Computergestütztes wissenschaftliches Rechnen
der Fakultät für Physik,
Universität Göttingen

Abschlussarbeit
Projekt 5 - Chemische Kinetik

Student:	Michael Lohmann
E-Mail:	m.lohmann@stud.uni-goettingen.de
Betreuer:	Burkhard Blobel
Versuchsdatum:	11.8.2014

Note:

Inhaltsverzeichnis

1 Aufgaben **3**

2 Runge-Kutta-Algorithmus **3**

3 Programmstruktur **3**

4 Auswertung **4**

5 Stabilität der Algorithmen **4**

Literatur **4**

1 Aufgaben

Die Bewegung von Teilchen zu bestimmen ist eine wichtige Voraussetzung um Prognosen zu erstellen, wie sich ein System verhält. Insbesondere ist es interessant (nicht nur für Chemiker) die Entwicklung der Dichte zweier verschiedener Stoffsorten, welche vermischt sind, zu beobachten, da diese in großem Maße für die Reaktionsfähigkeit entscheidend sind. Die Gleichungen, welche dieses System beschreiben lauten

$$\frac{du(t)}{dt} = a - u(t) + u^2(t) \cdot v(t) = f_u(u, t) \tag{1}$$

$$\frac{dv(t)}{dt} = b - u^2(t) \cdot v(t) = f_v(v, t) \tag{2}$$

wobei $u(t)$ und $v(t)$ die Dichten der beiden Molekülsorten sind und a und b zwei Konstanten, welche größer 0 sind. Außerdem kann die Dichte eines Stoffes natürlich nicht negativ werden, so dass $u(t)$ und $v(t)$ ebenfalls immer positiv sein müssen.

2 Runge-Kutta-Algorithmus

Der Runge-Kutta-Algorithmus ist ein Ansatz, Differentialgleichungen numerisch zu lösen. Er ist eine einfache und dabei relativ robuste Möglichkeit, Näherungen zu bekommen. Der wohl am häufigsten benutzte ist dabei derjenige 4. Ordnung. In der diskretisierten Form berechnet sich das folgende Glied aus dem vorherigen nach [1, S.130] durch

$$\boxed{y_{i+1} = y_i + (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)/6}$$

mit

$$\begin{aligned} k_1 &= \Delta t \cdot f(y_i,\, t_i) \\ k_2 &= \Delta t \cdot f(y_i + k_1/2,\, t_i + \Delta t/2) \\ k_3 &= \Delta t \cdot f(y_i + k_2/2,\, t_i + \Delta t/2) \\ k_4 &= \Delta t \cdot f(y_i + k_3,\, t_i + \Delta t) \end{aligned}$$

wobei $\dot{y} = f(y, t)$.

3 Programmstruktur

Das Programm beginnt mit der Definition der beiden Funktionen f_u und f_v, welche die jeweiligen Werte aus den Daten von u bzw. v berechnen. In der *main* werden zunächst die Variablen deklariert und soweit möglich initialisiert. Folgend werden die restlichen per Benutzereingabe eingelesen. Dabei wird auch überprüft, ob

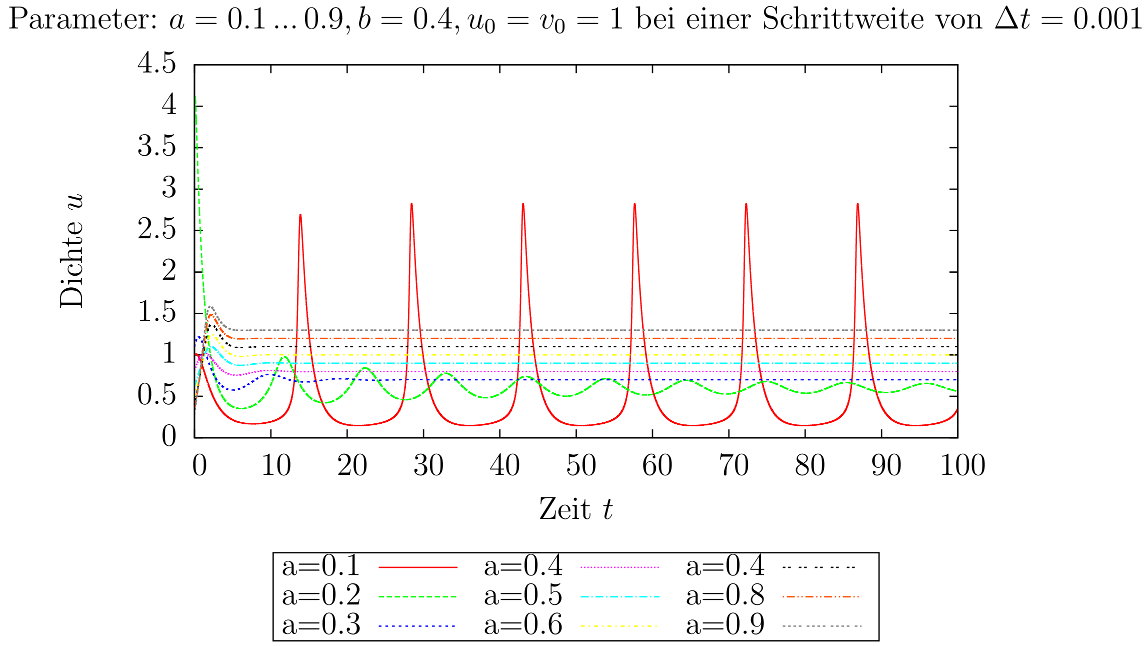


Abbildung 1: b04

4 Auswertung

Wie man in den Abb. 1 bis 6 sieht, nimmt mit größer werdendem Parameter a nicht nur der Grenzwert von u zu (von v ab), sondern die Schwingung von u und v wird im Verlauf der Zeit deutlich schneller abgebremst.

5 Stabilität der Algorithmen

Literatur

[1] J. Pitt-Francis and J. Whiteley (2012): *Guide To Scientific Computing in C++*, 1. Auflage, Springer London

Parameter: $a = 0.1 \ldots 0.9, b = 0.4, u_0 = v_0 = 1$ bei einer Schrittweite von $\Delta t = 0.001$

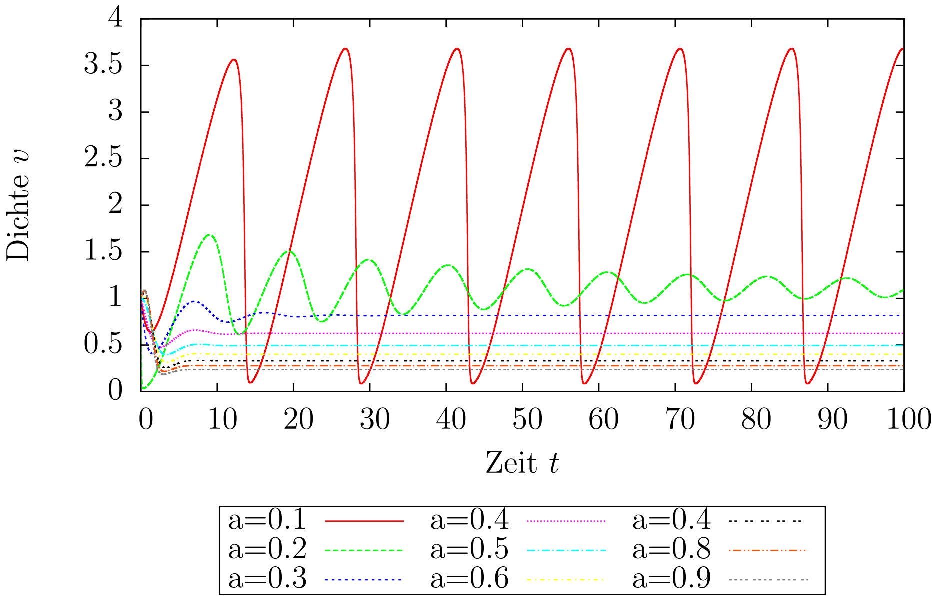


Abbildung 2: b04v

Parameter: $a = 0.1 \ldots 0.9, b = 0.4, u_0 = v_0 = 1$ bei einer Schrittweite von $\Delta t = 0.001$

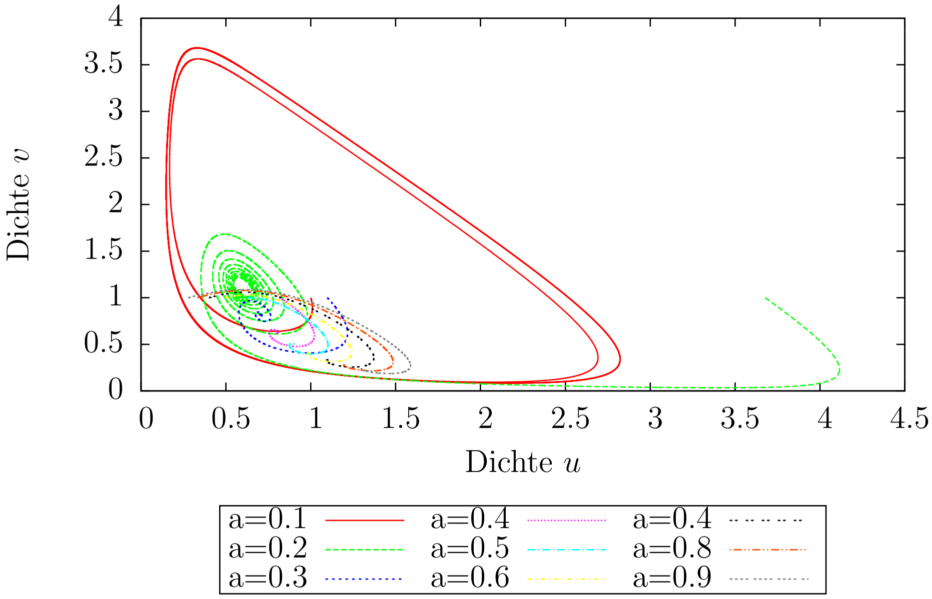


Abbildung 3: b04uv

Parameter: $a = 0.1 \ldots 0.9, b = 0.7, u_0 = v_0 = 1$ bei einer Schrittweite von $\Delta t = 0.001$

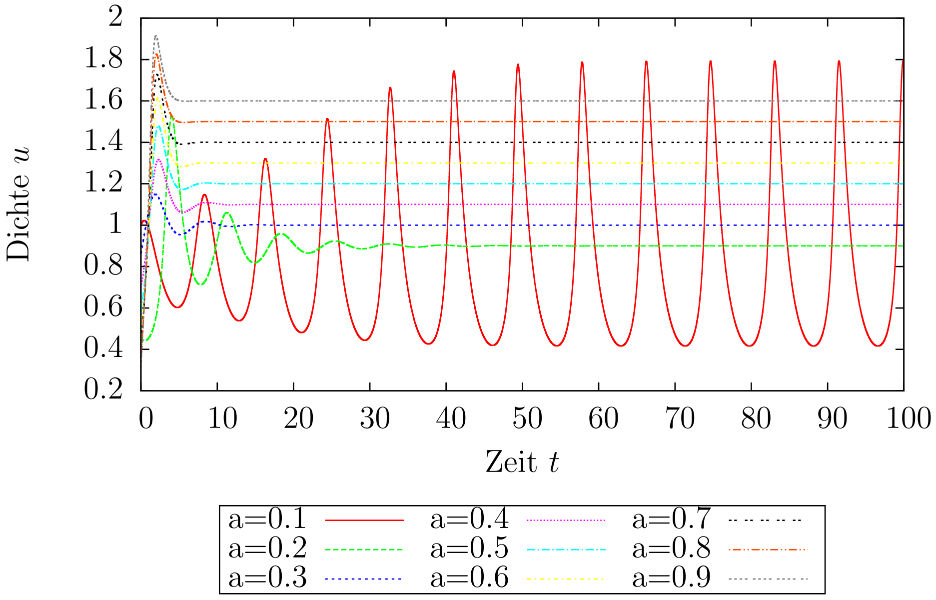


Abbildung 4: b07

Parameter: $a = 0.1 \ldots 0.9, b = 0.7, u_0 = v_0 = 1$ bei einer Schrittweite von $\Delta t = 0.001$

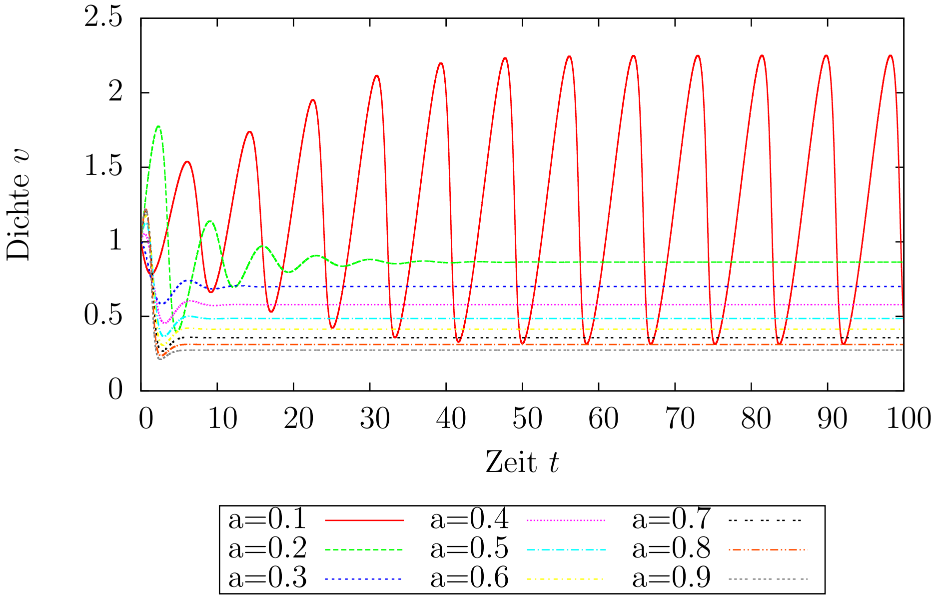


Abbildung 5: b07v

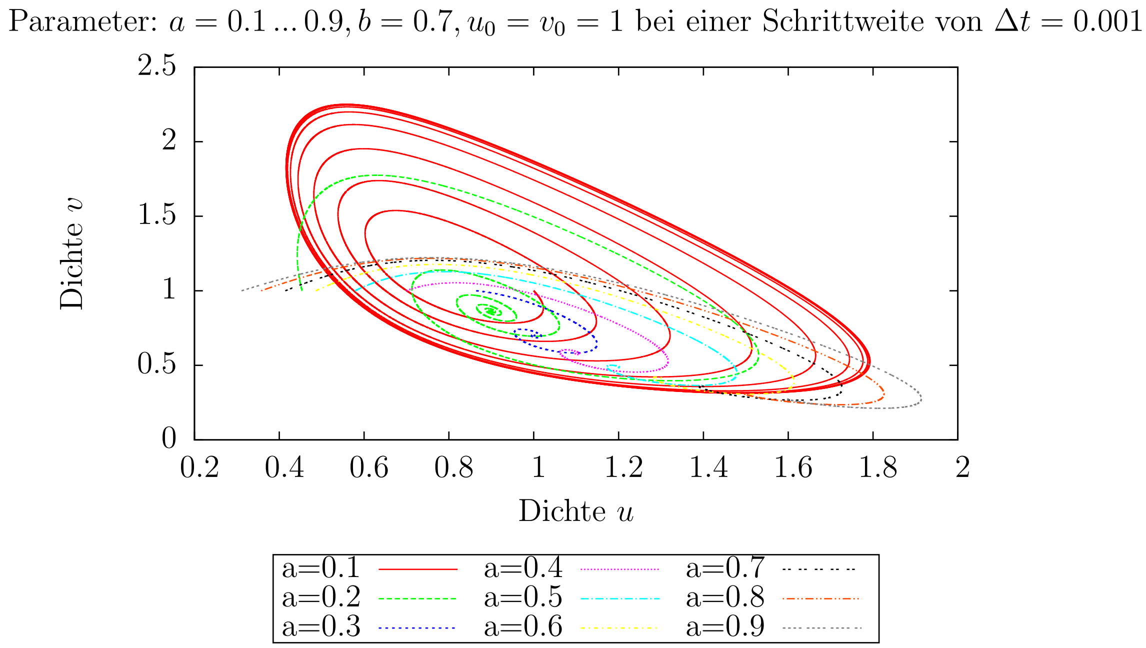
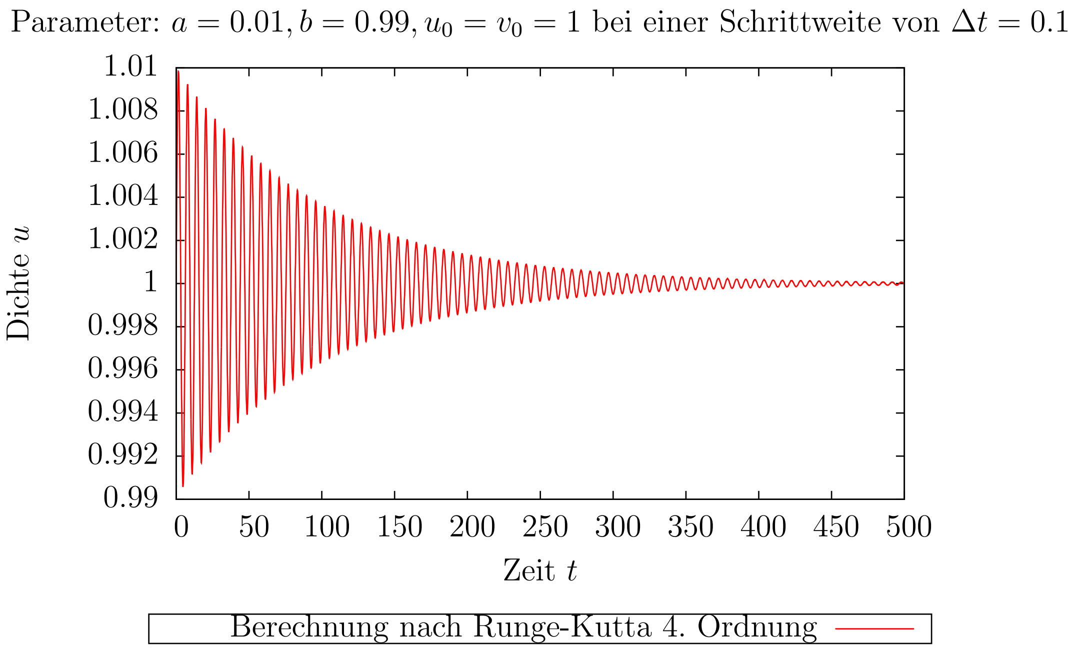
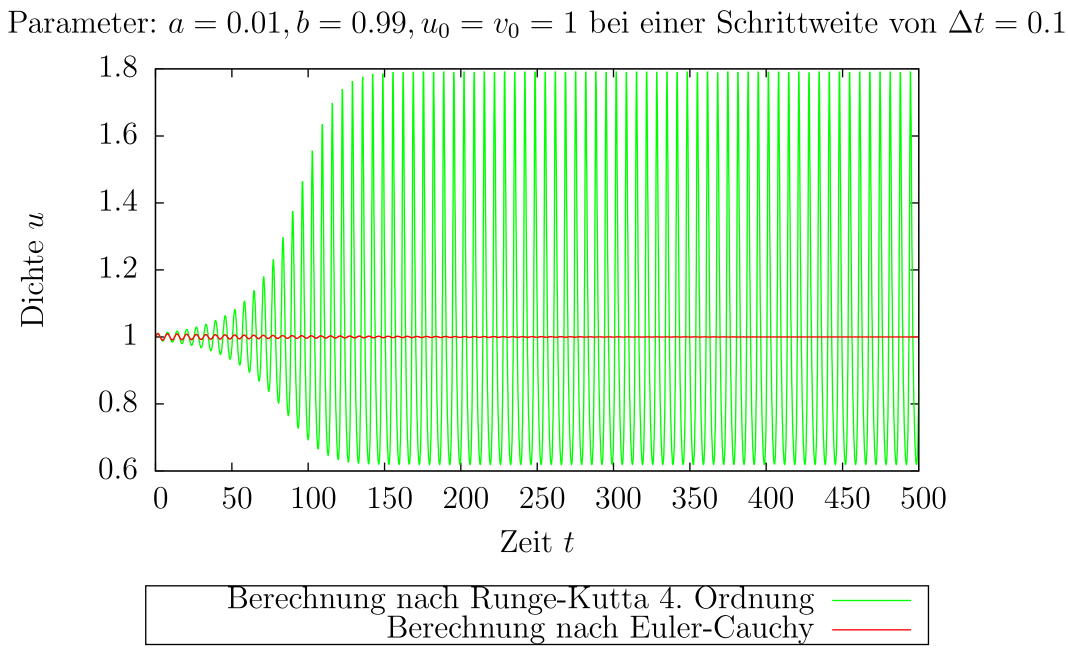


Abbildung 6: b07uv



(a) Runge-Kutta-Algorithmus 4. Ordnung



(b) Vergleich Runge-Kutta-Algorithmus mit Euler bei identischen Randbedingungen

Abbildung 7: Unterschiedliche Algorithmen im Vergleich