



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE GEOCIÊNCIAS
CURSO DE GRADUAÇÃO EM GEOFÍSICA
TRABALHO DE CONCLUSÃO DE CURSO**

CARLOS MATEUS BARRIGA NUNES

**CÁLCULO NUMÉRICO DE DERIVADAS PARA MODELAGEM
2D DE MÉTODOS ELETROMAGNÉTICOS**

BELÉM

2012

CARLOS MATEUS BARRIGA NUNES

**CÁLCULO NUMÉRICO DE DERIVADAS PARA MODELAGEM
2D DE MÉTODOS ELETROMAGNÉTICOS**

Trabalho de conclusão de curso apresentado à Faculdade de Geofísica da Universidade Federal Do Pará - UFPA como exigência para a obtenção do título de Bacharel em Geofísica .

Orientador: Prof. Dr. Cícero Roberto Teixeira Régis

BELÉM

2012

CARLOS MATEUS BARRIGA NUNES

CÁLCULO NUMÉRICO DE DERIVADAS PARA MODELAGEM
2D DE MÉTODOS ELETROMAGNÉTICOS

Trabalho de conclusão de curso apresentado à
Faculdade de Geofísica da Universidade Federal
Do Pará - UFPA como exigência para a obtenção
do título de Bacharel em Geofísica

Data de Aprovação :

Nota:

Banca de Dissertação:

Prof. Cícero Roberto Teixeira Régis (Orientador)
Doutor em Geofísica
Univesidade Federal do Pará

Prof. Vitor Cesar Tocantins de Souza
Doutor em Geofísica
Univesidade Federal do Pará

M.Sc. Walleson Gomes dos Santos
Mestre em Geofísica
Univesidade Federal do Pará

Aos meus filhos Davi Nunes e Samuel Nunes

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus que permitiu e me capacitou para chegar até aqui. À minha mãe pelo apoio e incentivo, à minha namorada Keylla Armanda pelo amor e dedicação e a minha família. Ao Prof. Dr. Cícero Roberto Teixeira Régis, pela oportunidade concedida de realizar este trabalho, pela honra da orientação, pela disposição, discussões e esclarecimentos. Ao Prof. Dr. Victor Cezar Tocantins de Souza pelas sugestões e esclarecimentos.

Agradeço ao PRH/ANP-06 pelo apoio financeiro, o qual tornou esse trabalho possível.

Agradeço também a todos que generosamente oneraram parte do seu tempo em discussões ou esclarecimentos inerentes ao desenvolvimento deste trabalho, grupo **EMMI**.

RESUMO

Na modelagem de métodos eletromagnéticos, calcular derivadas é geralmente uma tarefa indispensável. Neste trabalho, resolve-se numericamente a equação de Poisson com condições de Dirichlet homogênea utilizando o método dos elementos finitos. A discretização do domínio para implementação do método é obtida utilizando um software livre de geração de malha não estruturada, *Triangle*. Portanto, descreve-se como gerar uma malha usando o *Triangle*.

Para calcular a derivada numérica são utilizados dois métodos, o primeiro é baseado em derivar as funções bases, ou seja, a derivada em um nó é igual a média do gradiente das funções base na vizinhança $\frac{1}{2}$ deste nó. O segundo, trata-se de um ajuste móvel por mínimos quadrados ponderados, em que o peso é definido por uma função gaussiana. Os resultados da derivada numérica são obtidos a partir de sucessivos refinamentos da malha de elementos finitos e comparados com a derivada da solução analítica. Em primeiro lugar é usado a solução analítica para avaliar o erro adquirido pelo cálculo das derivadas e posteriormente a solução numérica. Então, avalia-se a eficiência dos métodos.

Palavras - chave: Método dos Elementos Finitos, Triangle, Malha não estruturada, recuperação de derivadas, ajuste móvel por mínimos quadrados.

ABSTRACT

modeling of electromagnetic methods. calculate derivatives is generally a task indispensable. In this paper, we solve numerically the Poisson equation with conditions homogeneous Dirichlet using the finite element method. The discretization of the domain for implementation of the method is achieved using a free software unstructured mesh generation, Triangle. Therefore, we describe how to generate a mesh using Triangle.

To calculate the numerical derivatives are used two methods. basing it on the first derive the basis functions, the derivative at a node is equal to the average of the gradients basis functions in the neighborhood of this node. Second, it is a sliding fit by least weighted square, where the weight is defn a Gaussian function. the results the numerical derivative are obtained from successive refinements of the mesh elements and compared with the finite derived from the analytical solution. First is used to analytical solution to evaluate the error Acquired by the calculation of the derivatives and then the numerical solution. Then, we evaluate the efficiency of the methods.

Key words: Finite element method, Triangle, unstructured mesh, recovery derived, sliding fit by least squares.

LISTA DE ILUSTRAÇÕES

2.1	representa o arquivo de entrada quadrado.poly. Seu segmento interno tem marcador de fronteira 33	18
2.2	Representa o arquivo de saída 'quadrado.1.poly	19
2.3	A figura representa o arquivo de saída quadrado.1.ele com 16 triângulos. . .	20
2.4	Esta figura representa o arquivo de entrada quadrado2.poly	21
2.5	A figura mostra 371 vértices do arquivo de saída quadrado'2.1.node.	21
2.6	Representa os 686 triângulos do arquivo de saída quadrado2.1.ele.	21
3.1	A figura mostra que a função base Ψ_i em um nó local j é 1 se $i = j$ e 0 se $i \neq j$.	25
3.2	Elemento qualquer de uma malha hipotética com numeração arbitrária. . . .	28
3.3	A figura representa o arquivo de entrada .poly, ele ilustra a geometria do domínio onde está sendo calculada a solução.	30
3.4	A figura representa o arquivo de saída .ele com discretização do domínio em 1372 nós e 2605 elementos, ele ilustra o último refinamento sobre o .poly de entrada figura(3.3)	30
3.5	Solução numérica da equação de Poisson sobre a malha da figura(3.4) com discretização do domínio em 2605 elementos.	31
3.6	Solução analítica da equação de Poisson sobre a malha da figura(3.3) com discretização do domínio em 2605 elementos.	31
3.7	A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.	32
3.8	A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.	33
3.9	A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.	33
4.1	domínio local arbitrário de uma malha hipotética	35
4.2	A figura ilustra as interpolações na vizinhança de um nó qualquer da malha onde a derivada será calculada	35
4.3	A figura mostra a função peso, em que $x_g = 1.0, z_g = 1.0, x_g - x_m = 0.3$ e $ z_g - z_m = 0.4$. Portanto, o peso é maior na origem onde se está calculando a derivada.	36

4.4	figura de uma malha hipotética ilustrando que o método independe da forma que o domínio é discretizado e que não há restrição para origem do domínio local.	37
4.5	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos	40
4.6	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos	41
4.7	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos	42
4.8	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos	43
4.9	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos	45
4.10	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos	46
4.11	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos	47
4.12	Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica . EMG e EMQMP são seus erros relativos	48

SUMÁRIO

1	INTRODUÇÃO	12
2	TRIANGLE	13
2.1	INTRODUÇÃO	13
2.2	INTERRUPTORES DE COMANDO	13
2.2.1	Descrição de alguns interruptores	13
2.3	FORMATOS DE ARQUIVOS	14
2.3.1	Arquivo .node	14
2.3.2	Arquivo .ele	15
2.3.3	Arquivo .poly	15
2.3.4	Marcadores de fronteira	17
2.3.5	Exemplos	17
3	MODELAGEM NUMÉRICA	23
3.1	ELEMENTOS FINITOS 2D	23
3.2	CASO ELETROSTÁTICO	23
3.2.1	Solução analítica	23
3.2.2	Solução numérica	24
3.3	RESULTADOS	30
4	RECUPERAÇÃO DE DERIVADAS	34
4.1	INTRODUÇÃO	34
4.2	DERIVADA ANALÍTICA	34
4.3	MÉDIA ARITMÉTICA DO GRADIENTE DAS FUNÇÕES BASE	34
4.4	AJUSTE MÓVEL POR MÍNIMOS QUADRADOS PONDERADOS	36
4.5	RESULTADOS	39

5 CONCLUSÃO

49

REFERÊNCIAS

50

1 INTRODUÇÃO

O cálculo numérico de derivadas é uma tarefa importante na modelagem numérica de Métodos Eletromagnéticos, por exemplo, na modelagem de dados magnetotelúrico marinho, as componentes x e z do campo elétrico e magnético são obtidas a partir da derivada das componentes y dos campos (SAITO, 2009). Além disso, é possível mencionar a utilização de um operador 2D no cálculo da derivada temporal para filtragem de dados sísmicos, (MELO, 2007). Sendo assim, depois de obter uma solução numérica pelo método dos elementos finitos (MEF), em geral, deseja-se recuperar as derivadas do campo encontrado em cada nó de uma malha ou em um ponto arbitrário no domínio do problema, seja ele escalar ou vetorial.

O MEF será aplicado à equação de Poisson usando uma malha não estruturada. Mesmo não sendo uma necessidade inerente ao problema de Poisson, a escolha de usar uma malha não estruturada em detrimento da regular, pode ser justificada ao tentar reduzir o erro de discretização ou melhorar o refinamento em uma região onde se quer maior aproximação da solução. Quando adicionado um número maior de nós, então o refinamento é difundido por toda a malha, dependendo mais tempo para obter a solução. A malha irregular controla a densidade de nós localmente. Entretanto, usar a malha não estruturada torna mais difícil o cálculo das derivadas, portanto, o objetivo do trabalho é obter por um método mais geral, que independa da geometria do elemento, as derivadas de soluções de elementos finitos em uma malha triangular não estruturada.

No capítulo 3, o MEF é usado para resolver equação de Poisson com condições de Dirichlet homogênea (RIJO, 1991). Um dos passos na implementação do MEF é a discretização do domínio, nesta etapa, utilizou-se o Triangle - software de geração automática de malha triangular não estruturada - O qual será abordado no capítulo 2

No capítulo 4, os cálculos das derivadas são realizados de duas maneiras. A primeira consiste basicamente em utilizar as derivadas das funções base. A segunda, um método mais geral independente da geometria do elemento, podendo ele ser, triangular, quadrangular, hexagonal e etc, consiste em um ajuste móvel por mínimos quadrados ponderado, (TABBARA, 1994). Outra forma de obter as derivadas seria deixar, I^+ e I^- , representar a função $f(x, z)$ na vizinhança de onde se quer calcular a derivada e encontrar tais funções por interpolação linear, (MELO, 2007). Por fim, o capítulo 5 apresenta as considerações sobre a eficiência dos métodos.

2 TRIANGLE

2.1 INTRODUÇÃO

Triangle é um gerador de malha 2D triangular de alta qualidade, uma vez que a malha poderá ser gerada com ângulos muito pequenos ou muito grandes, ou seja, sendo adequado para análise de elementos finitos, (www.cs.cmu.edu). As entradas do Triangle são arquivos com extensão .NODE, .POLY, .ELE e .AREA, contudo, também são arquivos de saída exceto o arquivo .AREA. Outros arquivos de saída existem, tais como .EDGE, .NEIGH, etc. Estes arquivos de saída não serão descritos neste trabalho.

2.2 INTERRUPTORES DE COMANDO

Arquivos de entrada podem guardar informações sobre a malha a ser gerada de forma separada, por exemplo, um arquivo contendo os vértices e outro as segmentos. Podem , conter restrições para refinamento e controle de densidade local da mesma, além disso, pode-se impor restrições de área e de ângulo máximo. Utilizar essas opções depende dos comandos que são usados sobre o arquivo de entrada. A sintaxe de comando é:

Triangle [-prq_a_uAcDjevngBPNEIOXzo_Y...] arquivo de entrada

o sublinhado significa que podem opcionalmente aparecer um parâmetro numérico após o interruptor de comando associado. Não é permitido espaço entre o interruptor e seu parâmetro

2.2.1 Descrição de alguns interruptores

- p** lê um PSLG do inglês Planar Straight Line Graph(arquivo .poly). Um PSLG é um conjunto de vértices e segmentos, estes segmentos são bordas onde seus pontos finais são todos os vértices no arquivo .poly em questão. Segmentos podem se cruzar somente em seus pontos finais
- q** impõe uma restrição de ângulo mínimo e portanto, está associado á qualidade da malha gerada. Vértices seram adicionados na malha para garantir que todos os ângulos estejam entre 20 e 140 graus. Opcionalmente, pode-se modificar o limite de ângulo mínimo, que por default é 20 graus, especificando este valor depois de -q. O valor especificado pode conter pontos decimais, mas não pode ter uma notação exponencial. Uma vez

escolhido o ângulo mínimo (teta), o máximo é determinado pela relação $180 - 2 \times \text{teta}$, sendo assim, se é estipulado 23.5 como ângulo mínimo, o máximo é 133.

- a é usado para impor uma área máxima sobre os triângulos da malha. Se um número acompanhar -a, então nenhum triângulo é gerado com área superior a este número. Entretanto, se nenhum número é especificado e além de -a é usado o -r, um arquivo .area é lido e usado para especificar um conjunto de restrições de área máxima. Este arquivo possui uma restrição de área para cada triângulo, portanto, muito importante para refinamento da malha e controle de densidade local. Contudo, se -a é usado mas o -r não, um arquivo .poly especifica restrições de área, uma vez que ele pode opcionalmente ter tais restrições para cada região na malha que é delimitada por segmentos de fronteira. Observe que este procedimento é útil para controle de densidade em uma primeira triangulação. Além disso, é possível impor restrições de área fixa e variável.
- j elimina vértices que não fazem parte da triangulação final do arquivo .node(arquivo que contém todos os vértices de entrada) de saída. Por padrão, Triangle copia todos os vértices de uma arquivo .node de entrada para um .node de saída mantendo a mesma ordem. -j evita que vértices de entrada que foram duplicados ou vértices que foram engolidos por buracos apareçam no arquivo .node de saída. Portanto, se dois vértices possuem a mesma coordenada, apenas um estará no arquivo de saída. Se vertices são eliminados num destes casos, a numeração dos vértices de entrada são diferentes no arquivo de saída.

2.3 FORMATOS DE ARQUIVOS

Os arquivos de entrada trazem informações, tais como, vértices, bordas, buracos, triângulos, restrições de área máxima. Vértices devem ser numerados consecutivamente e de forma consistente, ou seja, se vértices são rotulados, a fronteira em que eles ocorrem também deve ter a mesma numeração, além disso, '#' é utilizado para acrescentar comentários ao longo dos arquivos.

2.3.1 Arquivo .node

Quando este arquivo é lido, Triangle por padrão gera como saída os arquivos .poly, .node e .ele. Este arquivo contém as informações sobre os vértices de entrada de uma triangulação. Ele é construído da seguinte forma:

primeira linha: {Nº de vértices}{dimensão(é sempre 2)}{N atributo}{marcador de fronteira(que pode ser 0 ou 1)}

linha à baixo: {vértice}{x}{y}{atributo}{marcador de fronteira}

Os atributos são opcionais, normalmente valores de propriedades físicas, tais como resistividade ou densidade. Eles estão associados com cada nó da malha de elementos finitos. Estes parâmetros são transferidos para o arquivo `.node` de saída sem alterações. Se comandos que forcem pontos adicionais à malha de saída são usados(-q,-a,...), os atributos associados a esses pontos são desigandos por interpolação linear. A última coluna do arquivo é de marcadores de fronteira quando a quarta entrada da primeira linha é o 'número 1'

2.3.2 Arquivo `.ele`

Este arquivo lista os triângulos da malha de entrada ou de saída, sua forma é:

primeira linha: {N triângulos}{nós por triângulo}{N atributos}

linhas à baixo: {triângulo}{nó1}{nó2}{nó3}... {atributo}

Os nós são aqueles do arquivo `.node` e correspondem aos vértices dos triângulos, em que a numeração tem uma orientação anti-horária. Em geral os triângulos em um arquivo `.ele` gerado por Triangle terão três nós, a não ser que -o2 seja usado, então possuirá seis, os nós (4, 5 e 6) que estarão no ponto médio das bordas do triângulo, opostos aos vértices(1, 2 e 3) respectivamente.

Os atributos são como os do arquivo `.node`. Triangle tenta interpolar os atributos quando a malha é refinada, e a difusão de atributos estão limitadas à fronteiras externas e internas da malha.

2.3.3 Arquivo `.poly`

O arquivo `.poly` representa um PSLG e pode obter algumas informações adicionais. Quando Triangle lê este arquivo gera arquivo `.poly`, `.node` e `.ele`. O `.poly` de saída inclui todos os subsegmentos das bordas que são parte dos segmentos do `.poly` de entrada. Desde já, o arquivo de saída `.poly` é útil para encontrar bordas que estão associadas a segmentos de entrada e ajustar condições de fronteira, e além do mais, evitar que segmentos sejam perdidos ao refinar uma malha de saída.

Secção 1-nós

primeira linha: {N vértices}{dimensão(sempre 2, pois a malha é 2D)}{N atributos}{marcador de fronteira(que pode ser 0 ou 1)}

linha à baixo: {vértice}{x}{y}{atributo}{marcador de fronteira}

Secção 2-segmentos

Primeira linha: {N segmentos}{marcador de fronteira(0 ou 1)}

linha à baixo: {segmento}{ponto1}{ponto2}{marcador de fronteira}

Secção 3-buracos

primeira linha: {N buracos}

linha à baixo: {buraco}{x}{y}

Secção 4-opcional

primeira linha: {N atributos regionais e/ou restrição de área}

linhas à baixo: {região}{x}{y}{atributo}{área máxima}

A primeira Secção lista todos os vértices igual ao arquivo .node. Mas N vértices pode ser 0 indicando que os vértices serão listados separadamente em um arquivo .node.

Na primeira linha da Secção (2), têm-se o número segmentos que são bordas que serão executadas na triangulação. Na linha seguinte, cada segmento é especificado por uma lista de índices de seus dois pontos finais, em que cada ponto é incluído da lista de vértices. Os segmentos também podem incluir marcadores de fronteira. Portanto, marcadores de nós e segmentos devem ser consistentes.

A primeira linha da Secção (3) é o número de buracos existentes na malha, a linha que segue, lista buracos. Os buracos são especificados atribuindo uma coordenada dentro da região, então, Triangle triangula e depois cria os buracos que se difundem por toda região especificada, sendo bloqueado por segmentos no arquivo .poly.

A secção (4) é opcional. A primeira linha especifica o número de atributos e/ou o de restrições de área máxima regional. Na linha seguinte as colunas 1, 2, 3 e 4 especificam o número de regiões, coordenadas, área máxima e atributos regionais respectivamente. Triangle

lê essa secção somente se -A é usado ou se -a é usado sem está acompanhando por um número e -r não é usado. Restrições de área e atributos regionais, propagam-se da mesma forma que buracos. Os valores de atributos são aplicados apenas se -A é usado, enquanto restrições de área máxima se -a é usado. Caso seja necessário designar atributos sem impor uma área máxima para uma determinada região, usa-se um valor negativo.

2.3.4 Marcadores de fronteira

Marcadores de fronteira são rótulos, usados sobretudo para identificar quais bordas e vértices de saída estão associados com quais segmentos de um PSLG(.poly), e quais bordas e vértices acontecem em uma fronteira de triangulação, portanto, marcadores são comumente usados para se determinar onde as condições de fronteira serão aplicadas para malha de elementos finitos.

Os marcadores de fronteira associados com cada segmento em um arquivo de saída .poly e cada borda em um arquivo de saída .edge são escolhidos da seguinte forma: Se uma borda está totalmente ou parcialmente em segmento PSLG com um marcador diferente de zero, então a borda terá o mesmo marcador. Senão, se a borda estiver na fronteira da triangulação, então recebe o marcador de número um (1). Além disso, se nenhum dos casos se verificar o marcador é designado por zero (0).

No caso de vértices em uma arquivo de saída .node, se um vértice recebeu um marcador de fronteira diferente de zero em um arquivo de entrada .node, então seu marcador segue inalterado para o arquivo de saída .node. Se um vértice não recebeu um marcador diferente de zero e fica em um segmento PSLG com marcador diferente de zero, ele recebe o mesmo marcador. Contudo, se o vértice estiver em vários segmentos o marcador é escolhido arbitrariamente. Senão, se um vértice fica na fronteira de triangulação, seu marcador é designado por um (1). Por fim, um vértice terá seu marcador designado por zero se estiver fora de qualquer fronteira, seja ele interna ou externa(0).

2.3.5 Exemplos

Exemplo 1: Neste exemplo o arquivo de entrada é quadrado.poly, tabela(2.1), como o nome sugere, descreve um quadrado, mas com um segmento de fronteira interna horizontal que o divide em duas regiões, ver figura (2.1).

Tabela 2.1: A tabela é o arquivo de entrada quadrado.poly com 6 vértices, 5 segmentos e nenhum buraco

```
‡ quadrado com seis nós e marcador de fronteira
```

```
6 2 0 1
```

```
1 0.0 0.0 0
```

```
2 0.0 2.0 0
```

```
3 2.0 2.0 0
```

```
4 2.0 0.0 0
```

```
5 0.0 1.0 0
```

```
6 2.0 1.0 0
```

```
‡ cinco segmentos com marcado de fronteira
```

```
5 1
```

```
1 1 2 0
```

```
2 2 3 0
```

```
3 3 4 0
```

```
4 4 1 0
```

```
5 5 6 33 ‡ marcador da fronteira interna
```

```
‡ nenhum buraco
```

```
0
```

Figura 2.1: representa o arquivo de entrada quadrado.poly. Seu segmento interno tem marcador de fronteira 33



Executando `Triangle -pq30a0.4 quadrado.poly`, nota-se no arquivo, `'quadrado.1.node'` tabela (2.2), que sete vértices foram adicionados para satisfazer a restrição de ângulo mínimo e área máxima (nenhum triângulo tem área superior a 0.4). Os vértices 5, 6 e 8 ficam no segmento 33 (fronteira interna). Os vértices com marcador um (1) estão na fronteira externa. Os demais estão fora de segmentos especificados, Ver figuras (2.2) e (2.3).

Figura 2.2: Representa o arquivo de saída `'quadrado.1.poly'`



Tabela 2.2: A tabela é o arquivo de saída `quadrado.1.node` com 13 nós

```

13 2 0 1
1 0 0 1
2 0 2 1
3 2 2 1
4 2 0 1
5 0 1 33
6 2 1 33
7 1 0 1
8 1 1 33
9 1.5 0.5 0
10 0.5 0.5 0
11 1 2 1
12 0.5 1.5 0
13 1.5 1.5 0
# Generated by Triangle -pq30a0.4 quadrado.poly

```

Segue o arquivo de saída `quadrado.1.poly`, ver tabelas (2.3). Nota-se que o número de vértices neste arquivo é zero (0) indicando que vértices serão lidos separadamente de um arquivo `.node`. Quanto ao arquivo saída ele, possui uma estrutura semelhantes aos outros arquivos, portanto será omitido.

Figura 2.3: A figura representa o arquivo de saída quadrado.1.ele com 16 triângulos.

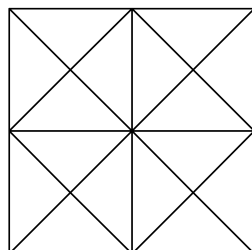


Tabela 2.3: A tabela é o arquivo de entrada quadrado.1.poly com 10 segmentos e nenhum buraco

```

0 2 0 1
10 1
1 1 5 1
2 5 2 1
3 2 11 1
4 3 6 1
5 6 4 1
6 4 7 1
7 5 8 33
8 7 1 1
9 8 6 33
10 11 3 1
0
# Generated by Triangle -pq30a0.4 quadrado.poly

```

Quando se refina uma malha, geralmente o que se quer é impor uma restrição rigorosa. Existem mais de uma maneira de se fazer isso. Um forma é usando `-q` com um ângulo grande seguido de `-a` com uma área menor do que a usada para gerar a malha. Outra forma é usar um arquivo `.area` especificando uma restrição de área máxima para cada triângulo e `-a` sem um número ao lado. Por último, pode-se usar a secção opcional do arquivo `.poly` para impor restrição de área. Uma área máxima no arquivo `.poly` é aplicada em uma região inteira. No próximo e único exemplo sobre refinamento, os arquivos de saída `.node`, `poly` e `.ele` serão omitidos.

Exemplo 2: Neste exemplo, será usado a parte opcional do arquivo `.poly`. O arquivo `quadrado2.poly`, tem como fronteira externa um quadrado com 10 nós e 3 segmentos de fronteira interna, ver tabela(2.4) e figura (2.4). Para os arquivos de saída `quadrado2.1.node` e `quadrado2.1.ele`, ver figura (2.5) e figura (2.6).

Figura 2.4: Esta figura representa o arquivo de entrada `quadrado2.poly`

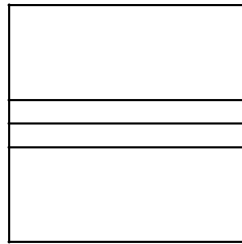


Figura 2.5: A figura mostra 371 vértices do arquivo de saída `quadrado'2.1.node`.

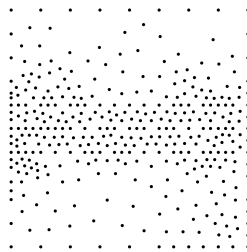


Figura 2.6: Representa os 686 triângulos do arquivo de saída `quadrado2.1.ele`.

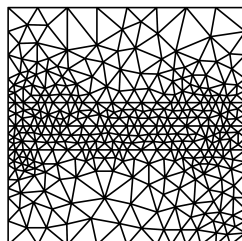


Tabela 2.4: A tabela é o arquivo de entrada quadrado2.poly com 10 vértices, 7 segmentos, sendo dois internos, nenhum buraco e 4 subregiões na secção opcional com restrição de área para duas regiões

```

# vértices
10 2 0 1
1 0.0 0.0 0
2 2.0 0.0 0
3 2.0 2.0 0
4 0.0 2.0 0
5 0.0 0.5 0
6 0.0 1.0 0
7 0.0 1.5 0
8 2.0 0.5 0
9 2.0 1.0 0
10 2.0 1.5 0
# segmentos
7 1
1 1 2 0
2 2 3 0
3 3 4 0
4 4 1 0
5 5 8 5
6 6 9 10
7 7 10 33
0 # nenhum buraco
# secção opcional
4
1 1.0 0.25 -1
2 1.0 0.75 0.005
3 1.0 1.25 0.005
4 1.0 1.75 -1

```

3 MODELAGEM NUMÉRICA

3.1 ELEMENTOS FINITOS 2D

Na maioria dos modelos matemáticos de sistemas físicos, a solução analítica costuma oferecer grandes dificuldades, ou até mesmo ser intratável. Uma alternativa de contorna a dificuldade de encontrar a solução analítica, é utilizar um método numérico que permita obter uma aproximação satisfatória da solução analítica. O MEF, é um método numérico para a resolução de equações diferenciais, sobretudo para as quais não existem solução analítica, ele pode ser entendido como derivado do método de Galerkin que é baseado no método dos resíduos ponderados, porém, com as funções pesos sendo as próprias funções base. Sendo assim, pode-se dizer que MEF é uma aplicação do método de galerkin restrito a uma classe de funções base. Portanto, do MEF para o problema descrito pela equação diferencial genérica (3.1), obtém-se uma solução aproximada a partir das projeções ortogonais de $u(x, z)$ no espaço gerado pela combinação linear das funções base no domínio Ω^e , (LAY, 2007).

$$L(u(x, z)) = h(x, z), \quad \text{para todo } (x, z) \in \Omega \quad (3.1)$$

L é um operador diferencial linear, u e h são funções de (x, z) , e $\Omega \subset R^2$ é o domínio do problema com condições de fronteira a serem especificadas.

3.2 CASO ELETROSTÁTICO

Considera-se aqui para aplicação do método de elementos finitos, o problema eletrostático de se encontrar o potencial associado a uma distribuição contínua de fonte no espaço livre. A equação de governo do sistema físico e dada pela equação de Poisson, (HAYT; BUCK, 2008),

$$-\nabla^2 u(x, z) = h(x, z), \quad (3.2)$$

no domínio $0 < x < a, 0 < z < b$, com condições de fronteira de Dirichlet homogêna.

3.2.1 Solução analítica

A solução analítica, (RIJO, 1991), é dada por uma série dupla de Fourier (3.3).

$$u(x, z) = \frac{4}{ab} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \frac{\sin \frac{m\pi x}{a} \sin \frac{n\pi z}{b}}{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \quad (3.3)$$

em que os coeficientes A_{mn} da séries são dados por

$$A_{mn} = \int_0^a \int_0^b h(x', z') \sin \frac{m\pi x'}{a} \sin \frac{n\pi z'}{b} dz' dx' \quad (3.4)$$

3.2.2 Solução numérica

Para implementação do MEF, segue-se a metodologia descrita em (RIJO, 1998), sendo assim, em primeiro lugar é necessário discretizar o domínio Ω em subdomínios triangulares Ω^e . Em seguida, deve-se definir as funções base que interpolam linearmente $u(x, z)$ em um domínio triangular arbitrário. Então, considerando um elemento genérico Ω^e da malha de elementos finitos, e considerando que a aproximação no elemento de $u(x, z)$ é dada pela função linear,

$$u^e(x, z) = \alpha + \beta x + \gamma z, \quad (3.5)$$

correspondente a um plano que intercepta $u(x, z)$ nos valores nodais (u_1^e, u_2^e, u_3^e) no elemento. Determina-se as constantes α , β e γ , montando um sistema de equações lineares a partir dos valores nodais e resolvendo pela regra de Cramer. Portanto, as constantes são dadas por:

$$\alpha = \frac{1}{2\Delta^e} \begin{vmatrix} u_1^e & x_1 & z_1 \\ u_2^e & x_2 & z_2 \\ u_3^e & x_3 & z_3 \end{vmatrix}, \quad \beta = \frac{1}{2\Delta^e} \begin{vmatrix} 1 & u_1^e & z_1 \\ 1 & u_2^e & z_2 \\ 1 & u_3^e & z_3 \end{vmatrix}, \quad (3.6)$$

$$\gamma = \frac{1}{2\Delta^e} \begin{vmatrix} 1 & x_1 & u_1^e \\ 1 & x_2 & u_2^e \\ 1 & x_3 & u_3^e \end{vmatrix}, \quad \Delta^e = \frac{1}{2} \begin{vmatrix} 1 & x_1 & z_1 \\ 1 & x_2 & z_2 \\ 1 & x_3 & z_3 \end{vmatrix}$$

em que Δ^e é a área do triângulo de vértices (x_1^e, z_1^e) , (x_2^e, z_2^e) , (x_3^e, z_3^e) . Então, calculando os determinante em (3.6) para α , β , γ , e reorganizando os termos e substituindo em (3.5), obtém-se

$$u^e(x, z) = \frac{a_1 + b_1x + c_1z}{2\Delta^e}u_1^e + \frac{a_2 + b_2x + c_2z}{2\Delta^e}u_2^e + \frac{a_3 + b_3x + c_3z}{2\Delta^e}u_3^e \quad (3.7)$$

em que,

$$\begin{aligned} a_1 &= x_2z_3 - x_3z_2, & b_1 &= z_2 - z_3, & c_1 &= x_3 - x_2, \\ a_2 &= x_3z_1 - x_1z_3, & b_2 &= z_3 - z_1, & c_2 &= x_1 - x_3, \\ a_3 &= x_1z_2 - x_2z_1, & b_3 &= z_1 - z_2, & c_3 &= x_2 - x_1, \end{aligned} \quad (3.8)$$

sendo assim, defini-se as funções base Ψ_1, Ψ_2, Ψ_3 por:

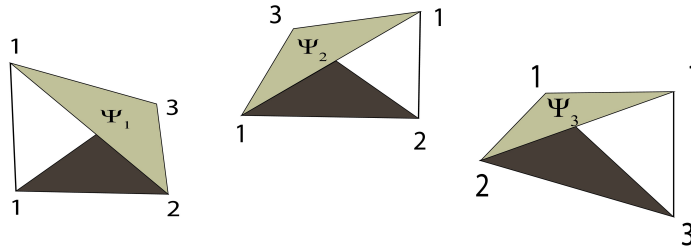
$$\begin{aligned} \Psi_1 &= \frac{1}{2\Delta^e}(a_1 + b_1x + c_1z) \\ \Psi_2 &= \frac{1}{2\Delta^e}(a_2 + b_2x + c_2z) \\ \Psi_3 &= \frac{1}{2\Delta^e}(a_3 + b_3x + c_3z) \end{aligned} \quad (3.9)$$

então, (3.7) pode ser escrita como uma combinação linear de (3.9)

$$u^e(x, z) = \sum_{i=1}^3 u_i^e \Psi_i \quad (3.10)$$

em que (3.9), tem as características descrita na figura (3.1).

Figura 3.1: A figura mostra que a função base Ψ_i em um nó local j é 1 se $i = j$ e 0 se $i \neq j$.



Portanto, considerando a aproximação da equação (3.2) em Ω^e , têm-se

$$-\left(\frac{\partial^2 u^e}{\partial x^2} + \frac{\partial^2 u^e}{\partial z^2}\right) - h(x, z) = \epsilon \quad (3.11)$$

Uma vez obtidas as funções base, aplica-se o critério de Galerkin em (3.11) para cada elemento Ω^e

$$- \int_{\Omega^e} \Psi_i \left(\frac{\partial^2 u^e}{\partial x^2} + \frac{\partial^2 u^e}{\partial z^2} + h(x, z) \right) dx dz = \int_{\Omega^e} \Psi_i \epsilon dx dz = 0 \quad (3.12)$$

e pela regra do produto da derivada de duas funções , obtem-se as seguinte identidades

$$\Psi_i \frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x} \left(\Psi_i \frac{\partial u}{\partial x} \right) - \frac{\partial \Psi_i}{\partial x} \frac{\partial u}{\partial x} \quad (3.13)$$

$$\Psi_i \frac{\partial^2 u}{\partial z^2} = \frac{\partial}{\partial z} \left(\Psi_i \frac{\partial u}{\partial z} \right) - \frac{\partial \Psi_i}{\partial z} \frac{\partial u}{\partial z}$$

substituindo (3.13) em (3.12), obtem-se

$$\begin{aligned} & \int_{\Omega^e} \left(\frac{\partial \Psi_i}{\partial x} \frac{\partial u^e}{\partial x} + \frac{\partial \Psi_i}{\partial z} \frac{\partial u^e}{\partial z} \right) dx dz \\ & - \int_{\Omega^e} \frac{\partial}{\partial x} \left[\Psi_i \left(\frac{\partial u^e}{\partial x} \right) + \frac{\partial}{\partial z} \left(\Psi_i \frac{\partial u^e}{\partial z} \right) \right] dx dz \\ & = \int_{\Omega^e} \Psi_i h(x, z) dx dz \end{aligned}$$

aplicando o teorema de Green na segunda integral do lado esquerdo,

$$\begin{aligned} \int_{\Omega^e} \frac{\partial}{\partial x} \left[\Psi_i \left(\frac{\partial u^e}{\partial x} \right) + \frac{\partial}{\partial z} \left(\Psi_i \frac{\partial u^e}{\partial z} \right) \right] dx dz &= \int_{\partial \Omega^e} -\Psi_i \frac{\partial u^e}{\partial z} dx + \Psi_i \frac{\partial u^e}{\partial x} dz \\ &= \int_{\partial \Omega^e} \Psi_i \nabla u^e \cdot \hat{n} dl \end{aligned}$$

e substituindo, resulta

$$\int_{\partial \Omega^e} \left(\frac{\partial \Psi_i}{\partial x} \frac{\partial u^e}{\partial x} + \frac{\partial \Psi_i}{\partial z} \frac{\partial u^e}{\partial z} \right) dx dz = \int_{\Omega^e} \Psi_i h(x, z) dx dz + \int_{\partial \Omega^e} \Psi_i \nabla u^e \cdot \hat{n} dl$$

A segunda integral do lado direito representa o fluxo do campo $(\Psi_i \nabla U^e)$ em $\partial \Omega^e$, considerando que o fluxo na borda de elementos adjacentes estão em sentidos opostos, então estas integrais se cancelam, sendo assim, a integral é diferente de zero somente para os elementos com bordas que acontecem na fronteira da triangulação.

Uma vez substituindo (3.10) nas integrais acima, têm-se a matriz e o vetor fonte do elemento Ω^e dadas por:

$$\sum_{j=1}^3 \left[\int_{\Omega^e} \left(\frac{\partial \Psi_i}{\partial x} \frac{\partial \Psi_j}{\partial x} + \frac{\partial \Psi_i}{\partial z} \frac{\partial \Psi_j}{\partial z} \right) dx dz \right] u_j^e = \int_{\Omega^e} \Psi_i h(x, z) dx dz + \int_{\partial \Omega^e} \Psi_i \nabla u^e \cdot \hat{n} dl$$

para $i = 1, 2, 3$. Representado a expressão acima na forma matricial,

$$K_{ij}^e u_j^e = h_i^e \quad i, j = 1, 2, 3 \quad (3.14)$$

em que

$$K_{ij}^e = \int_{\Omega^e} \left(\frac{\partial \Psi_i}{\partial x} \frac{\partial \Psi_j}{\partial x} + \frac{\partial \Psi_i}{\partial z} \frac{\partial \Psi_j}{\partial z} \right) dx dz \quad (3.15)$$

e o vetor fonte

$$h_i^e = \int_{\Omega^e} \Psi_i h(x, z) dx dz + \int_{\partial \Omega^e} \Psi_i \nabla u^e \cdot \hat{n} dl, \quad (3.16)$$

Substituindo (3.9) em (3.15) e observando que

$$\frac{\partial \Psi_i}{\partial x} = b_i, \quad \frac{\partial \Psi_i}{\partial z} = c_i \dots,$$

a matriz de elemento tem a forma:

$$K_{ij}^e = \frac{1}{4\Delta^e} \begin{pmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{pmatrix} \quad (3.17)$$

Para o vetor fonte, considera-se apenas a primeira integral de (3.16), pois, uma vez que impostas as condições de fronteira, não há contribuição da segunda integral no vetor fonte global. Agora considerando que a primeira integral possa ser aproximada por

$$h_i^e = \int_{\Omega^e} \Psi_i (h_1^e \Psi_1 + h_2^e \Psi_2 + h_3^e \Psi_3) dx dz$$

e usando a identidade,

$$\int_{\Omega^e} \Psi_1^\alpha \Psi_2^\beta \Psi_3^\gamma dx dz = 2\Delta^e \frac{\alpha! \beta! \gamma!}{(\alpha + \beta + \gamma + 2)} \quad (3.18)$$

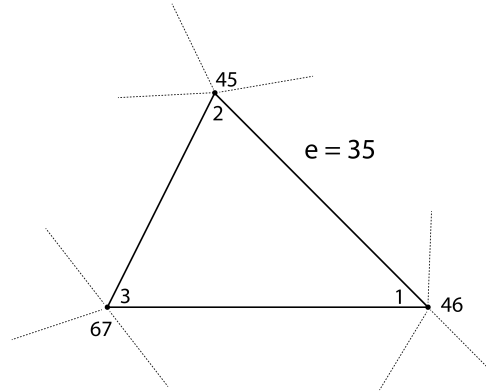
o vetor fonte do elemento tem a forma:

$$h_i^e = -\frac{\Delta^e}{12} \begin{pmatrix} 2h_1^e + h_2^e + h_3^e \\ h_1^e + 2h_2^e + h_3^e \\ h_1^e + h_2^e + 2h_3^e \end{pmatrix} \quad (3.19)$$

Monta-se o sistema de equações referente a matriz global relacionando os valores K_{ij} de cada elemento. Na montagem, utiliza-se a função de conectividade g_i^e que correlaciona os nós de cada elemento aos nós globais da malha, (SOBRINHO, 2006).

Considerando o elemento $e = 35$ de uma malha hipotética, ver figura 3.2, a função de conectividade é definida como

Figura 3.2: Elemento qualquer de uma malha hipotética com numeração arbitrária.



$$g_1^{35} = 46, g_2^{35} = 45, g_3^{35} = 67$$

sendo assim, adiciona-se K_{ij}^e a matriz global K pela correspondência g_i^{35} , logo, para o elemento $e = 35$, tem-se

$$K_{(g_i^{35})(g_j^{35})} = K_{(g_i^{35})(g_j^{35})} + K_{ij}^{35}$$

procede-se de forma análoga para o vetor fonte global. Portanto, o sistema global pode ser representado por

$$\mathbf{KU} = \mathbf{h} \quad (3.20)$$

ou seja,

$$\begin{pmatrix} K_{11} & \cdot & \cdot & \cdot & K_{1n} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \\ K_{n1} & \cdot & \cdot & \cdot & K_{nn} \end{pmatrix} \begin{pmatrix} U_1 \\ \cdot \\ \cdot \\ \cdot \\ U_n \end{pmatrix} = \begin{pmatrix} h_1 \\ \cdot \\ \cdot \\ \cdot \\ h_n \end{pmatrix} \quad (3.21)$$

em que n é o número de variáveis do sistema correspondente aos n nós globais da malha. A solução do sistema foi obtida por uma rotina baseada no método de Gauss-Compacto , (FRANCO, 2006).

3.3 RESULTADOS

As figuras que seguem representam soluções da equação de Poisson e o domínio sobre o qual as soluções foram calculadas. As curvas que representam as soluções e o erro relativo ao longo de $z = 1$ são mostradas nas figuras (3.7), (3.8) e (3.9) sendo os dois últimos resultados obtidos por dois refinamentos consecutivos do arquivo quadrado.poly.

Figura 3.3: A figura representa o arquivo de entrada .poly, ele ilustra a geometria do domínio onde está sendo calculada a solução.



Figura 3.4: A figura representa o arquivo de saída .ele com discretização do domínio em 1372 nós e 2605 elementos, ele ilustra o último refinamento sobre o .poly de entrada figura(3.3)

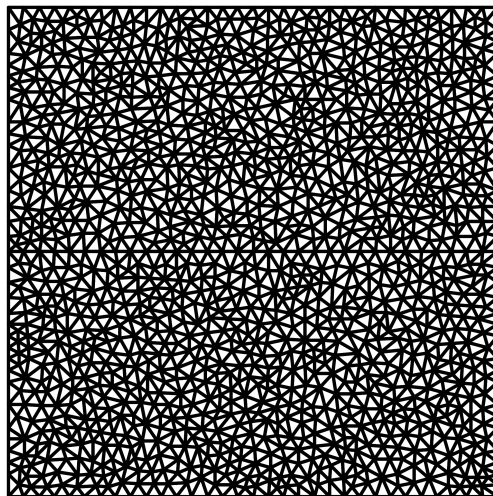


Figura 3.5: Solução numérica da equação de Poisson sobre a malha da figura(3.4) com discretização do domínio em 2605 elementos.

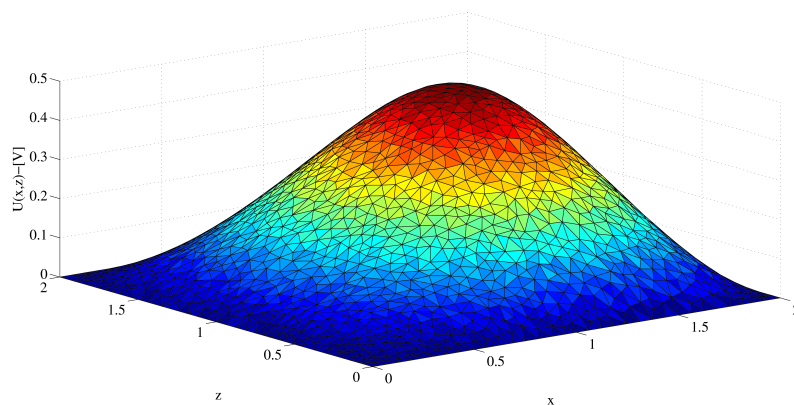
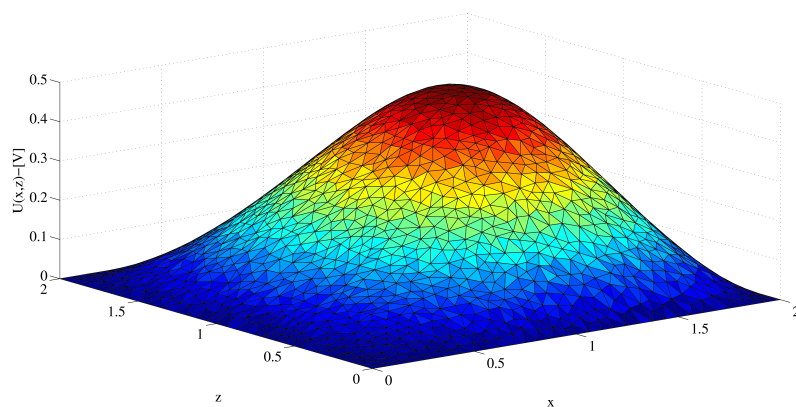
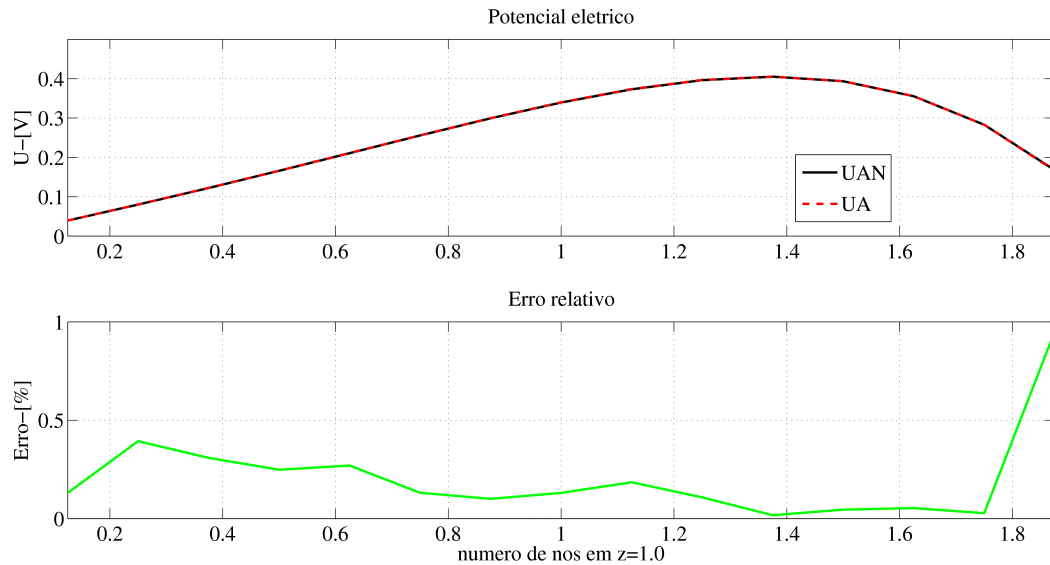


Figura 3.6: Solução analítica da equação de Poisson sobre a malha da figura(3.3) com discretização do domínio em 2605 elementos.



As figura(3.5) e figura(3.6) exibem a solução numérica e analítica da equação de Poisson calculada sobre a malha da figura(3.4) respectivamente. A malha foi gerada por - Triangle -pq32a0.0025 quadrado.poly.

Figura 3.7: A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.



O malha sobre a qual foram calculadas as soluções das curvas nas figuras (3.7), (3.8) e (3.9) foram obtidas pelo comando, Triangle -pq32a0.01 quadrado.poly, e variando o parâmetro referente a restrição de área em 0.005 e 0.0025 respectivamente. Para cada comando, obteve-se uma malha com 353, 704 e 1372 nós respectivamente. As curvas estão ao longo da reta $z = 1.0$ no domínio Ω , a partir delas é possível notar que, a aproximação da solução numérica em relação a analítica é maior a medida que a discretização é refinada devido a diminuição do erro relativo em cada refinamento. Nota-se que a figura (3.9) representa os potenciais das figuras (3.5) e (3.6) em $z = 1$. Portanto, embora as duas superfícies pareçam idênticas, o erro relativo mostra quão próximo UA está de UAN.

Figura 3.8: A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.

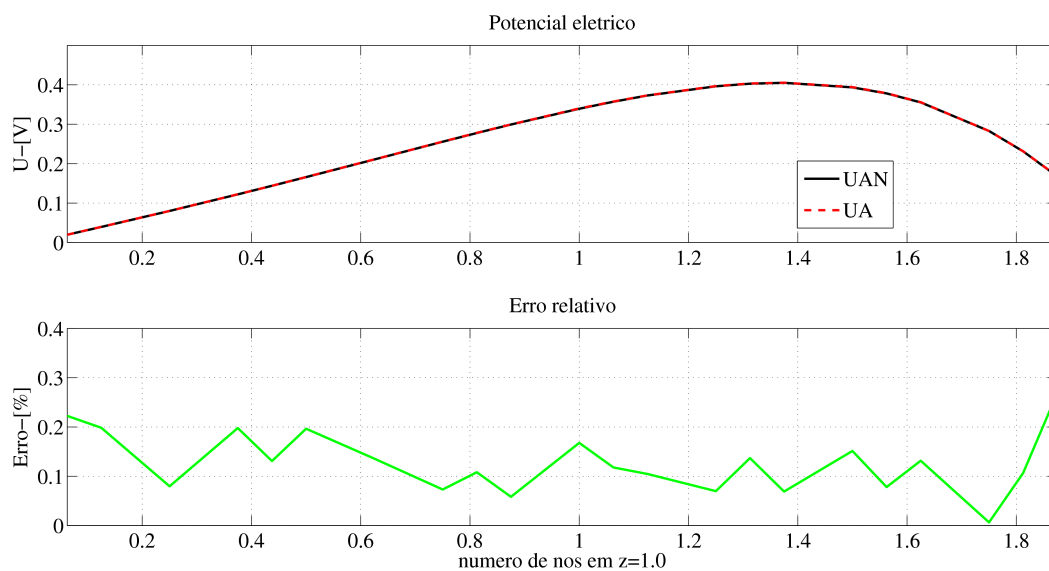
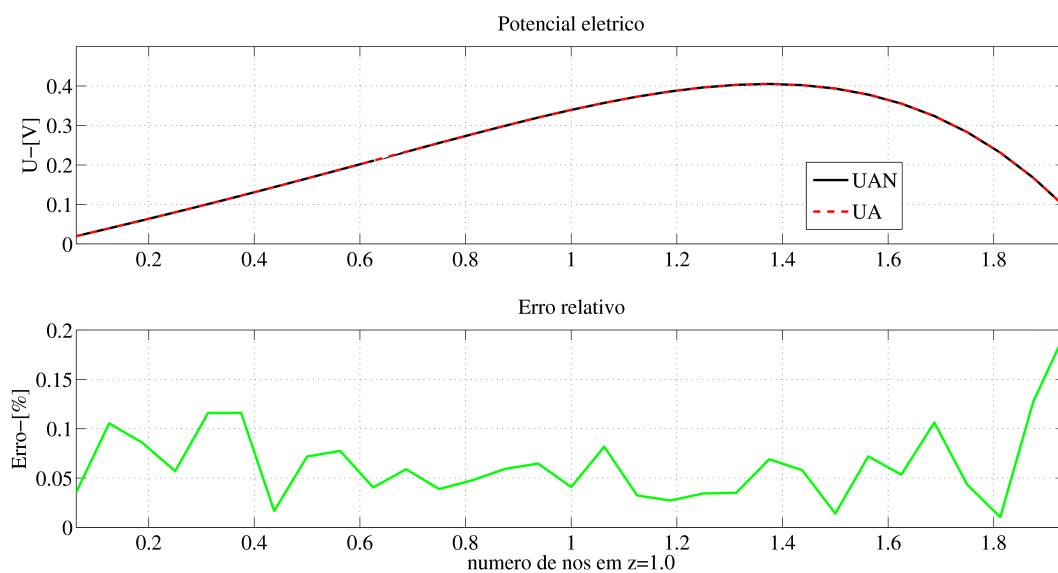


Figura 3.9: A figura mostra o resultado do potencial a partir da solução analítica (UAN) e numérica (UA), assim como o erro relativo de UA.



4 RECUPERAÇÃO DE DERIVADAS

4.1 INTRODUÇÃO

Depois de se obter uma solução de elementos finitos, em geral, deseja-se recuperar as derivas do campo encontrado em cada nó da malha, seja ele escalar ou vetorial. Existem algumas formas de se fazer isso, uma delas é usar as derivadas das funções de interpolação, que será descrito na próxima secção. Entretanto, busca-se fazer isso de uma forma mais geral, ou seja, independente da geometria do elemento, seja ela triangular, quadricular, hexagonal e etc. Para um método mais geral, usa-se uma técnica descrita em, (TABBARA, 1994).

4.2 DERIVADA ANALÍTICA

Para o problema da equação de Poisson com condições de fronteira de Dirichlet homogênea em que a solução é dada por,

$$u(x, z) = \frac{4}{ab} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} \frac{\sin \frac{m\pi x}{a} \sin \frac{n\pi z}{b}}{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}$$

as derivadas direcionais são,

$$\frac{\partial u(x, z)}{\partial x} = \frac{4\pi}{a^2 b} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} m \frac{\cos \frac{m\pi x}{a} \sin \frac{n\pi z}{b}}{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}$$

$$\frac{\partial u(x, z)}{\partial z} = \frac{4\pi}{ab^2} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{mn} n \frac{\sin \frac{m\pi x}{a} \cos \frac{n\pi z}{b}}{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}$$

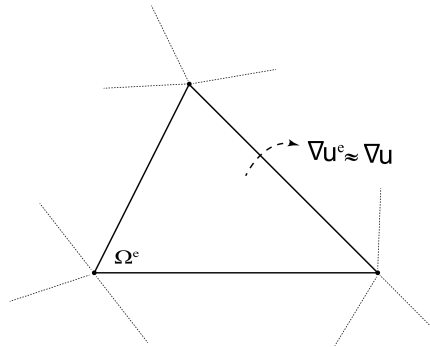
4.3 MÉDIA ARITMÉTICA DO GRADIENTE DAS FUNÇÕES BASE

Obter as derivadas direcionais em cada nó da malha através da derivada do elemento consiste em derivar a função u^e . lembrando que

$$u^e(x, z) = \sum_{i=1}^3 u_i^e \Psi_i$$

e Como $u^e(x, z)$ é a aproximação de $u(x, z)$ em Ω_e , logo, $\nabla u(x, z) \approx \nabla u^e(x, z)$ para $(x, z) \in \Omega_e$, desde que a discretização seja pequena o suficiente, tal que o plano que interpola a solução nos valores nodais possa representar satisfatoriamente $u(x, z)$, como ilustra a figura(4.1),

Figura 4.1: domínio local arbitrário de uma malha hipotética



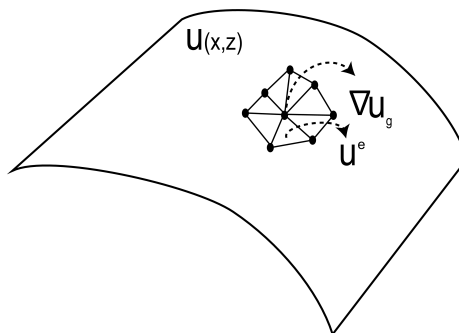
uma vez que,

$$\frac{\partial u^e(x, z)}{\partial x} = \frac{1}{2\Delta^e} \sum_{i=1}^3 b_i u_i^e \quad (4.1)$$

$$\frac{\partial u^e(x, z)}{\partial z} = \frac{1}{2\Delta^e} \sum_{i=1}^3 c_i u_i^e \quad (4.2)$$

então para determinar o $\nabla u_g(x, z)$ em cada nó da malha a partir de 4.1 e 4.2, considera-se uma média tomando o gradiente de cada elemento na vizinha do nó, ver figura (4.2),

Figura 4.2: A figura ilustra as interpolações na vizinhança de um nó qualquer da malha onde a derivada será calculada



portanto,

$$\frac{\partial u_g(x, z)}{\partial x} = \sum_{j=1}^{N_e} \frac{\partial u^e(x, z)}{\partial x} \quad (4.3)$$

$$\frac{\partial u_g(x, z)}{\partial z} = \sum_{j=1}^{N_e} \frac{\partial u^e(x, z)}{\partial z} \quad (4.4)$$

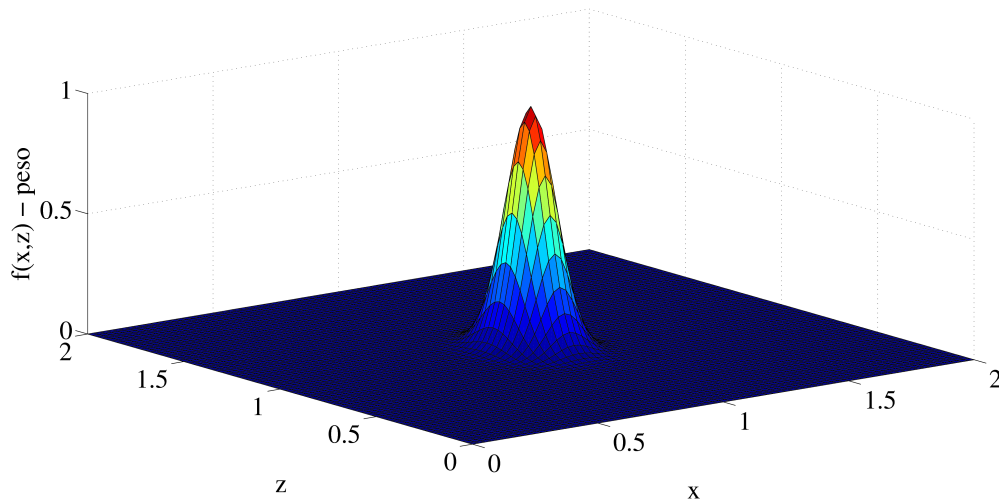
em que o subscrito g indica o nó no qual está sendo calculado a derivada direcional e N_e o número de elementos vizinhos a este nó

4.4 AJUSTE MÓVEL POR MÍNIMOS QUADRADOS PONDERADOS

Primeiramente, considera-se um problema de condições de fronteira com domínio Ω e contorno $\partial\Omega$, em que é discretizado por elementos finitos. A figura (4.4) ilustra e reforça a independência da geometria do elemento. Para calcular as derivadas em qualquer ponto da malha por ajuste móvel, defini-se um sub-domínio $\Gamma_g \in \Omega$, tendo a origem em (x_g, z_g) e raio R_g . Nota-se que somente os nós em Γ_g contribuem para o processo de ajuste móvel. Portanto, a idéia é resolver um problema de mínimos quadrados em cada sub-domínio Γ_g utilizando uma função peso, ver figura(), dado por (4.5), (TABBARA, 1994).

$$w(x, z) = \exp -c^2 \left[\left(\frac{x_g - x_i}{x_g - x_m} \right)^2 + \left(\frac{z_g - z_i}{z_g - z_m} \right)^2 \right] \quad (4.5)$$

Figura 4.3: A figura mostra a função peso, em que $x_g = 1.0, z_g = 1.0, |x_g - x_m| = 0.3$ e $|z_g - z_m| = 0.4$. Portanto, o peso é maior na origem onde se está calculando a derivada.



com $x_m = \text{Max}|x_i|$ e $z_m = \text{Max}|z_i|$, em que x_i, z_i ($i = 1, \dots, N$) são os nós no domínio Γ_g . $w(x, z)$ é uma função contínua e positiva que decresce monotonicamente a partir da origem. Nota-se que não há restrição sobre (x_g, z_g) , mas particularmente para os resultados que serão apresentados, refere-se ao nó da malha onde está sendo calculado a derivada.

Com o intuito de realizar uma interpolação suave no domínio local Γ_g , a solução $u_e(x, z)$ de elementos finitos interpolada em Γ_g é definida como,

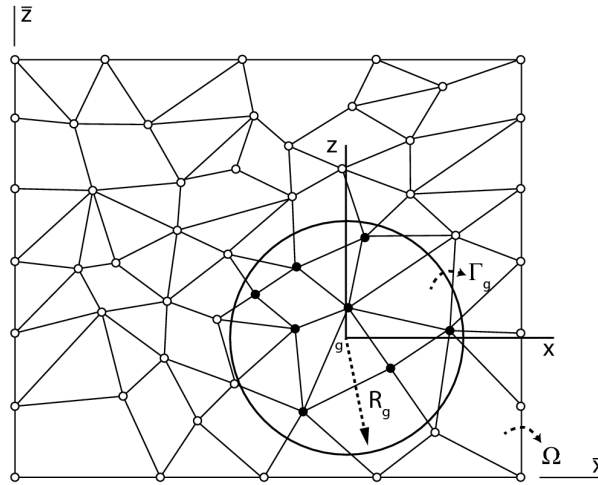
$$u_e^*(x, z) = \mathbf{f}(x, z) \mathbf{a} \quad \text{para todo ponto } (x, z) \in \Gamma_g$$

em que o produto do lado direito, refere-se a base quadrática

$$\mathbf{f} = [1 \quad x \quad z \quad xz \quad x^2 \quad z^2] \quad (4.6)$$

$$\mathbf{a}^T = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6] \quad (4.7)$$

Figura 4.4: figura de uma malha hipotética ilustrando que o método independe da forma que o domínio é discretizado e que não há restrição para origem do domínio local.



Para o problema de mínimos quadrado, deseja-se encontrar \mathbf{a} que minimize o vetor resíduo ϵ em

$$\sum_{i=1}^N \mathbf{u}_i^e - \sum_{i=1}^N \mathbf{f}_i \mathbf{a} = \epsilon \quad (4.8)$$

sendo assim, adicionando o peso w_i em (4.9) e considerando que

$$\left(\sum_{i=1}^N \mathbf{f}_i^T \right) \boldsymbol{\epsilon} = 0$$

tem-se

$$\sum_{i=1}^N \mathbf{f}_i^T \left(\sum_{i=1}^N w_i \mathbf{u}_i^e - \sum_{i=1}^N w_i \mathbf{f}_i \mathbf{a} \right) = \sum_{i=1}^N w_i \mathbf{f}_i^T \boldsymbol{\epsilon} = 0$$

$$\sum_{i=1}^N w_i \mathbf{f}_i^T \mathbf{u}_i^e = \sum_{i=1}^N w_i \mathbf{f}_i^T \mathbf{f}_i \mathbf{a}$$

$$b_i = M_i \mathbf{a} \tag{4.9}$$

$$\mathbf{a} = M_i^{-1} b_i \tag{4.10}$$

em que

$$M_i = \sum_{i=1}^N w_i \mathbf{f}_i^T \mathbf{f}_i, \quad b_i = \sum_{i=1}^N w_i \mathbf{f}_i^T \mathbf{u}_i^e \tag{4.11}$$

Observa-se que (4.11) só terá solução única se tiver pelo menos N^c nós em Γ_g , em que N^c é o número de coeficientes em \mathbf{a} . Além disso, é necessário controlar a densidade dos nós em Γ_g para manter uma representatividade local de $u_e(x, z)$ em volta de (x_g, z_g) . Como Γ_g é um círculo de área S_Γ e raio R_g e será definido para ter relação com um elemento h no domínio local, nota-se que essa relação não é linear, porque se fosse assim definido, o raio aumentaria com o quadrado de h e diminuiria com o inverso do quadrado, ou seja, para $h > 1$, a representatividade local de $u_e(x, z)$ é menor quanto maior for h e para $h < 1$ a unicidade de (4.11) não é garantida, sendo assim

$$R_g(n, h_{min}) = \alpha(n, h_{min})^{1/2} h_{min}$$

em que

$$\alpha(n, h_{min}) = \frac{n}{\pi h_{min}}$$

sendo $n = S_{\Gamma}/h_{min}$ e como as derivadas são calculadas em cada nó da malha, h_{min} é opcionalmente a área do menor elemento na vizinhança desse nó. Em todos os resultados n é tomado igual a 7.0.

Uma vez encontrado o menor elemento na vizinhança do nó onde se quer calcular a derivada, defini-se o raio do domínio local de acordo com a expressão acima. Para mapear os nós que se encontram dentro do domínio local, calcula-se a distância dos nós à origem do domínio para cada nó no arquivo .node correspondente, portanto, um nó arbitrário da malha de elementos finitos estará dentro do domínio local se a distância para origem do domínio for menor que R_g .

4.5 RESULTADOS

As derivadas numéricas para as componentes x e z do campo elétrico serão obtidas a partir das soluções analítica e numérica. Os resultados que serão mostrados, assim como no capítulo anterior estarão ao longo da reta $z = 1$ na malha de elementos finitos, ver figura(2.4).

Figura 4.5: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos

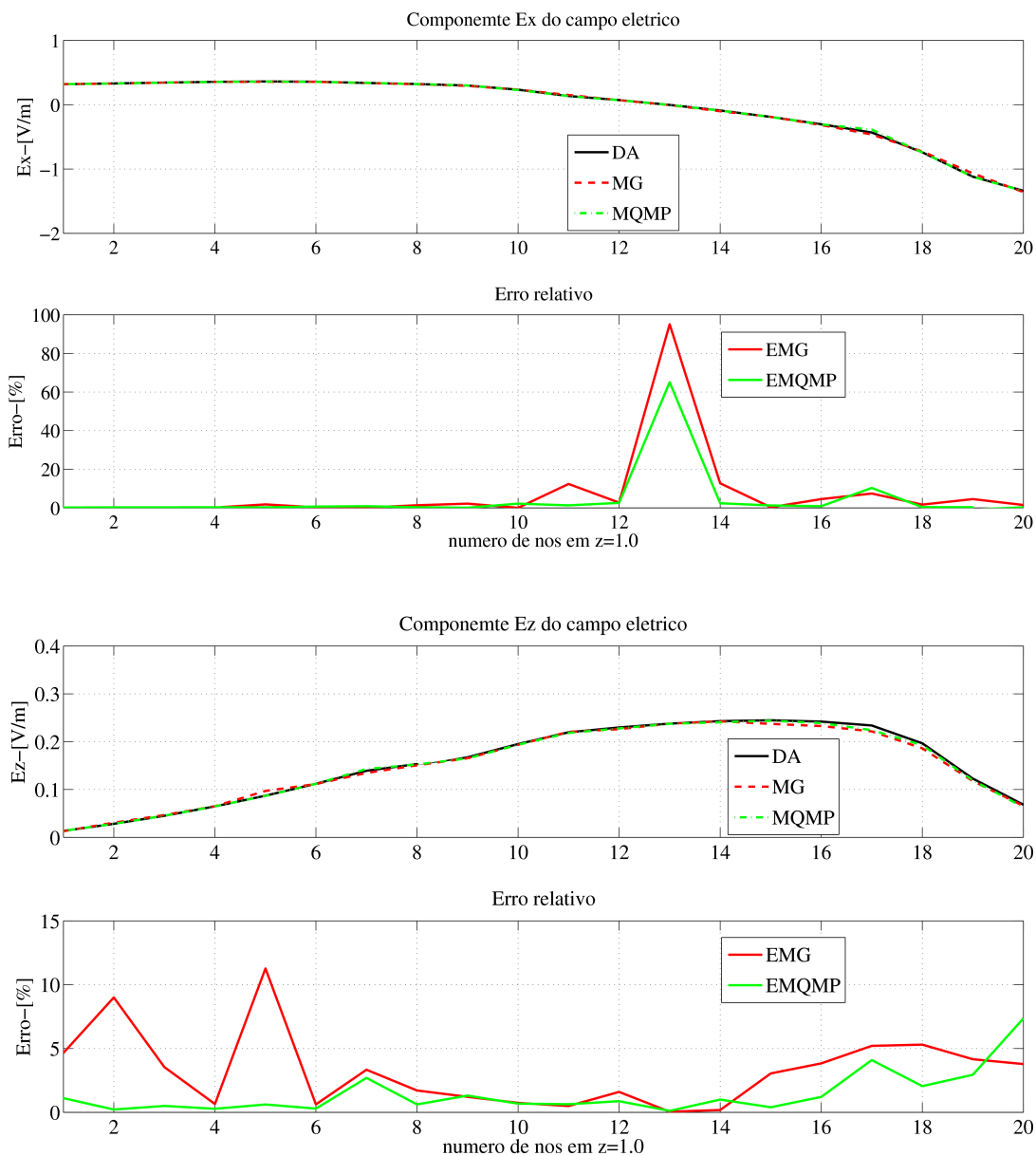


Figura 4.6: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos

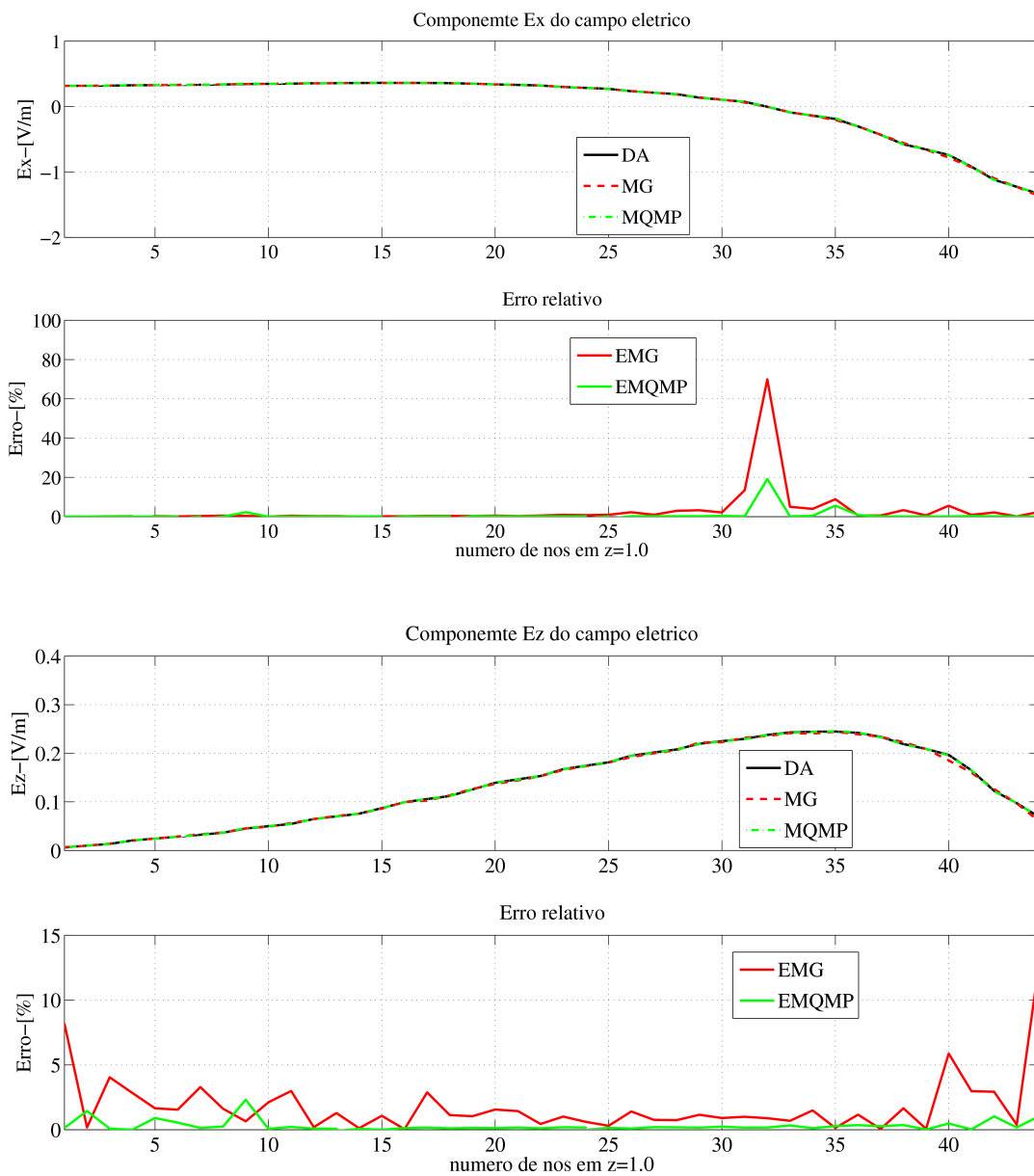


Figura 4.7: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos

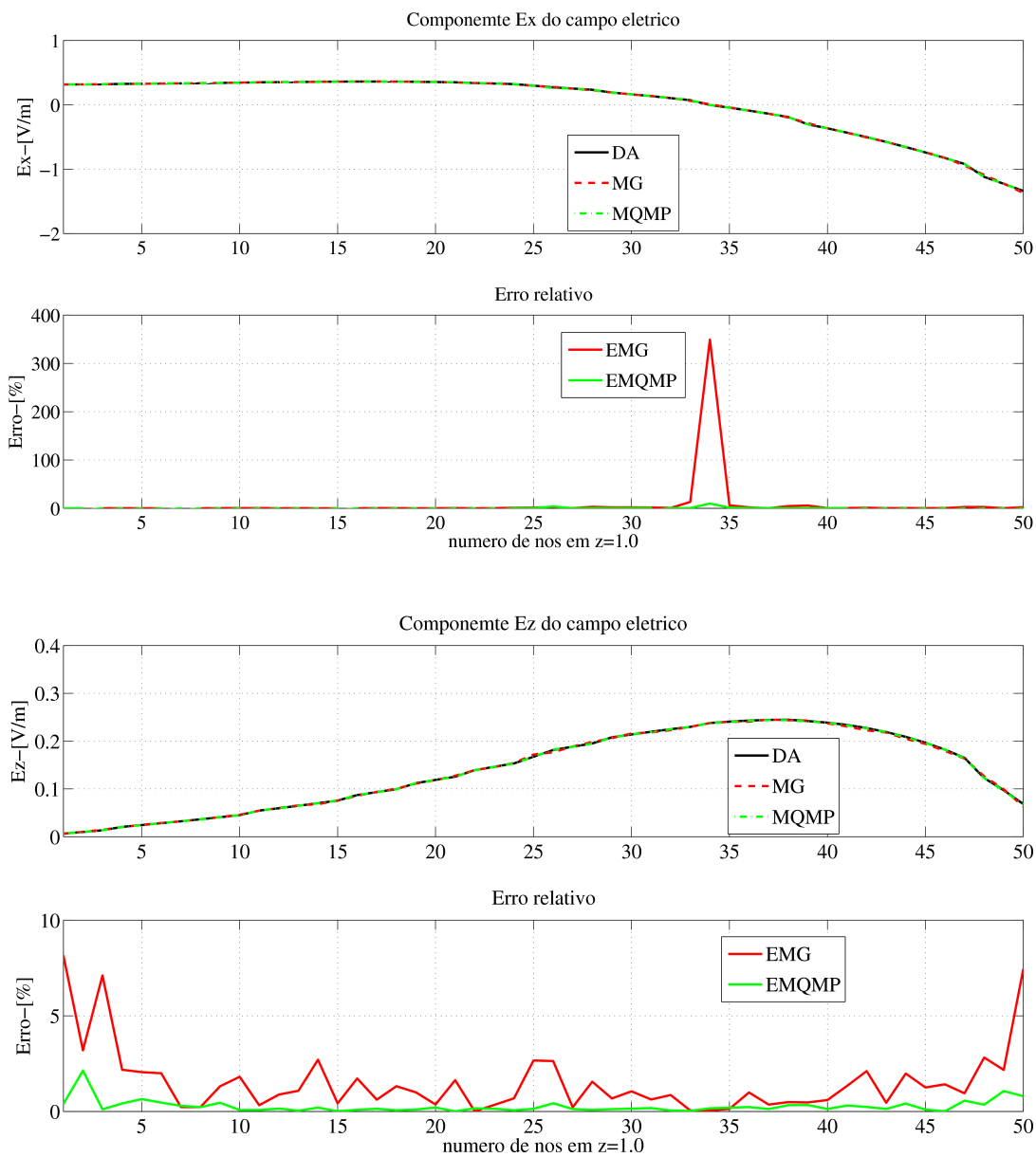
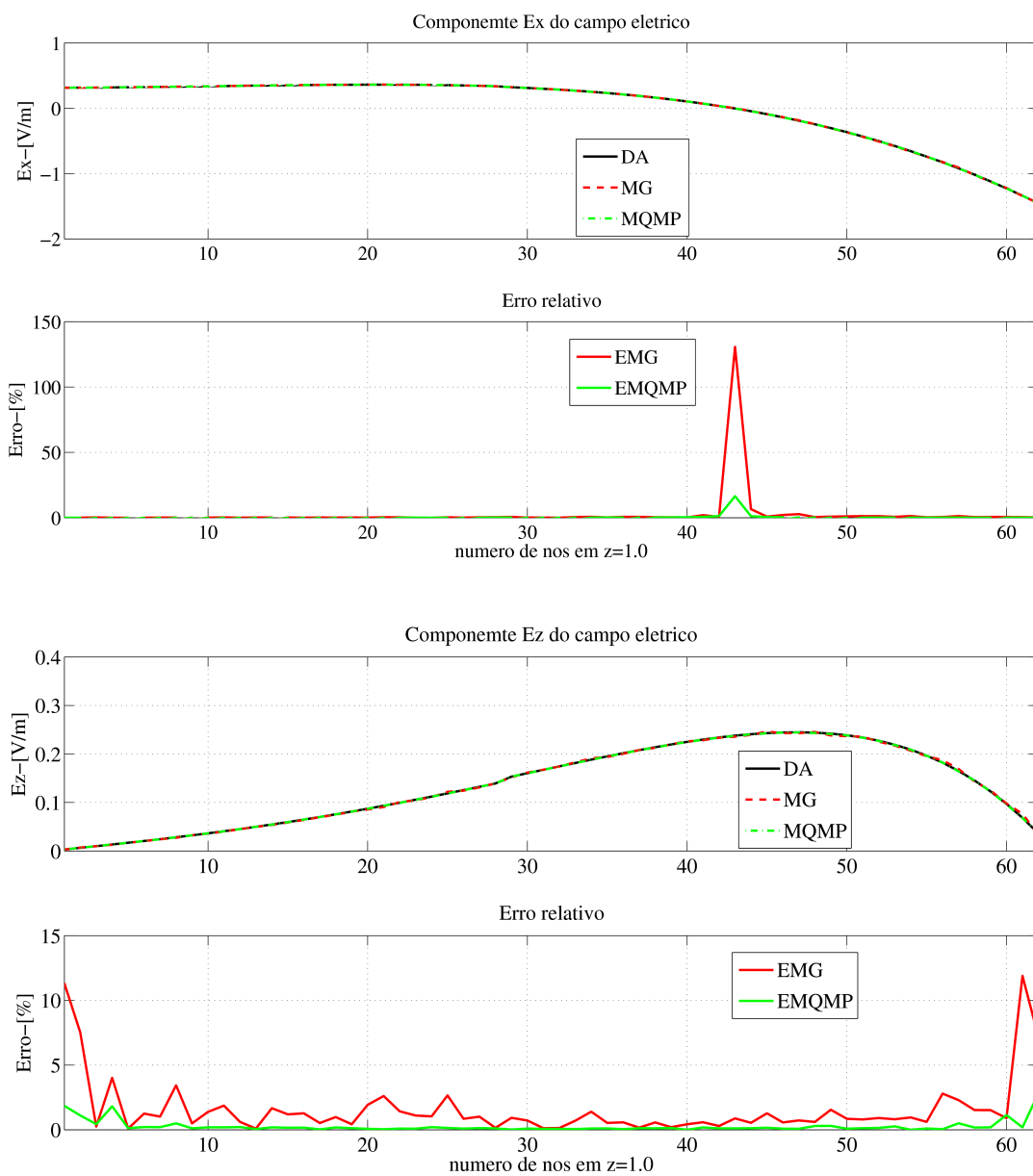


Figura 4.8: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente. EMG e EMQMP são seus erros relativos



Os resultados mostrados referentes as figuras (4.5), (4.6), (4.7) e (4.8) foram obtidos a partir de sucessivos refinamentos na malha da figura (2.4). Os refinamentos, diferente do capítulo três, foram obtidos usando a secção opcional do arquivo .poly, portanto, sem nenhum parâmetro acompanhando o interruptor -a, e variando a restrição de ângulo mínimo. Com isso foi possível obter um elevado número de nós ao longo da reta $z = 1.0$, sem aumentar exorbitantemente a quantidade de nós na malha. As discretizações foram realizadas com as seguintes sintaxes: Triangle -pq30a quadrado2.poly, com restrições de área de 0.005 e 0.00125, Triangle -pq32a... e Triangle -pq34a..., com restrições de área de 0.00125.

A solução analítica da equação (3.2) é usada para avaliar somente o erro introduzido pelo cálculo das derivadas. Observando os resultados das figuras (4.5), (4.6), (4.7) e (4.8), nota-se que o ajuste móvel por mínimos quadrados ponderados é mais eficiente que a média dos gradientes, este fato é traduzido pela redução do erro relativo a medida que a malha é refinada. Entretanto, ambos os métodos mostraram debilidades. Observou-se que as derivadas calculadas em regiões perto da fronteira e onde o campo é zero ou próximo de zero, apresentam erros relativos elevados. Contudo, mesmo para essas regiões, o ajuste móvel exibe uma aproximação mais efetiva e mais regular, no sentido de que o módulo da diferença do erro relativo entre nós vizinhos são em geral menores com este método.

Os resultados das figuras (4.9), (4.10), (4.11) e (4.12) também foram obtidos através dos mesmos refinamentos sobre malha da figura (2.4). Contudo, a solução analítica foi substituída pela numérica, portanto, qualquer erro acrescido no cálculo das derivadas está associado ao erro introduzido pela solução numérica. Todavia, observa-se que embora os erros relativos estejam amplificados, novamente as derivadas numéricas se aproximam da analítica a medida que a malha é refinada para ambos os métodos, e que o ajuste móvel continua apresentando resultados menos discrepantes.

Figura 4.9: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos

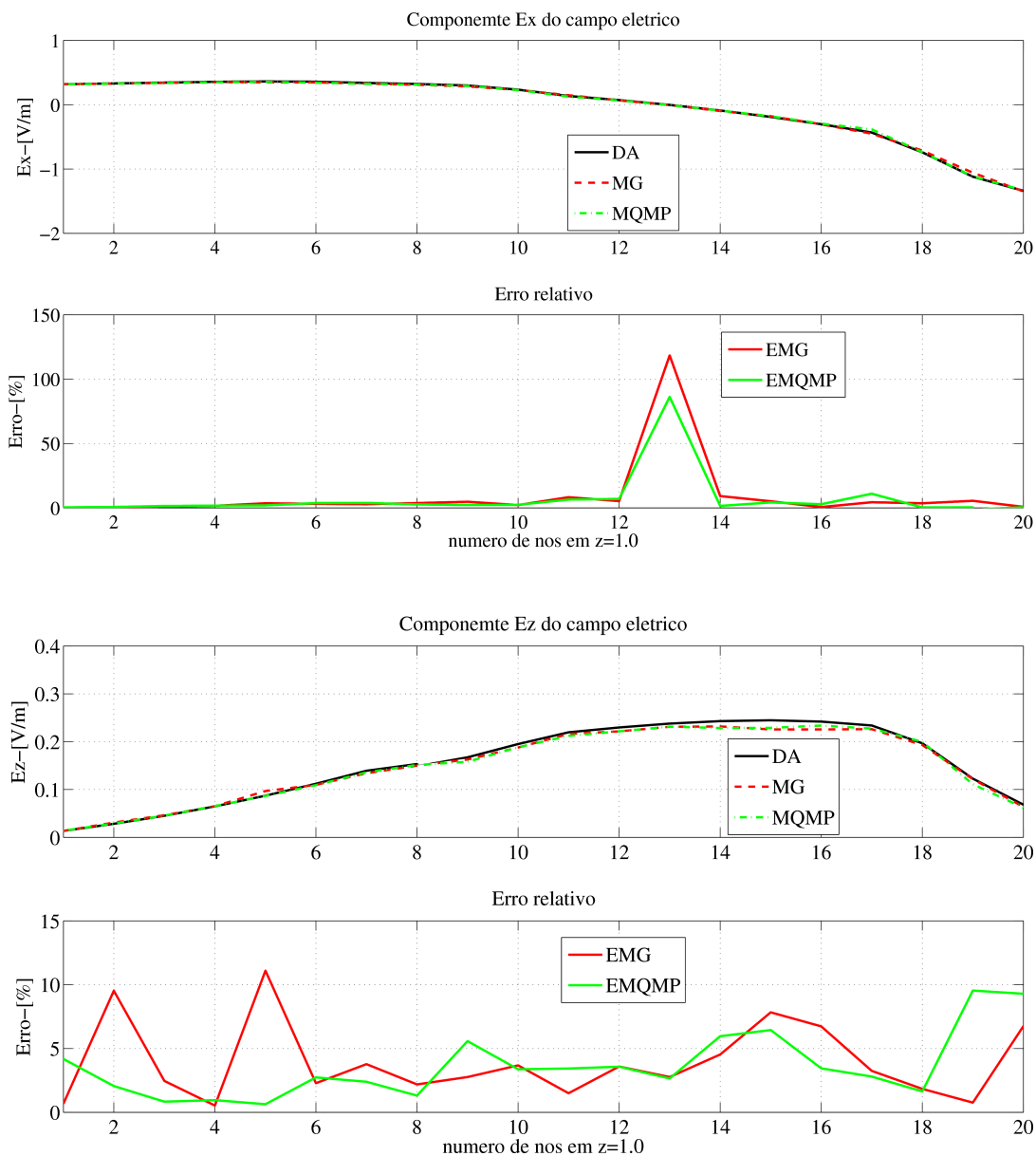


Figura 4.10: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos

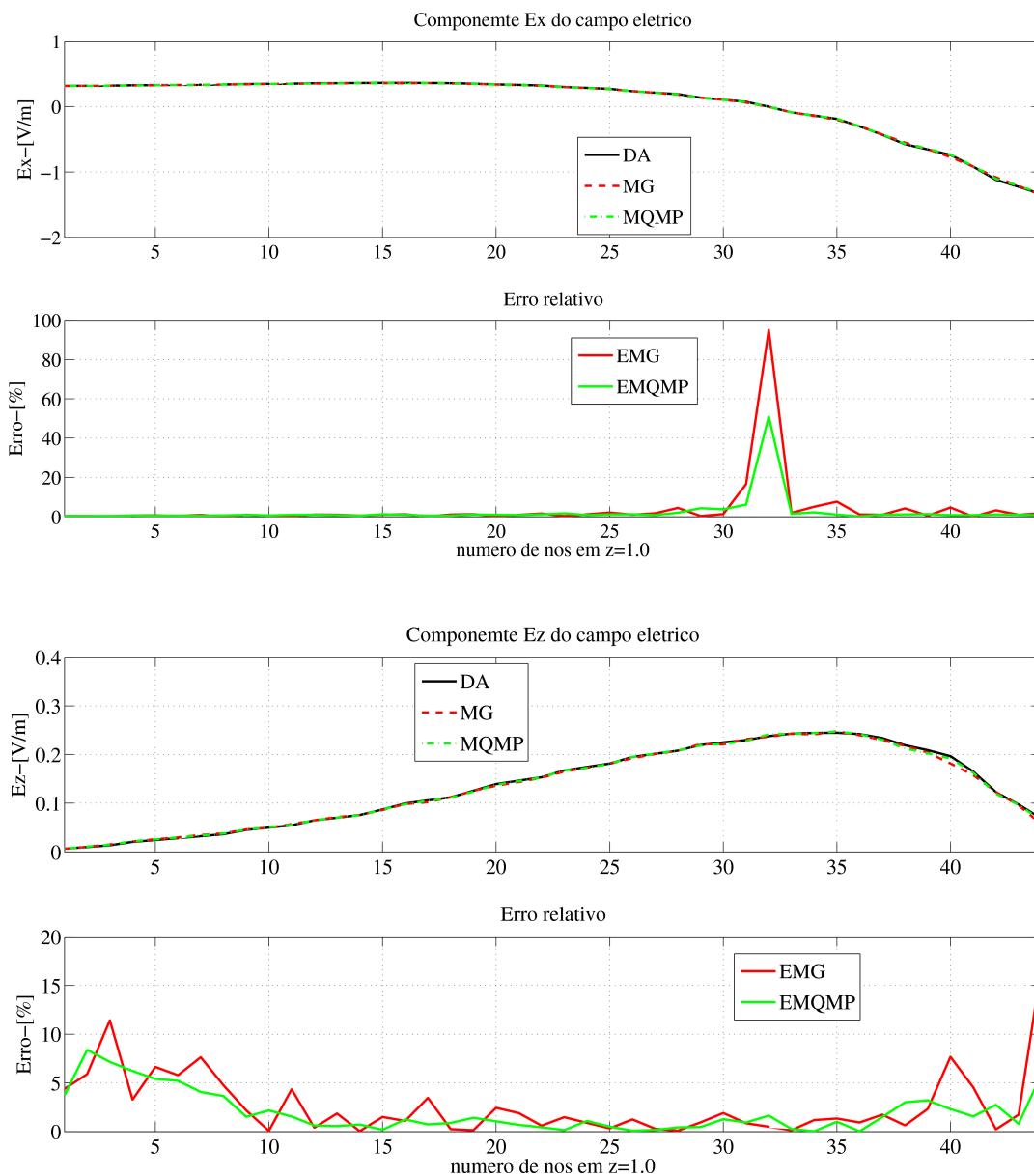


Figura 4.11: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos

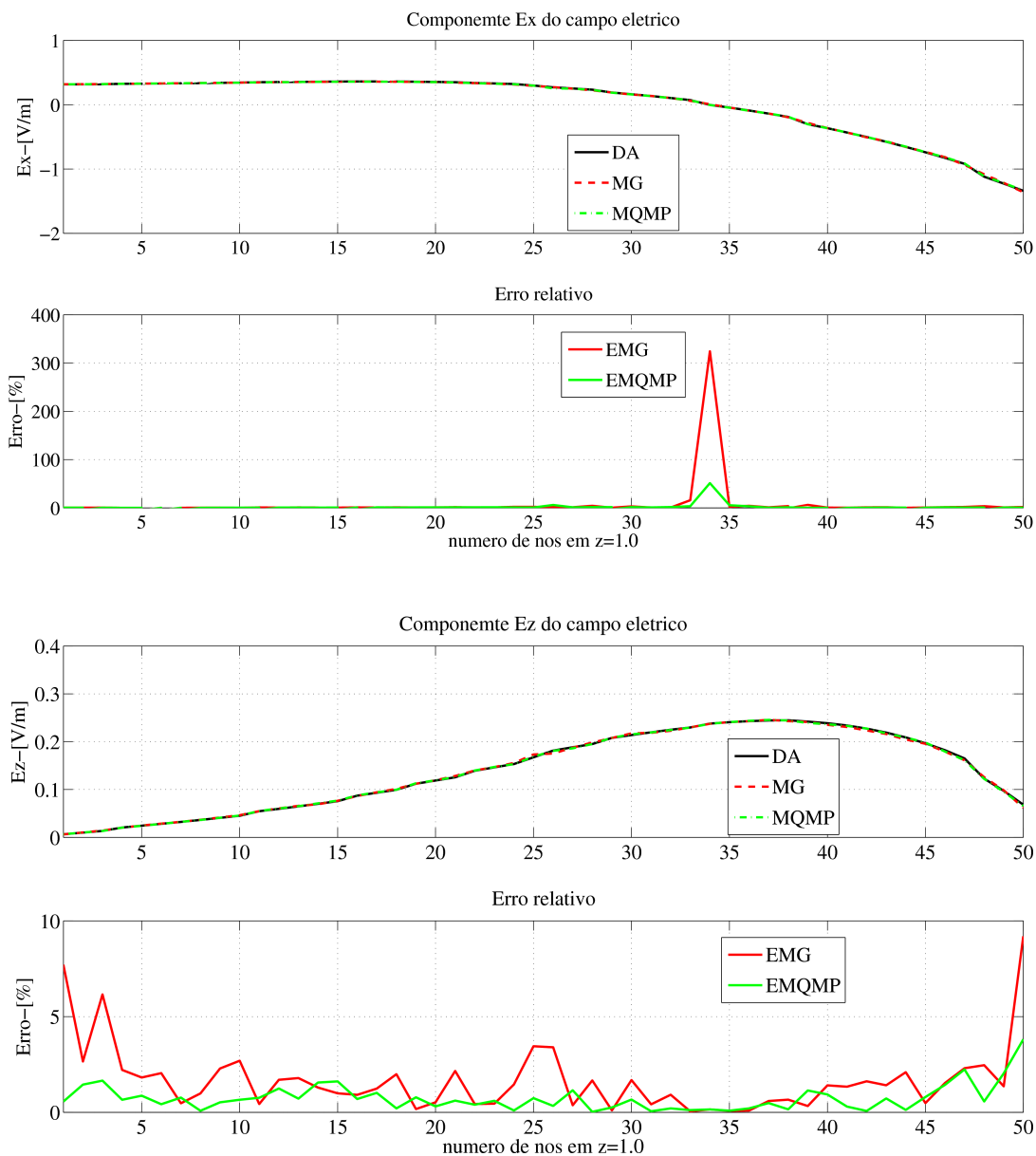
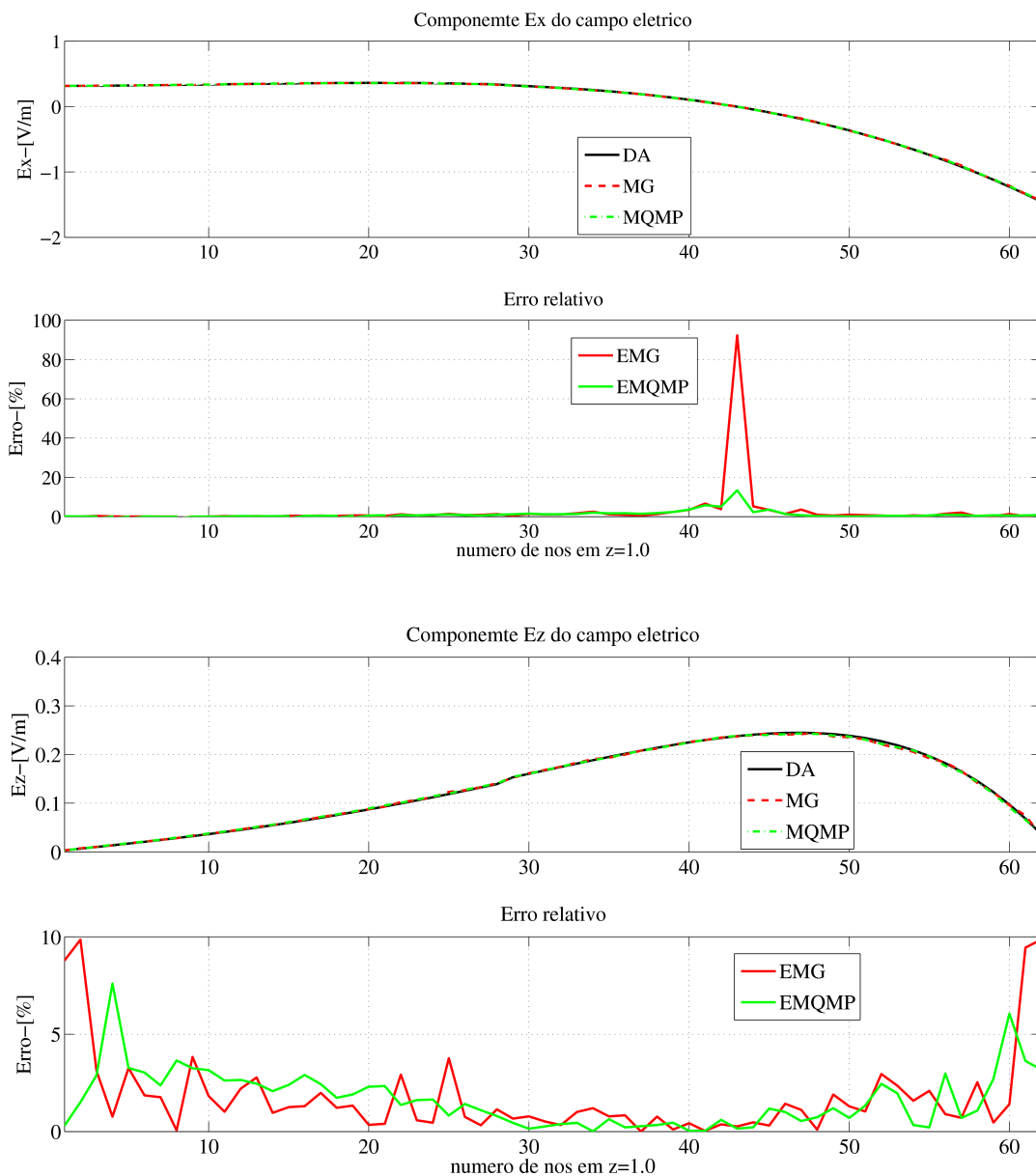


Figura 4.12: Na figura, DA, MG e MQMP representam as derivadas: analítica, por média dos gradientes das funções base e por ajuste móvel respectivamente, em que MG e MQMP são obtidos da solução numérica. EMG e EMQMP são seus erros relativos



5 CONCLUSÃO

Apresentou-se neste trabalho dois métodos para calcular numericamente as derivadas 2D, média aritmética do gradiente das funções base (MAGFB) e ajuste móvel por mínimos quadrados ponderado (AMMQP). Os algoritmos para calcular as derivadas foram validados a partir da solução analítica da equação de Poisson com condições de Dirichlet homogênea. Os primeiros resultados foram exibidos para comparar somente o erro introduzido pelo cálculos das derivadas, os demais são obtidos a partir da solução numérica de elementos finitos, que vale ressaltar, é de suma importância a familiarização com o método, tendo em vista a relevância do mesmo para modelagem de métodos eletromagnéticos.

Foi possível notar que os dois métodos, mostram efetiva melhora nos resultados a partir de sucessivos refinamentos sobre a malha. Entretanto, as debilidades mostradas pelas técnicas é traduzida pelo erro relativo em pontos na fronteira e onde o campo é zero, tais debilidades também mostraram ser contornáveis com maior refinamento, que se tratando de malha não estruturada, pode ser local. Contudo, o ajuste móvel se mostra menos discrepante, ou seja, apresenta resultados mais confiáveis, portanto mais eficiente. Uma vantagem pertinente do AMMQP em relação a MAGFB é que, as derivadas podem ser mais confiavelmente atribuídas à pontos arbitrários da malha, ou seja, fora de um ponto nodal, sem que seja necessário um elevado refinamento da malha.

O ajuste móvel de valores nodais por mínimos quadrados ponderados, oferece um recurso fácil e efetivo de calcular derivadas de problemas elípticos e lineares, (TABBARA, 1994). Além disso, o método é facilmente aplicado a elementos de geometria distinta, triângulos, quadrados, entre outros.

REFERÊNCIAS

- FRANCO, B. N. *Cálculo numérico*. São Paulo: Pearson Prentice Hall, 2006.
- HAYT Jr, W. H.; Buck. J. A. *Eletromagnetismo*. Tradução Amilton Soares Junior. São Paulo: McGraw Hill, 2008.
- JONATHAN RICHARD SHEWCHUK. TRIANGLE: A two-Dimensional Quality Mesh Generator And Delaunay. Disponível em: <<http://www.cs.cmu.edu/~quake/triangle.html>>. Acesso em: 15 setembro 2010.
- LAY, D. C. *Álgebra linear e suas aplicações*. Tradução Ricardo Camelier. Rio de Janeiro: LTC, 2007.
- MELO, P. E. M. *Novos Métodos Para Filtragem de dados sísmicos de reflexão*. 2007. 79 f. Dissertação (Doutorado em geofísica) – Universidade Federal da Bahia, Salvador – BA, 2007.
- RIJO, L. : *Teoria os métodos elétricos e Eletromagnéticos I e II*. Bélem-PA, 1991. Notas de aulas do curso
- SAITO, K. K. S. *Modelagem Numérica de dados Magnétotélúrico*. 2009. 46 f. Dissertação (Mestrado em geofísica) – Universidade Federal do Pará, Belém – PA, 2009.
- SOBRINHO, A. S. C. *Introdução ao método dos elementos finitos*. Rio de Janeiro: Ciência Moderna, 2006.
- TABBARA, M. et al. *Finite element derivative recovery by moving least square interpolants*. Computer Methods in Mechanics and Engineering, 117: p. 211-223, 1994.