

```

package mat;
import java.util.*;
import javax.swing.JOptionPane;

/**
 * Classe criada com o intuito de lidar com os cálculos necessários no
 programa
 * @author Danyel Clinário dos Santos
 * @email danyel.clinario@poli.ufrj.br
 */

public class MatFunc {

    public static double[] multiply_matrix2(double[][] a, double[] b) {

        double[] c=new double[3];

        for(int i=0;i<a.length;i++){
            c[i]=0;
            for(int k=0;k<a.length;k++){
                c[i]+=a[i][k]*b[k];
            }
        }
        return c;
    }

    public static double[] vet_func(double teta1, double teta2,
double[] x) { // x é aproximação da resposta
        double y[]= new double[3]; // c3 vira x[1]
        y[0]=2*Math.pow(x[1], 2)+Math.pow(x[0], 2)+6*Math.pow(x[2],
2)-1;
        y[1]=8*Math.pow(x[1], 3)+6*x[1]*Math.pow(x[0],
2)+36*x[1]*x[0]*x[2]+108*x[1]*Math.pow(x[2], 2)-teta1;
        y[2]=60*Math.pow(x[1], 4)+60*Math.pow(x[1], 2)*Math.pow(x[0],
2)+576*Math.pow(x[1], 2)*x[0]*x[2]+
        2232*Math.pow(x[1], 2)*Math.pow(x[2],
2)+252*Math.pow(x[2], 2)*Math.pow(x[0], 2)+1296*Math.pow(x[2], 3)+
        3348*Math.pow(x[2], 4) +24*Math.pow(x[0],
3)*x[2]+3*x[0]-teta2 ;

        return y;
    }

    public static double[][] jacobiana(double[] x) {
        double y[][]= new double[3][3];
        y[0][0]=2*x[0];
        y[0][1]=4*x[1];
        y[0][2]=12*x[2];
        y[1][0]=12*x[1]*x[0]+36*x[1]*x[2];

        y[1][1]=24*Math.pow(x[1],2)+6*Math.pow(x[0],2)+36*x[1]*x[2]+108*Mat
h.pow(x[2],2);
        y[1][2]=36*x[1]*x[0]+216*x[1]*x[2];
    }
}

```

```

        y[2][0]=120*Math.pow(x[1],2)*x[0]+576*Math.pow(x[1],2)*x[2]+504*Math.
h.pow(x[2],2)*x[0]+
                1296*Math.pow(x[2],3)+72*Math.pow(x[0],2)*x[2]+3;

        y[2][1]=240*Math.pow(x[1],3)+120*x[1]*Math.pow(x[0],2)+1152*x[1]*x[
0]*x[2]+4464*x[1]*Math.pow(x[2],2);

        y[2][2]=576*Math.pow(x[1],2)*x[0]+4464*Math.pow(x[1],2)*x[2]+504*x[
2]*Math.pow(x[0],2)+3888*Math.pow(x[2],2)*x[0]+
                13392*Math.pow(x[2],3)+24*Math.pow(x[0],3);

        return y;
    }

```

```

    public static boolean pivotamento(int posicao, double[][] A,
double[] b) {
        /**
         * Função de pivotamento necessária para algoritmos como
Gauss e Decomposição LU
         * @param A é a matriz dos coeficientes do sistema
         * @param b é a vetor de coeficientes dos termos
independentes
         * @param posicao é um auxiliar para o loop, indicando a
linha atual do pivo
         * @return uma variavel booleana que indica se foi necessário
fazer o pivotamento
         */

        boolean troca = false;
        // inicializando a variavel como false
        double pivo = A[posicao][posicao];
        // matriz utilizada para auxiliar no
pivotamento
        int linha_pivo = posicao;
        for (int i = posicao+1; i < b.length; i++) {
            if (Math.abs(A[i][posicao]) > Math.abs(pivo)){
                pivo = A[i][posicao];
                linha_pivo = i;
            }
        }
        if (posicao!= linha_pivo) {
            double[] temp = A[posicao];
            // variaveis temporarias para a troca de valores
            double[] tmp = A[linha_pivo];
            A[linha_pivo] = temp;
            A[posicao] = tmp;

            double temp_2 = b[posicao];
            // variaveis temporarias para a troca de valores
            double tmp_2 = b[linha_pivo];
            b[linha_pivo] = temp_2;
            b[posicao] = tmp_2;
            troca = true;
        }
    }

```

```

        System.gc();
        // Chamando garbage collector para liberar
        espaço
        return troca;
    }

    public static double[] subsretro(double[][] A, double[] b) {
        /**
         * Função de substituição retroativa usada em algoritmos como
         Decomposição LU e Cholesky
         * @param A é a matriz dos coeficientes do sistema, na
         Decomposição LU, representa a matriz U, por exemplo
         * @param b é a vetor de coeficientes dos termos
         independentes, na Decomposição LU, representa o vetor y, por exemplo
         * @return vetor resposta x
         */

        double[] x = new double[b.length];
        //Substituições Retroativas
        for (int i = (b.length-1); i > -1; i--) {
            double Somatorio = 0;
            for (int j = i+1; j < b.length; j++) {
                Somatorio= Somatorio+ (A[i][j] * x[j]);
            }
            x[i] = (b[i]-Somatorio)/A[i][i];
        }
        System.gc();
        //
        Chamando garbage collector para liberar espaço
        return x;
    }

```

```

    public static double[] subssuc(double[][] A, double[] b) {
        /**
         * Função de substituição sucessiva usada em algoritmos como
         Decomposição LU e Cholesky
         * @param A é a matriz dos coeficientes do sistema, na
         Decomposição LU, representa a matriz L, por exemplo
         * @param b é a vetor de coeficientes dos termos
         independentes, na Decomposição LU, representa o vetor b, por exemplo
         * @return vetor resposta x
         */

        double[] x = new double[b.length];
        //Substituições Sucessivas
        for (int i = 0; i < b.length; i++) {
            double Somatorio = 0;
            for (int j = 0; j < i; j++) {
                Somatorio= Somatorio+ (A[i][j] * x[j]);
            }
            x[i] = (b[i]-Somatorio)/A[i][i];
        }
    }

```

```

        System.gc();
    }

    Chamando garbage collector para liberar espaço
    return x;
}

public static double[] declu(double[][] A ,double[] b) { // b é
f(x)
    /**
     * Função de Decomposição LU
     * @param A é a matriz dos coeficientes do sistema
     * @param b é a vetor de coeficientes dos termos
independentes
     * @return um objeto do tipo Resposta que armazena a matriz
A, o vetor b, a matriz L, a matriz U, o vetor x e o determinante da
matriz lida.
     */

    double[][] L= new double [3][3];          // criando matriz L
    for(int i =0; i< 3; i++) {
        //transformando L em matriz identidade
        L[i][i]=1;
    }
    for(int k=0; k<3-1;k++) {
        //passando por cada etapa
        for(int i =k+1;i<3; i++) {
            // passando por cada linha
            boolean troca = pivotamento(k,A,b);
            double m = (-1)*(A[i][k]/A[k][k]);
            //fator m importante para L
            L[i][k]=-m;
            for (int j= k+1; j<3; j++ ) {
                // atualiza linha i, olhando cada coluna j
                A[i][j]= m * A[k][j] + A[i][j];
            }
            A[i][k] = 0;
        }
    }
    // por fim
    zera o elemento
    }
    }
    double det = 1;
    for (int i = 0; i<L[0].length; i++) {
        det*=A[i][i];
    }
    // É a
matriz "U" nesse caso
    }
    if(det != 0 && !(Double.isNaN(det))) {
        // se o determinante for diferente de 0 e válido (
        não é NaN), segue o algoritmo
    }
    }else {
        JOptionPane.showMessageDialog(null,"ERRO - SISTEMA
SINGULAR.");
        return null;
    }
}

```

```

    }

    double[] y = subssuc(L, b);
    double[] x = subsretro(A, y);

    System.out.print("Vetor resposta : ");
    for (int i = 0; i<x.length; i++) {
        System.out.print(x[i]+" ");
    }
    System.out.println();
    System.out.println("Acabou a Resposta ae: ");
    System.gc();
//
    Chamando garbage collector para liberar espaço
    return x;
}

public static double norma_vet(double[] v, double[] x) {
    /**
     * Função para calculo da norma dos vetores
     * @param v vetor atual
     * @param x é o vetor anterior
     * @return norma dos vetores
     */
    double v1=0;
    double v2=0;
    for(int i =0; i<x.length; i++) {
        v1+= Math.pow(v[i],2);
        v2+= Math.pow(x[i],2);
    }
    v1=Math.sqrt(v1);
    v2=Math.sqrt(v2);
    if( v2==0) {
        JOptionPane.showMessageDialog(null,"ERRO - DIVISÃO POR
ZERO");
        return 0;
    }
    return v1/v2;
}

public static double[][] invert(double a[][])
{
    int n = a.length;
    double x[][] = new double[n][n];
    double b[][] = new double[n][n];
    int index[] = new int[n];
    for (int i=0; i<n; ++i)
        b[i][i] = 1;

    gaussian(a, index);

    for (int i=0; i<n-1; ++i)

```

```

        for (int j=i+1; j<n; ++j)
            for (int k=0; k<n; ++k)
                b[index[j]][k]
                    -= a[index[j]][i]*b[index[i]][k];

    for (int i=0; i<n; ++i)
    {
        x[n-1][i] = b[index[n-1]][i]/a[index[n-1]][n-1];
        for (int j=n-2; j>=0; --j)
        {
            x[j][i] = b[index[j]][i];
            for (int k=j+1; k<n; ++k)
            {
                x[j][i] -= a[index[j]][k]*x[k][i];
            }
            x[j][i] /= a[index[j]][j];
        }
    }
    return x;
}

```

```

public static void gaussian(double a[][], int index[]) {
    int n = index.length;
    double c[] = new double[n];

    for (int i=0; i<n; ++i) {
        index[i] = i;
    }

    for (int i=0; i<n; ++i) {
        double c1 = 0;
        for (int j=0; j<n; ++j){
            double c0 = Math.abs(a[i][j]);
            if (c0 > c1) c1 = c0;
        }
        c[i] = c1;
    }

    int k = 0;
    for (int j=0; j<n-1; ++j){
        double pi1 = 0;
        for (int i=j; i<n; ++i) {
            double pi0 = Math.abs(a[index[i]][j]);
            pi0 /= c[index[i]];
            if (pi0 > pi1) {
                pi1 = pi0;
                k = i;
            }
        }

        int itmp = index[j];
        index[j] = index[k];
        index[k] = itmp;
    }
}

```

```

        for (int i=j+1; i<n; ++i){
            double pj = a[index[i]][j]/a[index[j]][j];

            a[index[i]][j] = pj;

            for (int l=j+1; l<n; ++l) {
                a[index[i]][l] -= pj*a[index[j]][l];
            }
        }
    }
}

```

```

public static double[] newton(double[] pontos, double tolm) {
    /**
     * Função do Método de Newton para sistemas não lineares
     * @param pontos são os pontos usados no calculo da função
     * @param tolm é a tolerancia maxima
     * @return um vetor contendo as constantes c2 c3 e c4
     */
    int k=0;
    double[] x = new double[3];

    x[0]=1;

    while(k<1000) {
        k += 1;

        double[] temp = new double[3];
        double[][] J = new double [3][3];

        J=jacobiana(x);

        double[] F = new double [3];
        F= vet_func(pontos[0], pontos[1], x);

        double[] delta_x = new double[3];

        delta_x = multiply_matrix2(invert(J),F);

        for(int i =0; i<x.length; i++) {
            temp[i]=x[i]-delta_x[i];
        }
        for(int i =0; i<x.length; i++) {
            x[i]=temp[i];
        }
        if (norma_vet(delta_x, x) < tolm) {
            System.out.println("CONVERGIU");
            for(double i : x ) {
                System.out.print(i);
                System.out.print(" ");
            }
        }
    }
}

```

```

        return x ;
    }
}
JOptionPane.showMessageDialog(null,"AVISO - Não foi possível
convergir a um resultado adequado");
return null;
}

```

```

public static double[] broyden(double[] pontos, double tolm) {
    /**
     * Função do Método de Broyden para sistemas não lineares
     * @param pontos são os pontos usados no calculo da função
     * @param tolm é a tolerancia maxima
     * @return um vetor contendo as constantes c2 c3 e c4
     */
    System.out.println("Chegou ate a func broyden");
    int k=0;
    double[] x = new double[3];
    double[] Y = new double [3];
    x[0]=1;

    double[][] J= new double[3][3];
    for(int i =0; i<x.length; i++) {
        J[i][i]=1;
    }

    while(k<1000) {
        k += 1;
        double[] temp = new double[3];
        double[] temp2 = new double[3];
        double[][] temp3 = new double[3][3];

        double[] F = new double [3];
        F= vet_func(pontos[0], pontos[1], x);

        double[] delta_x = new double[3];

        //delta_x = multiply_matrix2(invert(J),F);
        delta_x=declu(J, F);

        for(int i =0; i<x.length; i++) {
            Y[i]= -F[i]; // y = fxk - fxk-1, vou escrever y=-
fk-1 e depois somar fxk
        }

        for(int i =0; i<x.length; i++) {
            temp[i]=x[i]-delta_x[i];
        }
        for(int i =0; i<x.length; i++) {
            x[i]=temp[i];

```



```

    }

    for(int i =0; i<x.length; i++) {
        Y[i]+=x[i]; // y += f(x)
    }

    if (norma_vet(delta_x, x) < tol) {
        System.out.println("CONVERGIU");
        for(double i : x ) {
            System.out.print(i);
            System.out.print(" ");
        }
        return x ;
    }

    //atualizar j aqui

    //temp2=-J*deltax
    temp2=multiply_matrix2(J, delta_x);
    for(int i =0; i<x.length; i++) {
        temp2[i]= -temp2[i];
    }
    //temp2+=y
    for(int i =0; i<x.length; i++) {
        temp2[i]+= Y[i];
    }

    //CORREÇÃO

    double[][] temp4 =new double[3][1];
    temp4[0][0]=temp2[0];
    temp4[1][0]=temp2[1];
    temp4[2][0]=temp2[2];

    double[][] temp5 =new double[1][3];
    temp5[0][0]=delta_x[0];
    temp5[0][1]=delta_x[1];
    temp5[0][2]=delta_x[2];

    temp3[0][0] = temp4[0][0]*temp5[0][0];
    temp3[0][1] = temp4[0][0]*temp5[0][1];
    temp3[0][2] = temp4[0][0]*temp5[0][2];
    temp3[1][0] = temp4[1][0]*temp5[0][0];
    temp3[1][1] = temp4[1][0]*temp5[0][1];
    temp3[1][2] = temp4[1][0]*temp5[0][2];
    temp3[2][0] = temp4[2][0]*temp5[0][0];
    temp3[2][1] = temp4[2][0]*temp5[0][1];
    temp3[2][2] = temp4[2][0]*temp5[0][2];

    //temp3 = temp3/quadrado+quadrado+quadrado de delta x
    for(int i =0; i<x.length; i++) {

```

```

        for(int j =0; j<x.length; j++) {
            temp3[i][j] /=
Math.pow(delta_x[0],2)+Math.pow(delta_x[1],2)+Math.pow(delta_x[2],2);
        }
    }
    //J += temp3;
    for(int i =0; i<x.length; i++) {
        for(int j =0; j<x.length; j++) {
            J[i][j] += temp3[i][j];
        }
    }
}
JOptionPane.showMessageDialog(null,"AVISO - Não foi possível
convergir a um resultado adequado");
System.out.println("FALHOU");
return null;
}
}

```

```

package telas;
import mat.MatFunc;
import mat.Resposta;

import java.awt.BorderLayout;
import java.awt.Button;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.filechooser.FileNameExtensionFilter;

import java.awt.Toolkit;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;
import java.awt.event.ActionEvent;
import javax.swing.JTabbedPane;
import java.awt.Color;
import java.awt.Font;
import java.awt.Image;

```

```

import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.JRadioButton;
import javax.swing.AbstractButton;
import javax.swing.DropMode;
import javax.swing.ButtonGroup;
import javax.swing.JSeparator;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import java.awt.Dimension;
import javax.swing.JScrollPane;
import javax.swing.ScrollPaneConstants;
import javax.swing.border.SoftBevelBorder;
import javax.swing.border.BevelBorder;
import javax.swing.ImageIcon;
import java.awt.Rectangle;

public class Tela_principal extends JFrame {
    private JPanel contentPane;
    private final ButtonGroup buttonGroup = new ButtonGroup();

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tela_principal frame = new Tela_principal();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public static void calc(String caminhosaida , String
npontos_string, String tolmax_string, String metodo) throws IOException {
        int size=0;
        double [] pontos = new double[3];
        /*try{
            size = Integer.parseInt(npontos_string);
            if(size==1) {
                JOptionPane.showMessageDialog(null,"ERRO! - São
necessários dois pontos para realizar o cálculo.");
                return;
            }
        }catch (Exception e) {
            //JOptionPane.showMessageDialog(null,"ERRO! - A entrada
'número de pontos' deve ser um número inteiro.");

```

```

        return;
    }*/
    String[] parts = npontos_string.split(",");
    for(int i = 0; i < 2; i++ ) {
        pontos[i]=Double.parseDouble(parts[i]);
        //System.out.printf("%.2f ", pontos[i]);
    }

    double tolm = Double.parseDouble(tolmax_string);

    FileWriter arquivo_saida = new FileWriter(caminhosaida);
    PrintWriter escritor = new PrintWriter(arquivo_saida);

    escritor.print("Parâmetros lidos (o1, o2) : ");
    escritor.printf("%.2f, %.2f", pontos[0],pontos[1]);
    escritor.println();

    //escritor.close();

    switch (metodo) {
    case "newton":
        double[] resp = MatFunc.newton(pontos,tolm);
        escritor.printf("Tolerância estabelecida: %f\n",tolm);
        escritor.println("Método de Newton.");
        escritor.printf("Valor encontrado para as constantes c2,
c3 e c4: %.3f, %.3f, %.3f", resp[0], resp[1], resp[2]);
        arquivo_saida.close();
        JOptionPane.showMessageDialog(null,"Arquivo gerado com
sucesso em: "+caminhosaida);
        break;

        case "broyden":
            double[] resp1 =MatFunc.broyden(pontos,tolm);
            escritor.printf("Tolerância estabelecida: %f\n",tolm);
            escritor.println("Método de Broyden.");
            escritor.printf("Valor encontrado para as constantes c2,
c3 e c4: %.3f, %.3f, %.3f", resp1[0], resp1[1], resp1[2]);
            arquivo_saida.close();
            arquivo_saida.close();
            JOptionPane.showMessageDialog(null,"Arquivo gerado com
sucesso em: "+caminhosaida);
            break;

        }

    }

    /**
     * Create the frame.
     */
    public Tela_principal() {
        setResizable(false);

```

```

        setIconImage(Toolkit.getDefaultToolkit().getImage(Tela_principal.class.getResource("/imagens/logo1.jpg")));
        setTitle("Resolu\u00E7\u00E3o de Sistemas N\u00E3o Lineares");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 870, 339); // tamanho da tela

        JFileChooser openFileChooser;
        openFileChooser = new JFileChooser();
        openFileChooser.setCurrentDirectory(new File ("c:\\temp"));
        openFileChooser.setFileFilter(new
        FileNameExtensionFilter("arquivos em texto","txt"));

        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu mnAjuda = new JMenu("Ajuda");
        menuBar.add(mnAjuda);

        JMenuItem mntmManualDeInstrues = new JMenuItem("Manual de
        Instru\u00E7\u00F5es");
        mntmManualDeInstrues.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                Tela_ajuda telaajuda = new Tela_ajuda();
                telaajuda.setVisible(true);
            }
        });

        mntmManualDeInstrues.setHorizontalAlignment(SwingConstants.CENTER);
        mnAjuda.add(mntmManualDeInstrues);

        JMenu mnSobre = new JMenu("Sobre");
        menuBar.add(mnSobre);

        JMenuItem mntmSobreOPrograma = new JMenuItem("Sobre o
        programa");
        mntmSobreOPrograma.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Tela_sobre telasobre = new Tela_sobre();
                telasobre.setVisible(true);
            }
        });

        mntmSobreOPrograma.setHorizontalAlignment(SwingConstants.CENTER);
        mnSobre.add(mntmSobreOPrograma);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(new BorderLayout(0, 0));

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBackground(Color.LIGHT_GRAY);

```

```

contentPane.add(tabbedPane, BorderLayout.CENTER);

JLabel tolmax = new JLabel("Toler\u00E2ncia m\u00E9xima");
tolmax.setEnabled(true);
tolmax.setHorizontalTextPosition(SwingConstants.CENTER);
tolmax.setHorizontalAlignment(SwingConstants.CENTER);
tolmax.setFont(new Font("Segoe UI", Font.BOLD, 16));

JScrollPane scrollPane_texttol = new JScrollPane();

scrollPane_texttol.setVerticalScrollBarPolicy(ScrollPaneConstants.V
ERTICAL_SCROLLBAR_NEVER);

scrollPane_texttol.setHorizontalScrollBarPolicy(ScrollPaneConstants
.HORIZONTAL_SCROLLBAR_NEVER);
scrollPane_texttol.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));

JTextArea tolm = new JTextArea();
scrollPane_texttol.setViewportViewView(tolm);
tolm.setBorder(null);
tolm.setEnabled(true);
tolm.setDropMode(DropMode.INSERT);

ActionListener iterativoActionListener = new ActionListener()
{
    public void actionPerformed(ActionEvent actionEvent) {
        AbstractButton butjacobi = (AbstractButton)
actionEvent.getSource();
        AbstractButton butpot = (AbstractButton)
actionEvent.getSource();
    }
};

ActionListener decActionListener = new ActionListener() {
    public void actionPerformed(ActionEvent actionEvent) {
        AbstractButton butpot = (AbstractButton)
actionEvent.getSource();
    }
};

JPanel panel = new JPanel();
panel.setFont(new Font("Segoe UI", Font.PLAIN, 14));
panel.setBackground(Color.LIGHT_GRAY);
tabbedPane.addTab("Dados de Entrada", null, panel, null);

JLabel lblOrdemDaMatriz = new JLabel("Par\u00E2metros \u03B81
e \u03B82 (separados por v\u00E9rgula)");
lblOrdemDaMatriz.setToolTipText("");

```

```

        lblOrdemDaMatriz.setFont(new Font("Segoe UI", Font.BOLD,
16));

        lblOrdemDaMatriz.setHorizontalTextPosition(SwingConstants.CENTER);

        lblOrdemDaMatriz.setHorizontalAlignment(SwingConstants.CENTER);

        JLabel lblMtodoDeSolucao = new JLabel("M\u00E9todo para
c\u00E9lculo do valor aproximado");
        lblMtodoDeSolucao.setFont(new Font("Segoe UI", Font.BOLD, 17));

        lblMtodoDeSolucao.setHorizontalTextPosition(SwingConstants.CENTER);
        lblMtodoDeSolucao.setHorizontalAlignment(SwingConstants.LEFT);

        JRadioButton butnewton = new JRadioButton("M\u00E9todo de
Newton");
        butnewton.setFocusPainted(false);
        butnewton.setFont(new Font("Segoe UI", Font.PLAIN, 16));
        butnewton.setActionCommand("");
        butnewton.setBackground(Color.LIGHT_GRAY);
        buttonGroup.add(butnewton);

        JRadioButton butbroyden = new JRadioButton("M\u00E9todo de
Broyden");
        butbroyden.setFocusPainted(false);
        butbroyden.setFont(new Font("Segoe UI", Font.PLAIN, 16));
        butbroyden.setBackground(Color.LIGHT_GRAY);
        buttonGroup.add(butbroyden);

        butbroyden.addActionListener(iterativoActionListener);
        butnewton.addActionListener(decActionListener);

        JSeparator separator = new JSeparator();
        separator.setForeground(Color.GRAY);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBorder(new SoftBevelBorder(BevelBorder.LOWERED,
null, null, null, null));

        scrollPane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZON
TAL_SCROLLBAR_NEVER);

        scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_
SCROLLBAR_NEVER);

        JPanel panel_1 = new JPanel();
        panel_1.setFont(new Font("Segoe UI", Font.PLAIN, 14));
        panel_1.setBackground(new Color(192, 192, 192));
        tabbedPane.addTab("Dados de Sa\u00E9da", null, panel_1,
null);

        JLabel lblCaminhoParaSaida = new JLabel("Caminho para arquivo
de sa\u00E9da");

```

```

        lblCaminhoParaSaida.setHorizontalTextPosition(SwingConstants.CENTER
    );

    lblCaminhoParaSaida.setHorizontalAlignment(SwingConstants.LEFT);
    lblCaminhoParaSaida.setFont(new Font("Segoe UI", Font.BOLD,
17));

    JScrollPane scrollPane_2 = new JScrollPane();

    scrollPane_2.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZ
ONTAL_SCROLLBAR_NEVER);

    scrollPane_2.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICA
L_SCROLLBAR_NEVER);
    scrollPane_2.setBorder(new
SoftBevelBorder(BevelBorder.LOWERED, null, null, null, null));

    JLabel lblNewLabel = new JLabel("");
    lblNewLabel.setBackground(Color.WHITE);
    lblNewLabel.setBounds(new Rectangle(0, 0, 150, 200));
    ImageIcon icon = (new
ImageIcon(Tela_principal.class.getResource("/imagens/Minerva_UFRJ_Oficial
.png")));
    Image img = icon.getImage();
    Image imgScale =
img.getScaledInstance(lblNewLabel.getWidth(),
lblNewLabel.getHeight(), Image.SCALE_SMOOTH);
    ImageIcon scaledIcon = new ImageIcon(imgScale);
    lblNewLabel.setIcon(scaledIcon);

    JLabel Labelolho = new JLabel("");
    Labelolho.setVerticalAlignment(SwingConstants.TOP);
    Labelolho.setBackground(Color.WHITE);
    Labelolho.setBounds(new Rectangle(0, 0, 150, 150));
    ImageIcon icone = (new
ImageIcon(Tela_principal.class.getResource("/imagens/logol.jpg")));
    Image imge = icone.getImage();
    Image imgScaled =
imge.getScaledInstance(Labelolho.getWidth(),
Labelolho.getHeight(), Image.SCALE_SMOOTH);
    ImageIcon scaledIcane = new ImageIcon(imgScaled);
    Labelolho.setIcon(scaledIcane);

    JButton butPronto = new JButton("Pronto");

    GroupLayout gl_panel_1 = new GroupLayout(panel_1);
    gl_panel_1.setHorizontalGroup(
        gl_panel_1.createParallelGroup(Alignment.TRAILING)
            .addGroup(gl_panel_1.createSequentialGroup()
                .addContainerGap(231, Short.MAX_VALUE)
                .addComponent(lblNewLabel)
                .addGap(113)
                .addComponent(Labelolho)
            )
    );

```



```

        .addGap(205))
        .addGroup(Alignment.LEADING,
gl_panel_1.createSequentialGroup()
        .addContainerGap()
        .addComponent(lblCaminhoParaSaida,
GroupLayout.PREFERRED_SIZE, 258, GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(ComponentPlacement.UNRELATED)
        .addComponent(scrollPane_2,
GroupLayout.PREFERRED_SIZE, 419, GroupLayout.PREFERRED_SIZE)
        .addGap(18)
        .addComponent(butPronto)
        .addContainerGap(69, Short.MAX_VALUE))
    );
    gl_panel_1.setVerticalGroup(
        gl_panel_1.createParallelGroup(Alignment.LEADING)
        .addGroup(gl_panel_1.createSequentialGroup()
            .addGap(18)

            .addGroup(gl_panel_1.createParallelGroup(Alignment.TRAILING)

                .addComponent(lblCaminhoParaSaida, GroupLayout.PREFERRED_SIZE, 41,
GroupLayout.PREFERRED_SIZE)

                .addGroup(gl_panel_1.createSequentialGroup()

                    .addGroup(gl_panel_1.createParallelGroup(Alignment.BASELINE)

                        .addComponent(scrollPane_2, GroupLayout.PREFERRED_SIZE, 23,
GroupLayout.PREFERRED_SIZE)

                        .addComponent(butPronto))

                    .addGap(9)))

                .addGroup(gl_panel_1.createParallelGroup(Alignment.LEADING)

                    .addGroup(gl_panel_1.createSequentialGroup()

                        .addPreferredGap(ComponentPlacement.UNRELATED)

                        .addComponent(lblNewLabel, GroupLayout.PREFERRED_SIZE, 212,
Short.MAX_VALUE))

                    .addGroup(gl_panel_1.createSequentialGroup()

                        .addGap(21)

                        .addComponent(Labelolho)))

                    .addContainerGap())
            );

    JTextArea textpathsaida = new JTextArea();
    scrollPane_2.setViewportViewView(textpathsaida);
    textpathsaida.setDropMode(DropMode.INSERT);
    textpathsaida.setBorder(null);

```

```

panel_1.setLayout(gl_panel_1);

GridLayout gl_panel = new GridLayout(panel);
gl_panel.setHorizontalGroup(
    gl_panel.createParallelGroup(Alignment.LEADING)
        .addGroup(gl_panel.createSequentialGroup())

.addGroup(gl_panel.createParallelGroup(Alignment.LEADING)

.addGroup(gl_panel.createSequentialGroup())
                                .addGap(39)

.addComponent(lblOrdemDaMatriz)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(scrollPane, GroupLayout.PREFERRED_SIZE, 65,
GroupLayout.PREFERRED_SIZE)
                                .addGap(64)

.addComponent(tolmax)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(scrollPane_texttol, GroupLayout.PREFERRED_SIZE, 89,
GroupLayout.PREFERRED_SIZE))

.addGroup(gl_panel.createSequentialGroup())
                                .addGap(38)

.addComponent(lblMtodoDeSoluo)
                                .addGap(29)

.addComponent(butnewton)
                                .addGap(18)

.addComponent(butbroyden)

.addGroup(gl_panel.createSequentialGroup())
                                .addGap(20)

.addComponent(separator, GroupLayout.PREFERRED_SIZE, 777,
GroupLayout.PREFERRED_SIZE))
                                .addContainerGap(52, Short.MAX_VALUE))
    );
gl_panel.setVerticalGroup(
    gl_panel.createParallelGroup(Alignment.LEADING)
        .addGroup(gl_panel.createSequentialGroup())
            .addGap(31)

.addGroup(gl_panel.createParallelGroup(Alignment.BASELINE)

```

```

        .addComponent(lblOrdemDaMatriz, GroupLayout.PREFERRED_SIZE, 45,
        GroupLayout.PREFERRED_SIZE)
                                .addComponent(scrollPane,
        GroupLayout.PREFERRED_SIZE, 28, GroupLayout.PREFERRED_SIZE)
                                .addComponent(tolmax,
        GroupLayout.PREFERRED_SIZE, 45, GroupLayout.PREFERRED_SIZE)

        .addComponent(scrollPane_texttol, GroupLayout.PREFERRED_SIZE, 27,
        GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(ComponentPlacement.RELATED, 48, Short.MAX_VALUE)
                                .addComponent(separator,
        GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
                                .addGap(38)

        .addGroup(gl_panel.createParallelGroup(Alignment.BASELINE)

        .addComponent(lblMtodoDeSolucao, GroupLayout.PREFERRED_SIZE, 41,
        GroupLayout.PREFERRED_SIZE)
                                .addComponent(butnewton)
                                .addComponent(butbroyden,
        GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE))
                                .addGap(45))
    );

```

```

    JTextArea pontos = new JTextArea();
    scrollPane.setViewportViewView(pontos);
    pontos.setBorder(null);
    pontos.setWrapStyleWord(true);
    pontos.setMinimumSize(new Dimension(5, 20));
    pontos.setMaximumSize(new Dimension(5, 20));
    pontos.setColumns(1);
    pontos.setRows(1);

```

```

    pontos.setDropMode(DropMode.INSERT);
    panel.setLayout(gl_panel);

```

```

    ActionListener determinanteActionListener = new
    ActionListener() {
        public void actionPerformed(ActionEvent actionEvent) {
            AbstractButton butpot = (AbstractButton)
            actionEvent.getSource();
        }
    };

```

```

    ActionListener determinante_2ActionListener = new
    ActionListener() {
        public void actionPerformed(ActionEvent actionEvent) {
            AbstractButton butgauss= (AbstractButton)
            actionEvent.getSource();
        }
    };

```

```

        AbstractButton butjacobi = (AbstractButton)
actionEvent.getSource();
    }
};
butbroyden.addActionListener(determinante_2ActionListener);
butnewton.addActionListener(determinanteActionListener);

butPronto.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (butnewton.isSelected()) {
            try {
                calc(textpathsaida.getText(),
pontos.getText(), tolm.getText(), "newton" );
            } catch (IOException e1) {
                e1.printStackTrace();}
        }
        if (butbroyden.isSelected()) {
            try {
                calc( textpathsaida.getText(),
pontos.getText(), tolm.getText(), "broyden" );
            } catch (IOException e1) {
                e1.printStackTrace();}
        }
    }
});
}
}
}

```

```

package telas;
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JScrollBar;
import javax.swing.JScrollPane;
import javax.swing.border.EmptyBorder;
import java.awt.Toolkit;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import java.awt.Dimension;
import java.awt.Rectangle;
import java.awt.ScrollPane;
import java.awt.Component;
import javax.swing.JTextArea;
import java.awt.Font;
import javax.swing.ScrollPaneConstants;

```

```

public class Tela_ajuda extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tela_ajuda frame = new Tela_ajuda();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Tela_ajuda() {
        setResizable(false);
        //setResizable(false);
        setTitle("Ajuda");

        setIconImage(Toolkit.getDefaultToolkit().getImage(Tela_ajuda.class.
getResource("/imagens/logol.jpg")));
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setSize(500, 450);
        setLocationRelativeTo(null);
        getContentPane().setLayout(null);
        //contentPane = new JPanel();
        //contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        //setContentPane(contentPane);

        JLabel ajudalabel = new JLabel();
        ajudalabel.setVerticalTextPosition(SwingConstants.TOP);
        ajudalabel.setVerticalAlignment(SwingConstants.TOP);
        //ajudalabel.setBounds(new Rectangle(100, 100, 400, 300));
        ajudalabel.setAlignmentX(Component.CENTER_ALIGNMENT);
        ajudalabel.setToolTipText("");
        ajudalabel.setBorder(new EmptyBorder(5, 5, 5, 5));
        ajudalabel.setHorizontalTextPosition(SwingConstants.CENTER);
        ajudalabel.setHorizontalAlignment(SwingConstants.CENTER);
        ajudalabel.setFont(new Font("Segoe UI", Font.PLAIN, 14));
        ajudalabel.setText("<html> O programa gerado \u00E9 capaz de
resolver a equa\u00E7\u00E3o n\u00E3o linear apresentada <br> em aula
utilizando os dois m\u00E9todos aprendidos em aula:<br>\r\n- M\u00E9todo
de Newton<br>\r\n- M\u00E9todo de Broyden <br>\r\n<br>\r\n\r\nOs inputs
gerais do programa s\u00E3o:<br>\r\n- As constantes teta1 e
teta2<br>\r\n- O m\u00E9todo de solu\u00E7\u00E3o desejado<br>\r\n- O

```

caminho para o arquivo txt onde os outputs do programa ser\u00E3o escritos, <br>como a descri\u00E7\u00E3o os par\u00E2metros pontos lidos e o as constantes c2,c3 e c4.<br>\r\n\r\nPara que o programa funcione, <br> basta preencher as entradas corretamente e pressionar o bot\u00E3o \"pronto\" <br> na guia \"dados de sa\u00EDA\".<br> \r\n<br>\r\nOBS: <br> \r\n- O programa espera um \".\" para ler casas decimais. Por exemplo, escreva <br> \"0.5\" e n\u00E3o \"0,5\" (sem aspas). <br>\r\n- N\u00E3o separe as constantes com espa\u00E7o, apenas v\u00EDrgula, como : \"teta1,teta2\" <br>\r\n- \u00C9 recomendado preencher o caminho completo para os arquivos de <br> entrada e saida. Exemplo:

```
\"C:\\Users\\Myuser\\Desktop\\teste_entrada.txt\" <br>\r\n</html>");
        JScrollPane scroll = new JScrollPane(ajudalabel,
JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,ScrollPaneConstants.HORIZONTAL_SCROL
LLBAR_NEVER );
        scroll.setBounds(0, 0, 485, 413);
        scroll.setMinimumSize(new Dimension(500, 450));

        getContentPane().add(scroll);

    }
}
```

```
package telas;
```

```
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Font;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.SwingConstants;
import javax.swing.border.EmptyBorder;
import java.awt.Toolkit;
```

```
public class Tela_sobre extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tela_sobre frame = new Tela_sobre();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

        }
    }
});
}

/**
 * Create the frame.
 */
public Tela_sobre() {
    setResizable(false);
    setTitle("Sobre");

    setIconImage(Toolkit.getDefaultToolkit().getImage(Tela_sobre.class.
getResource("/imagens/logol.jpg")));
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);

    JLabel ajudalabel = new JLabel();
    //ajudalabel.setBounds(new Rectangle(100, 100, 400, 300));
    ajudalabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    ajudalabel.setToolTipText("");
    ajudalabel.setBorder(new EmptyBorder(5, 5, 5, 5));
    ajudalabel.setHorizontalTextPosition(SwingConstants.CENTER);
    ajudalabel.setHorizontalAlignment(SwingConstants.CENTER);
    ajudalabel.setFont(new Font("Segoe UI", Font.PLAIN, 14));
    ajudalabel.setText("<html> Programa desenvolvido pelo aluno
Danyel Clin\u00E9rio dos Santos, <br>utilizando o pacote de
desenvolvimento Java 11. O Programa <br> foi desenvolvido para a quarta
avalia\u00E7\u00E3o da disciplina \u00C1lgebra <br> Linear Computacional,
ministrada pelo Professor Lu\u00EDs Volnei <br> Sudati Sagrilo, no ano
de 2021, no per\u00EDodo 2020.2 da Faculdade <br>Federal do Rio de
Janeiro - UFRJ</html>");
    JScrollPane scroll = new JScrollPane(ajudalabel,
JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED, JScrollPane.HORIZONTAL_SCROLLBAR
_AS_NEEDED );
    scroll.setBounds(0, 0, 485, 413);
    scroll.setMinimumSize(new Dimension(500, 450));

    getContentPane().add(scroll);

}
}

```