Projeto UC-ESTRUTURAS MATEMÁTICAS

APLICAÇÃO DA TÉCNICA DE RECONSTRUÇÃO ALGÉBRICA PARA TOMOGRAFIA COMPUTADORIZADA

São Paulo
2024

Projeto UC-ESTRUTURAS MATEMÁTICAS

APLICAÇÃO DA TÉCNICA DE RECONSTRUÇÃO ALGÉBRICA PARA TOMOGRAFIA COMPUTADORIZADA

Nome dos Alunos:

Arthur Salatine de Moraes RA: 822151203
Eduardo Melo Maciel RA: 823127341
Felipe Pereira De Jesus RA: 823130181
Mariana Cardoso Brandão RA: 823146676
Victor Pinas Arnault RA: 82215768

São Paulo

2024

Sumário

Introdução

A tomografia computadorizada (TC) foi a primeira tecnologia de processamento de imagem de raios-x utilizada em diagnósticos médicos, que permite a visualização detalhada de estruturas internas do corpo humano. A qualidade das imagens obtidas por TC foi fundamental para um diagnóstico preciso e, para isso, os métodos de reconstrução de imagens desempenham um papel crucial. A reconstrução algébrica é um desses métodos, caracterizado por sua capacidade de produzir imagens de alta qualidade, mesmo com dados incompletos ou ruidosos. Este projeto visa explorar a aplicação da reconstrução algébrica na TC, comparando sua eficácia com outros métodos de reconstrução.

ART é um algoritmo iterativo utilizado para reconstruir uma imagem de suas projeções. Ela resolve um conjunto de equações lineares que relaciona uma imagem com suas projeções. Cada projeção refina a estimação da imagem para uma melhor correspondência com os dados da projeção.

Desenvolvimento

A técnica de reconstrução algébrica (ART, sigla em inglês) é uma classe de técnicas de resolução de um sistema sobredeterminado. Esse método é derivado de uma técnica iterativa produzida por Stefan Karczmarz, em 1937, e foi o método empregado na primeira máquina de tomografia computadorizada comercializada.

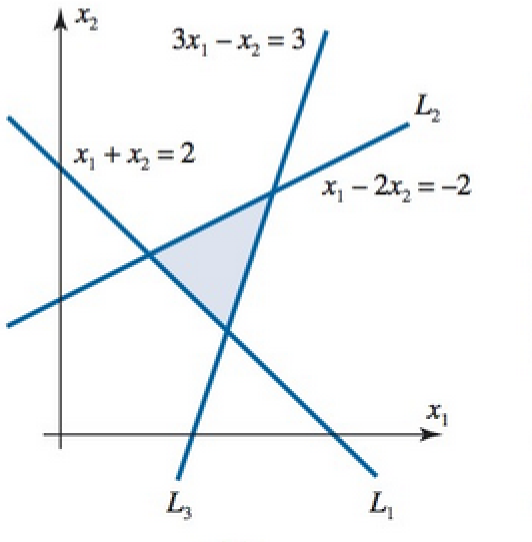
Para fins didáticos, considere o seguinte sistema de 3 equações em duas incógnitas:

$$L_1 : x_1 + x_2 = 2$$

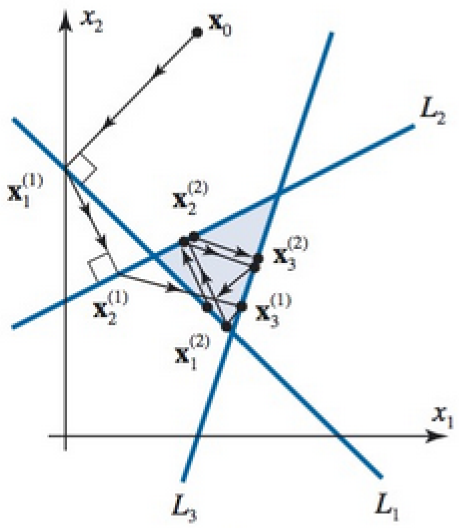
$$L_2 : x_1 - 2x_2 = -2$$

$$L_3 : 3x_1 - x_2 = 3$$

Geram as seguintes retas L_1, L_2 e L_3 no plano:



Na figura, vemos que as retas não tem um ponto comum entre si, logo não tem solução única, ou seja, não tem solução exata. Trata-se de um sistema possível e indeterminado. Todavia, os pontos em x_1 e x_2 do triângulo sombreado delimitado por essas três retas estão dentro da região do plano x_1x_2 e podem ser considerados como soluções aproximadas desse sistema. O procedimento iterativo parte da premissa de se ter soluções aproximadas dentro da condição de contorno posta pelo triângulo sombreado; ilustrado a seguir:



O procedimento iterativo é dado pelo seguinte algoritmo:

1. Escolhe-se algum ponto x_0 qualquer.

2. Faz-se a projeção ortogonal de x_0 sobre a reta L_1 e denotamos essa projeção por $x^{(1)}{}_1$. O expoente (1) indica que é a primeira iteração de uma sequência.

3. Projetamos $x_1^{(1)}$ ortogonalmente sobre a reta L_2 e denotamos essa projeção por $x_2^{(1)}$.

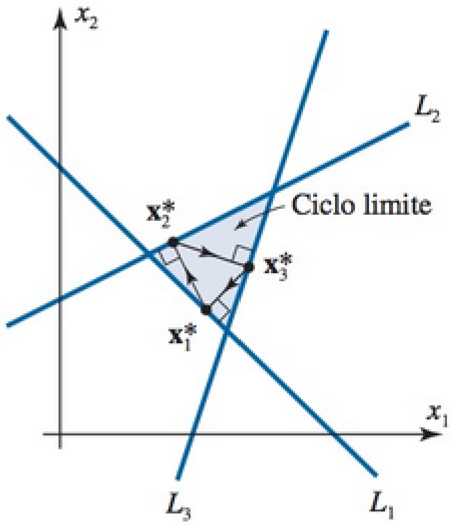
4. Projetamos $x^{(1)}{}_2$ ortogonalmente sobre a terceira reta L_3 e denotamos essa projeção por $x^{(1)}{}_3$.

5. Tomamos $x^{(1)}{}_3$ como ponto de origem da iteração e reinicia-se o processo e assim sucessivamente nos pontos em diante.

Esse algoritmos gera a seguinte sequência de pontos:

$$L_1 : x_1^{(1)}, x_1^{(2)}, x_1^{(3)}, \ldots$$
$$L_2 : x_2^{(1)}, x_2^{(2)}, x_2^{(3)}, \ldots$$
$$L_3 : x_3^{(1)}, x_3^{(2)}, x_3^{(3)}, \ldots$$

Que estão nas retas L_1, L_2 e L_3, respectivamente:



Vemos pela figura, que pode se se demonstrar que sempre que as três retas não forem paralelas, as sequências de pontos $x_i^{(*)}$ convergem em relação às retas. Esses pontos formam um ***ciclo limite*** do processo iterativo e podemos mostrar que o ciclo limite independe do ponto inicial x_0.

Em uma varredura de tomografia, as retas são os feixes de raios-x projetados no paciente enquanto o aparelho circunda o paciente deitado de frente a ele. A seguir, é mostrado o exemplo em código de uma simulação de varredura de tomografia computadorizada em linguagem de programação Java.

O plano x_1x_2 é dado por *gridSizeField*, pois o plano pode ser entendido como uma ‘grade de pontos’ aonde ficarão delimitadas as retas.

O tamanho da grade pode ser especificado pelo usuário. Assim como o número de iterações para o refinamento da imagem.

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

//Objeto de software CTScanSimulador
//Simulador de varredura de tomografia computadorizada
public class CTScanSimulator {

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new InputWindow());
    }
}
//Tela de inserção de dados pelo usuário
class InputWindow extends JFrame {
    private JTextField gridSizeField;
    private JTextField numProjectionsField;
    private JTextField iterationsField;
    private JTextField relaxationField;

```

```java
    public InputWindow() {
        setTitle("Simulador de Tomografia Computadorizada");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(5, 2, 10, 10));

        panel.add(new JLabel("Tamanho da grade (n):"));
        gridSizeField = new JTextField("4");
        panel.add(gridSizeField);

        panel.add(new JLabel("Número de projeções (m):"));
        numProjectionsField = new JTextField("8");
        panel.add(numProjectionsField);

        panel.add(new JLabel("Número de iterações:"));
        iterationsField = new JTextField("100");
        panel.add(iterationsField);

        panel.add(new JLabel("Fator de suavização:"));
        relaxationField = new JTextField("0.1");
        panel.add(relaxationField);

        JButton runButton = new JButton("Executar");
        runButton.addActionListener(new RunButtonListener());
        panel.add(runButton);

        add(panel);
        setVisible(true);
    }
```

Um outro conceito importante é o *relaxationField* ou ‘*fator de suavização*’. Este fator é o que ajusta a imagem obtida dentro da área triangular delimitada pelos feixes de raios-x e a aŕea triangular dentro do ciclo limite.

É este fator que determina a acuidade visual da imagem obtida.

Portanto, o algoritmo da ART é o seguinte:

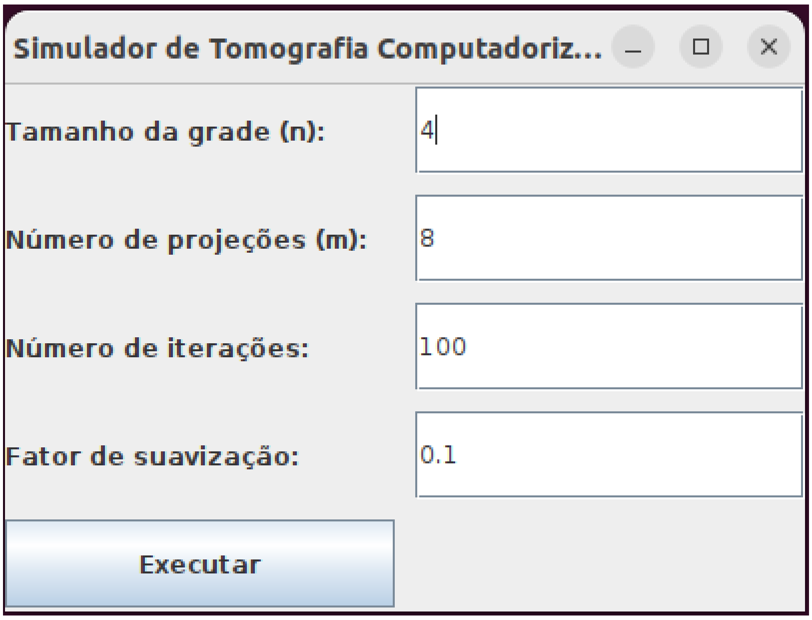
```java
    //Algoritmo da Técnica de Reconstrução Algébrica
    private double[] runART(int gridSize, int numProjections, int maxIterations, double relaxationFactor) {
        double[][] A = generateSystemMatrix(gridSize, numProjections);
        double[] b = generateProjectionData(numProjections);
        double[] x = new double[gridSize * gridSize];

        for (int iter = 0; iter < maxIterations; iter++) {
            for (int i = 0; i < numProjections; i++) {
                double aiDotX = 0.0;
                for (int j = 0; j < gridSize * gridSize; j++) {
                    aiDotX += A[i][j] * x[j];
                }
                double error = b[i] - aiDotX;
                for (int j = 0; j < gridSize * gridSize; j++) {
                    x[j] += relaxationFactor * error * A[i][j];
                }
            }
        }
        return x;
    }
```

```java
    // Matriz do sistema gerado pela varredura (CT Scan)
    private double[][] generateSystemMatrix(int gridSize, int numProjections) {
        double[][] A = new double[numProjections][gridSize * gridSize];
        for (int i = 0; i < numProjections; i++) {
            for (int j = 0; j < gridSize * gridSize; j++) {
                A[i][j] = (i + j) % 2;  // Padrão de exemplo
            }
        }
        return A;
    }
    //Dados projetados a partir das projeções ortogonais
    private double[] generateProjectionData(int numProjections) {
        double[] b = new double[numProjections];
        for (int i = 0; i < numProjections; i++) {
            b[i] = i + 1;  // Exemplo de dado projetado
        }
        return b;
    }
```

E a tela mostrando o resultado depois das iterações especificadas e dos dados inseridos é a seguinte:

```java
    //Tela de resultado
    //Imagem obtida em formato matricial
    private void displayResult(double[] image, int gridSize, int iterations) {
        resultArea.append("Imagem Reconstruída após " + iterations + " iterações:\n\n");
        for (int i = 0; i < image.length; i++) {
            resultArea.append(String.format("%.4f ", image[i]));
            if ((i + 1) % gridSize == 0) {
                resultArea.append("\n");
            }
        }
    }
}
```

Executando-se o código:



Resultado da Reconstrução

```
Imagem Reconstruída após 100 iterações:

0,9391 0,8141 0,9391 0,8141
0,9391 0,8141 0,9391 0,8141
0,9391 0,8141 0,9391 0,8141
0,9391 0,8141 0,9391 0,8141
```

Conclusão

Apesar de já existirem algoritmos mais precisos para aquisição de imagens de tomografia computadorizada, a técnica de reconstrução algébrica (ART) foi fundamental para adquirir as primeiras imagens nítidas do funcionamento do corpo humano e, justamente, foi o conceito base para a formulação de outros algoritmos mais precisos.

Referências:

chatGPT para construção do programa em Java (Previamente autorizado pelo professor Plácido Leitão Jr).
Wikipédia: https://en.wikipedia.org/wiki/Stefan_Kaczmarz
RORRES, Chris. ANTON, Howard. Álgebra Linear com aplicações. in: Capítulo 10 - Aplicações da álgebra linear. 10.12 - Tomografia computadorizada. página 615.
YOUTUBE. 'Como formatar trabalhos acadêmicos com normas ABNT com o Google Docs?' (https://www.youtube.com/watch?v=25IvSMeuPo4&list=PLXJiIWbFwH3nPJLwYzojgeFh4Y_M9-DGt&index=2&ab_channel=TecMundo)