Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

ALGORÍTIMO PARA SIMULAÇÃO NUMÉRICA DE POTENCIAIS ELÉTRICOS EM ELETROMAGNETISMO COM O USO DO MÉTODO DAS DIFERENÇAS FINITAS

Alexandre Maniçoba de Oliveira (UNISANTOS/USP)
amanicoba@pad.lsi.usp.br / alexandre.manicoba@unisantos.br
Héctor Dave Orrillo Ascama(USP/SP)
hector@pad.lsi.usp.br

Resumo: O grande interesse no estudo numérico de campos eletromagnéticos se deve a sua complexidade, o que torna tal estudo muito custoso. Para minimizar o custo e tempo envolvido na análise de campos de potenciais em eletromagnetismo, utiliza-se o computador para desempenhar a análise do fenômeno em estudo através da iteração e da otimização. Neste trabalho é apresentado o projeto de uma ferramenta computacional desenvolvida em C++ para simulação numérica de potenciais elétricos utilizando o método das diferenças finitas para o estudo da interação entre campos potenciais em uma superfície de um mesmo material, homogenia e com duas dimensões. O programa apresenta seus resultados na forma gráfica onde na posição cartesiana referente a posição da célula na matriz, é "plotado" no gráfico um símbolo (x para valores negativos e um quadrado para valores positivos) e a intensidade do potencial é representada através da cor do símbolo. Com base nos testes do programa, pode-se constatar seu satisfatório funcionamento, bem como comprovar a funcionalidade das técnicas de métodos de diferencias finitas.

Palavras-chaves: Algoritmo de Simulação Numérica, Elementos Finitos, Diferenças Finitas.

Abstract: *This paper presents the design of software written in C ++ and using finite element techniques to study the potential interaction between fields on a surface of the same material. Based on the testing program, we can verify their satisfactory operation and demonstrate the functionality of the techniques of finite methods.*

Keywords: *Numerical Simulation Algorithm, Finite Element, Finite Difference.*

1. Introdução

1.1 Motivação

Este trabalho foi motivado pela importância e contribuição que algoritmos de simulação numéricas realizam no que tange a redução significativa dos custos de análise de campos potenciais na área de eletromagnetismo.

1.2 Problema proposto

A Figura 1 apresenta uma superfície bidimensional onde dois segmentos de retas apresentam potenciais distintos. Nela os condutores em fitas, imersos em um meio dielétrico de permitividade igual a do vácuo, são representados em corte transversal e possuem potenciais de -10 e 10V respectivamente e todo contorno com potencial de 0V.

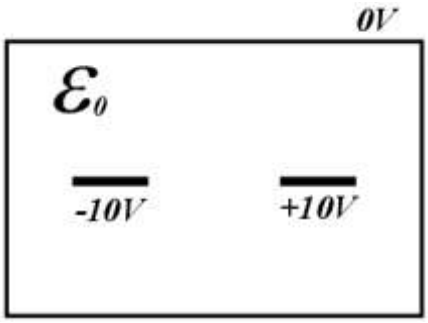

Figura 1 - Representação do corte transversal de uma dupla linha de transmissão.

Para realizar a análise da interação de campos magnéticos, foi necessário o desenvolvimento de um algoritmo.

1.3 Objetivo

Este trabalho tem por principal objetivo estudar a técnica de diferenças finitas e propor um algoritmo que promova a análise de um problema proposto no domínio do eletromagnetismo na forma de um estudo teórico dirigido aos métodos finitos e suas prováveis aplicações através de levantamento bibliográfico.

1.4 Estrutura do trabalho

Este trabalho está organizado em quatro seções. Inicialmente a Seção 1 contendo a introdução ao trabalho, na Seção 2 é feita uma revisão da literatura sobre conceitos, históricos e teorias sobre elementos finitos, na Seção 3 é apresentado o desenvolvimento do algoritmo propriamente dito e por fim na Seção 4 é apresentado a conclusão.

2. Fundamentação Teórica

2.1. Histórico do método de elementos finitos

Na década de 1950 iniciou-se o estudo do método de elementos finitos que tinha como proposta solucionar problemas na área de engenharia mecânica. Um exemplo de aplicação do método de elementos finitos é a análise das tensões em estruturas metálicas, termodifusão, dinâmica de fluidos, etc (BASTOS, 1996).

Na área do eletromagnetismo, pode-se citar como um dos precursores, embora não o primeiro, da utilização deste método, a publicação do artigo "Finite Elements Solution of Saturable Magnetic Fields Problems" por Silvester e Chari (1970) que propunham o desenvolvimento de soluções de eletromagnetismo através da utilização de método de elementos finitos, incluindo a resolução de problemas de comportamento não linear (SILVESTER e CHARI, 1970).

Até 1970, a resolução dos modelos matemáticos diferenciais de segunda ordem era considerada uma tarefa árdua e muito onerosa em matéria de custo computacional além de não serem totalmente satisfatórios para tais resolução.

2.2. A plataforma de desenvolvimento Dev C++

Para realizar o desenvolvimento do algoritmo proposto neste trabalho, foi utilizado o ambiente de desenvolvimento integrado e livre, Dev C++ 5 da Bloodshed Software (2011). Este ambiente de desenvolvimento pode desenvolver programas em linguagem C e em C++ e utiliza os compiladores livres (GNU) gcc e g++ respectivamente.

O Dev C++ pode ser obtido através do site do desenvolvedor pelo endereço eletrônico http://www.bloodshed.net/devcpp.html.

Para iniciar um novo projeto, basta seguir os seguintes passos:

Inicie o Dev C++, após o carregamento do Dev C++, selecione o menu **Arquivo** e em seguida **Novo** e por fim em **Projeto** como mostra a figura 2.

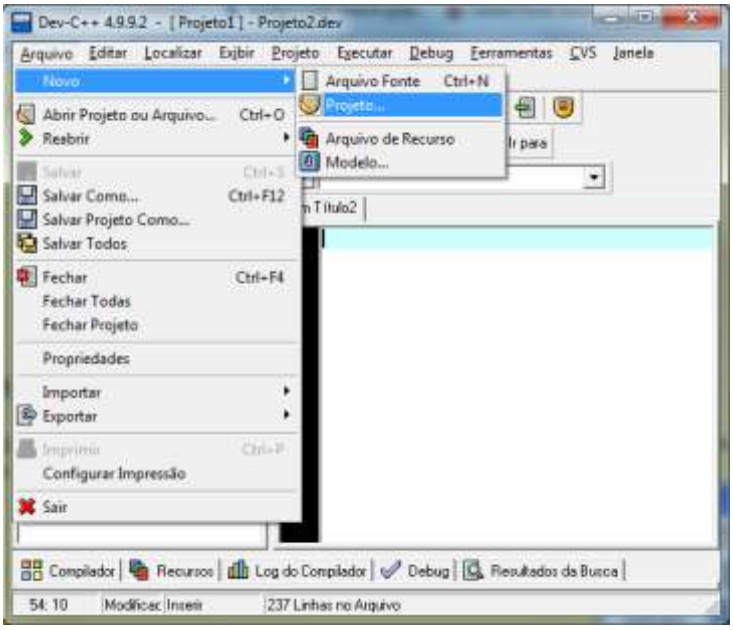

Figura 1 - Janela principal do Dev C++.

Após feito isso irá surgir uma nova janela, conforme a figura 3, onde é possível iniciar o novo projeto.

Figura 2 - Janela de inicio de novo projeto.

Para fazermos um programa do tipo "Olá mundo", selecione o item *Console Application*, de um nome a seu projeto e pressione o botão OK.

Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

Ao pressionar o botão, irá fechar a janela de novo projeto e você voltará a janela anterior conforme a figura 4.

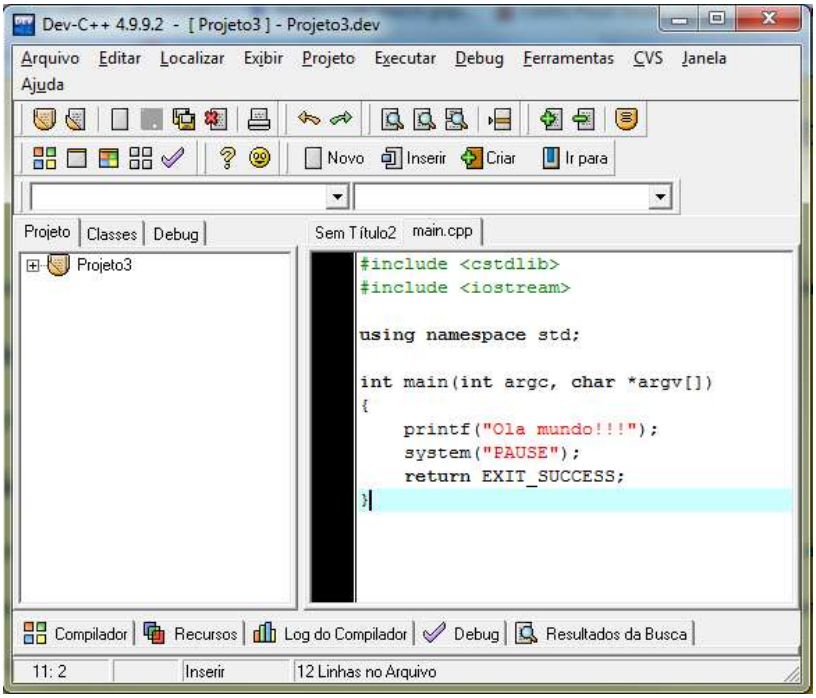

Figura 3 - Janela principal com o projeto já criado e o programa digitado.

Nela você poderá digitar o programa a seguir:

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main(int argc, char *argv[])
{
   printf("Ola mundo!!!");
   system("PAUSE");
   return EXIT_SUCCESS;
}
```

Para compilar e executar o programa clique no botão conforme a figura 5 ou pressione a tecla de função F9.

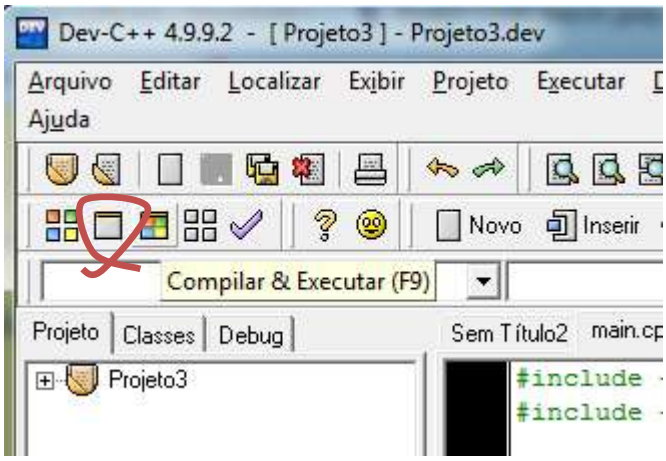

Figura 4 - Ilustração onde é realçado o botão para compilar e executar o programa.

A figura 6 mostra a tela que resultará da execução do programa Olá mundo.

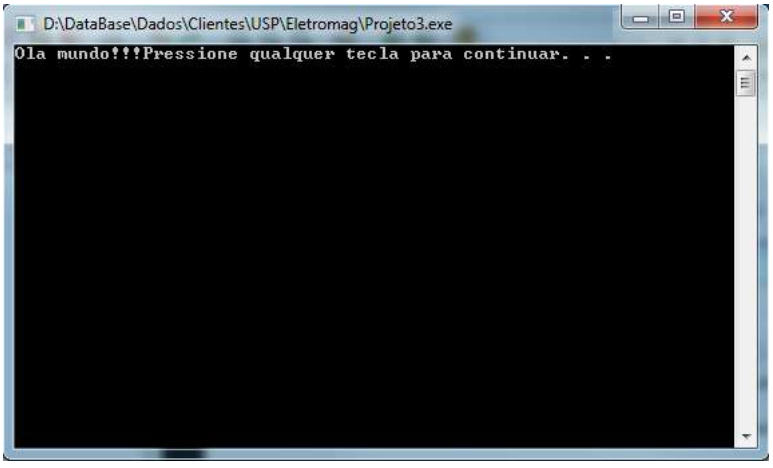

Figura 5 - Programa Olá mundo feito em Dev C++.

2.3. Método de diferenças finitas

O métodos de diferenças finitas é uma técnica para diferenciação de uma função com um dado conjunto finito de valores da variável dependente em determinados pontos conhecidos da variável independente.

O modelo matemático para calcular o declive é um cálculo discreto de dy/dx em um determinado ponto x_0. O resultado deste método é representado pela equação (1) e corresponde a uma formula de diferenças finitas para a resolução de uma diferenciação de primeira ordem (SILVESTER e FERRARI, 1996).

$$\frac{dy}{dx} \cong \frac{\Delta y}{\Delta x} = \frac{(y_1 - y_0)}{(x_1 - x_0)}$$

(1)

2.3. Método da Iteração

Em problemas de modelamento de distribuição de diferentes potenciais ao longo de uma superfície, um método de lápis-e-papel não é recomendado devido a inúmeras iterações que deve ser realizadas de forma a garantir um mínimo de precisão nos valores obtidos, isso tornaria muito oneroso o processo manual inviabilizando-o para problemas complexos (HAYT, 1974).

Para uma análise com grande precisão, faz-se necessário o uso de computadores e algoritmos iterativos. O método iterativo, descrito a seguir, é factível a nível computacional e traz grande precisão.

O método consiste em dividir uma dada região bidimensional com interações de potenciais de campo, em uma grade formada por quadrados de lados h conforme a figura 7.

Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

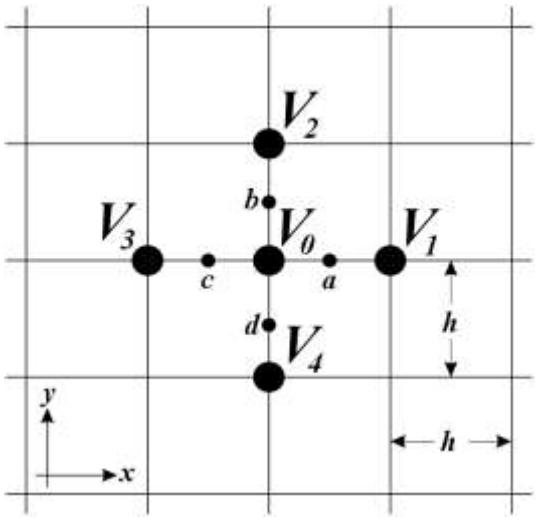

Figura 6 - Grade bidimensional. Ao centro o ponto de potencial ***Vo*** (HAYT, 1974).

O potencial V_0é aproximadamente igual à média aritmética de seus potenciais vizinhos na grade.

Assumindo que este problema proposto não varie em função de z, podemos definir então que o potencial de cada ponto adjacente ao ponto analisado V_0 são: V_1, V_2, V_3 e V_4. Se a região esta descarregada e contida em um meio dielétrico homogênio, temos $\boldsymbol{\nabla \cdot \vec{D} = 0}$ **e** $\boldsymbol{\nabla \cdot \vec{E} = 0}$**,** sendo assim temos, em duas dimensões:

$$\frac{\partial E_x}{\partial \mathrm{x}} + \frac{\partial E_y}{\partial \mathrm{y}} = 0$$

(2)

Com o gradiente de $E_x = -\partial V/\partial x$ e $E_y = -\partial V/\partial y$ obtemos o Laplaciano da equação em duas dimensões,

$$\nabla^2 \mathbf{V} = \frac{\partial^2 \mathrm{V}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{V}}{\partial \mathrm{y}^2} = 0$$

(3)

que equivale aproximadamente a:

$$\nabla^2 \mathbf{V} = \mathbf{V(x+1, y)} + \mathbf{V(x-1, y)} + \mathbf{V(x, y-1)} + \mathbf{V(x, y+1)} - \mathbf{4V(x, y)} = \mathbf{0}$$

(4)

correspondendo a uma convolução de **V** com a seguinte mascara (STRANG, 2008):

$$\begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

(5)

Aproximando-se os valores das derivadas parciais temos:

$$\left.\frac{\partial V}{\partial \mathrm{x}}\right|_{\mathrm{a}} \cong \frac{V_1 - V_0}{\mathrm{h}}$$

(6)

e

$$\left.\frac{\partial V}{\partial \mathrm{x}}\right|_{\mathrm{c}} \cong \frac{V_0 - V_3}{\mathrm{h}}$$

(7)

Com a aplicação de (7) e (6) na parcela em x equação (3) obtemos:

$$\left.\frac{\partial^2 \mathrm{V}}{\partial \mathrm{x}^2}\right|_{0} \cong \frac{\left.\frac{\partial V}{\partial \mathrm{x}}\right|_{\mathrm{a}} - \left.\frac{\partial V}{\partial \mathrm{x}}\right|_{\mathrm{c}}}{\mathrm{h}} \cong \frac{V_1 - V_0 - V_0 + V_3}{\mathrm{h}^2}$$

(8)

A análise discreta do potencial no eixo y é similar e temos:

$$\left.\frac{\partial^2 \mathrm{V}}{\partial \mathrm{y}^2}\right|_{0} \cong \frac{\left.\frac{\partial V}{\partial \mathrm{y}}\right|_{\mathrm{b}} - \left.\frac{\partial V}{\partial \mathrm{y}}\right|_{\mathrm{d}}}{\mathrm{h}} \cong \frac{V_2 - V_0 - V_0 + V_4}{\mathrm{h}^2}$$

(9)

Combinando a diferenciação de segunda ordem em duas dimensões, temos:

$$\nabla^2\mathbf{V} = \frac{\partial^2 \mathrm{V}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{V}}{\partial \mathrm{y}^2} \cong \frac{V_1 + V_2 + V_3 + V_4 - 4V_0}{\mathrm{h}^2} = 0 \tag{10}$$

ou simplesmente como em (4), temos :

$$V_0 = \frac{V_1 + V_2 + V_3 + V_4}{4} \tag{11}$$

3. Desenvolvimento do algoritmo proposto

3.1. Identificação da necessidade

A necessidade pode ser desde um vago descontentamento até a percepção de algo que não está bem (SHIGLEY, 1984).

Com base na atividade proposta na Seção **Erro! Fonte de referência não encontrada.**, faz necessário analisar um problema, no domínio do eletromagnetismo, de linhas de transmissão paralelas em fita e carregadas com potenciais distintos e representadas em corte transversal.

Segundo HAYT, (1974) não se justifica o uso de método iterativo manual (papel caneta), visto o elevado custo horas-homem e a baixa precisão e com base no problema proposto, surge à necessidade de se desenvolver resolver de forma rápida e precisa tal analise.

3.2. Definição do problema

O problema é mais específico que a necessidade. Em geral o problema é o ponto onde devemos atuar para atender a necessidade (SHIGLEY, 1984).

O problema nesse caso se resume na falta de um programa para solucionar a análise do exercício proposto de forma rápida e precisa.

3.3. Síntese

Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

Para atender a necessidade apresentada na Seção 1.3, pode-se idealizar um algoritmo de duas formas:

Primeira – Programa com uma matriz 5x7 que apresente os resultados em uma tela texto.

Segunda – Programa com uma matriz 50x70 e que pela dificuldade de se apresentar os resultados do processo de iteração na forma de texto, o faça na forma gráfica, representando por símbolos e cores os potenciais em cada ponto, distretizando em 3500 elementos finitos ao em vez de apenas 35.

3.4. Seleção e otimização

Três pré-protótipos foram feitos e segue um pequeno estudo do funcionamento de cada um:

Proposta 1 – Utilizado uma planilha eletrônica para simular a iteração na matriz 5x7 e os resultados obtidos podem ser vistos de forma geral na nas Figura 8.

Figura 7 - Planilha para discretização do problema proposto.

Como pode ser observado na Figura 9, a matriz do lado esquerdo representa o estado inicial da solução, onde as duas fitas em corte transversal são carregadas com -10 e 10V respectivamente, já na planilha da direita inicia-se o processo de iteração, onde o potencial $V_{22} = -2{,}5$ é dado por:

Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

$$\nabla^2 V_{22} = \frac{\partial^2 V_{22}}{\partial x^2} + \frac{\partial^2 V_{22}}{\partial y^2} \cong \frac{V_{23} + V_{12} + V_{21} + V_{21} - 4V_{22}}{h^2} = 0$$

sendo assim,

$$V_{22} = \frac{V_{23} + V_{12} + V_{21} + V_{21}}{4} = \frac{-10 + 0 + 0 + 0}{4} = -2{,}5$$

Figura 8 - Realce das duas primeiras fazes da primeira época do algoritmo proposto.

A Figura 10 apresenta o resultado após 8 iterações.

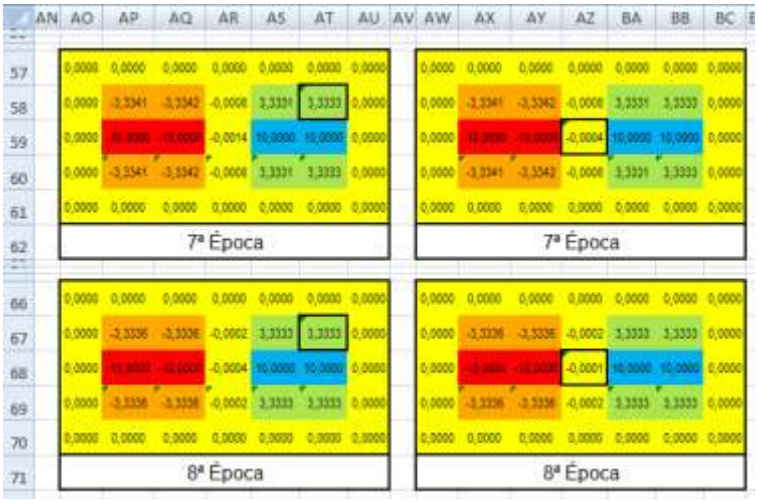

Figura 9 - Realce das quatro matrizes que representam as últimas fazes das duas últimas iterações.

Proposta 2 – Feito um algoritmo em C++ com o uso do ambiente de desenvolvimento DevC++ 5. O programa assim como a proposta 1, trabalha com matriz 5x7, utiliza tela texto para apresentar os resultados e pode ser visto na Figura 11. Nela se observa o estado inicial da matriz.

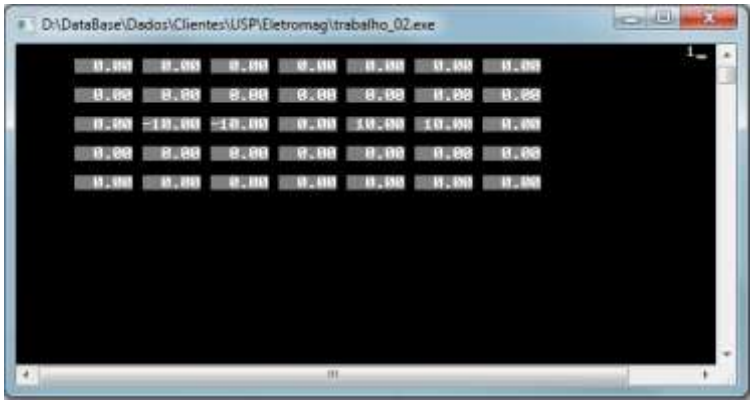

Figura 10 - Tela texto da proposta 2 feito em C++.

Na Figura 12 observa-se que o funcionamento do algoritmo segue o mesmo princípio da proposta 1 e o potencial também é $V_{22} = -2{,}5$.

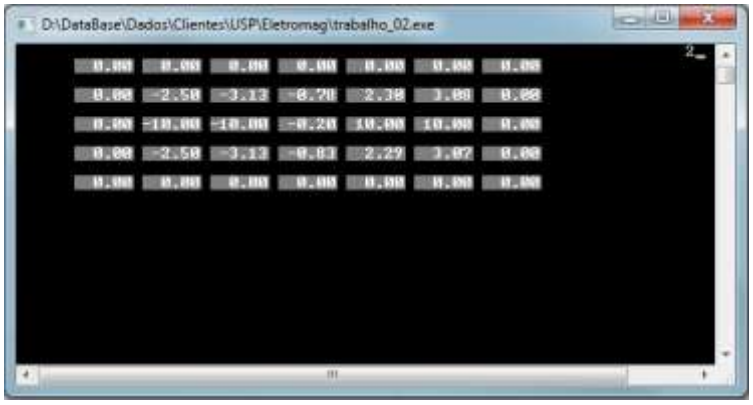

Figura 11 - Matriz após a primeira iteração completa.

Na Figura 13 apresenta-se o resultado do processo iterativo.

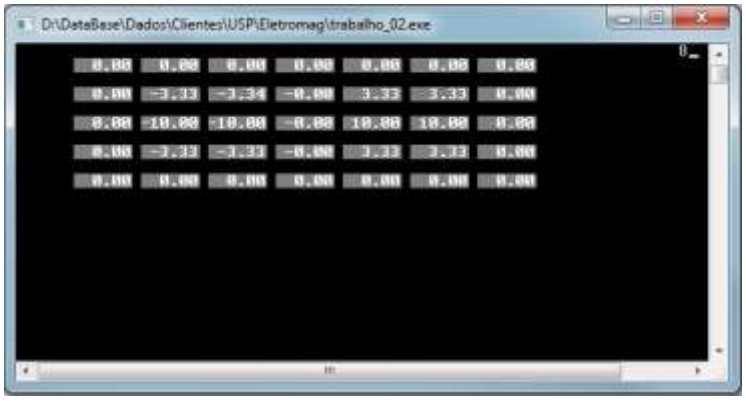

Figura 12 - Tela onde é apresentado o resultado da matriz após o final da 8ª iteração.

Proposta 3 - O algoritmo foi desenvolvido em C++ e assim como a proposta 2, foi utilizado o ambiente de desenvolvimento DevC++ 5. O programa utiliza o mesmo método de análise por diferenças finitas das propostas anteriores, porém se difere no tamanho da matriz que passa a ter 50x70 elementos discretos que pode ser visto na Figura 14 em seu estado inicial, dessa forma, o resultado passou a ser apresentado de forma gráfica, com cada ponto da matriz representado por símbolos específicos e variações de cores em 256 níveis diferentes.

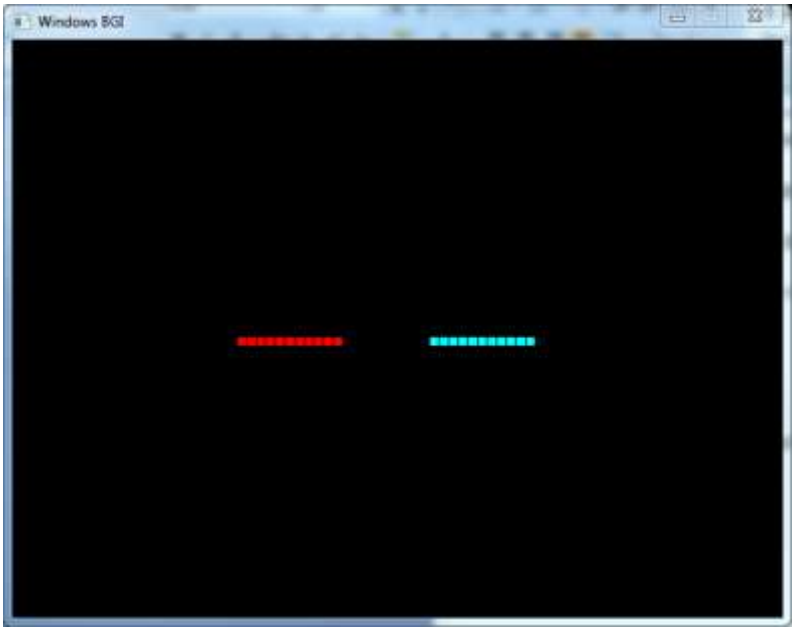

Figura 13 - Tela gráfica da proposta 3 com a representação da matriz 50x70 com os potenciais no instante inicial.

E na figura 15 é apresentado o resultado gráfico do processo de analise dos potenciais.

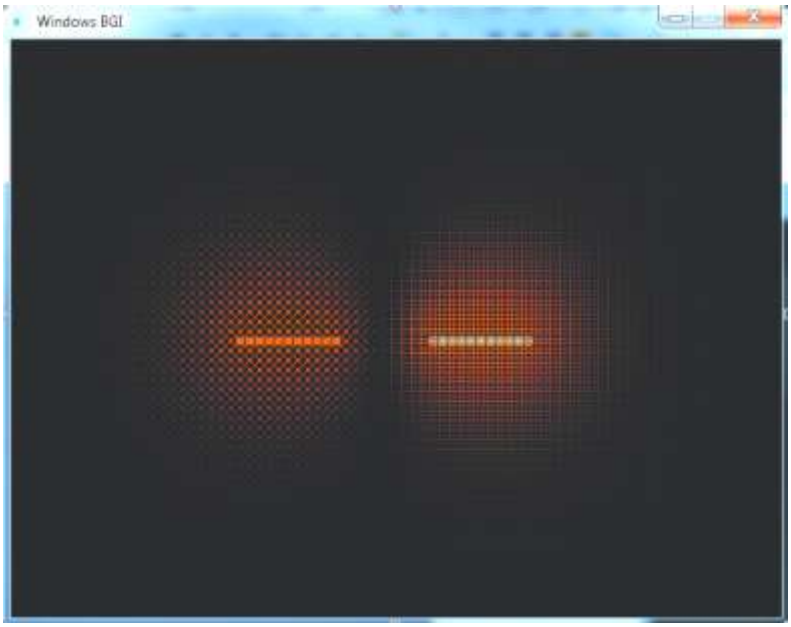

Figura 14 - Tela gráfica da proposta 3 com a representação da matriz 50x70 com os potenciais no instante final, após várias iterações, cerca de 300.

Com base nos resultados obtidos nestes 3 pré-protótipos, pode-se observar que o que melhor representou o estudo físico do fenômeno, foi a proposta 3, desta forma a mesma foi selecionada para o desenvolvimento e as duas demais propostas, foram descartadas em detrimento da seleção da proposta 3.

3.5. Detalhamento do projeto

A tela principal do programa desenvolvido solicita as seguintes informações:

- O valor do potencial P1
- A linha onde ele será inserido
- A coluna inicial para seu crescimento
- Seu comprimento

- O valor do potencial P2
- A linha onde ele será inserido
- A coluna inicial para seu crescimento
- Seu comprimento

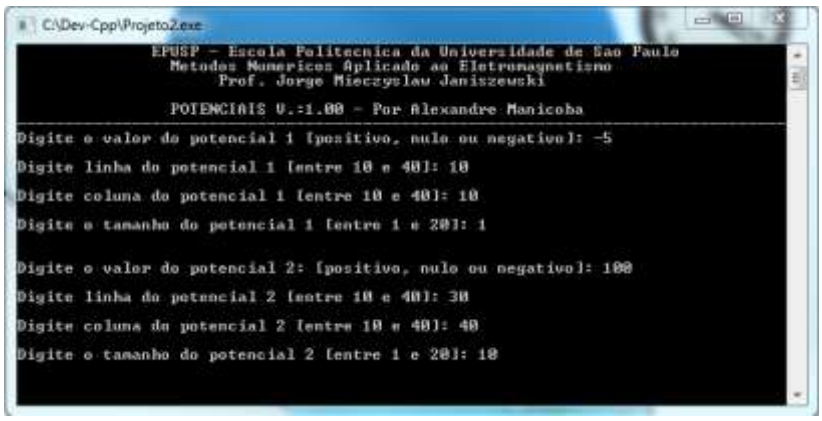

Figura 15 - Tela principal do programa desenvolvido co os dados dos potenciais sendo passados ao programa.

A Figura 16 ilustra a tela principal do programa desenvolvido co os dados dos potenciais sendo passados ao programa.

Na Figura 17 é apresentado a tela gráfica que exterioriza os valores de cada célula da matriz 50x70 onde valores negativos são representados por x, valores positivos são representados por quadrado com ponto ao centro e a intensidade dos valores de cada célula são representados pela variação na intensidade da cor vermelha de 0 (preto) a 256 (vermelho intenso).

Figura 16 - Tela gráfica onde são apresentados os pontos da matriz.

A Figura 18 mostra o resultado da análise onde ao potencial 1 foi atribuído a ddp de -5V e ao potencial 2 foi atribuída a ddp de 100V, observa-se assim que o algoritmo funciona de forma satisfatória.

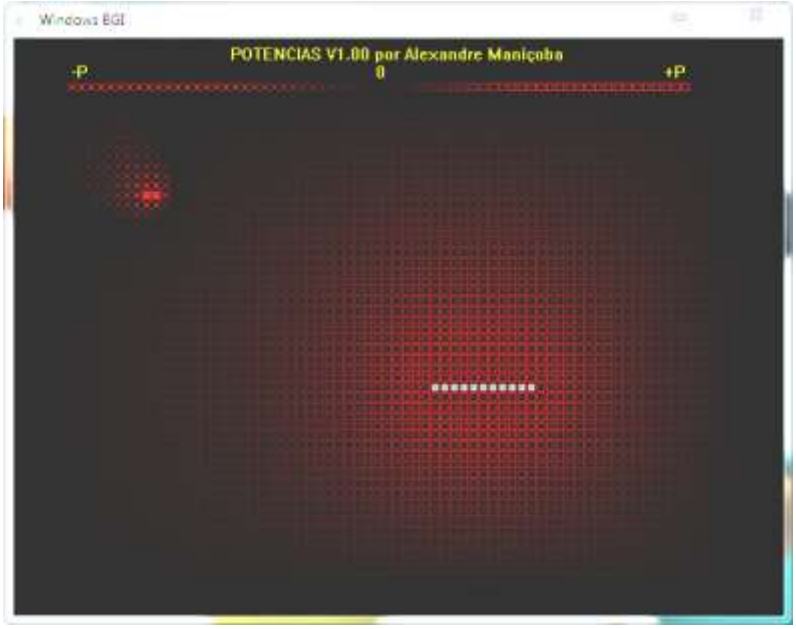

Figura 17 – Tela gráfica onde são apresentados os pontos da matriz após 300 iterações.

Segue o código fonte do programa proposto:

```
//Inclusão das bibliotécas utilizadas
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include <stdlib.h>

//Definições utilizadas

#define RWG 50 //Linhas da matriz gráfica
#define CLG 70 //Colunas da matriz gráfica

// Código principal
int main()
{
  // Declaração das variaveis
  int driver, mode;// variaveis usadas para modo gráfico
  int PN, PP, XPN, YPN, LPN,YPP, XPP, LPP,CX,CY;   // Variaveis utilizadas nos potenciais
  int rowG,colG, layer,layer1,layer2,i,j,l; // contadores
  int ll;  // variáveis a serem usadas ao longo do programa

  float k,kk; // Variáveis auxiliares
  float AG[RWG+10][CLG+10][2]; // Matriz 3D - Utilizada para armazenar os potenciais
  float BG[RWG+10][CLG+10]; //Matriz 2D - Utilizada para guardar a posição das celulas que
possuem potenciais diferentes de zero

  char exit; // Variável utilizada no loop

  // Atribuição dos valores iniciais das variáveis
  PN=-10; // Normalizador do potencial do ponto 1
  PP=10; // Normalizador do potencial do ponto 2
  CX=30; // Centralizador da matriz gráfica em x
  CY=50; // Centralizador da matriz gráfica em y
  exit = 'n';
  k=0;
  layer=0;  //Contador de camada
  ll=3;  // Raio do x e do quadrado a ser plotado

  detectgraph(&driver, &mode);   // Obtém o driver gráfico e o modo  de acordo com o hardware
  mode=2; // Força o modo SVGA 680x480x32bits
  initgraph(&driver, &mode, "");  // inicializa  modo gráfico de acordo com os parâmetros
obtidos
  ////////////////////////////////////////////////////////////////////////////
  // Rotina que monta a tela texto e recebe os valores dos potenciais
  ////////////////////////////////////////////////////////////////////////////
  printf("              EPUSP - Escola Politecnica da Universidade de Sao Paulo\n");
  printf("                Metodos Numericos Aplicado ao Eletromagnetismo\n");
  printf("                     Prof. Jorge Mieczyslaw Janiszewski\n\n");
  printf("                POTENCIAIS V.:1.00 - Por Alexandre Manicoba\n");
  printf("------------------------------------------------------------------------------");
  printf("Digite o valor do potencial 1 [positivo, nulo ou negativo]: ");
      scanf("%d",&PN);
      if (PN!=0) //Se o P1 for diferente de zero, pergunta as demais informações sobre o ponto
        {
          printf("\nDigite linha do potencial 1 [entre 10 e 40]: ");
          scanf("%d",&YPN);
          printf("\nDigite coluna do potencial 1 [entre 10 e 40]: ");
          scanf("%d",&XPN);
          printf("\nDigite o tamanho do potencial 1 [entre 1 e 20]: ");
          scanf("%d",&LPN);
       }
       else //Senão P1 é considerado ponto nulo
       {
        YPN=0;
        XPN=0;
        LPN=0;
       }
```

```
       printf("\n\nDigite o valor do potencial 2: [positivo, nulo ou negativo]: ");
       scanf("%d",&PP);
       if (PP!=0)//Se o P2 for diferente de zero, pergunta as demais informações sobre o ponto
        {
          printf("\nDigite linha do potencial 2 [entre 10 e 40]: ");
          scanf("%d",&YPP);
          printf("\nDigite coluna do potencial 2 [entre 10 e 40]: ");
          scanf("%d",&XPP);
          printf("\nDigite o tamanho do potencial 2 [entre 1 e 20]: ");
         scanf("%d",&LPP);
        }
       else //Senão P2 é considerado ponto nulo
        {
          YPP=0;
          XPP=0;
          LPP=0;
        }
       if((PN==0)&&(PP==0)) //Se P1 e P2 são nulos, são atribuidos os valores do exercício
solicitado pelo professor em sala de aula
        {
          PN=-10; //P1=-10V
          PP=10;  //P2=10V
          XPN=20; //Coluna inicial de P1
          YPN=25; //Linha de P1
          LPN=10; //Comprimento de P1
          XPP=40; //Coluna inicial de P2
          YPP=25; //Linha de P1
          LPP=10; //Comprimento de P2
          printf("\n
**************************************************************************");
          printf("\n  * Adotado os valores do exercício proposto em aula, P1=-10V e P2=10V
*");
          printf("\n
**************************************************************************");
        }
  ////////////////////////////////////////////////////////////////////////////
  // Rotina que limpa e prepara as matrizes
  ////////////////////////////////////////////////////////////////////////////

  //limpa as matrizes
  for(rowG=0;rowG<=RWG+5;rowG++)//Varre as linhas
    for(colG=0;colG<=CLG+5;colG++)//Varre as colunas
      {
        AG[rowG][colG][0]=0; //Limpa a primeira camada da matriz A
        AG[rowG][colG][1]=0; //Limpa a segunda camada da matriz A
        BG[rowG][colG]=0;    //Limpa a matriz B
      }
  //propara as matrizes
  for(colG=XPN;colG<=(XPN+LPN);colG++)//Varre as colunas ocupadas por P1
    {
      AG[YPN][colG][0]=PN; // Carrega a primeira camada da matriz A com P1
      AG[YPN][colG][1]=PN; // Carrega a segunda camada da matriz A com P1
      BG[YPN][colG]=1;     // Carrega a matriz B com o perfil de P1 (1 --> ponto<>0)
    }
   for(colG=XPP;colG<=(XPP+LPP);colG++)//Varre as colunas ocupadas por P2
    {
      AG[YPP][colG][0]=PP;// Carrega a primeira camada da matriz A com P2
      AG[YPP][colG][1]=PP;// Carrega a primeira camada da matriz A com P2
      BG[YPP][colG]=1;    // Carrega a matriz B com o perfil de P2 (1 --> ponto<>0)
    }
  ////////////////////////////////////////////////////////////////////////////
  //Rotina que imprime preliminarmente P1 e P2
  // Se o potencial for positivo, imprime em azul claro
  // Se o potencial for negativo, imprime em vermelho
  // Se o potencial for zero, imprime em preto
  ////////////////////////////////////////////////////////////////////////////
   for(rowG=1;rowG<=RWG+1;rowG++)//Varre as linhas
     {
        for(colG=1;colG<=CLG+1;colG++)//Varre as colunas
          {
            k=AG[rowG][colG][0]; //Carrega em k o valor da celula atual
            k=k*-25; //Normaliza k para um valor entre -250 e 250 (ajuste 10)
```

Periódico de Divulgação Científica da FALS
Ano VI - Nº XIII- JUL / 2012 - ISSN 1982-646X

```
            setcolor(int(k)); //Define a cor do pincel
            putpixel(CX+colG*8, CY+rowG*8,int(k)); //Plota um ponto
            for(l=1;l<=3;l++) //Amplia o raio l para crescer o solido quadrado
              rectangle((CX+colG*8-l),(CY+rowG*8-l),(CX+colG*8+l),(CY+rowG*8+l));
          }
      }

  ////////////////////////////////////////////////////////////////////////////
  // Rotina que desenha a escala de valores
  ////////////////////////////////////////////////////////////////////////////
  setcolor(YELLOW);  // cor de desenho ou escrita é amarela --  igual a setcolor(14)
  outtextxy( 20, 5,  "                                        POTENCIAS V1.00 por Alexandre
Maniçoba ");
  outtextxy( 20, 20, "       -P                                                    0
+P");
  // Desenha a escala
  for(i=0;i<=250;i=i+8)
      {
        //Desenha a variação negativa
        putpixel(40+i,40, 512-i);
        setcolor(512-i);
        rectangle((40+i-1),39,(40+i+1),41);
        line( (40+i-3),37,(40+i+3),43);
        line( (40+i+3),37,(40+i-3),43);
        //Desenha a variação positiva
        putpixel(40+270+i,40, 256+i);
        setcolor(256+i);
        rectangle(40+267+i,37,40+273+i,43);
      }
  ////////////////////////////////////////////////////////////////////////////
  // Rotina realiza a iteração da matriz
  ////////////////////////////////////////////////////////////////////////////
  do // Loop realizar n iterações na matriz
       {
          if(layer>=1) //se a camada escolhida for a 1 faz
            {
               layer1=0; //A camada que receberá a modificação é a camada 0 em função da
camada 1
               layer2=1; //A camada 1 será a camada modificadora da camada 0
               layer=-1; //Reseta o contador de camada
            }
           else  // Senão daz
            {
               layer1=1; //A camada que receberá a modificação é a camada 1 em função da
camada 0
               layer2=0; //A camada 0 será a camada modificadora da camada 1
            }
              layer++; //Incrementa o contador de camada
          ////////////////////////////////////////////////////////////////////////////
          // Rotina de varreduta da matriz
          ////////////////////////////////////////////////////////////////////////////
          for(rowG=1;rowG<=RWG;rowG++) // Varre as linhas da matriz
          {

              for(colG=1;colG<=CLG;colG++) // Varre as colunas da matriz
              {
                if(BG[rowG][colG]==0) //Se o valor da celula atual de B for igua a 0 faz a
media dos 4 pontos vizinhos
                  AG[rowG][colG][layer1]=(AG[rowG-
1][colG][layer2]+AG[rowG][colG+1][layer2]+AG[rowG+1][colG][layer2]+AG[rowG][colG-
1][layer2])/4;
              k=AG[rowG][colG][layer1]; //Atribui o valor da celula atual da matriz A a
variável k
              kk=k; // Duplicata de k

              if(k<=0) //Se o valor for negativo ou nulo faz
              {
                k=k*(256/PN)+257; // Normaliza o modulo do valor de k para algo entre 256 e
512
              }
              else //Se não,
              {
```

```
                  k=k*(256/PP)+257;; // Normaliza o valor de k para algo entre 256 e 512
                }

                if(k>510) k=510; // Mantem o valor de k abaixo de 510
                setcolor(int(k)); // Define a cor do pincel
                putpixel(CX+colG*8, CY+rowG*8, int(k)); // Imprime o ponto referente a celula
atual cuja cor esta em função do módulo do potencial

                if(kk>0) // Se o valor da celula atual for positivo faz
                {
                 rectangle((CX+colG*8-ll),(CY+rowG*8-ll),(CX+colG*8+ll),(CY+rowG*8+ll));   //
Imprime o quadradi com raio ll
                }
                else // Senão,
                {
                rectangle((CX+colG*8-1),(CY+rowG*8-1),(CX+colG*8+1),(CY+rowG*8+1));  //
Imprime um quadrado de raio 1
                // Imprime o X com raio l
                line( (CX+colG*8-ll),(CY+rowG*8-ll),(CX+colG*8+ll),(CY+rowG*8+ll));
                line( (CX+colG*8+ll),(CY+rowG*8-ll),(CX+colG*8-ll),(CY+rowG*8+ll));
                }

              }


        }

    } while (exit != 's');
  return 0;
  }
// Fim do algorítimo
```

4. Conclusão

Com base no levantamento bibliográfico realizado, pode-se desenvolver este algoritmo que utiliza o método de diferenças finitas, validando assim seu funcionamento e contribuindo de forma significativa para resolução de problemas no domínio do eletromagnetismo através da aplicação destas técnicas.

O programa idealizado rodou de forma satisfatória nos testes e apresentou bons resultados, ou seja, resultados legíveis, robustos e confiáveis ao longo de seus testes.

A técnica de diferenças finitas aplicada ao domínio do eletromagnetismo, demonstra ser muito robusta para a análise de matrizes com grande número de elementos.

Referências

BASTOS, João Pedro Assumpção. **"Eletromagnetismo e Cálculo de Campos"**, 3ª Edição, Ed. Da UFSC. 1996.

BLOODSHEED Software, **Dev C/C++ 5**, Disponivel em: <http://www.bloodshed.net/devcpp.html> e Acessado em 04/07/2011.

HAYT, William H. Jr. **"Engineering Electromagnetics"**, 3rd Edition, Ed. McGraw-Hill, 1974.

JANISZEWSKI, Jorge Mieczyslaw. **"Metodos Numericos Aplicados ao Eletromagnetismo"**. Notas de aula. Escola Politécnica da Universidade de São Paulo. Julho de 2011.

MANIÇOBA, A. O. **"PROJETO DE UMA MÁQUINA PARA SEPARAÇÃO E DIGITALIZAÇÃO DE GRÃOS DE MILHO (ZEA MAYS L.)."** 2008. 129 fl. Trabalho de Graduação em Engenharia – Universidade Católica de Santos, Santos, 2008.

SHIGLEY, Joseph Edward. **"Elementos de máquinas"**; Volume 1. 3ª.ed. Rio de Janeiro: LTC - Livros técnicos e científicos editora S.A., 1984. 347 p.

SILVESTER, Peter. P. e CHARI, M. V. K. **"Finite Element Solution of Saturable Magnetic Field Problems"**, IEEE Transactions. Vol. PAS-89 Issue:7, pp. 1642-1651, 1970.

SILVESTER, Peter P. e FERRARI, Ronald L. **"Finite Elements for Electrical Engineers"**, 3rd Edition, Ed. Cambridge. 1996.

STRANG, Gilbert. **"Mathematical Methods for Engineers I"**,Notes of Lecture 10, MIT Open Courseware. Massachusetts Institute of Technology. Disponível em: <http://www.youtube.com/watch?v=V5EjSvx1vw0> Acessado entre: 23/06/2011 a 4/07/2011.