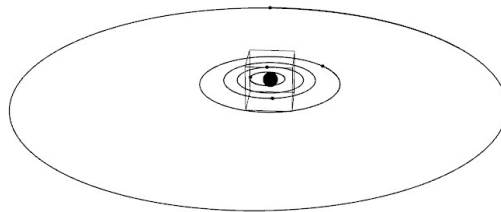


# Projektbericht: $n$ -Körper Simulation

Clemens Anschütz    Markus Pawellek  
clemens.anschuetz@uni-jena.de    markuspawellek@gmail.com



---

## Zusammenfassung

*In diesem Projekt haben wir uns mit der numerischen Lösung des  $n$ -Körper Problems beschäftigt. Hierbei sind wir vor allem auf die Vor- und Nachteile verschiedener numerischer Integratoren eingegangen.*

---

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Physikalische Grundlagen</b>	<b>2</b>
2.1	Das $n$ -Körper-Problem . . . . .	2
2.2	Das Zweikörperproblem . . . . .	4
2.3	Bahnelemente . . . . .	4
2.4	Dreikörperproblem . . . . .	4
<b>3</b>	<b>Methoden und Implementierung</b>	<b>4</b>
3.1	Programmstruktur . . . . .	4
3.2	Integratoren . . . . .	5
3.3	Explizites Euler-Verfahren . . . . .	6
3.4	Symplektisches Euler-Verfahren . . . . .	6
3.5	Leapfrog-Verfahren . . . . .	7
3.6	Runge-Kutta-Verfahren . . . . .	7
3.7	Adaptiver Zeitschritt . . . . .	7
3.8	Skalierung . . . . .	7

<b>4</b>	<b>Ergebnisse</b>	<b>7</b>
<b>5</b>	<b>Fazit</b>	<b>7</b>



# Projektbericht: $n$ -Körper Simulation

Clemens Anschütz      Markus Pawellek  
clemens.anschuetz@uni-jena.de      markuspawellek@gmail.com

---

## Zusammenfassung

*In diesem Projekt haben wir uns mit der numerischen Lösung des  $n$ -Körper Problems beschäftigt. Hierbei sind wir vor allem auf die Vor- und Nachteile verschiedener numerischer Integratoren eingegangen.*

---

## 1 Einleitung

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore ma-

gna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fu-

giat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore ma-

gna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## 2 Physikalische Grundlagen

### 2.1 Das $n$ -Körper-Problem

Das  $n$ -Körper-Problem betrachtet in seiner allgemeinen Form eine feste Anzahl von Punktmassen, die sich in einem Inertialsystem aufgrund der zwischen ihnen herrschenden Gravitationskraft bewegen.

Seien also  $n \in \mathbb{N}$  die Anzahl der Punktmassen,  $m_i \in \mathbb{R}^+$  die Masse und  $r_i: \mathbb{R}^+ \rightarrow \mathbb{R}^3$  die Kurve der  $i$ . Punktmasse für alle  $i \in \mathbb{N}$  mit  $i \leq n$ . Nach dem zweiten Newtonschen Axiom und dem Newtonschen Gravitationsgesetz mit der Gravitationskonstanten  $\gamma$  müssen die Kurven der Punktmassen den folgenden Differentialgleichungen für alle  $i \in \mathbb{N}$  mit  $i \leq n$  und alle  $t \in \mathbb{R}^+$

gehören.

$$m_i \ddot{r}_i(t) = \sum_{j=1, j \neq i}^n \gamma m_i m_j \frac{r_j(t) - r_i(t)}{\|r_j(t) - r_i(t)\|^3}$$

Der Einfachheit halber definieren wir die Beschleunigung  $a_i$  einer Punktmasse für alle  $i \in \mathbb{N}$  mit  $i \leq n$ .

$$a_i: \mathbb{R}^{3n} \rightarrow \mathbb{R}^3$$

$$a_i(r_1, \dots, r_n) := \sum_{j=1, j \neq i}^n \gamma m_j \frac{r_j - r_i}{\|r_j - r_i\|^3}$$

Für alle  $i \in \mathbb{N}$  mit  $i \leq n$  vereinfachen sich dann die Differentialgleichungen zu der folgenden Form.

$$\ddot{r}_i(t) = a_i(r_1(t), \dots, r_n(t))$$

Die Beschleunigung einer jeden Punktmasse hängt damit für jeden gegebenen Zeitpunkt vom gesamten Konfigurationsraum ab. Aus diesem Grund erscheint es sinnvoll die gesamte Problemstellung in den Konfigurationsraum zu überführen. Hierzu definieren wir als Erstes die verallgemeinerte Kurve im Konfigurationsraum.

$$r: \mathbb{R}^+ \rightarrow \mathbb{R}^{3n}$$

$$r(t) := (r_1(t), \dots, r_n(t))$$

Der nächste Schritt besteht darin, eine verallgemeinerte Beschleunigung im Konfigurationsraum zu definieren.

$$a: \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$$

$$a \circ r(t) := (a_1 \circ r(t), \dots, a_n \circ r(t))$$

Mit diesen Definitionen lässt sich das Problem nun wie folgt für alle  $t \in \mathbb{R}^+$  formulieren.

$$\ddot{r}(t) = a \circ r(t)$$

Bei dem gegebenen Gleichungssystem handelt es sich um ein gewöhnliches  $3n$ -dimensionales Differentialgleichungssystem zweiter Ordnung, welches sich, sofern  $n \geq 3$ , nur für sehr spezielle Fälle analytisch lösen lässt. Die später vorgestellten numerischen Integratoren zur Approximation der Lösung eines Anfangswertproblems sind jedoch nur für Differentialgleichungssysteme erster Ordnung anwendbar. Deshalb wollen wir zudem eine verallgemeinerte Geschwindigkeit im Konfigurationsraum einführen.

$$v: \mathbb{R}^+ \rightarrow \mathbb{R}^{3n}, \quad v := \dot{r}$$

Nach Definition muss damit auch die folgende Gleichung für alle  $t \in \mathbb{R}^+$  erfüllt sein.

$$\dot{v}(t) = a \circ r(t)$$

Diese Beziehung machen wir uns zu Nutze, um das Problem in den Phasenraum zu überführen. Wir definieren eine verallgemeinerte Kurve im Phasenraum, welche zu jedem Zeitpunkt die Orte und Geschwindigkeiten aller Punktmassen beschreibt.

$$p: \mathbb{R}^+ \rightarrow \mathbb{R}^{6n}, \quad p(t) := \begin{pmatrix} r(t) \\ v(t) \end{pmatrix}$$

Wie vorher, benötigen wir auch eine verallgemeinerte Beschleunigung im Phasenraum.

$$f: \mathbb{R}^{6n} \rightarrow \mathbb{R}^{6n}, \quad f \circ p(t) := \begin{pmatrix} v(t) \\ a \circ r(t) \end{pmatrix}$$

Aus den Definitionen von  $p$  und  $f$  folgt dann unmittelbar ein äquivalentes gewöhnliches  $6n$ -dimensionales Differentialgleichungssystem erster Ordnung in dem für alle  $t \in \mathbb{R}^+$  das folgende gilt.

$$\dot{p}(t) = f \circ p(t)$$

## 2.2 Das Zweikörperproblem

Das Zweikörperproblem ist eine spezielle Form des  $n$ -Körper-Problems, in welchem  $n = 2$  gilt und damit nur zwei Punktmassen, die sich gegenseitig beeinflussen, betrachtet werden. Das resultierende System von Differentialgleichungen lässt sich analytisch lösen und stellt gerade aus diesem Grund eine Möglichkeit dar numerische Integratoren in Bezug auf ihre Stabilität, Genauigkeit und Geschwindigkeit zu quantifizieren.

- Lösungen
- Energieerhaltung
- Drehimpulserhaltung

## 2.3 Bahnelemente

Ist die Lösung des Zweikörperproblems beschränkt beziehungsweise gebunden, so bewegen sich wie im vorigen Abschnitt erklärt, die beiden Punktmassen auf Ellipsen im Raum. Um die Bewegung einer Punktmasse zu beschreiben, sind, wie aus der theoretischen Mechanik bekannt, sechs Koordinaten notwendig. Im kartesischen Inertialsystem, indem auch das Problem formuliert wurde, charakterisieren drei dieser Koordinaten die Position und drei die Geschwindigkeit. Dieses System eignet sich für die numerische Simulation.

In der Astronomie sind diese sechs Koordinaten aber schwer zu messen. Aus diesem Grund haben sich die sogenannten Bahnelemente etabliert, die im klassischen Fall ebenfalls durch sechs Parameter  $(p, e, i, \Omega, \omega, t)$  angegeben werden. Hierbei bezeichnet  $p$  den Halbparameter,  $e$  die Exzentrizität,  $i$  die Inklination,  $\Omega$

die Winkel des aufsteigenden Knotens,  $\omega$  das Argument der Periapsis und  $t$  den Zeitbezug. Sie beschreiben direkt die Lage und Form der Ellipsen im Raum und eignen sich gerade deshalb zur quantitativen Beschreibungen von Planetenbahnen. Zur Beschreibung von Planeten eines Sonnensystems wird statt  $p$  auch oft die große Halbachse  $a$  verwendet. In Abbildung ?? ist noch einmal die Bedeutung einiger Bahnelemente skizziert.

$$a = -\gamma m M / 2e =$$

- Zweikörperproblem
- allgemeines  $n$ -Körper Problem

## 2.4 Dreikörperproblem

Auch das Dreikörperproblem ist eine spezielle Form des  $n$ -Körper-Problems, in welchem  $n = 3$  gilt. Es ist im allgemeinen nicht mehr analytisch lösbar. Allerdings existieren geschlossene Lösungen für diverse Spezialfälle.

# 3 Methoden und Implementierung

## 3.1 Programmstruktur

Das von uns erstellte Programm dient der numerischen Approximation von Lösungen des  $n$ -Körper-Problems für gegebene Anfangswerte. Um die Auswertung von Ergebnissen zu vereinfachen und intuitiver zu gestalten, implementierten wir neben den Integratoren eine graphische Ausgabe, eine einfache Form der Eingabe und diverse Möglichkeiten, um zwischen verschiedenen Einstellungen und

Integratoren zur Laufzeit des Programmes zu wechseln.

Bei der Ausgabe handelt es sich um eine perspektivische Projektion des betrachteten Raums auf den Bildschirm einer virtuellen Kamera. Position und Ausrichtung der Kamera lassen sich durch Mausinteraktionen beliebig einstellen. Zur Orientierung und Größeneinschätzung wird das Gitter eines Würfels mit Kantenlänge 1 im Koordinatensprung gezeigt. Die Punktmassen werden durch Kugeln dargestellt, deren Radien entsprechend ihrer Masse und ihres Abstandes zur Kamera skaliert werden. Um die zurückgelegten Kurven der Partikel zu visualisieren, speicherten wir eine feste Anzahl von Bahnpunkten und verbanden diese durch Strecken. Das Tracken der Bahnpunkte erfolgte nicht nach jedem Zeitschritt, sondern wurde durch ein Kriterium festgelegt, welches sicherstellte, dass nur Punkte festgehalten wurden, die eine signifikante Richtungsänderung der Kurve hervorriefen. In Abbildung ?? werden die angegebenen Methoden noch einmal anhand eines Beispiels demonstriert.

Um die noch folgenden Integratoren zu verstehen und implementieren zu können, muss im Vorfeld über eine feste Datenstruktur, welche beschreibt, wie Positionen, Geschwindigkeiten und Massen der einzelnen Partikel zu speichern sind, entschieden werden. Zwei typische Varianten für eine solche Datenstruktur werden mit AOS (Array-of-Structure) und SOA (Structure-of-Array) bezeichnet. Im Folgenden sind für diese beiden Fälle einfache C++-Quelltexte ange-

geben. In beiden Fällen werden auf die C++-Standardbibliothek und die Bibliothek Eigen zugegriffen. Im Programm legten wir uns auf die AOS-Datenstruktur fest, da sie intuitiver im Umgang ist.

Quelltext: SOA-Partikelsystem-Datenstruktur

```
#include <Eigen/Dense>
#include <vector>

struct Particle_system {
    std::vector<Eigen::
        Vector3f> positions;
    std::vector<Eigen::
        Vector3f> velocities;
    std::vector<float> masses
        ;
};
```

Quelltext: AOS-Partikelsystem-Datenstruktur

```
#include <Eigen/Dense>
#include <vector>

struct Particle {
    Eigen::Vector3f position;
    Eigen::Vector3f velocity;
    float mass;
};

using Particle_system = std
    ::vector<Particle>;
```

### 3.2 Integratoren

Integratoren oder auch numerische Integratoren bezeichnen in der numerischen Mathematik Algorithmen, die entweder den Wert eines bestimmten Integrals oder die Lösung einer Differentialgleichung numerisch approximieren. Wir möchten uns hier auf die numerische Lösung von gewöhnlichen Differentialgleichungen beschränken. Integratoren wer-



den immer dann verwendet, wenn analytische Lösungen nicht existieren oder zu kompliziert sind. In diesen Fällen reicht es aus, genäherte Lösungen mit beliebig kleinem Fehler zu bestimmen.

Gerade beim  $n$ -Körper-Problem für  $n \geq 3$  ist es demnach notwendig Integratoren zu verwenden, da allgemeine geschlossene Lösungen nicht existieren. Alle der hier verwendeten Integratoren lösen gewöhnliche Differentialgleichungssysteme erster Ordnung. Allerdings lässt sich das  $n$ -Körper-Problem wie in Abschnitt 2.1 gezeigt, in ein solches System überführen.

Integratoren lassen sich nach ihrer Stabilität oder Robustheit, ihrer Genauigkeit und ihrer Geschwindigkeit einordnen. Die Geschwindigkeit ist durch eine einfache Zeitmessung bestimmbar. Um die Stabilität eines Integrators einzuschätzen, betrachten wir die Gesamtenergie des Systems, welche nach dem Energieerhaltungssatz konstant bleiben muss. Die Simulation eines analytisch lösbaren Problems, wie zum Beispiel des Zweikörperproblems, macht es zudem möglich die Genauigkeit eines Integrators zu bewerten.

In den folgenden Abschnitten betrachten wir einige typische Integratoren, implementieren diese und schätzen sie nach den genannten Kriterien ein. Für  $k \in \mathbb{N}$  nennen wir  $t_k \in \mathbb{R}^+$  den  $k$ . Zeitpunkt mit der Bedingung  $t_k < t_{k+1}$  für alle  $k \in \mathbb{N}$ . Zudem nennen wir den  $\Delta t := t_{k+1} - t_k$  den Zeitschritt für äquidistante Zeitdiskretisierungen. Des Weiteren definieren wir  $p_k$  als die numerische Approximation des gewählten Integrators von  $p(t_k)$  für alle  $k \in \mathbb{N}$ .

### 3.3 Explizites Euler-Verfahren

Das explizite Euler-Verfahren ist eines der simpelsten und intuitivsten Verfahren zur Approximation von gewöhnlichen Differentialgleichungen erster Ordnung. Dieses einfache Verfahren nähert die Zeitableitung in der Differentialgleichung durch den diskreten rechtsseitigen Differenzenquotient. Gemäß unserer Notation aus Abschnitt 2.1 kann der explizite Euler-Algorithmus für alle  $k \in \mathbb{N}$  mit gegebenen Anfangswerten  $p_1$  wie folgt aufgeschrieben werden.

$$p_{k+1} = p_k + \Delta t \cdot f(p_k)$$

Eine äquivalente Formulierung für die Kurven des Konfigurationsraumes ist im folgenden Gleichungssystem gegeben.

$$\begin{pmatrix} r_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} r_k \\ v_k \end{pmatrix} + \Delta t \cdot \begin{pmatrix} v_k \\ a(r_k) \end{pmatrix}$$

Das explizite Euler-Verfahren konvergiert in erster Ordnung mit dem Zeitschritt  $\Delta t$ . Im Allgemeinen erhält es nicht die Energie des betrachteten Systems. CODEBEISPIELE

### 3.4 Symplektisches Euler-Verfahren

Verfahren wie der explizite Euler, die keine Energieerhaltung aufweisen, neigen dazu numerische Fehler in jedem Schritt aufzusummieren, wodurch die numerische Lösung mit der Zeit immer weiter von der analytischen weg driftet. Ein Lösungsansatz für dieses Problem sind symplektische Verfahren. Sie bestehen oft aus einer Mischung von expliziten und impliziten Methoden. Als einfachstes sei hier das semi-explizite Euler-

Verfahren gezeigt.

$$\begin{pmatrix} r_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} r_k \\ v_k \end{pmatrix} + \Delta t \cdot \begin{pmatrix} v_{k+1} \\ a(r_k) \end{pmatrix}$$

In der Umsetzung muss darauf geachtet werden zuerst die zweite Zeile des Gleichungssystems zu berechnen, da diese die für die erste Zeile notwendige Geschwindigkeit  $v_{k+1}$  liefert. Genau wie das explizite Euler-Verfahren konvergiert es in erster Ordnung. Im folgenden ist ein CODEBEISPIEL gezeigt.

### 3.5 Leapfrog-Verfahren

Ein weiteres symplektisches Verfahren ist das Leapfrog-Verfahren. Wie in den folgenden Gleichungen zu sehen, bezieht die Leapfrog-Methode auch Terme höherer Ordnung  $\Delta t^2$  in die Integration mit ein.

$$\begin{pmatrix} r_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} r_k \\ v_k \end{pmatrix} + \begin{pmatrix} v_k \Delta t + \frac{1}{2} a(r_k) \Delta t^2 \\ \frac{1}{2} (a(r_k) + a(r_{k+1})) \Delta t \end{pmatrix}$$

Das Verfahren konvergiert in zweiter Ordnung und ist damit präziser als die beiden zuvor eingeführten Euler-Verfahren. Die Implementierung kann wie im folgenden BEISPIEL gezeigt erfolgen.

### 3.6 Runge-Kutta-Verfahren

Das klassische Runge-Kutta-Verfahren (RK4) ist ein nicht-symplektisches In-

tegrationsverfahren, welches in vierter Ordnung konvergiert. Es ist eines der am häufigsten verwendeten Verfahren für gewöhnliche Differentialgleichungen. Die Idee des RK4 ist, in jedem Integrations-schritt die Ableitung an vier Punkten, dem Intervallanfangspunkt, zweimal in der Intervallmitte und dem Intervallendpunkt, abzuschätzen. Die Berechnung kann, wie in den folgenden Gleichungen dargestellt, erfolgen.

$$p_{k+1} = p_k + \frac{\Delta t}{6} \cdot (q_1 + 2q_2 + 2q_3 + q_4)$$

$$\begin{aligned} q_1 &= f(p_k) & q_2 &= f\left(p_k + \frac{\Delta t}{2} q_1\right) \\ q_3 &= f\left(p_k + \frac{\Delta t}{2} q_2\right) & q_4 &= f(p_k + \Delta t q_3) \end{aligned}$$

Es existieren neben dem RK4 auch noch weitere und allgemeinere Runge-Kutta-Verfahren für verschiedene Stufen und Konvergenzordnungen. Im Anschluss ist ein Beispielquelltext zur Implementierung des RK4 gezeigt.

### 3.7 Adaptiver Zeitschritt

### 3.8 Skalierung

## 4 Ergebnisse

## 5 Fazit