토마스 알고리즘

생성자	재 재환 김
태그	엔지니어링

1. 토마스 알고리즘의 이론

토마스 알고리즘은 **LU 분해법**의 변형으로, 삼중대각 행렬의 구조적 특징을 이용하여 계산을 효율적으로 수행하는 방법입니다. 이 알고리즘은 두 가지 주요 단계로 나누어집니다:

- **전진 대입 (Forward Elimination)**
- **후진 대입 (Backward Substitution)**

삼중대각 행렬 시스템은 다음과 같은 일반 형태를 가집니다:

$$A \cdot x = d$$

여기서 A는 $n \times n$ 크기의 삼중대각 행렬입니다. 즉, A는 다음과 같은 구조를 가집니다:

$$A = \begin{bmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_1 & b_2 & c_2 & \cdots & 0 \\ 0 & a_2 & b_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & a_{n-1} & b_n \end{bmatrix}$$

- a_i: 아래 대각선의 요소들 (하삼각선).
- b_i: 주 대각선의 요소들.
- c_i: 위 대각선의 요소들 (상삼각선).

---

2. 토마스 알고리즘의 수식

토마스 알고리즘은 다음과 같은 순서로 진행됩니다.

2.1 전진 대입 (Forward Elimination)

행렬 A의 요소를 수정하여 하삼각 행렬의 형태로 변환합니다. 이를 통해 bib_ibi와 did_idi를 차례로 업데이트합니다.

1. 처음에 각 행의 요소들을 다음과 같이 수정합니다:

$$m_i = \frac{a_{i-1}}{b_{i-1}}$$

여기서, m_i는 각 행에서 하삼각 성분을 제거하는 비율입니다.

1. 그 다음 주 대각선의 요소 b_i와 우변 벡터 d_i를 업데이트합니다:

$$b_i = b_i - m_i \cdot c_{i-1}$$

$$d_i = d_i - m_i \cdot d_{i-1}$$

2.2 후진 대입 (Backward Substitution)

전진 대입이 완료된 후, 후진 대입을 통해 해 x를 구합니다. 이때, 마지막 요소 x_n부터 시작하여 다음과 같이 해를 계산합니다:

1. 마지막 x_n은 아래와 같이 계산됩니다:

$$x_n = \frac{d_n}{b_n}$$

1. 그 이후로 역순으로 나머지 해들을 구합니다:

$$x_i = \frac{d_i - c_i \cdot x_{i+1}}{b_i}$$

이 과정을 통해 각 x_i를 효율적으로 계산할 수 있습니다.

---

3. 토마스 알고리즘의 구체적인 예시

문제 설정:

다음의 삼중대각 행렬 시스템을 해봅시다.

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{bmatrix}$$

이 시스템을 토마스 알고리즘을 사용하여 풉니다.

3.1 전진 대입 (Forward Elimination)

1. 첫 번째 행에서 m_2를 계산합니다:

$$m_2 = \frac{-1}{2.04} = -0.49$$

b_2와 d_2를 업데이트합니다:

$$b_2 = 2.04 - (-0.49)(-1) = 1.550$$

$$d_2 = 0.8 - (-0.49)(40.8) = 20.8$$

1. 두 번째 행에서 m_3를 계산합니다:

$$m_3 = \frac{-1}{1.550} = -0.645$$

b_3와 d_3를 업데이트합니다:

$$b_3 = 2.04 - (-0.645)(-1) = 1.395$$

$$d_3 = 0.8 - (-0.645)(20.8) = 14.221$$

1. 세 번째 행에서 m_4를 계산합니다:

$$m_4 = \frac{-1}{1.395} = -0.717$$

b_4와 d_4를 업데이트합니다:

$$b_4 = 2.04 - (-0.717)(-1) = 1.323$$

$$d_4 = 200.8 - (-0.717)(14.221) = 210.996$$

3.2 후진 대입 (Backward Substitution)

1. 마지막 해 x_4를 계산합니다:

$$x_4 = \frac{210.996}{1.323} = 159.480$$

1. x_3을 계산합니다:

$$x_3 = \frac{14.221 - (-1)(159.480)}{1.395} = 124.538$$

2. x_2을 계산합니다:

$$x_2 = \frac{20.8 - (-1)(124.538)}{1.550} = 93.778$$

1. x_1을 계산합니다:

$$x_1 = \frac{40.8 - (-1)(93.778)}{2.04} = 65.970$$

최종 해:

$$x_1 = 65.970, \quad x_2 = 93.778, \quad x_3 = 124.538, \quad x_4 = 159.480$$

---

4. 토마스 알고리즘의 실제 적용 사례

4.1 1차원 열전도 문제

- 1차원 열전도 방정식의 수치해석에서 유한 차분법을 사용하면 삼중대각 행렬 시스템이 생성됩니다. 토마스 알고리즘은 이 시스템을 효율적으로 해결합니다.

- 예를 들어, **양 끝 온도가 고정된 금속 막대의 내부 온도 분포**를 시간에 따라 계산할 때 토마스 알고리즘이 활용됩니다.

4.2 유체 역학

- **나비어-스토크스 방정식**의 수치해석, 특히 2차원 및 3차원 문제에서 유체의 속도와 압력 계산 시 유한 차분법이나 유한 요소법을 적용하면 삼중대각 행렬이 발생합니다. 이때 토마스 알고리즘을 사용하여 계산 효율을 향상시킵니다.
- **예시**: 직사각형 채널을 따라 흐르는 유체의 해석 문제에서 벽면 경계 조건을 적용하면 삼중대각 행렬이 생성되며, 토마스 알고리즘으로 이를 효과적으로 해결합니다.

4.3 금융공학

- **옵션 가격** 계산에 사용되는 블랙-숄즈 방정식(Black-Scholes Equation)은 편미분 방정식 형태입니다. 이를 유한 차분법으로 해석할 때 발생하는 삼중대각 행렬 시스템 해결에 토마스 알고리즘이 효과적으로 적용됩니다.