

Flugbahnberechnung für Fleckschuss und Flugobjektabschuss

Prof. Dr. Andreas Rudolph
Universität der Bundeswehr
München
Institut für Mathematik und Informatik
FB BW
Werner-Heisenberg-Weg 39
85577 Neubiberg
Email: Andreas.Rudolph@unibw.de

8. September 2015

Inhaltsverzeichnis

1	Überblick	1
1.1	Einführung	1
1.2	Mathematische Grundlagen	2
2	Der Algorithmus	13
2.1	Die Grundvoraussetzungen	13
2.2	Die beiden Algorithmen für Fleckschussberechnung	14
2.3	Ein Algorithmus für Flugobjektabschuss	17
2.4	Der Algorithmus für Abschuss aus der Bewegung heraus	20
3	Etwas Numerik	23
3.1	Einführung	23
3.2	Differenzgleichungen	24
3.3	Diskretisierungen	25
3.4	Differenzgleichung und Differentialgleichung - Einschrittverfahren	29
3.5	Konsistenzordnung und globaler Fehler - Einschrittverfahren	33
3.6	Mehrschrittverfahren	36
3.7	Steife Differentialgleichungen	48
3.8	Zusammenfassung	54
4	Einige Ergebnisse	55
4.1	Einführende Bemerkungen	55
4.2	Das Kaliber .223	56
4.3	Das Kaliber .308	57
4.4	Das Kaliber .338 Lapua Magnum	59
4.5	Erweiterungen	60
	Index	63

Kapitel 1

Überblick

1.1 Einführung

Bei allen Modellen zur Beschreibung der Flugbahn eines Geschosses liegt die Newton'sche Mechanik zugrunde, d. h. die Gesamtheit **aller** auf das Geschoss einwirkenden Kräfte \mathbf{F} ist gleich deren Masse m multipliziert mit seiner Beschleunigung \mathbf{a} :

$$\mathbf{F} = m \cdot \mathbf{a} = m \frac{d^2}{dt^2} \mathbf{x}$$

wenn $\mathbf{x} = \mathbf{x}(t)$ die Ortskoordinate des Geschosses zum Zeitpunkt t darstellt. Hierbei unterstellt man, dass das Projektil als Massenpunkt dargestellt werden kann - man beschreibt strenggenommen die Bahn seines Schwerpunktes¹.

In erster Näherung vernachlässigt man zuerst die Ausdehnung des Geschosses - Rotationseffekte wie die Präzession, die Nutation, den Magnus-Effekt und den Spin-Dämpfungseffekt betrachtet man dann in einem zweiten Schritt.

Formuliert man nun die Kräfte, die auf das Geschoss einwirken, und löst das obige Differentialgleichungssystem zweiter Ordnung

$$\mathbf{F} = m \cdot \mathbf{a} = m \frac{d^2}{dt^2} \mathbf{x}$$

dann erhält man damit die Flugbahn des Geschosses in parametrisierter Form - damit kann man in Abhängigkeit von der Zeit alle Größen bestimmen, die für das sportliche, aber auch für das militärische Schießen von Interesse sind.

Übrigens ist eine sicherlich lesenswerte Einführung in die Thematik das Buch von de Mestre (1990).

¹Anscheinend wird der Lagrange-Formalismus und die Hamilton-Jacobi Theorie bisher nicht verwendet.

1.2 Mathematische Grundlagen

Betrachtet man das vorliegende Differentialgleichungssystem zweiter Ordnung

$$\mathbf{F} = m \cdot \mathbf{a} = m \frac{d^2}{dt^2} \mathbf{x}$$

dann stellt sich die Frage, ob und unter welchen Bedingungen ein derartiges System lösbar ist, und wenn es lösbar ist, wann die Lösung eindeutig ist.

Dazu schreibt man das System um in ein System erster Ordnung - man erhält dadurch prinzipiell ein System der Gestalt

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$

dem noch entsprechende Anfangsbedingungen $\mathbf{y}(t_0) = \mathbf{y}_0$ hinzugefügt werden.

Hierbei ist nun

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{pmatrix} \text{ und } \mathbf{f}(t, \mathbf{y}) = \begin{pmatrix} f_1(t, y_1, \dots, y_n) \\ \vdots \\ f_n(t, y_1, \dots, y_n) \end{pmatrix}$$

Gesucht ist damit ein Vektor $\mathbf{y}(t)$ von Funktionen $y_1(t), \dots, y_n(t)$, der das obige Differentialgleichungssystem erster Ordnung samt Anfangsbedingungen erfüllt.

Beispiel 1

Wir betrachten das sogenannte Vakuum-Modell, d. h. der Luftwiderstand wird vernachlässigt.

Damit haben wir als Differentialgleichungssystem

$$\begin{aligned} m\ddot{x} &= 0 \\ m\ddot{y} &= -mg \end{aligned}$$

wenn nur die Flugweite $x = x(t)$ und die Flughöhe $y = y(t)$ betrachtet werden sollen. Der Punkt soll dabei die Ableitung nach der Zeit bedeuten.

Zusätzlich stellt man Anfangsbedingungen

$$\begin{aligned} x(0) &= x_0 \\ y(0) &= y_0 \\ v_x(0) &= v_0 \cos(\theta) \\ v_y(0) &= v_0 \sin(\theta) \end{aligned}$$

wenn v_0 die Anfangsgeschwindigkeit des Geschosses ist und g die Erdbeschleunigung.

Dieses Differentialgleichungssystem zweiter Ordnung überführen wir in

$$\begin{aligned}\ddot{x} &= 0 \\ \ddot{y} &= -g\end{aligned}$$

Wir definieren die Vektoren

$$\mathbf{y} = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}, \quad \mathbf{f}(t, \mathbf{y}) = \begin{pmatrix} \dot{x} \\ \dot{y} \\ 0 \\ -g \end{pmatrix}$$

und erhalten damit

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ 0 \\ -g \end{pmatrix} \quad \text{oder} \quad \dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$$

ein Differentialgleichungssystem erster Ordnung.

Damit stellt sich die Frage nach der Existenz und Eindeutigkeit einer Lösung eines derartigen Differentialgleichungssystems erster Ordnung.

Ein Satz, der die Existenz und Eindeutigkeit von Lösungen von Differentialgleichungssystemen erster Ordnung generell sicherstellt, ist der bekannte Satz von Picard und Lindelöf - dieser besagt:

Satz 1

Wir betrachten das Differentialgleichungssystem

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(u) = \mathbf{v}$$

Die Abbildung \mathbf{f} sei in dem $(n+1)$ -dimensionalen Zylinder

$$|x - u| < a, \quad \|\mathbf{y} - \mathbf{v}\| < b$$

stetig und beschränkt, sie soll weiter in diesem eine Lipschitz-Bedingung erfüllen, d. h. es gelte

$$\|\mathbf{f}(x, \mathbf{y})\| \leq A, \quad \|\mathbf{f}(x, \mathbf{y}_1) - \mathbf{f}(x, \mathbf{y}_2)\| \leq M \|\mathbf{y}_1 - \mathbf{y}_2\|$$

Dann gibt es für das Differentialgleichungssystem

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$$

genau eine Lösung $\mathbf{y} = \mathbf{y}(x)$, welche die Anfangsbedingung $\mathbf{y}(u) = \mathbf{v}$ erfüllt. und für $|x - u| < \alpha$ existiert, wobei

$$\alpha = \min\left\{a, \frac{b}{A}\right\}$$

ist.

BEWEIS

Siehe zum Beispiel Kamke (1969), S. 74, Satz 1 und S. 104, Satz 3.

Beispiel 2

Betrachten wir nochmals das obige Beispiel des Vakuum-Modells, dann besagt der Satz von Picard-Lindelöf, dass genau eine Lösung existiert.

Dies bedeutet, dass ein Vektor $\mathbf{y}(t)$ existiert mit

$$\mathbf{y}(t) = \begin{pmatrix} x(t) \\ y(t) \\ v_x(t) \\ v_y(t) \end{pmatrix}$$

der die Anfangsbedingungen $x(0) = x_0$, $y(0) = y_0$, $v_x(0) = v_0 \cos(\theta)$ und $v_y(0) = v_0 \sin(\theta)$ erfüllt, wenn v_0 die Anfangsgeschwindigkeit darstellt.

Diese Lösung kann man durch entsprechende Integrationen oder durch einen Potenzreihenansatz erhalten - es ergibt sich:

$$\begin{aligned} x(t) &= v_0 \cos(\theta)t \\ y(t) &= y_0 + v_0 \sin(\theta)t - \frac{1}{2}gt^2 \end{aligned}$$

und durch Differentiation

$$\begin{aligned} v_x(t) &= v_0 \cos(\theta) \\ v_y(t) &= v_0 \sin(\theta) - gt \end{aligned}$$

Damit hängt die Lösung von einem Parameter ab - dem Erhebungswinkel θ .

Fordert man nun, dass in einer vorgegebenen Entfernung $x(t_{\text{Ziel}})$ gelten soll

$$y(t_{\text{Ziel}}) = 0$$

dann stellt sich damit die Frage, wie der Winkel θ gewählt werden muss, um dies zu erreichen.

Insbesondere stellt sich die Frage, ob in jedem Fall ein entsprechender Winkel existiert.

Generell gilt hierzu folgender

Satz 2

Sei $\mathcal{G}(x, \mathbf{y})$ ein Gebiet, $\mathbf{c} = (c_1, \dots, c_k)$ ein k -dimensionaler Vektor, \mathcal{C} sei eine offene Menge in dem \mathbf{c} -Raum und $\mathcal{G} \times \mathcal{C}$ das Kartesische Produkt von \mathcal{G} und \mathcal{C} - d. h. der Bereich der Punkte $x, \mathbf{y}, \mathbf{c}$ eines $(n + k + 1)$ -dimensionalen Raumes mit $x, \mathbf{y} \in \mathcal{G}$ und $\mathbf{c} \in \mathcal{C}$.

Die Abbildung $\mathbf{f}(x, \mathbf{y}, \mathbf{c})$ sei in $\mathcal{G} \times \mathcal{C}$ stetig, so dass für jedes $\mathbf{c} \in \mathcal{C}$ durch jeden Punkt $x, \mathbf{y} \in \mathcal{G}$ mindestens eine Integralkurve der Differentialgleichung

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}, \mathbf{c})$$

geht. Es wird weiter vorausgesetzt, dass es auch nur **eine** solche von Rand zu Rand laufende Integralkurve gibt, die mit

$$\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c})$$

bezeichnet wird.

Dann gilt folgendes:

- (a) Wenn die Funktion $\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c})$ an einer Stelle $x_0, u_0, \mathbf{v}_0, \mathbf{c}_0$ existiert und wenn $\alpha \leq x \leq \beta$ ein abgeschlossenes Intervall ist, das x_0, u_0 als innere Punkte enthält und in dem $\mathbf{Y}(x, u_0, \mathbf{v}_0, \mathbf{c}_0)$ existiert, dann existieren die Funktionen $\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c})$ in diesem Intervall auch für alle hinreichend nahe an $u_0, \mathbf{v}_0, \mathbf{c}_0$ gelegenen $u, \mathbf{y}, \mathbf{c}$.
- (b) Die Funktionen $\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c})$ sind an jeder Stelle ihres Existenzbereiches stetige Funktionen ihrer Argumente $x, u, \mathbf{v}, \mathbf{c}$.

BEWEIS

Siehe Kamke (1969), S. 116, Satz 2.

Beispiel 3

Wir betrachten wiederum das Vakuum-Modell mit der Lösung

$$\begin{aligned} x(t) &= v_0 \cos(\theta)t \\ y(t) &= y_0 + v_0 \sin(\theta)t - \frac{1}{2}gt^2 \end{aligned}$$

Offensichtlich hängen die Lösungen in stetiger Weise von t und θ ab.

Gibt man nun ein $x(t_{\text{Ziel}}) = x_0$ vor und fordert, dass $y(t_{\text{Ziel}}) = 0$, dann liegt es nahe, das folgende Funktional zu minimieren:

$$h(\theta) = (x(t_{\text{Ziel}}) - v_0 \cos(\theta)t_{\text{Ziel}})^2 + (y(t_{\text{Ziel}}) - (y_0 + v_0 \sin(\theta)t_{\text{Ziel}} - \frac{1}{2}gt_{\text{Ziel}}^2))^2$$

Da es eine maximale Flugzeit gibt und ebenso einen maximalen Winkel für θ , wird die Funktion $h(\cdot)$ auf einem kompakten Intervall zu betrachten sein.

Aus Stetigkeitsgründen wird somit ein entsprechender Winkel existieren müssen.

Um den notwendigen Erhebungswinkel θ zu berechnen - will man in einer vorgegebenen Entfernung x_0 treffen - eliminieren wir zuerst den Parameter t , indem wir die erste Gleichung nach t auf lösen

$$t = \frac{x(t)}{v_0 \cos(\theta)}$$

und dann t in der zweiten Gleichung eliminieren:

$$y(x) = y_0 + \tan(\theta)x - \frac{1}{2}g \frac{x^2}{v_0^2 \cos^2(\theta)}$$

Wir beachten nun die Beziehung

$$\frac{1}{\cos^2(\theta)} = 1 + \tan^2(\theta)$$

und erhalten

$$0 = y(x_0) = y_0 + \tan(\theta)x_0 - \frac{gx_0^2}{2v_0^2} - \frac{gx_0^2}{2v_0^2} \tan^2(\theta)$$

Durch Umstellen ergibt sich hieraus eine quadratische Gleichung in $\tan(\theta)$

$$\frac{gx_0^2}{2v_0^2} \tan^2(\theta) - \tan(\theta)x_0 = y_0 - \frac{gx_0^2}{2v_0^2}$$

oder

$$\tan^2(\theta) - 2\frac{v_0^2 x_0}{gx_0^2} \tan(\theta) = \frac{2v_0^2 y_0}{gx_0^2} - 1$$

Mittels einer quadratischen Ergänzung ergibt sich hieraus

$$\tan^2(\theta) - 2\frac{v_0^2}{gx_0} \tan(\theta) + \left(\frac{v_0^2}{gx_0}\right)^2 = \left(\frac{v_0^2}{gx_0}\right)^2 + \frac{2v_0^2 y_0}{gx_0^2} - 1$$

und damit

$$\tan(\theta) = \frac{v_0^2}{gx_0} \pm \sqrt{\left(\frac{v_0^2}{gx_0}\right)^2 + \frac{2v_0^2 y_0}{gx_0^2} - 1}$$

woraus sich mithilfe der Umkehrfunktion des Tangens der gesuchte Winkel θ ergibt - sofern man fordert, dass der Winkel zwischen 0 und 45° liegen soll. Man nimmt hierzu die negative Wurzel.

In dem obigen Beispiel konnten wir den notwendigen Winkel θ analytisch bestimmen - dies ist sicherlich die Ausnahme.

Man kann aber das obige Beispiel verallgemeinern.

Sei hierzu ein Vektor \mathbf{y}_{Ziel} gegeben, dann ist ein x_{Ziel} und \mathbf{c}_{Ziel} gesucht mit

$$h(x_{Ziel}, \mathbf{c}_{Ziel}) = \|\mathbf{y}_{Ziel} - \mathbf{Y}(x_{Ziel}, u, v, \mathbf{c}_{Ziel})\|^2 = \min$$

Das Funktional

$$h(x, \mathbf{c}) = \|\mathbf{y}_{Ziel} - \mathbf{Y}(x, u, v, \mathbf{c})\|^2$$

soll somit minimiert werden.

Unterstellen wir nun, dass

- das ballistische Problem auf einer kompakten Teilmenge \mathcal{D} von $\mathcal{G} \times \mathcal{C}$ betrachtet werden kann
- \mathbf{y}_{Ziel} aus dem Bildraum $\mathbf{Y}(x, u, v, \mathbf{c})(\mathcal{D})$ entstammt

dann muss es nach dem Satz von Weierstrass vom Minimum und Maximum ein x_{Ziel} und ein \mathbf{c}_{Ziel} geben, welches das obige Funktional minimiert - das obige Funktional $h(\mathbf{c})$ ist ja nach dem vorhergehenden Satz stetig.

Das wesentliche Problem besteht damit darin, wie die Größe \mathbf{c}_{Ziel} berechnet werden kann.

Bevor wir einen Algorithmus angeben, um den notwendigen Parameter \mathbf{c} - in obigem Beispiel den notwendigen Winkel θ - zumindest iterativ zu bestimmen, wollen wir noch einen Satz angeben, der die Differenzierbarkeit der Integralkurven

$$\mathbf{y} = \mathbf{Y}(x, u, v, \mathbf{c})$$

sicherstellt und ein Differentialgleichungssystem erster Ordnung angibt - dies ist allerdings eher der Vollständigkeit halber.

Dabei sei $\mathbf{c} = (c_1, \dots, c_k)$ ein k -dimensionaler Vektor, $\mathbf{b} = (b_1, \dots, b_l)$ ein l -dimensionaler Vektor, \mathcal{C} eine offene Menge im \mathbf{c} -Raum und \mathcal{B} eine offene Menge im \mathbf{b} -Raum.

Ist $\mathcal{G}(x, \mathbf{y})$ ein Gebiet, dann sei $\mathcal{G} \times \mathcal{C} \times \mathcal{B}$ die Menge aller Punkte $x, \mathbf{y}, \mathbf{c}, \mathbf{b}$ mit $x, \mathbf{y} \in \mathcal{G}$, $\mathbf{c} \in \mathcal{C}$, $\mathbf{b} \in \mathcal{B}$.

Es gilt dann

Satz 3

Die Abbildung $\mathbf{f}(x, \mathbf{y}; \mathbf{c}, \mathbf{b})$ sei in $\mathcal{G} \times \mathcal{C} \times \mathcal{B}$ stetig und besitze stetige partielle Ableitungen bzgl $y_1, \dots, y_n, c_1, \dots, c_k$.

Für jeden Punkt \mathbf{c}, \mathbf{b} von $\mathcal{C} \times \mathcal{B}$ gibt es dann durch den Punkt $u, \mathbf{v} \in \mathcal{G}$ eine in \mathcal{G} von Rand zu Rand verlaufende Integralkurve

$$\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c}, \mathbf{b})$$

der Differentialgleichung

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}, \mathbf{c}, \mathbf{b})$$

Die Integralkurve $\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c}, \mathbf{b})$ ist an jeder Stelle $x, u, \mathbf{v}, \mathbf{c}, \mathbf{b}$ ihres Existenzbereiches nach $x, u, v_1, \dots, v_n, c_1, \dots, c_k$ partiell differenzierbar, und diese partiellen Ableitungen sind stetige Funktionen von $x, u, \mathbf{v}, \mathbf{c}, \mathbf{b}$.

Ferner ist die partielle Ableitung

$$\mathbf{w}_p = \frac{\partial \mathbf{Y}}{\partial c_p}, \quad 1 \leq p \leq k$$

die Lösung der linearen Differentialgleichung

$$\mathbf{w}' = \mathbf{f}_{c_p}(x, \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c}, \mathbf{b})) + \mathbf{F}(x, \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c}, \mathbf{b}), \mathbf{c}, \mathbf{b})\mathbf{w} \quad \text{mit} \quad \mathbf{w}_p(u) = \mathbf{0}$$

Hierbei ist \mathbf{F} die Matrix

$$\mathbf{F}(x, \mathbf{y}, \mathbf{c}, \mathbf{b}) = \frac{\partial f_p}{\partial y_q}, \quad p, q = 1, \dots, n$$

BEWEIS

Siehe Kamke (1969), S. 122, Satz 2.

Anmerkung

Ist die Abbildung $\mathbf{f}(x, \mathbf{y}, \mathbf{c}, \mathbf{b})$ entsprechend oft partiell differenzierbar, dann überträgt sich dies auf die Integralkurven

$$\mathbf{y} = \mathbf{Y}(x, u, \mathbf{v}, \mathbf{c}, \mathbf{b})$$

siehe hierzu Kamke (1969), S. 124, Satz 3, und S. 125, Satz 4.

Wir benötigen gegebenenfalls noch maximale und minimale Integralkurven.

Definition 1

Durch einen Punkt u, \mathbf{v} eines Gebiets $\mathcal{G}(x, \mathbf{y})$ möge eine Schar stetiger Kurven $\mathbf{y} = \mathbf{Y}(x)$ gehen.

Zwei Kurven $\mathbf{z}(x)$ und $\mathbf{Z}(x)$ dieser Schar heißen für ein Intervall - welches den Punkt u enthält - minimale oder maximale Kurve der Schar, wenn

$$\mathbf{z}(x) \leq \mathbf{Y}(x) \quad \text{bzw.} \quad \mathbf{Z}(x) \geq \mathbf{Y}(x)$$

für jede Kurve $\mathbf{Y}(x)$ der Schar gilt, sofern $\mathbf{z}(x)$, $\mathbf{Y}(x)$ bzw. $\mathbf{Z}(x)$, $\mathbf{Y}(x)$ an der Stelle x existieren.

Wir machen nun folgende Voraussetzungen:

- (a) Es sei u, \mathbf{v} ein Punkt des Gebiets $\mathcal{G}(x, \mathbf{y})$, $\mathbf{f}(x, \mathbf{y})$ stetig für die Punkte $x \neq u$ in \mathcal{G} , und für einen Bereich

$$0 < |x - u| < \delta, \quad \|\mathbf{y} - \mathbf{v}\| < \delta$$

soll es eine für $|x - u| < \delta$ integrierbare Funktion $A(x)$ geben, so dass

$$\|\mathbf{f}(x, \mathbf{y})\| \leq A(x)$$

für die obigen Punkte.

- (b) Die Komponenten der Abbildung $\mathbf{f}(x, \mathbf{y})$ erfüllen eine Monotonie-Voraussetzung

$$f_v(x, \mathbf{y}) \leq f_v(x, \bar{\mathbf{y}}) \quad \text{für} \quad x > u, \mathbf{y} \leq \bar{\mathbf{y}}, y_v = \bar{y}_v, v = 1, \dots, n$$

Unter den obigen Voraussetzungen kann man den folgenden Satz zeigen:

Satz 4

Erfüllt $\mathbf{f}(x, \mathbf{y})$ die obigen Voraussetzungen, so gibt es durch den Punkt u, \mathbf{v} für $x \geq u$ eine maximale und eine minimale Integralkurve, die beide nach rechts bis zum Rand von \mathcal{G} laufen.

BEWEIS

Siehe Kamke (1969), S. 127, Satz 2.

Beispiel 4

Wir betrachten nun die Situation, bei der der Luftwiderstand mit modelliert wird, und erhalten damit folgendes Differentialgleichungssystem:

$$\begin{aligned} \dot{v}_x &= -\hat{C}_d^* v v_x \\ \dot{v}_y &= -\hat{C}_d^* v v_y - g \\ \dot{v}_z &= -\hat{C}_d^* v v_z \end{aligned}$$

Hierbei ist Konstante \hat{C}_d^* das Luftwiderstandsgesetz - wir setzen hierbei voraus, dass \hat{C}_d^* eine stetige Funktion ist.

Löst man dieses Differentialgleichungssystem unter Berücksichtigung der Anfangsbedingungen $x(0)$, $y(0)$, $z(0)$, $v_x(0)$, $v_y(0)$ und $v_z(0)$. dann erhält man als Lösung einen Ortsvektor $\mathbf{x} = \mathbf{x}(t)$, der die Position des Projektils in Abhängigkeit von der Zeit t beschreibt.

Um dies zu erkennen, schreiben wir das Differentialgleichungssystem zweiter Ordnung als Differentialgleichungssystem erster Ordnung.

Wir definieren:

$$\mathbf{y}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \\ v_x(t) \\ v_y(t) \\ v_z(t) \end{pmatrix} \Rightarrow \dot{\mathbf{y}}(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \\ \dot{v}_x(t) \\ \dot{v}_y(t) \\ \dot{v}_z(t) \end{pmatrix}$$

und damit

$$\dot{\mathbf{y}}(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \dot{v}_x(t) \\ \dot{v}_y(t) \\ \dot{v}_z(t) \end{pmatrix} = \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \\ -\hat{C}_d^* v v_x \\ -\hat{C}_d^* v v_y - g \\ -\hat{C}_d^* v v_z \end{pmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

mit $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$.

Die Ausdrücke $\hat{C}_d^* v v_x$, $\hat{C}_d^* v v_y$ und $\hat{C}_d^* v v_z$ sind jeweils größer oder gleich Null, damit sind die entsprechenden Ausdrücke infolge des Negativzeichens jeweils kleiner oder gleich Null.

Somit sind die zugehörigen Komponenten von \mathbf{f} kleiner oder gleich 0, $-g$ und 0.

Die maximalen Integralkurven sind dann die des Vakuum-Modells zu den gleichen Anfangsbedingungen $x(0) = 0$, $y(0) = y_0$, $z(0) = 0$, $v_x(0) = v_0 \cos(\theta)$, $v_y(0) = v_0 \sin(\theta)$ und $v_z(0) = 0$.

Damit erhalten wir

$$\begin{aligned} x(t) &= v_0 \cos(\theta) t \\ y(t) &= y_0 + v_0 \sin(\theta) t - \frac{1}{2} g t^2 \\ z(t) &= 0 \end{aligned}$$

Wir berechnen nun die Flugzeit t_f bis zum Einschlag - dies ergibt

$$0 = y(t_f) = y_0 + v_0 \sin(\theta) t_f - \frac{1}{2} g t_f^2$$

Wir lösen nach t_f auf und erhalten

$$0 = \frac{2y_0}{g} + \frac{2v_0 \sin(\theta) t_f}{g} - t_f^2$$

oder

$$t_f^2 - \frac{2v_0 \sin(\theta) t_f}{g} + \left(\frac{v_0 \sin(\theta)}{g} \right)^2 = \frac{2y_0}{g} + \left(\frac{v_0 \sin(\theta)}{g} \right)^2$$

und damit

$$t_{f;1/2} = \frac{v_0 \sin(\theta)}{g} \pm \sqrt{\frac{2y_0}{g} + \left(\frac{v_0 \sin(\theta)}{g}\right)^2}$$

Beschränkt man sich auf die obere Wurzel, so erhält man für die Flugzeit

$$t_f = \frac{v_0 \sin(\theta)}{g} + \sqrt{\frac{2y_0}{g} + \left(\frac{v_0 \sin(\theta)}{g}\right)^2}$$

woraus sich für $y_0 = 0$ wiederum die bekannte Formel

$$t_f = \frac{2v_0 \sin(\theta)}{g}$$

ergibt.

Hieraus ergibt sich die Möglichkeit - da der Erhebungswinkel nach oben abgeschätzt werden kann - die Flugzeit nach oben abzuschätzen.

Allerdings kann die Flugzeit erheblich überschätzt werden.

Kapitel 2

Der Algorithmus

2.1 Die Grundvoraussetzungen

Wir setzen hier voraus, dass

- das Differentialgleichungssystem

$$\mathbf{y}' = \mathbf{f}(\mathbf{y})$$

welches die Geschossbewegung beschreibt, adäquat aufgestellt wurde

- die Endbedingungen

$$\mathbf{y}_{\text{Ziel}} = \mathbf{y}(t_{\text{Ziel}})$$

so gestellt wurden, dass gemäß der vorangestellten Sätze die Existenz einer Lösung gesichert ist

- zuerst die Situation betrachtet werden soll, dass als einziger zu bestimmender Parameter \mathbf{c} in der Lösung $\mathbf{Y}(x, u, \mathbf{v}, \mathbf{c})$ nur der Rohrerhöhungswinkel θ betrachtet werden soll.

Es stellt sich damit die Frage, wie dieser Winkel θ berechnet werden kann und insbesondere, wie gegebenenfalls auch ein notwendiger Vorhaltewinkel.

- dann die Situation betrachtet werden soll, dass ein Flugobjekt sich gemäß eines Differentialgleichungssystems bewegt und abgeschossen werden soll.

Es stellt sich damit die Frage nach dem notwendigen Rohrerhöhungs- und Vorhaltewinkel.

- zuletzt die Situation betrachtet werden soll, dass sowohl der Schütze wie auch das Flugobjekt sich jeweils gemäß eines Differentialgleichungssystems bewegen und das Flugobjekt abgeschossen werden soll.

2.2 Die beiden Algorithmen für Fleckschussberechnung

Der Ausgangspunkt ist das Differentialgleichungssystem

$$\mathbf{y}' = \mathbf{f}(\mathbf{y})$$

mit der Anfangsbedingung v , der Mündungsgeschwindigkeit.

Wir hatten bereits das Funktional

$$h(x_{\text{Ziel}}, \mathbf{c}_{\text{Ziel}}) = \| \mathbf{y}_{\text{Ziel}} - \mathbf{Y}(x_{\text{Ziel}}, u, v, \mathbf{c}_{\text{Ziel}}) \|^2 = \min$$

definiert.

Dieses Funktional

$$h(x, \mathbf{c}) = \| \mathbf{y}_{\text{Ziel}} - \mathbf{Y}(x, u, v, \mathbf{c}) \|^2$$

soll somit minimiert werden.

Dieses bedeutet, dass wir einen Zeitpunkt t_{Ziel} und einen Winkel θ^* finden müssen, so dass

$$\begin{aligned} x(t_{\text{Ziel}}) &= x_{\text{Ziel}} \\ y(t_{\text{Ziel}}) &= y_{\text{Ziel}} \end{aligned}$$

gilt (gegebenenfalls kommt noch ein Winkel für die z -Komponente hinzu, ein Seitenkorrekturwinkel, falls der Einfluss von Seitenwind mit eingerechnet werden soll).

Äquivalent hierzu ist

$$\begin{aligned} x(t_{\text{Ziel}}) - x_{\text{Ziel}} &= 0 \\ y(t_{\text{Ziel}}) - y_{\text{Ziel}} &= 0 \end{aligned}$$

Wir haben damit prinzipiell zwei Möglichkeiten:

- Wir können unser Problem als Optimierungsaufgabe auffassen.
- Wir können unser Problem als mehrdimensionales Nullstellenproblem formulieren.

Ganz allgemein betrachten wir im Fall der zweiten Möglichkeit ein nichtlineares System von Gleichungen der Form

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned}$$

Dabei sind f_1, f_2, \dots, f_n nichtlineare Funktionen.

Mit $\mathbf{F} = (f_1, \dots, f_n)^T$ und $\mathbf{x} = (x_1, \dots, x_n)^T$ kann das obige System auch in der Form

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

geschrieben werden - gesucht ist somit ein Vektor \mathbf{x} , der das obige System erfüllt.

Ein derartiges Problem kann mit Hilfe des mehrdimensionalen Newton-Verfahrens gelöst werden - eine natürliche Verallgemeinerung des eindimensionalen Verfahrens auf höhere Dimensionen zur Bestimmung von Nullstellen von Funktionen einer Veränderlicher.

Für das mehrdimensionale Newton-Verfahren benötigen wir die Funktionalmatrix

$$\mathbf{J}_{\mathbf{F}} = \left(\frac{\partial f_i}{\partial x_j} \right), \quad i, j = 1, 2, \dots, n$$

Ist somit ein Startvektor $\mathbf{x}^{(0)} \in \mathbb{R}^n$ gegeben, dann wird das mehrdimensionale Newton-Verfahren durch folgende Gleichungen beschrieben:

$$\begin{array}{l} \text{Löse } \mathbf{J}_{\mathbf{F}}(\mathbf{x}^{(k)}) \delta \mathbf{x}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k)}) \\ \text{Setze } \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta \mathbf{x}^{(k)} \end{array} \quad \text{für } k = 0, 1, 2, \dots$$

siehe hierzu Quarteroni u. a. (2014), S. 52 ff.

Wir müssen somit in jedem Iterationsschritt ein Gleichungssystem lösen - zusätzlich muss die Funktionalmatrix bereitgestellt werden.

Da die Bereitstellung der Funktionalmatrix sehr aufwändig ist, behilft man sich mit Näherungen für die Funktionalmatrix - dies führt auf sogenannte quasi-Newton-Methoden, siehe hierzu Quarteroni u. a. (2014), S. 55.

Hier bietet sich die Funktion `fsolve` aus dem Programm Octave an, eine Implementierung derartiger Verfahren (ist auch in Matlab verfügbar).

`fsolve` ist ein Programm zur Lösung von

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

wobei

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix}, \quad \mathbf{x} \in \mathbb{R}^n$$

und $\mathbf{0} \in \mathbb{R}^n$.

Gesucht ist hierbei ein \mathbf{x}^* mit $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$.

Informationen zu diesem Programm können im Buch von Quarteroni u. a. (2014) auf S. 73 gefunden werden.

Als weitere Möglichkeit bietet sich an, das Funktional $h(x, \mathbf{c})$ direkt zu minimieren.

Im Buch von Quarteroni u. a. (2014) wird hierzu das Programm `fminunc` zur Verfügung gestellt, sowohl in Octave wie auch in Matlab verfügbar - hier ist die BFGS-Methode implementiert, ein typischer Vertreter der quasi-Newton-Verfahren zur Minimierung von Funktionalen.

Beiden Verfahren ist dabei eine superlineare Konvergenz gemeinsam, es scheint aber so, dass das mehrdimensionale Newton-Verfahren etwas schneller konvergiert.

Im Kapitel 7 des Buches von Quarteroni u. a. (2014) sind weitere Verfahren zur Minimierung von Funktionalen angeführt - es ist sicherlich interessant, in unserer Situation die von Octave bzw. Matlab angebotenen Implementationen auszuprobieren.

Bereits mit `fsolve` wurden gute Erfahrungen gewonnen - das Verfahren scheint, wie gesagt, rasch zu konvergieren.

Anmerkung

Man muss hier folgenden Punkt sehen:

Im Normalfall kann das zugehörige Differentialgleichungssystem nur noch mit den Methoden der numerischen Mathematik gelöst werden.

Lesenswert sind sicherlich die Bücher von Strehmel u. a. (2012) oder Hermann (2004) über die Numerik gewöhnlicher Differentialgleichungen.

Das Buch von Shampine u. a. (2003) beschäftigt sich ausschließlich mit den in Matlab zur Verfügung gestellten Verfahren zum Lösen von Differentialgleichungen - eine kritische Würdigung und Abgrenzung findet man im Buch von Strehmel u. a. (2012).

Ein echtes Randwertproblem liegt wohl eher nicht vor - der Rohrerhebungswinkel ist eigentlich als Parameter zu werten, ebenso ein möglicher Seitenkorrekturwinkel, siehe hierzu auch zum Beispiel das Buch von Ascher u. a. (1988).

Im Fall vom Präzisionsschießen verbleibt noch, die Anzahl der Klicks zu berechnen, um die das Zielfernrohr verstellt werden muss.

Nun wird in der Regel der Verstellbereich eines Zielfernrohrs auf 100 Meter (im Angelsächsischen: 100 Yard) bezogen, und hier je nachdem auf MOA (Minute of Angle) oder im Europäischen auf Zentimeter. So sind gebräuchlich:

- 1/8 MOA - dies entspricht etwa 3,6 Millimeter, somit bedeutet ein Klick eine Verstellung von 3,6 Millimetern auf 100 Meter
- 1/4 MOA - dies entspricht etwa 7,2 Millimeter, somit bedeutet ein Klick eine Verstellung von 7,2 Millimetern auf 100 Meter
- 1 Zentimeter - dies entspricht 10 Millimeter, somit bedeutet ein Klick eine Verstellung von 10 Millimetern auf 100 Meter

Aus diesem Grunde ist es sinnvoll, die Flughöhe y_{100} auf 100 Meter zu berechnen, wenn der

Erhebungswinkel θ^* berechnet worden ist, mit dem auf die Zielentfernung x_{Ziel} geschossen werden kann.

Kennt man diese Flughöhe, dann kann man das Zielfernrohr auf diese Flughöhe y_{100} einstellen - die Waffe schießt dann auf x_{Ziel} Fleck.

Ist somit $y_{100} = 12$ Zentimeter, dann reichen 12 Klicks bei einer Verstellung von 1 Zentimeter pro Klick - die Waffe schießt dann auf 100 Meter 12 Zentimeter hoch und damit auf die Entfernung x_{Ziel} Fleck.

Um nun die Flughöhe berechnen zu können, muss die Flugzeit t_{100} berechnet werden - dies ist ein einfaches Nullstellenproblem.

Ist die Flugzeit t_{100} bekannt, kann die zugehörige Flughöhe berechnet werden.

Zur Berechnung der Flugzeit t_{100} ist die Funktion `fzero` hilfreich - diese ist in Octave wie auch in Matlab verfügbar.

Ein Prototyp wurde aufgesetzt - er funktioniert recht ordentlich.

2.3 Ein Algorithmus für Flugobjektabschuss

Auch hier ist das Differentialgleichungssystem

$$\mathbf{y}' = \mathbf{f}(\mathbf{y})$$

mit der Anfangsbedingung v , der Mündungsgeschwindigkeit, für die Flugbahn des Geschosses Ausgangspunkt für alle weiteren Überlegungen.

Zusätzlich kommt aber ein weiteres Differentialgleichungssystem

$$\mathbf{z}' = \mathbf{g}(\mathbf{z})$$

mit einer Anfangsbedingung $\mathbf{z}(t_0) = \mathbf{z}_0$ für das abzuschießende Flugobjekt hinzu.

Als Funktional - welches minimiert werden soll - wählen wir nun

$$h(x, \mathbf{c}) = \|\mathbf{z}(x) - \mathbf{Y}(x, u, v, \mathbf{c})\|^2$$

Dabei soll $\mathbf{z}(x)$ die Lösung des Differentialgleichungssystems

$$\mathbf{z}' = \mathbf{g}(\mathbf{z})$$

mit der Anfangsbedingung $\mathbf{z}(t_0) = \mathbf{z}_0$ sein.

Wie schon im vorhergehenden Abschnitt ausgeführt, kann diese Minimierung mit Hilfe des Programms `fminunc` durchgeführt werden.

Ein Treffer wäre gegeben, wenn das obige Funktional den Wert Null annimmt - dann ist das Flugobjekt zerstört (zumindest aus mathematischer Sicht).

Damit ist das Problem aus mathematischer Sicht zumindest gelöst.

Ist nun der erreichte Wert bei der Minimierung größer als Null, so ist damit eine Lösung im mathematischen Sinn nicht möglich.

Aus diesem Grunde ist in dieser Situation eine Anwendung der Funktion f_{\min} sinnvoll.

Beispiel 5

Wir setzen das Vakuum-Modell an mit der Lösung

$$\mathbf{y}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} v \cos(\alpha)t \\ y_0 + v \sin(\alpha)t - \frac{1}{2}gt^2 \end{pmatrix}$$

und für das Flugobjekt eine geradlinig gleichförmige Bewegung

$$\mathbf{z}(t) = \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} v_x t \\ v_y t \end{pmatrix} = \begin{pmatrix} a_x + v_x t \\ a_y + v_y t \end{pmatrix}$$

Dann ergibt sich

$$\mathbf{y} - \mathbf{z} = \begin{pmatrix} v \cos(\alpha)t - (a_x + v_x t) \\ y_0 + v \sin(\alpha)t - \frac{1}{2}gt^2 - (a_y + v_y t) \end{pmatrix}$$

Das Quadrat der Norm dieses Vektors soll nun gleich Null sein, somit

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v \cos(\alpha)t - (a_x + v_x t) \\ y_0 + v \sin(\alpha)t - \frac{1}{2}gt^2 - (a_y + v_y t) \end{pmatrix}$$

Damit müssen α^* und t^* so bestimmt werden, dass

$$\begin{aligned} 0 &= (v \cos(\alpha^*) - v_x)t^* - a_x \\ 0 &= y_0 - a_y + (v \sin(\alpha^*) - v_y)t^* - \frac{1}{2}g(t^*)^2 \end{aligned}$$

gilt.

Aus der ersten Gleichung folgt

$$t^* = \frac{a_x}{v \cos(\alpha^*) - v_x} = \frac{a_x}{v \left(\cos(\alpha^*) - \frac{v_x}{v} \right)}$$

Setzen wir t^* in die zweite Gleichung ein, so erhalten wir

$$0 = y_0 - a_y + (v \sin(\alpha^*) - v_y) \frac{a_x}{v \cos(\alpha^*) - v_x} - \frac{1}{2}g \left(\frac{a_x}{v \cos(\alpha^*) - v_x} \right)^2$$

Diese Gleichung kann nur noch numerisch nach α^* aufgelöst werden - aus numerischen Gründen ist es wohl besser, die Gleichung wie folgt umzuschreiben:

$$0 = (y_0 - a_y)(v \cos(\alpha^*) - v_x)^2 + a_x(v \sin(\alpha^*) - v_y)(v \cos(\alpha^*) - v_x) - \frac{1}{2} g a_x^2$$

Betrachten wir folgendes numerische Beispiel:

$$\begin{aligned} a_x &= 1000m & v_x &= -200\frac{m}{s} \\ a_y &= 20m & v_y &= -10\frac{m}{s} \end{aligned}$$

und $v = 800\frac{m}{s}$, $y_0 = 0$ - das Flugobjekt startet somit in 1000 Metern Entfernung in 20 Metern Höhe und fliegt auf den Ursprung zu mit ca. 720 Stundenkilometern.

Dann lautet die obige Gleichung

$$\begin{aligned} 0 &= (0 - 20)(800 \cos(\alpha^*) + 200)^2 + 1000(800 \sin(\alpha^*) + 10)(800 \cos(\alpha^*) + 200) - \frac{1}{2} 9,81 \cdot 1000^2 \\ &= -20(800 \cos(\alpha^*) + 200)^2 + 1000(800 \sin(\alpha^*) + 10)(800 \cos(\alpha^*) + 200) - \frac{1}{2} 9,81 \cdot 1000^2 \end{aligned}$$

Die Lösung ergibt einen Winkel von $\alpha^* = 1,0674^\circ$ - hierzu schreibt man am besten ein kleines Programm in Octave oder Matlab und benützt die Funktion `fzero`.

Die Flugzeit ergibt sich zu

$$t^* = \frac{a_x}{v \cos(\alpha^*) - v_x} = \frac{1000}{800 \cos(1,0674) - (-200)} = \frac{1000}{999,86} = 1,00013 \text{sec}$$

Ist nun bekannt, dass α^* klein ist, dann gilt

$$\cos(\alpha^*) \approx 1$$

Hieraus folgt

$$(a_y - y_0)(v \cos(\alpha^*) - v_x)^2 = a_x(v \sin(\alpha^*) - v_y)(v \cos(\alpha^*) - v_x) - \frac{1}{2} g a_x^2$$

und damit

$$(a_y - y_0)(v - v_x)^2 + \frac{1}{2} g a_x^2 = a_x(v \sin(\alpha^*) - v_y)(v - v_x)$$

Damit kann die Gleichung nach $\sin(\alpha^*)$ und somit nach α^* aufgelöst werden.

Setzt man die Werte aus dem obigen numerischen Beispiel ein, dann erhält man für α^* den Wert:

$$\alpha^* = 1,0675^\circ$$

Die Näherung ist somit in dieser Situation recht genau.

2.4 Der Algorithmus für Abschuss aus der Bewegung heraus

Man kann das obige Beispiel erweitern, insofern man davon ausgeht, dass der Schütze selber sich bewegt - zum Beispiel aus einem Hubschrauber heraus schießt, der sich ebenfalls mit einer gleichförmig geradlinigen Bewegung $\mathbf{s}(t)$ bewegt.

Wieder beschreibe $\mathbf{z}(t)$ die Bewegung des Zieles.

Wir setzen für die Bewegung (zum Beispiel des Hubschraubers) an:

$$\mathbf{s}(t) = \mathbf{c} + \mathbf{h} \cdot t$$

Hierbei sind \mathbf{c} und \mathbf{h} jeweils konstante Vektoren - $\mathbf{s}(t)$ kann durchaus auch durch ein Differentialgleichungssystem gegeben sein.

Ist nun

$$\mathbf{y}' = \mathbf{f}(\mathbf{y})$$

das zugehörige Differentialgleichungssystem mit der Anfangsbedingung $\mathbf{y}(t_0) = \mathbf{y}_0$ für die Geschossbewegung, dann betrachtet man stattdessen mit $\mathbf{w} = \mathbf{y} + \mathbf{s}$ das Differentialgleichungssystem

$$\mathbf{w}' = \mathbf{f}(\mathbf{w})$$

mit der Anfangsbedingung $\mathbf{w}(t_0) = \mathbf{y}(t_0) + \mathbf{s}(t_0)$ in Analogie zu McCoy (1999), S. 157 ff.

Löst man dieses System, dann erhält man mit \mathbf{w} die gesuchte Lösung für die Geschossbewegung.

Damit kann man dann aber gemäß des vorhergehenden Beispiels den gesuchten Vorhaltewinkel berechnen.

Beispiel 6

Wir setzen wiederum wie vorhin für die Geschossbewegung ein zweidimensionales Vakuum-Modell an und für das Flugobjekt eine geradlinig gleichförmige Bewegung

$$\mathbf{z}(t) = \begin{pmatrix} a_x \\ a_y \end{pmatrix} + \begin{pmatrix} v_x t \\ v_y t \end{pmatrix} = \begin{pmatrix} a_x + v_x t \\ a_y + v_y t \end{pmatrix}$$

Für den Hubschrauber setzen wir an

$$\mathbf{s}(t) = \begin{pmatrix} c_x + h_x t \\ c_y + h_y t \end{pmatrix}$$

Das Differentialgleichungssystem für \mathbf{w} lautet dann

$$\begin{aligned} \ddot{w}_x &= \frac{d}{dt}(\dot{x} + h_x) = 0 \\ \ddot{w}_y &= \frac{d}{dt}(\dot{y} + h_y) = -g \end{aligned}$$

Damit ergibt sich

$$(\mathbf{w} - \mathbf{z})(t) = \begin{pmatrix} v \cos(\alpha)t + c_x + h_x t - (a_x + v_x t) \\ y_0 + v \sin(\alpha)t + c_y + h_y t - \frac{1}{2}gt^2 - (a_y + v_y t) \end{pmatrix}$$

Das Quadrat der Norm dieses Vektors soll nun gleich Null sein, somit

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} v \cos(\alpha)t + c_x + h_x t - (a_x + v_x t) \\ y_0 + v \sin(\alpha)t + c_y + h_y t - \frac{1}{2}gt^2 - (a_y + v_y t) \end{pmatrix}$$

Damit müssen α^* und t^* so bestimmt werden, dass

$$\begin{aligned} 0 &= (v \cos(\alpha^*) + h_x - v_x)t^* + c_x - a_x \\ 0 &= y_0 + c_y - a_y + (v \sin(\alpha^*) + h_y - v_y)t^* - \frac{1}{2}g(t^*)^2 \end{aligned}$$

gilt.

Aus der ersten Gleichung folgt

$$t^* = \frac{a_x - c_x}{v \cos(\alpha^*) + h_x - v_x}$$

Setzen wir t^* in die zweite Gleichung ein, so erhalten wir

$$0 = y_0 + c_y - a_y + (v \sin(\alpha^*) + h_y - v_y) \frac{a_x}{v \cos(\alpha^*) + h_x - v_x} - \frac{1}{2}g \left(\frac{a_x}{v \cos(\alpha^*) + h_x - v_x} \right)^2$$

Diese Gleichung kann nur noch numerisch nach α^* aufgelöst werden - aus numerischen Gründen ist es wohl besser, die Gleichung wie folgt umzuschreiben:

$$0 = (y_0 + c_y - a_y)(v \cos(\alpha^*) + h_x - v_x)^2 + a_x(v \sin(\alpha^*) + h_y - v_y)(v \cos(\alpha^*) + h_x - v_x) - \frac{1}{2}ga_x^2$$

Fügen wir zu den Daten aus dem vorhergehenden Beispiel:

$$\begin{aligned} a_x &= 1000m & v_x &= -200\frac{m}{s} \\ a_y &= 20m & v_y &= -10\frac{m}{s} \end{aligned}$$

und $v = 800\frac{m}{s}$, $y_0 = 0$ noch $c_x = c_y = 0$ und $h_y = 0\frac{m}{s}$ und $h_x = 20\frac{m}{s}$ hinzu, dann lautet die obige Gleichung

$$\begin{aligned} 0 &= (0 - 20)(800 \cos(\alpha^*) + 220)^2 + 1000(800 \sin(\alpha^*) + 10)(800 \cos(\alpha^*) + 220) - \frac{1}{2}9,81 \cdot 1000^2 \\ &= -20(800 \cos(\alpha^*) + 220)^2 + 1000(800 \sin(\alpha^*) + 10)(800 \cos(\alpha^*) + 220) - \frac{1}{2}9,81 \cdot 1000^2 \end{aligned}$$

Die Lösung ergibt einen Winkel von $\alpha^* = 1,0892^\circ$.

Die Flugzeit ergibt sich zu

$$t^* = \frac{a_x}{v \cos(\alpha^*) + h_x - v_x} = \frac{1000}{800 \cos(1,0892) + 20 - (-200)} = \frac{1000}{1019,85} = 0,9805 \text{sec}$$

Man kann nun die Näherung aus dem vorhergehenden Beispiel übertragen.

Ist nun bekannt, dass α^* klein ist, dann gilt

$$\cos(\alpha^*) \approx 1$$

Damit ergibt sich

$$0 = -20(800 + 220)^2 + 1000 \cdot (800 \sin(\alpha^*) + 10) \cdot (800 + 220) - \frac{1}{2}9,81 \cdot 1000^2$$

oder

$$0 = -20 \cdot 1040400 + 1000 \cdot (800 \sin(\alpha^*) + 10) \cdot 1020 - 4,905 \cdot 1000000$$

Somit

$$2 \cdot 10404 + 4905 = (800 \sin(\alpha^*) + 10) \cdot 1020$$

Hieraus folgt durch Auflösen nach α^* :

$$\alpha^* = 1,0893^\circ$$

Kapitel 3

Etwas Numerik

3.1 Einführung

In den vorhergehenden Abschnitten betrachteten wir Differentialgleichungssystem

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}) \quad \text{mit} \quad \mathbf{y}(t_0) = \mathbf{y}_0$$

Nur in den seltensten Fällen ist es möglich, dieses System analytisch zu lösen.

Beispiel 7

Gegeben die Differentialgleichung

$$y' = \alpha y \quad \text{mit} \quad y(0) = 1$$

dann ist die Lösung offensichtlich

$$y(t) = \exp(\alpha t)$$

Beispiel 8

Die Differentialgleichung

$$y'' = -k \sin(y), \quad k \text{ Konstante}$$

(siehe Kamke (1969), S. 2) ist nicht mehr analytisch lösbar.

Man gelangt zu dieser Differentialgleichung, wenn man die Kraftgleichung für das mathematische Pendel aufstellt.

Dies bedeutet, dass man darauf angewiesen ist, die anstehenden Differentialgleichungssysteme mit den Mitteln der numerischen Mathematik zu lösen.

In den nachfolgenden Abschnitten wollen wir einige Hilfsmittel bereitstellen.

3.2 Differenzengleichungen

Der Ausgangspunkt unserer Betrachtungen ist somit ein Differentialgleichungssystem $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$,
ausgeschrieben

$$\begin{aligned}\dot{x}_1(t) &= f_1(t, x_1(t), \dots, x_n(t)) \\ \dot{x}_2(t) &= f_2(t, x_1(t), \dots, x_n(t)) \\ &\vdots \\ \dot{x}_n(t) &= f_n(t, x_1(t), \dots, x_n(t))\end{aligned}$$

Dabei sei $\dot{x}_i(t)$ die Ableitung der Funktion $x_i(t)$ nach t , $i = 1, \dots, n$.

Wir gehen davon aus, dass $t \in J \subset \mathbb{R}$ und

$$\mathbf{y}(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix} \in \Omega \subset \mathbb{R}^n$$

Die Abbildung $\mathbf{f} : J \times \Omega \rightarrow \mathbb{R}^n$ nennt man auch ein Vektorfeld und $\mathbf{y}(t)$ eine Lösung des Differentialgleichungssystems.

Im **Unterschied** zu dem bisher betrachteten Fall soll nun die Variable t ein diskreter Parameter $i \in \mathbb{N}_0$ sein.

Damit ist $\Omega \subset \mathbb{R}^n$ und $J \subset \mathbb{N}_0$.

Wir betrachten nun eine Folge von Vektorfeldern

$$\{\mathbf{f}_i(\mathbf{y})\}_{i \in J}, \quad \mathbf{f}_i : \Omega \rightarrow \mathbb{R}^n$$

mit $\mathbf{f}_i(\mathbf{y}) = \mathbf{f}(i, \mathbf{y})$.

Wir können damit definieren:

Definition 2

Unter einer Differenzengleichung erster Ordnung versteht man die Vorschrift

$$\mathbf{y}_{i+1} = \mathbf{f}_i(\mathbf{y}_i)$$

Eine Vektorfolge $\{\mathbf{y}\}_{i \in J}$, die die obige Gleichung erfüllt, heißt dann Lösung der Differenzengleichung.

Wie im Fall der Differentialgleichungssysteme kann man auch noch einen Anfangsvektor $\mathbf{y}_0 \in \mathbb{R}^n$ vorgeben - hieraus resultiert dann ein Anfangswertproblem

$$\begin{aligned}\mathbf{y}_{i+1} &= \mathbf{f}_i(\mathbf{y}_i), \quad i \in J \\ \mathbf{y}_0 &= \mathbf{v}\end{aligned}$$

3.3 Diskretisierungen

Wie wir anhand der ersten Beispiele gesehen haben, lässt sich nicht in jedem Fall die Lösung einer Differentialgleichung exakt angeben - man muss in dieser Situation auf die Möglichkeiten der numerischen Mathematik zurückgreifen.

Mit derartigen Techniken ist es allerdings nicht möglich, die Lösung in geschlossener Form anzugeben - man kann die Lösung nur an vorgegebenen diskreten Stellen berechnen.

Sei hierzu $[t_0, T]$ dasjenige Zeitintervall, auf dem eine eindeutige Lösung von

$$\begin{aligned} \mathbf{y}'(t) &= \mathbf{f}(t, \mathbf{y}(t)), & t \in J \\ \mathbf{y}(t_0) &= \mathbf{y}_0 \end{aligned}$$

existiert und auf dem man diese Lösung numerisch approximieren möchte.

Damit eine derartige Lösung existiert und eindeutig ist, setzen wir die Voraussetzungen des Satzes von Picard und Lindelöf voraus.

Das Intervall $[t_0, T]$ werde nun durch die Festlegung von $N + 1$ diskreten Punkten

$$t_0 < t_1 < \dots < t_N = T$$

in N Segmente unterteilt.

Wir definieren damit

Definition 3

Die diskreten Zeitpunkte bilden ein Gitter

$$J_h = \{t_0, t_1, \dots, t_N\}$$

und heißen daher Gitterpunkte oder auch Knoten.

Weiterhin bezeichnen wir mit

$$h_j = t_{j+1} - t_j, \quad j = 0, \dots, N-1$$

die Schrittweite von einem Gitterpunkt zum anderen.

Im Fall, dass die Schrittweiten konstant sind, d. h. $h_j = h$, somit

$$h = \frac{t_N - t_0}{N}$$

nennt man das Gitter äquidistant.

Im Fall eines nichtäquidistanten Gitter setzen wir

$$h = \max_{j=0, \dots, N-1} h_j$$

Wir können nun nur mit diskreten Werten der Funktion $\mathbf{y}(t)$ arbeiten, deshalb definieren wir

Definition 4

Unter einer Diskretisierung der Funktion $\mathbf{y}(t)$ mit $t_0 \leq t \leq t_N$ verstehen wir die Projektion von $\mathbf{y}(t)$ auf das zugrundeliegende Gitter J_h .

Wir erhalten damit die Folge $\mathbf{y}(t_i)_{t_i \in J_h}$.

Eine Funktion, die nur auf einem diskreten Gitter definiert ist, nennen wir auch eine Gitterfunktion.

Das Ziel ist nun, für die im allgemeinen unbekannte Gitterfunktion $\mathbf{y}(t_i)_{t_i \in J_h}$ eine genäherte Gitterfunktion \mathbf{y}_i , $i = 0, \dots, N$ zu berechnen.

Demzufolge sind Näherungen \mathbf{y}_i für $\mathbf{y}(t_i)$ mit $t_i \in J_h$ gesucht.

Diese Näherungen werden mit Hilfe eines numerischen Diskretisierungsverfahrens berechnet.

Definition 5

Unter einem numerischen Diskretisierungsverfahren zur Approximation der Lösung $\mathbf{y}(t)$ der obigen Anfangswertaufgabe

$$\begin{aligned} \mathbf{y}'(t) &= \mathbf{f}(t, \mathbf{y}(t)), \quad t \in J \\ \mathbf{y}(t_0) &= \mathbf{y}_0 \end{aligned}$$

verstehen wir eine Rechenvorschrift, welche eine Gitterfunktion \mathbf{y}_i , $i = 0, \dots, N$, erzeugt, für die

$$\mathbf{y}_i \approx \mathbf{y}(t_i), \quad t_i \in J_h$$

gilt.

Anmerkung

Um die Abhängigkeit der Näherung von der verwendeten Schrittweite zu kennzeichnen, wollen wir auch die Notation \mathbf{y}_i^h bzw. $\mathbf{y}^h(t_i)$ anstelle von \mathbf{y}_i verwenden.

Damit stellt sich die Frage, wie man zu einem geeigneten numerischen Diskretisierungsverfahren gelangen könnte.

Ein möglicher gangbarer Weg führt über die Integraldarstellung

$$\mathbf{y}(t) = \mathbf{y}_0 + \int_{t_0}^t \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau$$

der Anfangswertaufgabe.

Wendet man diese Integraldarstellung auf dem Intervall $[t_0, t_{i+1}]$ an, so erhält man

$$\mathbf{y}(t_{i+1}) = \mathbf{y}_0 + \int_{t_0}^{t_{i+1}} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau$$

Unterstellt man, dass für $\mathbf{y}(t_{i+1})$ eine Approximation \mathbf{y}_{i+1} bereits bekannt ist, dann lässt sich durch Interpolation ein Polynom vom Grad höchstens $i + 1$ finden, das durch die Punkte

$$(t_0, \mathbf{f}(t_0, \mathbf{y}_0)), (t_1, \mathbf{f}(t_1, \mathbf{y}_1)), \dots, (t_{i+1}, \mathbf{f}(t_{i+1}, \mathbf{y}_{i+1}))$$

verläuft und $\mathbf{f}(t, \mathbf{y})$ im betrachteten Intervall annähert.

Man kann nun das Interpolationspolynom im Integral einsetzen und das Integral berechnen - daraus resultiert

$$\mathbf{y}(t_{i+1}) \approx \mathbf{y}_0 + \sum_{j=0}^{i+1} w_{ij} \mathbf{f}(t_{i-j+1}, \mathbf{y}_{i-j+1}), \quad 0 \leq i \leq N-1$$

Damit stellt die rechte Seite des obigen Ausdrucks eine Näherung für $\mathbf{y}(t_{i+1})$ dar - sie wird als \mathbf{y}_{i+1} verwendet.

Hieraus resultiert aber ein Diskretisierungsverfahren zur Bestimmung einer Näherung \mathbf{y}_{i+1} für $\mathbf{y}(t_{i+1})$:

$$\mathbf{y}_{i+1} = \mathbf{y}_0 + \sum_{j=0}^{i+1} w_{ij} \mathbf{f}(t_{i-j+1}, \mathbf{y}_{i-j+1}), \quad 0 \leq i \leq N-1$$

Dieses Diskretisierungsverfahren resultiert aus einer Integration - deshalb bezeichnet man eine solche Vorschrift auch als Quadraturformel.

Ihrer Natur nach stellt diese eine spezielle Differenzgleichung dar.

Man unterscheidet:

Ist $w_{i0} \neq 0$, dann spricht man von einer impliziten Formel. Hierbei tritt das zu berechnende \mathbf{y}_{i+1} auch auf der rechten Seite auf und leider innerhalb der typischerweise nichtlinearen Funktion \mathbf{f} .

Deshalb müssen zur Lösung des daraus resultierenden Gleichungssystems iterative Techniken wie zum Beispiel das mehrdimensionale Newton-Verfahren eingesetzt werden.

Ist dagegen $w_{j0} = 0$, dann liegt eine explizite Formel vor - die Formel kann direkt ausgewertet werden.

Die Vor- und Nachteile der jeweiligen Verfahren liegen zum Beispiel in der sogenannten Stabilität - siehe zum Beispiel das Buch von Hermann (2004).

Nun ist die Formel

$$\mathbf{y}_{i+1} = \mathbf{y}_0 + \sum_{j=0}^{i+1} w_{ij} \mathbf{f}(t_{i-j+1}, \mathbf{y}_{i-j+1}), \quad 0 \leq i \leq N-1$$

für große N nicht sehr effizient - man modifiziert deshalb die Integraldarstellung

$$\mathbf{y}(t_{i+1}) = \mathbf{y}_0 + \int_{t_0}^{t_{i+1}} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau$$

und erhält

$$\mathbf{y}(t_{i+1}) = \mathbf{y}(t_{i-k+1}) + \int_{t_{i-k+1}}^{t_{i+1}} \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau$$

Geht man analog wie vorher vor, so erhält man hieraus die Quadraturformel

$$\mathbf{y}_{i+1} = \mathbf{y}_{i-k+1} + \sum_{j=0}^k \hat{w}_{ij} \mathbf{f}(t_{i-j+1}, \mathbf{y}_{i-j+1}), \quad 0 \leq i \leq N-1$$

Dies ist eine k -Schritt Differenzengleichung.

Beispiel 9

Eine recht einfache Strategie, die Funktion f auf dem Intervall $[t_i, t_{i+1}]$ zu approximieren, besteht darin, deren Wert an der Stelle (t_i, x_i) als Näherung heranzuziehen.

D. h. wir verwenden als Interpolationspolynom 0-ten Grades $P(t) = f(t_i, x_i)$.

Daraus ergibt sich mit $k = 1$

$$x(t_{i+1}) \approx x(t_i) + \int_{t_i}^{t_{i+1}} f(t_i, x_i) d\tau = x(t_i) + h_i f(t_i, x_i)$$

Damit nimmt das Diskretisierungsverfahren die Form

$$x_{i+1} = x_i + h_i f(t_i, x_i)$$

an.

Dieses Verfahren ist unter dem Namen Euler-Verfahren bekannt - es gehört zur Klasse der sogenannten Einschrittverfahren.

Definition 6

Unter einem Einschrittverfahren bzw. einer 1-Schritt-Differenzgleichung versteht man eine Vorschrift der Form

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \Phi(t_i, \mathbf{x}_i, \mathbf{x}_{i+1}; h_i), \quad i = 0, \dots, N-1$$

bei der in die Berechnung der Näherung \mathbf{x}_{i+1} für die exakte Lösung $\mathbf{x}(t_{i+1})$ nur die unmittelbar zuvor bestimmte Näherung \mathbf{x}_i an der Stelle t_i eingeht.

Da die Verfahrens- oder Inkrementfunktion Φ sowohl von \mathbf{x}_i als auch von \mathbf{x}_{i+1} abhängt, spricht man auch von einem impliziten Einschrittverfahren.

Ein explizites Einschrittverfahren ist dann durch die Vorschrift

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \Phi(t_i, \mathbf{x}_i; h_i), \quad i = 0, \dots, N-1$$

charakterisiert.

3.4 Differenzgleichung und Differentialgleichung - Einschrittverfahren

Wir wollen hier erst einmal ein explizites Einschrittverfahren voraussetzen.

Sei nun für eine gegebene Funktion \mathbf{f} die zugehörige Integralkurve durch $\{(t, \mathbf{x}(t)) : t \in J\}$ gegeben.

Damit gilt dann die Beziehung

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t))$$

Wir verfolgen nun die Integralkurve auf ihrem Weg von t nach $t+h$ - sie verläuft von $(t, \mathbf{x}(t))$ nach $(t+h, \mathbf{x}(t+h))$.

Nun wird das Einschrittverfahren

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \Phi(t_i, \mathbf{x}_i; h_i), \quad i = 0, \dots, N-1$$

die Differentialgleichung

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t))$$

normalerweise nicht exakt integrieren - es führt zu dem etwas abweichenden Punkt

$$(t+h, \mathbf{x}^h(t+h))$$

Wir nehmen nun an, dass an der Stelle $t = t_i$ die exakte Lösung der Differentialgleichung bekannt ist - es soll somit $\mathbf{x}_i = \mathbf{x}(t_i)$ gelten.

Dann lässt sich die mit dem Einschrittverfahren

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \Phi(t_i, \mathbf{x}_i; h_i), \quad i = 0, \dots, N-1$$

und der Schrittweite $h = h_i$ berechnete Näherung $\mathbf{x}^h(t+h)$ auch wie folgt darstellen

$$\mathbf{x}^h(t+h) = \mathbf{x}(t) + h\Phi(t, \mathbf{x}(t); h)$$

Nun ist das Diskretisierungsverfahren durch die Inkrementfunktion Φ charakterisiert.

Für die Konstruktion eines geeigneten Diskretisierungsverfahrens reicht die Forderung

$$\mathbf{x}^h(t+h) - \mathbf{x}(t+h) \rightarrow 0 \quad \text{für } h \rightarrow 0$$

nicht aus.

Dies liegt daran, dass die Richtungen der beiden Vektoren

$$\begin{pmatrix} \mathbf{x}(t+h) - \mathbf{x}(t) \\ h \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} \mathbf{x}^h(t+h) - \mathbf{x}(t) \\ h \end{pmatrix}$$

für $h \rightarrow 0$ voneinander verschieden bleiben können.

Damit könnte sich der Anstieg der numerisch berechneten Lösung vom Anstieg der exakten Lösung unterscheiden, selbst für $h \rightarrow 0$.

Deshalb stellen die entscheidenden Größen

$$\frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} \quad \text{und} \quad \frac{\mathbf{x}^h(t+h) - \mathbf{x}(t)}{h}$$

dar, die direkter Bestandteil des sogenannten lokalen Diskretisierungsfehlers sind:

Definition 7

Es sei $\mathbf{x}^h(t+h)$ das Resultat eines mit dem Einschrittverfahren

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i \Phi(t_i, \mathbf{x}_i; h_i), \quad i = 0, \dots, N-1$$

ausgeführten Schrittes, wobei ein auf der Lösungskurve $\mathbf{x}(t)$ des Anfangswertproblems

$$\begin{aligned}\mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0\end{aligned}$$

gelegener Startvektor \mathbf{x}_i verwendet wird. Dann nennt man

$$\delta(t+h, \mathbf{x}(t+h); h) = \frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} - \frac{\mathbf{x}^h(t+h) - \mathbf{x}(t)}{h} = \frac{\mathbf{x}(t+h) - \mathbf{x}^h(t+h)}{h}$$

den lokalen Diskretisierungsfehler des Einschrittverfahrens an der Stelle $t+h$, wobei $t \in J'_h = J_h - \{t_N\}$ ist.

Mit Hilfe des lokalen Diskretisierungsfehlers können wir nun die wesentliche Anforderung an ein Einschrittverfahren definieren - die sogenannte Konsistenz:

Definition 8

Sei wie bisher $\mathbf{x}(t)$ die exakte Lösung des Anfangswertproblems

$$\begin{aligned}\mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0\end{aligned}$$

auf dem Intervall $[x_0, x_0 + \alpha]$. Dann heißt das Einschrittverfahren konsistent mit dem Anfangswertproblem, wenn für jede Lipschitz-stetige Funktion \mathbf{f} und alle $t \in J'_h$ die Beziehung

$$\delta(t+h, \mathbf{x}(t+h); h) = O(h) \quad \text{für } h \rightarrow 0$$

gilt.

Aus der Formel

$$\delta(t+h, \mathbf{x}(t+h); h) = \frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h} - \frac{\mathbf{x}^h(t+h) - \mathbf{x}(t)}{h} = \frac{\mathbf{x}(t+h) - \mathbf{x}^h(t+h)}{h}$$

folgt, dass für $h \rightarrow 0$ der erste Summand gegen $\mathbf{f}(t, \mathbf{x}(t))$ konvergiert.

Ist somit das Verfahren konsistent, dann muss auch für den zweiten Term

$$\frac{\mathbf{x}^h(t+h) - \mathbf{x}(t)}{h} \rightarrow \mathbf{f}(t, \mathbf{x}(t)) \quad \text{für } h \rightarrow 0$$

gelten.

Damit erfüllt ein konsistentes Verfahren

$$\Phi(t, \mathbf{x}(t); 0) = \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J'_h$$

Das bedeutet, dass das numerische Ersatzproblem in das ursprüngliche Anfangswertproblem für $h \rightarrow 0$ übergeht.

Man kann den lokalen Diskretisierungsfehler auch etwas anders interpretieren.

Dazu substituiert man

$$\mathbf{x}^h(t+h) = \mathbf{x}(t) + h\Phi(t, \mathbf{x}(t); h)$$

in die Formel für den lokalen Diskretisierungsfehler und erhält:

$$\delta(t+h, \mathbf{x}(t+h); h) = \frac{1}{h} [\mathbf{x}(t+h) - \mathbf{x}(t) - h\Phi(t, \mathbf{x}(t); h)]$$

Damit stellt $h\delta(\cdot)$ das Residuum dar, welches sich ergibt, wenn die exakte Gitterfunktion $\{\mathbf{x}(t)\}$ anstelle der approximierenden Gitterfunktion $\{\mathbf{x}^h(t)\}$, $t \in J'_h$ in die Formel für das Einschrittverfahren eingesetzt wird.

Es verbleibt noch, implizite Einschrittverfahren

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i\Phi(t_i, \mathbf{x}_i, \mathbf{x}_{i+1}; h_i), \quad i = 0, \dots, N-1$$

zu betrachten.

Definition 9

Gegeben sei das implizite Einschrittverfahren

$$\mathbf{x}_{i+1} = \mathbf{x}_i + h_i\Phi(t_i, \mathbf{x}_i, \mathbf{x}_{i+1}; h_i), \quad i = 0, \dots, N-1$$

Derjenige Vektor $\delta(\cdot)$, der entsteht, wenn man die exakte Lösung in die Differenzengleichung einsetzt und das Resultat durch h dividiert, wird lokaler Diskretisierungsfehler genannt, d. h. für $t \in J'_h$ ist

$$\delta(t+h, \mathbf{x}(t+h); h) = \frac{1}{h} [\mathbf{x}(t+h) - \mathbf{x}(t) - h\Phi(t, \mathbf{x}(t); h)]$$

Um dann die Konsistenz zu definieren, hat man nur noch $\delta(\cdot)$ in die Konsistenzbedingung

$$\delta(t+h, \mathbf{x}(t+h); h) = O(h) \quad \text{für } h \rightarrow 0$$

einzusetzen.

3.5 Konsistenzordnung und globaler Fehler - Einschrittverfahren

Nun gibt es eine Vielzahl von numerischen Verfahren, um Differentialgleichungssysteme zu lösen - so gibt es beispielsweise ganze Familien von Runge-Kutta-Verfahren, siehe unter anderem das Buch von Hermann (2004), S. 20 ff.

Um diese miteinander vergleichen zu können, liegt es nahe, danach zu fragen, ob die Konsistenzbedingung

$$\delta(t+h, \mathbf{x}(t+h); h) = O(h) \quad \text{für } h \rightarrow 0$$

nicht auch mit einer größeren h -Potenz erfüllt werden kann - in diesem Fall reduziert sich der lokale Fehler bei Verkleinerung der Schrittweite schneller.

Wir definieren deshalb:

Definition 10

Ein Einschrittverfahren besitzt die Konsistenzordnung p , wenn für hinreichend glatte Lösungen $\mathbf{x}(t)$ des Anfangswertproblems

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

gilt

$$\delta(t_{i+1}, \mathbf{x}(t_{i+1}); h) = O(h^p)$$

wobei p die maximal mögliche positive ganze Zahl bezeichnet.

Da sich der lokale Fehler schneller reduziert, je größer die Konsistenzordnung ist, stellt sich die Frage, ob und inwieweit es möglich ist, Verfahren entsprechend hoher Konsistenzordnung zu konstruieren.

Da ein möglicher Weg zur Konstruktion derartiger Verfahren mit Hilfe von Taylor-Entwicklungen verläuft, ist es naheliegend, dass dies entsprechende Differenzierbarkeitsbedingungen an die Funktion \mathbf{f} des Anfangswertproblems

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

voraussetzt.

Ergebnisse dieses Typs können zum Beispiel im Buch von Hermann (2004) auf S. 35 ff. gefunden werden.

Ein Satz stellt unter entsprechenden Anforderungen an die Differenzierbarkeit von \mathbf{f} fest, dass im Fall der Runge-Kutta-Verfahren die Konsistenzordnung $p = 4$ erreicht werden kann - siehe hierzu Satz 2.3 auf S. 35 im Buch von Hermann (2004).

Weitere Ergebnisse - und die Sätze beinhalten auch Konstruktionsverfahren zur Berechnung der gesuchten Koeffizienten der zugehörigen Runge-Kutta-Verfahren - können in den Sätzen 2.7 auf S. 43 und 2.8 auf S. 45 gefunden werden.

Dabei wird die Konsistenzordnung nochmals erheblich erhöht.

Bisher hatten wir die numerische Integration nur über einem Intervall $[t_i, t_{i+1}]$ der Länge h betrachtet und den dort resultierenden lokalen Diskretisierungsfehler untersucht.

Dabei war der Ausgangspunkt, dass am Anfang des Intervalls der exakte Wert der Lösung $\mathbf{x}(t)$ des Anfangswertproblems vorliegt - dies ist natürlich nicht realistisch.

In Wirklichkeit ist es natürlich so, dass der Integrationsprozess sukzessive auf den auf dem Gitter J_h bestimmten Segmenten $[t_j, t_{j+1}]$ mit $j = 0, \dots, N-1$ stattfindet und dass man bereits am Ende des ersten Segments nur über eine Näherung \mathbf{x}_1^h für $\mathbf{x}(t_1)$ verfügt.

Demzufolge haben sich in der Approximation \mathbf{x}_{i+1}^h alle vorausgegangenen lokalen Fehler aufsummiert.

Somit müssen wir den an der Stelle t_{i+1} wie folgt definierten globalen Fehler studieren:

Definition 11

Es sei \mathbf{x}_{i+1}^h die mit einem numerischen Verfahren und der Schrittweite h erzeugte Näherung für die exakte Lösung $\mathbf{x}(t)$ des Anfangswertproblems

$$\begin{aligned}\mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0\end{aligned}$$

an der Stelle t_{i+1} . Dann ist der globale Fehler dieser Näherung zu

$$\mathbf{e}_{i+1}^h = \mathbf{x}(t_{i+1}) - \mathbf{x}_{i+1}^h$$

definiert.

Für den folgenden Satz definieren wir die nachstehende gestörte Differentialgleichung

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)) + \varepsilon(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{x}(t_0) = \mathbf{x}_0^h$$

mit

$$\varepsilon(t, \mathbf{y}(t)) = \Phi(t_i, \mathbf{x}_i^h, \mathbf{x}_{i+1}^h; h) - \mathbf{f}(t, \mathbf{y}) \quad \text{und} \quad t \in (t_i, t_{i+1})$$

Mit $\mathbf{y}(t_i) = \mathbf{x}_i^h$ folgt dann

$$\mathbf{y}(t_{i+1}) = \mathbf{x}_i^h + \int_{t_i}^{t_{i+1}} (\mathbf{f}(t, \mathbf{y}) + \varepsilon(t, \mathbf{y})) dt$$

$$\begin{aligned}
&= \mathbf{x}_i^h + \int_{t_i}^{t_{i+1}} \Phi(t, \mathbf{x}_i^h, \mathbf{x}_{i+1}^h; h) dt \\
&= \mathbf{x}_i^h + \Phi(t_i, \mathbf{x}_i^h, \mathbf{x}_{i+1}^h; h) \int_{t_i}^{t_{i+1}} dt \\
&= \mathbf{x}_i^h + h\Phi(t_i, \mathbf{x}_i^h, \mathbf{x}_{i+1}^h; h) = \mathbf{x}_{i+1}^h
\end{aligned}$$

Damit stimmt die exakte Lösung $\mathbf{y}(t)$ in den Gitterpunkten $t_i \in J_h$ mit \mathbf{x}_i^h überein.

Es gilt nun

Satz 5

Die Funktion \mathbf{f} sei Lipschitz-stetig mit einer Lipschitz-Konstanten L . Die Integrationsgrenze $T < \infty$ sei so gewählt, dass \mathbf{y} (wie in obigem gestörten Problem gewählt) in dem abgeschlossenen Gebiet Ω verbleibt.

Dann existiert eine Konstante c , so dass für alle i mit $t_0 + (i+1)h \leq T$ gilt

$$\| \mathbf{e}_{i+1}^h \| \leq c \max \| \delta(t_{j+1}, \mathbf{x}_j(t_{i+1}); h) \|$$

Insbesondere ist $\| \mathbf{e}_{i+1}^h \| = O(h^p)$, wobei p die Konsistenzordnung des verwendeten Integrationsverfahrens bezeichnet.

BEWEIS

Siehe Hermann (2004), S. 58, Satz 2.12.

Die Formel

$$\| \mathbf{e}_{i+1}^h \| \leq c \max \| \delta(t_{j+1}, \mathbf{x}_j(t_{i+1}); h) \|$$

legt nun folgende Definition nahe:

Definition 12

Ein Einschrittverfahren wird konvergent genannt, falls ein Gebiet Ω wie im vorhergehenden Satz existiert, so dass für ein festes $t = t_0 + (i+1)h \leq T$ gilt

$$\lim_{h \rightarrow 0, t \rightarrow \infty} \mathbf{e}_{i+1}^h = 0$$

Die größte positive ganze Zahl q , für die $\mathbf{e}_{i+1}^h = O(h^q)$ erfüllt ist, heißt Konvergenzordnung des Einschrittverfahrens.

Aus dem vorhergehenden Satz ergibt sich dann

Satz 6

Ein Einschrittverfahren ist konsistent mit der Ordnung p genau dann, wenn es konvergent mit der Ordnung p ist.

BEWEIS Siehe Hermann (2004), S. 59, Folgerung 2.1.

Anmerkung

Wir sollten noch darauf hinweisen, dass der vorletzte Satz auch für implizite Runge-Kutta-Verfahren gültig ist (siehe Hermann (2004), S. 59, Bemerkung 2.3).

3.6 Mehrschrittverfahren

Wir wollen nun die Betrachtungen der vorhergehenden Abschnitte auf die Klasse der Mehrschrittverfahren ausdehnen.

Wir hatten bereits gesehen, dass Diskretisierungen von Differentialgleichungen auch auf Mehrschrittverfahren führen können - einerseits lässt sich in der Integraldarstellung des Anfangswertproblems

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

nämlich

$$\mathbf{x}(t_{i+1}) = \mathbf{x}(t_{i-k+1}) + \int_{t_{i-k+1}}^{t_{i+1}} \mathbf{f}(\tau, \mathbf{x}(\tau)) d\tau$$

der Integrand durch eine interpolatorische Quadraturformel auf den Gitterpunkten $t_{i-k+1}, \dots, t_i, t_{i+1}$ approximieren.

Für $k = 2$ und einem äquidistanten Gitter mit der Schrittweite h führt dies auf die Simpsonregel

$$\mathbf{x}_{i+1}^h - \mathbf{x}_{i-1}^h = h \left[\frac{1}{3} \mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h) + \frac{4}{3} \mathbf{f}(t_i, \mathbf{x}_i^h) + \frac{1}{3} \mathbf{f}(t_{i-1}, \mathbf{x}_{i-1}^h) \right]$$

Andererseits kann man auch die ursprüngliche Differentialgleichung

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t))|_{t=t_{i+1}}$$

zum Ausgangspunkt nehmen und die Ableitung mittels einer numerischen Differentiationsformel approximieren - zum Beispiel durch

$$\mathbf{x}'(t_{i+1}) \approx \frac{1}{2h} [3\mathbf{x}(t_{i+1}) - 4\mathbf{x}(t_i) + \mathbf{x}(t_{i-1})]$$

Hieraus resultiert dann das Integrationsverfahren

$$\frac{3}{2}\mathbf{x}_{i+1}^h - 2\mathbf{x}_i^h + \frac{1}{2}\mathbf{x}_{i-1}^h = h\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

Beide so erhaltenen Verfahren gehören zur Klasse der linearen Mehrschrittverfahren:

Definition 13

Ein lineares Mehrschrittverfahren mit k Schritten zur Bestimmung einer Gitterfunktion $\mathbf{x}^h(t)$ für die Lösung $\mathbf{x}(t)$ des Anfangswertproblems

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

auf einem äquidistanten Gitter J_h ist durch die Vorgabe von k Startwerten

$$\mathbf{x}^h(t_j) = \mathbf{x}_j^h, \quad j = 0, 1, \dots, k-1$$

und einer Differenzengleichung

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

mit

$$\alpha_j, \beta_j \in \mathbb{R}, \quad \alpha_0 \neq 0 \quad \text{und} \quad |\alpha_k| + |\beta_k| > 0$$

definiert.

Hierzu sind folgende Anmerkungen zu machen:

- Die Verfahrensfunktion Φ ist hierbei gleich

$$\Phi(t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_{i+1}^h) = \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

Dies erklärt auch den Namen "linear". Die Verfahrensfunktion Φ hängt linear von den Funktionswerten $\mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$ ab.

- Die Forderung $\alpha_0 \neq 0$ sichert, dass die implizite Differenzgleichung

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

eine eindeutige Lösung besitzt.

- Die Forderung $|\alpha_k| + |\beta_k| > 0$ legt die Schrittzahl k eindeutig fest.
- Im Fall $\beta_0 = 0$ liegt ein explizites lineares Mehrschrittverfahren vor, d. h. zur Berechnung von \mathbf{x}_{i+1}^h braucht kein nichtlineares Gleichungssystem gelöst zu werden.

In der Praxis sind folgende Verfahrensklassen recht populär:

- Die Adams-Moulton-Formeln

$$\mathbf{x}_{i+1}^h = \mathbf{x}_i^h + h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

stellen implizite Integrationsformeln dar.

Zum Beispiel erhalten wir im Fall $k = 0$ die Formel

$$\mathbf{x}_{i+1}^h = \mathbf{x}_i^h + h \mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

siehe Hermann (2004), S. 89 oder Strehmel u. a. (2012), S. 103.

- Im Gegensatz dazu stehen die Integrationsformeln

$$\mathbf{x}_{i+1}^h = \mathbf{x}_i^h + h \sum_{j=1}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

Diese stellen explizite Integrationsformeln dar und werden Adams-Bashforth-Formeln genannt.

Zum Beispiel erhalten wir im Fall $k = 2$ die Formel

$$\mathbf{x}_{i+1}^h = \mathbf{x}_i^h + h \left(\frac{3}{2} \mathbf{f}(t_i, \mathbf{x}_i^h) - \frac{1}{2} \mathbf{f}(t_{i-1}, \mathbf{x}_{i-1}^h) \right)$$

siehe Hermann (2004), S. 90.

- Dann gibt es die Rückwärtigen Differenzen Formeln

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \beta_0 \mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

eine Klasse impliziter Integrationsformeln, die auch unter dem Namen “Backward Difference Formula “ oder BDF-Verfahren bekannt sind.

Zum Beispiel erhalten wir im Fall $k = 1$ und $\beta_0 = 1$ die Formel

$$\mathbf{x}_{i+1}^h - \mathbf{x}_i^h = h\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

siehe Hermann (2004), S. 91.

Unter den angegebenen Referenzen können auch weitere Formeln gefunden werden.

Wir können nun den Begriff des lokalen Fehlers bzw. lokalen Diskretisierungsfehlers definieren:

Definition 14

Der lokale Fehler des linearen Mehrschrittverfahrens

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

ist definiert zu

$$\hat{\delta}(t_{i+1}, \mathbf{x}(t_{i+1}); h) = \frac{1}{h} [\mathbf{x}(t_{i+1}) - \mathbf{x}_{i+1}^h]$$

wobei $\mathbf{x}(t_{i+1})$ die exakte Lösung und \mathbf{x}_{i+1}^h deren Approximation an der Stelle t_{i+1} bezeichnen.

Dabei setzen wir voraus, dass die Eingabedaten exakt gegeben sind.

Dies bedeutet:

Die Gleichung

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

wird als Berechnungsvorschrift für \mathbf{x}_{i+1}^h aufgefasst, bei der alle vorangegangenen Werte $\mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h$ korrekt vorgegeben sind.

Wir definieren nun einen Verschiebungsoperator E durch

$$Eu_i = u_{i+1}$$

wenn $\{u_i\}$ Lösung einer Differenzgleichung ist, und

$$E^k(\cdot) = E(E^{k-1}(\cdot))$$

Mit diesem Verschiebungsoperator lässt sich

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

kompakt in der Form schreiben

$$\left(\sum_{j=0}^k \alpha_j E^{k-j} \right) \mathbf{x}_{i-k+1}^h = h \left(\sum_{j=0}^k \beta_j E^{k-j} \right) \mathbf{f}(t_{i-k+1}, \mathbf{x}_{i-k+1}^h)$$

Definieren wir nun noch die charakteristische Polynome

$$\rho(\lambda) = \sum_{j=0}^k \alpha_j \lambda^{k-j}, \quad \sigma(\lambda) = \sum_{j=0}^k \beta_j \lambda^{k-j}$$

dann nimmt das lineare Mehrschrittverfahren die Gestalt an

$$\rho(E) \mathbf{x}_{i-k+1}^h = h \sigma(E) \mathbf{f}(t_{i-k+1}, \mathbf{x}_{i-k+1}^h)$$

Wir ordnen nun dem Mehrschrittverfahren den folgenden linearen Differenzenoperator zu:

$$\tilde{\delta}(t_{i+1}, \mathbf{x}(t_{i+1}; h) = \frac{1}{h} \sum_{j=0}^k (\alpha_j \mathbf{x}(t_{i-j+1}) - h \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}(t_{i-j+1})))$$

Hierbei sei $\mathbf{x}(t)$ eine differenzierbare Vektorfunktion, die auf einem Intervall, welches die Stützstellen $t_{i-k+1}, \dots, t_{i+1}$ enthält, definiert ist.

Der nachfolgende Satz stellt einen interessanten Zusammenhang zwischen $\tilde{\delta}(\cdot)$ und $\delta(\cdot)$ her:

Satz 7

Gegeben sei die Differentialgleichung

$$\mathbf{x}'(t) = \mathbf{f}(t, \mathbf{x}(t))$$

mit einer stetig differenzierbaren Funktion $\mathbf{f}(t, \mathbf{x})$ und ihrer Lösung $\mathbf{x}(t)$.

Dann gilt für den lokalen Fehler

$$\tilde{\delta}(t_{i+1}, \mathbf{x}(t_{i+1}; h) = \left(\alpha_0 \mathbf{Id} - h \beta_0 \frac{\partial \mathbf{f}(t_{i+1}, \boldsymbol{\eta})}{\partial \mathbf{x}} \right)^{-1} \delta(t_{i+1}, \mathbf{x}(t_{i+1}); h)$$

wobei $\boldsymbol{\eta} \in \mathbb{R}^n$ auf einem Segment liegt, dass $\mathbf{x}(t_{i+1})$ und \mathbf{x}_{i+1}^h verbindet.

BEWEIS

Siehe Hermann (2004), S. 92, Satz 3.1.

Dabei ist \mathbf{Id} die $n \times n$ Einheitsmatrix.

Damit besagt der Satz, dass im Wesentlichen $\tilde{\delta}(\cdot)$ und $\delta(\cdot)$ übereinstimmen.

Wir können damit definieren:

Definition 15

Unter dem lokalen Diskretisierungsfehler eines linearen Mehrschrittverfahrens wollen wir das Residuum $\delta(t_{i+1}, \mathbf{x}(t_{i+1}); h)$ verstehen, d. h.

$$\delta(t_{i+1}, \mathbf{x}(t_{i+1}); h) = \frac{1}{h} [\rho(E)\mathbf{x}(t_{i-k+1}) - h\sigma(E)\mathbf{f}(t_{i-k+1}, \mathbf{x}(t_{i-k+1}))]$$

Damit können wir dann die Konsistenz eines linearen Mehrschrittverfahrens definieren:

Definition 16

Ein lineares Mehrschrittverfahren

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

heißt konsistent mit dem Anfangswertproblem

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{f}(t, \mathbf{x}(t)), \quad t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned}$$

falls $\delta(\cdot) = O(h)$ für $h \rightarrow 0$ gilt.

Darüber hinaus wird die größtmögliche ganze Zahl $p \geq 1$, für die $\delta(\cdot) = O(h^p)$ erfüllt ist, Konsistenzordnung genannt.

Der folgende Satz gibt ein interessantes Kriterium für die Konsistenz:

Satz 8

Das lineare Mehrschrittverfahren ist genau dann konsistent, falls

$$\rho(1) = 0 \quad \text{und} \quad \dot{\rho}(1) = \sigma(1)$$

gilt.

BEWEIS

Siehe Hermann (2004), S. 93, Satz 3.2.

Wir wollen nun in Analogie zu den vorhergehenden Abschnitten die Konvergenz der linearen Mehrschrittverfahren zeigen.

Hierzu benötigen wir den Begriff der Wurzelstabilität.

Definition 17

Ein lineares Mehrschrittverfahren

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

heißt wurzelstabil (oder auch nullstabil bzw. D-stabil), wenn jede Lösung der zugehörigen linearen homogenen Differenzgleichung

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = \mathbf{0}$$

stabil im Sinne des folgenden Satzes ist.

Satz 9

Es sei \mathbf{A} eine reelle $k \times k$ -Matrix der Gestalt

$$\mathbf{A} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \dots & \dots & -\alpha_k \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

Dann ist die Lösung von

$$\begin{pmatrix} x_{i+1} \\ x_i \\ \vdots \\ \vdots \\ x_{i-k+2} \end{pmatrix} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \dots & \dots & -\alpha_k \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \dots \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_i \\ x_{i-1} \\ \vdots \\ \vdots \\ x_{i-k+1} \end{pmatrix}$$

genau dann stabil, falls alle Wurzeln λ_i des Polynoms

$$P(\lambda) = \lambda^k + \alpha_1 \lambda^{k-1} + \dots + \alpha_{k-1} \lambda + \alpha_k$$

die folgenden Bedingungen erfüllen:

- $|\lambda_i| \leq 1$, $i = 1, \dots, k$
- falls ein λ_j existiert mit $|\lambda_j| = 1$, dann ist λ_j eine einfache Wurzel des obigen Polynoms.

BEWEIS Siehe Hermann (2004), S. 104, Satz 3.6.

Hierzu definieren wir:

Sei eine Differenzgleichung der Form

$$x_{i+1} = G(x_i), \quad i = 0, 1, \dots$$

gegeben mit $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$, wobei der Startvektor x_0 bekannt sei.

Definition 18

Sei $G: \mathbb{R}^n \rightarrow \mathbb{R}^n$ gegeben. Dann heißt eine Lösung der obigen Differenzgleichung

- 1) stabil, wenn zu vorgegebenem $\varepsilon > 0$ ein $\delta > 0$ existiert, so dass für alle weiteren Lösungen $\{\hat{x}_i\}_{i=0}^{\infty}$ der obigen Differenzgleichung, welche $\|x_0 - \hat{x}_0\| \leq \delta$ erfüllen, gilt

$$\|x_i - \hat{x}_i\| \leq \varepsilon, \quad i = 1, 2, \dots$$

- 2) asymptotisch stabil, wenn zusätzlich gilt

$$\|x_i - \hat{x}_i\| \rightarrow 0 \quad \text{für} \quad i \rightarrow \infty$$

- 3) relativ stabil, wenn

$$\|x_i - \hat{x}_i\| \leq \varepsilon \|x_i\|, \quad i = 1, 2, \dots$$

Unter Beachtung dieses Satzes erhält man für die Wurzelstabilität eines linearen Mehrschrittverfahrens:

Satz 10

Das lineare Mehrschrittverfahren

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

ist wurzelstabil genau dann, wenn alle Wurzeln $\lambda_1, \dots, \lambda_k$ des Polynoms $\rho(\lambda)$ die Bedingungen

- $|\lambda_i| \leq 1$, $i = 1, \dots, k$, und
- jede Wurzel λ_j mit $|\lambda_j| = 1$ ist einfach

erfüllen.

BEWEIS

Siehe Hermann (2004), S. 106, Satz 3.7.

Nach diesen Vorbereitungen können wir die Konvergenz eines linearen Mehrschrittverfahrens definieren:

Definition 19

Ein lineares Mehrschrittverfahren wird konvergent genannt, falls

$$\lim_{h \rightarrow 0, i \rightarrow \infty} \mathbf{x}_i^h = \mathbf{x}(t)$$

für alle fixierten $t = t_0 + ih < T$ gilt.

Dabei wird vorausgesetzt, dass die Funktion $\mathbf{f}(t, \mathbf{x})$ des Anfangswertproblems Lipschitz-stetig ist, auf dem Intervall $[t_0, T]$ eine Lösung $\mathbf{x}(t)$ existiert und für alle Startwerte $\mathbf{x}_0^h, \dots, \mathbf{x}_{k-1}^h$ gilt

$$\| \mathbf{x}_j^h - \mathbf{x}(t_j) \| \rightarrow 0, \quad h \rightarrow 0, \quad j = 0, \dots, k-1$$

Wir können dann zeigen:

Satz 11

Ist das lineare Mehrschrittverfahren

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h \sum_{j=0}^k \beta_j \mathbf{f}(t_{i-j+1}, \mathbf{x}_{i-j+1}^h)$$

konvergent, dann ist es auch notwendigerweise

- wurzelstabil und
- konsistent, d. h. $\rho(1) = 0$ und $\dot{\rho}(1) = \sigma(1)$

BEWEIS

Siehe Hermann (2004), S. 110, Satz 3.8.

Die andere Richtung bedeutet, dass die Konsistenz und die Wurzelstabilität die Konvergenz implizieren - dies besagt der folgende, etwas längliche Satz:

Satz 12

Wir setzen voraus, dass das explizite lineare Mehrschrittverfahren

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = h \Phi(t_{i-k+1}, \dots, t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h; h)$$

wurzelstabil ist.

Des weiteren erfülle die Funktion Φ eine Lipschitzbedingung der Form

$$\| \Phi(t_1, \dots, t_k, \mathbf{u}_1, \dots, \mathbf{u}_k; h) - \Phi(t_1, \dots, t_k, \mathbf{v}_1, \dots, \mathbf{v}_k; h) \| \leq c \max_{1 \leq j \leq k} \| \mathbf{u}_j - \mathbf{v}_j \|_\infty$$

für alle $t_1, \dots, t_k \in [t_0, T]$, alle $\mathbf{u}_1, \dots, \mathbf{u}_k, \mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{R}^n$ und alle $h \geq 0$.

Die Konstante c sei von h unabhängig.

Des Weiteren werde vorausgesetzt, dass das Anfangswertproblem

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}), \quad t_0 \leq t \leq T, \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

eine Lösung $\mathbf{x}(t)$ auf $[t_0, T]$ besitzt.

Die Lösung der Differenzgleichung

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = h \Phi(t_{i-k+1}, \dots, t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h; h)$$

werde als Funktion von h dargestellt, d. h. $\mathbf{x}_i^h = \mathbf{x}_i(h)$, $i = 0, 1, \dots$

Dann existieren Konstanten c_1 und c_2 , die unabhängig von h sind, so dass gilt

$$\underbrace{\|\mathbf{x}(t_0 + jh) - \mathbf{x}_j(h)\|_\infty}_{\text{globaler Fehler}} \leq \underbrace{c_1 r(h)}_{\text{Startwerte}} + \underbrace{c_2 \delta(h)}_{\text{Verfahren}}, \quad j = k, k+1, \dots, \frac{T-t_0}{h}$$

Dabei sind $\delta(h)$ und $r(h)$ wie folgt erklärt:

$$\begin{aligned} \delta(h) &: \|\delta(t_{i+1}, \mathbf{x}(t_{i+1}); h)\|_\infty \leq \delta(h), \quad i = k-1, k, \dots \\ r(h) &: r(h) = \max_{0 \leq j \leq k-1} \|\mathbf{x}(t_0 + jh) - \mathbf{x}_j(h)\|_\infty \end{aligned}$$

mit $\mathbf{x}_0(h), \mathbf{x}_1(h), \dots, \mathbf{x}_{k-1}(h)$ als Anfangswerte von

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = h \Phi(t_{i-k+1}, \dots, t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h; h)$$

Gilt insbesondere

- $r(h) \rightarrow 0$ für $h \rightarrow 0$ und
- das Verfahren

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = h \Phi(t_{i-k+1}, \dots, t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h; h)$$

ist konsistent

dann besteht für jedes feste $t \in [t_0, T]$ die Beziehung

$$\lim_{h \rightarrow 0, j \rightarrow \infty} \mathbf{x}_j(h) = \mathbf{x}(t)$$

wobei der Grenzwert so zu bilden ist, dass $t = t_0 + jh$ fest bleibt.

BEWEIS

Siehe Hermann (2004), S. 112, Satz 3.9.

Es verbleibt noch die Frage nach der Konvergenzordnung eines linearen Mehrschrittverfahrens:

Satz 13

Unter den Voraussetzungen des vorhergehenden Satzes besitze die Differentialgleichung eine p -mal stetig differenzierbare Lösung.

Weiterhin sei das Verfahren

$$\alpha_0 \mathbf{x}_{i+1}^h + \alpha_1 \mathbf{x}_i^h + \dots + \alpha_k \mathbf{x}_{i-k+1}^h = h \Phi(t_{i-k+1}, \dots, t_i, \mathbf{x}_{i-k+1}^h, \dots, \mathbf{x}_i^h, h)$$

von der Konsistenzordnung p und $r(h) = O(h^p)$.

Dann erfüllt der globale Diskretisierungsfehler für $h \rightarrow 0$ und ih fest

$$\| \mathbf{x}(t_0 + ih) - \mathbf{x}_i(h) \| = O(h^p)$$

BEWEIS

Siehe Hermann (2004), S. 116, Satz 3.10.

Insbesondere kann man für bestimmte Verfahren aus der Klasse der BDF-Formeln die Konvergenz zeigen.

Hierzu benötigen wir sogenannte rückwärts genommene Differenzen:

Definition 20

Gegeben sei eine Folge $\{x_i\}_{i=0}^\infty$. Unter den zugehörigen rückwärts genommenen Differenzen $\nabla^0 x_i$ sollen die Ausdrücke

$$\nabla^0 x_i = x_i, \quad i = 0, 1, \dots$$

verstanden werden. Rückwärts genommene Differenzen höherer Ordnung $\nabla^j x_i$ werden rekursiv erklärt

$$\nabla^j x_i = \nabla^{j-1} x_i - \nabla^{j-1} x_{i-1}, \quad j \geq 1$$

Damit erhalten wir

$$\begin{aligned} \nabla^1 x_i &= \nabla^0 x_i - \nabla^0 x_{i-1} = x_i - x_{i-1} \\ \nabla^2 x_i &= \nabla^1 x_i - \nabla^1 x_{i-1} = x_i - x_{i-1} - x_{i-1} + x_{i-2} = x_i - 2x_{i-1} + x_{i-2} \\ &\vdots \\ \nabla^k x_i &= \sum_{j=0}^k (-1)^j \binom{k}{j} x_{i-j} \end{aligned}$$

Hiermit können die BDF-Formeln

$$\sum_{j=0}^k \alpha_j \mathbf{x}_{i-j+1}^h = h\beta_0 \mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

auch mit Hilfe der Rückwärts genommenen Differenzen geschrieben werden (siehe Hermann (2004), S. 165):

$$\sum_{j=0}^k \gamma_j \nabla^j \mathbf{x}_{i+1}^h = h\beta_0 \mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

Die noch frei wählbaren Koeffizienten γ_j können dann so bestimmt werden, dass eine möglichst hohe Konsistenzordnung entsteht.

Man kann dann zeigen (siehe Hermann (2004), S. 166 ff.), dass mit $\beta_0 = 1$ eine größtmögliche Konsistenzordnung einstellt, falls

$$\gamma_0 = 0, \quad \gamma_j = \frac{1}{j}, \quad j = 1, \dots, k$$

Damit erhält man für die BDF-Verfahren

$$\sum_{j=1}^k \frac{1}{j} \nabla^j \mathbf{x}_{i+1}^h = h\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

Man kann weiterhin zeigen, dass die Wurzelstabilität allerdings nur für $k \leq 6$ gegeben ist, nicht aber für $k > 6$ (siehe Hermann (2004), S. 168).

Damit erhalten wir den

Satz 14

Die BDF-Verfahren

$$\sum_{j=1}^k \frac{1}{j} \nabla^j \mathbf{x}_{i+1}^h = h\mathbf{f}(t_{i+1}, \mathbf{x}_{i+1}^h)$$

mit $k \leq 6$ besitzen die Konsistenzordnung $p = k$. Wegen der Wurzelstabilität sind diese Verfahren auch konvergent mit der Ordnung p .

BEWEIS

Siehe Hermann (2004), S. 168, Satz 4.9.

Explizit aufgelöste BDF-Formeln findet man im Buch von Strehmel u. a. (2012) auf S. 325 - hier ist eine Tabelle abgedruckt.

3.7 Steife Differentialgleichungen

Zur Zeit der Entstehung der digitalen Rechentechnik legte man die Annahme zugrunde, dass sich alle Anfangswertprobleme mit den damals bekannten einfachen Integrationstechniken wie beispielsweise den expliziten Runge-Kutta-Verfahren unter Verwendung einer geeigneten Schrittweitensteuerung in angemessener Zeit hinreichend genau berechnen lassen.

Dann entdeckte man sogenannte steife Differentialgleichungen - dies veränderte diese Auffassung doch sehr stark.

Derartige Differentialgleichungen treten in den Anwendungen recht häufig auf - Anwendungsfelder sind technische Systeme mit Zeitkonstanten, die sich um Größenordnungen unterscheiden.

Eine Ursache für die Steifheit eines Anfangswertproblems liegt in der Existenz von sehr unterschiedlichen Zeitskalen im jeweiligen Problem. Betrachten wir hierzu folgendes

Beispiel 10

Gegeben die beiden Anfangswertprobleme

$$\mathbf{y}'(t) = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \mathbf{y}(t) + \begin{pmatrix} 2 \sin(t) \\ 2(\cos(t) - \sin(t)) \end{pmatrix}, \quad \mathbf{y}(0) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

$$\mathbf{y}'(t) = \begin{pmatrix} -2 & 1 \\ 998 & -999 \end{pmatrix} \mathbf{y}(t) + \begin{pmatrix} 2 \sin(t) \\ 999(\cos(t) - \sin(t)) \end{pmatrix}, \quad \mathbf{y}(0) = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Wir bezeichnen die Koeffizientenmatrix des ersten Differentialgleichungssystems mit \mathbf{A}_1 und die des zweiten Differentialgleichungssystems mit \mathbf{A}_2 .

Man kann ausrechnen, dass beide Differentialgleichungssysteme die gleiche exakte Lösung besitzen, nämlich

$$\mathbf{y}(t) = 2 \exp(-t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$$

Löst man nun die beiden Differentialgleichungssysteme einmal mit einem Verfahren, welches auf die Lösung von steifen Differentialgleichungen spezialisiert ist, und einmal mit einem Verfahren, welches diese Eigenschaft nicht besitzt, dann erkennt man, dass das zweite Verfahren erheblich größeren Aufwand leisten muss, um das zweite DGL-System zu lösen.

Um dieses Verhalten zu verstehen, betrachten wir die allgemeine Lösung der zugehörigen Systeme - für das erste System ist diese gegeben durch

$$\mathbf{y}(t) = c_1 \exp(-t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} + c_2 \exp(-3t) \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$$

und für das zweite System durch

$$\mathbf{y}(t) = c_1 \exp(-t) \begin{pmatrix} 1 \\ 1 \end{pmatrix} + c_2 \exp(-1000t) \begin{pmatrix} 1 \\ -998 \end{pmatrix} + \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$$

mit $c_1, c_2 \in \mathbb{R}$.

Die Eigenwerte der Matrix \mathbf{A}_1 sind $\lambda_1 = -1$ und $\lambda_2 = -3$, die der Matrix \mathbf{A}_2 dagegen $\lambda_1 = -1$ und $\lambda_2 = -1000$.

Die Lösungen der Anfangswertprobleme sind durch $c_1 = 2$ und $c_2 = 0$ gegeben.

Damit enthält die allgemeine Lösung des zweiten Systems eine sehr schnell abklingende Lösungskomponente, nämlich

$$\exp(-1000t)$$

wogegen sich die allgemeine Lösung des ersten Systems relativ langsam ändert.

Hieraus folgt, dass ein explizites Verfahren bei dem zweiten System mit recht kleinen Schrittwerten rechnen muss - dies erklärt den hohen numerischen Aufwand.

Bei dem ersten System ist dies dagegen nicht der Fall.

Wir betrachten nun allgemein folgende Differentialgleichung mit einer konstanten reellen $n \times n$ Koeffizientenmatrix \mathbf{A}

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}(t)$$

Wir setzen voraus, dass die Matrix \mathbf{A} diagonalisierbar sei und $\Re(\lambda_i) < 0$ für alle Eigenwerte λ_i von \mathbf{A} gelte.

Die Lösung der Differentialgleichung kann man dann in der Form

$$\mathbf{x}(t) = \sum_{i=1}^n c_i \exp(\lambda_i t) \mathbf{v}^{(i)} + \mathbf{p}(t)$$

schreiben.

Da $\Re(\lambda_i) < 0$ für $i = 1, \dots, n$, geht der erste Summand - die allgemeine Lösung des homogenen Problems - für $t \rightarrow \infty$ gegen 0.

Damit haben wir, dass $\mathbf{x}(t) \rightarrow \mathbf{p}(t)$ für $t \rightarrow \infty$ - man nennt deshalb $\mathbf{p}(t)$ die stationäre Lösung des Problems.

Nun ist die Zielstellung bei einer derartigen Aufgabe die Bestimmung dieser stationären Lösung - hieraus folgt, dass die numerische Lösung solange verfolgt werden muss, bis der am langsamsten abklingende Term

$$c_i \exp(\lambda_i t) \mathbf{v}^{(i)}$$

vernachlässigbar klein ist.

Ist nun $|\Re(\lambda_i)|$ sehr klein, dann wird dieser Integrationsweg sehr lang.

Gilt somit

$$|\Re(\lambda_1)| \geq |\Re(\lambda_2)| \geq \dots \geq |\Re(\lambda_n)| \quad \text{und} \quad |\Re(\lambda_1)| \gg |\Re(\lambda_n)|$$

dann muss extrem lange mit einer recht kleinen Schrittweite gerechnet werden.

Dies legt folgende Definition nahe:

Definition 21

Die lineare inhomogene Differentialgleichung

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}(t)$$

heißt steif, falls gilt:

- $\Re(\lambda_i) < 0$ für alle Eigenwerte λ_i der reellen $n \times n$ Matrix \mathbf{A}
- $\max_{i=1,\dots,n} |\Re(\lambda_i)| \gg \min_{i=1,\dots,n} |\Re(\lambda_i)|$

Man versucht nun, die Steifheit eines Problems zu quantifizieren:

Definition 22

Das Steifheitsmaß S der Differentialgleichung

$$\mathbf{x}'(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}(t)$$

ist der Quotient

$$S = \frac{\max_{i=1,\dots,n} |\Re(\lambda_i)|}{\min_{i=1,\dots,n} |\Re(\lambda_i)|}$$

Das Problem besteht nun darin, die Schrittweite so zu wählen, dass $z = h\lambda$ im sogenannten Bereich der absoluten Stabilität liegt - siehe hierzu zum Beispiel Hermann (2004), S. 138 ff.

Viel sinnvoller ist es jedoch, Integrationsverfahren zu verwenden, deren Gebiet der absoluten Stabilität die gesamte komplexe Halbebene umfasst.

Diese Bedingung erfüllen zum Beispiel die impliziten Runge-Kutta-Verfahren.

Die Frage stellt sich nun, wie man diese Ideen auf nichtlineare Differentialgleichungen übertragen kann.

Wir gehen von dem Problem

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad x : [t_0, t_e] \rightarrow \mathbb{R}^n$$

aus.

Wir studieren nun das lokale Verhalten der exakten Lösung $\mathbf{x}(t)$ in einer Umgebung \mathbf{x}_i^h - dies sei die Näherungslösung an der Stelle t_i unter der Anfangsbedingung $\mathbf{x}(t_i) = \mathbf{x}_i^h$.

Dazu setzen wir für $\mathbf{x}(t)$ an

$$\mathbf{x}(t) = \mathbf{x}_i^h + \mathbf{w}(t), \quad t_i \leq t \leq t_i + h$$

Wir setzen dabei voraus, dass sowohl die Schrittweite h wie auch die Norm des Vektors $\mathbf{w}(t) = (w_1(t), \dots, w_n(t))^T$ klein sind.

Nun substituieren wir diesen Ansatz in die Differentialgleichung und linearisieren die rechte Seite - dies ergibt

$$\begin{aligned} \dot{\mathbf{w}}_j(t) &= \mathbf{f}_j(t_i + (t - t_i), x_{i,1}^h + w_1(t), x_{i,2}^h + w_2(t), \dots, x_{i,n}^h + w_n(t)) \\ &\approx \mathbf{f}_j(t_i, x_i^h) + (t - t_i) \frac{\partial \mathbf{f}_j(t_i, \mathbf{x}_i^h)}{\partial t} + \sum_{k=1}^n \frac{\partial \mathbf{f}_j(t_i, \mathbf{x}_i^h)}{\partial x_k} w_k(t), \quad j = 1, \dots, n \end{aligned}$$

Die so erhaltenen n linearen homogenen Differentialgleichungen für die Funktionen $w_1(t), w_2(t), \dots, w_n(t)$ lassen sich mit den Größen

$$\mathbf{J}(t_i) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

$$\mathbf{w}(t) = \begin{pmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_n(t) \end{pmatrix}, \quad \mathbf{f}_i = \begin{pmatrix} f_1(t_i, x_i^h) \\ f_2(t_i, x_i^h) \\ \vdots \\ f_n(t_i, x_i^h) \end{pmatrix}, \quad \mathbf{g}_i = \begin{pmatrix} \frac{\partial f_1}{\partial t}(t_i, x_i^h) \\ \frac{\partial f_2}{\partial t}(t_i, x_i^h) \\ \vdots \\ \frac{\partial f_n}{\partial t}(t_i, x_i^h) \end{pmatrix}$$

zusammenfassen, so dass

$$\dot{\mathbf{w}}(t) = \mathbf{J}(t_i) \mathbf{w}(t) + \mathbf{f}_i + (t - t_i) \mathbf{g}_i, \quad \mathbf{w}(t_i) = \mathbf{0}$$

Nun können wir die Definition der Steifheit übertragen:

Definition 23

Die nichtlineare Differentialgleichung

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x} : [t_0, t_e] \rightarrow \mathbb{R}^n$$

heißt steif in einer Umgebung der Stelle $t = t_i$, falls gilt

- $\Re(\lambda_i) < 0$ für alle Eigenwerte λ_i der $n \times n$ Matrix $\mathbf{J}(t_i)$
- $\max_{i=1, \dots, n} |\Re(\lambda_i)| \gg \min_{i=1, \dots, n} |\Re(\lambda_i)|$

Das Steifheitsmaß ist dann der Quotient

$$S(t_i, x_i^h) = \frac{\max_{i=1, \dots, n} |\Re(\lambda_i)|}{\min_{i=1, \dots, n} |\Re(\lambda_i)|}$$

Nun gibt es leider Beispiele, bei denen das obige Steifheitsmaß keinerlei Steifheit signalisiert, dessen Lösung sich qualitativ gleichartig verhält (siehe hierzu Hermann (2004), S. 160, Beispiel 4.6).

Deshalb suchte man nach einer verbesserten Charakterisierung steifer Probleme - dazu benötigt man folgende Norm:

Definition 24

Sei \mathbf{A} eine quadratische Matrix. Die Größe

$$\mu[\mathbf{A}] = \lim_{h \rightarrow 0} \frac{\|\mathbf{Id} + h\mathbf{A}\| - 1}{h}$$

heißt die logarithmische Norm von \mathbf{A} .

Diese Norm hängt offensichtlich von der verwendeten Matrix-Norm $\|\mathbf{A}\|$ ab - im folgenden Satz geben wir für bekannte Matrix-Normen die zugehörigen logarithmischen Normen an:

Satz 15

Es sei \mathbf{A} eine quadratische reelle Matrix. Dann gilt

$$\begin{aligned} \|\mathbf{A}\|_2 &= \sqrt{\lambda_{\max}(\mathbf{A}^T \mathbf{A})} \Rightarrow \mu_2[\mathbf{A}] = \lambda_{\max}\left(\frac{1}{2}(\mathbf{A} + \mathbf{A}^T)\right) \\ \|\mathbf{A}\|_\infty &= \max_{i=1, \dots, n} \sum_{j=1}^n |a_{ij}| \Rightarrow \mu_\infty[\mathbf{A}] = \max_{i=1, \dots, n} \left(a_{ii} + \sum_{j=1, j \neq i}^n |a_{ij}| \right) \\ \|\mathbf{A}\|_1 &= \max_{j=1, \dots, n} \sum_{i=1}^n |a_{ij}| \Rightarrow \mu_1[\mathbf{A}] = \max_{j=1, \dots, n} \left(a_{jj} + \sum_{i=1, i \neq j}^n |a_{ij}| \right) \end{aligned}$$

wobei $\lambda_{\max}(\mathbf{B})$ den maximalen Eigenwert der Matrix \mathbf{B} bezeichnet.

BEWEIS

Siehe Hermann (2004), S. 161, Satz 4.7.

Hiermit kann man dann definieren:

Definition 25

Das Anfangswertproblem

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(t, \mathbf{x}(t)), & t \in J \\ \mathbf{x}(t_0) &= \mathbf{x}_0\end{aligned}$$

wird als steif bezeichnet, falls eine logarithmische Matrix-Norm existiert, so dass für $t_0 \leq t \leq t_N$ und $\mathbf{y} \in \mathbb{R}^n$ aus einer Umgebung der exakten Lösung $\mathbf{x}(t)$ für die Jacobi-Matrix $\mathbf{f}_{\mathbf{x}}(t, \mathbf{x})$ gilt

$$(t_N - t_0) \sup \mu[\mathbf{f}_{\mathbf{x}}(t, \mathbf{y})] = \mu_0 \ll (t_N - t_0) \sup \|\mathbf{f}_{\mathbf{x}}(t, \mathbf{y})\|$$

In der Praxis ist die Bestimmung einer geeigneten logarithmischen Norm im Falle nichtlinearer Differentialgleichungen recht schwierig - man ersetzt die obige Beziehung durch

$$(t_N - t_0) \sup \|\mathbf{f}_{\mathbf{x}}(t, \mathbf{y})\| \gg 1$$

oder aber (mit L die klassische Lipschitzkonstante) durch

$$L(t_N - t_0) \gg 1$$

In der Ballistik sind die Flugzeiten teilweise recht ordentlich - damit werden manche Probleme die Definition der Steifheit erfüllen.

Es ist sinnvoll, nach Verfahren Ausschau zu halten, die mit steifen Differentialgleichungssystemen fertig werden.

Derartige Verfahren sind zum Beispiel die BDF-Verfahren - siehe Hermann (2004), S. 164 ff.

Ebenso weist Hermann (2004) darauf hin, dass sich steife Differentialgleichungssysteme nicht mit expliziten Runge-Kutta-Verfahren lösen lassen, siehe Hermann (2004), S. 171 ff.

Wohl geht dies mit den impliziten Runge-Kutta-Verfahren, siehe Hermann (2004), S. 171 ff.

In diesem Zusammenhang sei der Hinweis auf die Programmiersprache Octave gestattet - sie weist explizit auf die Möglichkeit hin, steife Differentialgleichungssysteme standardmäßig zu lösen.

3.8 Zusammenfassung

Offensichtlich kann man für eine Reihe von numerischen Verfahren die Konvergenz zeigen - hieraus ergibt sich als Konsequenz, dass auf dem betrachteten Integrationsintervall $[t_0, T]$ die exakte Lösung prinzipiell beliebig genau von der numerischen Lösung angenähert werden kann, sofern man die Schrittweite (im Rahmen der numerischen Möglichkeiten) entsprechend klein wählt.

Für die Differentialgleichungssysteme, die im Umfeld der Ballistik auftreten, ist dies sicherlich von Wichtigkeit. Je genauer die Lösung des Differentialgleichungssystems angenähert werden kann, umso genauer ist auch die Bestimmung der Flugbahn.

Kapitel 4

Einige Ergebnisse

4.1 Einführende Bemerkungen

Die nachfolgenden Berechnungen wurden mit den Daten aus dem Produktkatalog von Lapua abgeglichen - hier sind ballistische Daten bis gegebenenfalls 1000 Meter aufgeführt.

Die notwendigen Formfaktoren wurden dem Buch von Litz (2011) entnommen.

Die Berechnungen wurden mit der Programmiersprache Octave durchgeführt - einer kostenfreien Version von Matlab.

Die Differentialgleichungssysteme wurden dabei mit LSODE durchgeführt - standardmäßig wurden hierbei steife Differentialgleichungssysteme betrachtet.

Damit die Betrachtungen des vorhergehenden Abschnitts Anwendung finden können, benötigen wir Differentialgleichungssysteme

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t))$$

bei denen \mathbf{f} entsprechend oft differenzierbar ist - unter derartigen Voraussetzungen haben wir dann die entsprechenden Konvergenzeigenschaften des numerischen Verfahrens zum Lösen des Differentialgleichungssystems auf dem betrachteten Integrationsintervall.

Das erklärt dann auch, weshalb sich die berechnete Lösung bis zur Fleckschussweite "vernünftig" verhält.

Allerdings setzt dies voraus, dass die Luftwiderstandsbeiwerte des Luftwiderstandsgesetzes $G7$ entsprechend oft differenzierbar sein müssen.

Da $G7$ in Form von diskreten Werten vorliegt, wurden diese mit Hilfe einer gebrochen rationalen Funktion dargestellt - hierbei wurde der gleiche Zähler wie Nennergrad gleich 11 vorausgesetzt.

Das sich hieraus ergebende nichtlineare Ausgleichsproblem wurde mit Hilfe von GNUPLOT gelöst.

Für entsprechende theoretische Hintergründe sei auf das Buch von Peelen (2014) verwiesen -

interessanterweise gelangt der Verfasser auch hier auf eine gebrochen rationale Funktion für die Luftwiderstandsbeiwerte.

4.2 Das Kaliber .223

Wir betrachten zuerst das Kaliber .223 und hier Scenar-Geschosse mit 69 Grains (gleich 4,5 Gramm).

Zuerst Geschosse mit dem Produkt-Code GB541 (zuerst die Geschwindigkeit, dann die kinetische Energie):

0m	100m	200m	300m	600m
830	744	663	588	392
1550	1244	989	777	345

und nun mit dem Produkt-Code GB544 (wiederum zuerst die Geschwindigkeit, dann die kinetische Energie):

0m	100m	200m	300m	600m
850	805	730	659	472
1626	1307	1042	821	369

Im Fall des Produkt-Codes GB541 ergeben sich folgende Flughöhen (nach Herstellerangaben):

	100m	200m	300m	600m
100m	0	-129	-469	-3445
300m	156	184	0	-2506
600m	574	1020	1253	0

und berechnet:

	100m	200m	300m	600m
100m	0	-128	-467	-3409
300m	156	183	0	-2476
600m	568	1008	1238	0

Die erste Spalte beinhaltet dabei die Fleckschussweiten.

Im Fall des Produkt-Codes GB544 ergeben sich folgende Flughöhen (nach Herstellerangaben):

	100m	200m	300m	600m
100m	0	-138	-501	-3665
300m	167	196	0	-2661
600m	611	1083	1332	0

und berechnet:

	100m	200m	300m	600m
100m	0	-137	-495	-3618
300m	165	193	0	-2627
600m	603	1069	1313	0

Wiederum beinhaltet die erste Spalte die Fleckschussweiten.

Der Formfaktor ist nach Litz (2011), Seite 343, gleich 1,164.

4.3 Das Kaliber .308

Die Munition mit dem Produkt-Code GB491 hat folgende ballistische Daten (zuerst die Geschwindigkeit, dann die kinetische Energie):

0m	100m	200m	300m	600m	800	1000m
860	794	730	669	503	408	380
3698	3151	2663	2235	1264	833	548

Das Geschossgewicht beträgt hier 155 Grains oder 10 Gramm.

Für dieses Produkt ergeben sich folgende Flughöhen (nach Herstellerangaben):

	100m	200m	300m	600m	800m	1000m
100m	0	-115	-414	-2813	-6265	-12106
300m	138	161	0	-1984	-5160	-10726
600m	469	822	992	0	-2515	-7418

und berechnet:

	100m	200m	300m	600m	800m	1000m
100m	0	-114	-408	-2722	-5978	-11386
300m	136	158	0	-1906	-4889	-10024
600m	454	793	953	0	-2348	-6848

Wiederum beinhaltet die erste Spalte die Fleckschussweiten.

Der Formfaktor wurde dem Buch von Litz (2011), Seite 527, entnommen zu 0,988.

Man kann hier auf die Idee kommen, ob nicht vielleicht der Formfaktor optimiert werden könnte - in der Tat ergibt sich nach einigem Probieren mit dem Formfaktor 1,06 folgende Tabelle:

	100m	200m	300m	600m	800m	1000m
100m	0	-116	-416	-2818	-6278	-12176
300m	139	161	0	-1986	-5168	-10789
600m	470	823	993	0	-2520	-7479

Es liegt nahe, ein Schätzverfahren zu entwickeln für einen optimalen Formfaktor - die Daten sprechen an sich dafür.

Wir betrachteten weiterhin das Produkt D46 mit den folgenden ballistischen Daten (zuerst die Geschwindigkeit, dann die kinetische Energie):

0m	100m	200m	300m	600m	800
760	704	650	597	452	369
3466	2974	2533	2141	1228	815

Das Geschossgewicht beträgt hier 185 Grains oder 12 Gramm.

Für dieses Produkt ergeben sich folgende Flughöhen (nach Herstellerangaben):

	100m	200m	300m	600m	800m
100m	0	-157	-547	-3588	-7908
300m	182	207	0	-2495	-6450
600m	598	1039	1247	0	-3123

und berechnet:

	100m	200m	300m	600m	800m
100m	0	-157	-541	-3495	-7589
300m	180	204	0	-2412	-6146
600m	582	1008	1206	0	-2930

Wiederum beinhaltet die erste Spalte die Fleckschussweiten.

Der Formfaktor wurde dem Buch von Litz (2011), Seite 530, entnommen zu 1,081.

Auch hier kann man auf die Idee kommen, dass vielleicht der Formfaktor optimiert werden kann - in der Tat ergibt sich nach ein wenig Probieren mit dem Formfaktor 1,15 folgende Tabelle:

	100m	200m	300m	600m	800m
100m	0	-158	-550	-3597	-7904
300m	183	208	0	-2497	-6437
600m	599	1040	1249	0	-3108

Da auch in diesem Fall eine Verbesserung bzgl. des Formfaktors möglich war, liegt es nahe, ob nicht sogar ein Schätzverfahren denkbar wäre, mit welchem neben dem Formfaktor auch noch weitere Koeffizienten wie zum Beispiel im Fall des Magnus-Effektes geschätzt werden könnten.

Bei einem geschickten Versuchsaufbau sollte dies eigentlich mit relativ geringem Aufwand machbar sein - ein paar Schachteln Munition, ein Schießstand von 1000 oder 1500 Metern Länge, eine sehr genaue Waffe mit einem sehr guten Zielfernrohr, einige windstille Tage...

Die Konsequenzen wären weitreichend - in so einem Fall könnte auch die seitliche Abweichung durch den Drall mit berechnet werden.

4.4 Das Kaliber .338 Lapua Magnum

Die Munition mit dem Produkt-Code GB528 hat folgende ballistische Daten (zuerst die Geschwindigkeit, dann die kinetische Energie):

0m	100m	200m	300m	600m	800	1000m
830	789	750	712	605	539	478
6696	6054	5465	4925	3558	2827	2223

Das Geschossgewicht beträgt hier 300 Grains oder 19,4 Gramm.

Für dieses Produkt ergeben sich folgende Flughöhen (nach Herstellerangaben):

	100m	200m	300m	600m	800m	1000m
100m	0	-117	-413	-2583	-5231	-9289
300m	136	155	0	-1710	-4146	-7932
600m	421	725	855	0	-1865	-5082

und berechnet:

	100m	200m	300m	600m	800m	1000m
100m	0	-117	-407	-2508	-5182	-9179
300m	136	154	0	-1695	-4098	-7824
600m	418	719	848	0	-1838	-4999

Auch hier beinhaltet die erste Spalte die Fleckschussweiten.

Der Formfaktor wurde dem Buch von Litz (2011), Seite 557, entnommen zu 0,956.

Es ist auch hier sehr wahrscheinlich, dass Verbesserungen bzgl. des Formfaktors möglich sind - hier die Tabelle nach einigem Probieren für den Formfaktor 0,98.

	100m	200m	300m	600m	800m	1000m
100m	0	-117	-408	-2526	-5231	-9291
300m	136	155	0	-1709	-4143	-7931
600m	421	725	855	0	-1864	-5082

4.5 Erweiterungen

Der bisherige Programmcode kann relativ leicht erweitert werden:

- Im Fall des Vakuum-Modells wurde bereits ein Geländewinkel implementiert. Ist somit ein Anstieg des Geländes vorhanden (zum Beispiel bietet Nightforce die Möglichkeit, einen derartigen Winkel zu bestimmen, manche Laserentfernungsmesser bieten eine derartige Möglichkeit aber ebenfalls an), dann kann man den Geländewinkel in die Flughöhen einfließen lassen. Dies ist für die Einstellung des Zielfernrohrs von Bedeutung.

Mathematisch bedeutet dies nur eine Rotation des Koordinatensystems.

- Im Buch von Litz (2011) ist eine Formel angegeben, mit der die seitliche Abweichung des Geschosses geschätzt werden kann, gegeben unter anderem die Dralllänge. Stichwort: spin drift.

Dabei ist bei einem Rechtsdrall mit einer Abweichung nach rechts, im anderen Fall mit einer Abweichung nach links zu rechnen.

Dies kann man natürlich auch mit einem 6DOF-Modell berechnen, dazu müsste man allerdings entsprechende Koeffizienten kennen, zum Beispiel vom Magnus-Effekt.

Bei der angegebenen Formel ist dies nicht notwendig. Interessant ist eine derartige Korrektur insbesondere, falls Schüsse auf Entfernungen auf 900 und mehr Meter notwendig sind - hier können seitliche Abweichungen von 20 und mehr Zentimetern auftreten, die über das Zielfernrohr korrigiert werden müssen.

- Natürlich ist es möglich, auch die Abhängigkeit der Luftdichte von der Flughöhe mit einzuberechnen. Dies wäre nur noch die Abänderung einer Zeile Programmcode.
- Und natürlich wäre ein zusätzlicher Programm-Code interessant, mit dem der Formfaktor wie auch andere Faktoren wie zum Beispiel beim Magnus-Effekt oder dem Lift (aus Sicht des Statistikers) sauber geschätzt werden können.

Literaturverzeichnis

- [Ascher u. a. 1988] ASCHER, U. M. ; MATTHEIJ, R. M. M. ; RUSSEL, R. D.: *Numerical Solution of Boundary Value Problems of Ordinary Differential Equations*. SIAM, 1988
- [Hermann 2004] HERMANN, M.: *Numerik gewöhnlicher Differentialgleichungen*. Oldenbourg Verlag, 2004
- [Kamke 1969] KAMKE, E.: *Differentialgleichungen*. Akademische Verlagsgesellschaft, 1969
- [Litz 2011] LITZ, Bryan: *Applied Ballistics for Long-Range Shooting*. Applied Ballistics, LLC, 2011
- [McCoy 1999] MCCOY, R. L.: *Modern Exterior Ballistics*. Schiffer Military History, 1999
- [de Mestre 1990] MESTRE, N. de: *The Mathematics of Projectiles in Sport*. Cambridge University Press, 1990
- [Peelen 2014] PEELEN, J.: *Ballistik kleiner Kaliber: der Luftwiderstand*. Verlagshaus Monsenstein und Vannerdat OHG Münster, 2014
- [Quarteroni u. a. 2014] QUARTERONI, A. ; SALERI, F. ; GERVASIO, P.: *Scientific Computing with MATLAB and Octave*. Springer, 2014
- [Shampine u. a. 2003] SHAMPINE, L. F. ; GLADWELL, I. ; THOMPSON, S.: *Solving ODEs with MATLAB*. Cambridge University Press, 2003
- [Strehmel u. a. 2012] STREHMEL, K. ; WEINER, R. ; PODHAISKY, H.: *Numerik gewöhnlicher Differentialgleichungen*. Springer, 2012

Index

äquidistant, 25

Adams-Bashforth-Formeln, 38

Adams-Moulton-Formeln, 38

asymptotisch stabil, 43

BDF-Verfahren, 39

BFGS-Methode, 16

D-stabil, 42

Differenzgleichung, 24

Diskretisierung, 26

Diskretisierungsverfahren, 26

Einschrittverfahren, 29

fminunc, 16

fsolve, 15

Gitter, 25

Gitterfunktion, 26

Gitterpunkte, 25

Knoten, 25

konsistent, 31, 32, 41

Konsistenzordnung, 33, 41

konvergent, 35

Konvergenzordnung, 35

logarithmische Norm, 52

lokaler Diskretisierungsfehler, 31, 32, 41

mehrdimensionales Newton-Verfahren, 15

nullstabil, 42

Rückwärtigen Differenzen Formeln, 38

relativ stabil, 43

Schrittweite, 25

stabil, 43

steif, 50, 52

Steifheitsmaß, 50

wurzelstabil, 42

