

UNIVERSIDADE FEDERAL DO PARANÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM MÉTODOS NUMÉRICOS EM ENGENHARIA

Solução Numérica da Difusão de Calor 1D Transiente

Relatório do 1º Trabalho Computacional

Aluno: Romulo de Aguiar Beninca

1. Definição do Problema

O presente trabalho tem como objetivo desenvolver uma solução numérica para o problema clássico de difusão de calor unidimensional em regime transiente. O cenário físico consiste em uma placa plana de espessura L .

O trabalho foi desenvolvido em JavaScript e versionado no github por isso todo o código fonte e arquivos de entrada e saída estão disponíveis para consulta e reprodução dos resultados, no link :<https://rbeninca.github.io/cfd1/>

Conforme instrução do professor, para a modelagem matemática, partimos da equação diferencial parcial parabólica que governa o fenômeno, assumindo propriedades constantes e ausência de geração interna de calor, conforme estudamos no capítulo 4. Os detalhes do problema, incluindo a equação governante foram dados na Figura 1.

Dinâmica dos fluidos Computacional I

Relatório trabalho I

Romulo de Aguiar Benincá

Questão:

Implementar um programa um programa computacional para resolver com o método de volumes finitos o problema definido por:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad T(0, t) = T(L, t) = 0 \quad T(x, 0) = \sin\left(\frac{\pi x}{L}\right)$$

Dados: $L = 0,1\text{ m}$ $t_f = 20\text{s}$ $\alpha = 1,17 \times 10^4 \text{ m}^2/\text{s}$

$N = 10$ volumes de controle

$M = 5$ avanços no tempo

Assim, dados a especificação de enunciado temos que o problema trata da condução de calor em uma placa plana de espessura L , com propriedades constantes, em que temos a equação da difusão de calor dada como

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L, t > 0 \quad (0)$$

em que:

$T(x, t)$ é a temperatura

t é o tempo

x é a coordenada espacial em x

α é a constante de difusividade térmica no material m^2/s

Figura 1: Definição do problema, equação governante e parâmetros físicos.

Em suma, buscamos resolver numericamente a equação:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

2. Dedução Numérica (Volumes Finitos)

2.1. Discretização e Método θ

Para a solução numérica, adotou-se o Método dos Volumes Finitos (MVF). O domínio espacial $[0, L]$ foi particionado em N volumes de controle uniformes de largura Δx . A integração da equação governante foi realizada tanto no espaço quanto no tempo.

2. Discretização do domínio

Dividimos o domínio $[0, L]$ em N volumes de controle de largura

$$\Delta x = \frac{L}{N}$$

Assim, como cada volume é centrado em um ponto P

$$x_p = \left(i - \frac{1}{2}\right) \Delta x, \quad i = 1, \dots, N.$$

As faces este e leste do volume em torno de P estão em

$$x_w = x_p - \frac{\Delta x}{2}, \quad x_e = x_p + \frac{\Delta x}{2}$$

$$t^n = n \Delta t, \quad n = 0, 1, 2, \dots$$

3. Integrando a equação de volume de controle

Após integrarmos a derivada parcial (1) no volume de controle $[x_w, x_e]$ e no intervalo de tempo $[t^n, t^{n+1}]$

$$\int_{t^n}^{t^{n+1}} \int_{x_w}^{x_e} \frac{\partial T}{\partial t} dx dt = \Delta t \int_{t^n}^{t^{n+1}} \int_{x_w}^{x_e} \frac{\partial^2 T}{\partial x^2} dx dt \quad (1)$$

Assumindo que, dentro do volume, a derivada temporal é aproximadamente uniforme igual ao valor de P :

$$\int_{x_w}^{x_e} \frac{\partial T}{\partial t} dx \approx \Delta x \left(\frac{\partial T}{\partial t} \right)_P$$

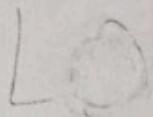
$$\int_{t^n}^{t^{n+1}} \int_{x_w}^{x_e} \frac{\partial T}{\partial t} dx dt \approx \Delta x [T_p^{n+1} - T_p^n] \quad (2)$$

Figura 2: Processo de discretização do domínio e integração da equação.

No que tange à discretização temporal, aplicou-se o **Método θ** . Para este trabalho, optou-se especificamente pelo esquema de **Crank-Nicolson** ($\theta = 0.5$).

$$\int_{x_w}^{x_e} \frac{\partial^2 T}{\partial x^2} dx \left. \frac{\partial T}{\partial x} \right|_e - \left. \frac{\partial T}{\partial x} \right|_w$$

$$\alpha \int_{t^n}^{t^{n+1}} \left[\left. \frac{\partial T}{\partial x} \right|_e - \left. \frac{\partial T}{\partial x} \right|_w \right] dt \quad (3)$$



$$\left. \frac{\partial T}{\partial x} \right|_e - \left. \frac{\partial T}{\partial x} \right|_w \approx \frac{T_w - 2T_p + T_E}{\Delta x} \quad (4)$$

$$\alpha \int_{t^n}^{t^{n+1}} \left[\left. \frac{\partial T}{\partial x} \right|_e - \left. \frac{\partial T}{\partial x} \right|_w \right] dt \approx \frac{\alpha}{\Delta x} \int_{t^n}^{t^{n+1}} (T_w - 2T_p + T_E) dt \quad (5)$$

método θ

$$\int_{t^n}^{t^{n+1}} (T_w - 2T_p + T_E) dt \approx \Delta t [\theta(T_p^{n+1} + 2T_p^n + T_E^n) + (1-\theta)(T_w^n - 2T_p^n + T_E^n)]. \quad (7)$$

Aplicando $\theta = \frac{1}{2}$ (Chandrikson)

$$\Delta x (T_p^{n+1} - T_p^n) = \frac{\alpha \Delta t}{2 \Delta x} [T_w^{n+1} + T_E^{n+1} - 2T_p^{n+1}) + T_w^n + T_E^n - 2T_p^n] \quad (8)$$

$$\Delta x T_p^{n+1} - \frac{\alpha \Delta t}{2 \Delta x} (T_w^{n+1} + T_E^{n+1} - 2T_p^{n+1}) = \Delta x T_p^n + \frac{\alpha \Delta t}{2 \Delta x} (T_w^n + T_E^n - 2T_p^n) \quad (9)$$

$$\left(\Delta x + \frac{\alpha \Delta t}{2 \Delta x} \right) T_p^{n+1} - \frac{\alpha \Delta t}{2 \Delta x} T_E^{n+1} = \Delta x T_p^n + \frac{\alpha \Delta t}{2 \Delta x} (T_w^n + T_E^n - 2T_p^n) \quad (10)$$

Escrivendo na forma algebrática:

$$a_p = \Delta x + \frac{\alpha \Delta t}{2 \Delta x}, \quad a_w = \frac{\alpha \Delta t}{2 \Delta x}, \quad a_E = \frac{\alpha \Delta t}{2 \Delta x}$$

$$b_p = \Delta x T_p^n + \frac{\alpha \Delta t}{2 \Delta x} (T_w^n + T_E^n - 2T_p^n)$$

Figura 3: Desenvolvimento algébrico da equação discretizada para volumes internos.

2.2. Tratamento dos Contornos

Um ponto crucial nesta implementação é o tratamento das condições de contorno. A dedução completa para os volumes adjacentes aos contornos (esquerdo e direito) é apresentada na Figura 4.

Contorno esquerdo

$$\left(\Delta x + \frac{\alpha \Delta t}{\Delta x} \right) T_1^{n+1} - \frac{\alpha \Delta t}{2 \Delta x} T_a - \frac{\alpha \Delta t}{2 \Delta x} T_2^{n+1} = \Delta x T_1^n + \frac{\alpha \Delta t}{2 \Delta x} (T_a + T_2^n - 2 T_1^n) \quad (11)$$

$$\alpha_p^{(1)} = \Delta x + \frac{\alpha \Delta t}{\Delta x}$$

$$\alpha_w^{(1)} = 0$$

$$\alpha_E^{(1)} = \frac{\alpha \Delta t}{2 \Delta x}$$

$$b_1 = \Delta x T_1^n + \frac{\alpha \Delta t}{2 \Delta x} (T_a + T_2^n - 2 T_1^n) \frac{\alpha \Delta t}{2 \Delta x}$$

Contorno direito

$$\left(\Delta x + \frac{\alpha \Delta t}{\Delta x} \right) T_N^{n+1} - \frac{\alpha \Delta t}{2 \Delta x} T_{N-1}^{n+1} - \frac{\alpha \Delta t}{2 \Delta x} = \Delta x T_N^n + \frac{\alpha \Delta t}{2 \Delta x} (T_{N-1}^n + T_b - 2 T_N^n) \quad (12)$$

$$\alpha_p^{(n)} = \Delta x + \frac{\alpha \Delta t}{\Delta x}$$

$$\alpha_w = \frac{\alpha \Delta t}{2 \Delta x}$$

$$\alpha_E = 0$$

$$b_N = \Delta x T_N^n + \frac{\alpha \Delta t}{2 \Delta x} (T_{N-1}^n + T_b - 2 T_N^n) + \frac{\alpha \Delta t}{2 \Delta x} T_b$$

$$\alpha_E = \theta \frac{\Delta t}{\Delta x}$$

$$\text{theta} \cdot \text{dt}/\text{dx}$$

$$\alpha_w = \theta \frac{\Delta t}{\Delta x}$$

$$\text{theta} \cdot \text{dt}/\text{dx}$$

$$\alpha_p = \alpha_E + \alpha_w + \frac{\Delta x}{\alpha}$$

$$\alpha_E^{(1)} + \alpha_w^{(1)} \frac{\Delta x}{\alpha}$$

Figura 4: Dedução dos coeficientes modificados para os volumes de contorno.

3. Da Teoria à Prática: Implementação Computacional

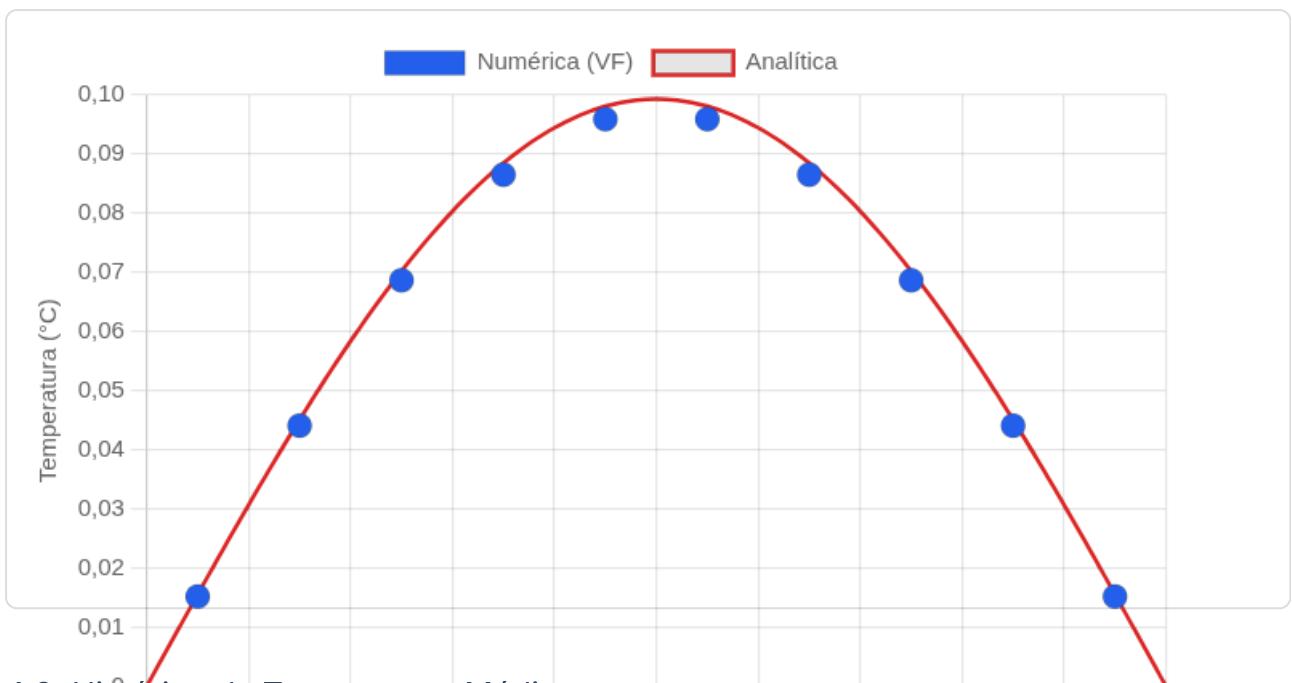
A implementação do código foi realizada utilizando a linguagem JavaScript. Para otimizar o cálculo, utilizamos variáveis auxiliares para agrupar os termos do passo de tempo anterior (n) que compõem o termo fonte. A tabela abaixo relaciona a matemática deduzida com as variáveis do código JavaScript apresentado na Seção 5:

Termo / Conceito	Expressão Algébrica (Dedução)	Variável no Código (JS)
Termo Difusivo (Parte Implícita)	$\theta \frac{\Delta t}{\Delta x}$	<code>termo_difusivo = theta * dt / dx</code>
Termo de Acúmulo	$\frac{\Delta x}{\alpha}$	<code>termo_acumulo = dx / alpha</code>
Coeficiente Central a_P (Contorno $i = 1$)	$a_P = \frac{\Delta x}{\alpha} + 3\theta \frac{\Delta t}{\Delta x}$	<code>aP[0] = 3.0 * termo_difusivo + termo_acumulo</code>
Termo Fonte b_P (Contribuição Central Explícita)	$\frac{\Delta x}{\alpha} - 3(1 - \theta) \frac{\Delta t}{\Delta x}$	<code>difusivo_explicito = (1.0 - theta) * dt / dx b1_contorno = termo_acumulo - 3.0 * difusivo_explicito</code>

4. Análise dos Resultados e Simulação

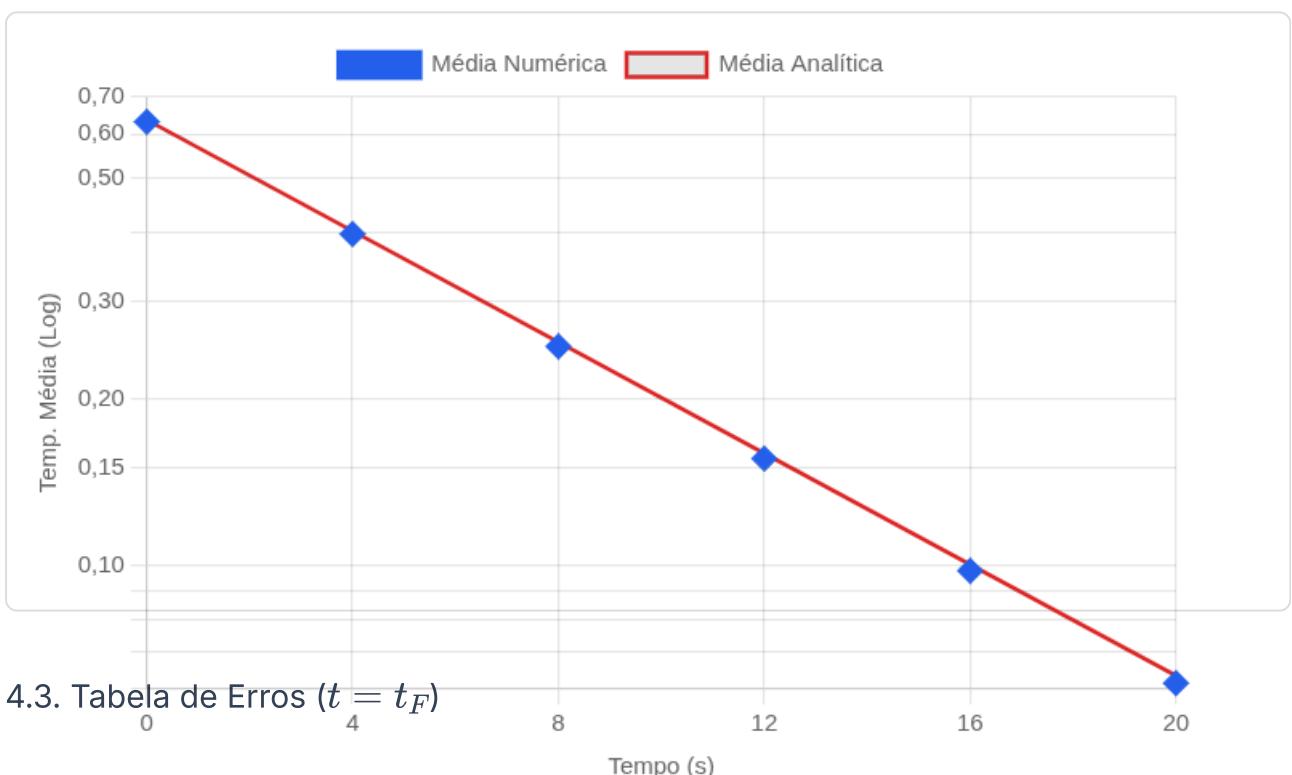
4.1. Perfil de Temperatura ($t = t_F$)

A comparação entre a solução numérica e a analítica para o tempo final configurado.



4.2. Histórico da Temperatura Média

Evolução da temperatura média da placa em escala logarítmica (eixo Y), demonstrando o decaimento exponencial.



4.3. Tabela de Erros ($t = t_F$)

Vol	Posição x (m)	$T_{analitica}$ (°C)	$T_{numerica}$ (°C)	Erro Absoluto
1	0.0050	0.015536	0.015191	3.4470e-4
2	0.0150	0.045087	0.044086	1.0003e-3

Vol	Posição x (m)	$T_{analitica}$ (°C)	$T_{numerica}$ (°C)	Erro Absoluto
3	0.0250	0.070224	0.068666	1.5581e-3
4	0.0350	0.088488	0.086524	1.9633e-3
5	0.0450	0.098089	0.095913	2.1763e-3
6	0.0550	0.098089	0.095913	2.1763e-3
7	0.0650	0.088488	0.086524	1.9633e-3
8	0.0750	0.070224	0.068666	1.5581e-3
9	0.0850	0.045087	0.044086	1.0003e-3
10	0.0950	0.015536	0.015191	3.4470e-4

5. Implementação em JavaScript

Abaixo apresenta-se o núcleo do algoritmo numérico implementado, que roda no navegador para gerar os resultados acima.

O código foi criado baseando-se no método TDMA para resolver sistemas tridiagonais, e usando com base o código em fortran disponibilizado pelo professor. Devido a natureza da web o arquivo de entrada foi adaptado e apresentado abaixo, similarmente a o arquivo utilizado em fortran como entrada.

O código javascript esta organizado da seguinte forma :

Funções Auxiliares: Funções matemáticas de apoio para cálculos analíticos e integração numérica.

Solução TDMA: Implementação do algoritmo Tri-Diagonal Matrix Algorithm para resolver o sistema linear.

Função de Interface: Função que lê os inputs da interface web e inicia a simulação.

Simulação Principal: Função que executa a simulação, monta a matriz, resolve o sistema e coleta resultados.

Geração de Arquivos: Funções para formatar e gerar os arquivos de entrada e saída.

Renderização de Resultados: Funções para atualizar a tabela e gráficos na interface web.

```
// --- SOLVER TDMA (Tri-Diagonal Matrix Algorithm) ---
function() {
    // Variáveis Globais de Gráfico para destruição correta
    let chartProfile = null;
    let chartHistory = null;

    // --- FUNÇÕES AUXILIARES ---
    const solucaoAnalitica = (pos, tempo, alpha, L) =>
        Math.exp(-alpha * Math.pow(Math.PI/L, 2) * tempo) * Math.sin(Math.PI*pos/L);

    const mediaAnalitica = (tempo, alpha, L) =>
        (2.0/Math.PI) * Math.exp(-alpha * Math.pow(Math.PI/L, 2) * tempo);
```

```

function calcMediaTrapezoidal(arrT, dx, N, Ta, Tb, L) {
    let soma = 0.0;
    soma += 0.25 * (Ta + arrT[0]) * dx;
    for(let i=1; i < N; i++) {
        soma += 0.5 * (arrT[i-1] + arrT[i]) * dx;
    }
    soma += 0.25 * (arrT[N-1] + Tb) * dx;
    return soma / L;
}

function resolverTDMA(a, b, c, d) {
    let n = d.length;
    let cp = new Float64Array(n);
    let dp = new Float64Array(n);
    let solucao = new Float64Array(n);
    cp[0] = c[0] / a[0];
    dp[0] = d[0] / a[0];
    for (let i = 1; i < n; i++) {
        let m = a[i] - b[i] * cp[i - 1];
        cp[i] = c[i] / m;
        dp[i] = (d[i] + b[i] * dp[i - 1]) / m;
    }
    solucao[n - 1] = dp[n - 1];
    for (let i = n - 2; i ≥ 0; i--) {
        solucao[i] = cp[i] * solucao[i + 1] + dp[i];
    }
    return solucao;
}

// --- FUNÇÃO DE INTERFACE (Lê Inputs e Chama Simulação) ---
// Tornar esta função globalmente acessível para o botão onClick
window.executarSimulacaoPeloUI = function() {
    // Ler valores
    const N = parseInt(document.getElementById('input_N').value);
    const M = parseInt(document.getElementById('input_M').value);
    const tF = parseFloat(document.getElementById('input_tF').value);
    const alpha = parseFloat(document.getElementById('input_alpha').value);
    const L = parseFloat(document.getElementById('input_L').value);
    const theta = parseFloat(document.getElementById('input_theta').value);
    const Ta = parseFloat(document.getElementById('input_Ta').value);
    const Tb = parseFloat(document.getElementById('input_Tb').value);

    executarSimulacao(N, M, tF, alpha, L, theta, Ta, Tb);
}

// --- SIMULAÇÃO PRINCIPAL ---
function executarSimulacao(N, M, tFinal, alpha, L, theta, Ta, Tb) {
    let dx = L/N;
    let dt = tFinal/M;

    // Arrays de Dados
    let T = new Float64Array(N);
    let To = new Float64Array(N);
    let x = new Float64Array(N);

    let historicoTempo = [0];
    let historicoT_num = [];
    let historicoT_anal = [];

    // Dados para relatório
    let report_aW = new Float64Array(N);
    let report_aE = new Float64Array(N);
    let report_aP = new Float64Array(N);
    let report_b_primeiro_passo = new Float64Array(N);

```

```

// Inicialização
for(let i=0; i= 0 ? '+' : '-';
let expVal = Math.abs(exponent).toString().padStart(2, '0');
return mantissa + 'E' + sign + expVal;
}

function gerarArquivoEntrada(N, M, tFinal, theta, alpha, L, Ta, Tb) {
    let txt = `trabalho1 ..... arg: nome do arquivo de saída de dados
NN : numero de volumes de controle {M} M: numero de passos de tempo
tFinal.toFixed(1)d0tf : tempo final [s] {theta}d0 theta: esquema (0.0=Expl, 0.5=CN,
1.0=Impl)
alpha.toExponential(2).replace('e','d')alpha : difusividade determinica [m2/s]{L}d0 L:
comprimento do domínio [m]
Ta.toFixed(1)d0Ta : temperatura contorno esquerdo [C]{Tb.toFixed(1)}d0 Tb:
temperatura contorno direito [C]
1 freq: frequencia de escrita dos resultados`;
    document.getElementById('output-arquivo-entrada').textContent = txt;
}

function gerarArquivoSaida(N, M, tFinal, theta, alpha, L, Ta, Tb, aW, aP, aE, b, x, T,
hTempo, hNum, hAnal) {
    let txt = "Solução Numérica da Eq. de Difusão de Calor 1D em Regime Transiente com
Propriedades Constantes\n\n";
    txt += "Dados de entrada\n\n";
    txt += `N = N : número de volumes de controle \n; txt+=\`{fmtE(alpha)} =
alpha: difusividade térmica [m2/s]\n`;
    txt += `fmtE(L) = L : comprimento do domínio [m]\n; txt+=\`{fmtE(Ta)} = Ta:
temperatura do contorno esquerdo (x=0) [°C]\n`;
    txt += `fmtE(Tb) = Tb : temperatura do contorno direito (x = L) [°C]\n; txt+=\`{fmtE(theta)} = theta: formulação temporal empregada\n\n\n`;

    txt += "Coeficientes e termos-fontes\nvol. oeste central
leste termo-fonte\n";
    for(let i=0; ii + 1{x[i].toFixed(4)}valAnal.toFixed(6)
{valNum.toFixed(6)}${err.toExponential(4)}`;
    }
    tbody.innerHTML = html;
}

function renderizarGraficos(N, x, T, tFinal, alpha, L, hTempo, hNum, hAnal) {
    // PERFIL
    const ctxP = document.getElementById('profileChart').getContext('2d');
    if(chartProfile) chartProfile.destroy();

    let dataAnal = [];
    for(let i=0; i<100; i++) { let pos = i*(L/100); dataAnal.push({x: pos, y:
solucaoAnalitica(pos, tFinal, alpha, L)}); }

    chartProfile = new Chart(ctxP, {
        type: 'scatter',
        data: {
            datasets: [
                { label: 'Numérica (VF)', data: Array.from(x).map((pos, i) => ({x: pos, y:
T[i]})), backgroundColor: '#2563eb', pointRadius: 6 },
                { label: 'Analítica', data: dataAnal, type: 'line', borderColor: '#dc2626',
borderWidth: 2, pointRadius: 0, fill: false, tension: 0.4 }
            ]
        },
        options: { responsive: true, maintainAspectRatio: false, scales: { x: { type:
'linear', title: {display:true, text: 'Posição (m)' } }, y: { title: {display:true, text:
'Temperatura (°C)'}, min: 0 } } }
    });
}

```

```

// HISTÓRICO
const ctxH = document.getElementById('historyChart').getContext('2d');
if(chartHistory) chartHistory.destroy();

chartHistory = new Chart(ctxH, {
    type: 'line',
    data: {
        labels: hTempo,
        datasets: [
            { label: 'Média Numérica', data: hNum, borderColor: '#2563eb',
backgroundColor: '#2563eb', showLine: false, pointStyle: 'rectRot', pointRadius: 6 },
            { label: 'Média Analítica', data: hAnal, borderColor: '#dc2626',
borderWidth: 2, pointRadius: 0 }
        ]
    },
    options: { responsive: true, maintainAspectRatio: false, scales: { x: { title:
{display:true, text: 'Tempo (s)'} }, y: { type: 'logarithmic', title: {display:true, text:
'Temp. Média (Log)' } } } }
});
}

// Inicialização Automática
executarSimulacaoPeloUI();
})();

// --- SIMULAÇÃO ---
// Esta função é chamada ao clicar em "Atualizar Simulação"
function executarSimulacao(N, M, tFinal, alpha, L, theta, Ta, Tb) {
    let dx = L/N;
    let dt = tFinal/M;
    // ... (Configuração de Coeficientes e Loop Temporal omitidos para brevidade) ...
}

```

Apêndice: Dados de Execução

Os dados abaixo são gerados dinamicamente com base nos parâmetros inseridos no painel de simulação.

ARQUIVO DE ENTRADA (REPRESENTAÇÃO)

```
trabalho1 ..... arq: nome do arquivo de saída de dados
 10          N:     numero de volumes de controle
 5           M:     numero de passos de tempo
 20.0d0      tf:    tempo final [s]
 0.5d0      theta: esquema (0.0=Expl, 0.5=CN, 1.0=Impl)
 1.17d-4     alpha: difusividade termica [m2/s]
 0.1d0       L:     comprimento do dominio [m]
 0.0d0       Ta:    temperatura contorno esquerdo [C]
 0.0d0       Tb:    temperatura contorno direito [C]
 1           freq:  frequencia de escrita dos resultados
```

ARQUIVO DE SAÍDA GERADO (TRABALHO1.TXT)

Solução Numérica da Eq. de Difusão de Calor 1D em Regime Transiente com Propriedades Constantes

Dados de entrada

```
10 = N:     número de volumes de controle
1.17000000000000E-04 = alpha: difusividade térmica [m2/s]
1.00000000000000E-01 = L:     comprimento do domínio [m]
0.00000000000000E+00 = Ta:    temperatura do contorno esquerdo (x=0) [°C]
0.00000000000000E+00 = Tb:    temperatura do contorno direito (x=L) [°C]
5.00000000000000E-01 = theta: formulação temporal empregada
```

Coeficientes e termos-fontes

vol.	oeste	central	leste	termo-fonte
1	0.00000000000000E+00	6.854700854700855E+02	2.00000000000000E+02	
1.030788802122646E+01				
2	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
2.991465616488281E+01				
3	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
4.659316933545514E+01				
4	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
5.871081845777348E+01				
5	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
6.508144360711930E+01				
6	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
6.508144360711930E+01				
7	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
5.871081845777348E+01				
8	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
4.659316933545503E+01				
9	2.00000000000000E+02	4.854700854700855E+02	2.00000000000000E+02	
2.991465616488307E+01				
10	2.00000000000000E+02	6.854700854700855E+02	0.00000000000000E+00	
1.030788802122612E+01				

Soluções numéricas

Volume numérico [m³]	Posição [m]	Temp. numérica [°C]	Temp. analítica [°C]	Erro
0	0.00000000000000E+00	0.00000000000000E+00	0.00000000000000E+00	
0.00000000000000E+00				
1	5.00000000000000E-03	1.519114550741685E-02	1.553584195521915E-02	
3.446964478022955E-04				
2	1.50000000000000E-02	4.408642135705011E-02	4.508676941050477E-02	
1.000348053454662E-03				
3	2.50000000000000E-02	6.866621111609580E-02	7.022428973789598E-02	
1.558078621800171E-03				
4	3.50000000000000E-02	8.652447370547317E-02	8.848776730434492E-02	
1.963293598871749E-03				
5	4.50000000000000E-02	9.591311795710199E-02	9.808944567651699E-02	
2.176327719415005E-03				
6	5.50000000000000E-02	9.591311795710201E-02	9.808944567651699E-02	
2.176327719414978E-03				
7	6.50000000000000E-02	8.652447370547314E-02	8.848776730434493E-02	
1.963293598871790E-03				
8	7.50000000000000E-02	6.866621111609575E-02	7.022428973789598E-02	
1.558078621800227E-03				
9	8.50000000000001E-02	4.408642135705020E-02	4.508676941050474E-02	
1.000348053454538E-03				
10	9.50000000000000E-02	1.519114550741667E-02	1.553584195521920E-02	
3.446964478025314E-04				
11	1.00000000000000E-01	0.00000000000000E+00	0.00000000000000E+00	
0.00000000000000E+00				

Soluções numéricas - Temperaturas Médias			
Tempo [s] [°C]	Numérica [°C]	Analítica [°C]	Erro numérico
0.00000000000000E+00	6.314235988979546E-01	6.366197723675814E-01	
5.196173469626753E-03			
4.00000000000000E+00	3.960704353050870E-01	4.011257975542927E-01	
5.055362249205719E-03			
8.00000000000000E+00	2.484414424746792E-01	2.527441220764715E-01	
4.302679601792231E-03			
1.20000000000000E+01	1.558388226865631E-01	1.592507677982495E-01	
3.411945111686354E-03			
1.60000000000000E+01	9.775236536395993E-02	1.003418272835587E-01	
2.589461919598773E-03			
2.00000000000000E+01	6.131671665325673E-02	6.322407384157178E-02	
1.907357188315043E-03			