

# GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

NUMERISCHE STRÖMUNGSMECHANIK

---

## Simulation mit FTCS- und BTCS-Verfahren

---

*Author:*  
Philip MARSZAL

*Prüfer:*  
Prof. Dr. Andreas  
TILGNER

3. August 2016



## Zusammenfassung

Your abstract.

## 1 Einleitung

## 2 Problemstellung

$$\frac{\partial T(\mathbf{x}, t)}{\partial t} + \text{Pe } \mathbf{u}_0 \cdot \nabla T(\mathbf{x}, t) = \nabla^2 T(\mathbf{x}, t). \quad (1)$$

## 3 Methoden

Zur Simulation von Strömungsmechanikproblemen kommen eine Vielzahl verschiedener Verfahren in Frage. Ein konzeptionell simples Verfahren, das häufig Anwendung findet, ist die *Finite-Differenzen-Methode*. Diese wird zum Lösen des hier gestellten Problems verwendet. Andere Verfahren die zur Lösung dieser Art von Problem eingesetzt werden können sind *Finite-Volumen-Methoden*, *Finite-Elemente-Verfahren* und *spektrale Verfahren*. Bei fast allen Verfahren liegt die Diskretisierung des integrierten Systems durch ein Gitter zu Grunde, welches die theoretisch kontinuierliche Lösungsfunktion approximiert.

### 3.1 Finite-Differenzen-Verfahren

Um eine partielle Differentialgleichung numerisch zu integrieren, müssen die Ableitungen ersetzt werden durch diskret berechenbare Terme. Ein naiver Ansatz ist die Näherung der Ableitung  $T'(x)$  einer Funktion an einem Punkt durch den Differenzenquotienten.

$$T'(x) = \lim_{h \rightarrow 0} \frac{T(x+h) - T(x)}{h}. \quad (2)$$

Bei der Verwendung eines Gitters entspricht hier  $h$  dem Abstand der zwei benachbarten Gitterpunkte. Der Differenzenquotient in Gleichung (2) entspricht dem Vorwärtsdifferenzenquotienten erster Ordnung in  $h$ .

Eine Approximation höherer Ordnung erhält man z.B. durch den zentralen Differenzenquotienten

$$T'(x) = \frac{T(x+h) - T(x-h)}{2h} + O(h^2), \quad (3)$$

oder für die zweite Ableitung als Vorwärtsdifferenzenquotient von  $T'(x)$ :

$$T''(x) = \frac{T(x+h) - 2T(x) + T(x-h)}{h^2} + O(h^2). \quad (4)$$

Durch Kombinationen unterschiedlicher Differenzenquotienten können Näherungen höherer Ordnung erhalten werden.

### 3.2 FTCS-Variante

Verwendet man zur Diskretisierung der PDGL eine Vorwärtsdifferenz für die Zeitableitung und eine zentrale Differenz für die Ortsableitungen, so spricht man von einem *FTCS*-Verfahren (**F**orwards **T**ime **C**entral **S**pace). Diskretisiert man Gleichung (1) auf diese Weise so erhält man die Differenzengleichung:

$$\begin{aligned} \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} + \text{Pe } u_{i,j} \frac{T_{i+1,j}^n - T_{i-1,j}^n}{\Delta x} + \text{Pe } v_{i,j} \frac{T_{i,j+1}^n - T_{i,j-1}^n}{\Delta y} \\ = \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{\Delta x^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{\Delta y^2} \end{aligned} \quad (5)$$

Diese Gleichung kann nun numerisch integriert werden, indem  $T_{i,j}^{n+1}$  für jeden Gitterpunkt berechnet wird. Allerdings ist darauf zu achten ob man  $T_{i,j}^{n+1}$  am Rand berechnet, denn hier müssen die Randbedingungen in die Gleichung eingebaut werden.

Die Dirichlet Randbedingungen lassen sich unproblematisch für  $T_{i,0}^n$  und  $T_{i,N_y+1}^n$  einsetzen. Diese Gitterpunkte verändern sich nicht im Laufe der Zeit, müssen also nicht integriert werden. Anders sieht dies allerdings für die Neumann Randbedingungen aus. Die Gitterpunkte mit  $i = 0$  und  $i = N_x + 1$  verändern sich sehr wohl im Laufe der Zeit.

Um zu vermeiden, dass man auf Gitterpunkte außerhalb des Gitters zugreift, muss man eine Vorwärtsdifferenz verwenden. Eine Näherung zweiter Ordnung für die Ableitung  $\frac{\partial T(x,t)}{\partial x}$  bei  $x = 0$  und  $x = 1$  lässt sich mit Gleichung (6)

berechnen:

$$\begin{aligned} T_{0,j}^{n+1} &= -\frac{1}{3} (T_{2,j}^{n+1} - 4T_{1,j}^{n+1}), \\ T_{N_x+1,j}^{n+1} &= -\frac{1}{3} (T_{N_x-1,j}^{n+1} - 4T_{N_x,j}^{n+1}). \end{aligned} \quad (6)$$

Mit Gleichung (5) und Gleichung (6) ist man nun in der Lage das Problem zu simulieren.

### 3.3 Die BTCS-Variante

Im Gegensatz zu dem FTCS-Verfahren, welches ein explizites Verfahren zu Lösung von partiellen Differentialgleichungen darstellt, handelt es sich bei der *BTCS*-Methode (**B**ackwards **T**ime **C**entral **S**pace) um ein implizites Verfahren. Implizite Verfahren erfordern die Lösung eines Gleichungssystems ehe ein Zeitschritt durchgeführt werden kann.

Im BTCS-Verfahren wird die Zeitableitung durch eine Rückwärtsdifferenz diskretisiert. Dadurch entsteht in dem vorliegenden Problem zunächst ein lineares Gleichungssystem aus  $(N_x + 1) \cdot (N_y + 1)$  Gleichungen für die  $T_{i,j}^n$ .

$$\begin{aligned} \frac{T_{i,j}^n - T_{i,j}^{n-1}}{\Delta t} + \text{Pe } u_{i,j} \frac{T_{i+1,j}^n - T_{i-1,j}^n}{\Delta x} + \text{Pe } v_{i,j} \frac{T_{i,j+1}^n - T_{i,j-1}^n}{\Delta y} \\ = \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{\Delta x^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{\Delta y^2} \end{aligned} \quad (7)$$

Führt man die Dirichlet-Randbedingungen ein so reduziert sich das Gleichungssystem auf  $N_x \cdot (N_y - 1)$  Gleichungen. Um die Neumann-Randbedingungen mit dem Gleichungssystem zu vereinigen betrachtet man den zentralen Differenzenquotienten an  $x = 0$  und  $x = 1$ :

$$\left. \frac{\partial T}{\partial x} \right|_{x=0} = \frac{T_{1,j}^n - T_{-1,j}^n}{\Delta x} = 0,$$

und einem analogen Quotienten für  $x = 1$ . Mithilfe dieser Gleichungen lassen sich die Terme  $T_{-1,j}^n$  und  $T_{N_x+2,j}^n$  aus dem Gleichungssystem eliminieren und somit die Neumann-Randbedingungen in die Gleichungen einbauen.

Um dieses System zu lösen bietet es sich an das Gitter  $T_{i,j}^n$  als  $N_x \cdot (N_y - 1)$  dimensionalen Vektor umzuschreiben. Die Lösung des Systems entspricht

dann einer Invertierung der Matrix  $M$ , die sich aus den Gleichungen als Gleichung (8) ergibt.

$$M = \begin{pmatrix} (1. + 4D) & (v_{0,1}A - D) & 0 & \dots & -2D & 0 & \dots & \dots \\ 0 & (1. + 4D) & (v_{0,2}A - D) & 0 & \dots & -2D & 0 & \dots \\ -(v_{0,2}A + D) & 0 & (1. + 4D) & (v_{0,3}A - D) & 0 & \dots & \dots & \dots \\ 0 & -(v_{0,3}A + D) & 0 & (1. + 4D) & (v_{0,4}A - D) & 0 & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & -(u_{i,j}A + D) & \dots & -(v_{i,j}A + D) & (1. + 4D) & (v_{i,j}A - D) & \dots & (u_{i,j}A - D) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & \dots & \dots & -2D & \dots & -(v_{N_x+1,N_y}A + D) & (1. + 4D) \end{pmatrix} \quad (8)$$

Hierbei wurde der Vektor  $\mathbf{T}^n$  so gewählt, dass die Spalten des Gitters aneinander gehängt werden, sprich die Komponenten des Vektors laufen zunächst im  $j$ -Index durch :  $\mathbf{T}^n = (T_{0,1}^n, T_{0,2}^n, \dots, T_{0,N_y}^n, T_{1,1}^n, \dots, T_{N_x+1,N_y}^n)$ . Die Terme  $A = \frac{Pe \Delta t}{\Delta x}$  und  $D = \frac{\Delta t}{\Delta x^2}$  entsprechen den konstanten Vorfaktoren im Advektions- und Diffusionsterm.

### 3.4 Der SOR-Algorithmus

Die Matrix  $M$  hat für vernünftige Gittergrößen und Zeitschritte  $\Delta t$  die Eigenschaft, dass sie diagonal dominant ist. Also die Summe der Beträge der nicht-diagonalen Einträge in jeder Spalte nicht den Betrag des diagonalen Eintrags dieser Spalte übersteigt.

Für Matrizen mit dieser Eigenschaft liefert das sogenannte Gauß-Seidel-Verfahren eine approximative Lösung des Gleichungssystems  $M\mathbf{x} = \mathbf{b}$ , die garantiert gegen die tatsächliche Lösung konvergiert. Zunächst wird  $M$  in eine Summe aus untere und oberer Dreiecksmatrix,  $L$  und  $U$ , und der Diagonale  $D$  zerlegt. Die Iterationsvorschrift des Gauß-Seidel-Verfahrens sieht nun wie folgt aus:

$$\mathbf{x}_{n+1} = (L + D)^{-1}(\mathbf{b} - U\mathbf{x}_n) \quad (9)$$

Der Vorteil dieser Methode liegt darin, dass das Produkt von  $(L + D)^{-1}$  mit einem Vektor einfach über Vorwärtssubstitution berechnet werden kann, dank der Dreiecksform von  $L + D$ .

Eine Weiterentwicklung des Gauß-Seidel-Algorithmus ist die *successive over-relaxation*-Methode (SOR). Dazu wird Gleichung (9) zunächst durch den

Korrekturvektor  $\mathbf{r}_n = M\mathbf{x}_n - \mathbf{b}$  ausgedrückt. Über den Relaxationsparameter  $\omega$  kann nun die Konvergenz des Gauß-Seidel-Verfahrens erheblich beschleunigt werden. Die Iterationsvorschrift des SOR-Algorithmus lautet:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \omega(L + D)^{-1}(\mathbf{r}_n) \quad (10)$$

## 4 Programmstruktur

## 5 Ergebnisse

## 6 Vergleich FTCS- und BTCS