

# Projektbericht – Ideal Gases

## **Einleitung:**

Im Rahmen des studentischen Projekts "Simulation idealer Gase" haben wir, ein Team aus Studierenden, uns darangesetzt, eine interaktive Lernumgebung zu konzipieren und umzusetzen. Unser erklärtes Ziel war es, gerade Studienanfängern der Ingenieurwissenschaften ein vertieftes, praktisches Verständnis von idealen Gasen zu ermöglichen. Hierfür haben sie die Möglichkeit, in einem geschlossenen idealen Gassystem die Auswirkungen von Veränderungen der Zustandsvariablen und Konstanten (wie Druck, Volumen, Temperatur, Innere Energie und Entropie) zu erkunden. Unsere Simulation haben wir in der Programmiersprache Julia entwickelt, mit besonderem Augenmerk auf eine modulare Struktur und eine einfache, pädagogisch wertvolle Darstellung der Zusammenhänge.

Die berechneten Werte ergeben sich aus einer makroskopischen Betrachtung des Systems mit Hilfe des idealen Gasgesetzes.

## **Beiträge:**

### **Jan Wichmann:**

- Initialisierung der "realen" Parameter des idealen Gases: Zu Beginn des Projekts musste ich die realen Parameter des idealen Gases (wie Volumen, Druck, Temperatur usw.) in die Simulation einführen. Diese Eingangsparameter bildeten die Grundlage für die nachfolgenden Berechnungen.
- Anpassung der „agent\_step!“ Funktion für eine effizientere Berechnung der Parameter bei Kollisionen der Partikel, unter Berücksichtigung der Impulserhaltung.
- Implementierung der idealen Gasgleichung: Mein Hauptbeitrag bestand darin, die physikalischen Berechnungen auf Basis der idealen Gasgleichung zu implementieren und zu optimieren.
- Anpassung der Slider: Ich löste ein Anfangsproblem, indem ich eigene Slider aus dem GLMakie-Paket entwickelte, was eine dynamische Einstellung der Slider-Werte ermöglichte.
- Optimierung der Rechenkapazität: Um die Effizienz der Berechnungen zu verbessern, verlagerte ich diese in die "on"-Funktionen der Slider.
- Implementierung eines Agenten-Hinzufügen/Entfernen-Features: Als abschließendes Feature habe ich die Möglichkeit geschaffen, Agenten auf Basis der Molzahl des Gases während der Laufzeit der Simulation hinzuzufügen oder zu entfernen.

### **Melanie Heinrich:**

- Berechnung und Skalierung der mittleren Geschwindigkeit der Partikel, unter Verwendung der Formel für eine makroskopische Betrachtung. Ersetzen der (mikroskopisch berechneten) Geschwindigkeitsverteilung beim Initialisieren des Systems.
- Neuverteilung von Partikelgeschwindigkeiten bei jedem Schritt des Modells, basierend auf der neuen mittleren Geschwindigkeit berechnet aus der Temperatur, so dass vorherige Geschwindigkeits-Erhöhen/-Verringerungen visuell erhalten bleiben.
- Berechnung der inneren Energie des Gases basierend auf der Temperatur unter Einbezug der Freiheitsgrade der Moleküle des Gases.
- Berechnung der Entropieänderung abhängig von dem abgelaufenen thermodynamischen Prozess von einem Schritt zum nächsten. Anzeige der Entropieänderung über den Verlauf der Simulationsschritte in einem Diagramm. Dafür wurden zuvor die molaren und spezifischen Wärmekapazitäten des Gases berechnet unter Einbezug der Freiheitsgrade.  
Die Entropieberechnung fehlt, wenn der Nutzer sich im Modus „Mol – Temperatur“ befindet, da Unklarheit beim letzten Treffen mit dem Kunden geherrscht hat welche Formel in dem Fall zu verwenden wäre und ob dieser Fall in der Praxis so ablaufen kann.

Nicht Bewertungsrelevante Aufgaben: - Überarbeitung der graphischen User-Oberfläche – Auslagern der universellen Gaskonstante – Verhinderung von

polytropischen Prozessen durch Stoppen der Volumenveränderung bei Modus-Wechsel und in nicht Volumen relevanten Modi. – Anpassung der Größe der Simulations-UI an die Größe des Bildschirms auf dem sie ausgeführt wird – Anpassung der Abfrage der Modell Variablen

### **Francisco Hella:**

- Ich habe mehrere Funktionen entwickelt, um das Volumen optisch zu verändern und den Benutzern eine ansprechende Interaktionsmöglichkeit zu bieten.
- Ich habe die Versionskontrolle überwacht und erfolgreich den Merge-Prozess durchgeführt, um sicherzustellen, dass alle Änderungen aus den verschiedenen Branches zusammengeführt werden und das System konsistent bleibt.
- In Zusammenarbeit mit meinem Kollegen Felix habe ich eine Geschwindigkeitsberechnung entwickelt, um eine realitätsnaher Darstellung der Agents bereitzustellen und eine genau Berechnung zu gewährleisten.
- Um die Codebasis zu verbessern, habe ich die physikalischen Berechnungen in ein separates Modul ausgelagert, um die Effizienz des Codes zu verbessern und die Wartbarkeit zu erleichtern

### **Felix Rollbühler**

- Transformation des IdealGas-Projekts: Das ursprüngliche ABM-Video-Projekt habe ich zu ABM-Playground umgewandelt, wodurch eine initiale, nutzerfreundliche Oberfläche geschaffen wurde.
- Implementierung von Partikelkollisionen mit Randbereichen: Ein spezifischer Bereich wurde definiert, in dem die Partikel, aufgrund ihrer sprunghaften Bewegung, an den Rändern ihre Richtung ändern (reflektieren). Zur Vermeidung von kontinuierlichen Richtungsänderungen langsamer Partikel in diesem Randbereich habe ich ein Zähler `last_bounce` eingeführt. Dieser ermöglicht eine erneute Richtungsänderung erst nach drei Modellschritten.
- Update: Später habe ich für die Partikelkollision im Randbereich eine Abfrage eingebaut, bei der die Richtung berücksichtigt wird und die Partikel im Randbereich nur noch ihre Richtung wechseln, wenn Sie sich in Richtung Rand bewegen.
- Durchführen eine Reihe kleinerer Verbesserungen und Optimierungen am Code und an der Benutzeroberfläche.
- Einführung einer Volumenänderung mittels Zylinder: Eine bewegliche Grenze wurde implementiert, welche mit den Buttons "Increase Volume", "Decrease Volume" und "Pause" gesteuert werden kann. Bei Betätigung des "Decrease Volume"-Buttons fährt der Zylinder autonom ein, bis entweder "Pause" oder "Increase Volume" gedrückt wird. Während dieses Prozesses steht eine Variable zur Verfügung, die die aktuelle Position des Zylinders für weitere Berechnungen festhält. Bei Kollisionen der Partikel mit dem einfahrenden oder ausfahrenden Zylinder wurde besonders auf korrektes optisches Verhalten geachtet: Geschwindigkeit und Bewegungsrichtung der Partikel werden entsprechend angepasst. Sobald der Zylinder gestoppt wird, wird eine neue Grenze für den Gasraum festgelegt.
- Konsolidierung verschiedener Versionen und Fehlerbehebung: In Zusammenarbeit mit Francisco wurden verschiedene Projektversionen zusammengeführt und daraus resultierende Probleme behoben.

