

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

ANÁLISE EQUAÇÃO DO CALOR UNIDIMENSIONAL

December 6, 2017

José Augusto

Contents

1	Introdução	3
2	Problema	3
2.1	Equação do calor unidimensional	3
2.2	Discretização	3
2.3	Implementação	4
3	Resultados	5
3.1	Resultados	5
3.2	Análise dos resultados	6
4	Conclusão	7

List of Figures

1	Decaimento do erro pelo aumento do número de nodos	6
---	--	---

1 INTRODUÇÃO

Este documento contém o relatório dos métodos empregados na simulação numérica da equação do calor unidimensional, apresenta os resultados obtidos e uma análise dos mesmos.

2 PROBLEMA

Essa seção apresenta o problema e a maneira como tal foi atacado para obtenção dos resultados.

2.1 Equação do calor unidimensional

A equação do calor é apresentada da seguinte maneira:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + g(x, t) \quad (1)$$

Tal que, $T(x, t) = \bar{U} \sin(\theta x) e^{\alpha\theta^2 - t}$, dessa maneira, aplicando as derivadas, obtemos:

$$\frac{\partial T}{\partial t} = -\bar{U}\alpha\theta^2 \sin(\theta x) e^{\alpha\theta^2 - t} \quad (2)$$

$$\frac{\partial^2 T}{\partial x^2} = -\bar{U}\theta^2 \sin(\theta x) e^{\alpha\theta^2 - t} \quad (3)$$

Desta maneira, substituindo (2) e (3) em (1), podemos encontrar o termo fonte $g(x, t)$.

$$\begin{aligned} \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} &= g(x, t) \\ g(x, t) &= -\bar{U}\alpha\theta^2 \sin(\theta x) e^{\alpha\theta^2 - t} + \alpha \bar{U}\theta^2 \sin(\theta x) e^{\alpha\theta^2 - t} \\ g(x, t) &= 0 \end{aligned}$$

Assim, a equação (1) pode ser reescrita da forma:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad (4)$$

2.2 Discretização

Para a discretização da equação diferencial, foi utilizado o método de Crank-Nicolson, que é baseado no método das diferenças centradas no espaço e da regra trapezoidal no tempo, com uma combinação do método de Euler implícito e explícito. Tem-se que tal método é um método de segunda ordem no tempo e no espaço. A discretização foi realizada a partir de (4), da seguinte maneira:

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

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{2\Delta x^2} \cdot [(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + (T_{i-1}^n - 2T_i^n + T_{i+1}^n)]$$

$$T_i^{n+1} - T_i^n = \frac{\alpha \cdot \Delta t}{2\Delta x^2} \cdot [(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + (T_{i-1}^n - 2T_i^n + T_{i+1}^n)]$$

Tomamos então $s = \frac{\alpha \cdot \Delta t}{2\Delta x^2}$, isso nos deixa com:

$$T_i^{n+1} - T_i^n = s[(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) + (T_{i-1}^n - 2T_i^n + T_{i+1}^n)]$$

Rearrajando os termos com tempo $n+1$ do lado esquerdo, resulta em:

$$T_i^{n+1} - s(T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}) = T_i^n + s(T_{i-1}^n - 2T_i^n + T_{i+1}^n)$$

Distribuindo o termo s e colocando valores em comum em evidência, a equação discretizada final é dada por:

$$-sT_{i-1}^{n+1} + (1 + 2s)T_i^{n+1} - sT_{i+1}^{n+1} = sT_{i-1}^n + (1 - 2s)T_i^n + sT_{i+1}^n \quad (5)$$

Para obtenção da solução do problema foram definidas as seguintes condições:

$$\begin{aligned} T(0, t) &= 0, \quad \forall t > 0 \\ T(2\pi, t) &= 0, \quad \forall t > 0 \\ T(x, 0) &= \bar{U} \sin(\theta x), \quad \forall x \in [0, 2\pi] \end{aligned}$$

Assim como visto na condição, o domínio é contido no intervalo $[0, 2\pi]$.

2.3 Implementação

A implementação do método foi realizada utilizando a linguagem C, o que afetou na captura dos valores de ponto flutuante, pois a precisão máxima fornecida pela linguagem é *d.dddddd*, a partir de certo ponto, não é possível afirmar qual o erro está sendo gerado pelo método. Como o principal intuito do problema é encontrar a razão do erro gerado pelo método, alguns parâmetros foram tomados como constantes, como, $\alpha = 1$, $\bar{U} = 1$ e $\theta = 1$, assim diminuindo possível dificuldades que poderiam atrapalhar na análise.

Como se trata de um método implícito, é necessário realizar a resolução de um sistema de equações a cada passo tempo, para resolução de tal sistema, foi utilizado o método de Gauss-Seidel.

A matriz criada é da forma:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ a & b & c & 0 & \cdots & 0 \\ 0 & a & b & c & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & c \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

$$a = \frac{\alpha \Delta t}{2\Delta x^2}$$

$$c = \frac{\alpha \Delta t}{2\Delta x^2}$$

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

Tal que $\alpha = 1$.

Nessa implementação, para o calculo do erro foi utilizada o norma L_∞ , como se trata de uma norma que captura o maior erro, foi implementada da seguinte maneira. A cada passo de tempo executado pelo programa, é realizado a captura do maior erro para o passo, seguindo a seguinte maneira:

$$\begin{aligned} an &\leftarrow \text{vetor solução analítica} \\ nu &\leftarrow \text{vetor solução numérica} \\ l &\leftarrow l = \{x \mid x = \sqrt[2]{(an_i - nu_i)^2}\} \\ \epsilon &\leftarrow \max(l) \end{aligned}$$

Retornamos ϵ e adicionamos em um vetor de erros, em que cada posição contém o maior erro cometido a cada passo de tempo, assim, quando o programa executa todos os passos de tempo, retiramos o maior valor desse vetor de erros, dessa forma é possível capturar o error de maior magnitude que ocorreu durante a execução do método levando em consideração todos os passos de tempo.

3 RESULTADOS

Essa seção apresenta os resultados que foram obtidos a partir da implementação explicada na seção anterior.

3.1 Resultados

Para analisar o valor do erro em relação a solução analítica, foram executadas diversas execuções em sequência com agluns parâmetros fixos, tais parâmetros foram $\Delta t = 0.0001$ e

cada execução teve 100 passos de tempos. Para esse teste variamos o valor de Δx , dobrando o número de nodos na malha a cada execução, ou seja, diminuindo o valor de Δx , com isso podemos analisar o que acontece com o erro entre a solução analítica e numérica. Dessa execução, os resultados obtidos foram:

Número de nodos	ϵ	Razão
2^3	0.000619	–
2^4	0.000140	4.4214
2^5	0.000033	4.2424
2^6	0.000008	4.1250
2^7	0.000002	4.0

3.2 Análise dos resultados

Podemos notar que a razão é o valor esperado, pois como citado anteriormente, o método de Crank-Nicolson é de segunda ordem tanto em espaço e tempo, ou seja, $\mathcal{O}(\Delta x^2)$, assim, ao dobrarmos o número de nodos na malha, esperamos que o erro diminua por um fator de 4. Mas pela dificuldade de representação de ponto flutuante em C, acompanhar a redução do erro só é possível até certo ponto.

Para quesito de visualização, o gráfico gerado a partir dos erros foi o seguinte:

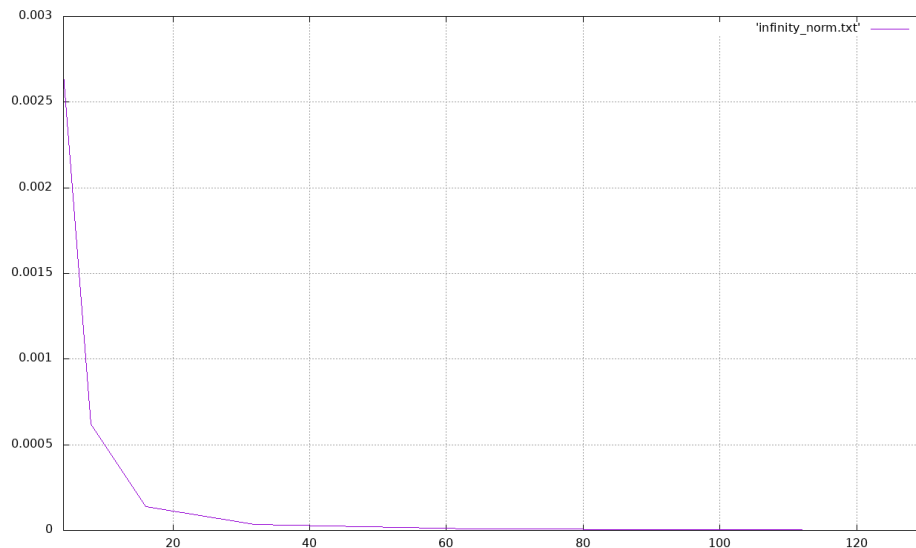


Figure 1: Decaimento do erro pelo aumento do número de nodos

Tanto pelo valores, como pelo gráfico, podemos notar que com um aumento do número de nodos na malha, a solução numérica se aproxima mais e mais da solução analítica.

4 CONCLUSÃO

Inicialmente um problema da equação do calor, representado por equações diferenciais, se utilizando do método de Crank-Nicolson, e feita a comparação dos resultados obtidos pelo método numérico com os resultados da solução analítica, é possível comprovar a relação de Δt , Δx e o erro.

Nesse aspecto, é possível encontrar valores ótimos para manter o sistema estável, mesmo Crank-Nicolson sendo um método incondicionalmente estável, ainda podem existir fatores que criam oscilações na resolução, o que pode se propagar com os passos tempos.

Com o nosso experimento realizado, foi possível comprovarmos que o erro do método é de segunda ordem, e que ao dobrar o número de nodos reduzimos o nosso erro máximo por 4.