INTEGRAÇÃO POR FRAÇÕES PARCIAIS

JOÃO COMINI CÉSAR DE ANDRADE

INTRODUÇÃO E OBJETIVOS

Objetivo Principal: Estudar algoritmos para a resolução de integrais racionais e implementar uma ferramenta computacional para realizar os cálculos.

O trabalho foca em três tipos de integrais:

1. Fatores Lineares Distintos
2. Fator Quadrático
3. Fatores Quadráticos Irredutíveis Distintos

Recursos Implementados:

- Um motor de integração simbólica escrito em **Java**.
- Detecção automática do tipo de denominador.
- Um parser de string para integração direta de expressões.

FUNDAMENTAÇÃO TEÓRICA

O QUE SÃO FRAÇÕES PARCIAIS?

Conceito: O método de Frações Parciais é uma técnica de álgebra que decompõe uma função racional complexa (uma divisão de polinômios) em uma soma de frações mais simples que sabemos como integrar.
Objetivo: Transformar uma integral difícil em várias integrais fáceis.
Pré-condição: O método se aplica diretamente quando o grau do numerador é menor que o grau do denominador.
Obs: Se o grau for maior, o programa primeiro executa a divisão polinomial.

REPRESENTAÇÃO COMPUTACIONAL

```java
// Construtor estático para um polinômio quadrático: ax^2 + bx + c
// Armazenado como [c, b, a]
public static Polynomial quadratic(double a, double b, double c) {
    return new Polynomial(List.of(c, b, a));
}
```

Polinômios foram representados usando uma "lista de coeficientes", conforme sugestão do trabalho.

Classe Polynomial.java:

- Armazena coeficientes em uma List<Double>, onde o índice da lista corresponde à potência de x.

```java
/**
 * Representa um polinômio como uma lista de coeficientes. [cite: 14]
 * O índice 0 é o termo constante (x^0), índice 1 é o termo (x^1), etc.
 * Ex: 5x^2 - 3x + 1 seria armazenado como [1.0, -3.0, 5.0]
 */
public class Polynomial {
    private List<Double> coefficients;
```

ALGORITMOS: FATORES LINEAREARES

Problema:

$$\int \frac{Ax + B}{(x - x1)\,(x - x2)} dx$$

Algoritmo:

1. Decomposição em: $\frac{C1}{x - x1} + \frac{C2}{x - x2}$
2. Resolver o sistema 2x2 para C1 e C2
3. A integral é $C_1 \ln|x - x_1| + C_2 \ln|x - x_2| + K$.

Implementação (solveCase1):

1. C1 e C2 calculados algebricamente.
2. Utilizadas funções auxiliares (formatTerm, combineTerms) para gerar uma string de saída limpa, omitindo termos nulos e tratando os sinais.

ALGORITMOS: FATORES LINEAREARES

TRECHO DO CÓDIGO (Classe: SymbolicIntegrator.java)

```
1  double C2 = (B + A * x2) / (x2 - x1);
2  double C1 = A - C2;
3
4  // Monta a string do resultado simbólico
5  String part1 = formatTerm(C1, String.format("ln|x - %.2f|", x1));
6  String part2 = formatTerm(C2, String.format("ln|x - %.2f|", x2));
7
8  String combined = combineTerms(part1, part2);
9  return String.format("Resultado: %s + K", combined);
```

ALGORITMOS: FATOR QUADRÁTICO

Problema:

$$\int \frac{Ax + B}{ax^2 + bx + c} dx$$

Algoritmo de identificação: É o algoritmo central. O programa deve primeiro "identificar o tipo de fatoração" usando o discriminante:
delta = $b^2 - 4ac$
Implementação(solveCase2):

1. Se delta > 0 (Raízes Reais): Fatora o denominador em a(x-x1)(x-x2) e reduz o problema ao Caso 1.
2. Se delta = 0 (Raiz Dupla): Aplica a decomposição $\frac{C1}{x - r} + \frac{C2}{(x - r)^2}$ e integra
3. Se delta < 0 (Irredutível): Usa a técnica de "completar o quadrado" e reescreve a integral, resultando em uma soma de ln e arctan.

ALGORITMOS: FATOR QUADRÁTICO

```java
/**
 * Resolve o Problema 2: integral((Ax+B) / (ax^2+bx+c)) dx [cite: 23]
 * Esta função identifica o tipo de raiz (delta) e resolve.
 */
public String solveCase2(double A, double B, double a, double b, double c) {
System.out.println("--- Resolvendo Caso 2 ---");
System.out.printf("Funcao: (%s) / (%s)\n",
            Polynomial.linear(A, B).toString(),
            Polynomial.quadratic(a, b, c).toString());

    double delta = b * b - 4 * a * c;

// ----------------------------------------------------
// Caso 2a: Delta > 0 (Duas raizes reais distintas)
// ----------------------------------------------------
    if (delta > EPSILON) {
        System.out.println("Info: Delta > 0. Reduzindo ao Caso 1.");
        double x1 = (-b + Math.sqrt(delta)) / (2 * a);
        double x2 = (-b - Math.sqrt(delta)) / (2 * a);

        // Lembre-se: (ax^2+bx+c) = a(x-x1)(x-x2)
        // A integral é (1/a) * integral((Ax+B) / (x-x1)(x-x2)) dx
        String case1Result = solveCase1(A, B, x1, x2);
        // Remove o cabeçalho "Resultado: " do solveCase1
        String integralPart = case1Result.substring(case1Result.indexOf(":") + 2);

        return String.format("Resultado: (1.0/%.2f) * (%s)", a, integralPart);
    }
```

```java
    // ----------------------------------------------------
    // Caso 2b: Delta = 0 (Uma raiz real dupla)
    // ----------------------------------------------------
    else if (Math.abs(delta) < EPSILON) {
        System.out.println("Info: Delta = 0. Caso de raiz dupla.");
        double r = -b / (2 * a); // A raiz dupla

        // Decomposição: (Ax+B) / (a(x-r)^2) = (1/a) * [C1/(x-r) + C2/(x-r)^2]
        // Ax + B = C1(x-r) + C2 = C1*x + (C2 - C1*r)
        double C1 = A;
        double C2 = B + C1 * r;

        System.out.printf("Decomposicao: (1/%.2f) * [%.2f/(x - %.2f) + %.2f/(x - %.2f)^2]\n", a, C1, r, C2, r);

        // Integrar: (1/a) * [ C1*ln|x-r| - C2/(x-r) ]
        String part1 = formatTerm(C1, String.format("ln|x - %.2f|", r));
        String part2 = formatTerm(-C2, String.format("1/(x - %.2f)", r));

        String combo = combineTerms(part1, part2);
        return String.format("Resultado: (1.0/%.2f) * (%s) + K", a, combo);
    }

    // ----------------------------------------------------
    // Caso 2c: Delta < 0 (Quadrático irredutível)
    // ----------------------------------------------------
    else {
        System.out.println("Info: Delta < 0. Caso de ln + arctan.");

        // 1. Reescrever o numerador (Ax+B) = K1 * (derivada) + K2
        // Derivada do denominador = 2ax + b
        // Ax+B = K1(2ax+b) + K2 = (2a*K1)x + (b*K1 + K2)
        double K1 = A / (2 * a);
        double K2 = B - K1 * b;

        // 2. Separar a integral:
        // I1 = K1 * integral((2ax+b) / (ax^2+bx+c)) dx
        // I2 = K2 * integral(1 / (ax^2+bx+c)) dx

        // 3. Resolver I1 (substituição u = ax^2+bx+c)
        String lnPart = formatTerm(K1, String.format("ln|%s|", Polynomial.quadratic(a, b, c).toString()));

        // 4. Resolver I2 (completar o quadrado e usar arctan)
        // Denom: a(x^2 + (b/a)x) + c
        //        = a(x + b/2a)^2 + c - b^2/4a
        //        = a(x + b/2a)^2 + (4ac - b^2)/4a
        //        = a( (x + b/2a)^2 + (-delta)/(4a^2) )
        //

        double u_offset = b / (2 * a); // u = x + u_offset
        double k_squared = -delta / (4 * a * a); // k^2
        double k = Math.sqrt(k_squared);

        // Integral de I2 = K2 * integral(1 / (a * (u^2 + k^2))) du
        //                = (K2/a) * (1/k) * arctan(u/k)
        //

        double arctanCoeff = (K2 / a) * (1 / k);
        String arctanPart = formatTerm(arctanCoeff,
                                String.format("arctan((x + %.2f) / %.2f)", u_offset, k));

        String combo = combineTerms(lnPart, arctanPart);
        return String.format("Resultado: %s + K", combo);
    }
}
```

ALGORITMOS: QUADRÁTICOS DISTINTOS

Problema:

$$\int \frac{Ax + B}{(x^2 + c1)(x^2 + c2)} dx$$

Algoritmo de Decomposição:

1. A forma da decomposição é: $\frac{C1x + D1}{x^2 + c1} + \frac{C2x + D2}{x^2 + c2}$
2. Isso gera um sistema 4x4 que se divide em dois sistemas 2x2 independentes (um para C1, C2 e outro para D1, D2).

Implementação(solveCase3):

1. Calcula C1, D1, C2, D2 diretamente.
2. Foi usada uma função auxiliar integrateQuadraticTerm que resolve a $\int \frac{Cx + D}{x^2 + c} dx$ na sua forma ln + arctan.

ALGORITMOS: QUADRÁTICOS DISTINTOS

TRECHO DO CÓDIGO (Classe: SymbolicIntegrator.java)

```java
public String solveCase3(double A, double B, double c1, double c2) {
System.out.println("--- Resolvendo Caso 3 ---");
System.out.printf("Funcao: (%s) / ((x^2 + %.2f)(x^2 + %.2f))\n",
            Polynomial.linear(A, B).toString(), c1, c2);

    if (Math.abs(c1 - c2) < EPSILON) {
        return "Erro: Denominadores quadraticos identicos (caso de raiz dupla).";
    }
    if (c1 < 0 || c2 < 0) {
        return "Erro: c1 ou c2 e negativo. Fatore em raizes lineares (Caso 1).";
    }

    // Decomposição: (C1*x + D1)/(x^2+c1) + (C2*x + D2)/(x^2+c2)
    // Ax+B = (C1x+D1)(x^2+c2) + (C2x+D2)(x^2+c1)
    //      = (C1+C2)x^3 + (D1+D2)x^2 + (C1c2+C2c1)x + (D1c2+D2c1)

    // Sistema para C (termos ímpares):
    // C1 + C2 = 0   (coef. x^3)
    // C1c2 + C2c1 = A (coef. x^1)
    double C2 = A / (c1 - c2);
    double C1 = -C2;
```

```java
    double D2 = B / (c1 - c2);
    double D1 = -D2;

System.out.printf("Decomposicao: (%.2fx + %.2f)/(x^2 + %.2f) + (%.2fx + %.2f)/(x^2 + %.2f)\n",
                        C1, D1, c1, C2, D2, c2);

    // Integrar termo a termo: integral( (Cx+D) / (x^2+c) ) dx
    String part1 = integrateQuadraticTerm(C1, D1, c1);
    String part2 = integrateQuadraticTerm(C2, D2, c2);

    String combo = combineTerms(part1, part2);
    return String.format("Resultado: %s + K", combo);
}

```

ALGORITMOS: MOTOR DE INTEGRAÇÃO

Além dos Requisitos: Foi implementado um "motor" de integração que automatiza todo o processo.

Divisão Polinomial (polynomialDivide):

1. Se a integral for imprópria (grau do numerador maior ou igual ao grau do denominador), o programa primeiro realiza a divisão polinomial.
2. Ele integra o quociente polinomial e aplica frações parciais apenas no resto.

Detecção Automática (integrate):

1. A função **integrate(numero, denom)** recebe as listas de coeficientes.
2. Ela analisa o denominador (grau, coeficientes) e direciona automaticamente para **solveCase2** ou **solveCase3**, sem que o usuário precise saber o caso.

Parser de String (parseAndIntegrate):

1. Este é o recurso de mais alto nível. O usuário pode inserir a integral como uma String.
2. O programa analisa a string, converte o numerador e o denominador em polinômios (listas de coeficientes) e chama o motor **integrate.**

RESULTADOS

Caso 1:

$$\int \frac{5x - 3}{x^2 - 2x - 3} dx$$

Resultado (Método solveCase1):

```
### TESTE 1 (Problema 1: Raizes Reais) ###
--- Resolvendo Caso 1 ---
Funcao: (5,00x - 3,00) / ((x - 3,00)(x - -1,00))
Decomposicao: 3,00/(x - 3,00) + 2,00/(x - -1,00)
Resultado: 3,00 ln|x - 3,00| + 2,00 ln|x - -1,00| + K
```

Resultado (Método parseAndIntegrate):

```
### TESTE 7 (Parser textual parseAndIntegrate) ###
--- Resolvendo Caso 2 ---
Funcao: (5,00x - 3,00) / (x^2 - 2,00x - 3,00)
Info: Delta > 0. Reduzindo ao Caso 1.
--- Resolvendo Caso 1 ---
Funcao: (5,00x - 3,00) / ((x - 3,00)(x - -1,00))
Decomposicao: 3,00/(x - 3,00) + 2,00/(x - -1,00)
Resultado (Detectado: Caso 2a: Delta>0 (duas raizes reais) - reduz ao Caso 1): (1.0/1,00) * (3,00 ln|x - 3,00| + 2,00 ln|x - -1,00| + K)
```

RESULTADOS

Caso 2:

$$\int \frac{3x+2}{x^2+2x+5}dx$$

Resultado (Método solveCase2):

```
### TESTE 4 (Problema 2: Delta < 0) ###
--- Resolvendo Caso 2 ---
Funcao: (3,00x + 2,00) / (x^2 + 2,00x + 5,00)
Info: Delta < 0. Caso de ln + arctan.
Resultado: 1,50 ln|x^2 + 2,00x + 5,00| - 0,50 arctan((x + 1,00) / 2,00) + K
```

RESULTADOS

Caso 3:

$$\int \frac{2x+1}{(x^2+1)(x^2+4)}\,dx$$

Resultado (Método parseAndIntegrate):

```
--- Resolvendo Caso 3 ---
Funcao: (2,00x + 1,00) / ((x^2 + 4,00)(x^2 + 1,00))
Decomposicao: (-0,67x + -0,33)/(x^2 + 4,00) + (0,67x + 0,33)/(x^2 + 1,00)
Resultado (Detectado: Caso 3: produto de quadraticos (x^2 + c1)(x^2 + c2)): (-0,33 ln|x^2 + 4,00| - 0,17 arctan(x / 2,00)) + (0,33 ln|x^2 + 1,00|
 + 0,33 arctan(x / 1,00)) + K
```

LIMITAÇÕES E MELHORIAS

Limitações Atuais:

1. **Parser:** O parser de string é funcional, mas simples. Ele lida com (x^2+1)(x^2+4), mas não com expressões algébricas mais complexas (ex: (x+1)^2).
2. **Escopo:** O programa não lida com fatores repetidos (ex: (x-1)^3 ou (x^2+1)^2).
3. **Saída:** A saída ainda é uma String formatada, não um objeto simbólico que possa ser reutilizado para outros cálculos (ex: diferenciação).

Possíveis Melhorias:

1. Expandir o parser para ser mais robusto.
2. Implementar os algoritmos para fatores lineares e quadráticos repetidos.
3. Criar um conjunto de classes de "Resultado Simbólico" para uma representação de dados mais estruturada.

GITHUB