

SIMULAÇÃO NUMÉRICA PARA O CÁLCULO DE
CAMPOS ELÉTRICOS EM DOMÍNIOS ILIMITADOS

Este exemplar corresponde à redação
final da tese devidamente corrigida e
defendida pelo Sr. João Nelson Hoffmann,
e aprovada pela Comissão Julgadora.

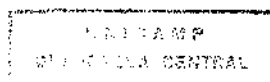
Campinas, 15 de abril de 1993

Petronio Pulino

Prof. Dr. Petronio Pulino

Orientador

Dissertação apresentada ao Instituto
de Matemática, Estatística e Ciência
da Computação, UNICAMP, como requisito
parcial para a obtenção do título de
Mestre em Matemática Aplicada, área de
Análise Numérica.



ÍNDICE

| | |
|--|-----------|
| Lista de símbolos e abreviações | <i>vi</i> |
| Resumo | <i>ix</i> |
| Introdução | <i>xi</i> |
| Capítulo 1 - Equações básicas da eletrostática | 1 |
| Capítulo 2 - Método de simulação de cargas | 4 |
| 2.1 Descrição do método | 4 |
| 2.2 Aplicação: Cilindro de comprimento infinito | 8 |
| Capítulo 3 - Método de diferenças finitas | 12 |
| 3.1 Descrição do método | 12 |
| 3.2 Aplicação: Cilindro de comprimento infinito | 14 |
| Capítulo 4 - Método híbrido de simulação de cargas e diferenças finitas | 18 |
| 4.1 Definição do problema | 18 |
| 4.2 Aplicação do método de diferenças finitas | 19 |
| 4.3 Aplicação do método de simulação de cargas | 21 |

| | |
|------------------------------|--|
| N, V | Número de nós sobre a discretização de um contorno no método de simulação de cargas |
| W | Número de nós sobre uma linha de descontinuidade na permissividade elétrica relativa do meio |
| Φ | Potencial elétrico (volt - V) |
| $\bar{\Phi}, \Phi_A, \Phi_B$ | Potencial elétrico dado sobre um contorno (volt - V) |
| Φ_0 | Potencial elétrico sobre $\Omega_A \cup \Omega_1 \cup \Omega_2$ (volt - V) |
| Φ_1 | Potencial elétrico sobre o contorno $\partial\Omega_1$ (volt - V) |
| Φ_2 | Potencial elétrico sobre o contorno $\partial\Omega_2$ (volt - V) |
| h, ph, qh, rh, sh | Passo da malha de diferenças finitas |
| S_0, S_1, Z_1 | Matriz de discretização de $\partial\Phi/\partial n$ sobre $\partial\Omega_1$ |
| E, S_4 | Matriz de discretização de $\partial\Phi/\partial n$ sobre $\partial\Omega_2$ |
| A, D, Z_2, S_2, S_3 | Matriz do problema de diferenças finitas |
| f | Vetor auxiliar da discretização de $\partial\Phi/\partial n$ sobre $\partial\Omega_1$ |
| I | Matriz identidade |
| q, r | Vetor de cargas elétricas |
| P_1, P_2 | Matriz com os coeficientes de potencial elétrico |
| F | Matriz com os coeficientes de campo elétrico |
| θ | Parâmetro de relaxação para a resolução do sistema de equações relativo ao método híbrido |
| a_j | Coefficiente para a discretização da derivada normal |

| | |
|---------------------------|---|
| Q, \bar{Q}, Q_i | Matriz ortogonal resultante de uma decomposição QR |
| R, \bar{R}, R_i | Matriz triangular superior resultante de uma decomposição QR |
| T | Matriz de iteração do método híbrido |
| u | Vetor auxiliar do processo iterativo |
| b, c | Vetor do lado direito de um sistema de equações |
| g, f | Função auxiliar |
| Δ | Operador Laplaciano |
| ∇ | Operador Gradiente |
| n | Vetor normal a uma superfície |
| t | Vetor tangencial a uma superfície |
| l, s, v | Designação genérica de comprimento, superfície e volume |
| \mathbb{R}^n | Espaço vetorial dos números reais, de dimensão n |
| $\mathbb{R}^{m \times n}$ | Espaço vetorial das matrizes reais, de dimensão $m \times n$ |

RESUMO

Um novo tratamento matemático é proposto ao método híbrido de simulação de cargas e diferenças finitas para o cálculo de campos eletrostáticos em domínios ilimitados, também aplicável ao método híbrido similar de simulação de cargas e elementos finitos.

O problema de simulação de cargas é tratado com técnicas de quadrados mínimos, utilizando a decomposição QR da matriz retangular resultante, e o método de Gram-Schmidt clássico ou modificado. Para a resolução do problema de diferenças finitas é proposto o método dos Gradientes Conjugados Quadráticos com pré-condicionamento, incluindo as necessárias técnicas de esparsidade. O sistema acoplado de equações lineares é resolvido utilizando resultados da teoria do ponto fixo. São propostas ainda novas discretizações para as condições de contorno, que conduzem a resultados mais precisos. Exemplos de aplicação são inclusos, com avaliação de resultados.

A B S T R A C T

A new mathematical treatment is proposed to the hybrid method of charge simulation and finite differences for the computation of unbounded electrostatic fields, also applicable to the similar hybrid method of charge simulation and finite elements.

The charge simulation problem is solved by making use of the least squares technique, including the QR decomposition of the resulting rectangular matrix, and the classical or the modified Gram-Schmidt method. The Conjugate Gradient Squared method with a preconditioning technique is proposed for the solution of the finite difference problem, which is stored in the computer with adequate sparsity techniques. The resulting coupled system of linear equations is solved by making use of some results of the fixed point theory. New procedures are also suggested for the discretization of the boundary conditions, which lead to results of increased precision. Case studies are included and the results are analysed.

INTRODUÇÃO

Na análise de campos eletromagnéticos, o uso de técnicas numéricas de discretização tem se tornado cada vez mais popular e representam o meio mais utilizado para tal, exceto provavelmente em problemas com soluções triviais. No entanto, as técnicas numéricas apresentam a desvantagem de que apenas um número finito de nós (ou elementos) pode ser utilizado para a discretização de um dado domínio, devido à limitação do espaço de memória disponível nos computadores.

Em contrapartida, é comum aos campos eletromagnéticos que o domínio do problema a ser resolvido estenda-se ao infinito. Surge então uma contradição nas exigências ao resolvermos estes casos. Por um lado a necessidade de um número infinito de nós discretizados para a modelagem do espaço infinito, e por outro a disponibilidade computacional que permite a consideração de um número finito de nós apenas.

Tornou-se então uma prática comum na utilização de técnicas de discretização, a limitação do domínio ("truncation") a uma certa distância da região de interesse. Ao tornarmos o respectivo contorno remoto o seu efeito diminuirá, até que o erro gerado seja tolerável. Embora isto venha requerer um elevado número de elementos, será obtida efetivamente uma solução, teoricamente em qualquer nível de precisão desejado.

O objetivo deste trabalho é apresentar uma técnica alternativa para a modelagem de problemas de eletrostática com domínios ilimitados, que denominaremos de método híbrido de simulação de cargas com diferenças finitas (ou simplesmente método híbrido), que se revelará muito mais eficiente do que a simples limitação do domínio. Na realidade, existem diversas técnicas disponíveis atualmente para o tratamento do mesmo problema, cada qual tendo suas próprias vantagens e desvantagens. Segue uma breve descrição dos principais métodos conhecidos com os respectivos comentários (vide Emson [4]):

1. Limitação do domínio ("truncation")

É um dos métodos mais utilizados, de simples aplicação, não requerendo alterações nos programas computacionais com técnicas de discretização eventualmente disponíveis.

Como verificação, podemos mover o contorno ampliando o domínio, e procedendo a uma comparação dos resultados. A repetição deste procedimento servirá como orientação para a precisão dos valores obtidos.

2. Aplicação ("Mapping")

A limitação do domínio do item anterior geralmente irá requerer uma região extensa e como consequência, uma grande quantidade de nós. Assim, uma aplicação ("Mapping") pode ser

utilizada para mover o contorno externo a uma distância maior da região de interesse. Em geral este procedimento funciona bem quando a região exterior não contém fontes.

3. Solução Analítica

Em muitos problemas podemos aplicar condições de contorno à malha discretizada, baseadas em uma solução analítica para a região externa. Ao movermos o contorno externo de modo a se distanciar das fontes, a solução sobre ele se aproxima da solução analítica obtida com fontes pontuais localizadas no centro do problema. A região exterior pode ser aproximada aplicando esta solução como condições de contorno (Dirichlet ou Neumann).

A vantagem deste método sobre o truncamento é que o contorno externo poderá ser localizado mais próximo da região de interesse, reduzindo o domínio a ser discretizado. Obviamente, a solução analítica referida deverá ser conhecida.

4. Elementos infinitos

Neste método a região exterior é dividida numa série de elementos de extensão infinita. Em uma aplicação do método, a variação da solução na direção infinita segue alguma função com decaimento (função inversa ou logarítmica) com a distância. Como alternativa, poderá ser estabelecida uma aplicação ("Mapping") relacionando os nós localizados no infinito.

5. Método híbrido de elementos de contorno com métodos de discretização

O método dos elementos de contorno pode ser aplicado à região exterior, sendo comumente utilizado em problemas lineares. Este método pode ser visto como a substituição de uma região por um conjunto de fontes. Em geral, haverá uma fonte localizada em cada nó do contorno, cujo valor é calculado, sendo que o campo devido a estas fontes pode ser expresso em termos de funções de Green apropriadas.

Já foi sugerido por alguns autores que as fontes equivalentes não necessitam ser localizadas precisamente sobre o contorno, o que evita a avaliação de certas singularidades. Também foi sugerido que o número de fontes não necessita ser igual ao número de nós. Sob estes aspectos, o método híbrido de elementos de contorno com métodos de discretização em muito se assemelha ao método seguinte, em princípio aplicável apenas a problemas de eletrostática.

6. Método híbrido de simulação de cargas com métodos de discretização

Em certos problemas de eletrostática que envolvem altos valores de potenciais elétricos (da ordem de centenas de quilovolts) a influência da superfície da terra não pode ser desprezada. Muitas vezes estes casos podem ser tratados de modo relativamente simples por meio de um método conhecido por "método

das imagens" que serve também como base para outro método de extensa aplicação na engenharia elétrica de alta tensão, conhecido por "simulação de cargas". Este método torna-se assim particularmente apropriado para aplicação em problemas de domínio ilimitado, que envolvam a superfície da terra (considerada idealmente como um plano condutor elétrico de potencial nulo).

Por outro lado, os métodos de discretização são, muitas vezes, a única ferramenta disponível para a resolução de problemas mais complexos, como por exemplo, os casos que envolvem vários materiais com diferentes constantes de permissividade elétrica relativa.

É previsível, portanto, que a utilização de um método híbrido de simulação de cargas com métodos de discretização (por exemplo o método de diferenças finitas) possa ser vantajosa em diversos problemas práticos. A conexão entre eles resulta de determinadas condições que são impostas sobre a fronteira comum arbitrariamente escolhida.

Finalizando esta breve retrospectiva, em geral não será possível dizer qual dos métodos acima mencionado é o melhor, visto que a escolha dependerá de diversos fatores, muitos dos quais específicos a cada problema. Por exemplo, se a precisão dos resultados é primordial, provavelmente os métodos 5 e 6 serão recomendáveis. Por outro lado, se tamanho do problema e rapidez dos cálculos são fatores significativos, o método 4 poderá ser preferível, e assim por diante.

No que se refere ao método híbrido de simulação de cargas com métodos de discretização, a alguns autores que já fizeram uso desse método (Abdel-Salam [1], Okubo [13], Steinbigler [20]) parece claro a existência de pontos que deveriam ser melhor estudados, sendo muitas vezes explicitamente referidos em suas pesquisas. Entre estes pontos podemos citar, por exemplo,

1) Discretização adequada para as condições de contorno sobre a fronteira arbitrária.

2) Discretização adequada para as condições de contorno sobre fronteiras com descontinuidade no parâmetro permissividade elétrica relativa dos materiais.

3) Redução no espaço de memória de computador necessário para o armazenamento de matrizes cheias resultantes da aplicação do método de simulação de cargas.

4) Técnicas de esparsidade adequadas para o armazenamento de matrizes referentes ao método de discretização utilizado.

5) Resolução do sistema linear acoplado resultante do método híbrido, uma vez que é composto por submatrizes com características diferentes, oriundas de problemas diferentes (simulação de cargas e métodos de discretização), que normalmente geram sistemas lineares mal-condicionados.

No estudo que se segue abordaremos todos os pontos acima com soluções inovadoras, procurando dar ao método híbrido um tratamento matemático mais rigoroso na tentativa de torná-lo competitivo e promissor como ferramenta de aplicação em problemas de interesse prático. Sob este aspecto, o presente trabalho pode ser resumido nos seguintes pontos:

1) O método de simulação de cargas é tratado como um problema de quadrados mínimos, sendo resolvido por meio da decomposição QR da matriz resultante. Deste modo, a capacidade de memória requerida pelo computador fica substancialmente reduzida.

2) O problema de discretização é armazenado no computador de modo bastante simples, sendo resolvido com técnicas eficazes apropriadas, incluindo pré-condicionamento do sistema linear respectivo.

3) São utilizadas técnicas adequadas para a discretização das condições de contorno, que conduzem a uma maior precisão nos resultados.

4) A resolução do sistema linear acoplado é feita utilizando resultados da teoria do ponto fixo, deste modo eliminando a dependência dos valores iniciais arbitrados para os potenciais elétricos.

Tomados em conjunto, os pontos acima representam uma sensível evolução em relação ao estágio anterior de desenvolvimento deste método para cálculo de campos elétricos em domínios ilimitados.

Abordaremos explicitamente o método híbrido de simulação de cargas com diferenças finitas, aplicado a problemas bidimensionais (ou tridimensionais com simetria de translação), muito embora praticamente todo o estudo seja igualmente aplicável ao método híbrido similar de simulação de cargas com elementos finitos. Da mesma forma, grande parte da abordagem teórica é igualmente aplicável a problemas tridimensionais com simetria axial, sendo que problemas deste tipo são também muito frequentes na engenharia elétrica de alta tensão.

CAPÍTULO 1

EQUAÇÕES BÁSICAS DA ELETROSTÁTICA

Os campos de natureza eletrostática são devidos à existência de cargas elétricas, cuja magnitude não varia com o tempo, e derivam basicamente de duas das equações de Maxwell:

$$\oint_s (\epsilon \epsilon_0 \mathbf{E}) \cdot d\mathbf{s} = \int_v \rho_v dv \quad 1.1$$

$$\oint_l \mathbf{E} \cdot d\mathbf{l} = 0 \quad 1.2$$

onde \mathbf{E} é o vetor intensidade de campo elétrico

ρ_v é a densidade volumétrica de carga elétrica

ϵ é a permissividade elétrica relativa do meio

ϵ_0 é a permissividade elétrica absoluta do vácuo

A equação 1.1 também é conhecida como Lei de Gauss e a equação 1.2 como Lei de Faraday. Das experiências de Faraday resulta também a relação (admitindo meio isótropo e linear)

$$\mathbf{D} = \epsilon \epsilon_0 \mathbf{E} \quad 1.3$$

onde \mathbf{D} é o vetor densidade de fluxo elétrico.

Como demonstrado no Apêndice I, o vetor \mathbf{E} pode ser obtido de uma função escalar auxiliar Φ , por meio de

$$\mathbf{E} = - \nabla \Phi \quad \text{I.4}$$

A função Φ é chamada de potencial elétrico do campo \mathbf{E} .

A equação que rege a distribuição da função potencial elétrico em problemas de eletrostática é

$$\nabla \cdot (\epsilon \epsilon_0 \nabla \Phi) = - \rho_v \quad \text{I.6}$$

conhecida como equação de Poisson, também deduzida no Apêndice I.

Obtida Φ pela resolução da equação I.6, podemos encontrar \mathbf{E} com a equação I.4.

As equações de Maxwell 1.1 e 1.2 podem ser utilizadas para obtermos condições de continuidade através de qualquer superfície do espaço tridimensional. No Apêndice II são deduzidas as respectivas equações

$$\mathbf{n} \cdot \mathbf{D}_1 = \mathbf{n} \cdot \mathbf{D}_2 \quad \text{II.1}$$

ou seja, a componente normal do vetor densidade de fluxo elétrico é contínua através da superfície, e

$$\mathbf{t} \cdot \mathbf{E}_1 = \mathbf{t} \cdot \mathbf{E}_2 \quad \text{II.2}$$

isto é, a componente tangencial do vetor intensidade de campo elétrico também é contínua através da superfície.

A condição II.2 pode ser substituída por uma condição de aplicação prática mais simples (vide Apêndice II),

$$\Phi(P_1) = \Phi(P_2) \qquad \text{II.3}$$

que retrata a continuidade da função potencial elétrico.

As equações I.4, I.6, II.1 e II.3 são as equações fundamentais que serão utilizadas nos problemas que trataremos nos capítulos seguintes.

CAPÍTULO 2

MÉTODO DE SIMULAÇÃO DE CARGAS

2.1 DESCRIÇÃO DO MÉTODO

O método de simulação de cargas (Singer [18]), designado abreviadamente de MSC, permite o cálculo de campos elétricos de modo relativamente simples em problemas em cujo domínio a permissividade elétrica relativa é constante.

A princípio, vamos considerar um problema bidimensional no qual uma carga pontual de valor q_j , localizada em (x_j, y_j) , produz um potencial elétrico $\Phi(P_i)$ sobre um ponto P_i , de coordenadas (x_i, y_i) (figura 2.1.1).

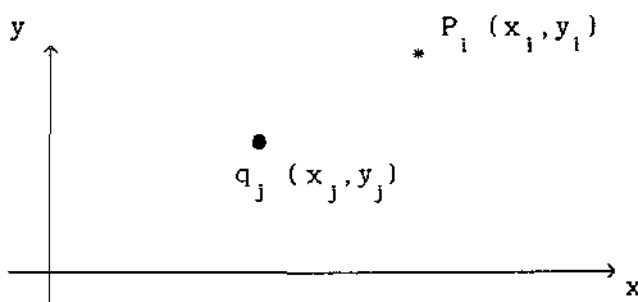


FIGURA 2.1.1 - Carga elétrica pontual

Vamos admitir que $\Phi(P_i)$ possa ser escrito como

$$\Phi(P_i) = p_{ij} \cdot q_j ,$$

onde p_{ij} é um coeficiente que não depende de q_j , o que é razoável supor em vista da linearidade do operador Laplaciano (Paris [15], cap. 3). O coeficiente p_{ij} será designado de "coeficiente de potencial elétrico, de Maxwell".

Para os problemas práticos que pretendemos estudar, será considerado $\Phi = 0$ ao longo do eixo x , o qual será coincidente com a superfície da terra. Para este caso, o coeficiente p_{ij} será determinado no Apêndice III, por meio da equação III.1.3 .

Utilizando o princípio da superposição (Paris [15], cap. 3), o potencial em P_i , na presença de K cargas q_j , será

$$\Phi(P_i) = \sum_{j=1}^K p_{ij} \cdot q_j \quad 2.1.1$$

A expressão 2.1.1 é a base para o método de simulação de cargas, exposto a seguir:

Suponhamos que a função potencial elétrico Φ seja conhecida ($\Phi = \bar{\Phi}$) sobre uma ou mais superfícies fechadas, as quais envolvem todas as distribuições de cargas elétricas do problema. Estas superfícies serão discretizadas por meio de N pontos.

Por outro lado, todas essas distribuições de cargas também serão discretizadas por meio de K cargas elementares ($K \leq N$) de valor q_j , $j=1...K$, a ser determinado (figura 2.1.2).

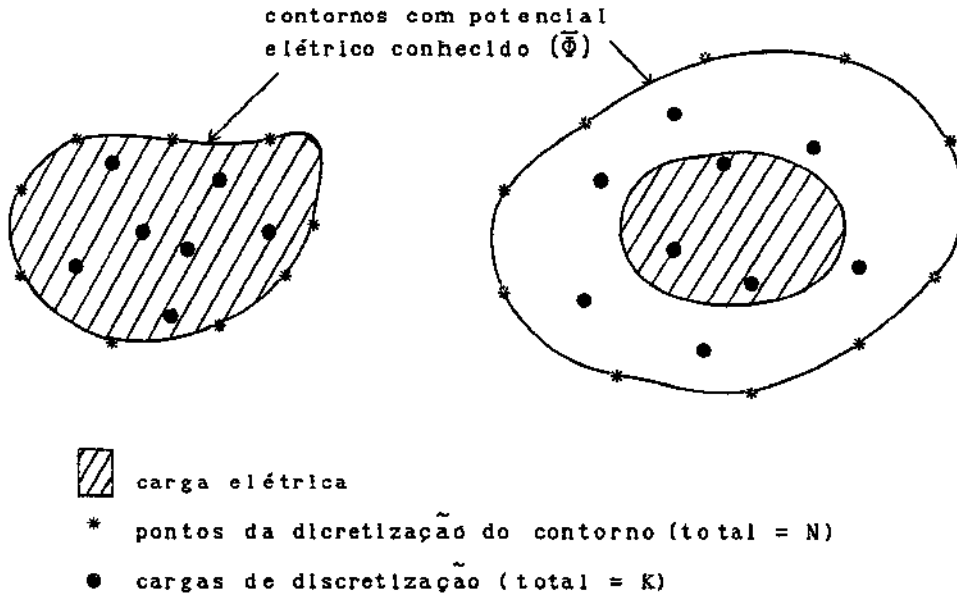


FIGURA 2.1.2 - Discretização no método de simulação de cargas

Estas cargas serão localizadas arbitrariamente no interior da respectiva região. Desta forma, podemos estabelecer um sistema de equações

$$\sum_{j=1}^K p_{ij} \cdot q_j = \bar{\Phi}(P_i) \quad , \quad i = 1...N \quad 2.1.2$$

Este sistema pode ser escrito na forma matricial

$$P \cdot q = \bar{\Phi} \quad , \quad 2.1.3$$

com $P \in \mathbb{R}^{N \times K}$ matriz cheia, o qual pode ser resolvido utilizando o método dos quadrados mínimos (Apêndice VII), uma vez que $K \leq N$. Determinado o vetor q , o potencial elétrico em qualquer ponto P_i externo às superfícies que envolvem as cargas é calculado utilizando a expressão 2.1.1.

A validade da equação 2.1.1 é assegurada pela unicidade de soluções do problema de Dirichlet (uma vez que os potenciais elétricos sobre os contornos são dados) e pelo princípio da superposição. Sua precisão dependerá essencialmente da discretização (quantidade de pontos N e quantidade de cargas K) utilizada nos cálculos.

Considerando que $E = -\nabla \phi$ (vide Capítulo 1), podemos escrever

$$E(P_i) = - \left[\frac{\partial(p_{ij} q_j)}{\partial x_i} e_1 + \frac{\partial(p_{ij} q_j)}{\partial y_i} e_2 \right] = (f_x^{ij} e_1 + f_y^{ij} e_2) q_j$$

onde e_1 e e_2 são os vetores unitários das direções x e y , respectivamente ,

f_x^{ij} e f_y^{ij} são chamados de "coeficientes de campo elétrico, de Maxwell", dados pelas expressões III.2.1 e III.2.2 .

Assim, em vista da linearidade do operador gradiente (7) e utilizando o princípio da superposição, o campo elétrico em P_i , na presença de K cargas q_j , será

$$E(P_i) = \sum_{j=1}^K \left(f_x^{ij} e_1 + f_y^{ij} e_2 \right) q_j \quad 2.1.4$$

2.2 APLICAÇÃO: CILINDRO DE COMPRIMENTO INFINITO

Vamos considerar um cilindro de comprimento infinito e com raio igual a 1,0 cm, cujo eixo é paralelo à superfície da terra e está situado a uma altura de 4 cm (figura 2.2.1).

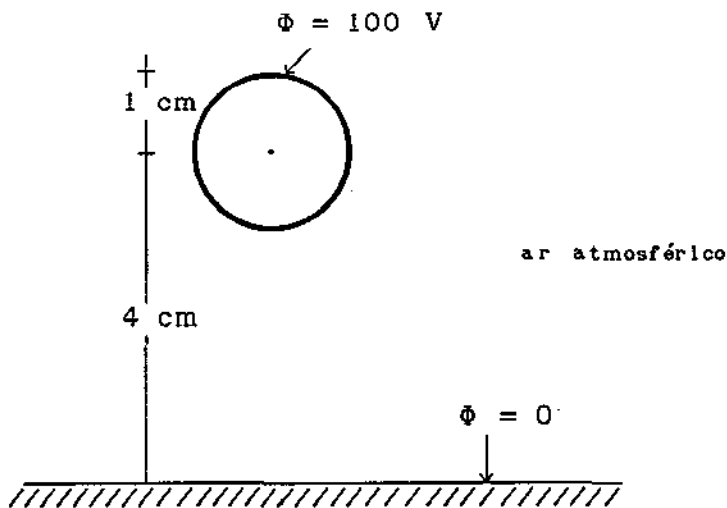


FIGURA 2.2.1 - Cilindro de comprimento infinito

Nosso objetivo é calcular a distribuição do potencial elétrico no espaço próximo à superfície do cilindro, admitindo que o potencial elétrico sobre a sua superfície é conhecido, $\bar{\Phi} = 100$ V.

Como se trata de um problema com apenas um material dielétrico (ar atmosférico externo à superfície do cilindro, que será considerado com $\epsilon = 1,0$), e visto que este problema pode ser reduzido a um caso bidimensional segundo um plano perpendicular ao eixo do cilindro, então o método de simulação de cargas, com as equações indicadas na Seção 2.1, torna-se aplicável. Para a resolução de problemas deste tipo foi desenvolvido o programa computacional MSC, descrito pela referência [9], cap. 8.

A verificação da precisão no MSC normalmente é feita calculando os potenciais elétricos em pontos sobre a superfície dada (valor exato $\Phi = 100$ V) . A tabela 2.2.1 abaixo apresenta alguns resultados, sendo que a coluna ERRO SUPERFICIAL indica a diferença encontrada em valores absolutos, tendo sido calculados os potenciais elétricos de 50 pontos equidistantes sobre a superfície do cilindro.

O problema apresentado, em particular, possui uma solução analítica dada por Johnk [10],

$$\Phi(x,y) = \frac{\bar{\Phi}}{2 \ln \left[\frac{h+a}{r} \right]} \cdot \ln \left[\frac{(y+a)^2 + x^2}{(y-a)^2 + x^2} \right]$$

onde $a = \sqrt{h^2 - r^2}$,

h = altura do centro do cilindro,

r = raio do cilindro,

$\bar{\Phi}$ = potencial elétrico dado na superfície do cilindro.

A coluna ERRO EXTERNO indica o máximo erro observado sobre um conjunto de cerca de 150 pontos externos e próximos ao cilindro.

TABELA 2.2.1

| Verificação da precisão dos resultados fornecidos pelo MSC | | | | |
|--|----|----|----------------------|------------------|
| Ítem | K | N | ERRO SUPERFICIAL (V) | ERRO EXTERNO (V) |
| 1 | 1 | 3 | 6,57 | 6,59 |
| 2 | 4 | 4 | 0,98 | 1,44 |
| 3 | 8 | 8 | 0,050 | 0,094 |
| 4 | 8 | 16 | 0,028 | 0,042 |
| 5 | 12 | 24 | 0,0003 | 0,0016 |

Como conclusão para este exemplo, ao tomarmos K em torno de 10 ou mais, obteremos excelentes níveis de precisão conforme indicado pela tabela.

A figura 2.2.2 mostra curvas eqüipotenciais nas proximidades do cilindro, e serve como ilustração para os resultados que podem ser obtidos com o método de simulação de cargas. Estas curvas resultam do cálculo de potenciais elétricos em diversos pontos (utilizando a equação 2.1.1), e procedendo à necessária interpolação, que pode ser feita com programas de computador apropriados.

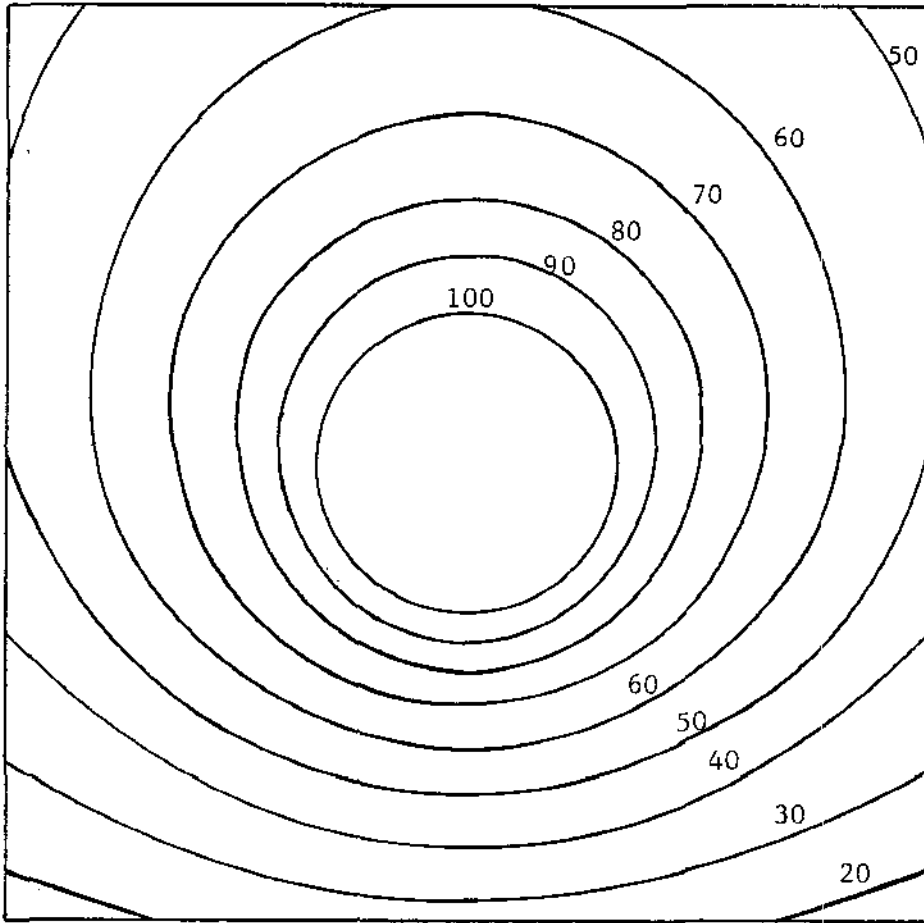


FIGURA 2.2.2 - Curvas eqüipotenciais

CAPÍTULO 3

MÉTODO DE DIFERENÇAS FINITAS

3.1 DESCRIÇÃO DO MÉTODO

O método de diferenças finitas (Forsythe [6], Mitchell [12]), designado abreviadamente de MDF, pode ser utilizado para a obtenção de uma solução aproximada para a equação de Poisson, desde que a região Ω em análise possua um contorno limitado. Como estaremos interessados em problemas de eletrostática sem fontes na região Ω , a equação de Poisson reduz-se a uma equação homogênea (equação de Laplace), cujas condições de contorno poderão ser do tipo Dirichlet ou Neumann, ou seja, o problema consiste em resolver

$$\begin{aligned} \Delta \phi &= 0 && \text{em } \Omega && 3.1.1 \\ \phi &= g && \text{em } \partial\Omega_A, \\ \partial\phi/\partial\mathbf{n} &= f && \text{em } \partial\Omega_B, && \partial\Omega_A \cup \partial\Omega_B = \partial\Omega \end{aligned}$$

onde Δ é o operador Laplaciano,

f e g são funções dadas.

Estaremos interessados em problemas com condições de Dirichlet, onde $\partial\Omega \equiv \partial\Omega_A$. Para tal, o problema 3.1.1 irá requerer a discretização do operador Laplaciano, considerada no Apêndice V.

Para a aplicação do MDF, a região de interesse é discretizada por meio de uma malha retangular (figura 3.1.1), com espaçamento máximo h entre nós, sendo $p, q, r, s \leq 1,0$.

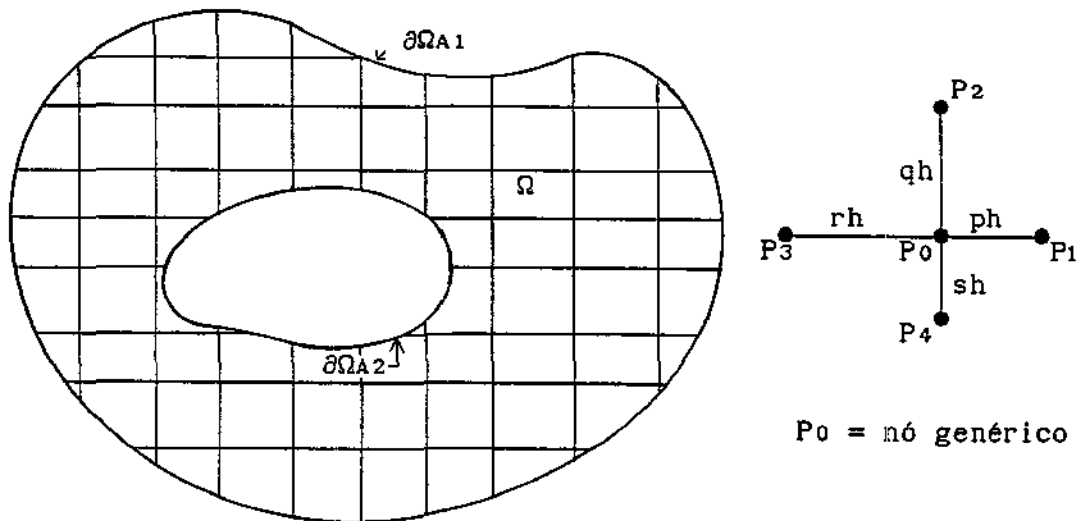


FIGURA 3.1.1 - Discretização no método de diferenças finitas

Esta malha de diferenças finitas irá gerar M nós no interior de Ω . Assim, a discretização de 3.1.1 por meio da expressão V.3 fornece, em um nó genérico P_0 ,

$$\frac{1}{p(p+r)}\phi(P_1) + \frac{1}{q(q+s)}\phi(P_2) + \frac{1}{r(p+r)}\phi(P_3) + \frac{1}{s(q+s)}\phi(P_4) - \left(\frac{1}{p \cdot r} + \frac{1}{q \cdot s} \right) \phi(P_0) = 0 \quad 3.1.2$$

Esta expressão aplicada a cada nó da discretização conduz a um sistema de equações lineares do tipo

$$D \Phi = c \quad 3.1.3$$

com $D \in \mathbb{R}^{M \times M}$, e sendo c o vetor que inclui a informação referente ao potencial elétrico dado sobre $\partial\Omega$. Pela observação da equação 3.1.2, notamos que a matriz D terá uma estrutura esparsa, e conterà no máximo cinco elementos não nulos em cada linha.

A resolução do sistema de equações 3.1.3 fornecerá a solução aproximada procurada para o problema 3.1.1, sobre os M nós da discretização.

3.2 APLICAÇÃO: CILINDRO DE COMPRIMENTO INFINITO

Vamos considerar que o cilindro de comprimento infinito da Seção 2.2 seja envolto por uma região fechada retangular Ω (figura 3.2.1) em cujo contorno $\partial\Omega_1$ e $\partial\Omega_A$ os respectivos potenciais Φ_1 e Φ_A são assumidos conhecidos.

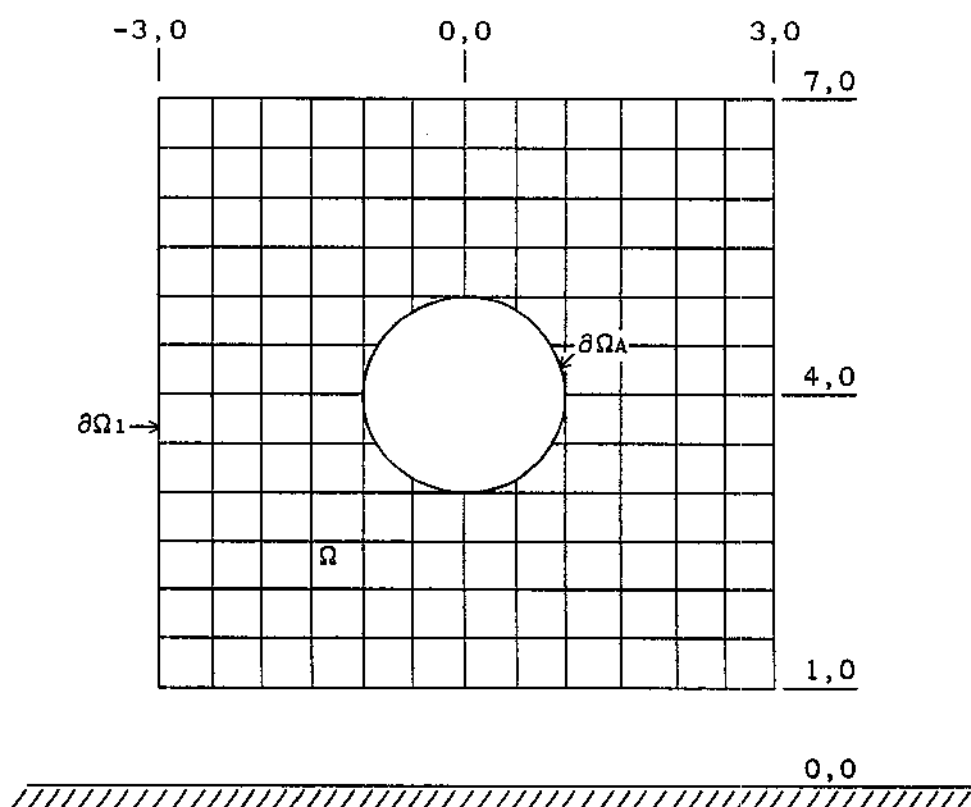


FIGURA 3.2.1 - Cilindro de comprimento infinito

O problema consiste em resolver

$$\Delta \Phi = 0 \quad \text{em } \Omega \quad 3.2.1$$

$$\Phi = \Phi_1 \quad \text{em } \partial\Omega_1$$

$$\Phi = \Phi_A = 100 \text{ V} \quad \text{em } \partial\Omega_A$$

utilizando o MDF, como descrito na Seção 3.1. O vetor Φ_1 poderá ser obtido por meio da solução analítica dada na seção 2.2, ou com um cálculo prévio por meio do MSC utilizando, por exemplo, os dados do item 5 da tabela 2.2.1 que fornecerão Φ_1 com um excelente nível de precisão.

Para tal, arbitraremos uma malha sobre Ω , com espaçamento h constante entre nós (exceto obviamente junto a $\partial\Omega$).

Aplicando a expressão 3.1.2 a cada um dos M nós, obteremos um sistema de equações do tipo 3.1.3, de ordem $M \times M$. Para a resolução do problema foi desenvolvido o programa computacional MDF, descrito pela referência [9], cap. 8.

A tabela 3.2.1 abaixo apresenta alguns resultados da aplicação do programa MDF, onde a coluna ERRO indica a máxima diferença encontrada em valores absolutos, calculando os potenciais elétricos sobre cada nó e comparando com os respectivos valores obtidos com a solução analítica dada na seção 2.2 .

TABELA 3.2.1

| Potencial elétrico sobre os nós de Ω Comparação MDF com a solução analítica | | | | |
|---|-------|-------|------|----------|
| N ^o | h | malha | M | ERRO (V) |
| 1 | 1,000 | 5x5 | 20 | 3,945 |
| 2 | 0,500 | 11x11 | 108 | 0,414 |
| 3 | 0,333 | 17x17 | 260 | 0,147 |
| 4 | 0,250 | 23x23 | 480 | 0,071 |
| 5 | 0,125 | 47x47 | 2012 | 0,018 |

Notamos que a coluna M refere-se ao número total de nós da região Ω , que não inclui os nós da malha que resultam internos ao cilindro.

Como conclusão, por exemplo, podemos esperar que na resolução de problemas deste tipo com h menor que aproximadamente 0,50 vezes o raio do cilindro, o erro máximo em cálculos com o método de diferenças finitas será menor do que 0,5 % (referido ao máximo valor de potencial elétrico do problema - 100 V).

CAPÍTULO 4

MÉTODO HÍBRIDO DE SIMULAÇÃO DE CARGAS E DIFERENÇAS FINITAS

4.1 DEFINIÇÃO DO PROBLEMA

Vamos considerar um problema como o mostrado na figura 4.1.1, onde nos interessa calcular os potenciais elétricos próximos à região A.

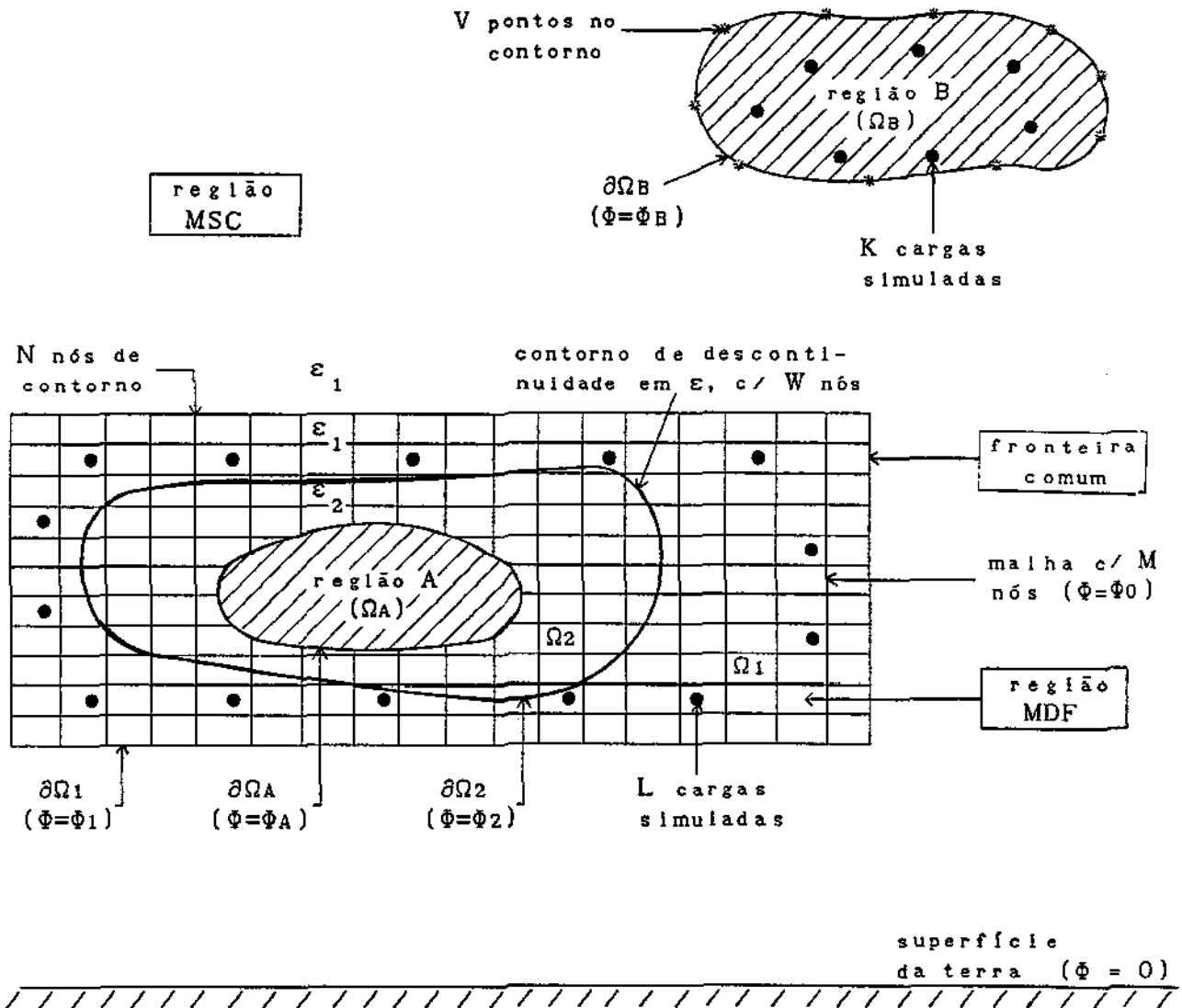


FIGURA 4.1.1 - Discretização no método híbrido

Para tal, escolhemos uma fronteira arbitrária (retangular por conveniência) $\partial\Omega_1$ englobando a região A (Ω_A), a região Ω_2 (região com permissividade elétrica relativa ϵ_2) e definindo assim uma região Ω_1 (com permissividade elétrica relativa ϵ_1).

Para a resolução deste problema, trataremos independentemente a região fechada, onde será utilizado o MDF, da região ilimitada, onde utilizaremos o MSC. Isto será válido desde que as condições de continuidade indicadas no Capítulo 1 sejam devidamente impostas à fronteira comum $\partial\Omega_1$.

4.2 APLICAÇÃO DO MÉTODO DE DIFERENÇAS FINITAS

Sobre a região fechada retangular arbitramos uma malha de diferenças finitas com M nós, o que definirá N nós sobre $\partial\Omega_1$ e W nós sobre $\partial\Omega_2$, conforme indicado na figura 4.1.1 .

Vamos admitir conhecido o vetor Φ_1 (potencial elétrico sobre a fronteira comum $\partial\Omega_1$), o que permite estabelecer os seguintes problemas de contorno:

Problema 1: $\Delta \Phi = 0$ em Ω_1 4.2.1

Com as condições de contorno:

$$\Phi = \Phi_1 \text{ em } \partial\Omega_1$$

$$\Phi = \Phi_2 \text{ em } \partial\Omega_2$$

$$\mathbf{n} \cdot \mathbf{D}_1 = g, \text{ ou seja, } -\epsilon_1 \epsilon_0 \left[\frac{\partial \Phi}{\partial \mathbf{n}} \right]_1 = g \text{ em } \partial\Omega_2$$

onde Δ é o operador Laplaciano e

g é uma função auxiliar,

Problema 2: $\Delta \Phi = 0$ em Ω_2 4.2.2

Com as condições de contorno:

$$\Phi = \Phi_A \text{ em } \partial\Omega_A$$

$$\Phi = \Phi_2 \text{ em } \partial\Omega_2$$

$$\mathbf{n} \cdot \mathbf{D}_2 = -g, \text{ ou seja, } -\epsilon_2 \epsilon_0 \left[\frac{\partial \Phi}{\partial \mathbf{n}} \right]_2 = -g \text{ em } \partial\Omega_2$$

Podemos resolver os problemas 4.2.1 e 4.2.2 simultaneamente com

a) $\Delta \Phi = 0$ em $\Omega_1 \cup \Omega_2$ 4.2.3

Com as condições de contorno:

b) $\Phi = \Phi_1$ em $\partial\Omega_1$

c) $\Phi = \Phi_2$ em $\partial\Omega_2$

d) $\Phi = \Phi_A$ em $\partial\Omega_A$

e) $\epsilon_1 \left[\frac{\partial \Phi}{\partial \mathbf{n}} \right]_1 + \epsilon_2 \left[\frac{\partial \Phi}{\partial \mathbf{n}} \right]_2 = 0$ em $\partial\Omega_2$

Designaremos Φ_0 ao vetor solução de 4.2.3a. Para a discretização de 4.2.3a, utilizamos a equação V.3 em cada nó de $\Omega_1 \cup \Omega_2$. Quando o respectivo nó é adjacente a $\partial\Omega_1$ ou $\partial\Omega_2$, sua equação conterà elementos do vetor Φ_1 ou Φ_2 respectivamente, devido às condições 4.2.3b e 4.2.3c. Adjacente a $\partial\Omega_A$, a equação de

discretização do Laplaciano para o respectivo nó conterá valores do vetor dado Φ_A , devido à condição 4.2.3d. Assim, o sistema de equações resultante de 4.2.3a, 4.2.3b, 4.2.3c e 4.2.3d terá a forma matricial

$$S_2 \Phi_1 + D \Phi_0 + S_3 \Phi_2 = c, \quad 4.2.4$$

onde $S_2 \in \mathbb{R}^{M \times N}$, matriz esparsa ,
 $D \in \mathbb{R}^{M \times M}$, matriz esparsa ,
 $S_3 \in \mathbb{R}^{M \times W}$, matriz esparsa ,
 $c \in \mathbb{R}^M$, vetor esparsa que inclui valores do vetor Φ_A .

Para atendermos à condição 4.2.3e, será necessário obter discretizações para $\partial\Phi/\partial\mathbf{n}$ sobre cada um dos W nós pertencentes a $\partial\Omega_2$. Utilizando a equação VI.2.1.7, teremos

$$\epsilon_1 \left[\frac{\partial\Phi}{\partial\mathbf{n}} \right]_1 (P_1) = \epsilon_1 \sum_{j=1}^3 a_{1j} [\Phi(P_1) - \Phi(P_{1j})]$$

e

$$\epsilon_2 \left[\frac{\partial\Phi}{\partial\mathbf{n}} \right]_2 (P_1) = \epsilon_2 \sum_{j=1}^3 a_{2j} [\Phi(P_1) - \Phi(P_{2j})]$$

onde $P_1 \in \partial\Omega_2$,
 $P_{1j} \in \Omega_1$,
 $P_{2j} \in \Omega_2$.

Assim, a discretização para 4.2.3e é

$$\epsilon_1 \sum_{j=1}^3 a_{1j} [\Phi(P_{1i}) - \Phi(P_{1j})] + \epsilon_2 \sum_{j=1}^3 a_{2j} [\Phi(P_{1i}) - \Phi(P_{2j})] = 0, \quad i=1 \dots W,$$

que em forma matricial fica

$$S_4 \Phi_0 + E \Phi_2 = 0 \quad 4.2.5$$

onde $S_4 \in \mathbb{R}^{W \times M}$, matriz esparsa com 6 elementos não nulos em cada linha,

$E \in \mathbb{R}^{W \times W}$, matriz diagonal.

4.3 APLICAÇÃO DO MÉTODO DE SIMULAÇÃO DE CARGAS

A região ilimitada será tratada pelo MSC, para o que discretizamos a superfície $\partial\Omega_B$ por meio de V pontos, arbitramos K cargas no interior da região B e L cargas no interior da região retangular fechada.

Vamos admitir conhecido o vetor Φ_1 (potenciais elétricos sobre a fronteira comum $\partial\Omega_1$). A aplicação do MSC conforme descrito no Capítulo 2 fornece as equações

$$\sum_{j=1}^K p_{1j} q_j + \sum_{j=1}^L p_{1j} q_j = \Phi_1(P_i), \quad i=1 \dots N$$

$$\sum_{j=1}^K p_{1j} q_j + \sum_{j=1}^L p_{1j} q_j = \Phi_B(P_i), \quad i=1 \dots V$$

onde q_j são as cargas discretizadas que simulam as distribuições de cargas do problema.

Estas equações podem ser escritas na forma matricial

$$P_1 q - I \Phi_1 = 0 \quad 4.3.1$$

$$e \quad P_2 q = \Phi_B, \quad 4.3.2$$

onde $P_1 \in \mathbb{R}^{N \times (K+L)}$, matriz cheia,

$P_2 \in \mathbb{R}^{V \times (K+L)}$, matriz cheia,

$I \in \mathbb{R}^{N \times N}$, matriz identidade,

$q \in \mathbb{R}^{K+L}$, vetor de cargas elétricas.

4.4 CONDIÇÕES DE CONTINUIDADE SOBRE A FRONTEIRA COMUM

A conexão entre o problema de diferenças finitas (Seção 4.2) e o problema de simulação de cargas (Seção 4.3) resulta da aplicação das condições de continuidade (Capítulo 1) sobre a fronteira comum $\partial\Omega_1$.

A condição de continuidade da componente tangencial do vetor intensidade de campo elétrico (expressão II.2) é satisfeita de imediato em vista da continuidade do potencial elétrico em $\partial\Omega_1$, bastando para isto considerar que os vetores Φ_1 das Seções 4.2 e 4.3 são idênticos.

A condição de continuidade da componente normal do vetor densidade de fluxo elétrico (expressão II.1) requer considerações adicionais. Podemos escrever esta condição como

$$\mathbf{n} \cdot \mathbf{D}_1(P_i) = \mathbf{n} \cdot \mathbf{D}_2(P_i) \quad , \quad i = 1 \dots N \quad ,$$

onde o índice (1) refere-se à região fechada (tratada pelo MDF) e o índice (2) à região ilimitada (tratada pelo MSC). Utilizando a expressão 1.3 temos,

$$\mathbf{n} \cdot [\epsilon_1 \epsilon_0 \mathbf{E}_1(P_i)] = \mathbf{n} \cdot [\epsilon_1 \epsilon_0 \mathbf{E}_2(P_i)] \quad , \quad \text{ou,}$$

$$\mathbf{n} \cdot \mathbf{E}_1(P_i) = \mathbf{n} \cdot \mathbf{E}_2(P_i) \quad , \quad i=1 \dots N \quad \quad 4.4.1$$

isto é, a componente normal do vetor intensidade de campo elétrico é contínua ao longo de $\partial\Omega_1$.

Podemos calcular $\mathbf{n} \cdot \mathbf{E}_1$ em cada ponto de $\partial\Omega_1$ utilizando a discretização proposta no Apêndice VI.1 (expressão VI.1.3). Esta discretização utiliza dois nós internos a Ω_1 , e seu erro é $O(h^2)$. Assim,

$$\begin{aligned} \mathbf{n} \cdot \mathbf{E}_1(P_i) &= - \mathbf{n} \cdot \nabla \phi(P_i) = - \frac{\partial \phi}{\partial \mathbf{n}}(P_i) = \\ &- [\alpha \phi(P_i) + \beta \phi(P_{i-1}) + \gamma \phi(P_{i-2})] \quad \quad 4.4.2 \end{aligned}$$

onde $P_i \in \partial\Omega_1$, $P_{i-1} \in \Omega_1$, $P_{i-2} \in \Omega_1$,

$$\alpha = 1/ph + 1/(ph+qh),$$

$$\beta = -(1/ph + 1/qh),$$

$$\gamma = 1/qh - 1/(ph+qh),$$

ph e qh são distâncias entre nós, conforme Apêndice

VI.1.

Para o cálculo de $\mathbf{n} \cdot \mathbf{E}_2$ utilizamos a expressão 2.1.4,

$$\mathbf{n} \cdot \mathbf{E}_2(P_i) = \mathbf{n} \cdot \sum_{j=1}^{K+L} \left(f_x^{ij} \mathbf{e}_1 + f_y^{ij} \mathbf{e}_2 \right) q_j$$

Visto que $\partial\Omega_1$ tem forma retangular com arestas alinhadas segundo os eixos cartesianos, teremos

$$\mathbf{n} \cdot f_x^{ij} \mathbf{e}_1 + \mathbf{n} \cdot f_y^{ij} \mathbf{e}_2 = \pm f_x^{ij} \text{ ou } \pm f_y^{ij},$$

dependendo da localização do nó i sobre $\partial\Omega_1$. Assim, podemos designar, genericamente

$$\mathbf{n} \cdot \mathbf{E}_2(P_i) = \sum_{j=1}^{K+L} f_{ij} q_j \quad 4.4.3$$

Como \mathbf{n} não está definido sobre os nós dos cantos de $\partial\Omega_1$, podemos escolher como aproximação (por exemplo) \mathbf{n} na direção de um dos eixos cartesianos. Assim, a condição de continuidade 4.4.1 sobre $\partial\Omega_2$ resulta em

$$\sum_{j=1}^{K+L} f_{ij} q_j + [\alpha\phi(P_i) + \beta\phi(P_{i-1}) + \gamma\phi(P_{i-2})] = 0, \quad i=1\dots N,$$

que pode ser escrito em forma matricial como

$$F q + S_0 \Phi_1 + S_1 \Phi_0 = 0 \quad 4.4.4$$

onde $F \in \mathbb{R}^{N \times (K+L)}$, matriz cheia,

$S_0 \in \mathbb{R}^{N \times N}$, matriz esparsa (predominantemente diagonal),

$S_1 \in \mathbb{R}^{N \times M}$, matriz esparsa.

4.5 CONSTRUÇÃO DO SISTEMA LINEAR

Podemos agora reunir as expressões 4.2.4, 4.2.5, 4.3.1, 4.3.2 e 4.4.4 num único sistema linear com incógnitas q , Φ_0 , Φ_1 e Φ_2 como segue

$$\begin{bmatrix} P_1 & -I & 0 & 0 \\ P_2 & 0 & 0 & 0 \\ F & S_0 & S_1 & 0 \\ 0 & S_2 & D & S_3 \\ 0 & 0 & S_4 & E \end{bmatrix} \begin{bmatrix} q \\ \Phi_1 \\ \Phi_0 \\ \Phi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \Phi_B \\ 0 \\ c \\ 0 \end{bmatrix} \quad 4.5.1$$

Este sistema de equações é equivalente ao sistema esquematizado na figura 4.5.1, que inclui também as dimensões das submatrizes.

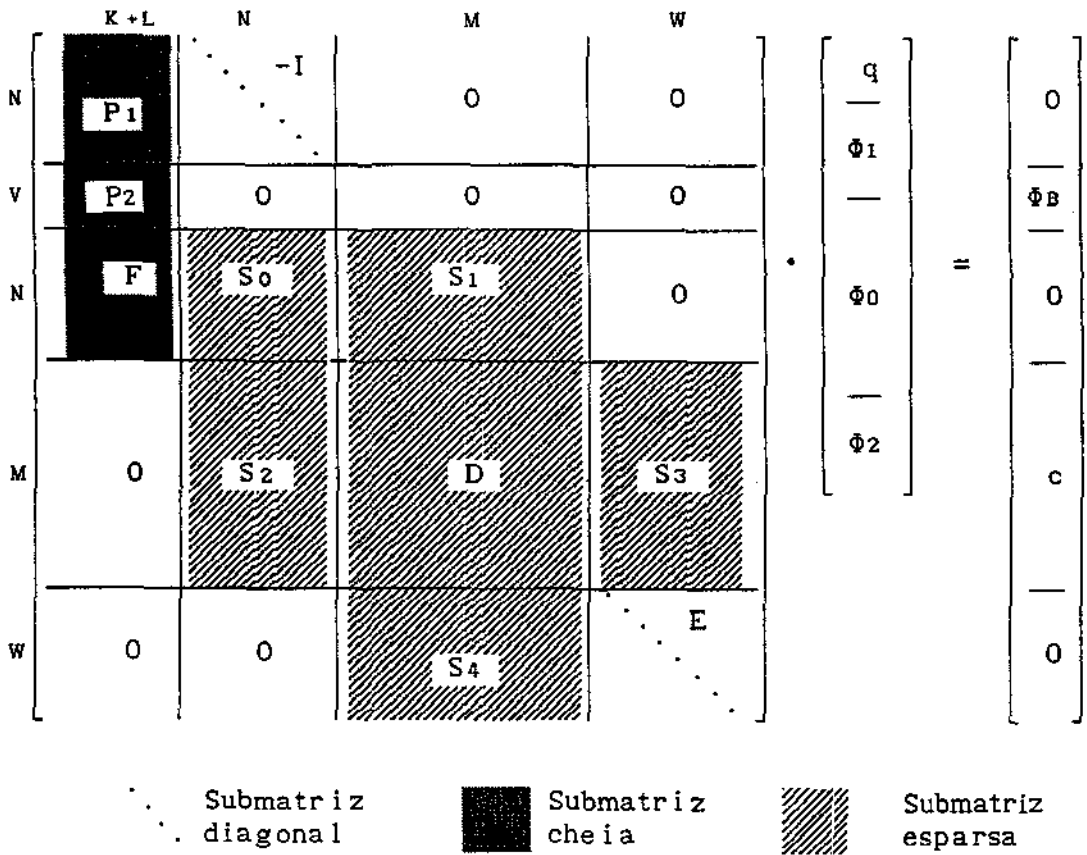


FIGURA 4.5.1. Sistema de equações representativo do método híbrido

Para facilitar a análise que se segue, este sistema de equações será reescrito na forma

$$\begin{bmatrix} P_1 & -I & 0 \\ P_2 & 0 & 0 \\ F & S_0 & Z_1 \\ 0 & Z_2 & A \end{bmatrix} \begin{bmatrix} q \\ \Phi_1 \\ \Phi \end{bmatrix} = \begin{bmatrix} 0 \\ \Phi_B \\ 0 \\ b \end{bmatrix} \quad 4.5.2$$

onde $A = \begin{bmatrix} D & S_3 \\ S_4 & E \end{bmatrix} \in \mathbb{R}^{(M+W) \times (M+W)}$, matriz esparsa ,

$\Phi = \begin{bmatrix} \Phi_0 \\ \Phi_2 \end{bmatrix} \in \mathbb{R}^{M+W}$, subvetor de incógnitas ,

$b = \begin{bmatrix} c \\ 0 \end{bmatrix} \in \mathbb{R}^{M+W}$, vetor esparsa ,

$Z_1 = [S_1 \ 0] \in \mathbb{R}^{N \times (M+W)}$, matriz esparsa ,

$Z_2 = \begin{bmatrix} S_2 \\ 0 \end{bmatrix} \in \mathbb{R}^{(M+W) \times N}$, matriz esparsa .

Uma vez reescrito na forma 4.5.2, podemos estabelecer um procedimento para a resolução do sistema de equações que representa o método híbrido.

4.6 RESOLUÇÃO DO SISTEMA LINEAR

O sistema linear 4.5.2 é composto por submatrizes com características diferentes, resultantes de problemas diferentes, ou seja,

1) Problema de quadrados mínimos, que origina submatrizes cheias e não quadradas (P_1 , P_2 , F),

2) Problema de diferenças finitas, que origina uma submatriz esparsa quadrada (A) não simétrica,

3) Problema de discretização na fronteira comum, que origina submatrizes esparsas quadrada (S₀) e não quadradas (Z₁, Z₂).

Deste modo, sugerimos que 4.5.2 seja resolvido por meio de um método iterativo que permita a separação dos problemas 1), 2) e 3). Assim, cada problema poderá ser resolvido independentemente, por meio do método mais apropriado a cada um.

Neste sentido, o sistema de equações 4.5.2 pode ser decomposto em

$$\begin{bmatrix} P_1 & -I \end{bmatrix} \begin{bmatrix} q \\ \Phi_1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \quad 4.6.1$$

$$\begin{bmatrix} P_2 & 0 & 0 \\ F & S_0 & Z_1 \end{bmatrix} \begin{bmatrix} q \\ \Phi_1 \\ \Phi \end{bmatrix} = \begin{bmatrix} \Phi_B \\ 0 \end{bmatrix} \quad 4.6.2$$

$$\begin{bmatrix} Z_2 & A \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \Phi \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} \quad 4.6.3$$

Notamos que uma vez conhecido o vetor de cargas q, obtemos os potenciais sobre a fronteira comum $\partial\Omega_1$ (vetor Φ_1) por meio de 4.6.1, o que por sua vez determina os potenciais em Ω_1 , Ω_2 e $\partial\Omega_2$ (vetor Φ) por meio de 4.6.3. Portanto, para a resolução completa do problema, basta a determinação do vetor de cargas q.

Assim, procuraremos uma expressão do tipo

$$q^{n+1} = T q^n + u \quad 4.6.4$$

onde $q \in \mathbb{R}^{K+L}$, $T \in \mathbb{R}^{(K+L) \times (K+L)}$ e $u \in \mathbb{R}^{K+L}$.

Pela teoria do ponto fixo aplicada a sistemas lineares (Golub [8], cap. 10), o processo iterativo 4.6.4 converge quando $\sigma(T) < 1$, onde $\sigma(T)$ é o raio espectral de T , para qualquer vetor inicial q^0 .

T é também conhecida como "matriz de iteração". Uma expressão para T pode ser obtida como segue.

Supomos inicialmente conhecido q^n . Utilizando a expressão 4.6.1, obtemos

$$\Phi_1^n = P_1 q^n \quad 4.6.5$$

o que é um simples produto matriz (cheia) - vetor.

A expressão 4.6.5 fornece os potenciais elétricos sobre $\partial\Omega_1$. Assim, o problema de diferenças finitas (Seção 4.2) é resolvido reescrevendo 4.6.3 como

$$A \Phi^n = b - Z_2 \Phi_1^n \quad 4.6.6$$

que representa um sistema linear onde A é uma matriz esparsa não simétrica. Este sistema pode ser resolvido utilizando um dos métodos dos Apêndices X, XI ou XII.

A resolução de 4.6.6 fornece, simbolicamente

$$\Phi^n = \begin{bmatrix} \Phi_0^n \\ \Phi_2^n \end{bmatrix} = A^{-1} (b - Z_2 \Phi_1^n) \quad 4.6.7$$

Uma vez obtidos os potenciais elétricos em todos os nós da região fechada (vetor Φ^n da expressão 4.6.7), podemos calcular o vetor intensidade de campo elétrico normal à fronteira comum $\partial\Omega_1$ resultante da aplicação da expressão 4.4.2 , representado pelo vetor auxiliar f^n ,

$$f^n = S_0 \Phi_1^n + Z_1 \Phi^n$$

Substituindo 4.6.7 nesta expressão obteremos

$$f^n = (S_0 - Z_1 A^{-1} Z_2) \Phi_1^n + Z_1 A^{-1} b$$

Utilizando a equação 4.6.2, onde designaremos o vetor q desta equação por r (apenas para facilitar a notação), teremos

$$\begin{bmatrix} P_2 \\ F \end{bmatrix} r^n = \begin{bmatrix} \Phi_B \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ f^n \end{bmatrix} = \begin{bmatrix} \Phi_B \\ -f^n \end{bmatrix}$$

Esta expressão representa um sistema linear com $V+N$ equações e $K+L$ incógnitas ($V+N \geq K+L$). Para a sua resolução podemos utilizar o método dos quadrados mínimos (vide Apêndice VII), por meio da decomposição $\begin{bmatrix} P_2 \\ F \end{bmatrix} = Q.R$, sendo Q matriz ortogonal e R matriz triangular superior. A solução r^n de quadrados mínimos será a solução de

$$R r^n = Q^T \begin{bmatrix} \Phi_B \\ -f^n \end{bmatrix} \quad 4.6.8$$

Este sistema linear é facilmente resolvido por "back-substitution". Simbolicamente teremos

$$r^n = R^{-1} Q^T \begin{bmatrix} \Phi_B \\ -f^n \end{bmatrix} \quad 4.6.9$$

Utilizando as equações 4.6.9 junto com 4.6.5, teremos

$$r^n = R^{-1} Q^T \begin{bmatrix} 0 \\ Z_1 A^{-1} Z_2 - S_0 \end{bmatrix} P_1 q^n + R^{-1} Q^T \begin{bmatrix} \Phi_B \\ -Z_1 A^{-1} b \end{bmatrix} \quad 4.6.10$$

Podemos tomar aqui, $q^{n+1} = r^n$ (vide expressão 4.6.4), e assim,

$$T = R^{-1} Q^T \begin{bmatrix} 0 \\ Z_1 A^{-1} Z_2 - S_0 \end{bmatrix} P_1$$

Isto porém representa um caso particular, onde alguns problemas podem conduzir a $\sigma(T) \geq 1$ não assegurando, portanto, a convergência.

Deste modo, sugerimos tomar um parâmetro θ , a ser arbitrado em cada problema, tal que torne $\sigma(T) < 1$. Podemos considerar

$$q^{n+1} = \theta q^n + (1-\theta) r^n, \quad 4.6.11$$

pois esta combinação assegura que ao final do processo iterativo, quando $r^n = q^n$, teremos $q^{n+1} = q^n = r^n$ para qualquer valor de θ .

Substituindo 4.6.10 em 4.6.11, chegaremos a

$$T = \theta I + (1-\theta) R^{-1}Q^T \begin{bmatrix} 0 \\ Z_1 A^{-1} Z_2 - S_0 \end{bmatrix} P_1 \quad 4.6.12$$

$$e, \quad u = (1-\theta) R^{-1}Q^T \begin{bmatrix} \Phi_B \\ -Z_1 A^{-1} b \end{bmatrix}, \quad 4.6.13$$

que são expressões para T e u do processo iterativo 4.6.4. A Seção 5.1 inclui um fluxograma relacionando as operações acima descritas.

Destes resultados podemos obter algumas conclusões importantes:

1) A análise de convergência independe dos potenciais elétricos dados do problema (estes potenciais compõe os vetores Φ_B e b), pois estes valores não entram na expressão 4.6.12.

2) A análise de convergência independe do valor inicial de q^0 , em conformidade com a teoria do ponto fixo. Porém, é possível obter uma boa estimativa para q^0 arbitrando Φ_1^0 (sobre o qual normalmente temos melhor intuição) e resolvendo o seguinte sistema linear obtido de 4.5.2,

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} q^0 = \bar{Q} \bar{R} q^0 = \begin{bmatrix} \Phi_1 \\ \Phi_B \end{bmatrix} \Rightarrow q^0 = \bar{R}^{-1} \bar{Q}^T \begin{bmatrix} \Phi_1 \\ \Phi_B \end{bmatrix} \quad 4.6.14$$

3) Outros fatores, além do parâmetro θ , afetam a convergência do método iterativo proposto (vide equação 4.6.12):

3.1) Quantidade e localização das cargas simuladas referentes ao MSC (afetam as submatrizes R, Q e P1).

3.2) O grau de refinamento e o tamanho da malha do MDF (interferem sobre todas as submatrizes da equação 4.6.12).

3.3) As regiões com potencial elétrico dado (Ω_A) (alteram A e eventualmente Z2).

3.4) Localização da região fechada relativa ao eixo $y = 0$ (altera R, Q e P1).

A complexidade da expressão 4.6.12 não permite estabelecer o grau de influência de cada uma das variáveis acima sobre $\sigma(T)$. Desta forma, é sugerido a escolha um problema típico, representativo da classe de problemas a estudar, de modo que seja possível uma avaliação da importância de cada um desses itens sobre $\theta_{\text{ótimo}}$, por meio de simulações computacionais.

CAPÍTULO 5

ASPECTOS COMPUTACIONAIS

Este capítulo trata de alguns algoritmos específicos necessários à implementação computacional do método híbrido, bem como de detalhes do pré-condicionamento utilizado para a resolução do problema de diferenças finitas. Mais detalhes da parte computacional são encontrados na referência [9].

5.1 MÉTODO ITERATIVO PARA A RESOLUÇÃO DO SISTEMA LINEAR

A parte computacional relativa ao método iterativo para a resolução do sistema de equações 4.5.2 pode ser resumida pelo fluxograma da figura 5.1.1, o qual é obtido a partir do procedimento descrito ao longo da Seção 4.6 .

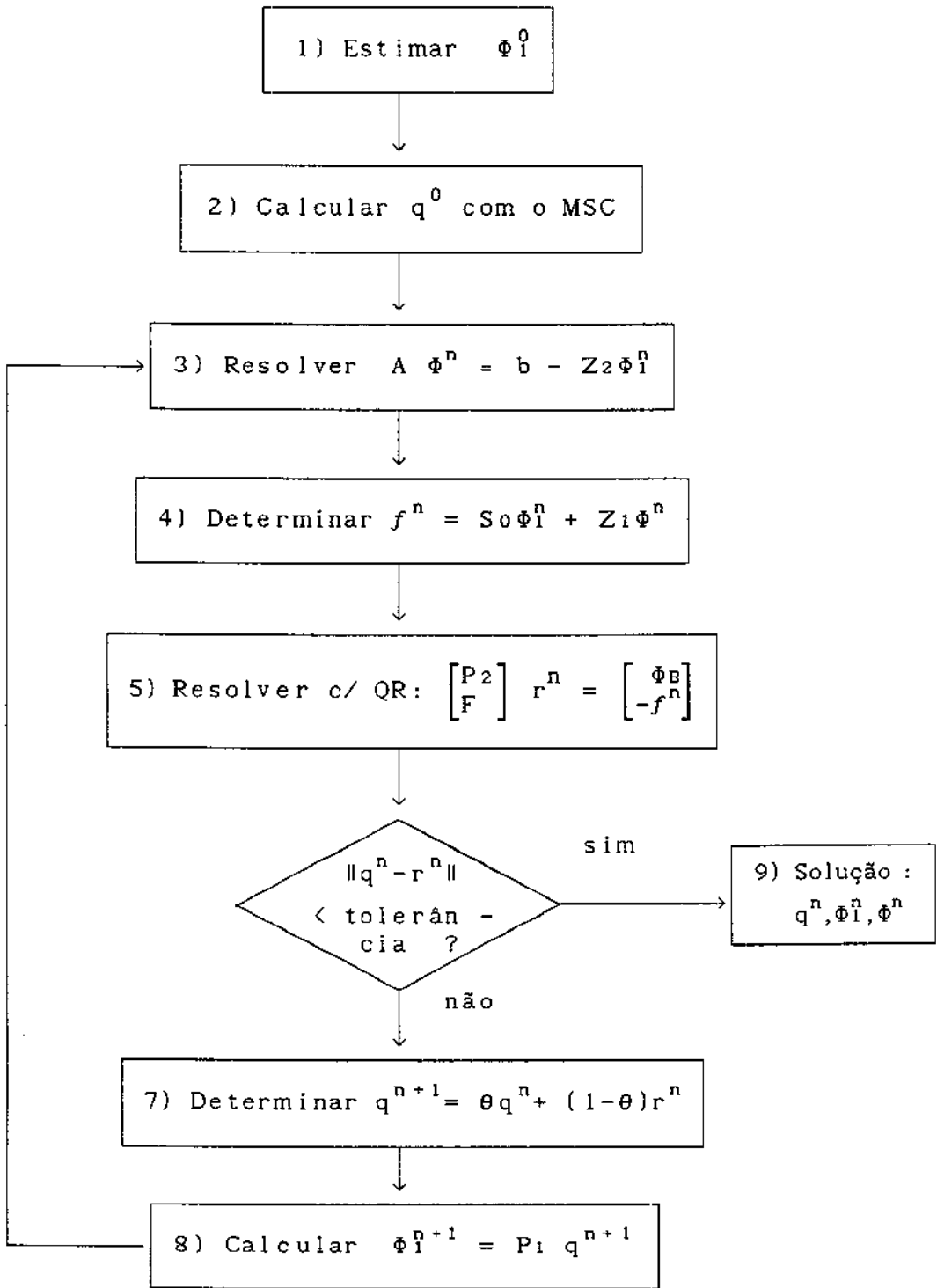


FIGURA 5.1.1 - Fluxograma do método iterativo

5.2 ALGORITMO PARA O PROBLEMA DE DIFERENÇAS FINITAS

O problema de diferenças finitas requer um algoritmo específico devido à existência de regiões com potencial elétrico dado (Ω_A) e de regiões com permissividade elétrica relativa diferentes (Ω_1 e Ω_2).

Vamos considerar um nó genérico k (figura 5.2.1), dado por suas coordenadas $X(k)$ e $Y(k)$, e vamos considerar que são dadas as distâncias da malha até $\partial\Omega_1$ (aux1, aux2, aux3 e aux4). Será necessário ainda considerar a eventual existência de uma ou mais regiões do tipo de Ω_A e Ω_2 .

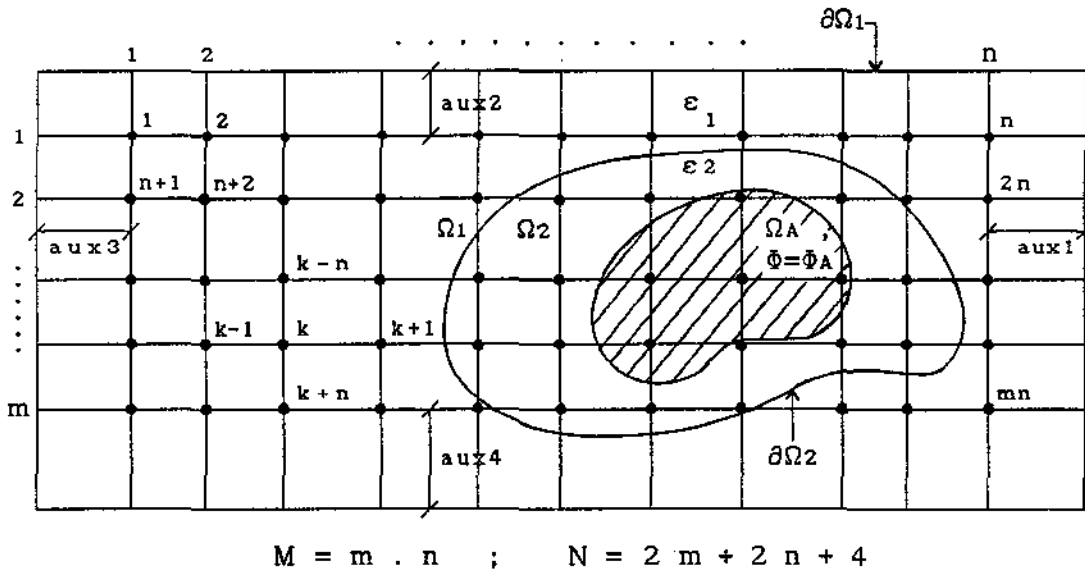


FIGURA 5.2.1 - Enumeração dos nós da malha de diferenças finitas

Aos nós da malha de diferenças finitas sobreposta à região fechada associamos um vetor (KOD) com M elementos. Este vetor identifica qual é a região à qual o respectivo nó pertence. Pode ser considerado, por exemplo, o seguinte critério:

KOD(k) = 0 indica que o nó "k" está situado sobre a
região Ω_1 ,

KOD(k) = +NUM indica que o nó "k" está situado sobre
uma região do tipo de Ω_A , de número NUM ,

KOD(k) = -NUM indica que o nó "k" está situado sobre
uma região do tipo de Ω_2 , de número NUM .

A matriz esparsa do problema de diferenças finitas será armazenada por meio de três vetores (ILIN, JCOL e VAL), conforme será exposto na Seção 5.3.1 .

Podemos agora construir um algoritmo para a geração do sistema de equações 4.2.4, representativo do problema de diferenças finitas incluso no método híbrido .

```

Definir vetor KOD(i), i=1...M
nelem=0
For i = 1 to m
  k0 = (i-1)*n
  For j = 1 to n
    k = k0+j
  
```

```

      If (KOD(k) > 0)      call ELEM(nelem,k,k,1,0)
                          c(k) =  $\Phi_A$  (k)
                          next j
      c(k) = 0
C   Definição de ph,qh,rh,sh
      ph=aux1; qh=aux2; rh=aux3; sh=aux4
      If (i≠1) qh = Y(k-n) - Y(k)
      If (i≠m) sh = Y(k) - Y(k+n)
      If (j≠1) rh = X(k) - X(k-1)
      If (j≠n) ph = X(k+1) - X(k)
C   Correção de ph,qh,rh,sh e do vetor c
      If (i≠1 e KOD(k-n)≠KOD(k)) corrigir qh
          If (KOD(k-n) > 0) corrigir c com  $\Phi_A$ 
      If (i≠m e KOD(k+n)≠KOD(k)) corrigir sh
          If (KOD(k+n) > 0) corrigir c com  $\Phi_A$ 
          If (KOD(k+n) < 0) enumerar nó sobre  $\partial\Omega_2$ 
      If (j≠1 e KOD(k-1)≠KOD(k)) corrigir rh
          If (KOD(k-1) > 0) corrigir c com  $\Phi_A$ 
      If (j≠n e KOD(k+1)≠KOD(k)) corrigir ph
          If (KOD(k+1) > 0) corrigir c com  $\Phi_A$ 
          If (KOD(k+1) < 0) enumerar nó sobre  $\partial\Omega_2$ 
C   Construção de D e S3, e correção do vetor c com  $c - S_2\Phi_1$ 
      If (i=1) corrigir c com  $\Phi_1$ 
          elseif (KOD(k-n) = 0 ) call ELEM(nelem,k,k-n,qh,sh)
          elseif (KOD(k-n) < 0 ) call ELEM(nelem,k, $\partial\Omega_2$ ,qh,sh)
      If (i=m) corrigir c com  $\Phi_1$ 
          elseif (KOD(k+n) = 0) call ELEM(nelem,k,k+n,sh,qh)
          elseif (KOD(k+n) < 0 ) call ELEM(nelem,k, $\partial\Omega_2$ ,sh,qh)
      If (j=1) corrigir c com  $\Phi_1$ 
          elseif (KOD(k-1) = 0) call ELEM(nelem,k,k-1,rh,ph)
          elseif (KOD(k-1) < 0 ) call ELEM(nelem,k, $\partial\Omega_2$ ,rh,ph)
      If (j=n) corrigir c com  $\Phi_1$ 
          elseif (KOD(k+1) = 0) call ELEM(nelem,k,k+1,ph,rh)
          elseif (KOD(k+1) < 0 ) call ELEM(nelem,k, $\partial\Omega_2$ ,ph,rh)
      nelem=nelem+1; ILIN(nelem)=k; JCOL(nelem)=k;
      VAL(nelem) = -1/(ph·rh)-1/(qh·sh)
      end
end
Subrotina ELEM(nelem,k1,k2,h1,h2)
      nelem=nelem+1; ILIN(nelem)=k1; JCOL(nelem)=k2;
      VAL(nelem) = 1/h1(h1+h2)
      return
end

```

5.3 ANÁLISE DE ESPARSIDADE

5.3.1 ESPARSIDADE NO MÉTODO DE DIFERENÇAS FINITAS

Para a resolução do sistema de equações 4.6.6, a matriz esparsa A deverá ser armazenada no computador apenas com seus elementos não nulos, que representam uma pequena percentagem do número total de elementos da matriz (da ordem de 1%, de um total de $M+W \times M+W$).

Podemos representar a matriz A por meio de três vetores,

- ILIN(i): vetor que armazena o número da linha referente ao i -ésimo elemento não nulo de A
- JCOL(i): vetor que armazena o número da coluna referente ao i -ésimo elemento não nulo de A
- VAL(i): vetor que armazena o valor numérico do elemento $A(\text{ILIN}(i), \text{JCOL}(i))$.

Com A armazenada deste modo, um produto matriz-vetor do tipo $y = A \cdot x$ poderá ser facilmente computado com o algoritmo

```
For i=1,NTOT
  y(ILIN(i)) = y(ILIN(i)) + VAL(i)*x(JCOL(i))
end
```

onde NTOT é o número total de elementos não nulos de A.

É fácil notar neste algoritmo que não é necessária nenhuma ordenação dos elementos de A durante a geração dos vetores ILIN, JCOL e VAL. Esta propriedade é especialmente apropriada para os problemas de discretização que pretendemos estudar.

Assim, para a solução de 4.6.6 podemos utilizar, em princípio, qualquer método que não requeira operações mais dispendiosas do que o produto matriz-vetor, o que torna aplicáveis os métodos propostos nos Apêndices X, XI e XII.

5.3.2 ESPARSIDADE NAS CONDIÇÕES DE CONTORNO

As submatrizes S_0 , Z_1 e Z_2 participam do método iterativo por meio dos produtos $S_0\Phi_1$, $Z_1\Phi$ e $Z_2\Phi_1$, conforme mostra o fluxograma da figura 5.1.1 .

Visto que S_0 contém apenas um elemento em cada linha e Z_1 dois elementos em cada linha (exceto nas linhas referentes aos nós dos cantos de $\delta\Omega_1$, onde S_0 tem três elementos e Z_1 tem estas linhas nulas), então S_0 e Z_1 poderão ser representadas por meio de vetores conforme segue:

$VET0(i)$, $VET1(1,i)$ e $VET1(2,i)$, que armazenam o valor numérico do elemento da linha "i" , respectivamente, e, $LK0(i)$, $LK1(1,i)$ e $LK1(2,i)$, que armazenam o número do elemento correspondente em Φ_1 e Φ , que será multiplicado por $VET0(i)$, $VET1(1,i)$ e $VET1(2,i)$, respectivamente.

Assim, nas linhas referentes aos cantos de $\partial\Omega_1$ ($i = 1, n+2, N-n-1$ e N), VET1 e LK1 poderão armazenar os elementos adicionais requeridos de S_0 nestas linhas.

Deste modo, o produto $PO = S_0\phi_1$ pode ser computado por:

```

For i = 1 to N
  PO(i) = VET0(i)*phi(LK0(i))
end
C Correções referentes aos cantos de  $\partial\Omega_1$  :
  PO(1) = PO(1) + VET1(1,1)*phi(LK1(1,1))
           + VET(2,1)*phi(LK1(2,1))
  PO(n+2) = PO(n+2) + VET1(1,n+2)*phi(LK1(1,n+2))
           + VET1(2,n+2)*phi(LK1(2,n+2))
  PO(N-n-1) = PO(N-n-1)
           + VET1(1,N-n-1)*phi(LK1(1,N-n-1))
           + VET1(2,N-n-1)*phi(LK1(2,N-n-1))
  PO(N) = PO(N) + VET1(1,N)*phi(LK1(1,N))
           + VET1(2,N)*phi(LK1(2,N))
end

```

De modo similar, $P_1 = Z_1 \Phi = [S_1 \ 0] \begin{bmatrix} \phi_0 \\ \phi_2 \end{bmatrix}$ pode ser computado por:

```

For i = 2 to N-1
  If (i=n+2 or i=N-n-1) next i
  P1(i) = VET1(1,i)*phi(LK1(1,i))
           + VET1(2,i)*phi(LK1(2,i))
end

```


Por sua vez, a submatriz Z_2 possui elemento não nulo apenas nas linhas correspondentes aos nós vizinhos a $\partial\Omega_1$. Cada uma destas linhas contém apenas um elemento, exceto as linhas correspondentes aos nós dos cantos da malha de diferenças finitas (4 linhas), que contém dois elementos não nulos cada. Assim, Z_2 poderá ser armazenada em dois vetores com dimensão $M+4$ (M = número de nós da malha), que conterão $N+4$ (N = número de nós em $\partial\Omega_1$) elementos não nulos, ou seja:

VET2(i) armazena o valor numérico do elemento

LK2(i) armazena o número do elemento correspondente em

Φ_1 , que será multiplicado por VET2(i)

Assim, o produto $P_2 = Z_2\Phi_1$ pode ser computado com o algoritmo:

```

For i = 1 to M
  If (LK2(i)=0) next i
P2(i) = VET2(i)* $\Phi_1$ (LK2(i))
end
P2(1) = P2(1) + VET2(M+1)* $\Phi_1$ (LK2(M+1))
P2(n) = P2(n) + VET2(M+2)* $\Phi_1$ (LK2(M+2))
P2(M-n+1) = P2(M-n+1) + VET2(M+3)* $\Phi_1$ (LK2(M+3))
P2(M) = P2(M) + VET2(M+4)* $\Phi_1$ (LK2(M+4))

```

5.4 PRÉ-CONDICIONAMENTO

5.4.1 INTRODUÇÃO

A matriz A resultante do problema de diferenças finitas e incluída na expressão 4.5.2, tem uma estrutura predominantemente penta-diagonal, em função da expressão 3.1.2, conforme ilustra a figura 5.4.1 (normalmente $M \gg W$).

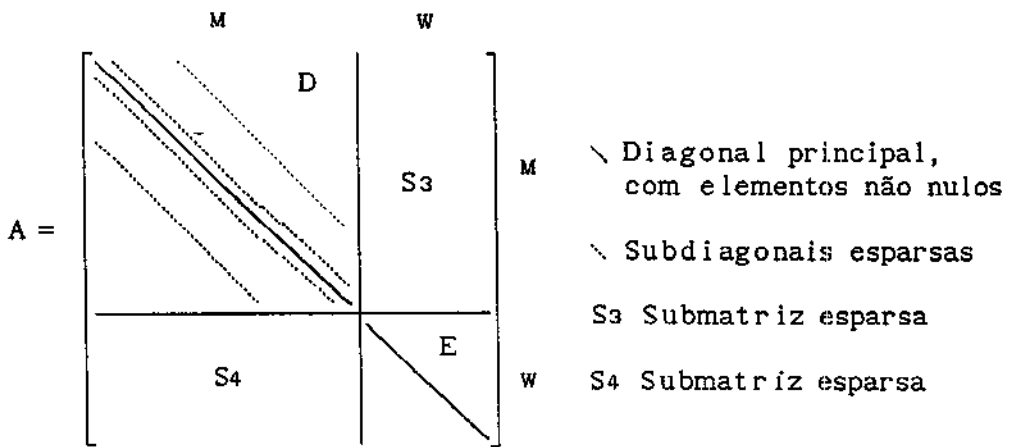


FIGURA 5.4.1 - Matriz resultante do problema de discretização

Assim, podemos esperar que um pré-condicionamento (vide Apêndice XIII) tridiagonal ou mesmo diagonal, possam ser convenientes.

5.4.2 PRÉ-CONDICIONAMENTO DIAGONAL

Será considerado o pré-condicionamento diagonal apenas para o método dos gradientes conjugados (Apêndice X), em virtude da relativa simplicidade para o cálculo computacional da diagonal principal de $A^T A$ (esse produto será necessário para a aplicação do método dos gradientes conjugados pois A não é simétrica).

Assim, podemos utilizar o algoritmo descrito no Apêndice XIII.2, com

$$M = \text{diag}(A^T A)$$

onde $\text{diag}(A^T A)$ é uma matriz esparsa, composta pela diagonal principal do produto $A^T A$.

5.4.3 PRÉ-CONDICIONAMENTO TRIDIAGONAL

Será considerado o pré-condicionamento tridiagonal para o método dos subespaços de Krylov (Apêndice XI) e para o método dos gradientes conjugados quadráticos (Apêndice XII), utilizando no Apêndice XIII.3 e XIII.4, respectivamente,

$$C = \text{trid}(A) , D = I$$

onde $\text{trid}(A)$ é uma matriz esparsa, composta pela parte tridiagonal da matriz A .

No caso do método dos gradientes conjugados, a matriz C deve ser simétrica e definida positiva. Tomando uma matriz B , bidiagonal de A (por exemplo, composta pela diagonal principal e pela diagonal inferior), temos que o produto $B^T B$ será uma matriz tridiagonal, simétrica e definida positiva. Assim, poderemos considerar um pré-condicionamento tridiagonal no Apêndice XIII.2, utilizando

$$M = B^T B$$

CAPÍTULO 6

APLICAÇÕES

Neste capítulo serão apresentados alguns exemplos didáticos da aplicação do método para cálculo de campos elétricos exposto no Capítulo 4 (Seções 6.1, 6.2 e 6.3), bem como serão apresentados exemplos práticos de sua aplicação em problemas com um material dielétrico (Seção 6.4) e com mais de um material dielétrico (Seção 6.5).

6.1 VERIFICAÇÃO DA PRECISÃO DOS CÁLCULOS

Nesta seção será feita uma verificação da precisão dos resultados em função da técnica utilizada para o tratamento da condição de continuidade $\mathbf{n} \cdot \mathbf{D}$ sobre $\partial\Omega_2$, apresentada na Seção 4.2 (expressão 4.2.3e).

Segundo a técnica proposta, a discretização para a derivada normal requer três nós internos à região além do nó sobre o contorno, conforme exposto no Apêndice VI.2.1. É também possível uma versão simplificada que utiliza apenas dois nós internos, além do nó sobre o contorno, conforme Apêndice VI.2.2.

Tomaremos o cilindro infinito da figura 2.2.1, já

utilizado anteriormente como exemplo de aplicação do MSC (Seção 2.2) e do MDF (Seção 3.2). Procedendo de modo similar ao Capítulo 3, Seção 3.2, arbitraremos uma região fechada Ω (figura 6.1.1) em cujos contornos $\partial\Omega_1$ e $\partial\Omega_A$ os respectivos potenciais elétricos Φ_1 e Φ_A são assumidos conhecidos. O vetor Φ_1 poderá ser dado por meio da solução analítica do problema, indicada na seção 2.2 .

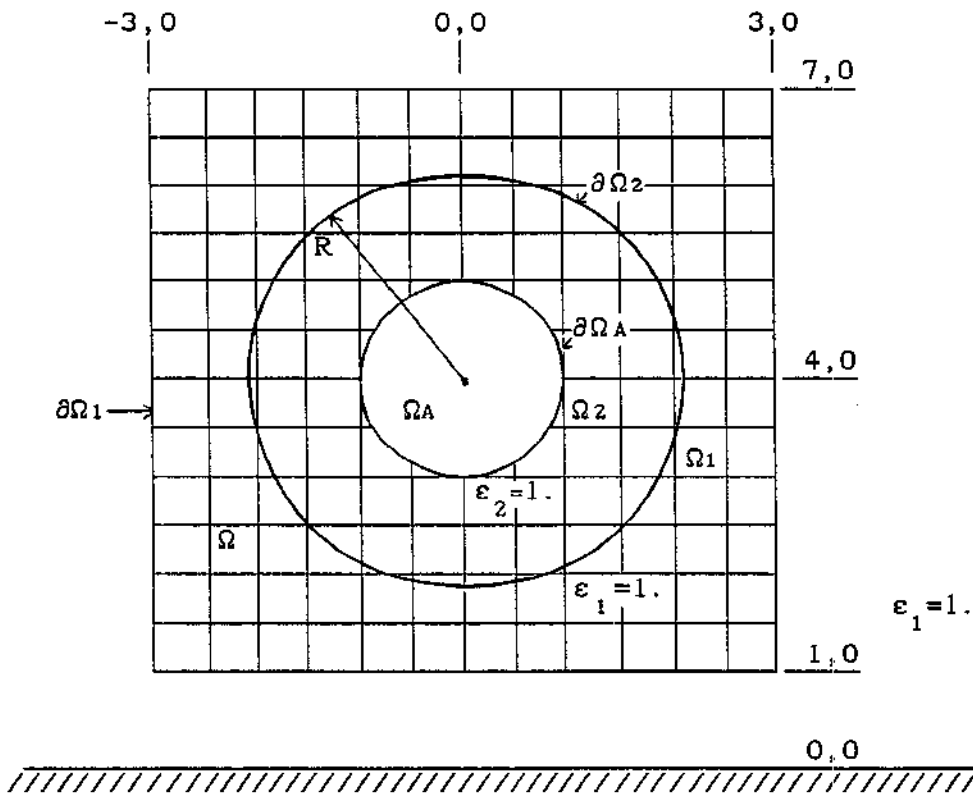


FIGURA 6.1.1 - Cilindro de comprimento infinito

Arbitraremos um contorno $\partial\Omega_2$ (de forma circular com raio R), que definirá $\Omega_1 \cup \Omega_2 = \Omega$. O problema assim colocado é

equivalente ao problema da região discretizada generalizado na figura 4.1.1 .

Com o objetivo de obtermos uma estimativa da ordem de grandeza do erro introduzido pelos métodos de discretização da derivada normal dados nos Apêndices VI.2.1 e VI.2.2, tomaremos $\epsilon_1 = \epsilon_2 = 1,0$. Deste modo, os resultados obtidos com a aplicação destes métodos serão comparados com a solução analítica dada na seção 2.2 .

A tabela 6.1.1 indica a máxima diferença em valor absoluto verificada para os potenciais elétricos calculados sobre os nós de Ω , em função do passo h considerado para a malha ($R = 1,75$ a $1,875$), utilizando o programa computacional MDF, descrito na referência [9], cap. 8 .

TABELA 6.1.1

| Máxima diferença no potencial elétrico calculado sobre $\Omega_1 \cup \Omega_2$ (V) | | | |
|---|------------------------------------|--------|-----------------------------|
| h | Comparação da solução analítica c/ | | Comparação VI.2.1 c/ VI.2.2 |
| | VI.2.1 | VI.2.2 | |
| 0,500 | 2,45 | 6,25 | 5,68 |
| 0,333 | 0,26 | 2,31 | 2,50 |
| 0,250 | 0,14 | 1,59 | 1,68 |
| 0,125 | 0,062 | 0,69 | 0,71 |

Podemos concluir, por exemplo, que o método que utiliza 3 nós internos (coluna VI.2.1) conduz a uma precisão sensivelmente melhor do que o método que utiliza apenas 2 nós (coluna VI.2.2).

Diferenças relativamente grandes foram obtidas com $h = 0,500$ devido ao fato de que, neste caso, a região Ω_2 apresenta uma discretização muito rudimentar, com poucos nós.

6.2 CILINDRO DE COMPRIMENTO INFINITO

O problema do cilindro de comprimento infinito da figura 2.2.1 foi resolvido com o método híbrido exposto no Capítulo 4, e os resultados foram comparados com a solução analítica dada na seção 2.2.

A tabela 6.2.1 indica a máxima diferença verificada no valor dos potenciais elétricos calculados sobre os nós de $\partial\Omega_1$, utilizando o programa computacional CCE referente ao método híbrido, descrito na referência [9].

TABELA 6.2.1

| Comparação do método híbrido com a solução analítica | |
|--|---|
| h | Diferença máxima sobre $\partial\Omega_1$ (V) |
| 1,000 | 2,448 |
| 0,500 | 0,310 |
| 0,333 | 0,128 |
| 0,250 | 0,068 |

Nestes cálculos foi considerada uma relação L/N (número de cargas/número de nós na fronteira) $\cong 0,50$, sendo que as cargas foram localizadas próximas a $\partial\Omega_1$, a uma distância de $2,0 h$.

Concluimos deste exemplo que o método híbrido poderá fornecer resultados com precisão melhor do que 1 % (referido ao máximo valor de potencial elétrico envolvido no problema, que é 100 V), dependendo do nível de discretização utilizado. O cálculo com $h = 1,0$ não apresentou bons resultados devido à pequena discretização para $\partial\Omega$ conseqüente deste caso.

6.3 CILINDROS PARALELOS

6.3.1 DESCRIÇÃO DO PROBLEMA

Suponhamos dois cilindros de comprimento infinito e paralelos à superfície da terra (figura 6.3.1.1), ambos com potencial elétrico de 100 V, e situados relativamente distantes do solo.

Arbitramos uma malha de dimensões 15×27 (figura 6.3.1.2) em torno destes cilindros (região Ω), com $M = 379$ nós. Temos então, $N = 88$, e tomamos $L/N \cong 0,50$.

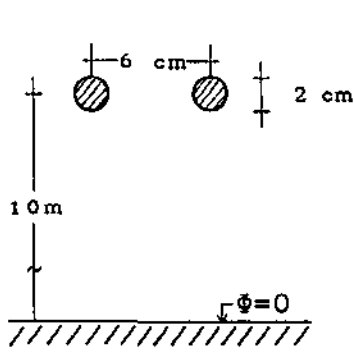


FIGURA 6.3.1.1 - Cí-
cindros paralelos

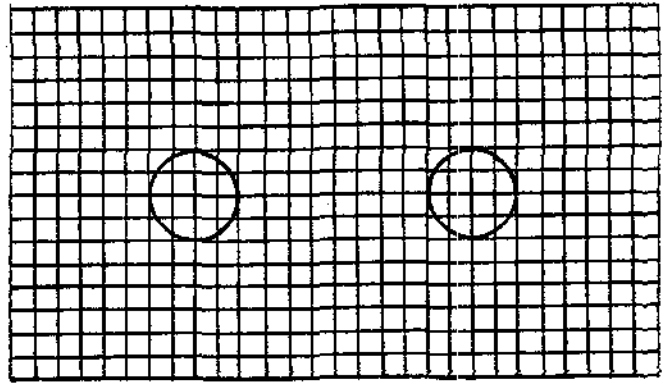


FIGURA 6.3.1.2 - Malha de
diferenças finitas

6.3.2 CURVAS EQUÍPOTENCIAIS

A aplicação do método híbrido nos dá valores numéricos de Φ_1 (sobre $\partial\Omega$) e Φ_0 (sobre Ω), com os quais obtemos as curvas equipotenciais indicadas na figura 6.3.2.1, mediante a utilização de programas de computador apropriados que geram a interpolação necessária.

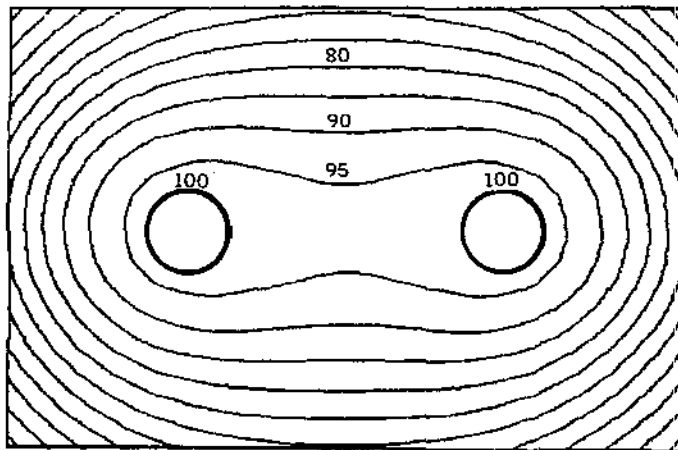


FIGURA 6.3.2.1 - Curvas equipotenciais

Este mesmo problema, porém considerando os cilindros com potenciais $+100\text{ V}$ e -100 V respectivamente, resulta nas curvas eqüipotenciais indicadas na figura 6.3.2.2.

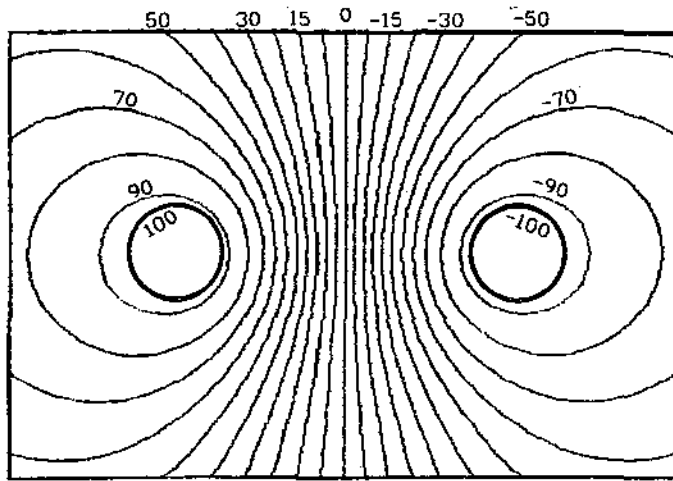


FIGURA 6.3.2.2 - Curvas eqüipotenciais

Podemos observar que a simples alteração nos potenciais elétricos dados sobre a superfície dos cilindros pode modificar completamente a distribuição de Φ sobre Ω . Em ambos os casos, o mesmo valor de θ conduziu à convergência do método iterativo, conforme conclusão 1 do final da Seção 4.6.

Notamos ainda na figura 6.3.2.2, a existência de uma linha de potencial elétrico nulo eqüidistante dos cilindros. Este resultado já era esperado visto que os cilindros possuem potenciais elétricos de mesmo valor e sinal contrário.

6.3.3 PROBLEMA DE QUADRADOS MÍNIMOS

A tabela 6.3.3.1 indica a máxima diferença obtida nos valores absolutos dos potenciais elétricos calculados sobre os nós de $\partial\Omega_1$, em função da relação L/N (nº de cargas simuladas em Ω / nº de nós sobre $\partial\Omega_1$), tomando como padrão de comparação L/N \cong 0,75 .

TABELA 6.3.3.1

| Máxima diferença no potencial elétrico calculado sobre $\partial\Omega_1$ | |
|---|---------|
| L/N | V |
| 0,75 | padrão |
| 0,50 | 0,00071 |
| 0,40 | 0,0060 |
| 0,30 | 0,017 |
| 0,20 | 0,096 |
| 0,10 | 1,05 |

Concluimos desta tabela que é possível a aplicação do método dos quadrados mínimos como meio para reduzir as dimensões do sistema de equações representativo do método híbrido de simulação de cargas e diferenças finitas, sem perda significativa na precisão dos resultados.

Pelo obtido neste exemplo, uma relação L/N \cong 20% seria suficiente para produzir resultados satisfatórios (precisão melhor do que 0,1%, referida ao máximo valor de potencial do problema, que é 100 V).

6.3.4 MÉTODO PARA A RESOLUÇÃO DO PROBLEMA DE DIFERENÇAS FINITAS

A escolha do método para a resolução do problema de diferenças finitas (Apêndices X, XI, XII e XIII) pode ser feita em função do tempo de CPU requerido para os cálculos. A tabela 6.3.4.1 mostra o tempo de CPU requerido para a resolução do exemplo em questão no computador IBM 3090-300S da COPEL (Cia. Paranaense de Energia), com $L/N \approx 0,33$.

TABELA 6.3.4.1

| Tempo de CPU (s) | | | | |
|------------------|---|-----------------|-----------------------------|--------------------------|
| | Método | sem precond. | com precond. tridiagonal | com precond. diagonal |
| 1 | Gradientes conjugados | 37,51 | 31,38 | 35,00 |
| 2 | Subespaços de Krylov | | | |
| | Dimensão = 5 | 18,93 | 15,20 | |
| | Dimensão = 10 | 18,94 | 18,03 | |
| | Dimensão = 15 | 30,73 | 28,53 | |
| 3 | Gradientes conjugados quadráticos (CGS) | 8,31 | 6,94 | |

Notamos que o método CGS (gradientes conjugados quadráticos), descrito no Apêndice XII apresentou melhor desempenho neste caso.

A pequena diferença resultante das aplicações do pré-condicionamento explica-se provavelmente pela matriz do problema ser relativamente bem condicionada.

6.3.5 LOCALIZAÇÃO DAS CARGAS SIMULADAS

A existência de singularidades nas expressões dos coeficientes de potencial elétrico p_{ij} (expressão III.1.3) e de campo elétrico f_{ij} (expressões III.2.1 e III.2.2) sugere que as cargas elétricas não devem ser arbitradas muito próximas a qualquer ponto da discretização do contorno.

A tabela 6.3.5.1 mostra a máxima diferença encontrada nos potenciais elétricos calculados sobre os nós de $\partial\Omega_1$ quando comparados com os potenciais calculados nestes nós pelo MSC, em função da localização das cargas no interior de Ω .

TABELA 6.3.5.1

| Máxima diferença no potencial elétrico calculado sobre $\partial\Omega_1$ | |
|---|--------|
| Dist. a $\partial\Omega_1$ | V |
| 3,0 x h | 0,0499 |
| 2,0 x h | 0,0539 |
| 1,0 x h | 0,215 |
| 0,50 x h | 1,028 |
| 0,25 x h | 1,883 |

Assim, tendo em vista a precisão esperada nos cálculos com o método híbrido (vide Seção 6.2), podemos sugerir que as cargas sejam localizadas a uma distância mínima de cerca de 2,0 h da fronteira $\partial\Omega$.

6.4 LINHA DE TRANSMISSÃO DE ENERGIA ELÉTRICA

No projeto de linhas de transmissão de energia elétrica de alta tensão (69.000 a 800.000 V), é fundamental o conhecimento dos níveis de campo elétrico em torno dos cabos energizados. Se estes níveis superam o tolerável pelo ar atmosférico, tem início o fenômeno chamado de "efeito corona", que tem implicações diretas com a economia das empresas de energia (as perdas de energia por quilômetro de linha são da ordem de US\$ 10000 por ano) e com o meio ambiente (ruído audível, interferências em rádio e TV, etc).

Este fenômeno pode ser controlado com a escolha adequada do potencial elétrico a ser imposto aos cabos, seu diâmetro, quantidade, localização, etc. Enfim, este é um problema típico (tridimensional com simetria plana) que pode ser analisado por meio do método híbrido exposto nos capítulos anteriores.

A figura 6.4.1 mostra a disposição dos cabos elétricos em uma linha de transmissão de energia elétrica (dimensões em cm).

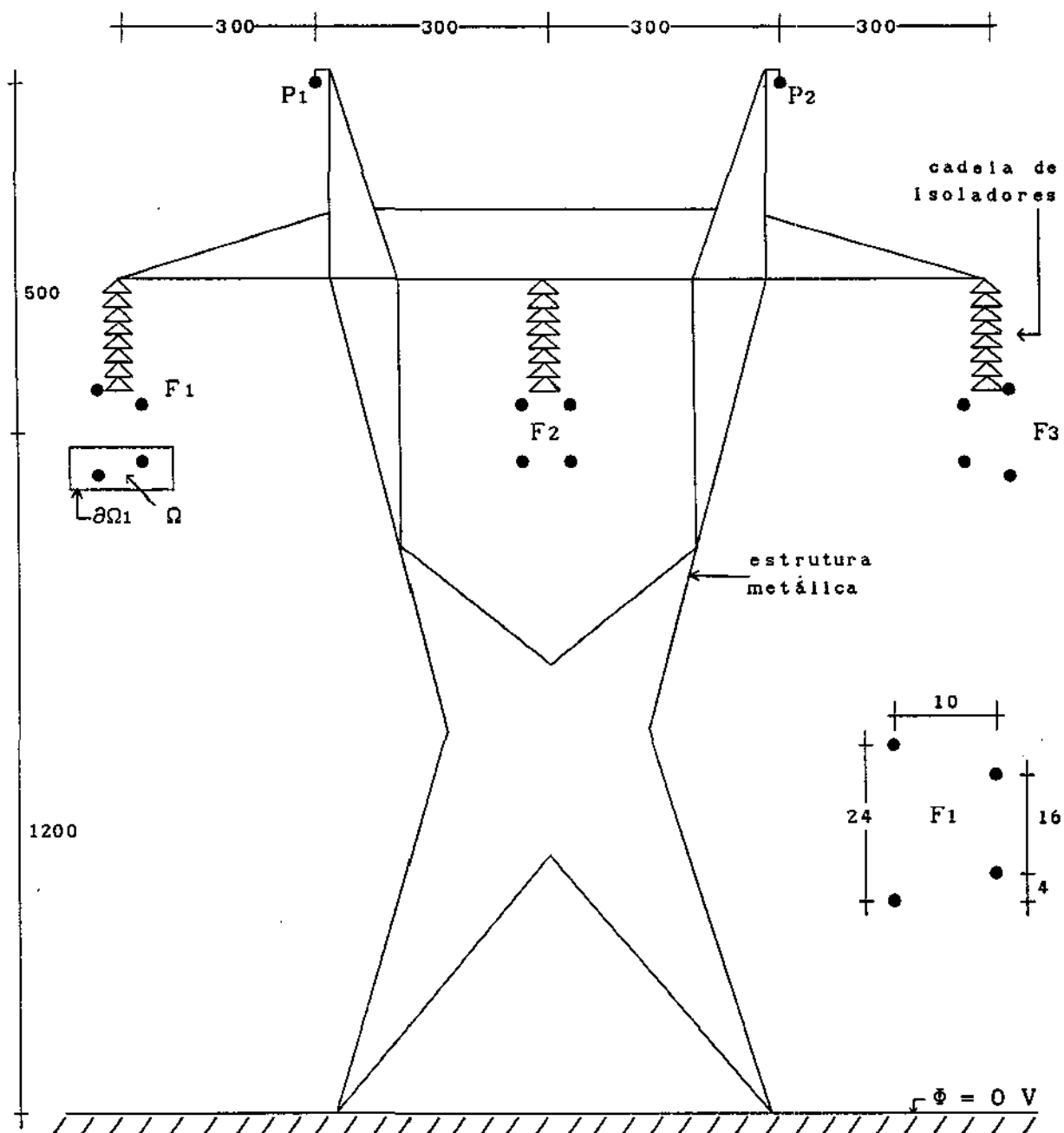


FIGURA 6.4.1 - Linha de transmissão de energia elétrica

P_1 e P_2 são chamados de cabos "pára-raios" e F_1 , F_2 e F_3 são denominados de "fases", cada uma composta de 4 cabos energizados. Os potenciais elétricos em relação à terra oscilam de forma senoidal, e são dados por

$$\Phi(F_1) = 296 \text{ sen } (2\pi f t) \quad \text{kV}$$

$$\Phi(F_2) = 296 \text{ sen } (2\pi f t + 2\pi/3) \quad \text{kV}$$

$$\Phi(F_3) = 296 \text{ sen } (2\pi f t + 4\pi/3) \quad \text{kV}$$

$$\Phi(P_1) = 0. \quad \text{V}$$

$$\Phi(P_2) = 0. \quad \text{V}$$

onde f = frequência da onda senoidal, igual a 60 hertz,

t = variável de tempo, em segundos.

Apesar de que este exemplo envolve potenciais elétricos variáveis no tempo, o fato do comprimento de uma onda de 60 Hz ser muito grande (cerca de 5000 km) permite analisar o caso como um problema de eletrostática (Olsen [14]).

Vamos supor que estaremos interessados em analisar a distribuição de potenciais elétricos em torno de F_1 . Assim, escolheremos o instante t onde $\Phi(F_1)$ é máximo, ou seja, $\Phi(F_1) = 296$. kV, $\Phi(F_2) = \Phi(F_3) = -148$. kV, e $\Phi(P_1) = \Phi(P_2) = 0$. V

Para exemplificar este caso, arbitramos uma região retangular $\partial\Omega_1$ abrangendo parte de F_1 , no interior da qual escolhemos uma malha irregular de dimensões 19×25 (figura 6.4.2), onde $M = 449$ nós ($N = 92$, $L/N \cong 0,50$).

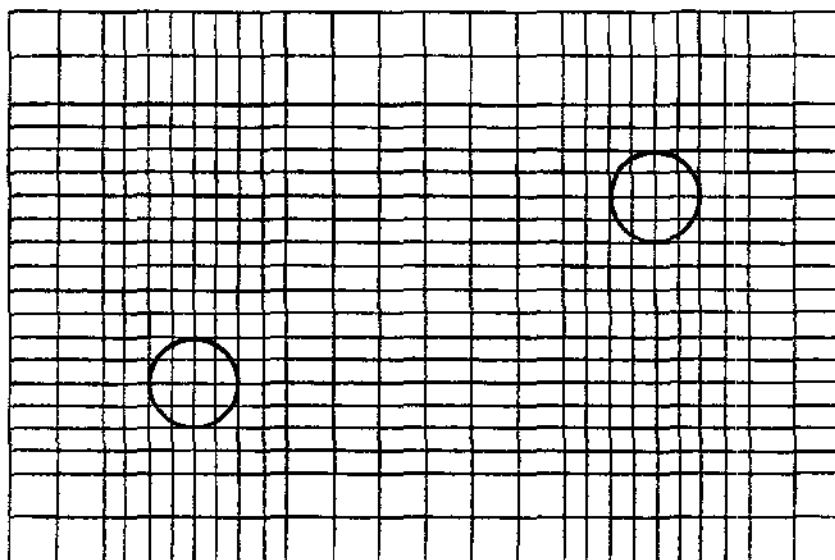


FIGURA 6.4.2 - Discretização de diferenças finitas

A aplicação do método híbrido nos dá os valores de Φ_0 e Φ_1 , com os quais obtemos as curvas eqüipotenciais indicados na figura 6.4.3.

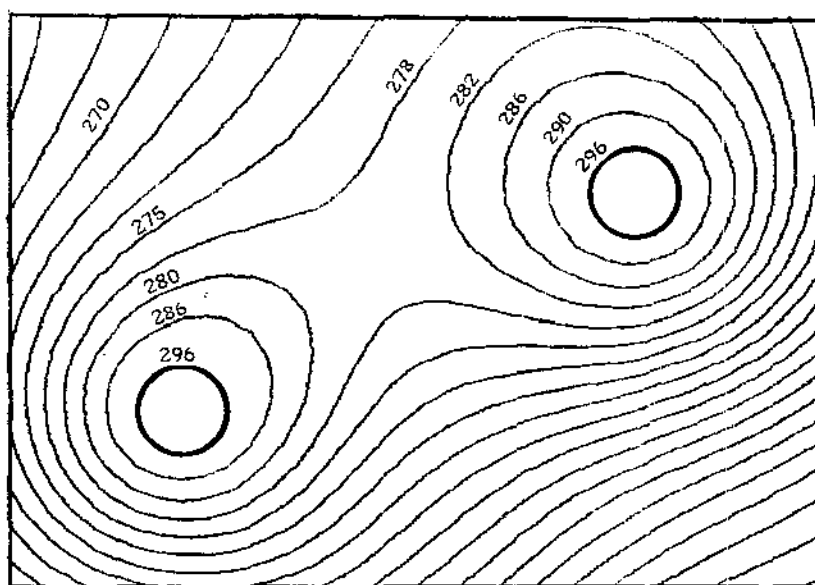


FIGURA 6.4.3 - Curvas eqüipotenciais

A partir da análise destes gráficos e/ou dos valores numéricos calculados de Φ sobre Ω , poderemos concluir pela viabilidade ou não da construção de uma linha de transmissão com as características deste exemplo, em função dos valores máximos admissíveis para Φ e/ou E .

6.5 PROBLEMA COM VÁRIOS MATERIAIS DIELÉTRICOS

6.5.1 DESCRIÇÃO DO PROBLEMA

A figura 6.5.1.1 representa um caso bidimensional que ilustra a aplicação do método híbrido em problemas com mais de um material dielétrico, estando associado com casos práticos envolvendo condutores com isolamento externo (unidades arbitrárias).

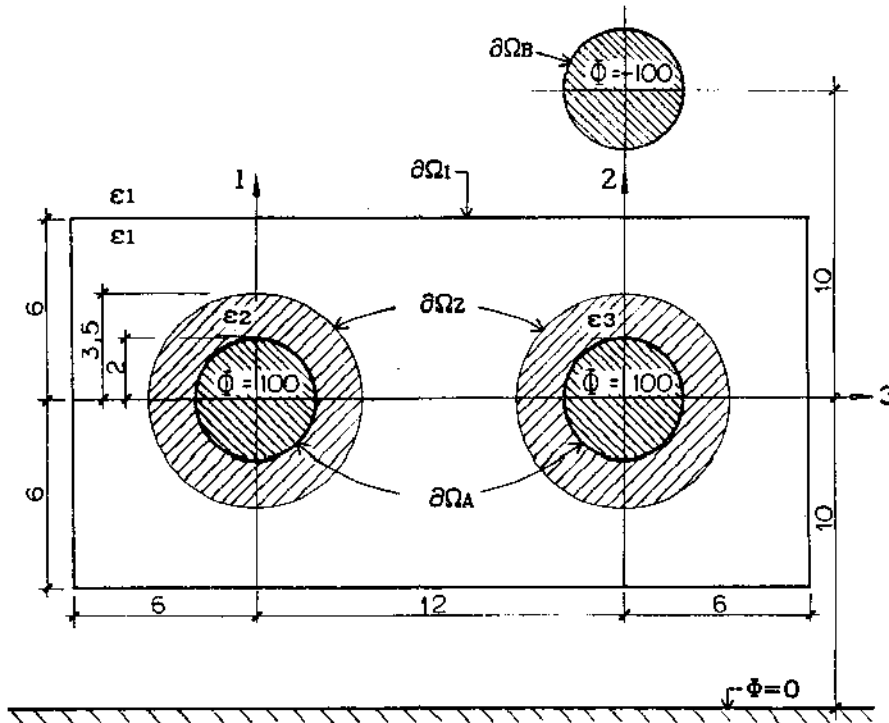


FIGURA 6.5.1.1 - Problema com vários dielétricos

No interior da região retangular a ser discretizada arbitramos uma malha com passo constante $h = 0,75$ (dimensões 15×31), o que gera $M = 423$, $N = 96$ e $W = 72$. Para a representação do eletrodo externo B, foi escolhido $K = V = 4$.

6.5.2 CURVAS EQÜIPOTENCIAIS E PERFIS DE POTENCIAL ELÉTRICO

A figura 6.5.2.1 mostra a distribuição de curvas eqüipotenciais no interior da região discretizada, considerando o caso $\epsilon_1 = 1,0$, $\epsilon_2 = 2,0$ e $\epsilon_3 = 5,0$.

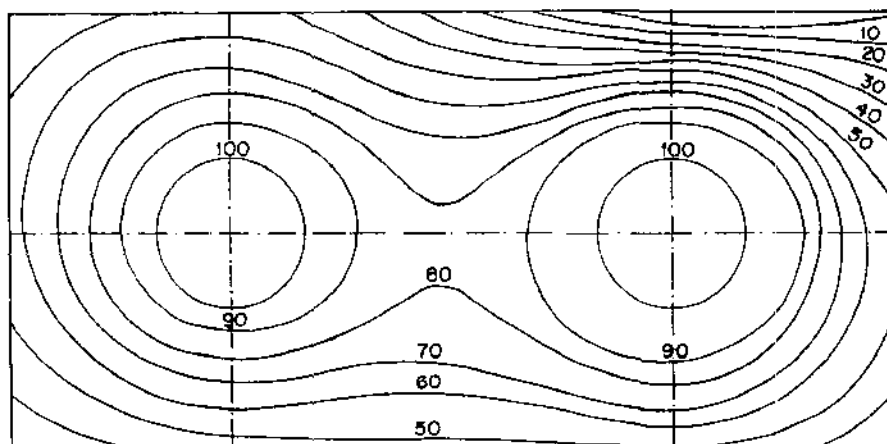


FIGURA 6.5.2.1 - Curvas eqüipotenciais

A figura 6.5.2.2 mostra a distribuição da função potencial elétrico ao longo das direções 1 e 2 da figura 6.5.1.1, calculada com $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1,0$, ilustrando o efeito da proximidade do eletrodo B.

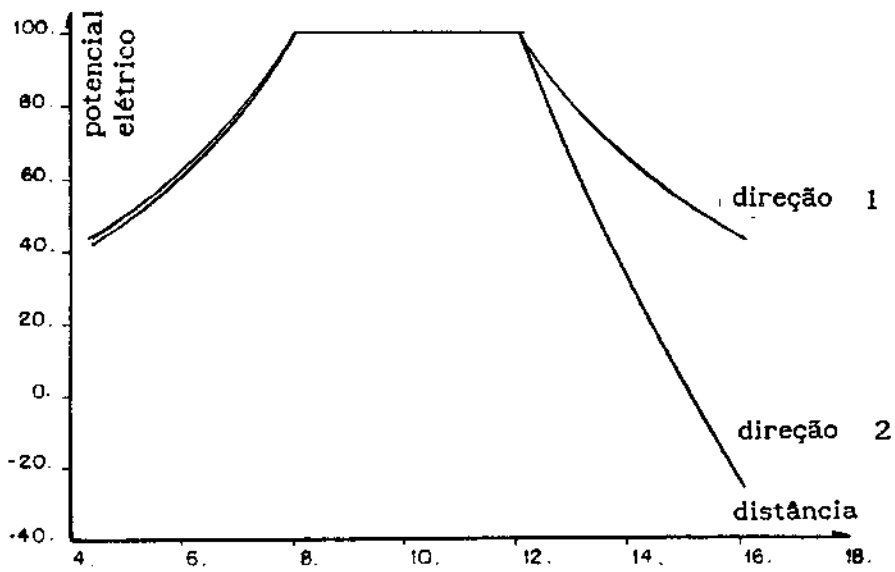


FIGURA 6.5.2.2 - Distribuição do potencial elétrico

A figura 6.5.2.3 mostra a distribuição da função potencial elétrico ao longo da direção 3, em função da variação no parâmetro permissividade elétrica ϵ .

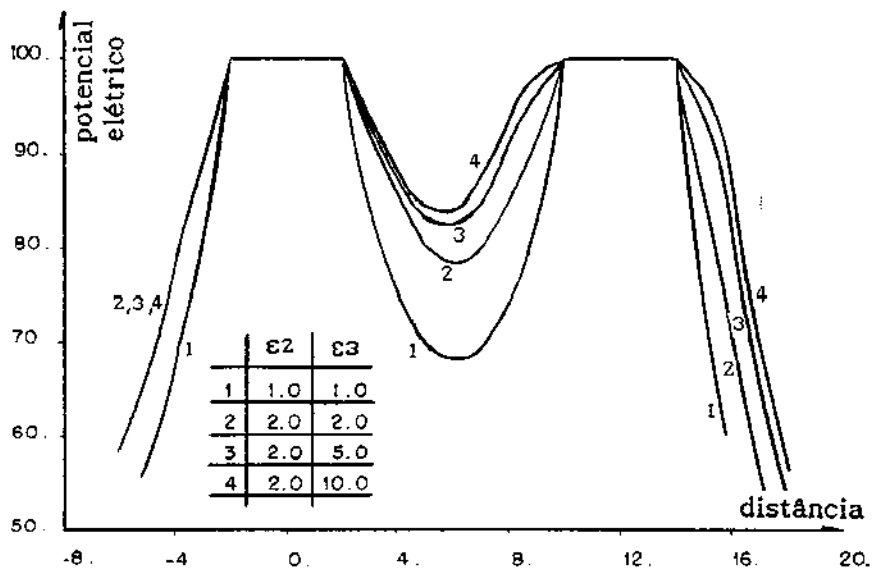


FIGURA 6.5.2.3 - Distribuição do potencial elétrico

6.5.3 PROBLEMA DE QUADRADOS MÍNIMOS

De modo similar ao caso da Seção 6.3.3, a tabela 6.5.3.1 indica a máxima diferença obtida nos potenciais elétricos calculados sobre os nós de $\partial\Omega_1$, em função da relação L/N (nº de cargas simuladas / nº de nós sobre $\partial\Omega_1$), tomando como padrão de comparação $L/N \cong 0,75$. Foi considerado o caso com $\epsilon_1 = 1,0$, $\epsilon_2 = 2,0$ e $\epsilon_3 = 5,0$.

TABELA 6.5.3.1

| Diferença no potencial elétrico calculado sobre $\partial\Omega_1$ | |
|--|--------|
| L/N | V |
| 0,75 | padrão |
| 0,50 | 0,0150 |
| 0,40 | 0,0299 |
| 0,30 | 0,173 |
| 0,20 | 1,165 |
| 0,10 | 5,782 |

Notamos que uma relação L/N $\cong 30\%$ a 50% conduz a resultados com precisão satisfatória.

6.5.4 MÉTODO PARA A RESOLUÇÃO DO PROBLEMA DE DIFERENÇAS FINITAS

A tabela 6.5.4.1 mostra o tempo de CPU requerido para a resolução do exemplo em questão no computador IBM 3090-300S da COPEL (com $L/N \cong 0,33$).

TABELA 6.5.4.1

| Tempo de CPU (s) | | | | | | | |
|--|---|--|-------|---------------------------|------|-----------------------|------|
| | Método | sem precond. | | c/precond. tridiagonal | | c/precon. diagonal | |
| | | A | B | A | B | A | B |
| 1 | Gradientes conjugados | 21,3 | 56,7 | 16,5 | 37,3 | 19,3 | 37,9 |
| 2 | Subespaços de Krylov | | | | | | |
| | Dimensão = 5 | 10,4 | 107,1 | 8,9 | 12,9 | | |
| | Dimensão = 10 | 10,2 | 74,5 | 9,7 | 15,2 | | |
| | Dimensão = 20 | 23,3 | 91,9 | 18,7 | 31,3 | | |
| 3 | Gradientes conjugados quadráticos (CGS) | 4,7 | 15,3 | 5,3 | 7,9 | | |
| A - Problema sem $\partial\Omega_2$ ($\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 1,0$) | | B - Problema com $\partial\Omega_2$, $\varepsilon_1 = 1,0 / \varepsilon_2 = 2,0 / \varepsilon_3 = 2,0$ | | | | | |

Assim como no exemplo da Seção 6.3, o método CGS (gradientes conjugados quadráticos), descrito no Apêndice XII apresentou melhor desempenho.

Este exemplo deixa claro a vantagem da utilização de pré-condicionamento, visto que a inclusão de condições de continuidade sobre $\partial\Omega_2$ (caso B da tabela 6.5.4.1) torna a matriz do problema de diferenças finitas mal condicionada.

6.6 OBSERVAÇÕES

A aplicação do método híbrido nos exemplos deste capítulo e em diversas outras simulações permitiu ainda a constatação dos seguintes pontos:

1) Uma tolerância de 1% a ser considerada para o vetor de cargas elétricas no processo iterativo principal (vide fluxograma da figura 5.1.1) é suficiente (não foi verificada diferença significativa nos resultados utilizando tolerância de 0,1%) .

2) Para a resolução do problema de diferenças finitas por algum dos métodos dos Apêndices X, XI ou XII, uma precisão de 10^{-5} (tanto para o valor absoluto da norma infinito do vetor do resíduo

como para o valor absoluto da norma infinito do vetor de incógnitas) mostrou-se suficiente.

3) Deve ser evitada a localização de cargas elétricas muito próximas entre si, pois isto tenderá a tornar as submatrizes P_1 e F submatrizes com posto incompleto (suas colunas tenderão a ser linearmente dependentes).

4) Chamando $[\theta_1; \theta_2]$ à faixa de convergência do parâmetro θ , e θ_{ot} ao valor de θ que permite atingir a solução em um menor número de iterações, foi observado em geral, $[\theta_1; \theta_2] \cong [0,6-0,8 ; 1,0)$, sendo que θ_{ot} sempre se localiza próximo a θ_1 .

5) O refinamento da malha (h) não altera significativamente $[\theta_1; \theta_2]$.

6) A redução na altura da região discretizada em relação ao eixo $y = 0$, reduz θ_1 . Próximo ao nível do solo, θ_1 (e θ_{ot}) tende a zero.

7) O número de regiões do tipo de Ω_A (figura 4.1.1) não altera significativamente $[\theta_1; \theta_2]$, desde que sua distância média a $\partial\Omega_1$ seja mantida aproximadamente constante.

8) A quantidade de cargas externas à região discretizada não interfere significativamente sobre $[\theta_1; \theta_2]$.

9) A proximidade de uma carga externa a $\partial\Omega_1$ tende a reduzir θ_1 , podendo chegar a $\theta_1 \cong 0,4$.

10) Não foi observada diferença significativa nos resultados ou no tempo de CPU para os cálculos, ao utilizarmos o método de Gram-Schmidt clássico ou o método de Gram-Schmidt modificado para a decomposição QR .

SUGESTÕES

Deixamos como sugestões que entendemos serem úteis não só para o complemento deste trabalho como também para pesquisas posteriores os seguintes pontos :

1) Obter uma estimativa analítica de erro, particularmente para o método de simulação de cargas,

2) Aplicação de técnicas de otimização para obter a melhor localização para as cargas elétricas arbitradas (vide Chow [3]),

3) Pesquisar uma discretização para a derivada normal sobre contornos de descontinuidade da permissividade elétrica relativa com erro $O(h^2)$,

4) Estender a aplicação do método híbrido a problemas tridimensionais com simetria axial (trata-se na realidade de um problema bidimensional) e a problemas tridimensionais sem qualquer simetria,

5) Pesquisar a formulação de problemas com a região discretizada por meio de "finite boxes" (Franz [7]), reduzindo assim a quantidade de nós requeridos.

CONCLUSÃO

Novas propostas para o método híbrido de simulação de cargas com diferenças finitas ou elementos finitos foram apresentadas.

Foi mostrado que o método de simulação de cargas com quadrados mínimos e decomposição QR pode conduzir a uma significativa economia no espaço de memória de computador utilizado originalmente (acima de 50 %), sem perda significativa da precisão dos resultados.

A resolução do problema na região discretizada por meio de um método adequado (a exemplo do CGS) pode conduzir a uma grande economia no tempo de CPU (às vezes acima de 80% , se comparado com o método dos gradientes conjugados).

Foram formuladas discretizações para as condições de contorno que conduzem a resultados com excelente nível de precisão.

A resolução do sistema acoplado de equações utilizando a teoria do ponto fixo independe da aproximação inicial arbitrada para os potenciais elétricos e amplia a gama de problemas que podem ser resolvidos pelo método híbrido, em relação ao estágio anterior de desenvolvimento do método.

Finalizando, resultados de aplicações práticas deste estudo evidenciam que as inovações apresentadas ao tratamento matemático do método híbrido representam em conjunto um novo passo à utilização deste método para o cálculo de campos elétricos em domínios ilimitados.

APÊNDICES

| | Página |
|------|--|
| I | Equação de Poisson 73 |
| II | Equações de continuidade sobre superfícies ... 76 |
| III | Determinação dos coeficientes de Maxwell 80 |
| IV | Método das imagens 83 |
| V | Discretização para o operador Laplaciano 86 |
| VI | Discretização para a derivada normal 88 |
| VII | Método dos quadrados mínimos 93 |
| VIII | Método de Gram-Schmidt clássico 95 |
| IX | Método de Gram-Schmidt modificado 97 |
| X | Método dos gradientes conjugados 99 |
| XI | Método dos subespaços de Krylov 101 |
| XII | Método dos gradientes conjugados quadráticos.. 104 |
| XIII | Pré-condicionamento 108 |

APÊNDICE I

EQUAÇÃO DE POISSON

A equação de Maxwell

$$\oint_s (\epsilon \epsilon_0 \mathbf{E}) \cdot d\mathbf{s} = \int_v \rho_v dv$$

pode ser escrita em sua forma diferencial (Johnk [10], cap. 2),
dividindo-a por Δv e tomando o limite para Δv tendendo a zero

$$\lim_{\Delta v \rightarrow 0} \frac{\oint_s (\epsilon \epsilon_0 \mathbf{E}) \cdot d\mathbf{s}}{\Delta v} = \lim_{\Delta v \rightarrow 0} \frac{\int_v \rho_v dv}{\Delta v}$$

que é equivalente a

$$\nabla \cdot (\epsilon \epsilon_0 \mathbf{E}) = \rho_v \quad \text{I.1}$$

onde ∇ é o operador $\frac{\partial}{\partial x} \mathbf{e}_1 + \frac{\partial}{\partial y} \mathbf{e}_2 + \frac{\partial}{\partial z} \mathbf{e}_3$

\mathbf{e}_1 , \mathbf{e}_2 e \mathbf{e}_3 são os vetores unitários das direções x ,
 y e z , respectivamente.

A forma diferencial de outra equação de Maxwell

$$\oint_1 \mathbf{E} \cdot d\mathbf{l} = 0$$

é obtida dividindo-a por Δs_1 , Δs_2 e Δs_3 , que representam uma superfície Δs orientada segundo direções ortogonais \mathbf{e}_1 , \mathbf{e}_2 e \mathbf{e}_3 , respectivamente, somando e tomando os limites para Δs_1 , Δs_2 e Δs_3 tendendo a zero, ou seja,

$$\mathbf{e}_1 \lim_{\Delta s_1 \rightarrow 0} \frac{\oint_1 \mathbf{E} \cdot d\mathbf{l}}{\Delta s_1} + \mathbf{e}_2 \lim_{\Delta s_2 \rightarrow 0} \frac{\oint_1 \mathbf{E} \cdot d\mathbf{l}}{\Delta s_2} + \mathbf{e}_3 \lim_{\Delta s_3 \rightarrow 0} \frac{\oint_1 \mathbf{E} \cdot d\mathbf{l}}{\Delta s_3} = 0$$

que é equivalente a

$$\nabla \times \mathbf{E} = 0 \quad \text{I.2}$$

Da álgebra vetorial temos a expressão (Johnk [10], cap. 4)

$$\nabla \times (\nabla \phi) = 0, \quad \text{I.3}$$

válida para qualquer função escalar e diferenciável ϕ . Portanto, I.2 e I.3 mostram que \mathbf{E} pode ser obtido de uma função escalar auxiliar ϕ , por meio de

$$\mathbf{E} = -\nabla \phi \quad \text{I.4}$$

A natureza de ϕ é esclarecida como segue. Suponha P_0 um ponto fixo no espaço com $\phi(P_0) = \phi_0$ e P um ponto qualquer. Assim

$$\int_{P_0}^P \mathbf{E} \cdot d\mathbf{l} = - \int_{P_0}^P (\nabla \Phi) \cdot d\mathbf{l} = - \int_{P_0}^P \left(\frac{\partial \Phi}{\partial x} dx + \frac{\partial \Phi}{\partial y} dy + \frac{\partial \Phi}{\partial z} dz \right)$$

ou seja

$$\int_{P_0}^P \mathbf{E} \cdot d\mathbf{l} = - \int_{P_0}^P d\Phi$$

de onde

$$\Phi(P) - \Phi(P_0) = - \int_{P_0}^P \mathbf{E} \cdot d\mathbf{l} \quad \text{I.5}$$

A função Φ , acima definida, é chamada de potencial elétrico do campo \mathbf{E} .

Combinando as equações I.1 e I.4 obtemos

$$\nabla \cdot (\epsilon \epsilon_0 \nabla \Phi) = - \rho_v \quad \text{I.6}$$

Esta equação é conhecida como a equação de Poisson, que rege a distribuição da função potencial em problemas de eletrostática.

APÊNDICE II

CONDIÇÕES DE CONTINUIDADE EM SUPERFÍCIES

As equações de Maxwell utilizadas no Apêndice I são também aplicáveis para obtermos condições de continuidade sobre qualquer superfície que separa o espaço tridimensional em duas regiões Ω_1 e Ω_2 (Johnk [10], cap. 3). Podemos generalizar a análise, considerando que a superfície em questão separa duas regiões com permissividade elétrica relativa diferentes (ϵ_1 e ϵ_2).

Inicialmente utilizamos a equação de Maxwell

$$\oint_s \mathbf{D} \cdot d\mathbf{s} = \int_v \rho_v dv$$

sobre um volume V convenientemente escolhido como indica a figura II.1, onde Δs representa uma superfície suficientemente pequena. Assim,

$$\lim_{\delta h \rightarrow 0} \oint_s \mathbf{D} \cdot d\mathbf{s} = \mathbf{D}_1 \cdot (\mathbf{n} \Delta s) + \mathbf{D}_2 \cdot (-\mathbf{n} \Delta s) = (\mathbf{D}_1 - \mathbf{D}_2) \cdot \mathbf{n} \Delta s$$

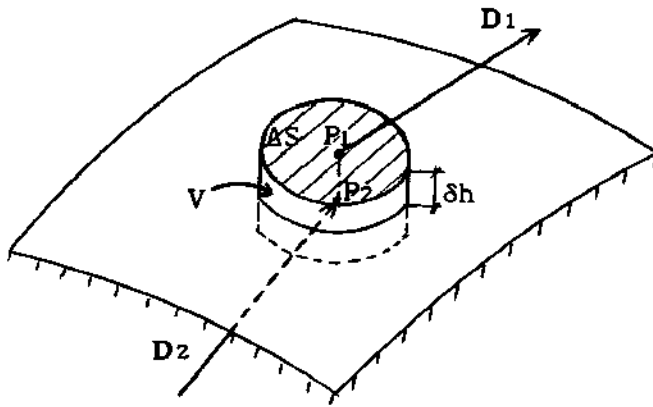


FIGURA II.1a - Continuidade sobre superfícies

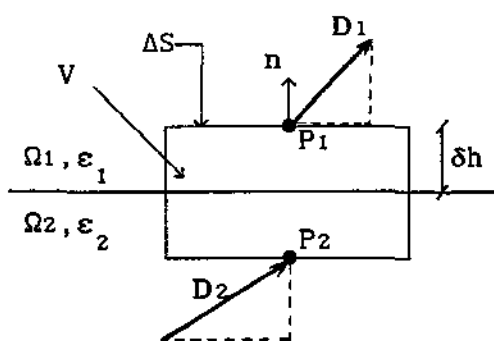


FIGURA II.1b - Continuidade sobre superfícies

Por outro lado, considerando que não há cargas em V , temos

$$\lim_{\delta h \rightarrow 0} \int_V \rho_v dv = 0$$

Assim obtemos a primeira condição de continuidade sobre a interface $\Omega_1 - \Omega_2$:

$$\mathbf{n} \cdot \mathbf{D}_1 = \mathbf{n} \cdot \mathbf{D}_2 \quad \text{II.1}$$

Utilizando agora a equação de Maxwell

$$\oint_1 \mathbf{E} \cdot d\mathbf{l} = 0$$

sobre uma linha fechada convenientemente escolhida como indica a figura II.2, com Δl suficientemente pequeno, obtemos

$$\lim_{\delta h \rightarrow 0} \oint_1 \mathbf{E} \cdot d\mathbf{l} = E_1 \cdot (t \Delta l) + E_2 \cdot (-t \Delta l) = 0$$

e,
$$t \cdot E_1 = t \cdot E_2 \quad \text{II.2}$$

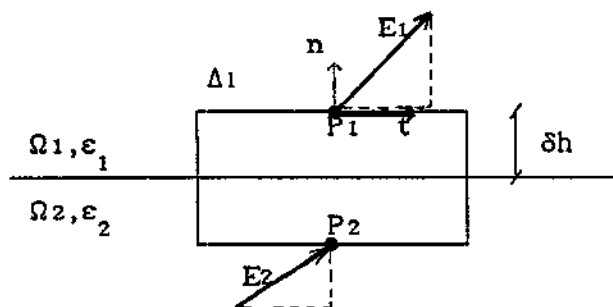


FIGURA II.2 - Continuidade sobre superfícies

No entanto, é possível mostrar que esta condição é satisfeita se o potencial for contínuo na interface $\Omega_1-\Omega_2$. De fato, seja a curva γ da figura II.3.

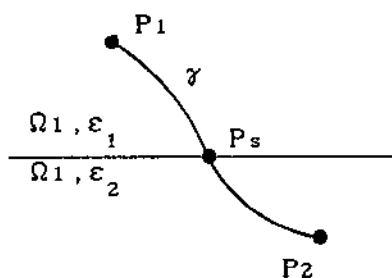


FIGURA II.3 - Continuidade do potencial elétrico

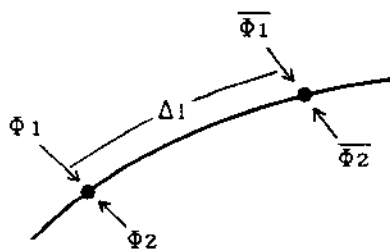


FIGURA II.4 - Continuidade do vetor E

Da equação I.8,

$$\lim_{\substack{P_2 \rightarrow P_s \\ P_1 \rightarrow P_s}} \phi(P_2) - \phi(P_1) = \lim_{\substack{P_2 \rightarrow P_s \\ P_1 \rightarrow P_s}} - \int_{P_1}^{P_2} \mathbf{E} \cdot d\mathbf{l} = 0$$

Logo, $\Phi(P_1) = \Phi(P_2)$ e o potencial elétrico é contínuo na interface. Agora, mostraremos que a condição $\Phi(P_1) = \Phi(P_2)$ implica na condição II.2.

Fazendo $\Delta l \rightarrow 0$ na figura II.4, e como $\Phi_1 = \Phi_2$ e $\overline{\Phi}_1 = \overline{\Phi}_2$, obtemos

$$\lim_{\Delta l \rightarrow 0} \frac{\overline{\Phi}_1 - \Phi_1}{\Delta l} = \lim_{\Delta l \rightarrow 0} \frac{\overline{\Phi}_2 - \Phi_2}{\Delta l}$$

ou
$$\mathbf{t} \cdot \mathbf{E}_1 = \mathbf{t} \cdot \mathbf{E}_2 . \quad \text{II.2}$$

Portanto, a condição II.2 é satisfeita com a continuidade do potencial elétrico:

$$\Phi(P_1) = \Phi(P_2) \quad \text{II.3}$$

APÊNDICE III

DETERMINAÇÃO DOS COEFICIENTES DE MAXWELL

III.1 COEFICIENTES DE POTENCIAL ELÉTRICO

Para a determinação do coeficiente de potencial elétrico p_{ij} de uma carga linearmente distribuída sobre um comprimento infinito e paralela à superfície da terra, consideraremos inicialmente o problema tridimensional da figura III.1 onde faremos uso das equações de Maxwell e do método das Imagens (Apêndice IV).

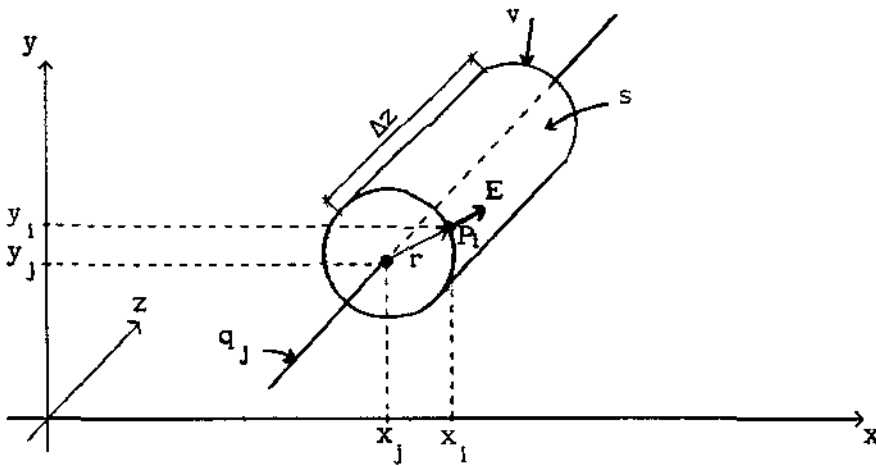


FIGURA III.1 - Carga linearmente distribuída sobre um comprimento infinito

Considerando a simetria cilíndrica do problema podemos escrever

$$(\epsilon\epsilon_0 E) \oint_s ds = \int_v \rho_v dv$$

ou,

$$(\epsilon\epsilon_0 E) 2\pi r \Delta z = q_j \Delta z ,$$

onde q_j é uma carga elétrica linearmente distribuída.

$$\text{Assim, } E = \frac{q_j}{2\pi\epsilon\epsilon_0 r} , \text{ ou, } E = \frac{q_j}{2\pi\epsilon\epsilon_0 r} \mathbf{a}_r = -\nabla\phi$$

onde \mathbf{a}_r é o vetor unitário na direção de r .

Lembrando que em coordenadas cilíndricas o gradiente da função ϕ é escrito como

$$\nabla\phi = \frac{\partial\phi}{\partial r} \mathbf{a}_r + \frac{1}{r} \frac{\partial\phi}{\partial\varphi} \mathbf{a}_\varphi + \frac{\partial\phi}{\partial z} \mathbf{a}_z$$

que, pela simetria do problema, fica reduzida a

$$\nabla\phi = \frac{\partial\phi}{\partial r} \mathbf{a}_r ,$$

logo,

$$\frac{\partial\phi}{\partial r} = \frac{-q_j}{2\pi\epsilon\epsilon_0 r} ,$$

de onde,

$$\phi = \frac{-q_j}{2\pi\epsilon\epsilon_0} \ln r + C ,$$

que em coordenadas cartesianas é escrito na forma

$$\phi(P_i) = \frac{-q_j}{4\pi\epsilon\epsilon_0} \ln \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right] + C \quad \text{III.1.1}$$

Utilizando o método das imagens é mostrado no Apêndice IV que o potencial elétrico em um ponto $P_1(x_1, y_1)$, localizado acima da superfície da terra (a qual é considerada com potencial elétrico nulo), é dado por

$$\Phi(P_1) = p_{1j} \cdot q_j \quad \text{III.1.2}$$

sendo

$$p_{1j} = \frac{1}{4\pi\epsilon\epsilon_0} \ln \left[\frac{(x_1 - x_j)^2 + (y_1 + y_j)^2}{(x_1 - x_j)^2 + (y_1 - y_j)^2} \right] \quad \text{III.1.3}$$

onde o eixo x deve ser coincidente com a superfície do solo.

III.2 COEFICIENTES DE CAMPO ELÉTRICO

Considerando as expressões I.4 e III.1.2, podemos escrever o vetor E para o ponto $P_1(x_1, y_1)$ como

$$E(P_1) = - \left[\frac{\partial(p_{1j} q_j)}{\partial x_1} e_1 + \frac{\partial(p_{1j} q_j)}{\partial y_1} e_2 \right] = (f_x^{1j} e_1 + f_y^{1j} e_2) q_j$$

onde,

$$f_x^{1j} = \frac{2}{4\pi\epsilon\epsilon_0} \left[\frac{x_1 - x_j}{(x_1 - x_j)^2 + (y_1 - y_j)^2} - \frac{x_1 - x_j}{(x_1 - x_j)^2 + (y_1 + y_j)^2} \right] \quad \text{III.2.1}$$

e,

$$f_y^{1j} = \frac{2}{4\pi\epsilon\epsilon_0} \left[\frac{y_1 - y_j}{(x_1 - x_j)^2 + (y_1 - y_j)^2} - \frac{y_1 + y_j}{(x_1 - x_j)^2 + (y_1 + y_j)^2} \right] \quad \text{III.2.2}$$

APÊNDICE IV

MÉTODO DAS IMAGENS

Alguns problemas de eletrostática com condições de contorno tipo Dirichlet podem ser resolvidos com o chamado "método das imagens", que consiste em substituir um certo tipo de carga por outra mais simples, que forneça precisamente a mesma solução que o problema original na região de interesse.

Vamos supor o problema bidimensional da figura IV.1, sendo Ω o semi-plano $y > 0$, e $\partial\Omega$ coincidente com eixo x .

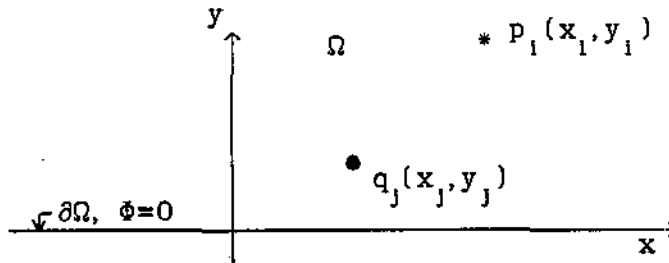


FIGURA IV.1 - Problema de Dirichlet sobre o semi-plano

Resolveremos o seguinte problema de contorno:

$$\Delta \Phi = -\rho \quad \text{em } \Omega \quad \text{IV.1}$$

$$\Phi = 0 \quad \text{em } \partial\Omega$$

Para atender à condição $\phi = 0$ em $\partial\Omega$, podemos arbitrar uma carga fictícia, de valor q_k , situada na região externa à região Ω , o que não altera a distribuição de cargas do problema IV.1. É fácil notar que uma carga de valor $q_k = -q_j$, localizada em $(x_k, y_k) = (x_j, -y_j)$ atende a esta condição, pois utilizando o princípio da superposição juntamente com a equação III.1.1 com $C = 0$, teremos em $P_i \in \partial\Omega$,

$$\begin{aligned} \phi(P_i) &= \frac{-q_j}{4\pi\epsilon\epsilon_0} \ln\left[(x_i - x_j)^2 + (-y_j)^2\right] + C \\ &+ \frac{-(-q_j)}{4\pi\epsilon\epsilon_0} \ln\left[(x_i - x_j)^2 + (y_j)^2\right] + C = 0 \end{aligned}$$

Assim sendo, a unicidade de soluções para o problema de Dirichlet e a validade do princípio da superposição nos asseguram que o potencial num ponto qualquer $P_i \in \Omega$ pode ser expresso por

$$\begin{aligned} \phi(P_i) &= \frac{-q_j}{4\pi\epsilon\epsilon_0} \ln\left[(x_i - x_j)^2 + (y_i - y_j)^2\right] + \\ &+ \frac{-(-q_j)}{4\pi\epsilon\epsilon_0} \ln\left[(x_i - x_j)^2 + (y_i + y_j)^2\right] = \\ &\frac{q_j}{4\pi\epsilon\epsilon_0} \ln\left[\frac{(x_i - x_j)^2 + (y_i + y_j)^2}{(x_i - x_j)^2 + (y_i - y_j)^2}\right] = P_{ij} \cdot q_j \end{aligned}$$

Desta forma, foi mostrado que o problema IV.1 é precisamente resolvido sobre $\Omega \cup \partial\Omega$ por meio do acréscimo de uma carga linear fictícia (imagem), localizada conforme a figura IV.2.

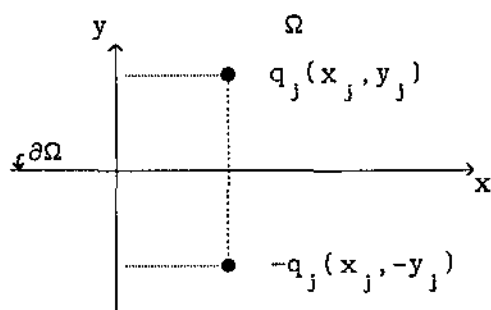


FIGURA IV.2 - Localização da carga imagem

APÊNDICE V

DISCRETIZAÇÃO DO OPERADOR LAPLACIANO

Seja o nó genérico P_0 (figura V.1), cujos nós vizinhos P_1 , P_2 , P_3 e P_4 estão a uma distância de ph , qh , rh e sh respectivamente, sendo $p, q, r, s \leq 1,0$.

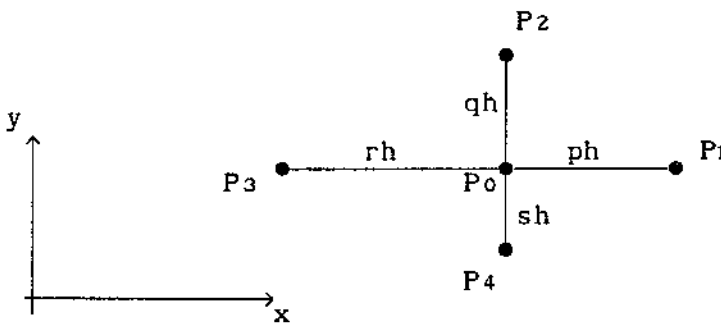


FIGURA V.1 - Discretização do operador Laplaciano

Utilizando a expansão de Taylor na direção x obtemos

$$\Phi(P_1) = \Phi(P_0) + ph \frac{\partial \Phi}{\partial x} (P_0) + \frac{(ph)^2}{2!} (\partial^2 \Phi / \partial x^2) (P_0) + O(h^3) \quad V.1$$

$$\Phi(P_3) = \Phi(P_0) - rh \frac{\partial \Phi}{\partial x} (P_0) + \frac{(rh)^2}{2!} (\partial^2 \Phi / \partial x^2) (P_0) + O(h^3) \quad V.2$$

Fazendo $r \cdot (V.1) + p \cdot (V.2)$ obtemos

$$\begin{aligned} r \Phi(P_1) + p \Phi(P_3) &= (p+r) \Phi(P_0) + \\ &pr(p+r)(h^2/2)(\partial^2 \Phi / \partial x^2) (P_0) + O(h^3) \end{aligned}$$

De onde temos em P_0 ,

$$\frac{h^2}{2} (\partial^2 \Phi / \partial x^2) = \frac{1}{p(p+r)} \Phi(P_1) + \frac{1}{r(p+r)} \Phi(P_3) - \frac{1}{p \cdot r} \Phi(P_0) + O(h^3)$$

Procedendo de modo similar na direção y teremos

$$\frac{h^2}{2} (\partial^2 \Phi / \partial y^2) = \frac{1}{q(q+s)} \Phi(P_2) + \frac{1}{s(q+s)} \Phi(P_4) - \frac{1}{q \cdot s} \Phi(P_0) + O(h^3)$$

Assim,

$$\frac{h^2}{2} \left[\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} \right] = \frac{h^2}{2} \Delta \Phi = \frac{1}{p(p+r)} \Phi(P_1) + \frac{1}{q(q+s)} \Phi(P_2) +$$

$$\frac{1}{r(p+r)} \Phi(P_3) + \frac{1}{s(q+s)} \Phi(P_4) - \left(\frac{1}{p \cdot r} + \frac{1}{q \cdot s} \right) \Phi(P_0) + O(h^3)$$

Como aproximação para a equação $\Delta \Phi = 0$ em P_0 podemos escrever

$$\begin{aligned} & \frac{1}{p(p+r)} \Phi(P_1) + \frac{1}{q(q+s)} \Phi(P_2) + \frac{1}{r(p+r)} \Phi(P_3) \\ & + \frac{1}{s(q+s)} \Phi(P_4) - \left(\frac{1}{p \cdot r} + \frac{1}{q \cdot s} \right) \Phi(P_0) = 0 \quad \text{V.3} \end{aligned}$$

Pode ser mostrado (Forsythe [6], cap. 3) que a discretização V.3 terá erro $O(h^2)$ se $p, q, r, s \leq 1$ apenas quando P_0 é adjacente ao contorno (no seu interior, p, q, r, s devem ser iguais a 1,0). Em qualquer outro caso o erro será $O(h)$.

APÊNDICE VI

DISCRETIZAÇÃO PARA A DERIVADA NORMAL

VI.1 MALHA COM NÓS ALINHADOS SEGUNDO A DIREÇÃO NORMAL

Procuraremos uma discretização para a derivada normal de uma função Φ sobre um ponto $P_i \in \partial\Omega$, admitindo a existência de nós internos a Ω alinhados segundo a direção normal (figura VI.1).

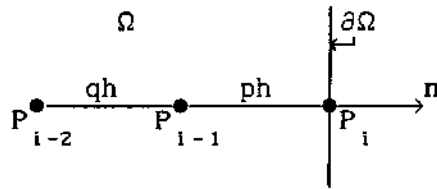


FIGURA VI.1 - Discretização para $\partial\Phi/\partial n$

Utilizando a expansão em séries de Taylor podemos escrever

$$\Phi(P_{i-1}) = \Phi(P_i) - ph \frac{\partial\Phi}{\partial n}(P_i) + \frac{(ph)^2}{2!} (\partial^2\Phi/\partial n^2)(P_i) + O(h^3) \quad \text{VI.1.1}$$

$$\begin{aligned} \Phi(P_{i-2}) = \Phi(P_i) - (ph+qh) \frac{\partial\Phi}{\partial n}(P_i) + \\ \frac{(ph+qh)^2}{2!} (\partial^2\Phi/\partial n^2)(P_i) + O(h^3) \end{aligned} \quad \text{VI.1.2}$$

Fazendo $(p+q)^2 \cdot (\text{VI.1.1}) - p^2 \cdot (\text{VI.1.2})$ teremos

$$\mathbf{n} \cdot \nabla \Phi(P_i) = \frac{\partial \Phi}{\partial \mathbf{n}}(P_i) = \alpha \Phi(P_i) + \beta \Phi(P_{i-1}) + \gamma \Phi(P_{i-2}) + O(h^2)$$

onde $\alpha = 1/ph + 1/(ph+qh)$

$$\beta = -(1/ph + 1/qh)$$

$$\gamma = 1/qh - 1/(ph+qh)$$

Assim, a aproximação

$$\frac{\partial \Phi}{\partial \mathbf{n}}(P_i) = \alpha \Phi(P_i) + \beta \Phi(P_{i-1}) + \gamma \Phi(P_{i-2}) \quad \text{VI.1.3}$$

terá erro $O(h^2)$.

VI.2 MALHA COM NÓS NÃO ALINHADOS SEGUNDO A DIREÇÃO NORMAL

VI.2.1 DISCRETIZAÇÃO COM 3 NÓS INTERNOS À REGIÃO

Seja $P_i \in \partial\Omega$ a origem de um sistema de coordenadas tal que o eixo x é tangente a $\partial\Omega$ em P_i , e o eixo y é positivo segundo a direção normal dirigida para dentro de Ω (figura VI.2).

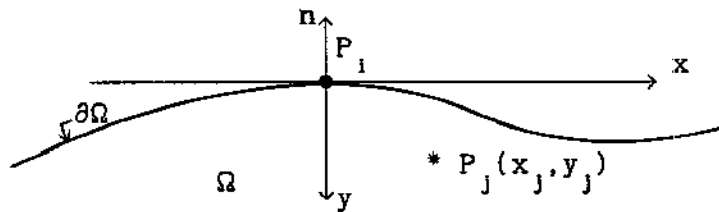


FIGURA VI.2 - Discretização para $\partial\Phi/\partial\mathbf{n}$

Pode ser mostrado (Synge [21]) que, para qualquer função derivável Φ , é válido em P_i ,

$$\frac{\partial^2 \Phi}{\partial x \partial y} = - \frac{\partial^2 \Phi}{\partial n \partial s} + K \frac{\partial \Phi}{\partial s} \quad \text{VI.2.1.1}$$

onde

\mathbf{n} = normal externa a $\partial\Omega$

s = comprimento de arco

K = curvatura de $\partial\Omega$ em P_1 .

Considerando $\Delta\Phi = 0$ em Ω e utilizando VI.2.1.1, é mostrado por Bramble [2] que

$$\begin{aligned} \Phi(P_j) = & \Phi(P_1) + x_j \left[1 + y_j K \right] \frac{\partial \Phi}{\partial s} (P_1) - \\ & y_j \frac{\partial \Phi}{\partial n} (P_1) + \frac{1}{2} (x_j^2 - y_j^2) (\partial^2 \Phi / \partial x^2)(P_1) - \\ & x_j y_j \frac{\partial^2 \Phi}{\partial n \partial s}(P_1) + O(x_j^3 + y_j^3) \end{aligned} \quad \text{VI.2.1.2}$$

onde P_j é um ponto interno a Ω , com coordenadas (x_j, y_j) .

Tomando três pontos internos a Ω ($j = 1, 2, 3$), e somando as respectivas expressões do tipo de VI.2.1.2 multiplicadas por um coeficiente de ponderação a_j obteremos

$$\begin{aligned} \sum_{j=1}^3 a_j [\Phi(P_1) - \Phi(P_j)] &= \sum_{j=1}^3 a_j y_j \frac{\partial \Phi}{\partial n} (P_1) - \\ \sum_{j=1}^3 a_j x_j [1 + y_j K] \frac{\partial \Phi}{\partial s} (P_1) &- \sum_{j=1}^3 \frac{1}{2} a_j (x_j^2 - y_j^2) (\partial^2 \Phi / \partial x^2)(P_1) + \\ \sum_{j=1}^3 a_j x_j y_j \frac{\partial^2 \Phi}{\partial n \partial s}(P_1) &+ O \left[\sum_{j=1}^3 a_j (x_j^3 + y_j^3) \right] \end{aligned} \quad \text{VI.2.1.3}$$

Para obtermos uma aproximação para $\frac{\partial \Phi}{\partial \mathbf{n}}(P_i)$, será necessário que

$$\sum_{j=1}^3 a_j y_j = 1$$

$$\sum_{j=1}^3 a_j x_j [1 + y_j K] = 0$$

$$\sum_{j=1}^3 \frac{1}{2} a_j (x_j^2 - y_j^2) = 0$$

Estas expressões representam um sistema de equações lineares com incógnitas a_j , que pode ser escrito na forma matricial,

$$\begin{bmatrix} y_1 & y_2 & y_3 \\ x_1(1+y_1 \cdot K) & x_2(1+y_2 \cdot K) & x_3(1+y_3 \cdot K) \\ x_1^2 - y_1^2 & x_2^2 - y_2^2 & x_3^2 - y_3^2 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{VI.2.1.4}$$

Uma vez resolvido o sistema linear VI.2.1.4, obtemos de VI.2.1.3 a seguinte expressão para $\frac{\partial \Phi}{\partial \mathbf{n}}(P_i)$:

$$\frac{\partial \Phi}{\partial \mathbf{n}}(P_i) = \sum_{j=1}^3 a_j [\Phi(P_i) - \Phi(P_j)] -$$

$$\sum_{j=1}^3 a_j x_j y_j (\partial^2 \Phi / \partial \mathbf{n} \partial s)(P_i) + O \left[\sum_{j=1}^3 a_j (x_j^3 + y_j^3) \right] \quad \text{VI.2.1.5}$$

Conforme mostrado por Bramble [2], os valores a_j estão limitados por um termo que é $O(h^{-1})$. Assim, se a função ϕ ou $\partial\phi/\partial n$ são conhecidos sobre $\partial\Omega$, podemos obter uma aproximação com erro $O(h^2)$ por meio de

$$\frac{\partial\phi}{\partial n}(P_1) \cong \sum_{j=1}^3 a_j [\phi(P_1) - \phi(P_j)] - \sum_{j=1}^3 a_j x_j y_j (\partial^2\phi/\partial n \partial s)(P_1) \quad \text{VI.2.1.6}$$

Se a função ϕ ou $\partial\phi/\partial n$ não são dadas em $\partial\Omega$, obtemos a seguinte aproximação com erro $O(h)$:

$$\frac{\partial\phi}{\partial n}(P_1) \cong \sum_{j=1}^3 a_j [\phi(P_1) - \phi(P_j)] \quad \text{VI.2.1.7}$$

VI.2.2 DISCRETIZAÇÃO COM 2 NÓS INTERNOS À REGIÃO

Como alternativa para o Apêndice VI.2.1, Bramble [2] fornece uma discretização com erro $O(h)$ utilizando apenas dois nós internos a Ω , que podemos escrever como

$$\frac{\partial\phi}{\partial n}(P_1) \cong \sum_{j=1}^2 a_j [\phi(P_1) - \phi(P_j)] \quad \text{VI.2.2.1}$$

onde os coeficientes a_j são calculados resolvendo o sistema de equações

$$\begin{bmatrix} y_1 & y_2 \\ x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{VI.2.2.2}$$

APÊNDICE VII

MÉTODO DOS QUADRADOS MÍNIMOS

Se $A \in \mathbb{R}^{M \times N}$ pode ser decomposta em $A = Q_1 R_1$ (Golub [8], cap. 6), com $Q_1 \in \mathbb{R}^{M \times M}$ matriz ortogonal e $R_1 \in \mathbb{R}^{M \times N}$ matriz triangular superior, e $R_1 = \begin{bmatrix} R \\ 0 \end{bmatrix}$, $R \in \mathbb{R}^{N \times N}$, então a norma euclidiana do vetor erro do sistema linear $Ax = b$ pode ser avaliada a partir de

$$\begin{aligned} \|Ax - b\|_2^2 &= \|Q_1 R_1 x - b\|_2^2 = \|R_1 x - Q_1^T b\|_2^2 = \\ &= \|Rx - c\|_2^2 + \|d\|_2^2, \end{aligned}$$

onde $Q_1^T b = \begin{bmatrix} c \\ d \end{bmatrix}$, $c \in \mathbb{R}^N$ e $d \in \mathbb{R}^{M-N}$.

Assim, a solução de quadrados mínimos é a solução de $Rx=c$, com resíduo $\|d\|_2^2$. O cálculo do resíduo pode ser dispensado se computarmos apenas as N colunas à esquerda de Q_1 (submatriz Q).

Para o cálculo da decomposição $A = Q_1 R_1 = QR$, podemos utilizar o método de Gram-Schmidt clássico (Apêndice VIII) ou o método de Gram-Schmidt modificado (Apêndice IX).

O seguinte algoritmo resolve o sistema linear $Rx = c$, sendo $R \in \mathbb{R}^{N \times N}$ triangular superior, armazenada como um vetor ordenado por linhas (conforme algoritmos dos Apêndices VIII e IX):

```
For i = N to 1
  Soma = 0.
  For j = i+1 to N
   iaux = (i-1)*N + j - (i-1)*i/2
    soma = soma + R(iaux)*x(j)
  end
 iaux = (i-1)*N + i - (i-1)*i/2
  x(i) = (c(i) - soma)/R(iaux)
end
```

APÊNDICE VIII

DECOMPOSIÇÃO QR - MÉTODO DE GRAM-SCHMIDT CLÁSSICO

Neste método, Q e R são computadas seqüencialmente por colunas (Golub [8], cap. 6).

Seja $A = [a_1 \ a_2 \ \dots \ a_N]$ e $Q = [q_1 \ q_2 \ \dots \ q_N]$, com $a_i, q_i \in \mathbb{R}^M$, e tal que $A = QR$. Então

$$a_k = \sum_{i=1}^k r_{ik} q_i = \sum_{i=1}^{k-1} r_{ik} q_i + r_{kk} q_k, \quad k=1 \dots N$$

$$\text{De onde: } r_{kk} q_k = z_k = a_k - \sum_{i=1}^{k-1} r_{ik} q_i, \quad k=1 \dots N,$$

de onde obtemos a coluna q_k .

Os r_{ik} devem ser tais que $z_k \perp q_i$, $i=1 \dots k-1$. Assim, podemos mostrar que $r_{ik} = q_i^T a_k$, $i=1 \dots k-1$, o que gera a coluna k da matriz R.

Deste modo, o algoritmo para o cálculo de Q e R fica

```
For k = 1 to N
```

```
  For i = 1 to k-1
```

$$r_{ik} = a_1^T a_k$$

$$\text{iaux} = (i-1)*N + k - (i-1)*1/2$$

```

R(iaux) = r_ik
a_k = a_k - r_ik a_i

end

r_kk = (a_k^T a_k)^{1/2}
a_k = a_k / r_kk
iaux = (k-1)*N + k - (k-1)*k/2

R(iaux) = r_kk

end

```

Neste algoritmo R é armazenada na forma de um vetor ordenado por linhas, de dimensão $(N+1)N/2$.

APÊNDICE IX

DECOMPOSIÇÃO QR - MÉTODO DE GRAM-SCHMIDT MODIFICADO

Neste método, Q é computada seqüencialmente por colunas e R por linhas (Golub [8], cap. 6).

Sejam A e Q definidos como no Apêndice VIII, e seja $r_i^T \in \mathbb{R}^N$ a i-ésima linha de R, tal que $A = QR$. Então

$$A = \sum_{i=1}^N q_i r_i^T = \sum_{i=1}^{k-1} q_i r_i^T + \sum_{i=k}^N q_i r_i^T, \text{ e}$$

$$A - \sum_{i=1}^{k-1} q_i r_i^T = \sum_{i=k}^N q_i r_i^T = \begin{bmatrix} 0 & A^{(k)} \end{bmatrix},$$

com $0 \in \mathbb{R}^{M \times k-1}$ e $A^{(k)} \in \mathbb{R}^{M \times N-(k-1)}$

Tomamos $z \in \mathbb{R}^M$ como a primeira coluna de $A^{(k)}$. Assim, $A^{(k)} = [z, B]$, $B = [b_{k+1} \ b_{k+2} \ \dots \ b_N] \in \mathbb{R}^{M \times (N-k)}$, de onde obtemos $z = q_k r_{kk}$. Assim,

$$r_{kk} = \|z\|_2, \text{ e } q_k = z / r_{kk}$$

Da condição de ortogonalidade dos q_i pode ser mostrado que $r_{kj} = q_k^T b_j$, $j=k+1 \dots N$.

$A^{(k+1)}$ é determinado por

$$A^{(k+1)} = B - q_k (r_{k,k+1}, r_{k,k+2}, \dots, r_{k,N})$$

Deste modo, o algoritmo para o cálculo de Q e R fica

```
For k = 1 to N
  rkk =  $\left( a_k^T a_k \right)^{1/2}$ 
  ak = ak / rkk
  iaux = (k-1)*N + k - (k-1)*k/2
  R(iaux) = rkk
  For j = k+1 to N
    rkj = akT aj
    iaux = (k-1)*N + j - (k-1)*k/2
    R(iaux) = rkj
    aj = aj - ak rkj
  end
end
```

Neste algoritmo R é armazenada na forma de um vetor ordenado por linhas, como no Apêndice VIII.

APÊNDICE X

MÉTODO DOS GRADIENTES CONJUGADOS

Seja o sistema linear $Ax = b$, onde $A \in \mathbb{R}^{N \times N}$ é simétrica e definida positiva. O método dos gradientes conjugados (Fernandes [5], Golub [8]) consiste na minimização da forma quadrática

$$F(x) = \frac{1}{2}x^T Ax - x^T b,$$

pois, se x^* é um ponto crítico de F , então $\nabla F = Ax^* - b = 0$. Como o Hessiano $H(x^*) > 0$ (A é definida positiva), então $F(x^*)$ é um ponto de mínimo. De $\nabla F = 0$ temos, portanto, que x^* é solução de $Ax = b$.

A idéia é construir uma seqüência tal que

$$x_k = x_{k-1} + t p_{k-1} \quad \text{X.1}$$

com $\langle Ap_{k-1}, p_{k-1} \rangle = 0$, e $p_0 = -\nabla F(x_0)$, onde t é um parâmetro escolhido tal que

$$F(x_k) = \min_t \{F(x_{k-1} + t p_{k-1})\}$$

Pode ser mostrado que:

$$\text{a) } t = \langle r_{k-1}, r_{k-1} \rangle / \langle Ap_{k-1}, p_{k-1} \rangle; \quad r_{k-1} = -\nabla F(x_{k-1}) \quad \text{X.2}$$

$$b) p_k = r_k + \beta p_{k-1}; \quad \beta = \langle r_k, r_k \rangle / \langle r_{k-1}, r_{k-1} \rangle \quad X.3$$

c) $\{p_0, p_1, \dots, p_{k-1}\}$ geram todo o sub-espaço \mathbb{R}^k , logo, a seqüência X.1 atinge a solução no máximo em N passos, para qualquer x_0 .

Se A não é simétrica e positiva definida, podemos tomar o sistema normal $A^T A x = A^T b$, pois $A^T A$ o será, desde que A tenha posto completo. Assim, a expressão X.2 pode ser escrita como

$$t = \langle r_{k-1}, r_{k-1} \rangle / \langle A^T A p_{k-1}, p_{k-1} \rangle = \langle r_{k-1}, r_{k-1} \rangle / \langle A p_{k-1}, A p_{k-1} \rangle$$

$$\text{com: } r_{k-1} = A^T b - A^T A x_{k-1} = A^T (b - A x_{k-1})$$

Neste caso, o algoritmo para o método dos gradientes conjugados fica

```

Arbitrar  $x_0$ 
 $r_0 = A^T (b - A x_0)$ 
 $p_0 = r_0$ 
For  $k = 1, 2, \dots$ 
  If  $(\|r_{k-1}\| < \text{tolerância})$  (Fim)
   $t = \langle r_{k-1}, r_{k-1} \rangle / \langle A p_{k-1}, A p_{k-1} \rangle$ 
   $x_k = x_{k-1} + t p_{k-1}$ 
   $r_k = A^T (b - A x_k)$ 
   $\beta = \langle r_k, r_k \rangle / \langle r_{k-1}, r_{k-1} \rangle$ 
   $p_k = r_k + \beta p_{k-1}$ 
end

```

APÊNDICE XI

MÉTODO DOS SUBESPAÇOS DE KRYLOV

Seja o sistema linear $Ax = b$, com $A \in \mathbb{R}^{N \times N}$ e seja \mathbb{K}^M o subespaço gerado pelo conjunto de vetores $\{v_1, v_2, \dots, v_m\}$, linearmente independentes. O método dos subespaços de Krylov (Fernandes [5], Saad [6]) buscará uma aproximação x_m para a solução x^* , a partir de uma aproximação inicial x_0 e de \mathbb{K}^M (variedade linear), sendo \mathbb{K}^M gerado por $v_i = A^{i-1}r_0$ e $r_0 = b - Ax_0$. Quando a matriz A é simétrica e definida positiva, os v_i assim construídos são linearmente independentes, como demonstrado para o método dos gradientes conjugados (Golub [8], cap. 10).

O procedimento para a determinação de x_m é construído considerando que o vetor solução pode ser escrito como

$$x^* = x_0 + z^*, \quad \text{XI.1}$$

o que conduz a

$$Az^* - r_0 = 0$$

Buscaremos uma aproximação $z_m \in \mathbb{K}^M$ para z^* , de modo a produzir o mínimo valor para a norma de $r_m = Az_m - r_0$. Para isto, a aproximação z_m deve satisfazer à condição de

Petrov-Galerkin (Saad [17]) $r_m \perp v_i$. Assim,

$$\langle r_m, v_i \rangle = \langle A z_m - r_0, v_i \rangle = 0$$

Como $z_m \in \mathbb{R}^M$, podemos escrever

$$z_m = \sum_{j=1}^M \alpha_j v_j = V_m y_m, \quad \text{XI.2}$$

com $V_m \in \mathbb{R}^{N \times M}$ e $y_m \in \mathbb{R}^M$. Assim,

$$\langle A \sum_{j=1}^M \alpha_j v_j - r_0, v_i \rangle = 0, \text{ ou,}$$

$$\langle A \sum_{j=1}^M \alpha_j v_j, v_i \rangle = \langle r_0, v_i \rangle, \text{ e,}$$

$$\sum_{j=1}^M \alpha_j \langle A v_j, v_i \rangle = \langle r_0, v_i \rangle, \quad i = 1 \dots M \quad \text{XI.3}$$

A resolução do sistema linear XI.3, de ordem $M \times M$, fornece a aproximação z_m por meio de XI.2, e x_m por meio de XI.1 tomando $x_m = x_0 + z_m$.

Devido à capacidade de memória para o armazenamento da matriz V_m , normalmente temos $M \ll N$. Se ao final dos cálculos $\|r_m\|$ não é desprezível, o processo recomeça com x_m em lugar de x_0 (método dos subespaços de Krylov com reinício).

Visto que, quando A é simétrica e definida positiva, os vetores v_i gerados ao longo do processo são linearmente

independentes, a convergência estará garantida neste caso após N/M passos.

O algoritmo para a implementação deste método fica

Arbitrar x_0 e M

$$r_0 = b - Ax_0$$

For $k = 0, 1, 2, \dots$

If ($\|r_k\| < \text{tolerância}$) (Fim)

$$\text{Calcular } V_m = (r_k, Ar_k, A^2r_k, \dots, A^{M-1}r_k)$$

Calcular $y_m = [\alpha_j]$ resolvendo o sistema linear

$$\sum_{j=1}^M \alpha_j \langle A v_j, v_i \rangle = \langle r_0, v_i \rangle, \quad i=1 \dots M$$

$$z_m = V_m y_m$$

$$x_{k+1} = x_k + z_m$$

$$r_{k+1} = b - A x_{k+1}$$

end

APÊNDICE XII

MÉTODO DOS GRADIENTES CONJUGADOS QUADRÁTICOS

Este método para a resolução de sistemas lineares não simétricos foi introduzido por Sonneweld [19], sendo também abordado por Fernandes [5].

Seja o sistema linear $A x = b$, com $A \in \mathbb{R}^{N \times N}$. Da estrutura do algoritmo tradicional do método dos gradientes conjugados (Sonneweld [19]), notamos que poderá ser escrito

$$\begin{aligned} r_k &= \varphi_k(A) r_0 \\ p_k &= \psi_k(A) r_0 \end{aligned} \tag{XII.1}$$

onde φ_k e ψ_k são polinômios de grau menor ou igual a k , sendo

$$\varphi_0(A) = I \text{ e } \psi_{-1}(A) = 0.$$

Utilizando estas expressões é possível mostrar que

$$\begin{aligned} \psi_k(A) &= \varphi_k(A) + \beta_k \psi_{k-1}(A), \text{ e} \\ \varphi_{k+1}(A) &= \varphi_k(A) - \alpha_k A \psi_k(A) \end{aligned} \tag{XII.2}$$

Denotaremos o conjunto dos polinômios reais de grau menor ou igual a N por Π^N , e definiremos a forma bilinear simétrica (\cdot, \cdot) em Π^N por

$$(\varphi, \psi) = [\varphi(A) r_0]^T \psi(A) r_0$$

Deste modo, o algoritmo para o método dos gradientes conjugados pode ser reescrito como

Arbitrar x_0

$$r_0 = b - A x_0$$

$$\varphi_0 = I ; \psi_{-1} = 0 ; \rho_{-1} = 1$$

For $k = 0, 1, 2, \dots$

If ($\|r_k\| < \text{tolerância}$) (Fim)

$$\rho_k = (\varphi_k, \varphi_k)$$

$$\beta_k = \rho_k / \rho_{k-1}$$

$$\psi_k = \varphi_k + \beta_k \psi_{k-1}$$

XII.3

$$p_k = \psi_k(A) r_0$$

$$\sigma_k = (\psi_k, A \psi_k)$$

$$\alpha_k = \rho_k / \sigma_k$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$\varphi_{k+1} = \varphi_k - \alpha_k A \psi_k$$

$$r_{k+1} = \varphi_{k+1}(A) r_0$$

end

Para a construção do algoritmo do método dos gradientes conjugados quadráticos faremos uso do seguinte

Teorema: Seja $[\cdot, \cdot]$ uma forma bilinear simétrica definida em Π^N e satisfazendo

$$[\varphi \psi, \chi] = [\varphi, \psi \chi], \quad \forall \varphi, \psi, \chi \in \Pi^N$$

e sejam as seqüências de polinômios φ_k e ψ_k construídas conforme XII.3, porém com $[\cdot, \cdot]$ ao invés de (\cdot, \cdot) . Então, se o algoritmo não for interrompido devido a uma divisão por zero, os polinômios φ_k e ψ_k satisfazem

$$[\varphi_n, \varphi_m] = \rho_n \delta_{nm},$$

$$[\psi_n, A \psi_m] = \sigma_n \delta_{nm}, \quad \forall n, m$$

onde δ_{nm} é o delta de Kronecker.

Demonstração: Vide Sonneveld [19]

Utilizando o resultado deste teorema, podemos construir um método iterativo com

$$\rho_k = [\varphi_0, \varphi_k^2]$$

$$\sigma_k = [\psi_0, A \psi_k^2]$$

onde os vetores $\varphi_k^2(A) r_0$ poderão ser interpretados como resíduos da estimativa x_k para a solução x^* , onde o efeito contractante de $\varphi_k(A)$ é acelerado, melhorando a convergência em relação ao método dos gradientes conjugados.

O algoritmo resultante fica (vide Sonneweld [19]),

Arbitrar x_0

$$r_0 = b - A x_0$$

$$\rho_{-1} = 1 ; q_0 = p_{-1} = 0$$

For $k = 0, 1, 2, \dots$

If ($\|r_k\| < \text{tolerância}$) (Fim)

$$\rho_k = r_0^T r_k$$

$$\beta_k = \rho_k / \rho_{k-1}$$

$$u_k = r_k + \beta_k q_k$$

$$p_k = u_k + \beta_k (q_k + \beta_k p_{k-1})$$

$$v_k = A p_k$$

$$\sigma_k = r_0^T v_k$$

$$\alpha_k = \rho_k / \sigma_k$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = b - A x_{k+1}$$

$$q_{k+1} = u_k - \alpha_k v_k$$

end

APÊNDICE XIII

PRÉ-CONDICIONAMENTO

XIII.1 INTRODUÇÃO

É fato conhecido que o método dos gradientes conjugados converge mais rapidamente se os autovalores da matriz A não diferem muito entre si (Luenberger [11], cap. 8). A idéia básica do pré-condicionamento é transformar o sistema linear $Ax = b$ (A simétrica e definida positiva) no sistema equivalente

$$(C^{-1}A D^{-1})(D x) = C^{-1}b \quad \text{XIII.1.1}$$

ou seja, $\tilde{A} \tilde{x} = \tilde{b}$,

com, $\tilde{A} = C^{-1}A D^{-1}$

$$\tilde{x} = D x$$

$$\tilde{b} = C^{-1}b ,$$

de tal modo que os autovalores de \tilde{A} estejam mais próximos entre si do que os autovalores de A .

As matrizes C e D normalmente são escolhidas aproveitando a estrutura de A , e de modo que os sistemas lineares do tipo $Dx = \tilde{x}$, $\tilde{b} = C^{-1}b$ e derivados sejam facilmente resolvidos.

XIII.2 MÉTODO DOS GRADIENTES CONJUGADOS COM PRÉ-CONDICIONAMENTO

Se definirmos as matrizes $C = D$, $M = C^2$, e o vetor z_k como a solução do sistema linear $Mz_k = r_k$, o algoritmo para o método dos gradientes conjugados aplicado à resolução do sistema normal $A^T A x = A^T b$ pode ser modificado (Golub [8], cap. 10) resultando em

```

Arbitrar  $x_0$ 
 $r_0 = A^T(b - Ax_0)$ 
 $z_0 = M^{-1}r_0$ 
 $p_0 = z_0$ 
For  $k = 1, 2, \dots$ 
    If ( $\|r_{k-1}\| < \text{tolerância}$ ) (Fim)
     $t = \langle r_{k-1}, z_{k-1} \rangle / \langle Ap_{k-1}, Ap_{k-1} \rangle$ 
     $x_k = x_{k-1} + tp_{k-1}$ 
     $r_k = A^T(b - Ax_k)$ 
     $z_k = M^{-1}r_k$ 
     $\beta = \langle r_k, z_k \rangle / \langle r_{k-1}, z_{k-1} \rangle$ 
     $p_k = z_k + \beta p_{k-1}$ 
end

```

XIII.3 MÉTODO DOS SUBESPAÇOS DE KRYLOV COM PRÉ-CONDICIONAMENTO

Escolhendo $D = I$ (matriz identidade), e substituindo A por $C^{-1}A$ e b por $C^{-1}b$, o algoritmo do Apêndice XI fica

Arbitrar x_0 e M

$$r_0 = C^{-1}(b - Ax_0)$$

For $k = 0, 1, 2, \dots$

If ($\|r_k\| < \text{tolerância}$) (Fim)

$$\text{Calcular } V_m = [r_k, C^{-1}Ar_k, (C^{-1}A)^2r_k, \dots, (C^{-1}A)^{M-1}r_k]$$

Calcular y_m resolvendo o sistema linear

$$\sum_{j=1}^M \alpha_j \langle C^{-1}A v_j, v_i \rangle = \langle r_0, v_i \rangle, \quad i=1 \dots M$$

$$z_m = V_m y_m$$

$$x_{k+1} = x_k + z_m$$

$$r_{k+1} = C^{-1}(b - Ax_{k+1})$$

end

XIII.4 MÉTODO DOS GRADIENTES CONJUGADOS QUADRÁTICOS COM PRÉ-CONDICIONAMENTO

Escolhendo $D = I$ (matriz identidade), e substituindo A por $C^{-1}A$ e b por $C^{-1}b$, o algoritmo do Apêndice XII fica

Arbitrar x_0

$$r_0 = C^{-1}(b - A x_0)$$

$$\rho_{-1} = 1 ; q_0 = p_{-1} = 0$$

For $k = 0, 1, 2, \dots$

If ($\|r_k\| < \text{tolerância}$) (Fim)

$$\rho_k = r_0^T r_k$$

$$\beta_k = \rho_k / \rho_{k-1}$$

$$u_k = r_k + \beta_k q_k$$

$$p_k = u_k + \beta_k (q_k + \beta_k p_{k-1})$$

$$v_k = C^{-1} A p_k$$

$$\sigma_k = r_0^T v_k$$

$$\alpha_k = \rho_k / \sigma_k$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = C^{-1}(b - A x_{k+1})$$

$$q_{k+1} = u_k - \alpha_k v_k$$

end

REFERÊNCIAS

- [1] ABDEL-SALAM, M., EL. MOHANDES, M. T., Combined method based on finite differences and charge simulation for calculating electric fields, IEEE Transactions on Industry Applications, vol. 25, pp. 1060-1066, Novembro-Dezembro 1989
- [2] BRAMBLE, J. H., HUBBARD, B. E., Aproximation of solutions of mixed boundary value problems for Poisson's equation by finite differences, Journal of the Association for Computing Machinery, Vol. 12, nº 1, pp 114-123, Janeiro 1965
- [3] CHOW, Y. L., CHARALAMBOUS, C., Static-field computations by the method of optimized simulated charges, IEE Proceedings, Vol. 126, nº 1, pp 123-125, Janeiro 1979
- [4] EMSON, C. R. I., Methods for the solution of open - boundary electromagnetic-field problems, IEE Proceedings, Vol. 135, Pt. A, nº 3, pp. 151-158, março 1988
- [5] FERNANDES, M. R., Métodos variacionais para sistemas lineares esparsos: Uma aplicação a superfícies livres de capilaridade, Tese de Mestrado, UNICAMP/IMECC/DMA, 1993
- [6] FORSYTHE, G. E., WASOW, W. R., Finite-difference methods for partial differential equations, John Willey & Sons, 1960

- [7] FRANZ, A. F., et al, Finite boxes - A generalization of the finite difference method suitable for semiconductor device simulation, IEEE Transactions on electron devices, Vol. ED-30, n° 9, pp. 1070-1082, Setembro 1983
- [8] GOLUB, G. H., VAN LOAN, C. F., Matrix computations, The John Hopkins University Press, 1989
- [9] HOFFMANN, J. N., Programa CCE - Manual de Utilização, UNICAMP/IMECC/DMA, Abril 1993
- [10] JOHNK, C. T. A., Engineering electromagnetic fields and waves, John Willey & Sons, 1975
- [11] LUENBERGER, D. G., Introduction to linear and nonlinear programming, Addison-Wesley Publishing Company, 1972
- [12] MITCHELL, A. R., GRIFFITHS, D. F., The finite difference method in partial differential equations, John Willey & Sons, 1987
- [13] OKUBO, H., IKEDA, M., HONDA, M., Electric field analysis by combination method, IEEE Transactions on Power Apparatus and Systems, Vol. 101, pp. 4039-4048, Outubro 1982
- [14] OLSEN, R. G., WONG, P. S., Characteristics of low frequency electric and magnetic fields in the vicinity of electric power lines, IEEE Transactions on Power Delivery, Vol. 7, n° 4, pp 2046-2055, Outubro 1992

- [15] PARIS, D. T., HURD, F. K., Basic electromagnetic theory, McGraw-Hill B. C. Inc, 1969
- [16] SAAD, Y., Krylov subspace methods on supercomputers, J. Soc. Industrial and Applied Mathematics, Vol. 10, n° 6, pp 1200-1232, Novembro 1989
- [17] SAAD, Y., SCHULTZ, M. H., Conjugate gradient-like algorithms for solving nonsymmetric linear systems, Math. Comput., n° 44, pp 417-424, 1985
- [18] SINGER, H., STEINBIGLER, H., WEIB, P., A charge simulation method for high voltage fields, IEEE Transactions on Power Apparatus and Systems, Vol. 93, pp. 1660-1668, 1974
- [19] SONNEWELD, P., CGS, a fast Lanczos-type solver for nonsymmetric linear systems, Siam J. Sci. Stat. Comput., Vol. 10, n° 1, pp 36-52, Janeiro 1989
- [20] STEINBIGLER, H., Combined application of finite element method and charge simulation method for the computation of electric fields, 3rd International Symposium on High Voltage Engineering, grupo 11-11, Agosto 1979
- [21] SYNGE, J. L., SCHILD, A., Tensor calculus, Univ. of Toronto Press, Toronto, 1952

PROGRAMA CCE
para
CÁLCULO DE CAMPOS ELÉTRICOS

MANUAL
DE
UTILIZAÇÃO

Preparado por:

João Nelson Hoffmann

*Anexo à redação da tese de mestrado
"SIMULAÇÃO NUMÉRICA PARA O CÁLCULO
DE CAMPOS ELÉTRICOS EM DOMÍNIOS
ILIMITADOS" - UNICAMP / IMECC / DMA*

Abril, 1993

ÍNDICE

| | |
|--|-----|
| Capítulo 1 - Introdução | 3 |
| Capítulo 2 - Aplicação: Linha de transmissão de energia elétrica | 6 |
| Capítulo 3 - Instruções para entrada de dados | 9 |
| Capítulo 4 - Instruções para dimensionamento das variáveis do comando "parameter" | 18 |
| Capítulo 5 - Instruções para a execução do programa . | 20 |
| Capítulo 6 - Resultados | 23 |
| Capítulo 7 - Descrição das variáveis | 26 |
| Capítulo 8 - Programas auxiliares | 47 |
| Capítulo 9 - Listagem dos programas | 56 |
| Referências | 129 |

CAPÍTULO 1

INTRODUÇÃO

Grande parte dos problemas de eletrostática somente podem ser resolvidos utilizando métodos numéricos, como por exemplo, o método de diferenças finitas.

Por outro lado, em certos problemas que envolvem altos valores de potenciais elétricos (da ordem de centenas de quilovolts), surgem complicações adicionais. Primeiramente, os campos elétricos muitas vezes não estão limitados por uma região fechada, mas estendem-se ao infinito. Além disso, a influência da superfície da terra não pode ser desprezada. Estes casos podem ser convenientemente tratados por meio de outro método numérico, conhecido por "simulação de cargas", de desenvolvimento relativamente recente e de extensa aplicação em estudos elétricos com altas tensões.

O método de diferenças finitas (MDF) é próprio para problemas com domínio limitado, com contornos relativamente complexos e com características elétricas não homogêneas. O método de simulação de cargas (MSC) tem características

opostas, ou seja, é adequado para problemas com domínio ilimitado e características elétricas constantes, sendo particularmente próprio em casos que envolvam a superfície da terra (considerada idealmente como um plano condutor de potencial elétrico nulo).

É previsível, portanto, que a utilização conjunta do MSC com o MDF (método híbrido) possa ser vantajosa em muitos problemas. Para a aplicação deste método, escolhe-se uma região fechada arbitrária em cujo interior é utilizado o MDF (à qual chamaremos de região MDF) , sendo o exterior tratado pelo MSC (região MSC). A conexão entre eles resultará de determinadas condições que serão impostas sobre a fronteira comum escolhida. Maiores detalhes são fornecidos pelas referências inclusas no final.

O objetivo deste manual é apresentar as instruções necessárias para a utilização adequada do programa computacional CCE (Capítulos 2 a 7), escrito em Fortran 77 (versão para microcomputadores). Este programa utiliza o método híbrido de simulação de cargas com diferenças finitas para o cálculo de campos elétricos em torno de condutores isolados (problema tridimensional com simetria plana) e sua base teórica é a referência [1], tendo sido abordado também por [2,3,4] .

Problemas deste tipo surgem em linhas de energia elétrica de alta tensão (de até cerca de 800.000 Volts) onde os cabos, em geral, não possuem isolamento superficial externo. Neste caso, a simples aplicação do método de simulação de cargas é suficiente, por meio do programa auxiliar MSC (Capítulo 8) que pode ser utilizado para confronto de resultados com o programa CCE. Problemas com condutores isolados, no entanto, somente podem ser analisados por meio de métodos numéricos do tipo do método híbrido.

--

CAPÍTULO 2

APLICAÇÃO : LINHA DE TRANSMISSÃO DE ENERGIA ELÉTRICA

A figura 2.1 ilustra uma aplicação do método híbrido de simulação de cargas e diferenças finitas. O objetivo é calcular a distribuição dos potenciais elétricos próximo aos condutores de uma linha de transmissão de energia elétrica, com a finalidade de verificar se os níveis de campo elétrico não superam ao tolerável pelo ar atmosférico (vide referência [1]). Os potenciais elétricos dos condutores são os seguintes:

$$\Phi(P_1) = \Phi(P_2) = 0. \text{ V}$$

$$\Phi(F_1) = 296. \text{ kV}$$

$$\Phi(F_2) = \Phi(F_3) = -148. \text{ kV}$$

Para exemplificar este caso, escolhemos uma região retangular Ω abrangendo parte de F_1 , com uma malha (figura 2.2) de $23 \times 35 = 805$ nós internos igualmente espaçados, 120 nós sobre a fronteira $\partial\Omega$ (vide figura 2.2), e escolhemos uma relação nº de cargas/nº de nós na fronteira de cerca de 50%.

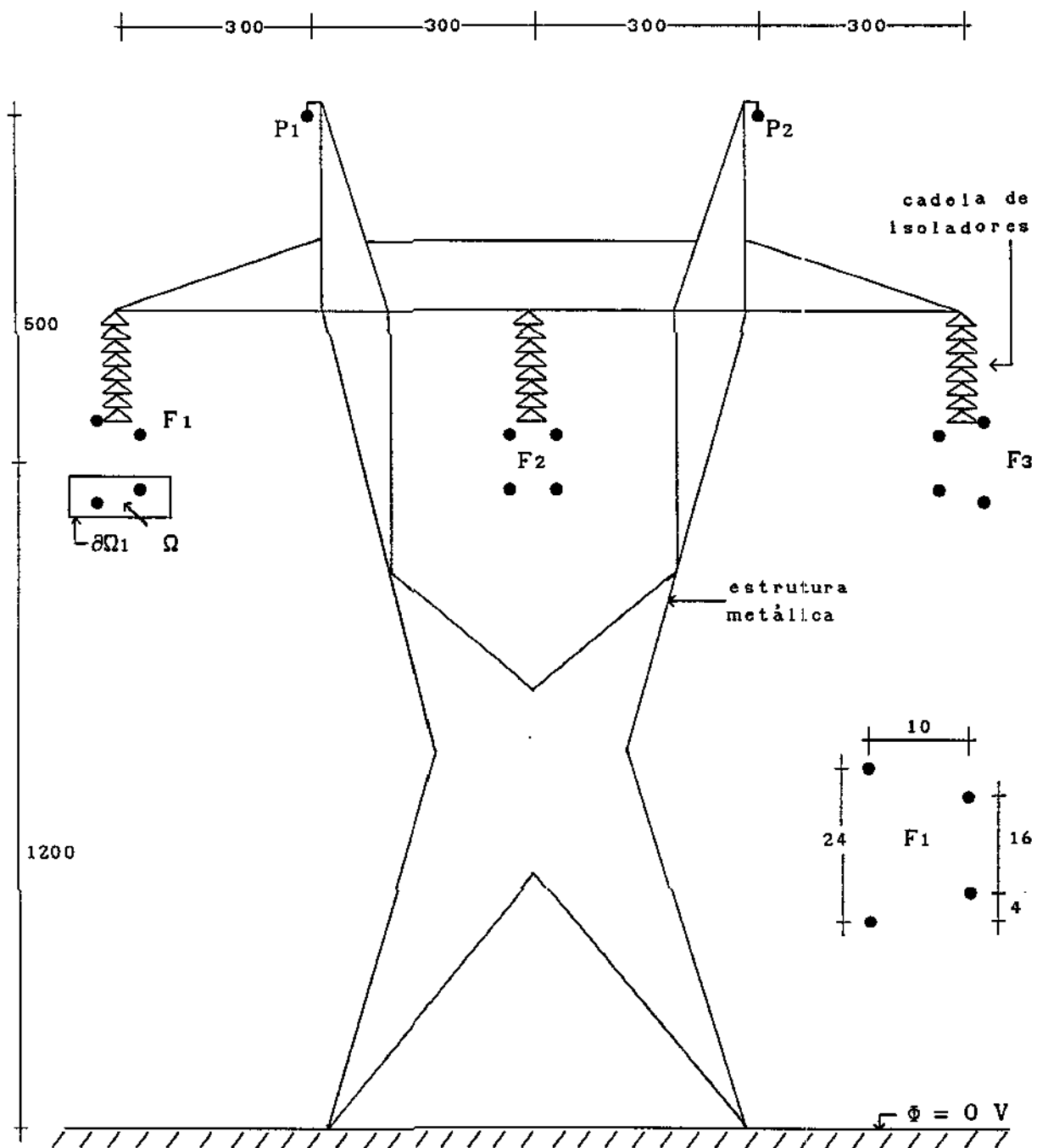


FIGURA 2.1 - Linha de transmissão de energia elétrica

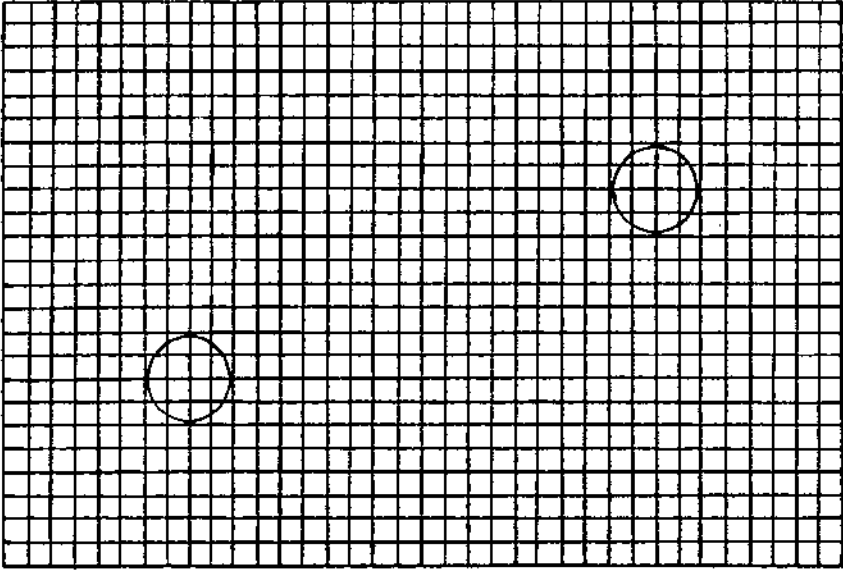


FIGURA 2.2 - Malha de diferenças finitas

Os dados de entrada e resultados para este problema serão detalhados nos capítulos seguintes (Capítulo 3 ao Capítulo 6). A execução do mesmo problema com o programa auxiliar MSC é feita na Capítulo 8 com a finalidade de comparação dos resultados.

CAPÍTULO 3

INSTRUÇÕES PARA A ENTRADA DE DADOS

Os dados para o problema apresentado na Capítulo 2 são descritos a seguir:

CONJUNTO DE DADOS:

1.^a linha: TESTE
2.^a linha: CST MQR
3.^a linha: 0 37 25 2 12 0 1.D-01 1.D-05 3 100 3000
4.^a linha: -4.0 14.0
5.^a linha: 8.0 -4.0
6.^a linha: 0. 1188. 1. 296. 0. 1.
10. 1192. 1. 296. 0. 1.
7.^a linha: 0. 1212. 0. 1211. 296. 0.7
10. 1208. 10. 1207. 296. 0.7
600. 1192. 600. 1191. -148. 0.7
610. 1192. 610. 1191. -148. 0.7
600. 1208. 600. 1207. -148. 0.7
610. 1208. 610. 1207. -148. 0.7
1200. 1188. 1200. 1187. -148. 0.7
1210. 1192. 1210. 1191. -148. 0.7
1200. 1212. 1200. 1211. -148. 0.7
1210. 1208. 1210. 1207. -148. 0.7
305. 1700. 305. 1699. 0. 0.7
905. 1700. 905. 1699. 0. 0.7

8.^a linha: -3.0 12
9.^a linha: 13.0 12
10.^a linha: 7.0 20
11.^a linha: -3.0 20
13.^a linha: 200. 0.7
14.^a linha: 5

INSTRUÇÕES:

1.^a linha: Identificação do estudo (formato A, colunas 01-20)

2.^a linha: Escolha dos métodos para cálculo (formato A):

Colunas 02-04: Método para resolução do problema
de diferenças finitas:

CG : método dos gradientes conjugados,
sem pré-condicionamento

CGD: idem, com pré-condicionamento diagonal

CGT: idem, com pré-condicionamento tridiagonal

KS : método dos subespaços de Krylov,
sem pré-condicionamento

KST: idem, com pré-condicionamento tridiagonal

CS : método dos gradientes conjugados
quadráticos, sem pré-condicionamento

CST: idem, com pré-condicionamento
tridiagonal

Colunas 06-08: Método para resolução do problema

de quadrados mínimos:

CQR: decomposição QR por meio do método
de Gram-Schmidt clássico

MQR: decomposição QR por meio do método
de Gram-Schmidt modificado

3.^a linha: Dados gerais (formatos I/F/D, colunas livres):

1) KOD1 = Código que indica como serão fornecidos os dados
para a construção da malha de diferenças finitas

KOD1 = 0 => Serão fornecidas apenas as coordenadas
das linhas extremas (bordas) da malha.

As coordenadas das linhas internas
serão computadas pelo programa,
assumindo passo constante.

KOD1 = 1 => Serão fornecidas as coordenadas de
todas as linhas horizontais e
verticais (inclusive bordas) que
comporão a malha de diferenças finitas.

Obs: As referidas coordenadas serão fornecidas
nos dados da 4.^a (abcissas) e 5.^a (ordenadas)
linhas

2)NO = Número de linhas verticais da malha de diferenças
finitas (inclusive bordas).

3)MO = Número de linhas horizontais da malha de diferenças
finitas (inclusive bordas).

- 4) NFDM = Número de condutores elétricos no interior da malha (região MDF)
- 5) NCSM = Número de cargas e pontos de contorno, externos à malha (região MSC)
- 6) KOD2 = Código que indica como serão localizadas as cargas no interior da região MDF

KOD2 = 0 => Será fornecida a localização (abscissa /ordenada) das linhas sobre as quais serão geradas as coordenadas das cargas, de forma igualmente espaçadas, conforme dados a serem fornecidos da 9.^a à 11.^a linhas.

KOD2 = N => Serão fornecidas as coordenadas (x,y) para cada uma das "N" cargas, conforme dados a serem fornecidos na 12.^a linha.

- 7) ERRO1 = Precisão desejada (%) para o vetor de cargas elétricas no método iterativo principal (sugestão: ERRO1 = 1.D-01 %).
- 8) ERRO2 = Precisão desejada (valor absoluto) na resolução do problema de diferenças finitas (sugestão: ERRO2 = 1.D-05).
- 9) N° de pontos internos à região para a discretização da derivada normal (indicar 2 ou 3)
- 10) Número máximo de iterações no processo iterativo principal

11) Número máximo de iterações na resolução do problema de diferenças finitas (sugestão: $3 \cdot NO \cdot MO$)

4.^a linha: Abscissas das linhas verticais da malha de diferenças finitas (formato F, colunas livres). As abscissas devem ser referidas ao centro de um condutor elétrico qualquer (interno à região MDF) e deverão estar ordenadas de modo crescente. Se $KOD1 = 0$, indicar apenas as abscissas das fronteiras verticais. Se $KOD1 = 1$, a quantidade de dados deve ser igual ao valor NO, fornecido na 3.^a linha.

5.^a linha: Ordenadas das linhas horizontais da malha de diferenças finitas (formato F, colunas livres). As ordenadas devem ser referidas ao centro do mesmo condutor elétrico mencionado nos dados da 4.^a linha, e deverão estar ordenadas de modo decrecente. Se $KOD1 = 0$, indicar apenas as ordenadas das fronteiras horizontais. Se $KOD1 = 1$, a quantidade de dados deve ser igual ao valor MO, fornecido na 3.^a linha.

6.^a linha: Dados dos condutores elétricos internos à região MDF (formato F, colunas livres). Incluir os seguintes dados (na ordem indicada):

- 1) Abscissa do centro, referida a uma origem arbitrária
- 2) Ordenada (altura em relação à superfície do solo) do centro
- 3) Raio do condutor

- 4) Potencial elétrico
- 5) Raio do isolamento externo do condutor
- 6) Permissividade elétrica relativa do isolamento externo

OBS: a) Repetir este conjunto de dados para cada condutor interno à região MDF.

b) A quantidade de conjunto de dados deve ser igual ao valor NFDM, indicado na 3.^a linha.

c) Os dados (abscissas e ordenadas) para a construção da malha de diferenças finitas indicados na 4.^a e 5.^a linhas são referidos ao centro do primeiro condutor elétrico (primeiro conjunto de dados) indicado

d) Se o condutor não tem isolamento externo, indicar raio do isolamento externo = 0.0 e atribuir um valor arbitrário à permissividade elétrica relativa.

7.^a linha: Dados das cargas elétricas e pontos de contorno da região MSC (formato F, colunas livres). Incluir os seguintes dados (na ordem indicada):

- 1) Abscissa da carga, referida à mesma origem escolhida na 6.^a linha de dados
- 2) Ordenada (altura em relação à superfície do solo) da carga
- 3) Abscissa do ponto de contorno, referida à mesma origem escolhida na 6.^a linha de dados

- 4) Ordenada (altura em relação à superfície do solo) do ponto de contorno
- 5) Potencial elétrico no ponto de contorno
- 6) Parâmetro θ para a carga elétrica (em geral, $0 \leq \theta < 1$)

OBS: a) Repetir este conjunto de dados para cada par carga elétrica/ponto de contorno da região MSC

b) A quantidade de conjunto de dados deve ser igual ao valor NCSM, indicado na 3.^a linha

8.^a à 11.^a linhas: Incluir estas linhas se $KOD2 = 0$

12.^a linha: Incluir esta linha se $KOD2 \neq 0$

8.^a linha: Dados para localização de um grupo de cargas internas à região MDF. Incluir os seguintes dados, na ordem indicada:

- 1) Abcissa de uma linha vertical (interna à região MDF) onde será localizado um conjunto com C cargas, igualmente espaçadas (formato F, colunas livres)
- 2) Quantidade C de cargas a ser localizadas nesta linha (formato I, colunas livres)

9.^a linha: Idem à 8.^a linha.

10.^a linha: Idem à 8.^a linha, porém é fornecida a ordenada de uma linha horizontal (interna à região MDF) onde será localizado o conjunto de cargas

11.^a linha: Idem à 10.^a linha.

OBS: As linhas verticais/horizontais fornecidas da 8.^a à 11.^a linha de dados formam um retângulo no interior da região MDF.

12.^a linha: Coordenadas de uma carga elétrica a ser localizada na região MDF (formato F, colunas livre). Incluir os seguintes dados (na ordem indicada):

- 1) Abcissa da carga, referida ao centro do primeiro condutor elétrico indicado na 6.^a linha de dados
- 2) Ordenada da carga, referida ao centro do primeiro condutor elétrico indicado na 6.^a linha de dados

OBS: A quantidade deste conjunto de dados (abcissa e ordenada para uma carga) deve ser igual ao valor KOD2, indicado na 3.^a linha de dados

13.^a linha: Estimativas iniciais (formato F, colunas livres). Incluir os seguintes dados (na ordem indicada):

1) Potencial elétrico inicial, estimado para a fronteira comum do método híbrido

2) Parâmetro θ para as cargas elétricas internas à região MDF

14ª linha: Incluir esta linha se o método escolhido para a resolução do problema de diferenças finitas, escolhido na 2ª linha, for KS ou KST. Indica a dimensão (NSUB) escolhida para os subespaços de Krylov (formato I, livre). Em geral, NSUB = 5 a 30.

CAPÍTULO 4

INSTRUÇÕES PARA DIMENSIONAMENTO DAS VARIÁVEIS DO COMANDO "PARAMETER"

As variáveis listadas neste capítulo devem estar convenientemente dimensionadas sob os respectivos comandos "parameter", antes da execução de cada problema.

Programa Principal:

NCS = Número máximo de cargas na região MSC

NFD = Número máximo de condutores na região MDF

NC = Número máximo de cargas na região MDF

NP = Número máximo de nós na fronteira comum

NOS = Número máximo de nós na malha MDF

NB = Número máximo de nós na curva de descontinuidade
da permissividade elétrica

Subrotina DADOS:

MAX = Número máximo de linhas verticais e linhas
horizontais da malha de diferenças finitas

Subrotina DIFER:

NOS = Idem ao parâmetro NOS do programa principal

NB = Idem ao parâmetro NB do programa principal

Subrotinas CG, KS e CGS:

MAX = Ordem máxima permissível do sistema de equações

Subrotina KS:

MSPACE = Máxima dimensão permissível para o
subespaço de Krylov.

CAPÍTULO 5

INSTRUÇÕES PARA A EXECUÇÃO DO PROGRAMA

Para a execução do programa CCE, verificar inicialmente a existência do módulo CCE.EXE, e de um módulo com o conjunto de dados, o qual deverá ter 4 caracteres no nome e extensão .DAT (por exemplo, DATA.DAT). A execução é feita como segue:

1.) Digitar CCE

2.) O programa responderá com :

```
ENTRE COM O NOME DO ARQUIVO DOS  
DADOS DE ENTRADA >>>
```

3.) Digitar apenas o nome do módulo de dados (por exemplo, DATA) sem extensão, que será assumida como sendo .DAT

4.) O programa responderá com:

```
ENTRE COM O NOME DO ARQUIVO PARA A  
SAIDA DOS RESULTADOS >>>
```

5.) Digite apenas o nome desejado , com 4 caracteres (por exemplo, SAID) e sem extensão. O programa irá gerar três módulos diferentes para a saída de resultados, que são:

SAID1.DAT : listagem dos dados de entrada lidos

SAID2.DAT : resultados finais

SAID3.DAT : listagem dos resultados na forma (x,y,z),

para utilização como entrada de dados

para determinação de curvas de nível

6.) O programa fornecerá resultados parciais em cada iteração do processo iterativo principal, na forma:

DIF. APOS NN ITERACOES : AAA.AA % ; CONVERGENCIA : BBB.BB %

onde: NN : Número da iteração

AAA.AA : variação percentual no vetor de cargas elétricas, relacionada com o parâmetro ERRO1 dos dados de entrada.

BBB.BB : indica a convergência do método para resolução do problema de diferenças finitas, expressa em percentagem da dimensão da matriz do sistema. A convergência está relacionada com o parâmetro ERRO2 dos dados de entrada.

7.) Quando $AAA.AA < ERRO1$, os cálculos serão finalizados e será apresentada a mensagem:

PARA VER OS RESULTADOS DIGITE :

EDIT SAID1.DAT (LISTAGEM DOS DADOS DE ENTRADA)

EDIT SAID2.DAT (CONTEM RESULTADOS FINAIS)

EDIT SAID3.DAT (CONTEM DADOS PARA CURVAS DE NIVEL)

FIM DA EXECUCAO ...

CAPÍTULO 6

RESULTADOS

A listagem dos resultados do programa CCE terá a seguinte forma:

6.1 MÓDULO SAID1.DAT

DADOS DE ENTRADA:

TESTE

CST MQR

0 37 25 2 12 0 1.D-01 1.D-05 3 100 3000

-4.0 14.0

8.0 -4.0

| | | | | | | |
|-------|-------|-------|-------|-------|-----|----|
| 0. | 1188. | 1. | | 296. | 0. | 1. |
| 10. | 1192. | 1. | | 296. | 0. | 1. |
| 0. | 1212. | 0. | 1211. | 296. | 0.7 | |
| 10. | 1208. | 10. | 1207. | 296. | 0.7 | |
| 600. | 1192. | 600. | 1191. | -148. | 0.7 | |
| 610. | 1192. | 610. | 1191. | -148. | 0.7 | |
| 600. | 1208. | 600. | 1207. | -148. | 0.7 | |
| 610. | 1208. | 610. | 1207. | -148. | 0.7 | |
| 1200. | 1188. | 1200. | 1187. | -148. | 0.7 | |
| 1210. | 1192. | 1210. | 1191. | -148. | 0.7 | |
| 1200. | 1212. | 1200. | 1211. | -148. | 0.7 | |
| 1210. | 1208. | 1210. | 1207. | -148. | 0.7 | |
| 305. | 1700. | 305. | 1699. | 0. | 0.7 | |
| 905. | 1700. | 905. | 1699. | 0. | 0.7 | |
| -3.0 | 12 | | | | | |
| 13.0 | 12 | | | | | |
| 7.0 | 20 | | | | | |
| -3.0 | 20 | | | | | |
| 200. | 0.7 | | | | | |
| 5 | | | | | | |

6.2 MÓDULO SAID2.DAT

RESULTADOS DO PROCESSO ITERATIVO :

| | | | | | |
|-----------|----|------------|------------|-----------------|--------|
| DIF. APOS | 1 | ITERACOES: | 2509.34 % | ; CONVERGENCIA: | 3.98 % |
| DIF. APOS | 2 | ITERACOES: | 7423.02 % | ; CONVERGENCIA: | 4.97 % |
| DIF. APOS | 3 | ITERACOES: | 2571.65 % | ; CONVERGENCIA: | 4.35 % |
| DIF. APOS | 4 | ITERACOES: | 36806.52 % | ; CONVERGENCIA: | 4.10 % |
| DIF. APOS | 5 | ITERACOES: | 1570.43 % | ; CONVERGENCIA: | 3.85 % |
| | | | | | |
| DIF. APOS | 22 | ITERACOES: | 1.77 % | ; CONVERGENCIA: | 0.75 % |
| DIF. APOS | 23 | ITERACOES: | 1.04 % | ; CONVERGENCIA: | 0.62 % |
| DIF. APOS | 24 | ITERACOES: | 0.60 % | ; CONVERGENCIA: | 0.37 % |
| DIF. APOS | 25 | ITERACOES: | 0.36 % | ; CONVERGENCIA: | 0.37 % |
| DIF. APOS | 26 | ITERACOES: | 0.18 % | ; CONVERGENCIA: | 0.37 % |
| DIF. APOS | 27 | ITERACOES: | 0.19 % | ; CONVERGENCIA: | 0.87 % |

RESULTADOS FINAIS APOS 28 ITERACOES:

CARGAS, EM $C/(4*PI*E)$:

| N.O | X | Y | $C/(4*PI*E)$ |
|-------|---------|----------|--------------|
| 1) | 0.000 | 1212.000 | 8.52271 |
| 2) | 10.000 | 1208.000 | 7.50827 |
| 3) | 600.000 | 1192.000 | -4.67227 |
| 4) | 610.000 | 1192.000 | -4.39109 |
| 5) | 600.000 | 1208.000 | -4.39735 |
| | | | |
| 68) | 13.000 | 1190.455 | 0.50827 |
| 69) | 13.000 | 1191.364 | 0.61833 |
| 70) | 13.000 | 1192.273 | 0.60507 |
| 71) | 13.000 | 1193.182 | 0.45170 |
| 72) | 13.000 | 1194.091 | 0.22684 |

POTENCIAIS DOS NOS DA MALHA (KV):

| N.O | X | Y | V |
|-------|--------|----------|-----------|
| 1) | -3.500 | 1195.500 | 263.97672 |
| 2) | -3.000 | 1195.500 | 265.28064 |
| 3) | -2.500 | 1195.500 | 266.53928 |
| 4) | -2.000 | 1195.500 | 267.74594 |
| 5) | -1.500 | 1195.500 | 268.89452 |
| | | | |
| 799) | 10.500 | 1184.500 | 254.45933 |

| | | | |
|------|--------|----------|-----------|
| 800) | 11.000 | 1184.500 | 253.48324 |
| 801) | 11.500 | 1184.500 | 252.45773 |
| 802) | 12.000 | 1184.500 | 251.38517 |
| 803) | 12.500 | 1184.500 | 250.26966 |
| 804) | 13.000 | 1184.500 | 249.11597 |
| 805) | 13.500 | 1184.500 | 247.92949 |

POTENCIAIS DOS NOS DA FRONTEIRA (KV):

| N.O | X | Y | V |
|-------|--------|----------|-----------|
| 1) | -4.000 | 1196.000 | 262.36303 |
| 2) | -3.500 | 1196.000 | 263.66910 |
| 3) | -3.000 | 1196.000 | 264.93749 |
| 4) | -2.500 | 1196.000 | 266.16241 |
| 5) | -2.000 | 1196.000 | 267.33858 |
| | | | |
| 116) | 12.000 | 1184.000 | 249.67222 |
| 117) | 12.500 | 1184.000 | 248.60733 |
| 118) | 13.000 | 1184.000 | 247.50755 |
| 119) | 13.500 | 1184.000 | 246.37696 |
| 120) | 14.000 | 1184.000 | 245.22000 |

6.3 MÓDULO SAID3.DAT

| | | |
|---------|-----------|----------|
| -3.5000 | 1196.0000 | 263.6690 |
| -3.0000 | 1196.0000 | 264.9374 |
| -2.5000 | 1196.0000 | 266.1624 |
| -2.0000 | 1196.0000 | 267.3385 |
| | | |
| 12.5000 | 1184.5000 | 250.2696 |
| 13.0000 | 1184.5000 | 249.1159 |
| 13.5000 | 1184.5000 | 247.9294 |
| 1.0000 | 1188.0000 | 296.0000 |
| 0.5934 | 1188.8048 | 296.0000 |
| | | |
| 10.7833 | 1191.3783 | 296.0000 |
| 10.9651 | 1192.2615 | 296.0000 |

CAPÍTULO 7

DESCRIÇÃO DAS VARIÁVEIS

As variáveis do programa CCE, cuja listagem completa da programação está inclusa no Capítulo 8, são descritas nas páginas conforme o índice a seguir:

| <u>Identificação</u> | <u>página</u> |
|--------------------------|---------------|
| Programa principal | 27 |
| Subrotinas : | |
| TECL | 30 |
| DADOS | 30 |
| MALHA | 31 |
| CARGA | 31 |
| INIC | 31 |
| ATUB1 | 31 |
| ATUB2 | 31 |
| ATUPOT | 32 |
| TESTE | 32 |
| ITERAT | 32 |
| CONDUT | 32 |
| FALHA | 32 |
| IMPR1 | 32 |
| IMPR2 | 33 |
| IMPR3 | 33 |
| IMPR4 | 33 |
| CHEPOT | 33 |
| CHECAM | 33 |
| PIJ | 33 |
| FXIJ | 34 |
| FYIJ | 34 |
| CONEX | 34 |
| DIFER | 34 |
| DEFKOD | 35 |
| PASSO | 35 |
| CORREC | 36 |
| BORDA | 36 |
| DIRICH | 36 |

| | |
|-----------------------|----|
| DIAG | 36 |
| VIZIN | 36 |
| COEF12 e COEF13 | 37 |
| KRAM2 E KRAM3 | 37 |
| COORD | 37 |
| NEUMAN | 38 |
| GRAD | 38 |
| PRODAX | 38 |
| PATX | 38 |
| SUPNOR | 39 |
| DIGATA | 39 |
| BIDATA | 39 |
| TRIDLU | 39 |
| SOLVDG | 39 |
| RESID | 40 |
| SOLVTD | 40 |
| SRCH | 40 |
| TRIDIG | 40 |
| FULLAX | 41 |
| VETOR | 41 |
| UPPER | 41 |
| SOLVE | 42 |
| DECOMP | 42 |
| DIAGON | 42 |
| KSPACE | 42 |
| GALERK | 43 |
| CG | 43 |
| KRYLOV | 44 |
| CGS | 45 |
| CGSQR | 45 |
| MGSQR | 46 |

PROGRAMA PRINCIPAL

Objetivo: Cálculo de potenciais elétricos em torno de condutores isolados pelo método híbrido de simulação de cargas e diferenças finitas

Variáveis:

NCS No. máximo de cargas na região MSC
NFD No. máximo de condutores na região MDF
NC No. máximo de cargas na região MDF

NP N° máximo de pontos no contorno comum
 NO N° máximo de nós na malha MDF
 NB N° máximo de nós na curva de
 descontinuidade da permissividade elétrica
 IDIM1...6 Parâmetros para dimensionamento de variáveis
 CHEIA1 Matriz cheia com os coeficientes de potencial
 elétrico (submatrizes P_1+P_2)
 CHEIA2 Matriz cheia com os coeficientes de potencial
 elétrico e de campo elétrico (submatrizes P_2+F)
 ILIN N° de linha na submatriz esparsa
 JCOL N° de coluna na submatriz esparsa
 ESP Valores numéricos da submatriz esparsa
 FIO Potenciais elétricos na fronteira
 FI1 Potenciais elétricos na malha MDF
 FIC Potenciais elétricos dos condutores da região MDF
 B2 Vetor do lado direito do
 problema de diferenças finitas
 WK1,WK2 Vetores auxiliares para o problema
 de quadrados mínimos (simulação de cargas)
 RO Vetor com valores de cargas elétricas
 NI Idem
 POT1 Potenciais dos condutores da região MDF
 V Vetor com os valores numéricos da matriz esparsa R
 XCAR,YCAR Coordenadas das cargas elétricas da região MDF
 XCOND,YCOND Coordenadas das cargas elétricas da região MSC
 XNO,YNO Coordenadas dos nós da malha MDF
 XFRONT,YFRONT Coordenadas dos nós no contorno comum
 XFR,YFR Coordenadas dos pontos com potencial
 dado, na região MSC
 XCG,YCG Coordenadas das cargas elétricas da região MSC
 X2,Y2 Coordenadas dos nós na região de
 descontinuidade da permissividade elétrica
 RAI Raio dos condutores da região MDF
 RAIO Raio da região de descontinuidade
 da permissividade elétrica

| | |
|-------------------|---|
| POT | Potencial nos pontos da região MSC |
| EPS | Permissividade elétrica na região MDF |
| TETA | Parâmetro θ para as cargas elétricas |
| DIAG | Diagonal da matriz do problema de diferenças finitas |
| SUB | Subdiagonal inferior da matriz do problema de diferenças finitas |
| SUP | Subdiagonal superior da matriz do problema de diferenças finitas |
| AUXFR | Vetor auxiliar com espaçamentos próximos à fronteira comum |
| ERRO1 | Precisão para o vetor de cargas elétricas |
| ERRO2 | Precisão para o método de resolução do problema de diferenças finitas |
| DIF | Norma infinito do vetor de cargas elétricas |
| POTO | Potencial inicial arbitrado para os nós da fronteira comum |
| VET1,VET2,LK1,LK2 | Vetores de conexão do sistema de equações |
| KOD | Códigos dos nós da malha de diferenças finitas |
| KONT | Contador de iterações do processo iterativo principal |
| ITER | Nº de iterações no método iterativo para a resolução do problema de diferenças finitas |
| LIMAX | Nº máximo de iterações no método iterativo para a resolução do problema de diferenças finitas |
| MAXIT | Nº máximo de iterações do processo iterativo principal |
| NELEM | Nº de elementos não nulos da submatriz esparsa do problema de diferenças finitas |
| NSUB | Dimensão do subespaço de Krylov |
| IFLAG | Código que indica pré-condicionamento ou não |
| NCSM | Nº de cargas e pontos de contorno na região MSC |
| NFDM | Nº de condutores na região MDF |
| NCG | Nº de cargas arbitradas internas à região MDF |
| NNO | Nº de nós internos à região MDF |
| NFR | Nº de nós na fronteira comum |

| | |
|-----------|--|
| NO | Nº de linhas verticais da região MDF, incl. bordas |
| MO | Nº de linhas horizontais da região MDF, incl. bordas |
| M2 | Nº de nós no contorno de descontinuidade da permissividade elétrica |
| ENTR | Nome do arquivo de entrada de dados |
| SAID | Nome do arquivo de saída de resultados |
| MET | Método utilizado para a resolução do problema de diferenças finitas |
| QR | Método utilizado para a resolução do problema de quadrados mínimos |
| LIN | Código numérico para a entrada de dados |
| LOUT1...3 | Códigos para a saída dos resultados |
| PTS | Nº de pontos no interior da região para discretização da derivada normal |

Subrotina TECL

Objetivo : Troca informações com o teclado

Variáveis: Idem ao programa principal

Subrotina DADOS

Objetivo: Leitura de dados e cálculos iniciais

Variáveis:

| | |
|------|---|
| MAX | Nº máximo de linhas verticais e/ou horizontais da malha de diferenças finitas |
| XYCG | Abcissa/ordenada de uma linha de cargas da região MDF |
| NC | Nº de cargas elétricas sobre a respectiva linha de cargas |
| XM | Abcissas das linhas verticais da malha de diferenças finitas |
| YM | Ordenadas das linhas horizontais da malha de diferenças finitas |

NOME Título do estudo

Demais variáveis: Idem ao programa principal

Subrotina MALHA

Objetivo: Determina as coordenadas dos nós da malha de diferenças finitas e dos nós da fronteira comum

Variáveis:

XO,YO Coordenadas do centro do primeiro condutor indicado para a região MDF

DELTAX Passo na direção x

DELTAY Passo na direção y

Demais variáveis: Conforme programa principal e/ou subrotinas precedentes

Subrotina CARGA

Objetivo: Determina as coordenadas das cargas na região de diferenças finitas

Variáveis:

INCR Incremento para a localização de cargas elétricas sobre a respectiva linha de cargas

AUX1,AUX2 Variáveis auxiliares

Demais variáveis: Idem ao programa principal e/ou subrotinas precedentes

Subrotina INIC

Objetivo: Define dados iniciais para os cálculos

Variáveis: Idem ao programa principal

Subrotina ATUB1

Objetivo: Atualiza o vetor do lado direito das equações de continuidade do campo elétrico

Variáveis: Idem ao programa principal

Subrotina ATUB2

Objetivo: Atualiza o vetor do lado direito das

equações de diferenças finitas

Variáveis: Idem ao programa principal

Subrotina ATUPOT

Objetivo: Atualiza o vetor de potenciais
na fronteira comum

Variáveis: Idem ao programa principal

Subrotina TESTE

Objetivo: Compara os valores das cargas na
fronteira e atualiza

Variáveis:

AUX Variável auxiliar

Demais variáveis: Idem ao programa principal

Subrotina ITERAT

Objetivo: Imprime resultados parciais

Variáveis:

PORC Nº de iterações no problema de diferenças finitas,
relativo à dimensão da matriz dos coeficientes (%)

Demais variáveis: Idem ao programa principal

Subrotina CONDUT

Objetivo: Define novos nós para plotagem,
sobre a superfície dos condutores

Variáveis: Idem ao programa principal

Subrotina FALHA

Objetivo: Imprime mensagens de erro

Variáveis:

NUM Variável auxiliar

Demais variáveis: Idem ao programa principal

Subrotina IMPR1

Objetivo: Imprime resultados finais

Variáveis: Idem ao programa principal

Subrotina IMPR2

Objetivo: Imprime resultados para plotagem
de curvas de nível

Variáveis: Idem ao programa principal

Subrotina IMPR3

Objetivo: Imprime resultados parciais

Variáveis: Idem ao programa principal

Subrotina IMPR4

Objetivo: Imprime resultados parciais

Variáveis:

POT,KOD Variáveis auxiliares

DIM Dimensão do problema de diferenças finitas

Demais variáveis: Idem ao programa principal

Subrotina CHEPOT

Objetivo: Determina os elementos das matrizes P₁ e P₂,
e armazena em CHEIA1 E CHEIA2

Variáveis:

AUX Variável auxiliar

Demais variáveis: Idem ao programa principal

Subrotina CHECAM

Objetivo: Copia os elementos da matriz P₂ e determina
os elementos da matriz F

Variáveis:

AUX Variável auxiliar

Demais variáveis: Idem ao programa principal

Subrotina PIJ

Objetivo: Calcula coeficiente de potencial elétrico

Variáveis:

AUX1,AUX2,AUX3 Variáveis auxiliares
XI,YI Coordenadas do ponto
XJ,YJ Coordenadas da carga
P Coeficiente de potencial elétrico

Subrotina FXIJ

Objetivo: Calcula coeficiente de campo
 elétrico na direção x

Variáveis:

AUX1...5 Variáveis auxiliares
XI,YI Coordenadas do ponto
XJ,YJ Coordenadas da carga
FX Coeficiente de campo elétrico na direção x

Subrotina FYIJ

Objetivo: Calcula coeficiente de campo
 elétrico na direção y

Variáveis:

AUX1...5 Variáveis auxiliares
XI,YI Coordenadas do ponto
XJ,YJ Coordenadas da carga
FY Coeficiente de campo elétrico na direção y

Subrotina CONEX

Objetivo: Determina vetores de conexão
 do sistema de equações

Variáveis: Idem ao programa principal

Subrotina DIFER

Objetivo: Determina os elementos da matriz
 de diferenças finitas

Variáveis:

HN,HS,HW,HE Passos relativos aos nós vizinhos
IN,IS,IW,IE N° dos nós vizinhos
COEF Coeficientes de ponderação para

a discretização da derivada normal

- P Armazena os nós vizinhos a um nó
pertencente à linha de descontinuidade
da permissividade elétrica
- BORDAX,BORDAY Indica o nº do nó da linha de descontinuidade
da permissividade elétrica que é vizinho ao
nó referido ao índice destes vetores.
- NUM Indica todos os nós possíveis que poderão compor
a discretização para a derivada normal
- VIZ Indica os nós que efetivamente compõem
a discretização para a derivada normal
- COD Código que indica a localização de um nó
- K Contador dos elementos não nulos da submatriz
esparsa do problema de diferenças finitas
- I0 Contador dos nós precedentes
- I1 Contador dos nós de uma linha da malha
- I2 Numero do nó da malha cuja equação
está sendo construída
- Demais variáveis: Idem ao programa principal

Subrotina DEFKOD

Objetivo: Define código de posição de cada nó
da malha de diferenças finitas

Variáveis:

DIST Distância do nó ao centro do condutor

AUXA,AUXB Variáveis auxiliares

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina PASSO

Objetivo: Determina o passo ph, qh, rh ou sh

Variáveis:

H Passo da malha

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina CORREC

Objetivo: Corrige o passo próximo à borda

Variáveis:

AUX Variável auxiliar

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina BORDA

Objetivo: Define nós sobre a curva de
 descontinuidade da permissividade
 elétrica (ϵ), e seus nós vizinhos

Variáveis:

AUX1,AUX2 Variáveis auxiliares

BORD Variável associada com BORDAX ou BORDAY
 da subrotina DIFER

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina DIRICH

Objetivo: Define um elemento da equação de
 diferenças finitas com condições de
 contorno de Dirichlet

Variáveis:

H1,H2 Variáveis (passos) auxiliares

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina DIAG

Objetivo: Define um elemento da diagonal do
 sistema de equações

Variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina VIZIN

Objetivo: Selecciona os nós vizinhos possíveis, em
cada nó sobre a curva de descontinuidade de ε

Variáveis:

IAUX Numeração de um nó

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotinas COEF12 e COEF13

Objetivo: Escolhe 2 ou 3 nós em cada lado da curva de
descontinuidade de ε , para compor as equações
com condição de Neumann

Variáveis:

ABC,ORD Coordenadas dos nós que irão compor
 as equações com condições de Neumann

L Índice para a variável NUM (subrotina DIFER)

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina KRAM2 e KRAM3

Objetivo: Resolve um sistema de equações 2x2 ou 3x3
que fornecerá os coeficientes que comporão
as equações da derivada normal

Variáveis:

L1,L2,L3 Linhas da matriz do sistema 3x3 que
 definirá os coeficientes a_i

D Cofatores da matriz

DELTA Determinante da matriz

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina COORD

Objetivo: Define as coordenadas de um nó interno à malha,
para compor as equações da derivada normal

Variáveis:

A,B,C,D,E,F Variáveis auxiliares

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina NEUMAN

Objetivo: Determina os elementos que compõe as
 equações de continuidade da derivada normal

Variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina GRAD

Objetivo: Calcula o resíduo $R = A^T (B - A X)$

Variáveis:

NELEM N° de elementos não nulos da matriz

NDIM Dimensão do sistema linear

WORK Vetor auxiliar

B Lado direito do sistema de equações

X Vetor de incógnitas

R Resíduo do sistema linear

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina PRODAX

Objetivo: Calcula o produto $Y = A X$

Variáveis:

X Vetor X

Y Vetor Y

NDIM Dimensão da matriz A

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina PATX

Objetivo: Calcula o produto $Y = A^T X$

Variáveis:

X Vetor X

Y Vetor $Y = A^T X$

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina SUPNOR

Objetivo: Calcula $\| X \|_{\infty}$

Variáveis:

X Vetor X

DIST Norma infinito do vetor X

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina DIGATA

Objetivo: Calcula a diagonal de $A^T A$

Variáveis:

NCOL N° de colunas da matriz A

DIAG Diagonal de $A^T A$

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina BIDATA

Objetivo: Calcula $A^T A$, sendo A matriz bidiagonal

Variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina TRIDLU

Objetivo: Decomposição LU de uma matriz tridiagonal

Variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina SOLVDG

Objetivo: Resolve um sistema linear $D X = B$, sendo
 D uma matriz diagonal

Variáveis:

B Lado direito do sistema de equações

X Vetor de incógnitas

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina SOLVTD

Objetivo: Resolve um sistema linear $LU X = B$, sendo
 L e U computadas conforme a subrotina TRIDLU

Variáveis:

B Lado direito do sistema de equações

X Vetor de incógnitas

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina SRCH

Objetivo: Localiza um elemento da matriz A

Variáveis:

INEW N° da linha do elemento procurado

JNEW N° da coluna do elemento procurado

ANEW Valor numérico do elemento procurado

Demais variáveis Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina TRIDIG

Objetivo: Determina a parte tridiagonal da matriz A

Variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina RESID

Objetivo: Calcula o resíduo $R = B - A X$

Variáveis:

NELEM N° de elementos não nulos da matriz

NDIM Dimensão do sistema linear

B Lado direito do sistema de equações

X Vetor de incógnitas

R Resíduo do sistema linear

Demais variáveis: Idem ao programa principal

e/ou subrotinas precedentes

Subrotina FULLAX

Objetivo: Calcula $Y = A X$, A matriz cheia

Variáveis:

A Matríz cheia A
X Vetor X
Y Produto $Y = A X$
MLIN Nº de linhas da matriz A
NCOL Nº de colunas da matriz A
MAX Dimensão máxima das colunas da matriz A
SOMA Variável auxiliar

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina UPPER

Objetivo: Resolve um sistema linear triangular superior

Variáveis:

U Matríz triangular superior (forma vetorizada)
B Lado direito do sistema de equações
X Vetor de incógnitas
N Dimensão do sistema linear
SOMA Variável auxiliar

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina VETOR

Objetivo: Calcula a relação entre os parâmetros I, J e K

Variáveis:

I Nº da linha do elemento
J Nº da coluna do elemento
N Dimensão da matriz triangular superior
K Índice para o vetor que armazena
a matriz triangular superior

Subrotina SOLVE

Objetivo: Resolve o sistema linear $LU x = b$

Variáveis:

A Matriz que armazena os valores
 das matrizes L e U

P Vetor que indica a permutação de
 linhas ocorrida na decomposição $A = LU$

B Lado direito do sistema de equações

X Vetor de incógnitas

N Dimensão do sistema de equações

MAX Dimensão máxima das colunas da matriz A

SOMA Variável auxiliar

IP Variável (índice) auxiliar

Subrotina DECOMP

Objetivo: Decomposição LU da matriz A

Variáveis:

A Matriz a ser decomposta na forma $A = LU$

P Vetor que indica a permutação de
 linhas ocorrida na decomposição $A = LU$

N Dimensão da matriz A

MAX Dimensão máxima das colunas da matriz A

PIVO Elemento pivô na decomposição $A = LU$

Subrotina DIAGON

Objetivo: Determina a diagonal de uma matriz A

Variáveis:

INEW N° da linha do elemento a ser procurado

JNEW N° da coluna do elemento a ser procurado

ANEW Valor numérico do elemento da diagonal

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina KSPACE

Objetivo: Construção dos subespaços de Krylov

Variáveis:

NMAX Dimensão máxima das colunas da matriz Q

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina GALERK

Objetivo: Calcula nova aproximação utilizando
o método de Galerkin

Variáveis:

KL Dimensão máxima das colunas da matriz Q

KC Dimensão máxima das colunas da matriz U

SUMB,SUBV Variáveis auxiliares

Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina CG

Objetivo: Resolução de sistema linear esparso
pelo método dos gradientes conjugados

Variáveis:

A Vetor com os elementos da matriz coeficientes

X Vetor de incógnitas

B Lado direito do sistema de equações

NDIM Dimensão do sistema de equações

MAX Dimensão máxima das colunas de A

R Resíduo do sistema normalizado $A^T A X = A^T B$

WORK Vetor auxiliar

VET Vetor direção de descida

EPS Precisão do processo iterativo

XNEW Variável auxiliar

STEP Parâmetro t para correção do vetor X

ALFA Produto interno $\langle r_k, z_k \rangle$

BETA Parâmetro β , para correção de VET

GAMA Produto interno $\langle A VET, A VET \rangle$

AUX Variável auxiliar

ERRO Norma infinito do vetor $X^k - X^{k-1}$

RES Norma infinito do vetor R
Demais variáveis: Idem ao programa principal
e/ou subrotinas precedentes

Subrotina KRYLOV

Objetivo: Resolução de sistema linear esparsos
pelo método dos subespaços de Krylov

Variáveis:

A Vetor com os elementos da matriz dos coeficientes
X Vetor de incógnitas
B Lado direito do sistema de equações
NDIM Dimensão do sistema de equações
MAX Dimensão máxima das colunas de A
Q Matriz com (MSPACE - 2) colunas que formam uma
base ortonormal para o subespaço de Krylov
U Matriz do sistema linear referente
ao método de Galerkin
Z Vetor pertencente ao subespaço de Krylov,
solução do sistema linear de Galerkin
P Vetor que indica a permutação de
linhas ocorrida na decomposição LU
MSPACE Dimensão do subespaço de Krylov
R Resíduo $B - A X$
WORK Vetor auxiliar
VET Vetor direção de descida
EPS Precisão do processo iterativo
XNEW Variável auxiliar
STEP Parâmetro t para correção do vetor X
ALFA Produto interno $\langle r_k, z_k \rangle$
BETA Parâmetro β , para correção de VET
GAMA Produto interno $\langle A VET, A VET \rangle$
AUX Variável auxiliar
ERRO Norma infinito do vetor $X^k - X^{k-1}$
RES Norma infinito do vetor R

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina CGS

Objetivo: Resolução de sistema linear esparso pelo
 método dos gradientes conjugados quadráticos

Variáveis:

A Vetor com os elementos da matriz dos coeficientes
X Vetor de incógnitas
B Lado direito do sistema de equações
NDIM Dimensão do sistema de equações
MAX Dimensão máxima das colunas de A
Q,U,P Vetores auxiliares do método CGS
R Resíduo $B - A X$
RO Resíduo inicial
EPS Precisão do processo iterativo
XNEW Variável auxiliar
ALFA Parâmetro para correção do vetor X
BETA Parâmetro β , para correção dos vetores U e P
GAMA Parâmetro ρ_{n-1}
SIGMA Produto interno $\langle RO, A P \rangle$
ERRO Norma infinito do vetor $X^k - X^{k-1}$
RES Norma infinito do vetor R

Demais variáveis: Idem ao programa principal
 e/ou subrotinas precedentes

Subrotina CGSQR

Objetivo: Decomposição QR de uma matriz por meio
 do método de Gram-Schmidt clássico

Variáveis:

A Matriz cheia de dimensão $M \times N$, com $M \geq N$
V Vetor que armazena a matriz R
M Dimensão das colunas de A
N Dimensão das linhas de A

MAX Dimensão máxima das colunas da matriz A
ALFA,SOMA Variáveis auxiliares
NORMA2 Norma euclidiana de uma coluna de Q

Subrotina MGSQR

Objetivo: Decomposição QR de uma matriz por meio
do método de Gram-Schmidt modificado

Variáveis: Idem à subrotina CGSQR

CAPÍTULO 8

PROGRAMAS AUXILIARES

8.1 PROGRAMA MSC

8.1.1 OBJETIVO

O método híbrido de simulação de cargas e diferenças finitas é adequado para o cálculo de campos elétricos em problemas que envolvem regiões com permissividades elétricas diferentes.

Para o cálculo de campos elétricos em problemas de regiões com permissividade elétrica constante, poderá também ser utilizado o método de simulação de cargas isoladamente, de aplicação mais simples do que o método híbrido.

Assim, o programa MSC terá aqui a finalidade de gerar dados para fins de comparação com os valores obtidos pelo programa CCE.

8.1.2 PROCEDIMENTO UTILIZADO

O procedimento para os cálculos segue basicamente os seguintes passos:

1º Passo: Arbitra-se K cargas no interior, e N pontos sobre a superfície dos condutores. Neste programa, é arbitrada uma carga sobre o eixo e as demais distribuídas no interior de cada condutor segundo uma faixa circular com raios a serem especificados (R1 e R2). Os pontos de contorno são distribuídos de modo eqüidistante sobre a superfície do condutor.

2º Passo: Determina-se a matriz $P = [p_{ij}]$, formada pelos coeficientes de potencial elétrico, por meio da expressão

$$p_{ij} = \frac{1}{4\pi\epsilon\epsilon_0} \ln \left[\frac{(x_1 - x_j)^2 + (y_1 + y_j)^2}{(x_1 - x_j)^2 + (y_1 - y_j)^2} \right] \quad 8.1.2.1$$

onde $i=1\dots N$, $j=1\dots K$,

(x_1, y_1) são as coordenadas de um ponto arbitrado sobre a superfície de um condutor,

(x_j, y_j) são as coordenadas de uma carga arbitrada no interior de um condutor .

3º Passo: Resolve-se o sistema linear $P q = \Phi$ (sendo Φ = vetor com os potenciais elétricos dos N pontos de contorno) assim obtido, de ordem $N \times K$. Para a determinação do vetor de cargas q, utiliza-se o método dos quadrados mínimos.

4º Passo: Calcula-se o potencial elétrico em qualquer ponto externo aos condutores ou sobre sua superfície, por meio da

expressão

$$\Phi(P_1) = \sum_{j=1}^K p_{1j} \cdot q_j \quad 8.1.2.2$$

com p_{1j} também calculados por meio da expressão 8.1.2.1, porém com

(x_1, y_1) são as coordenadas do ponto P_1 cujo potencial será calculado ,

(x_j, y_j) são as coordenadas de uma carga de valor q_j , situada no interior do condutor.

8.1.3 INSTRUÇÕES PARA A ENTRADA DE DADOS

EXEMPLO DE CONJUNTO DE DADOS:

```

1.ª linha:  TESTE
2.ª linha:  14
3.ª linha:   0.  1188.   1.  8  0.  .9  16  296.  0
              10.  1192.   1.  8  0.  .9  16  296.  0
              0.  1212.   1.  8  0.  .9  16  296.  0
              10.  1208.   1.  8  0.  .9  16  296.  0
              600.  1192.   1.  1  0.  .9   1 -148.  0
              610.  1192.   1.  1  0.  .9   1 -148.  0
              600.  1208.   1.  1  0.  .9   1 -148.  0
              610.  1208.   1.  1  0.  .9   1 -148.  0
              1200.  1188.   1.  1  0.  .9   1 -148.  0
              1210.  1192.   1.  1  0.  .9   1 -148.  0
              1200.  1212.   1.  1  0.  .9   1 -148.  0
              1210.  1208.   1.  1  0.  .9   1 -148.  0
              305.  1700.   1.  1  0.  .9   1   0.  0
              905.  1700.   1.  1  0.  .9   1   0.  0
4.ª linha: -3.0  1195.0  17  1.  11  1.
    
```

INSTRUÇÕES:

1.ª linha: Identificação do estudo (formato A, colunas 01-20)

2.ª linha: N.º de condutores elétricos

3.ª linha: Dados dos condutores elétricos

(formato I/F, colunas livres):

| | |
|-------|---|
| 0 | Abcissa do centro do condutor, referida a uma origem arbitrária |
| 1188. | Ordenada do centro do condutor, referida à superfície da terra |
| 1. | Raio do condutor |
| 8 | N.º de cargas elétricas que serão simuladas |
| 0. | Raio R1 para a localização das cargas |
| .9 | Raio R2 para a localização das cargas |
| 16 | N.º de pontos de contorno |
| 296. | Potencial elétrico do condutor |
| 0 | N.º de pontos de contorno para checagem do potencial elétrico |

OBS: A quantidade deste conjunto de dados deve ser igual ao número de condutores especificados na 1.ª linha

4.^a linha: Dados dos pontos para cálculo de potenciais

elétricos (formato I/F, colunas livres):

| | |
|--------|--|
| -3.0 | Abcissa da origem de uma malha para cálculo dos potenciais |
| 1195.0 | Ordenada da origem da malha para cálculo dos potenciais |
| 35 | Nº de pontos na direção x, para cálculo de potenciais a partir da origem |
| .5 | Distâncias entre pontos na direção x |
| 23 | Nº de pontos na direção y, para cálculo de potenciais a partir da origem |
| .5 | Distância entre pontos na direção y |

8.1.4 INSTRUÇÕES PARA DIMENSIONAMENTO DAS VARIÁVEIS DO COMANDO "PARAMETER"

As variáveis listadas nesta seção devem estar convenientemente dimensionadas sob os respectivos comandos "parameter", antes da execução de cada problema.

Programa Principal:

IDIM1 = Número máximo total de pontos de contorno

IDIM2 = Número máximo total de cargas simuladas

IDIM3 = Número máximo total de condutores

8.1.5 INSTRUÇÕES PARA A EXECUÇÃO DO PROGRAMA

As instruções para a execução do programa são idênticas às instruções do programa CCE (Capítulo 4), exceto que o módulo executável é o MSC.EXE. O arquivo SAID3.DAT não é gerado.

8.1.6 RESULTADOS

A listagem dos resultados do programa MSC tem a seguinte forma:

8.1.6.1 MÓDULO SAID1.DAT

TESTE

DADOS DE ENTRADA (APOS CALCULOS INICIAIS):

| I | XQ | YQ | XF | YF | POT |
|-------|----------|-----------|-----------|-----------|----------|
| 1 | 0.0000 | 1188.0000 | 0.9239 | 1188.3827 | 296.0000 |
| 2 | 0.0802 | 1188.1005 | 0.7071 | 1188.7071 | 296.0000 |
| 3 | -0.0572 | 1188.2507 | 0.3827 | 1188.9239 | 296.0000 |
| 4 | -0.3475 | 1188.1674 | 0.0000 | 1189.0000 | 296.0000 |
| 5 | -0.4634 | 1187.7769 | -0.3827 | 1188.9239 | 296.0000 |
| | | | | | |
| 41 | 305.0000 | 1700.0000 | -0.9239 | 1211.6173 | 296.0000 |
| 42 | 905.0000 | 1700.0000 | -0.7071 | 1211.2929 | 296.0000 |
| 43 | | | -0.3827 | 1211.0761 | 296.0000 |
| 44 | | | 0.0000 | 1211.0000 | 296.0000 |
| | | | | | |
| 72 | | | 1211.0000 | 1208.0000 | 148.0000 |
| 73 | | | 306.0000 | 1700.0000 | 0.0000 |
| 74 | | | 906.0000 | 1700.0000 | 0.0000 |

8.1.6.2 MÓDULO SAID2.DAT

RESULTADOS:

CARGAS (C/4.PI.E)

| | |
|-------|---------|
| 1 | 15.1517 |
| 2 | -7.1653 |
| 3 | 0.1014 |
| 4 | 0.0743 |
| 5 | 0.2967 |
| | |
| 41 | -1.6138 |
| 42 | 2.1973 |

POTENCIAIS (V)

| N.O | X | Y | POT |
|-------|---------|-----------|----------|
| 1 | -3.5000 | 1195.5000 | 264.4362 |
| 2 | -3.0000 | 1195.5000 | 265.7294 |
| 3 | -2.5000 | 1195.5000 | 266.9774 |
| 4 | -2.0000 | 1195.5000 | 268.1733 |
| 5 | -1.5000 | 1195.5000 | 269.3111 |
| | | | |
| 799 | 10.5000 | 1184.5000 | 254.7262 |
| 800 | 11.0000 | 1184.5000 | 253.7568 |
| 801 | 11.5000 | 1184.5000 | 252.7384 |
| 802 | 12.0000 | 1184.5000 | 251.6734 |
| 803 | 12.5000 | 1184.5000 | 250.5655 |
| 804 | 13.0000 | 1184.5000 | 249.4196 |
| 805 | 13.5000 | 1184.5000 | 248.2410 |

8.2 PROGRAMA MDF

8.2.1 OBJETIVO

O programa MDF é uma versão modificada e simplificada do programa CCE, cuja finalidade é o cálculo de

campos elétricos no interior de uma região retangular fechada pelo método de diferenças finitas. Para tal, é necessário fornecer os valores dos potenciais elétricos em todos os nós da fronteira retangular (problema de Dirichlet).

8.2.2 PROCEDIMENTO UTILIZADO

O programa MDF resolve o problema de discretização da mesma forma que o programa CCE. Como os potenciais na fronteira retangular são fornecidos (dados de entrada), o problema é resolvido de modo direto (com uma iteração apenas).

8.2.3 INSTRUÇÕES PARA A ENTRADA DE DADOS

A entrada de dados é idêntica ao programa CCE (Capítulo 3), onde aplicável. Em adição, será necessário criar um arquivo com os valores dos potenciais elétricos sobre a fronteira, listados na forma de 1 dado por linha, ordenados da esquerda para a direita e de cima para baixo sobre a fronteira retangular. O nome (incluindo a extensão) do arquivo será solicitado pelo programa no decorrer da execução.

8.2.4 INSTRUÇÕES PARA DIMENSIONAMENTO DAS VARIÁVEIS DO COMANDO "PARAMETER"

Idêntico ao programa CCE, onde aplicável (Capítulo 4).

8.2.5 INSTRUÇÕES PARA A EXECUÇÃO DO PROGRAMA

Similar ao programa CCE (Capítulo 5). O módulo executável é o MDF.EXE.

8.2.6 RESULTADOS

É gerado o módulo SAID1.DAT listando os dados de entrada e o módulo SAID2.DAT com os potenciais elétricos calculados da malha, seguindo o padrão do programa CCE. O arquivo SAID3.DAT não é gerado.

CAPÍTULO 9

LISTAGEM DOS PROGRAMAS

9.1 PROGRAMA CCE

A listagem do programa CCE está inclusa nas páginas seguintes, de acordo com o índice a seguir:

| <u>Identificação</u> | <u>página</u> |
|--------------------------|---------------|
| Programa principal | 58 |
| Subrotinas : | |
| TECL | 61 |
| DADOS | 63 |
| MALHA | 65 |
| CARGA | 66 |
| INIC | 67 |
| CHEPOT | 68 |
| CHECAM | 70 |
| PIJ | 70 |
| FXIJ | 71 |
| FYIJ | 71 |
| CONEX | 72 |
| DIFER | 74 |
| DEFKOD | 76 |
| PASSO | 77 |
| CORREC | 78 |
| BORDA | 78 |
| DIRICH | 79 |
| DIAG | 79 |
| VIZIN | 80 |
| COEF12 | 81 |
| COEF13 | 82 |
| COORD | 83 |
| KRAM2 | 84 |
| KRAM3 | 84 |
| NEUMAN | 85 |
| ATUB1 | 86 |
| ATUB2 | 87 |

| | |
|--------------|-----|
| ATUPOT | 87 |
| TESTE | 88 |
| ITERAT | 88 |
| CONDUT | 89 |
| FALHA | 89 |
| IMPR1 | 92 |
| IMPR2 | 93 |
| IMPR3 | 94 |
| IMPR4 | 94 |
| GRAD | 97 |
| RESID | 97 |
| PRODAX | 98 |
| PINNER | 98 |
| PATX | 99 |
| SUPNOR | 99 |
| DIGATA | 100 |
| BIDATA | 100 |
| TRIDLU | 101 |
| SOLVDG | 101 |
| SOLVTD | 102 |
| SRCH | 103 |
| FULLAX | 103 |
| TRIDIG | 104 |
| SOLVE | 105 |
| DECOMP | 106 |
| DIAGON | 107 |
| KSPACE | 107 |
| GALERK | 108 |
| VETOR | 109 |
| PRDATX | 110 |
| UPPER | 110 |
| CG | 111 |
| KRYLOV | 113 |
| CGS | 114 |
| MGSQR | 116 |
| CGSQR | 117 |


```

1          KOD(IDIM4),KONT,ITER,LIMAX,MAXIT,NELEM,NSUB,IFLAG
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
CHARACTER *9 ENTRI,SAID1,SAID2,SAID3
CHARACTER *3 MET,QR
COMMON /CONVRG/ERRO2,LIMAX,ITER
COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS

C
C
C          LEITURA DE DADOS E INICIALIZACAO

LIN = 5
LOUT1 = 6
LOUT2 = 6
LOUT3 = 6
CALL TECL (ENTRI,SAID1,SAID2,SAID3,1)
CALL DADOS (XCOND,YCOND,RAI,POT,RAIO,EPS,XNO,YNO,XFRONT,YFRONT,
1          XCAR,YCAR,XFR,YFR,XCG,YCG,POT1,AUXFR,ERRO1,ERRO2,
2          LIMAX,MAXIT,POT0,TETA,NSUB,MET,QR)
IF(NCSM .GT. NCS) CALL FALHA (1,NCSM)
IF(NFDM .GT. NFD) CALL FALHA (2,NFDM)
IF(NCG .GT. NC ) CALL FALHA (3,NCG )
IF(NFR .GT. NP ) CALL FALHA (4,NFR )
IF(NNO .GT. NOS) CALL FALHA (5,NNO )
CALL INIC (POT1,POT0,POT(1),NB,B2,WK1,WK2,RO,NI,FI0,FI1)
CALL CHEPOT (XFRONT,YFRONT,XCAR,YCAR,XFR,YFR,
1          XCG,YCG,IDIM1,CHEIA1,CHEIA2)

C
C
C          CALCULO DA APROXIMACAO INICIAL
          PARA O VETOR DE CARGAS ELETRICAS

IF(QR .EQ. 'CQR') THEN
CALL CGSQR (CHEIA2,V,NCSM+NFR,NCSM+NCG,IDIM1)
ELSE
CALL MGSQR (CHEIA2,V,NCSM+NFR,NCSM+NCG,IDIM1)
ENDIF
CALL PRDATX (CHEIA2,WK1,WK2,NCSM+NFR,NCSM+NCG,IDIM1)
CALL UPPER (V,WK2,RO,NCSM+NCG)

C
C
C          CALCULO DOS ELEMENTOS DO SISTEMA DE EQUACOES

CALL CHECAM (CHEIA1,POT1,XFRONT,YFRONT,XCAR,YCAR,
1          XCG,YCG,IDIM1,CHEIA2,WK1)
CALL CONEX (AUXFR,VET1,VET2,LK1,LK2)
CALL DIFER (XCOND,YCOND,RAI,POT,RAIO,EPS,KOD,XNO,YNO,
2          AUXFR,ILIN,JCOL,ESP,B2,X2,Y2,NELEM)
IF(M2 .GT. NB) CALL FALHA (6,M2)

C
C
C          DECOMPOSICAO QR DA SUBMATRIZ P2 + F

IF(QR .EQ. 'CQR') THEN
CALL CGSQR (CHEIA2,V,NCSM+NFR,NCSM+NCG,IDIM1)
ELSE
CALL MGSQR (CHEIA2,V,NCSM+NFR,NCSM+NCG,IDIM1)

```

ENDIF

OPCAO PARA IMPRESSAO DE RESULTADOS
PARCIAIS (SUBROTINAS IMPR3 E IMPR4)

CALL IMPR3 (XNO, YNO, XFRONT, YFRONT, XCAR, YCAR, XFR, YFR, XCG, YCG)
CALL IMPR4 (CHEIA1, CHEIA2, IDIM1, NNO+M2, ILIN, JCOL,
1 ESP, VET1, VET2, LK1, LK2, POT1, B2, NELEM)

CALCULOS PRELIMINARES PARA PRECONDICIONAMENTO

IFLAG = 0
IF(MET .EQ. 'CGT' .OR. MET .EQ. 'KST' .OR. MET .EQ. 'CST') THEN
CALL TRIDIG (ILIN, JCOL, ESP, NELEM, NNO+M2, DIAG, SUP, SUB)
IF(MET .EQ. 'CGT') CALL BIDATA (DIAG, SUP, SUB, NNO+M2)
CALL TRIDLU (DIAG, SUP, SUB, NNO+M2)
IFLAG = 1

ENDIF

IF(MET .EQ. 'CGD') THEN
CALL DIGATA (JCOL, ESP, NELEM, NNO+M2, DIAG)
IFLAG = 2

ENDIF

RESOLUCAO ITERATIVA DO SISTEMA LINEAR

KONT=0
KONT=KONT+1
CALL ATUB2 (B2, FI0, VET2, LK2, WK2)

RESOLUCAO DO SISTEMA DE
EQUACOES DE DIFERENCAS FINITAS

IF(MET.EQ.'CG ' .OR. MET.EQ.'CGD' .OR. MET.EQ.'CGT') THEN
CALL CG (ILIN, JCOL, ESP, FI1, WK2,
1 DIAG, SUP, SUB, NNO+M2, NELEM, IFLAG)

ENDIF

IF(MET.EQ.'KS ' .OR. MET.EQ.'KST') THEN
CALL KRYLOV (ILIN, JCOL, ESP, FI1, WK2,
1 DIAG, SUP, SUB, NNO+M2, NELEM, IFLAG, NSUB)

ENDIF

IF(MET.EQ.'CS ' .OR. MET.EQ.'CST') THEN
CALL CGS (ILIN, JCOL, ESP, FI1, WK2,
1 DIAG, SUP, SUB, NNO+M2, NELEM, IFLAG)

ENDIF

CALCULO DO VETOR DE CARGAS
PELO METODO DOS QUADRADOS MINIMOS

CALL ATUB1 (FI0, FI1, POT1, VET1, LK1, WK1)
CALL PRDATX (CHEIA2, WK1, WK2, NCSM+NFR, NCSM+NCG, IDIM1)
CALL UPPER (V, WK2, NI, NCSM+NCG)

C TESTE DE CONVERGENCIA E ATUALIZACAO
C DOS POTENCIAIS ELETRICOS NA FRONTEIRA
C

CALL TESTE (RO,NI,TETA,DIF)
 IF(DIF.LE.ERRO1) GO TO 10
CALL ATUPOT (CHEIA1,RO,IDIM1,FI0)
CALL ITERAT (ITER,KONT,DIF)
 IF(KONT.GE.MAXIT) CALL FALHA (7,KONT)
GO TO 20

C
C IMPRESSAO DOS RESULTADOS
C

10 CALL IMPR1 (RO,FI0,FI1,XCG,YCG,XCAR,YCAR,
1 XNO,YNO,XFRONT,YFRONT,KONT)
 CALL CONDUT (FIC,XNO,YNO,XCOND,YCOND,RAI,POT)
CALL IMPR2 (XFRONT,YFRONT,XNO,YNO,X2,Y2,FI0,FI1,FIC)
CALL TECL (ENTR1,SAID1,SAID2,SAID3,2)
STOP
END

C
C
C
C *****
C *****
C ***** S U B R O T I N A S *****
C *****
C *****
C *****

=====
C
C SUBROUTINE TECL (ENTR1,SAID1,SAID2,SAID3,KOD)
C

C
C -----
C
C OBJETIVO: TROCA INFORMACOES COM O TECLADO
C (APENAS PARA USO COM MICROCOMPUTADORES)
C
C

INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
CHARACTER *4 ENTR,SAID
CHARACTER *9 ENTR1,SAID1,SAID2,SAID3
COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
 IF(KOD.GT.1) GO TO 10
 LIN=5
 LOUT1=6
 LOUT2=7

```

LOUT3=8
WRITE(*,*)
WRITE(*,*) ' ENTRE COM O NOME DO ARQUIVO DOS '
WRITE(*,*)
WRITE(*,*) ' DADOS DE ENTRADA >>> '
READ(*,20) ENTR
ENTR1 = ENTR//'.DAT'
WRITE(*,*)
WRITE(*,*) ' ENTRE COM O NOME DO ARQUIVO PARA A '
WRITE(*,*)
WRITE(*,*) ' SAIDA DOS RESULTADOS >>> '
READ(*,20) SAID
SAID1 = SAID//'1.DAT'
SAID2 = SAID//'2.DAT'
20 SAID3 = SAID//'3.DAT'
FORMAT(A4)
WRITE(*,*)
WRITE(*,*) ' AGUARDE ... '
WRITE(*,*)
WRITE(*,*)
OPEN (UNIT=LIN,FILE=ENTR1,STATUS='OLD')
OPEN (UNIT=LOUT1,FILE=SAID1,STATUS='NEW')
OPEN (UNIT=LOUT2,FILE=SAID2,STATUS='NEW')
OPEN (UNIT=LOUT3,FILE=SAID3,STATUS='NEW')
10 RETURN
WRITE(*,*)
WRITE(*,*) ' PARA VER OS RESULTADOS DIGITE : '
WRITE(*,*)
WRITE(*,*) ' EDIT ',SAID1,' (PARA CHECAR OS DADOS DE ENTRADA) '
WRITE(*,*) ' EDIT ',SAID2,' (CONTEM OS RESULTADOS FINAIS) '
WRITE(*,*) ' EDIT ',SAID3,' (CONTEM OS DADOS PARA QGRID) '
WRITE(*,*)
WRITE(*,*) ' FIM DA EXECUCAO ... '
RETURN
END

```

```

C
C =====
C
SUBROUTINE DADOS (XCOND, YCOND, RAI, POT, RAI0, EPS, XNO, YNO, XFRONT,
1 YFRONT, XCAR, YCAR, XFR, YFR, XCG, YCG, POT1, AUXFR, ERRO1, ERRO2,
2 LIMAX, MAXIT, POT0, TETA, NSUB, MET, QR)
C
C -----
C
C OBJETIVO: LEITURA DE DADOS E CALCULOS INICIAIS
C
C
PARAMETER (MAX=50)
REAL *8 XCOND(*), YCOND(*), RAI(*), POT(*), RAI0(*), EPS(*),
1 XNO(*), YNO(*), XFRONT(*), YFRONT(*), XCAR(*), YCAR(*),
2 XFR(*), YFR(*), XCG(*), YCG(*), POT1(*), TETA(*), AUXFR(*),
3 XYCG(4), XM(MAX), YM(MAX), ERRO1, ERRO2, POT0
INTEGER *4 KOD1, KOD2, LIMAX, MAXIT, NC(4), NSUB
INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
INTEGER *4 LIN, LOU1, LOU2, LOU3
CHARACTER *20 NOME
CHARACTER *3 MET, QR
COMMON /IMPRES/LIN, LOU1, LOU2, LOU3
COMMON /DIMENS/NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
WRITE(LOU1, *)
WRITE(LOU1, *) ' DADOS DE ENTRADA: '
READ(LIN, 10) NOME
10 FORMAT(A20)
WRITE(LOU1, 10) NOME
READ(LIN, 20) MET, QR
20 FORMAT(2(1X, A3))
WRITE(LOU1, 60) MET, QR
60 FORMAT(2(1X, A3))
READ(LIN, *) KOD1, NO, MO, NFD, NCSM, KOD2,
1 ERRO1, ERRO2, MAXIT, LIMAX
IF(NO .GT. MAX .OR. MO .GT. MAX) CALL FALHA (14, 0)
WRITE(LOU1, *) KOD1, NO, MO, NFD, NCSM, KOD2,
1 ERRO1, ERRO2, MAXIT, LIMAX
IF(KOD1 .EQ. 0) THEN
READ(LIN, *) XM(1), XM(2), YM(1), YM(2)
WRITE(LOU1, *) XM(1), XM(2), YM(1), YM(2)
ELSE
READ(LIN, *) (XM(I), I=1, NO), (YM(I), I=1, MO)
WRITE(LOU1, *) (XM(I), I=1, NO), (YM(I), I=1, MO)
ENDIF
READ(LIN, *) (XCOND(I), YCOND(I), RAI(I),
1 POT(I), RAI0(I), EPS(I), I=1, NFD)
WRITE(LOU1, *) (XCOND(I), YCOND(I), RAI(I),
1 POT(I), RAI0(I), EPS(I), I=1, NFD)
READ(LIN, *) (XCG(I), YCG(I), XFR(I), YFR(I),
1 POT1(I), TETA(I), I=1, NCSM)
WRITE(LOU1, *) (XCG(I), YCG(I), XFR(I), YFR(I),
1 POT1(I), TETA(I), I=1, NCSM)

```

```

CALL MALHA (KOD1, XM, YM, XNO, YNO,
1          XFRONT, YFRONT, XCOND(1), YCOND(1))
IF(KOD2 .EQ. 0) THEN
  READ (LIN ,*) (XYCG(I), NC(I), I=1,4)
  WRITE (LOUT1,*) (XYCG(I), NC(I), I=1,4)
  CALL CARGA (XYCG, NC, XCOND(1), YCOND(1), XCAR, YCAR)
  ELSE
  NCG=KOD2
  DO 40 I=1, NCG
    READ (LIN ,*) XCAR(I), YCAR(I)
    WRITE (LOUT1,*) XCAR(I), YCAR(I)
    XCAR(I)=XCAR(I)+XCOND(1)
    YCAR(I)=YCAR(I)+YCOND(1)
40  CONTINUE
ENDIF
NNO=(NO-2)*(MO-2)
NFR=NO+NO+(MO-2)*2
AUXFR(1) = XNO(1)-XFRONT(1)
AUXFR(2) = XFRONT(NFR)-XNO(NNO)
AUXFR(3) = YFRONT(1)-YNO(1)
AUXFR(4) = YNO(NNO)-YFRONT(NFR)
AUXFR(5) = XNO(2)-XNO(1)
AUXFR(6) = XNO(NNO)-XNO(NNO-1)
AUXFR(7) = YNO(1)-YNO(NNO-1)
AUXFR(8) = YNO(NNO-NO+2)-YNO(NNO)
READ (LIN ,*) POT0, TETA(NCSM+1)
WRITE (LOUT1,*) POT0, TETA(NCSM+1)
DO 50 I=NCSM+2, NCSM+NCG
  TETA(I)=TETA(NCSM+1)
50 CONTINUE
IF(MET .EQ. 'KS' .OR. MET .EQ. 'KST') THEN
  READ (LIN ,*) NSUB
  WRITE (LOUT1,*) NSUB
ENDIF
WRITE (LOUT1,*)
WRITE (*,*)
WRITE (*,*) ' RESULTADOS DO PROCESSO ITERATIVO : '
WRITE (*,*)
WRITE (LOUT2,*)
WRITE (LOUT2,*) ' RESULTADOS DO PROCESSO ITERATIVO : '
WRITE (LOUT2,*)
RETURN
END

```



```

C
C =====
C
SUBROUTINE MALHA (KOD, XM, YM, XNO, YNO, XFRONT, YFRONT, X0, Y0)
C -----
C
C OBJETIVO: DETERMINA AS COORDENADAS DOS NOS DA MALHA DE
C           DIFERENCAS FINITAS E DOS NOS NA FRONTEIRA COMUM
C
C
1  REAL      *8 XM(*), YM(*), XNO(*), YNO(*), XFRONT(*), YFRONT(*),
    X0, Y0, DELTAX, DELTAY
    INTEGER *4 I, J, K, KOD
    INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
    COMMON /DIMENS/ NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
    IF(KOD .NE. 0) GO TO 10
    DELTAX = (XM(2)-XM(1))/DFLOAT(NO-1)
    DELTAY = (YM(2)-YM(1))/DFLOAT(MO-1)
    DO 20 I=2, NO
        XM(I) = XM(I-1) + DELTAX
20  CONTINUE
    DO 30 I=2, MO
        YM(I) = YM(I-1) + DELTAY
30  CONTINUE
10  DO 40 I=1, MO-2
        K=(I-1)*(NO-2)
        DO 50 J=1, NO-2
            XNO(J+K)=X0+XM(J+1)
            YNO(J+K)=Y0+YM(I+1)
50  CONTINUE
40  CONTINUE
    DO 60 I=1, NO
        XFRONT(I)=X0+XM(I)
        YFRONT(I)=Y0+YM(1)
        XFRONT(NO+(MO-2)*2+I)=X0+XM(I)
        YFRONT(NO+(MO-2)*2+I)=Y0+YM(MO)
60  CONTINUE
    DO 70 I=1, MO-2
        XFRONT(NO+I+I-1)=X0+XM(1)
        YFRONT(NO+I+I-1)=Y0+YM(I+1)
        XFRONT(NO+I+I) =X0+XM(NO)
        YFRONT(NO+I+I) =Y0+YM(I+1)
70  CONTINUE
    RETURN
END

```

C
C
C

SUBROUTINE CARGA (XYCG,NC,X0,Y0,XCAR,YCAR)

C
C
C
C
C

OBJETIVO: DETERMINA AS COORDENADAS DAS CARGAS NA
REGIAO DE DIFERENCAS FINITAS

REAL *8 X0,Y0,XCAR(*),YCAR(*),XYCG(*),INCR(4),AUX1,AUX2
INTEGER *4 NC(*)
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
AUX1=XYCG(2)-XYCG(1)
AUX2=XYCG(3)-XYCG(4)
INCR(1)=AUX2/DFLOAT(NC(1)-1)
INCR(2)=AUX2/DFLOAT(NC(2)-1)
INCR(3)=AUX1/DFLOAT(NC(3)-1)
INCR(4)=AUX1/DFLOAT(NC(4)-1)
DO 10 I=1,NC(3)
XCAR(I)=X0+XYCG(1)+(I-1)*INCR(3)
YCAR(I)=Y0+XYCG(3)
10 CONTINUE
NCG=NC(3)
DO 20 I=1,NC(4)
XCAR(NCG+I)=X0+XYCG(1)+(I-1)*INCR(4)
YCAR(NCG+I)=Y0+XYCG(4)
20 CONTINUE
NCG=NCG+NC(4)
DO 30 I=1,NC(1)-2
XCAR(NCG+I)=X0+XYCG(1)
YCAR(NCG+I)=Y0+XYCG(4)+I*INCR(1)
30 CONTINUE
NCG=NCG+NC(1)-2
DO 40 I=1,NC(2)-2
XCAR(NCG+I)=X0+XYCG(2)
YCAR(NCG+I)=Y0+XYCG(4)+I*INCR(2)
40 CONTINUE
NCG=NCG+NC(2)-2
RETURN
END

```

C
C =====
C
C SUBROUTINE INIC (POT1,POT0,POT,NB,B2,WK1,WK2,RO,NI,FI0,FI1)
C -----
C
C OBJETIVO: DEFINE DADOS INICIAIS PARA OS CALCULOS
C
C
1 REAL *8 POT1(*),B2(*),WK1(*),WK2(*),
RO(*),NI(*),FI0(*),FI1(*),POT0,POT
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
DO 10 I=1,NFR
WK1(I)=POT0
10 CONTINUE
DO 20 I=1,NCSM
WK1(NFR+I)=POT1(I)
20 CONTINUE
DO 30 I=1,NNO+NB
B2(I)=0.D0
WK2(I)=0.D0
30 CONTINUE
DO 40 I=1,NCSM+NCG
RO(I)=0.D0
NI(I)=0.D0
40 CONTINUE
DO 50 I=1,NFR
FI0(I)=POT0
50 CONTINUE
DO 60 I=1,NNO+NB
FI1(I)=POT
60 CONTINUE
RETURN
END

```



```

C
C =====
C
SUBROUTINE CHECAM (CHEIA1,POT1,XFRONT,YFRONT,XCAR,YCAR,XCG,YCG,
1          IDIM1,CHEIA2,WK1)
C
C -----
C
C OBJETIVO: COPIA OS ELEMENTOS DA MATRIZ P2 E DETERMINA
C OS ELEMENTOS DA MATRIZ F
C
C
C REAL      *8 CHEIA1(IDIM1,*),CHEIA2(IDIM1,*),POT1(*),WK1(*),
1          XFRONT(*),YFRONT(*),XCAR(*),YCAR(*),XCG(*),YCG(*),AUX
C INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C
C MATRIZ P2
C
C DO 10 I=1,NCSM
C DO 10 J=1,NCSM+NCG
C CHEIA2(I,J)=CHEIA1(NFR+I,J)
10 CONTINUE
C
C MATRIZ F
C
C DIRECAO Y
C
C DO 20 I=1,NO
C K=NFR-NO+I
C DO 30 J=1,NCSM
C CALL FYIJ (XFRONT(I),YFRONT(I),XCG(J),YCG(J),AUX)
C CHEIA2(NCSM+I,J) = AUX
C CALL FYIJ (XFRONT(K),YFRONT(K),XCG(J),YCG(J),AUX)
C CHEIA2(NCSM+K,J) = -AUX
30 CONTINUE
C DO 40 L=1,NCG
C J=NCSM+L
C CALL FYIJ (XFRONT(I),YFRONT(I),XCAR(L),YCAR(L),AUX)
C CHEIA2(NCSM+I,J) = AUX
C CALL FYIJ (XFRONT(K),YFRONT(K),XCAR(L),YCAR(L),AUX)
C CHEIA2(NCSM+K,J) = -AUX
40 CONTINUE
20 CONTINUE
C
C DIRECAO X
C
C DO 50 I=1,MO-2
C K=NO+I+I-1
C M=K+1
C DO 60 J=1,NCSM
C CALL FXIJ (XFRONT(K),YFRONT(K),XCG(J),YCG(J),AUX)
C CHEIA2(NCSM+K,J) = -AUX

```

```

        CALL FXIJ (XFRONT(M),YFRONT(M),XCG(J),YCG(J),AUX)
        CHEIA2(NCSM+M,J) = AUX
60      CONTINUE
        DO 70 L=1, NCG
            J=NCSM+L
            CALL FXIJ (XFRONT(K),YFRONT(K),XCAR(L),YCAR(L),AUX)
            CHEIA2(NCSM+K,J) = -AUX
            CALL FXIJ (XFRONT(M),YFRONT(M),XCAR(L),YCAR(L),AUX)
            CHEIA2(NCSM+M,J) = AUX
70      CONTINUE
50      CONTINUE
C
        DO 80 I=1, NCSM
            WK1(I)=POT1(I)
80      CONTINUE
        RETURN
    END

```

```

C
C =====
C
SUBROUTINE PIJ (XI,YI,XJ,YJ,P)
C
C -----
C

```

```

C
C OBJETIVO:  CALCULA COEFICIENTE DE POTENCIAL ELETRICO
C
C

```

```

        REAL      *8 XI,YI,XJ,YJ,P,AUX1,AUX2,AUX3
        AUX1=XI-XJ
        AUX2=YI-YJ
        AUX3=YI+YJ
        P=DLOG((AUX3*AUX3 + AUX1*AUX1)/(AUX2*AUX2 + AUX1*AUX1))
        RETURN
    END

```

```

C
C =====
C
C SUBROUTINE FXIJ (XI,YI,XJ,YJ,FX)
C -----
C
C OBJETIVO: CALCULA COEFICIENTE DE CAMPO ELETRICO NA DIRECAO X
C
C REAL *8 XI,YI,XJ,YJ,FX,AUX1,AUX2,AUX3,AUX4,AUX5
C AUX1=XI-XJ
C AUX2=YI-YJ
C AUX3=YI+YJ
C AUX4=AUX2*AUX2 + AUX1*AUX1
C AUX5=AUX3*AUX3 + AUX1*AUX1
C FX=2.D0*AUX1*(1.D0/AUX4 - 1.D0/AUX5)
C RETURN
C END
C
C =====
C
C SUBROUTINE FYIJ (XI,YI,XJ,YJ,FY)
C -----
C
C OBJETIVO: CALCULA COEFICIENTE DE CAMPO ELETRICO NA DIRECAO Y
C
C REAL *8 XI,YI,XJ,YJ,FY,AUX1,AUX2,AUX3,AUX4,AUX5
C AUX1=XI-XJ
C AUX2=YI-YJ
C AUX3=YI+YJ
C AUX4=AUX2*AUX2 + AUX1*AUX1
C AUX5=AUX3*AUX3 + AUX1*AUX1
C FY=2.D0*(AUX2/AUX4 - AUX3/AUX5)
C RETURN
C END

```

```

=====
SUBROUTINE CONEX (AUXFR,VET1,VET2,LK1,LK2)
-----

```

OBJETIVO: DETERMINA VETORES DE CONEXAO DO SISTEMA DE EQUACOES

```

REAL      *8 AUXFR(*),VET1(3,*),VET2(*)
INTEGER  *4 LK1(2,*),LK2(*)
INTEGER  *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
DO 10 I=1,NO
  VET1(1,I)      = 1.D0/AUXFR(3) + 1.D0/(AUXFR(3)+AUXFR(7))
  VET1(2,I)      = -1.D0/AUXFR(3) - 1.D0/AUXFR(7)
  VET1(3,I)      = 1.D0/AUXFR(7) - 1.D0/(AUXFR(3)+AUXFR(7))
  VET1(1,NFR-NO+I) = 1.D0/AUXFR(4) + 1.D0/(AUXFR(4)+AUXFR(8))
  VET1(2,NFR-NO+I) = -1.D0/AUXFR(4) - 1.D0/AUXFR(8)
  VET1(3,NFR-NO+I) = 1.D0/AUXFR(8) - 1.D0/(AUXFR(4)+AUXFR(8))
10 CONTINUE
DO 20 I=1,MO-2
  VET1(1,NO+I+I) = 1.D0/AUXFR(2) + 1.D0/(AUXFR(2)+AUXFR(6))
  VET1(2,NO+I+I) = -1.D0/AUXFR(2) - 1.D0/AUXFR(6)
  VET1(3,NO+I+I) = 1.D0/AUXFR(6) - 1.D0/(AUXFR(2)+AUXFR(6))
  VET1(1,NO+I+I-1) = 1.D0/AUXFR(1) + 1.D0/(AUXFR(1)+AUXFR(5))
  VET1(2,NO+I+I-1) = -1.D0/AUXFR(1) - 1.D0/AUXFR(5)
  VET1(3,NO+I+I-1) = 1.D0/AUXFR(5) - 1.D0/(AUXFR(1)+AUXFR(5))
20 CONTINUE
LK1(1,1)      = NO+1
LK1(2,1)      = NO+3
LK1(1,NO)     = NO+2
LK1(2,NO)     = NO+4
LK1(1,NFR-NO+1) = NFR-NO-1
LK1(2,NFR-NO+1) = NFR-NO-3
LK1(1,NFR)    = NFR-NO
LK1(2,NFR)    = NFR-NO-2
DO 30 I=1,NO-2
  LK1(1,I+1)   = I
  LK1(2,I+1)   = I+NO-2
  LK1(1,I+NFR-NO+1) = I+NNO-NO+2
  LK1(2,I+NFR-NO+1) = I+NNO-NO-NO+4
30 CONTINUE
DO 40 I=1,MO-2
  LK1(1,NO+I+I) = I*(NO-2)
  LK1(2,NO+I+I) = I*(NO-2)-1
  LK1(1,NO+I+I-1) = (I-1)*(NO-2)+1
  LK1(2,NO+I+I-1) = (I-1)*(NO-2)+2
40 CONTINUE
DO 50 I=1,NNO+4
  LK2(I)=0
VET2(I)=0.D0

```



```

50      CONTINUE
      DO 60 I=1,NO-2
          LK2 ( I ) = I+1
          VET2(I) = 1.D0/AUXFR(3)/(AUXFR(3)+AUXFR(7))
          LK2 ( I+NNO-NO+2 ) = I+(NFR-NO+1)
          VET2(I+NNO-NO+2) = 1.D0/AUXFR(4)/(AUXFR(4)+AUXFR(8))
60      CONTINUE
      DO 70 I=2,MO-3
          LK2 ( (I-1)*(NO-2)+1 ) = NO+I+I-1
          VET2( (I-1)*(NO-2)+1 ) = 1.D0/AUXFR(1)/(AUXFR(1)+AUXFR(5))
          LK2 ( I*(NO-2) ) = NO+I+I
          VET2( I*(NO-2) ) = 1.D0/AUXFR(2)/(AUXFR(2)+AUXFR(6))
70      CONTINUE
          LK2 (NNO+1) = NO+1
          VET2(NNO+1) = VET2(1)
          LK2 (NNO+2) = NO+2
          VET2(NNO+2) = VET2(1)
          LK2 (NNO+3) = NFR-NO-1
          VET2(NNO+3) = VET2(NNO-NO+3)
          LK2 (NNO+4) = NFR-NO
          VET2(NNO+4) = VET2(NNO-NO+3)
      RETURN
      END

```

```

=====
SUBROUTINE DIFER (XCOND, YCOND, RAI, POT, RAI0, EPS, KOD, XNO, YNO,
1          AUXFR, ILIN, JCOL, ESP, B2, X2, Y2, NELEM)
-----

```

OBJETIVO: DETERMINA OS ELEMENTOS DA MATRIZ DE DIFERENCAS FINITAS

```

PARAMETER (NOS=121, NB=50)

```

```

1 REAL *8 XCOND(*), YCOND(*), RAI(*), POT(*), RAI0(*), EPS(*),
2 XNO(*), YNO(*), B2(*), AUXFR(*), ESP(*), X2(*), Y2(*),
HN, HS, HW, HE, COEF(2, 3)

```

```

1 INTEGER *4 ILIN(*), JCOL(*), KOD(*), NELEM, P(2, NB), BORDAX(NOS),
BORDAY(NOS), K, NUM(9), VIZ(2, 3), COD

```

```

INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2, PTS

```

```

INTEGER *4 LIN, LOUT1, LOUT2, LOUT3

```

```

COMMON /IMPRES/LIN, LOUT1, LOUT2, LOUT3

```

```

COMMON /DIMENS/NCSM, NFD, NCG, NNO, NFR, NO, MO, M2, PTS

```

```

IF(NNO .GT. NOS) CALL FALHA (12, NNO)

```

```

K=0

```

```

M2=0

```

```

DO 20 I=1, NOS

```

```

    BORDAX(I) = 0

```

```

    BORDAY(I) = 0

```

```

CONTINUE

```

```

CALL DEFKOD (NB, XCOND, YCOND, RAI, RAI0, XNO, YNO, KOD)

```

DETERMINACAO DAS EQUACOES COM CONDICAO DE DIRICHLET

```

DO 30 I=1, MO-2

```

```

    I0=(I-1)*(NO-2)

```

```

    DO 30 I1=1, NO-2

```

```

        I2=I0+I1

```

```

        IF(KOD(I2) .GT. 0) GO TO 40

```

```

        HN=AUXFR(3)

```

```

        HS=AUXFR(4)

```

```

        HE=AUXFR(2)

```

```

        HW=AUXFR(1)

```

```

        IN=I2-NO+2

```

```

        IS=I2+NO-2

```

```

        IE=I2+1

```

```

        IW=I2-1

```

```

        CALL PASSO ( I, 1, XCOND, YCOND, XNO, YNO, X2, Y2, KOD,
1          RAI0, RAI, -1.00, IN, I2, 0, BORDAY, P, HN)

```

```

        CALL PASSO ( I, MO-2, XCOND, YCOND, XNO, YNO, X2, Y2, KOD,
1          RAI0, RAI, 1.00, IS, I2, 2, BORDAY, P, HS)

```

```

        CALL PASSO ( I1, 1, YCOND, XCOND, YNO, XNO, X2, Y2, KOD,
1          RAI0, RAI, 1.00, IW, I2, 1, BORDAX, P, HW)

```

```

        CALL PASSO ( I1, NO-2, YCOND, XCOND, YNO, XNO, X2, Y2, KOD,
1          RAI0, RAI, -1.00, IE, I2, 3, BORDAX, P, HE)

```

```

CALL DIRICH ( I, 1, KOD, IN, I2, B2, POT, HN, HS, ILIN, JCOL, ESP, K)
CALL DIRICH ( I, MO-2, KOD, IS, I2, B2, POT, HS, HN, ILIN, JCOL, ESP, K)
CALL DIRICH ( I1, 1, KOD, IW, I2, B2, POT, HW, HE, ILIN, JCOL, ESP, K)
CALL DIRICH ( I1, NO-2, KOD, IE, I2, B2, POT, HE, HW, ILIN, JCOL, ESP, K)
      CALL DIAG ( I2, HN, HS, HE, HW, ILIN, JCOL, ESP, K)
      GO TO 30
40      CALL DIAG ( I2, 2.D0, -1.D0, 2.D0, -1.D0, ILIN, JCOL, ESP, K)
      B2 ( I2) = POT ( KOD ( I2 ) )
30      CONTINUE
C
C      DETERMINACAO DAS EQUACOES COM CONDICOES DE NEUMANN
C
      IF ( M2 .GT. NB ) CALL FALHA ( 13, M2 )
      DO 50 I=1, M2
      COD=ABS ( KOD ( P ( 2, I ) ) )
      DO 60 M=1, 2
      L=0
      CALL VIZIN ( I, P, KOD, M, NUM, L )
      IF ( PTS .EQ. 2 ) CALL COEF12 ( I, M, NUM, L, COD, XCOND, YCOND,
1      XNO, YNO, X2, Y2, VIZ, COEF )
      IF ( PTS .EQ. 3 ) CALL COEF13 ( I, M, NUM, L, COD, XCOND, YCOND,
1      XNO, YNO, RAO, X2, Y2, VIZ, COEF )
60      CONTINUE
      CALL NEUMAN ( I, COD, COEF, VIZ, EPS, ILIN, JCOL, ESP, K )
50      CONTINUE
      NELEM=K
      RETURN
END

```

```
C
C
C -----
C SUBROUTINE DEFKOD (IDIM,XCOND,YCOND,RAI,RAIO,XNO, YNO,KOD)
C -----
C
C
C
C
C
C
C
C
```

```
OBJETIVO:  DEFINE CODIGO DE POSICAO DE CADA NO DA
            MALHA DE DIFERENCAS FINITAS
```

```
1  REAL      *8 XCOND(*),YCOND(*),RAI(*),RAIO(*),
      XNO(*),YNO(*),DIST,AUXA,AUXB
   INTEGER *4 KOD(*),J
   INTEGER *4 NCSM,NFDM, NCG,NNO,NFR,NO,MO,M2,PTS
   COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
      DO 10 I=1,NNO
         KOD(I)=0
         DO 20 J=1,NFDM
            AUXA=XNO(I)-XCOND(J)
            AUXB=YNO(I)-YCOND(J)
            DIST = DSQRT(AUXA*AUXA+AUXB*AUXB) - 1.D-3*RAI(J)
            IF(DIST .LE. RAI(J)) THEN
                                   KOD(I)=J
            ELSEIF(DIST .LE. RAIO(J)) THEN
                                   KOD(I)=-J
            ELSE
                                   GO TO 20
         ENDIF
      CONTINUE
20  CONTINUE
10  DO 30 I=NNO+1,NNO+IDIM
      KOD(I) = 0
30  CONTINUE
   RETURN
END
```

```

C
C =====
C
  SUBROUTINE PASSO (I,J,XCOND,YCOND,XNO,YNO,X2,Y2,KOD,
1      RAI,RAI,AUX,I1,I2,IAUX,BORD,P,H)
C
C -----
C
C OBJETIVO: DETERMINA O PASSO PH, QH, RH OU SH
C
C
  REAL      *8 XCOND(*),YCOND(*),XNO(*),YNO(*),X2(*),Y2(*),
1      RAI(*),RAI(*),H,AUX
  INTEGER *4 KOD(*),BORD(*),P(2,*)
  INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
  COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
  IF(I .EQ. J) RETURN
  H = DABS(YNO(I2)-YNO(I1))
  IF(KOD(I1) .EQ. KOD(I2)) RETURN
  IF(KOD(I1)) 10,20,30
10      CALL CORREC (XCOND,YCOND,XNO,YNO,I2,
1          ABS(KOD(I1)),RAIO,AUX,H)
      GO TO 40
20      CALL CORREC (XCOND,YCOND,XNO,YNO,I2,
1          ABS(KOD(I2)),RAIO,-AUX,H)
      GO TO 40
30      CALL CORREC (XCOND,YCOND,XNO,YNO,I2,
1          KOD(I1),RAI ,AUX,H)
40      IF(H .LT. (1.D-03)) CALL FALHA (15,I2)
      IF(IAUX.LE.1 .AND. KOD(I1).LE.0) I1 = NNO+BORD(I1)
      IF(IAUX.EQ.2 .AND. KOD(I1).LE.0) THEN
1          CALL BORDA (H,XNO,YNO,KOD,
              X2,Y2,I1,I2,0.D0,-1.D0,P,BORD)
      ENDIF
      IF(IAUX.EQ.3 .AND. KOD(I1).LE.0) THEN
1          CALL BORDA (H,YNO,XNO,KOD,
              X2,Y2,I1,I2,1.D0, 0.D0,P,BORD)
      ENDIF
      RETURN
  END
END

```



```

C
C =====
C
C SUBROUTINE DIRICH (I,J,KOD,I1,I2,B2,POT,H1,H2,ILIN,JCOL,ESP,K)

```

```

C -----
C
C OBJETIVO: DEFINE UM ELEMENTO DA EQUACAO DE DIFERENCAS FINITAS
C COM CONDICAOES DE CONTORNO DE DIRICHLET
C

```

```

C
C REAL *8 B2(*),POT(*),ESP(*),H1,H2
C INTEGER *4 KOD(*),ILIN(*),JCOL(*)
C IF(I .EQ. J) RETURN
C IF(KOD(I1) .GT. 0) THEN
C   B2(I2) = B2(I2) - POT(KOD(I1))/H1/(H1+H2)
C   ELSE
C     K=K+1
C     ILIN(K)=I2
C     JCOL(K)=I1
C     ESP(K)=1.D0/H1/(H1+H2)
C   ENDIF
C RETURN
C END

```

```

C
C =====
C
C SUBROUTINE DIAG (I2,HN,HS,HE,HW,ILIN,JCOL,ESP,K)

```

```

C -----
C
C OBJETIVO: DEFINE UM ELEMENTO DA DIAGONAL DO SISTEMA DE EQUACOES
C

```

```

C
C REAL *8 ESP(*),HN,HS,HE,HW
C INTEGER *4 ILIN(*),JCOL(*),K
C K=K+1
C ILIN(K)=I2
C JCOL(K)=I2
C ESP(K) = -1.D0/HN/HS - 1.D0/HE/HW
C RETURN
C END

```

```
C
C
C=====
C
C SUBROUTINE VIZIN (I,P,KOD,M,NUM,L)
C-----
C
C
C OBJETIVO: SELECIONA OS NOS VIZINHOS POSSIVEIS, EM CADA NOH SOBRE
C A CURVA DE DESCONTINUIDADE DA PERMISSIVIDADE ELETRICA
C
C
C INTEGER *4 P(2,*),KOD(*),NUM(*),IAUX
C INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
C COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
C DO 10 J=1,3
C DO 20 K=1,3
C IAUX = P(M,I) + (NO-2)*(J-2) + K-2
C IF(KOD(IAUX) .EQ. KOD(P(M,I))) THEN
C
C L=L+1
C NUM(L)=IAUX
C
C ENDIF
C CONTINUE
20 CONTINUE
10 IF(L .LT. PTS) CALL FALHA (16,I)
RETURN
END
```



```

C
C =====
C
  SUBROUTINE COEF13 (I,M,NUM,L,COD,XCOND,YCOND,
1      XNO,YNO,RAIO,X2,Y2,VIZ,COEF)
C
C -----
C
C OBJETIVO: ESCOLHE 3 NOS EM CADA LADO DA MALHA, PARA
C            COMPOR AS EQUACOES COM CONDICAO DE CONTORNO
C            DE NEUMANN (OPCAO COM 3 NOS)
C
C
  REAL      *8 XCOND(*),YCOND(*),XNO(*),YNO(*),RAIO(*),
1      X2(*),Y2(*),COEF(2,*),ABC(3),ORD(3)
  INTEGER *4 NUM(*),VIZ(2,*),COD
  INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
  COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
  DO 10 I1=1,L-2
    CALL COORD (M,NUM(I1),I,COD,XCOND,YCOND,
1      XNO,YNO,X2,Y2,ABC(1),ORD(1))
    DO 20 I2=I1+1,L-1
      CALL COORD (M,NUM(I2),I,COD,XCOND,YCOND,
1      XNO,YNO,X2,Y2,ABC(2),ORD(2))
      DO 30 I3=I2+1,L
        CALL COORD (M,NUM(I3),I,COD,XCOND,YCOND,
1      XNO,YNO,X2,Y2,ABC(3),ORD(3))
        CALL KRAM3 (M,ABC,ORD,RAIO(COD),COEF)
        DO 40 I4=1,3
          IF(COEF(M,I4) .LT. (0.D0)) GO TO 30
40      CONTINUE
          L=0
          VIZ(M,1)=NUM(I1)
          VIZ(M,2)=NUM(I2)
          VIZ(M,3)=NUM(I3)
          GO TO 50
30      CONTINUE
20      CONTINUE
10      CONTINUE
50      IF(L .NE. 0) THEN
          WRITE(LOUT2,*)
          WRITE(LOUT2,*) ' COEFICIENTE "A" < ZERO NO PONTO ',I
          WRITE(LOUT2,*) ' CALCULO PROSEGUE COM OS '
          WRITE(LOUT2,*) ' ULTIMOS PONTOS CALCULADOS ... '
          WRITE(LOUT2,*)
        ENDIF
      RETURN
    END
  END

```



```

C
C =====
C
SUBROUTINE COORD (M,NUM,I,COD,XCOND,YCOND,XNO,YNO,X2,Y2,ABC,ORD)
C -----
C
C OBJETIVO: DEFINE AS COORDENADAS DE UM NO INTERNO AA MALHA, PARA
C COMPOR AS EQUACOES DA DERIVADA NORMAL
C
C
      REAL      *8 XCOND(*),YCOND(*),XNO(*),YNO(*),X2(*),Y2(*),
1      ABC,ORD,A,B,C,D,E,F
      INTEGER *4 NUM,COD
      C = XCOND(COD) - X2(I)
      D = XNO(NUM) - X2(I)
      E = YCOND(COD) - Y2(I)
      F = YNO(NUM) - Y2(I)
      A = C*D + E*F
      B = D*E - C*F
      C = DSQRT (A*A + B*B)
      D = DSQRT (D*D + F*F)
      ABC = D*B/C
      ORD = -D*A/C
      IF(M .EQ. 2) ORD = DABS(ORD)
      RETURN
END

```



```

C
C =====
C
SUBROUTINE NEUMAN (I,COD,COEF,VIZ,EPS,ILIN,JCOL,ESP,K)
C -----
C
C OBJETIVO: DETERMINA OS ELEMENTOS QUE COMPOE AS EQUACOES
C DE CONTINUIDADE DA DERIVADA NORMAL
C
C
REAL *8 COEF(2,*),EPS(*),ESP(*)
INTEGER *4 VIZ(2,*),ILIN(*),JCOL(*),COD,K
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2,PTS
K = K+1
ILIN(K) = NNO+I
JCOL(K) = NNO+I
ESP(K) = 0.D0
DO 10 J=1,PTS
    ESP(K) = ESP(K) + COEF(1,J)
    ESP(K) = ESP(K) + EPS(COD)*COEF(2,J)
10 CONTINUE
DO 20 J=1,PTS
    K = K+1
    ILIN(K) = NNO+I
    JCOL(K) = VIZ(1,J)
    ESP(K) = -COEF(1,J)
20 CONTINUE
DO 30 J=1,PTS
    K = K+1
    ILIN(K) = NNO + I
    JCOL(K) = VIZ(2,J)
    ESP(K) = -EPS(COD)*COEF(2,J)
30 CONTINUE
RETURN
END

```

```

=====
SUBROUTINE ATUB1 (FIO,FII,POT1,VET1,LK1,WK1)
-----

```

```

OBJETIVO: ATUALIZA O VETOR DO LADO DIREITO DAS EQUACOES
           DE CONTINUIDADE DO CAMPO ELETRICO

```

```

REAL      *8 FIO(*),FII(*),POT1(*),VET1(3,*),WK1(*)
INTEGER   *4 LK1(2,*)
INTEGER   *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
DO 10 I=1,NCSM
  WK1(I) = POT1(I)
CONTINUE
DO 20 I=2,NFR-1
  IF(I .EQ. NO .OR. I .EQ. (NFR-NO+1)) GO TO 20
  WK1(NCSM+I) = - ( VET1(1,I)*FIO(I) +
1                   VET1(2,I)*FII(LK1(1,I)) +
2                   VET1(3,I)*FII(LK1(2,I)) )
CONTINUE
WK1(NCSM + 1) = - ( VET1(1,1)*FIO(1) +
1                   VET1(2,1)*FIO(LK1(1,1)) +
2                   VET1(3,1)*FIO(LK1(2,1)) )
WK1(NCSM + NO) = - ( VET1(1,NO)*FIO(NO) +
1                    VET1(2,NO)*FIO(LK1(1,NO)) +
2                    VET1(3,NO)*FIO(LK1(2,NO)) )
I=NFR-NO+1
WK1(NCSM + I) = - ( VET1(1,I)*FIO(I) +
1                   VET1(2,I)*FIO(LK1(1,I)) +
2                   VET1(3,I)*FIO(LK1(2,I)) )
WK1(NCSM + NFR) = - ( VET1(1,NFR)*FIO(NFR) +
1                     VET1(2,NFR)*FIO(LK1(1,NFR)) +
2                     VET1(3,NFR)*FIO(LK1(2,NFR)) )
RETURN
END

```

```

C
C =====
C
C      SUBROUTINE ATUB2 (B2,FI0,VET2,LK2,WK2)
C -----
C
C OBJETIVO:  ATUALIZA O VETOR DO LADO DIREITO DAS
C            EQUACOES DE DIFERENCAS FINITAS
C
C
C      REAL      *8 B2(*),FI0(*),WK2(*),VET2(*)
C      INTEGER   *4 LK2(*)
C      INTEGER   *4 NCSM,NFDM, NCG,NNO,NFR,NO,MO,M2
C      COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C      DO 10 I=1,NNO
C          WK2(I)=B2(I)
C          IF(LK2(I) .EQ. 0) GO TO 10
C          WK2(I)=WK2(I)-VET2(I)*FI0(LK2(I))
10      CONTINUE
C          WK2(      1) = WK2(      1) - VET2(NNO+1)*FI0(LK2(NNO+1))
C          WK2(     NO-2) = WK2(     NO-2) - VET2(NNO+2)*FI0(LK2(NNO+2))
C          WK2(NNO-NO+3) = WK2(NNO-NO+3) - VET2(NNO+3)*FI0(LK2(NNO+3))
C          WK2(      NNO) = WK2(      NNO) - VET2(NNO+4)*FI0(LK2(NNO+4))
C      RETURN
C      END
C
C =====
C
C      SUBROUTINE ATUPOT (CHEIA1,RO,IDIM1,FI0)
C -----
C
C OBJETIVO:  ATUALIZA O VETOR DE POTENCIAIS NA FRONTEIRA COMUM
C
C
C      REAL      *8 CHEIA1(IDIM1,*),RO(*),FI0(*)
C      INTEGER   *4 NCSM,NFDM, NCG,NNO,NFR,NO,MO,M2
C      COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C      DO 10 I=1,NFR
C          FI0(I) = 0.D0
C          DO 10 J=1,NCSM+NCG
C              FI0(I)=FI0(I)+CHEIA1(I,J)*RO(J)
10      CONTINUE
C      RETURN
C      END

```

```

C
C =====
C
SUBROUTINE TESTE (RO,NI,TETA,DIF)
C
C -----
C
C OBJETIVO: COMPARA OS VALORES DAS CARGAS NA FRONTEIRA E ATUALIZA
C
C
REAL *8 RO(*),NI(*),TETA(*),DIF,AUX
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
DIF=DABS(RO(1)-NI(1))
DIF=DABS((RO(1)-NI(1))/RO(1))*1.D2
DO 10 I=2,NCSM+NCG
AUX = DABS(RO(I)-NI(I))
AUX = DABS((RO(I)-NI(I))/RO(I))*1.D2
IF(AUX .GT. DIF) DIF=AUX
CONTINUE
DO 20 I=1,NCSM+NCG
RO(I)=(RO(I)*TETA(I) + NI(I)*(1.D0-TETA(I)))
CONTINUE
RETURN
END
C
C =====
C

```

```

SUBROUTINE ITERAT (ITER,KONT,DIF)
C
C -----
C
C OBJETIVO: IMPRESSAO DE RESULTADOS PARCIAIS
C
C
REAL *8 DIF
INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
PORC = FLOAT(ITER)/FLOAT(NNO+M2)*100.
WRITE(*,40) KONT,DIF,PORC
WRITE(LOUT2,40) KONT,DIF,PORC
40 FORMAT(' DIFERENCA APOS ',I3,' ITERACOES: ',F8.2,
1 ' % ; CONVERGENCIA: ',F6.2,' %')
RETURN
END

```



```

C
C =====
C
SUBROUTINE CONDOT (FIC,XNO,YNO,XCOND,YCOND,RAI,POT)
C
C -----
C
C OBJETIVO:  DEFINE NOVOS NOS PARA PLOTAGEM,
C            NA SUPERFICIE DOS CONDUTORES
C
C
1   REAL      *8 FIC(*),XNO(*),YNO(*),XCOND(*),YCOND(*),
      RAI(*),POT(*),ANG
   INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
   COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
      ANG=3.1415926D0/4.D0 + 0.15
      DO 60 I=1,NFDM
        DO 70 J=1,8
          XNO(NNO+(I-1)*8+J)=XCOND(I)+RAI(I)*DCOS(DFLOAT(J-1)*ANG)
          YNO(NNO+(I-1)*8+J)=YCOND(I)+RAI(I)*DSIN(DFLOAT(J-1)*ANG)
          FIC((I-1)*8+J)=POT(I)
70      CONTINUE
60      CONTINUE
      RETURN
      END

```

```

C
C =====
C
SUBROUTINE FALHA (KOD,NUM)
C
C -----
C
C OBJETIVO:  IMPRIME MENSAGENS DE ERRO
C
C
      INTEGER *4 NUM
      INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
      COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
      GO TO (01,02,03,04,05,06,07,08,09,10,
1      11,12,13,14,15,16,17,18 ) KOD
01      WRITE(LOUT2,51) NUM
51      FORMAT(/,T5,'N.O DE CARGAS NA REGIAO MSC (' ,I4,')',
1      /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2      /,T5,'A VARIABEL NCS NO PROGRAMA PRINCIPAL')
      GO TO 99
02      WRITE(LOUT2,52) NUM
52      FORMAT(/,T5,'N.O DE FONTES NA REGIAO MDF (' ,I4,')',
1      /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2      /,T5,'A VARIABEL NFD NO PROGRAMA PRINCIPAL')
      GO TO 99

```

```

03      WRITE(LOUT2,53) NUM
53      FORMAT(//,T5,'N.O DE CARGAS NA REGIAO MDF (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2         /,T5,'A VARIABEL  NC  NO PROGRAMA PRINCIPAL')
      GO TO 99
04      WRITE(LOUT2,54) NUM
54      FORMAT(//,T5,'N.O DE PONTOS NA FRONTEIRA COMUM (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2         /,T5,'A VARIABEL  NP  NO PROGRAMA PRINCIPAL')
      GO TO 99
05      WRITE(LOUT2,55) NUM
55      FORMAT(//,T5,'N.O DE NOS DA MALHA MDF (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2         /,T5,'A VARIABEL  NOS  NO PROGRAMA PRINCIPAL')
      GO TO 99
06      WRITE(LOUT2,56) NUM
56      FORMAT(//,T5,'N.O DE NOS COM DESCONTINUIDADE EM EPS (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2         /,T5,'A VARIABEL  NB  NO PROGRAMA PRINCIPAL')
      GO TO 99
07      WRITE(LOUT2,57) NUM
57      FORMAT(//,T5,'N.O DE ITERACOES NO PROGRAMA PRINCIPAL (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR O VALOR',
2         /,T5,'MAXIMO NOS DADOS DE ENTRADA')
      GO TO 99
08      WRITE(LOUT2,58) NUM
58      FORMAT(//,T5,'DIMENSAO DADA PARA OS SUB-ESPAÇOS DE',
1         /,T5,'KRYLOV (',I4,') EXCEDE O PERMISSIVEL.',
2         /,T5,'REDIMENSIONAR VARIABEL  MSPACE  NA',
3         /,T5,'SUBROTINA KRYLOV')
      GO TO 99
09      WRITE(LOUT2,59) NUM
59      FORMAT(//,T5,'DIMENSAO DO SISTEMA LINEAR ESPARSO (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR VARIABEL',
2         /,T5,'MLIN  NA SUBROTINA KRYLOV OU CG')
      GO TO 99
10      WRITE(LOUT2,60)
60      FORMAT(//,T5,'MATRIZ SINGULAR (SUBROTINA DECOMP)',
1         /,T5,'VERIFICAR DADOS DE ENTRADA')
      GO TO 99
11      WRITE(LOUT2,61)
61      FORMAT(//,T5,'MATRIZ SINGULAR (SUBROTINA TRIDLU)',
1         /,T5,'VERIFICAR DADOS DE ENTRADA')
      GO TO 99
12      WRITE(LOUT2,62) NUM
62      FORMAT(//,T5,'N.O DE NOS DA MALHA MDF (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR',
2         /,T5,'A VARIABEL  NOS  NA SUBROTINA DIFER')
      GO TO 99

```

```

13      WRITE(LOUT2,63) NUM
63      FORMAT(//,T5,'N.O DE NOS COM DESCONTINUIDADE EM EPS (',I4,')',
1         /,T5,'EXCEDE O PERMISSIVEL. REDIMENSIONAR A',
2         /,T5,'VARIAVEL NB NA SUBROTINA. DIFER')
      GO TO 99
14      WRITE(LOUT2,64)
64      FORMAT(//,T5,'N.O DE LINHAS HORIZONTAIS OU VERTICAIS NA',
1         /,T5,'MALHA DE DIFERENCAS FINITAS EXCEDE',
2         /,T5,'O PERMISSIVEL. REDIMENSIONAR VARIAVEL',
3         /,T5,'MAX NA SUBROTINA DADOS')
      GO TO 99
15      WRITE(LOUT2,65) NUM
65      FORMAT(//,T5,'BORDA DE DESCONTINUIDADE EM EPS COINCIDE',
1         /,T5,'COM O NO NUMERO ',I4,'. REDEFINIR MALHA NAO',
2         /,T5,'COINCIDENTE MODIFICANDO DADOS DE ENTRADA')
      GO TO 99
16      WRITE(LOUT2,66) NUM
66      FORMAT(//,T5,'NAO HA NUMERO DE PONTOS SUFICIENTES PARA',
1         /,T5,'DETERMINAR A DERIVADA NORMAL NO NOH NUMERO',
2         /,T5,I4,' DA BORDA DE DESCONTINUIDADE DE EPS',
3         /,T5,'EH NECESSARIO REFINAMENTO DA MALHA')
      GO TO 99
17      WRITE(LOUT2,67)
67      FORMAT(//,T5,'MATRIZ COM POSTO DEFICIENTE',
1         /,T5,'(SUBROTINA CGSOR OU MGSOR)',
2         /,T5,'CAUSA PROVAVEL: CARGAS COINCIDENTES',
3         /,T5,'OU MUITO PROXIMAS')
      GO TO 99
18      WRITE(LOUT2,68) NUM
68      FORMAT(//,T5,'NUMERO DE ITERACOES NA SUBROTINA CG, KRYLOV',
1         /,T5,'OU CGS (',I4,') SUPEROU AO MAXIMO ESPECIFICADO',
2         /,T5,'NOS DADOS DE ENTRADA. REDEFINIR VALOR MAXIMO.')
99      WRITE(*,*)' ERRO CODIGO N.O ',KOD
      WRITE(*,*)' EXECUCAO CANCELADA ... '
      STOP
      END

```


END

C
C
C
C
C
C
C
C
C
C

SUBROUTINE IMPR2 (XFRONT,YFRONT,XNO,YNO,X2,Y2,FI0,FI1,FIC)

OBJETIVO: IMPRIME RESULTADOS PARA PLOTAGEM DAS CURVAS DE NIVEL

```
1  REAL      *8 XFRONT(*),YFRONT(*),XNO(*),YNO(*),
      X2(*),Y2(*),FI0(*),FI1(*),FIC(*)
  INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
  INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
  COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
  COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
      DO 10 I=1,NFR
10      WRITE(LOUT3,*) XFRONT(I),YFRONT(I),FI0(I)
      CONTINUE
      DO 20 I=1,NNO
20      WRITE(LOUT3,*) XNO(I),YNO(I),FI1(I)
      CONTINUE
      DO 30 I=1,M2
30      WRITE(LOUT3,*) X2(I),Y2(I),FI1(NNO+I)
      CONTINUE
      J=NFDM*8
      DO 40 I=1,J
40      WRITE(LOUT3,*) XNO(NNO+I),YNO(NNO+I),FIC(I)
      CONTINUE
  RETURN
END
```

```

=====
SUBROUTINE IMPR3 (XNO, YNO, XFRONT, YFRONT, XCAR, YCAR,
1 XFR, YFR, XCG, YCG)
=====

```

OBJETIVO: IMPRIME RESULTADOS PARCIAIS

```

1 REAL *8 XNO(*), YNO(*), XFRONT(*), YFRONT(*), XCAR(*),
YCAR(*), XFR(*), YFR(*), XCG(*), YCG(*)
INTEGER *4 LIN, LOUT1, LOUT2, LOUT3
INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
COMMON /DIMENS/ NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
COMMON /IMPRES/ LIN, LOUT1, LOUT2, LOUT3
WRITE(LOUT2, *)
WRITE(LOUT2, *) '--- NOS DA MALHA (X,Y) ---'
WRITE(LOUT2, 10) (I, XNO(I), YNO(I), I=1, NNO)
WRITE(LOUT2, *)
WRITE(LOUT2, *) '---- PONTOS NA FRONTEIRA COMUM (X,Y) ----'
WRITE(LOUT2, 10) (I, XFRONT(I), YFRONT(I), I=1, NFR)
WRITE(LOUT2, *)
WRITE(LOUT2, *) '---- CARGAS NA FRONTEIRA COMUM (X,Y) ----'
WRITE(LOUT2, 10) (I, XCAR(I), YCAR(I), I=1, NCG)
WRITE(LOUT2, *)
WRITE(LOUT2, *) '---- PONTOS DE FRONTEIRA',
1 ' NA REGIAO CSM (X,Y) ----'
WRITE(LOUT2, 10) (I, XFR(I), YFR(I), I=1, NCSM)
WRITE(LOUT2, *)
WRITE(LOUT2, *) '---- CARGAS NA REGIAO CSM (X,Y) ----'
WRITE(LOUT2, 10) (I, XCG(I), YCG(I), I=1, NCSM)
10 FORMAT(81(I3, 2X, F10.5, 2X, F10.5, /))
RETURN
END

```

```

=====
SUBROUTINE IMPR4 (CHEIA1, CHEIA2, IDIM1, DIM, ILIN, JCOL, ESP, VET1,
1 VET2, LK1, LK2, POT1, B2, K)
=====

```

OBJETIVO: IMPRIME RESULTADOS PARCIAIS

```

1 REAL *8 CHEIA1(IDIM1, *), CHEIA2(IDIM1, *), ESP(*),
VET1(3, *), VET2(*), POT1(*), B2(*), POT(6)
INTEGER *4 ILIN(*), JCOL(*), LK1(2, *), LK2(*), KOD(11), K, DIM
INTEGER *4 LIN, LOUT1, LOUT2, LOUT3
INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
COMMON /DIMENS/ NCSM, NFD, NCG, NNO, NFR, NO, MO, M2

```

```

COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
WRITE(LOUT2,*)
WRITE(LOUT2,*)'          ---  MATRIZ  P1 + P2  ---'
L=0
KOD(1)=1
DO 10 I=1,NCSM+NFR
  DO 10 J=1,NCSM+NCG
    L=L+1
    IF(L .EQ. 1) GO TO 20
    IF(KOD(L-1) .NE. I) THEN
      WRITE(LOUT2,30)(KOD(M),KOD(5+M),POT(M),M=1,L-1)
      L=1
    ENDIF
20    KOD(L)=I
    KOD(5+L)=J
    POT(L)=CHEIA1(I,J)
    IF(L .LT. 5 .AND. I .NE. (NCSM+NFR)
1      .AND. J.NE.(NCSM+NCG) ) GO TO 10
    WRITE(LOUT2,30)(KOD(M),KOD(5+M),POT(M),M=1,L)
30    FORMAT(5(1X,2I3,'=',F8.4))
    L=0
10  CONTINUE
WRITE(LOUT2,*)
WRITE(LOUT2,*)'          ---  MATRIZ  P2 + F  ---'
L=0
KOD(1)=1
DO 110 I=1,NCSM+NFR
  DO 110 J=1,NCSM+NCG
    L=L+1
    IF(L .EQ. 1) GO TO 120
    IF(KOD(L-1) .NE. I) THEN
      WRITE(LOUT2,130)(KOD(M),KOD(5+M),POT(M),M=1,L-1)
      L=1
    ENDIF
120   KOD(L)=I
    KOD(5+L)=J
    POT(L)=CHEIA2(I,J)
    IF(L .LT. 5 .AND. I .NE. (NCSM+NFR)
1      .AND. J.NE.(NCSM+NCG) ) GO TO 110
    WRITE(LOUT2,130)(KOD(M),KOD(5+M),POT(M),M=1,L)
130   FORMAT(5(1X,2I3,'=',F8.4))
    L=0
110  CONTINUE
C
WRITE(LOUT2,*)
WRITE(LOUT2,*)'          ---  MATRIZ  D  ---'
L=0
KOD(1)=0
DO 210 I=1,K
  L=L+1
    IF(L .EQ. 1) GO TO 220
    IF(KOD(L-1) .NE. ILIN(I)) THEN
      WRITE(LOUT2,230)(KOD(M),KOD(5+M),POT(M),M=1,L-1)
      L=1
    ENDIF
220
230

```

```

                ENDIF
220    KOD(L)=ILIN(I)
        KOD(5+L)=JCOL(I)
        POT(L)=ESP(I)
        IF(L .LT. 5 .AND. I .NE. K) GO TO 210
        WRITE(LOUT2,230)(KOD(M),KOD(5+M),POT(M),M=1,L)
230    FORMAT(5(1X,2I3,'=',F8.4))
        L=0
C
210    CONTINUE
        WRITE(LOUT2,*)
        WRITE(LOUT2,*)'--- VETOR B1 ---'
        DO 310 I=1,NCSM
310    WRITE(LOUT2,320)I,POT1(I)
320    FORMAT(1X,'B1(',I2,') = ',F10.4)
C
        WRITE(LOUT2,*)
        WRITE(LOUT2,*)'---- ELEMENTOS NAO NULOS DO VETOR B2 ----'
        L=0
        DO 410 I=1,DIM
            IF(DABS(B2(I)) .LT. (1.D-04) .AND. I .NE. DIM) GO TO 410
            L=L+1
            KOD(L)=I
            POT(L)=B2(I)
            IF(L .LT. 5 .AND. I .NE. DIM) GO TO 410
            WRITE(LOUT2,420)(KOD(M),POT(M),M=1,L)
420    FORMAT(5(1X,I3,'=',F10.4))
            L=0
410    CONTINUE
C
        WRITE(LOUT2,*)
        WRITE(LOUT2,*)'          --- VETORES DE CONEXAO ---'
        WRITE(LOUT2,*)
        WRITE(LOUT2,*)'      I  VET1(1,I)  VET1(2,I)  VET1(3,I)  ',
1      'LK1(1,I)  LK1(2,I)  VET2(I)  LK2(I) '
        WRITE(LOUT2,*)
        DO 180 I=1,NNO+4
            IF(I .LE. NFR) THEN
                WRITE(LOUT2,190)I,(VET1(J,I),J=1,3),
1      (LK1(J,I),J=1,2),VET2(I),LK2(I)
190    FORMAT(1X,I4,3(3X,F8.5),5X,I3,8X,I3,4X,F8.5,3X,I3)
                ELSE
                WRITE(LOUT2,195)I,VET2(I),LK2(I)
195    FORMAT(1X,I4,55X,F9.5,3X,I3)
            ENDIF
180    CONTINUE
        RETURN
        END

```



```

C
C =====
C
C      SUBROUTINE GRAD (ILIN,JCOL,A,NELEM,NDIM,WORK,X,B,R)
C
C -----
C
C
C
C
C      OBJETIVO:  CALCULO DO RESIDUO R = AT .(B-A.X)
C
C
C      REAL      *8 A(*),X(*),R(*),B(*),WORK(*)
C      INTEGER *4 ILIN(*),JCOL(*),NDIM,NELEM
C      CALL PRODAX (ILIN,JCOL,A,NELEM,NDIM,X,WORK)
C      DO 10 I=1,NDIM
C          WORK(I) = B(I) - WORK(I)
10      CONTINUE
C      CALL PATX (ILIN,JCOL,A,NELEM,NDIM,WORK,R)
C      RETURN
C      END
C
C =====
C
C      SUBROUTINE RESID (ILIN,JCOL,A,NELEM,NDIM,X,B,R)
C
C -----
C
C
C
C
C      OBJETIVO:  CALCULO DO RESIDUO R = B - A.X
C
C
C      REAL      *8 A(*),X(*),R(*),B(*)
C      INTEGER *4 ILIN(*),JCOL(*),NELEM,NDIM
C      CALL PRODAX (ILIN,JCOL,A,NELEM,NDIM,X,R)
C      DO 10 I = 1,NDIM
C          R(I) = B(I) - R(I)
10      CONTINUE
C      RETURN
C      END

```

```
C
C
C=====
C
C SUBROUTINE PINNER (X,Y,NDIM,YTX)
C-----
C
C
```

```
C
C OBJETIVO: CALCULO DO PRODUTO INTERNO  $\langle X, Y \rangle = Y^T \cdot X$ 
C
```

```
C
C REAL *8 X(*),Y(*),YTX
C INTEGER *4 NDIM
C YTX = 0.D0
C DO 10 I=1,NDIM
C YTX = YTX + X(I)*Y(I)
10 CONTINUE
C RETURN
C END
```

```
C
C=====
C
C SUBROUTINE PRODAX (ILIN,JCOL,A,NELEM,NDIM,X,Y)
C-----
C
C
```

```
C
C OBJETIVO: CALCULO DO PRODUTO  $Y = A \cdot X$ 
C
```

```
C
C REAL *8 A(*),X(*),Y(*)
C INTEGER *4 ILIN(*),JCOL(*),NDIM,NELEM
C DO 10 I=1,NDIM
C Y(I) = 0.D0
10 CONTINUE
C DO 20 K=1,NELEM
C Y(ILIN(K)) = Y(ILIN(K)) + A(K)*X(JCOL(K))
20 CONTINUE
C RETURN
C END
```

```

C
C =====
C
C      SUBROUTINE PATX (ILIN,JCOL,A,NELEM,NDIM,X,Y)
C
C -----
C
C
C
C
C OBJETIVO:  CALCULO DO PRODUTO  $Y = A^T X$ 
C
C
C      REAL      *8 A(*),X(*),Y(*)
C      INTEGER  *4 ILIN(*),JCOL(*),NDIM,NELEM
C      DO 10 I=1,NDIM
C          Y(I) = 0.D0
10      CONTINUE
C      DO 20 K=1,NELEM
C          Y(JCOL(K)) = Y(JCOL(K)) + A(K)*X(ILIN(K))
20      CONTINUE
C      RETURN
C      END
C
C =====
C
C      SUBROUTINE SUPNOR (X,NDIM,DIST)
C
C -----
C
C
C
C OBJETIVO:  CALCULO DA NORMA INFINITO DO VETOR X
C
C
C
C
C      REAL      *8 X(*),DIST,AUX
C      INTEGER  *4 NDIM
C      DIST = DABS(X(1))
C      DO 10 I=2,NDIM
C          AUX = DABS(X(I))
C          IF(DIST .LT. AUX) DIST = AUX
10      CONTINUE
C      RETURN
C      END

```



```

C
C =====
C
C      SUBROUTINE TRIDLU (DIAG,SUP,SUB,MDIM)
C -----
C
C OBJETIVO:  DECOMPOSICAO LU DE UMA MATRIZ TRIDIAGONAL,
C            SENDO L = MATRIZ BIDIAGONAL INFERIOR
C            U = MATRIZ BIDIAGONAL SUPERIOR COM
C            DIAGONAL PRINCIPAL UNITARIA
C
C
C      REAL      *8 DIAG(*),SUP(*),SUB(*),EPS
C      INTEGER *4 MDIM,I,K
C      DATA EPS / 1.D-20 /
C      IF(DABS(DIAG(1)) .LE. EPS) GO TO 20
C      SUP(1) = SUP(1)/DIAG(1)
C      DO 10 I=2,MDIM
C        K = I - 1
C        DIAG(I) = DIAG(I) - SUB(K)*SUP(K)
C        IF(DABS(DIAG(I)) .LE. EPS) GO TO 20
C        IF(I .NE. MDIM) SUP(I) = SUP(I)/DIAG(I)
10      CONTINUE
C      RETURN
20      CALL FALHA (11,0)
C      STOP
C      END
C
C =====
C
C      SUBROUTINE SOLVDG (DIAG,B,X,MDIM)
C -----
C
C OBJETIVO:  RESOLUCAO DE UM SISTEMA LINEAR D X = B,
C            SENDO D UMA MATRIZ DIAGONAL
C
C
C      REAL      *8 DIAG(*),B(*),X(*)
C      INTEGER *4 MDIM
C      DO 10 I=1,MDIM
C        X(I) = B(I)/DIAG(I)
10      CONTINUE
C      RETURN
C      END

```

```
C
C
C=====
C
C SUBROUTINE SOLVTD (DIAG,SUP,SUB,B,X,MDIM)
C-----
C
C
C
C
C OBJETIVO: RESOLUCAO DE UM SISTEMA LINEAR LU X = B,
C           SENDO AS MATRIZES L E U COMPUTADAS
C           CONFORME SUBROTINA TRIDLU
C
C
C           REAL *8 DIAG(*),SUP(*),SUB(*),B(*),X(*)
C           INTEGER *4 MDIM
C
C RESOLUCAO DO SISTEMA TRIANGULAR INFERIOR
C
C           X(1) = B(1)/DIAG(1)
C           DO 10 I=2,MDIM
C             K = I -1
C             X(I) = ( B(I) - SUB(K)*X(K) )/DIAG(I)
C           CONTINUE
C
C RESOLUCAO DO SISTEMA TRIANGULAR SUPERIOR
C
C           DO 20 I=(MDIM-1),1,-1
C             K = I + 1
C             X(I) = X(I) - SUP(I)*X(K)
C           CONTINUE
C           RETURN
C           END
C
C
C
```

```

C
C =====
C
C      SUBROUTINE SRCH (ILIN,JCOL,A,NELEM,INEW,JNEW,ANEW)
C -----
C
C OBJETIVO: LOCALIZACAO DE UM ELEMENTO NA MATRIZ A
C
C      REAL      *8 A(*),ANEW
C      INTEGER *4 ILIN(*),JCOL(*),NELEM,INEW,JNEW
C      DO 10 K = 1,NELEM
C          IF(INEW .EQ. ILIN(K) .AND. JNEW .EQ. JCOL(K)) THEN
C              ANEW = A(K)
C              RETURN
C          END IF
C          ANEW = 0.D0
10      CONTINUE
C      RETURN
C      END
C
C =====
C
C      SUBROUTINE FULLAX (A,X,Y,MLIN,NCOL,MAX)
C -----
C
C OBJETIVO: CALCULO DE  $Y = A.X$ , A M X N, COM  $M > N$ 
C
C      REAL      *8 A(MAX,*), X(*), Y(*), SOMA
C      INTEGER *4 NCOL,MLIN
C      DO 10 I = 1,MLIN
C          SOMA = 0.D0
C          DO 20 J = 1,NCOL
C              SOMA = SOMA+A(I,J)*X(J)
20      CONTINUE
C          Y(I) = SOMA
10      CONTINUE
C      RETURN
C      END

```

```
C
C =====
C SUBROUTINE TRIDIG (ILIN,JCOL,A,NELEM,MDIM,DIAG,SUP,SUB)
C -----
C
C
C OBJETIVO: DETERMINACAO DA PARTE TRIDIAGONAL DA MATRIZ A
C
C
C REAL *8 A(*),DIAG(*),SUB(*),SUP(*),ANEW
C INTEGER *4 ILIN(*),JCOL(*),NELEM,MDIM,INEW,JNEW
C DO 10 I = 1,MDIM
C     INEW = I
C     JNEW = I
C     CALL SRCH (ILIN,JCOL,A,NELEM,INEW,JNEW,ANEW)
C     DIAG(I) = ANEW
10 CONTINUE
C     DO 20 I = 1,(MDIM-1)
C         INEW = I
C         JNEW = I + 1
C         CALL SRCH (ILIN,JCOL,A,NELEM,INEW,JNEW,ANEW)
C         SUP(I) = ANEW
20 CONTINUE
C     DO 30 J = 1,(MDIM-1)
C         INEW = J + 1
C         JNEW = J
C         CALL SRCH (ILIN,JCOL,A,NELEM,INEW,JNEW,ANEW)
C         SUB(J) = ANEW
30 CONTINUE
C     RETURN
C END
```



```

C
C =====
C
C SUBROUTINE SOLVE (A,P,B,X,N,MAX)
C -----
C
C
C OBJETIVO: RESOLUCAO DE LU X = B
C
C
C REAL *8 A(MAX,*),B(*),X(*),SOMA
C INTEGER *4 N,P(*),IP
C
C RESOLUCAO DO SISTEMA TRIANGULAR INFERIOR
C
C DO 20 I = 1,N
C   IP = P(I)
C   SOMA = 0.D0
C   DO 10 J = 1,(I-1)
C     SOMA = SOMA + A(I,J)*X(J)
10  CONTINUE
C   X(I) = B(IP) - SOMA
20  CONTINUE
C
C RESOLUCAO DO SISTEMA TRIANGULAR SUPERIOR
C
C DO 40 I = N,1,-1
C   SOMA = 0.D0
C   DO 30 J = (I+1),N
C     SOMA = SOMA + A(I,J)*X(J)
30  CONTINUE
C   X(I) = ( X(I) - SOMA )/A(I,I)
40  CONTINUE
C   RETURN
C END

```

```
C
C
C =====
C
C SUBROUTINE DECOMP (A,P,N,MAX)
C
C -----
C
C OBJETIVO: DECOMPOSICAO LU DA MATRIZ A (METODO
C DE ELIMINACAO DE GAUSS COM PIVOTEAMENTO)
C
C
C REAL *8 A(MAX,*),PIVO,EPS,AUX
C INTEGER *4 P(*),N
C DATA EPS / 1.D-30 /
C
C P = VETOR DE PERMUTACAO
C
C DO 10 I = 1,N
C   P(I) = I
C CONTINUE
C DO 20 K = 1,(N-1)
C
C CALCULO DO MAIOR ELEMENTO EM MODULO DA K-ESIMA COLUNA
C
C   L = K
C   PIVO = DABS(A(K,K))
C   DO 30 I = (K+1),N
C     AUX = DABS(A(I,K))
C     IF(PIVO .GE. AUX) GO TO 30
C     PIVO = AUX
C     L = I
C 30 CONTINUE
C   IF(PIVO .LE. EPS) GO TO 40
C   IF(L .EQ. K) GO TO 50
C
C TROCA DE LINHAS
C
C   DO 60 I = 1,N
C     AUX = A(K,I)
C     A(K,I) = A(L,I)
C     A(L,I) = AUX
C 60 CONTINUE
C   II = P(K)
C   P(K) = P(L)
C   P(L) = II
C
C INICIO DA DECOMPOSICAO L.U
C
C 50 DO 70 I = (K+1),N
C     PIVO = A(I,K)/A(K,K)
C     DO 80 J = (K+1),N
C       A(I,J) = A(I,J) - PIVO*A(K,J)
C 80 CONTINUE
```

```

          A(I,K) = PIVO
70      CONTINUE
20      CONTINUE
        IF(DABS(A(N,N)) .GT. EPS) RETURN
40      CALL FALHA (10,0)
        STOP
      END

```

```

C
C =====
C
      SUBROUTINE DIAGON (ILIN,JCOL,A,NELEM,MDIM,DIAG)

```

```

C
C -----
C
C
C OBJETIVO: DETERMINACAO DA DIAGONAL DE UMA MATRIZ A
C
C

```

```

      REAL *8 A(*),DIAG(*),ANEW
      INTEGER *4 ILIN(*),JCOL(*),NELEM,MDIM,INEW,JNEW
      DO 10 I=1,MDIM
          INEW = I
          JNEW = I
          CALL SRCH (ILIN,JCOL,A,NELEM,INEW,JNEW,ANEW)
          DIAG(I) = ANEW
10      CONTINUE
      RETURN
      END

```

```

C
C =====
C
      SUBROUTINE KSPACE (A,ILIN,JCOL,Q,DIAG,SUP,SUB,
1          IFLAG,NELEM,MDIM,NSUB,NMAX)

```

```

C
C -----
C
C
C OBJETIVO: CONSTRUCAO DOS SUBESPACOS DE KRYLOV
C
C

```

```

      REAL *8 A(*),Q(NMAX,*),DIAG(*),SUP(*),SUB(*)
      INTEGER *4 MDIM,NSUB,NELEM,ILIN(*),JCOL(*),IFLAG
      DO 10 KSPC = 2,NSUB
          CALL PRODAX (ILIN,JCOL,A,NELEM,MDIM,Q(1,KSPC-1),Q(1,KSPC))
          IF(IFLAG .EQ. 1) THEN
              CALL SOLVTD (DIAG,SUP,SUB,Q(1,KSPC),Q(1,KSPC),MDIM)
          END IF
10      CONTINUE
      RETURN
      END

```

```

=====
SUBROUTINE GALERK (A, ILIN, JCOL, NELEM, X, Q, U, Z, P, DIAG, SUP, SUB,
1          IFLAG, MDIM, NSUB, KL, KC)
-----

```

```

OBJETIVO:  CALCULO DA NOVA APROXIMACAO
           UTILIZANDO O METODO DE GALERKIN

```

```

REAL      *8 A(*), X(*), ERRO, SUMB, SUMV
REAL      *8 Q(KL,*), U(KC,*), Z(*), DIAG(*), SUP(*), SUB(*)
INTEGER   *4 MDIM, NSUB, NELEM, ILIN(*), JCOL(*), P(*), IFLAG
COMMON /TESTCV/ ERRO
DO 10 J = 1, NSUB
  Z(J) = 0.D0
DO 20 I = 1, NSUB
  U(I, J) = 0.D0
20 CONTINUE
10 CONTINUE
DO 30 J = 1, NSUB
  CALL PRODAX (ILIN, JCOL, A, NELEM, MDIM, Q(1, J), Q(1, NSUB+1))
  IF(IFLAG .EQ. 1) THEN
    CALL SOLVTD (DIAG, SUP, SUB, Q(1, NSUB+1), Q(1, NSUB+1), MDIM)
  END IF
DO 40 I = 1, NSUB
  SUMV = 0.D0
DO 50 L = 1, MDIM
  SUMV = SUMV + Q(L, I)*Q(L, NSUB+1)
50 CONTINUE
  U(I, J) = SUMV
40 CONTINUE
  SUMB = 0.D0
DO 60 L = 1, MDIM
  SUMB = SUMB + Q(L, J)*Q(L, 1)
60 CONTINUE
  Z(J) = SUMB
30 CONTINUE

```

```

RESOLUCAO DO SISTEMA LINEAR GERADO PELO METODO DE GALERKIN

```

```

IF(NSUB .EQ. 1) THEN
  Z(1) = Z(1) / U(1, 1)
ELSE
  CALL DECOMP (U, P, NSUB, KC)
  CALL SOLVE (U, P, Z, Q(1, NSUB+2), NSUB, KC)
END IF

```

```

CALCULO DA NOVA APROXIMACAO

```

```

CALL FULLAX (Q, Q(1, NSUB+2), Q(1, NSUB+1), MDIM, NSUB, KL)

```

```

C
C   CALCULO DA NORMA DO ERRO ABSOLUTO
C
      CALL SUPNOR (Q(1,NSUB+1),MDIM,ERRO)
      DO 70 L = 1,MDIM
        X(L) = X(L) + Q(L,NSUB+1)
70    CONTINUE
      RETURN
END

```

```

C
C   =====
C
C   SUBROUTINE VETOR (I,J,N,K)
C
C   -----
C
C   OBJETIVO: CALCULO DA RELACAO ENTRE I,J E K
C
C
      INTEGER *4 I,J,N,K,MENOS
      MENOS=((I-1)*I)/2
      K=(I-1)*N+J-MENOS
      RETURN
END

```



```

10      CONTINUE
      CALL PINNER (R,WORK,NDIM,ALFA)
20      ITER = ITER + 1
C
C      CALCULO DO          STEP = < R , R > / < A.R , A.R >
C
      CALL PRODAX (ILIN,JCOL,A,NELEM,NDIM,VET,WORK)
      CALL PINNER (WORK,WORK,NDIM,GAMA)
      STEP = ALFA / GAMA
C
C      NOVA APROXIMACAO   e      TESTE DE CONVERGENCIA
C
      ERRO = 0.D0
      DO 30 I=1,NDIM
        XNEW = X(I) + STEP*VET(I)
        AUX = DABS(XNEW-X(I))
        IF(ERRO .LT. AUX) ERRO = AUX
        X(I) = XNEW
30      CONTINUE
C
C      CALCULO DO RESIDUO
C
      CALL GRAD (ILIN,JCOL,A,NELEM,NDIM,WORK,X,B,R)
      IF(IFLAG .EQ. 0) THEN
        DO 15 I=1,NDIM
          WORK(I)=R(I)
15      CONTINUE
      ENDIF
      IF(IFLAG .EQ. 1) CALL SOLVTD (DIAG,SUP,SUB,R,WORK,NDIM)
      IF(IFLAG .EQ. 2) CALL SOLVDG (DIAG,R,WORK,NDIM)
      CALL SUPNOR (R,NDIM,RES)
C
C      CALCULO DA NOVA DIRECAO A-CONJUGADA
C
      BETA = ALFA
      CALL PINNER (R,WORK,NDIM,ALFA)
      BETA = ALFA / BETA
      DO 35 I=1,NDIM
        VET(I) = WORK(I) + BETA*VET(I)
35      CONTINUE
C
C      OPCAO PARA IMPRESSAO DE RESULTADOS PARCIAIS
      WRITE(LOUT2,200) ITER,ERRO,RES
C200  FORMAT(//,T5,' ITERACAO      : ',I3.3,/,T5,
C      1      'E R R O      : ',E12.5,/,T5,
C      2      'RESIDUO      : ',E12.5)
C
      IF(ERRO .LE. EPS .AND .RES .LE. EPS) RETURN
      IF(ITER .LE. LIMAX) GO TO 20
      CALL FALHA (18,ITER)
      STOP
      END

```



```

C
C =====
C
1   SUBROUTINE KRYLOV (ILIN,JCOL,A,X,B,
      DIAG,SUP,SUB,NDIM,NELEM,IFLAG,NSUB)
C
C -----
C OBJETIVO:  RESOLUCAO DE SISTEMA LINEAR ESPARSO
C            UTILIZANDO O METODO DOS SUBESPACOS DE
C            KRYLOV, SEM E COM PRECONDICIONAMENTO (IFLAG)
C
      PARAMETER (MAX = 1200, MSPACE = 2)
      REAL      *8 A(*),X(*),B(*),DIAG(*),SUP(*),SUB(*),
1             Q(MAX,MSPACE),U(MSPACE,MSPACE),Z(MSPACE),
2             EPS,RES,ERRO
      INTEGER *4 ILIN(*),JCOL(*),NELEM,NDIM,NSUB,LIMAX,ITER,
1             P(MSPACE),IFLAG
      INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
      COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
      COMMON /CONVRG/ EPS,LIMAX,ITER
      COMMON /TESTCV/ ERRO
      IF(NSUB .GT. MSPACE-2) CALL FALHA (8,NSUB)
      IF(NDIM .GT. MAX) CALL FALHA (9,NDIM)
      CALL RESID (ILIN,JCOL,A,NELEM,NDIM,X,B,Q(1,1))
      CALL SUPNOR (Q(1,1),NDIM,RES)
      ITER = 0
20     ITER = ITER + 1
C
C CONSTRUCAO DOS SUB - ESPACOS DE KRYLOV
      IF(IFLAG .EQ. 1) CALL SOLVTD
1             (DIAG,SUP,SUB,Q(1,1),Q(1,1),NDIM)
      CALL KSPACE (A,ILIN,JCOL,Q,DIAG,SUP,SUB,
1             IFLAG,NELEM,NDIM,NSUB,MAX)
C
C NOVA APROXIMACAO
      CALL GALERK (A,ILIN,JCOL,NELEM,X,Q,U,Z,P,DIAG,SUP,SUB,
1             IFLAG,NDIM,NSUB,MAX,MSPACE)
C
C TESTE DE CONVERGENCIA
      CALL RESID (ILIN,JCOL,A,NELEM,NDIM,X,B,Q(1,1))
      CALL SUPNOR (Q(1,1),NDIM,RES)
C
C OPCAO PARA IMPRESSAO DE RESULTADOS PARCIAIS
      WRITE(LOUT2,200) ITER,ERRO,RES
C200  FORMAT(//,T10,'I T E R A C A O      :      ',I3.3,//,T10,
C      1      'ERRO ABSOLUTO      :      ',D12.5,//,T10,
C      2      'R E S I D U O      :      ',D12.5,/)
      IF(ERRO .LE. EPS .AND. RES .LE. EPS) RETURN
      IF(ITER .LT. LIMAX) GO TO 20
      CALL FALHA (18,ITER)
      STOP
      END

```

```
=====
SUBROUTINE CGS (ILIN,JCOL,A,X,B,DIAG,SUP,SUB,NDIM,NELEM,IFLAG)
-----
```

```
OBJETIVO: RESOLUCAO DE SISTEMA LINEAR ESPARSO
           UTILIZANDO O METODO DOS GRADIENTES
           CONJUGADOS QUADRATICOS, SEM E COM
           PRECONDICIONAMENTO (IFLAG)
```

```
PARAMETER (MAX = 1200)
REAL      *8 A(*),X(*),B(*),DIAG(*),SUP(*),SUB(*),
1          Q(MAX),U(MAX),P(MAX),R(MAX),R0(MAX),
2          ALFA,BETA,SIGMA,GAMA,ERRO,AUX,RES,EPS,XNEW
INTEGER *4 ILIN(*),JCOL(*),NDIM,NELEM,ITER,LIMAX,IFLAG
INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
COMMON /CONVRG/ EPS,LIMAX,ITER
```

```
INICIO DO PROCESSO ITERATIVO
```

```
IF(NDIM .GT. MAX) CALL FALHA (9,NDIM)
ITER = 0
GAMA = 1.D0
CALL RESID (ILIN,JCOL,A,NELEM,NDIM,X,B,R)
IF(IFLAG .EQ. 1) CALL SOLVTD (DIAG,SUP,SUB,R,R,NDIM)
DO 10 I=1,NDIM
  Q(I) = 0.D0
  P(I) = 0.D0
  R0(I) = R(I)
CONTINUE
10  ITER = ITER + 1
20  CALL PINNER (R,R0,NDIM,BETA)
    ALFA = BETA
    BETA = BETA / GAMA
    DO 25 I=1,NDIM
      U(I) = R(I) + BETA*Q(I)
      P(I) = U(I) + BETA*( Q(I) + BETA*P(I) )
25  CONTINUE
    CALL PRODAX (ILIN,JCOL,A,NELEM,NDIM,P,Q)
    IF(IFLAG .EQ. 1) CALL SOLVTD (DIAG,SUP,SUB,Q,Q,NDIM)
    CALL PINNER (R0,Q,NDIM,SIGMA)
    GAMA = ALFA
    ALFA = ALFA / SIGMA
```

```
NOVA APROXIMACAO e TESTE DE CONVERGENCIA
```

```
ERRO = 0.D0
DO 30 I=1,NDIM
  Q(I) = U(I) - ALFA*Q(I)
```

```

XNEW = X(I) + ALFA*( U(I) + Q(I) )
AUX = DABS(XNEW-X(I))
IF(ERRO .LT. AUX) ERRO = AUX
X(I) = XNEW
30 CONTINUE
C
C CALCULO DO RESIDUO
C
CALL RESID (ILIN,JCOL,A,NELEM,NDIM,X,B,R)
IF(IFLAG .EQ. 1) CALL SOLVTD (DIAG,SUP,SUB,R,R,NDIM)
CALL SUPNOR (R,NDIM,RES)
C
C OPCAO PARA IMPRESSAO DE RESULTADOS PARCIAIS
C
WRITE(LOUT2,200) ITER,ERRO,RES
C200 FORMAT(//,T5,'ITERACAO : ',I3.3,/,T5,
C 1 'E R R O : ',E12.5,/,T5,
C 2 'RESIDUO : ',E12.5)
C
IF(ERRO .LE. EPS .AND. RES .LE. EPS) RETURN
IF(ITER .LE. LIMAX) GO TO 20
CALL FALHA (18,ITER)
STOP
END

```

```
C
C -----
C SUBROUTINE MGSQR (Q,R,MLIN,NCOL,MAX)
C -----
C
C
C OBJETIVO: DECOMPOSICAO QR DE UMA MATRIZ POR
C           MEIO DO METODO DE GRAM-SCHMIDT MODIFICADO
C
C
C REAL      *8 Q(MAX,*), R(*), EPS , SOMA
C INTEGER  *4 MLIN,NCOL
C DATA  EPS / 1.D-12 /
C   DO 80 K=1,NCOL
C     SOMA = 0.D0
C     DO 10 I=1,MLIN
C       SOMA = SOMA + Q(I,K)*Q(I,K)
10    CONTINUE
C     CALL VETOR (K,K,NCOL,KK)
C     R(KK) = DSQRT( SOMA )
C     IF(R(KK) .LE. EPS) CALL FALHA (17,0)
C     DO 20 I=1,MLIN
C       Q(I,K) = Q(I,K) / R(KK)
20    CONTINUE
C     DO 70 J=(K+1),NCOL
C       SOMA = 0.D0
C       DO 30 I=1,MLIN
C         SOMA = SOMA + Q(I,K)*Q(I,J)
30    CONTINUE
C       CALL VETOR (K,J,NCOL,KJ)
C       R(KJ) = SOMA
C       DO 60 I=1,MLIN
C         Q(I,J) = Q(I,J) - Q(I,K)*R(KJ)
60    CONTINUE
C       CONTINUE
C     CONTINUE
C   CONTINUE
C RETURN
C END
```

```

C
C =====
C
C SUBROUTINE CGSQR (A,V,M,N,MAX)
C -----
C
C OBJETIVO: DECOMPOSICAO QR DA MATRIZ A POR
C           MEIO DO METODO DE GRAM-SCHMIDT CLASSICO
C
C
C REAL      *8 A(MAX,*), V(*), EPS, ALFA, SOMA, NORMA2
C INTEGER   *4 M,N
C DATA    EPS / 1.D-12 /
C   DO 10 J=1,N
C     DO 30 K=1,(J-1)
C       ALFA=0.DO
C       DO 40 I=1,M
C         ALFA=ALFA + A(I,K)*A(I,J)
40      CONTINUE
C       CALL VETOR (K,J,N,KJ)
C       V(KJ)=ALFA
C       DO 50 I=1,M
C         A(I,J)=A(I,J)-ALFA*A(I,K)
50      CONTINUE
30      CONTINUE
C     SOMA=0.DO
C     DO 60 I=1,M
C       SOMA=SOMA + A(I,J)*A(I,J)
60      CONTINUE
C     NORMA2=DSQRT(SOMA)
C     CALL VETOR (J,J,N,JJ)
C     V(JJ)=NORMA2
C     IF(NORMA2 .LE. EPS) CALL FALHA (17,0)
C     DO 70 I=1,M
C       A(I,J)=A(I,J)/NORMA2
70      CONTINUE
10      CONTINUE
C   RETURN
C END

```



```

        L=NF+J-1
        POT(L)=POTCON(I)
        ANG=2.D0*3.1415926D0*DFLOAT(J)/FLOAT(MF)
        XF(L)=XCEN(I)+RAIO(I)*DCOS(ANG)
        YF(L)=YCEN(I)+RAIO(I)*DSIN(ANG)
30      CONTINUE
        NQ=NQ+MQ
        NF=NF+MF
10      CONTINUE
        READ(LIN,*)X,Y,NX,STEPX,NY,STEPY
        WRITE(LOUT1,*)
        WRITE(LOUT1,*)X,Y,NX,STEPX,NY,STEPY
        NQ=NQ-1
        NF=NF-1
        WRITE(LOUT1,*)
        WRITE(LOUT1,*)' APOS CALCULOS INICIAIS:'
        WRITE(LOUT1,*)' ====='
        WRITE(LOUT1,*)
        WRITE(LOUT1,*)' CARGAS / PONTOS DE CONTORNO:'
        WRITE(LOUT1,*)
        WRITE(LOUT1,*)'   I           XQ           YQ           XF   ',
1          '           YF           POT'
        DO 40 I=1,NQ
            WRITE(LOUT1,50)I,XQ(I),YQ(I),XF(I),YF(I),POT(I)
50          FORMAT(I3,5(3X,F9.4))
40      CONTINUE
        DO 60 I=NQ+1,NF
            WRITE(LOUT1,70)I,XF(I),YF(I),POT(I)
70          FORMAT(I3,24X,3(3X,F9.4))
60      CONTINUE
        RETURN
        END

```

```

C
C -----
C
C SUBROUTINE MATRIZ (NQ,NF,XQ,YQ,XF,YF,IDIM1,A)
C -----
C

```

OBJETIVO: CALCULO DOS ELEMENTOS DA MATRIZ DOS COEFICIENTES

```

C
C
C
C
C
C
C
C
10      REAL      *8 A(IDIM1,*),XQ(*),YQ(*),XF(*),YF(*)
        INTEGER  *4 NQ,NF
        DO 10 I=1,NF
            DO 10 J=1,NQ
                CALL PIJ (XF(I),YF(I),XQ(J),YQ(J),A(I,J))
10      CONTINUE
        RETURN
        END

```



```

C
C =====
C
C SUBROUTINE CHECK (N, XCEN, YCEN, POTCON, RAI0, NQ, XQ, YQ, Q, NC)
C -----
C
C OBJETIVO: CALCULOS FINAIS E IMPRESSAO DE RESULTADOS
C
C
C
C
C
1  REAL      *8 XCEN(*), YCEN(*), POTCON(*), RAI0(*), XQ(*), YQ(*),
    Q(*), XA, YA, RES, VAL, ANG, PORC
INTEGER *4 NC(*), N, NQ
COMMON /IMPRES/LIN, LOUT1, LOUT2, LOUT3
    WRITE(LOUT2, *)
    WRITE(LOUT2, *) 'RESULTADOS:'
    WRITE(LOUT2, *) '=====
    WRITE(LOUT2, *)
    WRITE(LOUT2, *) 'CARGAS (C/4.PI.E)'
    WRITE(LOUT2, *)
    DO 10 I=1, NQ
        WRITE(LOUT2, 20) I, Q(I)
20      FORMAT(I3, F9.4)
10    CONTINUE
    WRITE(LOUT2, *)
    WRITE(LOUT2, *) 'POTENCIAIS (P/CHECAGEM) SOBRE OS CONTORNOS (V)'
    WRITE(LOUT2, *) '=====
    WRITE(LOUT2, *)
    WRITE(LOUT2, *) ' N.O           X           Y           POT.DADO'
1    WRITE(LOUT2, *) '           POT.CALC.       DIF(%)'
    WRITE(LOUT2, *)
    L=0
    DO 30 I=1, N
        DO 30 J=1, NC(I)
            ANG = DFLOAT(J-1)*3.1415626D0*2.D0/DFLOAT(NC(I))
            XA = XCEN(I) + RAI0(I)*DCOS(ANG)
            YA = YCEN(I) + RAI0(I)*DSIN(ANG)
            RES=0.D0
            DO 40 K=1, NQ
                CALL PIJ (XA, YA, XQ(K), YQ(K), VAL)
                RES = RES + VAL*Q(K)
40          CONTINUE
            PORC = DABS(POTCON(I)-RES)/DABS(POTCON(I))*1.D2
            L = L+1
            WRITE(LOUT2, 50) L, XA, YA, POTCON(I), RES, PORC
50          FORMAT(I3, 5(3X, F9.4))
30    CONTINUE
    RETURN
END

```

```
C
C
C -----
C SUBROUTINE SAIDA (NQ,XQ,YQ,Q,X,Y,NX,STEPX,NY,STEPY)
C -----
C
C
C OBJETIVO: CALCULOS FINAIS E IMPRESSAO DE RESULTADOS
C
C
```

```
REAL *8 XQ(*),YQ(*),Q(*),X,Y,STEPX,STEPY,XA,YA,RES,VAL
INTEGER *4 NQ,NX,NY
COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
WRITE(LOUT2,*)
WRITE(LOUT2,*) 'POTENCIAIS CALCULADOS (V)'
WRITE(LOUT2,*) '===== '
WRITE(LOUT2,*)
WRITE(LOUT2,*) ' N.O          X          Y          POT'
WRITE(LOUT2,*)
L=0
DO 30 I=1,NY
  YA = Y - DFLOAT(I-1)*STEPY
  DO 30 J=1,NX
    XA = X + DFLOAT(J-1)*STEPX
    RES=0.D0
    DO 40 K=1,NQ
      CALL PIJ (XA,YA,XQ(K),YQ(K),VAL)
      RES = RES + VAL*Q(K)
40    CONTINUE
    L = L+1
    WRITE(LOUT2,50)L,XA,YA,RES
50    FORMAT(I3,3X,F9.4,3X,F9.4,3X,F9.4)
30    CONTINUE
RETURN
END
```

OBS: Demais subrotinas do programa MSC : Idem ao programa CCE


```

2          IDIM6 = IDIM2*(IDIM2+1)/2,    IDIM7 = 5*NOS+7*NB )
CC REAL *8 CHEIA1 (IDIM1, IDIM2), CHEIA2 (IDIM1, IDIM2), ESP (IDIM7),
REAL *8                                     ESP (IDIM7),
1          FI0 (NP), FI1 (IDIM4), FIC (IDIM3), B2 (IDIM4), WK1 (IDIM1+NB),
CC 2          WK2 (IDIM4+NP), RO (IDIM2), NI (IDIM2), POT1 (NCS), V (IDIM6),
2          WK2 (IDIM4+NP), RO (IDIM2), NI (IDIM2), POT1 (NCS),
3          XNO (NOS), YNO (NOS), XCAR (NC), YCAR (NC), XFRONT (NP), YFRONT (NP),
4          XFR (NCS), YFR (NCS), XCG (NCS), YCG (NCS), VET1 (3, NP),
6          VET2 (IDIM5), XCOND (NFD), YCOND (NFD), RAI (NFD), POT (NFD),
7          RAI0 (NFD), EPS (NFD), TETA (IDIM2), X2 (NB), Y2 (NB), DIAG (IDIM4),
8          SUP (IDIM4), SUB (IDIM4), AUXFR (8), ERRO1, ERRO2, DIF, POT0
INTEGER *4 LK1 (2, NP), LK2 (IDIM5), ILIN (IDIM7), JCOL (IDIM7),
1          KOD (IDIM4), KONT, ITER, LIMAX, MAXIT, NELEM, NSUB, IFLAG
INTEGER *4 NCSM, NFD, NCG, NNO, NFR, NO, MO, M2
INTEGER *4 LIN, LOUT1, LOUT2, LOUT3
CHARACTER *9 ENTR1, SAID1, SAID2, SAID3
CHARACTER *3 MET, QR
COMMON /CONVRG/ERRO2, LIMAX, ITER
COMMON /IMPRES/LIN, LOUT1, LOUT2, LOUT3
COMMON /DIMENS/NCSM, NFD, NCG, NNO, NFR, NO, MO, M2

C
C
C      LEITURA DE DADOS E INICIALIZACAO

LIN = 5
LOUT1 = 6
LOUT2 = 6
LOUT3 = 6
CALL TECL (ENTR1, SAID1, SAID2, SAID3, 1)
CALL DADOS (XCOND, YCOND, RAI, POT, RAI0, EPS, XNO, YNO, XFRONT, YFRONT,
1          XCAR, YCAR, XFR, YFR, XCG, YCG, POT1, AUXFR, ERRO1, ERRO2,
2          LIMAX, MAXIT, POT0, TETA, NSUB, MET, QR)
IF (NCSM .GT. NCS) CALL FALHA (1, NCSM)
IF (NFD .GT. NFD) CALL FALHA (2, NFD)
IF (NCG .GT. NC) CALL FALHA (3, NCG)
IF (NFR .GT. NP) CALL FALHA (4, NFR)
IF (NNO .GT. NOS) CALL FALHA (5, NNO)
CALL INIC (POT1, POT0, POT (1), NB, B2, WK1, WK2, RO, NI, FI0, FI1)
CC CALL CHEPOT (XFRONT, YFRONT, XCAR, YCAR, XFR, YFR,
CC 1          XCG, YCG, IDIM1, CHEIA1, CHEIA2)
C
C      CALCULO DA APROXIMACAO INICIAL
C      PARA O VETOR DE CARGAS ELETRICAS
C
CC IF (QR .EQ. 'CQR') THEN
CC     CALL CGSQR (CHEIA2, V, NCSM+NFR, NCSM+NCG, IDIM1)
CC     ELSE
CC     CALL MGSQR (CHEIA2, V, NCSM+NFR, NCSM+NCG, IDIM1)
CC ENDIF
CC CALL PRDATX (CHEIA2, WK1, WK2, NCSM+NFR, NCSM+NCG, IDIM1)
CC CALL UPPER (V, WK2, RO, NCSM+NCG)
C
C      CALCULO DOS ELEMENTOS DO SISTEMA DE EQUACOES
CC CALL CHECAM (CHEIA1, POT1, XFRONT, YFRONT, XCAR, YCAR,

```

```

CC      1          XCG, YCG, IDIM1, CHEIA2, WK1)
      CALL CONEX (AUXFR, VET1, VET2, LK1, LK2)
      CALL DIFER (XCOND, YCOND, RAI, POT, RATIO, EPS, KOD, XNO, YNO,
2      AUXFR, ILIN, JCOL, ESP, B2, X2, Y2, NELEM)
      IF(M2 .GT. NB) CALL FALHA (6, M2)
C
C      DECOMPOSICAO QR DA SUBMATRIZ P2 + F
C
CC      IF(QR .EQ. 'CQR') THEN
CC          CALL CGSQR (CHEIA2, V, NCSM+NFR, NCSM+NCG, IDIM1)
CC          ELSE
CC          CALL MGSQR (CHEIA2, V, NCSM+NFR, NCSM+NCG, IDIM1)
CC      ENDIF
C
C      OPCAO PARA IMPRESSAO DE RESULTADOS
C      PARCIAIS (SUBROTINAS IMPR3 E IMPR4)
C
C      CALL IMPR3 (XNO, YNO, XFRONT, YFRONT, XCAR, YCAR, XFR, YFR, XCG, YCG)
C      CALL IMPR4 (CHEIA1, CHEIA2, IDIM1, NNO+M2, ILIN, JCOL,
1      ESP, VET1, VET2, LK1, LK2, POT1, B2, NELEM)
C
C      CALCULOS PRELIMINARES PARA PRECONDICIONAMENTO
C
C      IFLAG = 0
C      IF(MET .EQ. 'CGT' .OR. MET .EQ. 'KST' .OR. MET .EQ. 'CST') THEN
C          CALL TRIDIG (ILIN, JCOL, ESP, NELEM, NNO+M2, DIAG, SUP, SUB)
C          IF(MET .EQ. 'CGT') CALL BIDATA (DIAG, SUP, SUB, NNO+M2)
C          CALL TRIDLU (DIAG, SUP, SUB, NNO+M2)
C          IFLAG = 1
C      ENDIF
C      IF(MET .EQ. 'CGD') THEN
C          CALL DIGATA (JCOL, ESP, NELEM, NNO+M2, DIAG)
C          IFLAG = 2
C      ENDIF
C
C      RESOLUCAO ITERATIVA DO SISTEMA LINEAR
C
CC          KONT=0
CC20         KONT=KONT+1
C
CC      DEFINICAO DO VETOR B
CC
      CALL LEITFI (NFR, FI0)
      CALL ATUB2 (B2, FI0, VET2, LK2, WK2)
C
C      RESOLUCAO DO SISTEMA DE
C      EQUACOES DE DIFERENCAS FINITAS
C
      IF(MET.EQ.'CG' .OR. MET.EQ.'CGD' .OR. MET.EQ.'CGT') THEN
1          CALL CG (ILIN, JCOL, ESP, FI1, WK2,
                  DIAG, SUP, SUB, NNO+M2, NELEM, IFLAG)
      ENDIF
      IF(MET.EQ.'KS' .OR. MET.EQ.'KST') THEN
          CALL KRYLOV (ILIN, JCOL, ESP, FI1, WK2,

```

```

1          DIAG,SUP,SUB,NNO+M2,NELEM,IFLAG,NSUB)
ENDIF
IF(MET.EQ.'CS' .OR. MET.EQ.'CST') THEN
    CALL CGS (ILIN,JCOL,ESP,FI1,WK2,
1          DIAG,SUP,SUB,NNO+M2,NELEM,IFLAG)
ENDIF

C
C          CALCULO DO VETOR DE CARGAS
C          PELO METODO DOS QUADRADOS MINIMOS
C
CC        CALL ATUB1 (FI0,FI1,POT1,VET1,LK1,WK1)
CC        CALL PRDATX (CHEIA2,WK1,WK2,NCSM+NFR,NCSM+NCG,IDIM1)
CC        CALL UPPER (V,WK2,NI,NCSM+NCG)
C
C          TESTE DE CONVERGENCIA E ATUALIZACAO
C          DOS POTENCIAIS ELETRICOS NA FRONTEIRA
C
CC        CALL TESTE (RO,NI,TETA,DIF)
CC        IF(DIF.LE.ERRO1) GO TO 10
CC        CALL ATUPOT (CHEIA1,RO,IDIM1,FI0)
CC        CALL ITERAT (ITER,KONT,DIF)
CC        IF(KONT.GE.MAXIT) CALL FALHA (7,KONT)
CC        GO TO 20
C
C          IMPRESSAO DOS RESULTADOS
C
CC10     CALL IMPR1 (RO,FI0,FI1,XCG,YCG,XCAR,YCAR,
CC        1          XNO,YNO,XFRONT,YFRONT,KONT)
CC        CALL CONDUT (FIC,XNO,YNO,XCOND,YCOND,RAI,POT)
CC        CALL IMPR2 (XFRONT,YFRONT,XNO,YNO,X2,Y2,FI0,FI1,FIC)
CC
CC        IMPRESSAO DOS RESULTADOS
CC
CALL IMPR5 (NNO,FI1)
CALL TECL (ENTR1,SAID1,SAID2,SAID3,2)
STOP
END

```

```

C
C *****
C *****
C ***** SUBROTINAS *****
C *****
C *****
C =====
C
C SUBROUTINE IMPR5 (NNO,XNO,YNO,FI1)
C -----
C
C OBJETIVO: IMPRIME RESULTADOS FINAIS
C
C REAL *8 XNO(*),YNO(*),FI1(*)
C INTEGER *4 NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C INTEGER *4 LIN,LOUT1,LOUT2,LOUT3
C COMMON /IMPRES/LIN,LOUT1,LOUT2,LOUT3
C COMMON /DIMENS/NCSM,NFDM,NCG,NNO,NFR,NO,MO,M2
C WRITE(LOUT2,*)
C WRITE(LOUT2,*) ' RESULTADOS FINAIS '
C WRITE(LOUT2,*)
C WRITE(LOUT2,*) ' POTENCIAIS DOS NOS DA MALHA (KV): '
C WRITE(LOUT2,*) ' N.O X Y V'
C WRITE(LOUT2,*)
C DO 10 I=1,NNO
C WRITE(LOUT2,30) I,XNO(I),YNO(I),FI1(I)
10 CONTINUE
C RETURN
C END

```

```
C
C -----
C
C SUBROUTINE LEITFI (NFR,FIO)
C -----
C
C
```

```
C OBJETIVO: LEITURA DE POTENCIAIS SOBRE A FRONTEIRA RETANGULAR
C
```

```
C
C REAL          *8 FIO(*)
C INTEGER       *4 LIN1,NFR
C CHARACTER     *20 ENTRAD
C   LIN1 = 5
C   WRITE(*,*)
C   WRITE(*,*) ' ENTRE COM NOME.EXTENSAO DO ARQUIVO DOS '
C   WRITE(*,*)
C   WRITE(*,*) ' DADOS DE POTENCIAIS NA FRONTEIRA >>> '
20  READ(*,20) ENTRAD
C   FORMAT(A20)
C   WRITE(*,*)
C   WRITE(*,*) ' AGUARDE ... '
C   WRITE(*,*)
C   WRITE(*,*)
C   OPEN (UNIT=LIN1,FILE=ENTRAD,STATUS='OLD')
C   DO 10 I=1,NFR
10  READ(LIN1,*) FIO(I)
C   CONTINUE
C   RETURN
END
```

OBS: Demais subrotinas do programa MDF : Idem ao programa CCE

REFERÊNCIAS

- [1] HOFFMANN, J. N., Simulação numérica para o cálculo de campos elétricos em domínios ilimitados, Tese de mestrado, UNICAMP/IMECC/DMA, Abril 1993

- [2] STEINBIGLER, H., Combined application of finite element method and charge simulation method for the computation of electric fields, 3rd International Symposium on High Voltage Engineering, grupo II-II, Agosto 1979

- [3] OKUBO, H., IKEDA, M., HONDA, M., Electric field analysis by combination method, IEEE Transactions on Power Apparatus and Systems, Vol. 101, pp. 4039-4048, Outubro 1982

- [4] ABDEL-SALAM, M., EL. MOHANDÉS, M. T., Combined method based on finite differences and charge simulation for calculating electric fields, IEEE Transactions on Industry Applications, vol. 25, pp. 1060-1066, Novembro-Dezembro 1989