

Berechnung der Flugbahn eines Partikels im zeitabhängigen Strömungsfeld

1 Ziel des Projekts

In einem zeitabhängigen Strömungsfeld soll die Bahn eines masselosen Partikels bestimmt werden. Aus einer Finiten-Elemente-Simulation der Kármánschen¹ Wirbelstraße liegen die Werte des Geschwindigkeitsfelds auf einem äquidistanten Rechteckgitter vor.

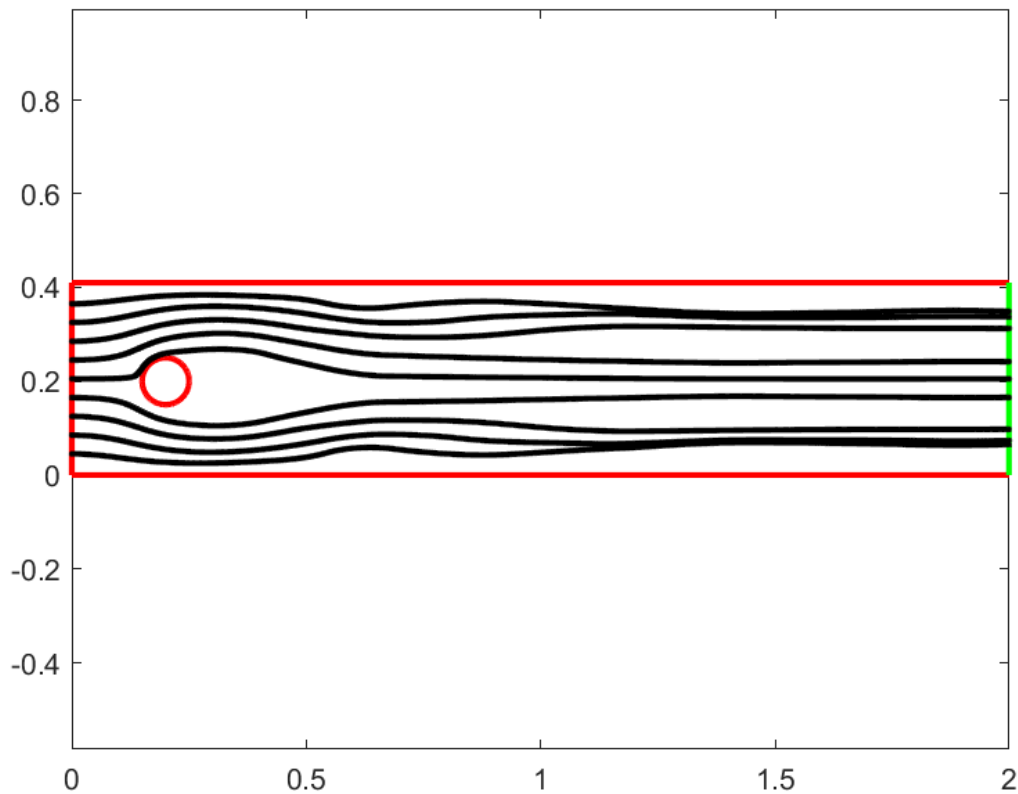


Abbildung 1: Bahnlinien mit Umströmung eines Körpers.

¹Theodore von Kármán (1881-1963)

2 Streamlines

Es soll die Bewegungsdifferentialgleichung

$$\dot{x}(t) = u(t, x(t)), \quad x(0) = x_0 \quad (2.1)$$

auf dem Intervall $t = [0, T]$ gelöst werden. Hierzu wird eine äquidistante Zerlegung gewählt

$$t_k = k \cdot dt, \quad k = 0, \dots, N-1, \quad \text{mit} \quad dt = T/(N-1). \quad (2.2)$$

Gegeben ist neben dem Startwert x_0 auch das Geschwindigkeitsfeld $u(0) = u_1$ und $u(T) = u_2$. Alle Werte auf dem Gitter dazwischen werden entsprechend der Gleichung 2.3 linear interpoliert.

$$u_k = u_1 + \gamma_k \cdot (u_2 - u_1) \quad \text{mit} \quad \gamma_k = \frac{k}{N-1}, \quad k = 0, \dots, N-1 \quad (2.3)$$

3 Funktionsaufruf

Zur Lösung des Anfangswertproblems (2.1) wird die Funktion

```
function [x1,y1] = pathline(u1,v1,u2,v2,x,y,x0,y0,T,N,method)
```

implementiert. Hier stellen $u1, v1$ die Komponenten des Geschwindigkeitsfelds in x- und y-Richtung zum Zeitpunkt $t = 0$ und $u2, v2$ die Komponenten des Geschwindigkeitsfelds zum Zeitpunkt $t = T$ dar. Die Länge des Zeitintervalls ist T und die Position des Partikels soll an N Punkten im Zeitintervall ausgewertet werden. Die Position (x_0, y_0) zum Zeitpunkt $t = 0$ ist hierbei bekannt. Die Eingabeparameter (x, y) enthalten die Koordinaten des äquidistanten Gitters, auf dem die Geschwindigkeitsfelder jeweils gegeben sind. Zusätzlich kann die numerische Methode zur Lösung der gewöhnlichen Differentialgleichung gewählt werden.

4 Implementierte Verfahren

Zur Lösung der gewöhnlichen Differentialgleichung kann aus den folgenden numerischen Methoden gewählt werden: Explizites Eulerverfahren, verbessertes Eulerverfahren, Prädiktor-Korrektor-Verfahren, implizites Eulerverfahren und Runge-Kutta-Verfahren.

4.1 Explizites Eulerverfahren

Zuerst wurde das explizite Eulerverfahren mit der folgenden Iterationsvorschrift implementiert:

$$x_{k+1} = x_k + dt \cdot u(t_k, x_k), \quad k = 0, \dots, N-1 \quad (4.1)$$

Zur Bestimmung von $u(t_k, x_k)$: Mit der in Abschnitt 2 beschriebenen Interpolation kann ein Geschwindigkeitsfeld $U(x_k)$ zum Zeitpunkt t_k definiert werden. Mittels der Funktion

```
function [u,v] = getValueMonic(U,V,dx,dy,sx,sy,x1k,y1k)
```

wird die Geschwindigkeit des Partikels an der Position (x_k, y_k) im übergebenen Feld (U, V) berechnet. Hierbei ist (dx, dy) der Abstand der Gitterpunkte, (sx, sy) die Anzahl der Gitterpunkte und die Werte der Positionen (x_k, y_k) seien so verschoben, dass das Gitter im Ursprung anliegt.

4.2 Verbessertes Eulerverfahren

Beim verbesserten Eulerverfahren wird der Geschwindigkeitswert nicht am Anfang, sondern in der Mitte des Intervalls gewählt:

$$x_{k+1} = x_k + dt \cdot u(t_{k+1/2}, x_{k+1/2}) \quad (4.2)$$

mit $t_{k+1/2} := t_k + \frac{dt}{2}$ und $x_{k+1/2} := x_k + \frac{dt}{2} \cdot u(t_k, x_k)$. Zunächst muss also der Wert $u(t_k, x_k)$ wie oben bestimmt werden. Anschließend soll ein neues Geschwindigkeitsfeld zum Zeitpunkt $t_{k+1/2}$ bestimmt werden. Hierzu lautet die Interpolationsvorschrift

$$u_{k+1/2} = u_1 + \gamma_{k+1/2} \cdot (u_2 - u_1) \quad \text{mit} \quad \gamma_{k+1/2} = \frac{k + 1/2}{N - 1}, \quad k = 0, \dots, N-2. \quad (4.3)$$

4.3 Prädiktor-Korrektor-Verfahren

Ein Prädiktor-Korrektor-Verfahren mit einfacher Korrektur ist das Euler-Heun-Verfahren. Hierbei wird zunächst ein Eulerschritt

$$\tilde{x}_{k+1} = x_k + dt \cdot u(t_k, x_k) \quad (4.4)$$

vorhergesagt und anschließend durch

$$x_{k+1} = x_k + \frac{dt}{2} (u(t_k, x_k) + u(t_k, \tilde{x}_{k+1})) \quad (4.5)$$

korrigiert. Bei der Implementierung wird der Eulerschritt wie in Unterabschnitt 4.1 durchgeführt, anschließend ein neues Geschwindigkeitsfeld zum Zeitpunkt t_{k+1} interpoliert und der Wert der Geschwindigkeit für x_{k+1} ausgewertet. Eine Erweiterung mit mehreren Korrekturen ist das Crank-Nicolson-Verfahren. Hierbei wird beispielsweise ausgehend von einem Eulerschritt $y_{k+1}^{(0)}$ eine Fixpunktiteration

$$x_{k+1}^{(i+1)} = x_k + \frac{dt}{2} (u(t_k, x_k) + u(t_k, x_{k+1}^{(i)})), \quad i = 0, \dots, \text{maxIt} \quad (4.6)$$

definiert. Sie wird abgebrochen, falls eine maximale Anzahl an Iterationen maxIt oder das Abbruchkriterium $\|x_{k+1}^{(i+1)} - x_{k+1}^{(i)}\| < \varepsilon$ erfüllt ist. Die Implementierung erfolgt analog zum Euler-Heun-Verfahren. Lediglich die Geschwindigkeit an der Stelle $x_{k+1}^{(i)}$ muss in jeder inneren Iteration neu bestimmt werden.

4.4 Implizites Eulerverfahren

Die Idee eines Fixpunktverfahrens wie bei Crank-Nicolson kann auch auf das implizite Eulerverfahren angewandt werden. Die Iterationsvorschrift lautet hier

$$x_{k+1} = x_k + dt \cdot u(t_{k+1}, x_{k+1}). \quad (4.7)$$

Es wird also wieder ausgehend von einem Eulerschritt $x_{k+1}^{(0)}$ eine Fixpunktiteration

$$x_{k+1}^{(i+1)} = x_k + dt \cdot u(t_{k+1}, x_{k+1}^{(i)}), \quad i = 0, \dots, \text{maxIt} \quad (4.8)$$

durchgeführt, welche abgebrochen wird, sobald eine maximale Anzahl an Iterationen oder ein Abbruchkriterium bezüglich einer Toleranz erreicht ist. Hierzu muss bei der Implementierung lediglich ein Mal das Geschwindigkeitsfeld zum Zeitpunkt t_{k+1} interpoliert und anschließend für alle inneren Iterierten ausgewertet werden.

4.5 Runge-Kutta-Verfahren

Zur Lösung der gewöhnlichen Differentialgleichung mittels eines Mehrschrittverfahrens wurden zwei verschiedene Varianten eines Runge-Kutta-Verfahrens vierter Ordnung implementiert. Die erste Variante folgt aus der Newtonschen $3/8$ -Regel.

$$\begin{aligned}
 k_1 &= u(t_k, x_k) \\
 k_2 &= u\left(t_{k+1/3}, x_k + \frac{dt}{3} \cdot k_1\right) \\
 k_3 &= u\left(t_{k+2/3}, x_k - \frac{dt}{3} \cdot k_1 + dt \cdot k_2\right) \\
 k_4 &= u(t_{k+1}, x_k + dt(k_1 - k_2 + k_3)) \\
 x_{k+1} &= x_k + \frac{dt}{8} \cdot (k_1 + 3(k_2 + k_3) + k_4)
 \end{aligned} \tag{4.9}$$

Alternativ wurde zudem die klassische Variante implementiert

$$\begin{aligned}
 k_1 &= u(t_k, x_k) \\
 k_2 &= u\left(t_{k+1/2}, x_k + \frac{dt}{2} \cdot k_1\right) \\
 k_3 &= u\left(t_{k+1/2}, x_k + \frac{dt}{2} \cdot k_2\right) \\
 k_4 &= u(t_{k+1}, x_k + dt \cdot k_3) \\
 x_{k+1} &= x_k + \frac{dt}{6} \cdot (k_1 + 2(k_2 + k_3) + k_4)
 \end{aligned} \tag{4.10}$$

$$\begin{aligned}
 k_1 &= u(t_k, x_k) \\
 k_2 &= u\left(t_{k+1/2}, x_k + \frac{dt}{2} \cdot k_1\right) \\
 k_3 &= u\left(t_{k+1/2}, x_k + \frac{dt}{2} \cdot k_2\right) \\
 k_4 &= u(t_{k+1}, x_k + dt \cdot k_3) \\
 x_{k+1} &= x_k + \frac{dt}{6} \cdot (k_1 + 2(k_2 + k_3) + k_4)
 \end{aligned} \tag{4.11}$$

Hierzu müssen in allen relevanten Zeitschritten die Geschwindigkeitsfelder analog zu (4.3) interpoliert und anschließend an den entsprechenden Orten ausgewertet werden.

5 Vergleich der Laufzeiten

Im Folgenden werden die Laufzeiten der verschiedenen implementierten Verfahren gegeneinander verglichen. Hierfür wurde jeweils eine unterschiedliche Anzahl an Zwischenschritten zur Berechnung der Lösung verwendet.

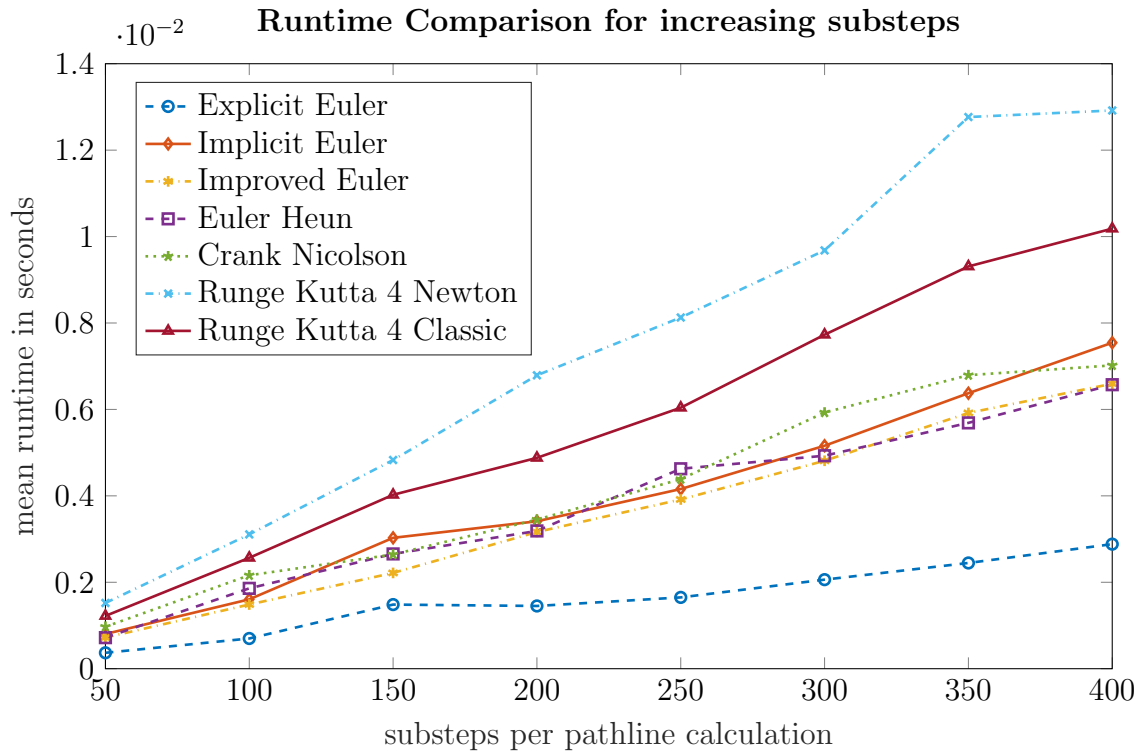


Abbildung 2: Vergleich der gemittelten Laufzeiten für unterschiedliche Anzahl an Zwischenschritten.

Wie in Abbildung 2 zu sehen ist, scheint das explizite Eulerverfahren zur Lösung des hier vorliegenden Anfangswertproblems am besten geeignet zu sein. Hier beträgt die Laufzeit zur Berechnung der Lösung deutlich weniger, als bei den anderen Verfahren und reagiert auch nicht so empfindlich auf die Erhöhung der Anzahl an Zwischenschritten.

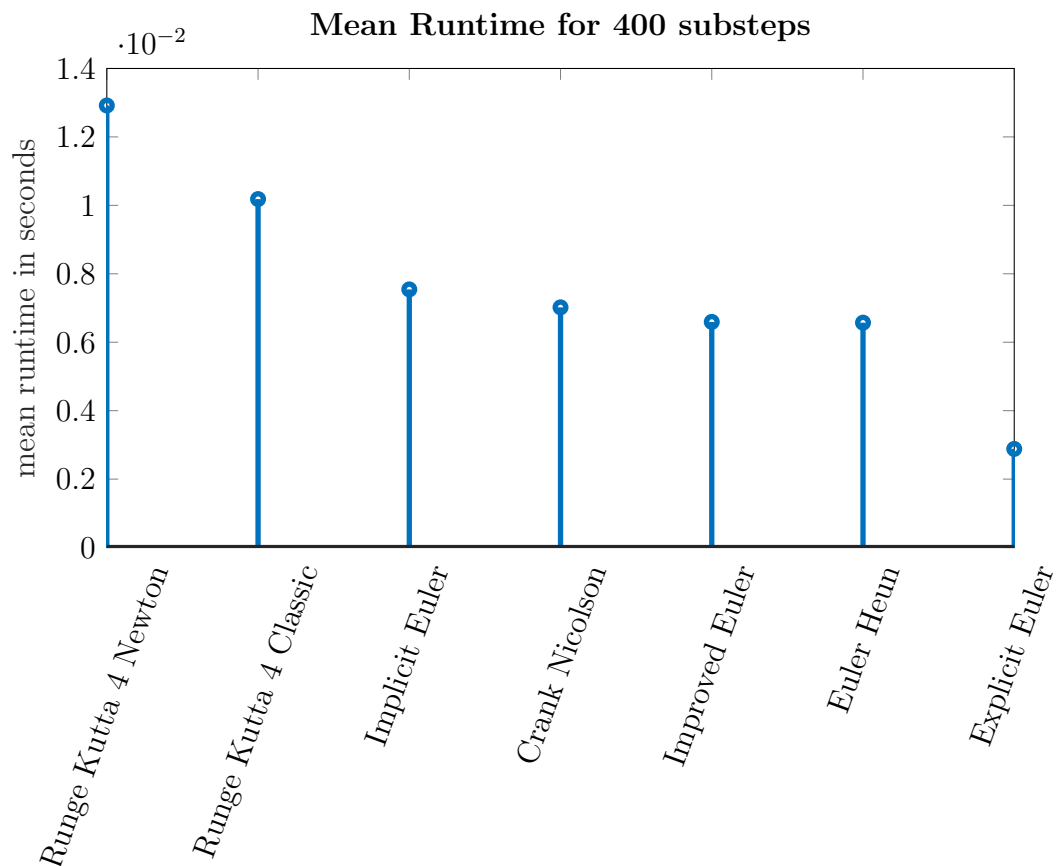


Abbildung 3: Sortierte Laufzeiten der unterschiedlichen numerischen Lösungsmethoden für 400 Zwischenschritte.

Mehrschrittverfahren liefern zwar bei einer kleineren Anzahl an Zeitschritten bessere Ergebnisse, benötigen hierfür jedoch auch mehr Rechenzeit, insbesondere je höher die Anzahl an Zwischenschritten wird. Aufgrund der hier schon sehr klein gewählten Schrittweite, lohnt sich die Verwendung dieser Verfahren nicht.

Die übrigen Verfahren ähneln sich im Hinblick auf die Laufzeit sehr und zeigen eine vergleichbare Entwicklung bei wachsender Zwischenschrittzahl.