



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Modellbildung und Simulation 2

Ausarbeitung

Simulation eines Airsoft Projektils

August 2023

Fachbereich Elektrotechnik, Maschinenbau und Technikjournalismus

Inhaltsverzeichnis

1	Projektvorstellung	1
1.1	Motivation	1
1.2	Berücksichtigte Größen	2
1.2.1	Simulationsgrößen des Projektils	2
1.2.2	Simulationsgrößen der Airsoft Waffe	2
1.2.3	Simulationsgrößen der Umwelt	2
2	Aufbau App-Designer	3
2.1	Vorstellung der App ARTEMIS	3
2.1.1	Eingabe der Simulationsparameter (Register Parameter)	3
2.1.2	BB-Parameter	4
2.1.3	Airsoft Parameter	4
2.1.4	Simulationsparameter	4
2.1.5	Output Gauges	4
2.1.6	Calculate	4
2.1.7	Standardparameter	4
2.2	Windgeschwindigkeit X,Y,Z-Achsen	5
2.3	Simulation Plot Overview	5
3	Erstellung der Simulation	6
3.1	Schräger Wurf	6
3.1.1	Herleitung der Modellgleichungen	6
3.1.2	Darstellung Ergebnisse	7
3.2	Berücksichtigung des Luftwiderstands	8
3.2.1	Herleitung der Modellgleichungen, Annahmen und Vereinfachungen	8
3.2.2	Darstellung Ergebnisse	8
3.3	Berücksichtigung mit dynamischen Windeffekten	10
3.3.1	Herleitung der Modellgleichungen, Annahmen und Vereinfachungen	10
3.3.2	Darstellung Ergebnisse	10
3.4	Berücksichtigung des Magnus-Effekts	11
3.4.1	Herleitung der Modellgleichungen, Annahmen und Vereinfachungen	11
3.4.2	Darstellung Ergebnisse	12
3.5	Berücksichtigung der Abnahme der Rotationsgeschwindigkeit	13
3.5.1	Herleitung der Modellgleichungen, Annahmen und Vereinfachungen	13
3.5.2	Darstellung Ergebnisse	14
3.6	Implementierung im dreidimensionalen Raum	15
3.6.1	Herleitung der Modellgleichungen, Annahmen und Vereinfachungen	15

3.6.2	Darstellung Ergebnisse	16
3.6.3	Gimbal Lock	16
4	Auswertung Simulation	18
4.1	Wahl des ODE-Solver	18
4.2	Magnus-Effekt ohne Luftreibung	19
4.3	Parameteranpassung	20
5	Zusammenfassung und Ausblick	21

Abkürzungsverzeichnis

BB	Baby Bullet
ODE	Ordinary differential equation
ARTEMIS	Airsoft Range and Trajectory Evaluation using MATLAB for Interactive Simulations

Abbildungsverzeichnis

1.1	Flug einer Kugel unter Einfluss des Magnus-Effekts	2
2.1	Bedienfläche Parameter	3
2.2	Bedienfläche Windeinstellungen	5
2.3	Simulation Plot Overview	5
3.1	Vergleich Flugbahn mit und ohne Luftreibung	9
3.2	Energieverlauf über die Zeit aufgetragen unter Einfluss von Luftreibung und Gravitation	9
3.3	Flugbahn mit beispielhaftem Wind	10
3.4	Vergleich Flugbahn mit und ohne Magnus-Effekt	12
3.5	Vergleich Flugbahn mit und ohne Luftreibung an der Rotation der Kugel .	14
3.6	Abnahme der Rotationsenergie	14
4.1	Vergleich der ODE-Solver in 2D	18
4.2	Vergleich der Laufzeiten der ODE-Solver in 2D	19
4.3	Vergleich Flugbahn mit und ohne Luftreibung	19

1 Projektvorstellung

Ziel des Projektes ist es, den Flug einer Airsoft Kugel (BB) zu simulieren. Bei der Simulation sollen die physikalischen Einflüsse der Kugel, sowie die Umwelteinflüsse berücksichtigt werden. Bei diesen handelt es sich, unter anderem, um die Geschwindigkeit der Kugel, den Luftwiderstand, den Magnus-Effekt, die Erdanziehung und den Einfluss des Windes. Die Auswirkungen der Einflussfaktoren sollen mathematisch und grafisch dargestellt werden. Die Darstellung soll im zweidimensionalen und dreidimensionalen Raum erfolgen. Die Programmierung der Simulation erfolgt mit Matlab R2022b. Um die Eingabe der wichtigsten Parameter einfach zu gestalten wird eine grafische Benutzeroberfläche verwendet. Diese wird mit dem App-Designer von Matlab R2022b realisiert.

1.1 Motivation

Airsoft ist ein Geländespiel, welches vom Konzept ähnlich wie Paintball ist. Der einzige Unterschied zu Paintball besteht darin, dass die Geschosse 6mm kleine Kunststoffkugeln (BB's) sind, anstelle von größeren, mit Farbe gefüllten Kugeln. Treffer sind so nicht offensichtlich, und Spieler geben selber zu erkennen, wenn sie getroffen worden sind. [9] Gespielt wird Airsoft mit Airsoftwaffen, welche sich vom Aussehen und Aufbau oftmals an echten Waffen orientiert. Eine Airsoftwaffe schießt die 6mm große, üblicherweise 0,12 bis 0,40 Gramm schwere Kunststoffkugel aus ihrem Lauf heraus. Die Kugel kann entweder durch Federkraft, durch komprimiertes Gas (CO_2 oder Butan/Propan Mischung) oder elektrisch beschleunigt werden. Die Mündungsenergie solcher Waffen befindet sich im Bereich zwischen 0,5 und 7,5 Joule. [1]

Während des Abschussvorgangs erhält die Kugel durch einen Hop-Up im Lauf eine Eigenrotation. Der Hop-Up ist eine einstellbare Verengung im oberen Segment des Laufs. Durch einen lokal höheren Reibkoeffizienten wird ein Drehmoment auf das Projektil eingeleitet. Die Kugel erhält einen Effekt, dessen Drehachse parallel zum Boden ist. Durch diese Rotation verhält sich die Kugel, wegen dem Einfluss des Magnus-Effekts, nicht wie bei einem senkrechten Wurf. Das Projektil verfolgt einen komplexen Pfad (Abbildung 1.1). Ziel der Simulation ist es dieses unintuitive Flugverhalten zu reproduzieren und ein besseres Verständnis für dieses Phänomen zu erhalten.

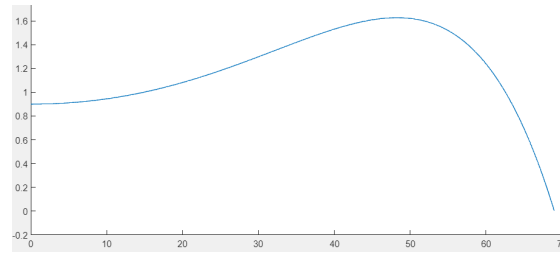


Abbildung 1.1: Flug einer Kugel unter Einfluss des Magnus-Effekts

1.2 Berücksichtigte Größen

Da es unmöglich ist, alle Einflüsse der Realität zu simulieren, wird sich in der Simulation auf die Größen und Effekte konzentriert, die einen ausgeprägten Einfluss auf die Simulation haben. Die ausgewählten Größen müssen mathematisch erfasst werden und eine nachvollziehbare Wechselwirkung aufeinander haben. Die im folgenden dargestellten Einflüsse sollen dazu dienen, ein möglichst genaues Abbild der Realität zu simulieren.

1.2.1 Simulationsgrößen des Projektils

- Projektilposition [m]
- Projektilgeschwindigkeit [$\frac{m}{s}$]
- Projektilmasse [kg]
- Projektildurchmesser [m]
- Drehung um eigene Achse [$\frac{rad}{s}$]

1.2.2 Simulationsgrößen der Airsoft Waffe

- Mündungsgeschwindigkeit [$\frac{m}{s}$]
- Abschusswinkel [deg]
- Abschusshöhe [m]

1.2.3 Simulationsgrößen der Umwelt

- Gravitation [$\frac{m}{s^2}$]
- Luftdichte [$\frac{kg}{m^3}$]
- Wind [$\frac{m}{s}$]

2 Aufbau App-Designer

2.1 Vorstellung der App ARTEMIS

Das Programm „Airsoft Range and Trajectory Evaluation using Matlab for Interactive Simulations “(ARTEMIS) soll eine einfache Bedingungsoberfläche für den Gebrauch der Simulationen ermöglichen. Ziel ist es die verschiedenen Parameter, der Airsoftwaffe, schnell und unkompliziert eingeben zu können, sowie die verschiedenen Störfaktoren zu berücksichtigen. Die Ergebnisse werden im zweidimensionalen und dreidimensionalen Raum grafisch dargestellt. Unter die grafische Darstellung fallen die kinetische Energie, die Flugdistanz als senkrechter Wurf, sowie die Flugbahn und Distanz der Kugel unter Berücksichtigung des Magnus-Effekts im zweidimensionalen und dreidimensionalen Raum.

2.1.1 Eingabe der Simulationsparameter (Register Parameter)

Die Eingabe der Parameter soll schnell und unkompliziert ermöglicht werden. Hierfür ist die App in verschiedene Register unterteilt. Im folgendem wird das Register für die Airsoft Parameter vorgestellt.

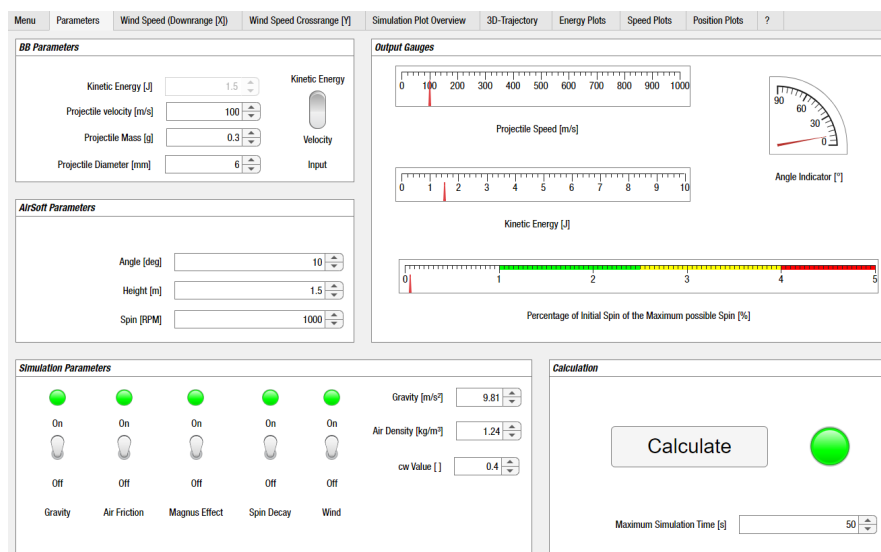


Abbildung 2.1: Bedienfläche Parameter

2.1.2 BB-Parameter

In diesem Bereich können die Parameter der Airsoft-BB-Kugel eingestellt werden. Die Berechnung kann entweder über die kinetische Energie [J] oder über die Projektilgeschwindigkeit [m/s] erfolgen. Des weiteren werden hier das Gewicht der Kugel [g] und der Durchmesser [mm] eingetragen. Es empfiehlt sich bei der Eingabe die Herstellerangaben zu verwenden, um ein möglichst genaues Ergebnis zu simulieren.

2.1.3 Airsoft Parameter

In diesem Feld werden der Abschusswinkel, die Abschusshöhe, sowie der Eigenrotation der Kugel bestimmt. Schusswinkel und Höhe sind auf den Anwender der der Airsoft bezogen. Bei der Rotation empfiehlt es sich Werte auszuprobieren, um die Charakteristik seiner Waffe zu bestimmen.

2.1.4 Simulationsparameter

In diesem Feld können verschiedene Effekte mit in die Simulation einbezogen werden. Dies dient dazu, verschiedene Einflussgrößen, gesondert zu betrachten (z.B. der Magnus-Effekt soll unter optimalen Bedingungen simuliert werden). Auslassen von Einflussgrößen führen zu nicht realitätsgetreuen Ergebnissen.

2.1.5 Output Gauges

In diesem Feld werden durch die vorherigen Parametereinstellungen erfasste Werte angezeigt. Die Projektilgeschwindigkeit, der Abschusswinkel und die kinetische Energie werden jeweils visualisiert. Allgemein dient dieses Feld zum überprüfen der eingegebenen Werte.

2.1.6 Calculate

Dieses Feld dient dazu die Simulation zu starten. Des weiteren kann hier das Zeitintervall bestimmt werden, sowie zwischen einer zweidimensionalen und dreidimensionalen Berechnung gewechselt werden.

2.1.7 Standardparameter

Das Programm ist beim Starten bereits mit sinnvollen Werten versehen. Insofern nicht anders erwähnt, werden diese für die folgenden Simulationen verwendet.

- Projektilgeschwindigkeit $100 \frac{m}{s}$
- Projektilmasse $0,3g$
- Projektildurchmesser $6mm$
- Abschusswinkel 10°
- Abschusshöhe $1,5m$
- Gravitation $9,81 \frac{m}{s^2}$
- Luftdichte $1,24 \frac{kg}{m^3}$
- Widerstandsbeiwert $c_w = 0,4$

2.2 Windgeschwindigkeit X,Y,Z-Achsen

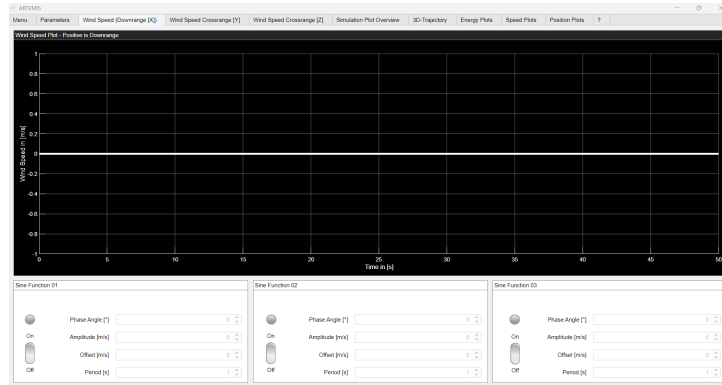


Abbildung 2.2: Bedienfläche Windeinstellungen

In den Registerkarten für Windgeschwindigkeit kann die Geschwindigkeit, die Richtung und das Verhalten des Windes eingestellt werden. Die Einstellungen für die einzelnen Achsen, erfolgen in separaten Registern. Das Verhalten, des Windes, kann über drei Sinusfunktionen, pro Achse, eingestellt werden. Die Funktionen werden über Phasenwinkel, Amplitude, Offset und Periode definiert.

2.3 Simulation Plot Overview

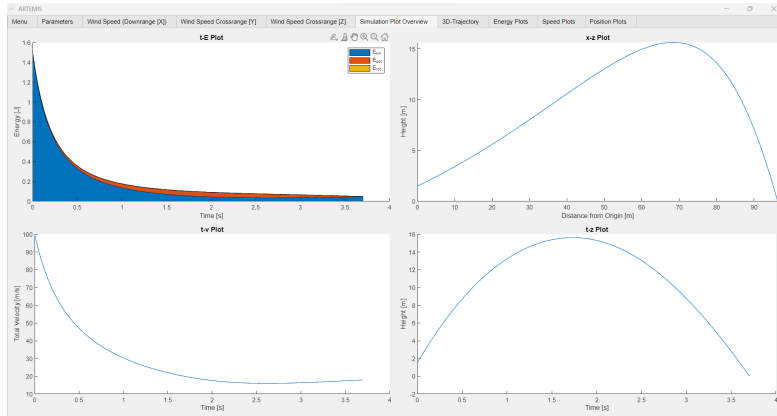


Abbildung 2.3: Simulation Plot Overview

Der Plot Overview stellt die Ergebnisse der Simulation grafisch da. In diesem Fenster werden die Flugbahn, der Energieabfall, die Geschwindigkeit, sowie die Höhe über die Zeit dargestellt. In den weiteren Registerkarten werden die verschiedenen Plots genauer dargestellt.

3 Erstellung der Simulation

Die Simulation des BB-Projektils erfolgt in mehreren Stufen und wird mit Matlab R2022b ausgeführt. Die erste Simulation erfolgt als schräger Wurf (Abschnitt 3.1). Dies dient dazu die grundlegendsten Gleichungen zu implementieren und die ersten Auswertungen zu erhalten. Darauf bezogen wird der Effekt des Luftwiderstandes (Abschnitt 3.2) eingeführt und mit dem schrägen Wurf verglichen. Zu erwarten ist, dass die Kugel durch den Luftwiderstand eine geringere Distanz zurücklegt als der schräge Wurf. Auf diesen Simulationen aufbauend werden dynamische Windeffekte in das Programm eingebaut (Abschnitt 3.3). Abhängig von der Windrichtung und dem Windverhalten, (stetig, böig ect.) können die dynamischen Windeffekte die Flugbahn der Kugel verschieden beeinflussen. Es ist davon auszugehen, dass die Effekte den Flug der Kugel entweder verlängern oder verkürzen, sowie später die Flugrichtung im dreidimensionalen Raum beeinflussen. Im weiteren wird der Magnus-Effekt (Abschnitt 3.4) in die Simulation einbezogen. Hier ist zu erwarten, dass sich die Flugbahn der Kugel nicht mehr, wie ein Schräger Wurf verhält, sondern einen Auftrieb erfährt (vgl. Abbildung 1.1). Zum Abschluss wird die Abnahme der Rotationsgeschwindigkeit, der Kugel, durch die Luftreibung betrachtet (Abschnitt 3.5). Zu erwarten ist, dass durch die Abnahme der Rotation, der Magnus-Effekt nachlässt und die Kugel eine kürzere Distanz zurücklegt. Ob die Abnahme des Magnus-Effektes, für das Ergebnis relevante Unterschiede erwirkt ist vor der Simulation noch eindeutig. Nach dem Aufstellen der Modellbildung für den zweidimensionalen Fall, wird das Programm auf die dritte Dimension ausgeweitet (4.6).

3.1 Schräger Wurf

Der schräge Wurf ist die sinnvolle Vereinfachung der Problemstellung. Hierbei werden alle Einflussgrößen bis auf die Gravitation vernachlässigt. Die relevanten, einstellbaren Parameter sind die Abschusshöhe, Abschusswinkel und Abschussgeschwindigkeit. Der schräge Wurf dient als Grundlage um den ODE Solver zu überprüfen und eventuelle konzeptionelle Fehler früh zu erkennen.

3.1.1 Herleitung der Modellgleichungen

Das Differenzialgleichungssystem für den schrägen Wurf sieht wie folgt aus [8, S. 186]:

$$\begin{array}{lll} \ddot{x} = 0; & x(0) = 0, & \dot{x}(0) = v_{x0} \\ \ddot{z} = -g; & z(0) = h, & \dot{z}(0) = v_{z0} \end{array}$$

Daraus ergibt sich das für den ODE Solver relevante Gleichungssystem [8, S. 186]:

$$\begin{aligned} dy(1) &= y_2, & y_1(0) &= 0 \\ dy(2) &= 0, & y_2(0) &= v_{x0} \\ dy(3) &= y_4, & y_3(0) &= 0 \\ dy(4) &= -g, & y_4(0) &= v_{z0} \end{aligned}$$

3.1.2 Darstellung Ergebnisse

Um Ergebnisse der numerischen Lösung zu verifizieren, kann für diesen Fall noch eine analytische Lösung herangezogen werden. Dies ist bei den folgenden Einflussgrößen nicht mehr möglich. Für die numerische und analytische Lösung werden identische Startbedingungen verwendet. Es wird eine Gravitation von $10 \frac{m}{s^2}$ angenommen. Der Abschusswinkel beträgt 45° , die Starthöhe beträgt 10m. Das Projektil besitzt eine Anfangsgeschwindigkeit von $10 \frac{m}{s}$ in x- und z-Richtung. Die Simulation ist beendet, wenn die Kugel auf dem Boden aufschlägt, der sich 10m unter der Starthöhe befindet. Die Erkennung für diesen Fall wird mit einer Eventfunktion realisiert. Der analytische Lösungsansatz für den schrägen Wurf beschreibt x- und z-Richtung mit unabhängigen Gleichungen:

$$\begin{aligned} x(t) &= v_x \cdot t \\ z(t) &= -\frac{1}{2} \cdot g \cdot t^2 + v_z \cdot t + h \end{aligned}$$

Nach dem Einsetzen der Anfangsbedingungen in die analytische und die numerische Lösung, ergeben sich folgende Daten:

Wert	Analytisch	Numerisch
Maximalhöhe	15m	14.99504...m
Flugweite	$10 \cdot (1 + \sqrt{3})$ m	27.32050807568 9062 m
Flugzeit	$1 + \sqrt{3}$ s	2.732050807568 906 s

Der Unterschied der beiden Verfahren befindet sich im tausendstel Bereich. Für die in der Simulation zu erwartenden Größen, sind diese Unsicherheiten akzeptabel. Der Vergleich zeigt, dass sich die größten Unsicherheiten in der Größenordnung eines Millimeters liegen. Da sich die typische Geschwindigkeit eines Airsoftprojektils bei $100 \frac{m}{s}$ befindet und die typische Schussweite bei 60m liegt, sind die zu erwartenden Unsicherheiten akzeptabel. Für die numerische Analyse wurde der ode45 Solver verwendet und wird für die weitere Modellbildung verwendet. Die Auswahl eines geeigneteren ODE Solvers wird nach Abschluss der Modellbildung überprüft.

3.2 Berücksichtigung des Luftwiderstands

3.2.1 Herleitung der Modellgleichungen, Annahmen und Vereinfachungen

Als erste Erweiterung zum Schwerfeld wird nun neben der Gravitation noch die Kraft des Luftwiderstandes berücksichtigt. Diese Kraft wirkt stets entgegen der Bewegungsrichtung der Kugel und wird durch folgende Formel definiert.

$$F_W = -c_w \cdot \frac{\rho}{2} \cdot A \cdot (|\vec{v}|)^2 \quad (3.1)$$

Für die Simulation in Matlab wird der Luftwiderstand für die einzelnen Bewegungsrichtungen separat betrachtet.

$$F_{W_x} = -c_w \cdot \frac{\rho}{2} \cdot A \cdot v_x^2$$
$$F_{W_z} = -c_w \cdot \frac{\rho}{2} \cdot A \cdot v_z^2$$

c_w ist ein dimensionsloser Widerstandsbeiwert, ρ die Luftdichte und A die angeströmte Querschnittsfläche der Kugel. Die Geschwindigkeit \vec{v} stellt die Differenz von Kugel- zu Windgeschwindigkeit dar. Da in diesem vereinfachten Fall, um den reinen Luftwiderstand bei Windstille ausgegangen wird, entspricht \vec{v} hier der Kugelgeschwindigkeit.[6, S. 26] Somit wird das für den ODE Solver relevante Gleichungssystem um den Anteil des Luftwiderstands erweitert.

$$dy(1) = y_2, \quad y_1(0) = 0 \quad (3.2)$$

$$dy(2) = -\frac{F_{W_x}}{m}, \quad y_2(0) = v_{x0} \quad (3.3)$$

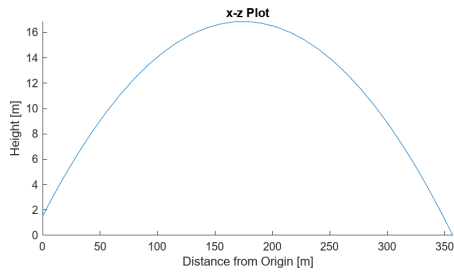
$$dy(3) = y_4, \quad y_3(0) = 0 \quad (3.4)$$

$$dy(4) = -g - \frac{F_{W_z}}{m}, \quad y_4(0) = v_{z0} \quad (3.5)$$

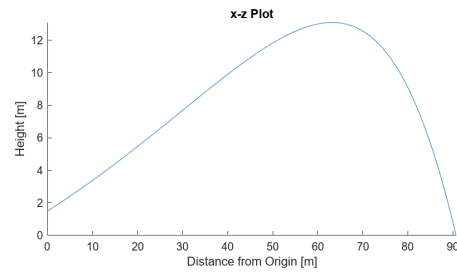
3.2.2 Darstellung Ergebnisse

Zu erwarten ist eine Abnahme in der Flugweite der Kugel gegenüber der Kurve nur mit Gravitationseinwirkung. Der Verlauf der Flugbahn sollte durch den Luftwiderstand von dem einer Parabel abweichen. Die Flugbahn ist nach Überschreiten des Hochpunktes steiler als im Anstieg.[7, 103ff]

Die Simulation wird mit dem Einbezug der Luftreibung durchgeführt und anschließend mit der Simulation ohne Luftreibung verglichen. Für beide Fälle werden die Standardparameter der App verwendet. In Abbildung 3.1 ist zu erkennen, dass der Verlauf mit Einbezug der Luftreibung von der Parabelform abweicht. Dies deckt sich mit den Annahmen. Der absteigene Flugbahnast bedeutend ist steiler, als der ansteigende Ast und deutet auf einen Abfall der horizontalen Geschwindigkeitskomponente hin.



(a) Flugbahn der Kugel im Schwerfeld



(b) Flugbahn der Kugel mit Luftreibung und Gravitation

Abbildung 3.1: Vergleich Flugbahn mit und ohne Luftreibung

Abbildung 3.2 zeigt den Verlauf der kinetischen sowie potentiellen Energie über die Zeit. Der deutliche Energieabfall, zu Beginn des Fluges, lässt sich anhand der Gleichung 3.1 erläutern. Der quadratische Anteil der Geschwindigkeit, welche der Kugelgeschwindigkeit entgegengerichtet ist, bewirkt einen logarithmischen Abfall im Energieverlaufdiagramm. Darüber hinaus geht aus der Abbildung hervor, dass der Anteil der kinetischen Energie deutlich größer ist als der Anteil der potentiellen Energie. Die hohe Startgeschwindigkeit wirkt sich auf die kinetische Energie aus. Anhand der Flugbahn und des Energieverlaufdiagramms lässt sich schlussfolgern, dass die Ergebnisse den physikalischen Prinzipien entsprechen.

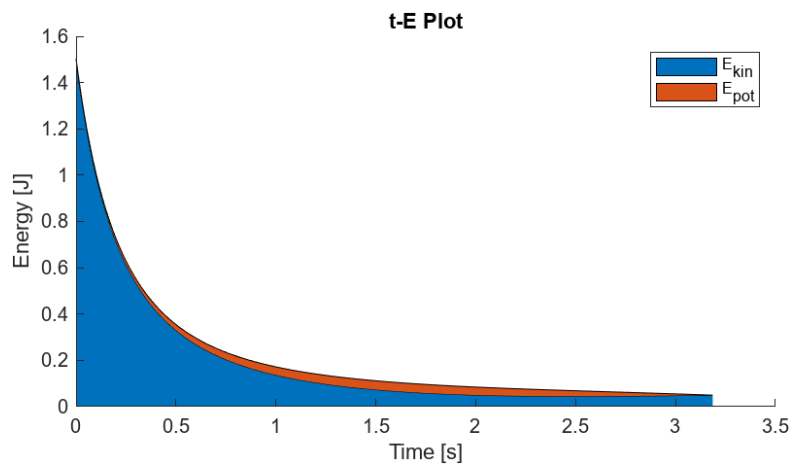


Abbildung 3.2: Energieverlauf über die Zeit aufgetragen unter Einfluss von Luftreibung und Gravitation

3.3 Berücksichtigung mit dynamischen Windeffekten

3.3.1 Herleitung der Modellgleichungen, Annahmen und Vereinfachungen

Wird die Windgeschwindigkeit und ihre Richtung in die Simulation einbezogen, muss die Relativgeschwindigkeit \vec{v}_{rel} verwendet werden. Zu erkennen in Gleichung 3.6 erfolgt die Differenz der Geschwindigkeiten komponentenweise. [6, S. 26]

$$\vec{v}_{rel} = \vec{v} - \vec{v}_{Wind} = \begin{bmatrix} v_x - v_{Wind_x} \\ v_y - v_{Wind_y} \\ v_z - v_{Wind_z} \end{bmatrix} = \begin{bmatrix} v_{rel_x} \\ v_{rel_y} \\ v_{rel_z} \end{bmatrix} \quad (3.6)$$

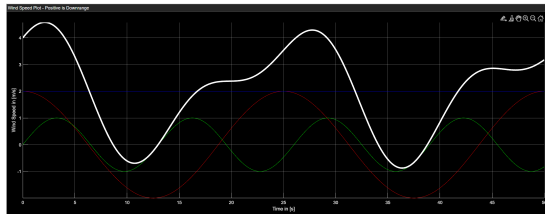
$$F_{W_x} = -c_w \cdot \frac{\rho}{2} \cdot A \cdot (v_{rel_x})^2$$

$$F_{W_z} = -c_w \cdot \frac{\rho}{2} \cdot A \cdot (v_{rel_z})^2$$

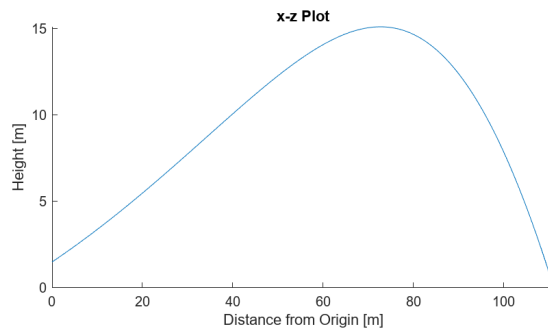
Da die Simulation im zweidimensionalen Raum erfolgt, verfällt die dritte Dimension in y-Richtung. Zusätzlich werden die Windgeschwindigkeiten von den Kugelgeschwindigkeiten subtrahiert, damit positive Windgeschwindigkeiten Rückenwind beziehungsweise Auftrieb bedeuten. Das Gleichungssystem bleibt unverändert und ist identisch mit den Gleichungen 3.2 bis 3.5.

3.3.2 Darstellung Ergebnisse

Je nach Gegenwind oder Rückenwind wird sich die Flugweite verringern oder erhöhen. Darüber hinaus ist bei konstanter Windgeschwindigkeit die Flugbahn vergleichbar mit dem Verlauf der Kurve bei Windstille. Der absteigende Flugbahnast ist weiterhin steiler als der ansteigende.



(a) Beispielhaftes Winddiagramm



(b) Flugbahn mit Gravitation, Luftreibung und Wind

Abbildung 3.3: Flugbahn mit beispielhaftem Wind

Die vielfältigen Einstellungs- und Überlagerungsmöglichkeiten von z.B. Aufwind und Rückenwind mit verschiedenen sinusbasierten Schwingungen lassen keine genaue Prognose auf die Flugbahn zu. Das Programm gibt dennoch nachvollziehbare Lösungen aus und die Ergebnisse passen zu den selbst gewählten Eingaben. In Abbildung 3.3 ist zu erkennen, dass unabhängig der Dynamik des Windes der Rückenwind im Allgemeinen zu einer erhöhten Flugweite führt.

3.4 Berücksichtigung des Magnus-Effekts

Der Magnus-Effekt ist ein Phänomen aus der Strömungsmechanik. Er beschreibt den Effekt eines runden Körpers (Kugel in dieser Simulation), der durch seine Eigenrotation, eine Querkraftwirkung erfährt. Diese entsteht dadurch, dass sich, durch die Rotation unter der Kugel ein Überdruck entsteht. Dieser sorgt dafür, dass die Kugel einen Auftrieb erfährt.

3.4.1 Herleitung der Modellgleichungen, Annahmen und Vereinfachungen

Der Magnus-Effekt lässt sich mathematisch mit der Gleichung 3.7 darstellen. Wobei c_m für den Magnus-Koeffizient, ρ für die Dichte der Luft und A für den Flächenquerschnitt steht. Durch $\omega(e_r \times v)$ wird die Wirkrichtung des Magnus-Effekts in Vektorform dargestellt. Hierbei ist e_r die Rotationsachse der Kugel, ω die Winkelgeschwindigkeit in $\frac{rad}{s}$ und v die zusammengesetzte Geschwindigkeit in $\frac{m}{s}$. Wie durch das Kreuzprodukt von $(e_r \times v)$ dargestellt wird, wirkt die Kraft des Magnus-Effekts orthogonal zur Geschwindigkeit der Kugel.[7, S. 18]

$$\vec{F}_M = \frac{(c_m \cdot \rho \cdot A)}{2} \cdot \omega(e_r \times \vec{v}) \quad (3.7)$$

Für die Simulation des Magnus-Effekts, bei einer Airsoft BB, wird, vereinfacht, die Gleichung 3.8 für eine glatte Kugel angenommen.

$$\vec{F}_M = \frac{4}{3}\pi\rho r^3 * (\vec{v} \times \omega) \quad (3.8)$$

Für die Simulation in Matlab wird zu Beginn die Kraft $F_M = \frac{4}{3}\pi\rho r^3 \cdot |\vec{v}_{rel}| \cdot \omega$ berechnet. Diese wird anschließend in ihre x - und z -Komponenten aufgeteilt.

$$F_{M_x} = |\vec{F}_M| \cdot \cos\left(\text{atan2}(v_{rel_x}; v_{rel_z}) + \frac{\pi}{2}\right)$$

$$F_{M_z} = |\vec{F}_M| \cdot \sin\left(\text{atan2}(v_{rel_x}; v_{rel_z}) + \frac{\pi}{2}\right)$$

v_x ist hierbei die relative Geschwindigkeit von der Kugel zum Boden, v_{Wind_x} ist die relative Geschwindigkeit zum Wind.

Anmerkung: die Matlabfunktion `atan2` arbeitet mit umgekehrter Reihenfolge der Parameter (`atan2(y; x)`) in dieser Dokumentation wird, standardmäßig mit der alphabetischen

Darstellung gearbeitet. Für die Bestimmung des Koordinatenpunkte werden abschließend die weiteren Umwelteinflüsse eingefügt.

$$\ddot{x} = \frac{F_{Wx}}{m} + \frac{F_{Mx}}{m}$$

$$\ddot{z} = \frac{F_{Wz}}{m} + \frac{F_{Mz}}{m} - g$$

Diese werden anschließend im ODE-Solver wie folgt dargestellt:

$$dy(1) = y_2, \quad y_1(0) = 0 \quad (3.9)$$

$$dy(2) = -\frac{F_{Wx}}{m} - \frac{F_{Mx}}{m}, \quad y_2(0) = v_{x0} \quad (3.10)$$

$$dy(3) = y_4, \quad y_3(0) = 0 \quad (3.11)$$

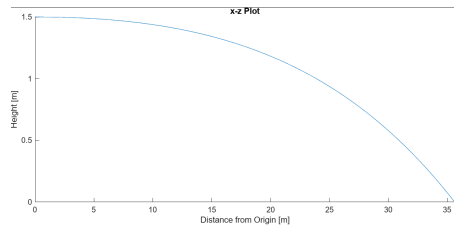
$$dy(4) = -\frac{F_{Wz}}{m} - \frac{F_{Mz}}{m} - g, \quad y_4(0) = v_{z0} \quad (3.12)$$

$$dy(5) = y_6, \quad y_5(0) = 0 \quad (3.13)$$

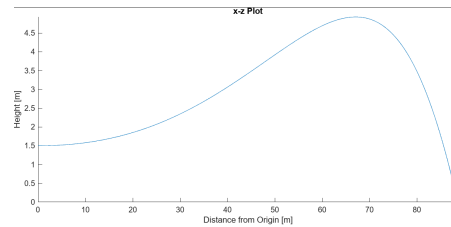
$$dy(6) = 0, \quad y_6(0) = \omega \quad (3.14)$$

Mit den Gleichungen 3.13 und 3.14 wird die Winkelgeschwindigkeit der Kugel berücksichtigt. Diese ist ausschlaggebend für den Einfluss des Magnus-Effekts auf die Flugbahn der Kugel.

3.4.2 Darstellung Ergebnisse



(a) Grader Schuss ohne Magnus-Effekt



(b) Grader Schuss mit Magnus-Effekt

Abbildung 3.4: Vergleich Flugbahn mit und ohne Magnus-Effekt

Aus der Simulation geht hervor, dass die Kugel, durch den Magnus-Effekt einen Auftrieb erfährt. Dieser Auftrieb sorgt dafür, dass die Flugbahn sich erhöht und eine größere Distanz zurückgelegt wird (vgl. Abb 3.4a und 3.4b). Für die Simulation aus Abbildung 3.4 wurden die Standardparameter verwendet und der die Rotation der Kugel auf $5000 \frac{\text{Umdrehungen}}{\text{min}}$ festgelegt. Der Abschusswinkel der Kugel beträgt 0 Grad. Die Ergebnisse der Simulationen verhalten sich wie den Erwartungen entsprechend.

3.5 Berücksichtigung der Abnahme der Rotationsgeschwindigkeit

Als letzter Einfluss wird die Auswirkung des Luftwiderstand auf die Rotation der Kugel betrachtet. Hierbei wird angenommen, dass die Rotation der Kugel durch den Luftwiderstand abnimmt. Dies soll dazu führen, dass der Magnus-Effekt einen schwächeren Einfluss auf die Flugbahn der Kugel hat.

3.5.1 Herleitung der Modellgleichungen, Annahmen und Vereinfachungen

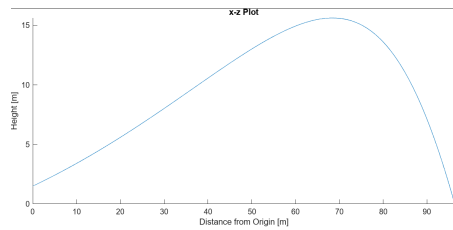
Der Einfluss der Abnahme der Rotationsgeschwindigkeit wird mit folgenden Formeln bestimmt. [2]

$$\begin{aligned}\tau_{visc} &= -8\pi r^3 \eta \dot{\phi} \\ \alpha &= \ddot{\phi} = \frac{\tau_{visc}}{I} \\ I_{Kugel} &= \frac{2}{5} \cdot m \cdot r^2\end{aligned}$$

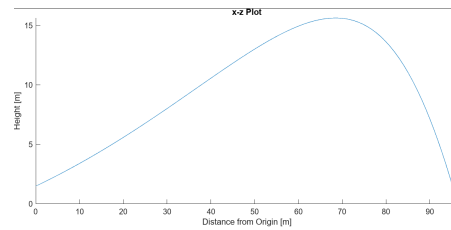
Hierbei ist τ_{visc} das viskose Drehmoment. Diese setzt sich aus der dynamischen Viskosität des Trägerelements η und der Winkelgeschwindigkeit $\dot{\phi} = \frac{d\phi}{dt}$ in Abhängigkeit zum Kugelvolumen zusammen. Die Winkelbeschleunigung $\alpha = \ddot{\phi}$ setzt sich aus der dynamischen Viskosität und dem Widerstandsmoment I_{Kugel} zusammen [3]. Diese wirkt entgegen der Rotation, die für den Magnus-Effekt verantwortlich ist (Gl. 3.8). Das resultierende Gleichungssystem, welches den Abfall der Drehgeschwindigkeit darstellt, ist:

$$\begin{aligned}dy(1) &= y_2, & y_1(0) &= 0 \\ dy(2) &= -\frac{F_{Wx}}{m} - \frac{F_{Mx}}{m}, & y_2(0) &= v_{x0} \\ dy(3) &= y_4, & y_3(0) &= 0 \\ dy(4) &= -\frac{F_{Wx}}{m} - \frac{F_{Mx}}{m} - g, & y_4(0) &= v_{z0} \\ dy(5) &= y_6, & y_5(0) &= 0 \\ dy(6) &= \frac{\tau_{visc}}{I_{Kugel}}, & y_6(0) &= \omega\end{aligned}$$

3.5.2 Darstellung Ergebnisse



(a) Magnus-Effekt mit Rotationsabnahme



(b) Magnus-Effekt ohne Rotationsabnahme

Abbildung 3.5: Vergleich Flugbahn mit und ohne Luftreibung an der Rotation der Kugel

Werden die Abbildung 3.5a und 3.5b verglichen, so wird deutlich, dass die Luftreibung, auf die Eigenrotation und somit auf den Magnus-Effekt eine vernachlässigbare Auswirkung hat. Die geringe Auswirkung lässt sich durch die geringe Abnahme der Rotationsenergie über den Zeitraum des Kugelfluges erklären (vgl. 3.6). Die Abnahme der Energie liegt bei $20 \cdot 10^{-9}$ Joule.

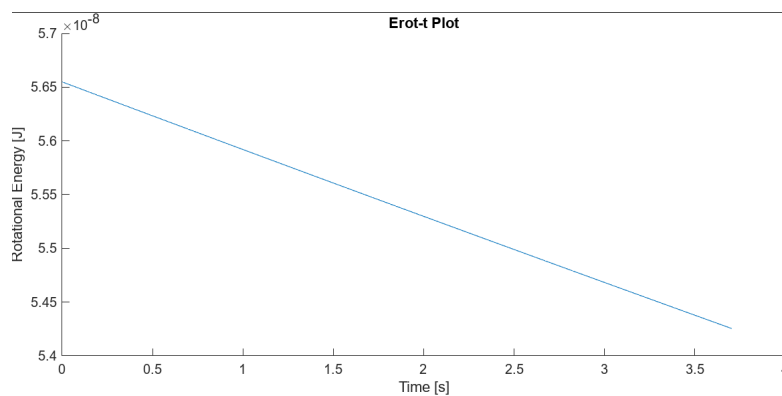


Abbildung 3.6: Abnahme der Rotationsenergie

3.6 Implementierung im dreidimensionalen Raum

Zur Vervollständigung von ARTEMIS wird die Simulation um die y-Achse erweitert. Dabei werden sämtliche zuvor berücksichtigten Faktoren sowie deren Auswirkungen in den nun dreidimensionalen Raum einbezogen. Eine dreidimensionale Simulation ermöglicht es primär Seitenwinde in die Simulation einzubeziehen.

3.6.1 Herleitung der Modellgleichungen, Annahmen und Vereinfachungen

Das bestehende Differenzialgleichungssystem kann mit kleinen Änderungen weiterverwendet werden. Die Bewegung in y-Richtung wird mit zwei zusätzlichen Termen berücksichtigt. Diese neuen Terme orientieren sich an den Gleichungen für die Bewegung in x-Richtung.

Die Implementierung des Magnus-Effekts kann nicht in das dreidimensionale System übernommen werden. Im zweidimensionalen Raum konnte der Magnus-Effekt zu jedem Zeitpunkt eindeutig bestimmt werden. Durch die Vereinfachung auf zwei Dimensionen konnte angenommen werden, dass die Drehachse zu jedem Zeitpunkt rechtwinklig auf der x- und z-Achse lag. Der Vektor der Magnuskraft ist orthogonal zur Bewegungsrichtung und Drehachse. Da die Bewegungsrichtung durch die Geschwindigkeit bestimmt werden konnte, war die Ermittlung der Richtung und Magnitude der Magnuskraft in x- und z-Richtung möglich.

Diese Möglichkeit entfällt in der dritten Dimension, da die Drehachse überall im Raum liegen könnte. Zur Vereinfachung des Problems wird angenommen, dass die Waffe stets gerade abgefeuert wird. Dies bedeutet, dass die Drehachse immer parallel zur x-y-Ebene ist. Um die Orientierung der Drehachse zu bestimmen, werden zuerst die x- und y-Komponenten der relativen Luftgeschwindigkeit ausgewertet. Aus dem resultierenden Winkel kann ein Einheitsvektor konstruiert werden, der orthogonal auf der Bewegungsrichtung liegt.

$$Azimuth = atan2(v_{rel_x}; v_{rel_z})$$
$$\vec{e}_r = \vec{A} = \begin{bmatrix} \sin(Azimuth + \frac{\pi}{2}) \\ \cos(Azimuth + \frac{\pi}{2}) \\ 0 \end{bmatrix}$$

Zusätzlich benötigt wird ein Einheitsvektor in Richtung der relativen Luftgeschwindigkeit:

$$\vec{B} = \begin{bmatrix} v_{rel_x} \\ v_{rel_y} \\ v_{rel_z} \end{bmatrix} \cdot \frac{1}{|\vec{v}_{rel}|}$$

Da der Kraftvektor der Magnuskraft orthogonal auf Achse und Fluggeschwindigkeit liegt, kann die Magnuskraft in der Simulation mit dem negativen Kreuzprodukt von A und B bestimmt werden.

$$\vec{F}_M = \begin{bmatrix} F_{M_x} \\ F_{M_y} \\ F_{M_z} \end{bmatrix} = -(\vec{A} \times \vec{B}) \cdot \left(\frac{4}{3} \pi \rho r^3 \cdot |\vec{v}_{rel}| \cdot \omega \right)$$

Mit der Berücksichtigung des Magnus-Effekts im dreidimensionalen Raum kann das Gleichungssystem aufgestellt werden.

$$\begin{aligned} dy(1) &= y_2, & y_1(0) &= 0 \\ dy(2) &= -\frac{F_{W_x}}{m} - \frac{F_{M_x}}{m}, & y_2(0) &= v_{x0} \\ dy(3) &= y_4, & y_3(0) &= 0 \\ dy(4) &= -\frac{F_{W_y}}{m} - \frac{F_{M_y}}{m}, & y_4(0) &= v_{y0} \\ dy(5) &= y_6, & y_5(0) &= 0 \\ dy(6) &= -\frac{F_{W_x}}{m} - \frac{F_{M_x}}{m} - g, & y_6(0) &= v_{z0} \\ dy(7) &= y_8, & y_7(0) &= 0 \\ dy(8) &= \frac{\tau_{visc}}{I_{Kugel}}, & y_8(0) &= \omega \end{aligned}$$

Ein Nachteil dieser Implementierung des Magnus-Effekts ist es, dass die Waffe nicht schräg simuliert werden kann. Die Drehachse liegt immer parallel zur x-y-Ebene. Flugbahnen um Hindernisse herum können somit nicht simuliert werden.

3.6.2 Darstellung Ergebnisse

Zu erwarten ist eine Parität zu den Ergebnissen im zweidimensionalen Raum. Diese Erwartungshaltung trifft zu, sowohl für den schiefen Wurf, als auch mit allen Effekten kombiniert. Der Luftwiderstand, Wind sowie der Magnuseffekt sind identisch zum dreidimensionalen Raum wie im zweidimensionalen.

3.6.3 Gimbal Lock

Die Implementierung des Magnus-Effekts auf diese Weise hat einen weiteren Nachteil. Wenn das Projektil einen Steilflug von 90° nach oben absolviert, die x- und y-Komponente des Geschwindigkeitsvektors somit exakt 0 ist, verliert das System einen Freiheitsgrad (Gimbal Lock). [5] Die Rotationsachse des Projektils kann sich so frei drehen, ohne eine Auswirkung auf die Simulation zu haben. Kleinste Änderungen in der Y-Komponente können so eine große Änderung für den Winkel ausmachen.

$$\text{atan2}(0; 1e-6) = 0 \tag{3.15}$$

$$\text{atan2}(0; -1e-6) = \pi \tag{3.16}$$

Der Vorzeichenwechsel des zweiten Parameters der *atan2* Funktion lässt den resultierenden Winkel um 180° drehen (Formel 3.15 und 3.16). Der ODE Solver reagiert auf diese Instabilität mit immer kleineren Schrittgrößen, die das Programm durch lange Rechenzeiten zum Erliegen bringen. Eine Lösung des Problems ist durch das Verfolgen der Orientierung des Projektils innerhalb des Gleichungssystems diese Unsicherheit zu beseitigen.

Jedoch tritt dieser Fall in der Realität nur bei hohen Abschusswinkeln auf. In der Praxis und im Airsoft-Spiel sind vor allem waagerechte Flugbahnen für die Simulation als relevant anzusehen. Für das Programm wurden daher beide Lösungsverfahren implementiert. Der Nutzer kann selbstständig zwischen 2D und 3D Simulation umschalten. Bei der 2D Simulation werden, die für die y-Achse relevanten Eingaben nicht betrachtet. Durch Umschalten ist es ebenfalls möglich die 2D Ergebnisse mit den 3D Ergebnissen zu vergleichen.

4 Auswertung Simulation

4.1 Wahl des ODE-Solver

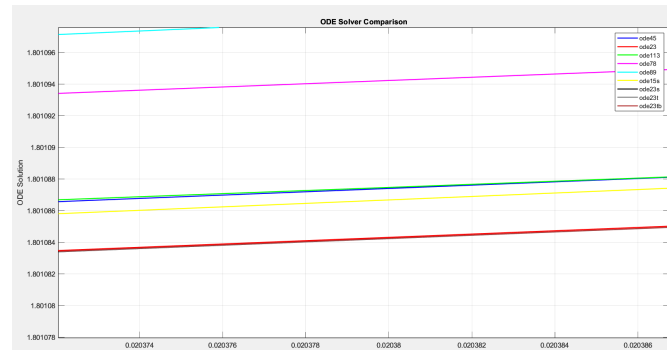
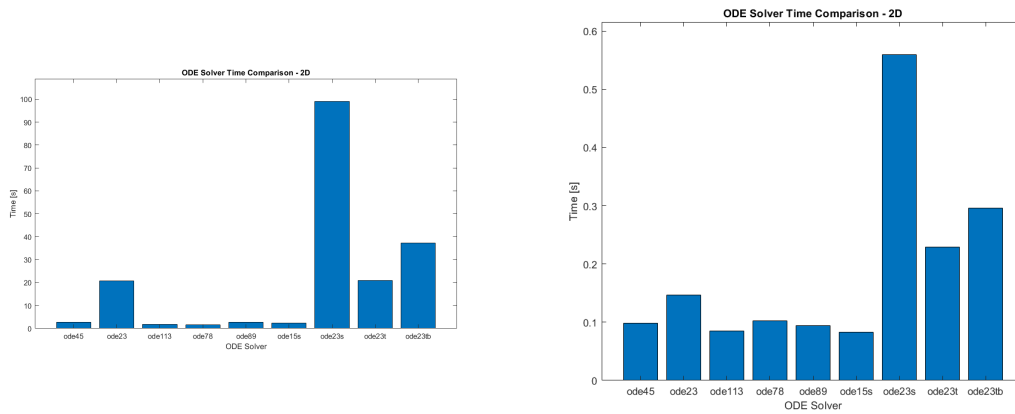


Abbildung 4.1: Vergleich der ODE-Solver in 2D

Wie in der Abbildung 4.1 dargestellt, unterscheiden sich die Kurven der Flugbahn im Bereich von $3 \cdot 10^{-6} \text{m}$. Diese Abweichung ist bei einer Fluglänge von ca. 60m für die Modellbildung vernachlässigbar. Der geeignetste Solver für die Simulation wäre der, der die geringste Rechenzeit benötigt um zu einem Ergebnis zu kommen. Dazu kommen zunächst alle Solver in Frage, bis auf ode15i, der voll implizit ist. Für die Laufzeitanalyse wurden Parameter gewählt, um eine unrealistische, aber sehr komplexe Flugbahn zu erzeugen, sowie ein Parametersatz, der ein gängiges Szenario darstellt. Die Rechenzeiten wurden mit einem i7-10510U CPU durchgeführt. Während dem Berechnungszeitlauf waren die Kerntemperaturen des CPU unterhalb der maximalen Temperatur, dass keine Messfehler durch eine automatische Leistungseinschränkung stattfanden.

Im ersten Durchlauf wurde eine Anfangsgeschwindigkeit von $300 \frac{\text{m}}{\text{s}}$ gewählt, sowie eine Rotationsgeschwindigkeit von $3 \cdot 10^5 \frac{\text{Umdrehungen}}{\text{min}}$. Dies resultiert in einer anfangs spiralförmigen Bewegung, die dann in eine sinusähnliche Bewegung in positiver X-Richtung wandelt. Im zweiten Durchlauf wurde eine realistischere Flugbahn gewählt. Die Mündungsgeschwindigkeit beträgt nun $100 \frac{\text{m}}{\text{s}}$, mit einem Spin von $5000 \frac{\text{Umdrehungen}}{\text{min}}$.



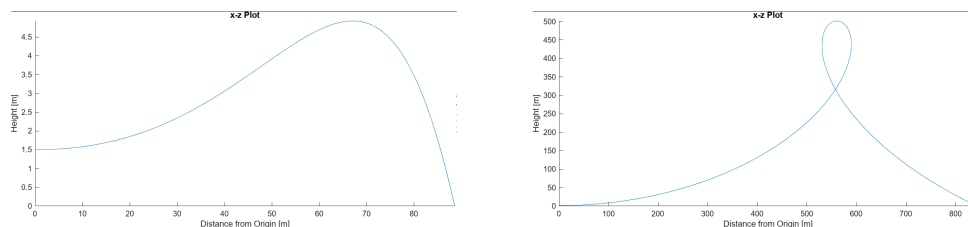
- (a) Vergleich der Laufzeiten der ODE-Solver in 2D, komplexe Flugbahn (b) Vergleich der Laufzeiten der ODE-Solver in 2D, realistische Flugbahn

Abbildung 4.2: Vergleich der Laufzeiten der ODE-Solver in 2D

Ersichtlich ist, dass die ode23 Solver deutlich mehr Zeit benötigen, um ein ähnliches Ergebnis zu erlangen. Die ode78 und ode113 Solver sind für die komplexe Flugbahn am schnellsten und benötigen knapp 1,5 Sekunden, beziehungsweise 1,6 Sekunden für die Berechnung. Für den realistischen Fall liegen alle ode Solver bis auf die ode23 Solver unter 0,1 Sekunden Rechenzeit.

Da komplexere Flugpfade im Programm mit dynamischen Windeffekten generiert werden können fällt die Auswahl auf den ode78 Solver, der eine höhere Genauigkeit, als der ode45 Solver besitzt. [4]

4.2 Magnus-Effekt ohne Luftreibung



(a) Flugbahn der Kugel mit Luftreibung

(b) Flugbahn der Kugel ohne Luftreibung

Abbildung 4.3: Vergleich Flugbahn mit und ohne Luftreibung

Für die Simulation wurden die Standard-Parameter verwendet und der Spin wurde auf 5000 $\frac{\text{Umdrehungen}}{\text{min}}$ eingestellt. Vergleicht man Abbildung 4.3a mit 4.3b, wird deutlich dass der Magnus-Effekt einen stärkeren Einfluss auf Flugkurve hat, wenn die Luftreibung vernachlässigt wird. Dies ist physikalisch nicht möglich, da der Magnus-Effekt durch die

Luftreibung zustande kommt, indem durch die Rotation der Kugel und der Luftreibung ein Überdruck entsteht, durch den das Projektil einen Auftrieb erfährt. Mathematisch wird die Kraft des Magnus-Effektes jedoch in Abhängigkeit zur Winkelgeschwindigkeit berechnet und die Luftreibung wird nicht in die Formeln einbezogen (vgl. Abschnitt 3.4.1). Durch den Vergleich wird deutlich, dass die Simulation des Magnus-Effekts, erst durch das Einbeziehen der Luftreibung, Ergebnisse liefert, die eine schlüssige Interpretation ermöglichen.

4.3 Parameteranpassung

Bei einigen Parametern besteht die Möglichkeit durch die Eingabe unrealistischer Werte das Programm an die Grenzen der Gültigkeit der physikalischen Gesetze zu bringen. Daher sind die Eingabewerte teilweise beschränkt worden. So zum Beispiel kann bei der Luftdichte keine Dichte für eine Flüssigkeit oder einen Feststoff eingegeben werden. Der Abschusswinkel ist auf -80° bis 80° limitiert, wodurch ein Schuss nach hinten unmöglich wird. Der Durchmesser sowie die Masse der Kugel wurden ebenfalls begrenzt, da ein Abschuss mit einer besonders großen und schweren Kugel in diesem Rahmen nicht vergleichbar wäre.

5 Zusammenfassung und Ausblick

Zusammenfassend lässt sich sagen, dass das Programm unter Eingabe sinnvoller Werte zu schlüssigen und nachvollziehbaren Ergebnissen kommt. Einfache Problemstellungen wie der schiefe Wurf im homogenen Schwerfeld sowohl mit als auch ohne Luftreibung sind vom Programm problemlos zu bewältigen.

Da die Berechnung des Magnuseffektes, wie bereits erwähnt, in Grenzfällen fehleranfällig wird, bedarf es hier einer Überarbeitung. Mögliche Lösungsansätze wären eine Berücksichtigung der Rotation der Achse in dem Differentialgleichungssystem, oder ein komplett anderer Ansatz mit Quaternionen.

Für einen besseren Vergleich der unterschiedlichen physikalischen Wirkungsweisen untereinander, bietet es sich an die Plots mehrerer Durchführungen in einem Graphen darzustellen. So könnte zum Beispiel der schiefe Wurf mit und ohne Luftreibung hinsichtlich seiner Wurfweite besser verglichen werden. Eine sinnvolle Erweiterung des Programms könnte der Import von realen Winddaten einer Wetterstation sein, um reale Parameter einzufügen und die Simulation näher an die Realität anzupassen. Durch das Schräghalten einer Airsoft-Pistole ändert sich die Wirkrichtung des Magnuseffektes. Somit kann durch die Hop-Up Funktion einen seitlichen Drall erzeugt werden. Dies bietet eine passende Grundlage, um horizontal gekrümmte Flugbahnen zu simulieren. Mit der Erweiterung des Programms um Hindernisse wie Wände und einer seitlich gekrümmten Flugbahn lässt sich ein Schuss um eine Ecke nachstellen. Ebenfalls kann das Programm ein Höhenprofil eines Geländes verwenden. So kann man eine Gebietskarte eines Airsoft-Geländes in das Programm einfügen, um taktische Möglichkeiten für aktive Spieler zu ermöglichen.

Literatur

- [1] Bussgeldkatalog. *Softair: Spielzeug oder echte Waffe?* URL: <https://www.bussgeldkatalog.net/softair/>.
- [2] Santiago Helbig u. a. „Self-consistent solution of magnetic and friction energy losses of a magnetic nanoparticle“. In: *Phys. Rev. B* 107 (5 Feb. 2023), S. 054416. DOI: 10.1103/PhysRevB.107.054416. URL: <https://link.aps.org/doi/10.1103/PhysRevB.107.054416>.
- [3] Russell C. Hibbler. *Technische Mechanik 3*. Pearson, 2021. ISBN: 978-3-86894-408-2.
- [4] MathWorks. *Choose an ODE solver*. URL: <https://de.mathworks.com/help/matlab/math/choose-an-ode-solver.html>.
- [5] Adrian E. Popa. *Re: What is meant by the term gimbal lock?* URL: <https://www.madsci.org/posts/archives/aug98/896993617.Eg.r.html>.
- [6] Georg Rill, Thomas Schaeffer und Fredrik Borchsenius. *Grundlagen und computergerechte Methodik der Mehrkörpersimulation*. Springer Vieweg Wiesbaden, 2020. ISBN: 978-3-658-28912-6.
- [7] Hans-Stefan Siller, Gilbert Greefrath und Werner Blum. *Neue Materialien für einen realitätsbezogenen Mathematikunterricht 4*. Springer Spektrum Wiesbaden, 2018. ISBN: 978-3-658-17599-3.
- [8] Gerd Steinebach. „Vorlesungsskript ModSim1“. In: ().
- [9] Hanno Terbuyken. *Airsoft eine Randsportart weitab vom Schuss*. URL: <https://www.evangelisch.de/inhalte/96547/22-10-2009/airsoft-eine-randsportart-weitab-vom-schuss>.