.5
34	34.0	0.75	6.0
35	35.0	0.75	5.5

	Número do incremento de tempo	Tempo
1	1.0	1.0E-6
2	2.0	0.01
3	3.0	0.02
4	4.0	0.03
5	5.0	0.04
6	6.0	0.05
7	7.0	0.06
8	8.0	0.07
9	9.0	0.08
10	10.0	0.09
11	11.0	0.1
12	12.0	0.2
13	13.0	0.3
14	14.0	0.4
15	15.0	0.5
16	16.0	0.6
17	17.0	0.7
18	18.0	0.8
19	19.0	0.9
20	20.0	1.0
21	21.0	1.1

Figura 47: Coordenadas dos nós e o número do incremento de tempo a que corresponde cada valor de tempo (exemplo de consolidação bidimensional).

De seguida serão mostrados alguns exemplos dos resultados que se podem ver graficamente na interface de saída de resultados. Escolhendo a opção "Deslocamento vertical e altura da coluna de água" nas opções do gráfico1 e deixando as outras opções por defeito, é mostrado no gráfico 1 uma representação dinâmica, ao longo do tempo, do deslocamento vertical de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador e também a altura da coluna de água num tubo colocado à profundidade do referido nó, considerando o nível freático à superfície. Ao mesmo tempo no gráfico 2 é mostrado uma representação dinâmica, ao longo do tempo, da tensão vertical efetiva e total e da pressão neutra no nó definido como mostra a Figura 48, onde foi definido o nó número 5 (Figura 9).

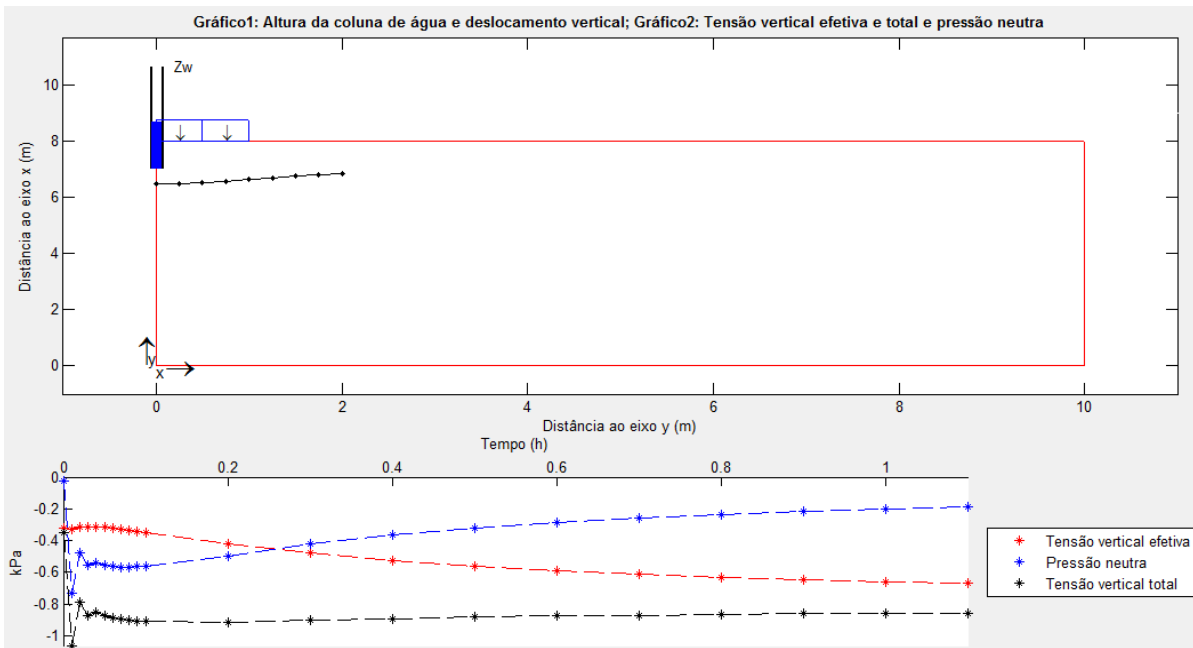


Figura 48: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão vertical efetiva e total (exemplo de consolidação bidimensional).

Nas opções do gráfico 2 o utilizador pode escolher representar a tensão horizontal efetiva e total em vez da tensão vertical efetiva e total como mostra a Figura 49, onde também foi definido o nó número 5 (Figura 9).

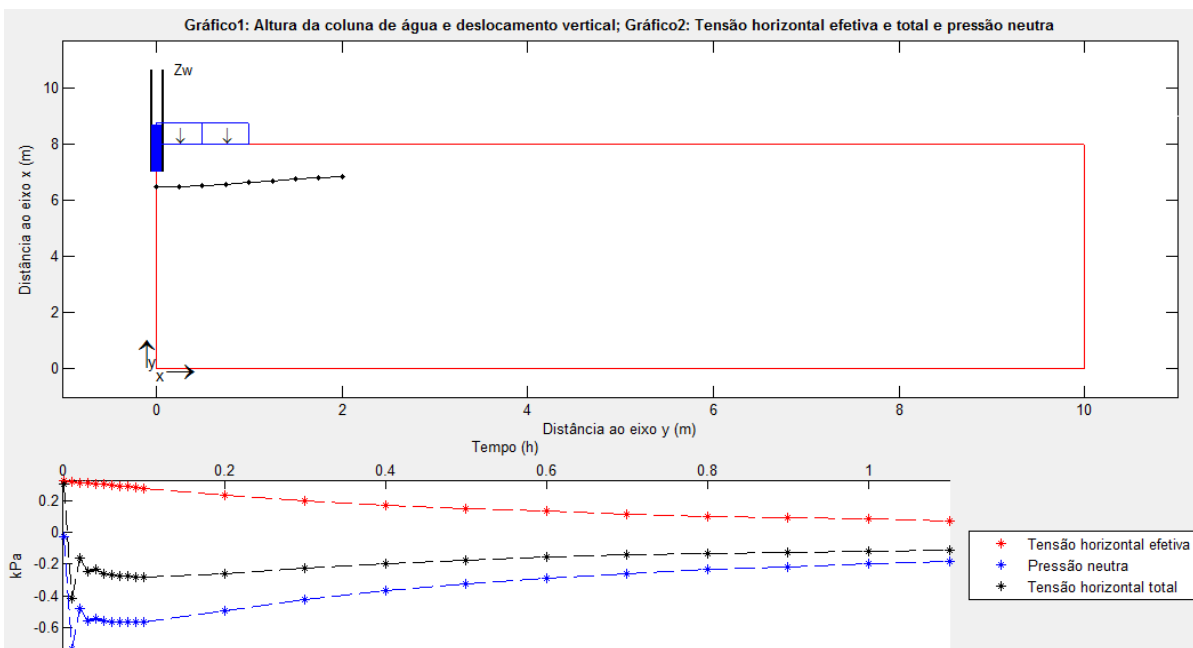


Figura 49: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com representação da tensão horizontal efetiva e total (exemplo de consolidação bidimensional).



O utilizador pode ainda adicionar outras representações às referidas anteriormente, nomeadamente vetores do deslocamento, vetores da tensão efetiva, isolinhas do deslocamento e isolinhas da tensão efetiva. No caso dos vetores do deslocamento, as opções disponíveis são entre vertical, horizontal e total sendo mostrado nas Figuras 50 a 53 exemplos com vetores do deslocamento vertical e com vetores do deslocamento total, sendo que nestes exemplos foi definido o nó número 1 (Figura 9) e como a pressão neutra à superfície é nula significa que a tensão efetiva e total têm o mesmo valor e são representadas sobrepostas no gráfico 2.

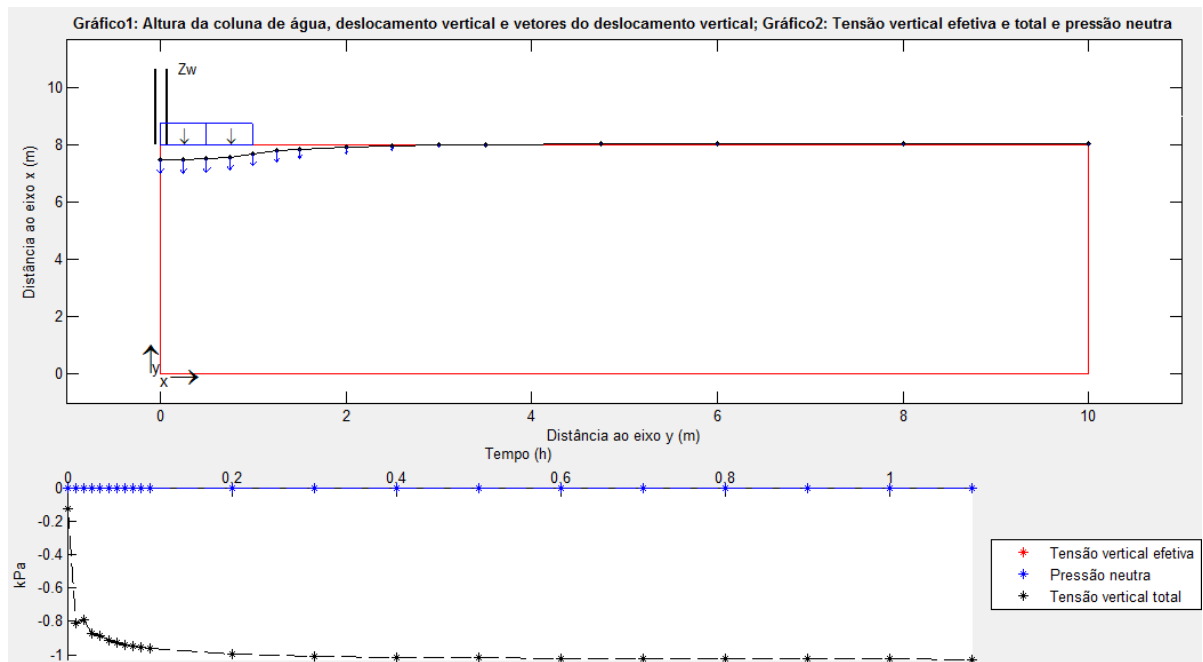


Figura 50: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento vertical nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional).

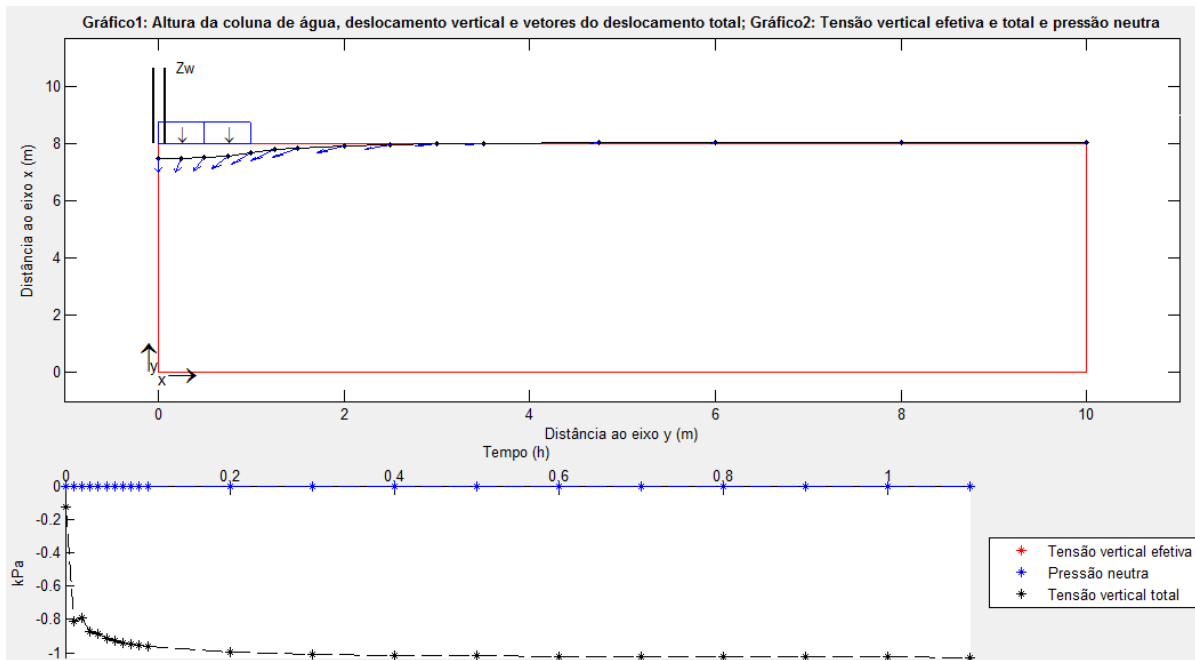


Figura 51: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional).

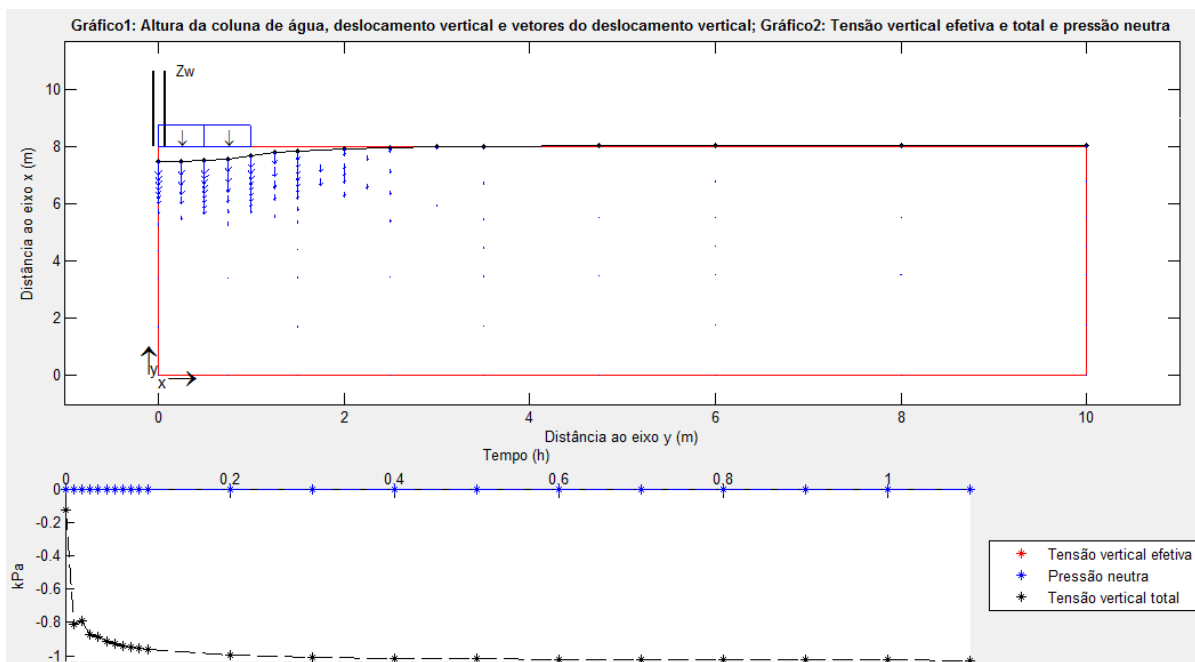


Figura 52: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento vertical em todos os nós (exemplo de consolidação bidimensional).

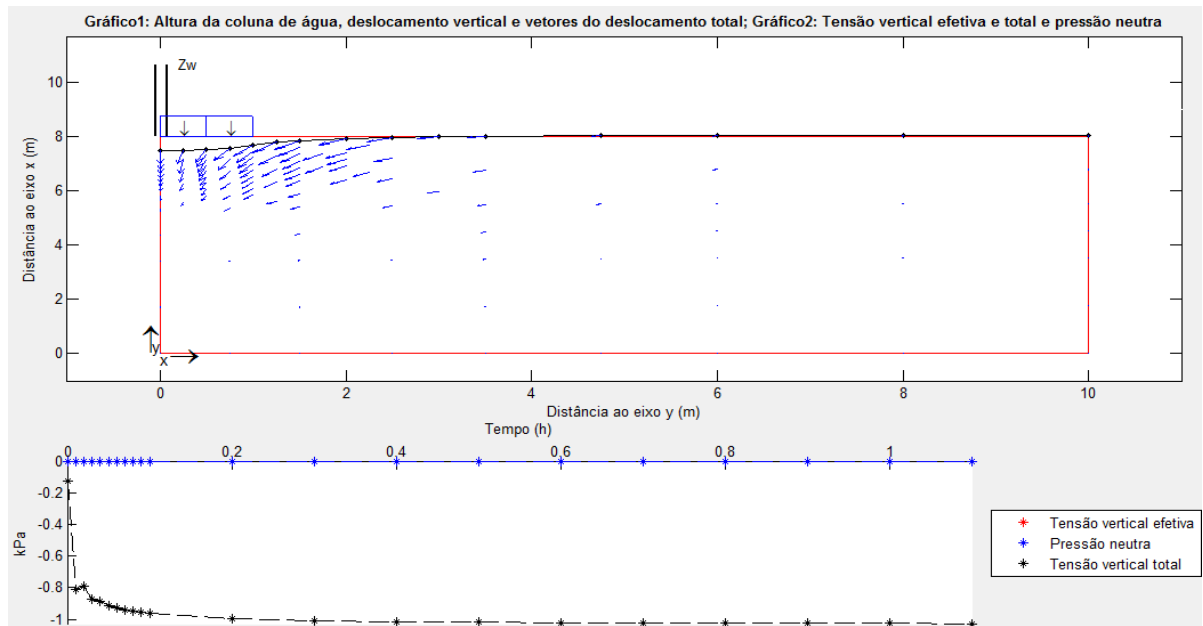


Figura 53: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores do deslocamento total em todos os nós (exemplo de consolidação bidimensional).

No caso dos vetores da tensão efetiva, as opções disponíveis são entre vertical e horizontal sendo mostrado nas Figuras 54 e 55 exemplos com vetores da tensão vertical efetiva, onde foi definido o nó número 1 (Figura 9).

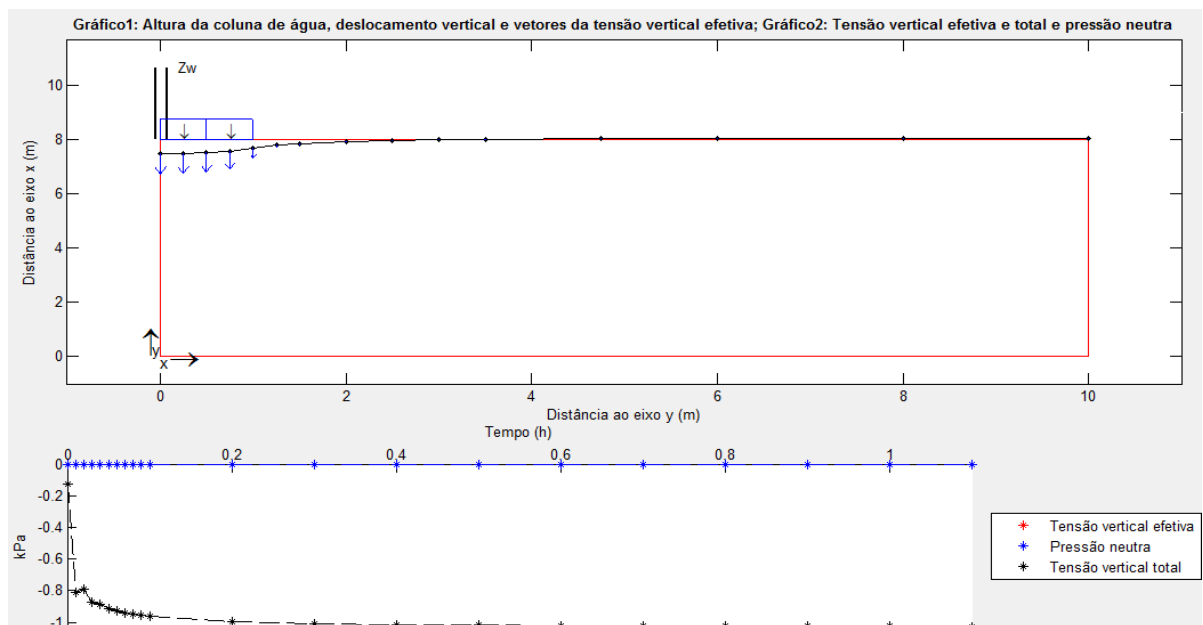


Figura 54: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva nos nós pertencentes à linha horizontal (exemplo de consolidação bidimensional).

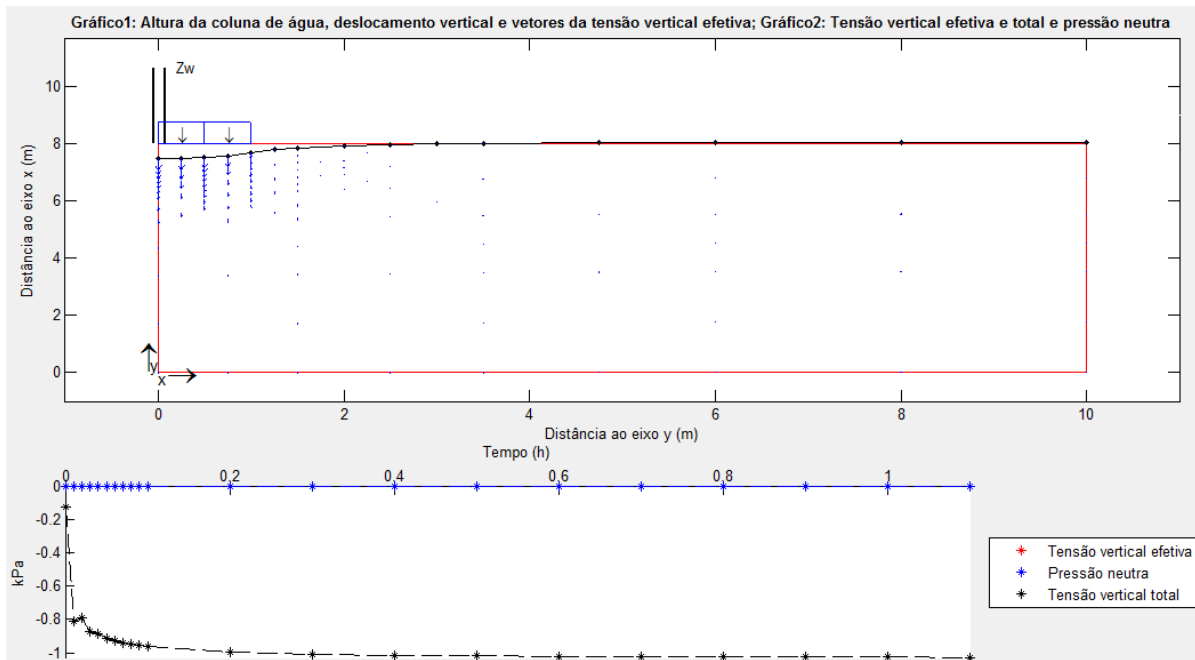


Figura 55: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com vetores da tensão vertical efetiva em todos os nós (exemplo de consolidação bidimensional).

No caso das isolinhas do deslocamento, as opções disponíveis são entre vertical e horizontal como mostra as Figuras 56 e 57, onde foi definido o nó número 95 (Figura 9) apenas com o objetivo de se poder visualizar melhor as isolinhas.

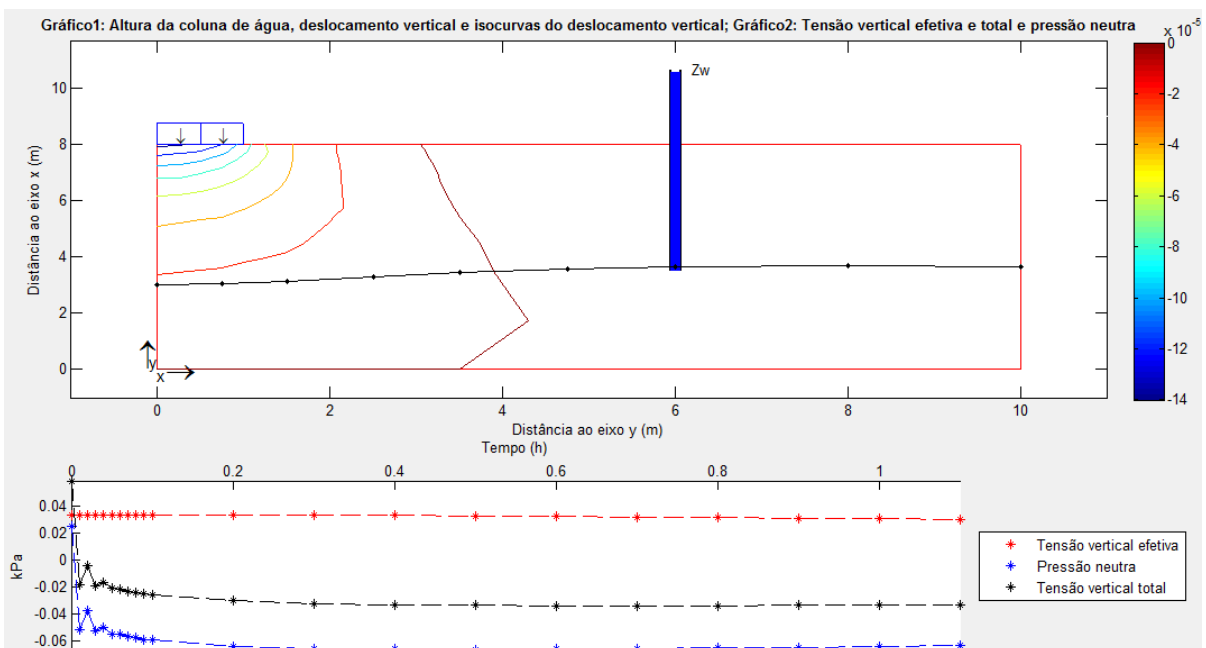


Figura 56: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento vertical (exemplo de consolidação bidimensional).

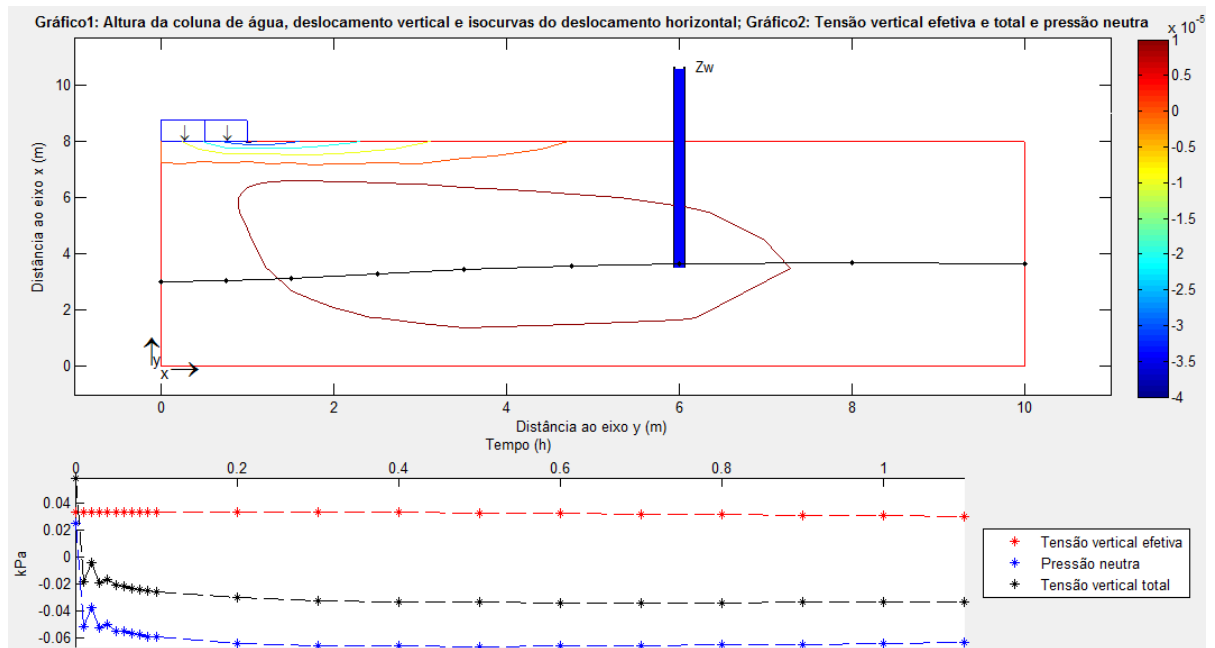


Figura 57: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas do deslocamento horizontal (exemplo de consolidação bidimensional).

Por fim, no caso das isolinhas da tensão efetiva, as opções disponíveis são entre vertical, horizontal e de corte ( $\tau_{xy}$ ) como mostra as Figuras 58 a 60, onde também foi definido o nó número 95 (Figura 9).

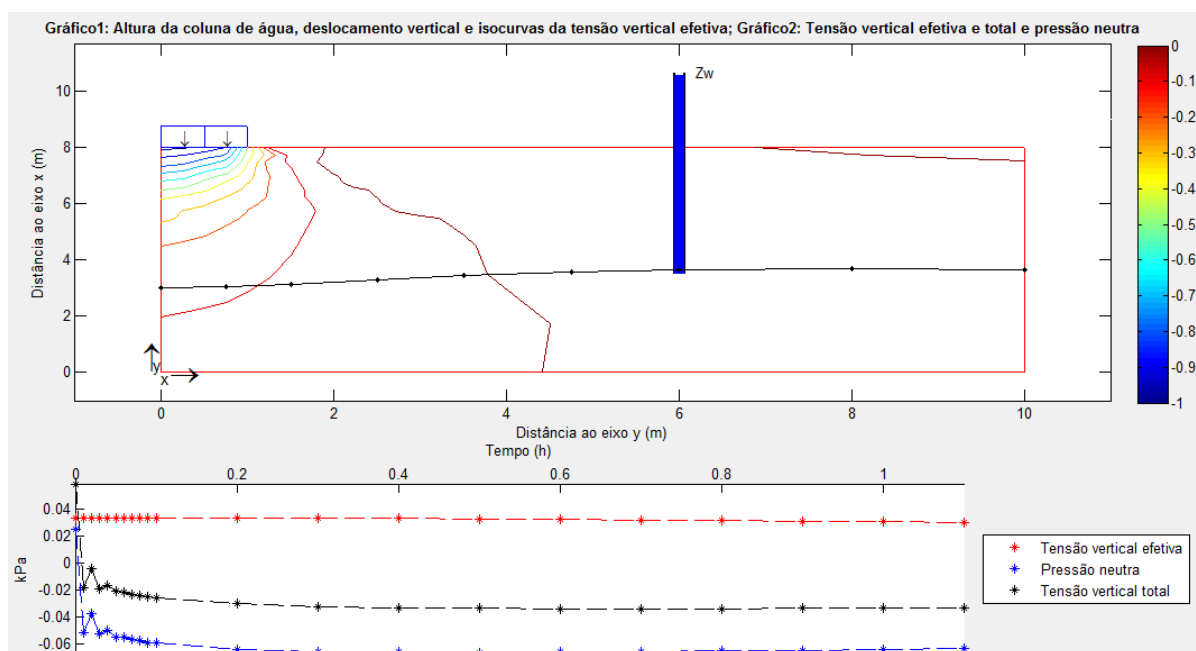


Figura 58: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão vertical efetiva (exemplo de consolidação bidimensional).

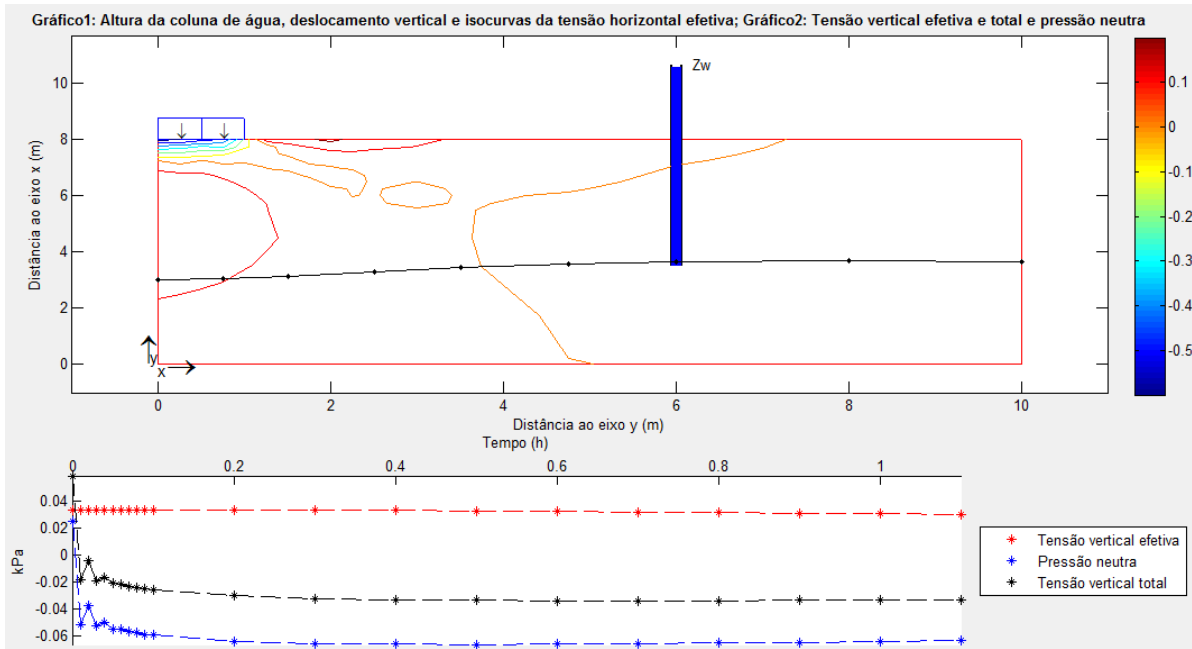


Figura 59: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão horizontal efetiva (exemplo de consolidação bidimensional).

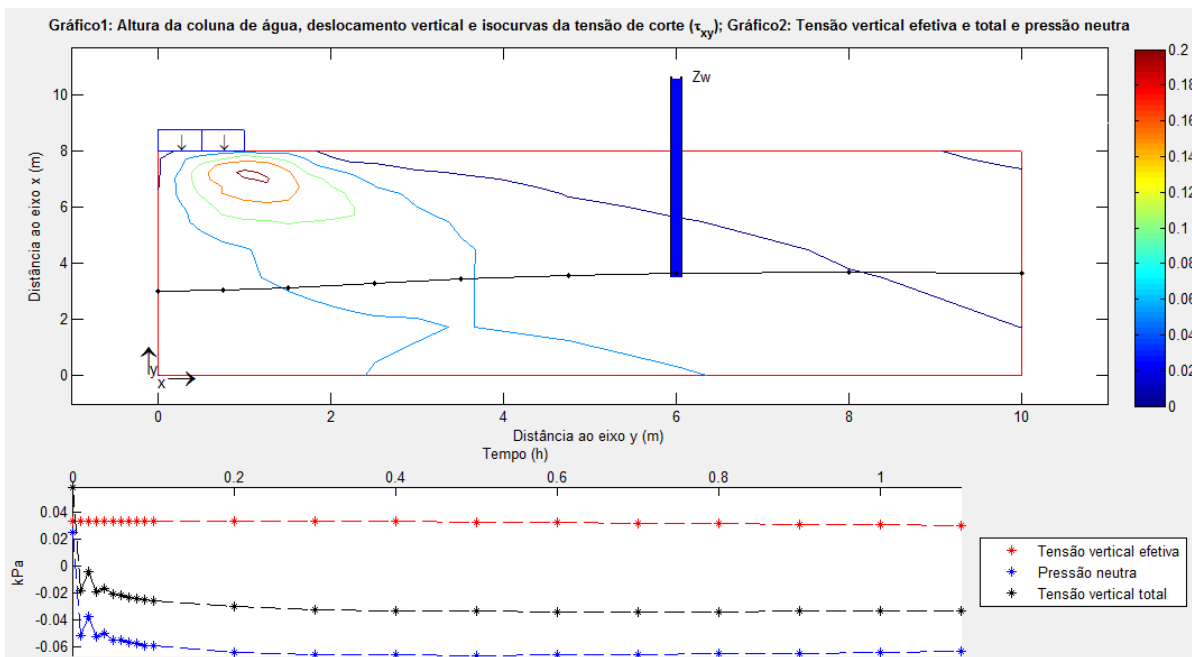


Figura 60: Exemplo da opção "Deslocamento vertical e altura da coluna de água" com isolinhas da tensão de corte ( $\tau_{xy}$ ) (exemplo de consolidação bidimensional).

Escolhendo a opção "Deslocamento total e altura da coluna de água" nas opções do gráfico 1 obtêm-se saídas gráficas semelhantes às obtidas na opção "Deslocamento vertical e altura da coluna de água", diferindo apenas desta no facto de passar a ser representado o deslocamento total de um conjunto de nós distribuídos numa linha horizontal à cota do nó definido pelo utilizador, como mostra a Figura 61, onde foi definido o nó número 1 (Figura 9).

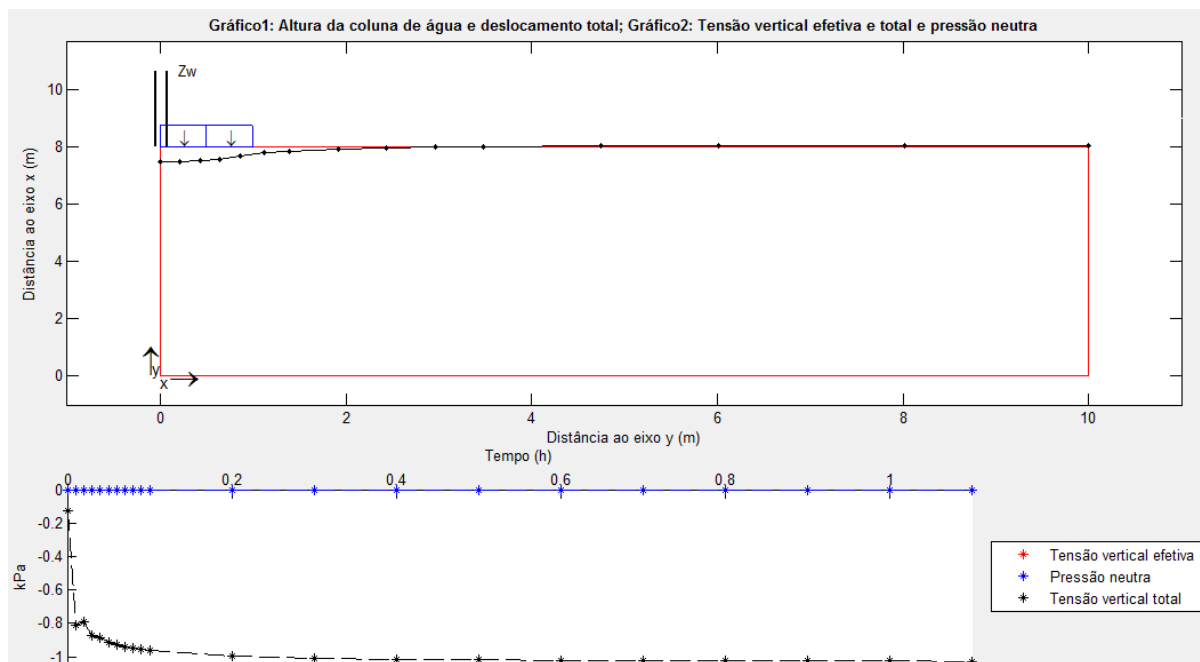


Figura 61: Exemplo da opção "Deslocamento total e altura da coluna de água" (exemplo de consolidação bidimensional).

De referir que nas duas opções do gráfico 1 referidas anteriormente o utilizador pode escolher entre usar o parâmetro "tempo" ou um "fator tempo T" (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas do gráfico 2.

A opção "Malha (deslocamento total)" das opções do gráfico 1 mostra uma representação dinâmica, ao longo do tempo, do deslocamento total de todos os nós da malha. Na Figura 62 é possível observar a malha deformada no final do tempo total atribuído no exemplo apresentado.

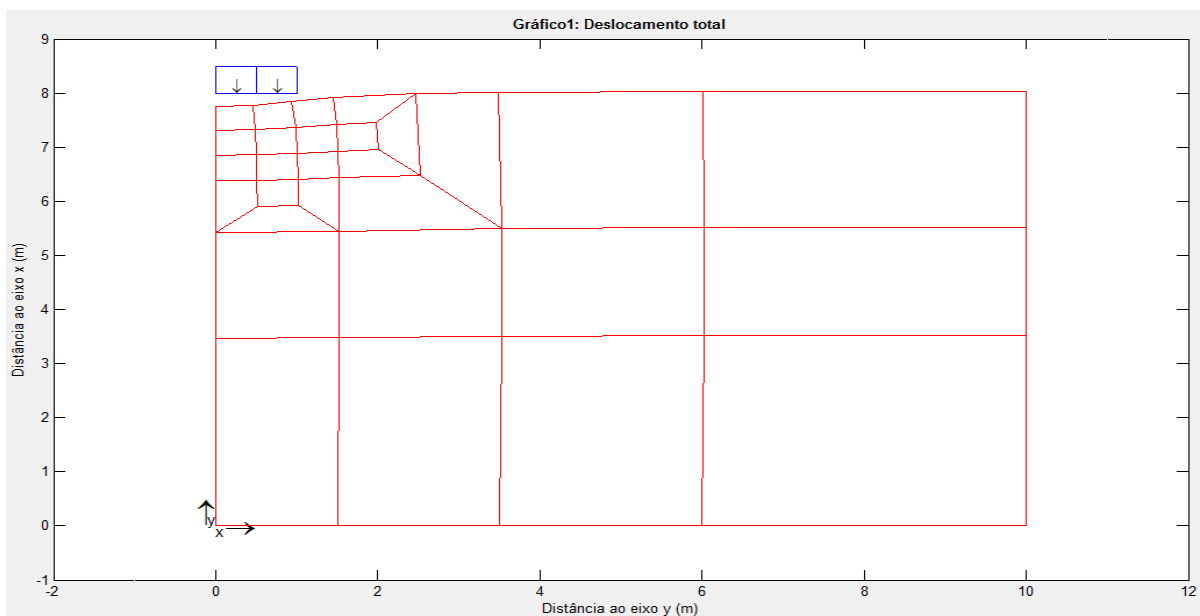


Figura 62: Exemplo da opção "Malha (deslocamento total)" (exemplo de consolidação bidimensional).

O utilizador pode ainda adicionar outras representações à referida anteriormente, nomeadamente vetores do deslocamento e vetores da tensão efetiva. No caso dos vetores do deslocamento, as opções disponíveis são entre vertical, horizontal e total, como mostram as Figuras 63 a 65.

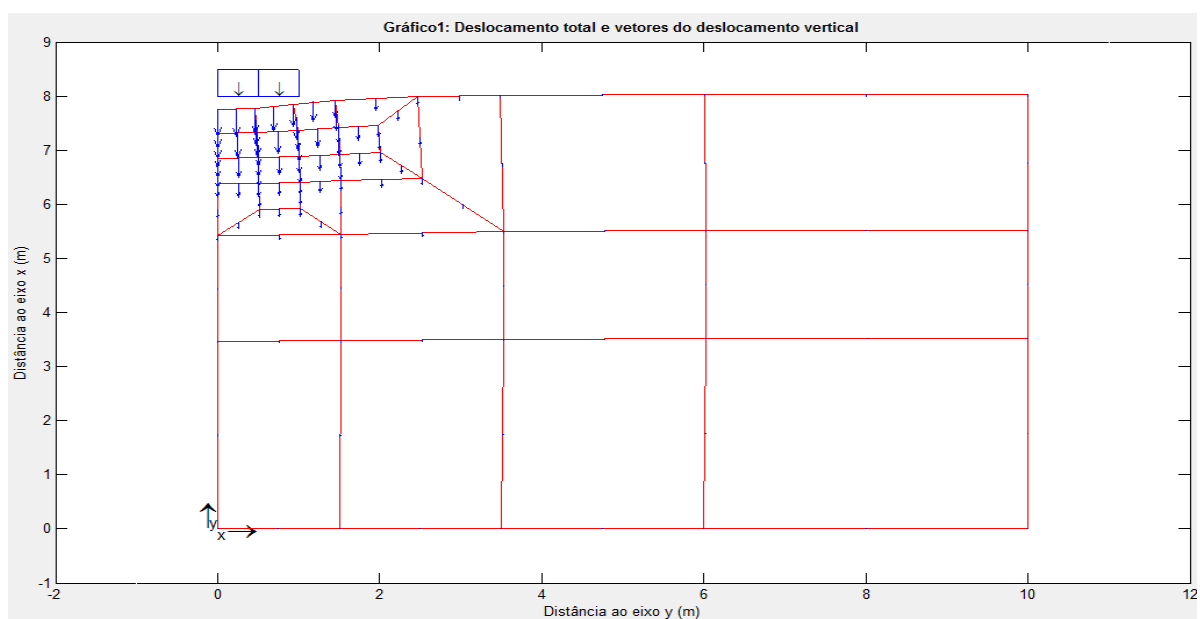


Figura 63: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento vertical (exemplo de consolidação bidimensional).



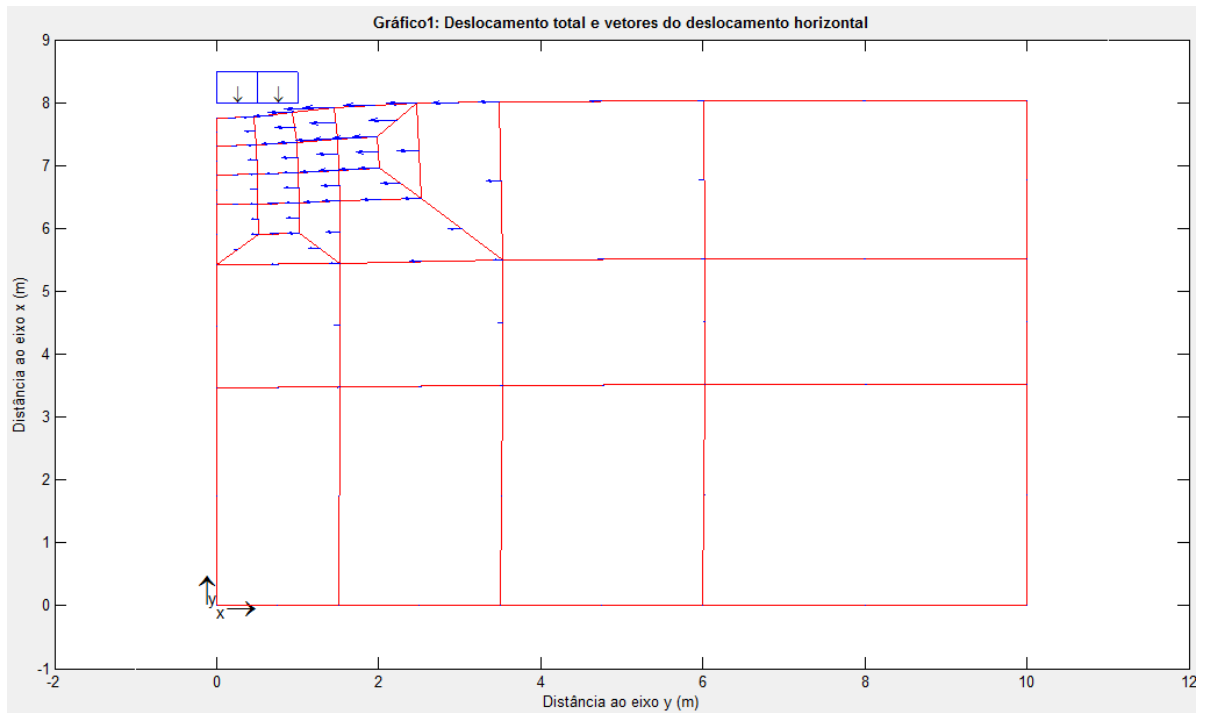


Figura 64: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento horizontal (exemplo de consolidação bidimensional).

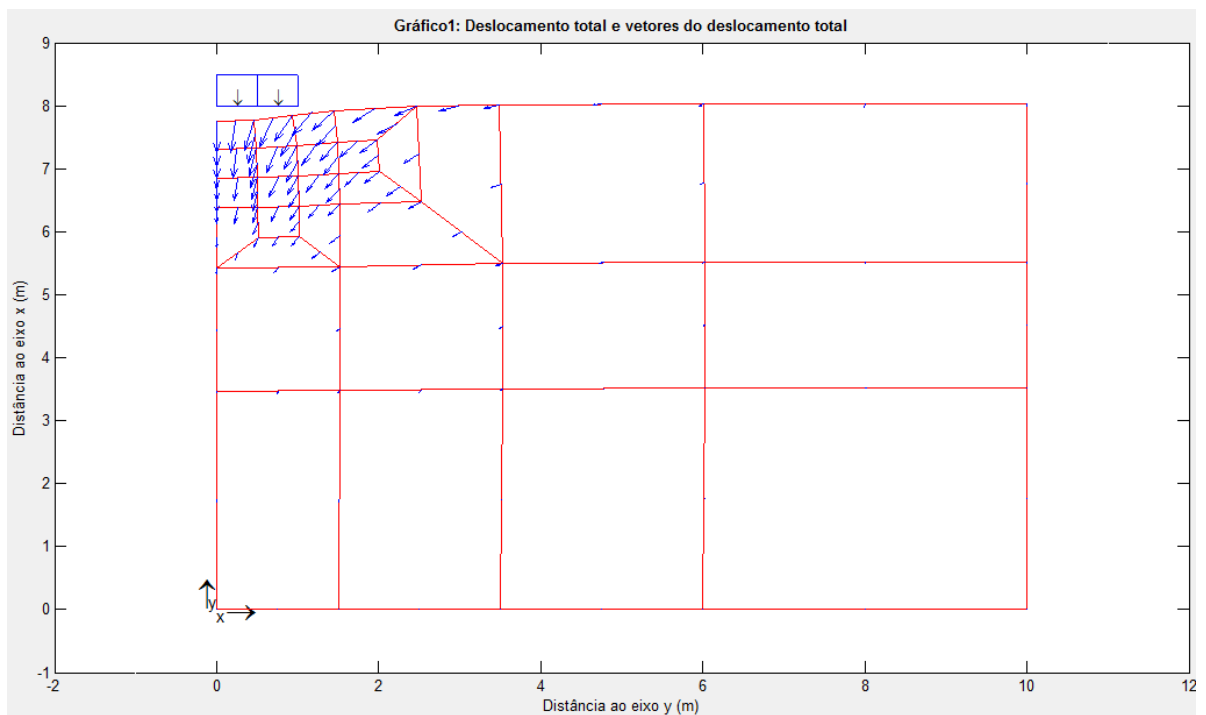


Figura 65: Exemplo da opção "Malha (deslocamento total)" com vetores do deslocamento total (exemplo de consolidação bidimensional).

No caso dos vetores da tensão efetiva, as opções disponíveis são entre vertical e horizontal, como mostram as Figuras 66 e 67.

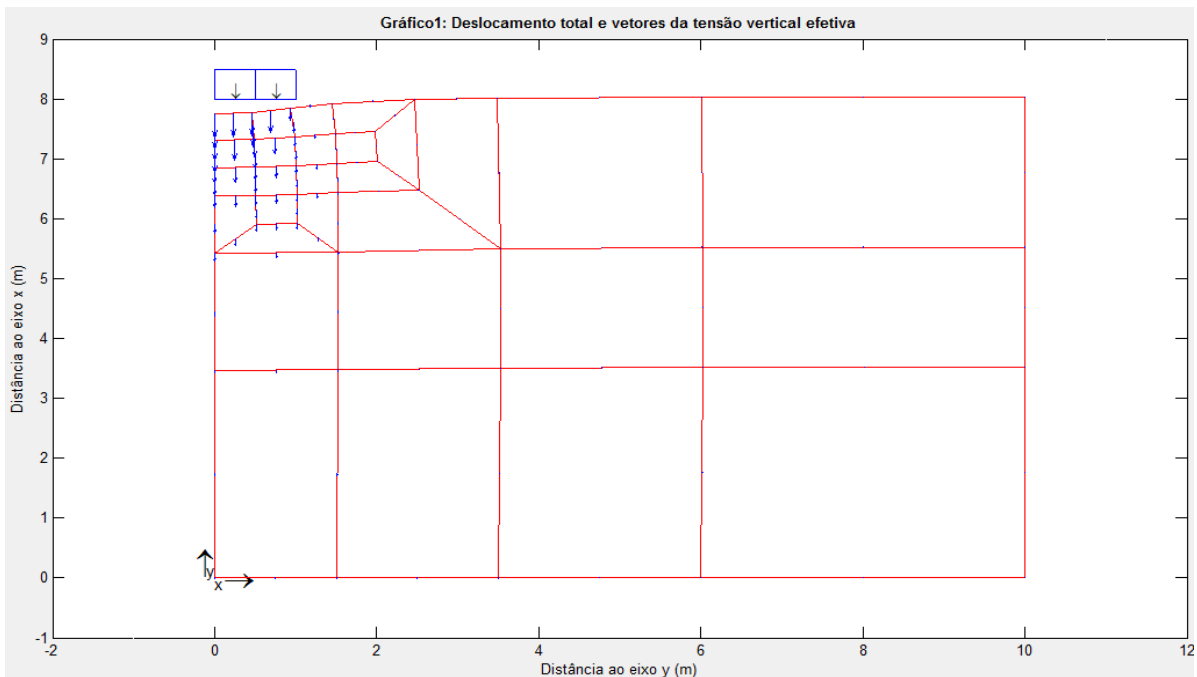


Figura 66: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação bidimensional).

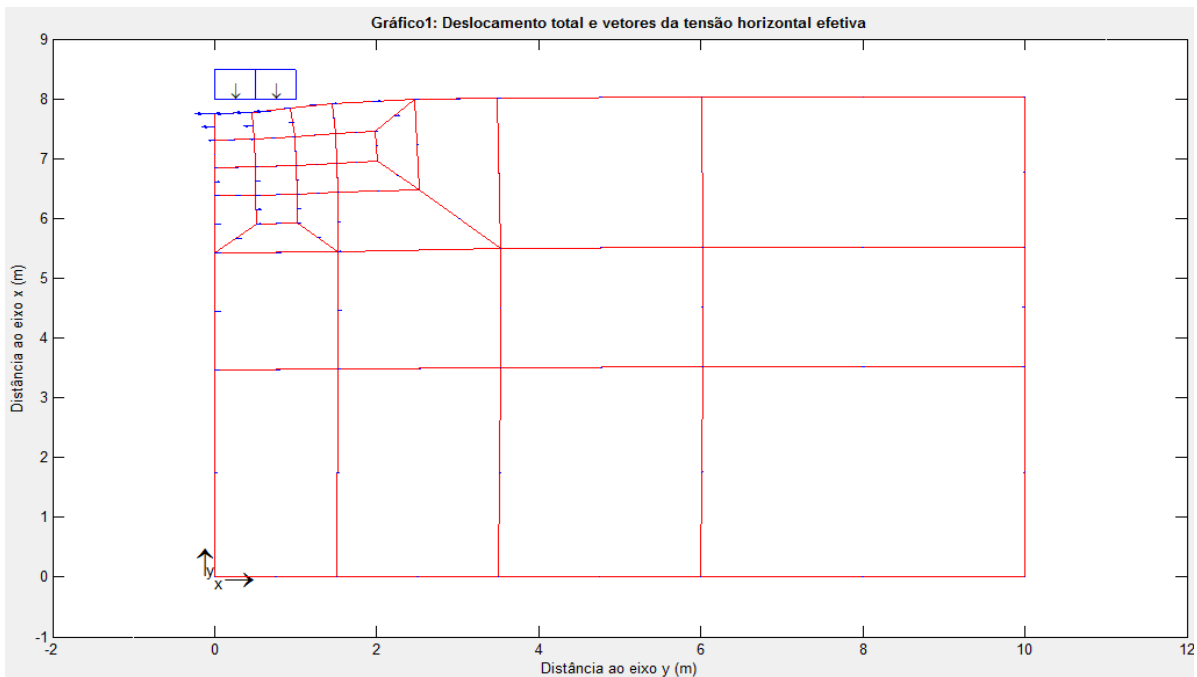


Figura 67: Exemplo da opção "Malha (deslocamento total)" com vetores da tensão horizontal efetiva (exemplo de consolidação bidimensional).

De referir que nas três opções do gráfico 1 referidas anteriormente é possível, a qualquer instante, parar a representação dinâmica clicando no botão "Pausa" e também ver os valores de um conjunto de parâmetros clicando no botão "Resultados", como mostra a Figura 68.

	Número do nó	Deslocamento vertical	Deslocamento horizontal	Tensão vertical efetiva
1	1.0	-1.44162E-4	0.0	-1.0257595634095693
2	2.0	-1.28352E-4	0.0	-0.9335904712404706
3	3.0	-1.13193E-4	0.0	-0.8511286529799051
4	4.0	-9.98611E-5	0.0	-0.7686668347193346
5	5.0	-8.80201E-5	0.0	-0.6713871976784498
6	6.0	-7.78534E-5	0.0	-0.5741075606375607
7	7.0	-6.9215E-5	0.0	-0.4991041655704851
8	8.0	-5.57872E-5	0.0	-0.3558402148302142
9	9.0	-4.5963E-5	0.0	-0.3223509845904912
10	10.0	-3.09101E-5	0.0	-0.19978683025404165
11	11.0	-2.08623E-5	0.0	-0.15663530047505084
12	12.0	-7.25819E-6	0.0	-0.09210643013856815
13	13.0	0.0	0.0	-0.04823649001584445
14	14.0	-1.42079E-4	-1.03295E-5	-1.0170190921690987
15	15.0	-1.11175E-4	-2.13962E-6	-0.8325237500000009
16	16.0	-8.62774E-5	1.63087E-6	-0.6393337500000005
17	17.0	-6.81626E-5	2.86714E-6	-0.46930513891829784
18	18.0	-4.96265E-5	3.14382E-6	-0.2953824733192195
19	19.0	-1.90644E-5	6.49558E-6	-0.13013479527874594
20	20.0	0.0	0.0	-0.03648980987727631
21	21.0	-1.34846E-4	-2.03905E-5	-0.9602315774428343
22	22.0	-1.18442E-4	-1.14874E-5	-0.86716375
23	23.0	-1.0446E-4	-3.81578E-6	-0.7546137499999999
24	24.0	-9.17271E-5	5.8805E-7	-0.6420637500000002
25	25.0	-8.11841E-5	3.13314E-6	-0.556911875
26	26.0	-7.22712E-5	4.64181E-6	-0.4717600000000002
27	27.0	-6.47541E-5	5.47215E-6	-0.4116996381335476
28	28.0	-5.83534E-5	5.97445E-6	-0.3496828908860924
29	29.0	-5.26857E-5	6.13661E-6	-0.3084511268258474
30	30.0	-1.20447E-4	-3.25853E-5	-0.9034440627165661
31	31.0	-9.13372E-5	-5.05169E-6	-0.6767037500000006
32	32.0	-7.30693E-5	4.37356E-6	-0.4744900000000001
33	33.0	-5.94459E-5	7.71262E-6	-0.3558212500000001
34	34.0	-4.91803E-5	8.79807E-6	-0.2717525000000006
35	35.0	-4.09714E-5	8.80169E-6	-0.21906334681406503

Figura 68: Exemplo dos resultados da opção "Malha (deslocamento total)" com vetores da tensão vertical efetiva (exemplo de consolidação bidimensional).

Nas opções do gráfico 1, a opção "Deslocamento vertical vs tempo" mostra um gráfico do deslocamento vertical do nó definido ao longo do tempo, como mostra a Figura 69, onde foi definido o nó número 1 (Figura 9).

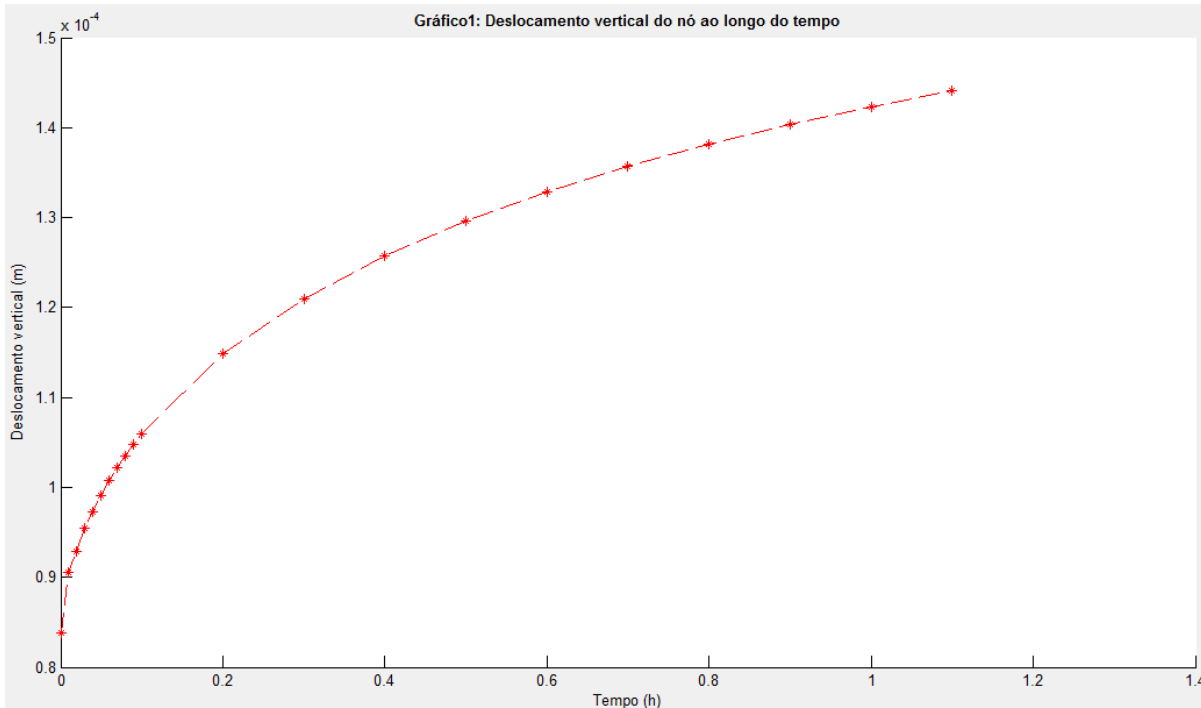


Figura 69: Exemplo da opção "Deslocamento vertical vs tempo" (exemplo de consolidação bidimensional).

Nesta opção o utilizador pode escolher entre usar o parâmetro "tempo" ou um "fator tempo  $T^m$ " (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas. Escolhendo a opção "Deslocamento vertical vs distância ao eixo y" nas opções do gráfico 1 é mostrado um gráfico do deslocamento vertical de um conjunto de nós distribuídos numa linha horizontal com a mesma ordenada do nó definido versus a distância de cada nó ao eixo de simetria, para o tempo definido, como mostra a Figura 70, onde foi definido o nó número 1 (Figura 9) e o incremento de tempo número 21 (Figura 47) que corresponde a um tempo de 1,1 horas.

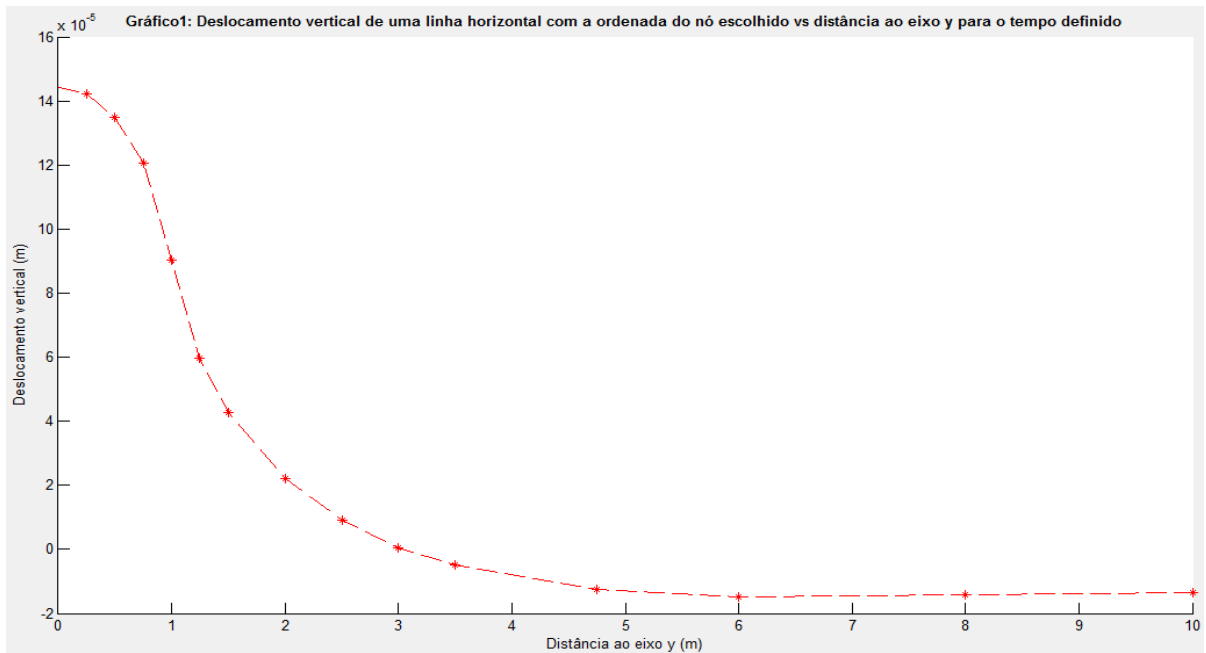


Figura 70: Exemplo da opção "Deslocamento vertical vs distância ao eixo y" (exemplo de consolidação bidimensional).

Nas opções do gráfico 1, escolhendo a opção "Pressão neutra vs tempo" é mostrado um gráfico da pressão neutra do nó definido ao longo do tempo como mostra a Figura 71, onde foi definido o nó número 5 (Figura 9).

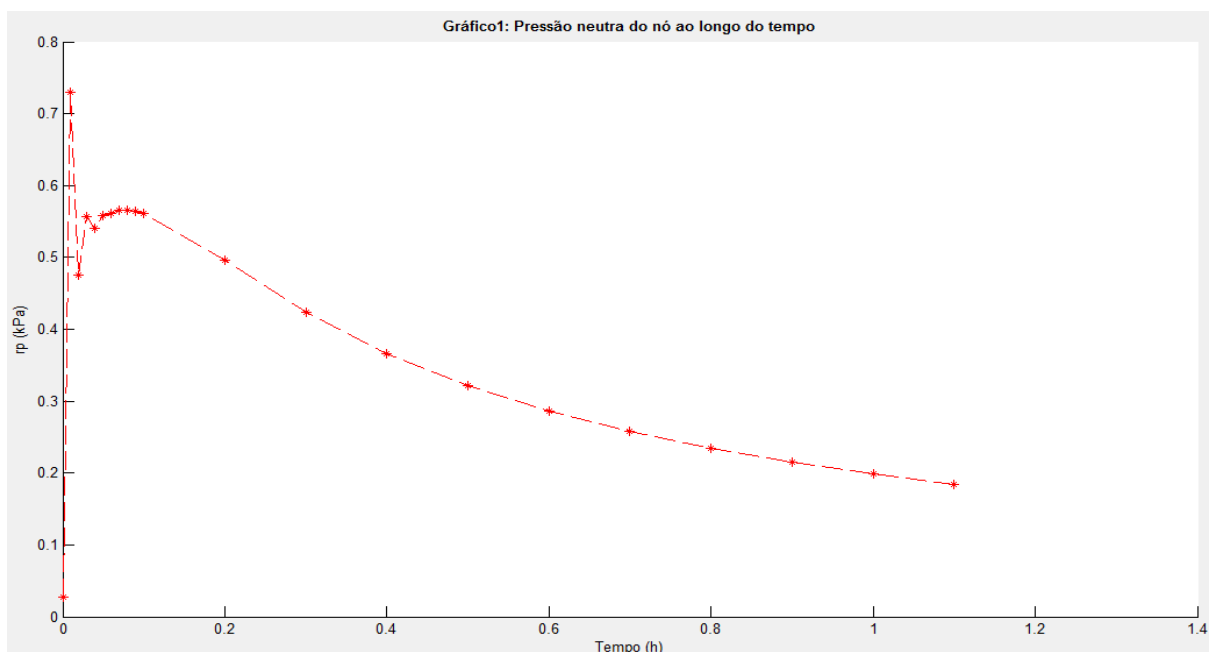


Figura 71: Exemplo da opção "Pressão neutra vs tempo" (exemplo de consolidação bidimensional).

Nesta opção o utilizador pode escolher entre usar o parâmetro "tempo" ou um fator "tempo T" (Equação 4) e também entre usar uma escala linear ou logarítmica no eixo das abcissas. No caso do eixo das ordenadas o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ).

Por fim, escolhendo a opção "Pressão neutra vs distância ao eixo y" nas opções do gráfico 1 é mostrado um gráfico da pressão neutra de um conjunto de nós distribuídos numa linha horizontal com a mesma ordenada do nó definido versus a distância de cada nó ao eixo de simetria, para o tempo definido, como mostra a Figura 72, onde foi definido o nó número 3 (Figura 9) e o incremento de tempo número 18 (Figura 47) que corresponde a um tempo de 0,8 horas.

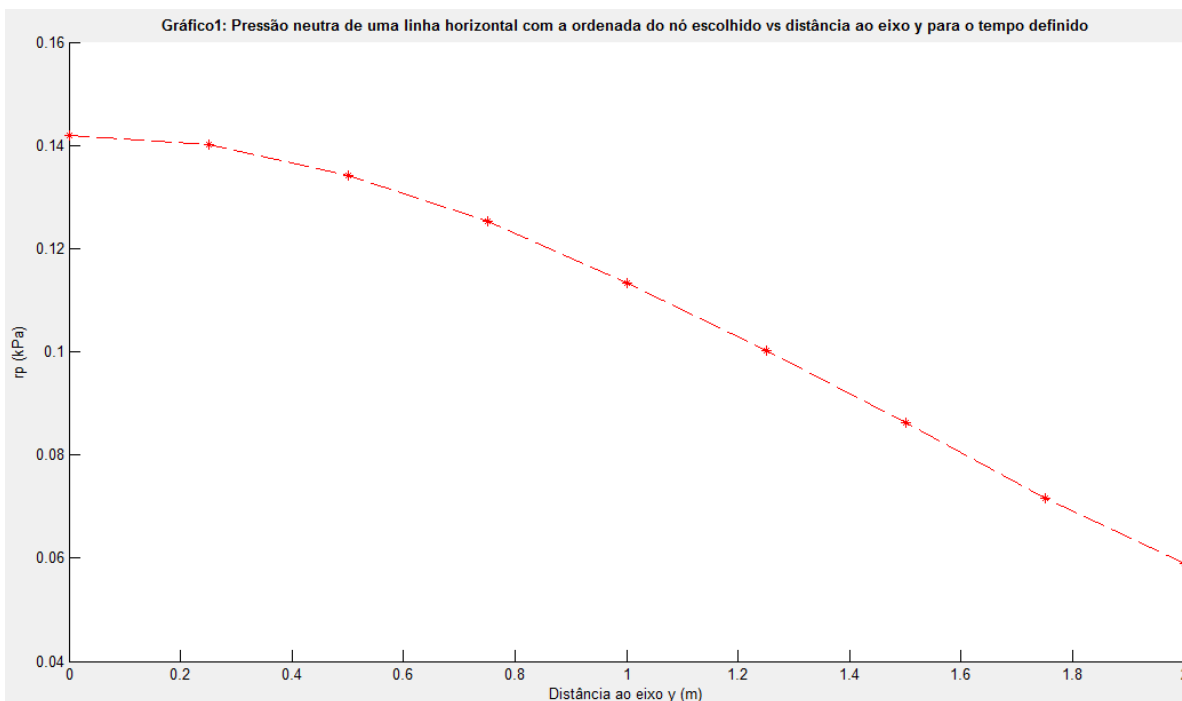


Figura 72: Exemplo da opção "Pressão neutra vs distância ao eixo y" (exemplo de consolidação bidimensional).

Nesta opção o utilizador pode escolher entre representar, no eixo das abcissas, a distância ao eixo de simetria ( $x$ ) ou a razão entre a distância ao eixo de simetria e a metade da largura da faixa carregada ( $x/a$ ). No caso do eixo das ordenadas o utilizador pode escolher entre representar a pressão neutra ( $r_p$ ) ou a razão entre a pressão neutra e a carga distribuída aplicada ( $r_p/p$ ).

## **6. CONCLUSÕES E ORIENTAÇÕES PARA FUTUROS DESENVOLVIMENTOS**

Esta dissertação teve como objetivo o melhoramento de um programa de cálculo automático sobre consolidação, existente na Universidade do Minho, para que este se tornasse mais amigável para o utilizador, principalmente para fins pedagógicos. O programa referido, que foi escrito em linguagem Fortran, permite o cálculo de problemas elásticos quer considerando só o estado de tensão e deformação do meio em análise, quer considerando o problema da consolidação acoplada.

Para que o programa se tornasse mais amigável para o utilizador, era indispensável o melhoramento do seu pré-processamento e pós-processamento, ou seja, que fosse facilitada a criação do ficheiro de dados necessário ao funcionamento do programa e que em vez de um enorme ficheiro de resultados, composto por uma pilha de números, fosse possível a visualização dos resultados graficamente.

Para esse efeito, foram criadas duas interfaces gráficas em MatLab, destinando-se uma à introdução de dados e outra à saída de resultados. Através da aplicação de dois exemplos, sendo um de consolidação unidimensional e outro de consolidação bidimensional, verificou-se que o objetivo desta dissertação foi atingido visto que o pré-processamento e pós-processamento do programa foram melhorados.

No caso do pré-processamento, as interfaces de introdução de dados básica e avançada cumpriram o seu objetivo visto que agora o utilizador não precisa de criar um ficheiro de texto, em que tinha de saber a ordem de introdução linha a linha de todos os parâmetros, pois as interfaces são tão amigáveis que facilitam o trabalho ao utilizador. Por isso, o utilizador tem menos trabalho e despende menos tempo na preparação e introdução de dados.

No caso do pós-processamento, a interface de saída de resultados também cumpriu o seu objetivo visto que agora o utilizador tem a possibilidade de visualizar um variado conjunto de resultados graficamente de forma rápida, inclusive representações dinâmicas que permitem

observar o fenómeno da consolidação. Para além disto, nesta dissertação verificou-se que os resultados obtidos através do programa são válidos.

No contexto pedagógico as melhorias introduzidas no programa existente passam a permitir mostrar aos alunos o andamento do processo de consolidação, nomeadamente a descida da água num tubo piezométrico colocado num dado ponto à medida que o tempo passa. Concomitantemente, os alunos podem ver a evolução em gráficos das pressões neutras, tensões efetivas e tensões totais e constatarem a diminuição gradual das pressões neutras à medida que as tensões efetivas vão aumentando com o tempo. Também podem verificar a validade do Princípio da Tensão Efetiva de Terzaghi, em que a tensão total é igual à soma da tensão efetiva com a pressão neutra.

No entanto, como melhoria do programa, sugere-se a possibilidade de introdução de dois tubos piezométricos em pontos situados a diferentes distâncias das fronteiras drenantes. Assim, os alunos passariam a ter a possibilidade de verificar a diferença de velocidade na descida da água nos dois tubos, podendo constatar que a água desce mais rapidamente num tubo colocado mais próximo da fronteira drenante.

Para futuros melhoramentos do programa são ainda dadas as seguintes sugestões:

- Pré-processamento:
  - O aproveitamento de todas as capacidades do programa, como por exemplo:
    - Uso de elementos finitos de Lagrange de 9 nós;
    - Uso de elementos tendo todos os nós as 2 componentes dos deslocamentos mas sendo só considerada a pressão neutra nos 4 nós dos vértices dos elementos.
  - A possibilidade de criar ficheiros de dados através de outros já existentes alterando os valores dos parâmetros que se pretende que sejam diferentes, de modo a que se veja as diferenças entre os resultados obtidos de uma forma mais rápida.
- Pós-processamento:
  - Adicionar outras representações gráficas;
  - A possibilidade de ver os resultados em qualquer ponto e não apenas nos nós dos elementos da malha;



- A possibilidade de ver os resultados para qualquer tempo e não apenas nos tempos em que foram calculados os resultados;
- A possibilidade de gravar as representações gráficas da interface de saídas gráficas como ficheiro de imagem;
- A possibilidade de gravar as representações dinâmicas da interface de saídas gráficas como ficheiro de vídeo.



## 7. REFERÊNCIAS BIBLIOGRÁFICAS

- Biot, M.A. (1941). 'General theory of three-dimensional consolidation', *Journal of Applied Physics*, vol. 12, pp. 155-164.
- Booker, J.R. (1973). 'A numerical method for the solution of Biot's consolidation theory', *The Quarterly Journal of Mechanics and Applied Mathematics*, vol. 26, pp. 457-470.
- Booker, J.R. & Small, J.C. (1975). 'An investigation of the stability of numerical solutions of Biot's equations of consolidation', *International Journal of Solids and Structures*, vol. 11, pp. 907-917.
- Chen, S.L., Chen, L.Z. & Zhang, L.M. (2005a). 'The axisymmetric consolidation of a semi-infinite transversely isotropic saturated soil', *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 29, pp. 1249-1270.
- Chen, R.P., Zhou, W.H., Wang, H.Z. & Chen, Y.M. (2005b). 'One-dimensional nonlinear consolidation of multi-layered soil by differential quadrature method', *Computers and Geotechnics*, vol. 32, pp. 358-369.
- Christian, J.T. & Boehmer, J.W. (1970). 'Plane strain consolidation by finite elements', *Journal of the Soil Mechanics and Foundations Division, ASCE*, vol. 96, pp. 1435-1457.
- Correia, R. (1980). *Resolução de problemas de consolidação bidimensional pelo método dos elementos finitos*, LNEC, Proc. 55/15/6188, Lisboa.
- Correia, R. (1982). *Aplicação de modelos matemáticos ao estudo de aterros construídos sobre solos argilosos moles*, LNEC, Proc. 55/11/7381, Lisboa.
- Ding, D. & Naylor, D.J. (1989). 'A virtual work derivation of the Biot consolidation finite element formulation', *Engineering Computations*, vol. 6, pp. 158-162.

Ferronato, M., Castelletto, N. & Gambolati, G. (2010). 'A fully coupled 3-D mixed finite element model of Biot consolidation', *Journal of Computational Physics*, vol. 229, pp. 4813-4830.

Geng, X., Xu, C. & Cai, Y. (2006). 'Non-linear consolidation analysis of soil with variable compressibility and permeability under cyclic loadings', *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 30, pp. 803-821.

Ghaboussi, J. & Wilson, E.L. (1973). 'Flow of compressible fluid in porous elastic media', *International Journal for Numerical Methods in Engineering*, vol. 5, pp. 419-442.

Hassanen, M. (2007). 'Application of scaled boundary finite element method on soil-structure interaction – two dimensions dynamic coupled consolidation analysis of fully saturated soils', *Proceedings of the 12th WSEAS International Conference on APPLIED MATHEMATICS*, Cairo, Egypt.

Kanok-Nukulchai, W. & Suaris, V.W. (1982). 'An eficiente finite element scheme for elastic porous media', *International Journal of Solids and Structures*, vol. 18, no. 1, pp. 37-49.

Krause, G. (1978). 'Finite element schemes for porous elastic media', *Journal of the Engineering Mechanics Division*, ASCE 104 (EM3), pp. 605-620.

Martins, F.F. (1993). 'Cálculo elasto-plástico de maciços terrosos em consolidação. Programa e aplicações', Tese de Doutorado em Engenharia Civil, Universidade do Minho.

Menéndez, C., Nieto, P.J.G., Ortega, F.A. & Bello, A. (2009). 'Mathematical modeling and study of the consolidation of an elastic saturated soil with an incompressible fluid by FEM', *Mathematical and Computer Modelling*, vol. 49, pp. 2002-2018.

Menéndez, C., Nieto, P.J.G., Ortega, F.A. & Bello, A. (2010). 'Non-linear analysis of the consolidation of an elastic saturated soil with incompressible fluid and variable permeability by FEM', *Applied Mathematics and Computation*, vol. 216, pp. 458-476.

Mira, P., Pastor, M., Li, T. & Liu, X. (2003). 'A new stabilized enhanced strain element with equal order of interpolation for soil consolidation problems', *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 4257-4277.

Moreira, M. (2005). 'Um modelo de elementos finitos para análise acoplada de problemas de adensamento com simetria axial', Tese de Mestrado em Engenharia Oceânica, Fundação Universidade Federal do Rio Grande.

NG, A.K.L. & Small, J.C. (2000). 'Use of coupled finite element analysis in unsaturated soil problems', *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 24, pp. 73-94.

Perrone, V. (1998). 'One dimensional computer analysis of simultaneous consolidation and creep of clay, Phd Thesis, Faculty of the Virginia Polytechnic Institute and State University.

Qin, A., Sun, D. & Tan, Y. (2010). 'Analytical solution to one-dimensional consolidation in unsaturated soils under loading varying exponentially with time', *Computers and Geotechnics*, vol. 37, pp. 233-238.

Rado, H., Beneyto, P., Mroginski, J. & Awruch, A. (2009). 'Influence of the saturation-suction relationship in the formulation of non-saturated soil consolidation models', *Mathematical and Computer Modelling*, vol. 49, pp. 1058-1070.

Reed, M.B. (1984). 'An investigation of numerical errors in the analysis of consolidation by finite elements', *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 8, pp. 243-257.

Rendulic, L. (1936). 'Porenziffer und porenwasserdruck in tonen', *Der Bauingenieur*, vol.17, no. 51/53, pp. 559-564.

Rujikiatkamjorn, C. & Indraratna, B. (2006). 'Three-dimensional analysis of soft soil consolidation improved by prefabricated vertical drains', *ASCE Special Geotechnical Publication*, vol. 152, pp. 161-168.

Sandhu, R.S. & Wilson, E.L. (1969). 'Finite element analysis of seepage in elastic media', *Journal of the Engineering Mechanics Division*, ASCE, vol. 95, pp. 641-652.

Schiffman, R.L., Chen, A.T. & Jordan, J.C. (1969). 'An analysis of consolidation theories', *Journal of the Soil Mechanics and Foundations Division*, ASCE, no. SM1, pp. 285-312.

Suoangkat, H. (1994). 'On finite element analysis of nonlinear consolidation', Master's Thesis in Mechanical Engineering, Massachusetts Institute of Technology.

Terzaghi, K. (1943). *Theoretical soil mechanics*, John Wiley & Sons Inc, New York.

Verruijt, A. (1978). 'Applications of the finite element method in Geomechanics', Delft, Delft University of Technology.

Walker, R. (2011). 'Vertical drain consolidation analysis in one, two and three dimensions', *Computers and Geotechnics*, vol. 38, pp. 1069-1077.

Wang, Z.L., Wang, J.G. & Shen, R.F. (2008). 'The application of a meshless method to consolidation analysis of saturated soils with anisotropic damage', *Computers & Geosciences*, vol. 34, pp. 849-859.

Xie, K.H. & Leo, C.J. (2004). 'Analytical solutions of one-dimensional large strain consolidation of saturated and homogeneous clays', *Computers and Geotechnics*, vol. 31, pp. 301-314.

Xie, K.H., Wang, K., Wang, Y.L. & Li, C.X. (2010). 'Analytical solution for one-dimensional consolidation of clayey soils with a threshold gradient', *Computers and Geotechnics*, vol. 37, pp. 487-493.

Xie, K.H., Xie, X.Y. & Gao, X. (1999). 'Theory of one dimensional consolidation of two-layered soil with partially drained boundaries', *Computers and Geotechnics*, vol. 24, pp. 265-278.

Xie, K.H., Xie, X.Y. & Li, X.B. (2008). 'Analytical theory for one-dimensional consolidation of clayey soils exhibiting rheological characteristics under time-dependent loading', *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 32, pp. 1833-1855.

Zhang, J. & Tao, G. (2011). 'A new model for soil consolidation based on microstructure', *Journal of Shanghai Jiaotong University*, vol. 16, pp. 78-82.

Zhu, G. & Yin, J. (2004). 'Consolidation analysis of soil with vertical and horizontal drainage under ramp loading considering smear effects', *Geotextiles and Geomembranes*, vol. 22, pp. 63-74.





# **ANEXOS**



# A1. CÓDIGOS EM MATLAB DAS INTERFACES

## function varargout = menu\_principal(varargin)

```
function togglebutton1_Callback(hObject, eventdata, handles)
c_fich_dados=get(handles.togglebutton1,'value');
if c_fich_dados==1
    set(handles.pushbutton2,'visible','off');
    set(handles.pushbutton3,'visible','off');
    set(handles.pushbutton4,'visible','on');
    set(handles.pushbutton5,'visible','on');
else
    set(handles.pushbutton2,'visible','on');
    set(handles.pushbutton3,'visible','on');
    set(handles.pushbutton4,'visible','off');
    set(handles.pushbutton5,'visible','off');
end
function pushbutton2_Callback(hObject, eventdata, handles)
filename_DAT = uigetfile({'*.DAT'},'Escolha o ficheiro de dados');
pause(0.1)
if filename_DAT~=0
    existe_filename_RES=1;
    while existe_filename_RES>0
        filename_RES = uiputfile({'*.RES'},'Guardar como (Ficheiro de resultados):');
        pause(0.1)
        if filename_RES~=0
            existe_filename_RES = fopen(filename_RES,'rt');
            if existe_filename_RES<0
                fid2=fopen('nome_ficheiros.txt','w');
                fprintf(fid2,'%s \n',filename_DAT);
                fprintf(fid2,'%s \n',filename_RES);
                fclose(fid2);
                pause(0.1)
                system('CONSOL.exe')
                pause(0.1)
                resposta3=questdlg('Ver resultados?',','Sim','Não','Sim');
                pause(0.1)
                delete nome_ficheiros.txt
                if strcmp(resposta3,'Sim')
                    ler_ficheiro_abrir_menu()
                end
            end
        else
            existe_filename_RES=-1;
        end
    end
end
function pushbutton3_Callback(hObject, eventdata, handles)
ler_ficheiro_abrir_menu();
if filename_RES~=0
    delete(menu_principal);
end
function pushbutton4_Callback(hObject, eventdata, handles)
menu_basico_fich_dados_pag1()
delete(menu_principal);
function pushbutton5_Callback(hObject, eventdata, handles)
menu_fich_dados_pag1()
delete(menu_principal);
function figure1_CloseRequestFcn(hObject, eventdata, handles)
resposta=questdlg('Fechar programa?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(hObject);
end
function figure1_CreateFcn(hObject, eventdata, handles)
warning off all
```



```

        end
        save teste
        delete(menu_basico_fich_dados_pag1);
        menu_basico_fich_dados_pag2()
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function togglebutton1_Callback(hObject, eventdata, handles)
unidimensional=get(handles.togglebutton1,'value');
if unidimensional==1
    set(handles.togglebutton2,'value',0);
    set(handles.text2,'visible','on');
    set(handles.a,'visible','on');
    winopen('unidimensional.png')
else
    set(handles.text2,'visible','off');
    set(handles.a,'visible','off');
end
function togglebutton2_Callback(hObject, eventdata, handles)
bidimensional=get(handles.togglebutton2,'value');
if bidimensional==1
    set(handles.togglebutton1,'value',0);
    set(handles.text2,'visible','on');
    set(handles.a,'visible','on');
    winopen('bidimensional.png')
else
    set(handles.text2,'visible','off');
    set(handles.a,'visible','off');
end
function a_Callback(hObject, eventdata, handles)
a=get(handles.a,'String');
if isempty(a)==0
    a=str2double(get(handles.a,'String'));
    if a<0 || isnan(a)
        set(handles.a,'String','');
        if a<0
            errordlg('Valor de "a" tem que ser maior ou igual a 0','Erro');
        elseif isnan(a)
            errordlg('Valor de "a" tem que ser numérico','Erro');
        end
    end
end
end
function axes1_CreateFcn(hObject, eventdata, handles)
imagem=imread('unidimensional2.png');
image(imagem)
axis off
function axes2_CreateFcn(hObject, eventdata, handles)
imagem=imread('bidimensional2.png');
image(imagem)
axis off
function figure1_CloseRequestFcn(hObject, eventdata, handles)
resposta=questdlg('Fechar programa?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(hObject);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
end

```

#### **function varargout = menu\_basico\_fich\_dados\_pag2(varargin)**

```

function nmats_Callback(hObject, eventdata, handles)
NMATS = get(handles.nmats,'String');
if isempty(NMATS)==0
    load teste NELEM
    NMATS = str2double(get(handles.nmats,'String'));
    if NMATS<1 || NMATS>NELEM || isnan(NMATS) || isnan(NELEM)
        set(handles.nmats,'String','');
        if isnan(NELEM)
            errordlg('Introduza primeiro o valor de "NELEM"', 'Erro');
        elseif NMATS<1
            errordlg('Valor de "NMATS" tem que ser maior ou igual a 1','Erro');
        elseif NMATS>NELEM
            errordlg('Valor de "NMATS" não pode ser maior que "NELEM"', 'Erro');
        elseif isnan(NMATS)

```

```

        errordlg('Valor de "NMATS" tem que ser numérico','Erro');
    end
end
end
function linca_Callback(hObject, eventdata, handles)
LINCA = get(handles.linca,'String');
if isempty(LINCA)==0
    LINCA = str2double(get(handles.linca,'String'));
    if LINCA<0 || LINCA>1 || isnan(LINCA)
        set(handles.linca,'String','');
        if LINCA<0 || LINCA>1
            errordlg('Valor de "LINCA" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(LINCA)
            errordlg('Valor de "LINCA" tem que ser numérico','Erro');
        end
    end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag2);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
NVFIX = str2double(get(handles.nvfix,'String'));
NPRFX = str2double(get(handles.nprfx,'String'));
NMATS = str2double(get(handles.nmats,'String'));
NGAUS = str2double(get(handles.ngaus,'String'));
TELEM = str2double(get(handles.telem,'String'));
NINCR = str2double(get(handles.nincr,'String'));
if TELEM==2
    NPPRE = str2double(get(handles.nppre,'String'));
    if isnan(NPPRE)==0
        Sair=1;
    else
        Sair=0;
    end
else
    Sair=1;
end
LINCA = str2double(get(handles.linca,'String'));
if isnan(NVFIX)==0 && isnan(NPRFX)==0 && isnan(NMATS)==0 && isnan(NGAUS)==0 && isnan(TELEM)==0 &&
isnan(NINCR)==0 && isnan(LINCA)==0 && Sair==1
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        save teste2
        delete(menu_basico_fich_dados_pag2);
        if NMATS>1
            menu_basico_fich_dados_pag3()
        else
            load teste NELEM
            for i=1:NELEM
                NUMEL(1,i)=i;
            end
            MATNO(1,1:NELEM)=1;
            NNODE(1,1:NELEM)=8;
            save teste3 NUMEL MATNO NNODE
            menu_basico_fich_dados_pag4()
        end
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function nvfix_Callback(hObject, eventdata, handles)
NVFIX = get(handles.nvfix,'String');
if isempty(NVFIX)==0
    load teste NPOIN
    NVFIX = str2double(get(handles.nvfix,'String'));
    if NVFIX<1 || NVFIX>NPOIN || isnan(NVFIX) || isnan(NPOIN)
        set(handles.nvfix,'String','');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif NVFIX<1
            errordlg('Valor de "NVFIX" tem que ser maior ou igual a 1','Erro');
        end
    end
end

```

```

elseif NVFIX>NPOIN
    errordlg('Valor de "NVFIX" não pode ser maior que "NPOIN"', 'Erro');
elseif isnan(NVFIX)
    errordlg('Valor de "NVFIX" tem que ser numérico', 'Erro');
end
end
end
function nprfx_Callback(hObject, eventdata, handles)
NPRFX = get(handles.nprfx, 'String');
if isempty(NPRFX)==0
    load teste NPOIN
    NPRFX = str2double(get(handles.nprfx, 'String'));
    if NPRFX<0 || NPRFX>NPOIN || isnan(NPRFX) || isnan(NPOIN)
        set(handles.nprfx, 'String', '');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif NPRFX<0
            errordlg('Valor de "NPRFX" tem que ser maior ou igual a 0', 'Erro');
        elseif NPRFX>NPOIN
            errordlg('Valor de "NPRFX" não pode ser maior que "NPOIN"', 'Erro');
        elseif isnan(NPRFX)
            errordlg('Valor de "NPRFX" tem que ser numérico', 'Erro');
        end
    end
end
end
function ngaus_Callback(hObject, eventdata, handles)
NGAUS = get(handles.ngaus, 'String');
if isempty(NGAUS)==0
    NGAUS = str2double(get(handles.ngaus, 'String'));
    if NGAUS<2 || NGAUS>3 || isnan(NGAUS)
        set(handles.ngaus, 'String', '');
        if NGAUS<2 || NGAUS>3
            errordlg('Valor de "NGAUS" só pode ser igual a 2 ou 3', 'Erro');
        elseif isnan(NGAUS)
            errordlg('Valor de "NGAUS" tem que ser numérico', 'Erro');
        end
    end
end
end
function nincr_Callback(hObject, eventdata, handles)
NINCR = get(handles.nincr, 'String');
if isempty(NINCR)==0
    NINCR = str2double(get(handles.nincr, 'String'));
    if NINCR<1 || isnan(NINCR)
        set(handles.nincr, 'String', '');
        if NINCR<1
            errordlg('Valor de "NINCR" tem que ser maior ou igual a 1', 'Erro');
        elseif isnan(NINCR)
            errordlg('Valor de "NINCR" tem que ser numérico', 'Erro');
        end
    end
end
end
function telem_Callback(hObject, eventdata, handles)
TELEM = get(handles.telem, 'String');
if isempty(TELEM)==0
    TELEM = str2double(get(handles.telem, 'String'));
    if TELEM<0 || TELEM>1 || isnan(TELEM)
        set(handles.telem, 'String', '');
        if TELEM<0 || TELEM>1
            errordlg('Valor de "TELEM" só pode ser igual a 0 ou 1', 'Erro');
        elseif isnan(TELEM)
            errordlg('Valor de "TELEM" tem que ser numérico', 'Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_basico_fich_dados_pag2);
    menu_basico_fich_dados_pag1()
end
end

```

### **function varargout = menu\_basico\_fich\_dados\_pag3(varargin)**

```
function matno_Callback(hObject, eventdata, handles)
MATNO = get(handles.matno,'String');
if isempty(MATNO)==0
    load teste2 NMATS
    MATNO = str2double(get(handles.matno,'String'));
    if MATNO<1 || MATNO>NMATS || isnan(MATNO)
        set(handles.matno,'String','');
        if MATNO<1
            errordlg('Valor de "MATNO" tem que ser maior ou igual a 1','Erro');
        elseif MATNO>NMATS
            errordlg('Valor de "MATNO" nao pode ser maior que "NMATS','Erro');
        elseif isnan(MATNO)
            errordlg('Valor de "MATNO" tem que ser numérico','Erro');
        end
    end
end
function matno_CreateFcn(hObject, eventdata, handles)
load teste2 NMATS
if NMATS==1
    set(hObject,'backgroundcolor',[.502 .502 .502]);
    set(hObject,'foregroundcolor',[1 1 1]);
    set(hObject,'enable','inactive');
    set(hObject,'string','1');
else
    set(hObject,'backgroundcolor',[1 1 1]);
    set(hObject,'foregroundcolor',[0 0 0]);
    set(hObject,'enable','on');
    set(hObject,'string','');
end
function nnode_Callback(hObject, eventdata, handles)
NNODE = get(handles.nnode,'String');
if isempty(NNODE)==0
    load teste MNODE
    NNODE = str2double(get(handles.nnode,'String'));
    if NNODE<1 || NNODE>MNODE || isnan(NNODE)
        set(handles.nnode,'String','');
        if NNODE<1
            errordlg('Valor de "NNODE" tem que ser maior ou igual a 1','Erro');
        elseif NNODE>MNODE
            errordlg('Valor de "NNODE" nao pode ser maior que "MNODE','Erro');
        elseif isnan(NNODE)
            errordlg('Valor de "NNODE" tem que ser numérico','Erro');
        end
    end
end
function nnode_CreateFcn(hObject, eventdata, handles)
load teste MNODE
set(hObject,'backgroundcolor',[.502 .502 .502]);
set(hObject,'foregroundcolor',[1 1 1]);
set(hObject,'enable','inactive');
set(hObject,'string',MNODE);
function lnods_Callback(hObject, eventdata, handles)
LNODS = get(handles.lnods,'String');
if isempty(LNODS)==0
    NNODE = str2double(get(handles.nnode,'String'));
    LNODS = str2num(get(handles.lnods,'String'));
    tamanho_LNODS=size(LNODS);
    if isnan(NNODE) || tamanho_LNODS(2)~=NNODE || isempty(LNODS)
        set(handles.lnods,'String','');
        if isnan(NNODE)
            errordlg('Introduza primeiro o valor de "NNODE"', 'Erro');
        elseif isempty(LNODS)
            errordlg('Valor de "LNODS" tem que ser numérico','Erro');
        elseif tamanho_LNODS(2)~=NNODE
            errordlg('O número de nós tem que ser igual a "NNODE"', 'Erro');
        end
    end
end
function lnods_CreateFcn(hObject, eventdata, handles)
load teste LNODS MNODE
set(hObject,'backgroundcolor',[.502 .502 .502]);
set(hObject,'foregroundcolor',[1 1 1]);
set(hObject,'enable','inactive');
set(hObject,'String',num2str(LNODS(1,1:MNODE)));
```



```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_basico_fich_dados_pag3);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste NELEM
Numel=str2double(get(handles.numel, 'String'));
MATNO = str2double(get(handles.matno, 'String'));
if NELEM==Numel && isnan(MATNO)==0
    load teste3 NUMEL MATNO NNODE
    NUMEL(1, NELEM) = str2double(get(handles.numel, 'String'));
    MATNO(1, NELEM) = str2double(get(handles.matno, 'String'));
    NNODE(1, NELEM) = str2double(get(handles.nnode, 'String'));
    save teste3 NUMEL MATNO NNODE
    resposta=questdlg('Seguinte?', 'Sim', 'Não', 'Sim');
    if strcmp(resposta, 'Sim')
        delete(menu_basico_fich_dados_pag3);
        menu_basico_fich_dados_pag4()
    end
elseif isnan(MATNO)
    errordlg('Preencha todos os campos', 'Erro');
elseif NELEM~=Numel
    errordlg('Introduza os dados dos elementos em falta', 'Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_basico_fich_dados_pag3);
    menu_basico_fich_dados_pag2()
end
function pushbutton4_Callback(hObject, eventdata, handles)
resposta=questdlg('Elemento anterior?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    Numel=str2double(get(handles.numel, 'String'));
    Numel=Numel-1;
    set(handles.numel, 'String', Numel);
    if Numel==1
        set(handles.pushbutton4, 'Visible', 'off');
    end
    load teste3 NNODE MATNO
    load teste2 NMATS
    load teste LNODS
    set(handles.pushbutton5, 'Visible', 'on');
    if NMATS>1
        set(handles.matno, 'String', MATNO(1, Numel));
    end
    set(handles.lnodes, 'String', num2str(LNODS(Numel, 1:NNODE(1, Numel))));
end
function pushbutton5_Callback(hObject, eventdata, handles)
load teste NELEM MNODE
load teste2 NMATS
Numel = str2double(get(handles.numel, 'String'));
MATNO = str2double(get(handles.matno, 'String'));
if isnan(MATNO)==0
    if Numel>1
        load teste3 NUMEL MATNO NNODE
    end
    NUMEL(1, Numel)=str2double(get(handles.numel, 'String'));
    MATNO(1, Numel)=str2double(get(handles.matno, 'String'));
    NNODE(1, Numel)=str2double(get(handles.nnode, 'String'));
    resposta=questdlg('Elemento seguinte?', 'Sim', 'Não', 'Sim');
    if strcmp(resposta, 'Sim')
        Numel=Numel+1;
        if Numel==NELEM
            set(handles.pushbutton5, 'Visible', 'off');
        end
        set(handles.pushbutton4, 'Visible', 'on');
        if NMATS>1
            set(handles.matno, 'String', '');
        end
        load teste LNODS
        set(handles.lnodes, 'String', num2str(LNODS(Numel, 1:MNODE)));
    end
end

```

```

        set(handles.numel,'String',Numel);
        save teste3 NUMEL MATNO NNODE
    end
else
    errorDlg('Preencha todos os campos','Erro');
end
function pushbutton5_CreateFcn(hObject, eventdata, handles)
load teste NELEM
if NELEM==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.
function axes1_CreateFcn(hObject, eventdata, handles)
load teste unidimensional bidimensional
if unidimensional==1
    imagem=imread('unidimensional2.png');
    image(imagem)
    axis off
elseif bidimensional==1
    imagem=imread('bidimensional2.png');
    image(imagem)
    axis off
end
function pushbutton6_Callback(hObject, eventdata, handles)
load teste unidimensional bidimensional
if unidimensional==1
    winopen('unidimensional.png')
elseif bidimensional==1
    winopen('bidimensional.png')
end

```

#### **function varargout = menu\_basico\_fich\_dados\_pag4(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag4);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste2 NVFIX
Contagem=str2double(get(handles.contagem,'String'));
if Contagem==NVFIX
    NOFIX=str2double(get(handles.nofix,'String'));
    IFPRE1=str2double(get(handles.ifpre1,'String'));
    IFPRE2=str2double(get(handles.ifpre2,'String'));
    PRESC1=str2double(get(handles.presc1,'String'));
    PRESC2=str2double(get(handles.presc2,'String'));
    if isnan(NOFIX)==0 && isnan(IFPRE1)==0 && isnan(IFPRE2)==0 && isnan(PRESC1)==0 && isnan(PRESC2)==0
        if Contagem>1
            load teste4 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
        end
        NOFIX(1,Contagem)=str2double(get(handles.nofix,'String'));
        IFPRE1(1,Contagem)=str2double(get(handles.ifpre1,'String'));
        IFPRE2(1,Contagem)=str2double(get(handles.ifpre2,'String'));
        PRESC1(1,Contagem)=str2double(get(handles.presc1,'String'));
        PRESC2(1,Contagem)=str2double(get(handles.presc2,'String'));
        save teste4 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
        resposta=questdlg('Seguinte?',',','Sim','Não','Sim');
        if strcmp(resposta,'Sim')
            delete(menu_basico_fich_dados_pag4);
            menu_basico_fich_dados_pag5()
        end
    else
        errorDlg('Preencha todos os campos','Erro');
    end
else
    errorDlg('Introduza os dados dos nós restringidos em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)

```

```

resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag4);
    load teste2 NMATS
    if NMATS>1
        menu_basico_fich_dados_pag3()
    else
        menu_basico_fich_dados_pag2()
    end
end
end
function presc1_Callback(hObject, eventdata, handles)
PRESC1=get(handles.presc1,'String');
if isempty(PRESC1)==0
    PRESC1=str2double(get(handles.presc1,'String'));
    if isnan(PRESC1)
        set(handles.presc1,'String','');
        if isnan(PRESC1)
            errordlg('Valor de "PRESC1" tem que ser numérico','Erro');
        end
    end
end
end
function presc2_Callback(hObject, eventdata, handles)
PRESC2=get(handles.presc2,'String');
if isempty(PRESC2)==0
    PRESC2=str2double(get(handles.presc2,'String'));
    if isnan(PRESC2)
        set(handles.presc2,'String','');
        if isnan(PRESC2)
            errordlg('Valor de "PRESC2" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
NOFIX=str2double(get(handles.nofix,'String'));
IFPRE1=str2double(get(handles.ifpre1,'String'));
IFPRE2=str2double(get(handles.ifpre2,'String'));
PRESC1=str2double(get(handles.presc1,'String'));
PRESC2=str2double(get(handles.presc2,'String'));
if isnan(NOFIX)==0 && isnan(IFPRE1)==0 && isnan(IFPRE2)==0 && isnan(PRESC1)==0 && isnan(PRESC2)==0
    if Contagem>1
        load teste4 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
    end
    NOFIX(1,Contagem)=str2double(get(handles.nofix,'String'));
    IFPRE1(1,Contagem)=str2double(get(handles.ifpre1,'String'));
    IFPRE2(1,Contagem)=str2double(get(handles.ifpre2,'String'));
    PRESC1(1,Contagem)=str2double(get(handles.presc1,'String'));
    PRESC2(1,Contagem)=str2double(get(handles.presc2,'String'));
    save teste4 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
    resposta=questdlg('Nó restringido seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.nofix,'String','');
        set(handles.ifpre1,'String','');
        set(handles.ifpre2,'String','');
        set(handles.presc1,'String','');
        set(handles.presc2,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste2 NVFIX
        if Contagem==NVFIX
            set(handles.pushbutton4,'Visible','off');
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
end
function nofix_Callback(hObject, eventdata, handles)
NOFIX2=get(handles.nofix,'String');
if isempty(NOFIX2)==0
    load teste NPOIN
    NOFIX2=str2double(get(handles.nofix,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    if NOFIX2<1 || NOFIX2>NPOIN || isnan(NOFIX2)
        set(handles.nofix,'String','');
        if NOFIX2<1

```

```

        errordlg('Valor de "NOFIX" tem que ser maior ou igual a 1','Erro');
elseif NOFIX2>NPOIN
    errordlg('Valor de "NOFIX" nao pode ser maior que "NPOIN','Erro');
elseif isnan(NOFIX2)
    errordlg('Valor de "NOFIX" tem que ser numérico','Erro');
end
elseif Contagem>1
    load teste4 NOFIX
    for i=1:Contagem-1
        if NOFIX2==NOFIX(1,i)
            set(handles.nofix,'String','');
            errordlg('O número do nó já está a ser usado','Erro');
        end
    end
end
end
function ifpre1_Callback(hObject, eventdata, handles)
IFPRE1=get(handles.ifpre1,'String');
if isempty(IFPRE1)==0
    IFPRE1=str2double(get(handles.ifpre1,'String'));
    if IFPRE1<0 || IFPRE1>1 || isnan(IFPRE1)
        set(handles.ifpre1,'String','');
        if IFPRE1<0 || IFPRE1>1
            errordlg('Valor de "IFPRE1" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(IFPRE1)
            errordlg('Valor de "IFPRE1" tem que ser numérico','Erro');
        end
    end
end
function ifpre2_Callback(hObject, eventdata, handles)
IFPRE2=get(handles.ifpre2,'String');
if isempty(IFPRE2)==0
    IFPRE2=str2double(get(handles.ifpre2,'String'));
    if IFPRE2<0 || IFPRE2>1 || isnan(IFPRE2)
        set(handles.ifpre2,'String','');
        if IFPRE2<0 || IFPRE2>1
            errordlg('Valor de "IFPRE2" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(IFPRE2)
            errordlg('Valor de "IFPRE2" tem que ser numérico','Erro');
        end
    end
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Nó restringido anterior?',','Sim'.Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste4 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
    set(handles.pushbutton4,'Visible','on');
    set(handles.nofix,'String',NOFIX(1,Contagem));
    set(handles.ifpre1,'String',IFPRE1(1,Contagem));
    set(handles.ifpre2,'String',IFPRE2(1,Contagem));
    set(handles.presc1,'String',PRESC1(1,Contagem));
    set(handles.presc2,'String',PRESC2(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste2 NVFIX
if NVFIX==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.
function axes1_CreateFcn(hObject, eventdata, handles)
Ver function axes1_CreateFcn do ficheiro menu_basico_fich_dados_pag3.
function pushbutton7_Callback(hObject, eventdata, handles)
load teste unidimensional bidimensional
if unidimensional==1
    winopen('unidimensional.png')
elseif bidimensional==1
    winopen('bidimensional.png')
end

```

end

**function varargout = menu\_basico\_fich\_dados\_pag5(varargin)**

```
function prepr_Callback(hObject, eventdata, handles)
PREPR=get(handles.prepr,'String');
if isempty(PREPR)==0
    PREPR=str2double(get(handles.prepr,'String'));
    if isnan(PREPR)
        set(handles.prepr,'String','');
        if isnan(PREPR)
            errordlg('Valor de "PREPR" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag5);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste2 NPRFX
Contagem=str2double(get(handles.contagem,'String'));
if Contagem==NPRFX
    NOPRF=str2double(get(handles.nopr, 'String'));
    PREPR=str2double(get(handles.prepr,'String'));
    if isnan(NOPRF)==0 && isnan(PREPR)==0
        if Contagem>1
            load teste5 NOPRF PREPR
        end
        NOPRF(1,Contagem)=str2double(get(handles.nopr,'String'));
        PREPR(1,Contagem)=str2double(get(handles.prepr,'String'));
        save teste5 NOPRF PREPR
        resposta=questdlg('Seguinte?', 'Sim','Não','Sim');
        if strcmp(resposta,'Sim')
            delete(menu_basico_fich_dados_pag5);
            menu_basico_fich_dados_pag6()
        end
    else
        errordlg('Preencha todos os campos','Erro');
    end
else
    errordlg('Introduza os dados dos nós em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag5);
    menu_basico_fich_dados_pag4()
end
function noprf_Callback(hObject, eventdata, handles)
NOPRF2=get(handles.nopr, 'String');
if isempty(NOPRF2)==0
    load teste NPOIN
    NOPRF2=str2double(get(handles.nopr,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    if NOPRF2<1 || NOPRF2>NPOIN || isnan(NOPRF2)
        set(handles.nopr,'String','');
        if NOPRF2<1
            errordlg('Valor de "NOPRF" tem que ser maior ou igual a 1','Erro');
        elseif NOPRF2>NPOIN
            errordlg('Valor de "NOPRF" nao pode ser maior que "NPOIN','Erro');
        elseif isnan(NOPRF2)
            errordlg('Valor de "NOPRF" tem que ser numérico','Erro');
        end
    elseif Contagem>1
        load teste5 NOPRF
        for i=1:Contagem-1
            if NOPRF2==NOPRF(1,i)
                set(handles.nopr,'String','');
                errordlg('O número do nó já está a ser usado','Erro');
            end
        end
    end
end
```

```

    end
end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
NOPRF=str2double(get(handles.nopr,'String'));
PREPR=str2double(get(handles.prepr,'String'));
if isnan(NOPRF)==0 && isnan(PREPR)==0
    if Contagem>1
        load teste5 NOPRF PREPR
    end
    NOPRF(1,Contagem)=str2double(get(handles.nopr,'String'));
    PREPR(1,Contagem)=str2double(get(handles.prepr,'String'));
    save teste5 NOPRF PREPR
    resposta=questdlg('Nó seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.nopr,'String','');
        set(handles.prepr,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste2 NPRFX
        if Contagem==NPRFX
            set(handles.pushbutton4,'Visible','off');
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Nó anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste5 NOPRF PREPR
    set(handles.pushbutton4,'Visible','on');
    set(handles.nopr,'String',NOPRF(1,Contagem));
    set(handles.prepr,'String',PREPR(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste2 NPRFX
if NPRFX==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.
function axes1_CreateFcn(hObject, eventdata, handles)
Ver function axes1_CreateFcn do ficheiro menu_basico_fich_dados_pag3.
function pushbutton6_Callback(hObject, eventdata, handles)
load teste unidimensional bidimensional
if unidimensional==1
    winopen('unidimensional.png')
elseif bidimensional==1
    winopen('bidimensional.png')
end

function varargout = menu_basico_fich_dados_pag6(varargin)

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag6);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
ESSE=1;
AALFA=str2double(get(handles.aalfa,'String'));

```

```

GAMAW=str2double(get(handles.gamaw,'String'));
PERMB=str2double(get(handles.permb,'String'));
if isnan(ESSE)==0 && isnan(AALFA)==0 && isnan(GAMAW)==0 && isnan(PERMB)==0
    PERMB=[PERMB 0 0 PERMB];
    save teste6 ESSE AALFA GAMAW PERMB
    resposta=questdlg('Seguinte?',','Sim','Nao','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_basico_fich_dados_pag6);
        menu_basico_fich_dados_pag7()
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag6);
    menu_basico_fich_dados_pag5()
end
function permb_Callback(hObject, eventdata, handles)
PERMB=get(handles.permb,'String');
if isempty(PERMB)==0
    PERMB=str2double(get(handles.permb,'String'));
    if isnan(PERMB)
        set(handles.permb,'String','');
        if isnan(PERMB)
            errordlg('Valor de "PERMB" tem que ser numérico','Erro');
        end
    end
end
end
function aalfa_Callback(hObject, eventdata, handles)
AALFA=get(handles.aalfa,'String');
if isempty(AALFA)==0
    AALFA=str2double(get(handles.aalfa,'String'));
    if isnan(AALFA)
        set(handles.aalfa,'String','');
        if isnan(AALFA)
            errordlg('Valor de "AALFA" tem que ser numérico','Erro');
        end
    end
end
end
function gamaw_Callback(hObject, eventdata, handles)
GAMAW=get(handles.gamaw,'String');
if isempty(GAMAW)==0
    GAMAW=str2double(get(handles.gamaw,'String'));
    if isnan(GAMAW)
        set(handles.gamaw,'String','');
        if isnan(GAMAW)
            errordlg('Valor de "GAMAW" tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

#### **function varargout = menu\_basico\_fich\_dados\_pag7(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag7);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste2 NMATS
Numat=str2double(get(handles.numat,'String'));
if Numat==NMATS
    PROPS1=str2double(get(handles.props1,'String'));
    PROPS2=str2double(get(handles.props2,'String'));
    PROPS3=str2double(get(handles.props3,'String'));
    PROPS4=str2double(get(handles.props4,'String'));
    if isnan(PROPS1)==0 && isnan(PROPS2)==0 && isnan(PROPS3)==0 && isnan(PROPS4)==0
        if Numat>1

```

```

load teste7 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
end
NUMAT(1,Numat)=str2double(get(handles.numat,'String'));
PROPS1(1,Numat)=str2double(get(handles.props1,'String'));
PROPS2(1,Numat)=str2double(get(handles.props2,'String'));
PROPS3(1,Numat)=str2double(get(handles.props3,'String'));
PROPS4(1,Numat)=str2double(get(handles.props4,'String'));
PROPS5(1,Numat)=0;
save teste7 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
resposta=questdlg('Seguinte?',',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag7);
    menu_basico_fich_dados_pag8()
end
else
    errordlg('Preencha todos os campos','Erro');
end
end
errordlg('Introduza os dados dos materiais em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag7);
    menu_basico_fich_dados_pag6()
end
function props3_Callback(hObject, eventdata, handles)
PROPS32=get(handles.props3,'String');
if isempty(PROPS32)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    if PROPS32<0 || isnan(PROPS32)
        set(handles.props3,'String','');
        if PROPS32<0
            errordlg('Valor de "PROPS3" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS32)
            errordlg('Valor de "PROPS3" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS22)==0 && isnan(PROPS42)==0
        load teste7 PROPS1 PROPS2 PROPS3 PROPS4
        for i=1:NUMAT-1
            if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i)
                set(handles.props3,'String','');
                errordlg('Já existe um material com as mesmas propriedades','Erro');
            end
        end
    end
end
function props1_Callback(hObject, eventdata, handles)
PROPS12=get(handles.props1,'String');
if isempty(PROPS12)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    if PROPS12<0 || isnan(PROPS12)
        set(handles.props1,'String','');
        if PROPS12<0
            errordlg('Valor de "PROPS1" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS12)
            errordlg('Valor de "PROPS1" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS22)==0 && isnan(PROPS32)==0 && isnan(PROPS42)==0
        load teste7 PROPS1 PROPS2 PROPS3 PROPS4
        for i=1:NUMAT-1
            if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i)
                set(handles.props1,'String','');
                errordlg('Já existe um material com as mesmas propriedades','Erro');
            end
        end
    end
end
end
end

```



```

function pushbutton4_Callback(hObject, eventdata, handles)
Numat=str2double(get(handles.numat,'String'));
PROPS1=str2double(get(handles.props1,'String'));
PROPS2=str2double(get(handles.props2,'String'));
PROPS3=str2double(get(handles.props3,'String'));
PROPS4=str2double(get(handles.props4,'String'));
if isnan(PROPS1)==0 && isnan(PROPS2)==0 && isnan(PROPS3)==0 && isnan(PROPS4)==0
    if Numat>1
        load teste7 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    end
    NUMAT(1,Numat)=str2double(get(handles.numat,'String'));
    PROPS1(1,Numat)=str2double(get(handles.props1,'String'));
    PROPS2(1,Numat)=str2double(get(handles.props2,'String'));
    PROPS3(1,Numat)=str2double(get(handles.props3,'String'));
    PROPS4(1,Numat)=str2double(get(handles.props4,'String'));
    PROPS5(1,Numat)=0;
    save teste7 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    resposta=questdlg('Material seguinte?', 'Sim', 'Nao', 'Sim');
    if strcmp(resposta, 'Sim')
        Numat=Numat+1;
        set(handles.numat, 'String', Numat);
        set(handles.props1, 'String', '');
        set(handles.props2, 'String', '');
        set(handles.props3, 'String', '');
        set(handles.props4, 'String', '');
        set(handles.pushbutton5, 'Visible', 'on');
        load teste2 NMATS
        if Numat==NMATS
            set(handles.pushbutton4, 'Visible', 'off');
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function props2_Callback(hObject, eventdata, handles)
PROPS22=get(handles.props2,'String');
if isempty(PROPS22)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    if PROPS22<0 || isnan(PROPS22)
        set(handles.props2,'String','');
        if PROPS22<0
            errordlg('Valor de "PROPS2" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS22)
            errordlg('Valor de "PROPS2" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS32)==0 && isnan(PROPS42)==0
        load teste7 PROPS1 PROPS2 PROPS3 PROPS4
        for i=1:NUMAT-1
            if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i)
                set(handles.props2,'String','');
                errordlg('Já existe um material com as mesmas propriedades','Erro');
            end
        end
    end
end
end
function props4_Callback(hObject, eventdata, handles)
PROPS42=get(handles.props4,'String');
if isempty(PROPS42)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    if PROPS42<0 || isnan(PROPS42)
        set(handles.props4,'String','');
        if PROPS42<0
            errordlg('Valor de "PROPS4" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS42)
            errordlg('Valor de "PROPS4" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS22)==0 && isnan(PROPS32)==0
        load teste7 PROPS1 PROPS2 PROPS3 PROPS4
    end
end

```

```

for i=1:NUMAT-1
    if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i)
        set(handles.props4,'String','');
        errordlg('Já existe um material com as mesmas propriedades','Erro');
    end
end
end
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Material anterior?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Numat=str2double(get(handles.numat,'String'));
    Numat=Numat-1;
    set(handles.numat,'String',Numat);
    if Numat==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste8 PROPS1 PROPS2 PROPS3 PROPS4
    set(handles.pushbutton4,'Visible','on');
    set(handles.props1,'String',PROPS1(1,Numat));
    set(handles.props2,'String',PROPS2(1,Numat));
    set(handles.props3,'String',PROPS3(1,Numat));
    set(handles.props4,'String',PROPS4(1,Numat));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste2 NMATS
if NMATS==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

#### **function varargout = menu\_basico\_fich\_dados\_pag8(varargin)**

```

function incrt_Callback(hObject, eventdata, handles)
INCR= get(handles.incrt,'String');
if isempty(INCR)==0
    INCR=str2double(get(handles.incrt,'String'));
    if INCR<0 || isnan(INCR)
        set(handles.incrt,'String','');
        if INCR<0
            errordlg('Valor de "INCR" tem que ser maior ou igual a 0','Erro');
        elseif isnan(INCR)
            errordlg('Valor de "INCR" tem que ser numérico','Erro');
        end
    end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag8);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste2 NINCR
Contagem=str2double(get(handles.contagem,'String'));
if Contagem==NINCR
    INCR=str2double(get(handles.incrt,'String'));
    if isnan(INCR)==0
        if Contagem>1
            load teste8 INCR ISOUT
        end
        INCR(1,Contagem)=str2double(get(handles.incrt,'String'));
        ISOUT(1,Contagem)=1;
        save teste8 INCR ISOUT
        resposta=questdlg('Seguinte?', 'Sim','Não','Sim');
        if strcmp(resposta,'Sim')
            delete(menu_basico_fich_dados_pag8);
            menu_basico_fich_dados_pag9()
        end
    else

```

```

        errordlg('Preencha todos os campos','Erro');
    end
else
    errordlg('Introduza os dados dos incrementos de tempo em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag8);
    menu_basico_fich_dados_pag7()
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
INCR=str2double(get(handles.incr,'String'));
if isnan(INCR)==0
    if Contagem>1
        load teste8 INCR ISOUT
    end
    INCR(1,Contagem)=str2double(get(handles.incr,'String'));
    ISOUT(1,Contagem)=1;
    save teste8 INCR ISOUT
    resposta=questdlg('Incremento de tempo seguinte?','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.incr,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste2 NINCR
        if Contagem==NINCR
            set(handles.pushbutton4,'Visible','off');
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Incremento de tempo anterior?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste8 INCR
    set(handles.pushbutton4,'Visible','on');
    set(handles.incr,'String',INCR(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste2 NINCR
if NINCR==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

#### **function varargout = menu\_basico\_fich\_dados\_pag9(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag9);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
IGRAV = str2double(get(handles.igrav,'String'));
INIST = str2double(get(handles.inist,'String'));
if isnan(IGRAV)==0 && isnan(INIST)==0
    resposta=questdlg('Seguinte?','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        save teste9
    end
end

```

```

delete(menu_basico_fich_dados_pag9);
if IGRAV==1
    menu_basico_fich_dados_pag10()
elseif INIST==1
    menu_basico_fich_dados_pag11()
else
    menu_basico_fich_dados_pag13()
end
end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag9);
    menu_basico_fich_dados_pag8()
end
function igrav_Callback(hObject, eventdata, handles)
IGRAV=get(handles.igrav,'String');
if isempty(IGRAV)==0
    IGRAV=str2double(get(handles.igrav,'String'));
    if IGRAV<0 || IGRAV>1 || isnan(IGRAV)
        set(handles.igrav,'String','');
        if IGRAV<0 || IGRAV>1
            errordlg('Valor de "IGRAV" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(IGRAV)
            errordlg('Valor de "IGRAV" tem que ser numérico','Erro');
        end
    end
end
end
function inist_Callback(hObject, eventdata, handles)
INIST=get(handles.inist,'String');
if isempty(INIST)==0
    INIST=str2double(get(handles.inist,'String'));
    if INIST<0 || INIST>1 || isnan(INIST)
        set(handles.inist,'String','');
        if INIST<0 || INIST>1
            errordlg('Valor de "INIST" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(INIST)
            errordlg('Valor de "INIST" tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

### **function varargout = menu\_basico\_fich\_dados\_pag10(varargin)**

```

function gravity_Callback(hObject, eventdata, handles)
GRAVY1 = get(handles.gravity,'String');
if isempty(GRAVY1)==0
    GRAVY1 = str2double(get(handles.gravity,'String'));
    if isnan(GRAVY1)
        set(handles.gravity,'String','');
        if isnan(GRAVY1)
            errordlg('Valor de "GRAVY" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag10);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
THETA = str2double(get(handles.theta,'String'));
GRAVY1 = str2double(get(handles.gravity,'String'));
if isnan(THETA)==0 && isnan(GRAVY1)==0
    save teste10 THETA GRAVY1
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')

```

```

        delete(menu_basico_fich_dados_pag10);
        load teste9 INIST
        if INIST==1
            menu_basico_fich_dados_pag11()
        else
            menu_basico_fich_dados_pag13()
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag10);
    menu_basico_fich_dados_pag9()
end
function theta_Callback(hObject, eventdata, handles)
THETA = get(handles.theta,'String');
if isempty(THETA)==0
    THETA = str2double(get(handles.theta,'String'));
    if isnan(THETA)
        set(handles.theta,'String','');
        if isnan(THETA)
            errordlg('Valor de "THETA" tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

#### **function varargout = menu\_basico\_fich\_dados\_pag11(varargin)**

```

function gravity_Callback(hObject, eventdata, handles)
GRAVY2 = get(handles.gravity,'String');
if isempty(GRAVY2)==0
    GRAVY2 = str2double(get(handles.gravity,'String'));
    if isnan(GRAVY2)
        set(handles.gravity,'String','');
        if isnan(GRAVY2)
            errordlg('Valor de "GRAVY" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag11);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
SURCO = str2double(get(handles.surco,'String'));
GRAVY2 = str2double(get(handles.gravity,'String'));
if isnan(SURCO)==0 && isnan(GRAVY2)==0
    save teste11 SURCO GRAVY2
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_basico_fich_dados_pag11);
        menu_basico_fich_dados_pag12()
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag11);
    load teste9 IGRAV
    if IGRAV==1
        menu_basico_fich_dados_pag10()
    else
        menu_basico_fich_dados_pag9()
    end
end

```

```

end
function surco_Callback(hObject, eventdata, handles)
SURCO = get(handles.surco,'String');
if isempty(SURCO)==0
    SURCO = str2double(get(handles.surco,'String'));
    if isnan(SURCO)
        set(handles.surco,'String','');
        if isnan(SURCO)
            errordlg('Valor de "SURCO" tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

**function varargout = menu\_basico\_fich\_dados\_pag12(varargin)**

```

function ko_Callback(hObject, eventdata, handles)
KO = get(handles.ko,'String');
if isempty(KO)==0
    KO = str2double(get(handles.ko,'String'));
    if isnan(KO)
        set(handles.ko,'String','');
        if isnan(KO)
            errordlg('Valor de "KO" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag12);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat nome_ficheiros.txt
    menu_principal()
end
end
function pushbutton2_Callback(hObject, eventdata, handles)
IK = str2double(get(handles.ik,'String'));
if IK==0
    KO = str2double(get(handles.ko,'String'));
    if isnan(KO)==0
        Sair=1;
    else
        Sair=0;
    end
end
else
    Sair=1;
end
if isnan(IK)==0 && Sair==1
    save teste12
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_basico_fich_dados_pag12);
        menu_basico_fich_dados_pag13()
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag12);
    menu_basico_fich_dados_pag11()
end
end
function ik_Callback(hObject, eventdata, handles)
IK = get(handles.ik,'String');
if isempty(IK)==0
    IK = str2double(get(handles.ik,'String'));
    if isnan(IK) || IK~=0 && IK~=1
        set(handles.ik,'String','');
        set(handles.text4,'visible','off');
        set(handles.ko,'visible','off');
        if isnan(IK)
            errordlg('Valor de "IK" tem que ser numérico','Erro');
        elseif IK~=0 && IK~=1

```

```

        errordlg('Valor de "IK" tem que ser igual a 0 ou 1','Erro');
    end
elseif IK==0
    set(handles.text4,'visible','on');
    set(handles.ko,'visible','on');
else
    set(handles.text4,'visible','off');
    set(handles.ko,'visible','off');
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

**function varargout = menu\_basico\_fich\_dados\_pag13(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag13);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
PRESS11=str2double(get(handles.press11,'String'));
PRESS12=str2double(get(handles.press12,'String'));
if isnan(PRESS11)==0 && isnan(PRESS12)==0
    load teste NEDGE NPOIN
    PRESS11(1,1:NEDGE)=str2double(get(handles.press11,'String'))*-1;
    PRESS12(1,1:NEDGE)=str2double(get(handles.press12,'String'))*-1;
    PRESS21=PRESS11;
    PRESS22=PRESS12;
    PRESS31=PRESS11;
    PRESS32=PRESS12;
    IBODY=0;
    IPOUT=1;
    QPONT=NPOIN;
    CAUDL=0;
    NGASH1=NPOIN;
    XGASH=0;
    YGASH=0;
    NGASH2=NPOIN;
    PGASH=0;
    save teste13
    resposta=questdlg('Concluir?', 'Sim','Nao','Sim');
    pause(0.1)
    if strcmp(resposta,'Sim')
        resposta2=questdlg('Criar ficheiro de resultados?', 'Sim','Nao','Sim');
        pause(0.1)
        if strcmp(resposta2,'Sim')
            resposta3=questdlg('Ver resultados?', 'Sim','Nao','Sim');
            pause(0.1)
        end
        basico=1;
        escrita_ficheiro_dados()
        pause(0.1)
        if filename_DAT~=0
            delete(menu_basico_fich_dados_pag13);
            delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
teste12.mat teste13.mat nome_ficheiros.txt
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_basico_fich_dados_pag13);
    load teste9 IGRAV INIST
    if INIST==1
        menu_basico_fich_dados_pag12()
    elseif IGRAV==1
        menu_basico_fich_dados_pag10()
    else

```

```

    menu_basico_fich_dados_pag9()
end
end
function press11_Callback(hObject, eventdata, handles)
PRESS11 = get(handles.press11,'String');
if isempty(PRESS11)==0
    PRESS11 = str2double(get(handles.press11,'String'));
    if isnan(PRESS11)
        set(handles.press11,'String','');
        if isnan(PRESS11)
            errordlg('Valor de "PRESS(1,1)" tem que ser numérico','Erro');
        end
    end
end
end
function press12_Callback(hObject, eventdata, handles)
PRESS12 = get(handles.press12,'String');
if isempty(PRESS12)==0
    PRESS12 = str2double(get(handles.press12,'String'));
    if isnan(PRESS12)
        set(handles.press12,'String','');
        if isnan(PRESS12)
            errordlg('Valor de "PRESS(1,2)" tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_basico_fich_dados_pag1.

```

## A1.1.2. Interface de introdução de dados avançada

### **function varargout = menu\_fich\_dados\_pag1(varargin)**

```

function npoin_Callback(hObject, eventdata, handles)
NPOIN = get(handles.npoin,'String');
if isempty(NPOIN)==0
    NPOIN = str2double(get(handles.npoin,'String'));
    if NPOIN<1 || isnan(NPOIN)
        set(handles.npoin,'String','');
        if NPOIN<1
            errordlg('Valor de "NPOIN" tem que ser maior ou igual a 1','Erro');
        elseif isnan(NPOIN)
            errordlg('Valor de "NPOIN" tem que ser numérico','Erro');
        end
    end
end
end
function nmats_Callback(hObject, eventdata, handles)
NMATS = get(handles.nmats,'String');
if isempty(NMATS)==0
    NELEM = str2double(get(handles.nelem,'String'));
    NMATS = str2double(get(handles.nmats,'String'));
    if NMATS<1 || NMATS>NELEM || isnan(NMATS) || isnan(NELEM)
        set(handles.nmats,'String','');
        if isnan(NELEM)
            errordlg('Introduza primeiro o valor de "NELEM"', 'Erro');
        elseif NMATS<1
            errordlg('Valor de "NMATS" tem que ser maior ou igual a 1','Erro');
        elseif NMATS>NELEM
            errordlg('Valor de "NMATS" não pode ser maior que "NELEM"', 'Erro');
        elseif isnan(NMATS)
            errordlg('Valor de "NMATS" tem que ser numérico','Erro');
        end
    end
end
end
function linca_Callback(hObject, eventdata, handles)
Ver function linca_Callback do ficheiro menu_basico_fich_dados_pag2.
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag1);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
end

```



```

function pushbutton2_Callback(hObject, eventdata, handles)
NPOIN = str2double(get(handles.npoin,'String'));
NELEM = str2double(get(handles.nelem,'String'));
NVFIX = str2double(get(handles.nvfix,'String'));
NPRFX = str2double(get(handles.nprfx,'String'));
NTYPE = str2double(get(handles.ntype,'String'));
MNODE = str2double(get(handles.mnode,'String'));
NMATS = str2double(get(handles.nmats,'String'));
NGAUS = str2double(get(handles.ngaus,'String'));
TELEM = str2double(get(handles.telem,'String'));
NINCR = str2double(get(handles.nincr,'String'));
if TELEM==2
    NPPRE = str2double(get(handles.nppre,'String'));
    if isnan(NPPRE)==0
        Sair=1;
    else
        Sair=0;
    end
else
    Sair=1;
end
LINCA = str2double(get(handles.linca,'String'));
titulo_problema='bidimensional ';
NDOFN=2;
NPROP=5;
NDIME=2;
NSTRE=3;
if isempty(titulo_problema)==0 && isnan(NPOIN)==0 && isnan(NELEM)==0 && isnan(NVFIX)==0 && isnan(NPRFX)==0 &&
isnan(NTYPE)==0 && isnan(MNODE)==0 && isnan(NMATS)==0 && isnan(NGAUS)==0 && isnan(NSTRE)==0 &&
isnan(TELEM)==0 && isnan(NINCR)==0 && isnan(LINCA)==0 && Sair==1
    resposta=questdlg('Seguinte?',','Sim','Nao','Sim');
    if strcmp(resposta,'Sim')
        save teste
        delete(menu_fich_dados_pag1);
        menu_fich_dados_pag2()
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function nelem_Callback(hObject, eventdata, handles)
NELEM = get(handles.nelem,'String');
if isempty(NELEM)==0
    NPOIN = str2double(get(handles.npoin,'String'));
    NELEM = str2double(get(handles.nelem,'String'));
    if NELEM<1 || NELEM>NPOIN || isnan(NELEM) || isnan(NPOIN)
        set(handles.nelem,'String','');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif NELEM<1
            errordlg('Valor de "NELEM" tem que ser maior ou igual a 1', 'Erro');
        elseif NELEM>NPOIN
            errordlg('Valor de "NELEM" não pode ser maior que "NPOIN"', 'Erro');
        elseif isnan(NELEM)
            errordlg('Valor de "NELEM" tem que ser numérico', 'Erro');
        end
    end
end
function nvfix_Callback(hObject, eventdata, handles)
NVFIX = get(handles.nvfix,'String');
if isempty(NVFIX)==0
    NPOIN = str2double(get(handles.npoin,'String'));
    NVFIX = str2double(get(handles.nvfix,'String'));
    if NVFIX<1 || NVFIX>NPOIN || isnan(NVFIX) || isnan(NPOIN)
        set(handles.nvfix,'String','');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif NVFIX<1
            errordlg('Valor de "NVFIX" tem que ser maior ou igual a 1', 'Erro');
        elseif NVFIX>NPOIN
            errordlg('Valor de "NVFIX" não pode ser maior que "NPOIN"', 'Erro');
        elseif isnan(NVFIX)
            errordlg('Valor de "NVFIX" tem que ser numérico', 'Erro');
        end
    end
end
function nprfx_Callback(hObject, eventdata, handles)

```

```

NPRFX = get(handles.nprfx,'String');
if isempty(NPRFX)==0
    NPOIN = str2double(get(handles.npoin,'String'));
    NPRFX = str2double(get(handles.nprfx,'String'));
    if NPRFX<0 || NPRFX>NPOIN || isnan(NPRFX) || isnan(NPOIN)
        set(handles.nprfx,'String','');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif NPRFX<0
            errordlg('Valor de "NPRFX" tem que ser maior ou igual a 0','Erro');
        elseif NPRFX>NPOIN
            errordlg('Valor de "NPRFX" não pode ser maior que "NPOIN"', 'Erro');
        elseif isnan(NPRFX)
            errordlg('Valor de "NPRFX" tem que ser numérico','Erro');
        end
    end
end
function ntype_Callback(hObject, eventdata, handles)
NTYPE = get(handles.ntype,'String');
if isempty(NTYPE)==0
    NTYPE = str2double(get(handles.ntype,'String'));
    if NTYPE<1 || NTYPE>3 || isnan(NTYPE)
        set(handles.ntype,'String','');
        if NTYPE<1 || NTYPE>3
            errordlg('Valor de "NTYPE" só ser igual a 1, 2 ou 3','Erro');
        elseif isnan(NTYPE)
            errordlg('Valor de "NTYPE" tem que ser numérico','Erro');
        end
    end
end
function mnode_Callback(hObject, eventdata, handles)
MNODE = get(handles.mnode,'String');
if isempty(MNODE)==0
    NPOIN = str2double(get(handles.npoin,'String'));
    NELEM = str2double(get(handles.nelem,'String'));
    MNODE = str2double(get(handles.mnode,'String'));
    if MNODE~=4 && MNODE~=8 && MNODE~=9 || NELEM*MNODE<NPOIN || isnan(MNODE) || isnan(NPOIN) || isnan(NELEM)
        set(handles.mnode,'String','');
        if isnan(NPOIN)
            errordlg('Introduza primeiro o valor de "NPOIN"', 'Erro');
        elseif isnan(NELEM)
            errordlg('Introduza primeiro o valor de "NELEM"', 'Erro');
        elseif isnan(MNODE)
            errordlg('Valor de "MNODE" tem que ser numérico','Erro');
        elseif MNODE~=4 && MNODE~=8 && MNODE~=9
            errordlg('Valor de "MNODE" só ser igual a 4, 8 ou 9','Erro');
        elseif NELEM*MNODE<NPOIN
            errordlg('Valor de "NELEMxMNODE" não pode ser menor que "NPOIN"', 'Erro');
        end
    end
end
function ngaus_Callback(hObject, eventdata, handles)
Ver function ngaus_Callback do ficheiro menu_basico_fich_dados_pag2.
function nincr_Callback(hObject, eventdata, handles)
Ver function nincr_Callback do ficheiro menu_basico_fich_dados_pag2.
function telem_Callback(hObject, eventdata, handles)
Ver function telem_Callback do ficheiro menu_basico_fich_dados_pag2.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
resposta=questdlg('Fechar programa?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(hObject);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
end

```

### **function varargout = menu\_fich\_dados\_pag2(varargin)**

```

function matno_Callback(hObject, eventdata, handles)
MATNO = get(handles.matno,'String');
if isempty(MATNO)==0
    load teste NMATS
    MATNO = str2double(get(handles.matno,'String'));
    if MATNO<1 || MATNO>NMATS || isnan(MATNO)
        set(handles.matno,'String','');
        if MATNO<1

```

```

        errordlg('Valor de "MATNO" tem que ser maior ou igual a 1','Erro');
    elseif MATNO>NMATS
        errordlg('Valor de "MATNO" nao pode ser maior que "NMATS','Erro');
    elseif isnan(MATNO)
        errordlg('Valor de "MATNO" tem que ser numérico','Erro');
    end
end
end
function matno_CreateFcn(hObject, eventdata, handles)
load teste NMATS
if NMATS==1
    set(hObject,'backgroundcolor',[.502 .502 .502]);
    set(hObject,'foregroundcolor',[1 1 1]);
    set(hObject,'enable','inactive');
    set(hObject,'string','1');
else
    set(hObject,'backgroundcolor',[1 1 1]);
    set(hObject,'foregroundcolor',[0 0 0]);
    set(hObject,'enable','on');
    set(hObject,'string','');
end
function nnode_Callback(hObject, eventdata, handles)
Ver function nnode_Callback do ficheiro menu_basico_fich_dados_pag3.
function nnode_CreateFcn(hObject, eventdata, handles)
Ver function nnode_CreateFcn do ficheiro menu_basico_fich_dados_pag3.
function lnods_Callback(hObject, eventdata, handles)
Ver function lnods_Callback do ficheiro menu_basico_fich_dados_pag3.
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag2);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste NELEM TELEM
Numel=str2double(get(handles.numel,'String'));
MATNO = str2double(get(handles.matno,'String'));
NNODE = str2double(get(handles.nnode,'String'));
LNODS = str2num(get(handles.lnods,'String'));
tamanho_LNODS=size(LNODS);
if NELEM==Numel && isnan(MATNO)==0 && tamanho_LNODS(2)==NNODE
    load teste2 NUMEL MATNO NNODE LNODS
    NUMEL(1,NELEM) = str2double(get(handles.numel,'String'));
    MATNO(1,NELEM) = str2double(get(handles.matno,'String'));
    NNODE(1,NELEM) = str2double(get(handles.nnode,'String'));
    LNODS(NELEM,1:NNODE(1,NELEM)) = str2num(get(handles.lnods,'String'));
    save teste2 NUMEL MATNO NNODE LNODS
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_fich_dados_pag2);
        if TELEM==2
            menu_fich_dados_pag3()
        else
            menu_fich_dados_pag4()
        end
    end
end
elseif isnan(MATNO) || tamanho_LNODS(2)~=NNODE
    errordlg('Preencha todos os campos','Erro');
elseif NELEM~=Numel
    errordlg('Introduza os dados dos elementos em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag2);
    menu_fich_dados_pag1()
end
function pushbutton4_Callback(hObject, eventdata, handles)
resposta=questdlg('Elemento anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Numel=str2double(get(handles.numel,'String'));
    Numel=Numel-1;
    set(handles.numel,'String',Numel);
    if Numel==1

```

```

        set(handles.pushbutton4,'Visible','off');
    end
    load teste2 LNODS NNODE MATNO
    load teste NMATS
    set(handles.pushbutton5,'Visible','on');
    if NMATS>1
        set(handles.matno,'String',MATNO(1,Numel));
    end
    set(handles.lnodes,'String',num2str(LNODS(Numel,1:NNODE(1,Numel))));
end
function pushbutton5_Callback(hObject, eventdata, handles)
load teste NELEM NMATS
Numel = str2double(get(handles.numel,'String'));
MATNO = str2double(get(handles.matno,'String'));
NNODE = str2double(get(handles.nnode,'String'));
LNODS = str2num(get(handles.lnodes,'String'));
tamanho_LNODS=size(LNODS);
if isnan(MATNO)==0 && tamanho_LNODS(2)==NNODE
    if Numel>1
        load teste2 NUMEL MATNO NNODE LNODS
    end
    NUMEL(1,Numel)=str2double(get(handles.numel,'String'));
    MATNO(1,Numel)=str2double(get(handles.matno,'String'));
    NNODE(1,Numel)=str2double(get(handles.nnode,'String'));
    LNODS(Numel,1:NNODE(1,Numel))=str2num(get(handles.lnodes,'String'));
    resposta=questdlg('Elemento seguinte?', 'Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Numel=Numel+1;
        if Numel==NELEM
            set(handles.pushbutton5,'Visible','off');
        end
        set(handles.pushbutton4,'Visible','on');
        if NMATS>1
            set(handles.matno,'String','');
        end
        set(handles.lnodes,'String','');
        set(handles.numel,'String',Numel);
        save teste2 NUMEL MATNO NNODE LNODS
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton5_CreateFcn(hObject, eventdata, handles)
Ver function pushbutton5_CreateFcn do ficheiro menu_basico_fich_dados_pag3.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag4(varargin)**

```

function ipoin_Callback(hObject, eventdata, handles)
IPOIN2=get(handles.ipoin,'String');
if isempty(IPOIN2)==0
    load teste NPOIN
    IPOIN2=str2double(get(handles.ipoin,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    if IPOIN2<1 || IPOIN2>NPOIN || isnan(IPOIN2)
        set(handles.ipoin,'String','');
        if IPOIN2<1
            errordlg('Valor de "IPOIN" tem que ser maior ou igual a 1','Erro');
        elseif IPOIN2>NPOIN
            errordlg('Valor de "IPOIN" nao pode ser maior que "NPOIN','Erro');
        elseif isnan(IPOIN2)
            errordlg('Valor de "IPOIN" tem que ser numérico','Erro');
        end
    elseif Contagem>1
        load teste4 IPOIN
        for i=1:Contagem-1
            if IPOIN2==IPOIN(i,1)
                set(handles.ipoin,'String','');
                errordlg('O número do nó já está a ser usado','Erro');
            end
        end
    end
end
function coord2_Callback(hObject, eventdata, handles)

```

```

COORD22=get(handles.coord2,'String');
if isempty(COORD22)==0
    COORD22=str2double(get(handles.coord2,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    COORD12=str2double(get(handles.coord1,'String'));
    if COORD22<0 || isnan(COORD22)
        set(handles.coord2,'String','');
        if COORD22<0
            errordlg('Valor de "COORD2" tem que ser maior ou igual a 0','Erro');
        elseif isnan(COORD22)
            errordlg('Valor de "COORD2" tem que ser numérico','Erro');
        end
    elseif isnan(COORD12)==0 && Contagem>1
        load teste4 COORD1 COORD2
        for i=1:Contagem-1
            if COORD12==COORD1(i,1) && COORD22==COORD2(i,1)
                set(handles.coord2,'String','');
                errordlg('Já existe um nó com as mesmas coordenadas','Erro');
            end
        end
    end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag4);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
IPOIN=str2double(get(handles.ipoin,'String'));
COORD1=str2double(get(handles.coord1,'String'));
COORD2=str2double(get(handles.coord2,'String'));
if isnan(IPOIN)==0 && isnan(COORD1)==0 && isnan(COORD2)==0
    if Contagem>1
        load teste4 IPOIN COORD1 COORD2
    end
    IPOIN(Contagem,1)=str2double(get(handles.ipoin,'String'));
    COORD1(Contagem,1)=str2double(get(handles.coord1,'String'));
    COORD2(Contagem,1)=str2double(get(handles.coord2,'String'));
    save teste4 IPOIN COORD1 COORD2
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_fich_dados_pag4);
        load teste TELEM
        if TELEM==2
            menu_fich_dados_pag5()
        else
            menu_fich_dados_pag5b()
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag4);
    load teste TELEM
    if TELEM==2
        menu_fich_dados_pag3()
    else
        menu_fich_dados_pag2()
    end
end
function coord1_Callback(hObject, eventdata, handles)
COORD12=get(handles.coord1,'String');
if isempty(COORD12)==0
    COORD12=str2double(get(handles.coord1,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    COORD22=str2double(get(handles.coord2,'String'));
    if COORD12<0 || isnan(COORD12)
        set(handles.coord1,'String','');
        if COORD12<0

```

```

        errordlg('Valor de "COORD1" tem que ser maior ou igual a 0','Erro');
    elseif isnan(COORD12)
        errordlg('Valor de "COORD1" tem que ser numérico','Erro');
    end
elseif isnan(COORD22)==0 && Contagem>1
    load teste4 COORD1 COORD2
    for i=1:Contagem-1
        if COORD12==COORD1(i,1) && COORD22==COORD2(i,1)
            set(handles.coord1,'String','');
            errordlg('Já existe um nó com as mesmas coordenadas','Erro');
        end
    end
end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
IPOIN=str2double(get(handles.ipoin,'String'));
COORD1=str2double(get(handles.coord1,'String'));
COORD2=str2double(get(handles.coord2,'String'));
if isnan(IPOIN)==0 && isnan(COORD1)==0 && isnan(COORD2)==0
    if Contagem>1
        load teste4 IPOIN COORD1 COORD2
    end
    IPOIN(Contagem,1)=str2double(get(handles.ipoin,'String'));
    COORD1(Contagem,1)=str2double(get(handles.coord1,'String'));
    COORD2(Contagem,1)=str2double(get(handles.coord2,'String'));
    save teste4 IPOIN COORD1 COORD2
    resposta=questdlg('Ponto nodal seguinte?','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.ipoin,'String','');
        set(handles.coord1,'String','');
        set(handles.coord2,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste NPOIN
        if Contagem==NPOIN
            set(handles.pushbutton4,'Visible','off');
        end
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Ponto nodal anterior?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste4 IPOIN COORD1 COORD2
    set(handles.pushbutton4,'Visible','on');
    set(handles.ipoin,'String',IPOIN(Contagem,1));
    set(handles.coord1,'String',COORD1(Contagem,1));
    set(handles.coord2,'String',COORD2(Contagem,1));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste NPOIN
if NPOIN==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

**function varargout = menu\_fich\_dados\_pag5b(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag5b);

```

```

delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste NVFIX
Contagem=str2double(get(handles.contagem,'String'));
if Contagem==NVFIX
NOFIX=str2double(get(handles.nofix,'String'));
IFPRE1=str2double(get(handles.ifpre1,'String'));
IFPRE2=str2double(get(handles.ifpre2,'String'));
PRESC1=str2double(get(handles.presc1,'String'));
PRESC2=str2double(get(handles.presc2,'String'));
if isnan(NOFIX)==0 && isnan(IFPRE1)==0 && isnan(IFPRE2)==0 && isnan(PRESC1)==0 && isnan(PRESC2)==0
if Contagem>1
load teste5 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
end
NOFIX(1,Contagem)=str2double(get(handles.nofix,'String'));
IFPRE1(1,Contagem)=str2double(get(handles.ifpre1,'String'));
IFPRE2(1,Contagem)=str2double(get(handles.ifpre2,'String'));
PRESC1(1,Contagem)=str2double(get(handles.presc1,'String'));
PRESC2(1,Contagem)=str2double(get(handles.presc2,'String'));
save teste5 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
resposta=questdlg('Seguinte?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
delete(menu_fich_dados_pag5b);
menu_fich_dados_pag6()
end
else
errordlg('Preencha todos os campos','Erro');
end
else
errordlg('Introduza os dados dos nós restringidos em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
delete(menu_fich_dados_pag5b);
menu_fich_dados_pag4()
end
function presc1_Callback(hObject, eventdata, handles)
Ver function presc1_Callback do ficheiro menu_basico_fich_dados_pag4.
function presc2_Callback(hObject, eventdata, handles)
Ver function presc2_Callback do ficheiro menu_basico_fich_dados_pag4.
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem,'String'));
NOFIX=str2double(get(handles.nofix,'String'));
IFPRE1=str2double(get(handles.ifpre1,'String'));
IFPRE2=str2double(get(handles.ifpre2,'String'));
PRESC1=str2double(get(handles.presc1,'String'));
PRESC2=str2double(get(handles.presc2,'String'));
if isnan(NOFIX)==0 && isnan(IFPRE1)==0 && isnan(IFPRE2)==0 && isnan(PRESC1)==0 && isnan(PRESC2)==0
if Contagem>1
load teste5 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
end
NOFIX(1,Contagem)=str2double(get(handles.nofix,'String'));
IFPRE1(1,Contagem)=str2double(get(handles.ifpre1,'String'));
IFPRE2(1,Contagem)=str2double(get(handles.ifpre2,'String'));
PRESC1(1,Contagem)=str2double(get(handles.presc1,'String'));
PRESC2(1,Contagem)=str2double(get(handles.presc2,'String'));
save teste5 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
resposta=questdlg('Nó restringido seguinte?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
Contagem=Contagem+1;
set(handles.contagem,'String',Contagem);
set(handles.nofix,'String','');
set(handles.ifpre1,'String','');
set(handles.ifpre2,'String','');
set(handles.presc1,'String','');
set(handles.presc2,'String','');
set(handles.pushbutton5,'Visible','on');
load teste NVFIX
if Contagem==NVFIX
set(handles.pushbutton4,'Visible','off');
end
end
end

```

```

else
    errordlg('Preencha todos os campos','Erro');
end
function nofix_Callback(hObject, eventdata, handles)
NOFIX2=get(handles.nofix,'String');
if isempty(NOFIX2)==0
    load teste NPOIN
    NOFIX2=str2double(get(handles.nofix,'String'));
    Contagem=str2double(get(handles.contagem,'String'));
    if NOFIX2<1 || NOFIX2>NPOIN || isnan(NOFIX2)
        set(handles.nofix,'String','');
        if NOFIX2<1
            errordlg('Valor de "NOFIX" tem que ser maior ou igual a 1','Erro');
        elseif NOFIX2>NPOIN
            errordlg('Valor de "NOFIX" nao pode ser maior que "NPOIN','Erro');
        elseif isnan(NOFIX2)
            errordlg('Valor de "NOFIX" tem que ser numérico','Erro');
        end
    elseif Contagem>1
        load teste5 NOFIX
        for i=1:Contagem-1
            if NOFIX2==NOFIX(1,i)
                set(handles.nofix,'String','');
                errordlg('O número do nó já está a ser usado','Erro');
            end
        end
    end
end
function ifpre1_Callback(hObject, eventdata, handles)
Ver function ifpre1_Callback do ficheiro menu_basico_fich_dados_pag4.
function ifpre2_Callback(hObject, eventdata, handles)
Ver function ifpre2_Callback do ficheiro menu_basico_fich_dados_pag4.
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Nó restringido anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste5 NOFIX IFPRE1 IFPRE2 PRESC1 PRESC2
    set(handles.pushbutton4,'Visible','on');
    set(handles.nofix,'String',NOFIX(1,Contagem));
    set(handles.ifpre1,'String',IFPRE1(1,Contagem));
    set(handles.ifpre2,'String',IFPRE2(1,Contagem));
    set(handles.presc1,'String',PRESC1(1,Contagem));
    set(handles.presc2,'String',PRESC2(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste NVFIX
if NVFIX==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag6(varargin)**

```

function prepr_Callback(hObject, eventdata, handles)
Ver function prepr_Callback do ficheiro menu_basico_fich_dados_pag5.
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag6);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste NPRFX
Contagem=str2double(get(handles.contagem,'String'));
if Contagem==NPRFX

```



```

NOPRF=str2double(get(handles.nopr, 'String'));
PREPR=str2double(get(handles.prepr, 'String'));
if isnan(NOPRF)==0 && isnan(PREPR)==0
    if Contagem>1
        load teste6 NOPRF PREPR
    end
    NOPRF(1,Contagem)=str2double(get(handles.nopr, 'String'));
    PREPR(1,Contagem)=str2double(get(handles.prepr, 'String'));
    save teste6 NOPRF PREPR
    resposta=questdlg('Seguinte?', 'Sim', 'Não', 'Sim');
    if strcmp(resposta, 'Sim')
        delete(menu_fich_dados_pag6);
        menu_fich_dados_pag7()
    end
else
    errordlg('Preencha todos os campos', 'Erro');
end
end
errordlg('Introduza os dados dos nós em falta', 'Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag6);
    load teste TELEM
    if TELEM==2
        menu_fich_dados_pag5()
    else
        menu_fich_dados_pag5b()
    end
end
function nopr_Callback(hObject, eventdata, handles)
NOPRF2=get(handles.nopr, 'String');
if isempty(NOPRF2)==0
    load teste NPOIN
    NOPRF2=str2double(get(handles.nopr, 'String'));
    Contagem=str2double(get(handles.contagem, 'String'));
    if NOPRF2<1 || NOPRF2>NPOIN || isnan(NOPRF2)
        set(handles.nopr, 'String', '');
        if NOPRF2<1
            errordlg('Valor de "NOPRF" tem que ser maior ou igual a 1', 'Erro');
        elseif NOPRF2>NPOIN
            errordlg('Valor de "NOPRF" nao pode ser maior que "NPOIN", 'Erro');
        elseif isnan(NOPRF2)
            errordlg('Valor de "NOPRF" tem que ser numérico', 'Erro');
        end
    elseif Contagem>1
        load teste6 NOPRF
        for i=1:Contagem-1
            if NOPRF2==NOPRF(1,i)
                set(handles.nopr, 'String', '');
                errordlg('O número do nó já está a ser usado', 'Erro');
            end
        end
    end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem=str2double(get(handles.contagem, 'String'));
NOPRF=str2double(get(handles.nopr, 'String'));
PREPR=str2double(get(handles.prepr, 'String'));
if isnan(NOPRF)==0 && isnan(PREPR)==0
    if Contagem>1
        load teste6 NOPRF PREPR
    end
    NOPRF(1,Contagem)=str2double(get(handles.nopr, 'String'));
    PREPR(1,Contagem)=str2double(get(handles.prepr, 'String'));
    save teste6 NOPRF PREPR
    resposta=questdlg('Nó seguinte?', 'Sim', 'Não', 'Sim');
    if strcmp(resposta, 'Sim')
        Contagem=Contagem+1;
        set(handles.contagem, 'String', Contagem);
        set(handles.nopr, 'String', '');
        set(handles.prepr, 'String', '');
        set(handles.pushbutton5, 'Visible', 'on');
        load teste NPRFX
        if Contagem==NPRFX

```

```

        set(handles.pushbutton4,'Visible','off');
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Nó anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste6 NOPRF PREPR
    set(handles.pushbutton4,'Visible','on');
    set(handles.nopr,'String',NOPRF(1,Contagem));
    set(handles.prepr,'String',PREPR(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste NPRFX
if NPRFX==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag7(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag7);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
ESSE=1;
AALFA = str2double(get(handles.aalfa,'String'));
GAMAW = str2double(get(handles.gamaw,'String'));
PERMB=str2num(get(handles.permb,'String'));
tamanho_PERMB=size(PERMB);
if isnan(ESSE)==0 && isnan(AALFA)==0 && isnan(GAMAW)==0 && tamanho_PERMB(2)==4
    save teste7 ESSE AALFA GAMAW PERMB
    resposta=questdlg('Seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_fich_dados_pag7);
        menu_fich_dados_pag8()
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag7);
    menu_fich_dados_pag6()
end
function permb_Callback(hObject, eventdata, handles)
PERMB=get(handles.permb,'String');
if isempty(PERMB)==0
    PERMB=str2num(get(handles.permb,'String'));
    tamanho_PERMB=size(PERMB);
    if tamanho_PERMB(2)~=4 || isempty(PERMB)
        set(handles.permb,'String','');
        if isempty(PERMB)
            errordlg('Valor de "PERMB" tem que ser numérico','Erro');
        elseif tamanho_PERMB(2)~=4
            errordlg('Tem que introduzir 4 valores (matriz 2x2)','Erro');
        end
    end
end
end

```

```

end
function aalfa_Callback(hObject, eventdata, handles)
Ver function aalfa_Callback do ficheiro menu_basico_fich_dados_pag6.
function gamaw_Callback(hObject, eventdata, handles)
Ver function gamaw_Callback do ficheiro menu_basico_fich_dados_pag6.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

**function varargout = menu\_fich\_dados\_pag8(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag8);
    delete teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste NMATS
Numat=str2double(get(handles.numat, 'String'));
if Numat==NMATS
    PROPS1=str2double(get(handles.props1, 'String'));
    PROPS2=str2double(get(handles.props2, 'String'));
    PROPS3=str2double(get(handles.props3, 'String'));
    PROPS4=str2double(get(handles.props4, 'String'));
    PROPS5=0;
if isnan(PROPS1)==0 && isnan(PROPS2)==0 && isnan(PROPS3)==0 && isnan(PROPS4)==0 && isnan(PROPS5)==0
    if Numat>1
        load teste8 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    end
    NUMAT(1, Numat)=str2double(get(handles.numat, 'String'));
    PROPS1(1, Numat)=str2double(get(handles.props1, 'String'));
    PROPS2(1, Numat)=str2double(get(handles.props2, 'String'));
    PROPS3(1, Numat)=str2double(get(handles.props3, 'String'));
    PROPS4(1, Numat)=str2double(get(handles.props4, 'String'));
    PROPS5(1, Numat)=0;
    save teste8 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    resposta=questdlg('Seguinte?', 'Sim', 'Não', 'Sim');
    if strcmp(resposta, 'Sim')
        delete(menu_fich_dados_pag8);
        menu_fich_dados_pag9()
    end
else
    errordlg('Preencha todos os campos', 'Erro');
end
else
    errordlg('Introduza os dados dos materiais em falta', 'Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim', 'Não', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag8);
    menu_fich_dados_pag7()
end
function props3_Callback(hObject, eventdata, handles)
PROPS32=get(handles.props3, 'String');
if isempty(PROPS32)==0
    NUMAT=str2double(get(handles.numat, 'String'));
    PROPS12=str2double(get(handles.props1, 'String'));
    PROPS22=str2double(get(handles.props2, 'String'));
    PROPS32=str2double(get(handles.props3, 'String'));
    PROPS42=str2double(get(handles.props4, 'String'));
    PROPS52=0;
if PROPS32<0 || isnan(PROPS32)
    set(handles.props3, 'String', '');
    if PROPS32<0
        errordlg('Valor de "PROPS3" tem que ser maior ou igual a 0', 'Erro');
    elseif isnan(PROPS32)
        errordlg('Valor de "PROPS3" tem que ser numérico', 'Erro');
    end
elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS22)==0 && isnan(PROPS42)==0 && isnan(PROPS52)==0
    load teste8 PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    for i=1:NUMAT-1

```

```

        if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i) &&
PROPS52==PROPS5(1,i)
            set(handles.props3,'String','');
            errordlg('Já existe um material com as mesmas propriedades','Erro');
        end
    end
end
function props1_Callback(hObject, eventdata, handles)
PROPS12=get(handles.props1,'String');
if isempty(PROPS12)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    PROPS52=0;
    if PROPS12<0 || isnan(PROPS12)
        set(handles.props1,'String','');
        if PROPS12<0
            errordlg('Valor de "PROPS1" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS12)
            errordlg('Valor de "PROPS1" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS22)==0 && isnan(PROPS32)==0 && isnan(PROPS42)==0 && isnan(PROPS52)==0
        load teste8 PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
        for i=1:NUMAT-1
            if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i) &&
PROPS52==PROPS5(1,i)
                set(handles.props1,'String','');
                errordlg('Já existe um material com as mesmas propriedades','Erro');
            end
        end
    end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Numat=str2double(get(handles.numat,'String'));
PROPS1=str2double(get(handles.props1,'String'));
PROPS2=str2double(get(handles.props2,'String'));
PROPS3=str2double(get(handles.props3,'String'));
PROPS4=str2double(get(handles.props4,'String'));
PROPS5=0;
if isnan(PROPS1)==0 && isnan(PROPS2)==0 && isnan(PROPS3)==0 && isnan(PROPS4)==0 && isnan(PROPS5)==0
    if Numat>1
        load teste8 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    end
    NUMAT(1,Numat)=str2double(get(handles.numat,'String'));
    PROPS1(1,Numat)=str2double(get(handles.props1,'String'));
    PROPS2(1,Numat)=str2double(get(handles.props2,'String'));
    PROPS3(1,Numat)=str2double(get(handles.props3,'String'));
    PROPS4(1,Numat)=str2double(get(handles.props4,'String'));
    PROPS5(1,Numat)=0;
    save teste8 NUMAT PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    resposta=questdlg('Material seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Numat=Numat+1;
        set(handles.numat,'String',Numat);
        set(handles.props1,'String','');
        set(handles.props2,'String','');
        set(handles.props3,'String','');
        set(handles.props4,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste NMATS
        if Numat==NMATS
            set(handles.pushbutton4,'Visible','off');
        end
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function props2_Callback(hObject, eventdata, handles)
PROPS22=get(handles.props2,'String');
if isempty(PROPS22)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));

```

```

PROPS32=str2double(get(handles.props3,'String'));
PROPS42=str2double(get(handles.props4,'String'));
PROPS52=0;
if PROPS22<0 || isnan(PROPS22)
    set(handles.props2,'String','');
    if PROPS22<0
        errordlg('Valor de "PROPS2" tem que ser maior ou igual a 0','Erro');
    elseif isnan(PROPS22)
        errordlg('Valor de "PROPS2" tem que ser numérico','Erro');
    end
elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS32)==0 && isnan(PROPS42)==0 && isnan(PROPS52)==0
    load teste8 PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    for i=1:NUMAT-1
        if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i) &&
PROPS52==PROPS5(1,i)
            set(handles.props2,'String','');
            errordlg('Já existe um material com as mesmas propriedades','Erro');
        end
    end
end
end
function props4_Callback(hObject, eventdata, handles)
PROPS42=get(handles.props4,'String');
if isempty(PROPS42)==0
    NUMAT=str2double(get(handles.numat,'String'));
    PROPS12=str2double(get(handles.props1,'String'));
    PROPS22=str2double(get(handles.props2,'String'));
    PROPS32=str2double(get(handles.props3,'String'));
    PROPS42=str2double(get(handles.props4,'String'));
    PROPS52=0;
    if PROPS42<0 || isnan(PROPS42)
        set(handles.props4,'String','');
        if PROPS42<0
            errordlg('Valor de "PROPS4" tem que ser maior ou igual a 0','Erro');
        elseif isnan(PROPS42)
            errordlg('Valor de "PROPS4" tem que ser numérico','Erro');
        end
    elseif NUMAT>1 && isnan(PROPS12)==0 && isnan(PROPS22)==0 && isnan(PROPS32)==0 && isnan(PROPS52)==0
        load teste8 PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
        for i=1:NUMAT-1
            if PROPS12==PROPS1(1,i) && PROPS22==PROPS2(1,i) && PROPS32==PROPS3(1,i) && PROPS42==PROPS4(1,i) &&
PROPS52==PROPS5(1,i)
                set(handles.props4,'String','');
                errordlg('Já existe um material com as mesmas propriedades','Erro');
            end
        end
    end
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Material anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Numat=str2double(get(handles.numat,'String'));
    Numat=Numat-1;
    set(handles.numat,'String',Numat);
    if Numat==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste8 PROPS1 PROPS2 PROPS3 PROPS4 PROPS5
    set(handles.pushbutton4,'Visible','on');
    set(handles.props1,'String',PROPS1(1,Numat));
    set(handles.props2,'String',PROPS2(1,Numat));
    set(handles.props3,'String',PROPS3(1,Numat));
    set(handles.props4,'String',PROPS4(1,Numat));
    set(handles.props5,'String',PROPS5(1,Numat));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste NMATS
if NMATS==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

### **function varargout = menu\_fich\_dados\_pag9(varargin)**

**function** incrt\_Callback(hObject, eventdata, handles)

Ver function incrt\_Callback do ficheiro menu\_basico\_fich\_dados\_pag8.

**function** isout\_Callback(hObject, eventdata, handles)

ISOUT=get(handles.isout,'String');

if isempty(ISOUT)==0

    ISOUT=str2double(get(handles.isout,'String'));

    if ISOUT<0 || ISOUT>1 || isnan(ISOUT)

        set(handles.isout,'String','');

        if ISOUT<0 || ISOUT>1

            errordlg('Valor de "ISOUT" só pode ser igual a 0 ou 1','Erro');

        elseif isnan(ISOUT)

            errordlg('Valor de "ISOUT" tem que ser numérico','Erro');

        end

    end

end

**function** pushbutton1\_Callback(hObject, eventdata, handles)

resposta=questdlg('Menu inicial?',','Sim','Não','Sim');

if strcmp(resposta,'Sim')

    delete(menu\_fich\_dados\_pag9);

    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat teste13.mat teste14.mat teste15.mat nome\_ficheiros.txt

    menu\_principal()

end

**function** pushbutton2\_Callback(hObject, eventdata, handles)

load teste NINCR

Contagem=str2double(get(handles.contagem,'String'));

if Contagem==NINCR

    INCRT=str2double(get(handles.incr,'String'));

    ISOUT=str2double(get(handles.isout,'String'));

    if isnan(INCRT)==0 && isnan(ISOUT)==0

        if Contagem>1

            load teste9 INCRT ISOUT

        end

        INCRT(1,Contagem)=str2double(get(handles.incr,'String'));

        ISOUT(1,Contagem)=str2double(get(handles.isout,'String'));

        save teste9 INCRT ISOUT

        resposta=questdlg('Seguinte?',','Sim','Não','Sim');

        if strcmp(resposta,'Sim')

            delete(menu\_fich\_dados\_pag9);

            menu\_fich\_dados\_pag10()

        end

    else

        errordlg('Preencha todos os campos','Erro');

    end

else

    errordlg('Introduza os dados dos incrementos de tempo em falta','Erro');

end

**function** pushbutton3\_Callback(hObject, eventdata, handles)

resposta=questdlg('Anterior?',','Sim','Não','Sim');

if strcmp(resposta,'Sim')

    delete(menu\_fich\_dados\_pag9);

    menu\_fich\_dados\_pag8()

end

**function** pushbutton4\_Callback(hObject, eventdata, handles)

Contagem=str2double(get(handles.contagem,'String'));

INCRT=str2double(get(handles.incr,'String'));

ISOUT=str2double(get(handles.isout,'String'));

if isnan(INCRT)==0 && isnan(ISOUT)==0

    if Contagem>1

        load teste9 INCRT ISOUT

    end

    INCRT(1,Contagem)=str2double(get(handles.incr,'String'));

    ISOUT(1,Contagem)=str2double(get(handles.isout,'String'));

    save teste9 INCRT ISOUT

    resposta=questdlg('Incremento de tempo seguinte?',','Sim','Não','Sim');

    if strcmp(resposta,'Sim')

        Contagem=Contagem+1;

        set(handles.contagem,'String',Contagem);

        set(handles.incr,'String','');

        set(handles.isout,'String','');

        set(handles.pushbutton5,'Visible','on');

        load teste NINCR

        if Contagem==NINCR

            set(handles.pushbutton4,'Visible','off');

```

        end
    end
else
    errorldg('Preencha todos os campos','Erro');
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Incremento de tempo anterior?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste9 INCR T ISOUT
    set(handles.pushbutton4,'Visible','on');
    set(handles.incr,'String',INCR T(1,Contagem));
    set(handles.isout,'String',ISOUT(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste NINCR
if NINCR==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag10(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag10);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
TITLE='Carga distribuída';
IPLD = str2double(get(handles.iplod,'String'));
if IPLD==1
    Numero_Cargas_Pontuais=str2double(get(handles.numero_cargas_pontuais,'String'));
    if isnan(Numero_Cargas_Pontuais)==0
        Sair=1;
    else
        Sair=0;
    end
else
    Sair=1;
end
IGRAV = str2double(get(handles.igrav,'String'));
IEDGE = str2double(get(handles.iedge,'String'));
if IEDGE==1
    if Sair==1
        NEDGE=str2double(get(handles.nedge,'String'));
        if isnan(NEDGE)
            Sair=0;
        end
    end
end
INIST = str2double(get(handles.inist,'String'));
if isnan(IPLD)==0 && isnan(IGRAV)==0 && isnan(IEDGE)==0 && isnan(INIST)==0 && Sair==1
    resposta=questdlg('Seguinte?',','Sim','Nao','Sim');
    if strcmp(resposta,'Sim')
        save teste10
        delete(menu_fich_dados_pag10);
        if IPLD==1
            menu_fich_dados_pag11()
        elseif IGRAV==1
            menu_fich_dados_pag12()
        elseif INIST==1
            menu_fich_dados_pag13()
        elseif IEDGE==1

```

```

    menu_fich_dados_pag15()
else
    load teste NPOIN
    IBODY=0;
    IPOUT=1;
    QPONT=NPOIN;
    CAUDL=0;
    NGASH1=NPOIN;
    XGASH=0;
    YGASH=0;
    NGASH2=NPOIN;
    PGASH=0;
    save teste10
    resposta=questdlg('Concluir?',','Sim','Não','Sim');
    pause(0.1)
    if strcmp(resposta,'Sim')
        resposta2=questdlg('Criar ficheiro de resultados?',','Sim','Não','Sim');
        pause(0.1)
        if strcmp(resposta2,'Sim')
            resposta3=questdlg('Ver resultados?',','Sim','Não','Sim');
            pause(0.1)
        end
        basico=0;
        escrita_ficheiro_dados()
        pause(0.1)
        if filename_DAT==0
            menu_fich_dados_pag10()
        else
            delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
teste12.mat teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
        end
    end
end
end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag10);
    menu_fich_dados_pag9()
end
function igrav_Callback(hObject, eventdata, handles)
Ver function igrav_Callback do ficheiro menu_basico_fich_dados_pag9.
function iedge_Callback(hObject, eventdata, handles)
IEDGE=get(handles.iedge,'String');
if isempty(IEDGE)==0
    IEDGE=str2double(get(handles.iedge,'String'));
    if IEDGE<0 || IEDGE>1 || isnan(IEDGE)
        set(handles.iedge,'String','');
        set(handles.nedge,'Visible','off');
        set(handles.text22,'visible','off');
        if IEDGE<0 || IEDGE>1
            errordlg('Valor de "IEDGE" só pode ser igual a 0 ou 1','Erro');
        elseif isnan(IEDGE)
            errordlg('Valor de "IEDGE" tem que ser numérico','Erro');
        end
    elseif IEDGE==1
        set(handles.nedge,'Visible','on');
        set(handles.text22,'visible','on');
    else
        set(handles.nedge,'Visible','off');
        set(handles.text22,'visible','off');
    end
end
end
function inist_Callback(hObject, eventdata, handles)
Ver function inist_Callback do ficheiro menu_basico_fich_dados_pag9.
function iplod_Callback(hObject, eventdata, handles)
IPLOD=get(handles.iplod,'String');
if isempty(IPLOD)==0
    IPLOD=str2double(get(handles.iplod,'String'));
    if IPLOD<0 || IPLOD>1 || isnan(IPLOD)
        set(handles.iplod,'String','');
        set(handles.numero_cargas_pontuais,'Visible','off');
        set(handles.text21,'visible','off');
    end
end

```



```

    if IPILOD<0 || IPILOD>1
        errordlg('Valor de "IPILOD" só pode ser igual a 0 ou 1','Erro');
    elseif isnan(IPILOD)
        errordlg('Valor de "IPILOD" tem que ser numérico','Erro');
    end
elseif IPILOD==1
    set(handles.numero_cargas_pontuais,'Visible','on');
    set(handles.text21,'visible','on');
else
    set(handles.numero_cargas_pontuais,'Visible','off');
    set(handles.text21,'visible','off');
end
end
function numero_cargas_pontuais_Callback(hObject, eventdata, handles)
Numero_Cargas_Pontuais=get(handles.numero_cargas_pontuais,'String');
if isempty(Numero_Cargas_Pontuais)==0
    load teste NPOIN
    Numero_Cargas_Pontuais=str2double(get(handles.numero_cargas_pontuais,'String'));
    if Numero_Cargas_Pontuais<1 || Numero_Cargas_Pontuais>NPOIN || isnan(Numero_Cargas_Pontuais)
        set(handles.numero_cargas_pontuais,'String','');
        if Numero_Cargas_Pontuais<1
            errordlg('Número de nós tem que ser maior ou igual a 1','Erro');
        elseif Numero_Cargas_Pontuais>NPOIN
            errordlg('Número de nós não pode ser maior que "NPOIN",'Erro');
        elseif isnan(IPILOD)
            errordlg('Valor introduzido tem que ser numérico','Erro');
        end
    end
end
end
function nedge_Callback(hObject, eventdata, handles)
NEDGE=get(handles.nedge,'String');
if isempty(NEDGE)==0
    load teste NELEM
    NEDGE=str2double(get(handles.nedge,'String'));
    if NEDGE<1 || NEDGE>NELEM || isnan(NEDGE)
        set(handles.nedge,'String','');
        if NEDGE<1
            errordlg('Número de lados tem que ser maior ou igual a 1','Erro');
        elseif NEDGE>NELEM
            errordlg('Número de lados não pode ser maior que "NPOIN",'Erro');
        elseif isnan(NEDGE)
            errordlg('Valor introduzido tem que ser numérico','Erro');
        end
    end
end
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag11(varargin)**

```

function lodpt_Callback(hObject, eventdata, handles)
LODPT = get(handles.lodpt,'String');
if isempty(LODPT)==0
    load teste NPOIN
    LODPT = str2double(get(handles.lodpt,'String'));
    if LODPT<1 || LODPT>NPOIN || isnan(LODPT)
        set(handles.lodpt,'String','');
        if LODPT<1
            errordlg('Valor de "LODPT" tem que ser maior ou igual a 1','Erro');
        elseif LODPT>NPOIN
            errordlg('Valor de "LODPT" não pode ser maior que "NPOIN",'Erro');
        elseif isnan(LODPT)
            errordlg('Valor de "LODPT" tem que ser numérico','Erro');
        end
    end
end
end
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag11);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
    teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)

```

```

load teste10 Numero_Cargas_Pontuais
Contagem = str2double(get(handles.contagem,'String'));
if Contagem==Numero_Cargas_Pontuais
    LODPT = str2double(get(handles.lodpt,'String'));
    POINT1 = str2double(get(handles.point1,'String'));
    POINT2 = str2double(get(handles.point2,'String'));
    if isnan(LODPT)==0 && isnan(POINT1)==0 && isnan(POINT2)==0
        if Contagem>1
            load teste11 LODPT POINT1 POINT2
            end
            LODPT(1,Contagem) = str2double(get(handles.lodpt,'String'));
            POINT1(1,Contagem) = str2double(get(handles.point1,'String'));
            POINT2(1,Contagem) = str2double(get(handles.point2,'String'));
            save teste11 LODPT POINT1 POINT2
            resposta=questdlg('Seguinte?',','Sim','Nao','Sim');
            if strcmp(resposta,'Sim')
                delete(menu_fich_dados_pag11);
                load teste10 IGRAV INIST IEDGE
                if IGRAV==1
                    menu_fich_dados_pag12()
                elseif INIST==1
                    menu_fich_dados_pag13()
                elseif IEDGE==1
                    menu_fich_dados_pag15()
                else
                    load teste NPOIN
                    IBODY=0;
                    IPOUT=1;
                    QPONT=NPOIN;
                    CAUDL=0;
                    NGASH1=NPOIN;
                    XGASH=0;
                    YGASH=0;
                    NGASH2=NPOIN;
                    PGASH=0;
                    save teste11
                    resposta=questdlg('Concluir?',','Sim','Nao','Sim');
                    pause(0.1)
                    if strcmp(resposta,'Sim')
                        resposta2=questdlg('Criar ficheiro de resultados?',','Sim','Nao','Sim');
                        pause(0.1)
                        if strcmp(resposta2,'Sim')
                            resposta3=questdlg('Ver resultados?',','Sim','Nao','Sim');
                            pause(0.1)
                        end
                        basico=0;
                        escrita_ficheiro_dados()
                        pause(0.1)
                        if filename_DAT==0
                            menu_fich_dados_pag11()
                        else
                            delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
                            teste12.mat teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
                        end
                    end
                end
            end
        else
            errordlg('Preencha todos os campos','Erro');
        end
    else
        errordlg('Introduza os dados das cargas pontuais em falta','Erro');
    end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag11);
    menu_fich_dados_pag10()
end
function point2_Callback(hObject, eventdata, handles)
POINT2 = get(handles.point2,'String');
if isempty(POINT2)==0
    POINT2 = str2double(get(handles.point2,'String'));
    if isnan(POINT2)
        set(handles.point2,'String','');
        if isnan(POINT2)

```

```

        errordlg('Valor de "POINT2" tem que ser numérico','Erro');
    end
end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem = str2double(get(handles.contagem,'String'));
LODPT = str2double(get(handles.lodpt,'String'));
POINT1 = str2double(get(handles.point1,'String'));
POINT2 = str2double(get(handles.point2,'String'));
if isnan(LODPT)==0 && isnan(POINT1)==0 && isnan(POINT2)==0
    if Contagem>1
        load teste11 LODPT POINT1 POINT2
    end
    LODPT(1,Contagem) = str2double(get(handles.lodpt,'String'));
    POINT1(1,Contagem) = str2double(get(handles.point1,'String'));
    POINT2(1,Contagem) = str2double(get(handles.point2,'String'));
    save teste11 LODPT POINT1 POINT2
    resposta=questdlg('Carga pontual seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.lodpt,'String','');
        set(handles.point1,'String','');
        set(handles.point2,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste10 Numero_Cargas_Pontuais
        if Contagem==Numero_Cargas_Pontuais
            set(handles.pushbutton4,'Visible','off');
        end
    end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function point1_Callback(hObject, eventdata, handles)
POINT1 = get(handles.point1,'String');
if isempty(POINT1)==0
    POINT1 = str2double(get(handles.point1,'String'));
    if isnan(POINT1)
        set(handles.point1,'String','');
        if isnan(POINT1)
            errordlg('Valor de "POINT1" tem que ser numérico','Erro');
        end
    end
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Carga pontual anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste11 LODPT POINT1 POINT2
    set(handles.pushbutton4,'Visible','on');
    set(handles.lodpt,'String',LODPT(1,Contagem));
    set(handles.point1,'String',POINT1(1,Contagem));
    set(handles.point2,'String',POINT2(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste10 Numero_Cargas_Pontuais
if Numero_Cargas_Pontuais==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

**function varargout = menu\_fich\_dados\_pag12(varargin)**

```

function gravity_Callback(hObject, eventdata, handles)
Ver function gravity_Callback do ficheiro menu_basico_fich_dados_pag10.
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Não','Sim');

```

```

if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag12);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
THETA = str2double(get(handles.theta,'String'));
GRAVY1 = str2double(get(handles.gravy,'String'));
if isnan(THETA)==0 && isnan(GRAVY1)==0
    save teste12 THETA GRAVY1
    resposta=questdlg('Seguinte?',','Sim','Nao','Sim');
    if strcmp(resposta,'Sim')
        delete(menu_fich_dados_pag12);
        load teste10 INIST IEDGE
        if INIST==1
            menu_fich_dados_pag13()
        elseif IEDGE==1
            menu_fich_dados_pag15()
        else
            load teste NPOIN
            IBODY=0;
            IPOUT=1;
            QPONT=NPOIN;
            CAUDL=0;
            NGASH1=NPOIN;
            XGASH=0;
            YGASH=0;
            NGASH2=NPOIN;
            PGASH=0;
            save teste12
            resposta=questdlg('Concluir?',','Sim','Nao','Sim');
            pause(0.1)
            if strcmp(resposta,'Sim')
                resposta2=questdlg('Criar ficheiro de resultados?',','Sim','Nao','Sim');
                pause(0.1)
                if strcmp(resposta2,'Sim')
                    resposta3=questdlg('Ver resultados?',','Sim','Nao','Sim');
                    pause(0.1)
                end
                basico=0;
                escrita_ficheiro_dados()
                pause(0.1)
                if filename_DAT==0
                    menu_fich_dados_pag12()
                else
                    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
teste12.mat teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
                end
            end
        end
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag12);
    load teste10 IPLOD
    if IPLOD==1
        menu_fich_dados_pag11()
    else
        menu_fich_dados_pag10()
    end
end
function theta_Callback(hObject, eventdata, handles)
Ver function theta_Callback do ficheiro menu_basico_fich_dados_pag10.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

function varargout = menu_fich_dados_pag13(varargin)

function gravy_Callback(hObject, eventdata, handles)
Ver function gravy_Callback do ficheiro menu_basico_fich_dados_pag11.

```

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim', 'Nao', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag13);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
SURCO = str2double(get(handles.surco, 'String'));
GRAVY2 = str2double(get(handles.gravy, 'String'));
if isnan(SURCO)==0 && isnan(GRAVY2)==0
    save teste13 SURCO GRAVY2
    resposta=questdlg('Seguinte?', 'Sim', 'Nao', 'Sim');
    if strcmp(resposta, 'Sim')
        delete(menu_fich_dados_pag13);
        menu_fich_dados_pag14()
    end
end
else
    errordlg('Preencha todos os campos', 'Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?', 'Sim', 'Nao', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag13);
    load teste10 IPLOD IGRAV
    if IGRAV==1
        menu_fich_dados_pag12()
    elseif IPLOD==1
        menu_fich_dados_pag11()
    else
        menu_fich_dados_pag10()
    end
end
end
function surco_Callback(hObject, eventdata, handles)
Ver function surco_Callback do ficheiro menu_basico_fich_dados_pag11.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

#### **function varargout = menu\_fich\_dados\_pag14(varargin)**

```

function ko_Callback(hObject, eventdata, handles)
Ver function ko_Callback do ficheiro menu_basico_fich_dados_pag12.
function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim', 'Nao', 'Sim');
if strcmp(resposta, 'Sim')
    delete(menu_fich_dados_pag14);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
IK = str2double(get(handles.ik, 'String'));
if IK==0
    KO = str2double(get(handles.ko, 'String'));
    if isnan(KO)==0
        Sair=1;
    else
        Sair=0;
    end
end
else
    Sair=1;
end
if isnan(IK)==0 && Sair==1
    save teste14
    resposta=questdlg('Seguinte?', 'Sim', 'Nao', 'Sim');
    if strcmp(resposta, 'Sim')
        delete(menu_fich_dados_pag14);
        load teste10 IEDGE
        if IEDGE==1
            menu_fich_dados_pag15()
        else
            load teste NPOIN
            IBODY=0;
            IPOUT=1;
        end
    end
end

```

```

QPONT=NPOIN;
CAUDL=0;
NGASH1=NPOIN;
XGASH=0;
YGASH=0;
NGASH2=NPOIN;
PGASH=0;
save teste14
resposta=questdlg('Concluir?',','Sim','Nao','Sim');
pause(0.1)
if strcmp(resposta,'Sim')
    resposta2=questdlg('Criar ficheiro de resultados?',','Sim','Nao','Sim');
    pause(0.1)
    if strcmp(resposta2,'Sim')
        resposta3=questdlg('Ver resultados?',','Sim','Nao','Sim');
        pause(0.1)
    end
    basico=0;
    escrita_ficheiro_dados()
    pause(0.1)
    if filename_DAT==0
        menu_fich_dados_pag14()
    else
        delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
teste12.mat teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    end
end
end
end
else
    errordlg('Preencha todos os campos','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag14);
    menu_fich_dados_pag13()
end
function ik_Callback(hObject, eventdata, handles)
Ver function ik_Callback do ficheiro menu_basico_fich_dados_pag12.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

**function varargout = menu\_fich\_dados\_pag15(varargin)**

```

function pushbutton1_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?',','Sim','Nao','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag15);
    delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat teste12.mat
teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
    menu_principal()
end
function pushbutton2_Callback(hObject, eventdata, handles)
load teste10 NEDGE
Contagem = str2double(get(handles.contagem,'String'));
if NEDGE==Contagem
    NEASS = str2double(get(handles.neass,'String'));
    NOPRS123 = str2num(get(handles.noprs123,'String'));
    PRESS11 = str2double(get(handles.press11,'String'));
    PRESS12 = str2double(get(handles.press12,'String'));
    PRESS21 = str2double(get(handles.press21,'String'));
    PRESS22 = str2double(get(handles.press22,'String'));
    load teste MNODE NPOIN
    if MNODE==8 || MNODE==9
        PRESS31 = str2double(get(handles.press31,'String'));
        if isnan(PRESS31)==0
            Sair=1;
        else
            Sair=0;
        end
        PRESS32 = str2double(get(handles.press32,'String'));
        if Sair==1
            if isnan(PRESS32)
                Sair=0;
            end
        end
    end
end

```

```

        end
    end
    n=3;
else
    n=2;
    Sair=1;
end
if isnan(NEASS)==0 && isempty(NOPRS123)==0 && isnan(PRESS11)==0 && isnan(PRESS12)==0 && isnan(PRESS21)==0 &&
isnan(PRESS22)==0 && Sair==1
    if Contagem>1
        load teste15 NEASS NOPRS123 PRESS11 PRESS12 PRESS21 PRESS22 PRESS31 PRESS32
    end
    NEASS(1,Contagem) = str2double(get(handles.neass,'String'));
    NOPRS123(Contagem,1:n) = str2num(get(handles.noprs123,'String'));
    PRESS11(1,Contagem) = str2double(get(handles.press11,'String'))*-1;
    PRESS12(1,Contagem) = str2double(get(handles.press12,'String'))*-1;
    PRESS21(1,Contagem) = str2double(get(handles.press21,'String'))*-1;
    PRESS22(1,Contagem) = str2double(get(handles.press22,'String'))*-1;
    PRESS31(1,Contagem) = str2double(get(handles.press31,'String'))*-1;
    PRESS32(1,Contagem) = str2double(get(handles.press32,'String'))*-1;
    IBODY=0;
    IPOUT=1;
    QPONT=NPOIN;
    CAUDL=0;
    NGASH1=NPOIN;
    XGASH=0;
    YGASH=0;
    NGASH2=NPOIN;
    PGASH=0;
    save teste15
    resposta=questdlg('Concluir?',','Sim','Não','Sim');
    pause(0.1)
    if strcmp(resposta,'Sim')
        resposta2=questdlg('Criar ficheiro de resultados?',','Sim','Não','Sim');
        pause(0.1)
        if strcmp(resposta2,'Sim')
            resposta3=questdlg('Ver resultados?',','Sim','Não','Sim');
            pause(0.1)
        end
        basico=0;
        escrita_ficheiro_dados()
        pause(0.1)
        if filename_DAT~=0
            delete(menu_fich_dados_pag15);
            delete teste.mat teste2.mat teste3.mat teste4.mat teste5.mat teste6.mat teste7.mat teste8.mat teste9.mat teste10.mat teste11.mat
            teste12.mat teste13.mat teste14.mat teste15.mat nome_ficheiros.txt
        end
    end
    else
        errordlg('Preencha todos os campos','Erro');
    end
else
    errordlg('Introduza os dados das cargas distribuídas em falta','Erro');
end
function pushbutton3_Callback(hObject, eventdata, handles)
resposta=questdlg('Anterior?',','Sim','Não','Sim');
if strcmp(resposta,'Sim')
    delete(menu_fich_dados_pag15);
    load teste10 IPLOD IGRAV INIST
    if INIST==1
        menu_fich_dados_pag14()
    elseif IGRAV==1
        menu_fich_dados_pag12()
    elseif IPLOD==1
        menu_fich_dados_pag11()
    else
        menu_fich_dados_pag10()
    end
end
function noprs123_Callback(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    n=2;
else
    n=3;
end
end

```

```

NOPRS123 = get(handles.noprs123,'String');
if isempty(NOPRS123)==0
    NOPRS123 = str2num(get(handles.noprs123,'String'));
    tamanho_NOPRS123=size(NOPRS123);
    if tamanho_NOPRS123(2)~=n || isempty(NOPRS123)
        set(handles.noprs123,'String','');
        if n==3
            if isempty(NOPRS123)
                errordlg('Valor de [nó(1) nó(2) nó(3)] tem que ser numérico','Erro');
            elseif tamanho_NOPRS123(2)~=n
                errordlg('O número de nós tem que ser igual a 3','Erro');
            end
        elseif n==2
            if isempty(NOPRS123)
                errordlg('Valor de [nó(1) nó(2)] tem que ser numérico','Erro');
            elseif tamanho_NOPRS123(2)~=n
                errordlg('O número de nós tem que ser igual a 2','Erro');
            end
        end
    end
end
function press11_Callback(hObject, eventdata, handles)
Ver function press11_Callback do ficheiro menu_basico_fich_dados_pag13.
function press32_Callback(hObject, eventdata, handles)
PRESS32 = get(handles.press32,'String');
if isempty(PRESS32)==0
    PRESS32 = str2double(get(handles.press32,'String'));
    if isnan(PRESS32)
        set(handles.press32,'String','');
        if isnan(PRESS32)
            errordlg('Valor de "PRESS(3,2)" tem que ser numérico','Erro');
        end
    end
end
function press32_CreateFcn(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function neass_Callback(hObject, eventdata, handles)
NEASS = get(handles.neass,'String');
if isempty(NEASS)==0
    load teste NELEM
    NEASS = str2double(get(handles.neass,'String'));
    if NEASS<1 || NEASS>NELEM || isnan(NEASS)
        set(handles.neass,'String','');
        if NEASS<1
            errordlg('Valor de "NEASS" tem que ser maior ou igual a 1','Erro');
        elseif NEASS>NELEM
            errordlg('Valor de "NEASS" não pode ser maior que "NELEM"', 'Erro');
        elseif isnan(NEASS)
            errordlg('Valor de "NEASS" tem que ser numérico','Erro');
        end
    end
end
function pushbutton4_Callback(hObject, eventdata, handles)
Contagem = str2double(get(handles.contagem,'String'));
NEASS = str2double(get(handles.neass,'String'));
NOPRS123 = str2num(get(handles.noprs123,'String'));
PRESS11 = str2double(get(handles.press11,'String'));
PRESS12 = str2double(get(handles.press12,'String'));
PRESS21 = str2double(get(handles.press21,'String'));
PRESS22 = str2double(get(handles.press22,'String'));
load teste MNODE
if MNODE==8 || MNODE==9
    PRESS31 = str2double(get(handles.press31,'String'));
    if isnan(PRESS31)==0
        Seguinte=1;
    else
        Seguinte=0;
    end
    PRESS32 = str2double(get(handles.press32,'String'));
    if Seguinte==1
        if isnan(PRESS32)

```



```

        Seguinte=0;
    end
end
n=3;
else
    n=2;
    Seguinte=1;
end
if isnan(NEASS)==0 && isempty(NOPRS123)==0 && isnan(PRESS11)==0 && isnan(PRESS12)==0 && isnan(PRESS21)==0 &&
isnan(PRESS22)==0 && Seguinte==1
    if Contagem>1
        load teste15 NEASS NOPRS123 PRESS11 PRESS12 PRESS21 PRESS22 PRESS31 PRESS32
    end
    NEASS(1,Contagem) = str2double(get(handles.neass,'String'));
    NOPRS123(Contagem,1:n) = str2num(get(handles.noprs123,'String'));
    PRESS11(1,Contagem) = str2double(get(handles.press11,'String'))*-1;
    PRESS12(1,Contagem) = str2double(get(handles.press12,'String'))*-1;
    PRESS21(1,Contagem) = str2double(get(handles.press21,'String'))*-1;
    PRESS22(1,Contagem) = str2double(get(handles.press22,'String'))*-1;
    PRESS31(1,Contagem) = str2double(get(handles.press31,'String'))*-1;
    PRESS32(1,Contagem) = str2double(get(handles.press32,'String'))*-1;
    save teste15
    resposta=questdlg('Carga distribuída seguinte?',','Sim','Não','Sim');
    if strcmp(resposta,'Sim')
        Contagem=Contagem+1;
        set(handles.contagem,'String',Contagem);
        set(handles.neass,'String','');
        set(handles.noprs123,'String','');
        set(handles.press11,'String','');
        set(handles.press12,'String','');
        set(handles.press21,'String','');
        set(handles.press22,'String','');
        set(handles.press31,'String','');
        set(handles.press32,'String','');
        set(handles.pushbutton5,'Visible','on');
        load teste10 NEDGE
        if Contagem==NEDGE
            set(handles.pushbutton4,'Visible','off');
        end
    end
else
    errordlg('Preencha todos os campos','Erro');
end
function press12_Callback(hObject, eventdata, handles)
Ver function press12_Callback do ficheiro menu_basico_fich_dados_pag13.
function press21_Callback(hObject, eventdata, handles)
PRESS21 = get(handles.press21,'String');
if isempty(PRESS21)==0
    PRESS21 = str2double(get(handles.press21,'String'));
    if isnan(PRESS21)
        set(handles.press21,'String','');
        if isnan(PRESS21)
            errordlg('Valor de "PRESS(2,1)" tem que ser numérico','Erro');
        end
    end
end
function press22_Callback(hObject, eventdata, handles)
PRESS22 = get(handles.press22,'String');
if isempty(PRESS22)==0
    PRESS22 = str2double(get(handles.press22,'String'));
    if isnan(PRESS22)
        set(handles.press22,'String','');
        if isnan(PRESS22)
            errordlg('Valor de "PRESS(2,2)" tem que ser numérico','Erro');
        end
    end
end
function press31_Callback(hObject, eventdata, handles)
PRESS31 = get(handles.press31,'String');
if isempty(PRESS31)==0
    PRESS31 = str2double(get(handles.press31,'String'));
    if isnan(PRESS31)
        set(handles.press31,'String','');
        if isnan(PRESS31)
            errordlg('Valor de "PRESS(3,1)" tem que ser numérico','Erro');
        end
    end
end

```

```

end
end
function pushbutton5_Callback(hObject, eventdata, handles)
resposta=questdlg('Carga distribuída anterior?', 'Sim','Não','Sim');
if strcmp(resposta,'Sim')
    Contagem=str2double(get(handles.contagem,'String'));
    Contagem=Contagem-1;
    set(handles.contagem,'String',Contagem);
    if Contagem==1
        set(handles.pushbutton5,'Visible','off');
    end
    load teste MNODE
    if MNODE==4
        n=2;
    else
        n=3;
    end
    load teste15 NEASS NOPRS123 PRESS11 PRESS12 PRESS21 PRESS22 PRESS31 PRESS32
    set(handles.pushbutton4,'Visible','on');
    set(handles.neass,'String',NEASS(1,Contagem));
    set(handles.noprs123,'String',num2str(NOPRS123(Contagem,1:n)));
    set(handles.press11,'String',PRESS11(1,Contagem));
    set(handles.press12,'String',PRESS12(1,Contagem));
    set(handles.press21,'String',PRESS21(1,Contagem));
    set(handles.press22,'String',PRESS22(1,Contagem));
    set(handles.press31,'String',PRESS31(1,Contagem));
    set(handles.press32,'String',PRESS32(1,Contagem));
end
function pushbutton4_CreateFcn(hObject, eventdata, handles)
load teste10 NEDGE
if NEDGE==1
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function text19_CreateFcn(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function press31_CreateFcn(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function text22_CreateFcn(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    set(hObject,'visible','off')
else
    set(hObject,'visible','on')
end
function text16_CreateFcn(hObject, eventdata, handles)
load teste MNODE
if MNODE==4
    set(hObject,'string',{'Lista dos números, em sentido direto, dos nós';'que formam o lado carregado [nó(1) nó(2):]'})
else
    set(hObject,'string',{'Lista dos números, em sentido direto, dos nós';'que formam o lado carregado [nó(1) nó(2) nó(3):]'})
end
function figure1_CloseRequestFcn(hObject, eventdata, handles)
Ver function figure1_CloseRequestFcn do ficheiro menu_fich_dados_pag1.

```

### **function escrita ficheiro dados**

```

existe_filename_DAT=1;
while existe_filename_DAT>0
    filename_DAT = uiputfile({'*.DAT'},'Guardar como (Ficheiro de dados):');
    pause(0.1)
    if filename_DAT~=0
        existe_filename_DAT = fopen(filename_DAT,'rt');
    else

```

```

    existe_filename_DAT=-1;
end
end
if filename_DAT~=0
if basico==1
load teste
load teste2
load teste3
load teste4
load teste5
load teste6
load teste7
load teste8
load teste9
if IGRAV==1
load teste10
end
if INIST==1
load teste11
load teste12
end
load teste13
else
load teste
load teste2
if TELEM==2
load teste3
end
load teste4
load teste5
load teste6
load teste7
load teste8
load teste9
load teste10
if IPLOD==1
load teste11
end
if IGRAV==1
load teste12
end
if INIST==1
load teste13
load teste14
end
if IEDGE==1
load teste15
end
end
fid=fopen(filename_DAT,'w');
fprintf(fid,'%s \n',titulo_problema);
fprintf(fid,'%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g \n',[NPOIN NELEM NVFIX NPRFX NTYPE MNODE NDOFN
NMATS NPROP NGAUS NDIME NSTRE TELEM NINCR]);
if TELEM==2
fprintf(fid,'%g \n',NPPRE);
end
fprintf(fid,'%g \n',LINCA);
if MNODE==9
for i=1:NELEM
fprintf(fid,'%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g \n',[NUMEL(1,i) MATNO(1,i) LNODS(i,1:MNODE) NNODE(1,i)]);
end
elseif MNODE==8
for i=1:NELEM
fprintf(fid,'%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g \n',[NUMEL(1,i) MATNO(1,i) LNODS(i,1:MNODE) NNODE(1,i)]);
end
elseif MNODE==4
for i=1:NELEM
fprintf(fid,'%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g \n',[NUMEL(1,i) MATNO(1,i) LNODS(i,1:MNODE) NNODE(1,i)]);
end
end
if TELEM==2
if MNODE==8
for i=1:NELEM
fprintf(fid,'%g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g %g \n',[NUMEL2(1,i) LPNOD(i,1:MNODE)]);
end
end
end

```

```

if MNODE==4
  for i=1:NELEM
    fprintf(fid,'%g %g %g %g %g \n',[NUMEL2(1,i) LPNOD(i,1:MNODE)]);
  end
end
end
tamanho_IPOIN=size(IPOIN);
for i=1:tamanho_IPOIN(1)
  fprintf(fid,'%g %g %g \n',[IPOIN(i,1) COORD1(i,1) COORD2(i,1)]);
end
for i=1:NVFIX
  if TELEM==2
    fprintf(fid,'%g %g %g %g %g %g \n',[NOFIX(1,i) KELEM(1,i) IFPRE1(1,i) IFPRE2(1,i) PRESC1(1,i) PRESC2(1,i)]);
  else
    fprintf(fid,'%g %g %g %g %g \n',[NOFIX(1,i) IFPRE1(1,i) IFPRE2(1,i) PRESC1(1,i) PRESC2(1,i)]);
  end
end
end
for i=1:NPRFX
  fprintf(fid,'%g %g \n',[NOPRF(1,i) PREPR(1,i)]);
end
fprintf(fid,'%g %g \n',[ESSE AALFA]);
fprintf(fid,'%g \n',GAMAW);
fprintf(fid,'%E %E %E \n',PERMB);
for i=1:NMATS
  fprintf(fid,'%g %g %g %g %g \n',[NUMAT(1,i) PROPS1(1,i) PROPS2(1,i) PROPS3(1,i) PROPS4(1,i) PROPS5(1,i)]);
end
end
for i=1:NINCR
  fprintf(fid,'%G %g \n',[INCRT(1,i) ISOUT(1,i)]);
end
end
fprintf(fid,'%s \n',TITLE);
fprintf(fid,'%g %g %g %g \n',[IPLD IGRAV IEDGE INIST]);
if IPLD==1
  for i=1:Numero_Cargas_Pontuais
    fprintf(fid,'%g %g %g \n',[LODPT(1,i) POINT1(1,i) POINT2(1,i)]);
    if i==Numero_Cargas_Pontuais
      fprintf(fid,'%g %g %g \n',[NPOIN 0 0]);
    end
  end
end
end
if IGRAV==1
  fprintf(fid,'%g %g \n',[THETA GRAVY1]);
end
end
if INIST==1
  fprintf(fid,'%g %g \n',[SURCO GRAVY2]);
  fprintf(fid,'%g \n',IK);
  if IK==0
    fprintf(fid,'%g \n',KO);
  end
end
end
if IEDGE==1
  fprintf(fid,'%g \n',NEDGE);
  if MNODE==4
    for i=1:NEDGE
      fprintf(fid,'%g %g %g \n',[NEASS(1,i) NOPRS123(i,1:2)]);
      fprintf(fid,'%g %g %g %g \n',[PRESS11(1,i) PRESS21(1,i) PRESS12(1,i) PRESS22(1,i)]);
    end
  else
    for i=1:NEDGE
      fprintf(fid,'%g %g %g %g \n',[NEASS(1,i) NOPRS123(i,1:3)]);
      fprintf(fid,'%g %g %g %g %g \n',[PRESS11(1,i) PRESS21(1,i) PRESS31(1,i) PRESS12(1,i) PRESS22(1,i) PRESS32(1,i)]);
    end
  end
end
end
fprintf(fid,'%g %g \n',[IBODY IPOUT]);
if IPOUT==1
  fprintf(fid,'%g %g \n',[QPONT CAUDL]);
end
end
if TELEM==2
  fprintf(fid,'%g %g %g %g \n',[NGASH1 IELEM XGASH YGASH]);
  fprintf(fid,'%g %g \n',[NGASH2 PGASH]);
else
  fprintf(fid,'%g %g %g \n',[NGASH1 XGASH YGASH]);
  fprintf(fid,'%g %g \n',[NGASH2 PGASH]);
end
end
fclose(fid);
if strcmp(resposta2,'Sim')

```

```

existe_filename_RES=1;
while existe_filename_RES>0
    filename_RES = uiputfile({'*.RES'},'Guardar como (Ficheiro de resultados):');
    pause(0.1)
    if filename_RES~=0
        existe_filename_RES = fopen(filename_RES,'rt');
        if existe_filename_RES<0
            fid2=fopen('nome_ficheiros.txt','w');
            fprintf(fid2,'%s \n',filename_DAT);
            fprintf(fid2,'%s \n',filename_RES);
            fclose(fid2);
            pause(0.1)
            system('CONSOL.exe')
            pause(0.1)
            if strcmp(resposta3,'Sim')
                ler_ficheiro_abrir_menu()
            else
                menu_principal()
            end
        end
    else
        existe_filename_RES=-1;
    end
end
else
    menu_principal()
end
end
end

```

## A1.2. Interface de saída de resultados

### function ler\_ficheiro\_abrir\_menu

```

if exist('filename_RES','var')==0
    filename_RES = uigetfile({'*.RES'},'Escolha o ficheiro de resultados');
    pause(0.1)
end
if filename_RES~=0
    fid = fopen(filename_RES,'rt');
    fseek(fid,20,0);
    titulo_problema=fgets(fid,14);
    linha1=fgetl(fid);
    dados=fscanf(fid,'%s %c %f %*s %*c %f %*s %*c %f %*s %*c %f %*s %*c %f \n',[5, inf]);
    dados=dados';
    NPOIN=dados(1,1);
    NELEM=dados(1,2);
    NVFIX=dados(1,3);
    NPRFX=dados(1,4);
    NTYPE=dados(1,5);
    MNODE=dados(2,1);
    if MNODE==4
        NODEG=2;
    else
        NODEG=3;
    end
    NDOFN=dados(2,2);
    NMATS=dados(2,3);
    NPROP=dados(2,4);
    NGAUS=dados(2,5);
    NDIME=dados(3,1);
    NSTRE=dados(3,2);
    MEVAB=dados(3,3);
    TELEM=dados(3,4);
    NINCR=dados(3,5);
    if TELEM==2
        NPPRE=dados(4,1);
        LINCA=dados(4,2);
    else
        LINCA=dados(4,1);
    end
    linha2=fgetl(fid);
    elementos=fscanf(fid,'%f %f %f %f %f %f %f %f %f %f \n',[3+MNODE, NELEM]);
    elementos=elementos';
end

```

```

linha3=fgetl(fid);
linha4=fgetl(fid);
linha5=fgetl(fid);
linha6=fgetl(fid);
linha7=fgetl(fid);
coordenadas=fscanf(fid,'%f %f %f %f\n',[1+NDIME, NPOIN]);
coordenadas=coordenadas';
linha8=fgetl(fid);
linha9=fgetl(fid);
linha10=fgetl(fid);
linha11=fgetl(fid);
linha12=fgetl(fid);
if TELEM==2
else
    nos_restringidos1=fscanf(fid,'%f %f %f %f\n',[2+NDOFN, NVFIX]);
    nos_restringidos1=nos_restringidos1';
end
nos_restringidos2=fscanf(fid,'%f %f\n',[2, NPRFX]);
nos_restringidos2=nos_restringidos2';
linha13=fgetl(fid);
linha14=fgetl(fid);
fator_majoraçao_S=fscanf(fid,'%s %s %s %s %s %c%f',[1, 1]);
alfa=fscanf(fid,'%s %c%f',[1, 1]);
gama_agua=fscanf(fid,'%s %s %s %s %s %c%f',[1, 1]);
linha15=fgetl(fid);
linha16=fgetl(fid);
linha17=fgetl(fid);
linha18=fgetl(fid);
matriz_permeabilidade=fscanf(fid,'%f %f %f %f',[2, 2]);
matriz_permeabilidade=matriz_permeabilidade';
linha19=fgetl(fid);
linha20=fgetl(fid);
linha21=fgetl(fid);
linha22=fgetl(fid);
linha23=fgetl(fid);
propriedades_materiais=fscanf(fid,'%f %f %f %f %f %f\n',[6, NMATS]);
propriedades_materiais=propriedades_materiais';
linha24=fgetl(fid);
linha25=fgetl(fid);
linha26=fgetl(fid);
linha27=fgetl(fid);
linha28=fgetl(fid);
incr_tempo_tensoes=fscanf(fid,'%f %f\n',[2, NINCR]);
incr_tempo_tensoes=incr_tempo_tensoes';
linha29=fgetl(fid);
linha30=fgetl(fid);
linha31=fgetl(fid);
linha32=fgetl(fid);
linha33=fgetl(fid);
linha34=fgetl(fid);
linha35=fgetl(fid);
linha36=fgetl(fid);
linha37=fgetl(fid);
linha38=fgetl(fid);
linha39=fgetl(fid);
titulo_carga=fgetl(fid);
tipo_carga=fscanf(fid,'%f %f %f %f',[4, 1]);
tipo_carga=tipo_carga';
IPLOD=tipo_carga(1,1);
IGRAV=tipo_carga(1,2);
IEDGE=tipo_carga(1,3);
INIST=tipo_carga(1,4);
if IPLOD==1
    cargas_pontuais=fscanf(fid,'%f %f %f\n',[3, inf]);
    cargas_pontuais=cargas_pontuais';
end
if IGRAV==1
    gravidade=fscanf(fid,'%s %s %s %s %c %f %s %s %c %f',[2, 1]);
    gravidade=gravidade';
end
if INIST==1
    tensao_inicial=fscanf(fid,'%s %f %s %c %c %c %c %c %c %c %c %f',[2, 1]);
    tensao_inicial=tensao_inicial';
end
if IEDGE==1
    linha41=fgetl(fid);

```



```

reacoes(1:contagem_num_reacoes,3*t-2:3*t)=reac;
linha65=fgetl(fid);
linha66=fgetl(fid);
linha67=fgetl(fid);
if LINCA==1 && TELEM==1
    if t==2
        caud=fscanf(fid,'%f %f \n',[2, NPOIN]);
        caud=caud';
        caudais(1:NPOIN,2*t-1:2*t)=caud;
    elseif t>2
        tamanho_nos_restringidos2=size(nos_restringidos2);
        caud=fscanf(fid,'%f %f \n',[2, tamanho_nos_restringidos2(1)]);
        caud=caud';
        caudais(1:tamanho_nos_restringidos2(1),2*t-1:2*t)=caud;
    end
elseif LINCA==1 && TELEM~=1
    if t>1
        tamanho_nos_restringidos2=size(nos_restringidos2);
        caud=fscanf(fid,'%f %f \n',[2, tamanho_nos_restringidos2(1)]);
        caud=caud';
        caudais(1:tamanho_nos_restringidos2(1),2*t-1:2*t)=caud;
    end
elseif TELEM==1 && t==2 && LINCA~=1
    caud=fscanf(fid,'%f %f \n',[2, NPOIN]);
    caud=caud';
    caudais(1:NPOIN,2*t-1:2*t)=caud;
else
    tamanho_nos_restringidos2=size(nos_restringidos2);
    caud=fscanf(fid,'%f %f \n',[2, tamanho_nos_restringidos2(1)]);
    caud=caud';
    caudais(1:tamanho_nos_restringidos2(1),2*t-1:2*t)=caud;
end
linha68=fgetl(fid);
calculo_tensoes=incr_tempo_tensoes(1:NINCR,2);
if max(calculo_tensoes)==1 && incr_tempo_tensoes(t,2)==1
    if NTYPE==3
        else
            linha69=fgetl(fid);
            linha70=fgetl(fid);
            linha71=fgetl(fid);
            linha72=fgetl(fid);
        end
    for n=1:NELEM
        if incr_tempo_tensoes(t,2)==1
            linha73=fgetl(fid);
            linha74=fgetl(fid);
            if LINCA==0 || LINCA==1 && n>1 || LINCA==1 && t>1
                linha75=fgetl(fid);
            end
            if NGAUS==2
                aux_NGAUS=4;
            elseif NGAUS==3
                aux_NGAUS=9;
            end
            if NTYPE==3
                else
                    for i=1:aux_NGAUS
                        ponto1=fscanf(fid,'%f %f %f %f %f %f %f %f',[4+NDIME, 1]);
                        ponto1=ponto1';
                        ponto1(i,1:4+NDIME)=ponto1;
                        ponto2=fscanf(fid,'%f %f %f',[3, 1]);
                        ponto2=ponto2';
                        ponto2(i,1:3)=ponto2;
                    end
                end
                tensoes1(aux_NGAUS*n-(aux_NGAUS-1):aux_NGAUS*n,(4+NDIME)*t-(4+NDIME-1):(4+NDIME)*t)=ponto1;
                tensoes2(aux_NGAUS*n-(aux_NGAUS-1):aux_NGAUS*n,3*t-2:3*t)=ponto2;
            end
        end
    end
end
end
end
end
fclose(fid);
if IEDGE==1
    for i=1:NEDGE
        n=0;

```



```

tamanho_local_carga=size(local_carga);
for j=1:tamanho_local_carga(2)-1
    if valor_carga(2*i-1,j)~=0 || valor_carga(2*i,j)~=0
        n=n+1;
        m(i,1)=n;
        aux3=local_carga(i,j+1);
        Xcoord_c=coordenadas(aux3,2);
        Ycoord_c=coordenadas(aux3,3);
        Xcoord_carga(i,m(i,1))=Xcoord_c;
        Ycoord_carga(i,m(i,1))=Ycoord_c;
    end
end
Xmax_carga(i,1)=max(Xcoord_carga(i,1:m(i,1)));
Xmin_carga(i,1)=min(Xcoord_carga(i,1:m(i,1)));
Ymax_carga(i,1)=max(Ycoord_carga(i,1:m(i,1)));
Ymin_carga(i,1)=min(Ycoord_carga(i,1:m(i,1)));
end
a=max(Xmax_carga)-min(Xmin_carga);
G=propriedades_materiais(1,2)/(2*(1+propriedades_materiais(1,3)));
tempo_mod=(2*G*matriz_permeabilidade(1,1)*tempo)/(gama_agua*a^2);
for i=1:NEDGE*2
    max_valor_c(1,i)=max(abs(valor_carga(i,1:NODEG)));
end
max_valor_carga=max(max_valor_c);
end
save dados
pause(0.1)
menu_resultados()
delete(menu_principal)
end

```

#### **function varargout = menu\_resultados(varargin)**

```

function popuptenu1_Callback(hObject, eventdata, handles)
load busy busy
if busy==0
    set(handles.pushbutton9,'visible','off');
end
load dados tensoes1 IEDGE titulo_problema
Popuptenu1=get(handles.popuptenu1,'value');
if Popuptenu1==1
    set(handles.popuptenu3,'visible','off');
    set(handles.popuptenu4,'visible','off');
    set(handles.uipanel5,'visible','off');
    set(handles.popuptenu2,'visible','off');
    set(handles.text1,'visible','off');
    set(handles.edit1,'visible','off');
    set(handles.text2,'visible','off');
    set(handles.edit2,'visible','off');
    set(handles.pushbutton1,'visible','off');
    set(handles.uipanel2,'Position',[0.8513 0.8176 0.1170 0.1022]);
    set(handles.popuptenu1,'Position',[0.0680 0.2653 0.7687 0.4489]);
    set(handles.popuptenu3,'value',1);
    set(handles.popuptenu4,'value',1);
    set(handles.popuptenu2,'value',1);
    set(handles.pushbutton2,'Position',[0.8513 0.7563 0.0534 0.0362]);
    set(handles.pushbutton3,'Position',[0.9148 0.7563 0.0534 0.0362]);
    set(handles.popuptenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';Tensão horizontal efetiva e total e pressão neutra'});
elseif Popuptenu1==2 || Popuptenu1==3
    set(handles.popuptenu4,'visible','off');
    set(handles.text2,'visible','off');
    set(handles.edit2,'visible','off');
    set(handles.uipanel2,'Position',[0.8513 0.7610 0.1170 0.1588]);
    set(handles.popuptenu1,'Position',[0.0680 0.5764 0.7687 0.2588]);
    set(handles.popuptenu3,'Position',[0.0680 0.1647 0.7687 0.2588]);
    set(handles.popuptenu3,'value',1);
    if exist('tensoes1','var')==1
        set(handles.popuptenu3,'string',{'-----';'Vetores do deslocamento';'Vetores da tensão efetiva';'Isolinhas do deslocamento';'Isolinhas da tensão efetiva'});
    else
        set(handles.popuptenu3,'string',{'-----';'Vetores do deslocamento';'Isolinhas do deslocamento'});
    end
end
set(handles.popuptenu4,'value',1);

```

```

set(handles.popupmenu4,'string',{'Vertical (nos nós pertencentes à linha horizontal)';'Horizontal (nos nós pertencentes à linha horizontal)';'Total (nos nós pertencentes à linha horizontal)';'Vertical (em todos os nós)';'Horizontal (em todos os nós)';'Total (em todos os nós)'});
set(handles.uipanel5,'Position',[0.8513 0.6368 0.1170 0.1022]);
set(handles.popupmenu2,'Position',[0.0680 0.2653 0.7687 0.4489]);
set(handles.popupmenu2,'value',1);
if exist('tensoes1','var')==1
    set(handles.text1,'Position',[0.8513 0.5818 0.0627 0.0236]);
    set(handles.edit1,'Position',[0.9171 0.5818 0.0201 0.0299]);
    set(handles.pushbutton1,'Position',[0.8761 0.5252 0.0534 0.0362]);
    set(handles.pushbutton2,'Position',[0.8513 0.5252 0.0534 0.0362]);
    set(handles.pushbutton3,'Position',[0.9148 0.5252 0.0534 0.0362]);
else
    set(handles.text1,'Position',[0.8513 0.7186 0.0627 0.0236]);
    set(handles.edit1,'Position',[0.9171 0.7186 0.0201 0.0299]);
    set(handles.pushbutton1,'Position',[0.8761 0.6651 0.0534 0.0362]);
    set(handles.pushbutton2,'Position',[0.8513 0.6651 0.0534 0.0362]);
    set(handles.pushbutton3,'Position',[0.9148 0.6651 0.0534 0.0362]);
end
set(handles.popupmenu3,'visible','on');
if exist('tensoes1','var')==1
    set(handles.uipanel5,'visible','on');
    set(handles.popupmenu2,'visible','on');
end
set(handles.text1,'visible','on');
set(handles.edit1,'visible','on');
set(handles.pushbutton1,'visible','on');
elseif Popumenu1==4
    set(handles.popupmenu4,'visible','off');
    set(handles.uipanel5,'visible','off');
    set(handles.popupmenu2,'visible','off');
    set(handles.text1,'visible','off');
    set(handles.edit1,'visible','off');
    set(handles.text2,'visible','off');
    set(handles.edit2,'visible','off');
    set(handles.uipanel2,'Position',[0.8513 0.7610 0.1170 0.1588]);
    set(handles.popupmenu1,'Position',[0.0680 0.5764 0.7687 0.2588]);
    set(handles.popupmenu3,'Position',[0.0680 0.1647 0.7687 0.2588]);
    set(handles.popupmenu3,'value',1);
    if exist('tensoes1','var')==1
        set(handles.popupmenu3,'string',{'-----';'Vetores do deslocamento';'Vetores da tensão efetiva'});
    else
        set(handles.popupmenu3,'string',{'-----';'Vetores do deslocamento'});
    end
    set(handles.popupmenu4,'value',1);
    set(handles.popupmenu4,'string',{'Vertical';'Horizontal';'Total'});
    set(handles.popupmenu2,'value',1);
    set(handles.pushbutton1,'Position',[0.8761 0.7060 0.0534 0.0362]);
    set(handles.pushbutton2,'Position',[0.8513 0.7060 0.0534 0.0362]);
    set(handles.pushbutton3,'Position',[0.9148 0.7060 0.0534 0.0362]);
    set(handles.popupmenu3,'visible','on');
    set(handles.pushbutton1,'visible','on');
elseif Popumenu1>4
    set(handles.popupmenu3,'visible','off');
    set(handles.popupmenu4,'visible','off');
    set(handles.uipanel5,'visible','off');
    set(handles.popupmenu2,'visible','off');
    set(handles.uipanel2,'Position',[0.8513 0.8176 0.1170 0.1022]);
    set(handles.popupmenu1,'Position',[0.0680 0.2653 0.7687 0.4489]);
    set(handles.popupmenu3,'value',1);
    set(handles.popupmenu4,'value',1);
    set(handles.popupmenu2,'value',1);
    set(handles.text1,'Position',[0.8513 0.7767 0.0627 0.0236]);
if Popumenu1==6 || Popumenu1==8
    set(handles.edit1,'Position',[0.9473 0.7767 0.0201 0.0299]);
    set(handles.text2,'Position',[0.8513 0.7203 0.0976 0.0472]);
    set(handles.edit2,'Position',[0.9473 0.7264 0.0201 0.0299]);
    set(handles.pushbutton1,'Position',[0.8761 0.6682 0.0534 0.0362]);
    set(handles.pushbutton2,'Position',[0.8513 0.6682 0.0534 0.0362]);
    set(handles.pushbutton3,'Position',[0.9148 0.6682 0.0534 0.0362]);
    set(handles.edit1,'visible','on');
    set(handles.text2,'visible','on');
    set(handles.edit2,'visible','on');
    set(handles.pushbutton1,'visible','on');
else
    set(handles.text2,'visible','off');

```

```

set(handles.edit2,'visible','off');
set(handles.edit1,'Position',[0.9171 0.7767 0.0201 0.0299]);
set(handles.pushbutton1,'Position',[0.8761 0.7264 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.7264 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.7264 0.0534 0.0362]);
set(handles.edit1,'visible','on');
set(handles.pushbutton1,'visible','on');
end
set(handles.text1,'visible','on');
end
if Popupmenu1==2 || Popupmenu1==3 || Popupmenu1==5 || Popupmenu1==7
if Popupmenu1==7
if IEDGE==1
if strcmp(titulo_problema,'unidimensional')
set(handles.uipanel8,'position',[0.8513 0.380 0.117 0.0794]);
elseif strcmp(titulo_problema,'bidimensional ')
set(handles.uipanel8,'position',[0.8513 0.460 0.117 0.0794]);
end
set(handles.uipanel8,'visible','on');
set(handles.radiobutton12,'position',[0.129 0.550 0.551 0.350]);
set(handles.radiobutton13,'position',[0.129 0.100 0.551 0.350]);
else
set(handles.uipanel8,'visible','off');
set(handles.radiobutton12,'value',1);
end
else
set(handles.uipanel8,'visible','off');
end
set(handles.uipanel6,'visible','on');
set(handles.radiobutton4,'visible','on');
set(handles.radiobutton4,'string','Escala linear');
set(handles.radiobutton5,'visible','on');
set(handles.radiobutton5,'string','Escala logarítmica');
if IEDGE==1
if strcmp(titulo_problema,'unidimensional')
set(handles.uipanel6,'position',[0.8513 0.2893 0.117 0.0794]);
set(handles.radiobutton2,'visible','off');
set(handles.radiobutton3,'visible','off');
set(handles.radiobutton4,'position',[0.129 0.550 0.551 0.350]);
set(handles.radiobutton5,'position',[0.129 0.100 0.800 0.350]);
elseif strcmp(titulo_problema,'bidimensional ')
set(handles.uipanel6,'position',[0.8513 0.2893 0.117 0.1588]);
set(handles.radiobutton2,'visible','on');
set(handles.radiobutton3,'visible','on');
end
set(handles.radiobutton2,'position',[0.129 0.765 0.551 0.176]);
set(handles.radiobutton2,'string','Tempo');
set(handles.radiobutton3,'position',[0.129 0.565 0.551 0.176]);
set(handles.radiobutton3,'string','Fator T');
else
set(handles.uipanel6,'position',[0.8513 0.2893 0.117 0.0794]);
set(handles.radiobutton2,'visible','off');
set(handles.radiobutton2,'value',1);
set(handles.radiobutton3,'visible','off');
set(handles.radiobutton4,'position',[0.129 0.550 0.551 0.350]);
set(handles.radiobutton5,'position',[0.129 0.100 0.800 0.350]);
end
if exist('tensoes1','var')==0
if Popupmenu1==2 || Popupmenu1==3
set(handles.uipanel6,'visible','off');
set(handles.uipanel8,'visible','off');
end
end
elseif Popupmenu1==8
if IEDGE==1
if strcmp(titulo_problema,'unidimensional')
set(handles.uipanel6,'visible','off');
set(handles.uipanel8,'position',[0.8513 0.2893 0.117 0.0794]);
elseif strcmp(titulo_problema,'bidimensional ')
set(handles.uipanel6,'visible','on');
set(handles.uipanel8,'position',[0.8513 0.380 0.117 0.0794]);
end
set(handles.uipanel6,'position',[0.8513 0.2893 0.117 0.0794]);
set(handles.radiobutton2,'position',[0.129 0.550 0.551 0.350]);
set(handles.radiobutton2,'visible','on');
set(handles.radiobutton2,'string','x');

```

```

set(handles.radiobutton3,'position',[0.129 0.100 0.551 0.350]);
set(handles.radiobutton3,'visible','on');
set(handles.radiobutton3,'string','x/a');
set(handles.radiobutton4,'visible','off');
set(handles.radiobutton5,'visible','off');
set(handles.uipanel8,'visible','on');
set(handles.radiobutton12,'position',[0.129 0.550 0.551 0.350]);
set(handles.radiobutton13,'position',[0.129 0.100 0.551 0.350]);
else
set(handles.uipanel6,'visible','off');
set(handles.radiobutton2,'value',1);
set(handles.uipanel8,'visible','off');
set(handles.radiobutton12,'value',1);
end
else
set(handles.uipanel6,'visible','off');
set(handles.uipanel8,'visible','off');
end
if Popumenu1>1
load busy busy
if busy==1
set(handles.pushbutton1,'visible','off');
else
set(handles.pushbutton1,'visible','on');
end
end
set(handles.popupmenu2,'value',1);
set(handles.popupmenu3,'value',1);
set(handles.popupmenu4,'value',1);
set(handles.popupmenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';Tensão horizontal efetiva e total e pressão neutra'});
function popupmenu1_CreateFcn(hObject, eventdata, handles)
load dados titulo_problema
if strcmp(titulo_problema,'unidimensional')
set(hObject,'string',{'-----';'Deslocamento vertical e altura da coluna de água';'Deslocamento total e altura da coluna de
água';'Malha (deslocamento total)';'Deslocamento vertical vs tempo';'Deslocamento vertical vs profundidade';'Pressão neutra vs
tempo';'Pressão neutra vs profundidade'});
elseif strcmp(titulo_problema,'bidimensional ')
set(hObject,'string',{'-----';'Deslocamento vertical e altura da coluna de água';'Deslocamento total e altura da coluna de
água';'Malha (deslocamento total)';'Deslocamento vertical vs tempo';'Deslocamento vertical vs distância ao eixo y';'Pressão neutra vs
tempo';'Pressão neutra vs distância ao eixo y'});
end
function popupmenu2_Callback(hObject, eventdata, handles)
load busy busy
if busy==0
set(handles.pushbutton9,'visible','off');
end
function popupmenu3_Callback(hObject, eventdata, handles)
load busy busy
if busy==0
set(handles.pushbutton9,'visible','off');
end
load dados tensoes1
Popumenu1=get(handles.popupmenu1,'value');
Popumenu3=get(handles.popupmenu3,'value');
Popumenu4=get(handles.popupmenu4,'value');
if Popumenu1==2 || Popumenu1==3
if Popumenu3==1
set(handles.popupmenu4,'visible','off');
set(handles.uipanel2,'Position',[0.8513 0.7610 0.1170 0.1588]);
set(handles.popupmenu1,'Position',[0.0680 0.5764 0.7687 0.2588]);
set(handles.popupmenu3,'Position',[0.0680 0.1647 0.7687 0.2588]);
set(handles.popupmenu4,'value',1);
set(handles.uipanel5,'Position',[0.8513 0.6368 0.1170 0.1022]);
set(handles.popupmenu2,'Position',[0.0680 0.2653 0.7687 0.4489]);
if exist('tensoes1','var')==1
set(handles.text1,'Position',[0.8513 0.5818 0.0627 0.0236]);
set(handles.edit1,'Position',[0.9171 0.5818 0.0201 0.0299]);
set(handles.pushbutton1,'Position',[0.8761 0.5252 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.5252 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.5252 0.0534 0.0362]);
else
set(handles.text1,'Position',[0.8513 0.7186 0.0627 0.0236]);
set(handles.edit1,'Position',[0.9171 0.7186 0.0201 0.0299]);
set(handles.pushbutton1,'Position',[0.8761 0.6651 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.6651 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.6651 0.0534 0.0362]);

```

```

end
else
set(handles.uipanel2,'Position',[0.8513 0.7233 0.1170 0.1965]);
set(handles.popupmenu1,'Position',[0.0680 0.6697 0.7687 0.2018]);
set(handles.popupmenu3,'Position',[0.0680 0.3853 0.7687 0.2018]);
set(handles.popupmenu4,'Position',[0.0680 0.1101 0.7687 0.2018]);
set(handles.uipanel5,'Position',[0.8513 0.6038 0.1170 0.1085]);
set(handles.popupmenu2,'Position',[0.0816 0.3208 0.7687 0.4150]);
if exist('tensoes1','var')==1
set(handles.text1,'Position',[0.8513 0.5550 0.0627 0.0236]);
set(handles.edit1,'Position',[0.9179 0.5550 0.0201 0.0299]);
set(handles.pushbutton1,'Position',[0.8761 0.4906 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.4906 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.4906 0.0534 0.0362]);
else
set(handles.text1,'Position',[0.8513 0.6730 0.0627 0.0236]);
set(handles.edit1,'Position',[0.9179 0.6730 0.0201 0.0299]);
set(handles.pushbutton1,'Position',[0.8761 0.6132 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.6132 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.6132 0.0534 0.0362]);
end
set(handles.popupmenu4,'visible','on');
end
elseif Popupmenu1==4
if Popupmenu3==1
set(handles.popupmenu4,'visible','off');
set(handles.uipanel2,'Position',[0.8513 0.7610 0.1170 0.1588]);
set(handles.popupmenu1,'Position',[0.0680 0.5764 0.7687 0.2588]);
set(handles.popupmenu3,'Position',[0.0680 0.1647 0.7687 0.2588]);
set(handles.popupmenu4,'value',1);
set(handles.pushbutton1,'Position',[0.8761 0.7060 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.7060 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.7060 0.0534 0.0362]);
else
set(handles.uipanel2,'Position',[0.8513 0.7233 0.1170 0.1965]);
set(handles.popupmenu1,'Position',[0.0680 0.6697 0.7687 0.2018]);
set(handles.popupmenu3,'Position',[0.0680 0.3853 0.7687 0.2018]);
set(handles.popupmenu4,'Position',[0.0680 0.1101 0.7687 0.2018]);
set(handles.pushbutton1,'Position',[0.8761 0.6651 0.0534 0.0362]);
set(handles.pushbutton2,'Position',[0.8513 0.6651 0.0534 0.0362]);
set(handles.pushbutton3,'Position',[0.9148 0.6651 0.0534 0.0362]);
set(handles.popupmenu4,'visible','on');
end
end
if Popupmenu3==5 && Popupmenu4==3
set(handles.popupmenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';Tensão horizontal efetiva e total e pressão neutra';Tensão de corte (tau xy)'});
else
set(handles.popupmenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';Tensão horizontal efetiva e total e pressão neutra'});
end
if Popupmenu1==2 || Popupmenu1==3
if exist('tensoes1','var')==1
if Popupmenu3==4
set(handles.popupmenu4,'string',{'Vertical';'Horizontal'});
elseif Popupmenu3==5
set(handles.popupmenu4,'string',{'Vertical';'Horizontal';'De corte (tau xy)'});
elseif Popupmenu3==3
set(handles.popupmenu4,'string',{'Vertical (nos nós pertencentes à linha horizontal)';'Horizontal (nos nós pertencentes à linha horizontal)';'Vertical (em todos os nós)';'Horizontal (em todos os nós)'});
else
set(handles.popupmenu4,'string',{'Vertical (nos nós pertencentes à linha horizontal)';'Horizontal (nos nós pertencentes à linha horizontal)';'Total (nos nós pertencentes à linha horizontal)';'Vertical (em todos os nós)';'Horizontal (em todos os nós)';'Total (em todos os nós)'});
end
else
if Popupmenu3==3
set(handles.popupmenu4,'string',{'Vertical';'Horizontal'});
else
set(handles.popupmenu4,'string',{'Vertical (nos nós pertencentes à linha horizontal)';'Horizontal (nos nós pertencentes à linha horizontal)';'Total (nos nós pertencentes à linha horizontal)';'Vertical (em todos os nós)';'Horizontal (em todos os nós)';'Total (em todos os nós)'});
end
end
elseif Popupmenu1==4
if Popupmenu3==3
set(handles.popupmenu4,'string',{'Vertical';'Horizontal'});

```

```

else
    set(handles.popupmenu4,'string',{'Vertical';'Horizontal';'Total'});
end
end
set(handles.popupmenu2,'value',1);
set(handles.popupmenu4,'value',1);
function popupmenu4_Callback(hObject, eventdata, handles)
load busy busy
if busy==0
    set(handles.pushbutton9,'visible','off');
end
PopupMenu3=get(handles.popupmenu3,'value');
PopupMenu4=get(handles.popupmenu4,'value');
if Popupmenu3==5 && Popupmenu4==3
    set(handles.popupmenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';'Tensão horizontal efetiva e total e pressão neutra';'Tensão de corte (tau xy)'});
else
    set(handles.popupmenu2,'string',{'Tensão vertical efetiva e total e pressão neutra';'Tensão horizontal efetiva e total e pressão neutra'});
end
set(handles.popupmenu2,'value',1);
function pushbutton1_Callback(hObject, eventdata, handles)
load dados tensoes1 IEDGE IPLOD
PopupMenu1=get(handles.popupmenu1,'value');
PopupMenu2=get(handles.popupmenu2,'value');
PopupMenu3=get(handles.popupmenu3,'value');
PopupMenu4=get(handles.popupmenu4,'value');
eixo_tempo=get(handles.radiobutton2,'value');
eixo_x=get(handles.radiobutton2,'value');
escala_linear=get(handles.radiobutton4,'value');
eixo_rp=get(handles.radiobutton12,'value');
no=str2double(get(handles.edit1,'String'));
if Popupmenu1==6 || Popupmenu1==8
    t2=str2double(get(handles.edit2,'String'));
    if isnan(t2)
        Continuar=0;
    else
        Continuar=1;
    end
else
    Continuar=1;
end

if Continuar==1
    if Popupmenu1~=4
        if isnan(no)
            Continuar=0;
        end
    end
end

if Continuar==1
    if IEDGE==1 || IPLOD==1

vdy=0;vdx=0;vdx_y=0;vdy_t=0;vdx_t=0;vdx_y_t=0;vty=0;vtx=0;vty_t=0;vtx_t=0;idY=0;idX=0;idXY=0;itY=0;itX=0;itXY=0;ty=0;tx=0;txy=0;hxy=0;
        set(handles.pushbutton1,'visible','off');
        set(handles.pushbutton2,'visible','on');
        set(handles.pushbutton3,'visible','on');
        busy=1;
        save busy busy
        set(handles.radiobutton2,'enable','inactive');
        set(handles.radiobutton3,'enable','inactive');
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(gca,'yTickmode','auto')
        set(gca,'yTickLabelmode','auto')
        set(gca,'xTickmode','auto')
        set(gca,'xTickLabelmode','auto')
        set(gca,'yLimmode','auto')
        set(gca,'xLimmode','auto')
        if Popupmenu1==2
            set(handles.pushbutton9,'visible','on');
            if exist('tensoes1','var')==1
                set(handles.axes1,'visible','off');
            end
        end
    end
end

```

```

set(handles.axes2,'visible','off');
set(handles.axes3,'visible','on');
set(handles.axes4,'visible','on');
else
set(handles.axes1,'visible','on');
set(handles.axes2,'visible','off');
set(handles.axes3,'visible','off');
set(handles.axes4,'visible','off');
end
if Popupmenu2==1
ty=1;
if Popupmenu3==2
if Popupmenu4==1
vdy=1;
elseif Popupmenu4==2
vdx=1;
elseif Popupmenu4==3
vdx=1;
elseif Popupmenu4==4
vdy_t=1;
elseif Popupmenu4==5
vdx_t=1;
elseif Popupmenu4==6
vdx_t=1;
end
elseif Popupmenu3==3
if Popupmenu4==1
if exist('tensoes1','var')==1
vty=1;
else
idY=1;
end
elseif Popupmenu4==2
if exist('tensoes1','var')==1
vtx=1;
else
idX=1;
end
elseif Popupmenu4==3
vty_t=1;
elseif Popupmenu4==4
vtx_t=1;
end
elseif Popupmenu3==4
if Popupmenu4==1
idY=1;
elseif Popupmenu4==2
idX=1;
end
elseif Popupmenu3==5
if Popupmenu4==1
itY=1;
elseif Popupmenu4==2
itX=1;
elseif Popupmenu4==3
itXY=1;
end
end
elseif Popupmenu2==2
tx=1;
if Popupmenu3==2
if Popupmenu4==1
vdy=1;
elseif Popupmenu4==2
vdx=1;
elseif Popupmenu4==3
vdx=1;
elseif Popupmenu4==4
vdy_t=1;
elseif Popupmenu4==5
vdx_t=1;
elseif Popupmenu4==6
vdx_t=1;
end
elseif Popupmenu3==3
if Popupmenu4==1

```

```

        vty=1;
    elseif Popupmenu4==2
        vtx=1;
    elseif Popupmenu4==3
        vty_t=1;
    elseif Popupmenu4==4
        vtx_t=1;
    end
elseif Popupmenu3==4
    if Popupmenu4==1
        idY=1;
    elseif Popupmenu4==2
        idX=1;
    end
elseif Popupmenu3==5
    if Popupmenu4==1
        itY=1;
    elseif Popupmenu4==2
        itX=1;
    elseif Popupmenu4==3
        itXY=1;
    end
end
end
elseif Popupmenu2==3
    txy=1;
    if Popupmenu3==5
        if Popupmenu4==3
            itXY=1;
        end
    end
end
end
graficos_p_hy_ty();
pause(0.1)
elseif Popupmenu1==3
    set(handles.pushbutton9,'visible','on');
    if exist('tensoes1','var')==1
        set(handles.axes1,'visible','off');
        set(handles.axes2,'visible','off');
        set(handles.axes3,'visible','on');
        set(handles.axes4,'visible','on');
    else
        set(handles.axes1,'visible','on');
        set(handles.axes2,'visible','off');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
    end
end
hxy=1;
if Popupmenu2==1
    ty=1;
    if Popupmenu3==2
        if Popupmenu4==1
            vdy=1;
        elseif Popupmenu4==2
            vdx=1;
        elseif Popupmenu4==3
            vdx=1;
        elseif Popupmenu4==4
            vdy_t=1;
        elseif Popupmenu4==5
            vdx_t=1;
        elseif Popupmenu4==6
            vdx_t=1;
        end
    elseif Popupmenu3==3
        if Popupmenu4==1
            if exist('tensoes1','var')==1
                vty=1;
            else
                idY=1;
            end
        elseif Popupmenu4==2
            if exist('tensoes1','var')==1
                vtx=1;
            else
                idX=1;
            end
        end
    end
end
end

```



```

elseif Popupmenu4==3
    vty_t=1;
elseif Popupmenu4==4
    vtx_t=1;
end
elseif Popupmenu3==4
    if Popupmenu4==1
        idY=1;
    elseif Popupmenu4==2
        idX=1;
    end
elseif Popupmenu3==5
    if Popupmenu4==1
        itY=1;
    elseif Popupmenu4==2
        itX=1;
    elseif Popupmenu4==3
        itXY=1;
    end
end
elseif Popupmenu2==2
    tx=1;
    if Popupmenu3==2
        if Popupmenu4==1
            vdy=1;
        elseif Popupmenu4==2
            vdx=1;
        elseif Popupmenu4==3
            vdx=1;
        elseif Popupmenu4==4
            vdy_t=1;
        elseif Popupmenu4==5
            vdx_t=1;
        elseif Popupmenu4==6
            vdx_t=1;
        end
    elseif Popupmenu3==3
        if Popupmenu4==1
            vty=1;
        elseif Popupmenu4==2
            vtx=1;
        elseif Popupmenu4==3
            vty_t=1;
        elseif Popupmenu4==4
            vtx_t=1;
        end
    elseif Popupmenu3==4
        if Popupmenu4==1
            idY=1;
        elseif Popupmenu4==2
            idX=1;
        end
    elseif Popupmenu3==5
        if Popupmenu4==1
            itY=1;
        elseif Popupmenu4==2
            itX=1;
        elseif Popupmenu4==3
            itXY=1;
        end
    end
end
elseif Popupmenu2==3
    txy=1;
    if Popupmenu3==5
        if Popupmenu4==3
            itXY=1;
        end
    end
end
end
graficos_p_hy_ty();
pause(0.1)
elseif Popupmenu1==4
    set(handles.pushbutton9,'visible','on');
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');

```

```

set(handles.axes4,'visible','off');
axes(handles.axes1)
if Popupmenu3==2
    if Popupmenu4==1
        vdy=1;
    elseif Popupmenu4==2
        vdx=1;
    elseif Popupmenu4==3
        vdx=1;
    end
elseif Popupmenu3==3
    if Popupmenu4==1
        vty=1;
    elseif Popupmenu4==2
        vtx=1;
    end
end
desl_x_y_malha();
pause(0.1)
elseif Popupmenu1==5
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes1)
    grafico_desl_y_no_vs_t();
elseif Popupmenu1==6
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes1)
    grafico_desl_y_nos_profund();
elseif Popupmenu1==7
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes1)
    grafico_rpp_vs_t();
elseif Popupmenu1==8
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes1)
    grafico_rpp_vs_xa();
end
busy=0;
save busy busy
pause on
set(handles.radiobutton2,'enable','on');
set(handles.radiobutton3,'enable','on');
set(handles.pushbutton2,'visible','off');
set(handles.pushbutton3,'visible','off');
set(handles.pushbutton2,'String','Pausa');
Popupmenu1=get(handles.popupmenu1,'value');
pause(0.1)
if Popupmenu1>1
    set(handles.pushbutton1,'visible','on');
end
else
    errordlg('Nenhuma carga aplicada','Erro');
end
else
    errordlg('Preencha todos os campos','Erro');
end
function edit1_Callback(hObject, eventdata, handles)
load busy busy
if busy==0
    set(handles.pushbutton9,'visible','off');
end
no=get(handles.edit1,'String');
if isempty(no)==0
    load dados NPOIN
    no=str2double(get(handles.edit1,'String'));

```

```

if isnan(no) || no>NPOIN || no<1
    set(handles.edit1,'String','');
    if no>NPOIN
        errordlg('Número do nó não pode ser maior que "NPOIN"', 'Erro');
    elseif no<1
        errordlg('Número do nó tem que ser maior ou igual a 1', 'Erro');
    elseif isnan(no)
        errordlg('Número do nó tem que ser um valor numérico', 'Erro');
    end
end
end
end
function edit2_Callback(hObject, eventdata, handles)
t2=get(handles.edit2,'String');
if isempty(t2)==0
    load dados NINCR
    t2=str2double(get(handles.edit2,'String'));
    if isnan(t2) || t2>NINCR || t2<1
        set(handles.edit2,'String','');
        if t2>NINCR
            errordlg('Número do incremento de tempo não pode ser maior que "NINCR"', 'Erro');
        elseif t2<1
            errordlg('Número do incremento de tempo tem que ser maior ou igual a 1', 'Erro');
        elseif isnan(t2)
            errordlg('Número do incremento de tempo tem que ser um valor numérico', 'Erro');
        end
    end
end
end
function figure1_CreateFcn(hObject, eventdata, handles)
busy=0;
save busy busy
load dados NELEM MNODE elementos coordenadas NDIME IEDGE valor_carga NEDGE Xmin_carga Xmax_carga Ymin_carga IPLOD
cargas_pontuais a
malha()
title('Representação do problema em análise', 'FontWeight', 'bold')
xlabel('Distância ao eixo x (m)')
ylabel('Distância ao eixo y (m)')
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.pushbutton2, 'visible', 'off');
set(handles.pushbutton2, 'String', 'Pausa');
set(handles.pushbutton3, 'visible', 'off');
pause off
function pushbutton2_Callback(hObject, eventdata, handles)
pushbutton2_string=get(handles.pushbutton2,'String');
if strcmp(pushbutton2_string,'Pausa')
    set(handles.pushbutton2,'String','Continuar');
    pushbutton2_string=get(handles.pushbutton2,'String');
    while strcmp(pushbutton2_string,'Continuar')
        pause(1)
        pushbutton2_string=get(handles.pushbutton2,'String');
    end
end
else
    set(handles.pushbutton2,'String','Pausa');
end
end
function pushbutton6_Callback(hObject, eventdata, handles)
resposta=questdlg('Menu inicial?', 'Sim', 'Não', 'Sim');
if strcmp(resposta,'Sim')
    delete(menu_resultados);
    delete(ajuda_dados);
    delete(ajuda_malha);
    if exist('dados3.mat','file')
        delete(ajuda_resultados);
    end
    delete busy.mat dados.mat dados2.mat dados3.mat dados4.mat
    menu_principal()
end
end
function pushbutton7_Callback(hObject, eventdata, handles)
ajuda_malha()
function pushbutton8_Callback(hObject, eventdata, handles)
ajuda_dados()
function pushbutton9_Callback(hObject, eventdata, handles)
ajuda_resultados()
function figure1_CloseRequestFcn(hObject, eventdata, handles)
resposta=questdlg('Fechar programa?', 'Sim', 'Não', 'Sim');
if strcmp(resposta,'Sim')
    delete(hObject);
    delete(ajuda_dados);

```

```

delete(ajuda_malha);
if exist('dados3.mat','file')
    delete(ajuda_resultados);
end
delete busy.mat dados.mat dados2.mat dados3.mat dados4.mat
end
function radiobutton2_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
escala_linear=get(handles.radiobutton4,'value');
eixo_rp=get(handles.radiobutton12,'value');
eixo_tempo=get(handles.radiobutton2,'value');
eixo_x=get(handles.radiobutton2,'value');
if eixo_tempo==1
    set(handles.radiobutton3,'value',0);
else
    set(handles.radiobutton3,'value',1);
end
no=get(handles.edit1,'String');
if isempty(no)==0
    no=str2double(get(handles.edit1,'String'));
    if Popupmenu1==5
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','on');
        set(handles.axes2,'visible','off');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes1)
        grafico_desl_y_no_vs_t()
    elseif Popupmenu1==7
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','off');
        set(handles.axes2,'visible','on');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes2)
        grafico_rpp_vs_t()
    elseif Popupmenu1==8
        t2=get(handles.edit2,'String');
        if isempty(t2)==0
            t2=str2double(get(handles.edit2,'String'));
            cla(handles.axes1,'reset')
            cla(handles.axes2,'reset')
            cla(handles.axes3,'reset')
            cla(handles.axes4,'reset')
            set(handles.axes1,'visible','on');
            set(handles.axes2,'visible','off');
            set(handles.axes3,'visible','off');
            set(handles.axes4,'visible','off');
            axes(handles.axes1)
            grafico_rpp_vs_xa()
        end
    end
end
function radiobutton3_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
escala_linear=get(handles.radiobutton4,'value');
eixo_rp=get(handles.radiobutton12,'value');
eixo_tempo_mod=get(handles.radiobutton3,'value');
eixo_xa=get(handles.radiobutton3,'value');
if eixo_tempo_mod==1
    eixo_tempo=0;
    eixo_x=0;
    set(handles.radiobutton2,'value',0);
else
    eixo_tempo=1;
    eixo_x=1;
    set(handles.radiobutton2,'value',1);
end
no=get(handles.edit1,'String');
if isempty(no)==0

```

```

no=str2double(get(handles.edit1,'String'));
if Popupmenu1==5
    cla(handles.axes1,'reset')
    cla(handles.axes2,'reset')
    cla(handles.axes3,'reset')
    cla(handles.axes4,'reset')
    set(handles.axes1,'visible','on');
    set(handles.axes2,'visible','off');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes1)
    grafico_desl_y_no_vs_t()
elseif Popupmenu1==7
    cla(handles.axes1,'reset')
    cla(handles.axes2,'reset')
    cla(handles.axes3,'reset')
    cla(handles.axes4,'reset')
    set(handles.axes1,'visible','off');
    set(handles.axes2,'visible','on');
    set(handles.axes3,'visible','off');
    set(handles.axes4,'visible','off');
    axes(handles.axes2)
    grafico_rpp_vs_t()
elseif Popupmenu1==8
    t2=get(handles.edit2,'String');
    if isempty(t2)==0
        t2=str2double(get(handles.edit2,'String'));
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','on');
        set(handles.axes2,'visible','off');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes1)
        grafico_rpp_vs_xa()
    end
end
end
end
function radiobutton4_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
escala_linear=get(handles.radiobutton4,'value');
if escala_linear==1
    set(handles.radiobutton5,'value',0);
    if Popupmenu1==2 || Popupmenu1==3
        set(handles.axes4,'xscale','linear');
    elseif Popupmenu1==5
        set(handles.axes1,'xscale','linear');
    elseif Popupmenu1==7
        set(handles.axes2,'xscale','linear');
    end
else
    set(handles.radiobutton5,'value',1);
    if Popupmenu1==2 || Popupmenu1==3
        set(handles.axes4,'xscale','log');
    elseif Popupmenu1==5
        set(handles.axes1,'xscale','log');
    elseif Popupmenu1==7
        set(handles.axes2,'xscale','log');
    end
end
end
function radiobutton5_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
escala_logaritmica=get(handles.radiobutton5,'value');
if escala_logaritmica==1
    set(handles.radiobutton4,'value',0);
    if Popupmenu1==2 || Popupmenu1==3
        set(handles.axes4,'xscale','log');
    elseif Popupmenu1==5
        set(handles.axes1,'xscale','log');
    elseif Popupmenu1==7
        set(handles.axes2,'xscale','log');
    end
else
    set(handles.radiobutton4,'value',1);

```

```

if Popupmenu1==2 || Popupmenu1==3
    set(handles.axes4,'xscale','linear');
elseif Popupmenu1==5
    set(handles.axes1,'xscale','linear');
elseif Popupmenu1==7
    set(handles.axes2,'xscale','linear');
end
end
function radiobutton12_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
eixo_tempo=get(handles.radiobutton2,'value');
eixo_x=get(handles.radiobutton2,'value');
escala_linear=get(handles.radiobutton4,'value');
eixo_rp=get(handles.radiobutton12,'value');
if eixo_rp==1
    set(handles.radiobutton13,'value',0);
else
    set(handles.radiobutton13,'value',1);
end
no=get(handles.edit1,'String');
if isempty(no)==0
    no=str2double(get(handles.edit1,'String'));
    if Popupmenu1==7
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','off');
        set(handles.axes2,'visible','on');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes2)
        grafico_rpp_vs_t()
    elseif Popupmenu1==8
        t2=get(handles.edit2,'String');
        if isempty(t2)==0
            t2=str2double(get(handles.edit2,'String'));
            cla(handles.axes1,'reset')
            cla(handles.axes2,'reset')
            cla(handles.axes3,'reset')
            cla(handles.axes4,'reset')
            set(handles.axes1,'visible','on');
            set(handles.axes2,'visible','off');
            set(handles.axes3,'visible','off');
            set(handles.axes4,'visible','off');
            axes(handles.axes1)
            grafico_rpp_vs_xa()
        end
    end
end
function radiobutton13_Callback(hObject, eventdata, handles)
Popupmenu1=get(handles.popupmenu1,'value');
eixo_tempo=get(handles.radiobutton2,'value');
eixo_x=get(handles.radiobutton2,'value');
escala_linear=get(handles.radiobutton4,'value');
eixo_rpp=get(handles.radiobutton13,'value');
if eixo_rpp==1
    eixo_rp=0;
    set(handles.radiobutton12,'value',0);
else
    eixo_rp=1;
    set(handles.radiobutton12,'value',1);
end
no=get(handles.edit1,'String');
if isempty(no)==0
    no=str2double(get(handles.edit1,'String'));
    if Popupmenu1==7
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','off');
        set(handles.axes2,'visible','on');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes2)
    end
end

```

```

    grafico_rpp_vs_t()
elseif Popupmenu1==8
    t2=get(handles.edit2,'String');
    if isempty(t2)==0
        t2=str2double(get(handles.edit2,'String'));
        cla(handles.axes1,'reset')
        cla(handles.axes2,'reset')
        cla(handles.axes3,'reset')
        cla(handles.axes4,'reset')
        set(handles.axes1,'visible','on');
        set(handles.axes2,'visible','off');
        set(handles.axes3,'visible','off');
        set(handles.axes4,'visible','off');
        axes(handles.axes1)
        grafico_rpp_vs_xa()
    end
end
end
end

```

#### **function varargout = ajuda\_dados(varargin)**

```

function figure1_CreateFcn(hObject, eventdata, handles)
load dados
for i=1:NPOIN
    numero_no_x_y(i,1:3)=[coordenadas(i,1) coordenadas(i,2) coordenadas(i,3)];
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    for i=1:NINCR
        numero_incremento_tempo(i,1:3)=[i tempo(i,1) tempo_mod(i,1)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Fator T'};
else
    for i=1:NINCR
        numero_incremento_tempo(i,1:2)=[i tempo(i,1)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo'};
end
nome_colunas1={'Número do nó','X','Y'};
tabela1=uitable(numero_no_x_y,nome_colunas1,'Position',[200 50 344 525],'ColumnWidth',100);
tabela2=uitable(numero_incremento_tempo,nome_colunas2,'Position',[700 50 421 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')

```

#### **function varargout = ajuda\_malha(varargin)**

```

function figure1_CreateFcn(hObject, eventdata, handles)
load dados
malha()
for i=1:NPOIN
    text(coordenadas(i,2),coordenadas(i,3),num2str(coordenadas(i,1)), 'FontSize',7,'HorizontalAlignment','right','VerticalAlignment','top');
    text(coordenadas(i,2),coordenadas(i,3),'\bullet', 'FontSize',6,'HorizontalAlignment','center');
end

```

#### **function varargout = ajuda\_resultados(varargin)**

```

function figure1_CreateFcn(hObject, eventdata, handles)
load dados3
load dados tensoes1 tempo tempo_mod NPOIN IEDGE IPLOD titulo_problema
if IEDGE==1 || IPLOD==1
    if Popupmenu1==2 || Popupmenu1==3
        if exist('tensoes1','var')==1
            if hxy==1
                if vdy_t==1 || vdx_t==1 || vdx_t==1
                    if ty==1
                        load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc y22 x22
                    else
                        load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x y22 x22
                    end
                end
                tamanho_no_x=size(no_x);
                for i=1:tamanho_no_x
                    if no_x(i,1)==no
                        local_no=i;
                    end
                end
            end
        end
    end
end

```

```

end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nível_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1
nível_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
        else
            if ty==1
                for t=1:t1
                    data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nível_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
            end
            else
                for t=1:t1
                    data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nível_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
            end
        end
        data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
        for i=1:NPOIN
            data3(i,1:3)=[i y22(i,t1) x22(i,t1)];
        end
        nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
        nome_colunas3={'Número do nó','Deslocamento vertical','Deslocamento horizontal'};
        tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
        tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
        tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
        set(tabela1,'editable',0,'units','normalized')
        set(tabela2,'editable',0,'units','normalized')
        set(tabela3,'editable',0,'units','normalized')
    elseif vty==1 || vtx==1 || vtxy==1
        if ty==1
            load dados2 no P nível_agua desl_y no_x t1 desl_x total zc
        else
            load dados2 no P nível_agua desl_y no_x t1 desl_x total zc_x
        end
        load dados4
        tamanho_no_x=size(no_x);
        for i=1:tamanho_no_x
            if no_x(i,1)==no
                local_no=i;
            end
        end
        if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
            if ty==1
                for t=1:t1
                    data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nível_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
            end
            else
                for t=1:t1
                    data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1
nível_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
            end
        end
        else
            if ty==1
                for t=1:t1

```



```

        data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
end
if vty==1
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1) ty1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva'};
elseif vtx==1
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1) tx1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal','Tensão horizontal efetiva'};
else
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1) ty1(:,t1) tx1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva','Tensão
horizontal efetiva'};
end
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
elseif vty_t==1 || vtx_t==1 || vtxy_t==1
if ty==1
    load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc y22 x22
else
    load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x y22 x22
end
load dados4
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
tamanho_no_x=size(no_x);
for i=1:tamanho_no_x
    if no_x(i,1)==no
        local_no=i;
    end
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1
nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
else
    if ty==1
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
end
if vty_t==1
    for i=1:NPOIN
        data3(i,1:4)=[i y22(i,t1) x22(i,t1) ty1(i,t1)];
    end
end

```

```

nome_colunas3={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva'};
tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
elseif vtx_t==1
for i=1:NPOIN
data3(i,1:4)=[i y22(i,t1) x22(i,t1) tx1(i,t1)];
end
nome_colunas3={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão horizontal efetiva'};
tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
else
for i=1:NPOIN
data3(i,1:5)=[i y22(i,t1) x22(i,t1) ty1(i,t1) tx1(i,t1)];
end
nome_colunas3={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva','Tensão
horizontal efetiva'};
tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
end
nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
set(tabela3,'editable',0,'units','normalized')
elseif itY==1
load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc2
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
if ty==1
nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
else
nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
end
else
if ty==1
nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
else
nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
end
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
tamanho_no_x=size(no_x);
for i=1:tamanho_no_x
if no_x(i,1)==no
local_no=i;
end
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
for t=1:t1
data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1
nivel_agua(1,t)];
end
else
for t=1:t1
data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
end
end
nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
elseif itX==1
if ty==1
load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
else
load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc2
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
tamanho_no_x=size(no_x);
for i=1:tamanho_no_x
if no_x(i,1)==no
local_no=i;
end
end
end
end

```

```

if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nível_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1
nível_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
else
    if ty==1
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nível_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nível_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
elseif itXY==1
    if ty==1
        load dados2 no P nível_agua desl_y no_x t1 desl_x total zc
    elseif tx==1
        load dados2 no P nível_agua desl_y no_x t1 desl_x total zc_x
    else
        load dados2 no P nível_agua desl_y no_x t1 desl_x zc2
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nível_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        elseif tx==1
            for t=1:t1
                data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1
nível_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) zc2(1,t) P(1,t)*-1 nível_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tau xy','Pressão neutra','Zw'};
        end
    else
        if ty==1
            for t=1:t1

```

```

        data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
    elseif tx==1
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tau
xy','Pressão neutra','Zw'};
    end
    end
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
else
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
    else
        load dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x
    end
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1
nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:9)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1
nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    else
        if ty==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Tensão
horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
end
end
else
    if vdy_t==1 || vdx_t==1 || vdx_y_t==1
        if ty==1

```

```

        load dados2 no P nivel_agua desl_y no_x t1 total zc y22
    else
        load dados2 no P nivel_agua desl_y no_x t1 total zc_x y22
    end
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    else
        if ty==1
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
    for i=1:NPOIN
        data3(i,1:2)=[i y22(i,t1)];
    end
    nome_colunas1={'Número do nó','X','Deslocamento vertical'};
    nome_colunas3={'Número do nó','Deslocamento vertical'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
    tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
    set(tabela3,'editable',0,'units','normalized')
elseif vty==1 || vtx==1 || vtxy==1
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc
    else
        load dados2 no P nivel_agua desl_y no_x t1 total zc_x
    end
    load dados4
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão vertical total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
        end
    end
end

```

```

        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
    else
    if ty==1
    for t=1:t1
        data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
    else
    for t=1:t1
        data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
    end
    end
    if vty==1
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) ty1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Tensão vertical efetiva'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 515 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[535 50 740 525],'ColumnWidth',200);
    elseif vtx==1
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) tx1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Tensão horizontal efetiva'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 515 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[535 50 740 525],'ColumnWidth',200);
    else
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) ty1(:,t1) tx1(:,t1)];
    nome_colunas1={'Número do nó','X','Deslocamento vertical','Tensão vertical efetiva','Tensão horizontal efetiva'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 640 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[660 50 615 525],'ColumnWidth',200);
    end
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
    elseif vty_t==1 || vtx_t==1 || vtxy_t==1
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc y22
    else
        load dados2 no P nivel_agua desl_y no_x t1 total zc_x y22
    end
    load dados4
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
    for t=1:t1
        data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão vertical
total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
    for t=1:t1
        data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
    end
    else
    if ty==1
    for t=1:t1
        data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
    else
    for t=1:t1
        data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end

```

```

        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
    end
end
if vty_t==1
    for i=1:NPOIN
        data3(i,1:3)=[i y22(i,t1) ty1(i,t1)];
    end
    nome_colunas3={'Número do nó','Deslocamento vertical','Tensão vertical efetiva'};
elseif vtx_t==1
    for i=1:NPOIN
        data3(i,1:3)=[i y22(i,t1) tx1(i,t1)];
    end
    nome_colunas3={'Número do nó','Deslocamento vertical','Tensão horizontal efetiva'};
else
    for i=1:NPOIN
        data3(i,1:4)=[i y22(i,t1) ty1(i,t1) tx1(i,t1)];
    end
    nome_colunas3={'Número do nó','Deslocamento vertical','Tensão vertical efetiva','Tensão horizontal efetiva'};
end
nome_colunas1={'Número do nó','X','Deslocamento vertical'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
set(tabela3,'editable',0,'units','normalized')
elseif itY==1
    load dados2 no P nivel_agua desl_y no_x t1 total zc2
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão vertical
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        else
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    else
        if ty==1
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
        else
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
        end
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
    else
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
    end
    nome_colunas1={'Número do nó','X','Deslocamento vertical'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
elseif itX==1
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc
    else
        load dados2 no P nivel_agua desl_y no_x t1 total zc2
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x

```

```

        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator  $\Gamma$ ','Deslocamento vertical','Tensão vertical
total','Tensão vertical efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator  $\Gamma$ ','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        end
    else
        if ty==1
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
        end
    end
    nome_colunas1={'Número do nó','X','Deslocamento vertical'};
    tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
    tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
    set(tabela1,'editable',0,'units','normalized')
    set(tabela2,'editable',0,'units','normalized')
elseif itXY==1
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc
    elseif tx==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc_x
    else
        load dados2 no P nivel_agua desl_y no_x t1 zc2
    end
    data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator  $\Gamma$ ','Deslocamento vertical','Tensão vertical
total','Tensão vertical efetiva','Pressão neutra','Zw'};
        elseif tx==1
            for t=1:t1
                data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator  $\Gamma$ ','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator  $\Gamma$ ','Deslocamento vertical','Tau xy','Pressão
neutra','Zw'};
        end
    else
        if ty==1
            for t=1:t1

```



```

        data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
    end
    nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
    elseif tx==1
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) desl_y(local_no,t) zc2(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tau xy','Pressão neutra','Zw'};
    end
end
nome_colunas1={'Número do nó','X','Deslocamento vertical'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
else
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 total zc
    else
        load dados2 no P nivel_agua desl_y no_x t1 total zc_x
    end
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão vertical
total','Tensão vertical efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:8)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Tensão horizontal
total','Tensão horizontal efetiva','Pressão neutra','Zw'};
    end
else
    if ty==1
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão vertical total','Tensão vertical
efetiva','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) desl_y(local_no,t) total(1,t) zc_x(1,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Tensão horizontal total','Tensão
horizontal efetiva','Pressão neutra','Zw'};
    end
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
nome_colunas1={'Número do nó','X','Deslocamento vertical'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
end
else
    if hxy==1
        if vdy_t==1 || vdx_t==1 || vdx_t==1
            if ty==1
                load dados2 no P nivel_agua desl_y no_x t1 desl_x y22 x22
            end
        end
    end
end

```

```

else
    load dados2 no P nivel_agua desl_y no_x t1 desl_x y22 x22
end
tamanho_no_x=size(no_x);
for i=1:tamanho_no_x
    if no_x(i,1)==no
        local_no=i;
    end
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:7)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Pressão neutra','Zw'};
    end
else
    if ty==1
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Pressão
neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Pressão
neutra','Zw'};
    end
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
for i=1:NPOIN
    data3(i,1:3)=[i y22(i,t1) x22(i,t1)];
end
nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
nome_colunas3={'Número do nó','Deslocamento vertical','Deslocamento horizontal'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
set(tabela3,'editable',0,'units','normalized')
else
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1 desl_x
    else
        load dados2 no P nivel_agua desl_y no_x t1 desl_x
    end
    tamanho_no_x=size(no_x);
    for i=1:tamanho_no_x
        if no_x(i,1)==no
            local_no=i;
        end
    end
    if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
        if ty==1
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Pressão neutra','Zw'};
        else
            for t=1:t1
                data2(t,1:7)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
            end
            nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Deslocamento
horizontal','Pressão neutra','Zw'};
        end
    end
end

```

```

else
    if ty==1
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Pressão
neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) desl_y(local_no,t) desl_x(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Deslocamento horizontal','Pressão
neutra','Zw'};
    end
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1) desl_x(:,t1)];
nome_colunas1={'Número do nó','X','Deslocamento vertical','Deslocamento horizontal'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
end
else
    if vdy_t==1 || vdx_t==1 || vdx_y_t==1
        if ty==1
            load dados2 no P nivel_agua desl_y no_x t1 y22
        else
            load dados2 no P nivel_agua desl_y no_x t1 y22
        end
        tamanho_no_x=size(no_x);
        for i=1:tamanho_no_x
            if no_x(i,1)==no
                local_no=i;
            end
        end
        end
        if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
            if ty==1
                for t=1:t1
                    data2(t,1:6)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Pressão neutra','Zw'};
            else
                for t=1:t1
                    data2(t,1:6)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Pressão neutra','Zw'};
            end
        end
        else
            if ty==1
                for t=1:t1
                    data2(t,1:5)=[t tempo(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Pressão neutra','Zw'};
            else
                for t=1:t1
                    data2(t,1:5)=[t tempo(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
                end
                nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Pressão neutra','Zw'};
            end
        end
        data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
        for i=1:NPOIN
            data3(i,1:2)=[i y22(i,t1)];
        end
        nome_colunas1={'Número do nó','X','Deslocamento vertical'};
        nome_colunas3={'Número do nó','Deslocamento vertical'};
        tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
        tabela2=uitable(data2,nome_colunas2,'Position',[410 50 565 525],'ColumnWidth',200);
        tabela3=uitable(data3,nome_colunas3,'Position',[980 50 294 525],'ColumnWidth',125);
        set(tabela1,'editable',0,'units','normalized')
        set(tabela2,'editable',0,'units','normalized')
        set(tabela3,'editable',0,'units','normalized')
    end
else
    if ty==1
        load dados2 no P nivel_agua desl_y no_x t1
    else

```

```

    load dados2 no P nivel_agua desl_y no_x t1
end
tamanho_no_x=size(no_x);
for i=1:tamanho_no_x
    if no_x(i,1)==no
        local_no=i;
    end
end
if IEDGE==1 && strcmp(titulo_problema,'bidimensional ')
    if ty==1
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:6)=[t tempo(t,1) tempo_mod(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Fator T','Deslocamento vertical','Pressão neutra','Zw'};
    end
else
    if ty==1
        for t=1:t1
            data2(t,1:5)=[t tempo(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Pressão neutra','Zw'};
    else
        for t=1:t1
            data2(t,1:5)=[t tempo(t,1) desl_y(local_no,t) P(1,t)*-1 nivel_agua(1,t)];
        end
        nome_colunas2={'Número do incremento de tempo','Tempo','Deslocamento vertical','Pressão neutra','Zw'};
    end
end
data1=[no_x(:,1) no_x(:,2) desl_y(:,t1)];
nome_colunas1={'Número do nó','X','Deslocamento vertical'};
tabela1=uitable(data1,nome_colunas1,'Position',[17.5 50 390 525],'ColumnWidth',125);
tabela2=uitable(data2,nome_colunas2,'Position',[410 50 865 525],'ColumnWidth',200);
set(tabela1,'editable',0,'units','normalized')
set(tabela2,'editable',0,'units','normalized')
end
end
elseif Popupmenu1==4
    if vty==1
        load dados2 t ty1
        load dados deslocamentos
        for i=1:NPOIN
            data1(i,1:4)=[i deslocamentos(i,3*t) deslocamentos(i,3*t-1) ty1(i,t)];
        end
        nome_colunas1={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva'};
        tabela1=uitable(data1,nome_colunas1,'Position',[370 50 544 525],'ColumnWidth',125);
        set(tabela1,'editable',0,'units','normalized')
    elseif vtx==1
        load dados2 t tx1
        load dados deslocamentos
        for i=1:NPOIN
            data1(i,1:4)=[i deslocamentos(i,3*t) deslocamentos(i,3*t-1) tx1(i,t)];
        end
        nome_colunas1={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão horizontal efetiva'};
        tabela1=uitable(data1,nome_colunas1,'Position',[370 50 544 525],'ColumnWidth',125);
        set(tabela1,'editable',0,'units','normalized')
    elseif vtxy==1
        load dados2 t ty1 tx1
        load dados deslocamentos
        for i=1:NPOIN
            data1(i,1:5)=[i deslocamentos(i,3*t) deslocamentos(i,3*t-1) ty1(i,t) tx1(i,t)];
        end
        nome_colunas1={'Número do nó','Deslocamento vertical','Deslocamento horizontal','Tensão vertical efetiva','Tensão horizontal
efetiva'};
        tabela1=uitable(data1,nome_colunas1,'Position',[310 50 669 525],'ColumnWidth',125);
        set(tabela1,'editable',0,'units','normalized')
    else
        load dados2 t
        load dados deslocamentos
        for i=1:NPOIN
            data1(i,1:3)=[i deslocamentos(i,3*t) deslocamentos(i,3*t-1)];
        end
    end
end

```

```

end
nome_colunas1={'Número do nó','Deslocamento vertical','Deslocamento horizontal'};
tabela1=uitable(data1,nome_colunas1,'Position',[430 50 419 525],'ColumnWidth',125);
set(tabela1,'editable',0,'units','normalized')
end
end
end
end

```

### function caixa

```

if strcmp(titulo_problema,'unidimensional')
    valor_escala_0_05=0.015;
    valor_escala_0_07=0.020;
    valor_escala_0_2=0.050;
elseif strcmp(titulo_problema,'bidimensional ')
    valor_escala_0_05=0.05;
    valor_escala_0_07=0.07;
    valor_escala_0_2=0.2;
end
for n=1:NPOIN
    Xcoord(1,n)=coordenadas(n,2);
    Ycoord(1,n)=coordenadas(n,3);
end
Xmax=max(Xcoord);
Xmin=min(Xcoord);
Ymax=max(Ycoord);
Ymin=min(Ycoord);
axis([Xmin-1*a Xmax+1*a Ymin-1*a Ymax+Ymax/3+1*a])
plot([Xmin,Xmin],[Ymin,Ymax],'r-');
hold on
plot([Xmin,Xmax],[Ymax,Ymax],'r-');
hold on
plot([Xmax,Xmax],[Ymin,Ymax],'r-');
hold on
plot([Xmin,Xmax],[Ymin,Ymin],'r-');
hold on
text(Xmin,Ymin,' \rightarrow','FontSize',20,'HorizontalAlignment','left','VerticalAlignment','middle');
text(Xmin,Ymin,'x','FontSize',11,'HorizontalAlignment','left','VerticalAlignment','cap');
text(Xmin,Ymin,' \uparrow','FontSize',20,'HorizontalAlignment','right','VerticalAlignment','baseline');
text(Xmin,Ymin,'y','FontSize',11,'HorizontalAlignment','right','VerticalAlignment','baseline');
plot(Xmax+1*a,Ymax+1*a,'w');
plot(Xmin-1*a,Ymin-1*a,'w');
plot([coordenadas(no,2)-valor_escala_0_05*a,coordenadas(no,2)-valor_escala_0_05*a],[coordenadas(no,3),Ymax+Ymax/3],'k-','linewidth',2);
plot([coordenadas(no,2)+valor_escala_0_07*a,coordenadas(no,2)+valor_escala_0_07*a],[coordenadas(no,3),Ymax+Ymax/3],'k-','linewidth',2);
text(coordenadas(no,2)+valor_escala_0_2*a,Ymax+Ymax/3,'Zw');
if IEDGE==1
    numero_nos=size(valor_carga);
    for i=1:NEDGE
        plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)'],'b-');
        hold on
        plot([Xmin_carga(i,1),Xmin_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.75*a'],'b-');
        hold on
        plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1)+0.75*a,Ymin_carga(i,1)+0.75*a'],'b-');
        hold on
        plot([Xmax_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.75*a'],'b-');
        if max(valor_carga(2*i-1,1:numero_nos(2)))>0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
        elseif min(valor_carga(2*i-1,1:numero_nos(2)))<0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),'
\uparrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
        end
        if max(valor_carga(2*i,1:numero_nos(2)))>0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),'
\leftarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        elseif min(valor_carga(2*i,1:numero_nos(2)))<0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),'
\rightarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        end
    end
end
if IPLOD==1
    numero_cargas_pontuais=size(cargas_pontuais);
    for i=1:numero_cargas_pontuais

```

```

        if cargas_pontuais(i,3)~=0
            if cargas_pontuais(i,3)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.75*a,'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.75*a,'
\uparrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            end
        end
        if cargas_pontuais(i,2)~=0
            if cargas_pontuais(i,2)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.75*a,'
\leftarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.75*a,'
\rightarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            end
        end
    end
end
elseif IPLOD==1
    numero_cargas_pontuais=size(cargas_pontuais);
    for i=1:numero_cargas_pontuais
        if cargas_pontuais(i,3)~=0
            if cargas_pontuais(i,3)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3),'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3),'
\uparrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            end
        end
        if cargas_pontuais(i,2)~=0
            if cargas_pontuais(i,2)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3),'
\leftarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3),'
\rightarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            end
        end
    end
end
end

```

### **function graficos p hv ty**

```

save dados3 Popupmenu1 vdy vdx vdx_y vdy_t vdx_t vdy_t vty vtx vty_t vtx_t idY idX idXY itY itX itXY ty tx txy hxy
load dados coordenadas NINCR pressao_neutra gama_agua NPOIN deslocamentos NELEM elementos aux_NGAUS valor_carga NEDGE
Xmin_carga Xmax_carga Ymin_carga tempo incr_tempo_tensoes tempo_mod cargas_pontuais a titulo_problema
no_desl=no;
no_tensoes=no;
if strcmp(titulo_problema,'unidimensional')
    valor_escala_0_2=0.075;
    valor_escala_0_5a=0.40;
    valor_escala_0_3=0.05;
    valor_escala_0_25=0.05;
    valor_escala_0_75=0.4;
    valor_escala_0_4=0.4;
    valor_escala_0_5b=0.35;
elseif strcmp(titulo_problema,'bidimensional ')
    valor_escala_0_2=0.2;
    valor_escala_0_5a=0.5;
    valor_escala_0_3=0.3;
    valor_escala_0_25=0.25;
    valor_escala_0_75=0.75;
    valor_escala_0_4=0.4;
    valor_escala_0_5b=0.5;
end
y_no_desl=coordenadas(no_desl,3);
for t=1:NINCR
    P(t)=pressao_neutra(no,2*t)*-1;
end
nivel_agua=P/gama_agua;
Pmax=max(abs(nivel_agua));
XP0=coordenadas(no,2);

```

```

YP0=coordenadas(no,3);
n=1;
for i=1:NPOIN
    if coordenadas(i,3)==y_no_desl
        for t=1:NINCR
            desl_y(n,t)=deslocamentos(i,3*t);
            if hxy==1
                desl_x(n,t)=deslocamentos(i,3*t-1);
            end
            if vdy==1 || vdx==1 || vdx==1 || vty==1 || vtx==1
                if vdy==1 || vdx==1
                    if vdy==1
                        x1(n,t)=0;
                    end
                    if t==1
                        y1(n,t)=deslocamentos(i,3*t)-0;
                    else
                        y1(n,t)=deslocamentos(i,3*t)-deslocamentos(i,3*(t-1));
                    end
                end
                if vdx==1 || vdx==1
                    if vdx==1
                        y1(n,t)=0;
                    end
                    if t==1
                        x1(n,t)=deslocamentos(i,3*t-1)-0;
                    else
                        x1(n,t)=deslocamentos(i,3*t-1)-deslocamentos(i,3*(t-1)-1);
                    end
                end
                if hxy==1
                    x21(n,t)=coordenadas(i,2);
                    x22(n,t)=deslocamentos(i,3*t-1);
                else
                    x2(n,t)=coordenadas(i,2);
                end
                y21(n,t)=coordenadas(i,3);
                y22(n,t)=deslocamentos(i,3*t);
            end
            no_x(n,1:2)=[coordenadas(i,1) coordenadas(i,2)];
            max_de_y(n,1)=max(abs(desl_y(n,1:NINCR)));
            if hxy==1
                max_de_x(n,1)=max(abs(desl_x(n,1:NINCR)));
            end
            n=n+1;
        end
    end
    if vdy_t==1 || vdx_t==1 || vdx_t==1 || vty_t==1 || vtx_t==1
        for t=1:NINCR
            if vdy_t==1 || vdx_t==1
                if vdy_t==1
                    x1(i,t)=0;
                end
                if t==1
                    y1(i,t)=deslocamentos(i,3*t)-0;
                else
                    y1(i,t)=deslocamentos(i,3*t)-deslocamentos(i,3*(t-1));
                end
            end
            if vdx_t==1 || vdx_t==1
                if vdx_t==1
                    y1(i,t)=0;
                end
                if t==1
                    x1(i,t)=deslocamentos(i,3*t-1)-0;
                else
                    x1(i,t)=deslocamentos(i,3*t-1)-deslocamentos(i,3*(t-1)-1);
                end
            end
            if hxy==1
                x21(i,t)=coordenadas(i,2);
                x22(i,t)=deslocamentos(i,3*t-1);
            else
                x2(i,t)=coordenadas(i,2);
            end
            y21(i,t)=coordenadas(i,3);
        end
    end
end

```

```

        y22(i,t)=deslocamentos(i,3*t);
    end
end
end
tamanho_no_x=size(no_x);
if vty==1 || vtx==1 || vty_t==1 || vtx_t==1
    ty2=0;
    tx2=0;
    if ty==1
        ty=0;
        ty2=1;
    elseif tx==1
        tx=0;
        tx2=1;
    end
    if vty==1 || vtx==1
        for v=1:tamanho_no_x(1)
            no_tensoes=no_x(v,1);
            vtxy2=0;
            if vtx==1
                tensao();
                tx1(v,1:NINCR)=zc_x;
                if vtxy2==0
                    ty1(v,1:NINCR)=0;
                end
            end
            if vty==1
                tensao();
                ty1(v,1:NINCR)=zc;
                if vtxy2==0
                    tx1(v,1:NINCR)=0;
                end
            end
            max_tx1(v,1)=max(abs(tx1(v,1:NINCR)));
            max_ty1(v,1)=max(abs(ty1(v,1:NINCR)));
        end
    elseif vty_t==1 || vtx_t==1
        for v=1:NPOIN
            no_tensoes=v;
            vtxy2_t=0;
            if vtx_t==1
                tensao();
                tx1(v,1:NINCR)=zc_x;
                if vtxy2_t==0
                    ty1(v,1:NINCR)=0;
                end
            end
            if vty_t==1
                tensao();
                ty1(v,1:NINCR)=zc;
                if vtxy2_t==0
                    tx1(v,1:NINCR)=0;
                end
            end
            max_tx1(v,1)=max(abs(tx1(v,1:NINCR)));
            max_ty1(v,1)=max(abs(ty1(v,1:NINCR)));
        end
    end
    if ty2==1
        ty=1;
    elseif tx2==1
        tx=1;
    end
    save dados4 ty1 tx1
end
if vdy==0 && vdx==0 && vdx_y==0 && vdy_t==0 && vdx_t==0 && vdx_y_t==0 && vty==0 && vtx==0 && vty_t==0 && vtx_t==0
    desl_y_max=max(max_de_y);
    if hxy==1
        desl_x_max=max(max_de_x);
        desl_max=max([desl_y_max desl_x_max]);
    end
elseif vdy==1 || vdx==1 || vdx_y==1 || vdy_t==1 || vdx_t==1 || vdx_y_t==1 || vty==1 || vtx==1 || vty_t==1 || vtx_t==1
    if hxy==1
        desl_y_max=max(max_de_y);
        desl_x_max=max(max_de_x);
        desl_max=max([desl_y_max desl_x_max]);
    end
end

```



```

else
    desl_max=max(max_de_y);
end
if vty==1 || vtx==1 || vty_t==1 || vtx_t==1
    t_max=max([max_tx1;max_ty1]);
end
end
if vdy==1 || vdx==1 || vdx_t==1
    for t=1:NINCR
        desl_max_x1y1(1,t)=max(abs([x1(1:tamanho_no_x(1),t);y1(1:tamanho_no_x(1),t)]));
    end
elseif vdy_t==1 || vdx_t==1 || vdx_t==1
    for t=1:NINCR
        desl_max_x1y1(1,t)=max(abs([x1(1:NPOIN,t);y1(1:NPOIN,t)]));
    end
end
if vdy==1 || vdx==1 || vdx_t==1
    for t=1:NINCR
        if vdy==1 || vdx_t==1
            y1(1:tamanho_no_x(1),t)=y1(1:tamanho_no_x(1),t)*0.5*a/(desl_max_x1y1(1,t));
        end
        if vdx==1
            x1(1:tamanho_no_x(1),t)=x1(1:tamanho_no_x(1),t)*valor_escala_0_2*a/(desl_max_x1y1(1,t));
        end
        if vdx_t==1
            x1(1:tamanho_no_x(1),t)=x1(1:tamanho_no_x(1),t)*0.5*a/(desl_max_x1y1(1,t));
        end
        y2(1:tamanho_no_x(1),t)=y21(1:tamanho_no_x(1),t)+y22(1:tamanho_no_x(1),t)*valor_escala_0_5b*a/(desl_max);
        if hxy==1
            x2(1:tamanho_no_x(1),t)=x21(1:tamanho_no_x(1),t)+x22(1:tamanho_no_x(1),t)*valor_escala_0_5b*a/(desl_max);
        end
    end
elseif vdy_t==1 || vdx_t==1 || vdx_t==1
    if hxy==1
        for t=1:NINCR
            max_y22(1,t)=max(abs(y22(:,t)));
            max_x22(1,t)=max(abs(x22(:,t)));
        end
        max2_y22=max(max_y22);
        max2_x22=max(max_x22);
        desl_max=max([max2_y22 max2_x22]);
    else
        for t=1:NINCR
            max_y22(1,t)=max(abs(y22(:,t)));
        end
        max2_y22=max(max_y22);
        desl_max=max2_y22;
    end
    for t=1:NINCR
        if vdy_t==1 || vdx_t==1
            y1(1:NPOIN,t)=y1(1:NPOIN,t)*valor_escala_0_5a*a/(desl_max_x1y1(1,t));
        end
        if vdx_t==1
            x1(1:NPOIN,t)=x1(1:NPOIN,t)*valor_escala_0_3*a/(desl_max_x1y1(1,t));
        end
        if vdx_t==1
            x1(1:NPOIN,t)=x1(1:NPOIN,t)*valor_escala_0_5a*a/(desl_max_x1y1(1,t));
        end
        y2(1:NPOIN,t)=y21(1:NPOIN,t)+y22(1:NPOIN,t)*valor_escala_0_5b*a/(desl_max);
        if hxy==1
            x2(1:NPOIN,t)=x21(1:NPOIN,t)+x22(1:NPOIN,t)*valor_escala_0_5b*a/(desl_max);
        end
    end
elseif vty==1 || vtx==1
    if vty==1
        ty1(1:tamanho_no_x(1),1:NINCR)=ty1(1:tamanho_no_x(1),1:NINCR)*valor_escala_0_75*a/(t_max);
    end
    if vtx==1
        tx1(1:tamanho_no_x(1),1:NINCR)=tx1(1:tamanho_no_x(1),1:NINCR)*valor_escala_0_25*a/(t_max);
    end
end
y2(1:tamanho_no_x(1),1:NINCR)=y21(1:tamanho_no_x(1),1:NINCR)+y22(1:tamanho_no_x(1),1:NINCR)*valor_escala_0_5b*a/(desl_max);
if hxy==1

```

```

x2(1:tamanho_no_x(1),1:NINCR)=x21(1:tamanho_no_x(1),1:NINCR)+x22(1:tamanho_no_x(1),1:NINCR)*valor_escala_0_5b*a/(desl_max
);
end
elseif vty_t==1 || vtx_t==1
if hxy==1
for t=1:NINCR
max_y22(1,t)=max(abs(y22(:,t)));
max_x22(1,t)=max(abs(x22(:,t)));
end
max2_y22=max(max_y22);
max2_x22=max(max_x22);
desl_max=max([max2_y22 max2_x22]);
else
for t=1:NINCR
max_y22(1,t)=max(abs(y22(:,t)));
end
max2_y22=max(max_y22);
desl_max=max2_y22;
end
if vty_t==1
ty1(1:NPOIN,1:NINCR)=ty1(1:NPOIN,1:NINCR)*valor_escala_0_4*a/(t_max);
end
if vtx_t==1
tx1(1:NPOIN,1:NINCR)=tx1(1:NPOIN,1:NINCR)*valor_escala_0_25*a/(t_max);
end
y2(1:NPOIN,1:NINCR)=y21(1:NPOIN,1:NINCR)+y22(1:NPOIN,1:NINCR)*valor_escala_0_5b*a/(desl_max);
if hxy==1
x2(1:NPOIN,1:NINCR)=x21(1:NPOIN,1:NINCR)+x22(1:NPOIN,1:NINCR)*valor_escala_0_5b*a/(desl_max);
end
end
if exist('tensoes1','var')==1
if ty==1 && vtx==0 && vty_t==0 && itX==0 && itXY==0 || tx==1 && vtx==0 && vty_t==0 && itX==0 && itXY==0
if vty==1 || vty_t==1
no_tensoes=no;
end
tensao();
if ty==1
if itY==1
zc2=zc;
end
elseif tx==1
if itY==1
zc2=zc_x;
end
end
elseif ty==1 && vtx==1 || tx==1 && vtx==1
no_tensoes=no;
vtx=0;
tensao();
vtx=1;
elseif ty==1 && vty_t==1 || tx==1 && vty_t==1
no_tensoes=no;
vty_t=0;
tensao();
vty_t=1;
elseif ty==1 && itX==1 || tx==1 && itX==1
itX=0;
tensao();
itX=1;
if tx==1
zc2=zc_x;
end
elseif ty==1 && itXY==1 || tx==1 && itXY==1 || txy==1 && itXY==1
itXY=0;
tensao();
itXY=1;
if txy==1
zc2=zc_xy;
end
end
if ty==1
total=P*-1+zc;
min_zc=min([min(zc) min(P*-1) min(total)]);
max_zc=max([max(zc) max(P*-1) max(total)]);
if min_zc>0

```

```

        min_zc=0;
    end
    if max_zc<0
        max_zc=0;
    end
elseif tx==1
    total=P*-1+zc_x;
    min_zc_x=min([min(zc_x) min(P*-1) min(total)]);
    max_zc_x=max([max(zc_x) max(P*-1) max(total)]);
    if min_zc_x>0
        min_zc_x=0;
    end
    if max_zc_x<0
        max_zc_x=0;
    end
elseif txy==1
    if min(zc_xy)>=0
        min_zc_xy=0;
    else
        min_zc_xy=min(zc_xy);
    end
    if max(zc_xy)<=0
        max_zc_xy=0;
    else
        max_zc_xy=max(zc_xy);
    end
end
end
for t1=1:NINCR
    if t1==1
        if exist('tensoes1','var')==1
            axes(handles.axes3)
        else
            axes(handles.axes1)
        end
        if t1==1
            hold off
        end
        caixa();
        axis([Xmin-1*a Xmax+1*a Ymin-1*a Ymax+Ymax/3+1*a])
        hold('all')
        plot([XP0,XP0],[YP0,Ymax],'linewidth',7);
        plot([no_x(1:tamanho_no_x(1),2),no_x(1:tamanho_no_x(1),2)],[y_no_desl,y_no_desl],'-k','linewidth',1);
        if exist('tensoes1','var')==1
            if vdy==1 && ty==1 || vdy_t==1 && ty==1
                if hxy==1
                    title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
                else
                    title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
                end
            elseif vdy==1 && tx==1 || vdy_t==1 && tx==1
                if hxy==1
                    title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
                else
                    title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
                end
            elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
                if hxy==1
                    title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
                else
                    title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
                end
            elseif vdx==1 && tx==1 || vdx_t==1 && tx==1
                if hxy==1
                    title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
                else
                    title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
                end
            end
        end
    end
end

```



```

else
    title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas do deslocamento horizontal; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
end
elseif itY==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão vertical efetiva; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão vertical efetiva; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itY==1 && tx==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão vertical efetiva; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão vertical efetiva; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itX==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão
vertical efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itX==1 && tx==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itXY==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
vertical efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
vertical efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itXY==1 && tx==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
    end
elseif itXY==1 && txy==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão de
corte ( $\tau_{xy}$ )','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão de
corte ( $\tau_{xy}$ )','FontWeight','bold')
    end
elseif ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água e deslocamento total; Gráfico2: Tensão vertical efetiva e total e pressão
neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água e deslocamento vertical; Gráfico2: Tensão vertical efetiva e total e pressão
neutra','FontWeight','bold')
    end
elseif tx==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água e deslocamento total; Gráfico2: Tensão horizontal efetiva e total e pressão
neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água e deslocamento vertical; Gráfico2: Tensão horizontal efetiva e total e pressão
neutra','FontWeight','bold')
    end
end

```

```

end
else
if vdy==1 && ty==1 || vdy_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical','FontWeight','bold')
end
elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal','FontWeight','bold')
end
elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento total','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento total','FontWeight','bold')
end
elseif idY==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas do deslocamento vertical','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas do deslocamento vertical','FontWeight','bold')
end
elseif idX==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas do deslocamento horizontal','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas do deslocamento horizontal','FontWeight','bold')
end
elseif ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água e deslocamento total','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água e deslocamento vertical','FontWeight','bold')
end
end
end
xlabel('Distância ao eixo y (m)')
ylabel('Distância ao eixo x (m)')
hold off
if exist('tensoes1','var')==1
axes(handles.axes4)
if escala_linear==1
set(gca,'xscale','linear');
else
set(gca,'xscale','log');
end
if t1==1
hold off
end
if eixo_tempo==1
tempo_usado=tempo;
else
tempo_usado=tempo_mod;
end
if ty==1
axis([0 max(tempo_usado) min_zc max_zc])
if abs(min_zc)>abs(max_zc)
set(handles.axes4,'XAxisLocation','top');
else
set(handles.axes4,'XAxisLocation','bottom');
end
elseif tx==1
axis([0 max(tempo_usado) min_zc_x max_zc_x])
if abs(min_zc_x)>abs(max_zc_x)
set(handles.axes4,'XAxisLocation','top');
else
set(handles.axes4,'XAxisLocation','bottom');
end
elseif txy==1
axis([0 max(tempo_usado) min_zc_xy max_zc_xy])
if abs(min_zc_xy)>abs(max_zc_xy)
set(handles.axes4,'XAxisLocation','top');

```

```

else
    set(handles.axes4,'XAxisLocation','bottom');
end
end
hold('all');
if eixo_tempo==1
    xlabel('Tempo (s)')
else
    xlabel('Fator T` (s)')
end
ylabel('kPa')
pause(0.1)
if idY==1 || idX==1 || itY==1 || itX==1 || itXY==1
    for t2=1:NINCR
        tx2=0;
        if tx==1
            tx2=1;
            tx=0;
        end
        isocurvas();
        tamanho_XYZ=size(X);
        X2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=X;
        Y2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=Y;
        Z2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=Z;
        if tx2==1
            tx=1;
        end
    end
else
    pause(5)
end
else
    pause(0.1)
    if idY==1 || idX==1
        for t2=1:NINCR
            tx2=0;
            if tx==1
                tx2=1;
                tx=0;
            end
            isocurvas();
            tamanho_XYZ=size(X);
            X2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=X;
            Y2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=Y;
            Z2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t2-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t2)=Z;
            if tx2==1
                tx=1;
            end
        end
    else
        pause(5)
    end
end
end
if exist('tensoes1','var')==1
    axes(handles.axes3)
else
    axes(handles.axes1)
end
caixa();
axis([Xmin-1*a Xmax+1*a Ymin-1*a Ymax+Ymax/3+1*a])
hold('all')
if idY==1 || idX==1 || itY==1 || itX==1 || itXY==1
    contour(X2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t1-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t1),Y2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t1-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t1),Z2(1:tamanho_XYZ(1),tamanho_XYZ(2)*t1-(tamanho_XYZ(2)-1):tamanho_XYZ(2)*t1))
    colorbar('location','EastOutside')
end
plot([XP0,XP0],[YP0,Ymax+nivel_agua(t1)*(Ymax/3)/Pmax],'linewidth',7);
if hxy==1
    if vdx==1 || vdx_t==1 || vtx==1 || vtx_t==1
        plot([no_x(1:tamanho_no_x(1),2)+desl_x(1:tamanho_no_x(1),1)*valor_escal_a_0_5b*a/(desl_max),no_x(1:tamanho_no_x(1),2)+desl_x(1:tamanho_no_x(1),1)*valor_escal_a_0_5b*a/(desl_max)],[y_no_desl+desl_y(1:tamanho_no_x(1),1)*valor_escal_a_0_5b*a/(desl_max),y_no_desl+desl_y(1:tamanho_no_x(1),1)*valor_escal_a_0_5b*a/(desl_max)],'k','linewidth',1);
    else

```

```

plot([no_x(1:tamanho_no_x(1),2)+desl_x(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max),no_x(1:tamanho_no_x(1),2)+desl_x(1:taman
ho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)], [y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max),y_no_de
sl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)], '-k.', 'linewidth', 1);
end
elseif vdy==1 || vdx==1 || vdy_t==1 || vdx_t==1 || vty==1 || vty_t==1

plot([no_x(1:tamanho_no_x(1),2),no_x(1:tamanho_no_x(1),2)], [y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)
,y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)], '-k.', 'linewidth', 1);
elseif vdx==1 || vdx_t==1 || vtx==1 || vtx_t==1

plot([no_x(1:tamanho_no_x(1),2),no_x(1:tamanho_no_x(1),2)], [y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)
,y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_max)], 'k.', 'linewidth', 1);
else

plot([no_x(1:tamanho_no_x(1),2),no_x(1:tamanho_no_x(1),2)], [y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_y_m
ax),y_no_desl+desl_y(1:tamanho_no_x(1),t1)*valor_escal_a_0_5b*a/(desl_y_max)], '-k.', 'linewidth', 1);
end
if vdy==1 || vdx==1 || vdx==1
quiver(x2(1:tamanho_no_x(1),t1),y2(1:tamanho_no_x(1),t1),x1(1:tamanho_no_x(1),t1),y1(1:tamanho_no_x(1),t1),0,'b');
elseif vdy_t==1 || vdx_t==1 || vdx==1
quiver(x2(1:NPOIN,t1),y2(1:NPOIN,t1),x1(1:NPOIN,t1),y1(1:NPOIN,t1),0,'b');
elseif vty==1 || vtx==1
quiver(x2(1:tamanho_no_x(1),t1),y2(1:tamanho_no_x(1),t1),tx1(1:tamanho_no_x(1),t1),ty1(1:tamanho_no_x(1),t1),0,'b');
elseif vty_t==1 || vtx_t==1
quiver(x2(1:NPOIN,t1),y2(1:NPOIN,t1),tx1(1:NPOIN,t1),ty1(1:NPOIN,t1),0,'b');
end
hold off
if exist('tensoes1','var')==1
if vdy==1 && ty==1 || vdy_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical; Gráfico2: Tensão vertical efetiva e
total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
end
elseif vdy==1 && tx==1 || vdy_t==1 && tx==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
end
elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
end
elseif vdx==1 && tx==1 || vdx_t==1 && tx==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
end
elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento total; Gráfico2: Tensão vertical efetiva e
total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento total; Gráfico2: Tensão vertical efetiva e
total e pressão neutra','FontWeight','bold')
end
elseif vdx==1 && tx==1 || vdx_t==1 && tx==1
if hxy==1
title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento total; Gráfico2: Tensão horizontal efetiva e
total e pressão neutra','FontWeight','bold')
else
title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento total; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
end

```





```

        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão vertical efetiva; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão vertical efetiva; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
    end
    elseif itX==1 && ty==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
        end
    elseif itX==1 && tx==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão horizontal
efetiva e total e pressão neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão horizontal efetiva; Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
        end
    elseif itXY==1 && ty==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão vertical
efetiva e total e pressão neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
vertical efetiva e total e pressão neutra','FontWeight','bold')
        end
    elseif itXY==1 && tx==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão
horizontal efetiva e total e pressão neutra','FontWeight','bold')
        end
    elseif itXY==1 && txy==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão de corte
( $\tau_{xy}$ ),'FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas da tensão de corte ( $\tau_{xy}$ ); Gráfico2: Tensão de
corte ( $\tau_{xy}$ ),'FontWeight','bold')
        end
    elseif ty==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água e deslocamento total; Gráfico2: Tensão vertical efetiva e total e pressão
neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água e deslocamento vertical; Gráfico2: Tensão vertical efetiva e total e pressão
neutra','FontWeight','bold')
        end
    elseif tx==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água e deslocamento total; Gráfico2: Tensão horizontal efetiva e total e pressão
neutra','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água e deslocamento vertical; Gráfico2: Tensão horizontal efetiva e total e pressão
neutra','FontWeight','bold')
        end
    end
else
    if vdy==1 && ty==1 || vdy_t==1 && ty==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento vertical','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento vertical','FontWeight','bold')
        end
    elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
        if hxy==1
            title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento horizontal','FontWeight','bold')
        else
            title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento horizontal','FontWeight','bold')
        end
    end
end

```

```

elseif vdx==1 && ty==1 || vdx_t==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e vetores do deslocamento total','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e vetores do deslocamento total','FontWeight','bold')
    end
elseif idY==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas do deslocamento vertical','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas do deslocamento vertical','FontWeight','bold')
    end
elseif idX==1 && ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água, deslocamento total e isocurvas do deslocamento horizontal','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água, deslocamento vertical e isocurvas do deslocamento horizontal','FontWeight','bold')
    end
elseif ty==1
    if hxy==1
        title('Gráfico1: Altura da coluna de água e deslocamento total','FontWeight','bold')
    else
        title('Gráfico1: Altura da coluna de água e deslocamento vertical','FontWeight','bold')
    end
end
end
xlabel('Distância ao eixo y (m)')
ylabel('Distância ao eixo x (m)')
if exist('tensoes1','var')==1
    axes(handles.axes4)
    if t1==1
        hold off
    end
    if ty==1
        axis([0 max(tempo_usado) min_zc max_zc])
        if abs(min_zc)>abs(max_zc)
            set(handles.axes4,'XAxisLocation','top');
        else
            set(handles.axes4,'XAxisLocation','bottom');
        end
    elseif tx==1
        axis([0 max(tempo_usado) min_zc_x max_zc_x])
        if abs(min_zc_x)>abs(max_zc_x)
            set(handles.axes4,'XAxisLocation','top');
        else
            set(handles.axes4,'XAxisLocation','bottom');
        end
    elseif txy==1
        axis([0 max(tempo_usado) min_zc_xy max_zc_xy])
        if abs(min_zc_xy)>abs(max_zc_xy)
            set(handles.axes4,'XAxisLocation','top');
        else
            set(handles.axes4,'XAxisLocation','bottom');
        end
    end
end
hold('all');
if incr_tempo_tensoes(t1,2)==1
    if itY==1 || tx==1 && itX==1 || txy==1 && itXY==1
        plot(tempo_usado(t1),zc2(t1),'r*');
        if min(incr_tempo_tensoes(1:NINCR,2))==1
            if t1>1
                plot([tempo_usado(t1-1),tempo_usado(t1)],[zc2(t1-1),zc2(t1)],'r--');
            end
        end
    else
        if ty==1
            plot(tempo_usado(t1),zc(t1),'r*');
            if min(incr_tempo_tensoes(1:NINCR,2))==1
                if t1>1
                    plot([tempo_usado(t1-1),tempo_usado(t1)],[zc(t1-1),zc(t1)],'r--');
                end
            end
        elseif tx==1
            plot(tempo_usado(t1),zc_x(t1),'r*');
            if min(incr_tempo_tensoes(1:NINCR,2))==1
                if t1>1

```

```

        plot([tempo_usado(t1-1),tempo_usado(t1)],[zc_x(t1-1),zc_x(t1)],'r--');
    end
end
end
end
end
if ty==1 || tx==1
    plot(tempo_usado(t1),P(t1)*-1,'b*');
    if incr_tempo_tensoes(t1,2)==1
        plot(tempo_usado(t1),total(t1),'k*');
    end
    if t1>1
        plot([tempo_usado(t1-1),tempo_usado(t1)],[P(t1-1)*-1,P(t1)*-1],'b--');
        if min(incr_tempo_tensoes(1:NINCR,2))==1
            plot([tempo_usado(t1-1),tempo_usado(t1)],[total(t1-1),total(t1)],'k--');
        end
    end
end
end
if ty==1
    legend('Tensão vertical efetiva','Pressão neutra','Tensão vertical total','Location','EastOutside')
elseif tx==1
    legend('Tensão horizontal efetiva','Pressão neutra','Tensão horizontal total','Location','EastOutside')
elseif txy==1
    legend('Tensão de corte ( $\tau_{xy}$ )','Location','EastOutside')
end
if eixo_tempo==1
    xlabel('Tempo (s)')
else
    xlabel('Fator T (s)')
end
ylabel('kPa')
hold on
end
if exist('tensoes1','var')==1
    if hxy==1
        if vdy_t==1 || vdx_t==1 || vdx_t==1
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc y22 x22
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x y22 x22
            end
        elseif vty==1 || vtx==1
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x
            end
        elseif vty_t==1 || vtx_t==1
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc y22 x22
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x y22 x22
            end
        elseif itY==1
            save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc2
        elseif itX==1
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc2
            end
        elseif itXY==1
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
            elseif tx==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x zc2
            end
        else
            if ty==1
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc
            else
                save dados2 no P nivel_agua desl_y no_x t1 desl_x total zc_x
            end
        end
    end
end
end

```



### **function tensao**

```
x_y_no=[coordenadas(no_tensoes,2) coordenadas(no_tensoes,3)];
elementos_no=0;
n=1;
for i=1:NELEM
    for j=4:4+elementos(i,3)-1
        if elementos(i,j)==no_tensoes
            elementos_no(n,1:elementos(i,3)+1)=[elementos(i,1) elementos(i,4:4+elementos(i,3)-1)];
            n=n+1;
        end
    end
end
elementos_no=elementos_no';
tamanho_elementos_no=size(elementos_no);
tamanho=tamanho_elementos_no(2)*4;
if vtx==1 || tx==1 || vtx_t==1 || itX==1
    x_y_tensoes=0;
    valor_tensao_x=0;
    for i=1:tamanho_elementos_no(2)
        x_y_tensoes(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,1:2)=[tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),1) tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),2)];
        for t=1:NINCR
            valor_tensao_x(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,t)=tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),6*t-3)*-1;
        end
    end
end
if txy==1 || itXY==1
    x_y_tensoes=0;
    valor_tensao_y=0;
    for i=1:tamanho_elementos_no(2)
        x_y_tensoes(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,1:2)=[tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),1) tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),2)];
        for t=1:NINCR
            valor_tensao_xy(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,t)=tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),6*t-1)*-1;
        end
    end
end
if vty==1 || ty==1 || vty_t==1 || itY==1
    x_y_tensoes=0;
    valor_tensao_y=0;
    for i=1:tamanho_elementos_no(2)
        x_y_tensoes(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,1:2)=[tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),1) tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),2)];
        for t=1:NINCR
            valor_tensao_y(aux_NGAUS*i-(aux_NGAUS-1):aux_NGAUS*i,t)=tensoes1(aux_NGAUS*elementos_no(1,i)-(aux_NGAUS-1):aux_NGAUS*elementos_no(1,i),6*t-2)*-1;
        end
    end
end
valor_tensao_no();
```

### **function valor\_tensao\_no**

```
xi=0;
yi=0;
z_i=0;
xi(1,1:tamanho)=x_y_tensoes(1:tamanho,1);
yi(1,1:tamanho)=x_y_tensoes(1:tamanho,2);
if vtx==1 || tx==1 || vtx_t==1 || itX==1
    z_i=valor_tensao_x'*-1;
elseif txy==1 || itXY==1
    z_i=valor_tensao_xy'*-1;
elseif vty==1 || ty==1 || vty_t==1 || itY==1
    z_i=valor_tensao_y'*-1;
end
tamanho_xi=size(xi);
num_colunas_xi=tamanho_xi(2);
xc=x_y_no(1,1);
yc=x_y_no(1,2);
for t=1:NINCR
    zi=z_i(t,1:tamanho);
    soma_zi=sum(zi);
```

```

soma_xi=sum(xi);
soma_yi=sum(yi);
soma_xi_zi=0.0;
for i=1:num_colunas_xi
    soma_xi_zi=soma_xi_zi+xi(i)*zi(i);
end
soma_yi_zi=0.0;
for i=1:num_colunas_xi
    soma_yi_zi=soma_yi_zi+yi(i)*zi(i);
end
soma_xi_xi=0.0;
for i=1:num_colunas_xi
    soma_xi_xi=soma_xi_xi+xi(i)*xi(i);
end
soma_xi_yi=0.0;
for i=1:num_colunas_xi
    soma_xi_yi=soma_xi_yi+xi(i)*yi(i);
end
soma_yi_yi=0.0;
for i=1:num_colunas_xi
    soma_yi_yi=soma_yi_yi+yi(i)*yi(i);
end
A(1,1)=num_colunas_xi;
A(1,2)=soma_xi;
A(1,3)=soma_yi;
A(2,1)=soma_xi;
A(2,2)=soma_xi_xi;
A(2,3)=soma_xi_yi;
A(3,1)=soma_yi;
A(3,2)=soma_xi_yi;
A(3,3)=soma_yi_yi;
b(1)=soma_zi;
b(2)=soma_xi_zi;
b(3)=soma_yi_zi;
x=A\b';
x(1);
x(2);
x(3);
xi;
yi;
zi;
if vtx==1 || tx==1 || vtx_t==1 || itX==1
    zc_x(t)=x(1)+x(2)*xc+x(3)*yc;
elseif txy==1 || itXY==1
    zc_xy(t)=x(1)+x(2)*xc+x(3)*yc;
elseif vty==1 || ty==1 || vty_t==1 || itY==1
    zc(t)=x(1)+x(2)*xc+x(3)*yc;
end
end
end

```

### **function isocurvas**

```

if idY==1 || idX==1
    t=t2;
elseif itX==1 || itXY==1 || itY==1
    t3=t2;
end
for i=1:NPOIN
    x_nos(1,i)=coordenadas(i,2);
    y_nos(1,i)=coordenadas(i,3);
end
x_nos=sort(x_nos);
y_nos=sort(y_nos);
m=2;
n=2;
x_nos2(1,1)=x_nos(1,1);
y_nos2(1,1)=y_nos(1,1);
for i=2:NPOIN
    if x_nos(1,i)~=x_nos(1,i-1)
        x_nos2(1,m)=x_nos(1,i);
        m=m+1;
    end
    if y_nos(1,i)~=y_nos(1,i-1)
        y_nos2(1,n)=y_nos(1,i);
        n=n+1;
    end
end

```

```

end
end
tamanho_x_nos2=size(x_nos2);
tamanho_y_nos2=size(y_nos2);
for i=1:tamanho_x_nos2(2)
    for j=1:tamanho_y_nos2(2)
        for n=1:NPOIN
            if coordenadas(n,2)==x_nos2(1,i)
                if coordenadas(n,3)==y_nos2(1,j)
                    malha(j,i)=coordenadas(n,1);
                end
            end
        end
    end
end
end
if idY==1 || idX==1
    tamanho_malha=size(malha);
    for i=1:tamanho_malha(1)
        for j=1:tamanho_malha(2)
            if malha(i,j)~=0
                x_nos3(i,j)=coordenadas(malha(i,j),2);
                y_nos3(i,j)=coordenadas(malha(i,j),3);
                if idY==1
                    idy(i,j)=deslocamentos(malha(i,j),3*t);
                elseif idX==1
                    idx(i,j)=deslocamentos(malha(i,j),3*t-1);
                end
            end
        end
    end
end
elseif itX==1 || itXY==1 || itY==1
    tamanho_malha=size(malha);
    for z=1:tamanho_malha(1)
        for w=1:tamanho_malha(2)
            if malha(z,w)~=0
                x_nos3(z,w)=coordenadas(malha(z,w),2);
                y_nos3(z,w)=coordenadas(malha(z,w),3);
                no_tensoes=malha(z,w);
                if itX==1
                    tensao()
                    itx(z,w)=zc_x(1,t3);
                elseif itXY==1
                    tensao()
                    itxy(z,w)=zc_xy(1,t3);
                elseif itY==1
                    tensao()
                    ity(z,w)=zc(1,t3);
                end
            end
        end
    end
end
end
[X,Y]=meshgrid(x_nos2,y_nos2);
if idY==1
    Z=griddata(x_nos3,y_nos3,idy,X,Y,'linear');
elseif idX==1
    Z=griddata(x_nos3,y_nos3,idx,X,Y,'linear');
elseif itX==1
    Z=griddata(x_nos3,y_nos3,itx,X,Y,'linear');
elseif itXY==1
    Z=griddata(x_nos3,y_nos3,itxy,X,Y,'linear');
elseif itY==1
    Z=griddata(x_nos3,y_nos3,ity,X,Y,'linear');
end
end

```

### **function malha**

```

x_min2=0;y_min2=0;x_max2=0;y_max2=0;
for n=1:NELEM
    plot_no(1:4,1:2)=99999;
    nos_elem=elementos(n,4:3+MNODE);
    for i=1:MNODE
        x_y_nos_elem(i,1:NDIME)=[coordenadas(nos_elem(1,i),2) coordenadas(nos_elem(1,i),3)];
    end
    x_min=min(x_y_nos_elem(1:MNODE,1));

```



```

x_max=max(x_y_nos_elem(1:MNODE,1));
y_min=min(x_y_nos_elem(1:MNODE,2));
y_max=max(x_y_nos_elem(1:MNODE,2));
for i=1:MNODE
  if x_y_nos_elem(i,1)==x_min
    if x_y_nos_elem(i,2)==y_max
      plot_no(1,1:NDIME)=x_y_nos_elem(i,1:NDIME);
    else
      if x_y_nos_elem(i,2)==y_min
        plot_no(2,1:NDIME)=x_y_nos_elem(i,1:NDIME);
      else
        if plot_no(1,1:NDIME)==99999
          plot_no(1,1:NDIME)=x_y_nos_elem(i,1:NDIME);
        else
          if x_y_nos_elem(i,2)>plot_no(1,2)
            plot_no(1,1:NDIME)=x_y_nos_elem(i,1:NDIME);
          end
        end
        if plot_no(2,1:NDIME)==99999
          plot_no(2,1:NDIME)=x_y_nos_elem(i,1:NDIME);
        else
          if x_y_nos_elem(i,2)<plot_no(2,2)
            plot_no(2,1:NDIME)=x_y_nos_elem(i,1:NDIME);
          end
        end
      end
    end
  end
end
else
  if x_y_nos_elem(i,1)==x_max
    if x_y_nos_elem(i,2)==y_max
      plot_no(4,1:NDIME)=x_y_nos_elem(i,1:NDIME);
    else
      if x_y_nos_elem(i,2)==y_min
        plot_no(3,1:NDIME)=x_y_nos_elem(i,1:NDIME);
      else
        if plot_no(3,1:NDIME)==99999
          plot_no(3,1:NDIME)=x_y_nos_elem(i,1:NDIME);
        else
          if x_y_nos_elem(i,2)<plot_no(3,2)
            plot_no(3,1:NDIME)=x_y_nos_elem(i,1:NDIME);
          end
        end
      end
    end
  end
end
else
  if x_y_nos_elem(i,2)==y_max
    if plot_no(1,1:NDIME)==99999
      plot_no(1,1:NDIME)=x_y_nos_elem(i,1:NDIME);
    else
      if x_y_nos_elem(i,1)<plot_no(1,1)
        plot_no(1,1:NDIME)=x_y_nos_elem(i,1:NDIME);
      end
    end
    if plot_no(4,1:NDIME)==99999
      plot_no(4,1:NDIME)=x_y_nos_elem(i,1:NDIME);
    else
      if x_y_nos_elem(i,1)>plot_no(4,1)
        plot_no(4,1:NDIME)=x_y_nos_elem(i,1:NDIME);
      end
    end
  end
end
else
  if x_y_nos_elem(i,2)==y_min
    if plot_no(3,1:NDIME)==99999
      plot_no(3,1:NDIME)=x_y_nos_elem(i,1:NDIME);
    else
      if x_y_nos_elem(i,1)>plot_no(1,1)
        plot_no(3,1:NDIME)=x_y_nos_elem(i,1:NDIME);
      end
    end
  end
end
end
end
end
end
end
x_min2=min([x_min2 x_min]);
y_min2=min([y_min2 y_min]);

```

```

x_max2=max([x_max2 x_max]);
y_max2=max([y_max2 y_max]);
plot_no2(1:4,2*n-1:2*n)=plot_no;
end
plot(x_min2-1*a,y_min2-1*a,'w');
hold on
plot(x_max2+1*a,y_max2+1*a,'w');
hold on
text(x_min2,y_min2,' \rightarrow', 'FontSize',20,'HorizontalAlignment','left','VerticalAlignment','middle');
text(x_min2,y_min2,'x', 'FontSize',11,'HorizontalAlignment','left','VerticalAlignment','cap');
text(x_min2,y_min2,' \uparrow', 'FontSize',20,'HorizontalAlignment','right','VerticalAlignment','baseline');
text(x_min2,y_min2,'y', 'FontSize',11,'HorizontalAlignment','right','VerticalAlignment','baseline');
for n=1:NELEM
    plot([plot_no2(1,2*n-1),plot_no2(2,2*n-1)],[plot_no2(1,2*n),plot_no2(2,2*n)],'r-');
    hold on
    plot([plot_no2(2,2*n-1),plot_no2(3,2*n-1)],[plot_no2(2,2*n),plot_no2(3,2*n)],'r-');
    hold on
    plot([plot_no2(3,2*n-1),plot_no2(4,2*n-1)],[plot_no2(3,2*n),plot_no2(4,2*n)],'r-');
    hold on
    plot([plot_no2(4,2*n-1),plot_no2(1,2*n-1)],[plot_no2(4,2*n),plot_no2(1,2*n)],'r-');
    hold on
end
if IEDGE==1
    numero_nos=size(valor_carga);
    for i=1:NEDGE
        plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)],'b-');
        hold on
        plot([Xmin_carga(i,1),Xmin_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.5*a], 'b-');
        hold on
        plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1)+0.5*a,Ymin_carga(i,1)+0.5*a], 'b-');
        hold on
        plot([Xmax_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.5*a], 'b-');
        if max(valor_carga(2*i-1,1:numero_nos(2)))>0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),
\downarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
        elseif min(valor_carga(2*i-1,1:numero_nos(2)))<0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),
\uparrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
        end
        if max(valor_carga(2*i,1:numero_nos(2)))>0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),
\leftarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        elseif min(valor_carga(2*i,1:numero_nos(2)))<0
            text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),
\rightarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        end
    end
end
if IPLOD==1
    numero_cargas_pontuais=size(cargas_pontuais);
    for i=1:numero_cargas_pontuais
        if cargas_pontuais(i,3)~=0
            if cargas_pontuais(i,3)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\downarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\uparrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
            end
        end
        if cargas_pontuais(i,2)~=0
            if cargas_pontuais(i,2)<0
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\leftarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            else
                text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\rightarrow', 'FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
            end
        end
    end
end
elseif IPLOD==1
    numero_cargas_pontuais=size(cargas_pontuais);
    for i=1:numero_cargas_pontuais
        if cargas_pontuais(i,3)~=0
            if cargas_pontuais(i,3)<0

```

```

        text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
    else
        text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\uparrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
    end
    if cargas_pontuais(i,2)~=0
        if cargas_pontuais(i,2)<0
            text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\leftarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        else
            text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\rightarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','bottom');
        end
    end
end
end
end

```

### **function desl\_x\_v\_malha**

```

save dados3 Popupmenu1 vty vtx
load dados titulo_problema NINCR NPOIN deslocamentos NELEM MNODE elementos coordenadas NDIME valor_carga NEDGE
Xmin_carga Xmax_carga Ymin_carga Ymax_carga aux_NGAUS cargas_pontuais a
if strcmp(titulo_problema,'unidimensional')
    valor_escala=0.15*a;
    valor_escala_0_5=0.2;
    valor_escala_0_4a=0.075;
    valor_escala_0_4b=0.2;
    valor_escala_0_25=0.075;
elseif strcmp(titulo_problema,'bidimensional ')
    valor_escala=0.25*a;
    valor_escala_0_5=0.5;
    valor_escala_0_4a=0.4;
    valor_escala_0_4b=0.4;
    valor_escala_0_25=0.25;
end
for t=1:NINCR
    x_max_t(1,t)=max(deslocamentos(1:NPOIN,3*t-1));
    x_min_t(1,t)=min(deslocamentos(1:NPOIN,3*t-1));
    y_max_t(1,t)=max(deslocamentos(1:NPOIN,3*t));
    y_min_t(1,t)=min(deslocamentos(1:NPOIN,3*t));
    if vdy==1 || vdx==1 || vdxv==1
        if vdy==1 || vdxv==1
            if t==1
                desl_max_vdy(1,t)=max(abs(deslocamentos(1:NPOIN,3*t)-0));
            else
                desl_max_vdy(1,t)=max(abs(deslocamentos(1:NPOIN,3*t)-deslocamentos(1:NPOIN,3*(t-1))));
            end
        end
        if vdx==1 || vdxv==1
            if t==1
                desl_max_vdx(1,t)=max(abs(deslocamentos(1:NPOIN,3*t)-0));
            else
                desl_max_vdx(1,t)=max(abs(deslocamentos(1:NPOIN,3*t)-deslocamentos(1:NPOIN,3*(t-1)-1)));
            end
        end
        if vdxv==1
            desl_max_vdxv(1,t)=max([desl_max_vdy(1,t) desl_max_vdx(1,t)]);
        end
    end
end
x_max_total=abs(max(x_max_t));
x_min_total=abs(min(x_min_t));
y_max_total=abs(max(y_max_t));
y_min_total=abs(min(y_min_t));
desl_max=max([x_max_total x_min_total y_max_total y_min_total]);
if vty==1 || vtx==1
    for v=1:NPOIN
        no_tensoes=v;
        if vtx==1
            tensao();
            tx1(v,1:NINCR)=zc_x;
            max_tx1(v,1)=max(abs(tx1(v,1:NINCR)));
        end
    end
end

```

```

    if vty==1
        tensao();
        ty1(v,1:NINCR)=zc;
        max_ty1(v,1)=max(abs(ty1(v,1:NINCR)));
    end
end
if vty==1
    t_max=max(max_ty1);
elseif vtx==1
    t_max=max(max_tx1);
end
end
x_min2=0;y_min2=0;x_max2=0;y_max2=0;
for t=1:NINCR
    if vdx==1 || vdx==1 || vdy==1 || vtx==1 || vty==1
        nos(1,1:NPOIN)=0;
    end
    for n=1:NELEM
        if vdx==1 || vdx==1 || vdy==1 || vtx==1 || vty==1
            m=1;
            x2(1,1:MNODE)=0;
            y2(1,1:MNODE)=0;
            if vdx==1 || vdx==1 || vdy==1
                x1(1,1:MNODE)=0;
                y1(1,1:MNODE)=0;
            elseif vtx==1 || vty==1
                t_x1(1,1:MNODE)=0;
                t_y1(1,1:MNODE)=0;
            end
        end
        plot_no(1:4,1:2)=99999;
        nos_elem=elementos(n,4:3+MNODE);
        for i=1:MNODE
            x_y_nos_elem(i,1:NDIME)=[coordenadas(nos_elem(1,i),2) coordenadas(nos_elem(1,i),3)];
            x_y_nos_elem_tempo(i,1:NDIME)=[coordenadas(nos_elem(1,i),2)-deslocamentos(nos_elem(1,i),3*t-1)*valor_escala/desl_max
            coordenadas(nos_elem(1,i),3)+deslocamentos(nos_elem(1,i),3*t)*valor_escala/desl_max];
            if vdy==1 || vdx==1 || vdx==1 || vty==1 || vtx==1
                if nos(1,coordenadas(nos_elem(1,i),1))=coordenadas(nos_elem(1,i),1)
                    w=1;
                else
                    nos(1,coordenadas(nos_elem(1,i),1))=coordenadas(nos_elem(1,i),1);
                    if vdy==1
                        if t==1
                            y1(m)=(deslocamentos(nos_elem(1,i),3*t)-0)*valor_escala_0_5*a/desl_max_vdy(1,t);
                        else
                            y1(m)=(deslocamentos(nos_elem(1,i),3*t)-deslocamentos(nos_elem(1,i),3*(t-1)))*valor_escala_0_5*a/desl_max_vdy(1,t);
                        end
                        x1(m)=0;
                    elseif vdx==1
                        if t==1
                            x1(m)=(deslocamentos(nos_elem(1,i),3*t-1)-0)*valor_escala_0_4a*a/desl_max_vdx(1,t);
                        else
                            x1(m)=(deslocamentos(nos_elem(1,i),3*t-1)-deslocamentos(nos_elem(1,i),3*(t-1)-
                            1))*valor_escala_0_4a*a/desl_max_vdx(1,t);
                        end
                        y1(m)=0;
                    elseif vdx==1
                        if t==1
                            y1(m)=(deslocamentos(nos_elem(1,i),3*t)-0)*valor_escala_0_5*a/desl_max_vdxy(1,t);
                            x1(m)=(deslocamentos(nos_elem(1,i),3*t-1)-0)*valor_escala_0_5*a/desl_max_vdxy(1,t);
                        else
                            y1(m)=(deslocamentos(nos_elem(1,i),3*t)-deslocamentos(nos_elem(1,i),3*(t-
                            1)))*valor_escala_0_5*a/desl_max_vdxy(1,t);
                            x1(m)=(deslocamentos(nos_elem(1,i),3*t-1)-deslocamentos(nos_elem(1,i),3*(t-1)-
                            1))*valor_escala_0_5*a/desl_max_vdxy(1,t);
                        end
                    elseif vty==1 || vtx==1
                        no_tensoes=nos_elem(1,i);
                        if vty==1
                            t_x1(m)=0;
                            t_y1(m)=ty1(no_tensoes,t)*valor_escala_0_4b*a/t_max;
                        elseif vtx==1
                            t_x1(m)=tx1(no_tensoes,t)*valor_escala_0_25*a/t_max;
                            t_y1(m)=0;
                        end
                    end
                end
            end
        end
    end
end

```

```

        x2(m)=coordenadas(nos_elem(1,i),2)+deslocamentos(nos_elem(1,i),3*t-1)*valor_escala/desl_max;
        y2(m)=coordenadas(nos_elem(1,i),3)+deslocamentos(nos_elem(1,i),3*t)*valor_escala/desl_max;
        m=m+1;
    end
end
end
x_min=min(x_y_nos_elem(1:MNODE,1));
x_max=max(x_y_nos_elem(1:MNODE,1));
y_min=min(x_y_nos_elem(1:MNODE,2));
y_max=max(x_y_nos_elem(1:MNODE,2));
for i=1:MNODE
    if x_y_nos_elem(i,1)==x_min
        if x_y_nos_elem(i,2)==y_max
            plot_no(1,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
        else
            if x_y_nos_elem(i,2)==y_min
                plot_no(2,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            else
                if plot_no(1,1:NDIME)==99999
                    plot_no(1,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
                else
                    if x_y_nos_elem(i,2)>plot_no(1,2)
                        plot_no(1,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
                    end
                end
            end
            if plot_no(2,1:NDIME)==99999
                plot_no(2,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            else
                if x_y_nos_elem(i,2)<plot_no(2,2)
                    plot_no(2,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
                end
            end
        end
    end
end
end
if x_y_nos_elem(i,1)==x_max
    if x_y_nos_elem(i,2)==y_max
        plot_no(4,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
    else
        if x_y_nos_elem(i,2)==y_min
            plot_no(3,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
        else
            if plot_no(3,1:NDIME)==99999
                plot_no(3,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            else
                if x_y_nos_elem(i,2)<plot_no(3,2)
                    plot_no(3,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
                end
            end
        end
    end
end
end
else
    if x_y_nos_elem(i,2)==y_max
        if plot_no(1,1:NDIME)==99999
            plot_no(1,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
        else
            if x_y_nos_elem(i,1)<plot_no(1,1)
                plot_no(1,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            end
        end
        if plot_no(4,1:NDIME)==99999
            plot_no(4,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
        else
            if x_y_nos_elem(i,1)>plot_no(4,1)
                plot_no(4,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            end
        end
    end
else
    if x_y_nos_elem(i,2)==y_min
        if plot_no(3,1:NDIME)==99999
            plot_no(3,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
        else
            if x_y_nos_elem(i,1)>plot_no(1,1)
                plot_no(3,1:NDIME)=x_y_nos_elem_tempo(i,1:NDIME);
            end
        end
    end
end
end
end

```

```

        end
    end
end
end
end
x_min2=min([x_min2 x_min]);
y_min2=min([y_min2 y_min]);
x_max2=max([x_max2 x_max]);
y_max2=max([y_max2 y_max]);
plot_no2(1:4,2*n-1:2*n)=plot_no;
if vdx==1 || vdx==1 || vdy==1
    x2_2(n,1:MNODE)=x2;
    y2_2(n,1:MNODE)=y2;
    x1_2(n,1:MNODE)=x1;
    y1_2(n,1:MNODE)=y1;
elseif vtx==1 || vty==1
    x2_2(n,1:MNODE)=x2;
    y2_2(n,1:MNODE)=y2;
    t_x1_2(n,1:MNODE)=t_x1;
    t_y1_2(n,1:MNODE)=t_y1;
end
end
plot_no3(4*t-3:4*t,1:2*NELEM)=plot_no2;
if vdx==1 || vdx==1 || vdy==1
    x2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=x2_2;
    y2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=y2_2;
    x1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=x1_2;
    y1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=y1_2;
elseif vtx==1 || vty==1
    x2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=x2_2;
    y2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=y2_2;
    t_x1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=t_x1_2;
    t_y1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t)=t_y1_2;
end
end
for t=1:NINCR
    if t==1
        hold off
        malha();
        if vdy==1
            title('Gráfico1: Deslocamento total e vetores do deslocamento vertical','FontWeight','bold')
        elseif vdx==1
            title('Gráfico1: Deslocamento total e vetores do deslocamento horizontal','FontWeight','bold')
        elseif vdx==1
            title('Gráfico1: Deslocamento total e vetores do deslocamento total','FontWeight','bold')
        elseif vty==1
            title('Gráfico1: Deslocamento total e vetores da tensão vertical efetiva','FontWeight','bold')
        elseif vtx==1
            title('Gráfico1: Deslocamento total e vetores da tensão horizontal efetiva','FontWeight','bold')
        else
            title('Gráfico1: Deslocamento total','FontWeight','bold')
        end
        xlabel('Distância ao eixo y (m)')
        ylabel('Distância ao eixo x (m)')
        pause(2)
        hold off
    end
    plot(x_min2-1*a,y_min2-1*a,'w');
    hold on
    plot(x_max2+1*a,y_max2+1*a,'w');
    hold on
    text(x_min2,y_min2,' \rightarrow', 'FontSize',20,'HorizontalAlignment','left','VerticalAlignment','middle');
    text(x_min2,y_min2,'x', 'FontSize',11,'HorizontalAlignment','left','VerticalAlignment','cap');
    text(x_min2,y_min2,' \uparrow', 'FontSize',20,'HorizontalAlignment','right','VerticalAlignment','baseline');
    text(x_min2,y_min2,'y', 'FontSize',11,'HorizontalAlignment','right','VerticalAlignment','baseline');
    for n=1:NELEM
        plot([plot_no3(4*t-3,2*n-1),plot_no3(4*t-2,2*n-1)],[plot_no3(4*t-3,2*n),plot_no3(4*t-2,2*n)],'r-');
        hold on
        plot([plot_no3(4*t-2,2*n-1),plot_no3(4*t-1,2*n-1)],[plot_no3(4*t-2,2*n),plot_no3(4*t-1,2*n)],'r-');
        hold on
        plot([plot_no3(4*t-1,2*n-1),plot_no3(4*t,2*n-1)],[plot_no3(4*t-1,2*n),plot_no3(4*t,2*n)],'r-');
        hold on
        plot([plot_no3(4*t,2*n-1),plot_no3(4*t-3,2*n-1)],[plot_no3(4*t,2*n),plot_no3(4*t-3,2*n)],'r-');
        hold on
    end
end
if vdx==1 || vdx==1 || vdy==1

```

```

    quiver(x2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),y2_3(1:NELEM,MNODE*t-(MNODE-
1):MNODE*t),x1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),y1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),0);
    hold on
    elseif vtx==1 || vty==1
        quiver(x2_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),y2_3(1:NELEM,MNODE*t-(MNODE-
1):MNODE*t),t_x1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),t_y1_3(1:NELEM,MNODE*t-(MNODE-1):MNODE*t),0);
        hold on
    end
    if IEDGE==1
        numero_nos=size(valor_carga);
        aux_plot=1;
        for i=1:NEDGE
            if max(valor_carga(2*i-1,1:numero_nos(2)))<=0 || min(valor_carga(2*i-1,1:numero_nos(2)))<0 ||
max(valor_carga(2*i,1:numero_nos(2)))>0 || min(valor_carga(2*i,1:numero_nos(2)))<0
                aux_plot=0;
            end
        end
        if aux_plot==1
            if IPLOD==1
                if max(cargas_pontuais(1:numero_cargas_pontuais-1,3))>0 || max(cargas_pontuais(1:numero_cargas_pontuais-1,2))~=0 ||
min(cargas_pontuais(1:numero_cargas_pontuais-1,2))~=0
                    aux_plot=0;
                end
            end
        end
        if aux_plot==1
            for i=1:NEDGE
                if max(valor_carga(2*i-1,1:numero_nos(2)))>0 && min(valor_carga(2*i-1,1:numero_nos(2)))>=0 &&
max(valor_carga(2*i,1:numero_nos(2)))=0 && min(valor_carga(2*i,1:numero_nos(2)))=0
                    plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)],'b-');
                    hold on
                    plot([Xmin_carga(i,1),Xmin_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.5*a], 'b-');
                    hold on
                    plot([Xmin_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1)+0.5*a,Ymin_carga(i,1)+0.5*a], 'b-');
                    hold on
                    plot([Xmax_carga(i,1),Xmax_carga(i,1)],[Ymin_carga(i,1),Ymin_carga(i,1)+0.5*a], 'b-');
                    text((Xmin_carga(i,1)+Xmax_carga(i,1))/2,Ymin_carga(i,1),
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
                end
            end
            if IPLOD==1
                numero_cargas_pontuais=size(cargas_pontuais);
                if max(cargas_pontuais(1:numero_cargas_pontuais-1,3))<=0 && max(cargas_pontuais(1:numero_cargas_pontuais-1,2))=0 &&
min(cargas_pontuais(1:numero_cargas_pontuais-1,2))=0
                    for i=1:numero_cargas_pontuais-1
                        if max(cargas_pontuais(i,3))<0
                            text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3)+0.5*a,'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
                        end
                    end
                end
            end
        end
        elseif IPLOD==1
            numero_cargas_pontuais=size(cargas_pontuais);
            if max(cargas_pontuais(1:numero_cargas_pontuais-1,3))<=0 && max(cargas_pontuais(1:numero_cargas_pontuais-1,2))=0 &&
min(cargas_pontuais(1:numero_cargas_pontuais-1,2))=0
                for i=1:numero_cargas_pontuais-1
                    if max(cargas_pontuais(i,3))<0
                        text(coordenadas(cargas_pontuais(i,1),2),coordenadas(cargas_pontuais(i,1),3),'
\downarrow','FontSize',11,'HorizontalAlignment','center','VerticalAlignment','baseline');
                    end
                end
            end
        end
    end
    if vdy==1
        title('Gráfico1: Deslocamento total e vetores do deslocamento vertical','FontWeight','bold')
    elseif vdx==1
        title('Gráfico1: Deslocamento total e vetores do deslocamento horizontal','FontWeight','bold')
    elseif vdxy==1
        title('Gráfico1: Deslocamento total e vetores do deslocamento total','FontWeight','bold')
    elseif vty==1
        title('Gráfico1: Deslocamento total e vetores da tensão vertical efetiva','FontWeight','bold')
    elseif vtx==1
        title('Gráfico1: Deslocamento total e vetores da tensão horizontal efetiva','FontWeight','bold')
    else

```

```

    title('Gráfico1: Deslocamento total','FontWeight','bold')
end
xlabel('Distância ao eixo y (m)')
ylabel('Distância ao eixo x (m)')
if vty==1
    save dados2 t ty1
elseif vtx==1
    save dados2 t tx1
else
    save dados2 t
end
pause(0.5)
hold off
end

```

### **function grafico desl v no vs t**

```

load dados coordenadas NINCR deslocamentos tempo tempo_mod
y_no_desl=coordenadas(no,3);
for t=1:NINCR
    desl_y(t)=deslocamentos(no,3*t)*-1;
end
hold('all');
if eixo_tempo==1
    plot(tempo,desl_y,'-r*')
    xlabel('Tempo (s)')
else
    plot(tempo_mod,desl_y,'-r*')
    xlabel('Fator T` (s)')
end
title('Gráfico1: Deslocamento vertical do nó ao longo do tempo','FontWeight','bold')
ylabel('Deslocamento vertical (m)')
if escala_linear==1
    set(gca,'xscale','linear');
else
    set(gca,'xscale','log');
end
end

```

### **function grafico desl v nos profund**

```

load dados coordenadas NPOIN deslocamentos titulo_problema
if strcmp(titulo_problema,'unidimensional')
    x_no=coordenadas(no,2);
    n=1;
    for i=1:NPOIN
        if coordenadas(i,2)==x_no
            desl_y(n,1:3)=[coordenadas(i,1) coordenadas(i,3) deslocamentos(i,3*t)*-1];
            n=n+1;
        end
    end
    tamanho_desl_y=size(desl_y);
    hold('all');
    plot(max(desl_y(1:tamanho_desl_y(1),2))-desl_y(1:tamanho_desl_y(1),2),desl_y(1:tamanho_desl_y(1),3),'-r*')
    title('Gráfico1: Deslocamento vertical de uma linha vertical com a abcissa do nó escolhido vs profundidade para o tempo
definido','FontWeight','bold')
    xlabel('Profundidade (m)')
    ylabel('Deslocamento vertical (m)')
elseif strcmp(titulo_problema,'bidimensional ')
    y_no=coordenadas(no,3);
    n=1;
    for i=1:NPOIN
        if coordenadas(i,3)==y_no
            desl_y(n,1:3)=[coordenadas(i,1) coordenadas(i,2) deslocamentos(i,3*t)*-1];
            n=n+1;
        end
    end
    tamanho_desl_y=size(desl_y);
    hold('all');
    plot(desl_y(1:tamanho_desl_y(1),2),desl_y(1:tamanho_desl_y(1),3),'-r*')
    title('Gráfico1: Deslocamento vertical de uma linha horizontal com a ordenada do nó escolhido vs distância ao eixo y para o tempo
definido','FontWeight','bold')
    xlabel('Distância ao eixo y (m)')
    ylabel('Deslocamento vertical (m)')
end
end

```



### **function grafico rpp vs t**

```
load dados NINCR pressao_neutra tempo_max_valor_carga tempo_mod
for t=1:NINCR
    P(t)=pressao_neutra(no,2*t)*-1;
end
if eixo_rp==1
    rpp=P;
else
    rpp=P/max_valor_carga;
end
if eixo_tempo==1
    plot(tempo,rpp,'-r*')
    xlabel('Tempo (s)')
else
    plot(tempo_mod,rpp,'-r*')
    xlabel('Fator T (s)')
end
title('Gráfico1: Pressão neutra do nó ao longo do tempo','FontWeight','bold')
if eixo_rp==1
    ylabel('rp (kPa)')
else
    ylabel('rp/p')
end
box off
if escala_linear==1
    set(gca,'xscale','linear');
else
    set(gca,'xscale','log');
end
end
```

### **function grafico rpp vs xa**

```
load dados coordenadas NPOIN pressao_neutra max_valor_carga a titulo_problema
if strcmp(titulo_problema,'unidimensional')
    x_no=coordenadas(no,2);
    n=1;
    for i=1:NPOIN
        if coordenadas(i,2)==x_no
            nos_y(n,1:3)=[coordenadas(i,1) coordenadas(i,3) (pressao_neutra(i,2*t2)*-1)];
            n=n+1;
        end
    end
    tamanho_nos_y=size(nos_y);
    hold('all');
    if eixo_rp==1
        plot(max(nos_y(1:tamanho_nos_y(1),2))-nos_y(1:tamanho_nos_y(1),2),nos_y(1:tamanho_nos_y(1),3),'-r*')
        ylabel('rp (kPa)')
    else
        plot(max(nos_y(1:tamanho_nos_y(1),2))-nos_y(1:tamanho_nos_y(1),2),nos_y(1:tamanho_nos_y(1),3)/max_valor_carga,'-r*')
        ylabel('rp/p')
    end
    xlabel('Profundidade (m)')
    title('Gráfico1: Pressão neutra de uma linha vertical com a abcissa do nó escolhido vs profundidade para o tempo definido','FontWeight','bold')
    box off
elseif strcmp(titulo_problema,'bidimensional ')
    y_no=coordenadas(no,3);
    n=1;
    for i=1:NPOIN
        if coordenadas(i,3)==y_no
            nos_y(n,1:3)=[coordenadas(i,1) coordenadas(i,2) (pressao_neutra(i,2*t2)*-1)];
            n=n+1;
        end
    end
    tamanho_nos_y=size(nos_y);
    hold('all');
    if eixo_rp==1
        if eixo_x==1
            plot(nos_y(1:tamanho_nos_y(1),2),nos_y(1:tamanho_nos_y(1),3),'-r*')
            xlabel('Distância ao eixo y (m)')
        else
            plot(nos_y(1:tamanho_nos_y(1),2)/a,nos_y(1:tamanho_nos_y(1),3),'-r*')
            xlabel('x/a (m)')
        end
    end
end
```

```

    ylabel('rp (kPa)')
else
    if eixo_x==1
        plot(nos_y(1:tamanho_nos_y(1),2),nos_y(1:tamanho_nos_y(1),3)/max_valor_carga,'-r*')
        xlabel('Distância ao eixo y (m)')
    else
        plot(nos_y(1:tamanho_nos_y(1),2)/a,nos_y(1:tamanho_nos_y(1),3)/max_valor_carga,'-r*')
        xlabel('x/a (m)')
    end
    ylabel('rp/p')
end
title('Gráfico1: Pressão neutra de uma linha horizontal com a ordenada do nó escolhido vs distância ao eixo y para o tempo
definido','FontWeight','bold')
box off
end

```