Universität Bern

Bachelor-Thesis

Physik

Bestimmung von Ort und Geschwindigkeit passiver Raumobjekte durch Dopplerdaten daran reflektierter elektromagnetischer Wellen

Autor:
Daniel Zahnd

Arbeitsleiter:
Prof. Dr. Thomas Schildknecht

5. August 2022

Zusammenfassung

Die Umlaufbahnen der Erde werden mit jeder Weltraummission zunehmend durch Raumschrott bevölkert. Diese Objekte gefährden neue Raumfahrtprojekte, da solche Raumobjekte in der Regel sehr hohe Geschwindigkeiten der Grössenordnung 10^3 $\mathrm{m\,s^{-1}}$ aufweisen und damit verheerenden Schaden an Raumfähren, Satelliten oder Raketen verursachen können. Deswegen ist die Kenntnis der Bahnelemente von möglichst allen Raumschrottteilen in den Erdumlaufbahnen von allergrösster Wichtigkeit für jedes Raumfahrtprojekt. Die vorliegende Arbeit untersucht zwei kostengünstige Verfahren, die zur Bestimmung von Bahnelementen von Raumobjekten verwendet werden können. Kostengünstig ist die Technik, weil die Bestimmung der Bahnelemente durch Dopplerdaten von an Raumobjekten reflektierten elektromagnetischen Wellen erfolgt, wozu ausschliesslich ein hochenergetischer Radiowellensender, ein bis sechs Radiowellenempfänger sowie ein Rechner zur Auswertung der Daten erforderlich sind. Solche Einrichtungen bestehen vielerorts bereits, weswegen die in vorliegender Arbeit beschriebenen Verfahren lediglich an die Geometrien der bestehenden Anlagen angepasst werden müssen - darin bestehen hinsichtlich der aufzuwendenden finanziellen Mittel die grossen Vorteile der beschriebenen Methoden.

Abstract

With every new space mission, the earth orbits are increasingly populated with space debris. These objects endanger new space flight projects, since such objects typically have large velocities of order 10^3 $\mathrm{m\,s^{-1}}$ and therefore can have devastating effects on a space flight device upon impact. Hence, knowledge of the orbital elements of space debris is of utmost importance for every space flight project. The thesis at hand examines two cost-efficient methods, which can be used in order to determine orbital elements of space debris. These methods are cost-efficient, because the determination of orbital elements takes place using doppler data of electromagnetic waves, which are reflected on space debris objects. The only requirements therefore are a high-energy radiowave emitter, one to six radiowave receivers and a computer to evaluate the doppler data. Such facilities already exist in many places, hence the methods described in this thesis need only be adapted to the geometries of these existing facilities - herein the big advantages of the described methods with regard to cost-efficiency are motivated.

Inhaltsverzeichnis

1 Einleitung

Die Umlaufbahnen der Erde werden mit jeder Weltraummission zunehmend durch Raumschrott bevölkert. Diese Objekte gefährden neue Raumfahrtprojekte, da solche Raumobjekte in der Regel sehr hohe Geschwindigkeiten der Grössenordnung $10^3\ \mathrm{m\,s^{-1}}$ aufweisen und damit verheerenden Schaden an Raumfähren, Satelliten oder Raketen verursachen können. Deswegen ist die Kenntnis der Bahnelemente von möglichst allen Raumschrottteilen in den Erdumlaufbahnen von allergrösster Wichtigkeit für jedes Raumfahrtprojekt. Die Bestimmung von Raumschrott-Bahnelementen ist aber technisch anspruchsvoll, da es sich um passive Objekte handelt - also Objekte, die keine Signale aussenden können. Deswegen ist dieses Unterfangen sehr aufwendig und kostenintensiv. Die vorliegende Arbeit untersucht daher zwei kostengünstige Verfahren, die zur Bestimmung von Bahnelementen von Raumobjekten verwendet werden könnten. Kostengünstig ist die Technik, weil die Bestimmung der Bahnelemente durch Dopplerdaten von an Raumobjekten reflektierten elektromagnetischen Wellen erfolgt, wozu ausschliesslich ein hochenergetischer Radiowellensender, ein bis sechs Radiowellenempfänger sowie ein Rechner zur Auswertung der Daten erforderlich sind. Solche Einrichtungen bestehen vielerorts bereits, weswegen die in vorliegender Arbeit beschriebenen Verfahren lediglich an die Geometrien der bestehenden Anlagen angepasst werden müssen - darin bestehen hinsichtlich der aufzuwendenden finanziellen Mittel die grossen Vorteile der beschriebenen Verfahren.

Ein erstes in vorliegender Arbeit dargelegtes Verfahren basiert auf den Arbeiten von Paul B. Richards [**?**]. Hierbei handelt es sich um ein Verfahren mit einem Sender und einem Empfänger - diese müssen aber spezielle Geometrien aufweisen, wie in der Beschreibung des Verfahrens deutlich wird.

Ein zweites in dieser Arbeit untersuchtes Verfahren ist durch die Arbeiten von Iman Shames et al. inspiriert [**?**]. Dabei handelt es sich um ein Verfahren mit einem Sender und sechs Empfängern - diese müssen im Gegensatz zum ersten Verfahren aber keine speziellen Geometrien aufweisen, worin der Vorteil dieser Methode besteht.

Beide Verfahren bedienen sich des Dopplereffektes für elektromagnetische Wellen. Das Grundprinzip beider Verfahren ist es, aus gemessenen Dopplerverschiebungen von an Raumobjekten reflektierten elektromagnetischen Wellen Rückschlüsse auf deren Position und Geschwindigkeit zu gewinnen.

2 Methode 1: Ein Sender, ein Empfänger

2.1 Theoretische Herleitung der Methode

2.1.1 Trajektorie eines Objektes

Es sei durch $\vec{\rho}(t)$ die Trajektorie eines Objektes beschrieben. Die Geschwindigkeit $\vec{v}(t)$ des Objektes ist folglich durch

$$\vec{v}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\vec{\rho}(t) = \dot{\vec{\rho}}(t) \tag{2.1}$$

gegeben. Der Betrag der Trajektorie zu der Zeit t sei durch $\rho(t) \doteq |\vec{\rho}(t)|$ notiert, der Betrag der Geschwindigkeit zur Zeit t durch $v(t) \doteq |\vec{v}(t)|$. Die Zeitableitung des Betrages der Trajektorie sei geschrieben durch

$$\dot{\rho}(t) = \frac{\mathrm{d}}{\mathrm{d}t}|\vec{\rho}(t)| = \frac{\mathrm{d}}{\mathrm{d}t}\sqrt{\vec{\rho}(t)\cdot\vec{\rho}(t)} = \frac{\vec{\rho}(t)\cdot\dot{\vec{\rho}}(t)}{|\vec{\rho}(t)|} = \frac{\vec{\rho}(t)\cdot\dot{\vec{\rho}}(t)}{\rho(t)}, \tag{2.2}$$

was nach Paul B. Richards als Range-Rate bezeichnet wird [**?**, S.1729]. Die Dimension von $\dot{\rho}(t)$ ist eine Geschwindigkeit, da $[\dot{\rho}(t)] = \mathrm{m\,s^{-1}}$ gilt. Wie aus Gleichung (2.2) ersichtlich ist, kann die Range-Rate positiv oder negativ sein. Die Range-Rate $\dot{\rho}(t)$ ist zu unterscheiden vom Betrag der Geschwindigkeit $|\dot{\vec{\rho}}(t)|$, da im Allgemeinen

$$\dot{\rho}(t) = \frac{\mathrm{d}}{\mathrm{d}t}|\vec{\rho}(t)| \neq |\frac{\mathrm{d}}{\mathrm{d}t}\vec{\rho}(t)| = |\dot{\vec{\rho}}(t)| \tag{2.3}$$

gilt.

Es sei ferner bemerkt, dass hier und fortan in diesem Dokument $c = 2.997\,924\,58\mathrm{e}8\,\mathrm{m\,s^{-1}}$ die Lichtgeschwindigkeit im Vakuum bezeichnet.

2.1.2 Dopplereffekt zur Ortung eines passiven Objektes im Raum

Mittels des Dopplereffekts für elektromagnetische Wellen soll ein passives Objekt im Raum geortet werden. Die Situation hierzu ist in Abbildung 1 visualisiert.

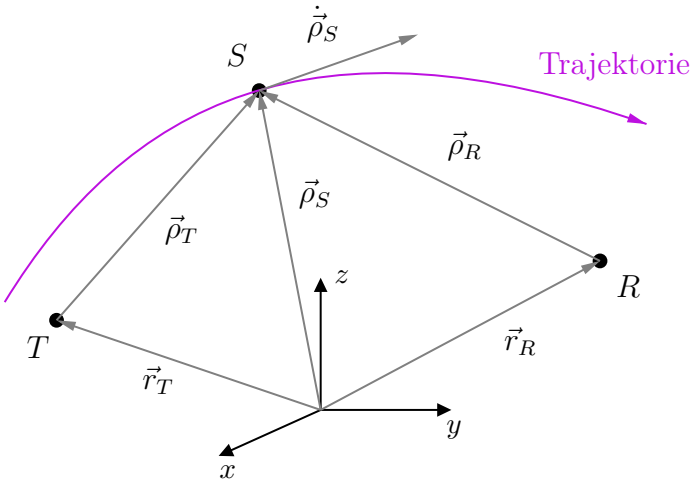


Abbildung 1: Skizze zur Situation betreffend der Ortung eines passiven Objektes (Subskript S) mittels des nichtrelativistischen Dopplereffektes.

Der Dopplereffekt wird nun für nichtrelativistische Geschwindigkeiten des Raumobjektes, beziehungsweise der Sender- und Empfängerstationen T und R - die über einen konstanten Abstand $\overline{TR} = \text{cst}$ verfügen - in Form der Range-Raten $\dot{\rho}_T$ und $\dot{\rho}_R$ formuliert. Wie der Abbildung 1 und den Definitionen der Grössen $\dot{\rho}_T$ und $\dot{\rho}_R$ zu entnehmen ist, geben $\dot{\rho}_T$ die Längenänderung des Ortsvektors $\vec{\rho}_T$ und $\dot{\rho}_R$ entsprechend die Längenänderung des Ortsvektors $\vec{\rho}_R$ an. Um nun den Dopplereffekt zu formulieren, verwendet man zweimal die Formel (5.43). Von der Transmitterstation T werde eine elektromagnetische sphärische Welle konstanter Frequenz f_T emittiert. Diese Frequenz wird am Raumobjekt S dopplerverschoben mit der Frequenz f_S empfangen und reflektiert. Die reflektierten Wellen werden wiederum von der Empfängerstation R dopplerverschoben mit der Frequenz f_R registriert. Für die Frequenzen f_S und f_R gelten nach Gleichung (5.43) die Ausdrücke

$$f_S = f_T \left(\frac{c - \dot{\rho}_T}{c} \right), \quad f_R = f_S \left(\frac{c - \dot{\rho}_R}{c} \right). \tag{2.4}$$

Berechnet man die Differenz $f_T - f_R$ unter Benutzung der Relationen (2.4), so folgt

$$\begin{aligned} f_T - f_R &= f_T - f_S \frac{c - \dot{\rho}_R}{c} = f_T - f_T \frac{c - \dot{\rho}_T}{c} \frac{c - \dot{\rho}_R}{c} && (2.5)\\ &= f_T - \frac{f_T}{c^2}(c^2 - c\dot{\rho}_R - c\dot{\rho}_T + \dot{\rho}_T\dot{\rho}_R) && (2.6)\\ &= \frac{f_T c^2 - f_T c^2 + f_T c\dot{\rho}_R + f_T c\dot{\rho}_T - f_T\dot{\rho}_T\dot{\rho}_R}{c^2} = \frac{c f_T(\dot{\rho}_T + \dot{\rho}_R) - f_T\dot{\rho}_T\dot{\rho}_R}{c^2} && (2.7)\\ &= \frac{f_T}{c}(\dot{\rho}_T + \dot{\rho}_R) - f_T \underbrace{\frac{\dot{\rho}_T}{c}\frac{\dot{\rho}_R}{c}}_{= \mathcal{O}[(\frac{\dot{\rho}}{c})^2]} \approx \frac{f_T}{c}(\dot{\rho}_T + \dot{\rho}_R), && (2.8) \end{aligned}$$

wobei die Approximation im letzten Schritt zulässig ist, wenn für die Range-Raten $\dot{\rho}_T \ll c$ und $\dot{\rho}_R \ll c$ gilt, weil dann

$$\dot{\rho} \ll c \quad \Leftrightarrow \quad \frac{\dot{\rho}}{c} \ll 1 \quad \Rightarrow \quad \left(\frac{\dot{\rho}}{c}\right)^2 \approx 0 \tag{2.9}$$

resultiert. Überdies wurden Terme der Ordnung $\mathcal{O}[(\dot{\rho}/c)^2]$ bereits in der Herleitung des nichtrelativistischen Dopplereffektes (5.43) aus dem allgemeinen relativistischen Dopplereffekt (5.41) gemäss Gleichung (5.42) vernachlässigt, somit müssen weitere solche - durch Umformungen entstandene - Terme auch vernachlässigt werden, wenn konsequent eine nichtrelativistische Approximation erster Ordnung in $\dot{\rho}/c$ gemacht werden soll. Somit gibt Richards die für die folgenden Betrachtungen grundlegende Gleichung

$$\dot{\rho}_T(t) + \dot{\rho}_R(t) = \frac{c}{f_T(t)}[f_T(t) - f_R(t)] \tag{2.10}$$

an [**?**, S.1729].

2.1.3 Zweikörperproblem

Zunächst sei das allgemeine Zweikörperproblem mathematisch hergeleitet. Das allgemeine Gravitationsgesetz lässt sich zu

$$M_i\ddot{\vec{x}}_i(t) = -GM_i \sum_{\substack{j=1 \\ j\neq i}}^{N} M_j \frac{\vec{x}_i(t) - \vec{x}_j(t)}{|\vec{x}_i(t) - \vec{x}_j(t)|^3}, \quad i \in \{1, \ldots, N\}, \quad N \in \mathbb{N} \tag{2.11}$$

angeben. Im Zweikörperproblem werden zwei Massen M_1 und M_2 betrachtet, die sich zur Zeit t an den Orten $\vec{x}_1(t)$ und $\vec{x}_2(t)$ befinden. Der zeitabhängige Relativvektor $\vec{r}(t)$ ist mitsamt seiner zweiten zeitlichen Ableitung hierbei durch

$$\vec{r}(t) = \vec{x}_1(t) - \vec{x}_2(t) \quad \Rightarrow \quad \ddot{\vec{r}}(t) = \ddot{\vec{x}}_1(t) - \ddot{\vec{x}}_2(t) \tag{2.12}$$

gegeben. Eine Visualisierung der geschilderten Situation ist in Abbildung 2 zu finden.

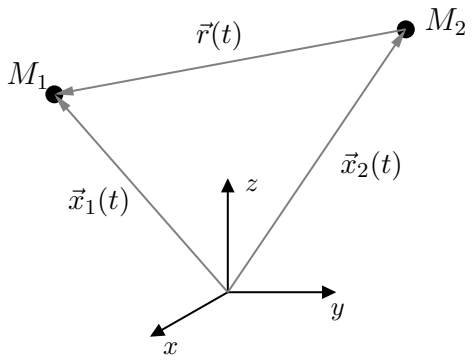


Abbildung 2: Skizze zum Zweikörperproblem.

Mit dem allgemeinen Gravitationsgesetz für zwei Massen M_1, M_2 folgt hiermit

$$\begin{aligned} \ddot{\vec{r}} = \ddot{\vec{x}}_1 - \ddot{\vec{x}}_2 = & -GM_2 \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3} - GM_1 \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3} \\ = & -\underbrace{G(M_1 + M_2)}_{\doteq \mu} \frac{\vec{x}_1 - \vec{x}_2}{|\vec{x}_1 - \vec{x}_2|^3} = -\mu \frac{\vec{r}}{|\vec{r}|^3}. \end{aligned} \tag{2.13}$$

Damit ist die allgemeine Formel für das Zweikörperproblem hergeleitet. Wird ein Raumobjekt mit einer in Relation zur Erdmasse vernachlässigbaren Masse von der Erde aus beobachtet und beschrieben, so ist als Grundlage der Bewegungsgleichungen vom Zweikörperproblem auszugehen. Dabei gelten $M_1 = M_E = 5.972\text{e}24\,\text{kg}$ und $M_2 \lll M_E$, sodass die Konstante μ zu $\mu = G(M_E + M_2) \approx GM_E$ angegeben werden kann.

2.1.4 Bezug der Ortskoordinaten zu den Dopplerdaten

Um die Positionen der Transmitter- und Empfängerstationen $\vec{\rho}_T(t)$ und $\vec{\rho}_R(t)$ auszudrücken, können aus einer Betrachtung der Abbildung 1 die zwei Möglichkeiten

$$\vec{\rho}_T(t) = \vec{\rho}_S(t) - \vec{r}_T(t), \quad \vec{\rho}_R(t) = \vec{\rho}_S(t) - \vec{r}_R(t) \tag{2.14}$$

abgelesen werden. Die drei Vektoren $\vec{r}_T(t), \vec{r}_R(t)$ und $\vec{\rho}_S(t)$ können komponentenweise zu

$$\vec{r}_T(t) = \begin{pmatrix} x_T(t) \\ y_T(t) \\ z_T(t) \end{pmatrix}, \quad \vec{r}_R(t) = \begin{pmatrix} x_R(t) \\ y_R(t) \\ z_R(t) \end{pmatrix}, \quad \vec{\rho}_S(t) = \begin{pmatrix} x_S(t) \\ y_S(t) \\ z_S(t) \end{pmatrix} \tag{2.15}$$

geschrieben werden. Über die Gleichung

$$\begin{aligned} \dot{\rho}_i = \frac{\vec{\rho}_i(t) \cdot \dot{\vec{\rho}}_i(t)}{|\vec{\rho}_i(t)|} = \frac{[\vec{\rho}_S(t) - \vec{r}_i(t)] \cdot [\dot{\vec{\rho}}_S(t) - \dot{\vec{r}}_i(t)]}{|\vec{\rho}_S(t) - \vec{r}_i(t)|} \\ = \frac{[x_S(t) - x_i(t)][\dot{x}_S(t) - \dot{x}_i(t)] + [(y_S(t) - y_i(t)][(\dot{y}_S(t) - \dot{y}_i(t)] + [z_S(t) - z_i(t)][\dot{z}_S(t) - \dot{z}_i(t)]}{([x_S(t) - x_i(t)]^2 + [y_S(t) - y_i(t)]^2 + [y_S(t) - y_i(t)]^2)^{1/2}} \end{aligned} \tag{2.16}$$

ist nach Richards für $i \in \{T, R\}$ der Bezug zu den Dopplerdaten eines Raumobjekts hergestellt - dies über die Definition der Range-Rate und der grundlegenden Dopplergleichung (2.10) [**?**, S.1729].

2.2 Berechnung einer ersten Bahnbestimmung

2.2.1 Parametrische Repräsentation der Dopplerkurve

Mittels den Definitionen $\lambda_T(t) \doteq {}^{c}/_{f_T(t)}$ und $\Delta f(t) \doteq f_T(t) - f_R(t)$ kann die Dopplergleichung (2.10) für die Range-Raten $\dot{\rho}_T(t)$ und $\dot{\rho}_R(t)$ durch

$$\dot{\rho}_T(t) + \dot{\rho}_R(t) = \lambda_T(t)\Delta f(t) \tag{2.17}$$

notiert werden. Die Senderfrequenz soll konstant und von null verschieden sein sein, also gelten $f_T = \text{cst} \neq 0$ und $\lambda_T = \text{cst} \neq 0$ - die Empfängerfrequenz f_R bleibt aufgrund des Dopplereffektes aber zeitabhängig. Eine Integration obiger Gleichung über das Zeitintervall $[t_0, t]$ liefert die Beträge der Ortsvektoren $\vec{\rho}_T$ und $\vec{\rho}_R$, denn es gilt

$$\int_{t'=t_0}^{t} [\dot{\rho}_T(t') + \dot{\rho}_R(t')]\,\mathrm{d}t' = \rho_T(t) + \rho_R(t) - [\rho_T(t_0) + \rho_R(t_0)] = \lambda_T \int_{t'=t_0}^{t} \Delta f(t')\,\mathrm{d}t'. \tag{2.18}$$

Richards stellt fest, dass die Summe der Absolutwerte der Vektoren $\vec{\rho}_T(t)$ und $\vec{\rho}_R(t)$ gleich der doppelten grossen Halbachse eines Rotationsellipsoides ist - mit der Sendestation T in einem Brennpunkt und der Empfängerstation R im anderen Brennpunkt des Ellipsoides [**?**, S.1730]. Mittels der Abbildung 3 ist nachvollziehbar, dass aufgrund der Ellipsoidgeometrie die Beziehung $2a(t) = \rho_T(t) + \rho_R(t)$ gilt, dass also das Raumobjekt zum Zeitpunkt t auf

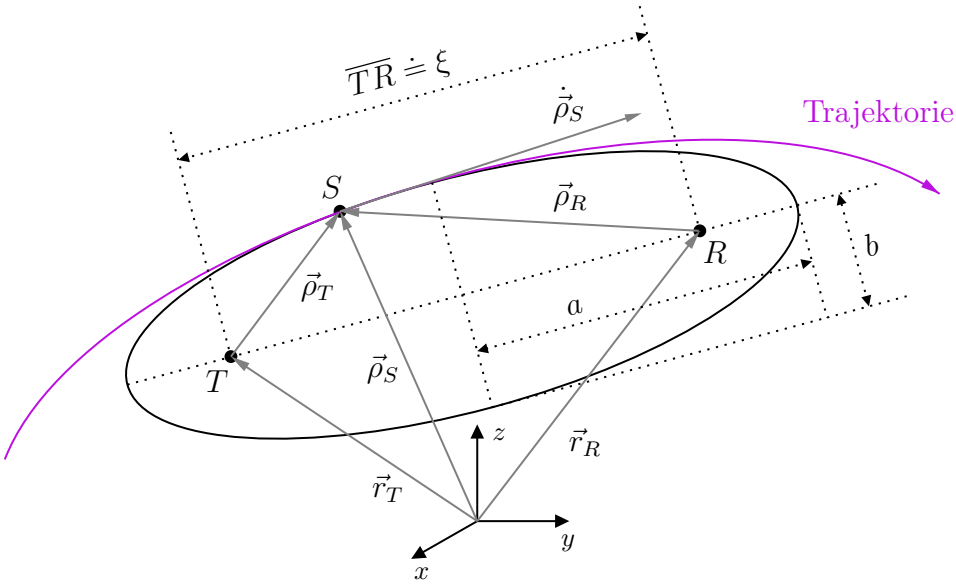


Abbildung 3: Skizze des Problems zu einer ersten Bahnbestimmung. Sämtliche Vektoren und Skalare sind im zeitabhängigen Sinne zu verstehen. Die Punkte T und R liegen in den Brennpunkten des gezeichneten schwarzen Ellipsoides mit grosser Halbachse a und kleiner Halbachse b. Die gezeichnete Ellipse ist als dreidimensionales Rotationsellipsoid zu verstehen, wobei die kleine Halbachse b überall orthogonal zur grossen Halbachse a ist.

einem Rotationsellipsoid mit grosser Halbachse $a(t)$ liegen muss, wobei die Sender- und Empfängerstationen in den zwei Brennpunkten des hierdurch definierten Rotationsellipsoides liegen. Das heisst, die Sender- und Empfängerstationen T und R liegen auf der Rotationsachse des gezeichneten Rotationsellipsoides. Somit gilt gemäss Gleichung (2.18)

$$\rho_T(t) + \rho_R(t) = \lambda_T \int_{t'=t_0}^{t} \Delta f(t')\,\mathrm{d}t' + [\rho_T(t_0) + \rho_R(t_0)] = 2a(t). \tag{2.19}$$

Aus der Kenntnis der grossen Halbachse $a(t)$ sowie dem konstanten Abstand $\overline{TR} = \xi$ beider Brennpunkte des durch $a(t)$ definierten Rotationsellipsoides kann die Exzentrizität $e(t)$ wiederum aufgrund ellipsoidgeometrischer Gesetze zu

$$e(t) = \frac{\xi}{2a(t)} \tag{2.20}$$

berechnet werden. Durch die Kenntnis des Terms $[\rho_T(t_0) + \rho_R(t_0)]$ und einer geeigneten Wahl der Zeit $t' \in [t_0, t]$ sind $\rho_T(t') + \rho_R(t')$ gegeben. Um einen Wert von $[\rho_T(t_0) + \rho_R(t_0)]$ zu ermitteln, kann nach Richards zunächst die Gleichung (2.17) durch

$$\frac{\vec{\rho}_T(t) \cdot \dot{\vec{\rho}}_T(t)}{\rho_T(t)} + \frac{\vec{\rho}_R(t) \cdot \dot{\vec{\rho}}_R(t)}{\rho_R(t)} = \lambda_T(t)\Delta f(t). \tag{2.21}$$

ausgedrückt werden [**?**, S.1730]. Im folgenden sei nun der Ursprung des in Abbildung 3 gezeigten rechtshändigen Koordinatensysstems topozentrisch in die Transmitterposition gelegt, mit der negativen z-Achse in Richtung des Empfängers zeigend, wie es in Abbildung 4 realisiert ist. Es gilt dann

$$\vec{r}_T = \vec{0} \quad \Leftrightarrow \quad \vec{\rho}_S = \vec{\rho}_T \quad \Rightarrow \quad \dot{\vec{\rho}}_S = \dot{\vec{\rho}}_T. \tag{2.22}$$

Ferner kann die Approximation

$$\dot{\vec{\rho}}_T = \dot{\vec{\rho}}_R = \dot{\vec{\rho}}_S \tag{2.23}$$

gemacht werden, falls die Distanz $\xi = \overline{TR}$ viel kleiner als die Flughöhe $x_{S_{min}}$ des Satelliten über der Erdoberfläche ist, also $\xi \ll x_{S_{min}}$ gilt. Satellitenflughöhen bewegen sich im Intervall $x_{S_{min}} \in [160\,\text{km}, 40\,000\,\text{km}]$, daher ist die Bedingung für obige Approximation bei einer Distanz $\xi \approx 1\,\text{km}$ des Senders vom Empfänger als erfüllt zu betrachten. Deshalb sei für die weitere Herleitung der Ortungsmethodik nach Richards von ebendieser Bedingung und der daraus folgenden Approximation ausgegangen. Hierdurch vereinfacht sich die obige Gleichung (2.21) zu

$$\dot{\vec{\rho}}_S(t) \cdot \left(\frac{\vec{\rho}_T(t)}{\rho_T(t)} + \frac{\vec{\rho}_R(t)}{\rho_R(t)} \right) = \lambda_T(t)\Delta f(t). \tag{2.24}$$

Nun sei Richards folgend der Vektor $\vec{p}(t)$ durch

$$\vec{p}(t) = \frac{1}{2} \left[\frac{\rho_T(t)\rho_R(t)}{1 - \left(\frac{\xi}{\rho_T(t) + \rho_R(t)} \right)^2} \right]^{\frac{1}{2}} \left(\frac{\vec{\rho}_T(t)}{\rho_T(t)} + \frac{\vec{\rho}_R(t)}{\rho_R(t)} \right) \tag{2.25}$$

definiert [**?**, S.1730]. Der Betrag $|\vec{p}(t)|$ ist dann gegeben durch

$$p(t) \doteq |\vec{p}(t)| = \sqrt{\frac{1}{4} \frac{\rho_T \rho_R}{1 - \frac{\xi^2}{[\rho_T + \rho_R]^2}} \left(\frac{\rho_T^2}{\rho_T^2} + 2\frac{\vec{\rho}_T \cdot \vec{\rho}_R}{\rho_T \rho_R} + \frac{\rho_R^2}{\rho_R^2} \right)} \tag{2.26}$$

$$= \sqrt{\frac{\rho_T \rho_R}{1 - \frac{\xi^2}{[\rho_T + \rho_R]^2}}} = \sqrt{\frac{\rho_T \rho_R}{(\rho_T + \rho_R)^2 - \xi^2} (\rho_T + \rho_R)^2} = \frac{1}{2}[\rho_T(t) + \rho_R(t)] = a(t). \tag{2.27}$$

Damit kann die Dopplergleichung (2.17) zu

$$\dot{p}(t) = \frac{\vec{p}(t) \cdot \dot{\vec{p}}(t)}{p(t)} = \lambda_T(t) \frac{\Delta f(t)}{2} \tag{2.28}$$

geschrieben werden. Der Zeitpunkt, zu dem das Raumobjekt die Distanz $\rho_T + \rho_R$ minimiert, sei mit t_c angeschrieben. Um diesen Zeitpunkt mittels Gleichung (2.19) zu finden, berechnet man entsprechend dem analytischen Standardverfahren

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\lambda_T \int_{t'=t_0}^{t} \Delta f(t')\,\mathrm{d}t' + [\rho_T(t_0) + \rho_R(t_0)]\right) = \lambda_T \Delta f(t) \stackrel{!}{=} 0 \quad \Leftrightarrow \quad \Delta f(t) = 0. \tag{2.29}$$

Diese Bedingung ist erfüllt, wenn t_c gleich t_0 in Gleichung (2.19) gesetzt wird, also $t_c = t_0$. Dann gilt also

$$p_0 \doteq p(t_0) = \frac{1}{2}[\rho_T(t_0) + \rho_R(t_0)] = a(t_0) \doteq a_0. \tag{2.30}$$

Für t_0 gilt ferner aufgrund von Gleichung (2.28)

$$\Delta f(t_0) = 0 \quad \Rightarrow \quad \vec{p}(t_0) \cdot \dot{\vec{p}}(t_0) = 0 \quad \Rightarrow \quad \vec{p}_0 \doteq \vec{p}(t_0) \perp \dot{\vec{p}}(t_0) \doteq \dot{\vec{p}}_0. \tag{2.31}$$

Aus Gleichungen (2.24) und (2.25) folgt überdies, dass

$$\begin{aligned} \Delta f(t_0) = 0 \quad \Rightarrow \quad \dot{\vec{\rho}}_S(t_0) \cdot \left(\frac{\vec{\rho}_T(t_0)}{\rho_T(t_0)} + \frac{\vec{\rho}_R(t_0)}{\rho_R(t_0)}\right) = \alpha \dot{\vec{\rho}}_S(t_0) \cdot \vec{p}\,(t_0) = \lambda_T(t_0)\Delta f(t_0) = 0 \\ \Rightarrow \quad \dot{\vec{\rho}}_S(t_0) \perp \vec{p}\,(t_0) \end{aligned} \tag{2.32}$$

gilt, wobei $\alpha \in \mathbb{R}$ ein passender Skalar ist. Es sei also folgendes festgehalten: Das Raumobjekt liegt zur Zeit t_0 auf einem Rotationsellipsoid der grossen Halbachse $a(t_0)$ und hat eine Geschwindigkeit $\dot{\vec{\rho}}_S(t_0)$, die per Definition der Geschwindigkeit tangential zu jenem Rotationsellipsoid ist. Ferner ist $\vec{p}_0$ orthogonal zur Geschwindigkeit $\dot{\vec{\rho}}_S(t_0)$ und ebenfalls orthogonal zu $\dot{\vec{p}}_0$, somit liegen beide Vektoren $\dot{\vec{\rho}}_S(t_0)$ und $\dot{\vec{p}}_0$ in der Ebene, die tangential ist zur Position des Raumobjektes auf dem durch $a(t_0)$ und $e(t_0) = \xi/2a(t_0)$ definierten Rotationsellipsoid zur Zeit t_0.

Der Vektor $\vec{p}(t)$ sei nun nach Richards in eine Taylorreihe um den Entwicklungspunkt t_0 entwickelt, was in

$$\vec{p}(t) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\frac{\mathrm{d}^k \vec{p}(t)}{\mathrm{d}t^k}\right)\Bigg|_{t=t_0} (t - t_0)^k = \vec{p}(t_0) + \dot{\vec{p}}(t_0)(t - t_0) + \mathcal{O}(t^2) \tag{2.33}$$

resultiert [**?**, S.1730]. Quadrieren und Anwenden der Relation (2.31) führt zu

$$p(t)^2 = p(t_0)^2 + |\dot{\vec{p}}(t_0)|^2 (t - t_0)^2 = p_0^2 + |\dot{\vec{p}}_0|^2 (t - t_0)^2. \tag{2.34}$$

Mit diesem Ergebnis gilt in einer kleinen Umgebung um $t \in [t_0 - \epsilon, t_0 + \epsilon]$ die Gleichung

$$\dot{p}(t) = \frac{\mathrm{d}}{\mathrm{d}t}|\vec{p}(t)| = \frac{\vec{p}(t) \cdot \dot{\vec{p}}(t)}{p(t)} \stackrel{(2.31)}{=} \frac{|\dot{\vec{p}}_0|^2 (t - t_0)}{[p_0^2 + |\dot{\vec{p}}_0|^2 (t - t_0)^2]^{\frac{1}{2}}} = \lambda_T(t) \frac{\Delta f(t)}{2}. \tag{2.35}$$

Hierbei ist zu bemerken, dass eine Anzahl von Tupeln $(t, \Delta f(t))$ die eigentlichen Messdaten konstituieren. Somit ist ein Bezug von den Messungen zum gesuchten «closest-approach»-Rotationsellipsoid über obige Gleichung hergestellt.

2.2.2 Ausgleichungsrechnung der Messdaten

Zwecks der Findung der Zeit des «closest-approach» t_0 sowie der dazugehörigen grossen Bahnhalbachse $a(t_0)$ müssen die Messdaten $\Delta f(t)$ nach Richards nun an ein funktionales Modell angeglichen werden [**?**, S.1731]. Zunächst werden die Messdaten $\Delta f(t)$ für ein mutmasslich die Zeit t_0 beinhaltendes Zeitintervall an ein Polynom dritten Grades

$$\dot{p}(t) = A + Bt + Ct^2 + Dt^3 \tag{2.36}$$

angeglichen. Hierzu sei auf die zahlreichen Lehrbücher zur Ausgleichsrechnung verwiesen. Weil für den «closest-approach» $\Delta f(t_0) = 0$ gefordert ist, kann t_0 als Nullstelle von $\dot{p}(t) = A + Bt + Ct^2 + Dt^3 = \lambda_T(t)^{\Delta f(t)}/_2 \stackrel{!}{=} 0$ ermittelt werden.

In einem zweiten Schritt, sobald t_0 aus dem ersten Schritt bekannt ist, können alle Messdaten an ein weiteres Polynom dritten Grades angeglichen werden, namentlich

$$\dot{p}(t) = \tilde{B}(t - t_0) + \tilde{C}(t - t_0)^2 + \tilde{D}(t - t_0)^3. \tag{2.37}$$

In einem dritten Schritt wird Gleichung (2.35) geschrieben durch

$$\frac{(t - t_0)^2}{\dot{p}(t)^2} = \frac{p_0^2 + |\dot{\vec{p}}_0|^2}{|\dot{\vec{p}}_0|^4 (t - t_0)^2} \quad \Leftrightarrow \quad \frac{\tau^2}{\dot{p}(t)^2} = \hat{A} + \hat{C}\tau^2, \tag{2.38}$$

wobei die Variablen

$$\hat{A} = \frac{p_0^2}{|\dot{\vec{p}}_0|^4}, \quad \hat{C} = \frac{1}{|\dot{\vec{p}}_0|^2} \quad \tau = t - t_0 \tag{2.39}$$

eingeführt wurden. Anschliessend wird die in Schritt 3 erhaltene obige Gleichung (2.38) an die in Schritt 2 erhaltene Gleichung (2.37) angeglichen. Daraus folgen konkrete Werte für $\hat{A}$ und $\hat{C}$, wobei der gesuchte Parameter $p_0 = p(t_0)$ durch

$$p(t_0)^2 = \frac{\hat{A}}{\hat{C}^2} \quad \Leftrightarrow \quad p_0 = \pm\sqrt{\frac{\hat{A}}{\hat{C}^2}} \tag{2.40}$$

gegeben und gemäss Relation (2.30) identisch zu $a(t_0)$ ist. Somit sind die Parameter $p_0 = a_0$ und t_0 bekannt, woraus durch die Kenntnis des Abstandes $\overline{TR} = \xi$ die Exzentrizität $e_0 = \xi/_{2a_0}$ des Rotationsellipsoides im «closest-approach» berechnet werden kann.

2.2.3 Geometrie der Messapparatur

Um weitere Berechnungen oder analytische Herleitungen durchzuführen, muss zuerst die Geometrie der Messapparatur definiert werden. Richards benutzte für seine Arbeiten ein sogenanntes DOPLOC-System, dessen Geometrie auch in vorliegender Arbeit verwendet und nachfolgend illustiert werden soll [**?**, S. 1732]. Die Geometrie der Messapparatur sowie das benutzte topozentrische Koordinatensystem sind in Abbildung 4 dargestellt.

Mittels der in Abbildung 4 beschriebenen Geometrie kann nun ein Bezug zum Zweikörperproblem hergestellt werden. Zur Herleitung dieses Bezuges betrachte man ferner Abbildung 5. Sobald t_0, $p_0 = a_0$ und damit $e_0 \doteq e(t_0) = \overline{TR}/_{2p_0} = \xi/_{2a_0}$ durch eine Ausgleichung der Messdaten mit dem analytischen Ausdruck (2.35) bekannt sind, ist das «closest-approach»-Rotationsellipsoid definiert, da die Distanz $\xi = \overline{TR}$ zwischen den Brennpunkten des Rotationsellipsoides durch die Position der Messstationen gegeben

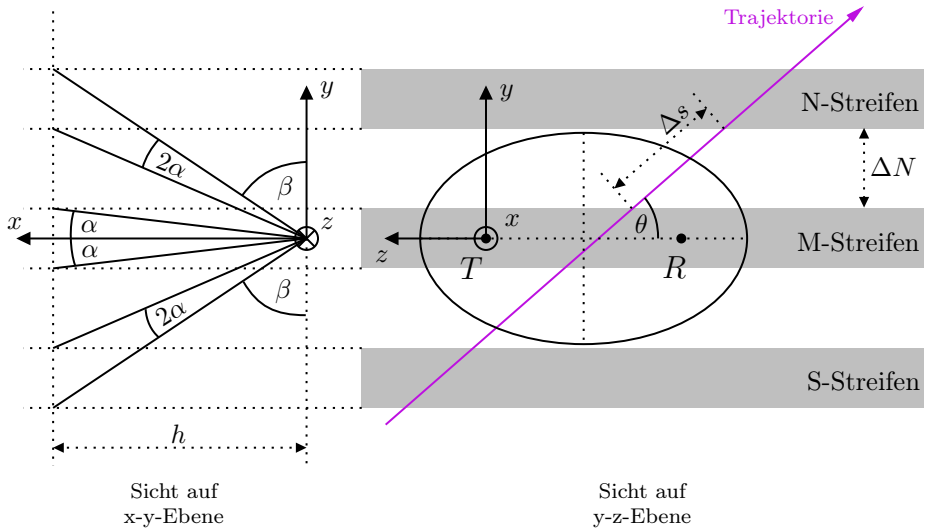


Abbildung 4: Geometrie der Messapparatur und Definition des benutzten, topozentrischen Koordinatensystems. Die ersichtliche Elllipse rechts im Bild ist die Projektion des «closest-approach»-Rotationsellipsoides auf die y-z-Ebene.

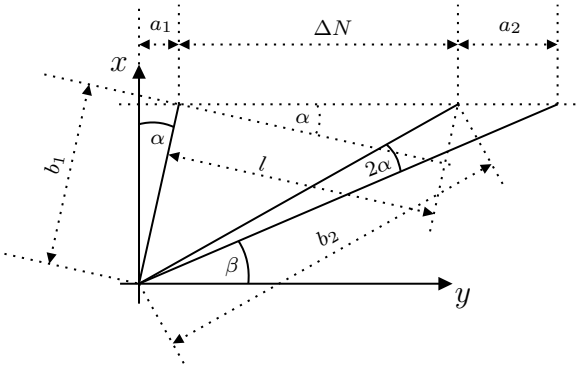


Abbildung 5: Abbildung zur Herstellung des Bezuges der Messgeometrie zum Zweikörperproblem.

ist. Gemäss obigen Überlegungen muss das Raumobjekt zur Zeit t_0 auf diesem Rotationsellipsoid liegen und hat eine Geschwindigkeit, die in der Tangentialebene zu diesem Rotationsellipsoid liegt. Also bleibt zu bestimmen, wo auf dem Rotationsellipsoid sich das Raumobjekt zur Zeit t_0 befindet und wie sich seine Geschwindigkeitskomponenten $\dot{\vec{\rho}}_S(t) = (\dot{x}_S(t), \dot{y}_S(t), \dot{z}_S(t))^T$ gestalten. An dieser Stelle wurde die in Beziehung (2.15) eingeführte Notation verwendet.

Es wird nun im Folgenden angenommen, dass sich das Raumobjekt während seines Durchfluges der Zeit T_D durch die drei Messstreifen des Transmitters in einer Ebene konstanter Höhe h bewege, wobei sich die Höhe h auf das in Abbildung 4 definierte, topozentrische Koordinatensystem bezieht. Ferner sei davon ausgegangen, dass das Raumobjekt während T_D eine konstante Geschwindigkeit v_D aufweise, die zu einem zirkulären Orbit um die Erde korrespondiere. Hierbei ist das dritte Kepler'sche Gesetz

$$\frac{d^3}{U^2} = \frac{\mu}{(2\pi)^2} \tag{2.41}$$

zu beachten, wobei d die grosse Halbachse der durch die Relativbewegung zweier Himmelskörper der Massen M_1 und M_2 im Zweikörperproblem beschriebenen Ellipse ist, U die Umlaufszeit des einen Körpers um den Anderen bezeichnet und $\mu = G(M_1 + M_2)$ eine

Konstante ist. Im Fall des zirkulären Orbits eines Raumobjektes der Masse $M_S \ll M_E$ um die Erde M_E gelten

$$d = R_E + x_S, \quad \frac{(2\pi)^2}{U} = \omega^2, \quad v_D^2 = (R_E + x_S)^2\omega^2, \tag{2.42}$$

wobei R_E der Radius der Erde und ω die konstante Winkelgeschwindigkeit des beobachteten Raumobjektes um die Erde sind. Dies führt mit dem dritten Kepler'schen Gesetz (2.41) zu

$$\frac{d^3}{U^2} = \frac{\mu}{(2\pi)^2} \quad \Leftrightarrow \quad v_D = \sqrt{\frac{\mu}{R_E + x_S}} \overset{M_S \ll M_E}{\approx} \sqrt{\frac{GM_E}{R_E + x_S}}. \tag{2.43}$$

Nach Richards sei nun entsprechend Abbildung 4 mit

$$\Delta s \doteq v_D \Delta t \tag{2.44}$$

jene Distanz gegeben, die das Raumobjekt zwischen dem M-Streifen und dem N-Streifen zurücklegt [**?**, S.1732]. Die Zeit Δt bezeichnet hierbei die dafür benötigte Zeitspanne, die zu der entsprechenden Lücke in den Dopplermessdaten korrespondiert. Aus Abbildung 4 ist ersichtlich, dass die Distanz Δs durch

$$\Delta s = \frac{\Delta N}{\sin(\theta)} \tag{2.45}$$

gegeben ist. Um die Grösse ΔN zu ermitteln, muss die in Abbildung 5 gezeichnete Geometrie betrachtet werden. Man entnimmt dieser Abbildung unter Anwendung elementarer trigonometrischer Beziehungen die Gleichungen

$$\cos(\alpha) = \frac{x_S}{b_1}, \quad \sin(2\alpha + \beta) = \frac{x_S}{b_2}, \quad \sin(\alpha) = \frac{a_1}{b_1}. \tag{2.46}$$

Durch Anwendung des Satzes von Pythagoras auf das durch die Seitenlängen $x_S, (a_1 + \Delta N), b_2$ gegebene Dreieck ergibt sich die quadratische Gleichung

$$x_S^2 + (a_1 + \Delta N)^2 = b_2^2 \quad \overset{(2.46)}{\Rightarrow} \quad c_1\Delta N^2 + c_2\Delta N + c_3 = 0, \tag{2.47}$$

wobei

$$c_1 = 1, \quad c_2 = 2x_S \tan(\alpha), \quad c_3 = x_S^2 + x_S^2 \tan^2(\alpha) - \frac{x_S^2}{\sin^2(2\alpha + \beta)} \tag{2.48}$$

sind. Die quadratische Lösungsformel

$$\frac{-c_2 \pm \sqrt{c_2^2 - 4c_1c_3}}{2c_1} \tag{2.49}$$

ergibt somit die Lösung für die gesuchte Beziehung ΔN zu

$$\Delta N_\pm = x_S[\pm \cot(2\alpha + \beta) - \tan(\alpha)], \tag{2.50}$$

wovon nur erstere Lösung mit positivem Vorzeichen $\Delta N_+ \doteq \Delta N$ verwendet werden soll. Mit den obigen Resultaten (2.43) und (2.45) ergibt sich dann ein Ausdruck für den Winkel θ zu

$$\sin(\theta) = \frac{\Delta N}{\Delta s} \overset{(2.43),(2.45)}{=} [\cot(2\alpha + \beta) - \tan(\alpha)]\frac{x_S}{\Delta t}\sqrt{\frac{R_E + x_S}{GM_E}}. \tag{2.51}$$

2.2.4 Ort und Geschwindigkeit des Raumobjektes berechnen

Zunächst seien die Geschwindigkeitskomponenten $\dot{\vec{\rho}}_S(t_0) = (\dot{x}_S(t_0), \dot{y}_S(t_0), \dot{z}_S(t_0))^T$ zur Zeit t_0 berechnet. Die Flughöhe bleibt gemäss Annahme im betrachteten Zeitintervall konstant, daher ist

$$\dot{x}_S(t_0) = 0. \tag{2.52}$$

Die anderen beiden Geschwindigkeitskomponenten ergeben sich aus Abbildung 4 zu

$$\dot{y}_S(t_0) = v_D \sin[\theta(t_0)] = [\cot(2\alpha + \beta) - \tan(\alpha)]\frac{x_S(t_0)}{\Delta t} \tag{2.53}$$

und

$$\dot{z}_S(t_0) = v_D \cos[\theta(t_0)] = \sqrt{\frac{GM_E}{R_E + x_S}} \cos\left[\arcsin\left(\frac{[\cot(2\alpha + \beta) - \tan(\alpha)]x_S(t_0)}{\Delta t\sqrt{\frac{GM_E}{R_E + x_S(t_0)}}}\right)\right], \tag{2.54}$$

wobei alle Grössen bekannt sind, ausser die Höhe $x_S(t_0)$ des Flugobjektes. Die Höhe des Flugobjektes kann aber eingeschränkt werden, da durch Gleichung (2.51) eine obere Schranke für die Flughöhe bei $\theta = \pi/2 + 2\pi k, k \in \mathbb{Z}$ gegeben ist. Diese Höhe sei mit $x_{S_{max}}(t_0)$ bezeichnet. Somit können die möglichen Werte für $x_S(t_0)$ zu $x_S(t_0) \in [0, x_{S_{max}}(t_0)]$ eingeschränkt werden, wobei an die in Beziehung (2.23) gemachte Annahme einer niedrigen Flughöhe $x_S(t_0) \ll R_E$ erinnert sei.

Die Positionskomponenten des Vektors $\vec{\rho}_S(t_0) = (x_S(t_0), y_S(t_0), z_S(t_0))^T$ ergeben sich sodann durch die folgenden Überlegungen an der Geometrie des Rotationsellipsoides. Die Bestimmungsgleichung für ein allgemeines Ellipsoid, dessen Zentrum sich im Koordinatenursprung befindet, lautet in kartesischen Koordinaten

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{2.55}$$

mit a der grossen Halbachse entlang der x-Achse, b der ersten kleinen Halbachse entlang der y-Achse und c der zweiten kleinen Halbachse entlang der z-Achse. Im Fall des Rotationsellipsoides mit Rotationsachse entlang der grossen Halbachse gilt $b = c$ und somit

$$\frac{x^2}{a^2} + \frac{y^2 + z^2}{b^2} = 1. \tag{2.56}$$

Wenn der Koordinatenursprung nach links in den Brennpunkt mit negativer x-Koordinate geschoben werden soll, so muss zu der x-Koordinate im neuen Koordinatensystem der Term ae mit der numerischen Exzentrizität e hinzuaddiert werden, sodass die Bestimmungsgleichung der Ellipse dann auf

$$\frac{(x + ae)^2}{a^2} + \frac{y^2 + z^2}{b^2} = 1 \tag{2.57}$$

lautet. Wenn nun die Koordinatentransformation

$$x \to -z, \quad y \to y, \quad z \to -x \tag{2.58}$$

gemacht wird, lautet die Bestimmungsgleichung

$$\frac{(z + ae)^2}{a^2} + \frac{x^2 + y^2}{b^2} = 1 \quad \Leftrightarrow \quad \frac{(z + ae)^2}{\frac{a^2}{b^2}(b^2 - x^2)} + \frac{y^2}{b^2 - x^2} = 1, \tag{2.59}$$

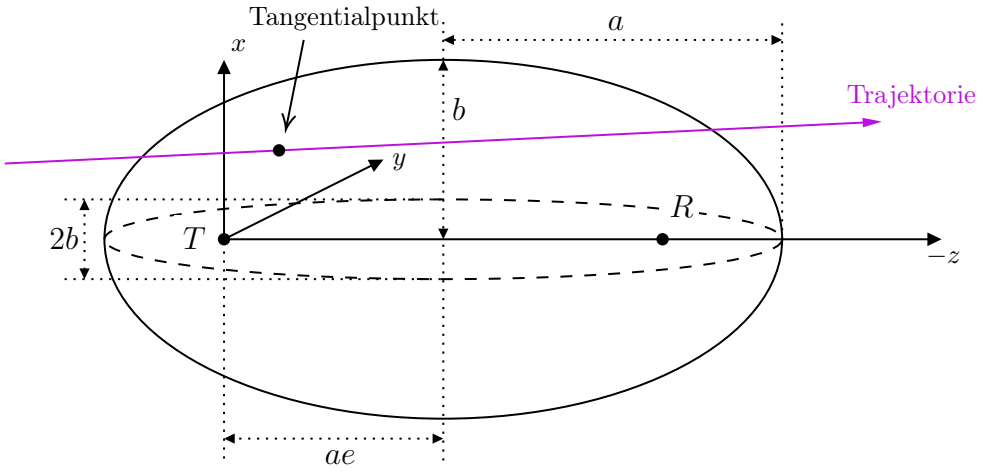


Abbildung 6: Skizze des «closest-approach»-Rotationsellipsoides mit T und R in den Brennpunkten. Die Trajektorie des Raumobjektes ist zum Zeitpunkt t_0 des «closest-approach» tangential zum gezeichneten Rotationsellipsoid, berührt das Rotationsellipsoid also in einem Tangentialpunkt. Ferner hat die Trajektorie gemäss Annahme eine Konstante Höhe von $x = h = x_S(t_0) = \text{cst}$.

was die Situation in Abbildung 6 wiedergibt. Weil die Trajektorie tangential zum Rotationsellipsoid ist und gemäss Annahme lokal eine konstante topozentrische Höhe von $h = x_S(t_0)$ hat, befindet sie sich in einer Ebene parallel zur y-z-Ebene. Aufgrund der konstanten Höhe kann die Variable x als fixiert betrachtet werden. Somit muss die Trajektorie tangential zum Rotationsellipsoid in der y-z-Ebene sein. Obige Bestimmungsgleichung (2.59) des Rotationsellipsoides wird nun so umgeformt, dass y in Abhängigkeit von z wiedergegeben wird:

$$y(z) = \pm \left[b^2 - x^2 - \frac{b^2}{a^2}(z + ae)^2 \right]^{1/2}, \quad x = \text{cst} \, . \tag{2.60}$$

Aufgrund der Tangentialität der Trajektorie, welche die z-Achse in der y-z-Ebene unter dem Winkel θ wie in Abbildung 4 ersichtlich schneidet, muss für die Steigung von $y(z)$ die Forderung

$$\frac{\mathrm{d}y(z)}{\mathrm{d}z} \stackrel{!}{=} \tan(\theta) \tag{2.61}$$

erfüllt sein. Hieraus ergibt sich

$$\frac{\mathrm{d}y(z)}{\mathrm{d}z} = \pm \frac{\frac{b^2}{a^2}(z + ae)^2}{\left[b^2 - x^2 - \frac{b^2}{a^2}(z + ae)^2 \right]^{1/2}} \stackrel{!}{=} \tan(\theta) \quad \Leftrightarrow \quad \tan^2(\theta) = \frac{b^4}{a^4} \frac{(z + ae)^2}{y^2}. \tag{2.62}$$

Die erhaltene Relation wird nun wiederum in die Bestimmungsgleichung (2.59) eingesetzt, woraus für die y-Koordinate des Raumobjektes die Relation

$$y^2 = b^2 - x^2 - \frac{b^2}{a^2}(z + ae)^2 = \frac{\left(\frac{b}{a}\right)^2 (b^2 - x^2)}{\left(\frac{b}{a}\right)^2 + \tan^2(\theta)} \tag{2.63}$$

und für die z-Koordinate mit der Definition $\tilde{z} \doteq z + ae$ der Ausdruck

$$\tilde{z}^2 = (z + ae)^2 = y^2 \tan^2(\theta) \frac{a^4}{b^4} = \frac{\left(\frac{a}{b}\right)^2 (b^2 - x^2)}{\left(\frac{b}{a}\right)^2 + \tan^2(\theta)} \tan^2(\theta) \tag{2.64}$$

folgen. Zwecks der Bezugnahme auf das «closest-approach»-Rotationsellipsoid gilt es, in den oben erhaltenen Formeln die Ersetzungen

$$k \to k_S(t_0), \; k \in \{x, y, z\}, \qquad l \to l(t_0), \; l \in \{a, b, e, \theta\} \tag{2.65}$$

vorzunehmen.

Eine Zusammenstellung der Geschwindigkeits- und Positionsdaten des Raumobjektes zur Zeit t_0 kann somit gemäss Richards durch

$$\dot{x}_S(t_0) = 0, \tag{2.66}$$
$$\dot{y}_S(t_0) = v_D(t_0)\sin[\theta(t_0)], \tag{2.67}$$
$$\dot{z}_S(t_0) = v_D(t_0)\cos[\theta(t_0)], \tag{2.68}$$
$$x_S(t_0) \in [0, x_{S_{max}}(t_0)], \tag{2.69}$$
$$y_S(t_0) = \pm\sqrt{\frac{\left[\frac{b(t_0)}{a(t_0)}\right]^2[b(t_0)^2 - x_S(t_0)^2]}{\left[\frac{b(t_0)}{a(t_0)}\right]^2 + \tan^2[\theta(t_0)]}}, \tag{2.70}$$
$$z_S(t_0) = \pm\sqrt{\frac{\left[\frac{a(t_0)}{b(t_0)}\right]^2[b(t_0)^2 - x_S(t_0)^2]}{\left[\frac{b(t_0)}{a(t_0)}\right]^2 + \tan^2[\theta(t_0)]}\tan^2[\theta(t_0)] - a(t_0)e(t_0)} \tag{2.71}$$

gegeben werden, wobei die Relationen

$$v_D(t_0) = \sqrt{\frac{GM_E}{R_E + x_S(t_0)}}, \tag{2.72}$$
$$\theta(t_0) = \arcsin\left([\cot(2\alpha+\beta) - \tan(\alpha)]\frac{x_S(t_0)}{\Delta t}\sqrt{\frac{R_E + x_S(t_0)}{GM_E}}\right), \tag{2.73}$$
$$a(t_0) = p(t_0), \tag{2.74}$$
$$b(t_0) = a(t_0)\sqrt{1 - e(t_0)^2}, \tag{2.75}$$
$$e(t_0) = \xi/_{2a(t_0)}, \tag{2.76}$$
$$\xi = \overline{TR} \tag{2.77}$$

gelten [**?**, S.1732]. Für eine erlaubte gegebene Höhe $x_S(t_0)$ gibt es aufgrund der Ellipsoidgeometrie ferner vier mögliche Lösungen $(y_S(t_0), z_S(t_0))$. Welche der vier Lösungen die zutreffende ist, kann

(1) durch eine Beobachtung der Flugrichtung Nord-Süd oder Süd-Nord und

(2) durch die Position des Raumobjektes nördlich oder südlich des M-Streifens zur Zeit t_0

ermittelt werden.

Eine Mehrdeutigkeit der möglichen Lösungen ist jedoch durch die bedingte Variabilität der Flughöhe $x_S(t_0)$ gegeben. Nachfolgend sei qualitativ ein anzuwendendes Verfahren beschrieben, welches zur realitätsnächsten Wahl des Wertes $x_S(t_0)$ angewandt werden kann. Zunächst seien die Vektoren $\vec{\rho}_S(t_0)$ und $\dot{\vec{\rho}}_S(t_0)$ für $n \in \mathbb{N}$ verschiedene Werte von $x_S(t_0) \in [0, x_{S_{max}}(t_0)]$ berechnet. Hieraus können für alle resultierenden Fälle von Tupeln $(\vec{\rho}_S, \dot{\vec{\rho}}_S)$ die Bahnelemente nach den Formeln des Zweikörperproblems berechnet werden. Aus den Bahnelementen aller Anfangswertprobleme können die Ephemeriden berechnet werden, also $\vec{\rho}_S(t)$ und $\dot{\vec{\rho}}_S(t)$ zu allen Zeiten $t \in \mathbb{R}$ für alle Tupel $(\vec{\rho}_S, \dot{\vec{\rho}}_S)$. Diese wiederum können in zu erwartende Dopplerkurven $\Delta f(t)$ übersetzt werden, die dann mit

der gemessenen Dopplerkurve abgeglichen werden können. Mit der Methode der kleinsten Quadrate kann nun die Dopplerkurve von jenem Tupel $(\vec{\rho}_S, \dot{\vec{\rho}}_S)$ ermittelt werden, dessen aus den Ephemeriden berechnete Dopplerkurve die kleinsten Residuen gegenüber der tatsächlichen Dopplerkurve aufweist. Da zum ermittelten Tupel auch eine Höhe $x_S(t_0)$ korrespondiert, ist somit die Höhe des Raumobjektes determiniert. Als Ausblick ist es durchaus denkbar, den so erhaltenen Orbit des Raumobjektes durch weitere Messungen mithilfe der Ausgleichsrechnung aus der Astrodynamik weiter zu verbessern und somit genauere Ephemeriden für das untersuchte Raumobjekt zu erhalten.

3 Methode 2: Ein Sender, mehrere Empfänger

3.1 Theoretische Herleitung der Methode

Die nachfolgend beschriebene Methode zur Bestimmung von Geschwindigkeit und Position eines Raumobjektes mittels des Dopplereffektes für elektromagnetische Wellen involviert die Lösung eines nichtlinearen Gleichungssystems, setzt aber a priori keine Physik voraus als nur den Dopplereffekt für elektromagnetische Wellen. Iman Shames et al. untersuchten in ihren Arbeiten ein zweidimensionales System bestehend aus $n \in \mathbb{N}$ Messstationen, die je über Sende- und Empfangsmöglichkeit verfügen, womit sie Betrachtungen zu Lösbarkeit und Lösungsmengen des resultierenden nichtlinearen Gleichungssysteme durchführten [**?**, S.266]. Inspiriert durch die Betrachtungen von Shames et al. wird in der vorliegenden Arbeit ein Verfahren entwickelt, welches zur Orts- und Geschwindigkeitsbestimmung von Raumobjekten im dreidimensionalen Raum verwendet werden kann und als messtechnische Voraussetzung eine Sendestation und sechs Empfängerstationen bedingt.

Zunächst sei die in Abbildung 7 gegebene Messanordnung betrachtet. Als Ursprung

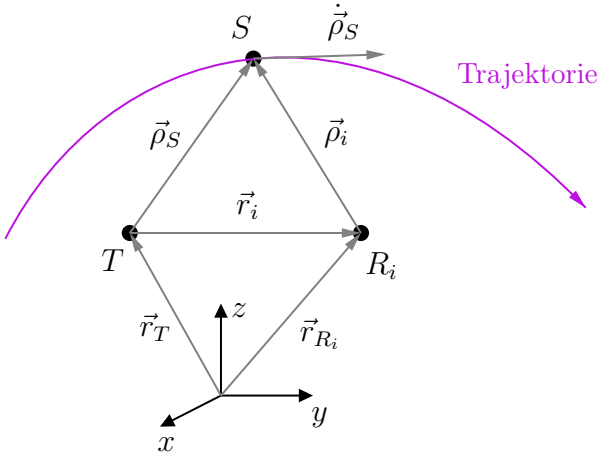


Abbildung 7: Messaufbau zur Methode der Lösung eines nichtlinearen Gleichungssystems. Der Ursprung des Koordinatensystems kann geozentrisch mit der x-Achse in Richtung des Kreuzungspunktes von Nullmeridian und Äquator und der z-Achse in Richtung Nordpol gewählt, oder topozentrisch mit der z-Achse normal zur Erdoberfläche, der x-Achse parallel zu den Breitengraden und der y-Achse parallel zu den Längengraden angesetzt werden.

des Koordinatensystems sei im Folgenden ein topozentrisches System mit Ursprung in T gewählt, wobei die z-Achse senkrecht zur Erdoberfläche stehe, die y-Achse parallel zum lokalen Längengrad und die x-Achse parallel zum lokalen Breitengrad seien. Ferner befinde sich in T eine Sendestation, welche sphärische elektromagnetische Wellen der konstanten Frequenz f_T in einen bestimmten geometrischen Bereich aussende. Es seien $n \in \mathbb{N}$ Empfängerstationen an den Orten $R_i, i \in \{1, \ldots, n\}$ vorhanden, welche die von einem sich im Raum bewegenden Raumobjekt reflektierten und deshalb dopplerverschobenen elektromagnetischen Wellen registrieren, die ursprünglich vom Transmitter in T ausgehen. Die gemessenen Frequenzen an der Empfängerstation R_i seien mit f_i bezeichnet. Ist die Betragsgeschwindigkeit

$$\dot{\rho}_S(t) = \frac{\mathrm{d}}{\mathrm{d}t}|\vec{\rho}_S(t)| = \frac{\vec{\rho}_S(t) \cdot \dot{\vec{\rho}}_S(t)}{|\vec{\rho}_S(t)|} \tag{3.1}$$

des Raumobjektes im Vergleich zur Lichtgeschwindigkeit c vernachlässigbar - gilt also

$\dot{\rho}_S \ll c$ - so ist die beim Raumobjekt zu messende Frequenz f_S gemäss der Dopplerformel (5.43) gegeben durch

$$f_S(\dot{\rho}_S(t)) = f_T \left(\frac{c - \dot{\rho}_S(t)}{c} \right). \tag{3.2}$$

Beim Empfänger R_i gilt dann gemäss derselben Dopplerformel

$$\begin{aligned} f_i(\dot{\rho}_S(t), \dot{\rho}_i(t)) &= f_S(\dot{\rho}_S(t)) \left(\frac{c - \dot{\rho}_i(t)}{c} \right) = f_T \left(\frac{c - \dot{\rho}_S(t)}{c} \right) \left(\frac{c - \dot{\rho}_i(t)}{c} \right) \\ &= \frac{f_T}{c^2} [c^2 - c\dot{\rho}_S(t) - c\dot{\rho}_i(t) + \dot{\rho}_S(t)\dot{\rho}_i(t)] \approx f_T \left(1 - \frac{\dot{\rho}_S(t)}{c} - \frac{\dot{\rho}_i(t)}{c} \right), \end{aligned} \tag{3.3}$$

wobei Terme der Ordnung $\mathcal{O}[(\dot{\rho}/c)^2]$ in letzterer Approximation vernachlässigt wurden. Dies ist zulässig, da $\dot{\rho}_S \ll c$ gefordert wurde und diese Forderung für Raumobjekte in Erdumlaufbahnen erfüllt ist - denn deren Geschwindigkeiten sind von der Grössenordnung $10\text{e}3\,\mathrm{m\,s^{-1}}$. Für die gemessene Dopplerverschiebung Δf_i des Signals f_T bei der Messstation R_i gilt also

$$\Delta f_i(\dot{\rho}_S(t), \dot{\rho}_i(t)) = f_T - f_i(\dot{\rho}_S(t), \dot{\rho}_i(t)) = \frac{f_T}{c} [\dot{\rho}_S(t) + \dot{\rho}_i(t)]. \tag{3.4}$$

Man berechnet mit $\vec{\rho}_i(t) = \vec{\rho}_S(t) - \vec{r}_i$ sodann

$$\dot{\rho}_i(t) = \frac{\mathrm{d}}{\mathrm{d}t} \sqrt{\vec{\rho}_i(t) \cdot \vec{\rho}_i(t)} = \frac{\mathrm{d}}{\mathrm{d}t} \sqrt{\vec{\rho}_S(t)^2 - 2\vec{\rho}_S(t) \cdot \vec{\rho}_i(t) + \vec{\rho}_i(t)^2} = \frac{(\vec{\rho}_S(t) - \vec{r}_i) \cdot \dot{\vec{\rho}}_S(t)}{|\vec{\rho}_S(t) - \vec{r}_i|}. \tag{3.5}$$

Ferner ist Betragsänderung $\dot{\vec{\rho}}_S(t)$ des Raumobjektvektors in Abhängigkeit von der Zeit gemäss Gleichung (3.1) durch

$$\dot{\vec{\rho}}_S(t) = \frac{\vec{\rho}_S(t) \cdot \dot{\vec{\rho}}_S(t)}{|\vec{\rho}_S(t)|} \tag{3.6}$$

gegeben. Unter Benutzung von Gleichung (3.4) folgt daraus für die Dopplerverschiebungen bei den Empfängerstationen die Gleichung

$$\begin{aligned} \Delta f_i(\vec{\rho}_S(t), \dot{\vec{\rho}}_S(t)) &= \frac{f_T}{c} [\dot{\rho}_S(t) + \dot{\rho}_i(t)] = \frac{f_T}{c} \left[\frac{\vec{\rho}_S(t) \cdot \dot{\vec{\rho}}_S(t)}{|\vec{\rho}_S(t)|} + \frac{(\vec{\rho}_S(t) - \vec{r}_i) \cdot \dot{\vec{\rho}}_S(t)}{|\vec{\rho}_S(t) - \vec{r}_i|} \right] \\ &= \frac{f_T}{c} \dot{\vec{\rho}}_S(t) \cdot \left[\frac{\vec{\rho}_S(t)}{|\vec{\rho}_S(t)|} + \frac{(\vec{\rho}_S(t) - \vec{r}_i)}{|\vec{\rho}_S(t) - \vec{r}_i|} \right]. \end{aligned} \tag{3.7}$$

Mit der Definition

$$\delta f_i \doteq \frac{\Delta f_i c}{f_T} \tag{3.8}$$

kann man das Gleichungssystem kompakt durch

$$\delta f_i(\vec{\rho}_S(t), \dot{\vec{\rho}}_S(t)) - \dot{\vec{\rho}}_S(t) \cdot \left[\frac{\vec{\rho}_S(t)}{|\vec{\rho}_S(t)|} + \frac{(\vec{\rho}_S(t) - \vec{r}_i)}{|\vec{\rho}_S(t) - \vec{r}_i|} \right] = 0, \quad i \in \{1, \ldots, n\}, \quad n \in \mathbb{N} \tag{3.9}$$

wiedergeben. Durch die Bezeichnungen

$$\vec{\rho}_S(t) = \begin{pmatrix} x_S(t) \\ y_S(t) \\ z_S(t) \end{pmatrix}, \quad \dot{\vec{\rho}}_S(t) = \begin{pmatrix} \dot{x}_S(t) \\ \dot{y}_S(t) \\ \dot{z}_S(t) \end{pmatrix}, \quad \vec{r}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \mathrm{cst} \tag{3.10}$$

kann das Gleichungssystem für n Empfängerstationen zu

$$\begin{array}{rcl} \frac{\dot{x}_S x_S+\dot{y}_S y_S+\dot{z}_S z_S}{\sqrt{x_S^2+y_S^2+z_S^2}} + \frac{\dot{x}_S(x_S-x_1)+\dot{y}_S(y_S-y_1)+\dot{z}_S(z_S-z_1)}{\sqrt{(x_S-x_1)^2+(y_S-y_1)^2+(z_S-z_1)^2}} - \delta f_1 &=& 0 \\ & \vdots & \\ \frac{\dot{x}_S x_S+\dot{y}_S y_S+\dot{z}_S z_S}{\sqrt{x_S^2+y_S^2+z_S^2}} + \frac{\dot{x}_S(x_S-x_n)+\dot{y}_S(y_S-y_n)+\dot{z}_S(z_S-z_n)}{\sqrt{(x_S-x_n)^2+(y_S-y_n)^2+(z_S-z_n)^2}} - \delta f_n &=& 0 \end{array} \tag{3.11}$$

ausgeschrieben werden, wobei die explizite Anschrift der Zeitabhängigkeiten wegen Übersichtlichkeitsgründen weggelassen wurde.

Im nächsten Abschnitt geht es darum herauszufinden, wann und unter welchen Bedingungen dieses Gleichungssystem lösbar ist. In einem weiteren Abschnitt sollen schliesslich Betrachtungen zur konkreten numerischen Lösung und deren Implementierung in eine Programmiersprache durchgeführt werden.

3.2 Lösbarkeit und Lösungsverfahren von linearen und nichtlinearen Gleichungssystemen

3.2.1 Lösbarkeit von linearen Gleichungssystemen

Es sei ein lineares Gleichungssystem mit $m \in \mathbb{N}$ Gleichungen und $n \in \mathbb{N}$ Unbekannten durch

$$\begin{array}{rcl} c_{11}x_1 + c_{12}x_2 + \ldots + c_{1n}x_n &=& b_1 \\ c_{21}x_1 + c_{22}x_2 + \ldots + c_{2n}x_n &=& b_2 \\ & \vdots & \\ c_{m1}x_1 + c_{m2}x_2 + \ldots + c_{mn}x_n &=& b_m \end{array} \tag{3.12}$$

gegeben. Dieses Gleichungssystem kann mittels den Definitionen

$$C \doteq \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \ldots & c_{mn} \end{pmatrix}, \quad B \doteq \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}, \quad X \doteq \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{3.13}$$

als

$$CX = B \tag{3.14}$$

geschrieben werden. Ferner sei die erweiterte Koeffizientenmatrix $(C|B)$ durch

$$(C|B) = \begin{pmatrix} c_{11} & c_{12} & \ldots & c_{1n} & b_1 \\ c_{21} & c_{22} & \ldots & c_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m1} & c_{m2} & \ldots & c_{mn} & b_m \end{pmatrix} \tag{3.15}$$

definiert. Das obige lineare Gleichungssystem besitzt eine eindeutige Lösung genau dann, wenn

$$\text{rank}[(C|B)] = \text{rank}[C] = n \tag{3.16}$$

gilt, der Rang der erweiterten Koeffizientenmatrix $(C|B)$ also gleich der dem Rang der Koeffizientenmatrix C und auch gleich der Anzahl Unbekannten n des Gleichungssystems ist. Im Fall von $n = m$ ist das Gleichungssystem genau dann eindeutig lösbar, wenn

$$\det(C) \neq 0 \tag{3.17}$$

gilt.

3.2.2 Lösungsverfahren für nichtlineare Gleichungen

Es sei $f(t)$ eine glatte und eindimensionale Funktion. Um die Nullstelle von $f(t)$ zu finden, kann das sogenannte Newton-Verfahren angewandt werden. Die Nullstelle von $f(t)$ zu finden heisst, eine Lösung der nichtlinearen Gleichung

$$f(t) = 0 \tag{3.18}$$

zu suchen. Hierzu wird die Funktion mittels einer Taylorentwicklung um einen Punkt $t_{i-1}, i \in \mathbb{N}$ entwickelt:

$$f(t) = \sum_{k=0}^{\infty} \frac{1}{k!} \frac{\mathrm{d}f(t)}{\mathrm{d}t}\bigg|_{t=t_{i-1}} (t - t_{i-1})^k = f(t_{i-1}) + \frac{\mathrm{d}f(t)}{\mathrm{d}t}\bigg|_{t=t_{i-1}} (t - t_{i-1}) + \mathcal{O}(t^2). \tag{3.19}$$

Die Entwicklung wird nach erster Ordnung abgebrochen und der Funktionswert soll an der Stelle t_i berechnet werden, sodass die Approximation

$$f(t_i) \approx f(t_{i-1}) + \frac{\mathrm{d}f(t)}{\mathrm{d}t}\bigg|_{t=t_{i-1}} (t_i - t_{i-1}) = f(t_{i-1}) + J(t_{i-1}) \cdot (t_i - t_{i-1}) = 0 \tag{3.20}$$

resultiert, wobei die Notation

$$J(t_i) \doteq \frac{\mathrm{d}f(t)}{\mathrm{d}t}\bigg|_{t=t_i} \tag{3.21}$$

verwendet wurde. Dabei gilt t_{i-1} als Startwert der Iteration, der möglichst Nahe an einer Vermuteten Nullstelle von $f(t)$ gewählt werden sollte. Das Gleichungssystem kann nun mittels analytischen oder numerischen Methoden nach t_i aufgelöst werden. Sobald t_i bekannt ist, wird ein Iterationsschritt durchgeführt, das heisst, dass t_i in die Gleichung

$$f(t_i) + J(t_i)(t_{i+1} - t_i) = 0 \tag{3.22}$$

eingesetzt und nach t_{i+1} aufgelöst wird. Hierauf folgt wiederum der nächste Iterationsschritt. Die Iteration wird abgebrochen, sobald die gewünschte relative Verbesserung $|f(t_j) - f(t_{j-1})|, j \in \mathbb{N}$ kleiner als ein vorgegebener Grenzwert $\varepsilon \lll 1$ ist, also

$$|f(t_j) - f(t_{j-1})| < \varepsilon \tag{3.23}$$

gilt. Der erhaltene Wert t_j gilt dann als genäherte Lösung der nichtlinearen Gleichung (3.18).

3.2.3 Lokales Lösungsverfahren für nichtlineare Gleichungssysteme

Ein nichtlineares Gleichungssystem kann durch Linearisierung in ein lineares Gleichungssystem überführt werden. Jede Gleichung kann prinzipiell in der Form $G = 0$ geschrieben werden, wobei G durch die Gleichungsbestandteile vor der Umformung zu $G = 0$ definiert ist. Es sei nun also ein nichtlineares, glattes Gleichungssystem mit $m \in \mathbb{N}$ Gleichungen $F_1(x_1, \ldots, x_n) = 0, \ldots, F_m(x_1, \ldots, x_n) = 0$ und $n \in \mathbb{N}$ Unbekannten $x_1, \ldots, x_n$ gegeben. Das Gleichungssystem kann also komponentenweise zu

$$\begin{aligned} F_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ F_m(x_1, \ldots, x_n) &= 0 \end{aligned} \tag{3.24}$$

geschrieben werden. Mittels den durch

$$F \doteq (F_1, \ldots, F_m) \in \mathbb{R}^m, \qquad x \doteq (x_1, \ldots, x_n) \in \mathbb{R}^n \tag{3.25}$$

definierten Vektoren kann das Gleichungssystem kompakter als

$$F(x) = 0 \tag{3.26}$$

wiedergegeben werden. Das oben für den eindimensionalen Fall beschriebene Newton-Verfahren kann zum allgemeinen Fall mit m Gleichungen und n Variablen verallgemeinert werden. Die Funktion $F(x)$ wird also linearisiert, das heisst, deren Taylorentwicklung um den Vektor $x_{i-1} = (x_{i-1,1}, \ldots, x_{i-1,n})$ wird nach erster Ordnung abgebrochen. Dann gilt

$$F(x) \approx F(x_{i-1}) + \nabla F(x_{i-1})(x - x_{i-1}) = F(x_{i-1}) + J(x_{i-1})(x - x_{i-1}) = 0, \tag{3.27}$$

wobei $J(u) \doteq \nabla F(u)$ die Jakobimatrix im Punkt $u \in \mathbb{R}^n$ ist:

$$J(u) = \nabla F(u) = \begin{pmatrix} \frac{\partial F_1(x_1,\ldots,x_n)}{\partial x_1}\Big|_u & \frac{\partial F_1(x_1,\ldots,x_n)}{\partial x_2}\Big|_u & \cdots & \frac{\partial F_1(x_1,\ldots,x_n)}{\partial x_n}\Big|_u \\ \frac{\partial F_2(x_1,\ldots,x_n)}{\partial x_1}\Big|_u & \frac{\partial F_2(x_1,\ldots,x_n)}{\partial x_2}\Big|_u & \cdots & \frac{\partial F_2(x_1,\ldots,x_n)}{\partial x_n}\Big|_u \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m(x_1,\ldots,x_n)}{\partial x_1}\Big|_u & \frac{\partial F_m(x_1,\ldots,x_n)}{\partial x_2}\Big|_u & \cdots & \frac{\partial F_m(x_1,\ldots,x_n)}{\partial x_n}\Big|_u \end{pmatrix}. \tag{3.28}$$

Der Vektor x_{i-1} gilt dabei als Startwert der Iteration, der möglichst nahe an einer vermuteten Nullstelle von $F(x)$ gewählt werden sollte. Ausgewertet im Vektor $x_i = (x_{i,1}, \ldots, x_{i,n}) \in \mathbb{R}^n$ lautet das zu lösende lineare Gleichungssystem nun

$$F(x_{i-1}) + J(x_{i-1})(x_i - x_{i-1}) = 0, \tag{3.29}$$

woraus der Lösungsvektor x_i mittels Methoden der linearen Algebra berechnet wird. Zu diesem Zweck wird der Vektor $\delta_{i,i-1} \doteq x_i - x_{i-1} \in \mathbb{R}^n$ definiert. Dann ist das System

$$F(x_{i-1}) = -J(x_{i-1})\delta_{i,i-1} \quad \Leftrightarrow \quad \delta_{i,i-1} = -J(x_{i-1})^{-1}F(x_{i-1}) \tag{3.30}$$

zu lösen, was mittels Gauss'schem Eliminationsverfahren erfolgen kann. Daraus ist zu erkennen, dass das beschriebene Verfahren im Mehrdimensionalen nur funktioniert, wenn die Jakobimatrix $J(x)$ für alle $x \in \mathbb{R}^n$ invertierbar, das heisst

$$\det[J(x)] \neq 0 \; \forall x \in \mathbb{R}^n \tag{3.31}$$

ist. Dies wiederum ist nur möglich, falls die Jakobimatrix quadratisch ist, was

$$n = m \tag{3.32}$$

bedeutet. Hat man nun $\delta_{i,i-1}$ berechnet, so kann $x_i = x_{i-1} + \delta_{i,i-1}$ kalkuliert und in Gleichung

$$F(x_i) + J(x_i)(x_{i+1} - x_i) = 0 \tag{3.33}$$

eingesetzt werden, womit ein weiterer Iterationsschritt gegeben ist. Es sei durch

$$\|\cdot\| : \mathbb{R}^m \to \mathbb{R}, \quad F = (F_1, \ldots, F_m) \mapsto \|F\| = \sqrt{F_1^2 + \cdots + F_m^2} \tag{3.34}$$

die euklidische Norm auf $\mathbb{R}^m$ definiert. Sobald die relative Iterationsgenauigkeit $\|F(x_j) - F(x_{j-1})\|, j \in \mathbb{N}$ einen gewünschten Wert $\varepsilon \lll 1$ unterschreitet, also

$$\|F(x_j) - F(x_{j-1})\| < \varepsilon \tag{3.35}$$

gilt, wird die Iteration abgebrochen und der erhaltene Vektor x_j gilt als genäherter Lösungsvektor des nichtlinearen Gleichungssystems (3.24) mit $n = m$. Sind die Bedingungen

(1) $n = m$,

(2) $\det[J(x)] \neq 0 \; \forall x \in \mathbb{R}^n$ und

(3) $J(u)$ lipschitzstetig für alle $u \in \mathbb{R}^n$, das heisst $\exists L > 0 : \|J(x)^{-1} \cdot [J(y) - J(x)]\| \leq L \cdot \|y - x\| \; \forall x, y \in \mathbb{R}^n$,

für ein nichtlineares und glattes Gleichungssystem erfüllt, so konvergiert das beschriebene Newton-Verfahren lokal quadratisch, wie es der *Satz von Kantorowitsch* garantiert. Das bedeutet, dass das betrachtete Gleichungssystem bei Erfülltsein obiger Bedingungen mindestens eine Lösung hat und dass das beschriebene Newton-Verfahren lokal quadratisch gegen eine Lösung des Gleichungssystems $F(x) = 0$ konvergiert, sofern der Startvektor x_{i-1} der Iteration nahe genug bei einer tatsächlichen Lösung dieses Gleichungssystems liegt. Die obigen Bedingungen an die lokale Konvergenz des Newton-Verfahrens sollen im nachfolgenden Satz nach Armin Iske statuiert und bewiesen werden [**?**, S.113-116]:

Satz 1 (Zur lokalen Konvergenz des Newton-Verfahrens)**.** *Es sei $F : \mathbb{R}^n \supset U \to \mathbb{R}^n$ eine stetig differenzierbare Funktion, wobei U offen und konvex sei. $U \subset \mathbb{R}^n$ ist konvex genau dann, wenn für alle $a, b \in U$ und für alle $\lambda \in \mathbb{R}$ mit $0 \leq \lambda \leq 1$ stets $\lambda a + (1 - \lambda)b \in U$ gilt. Ferner sei $x^\star \in U$ eine Nullstelle von $F(x)$, das heisst $F(x^\star) = 0$. Es sollen zudem die folgenden Bedingungen gelten:*

(1) Die Jakobimatrix $J(u)$ sei regulär für alle $u \in U$, das heisst $\det[J(u)] \neq 0 \; \forall u \in U \subset \mathbb{R}^n$.

(2) Die Jakobimatrix $J(u)$ sei lipschitzstetig für alle $u \in U$, das heisst $\exists L > 0 : \|J(x)^{-1} \cdot [J(y) - J(x)]\| \leq L \cdot \|y - x\| \; \forall x, y \in U \subset \mathbb{R}^n$.

Dann konvergiert die Folge $(x_k)_{k \in \mathbb{N}_0}$ quadratisch gegen $x^\star$. Das heisst, dass $\|x_{k+1} - x^\star\| \leq L\|x_k - x^\star\|^2$ für alle $k \in \mathbb{N}_0$ gilt, wobei für den Startwert x_0 die Ungleichung $r \doteq \frac{1}{L} > \|x_0 - x^\star\|$ mit $K_r(x^\star) \subset U$ gilt. Ferner ist $x^\star$ eine eindeutige Nullstelle von $F(x)$ innerhalb der Kugel $K_r(x^\star)$, wobei $x_k \in K_r(x^\star)$ für alle $k \in \mathbb{N}_0$.

Beweis. Für $x, y \in U$ sei die Funktion $\phi(t) \doteq J(x)^{-1} \cdot F(x + t(y - x))$ betrachtet, wobei $t \in [0, 1]$ sei. Da $J(x)$ regulär und ferner $F(x)$ stetig differenzierbar sind, gilt für die Ableitung $\phi'(t)$ das Ergebnis $\phi'(t) = \frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = J(x)^{-1} \cdot J(x + t(y - x)) \cdot (y - x)$. Daraus folgt für die euklidische Norm $\|\cdot\|$ auf $\mathbb{R}^n$, dass

$$\begin{aligned}
\|\phi'(t) - \phi'(0)\| &= \|J(x)^{-1} \cdot [J(x - t(y - x)) - J(x)] \cdot (y - x)\| \\
&\leq \|J(x)^{-1} \cdot [J(x - t(y - x)) - J(x)]\| \cdot \|y - x\| \\
&\leq L \cdot t \cdot \|y - x\|^2
\end{aligned}$$

gilt. Unter Anwendung der Definitionsgleichung der Funktion $\phi(t)$ ergibt sich ferner die folgende Abschätzung:

$$\begin{aligned}
\|J(x)^{-1} \cdot [F(y) - F(x) - J(x) \cdot (y - x)]\| &= \|\phi(1) - \phi(0) - \phi'(0)\| \\
&\leq \| \int_0^1 [\phi'(t) - \phi'(0)] \, \mathrm{d}t \, \| \leq \int_0^1 \|\phi'(t) - \phi'(0)\| \, \mathrm{d}t \\
&\leq \int_0^1 L \cdot t \cdot \|y - x\|^2 \, \mathrm{d}t = L\|y - x\|^2.
\end{aligned}$$

Der Fehler $x_{k+1} - x^\star$ der Newton-Interation kann folgendermassen dargestellt werden:

$$\begin{aligned} x_{k+1} - x^\star &= x_k - J(x_k)^{-1}F(x_k) - x^\star \\ &= -J(x_k)^{-1} \cdot [F(x_k) - \underbrace{F(x^\star)}_{=0}] + (x_k - x^\star) \\ &= J(x_k)^{-1} \cdot [F(x^\star) - F(x_k) - J(x_k) \cdot (x^\star - x_k)]. \end{aligned}$$

Gemäss Voraussetzung des Satzes gilt $0 < \|x_k - x^\star\| < r = \frac{1}{L}$. Daraus folgt mit obigen Abschätzungen

$$\|x_{k+1} - x^\star\| \leq L\|x_k - x^\star\|^2 < Lr\|x_k - x^\star\| = \|x_k - x^\star\|.$$

Die nichtnegative Folge $(\|x_k - x^\star\|)_{k\in\mathbb{N}_0}$ ist somit streng monoton fallend und daher konvergent gegen eine reelle Zahl d mit $0 \leq d < r = \frac{1}{L}$ und $d \leq Ld^2$, woraus $d = 0$ folgt. Daher konvergiert die Folge $(x_k)_{k\in\mathbb{N}_0}$ quadratisch gegen $x^\star$. Ferner ist die Nullstelle $x^\star$ eindeutig, denn falls $x^{\star\star} \in K_r(x^\star)$ eine weitere Nullstelle von $F(x)$ ist, so resultiert daraus ein Widerspruch mit

$$\begin{aligned} \|x^{\star\star} - x^\star\| &= \|J(x^\star)^{-1} \cdot [F(x^{\star\star}) - F(x^\star) - J(x^\star)] \cdot (x^\star - x^{\star\star})\| \\ &\leq L\|x^\star - x^{\star\star}\|^2 = L\|x^{\star\star} - x^\star\|^2 < \|x^{\star\star} - x^\star\|. \end{aligned}$$

Folglich ist $x^\star$ die einzige Nullstelle von $F(x)$ in $K_r(x^\star)$. □

Der eben bewiesene Satz über die minimalen Konvergenzvoraussetzungen für das mehrdimensionale Newton-Verfahren zur Lösung nichtlinearer Gleichungssysteme ist auch als *Satz von Kantorowitsch* bekannt.

Nach Tilo Arens et al. besagt die multidimensionale Analysis, dass jede lineare Abbildung zwischen endlichdimensionalen normierten Vektorräumen lipschitzstetig ist [**?**, S.780]. Handelt es sich also bei $(\mathbb{R}^n, \|\cdot\| = \langle\cdot,\cdot\rangle^{1/2})$ um einen endlichdimensionalen Vektorraum, auf dem die in obigem Satz definierte stetig differenzierbare Funktion $F : \mathbb{R} \supset U \to \mathbb{R}^n$ definiert ist, so ist diese Funktion automatisch lipschitzstetig, wenn dessen Jakobimatrix $J(u)$ für alle $u \in U$ existiert und invertierbar ist - denn die Jakobimatrix $J(u)$ kann als lineare Abbildung $J(u) : \mathbb{R}^n \supset U \to \mathbb{R}^n$, $v \mapsto J(u)v$ zwischen zwei normierten Vektorräumen $(U, \|\cdot\|)$ und $(\mathbb{R}^n, \|\cdot\|)$ aufgefasst werden.

3.2.4 Globales Lösungsverfahren für nichtlineare Gleichungssysteme

Dass das im vorigen Abschnitt hergeleitete Newton'sche Lösungsverfahren nur lokal quadratisch konvergent ist bedeutet, dass das Lösungsverfahren für einen Startvektor $x_{i-1}, i \in \mathbb{N}$, der von einer Nullstelle $x^\star$ der Funktion $F(x)$ weit entfernt ist, nicht notwendigerweise gegen diese Nullstelle $x^\star$ konvergieren muss, sondern divergieren kann. Es geht also darum, das lokal konvergente Verfahren zur globalen Konvergenz zu verallgemeinern.

Das lokal konvergente Verfahren ist wie folgt charakterisiert. Hat man einen Startvektor $x_{i-1}, i \in \mathbb{N}$, so lautet das zu lösende lineare Gleichungssystem auf

$$F(x_{i-1}) + J(x_{i-1})(x_i - x_{i-1}) = 0. \tag{3.36}$$

Mit der Definition des Newton-Schrittes $\delta_{i,i-1} = x_i - x_{i-1}$ kann die Lösung durch

$$F(x_{i-1}) = -J(x_{i-1})\delta_{i,i-1} \quad \Leftrightarrow \quad \delta_{i,i-1} = -J(x_{i-1})^{-1}F(x_{i-1}) \tag{3.37}$$

beschrieben werden. Der nächste Interationsstartvektor x_i ist sodann durch $x_i = x_{i-1}+\delta_{i,i-1}$ gegeben. Wenn der Startvektor x_{i-1} aber zu weit entfernt von einer tatsächlichen Nullstelle $x^\star$ von $F(x)$ entfernt ist, kann es nun aber gemäss der lediglich lokalen Konvergenz des Verfahrens sein, dass das Iterationsverfahren mit dem neuen Startvektor x_i einen Newton-Schritt $\delta_{i+1,i}$ derart liefert, dass

$$\|x_{i+1} - x^\star\| = \|x_i + \delta_{i+1,i} - x^\star\| \geq \|x_{i-1} + \delta_{i,i-1} - x^\star\| = \|x_i - x^\star\| \tag{3.38}$$

gilt - das Verfahren also divergiert.

Um das Verfahren zur globalen Konvergenz zu erweitern, definiert man zunächst die Funktion

$$f(x) \doteq \frac{1}{2}F(x) \cdot F(x) = \frac{1}{2}\|F(x)\|^2 \geq 0 \tag{3.39}$$

und berechnet deren Gradient zu

$$\nabla f(x) = \frac{1}{2}(F(x) \cdot \nabla F(x) + \nabla F(x) \cdot F(x)) = F(x) \cdot \nabla F(x) = F(x) \cdot J(x). \tag{3.40}$$

Damit erhält man nach William Press et al. für den Ausdruck $\nabla f(x_{i-1}) \cdot \delta_{i,i-1}$ die Ungleichung

$$\nabla f(x_{i-1}) \cdot \delta_{i,i-1} = [F(x_{i-1}) \cdot J(x_{i-1})] \cdot [-J(x_{i-1})^{-1} \cdot F(x_{i-1})] = -F(x_{i-1}) \cdot F(x_{i-1}) < 0, \tag{3.41}$$

woraus folgt, dass der Newton-Schritt $\delta_{i,i-1}$ stets in Richtung des negativen Gradienten von $f(x)$ erfolgt [?, S.383]. Ferner sei bemerkt, dass jede Lösung von $F(x) = 0$ die Funktion $f(x) = \frac{1}{2}F(x) \cdot F(x)$ minimieren wird, da $f(x) \geq 0$ per Definition von $f(x)$.

Man kann Newton-Schritte $\delta_{i,i-1}, i \in \mathbb{N}$ dahingehend prüfen, ob diese die Funktion $f(x)$ weiter minimieren, also ob

$$f(x_i) = f(x_{i-1} + \delta_{i,i-1}) < f(x_{i-1}), \quad i \in \mathbb{N} \tag{3.42}$$

gilt. Obwohl nach Gleichung (3.41) jeder Newton-Schritt $\delta_{i,i-1}$ in abnehmender Richtung von $f(x)$ getätigt wird, ist gemäss Press et al. nicht gegeben, dass $f(x_i) = f(x_{i-1}+\delta_{i,i-1}) < f(x_{i-1})$ erfüllt ist - das liegt daran, dass im Newton-Verfahren per Definition quadratische Terme $\mathcal{O}(\delta_{i,i+1}^2) \approx 0$ vernachlässigt werden, die so erhaltene Approximation von $F(x_i)$ deshalb bei der Durchführung des vollen Newton-Schrittes $x_i = x_{i-1} + \delta_{i,i-1}$ nicht mehr gültig ist und folglich $f(x_i) = f(x_{i-1} + \delta_{i,i-1}) \not< f(x_{i-1})$ gilt [?, S.384]. Damit das Newton-Verfahren gegen eine Nullstelle $x^\star$ von $F(x)$ konvergiert, muss die obige Ungleichung (3.42) aber für jeden Newton-Schritt $\delta_{i,i-1}, i \in \mathbb{N}$ gelten. Daher kann man nach Press et al. ein Kriterium an die Akzeptanz des Newton-Schrittes $\delta_{i,i-1}$ definieren - nämlich das Kriterium, dass der Newton-Schritt $\delta_{i,i-1}$ die Funktion $f(x)$ kleiner machen muss, dass also die Ungleichung (3.42) für jeden Newton-Schritt $\delta_{i,i-1}, i \in \mathbb{N}$ erfüllt sein muss [?, S.384]. Wenn die Ungleichung (3.42) für einen Newton-Schritt $\delta_{i,i-1}, i \in \mathbb{N}$ nicht erfüllt ist, so kann man den Newton-Schritt $\delta_{i,i-1}$ entlang der hierdurch definierten Richtung gemäss

$$\tilde{x}_i = x_{i-1} + \lambda\delta_{i,i-1} = x_{i-1} + \tilde{\delta}_{i,i-1}, \quad 0 < \lambda \leq 1 \tag{3.43}$$

variieren, bis man einen Newton-Schritt $\tilde{\delta}_{i,i-1}$ erhält, für den die Ungleichung

$$f(\tilde{x}_i) = f(x_{i-1} + \lambda\delta_{i,i-1}) = f(x_{i-1} + \tilde{\delta}_{i,i-1}) < f(x_{i-1}) \tag{3.44}$$

erfüllt ist. Das ist möglich, weil - wie oben erwähnt - jeder Newton-Schritt $\delta_{i,i-1}$ stets in Richtung des negativen Gradienten von $f(x)$ erfolgt. Dieser Newton-Schritt $\tilde{\delta}_{i,i-1}$ wird dann in der Newton-Iteration für den Iterationsschritt $i-1 \to i$ verwendet. Dieses Verfahren - welches auch *Line Searching* oder *Backtracking* genannt wird - garantiert nach Press et al., dass das Newton-Verfahren global konvergiert, ausgenommen jener Fälle, bei denen dieses Verfahren ein lokales Minimum von $f(x)$ liefert [?, S.384].

3.3 Lösungsalgorithmen

3.3.1 Lösungsalgorithmus für lokal konvergentes Lösungsverfahren

Der nachfolgend elementar beschriebene Algorithmus entspricht dem Newton-Verfahren im mehrdimensionalen Raum, welches gemäss den Erläuterungen im Abschnitt 3.2.3 lokal quadratisch konvergent ist, sofern die lokalen Konvergenzbedingungen (1) und (2) nach Satz 1 erfüllt sind. Vorausgesetzt sei ein glattes Gleichungssystem $F(x) = 0$ mit

$$F : \mathbb{R}^n \to \mathbb{R}^n, \quad x = (x_1, \ldots, x_n) \mapsto F(x) = (F_1(x_1, \ldots, x_n), \ldots, F_n(x_1, \ldots, x_n)) \quad (3.45)$$

und $n \in \mathbb{N}$, welches lokal die Konvergenzbedingungen an das Newton-Verfahren erfülle. Beispielsweise kann der Algorithmus also auf das in Beziehung (3.11) definierte Gleichungssystem angewandt werden. Ferner bedingt der Algorithmus, dass mit $\|\cdot\|$ die euklidische Norm (3.34) im Raum $\mathbb{R}^n$ gegeben ist. Der Algorithmus ist dann wie folgt gegeben:

(L1) Setze $i = 1$ und wähle einen Punkt $x_{i-1} \in \mathbb{R}^n$ nahe einem vermuteten Lösungsvektor $x \in \mathbb{R}^n$ des Gleichungssystems $F(x) = 0$.

(L2) Linearisiere die Funktion $F(x)$ im Punkt $x_{i-1} \in \mathbb{R}^n$ und löse die linearisierte Gleichung $F(x_{i-1}) + J(x_{i-1})\delta_{i,i-1} = 0$ nach $\delta_{i,i-1} \doteq (x_i - x_{i-1})$ auf.

(L3) Setze $x_i = x_{i-1} + \delta_{i,i-1}$ und erhöhe den Wert von i um eine Einheit.

(L4) Wiederhole Schritte (L2) und (L3), bis die relative Iterationsgenauigkeit $\|F(x_j) - F(x_{j-1})\|, j \in \mathbb{N}$ einen gewünschten Wert $\varepsilon \lll 1$ unterschreitet, also $\|F(x_j) - F(x_{j-1})\| < \varepsilon$ gilt.

(L5) Beende den Algorithmus mit dem berechneten Lösungsvektor $x_j \in \mathbb{R}^n$, $j \in \mathbb{N}$.

3.3.2 Lösungsalgorithmus für global konvergentes Lösungsverfahren

Der nachfolgend elementar beschriebene Algorithmus entspricht dem Newton-Verfahren im mehrdimensionalen Raum mit der Modifikation des Line Searching oder Backtracking, wodurch gemäss den Erläuterungen im Abschnitt 3.2.4 globale Konvergenz des Verfahrens erreicht wird, sofern die lokalen Konvergenzbedingungen (1) und (2) gemäss Satz 1 erfüllt sind. Vorausgesetzt sei ein glattes Gleichungssystem $F(x) = 0$ mit

$$F : \mathbb{R}^n \to \mathbb{R}^n, \quad x = (x_1, \ldots, x_n) \mapsto F(x) = (F_1(x_1, \ldots, x_n), \ldots, F_n(x_1, \ldots, x_n)) \quad (3.46)$$

und $n \in \mathbb{N}$, welches lokal die Konvergenzbedingungen des Newton-Verfahrens erfülle. Zudem sei die nachfolgend im Lösungsalgorithmus verwendete Funktion $f(x)$ durch

$$f(x) \doteq \frac{1}{2} F(x) \cdot F(x) \quad (3.47)$$

definiert. Beispielsweise kann der Algorithmus also auf das in Beziehung (3.11) definierte Gleichungssystem angewandt werden. Ferner bedingt der Algorithmus, dass mit $\|\cdot\|$ die euklidische Norm (3.34) im Raum $\mathbb{R}^n$ gegeben ist. Der Algorithmus ist dann wie folgt gegeben:

(G1) Setze $i = 1$ und wähle einen Punkt $x_{i-1} \in \mathbb{R}^n$ (nahe einem vermuteten Lösungsvektor $x \in \mathbb{R}^n$ des Gleichungssystems $F(x) = 0$).

(G2) Linearisiere die Funktion $F(x)$ im Punkt $x_{i-1} \in \mathbb{R}^n$ und löse die linearisierte Gleichung $F(x_{i-1}) + J(x_{i-1})\delta_{i,i-1} = 0$ nach $\delta_{i,i-1} \doteq (x_i - x_{i-1})$ auf.

(G3) Variiere den Newton-Schritt $\delta_{i,i-1}$ gemäss $\tilde{x}_i = x_{i-1} + \lambda\delta_{i,i-1} = x_{i-1} + \tilde{\delta}_{i,i-1}, \quad 0 < \lambda \leq 1$, bis die Ungleichung $f(\tilde{x}_i) = f(x_{i-1} + \lambda\delta_{i,i-1}) = f(x_{i-1} + \tilde{\delta}_{i,i-1}) < f(x_{i-1})$ erfüllt ist.

(G4) Setze $x_i = x_{i-1} + \tilde{\delta}_{i,i-1}$ und erhöhe den Wert von i um eine Einheit.

(G5) Wiederhole Schritte (G2) bis (G4), bis die relative Iterationsgenauigkeit $\|F(x_j) - F(x_{j-1})\|, j \in \mathbb{N}$ einen gewünschten Wert $\varepsilon \lll 1$ unterschreitet, also $\|F(x_j) - F(x_{j-1})\| < \varepsilon$ gilt.

(G6) Beende den Algorithmus mit dem berechneten Lösungsvektor $x_j \in \mathbb{R}^n$, $j \in \mathbb{N}$.

3.4 Numerische Lösung des Gleichungssystems

3.4.1 Formulierung des Gleichungssystems

Das in Abschnitt 3.1 hergeleitete Gleichungssystem (3.9) für die vorliegend betrachtete Methode zur Ortung von Raumobjekten mittels Dopplerdaten daran reflektierter elektromagnetischen Wellen soll nun zur Implementierung in eine Programmiersprache in einer den Lösungsalgorithmen entsprechenden Notation ausformuliert werden.

Der Vektor der Variablen ist hierbei durch

$$\rho = (\vec{\rho}_S, \dot{\vec{\rho}}_S) = (x_S, y_S, z_S, \dot{x}_S, \dot{y}_S, \dot{z}_S) \in \mathbb{R}^6 \tag{3.48}$$

gegeben, wobei sämtliche Komponenten dieses Vektors zeitabhängig sind. Das explizite Notieren der Zeitabhängigkeiten soll aber der Übersichtlichkeit halber hier und im Folgenden weggelassen werden. Die Komponenten des Gleichungssystems

$$F(\rho) = 0 \tag{3.49}$$

sind sodann durch

$$F_1(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_1) + \dot{y}_S(y_S - y_1) + \dot{z}_S(z_S - z_1)}{\sqrt{(x_S - x_1)^2 + (y_S - y_1)^2 + (z_S - z_1)^2}} - \delta f_1 = 0 \tag{3.50}$$

$$F_2(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_2) + \dot{y}_S(y_S - y_2) + \dot{z}_S(z_S - z_2)}{\sqrt{(x_S - x_2)^2 + (y_S - y_2)^2 + (z_S - z_2)^2}} - \delta f_2 = 0 \tag{3.51}$$

$$F_3(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_3) + \dot{y}_S(y_S - y_3) + \dot{z}_S(z_S - z_3)}{\sqrt{(x_S - x_3)^2 + (y_S - y_3)^2 + (z_S - z_3)^2}} - \delta f_3 = 0 \tag{3.52}$$

$$F_4(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_4) + \dot{y}_S(y_S - y_4) + \dot{z}_S(z_S - z_4)}{\sqrt{(x_S - x_4)^2 + (y_S - y_4)^2 + (z_S - z_4)^2}} - \delta f_4 = 0 \tag{3.53}$$

$$F_5(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_5) + \dot{y}_S(y_S - y_5) + \dot{z}_S(z_S - z_5)}{\sqrt{(x_S - x_5)^2 + (y_S - y_5)^2 + (z_S - z_5)^2}} - \delta f_5 = 0 \tag{3.54}$$

$$F_6(\rho) = \frac{\dot{x}_S x_S + \dot{y}_S y_S + \dot{z}_S z_S}{\sqrt{x_S^2 + y_S^2 + z_S^2}} + \frac{\dot{x}_S(x_S - x_6) + \dot{y}_S(y_S - y_6) + \dot{z}_S(z_S - z_6)}{\sqrt{(x_S - x_6)^2 + (y_S - y_6)^2 + (z_S - z_6)^2}} - \delta f_6 = 0 \tag{3.55}$$

gegeben, wobei δf_i durch

$$\delta f_i \doteq \frac{\Delta f_i c}{f_T} \tag{3.56}$$

definiert ist. Hierbei ist Δf_i die beim Empfänger i mit $i \in \{1, \ldots, 6\}$ gemessene Dopplerverschiebung des am Raumobjekt reflektierten Signals zur Trägerfrequenz f_T.

Die Jakobimatrix $J(u)$ in einem Punkt $u \in U \subset \mathbb{R}^6$ ist gegeben durch

$$J(u) = \begin{pmatrix} \frac{\partial F_1(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_1(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_1(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_1(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_1(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_1(\rho)}{\partial \dot{z}_S}\Big|_u \\ \frac{\partial F_2(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_2(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_2(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_2(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_2(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_2(\rho)}{\partial \dot{z}_S}\Big|_u \\ \frac{\partial F_3(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_3(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_3(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_3(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_3(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_3(\rho)}{\partial \dot{z}_S}\Big|_u \\ \frac{\partial F_4(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_4(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_4(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_4(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_4(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_4(\rho)}{\partial \dot{z}_S}\Big|_u \\ \frac{\partial F_5(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_5(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_5(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_5(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_5(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_5(\rho)}{\partial \dot{z}_S}\Big|_u \\ \frac{\partial F_6(\rho)}{\partial x_S}\Big|_u & \frac{\partial F_6(\rho)}{\partial y_S}\Big|_u & \frac{\partial F_6(\rho)}{\partial z_S}\Big|_u & \frac{\partial F_6(\rho)}{\partial \dot{x}_S}\Big|_u & \frac{\partial F_6(\rho)}{\partial \dot{y}_S}\Big|_u & \frac{\partial F_6(\rho)}{\partial \dot{z}_S}\Big|_u \end{pmatrix}, \tag{3.57}$$

wobei die darin vorkommenden Ableitungen für $i \in \{1, \ldots, 6\}$ durch

$$\frac{\partial F_i(\rho)}{\partial x_S} = \frac{\dot{x}_S}{|\vec{\rho}_S - \vec{r}_i|} + \frac{\dot{x}_S}{|\vec{\rho}_S|} - \frac{x_S}{|\vec{\rho}_S|^3}[\dot{\vec{\rho}}_S \cdot \vec{\rho}_S] - \frac{x_S - x_i}{|\vec{\rho}_S - \vec{r}_i|^3}[\dot{\vec{\rho}}_S \cdot (\vec{\rho}_S - \vec{r}_i)], \tag{3.58}$$

$$\frac{\partial F_i(\rho)}{\partial y_S} = \frac{\dot{y}_S}{|\vec{\rho}_S - \vec{r}_i|} + \frac{\dot{y}_S}{|\vec{\rho}_S|} - \frac{y_S}{|\vec{\rho}_S|^3}[\dot{\vec{\rho}}_S \cdot \vec{\rho}_S] - \frac{y_S - y_i}{|\vec{\rho}_S - \vec{r}_i|^3}[\dot{\vec{\rho}}_S \cdot (\vec{\rho}_S - \vec{r}_i)], \tag{3.59}$$

$$\frac{\partial F_i(\rho)}{\partial z_S} = \frac{\dot{z}_S}{|\vec{\rho}_S - \vec{r}_i|} + \frac{\dot{z}_S}{|\vec{\rho}_S|} - \frac{z_S}{|\vec{\rho}_S|^3}[\dot{\vec{\rho}}_S \cdot \vec{\rho}_S] - \frac{z_S - z_i}{|\vec{\rho}_S - \vec{r}_i|^3}[\dot{\vec{\rho}}_S \cdot (\vec{\rho}_S - \vec{r}_i)], \tag{3.60}$$

$$\frac{\partial F_i(\rho)}{\partial \dot{x}_S} = \frac{x_S}{|\vec{\rho}_S|} + \frac{x_S - x_i}{|\vec{\rho}_S - \vec{r}_i|}, \tag{3.61}$$

$$\frac{\partial F_i(\rho)}{\partial \dot{y}_S} = \frac{y_S}{|\vec{\rho}_S|} + \frac{y_S - y_i}{|\vec{\rho}_S - \vec{r}_i|}, \tag{3.62}$$

$$\frac{\partial F_i(\rho)}{\partial \dot{z}_S} = \frac{z_S}{|\vec{\rho}_S|} + \frac{z_S - z_i}{|\vec{\rho}_S - \vec{r}_i|} \tag{3.63}$$

gegeben sind.

3.4.2 Lösbarkeit des Gleichungssystems

Es sei angenommen, dass offene und konvexe Gebiete $U \subset \mathbb{R}^6$ existieren, sodass die lokalen Konvergenzbedingungen (1) und (2) an das Newton-Verfahren gemäss Satz 1 für das Gleichungssystem (3.49) auf U erfüllt sind, wobei als Messanordnung die in Abbildung 8 dargestellte Geometrie verwendet wird. Aus Satz 1 folgt dann, dass es für das betrachtete Gleichungssystem (3.49) mindestens eine reelle Lösung gibt.

In der vorliegenden Arbeit wird von einem mathematischen Existenzbeweis solcher Gebiete $U \subset \mathbb{R}^6$ abgesehen, da die numerische Lösungsroutine `numsolver.py` für die allermeisten Testkoordinaten $\rho_{ti} = (\vec{\rho}_{S,ti}, \dot{\vec{\rho}}_{S,ti}) = (x_{S,ti}, y_{S,ti}, z_{S,ti}, \dot{x}_{S,ti}, \dot{y}_{S,ti}, \dot{z}_{S,ti})$ zuverlässig Lösungen des Gleichungssystems (3.49) liefert - das im Programm `numsolver.py` implementierte Newton-Lösungsverfahren also konvergiert. Daher kann davon ausgegangen werden, dass ein mathematischer Existenzbeweis entsprechender Gebiete $U \subset \mathbb{R}^6$ prinzipiell möglich ist und folglich offene und konvexe Gebiete $U \subset \mathbb{R}^6$ existieren, sodass die lokalen Konvergenzbedingungen (1) und (2) an das Newton-Verfahren gemäss Satz 1 für das Gleichungssystem (3.49) auf U erfüllt sind - das Gleichungssystem (3.49) ist also reell lösbar.

3.4.3 Implementation des numerischen Lösungsverfahrens

Zur Implementierung eines Testprogramms zur Lösung des nichtlinearen Gleichungssystems (3.49) mit $n = 6$ wird eine Konfiguration der Sender- und Empfängerstationen T und $R_1, \ldots, R_6$ verwendet, wie sie in Abbildung 8 ersichtlich ist.

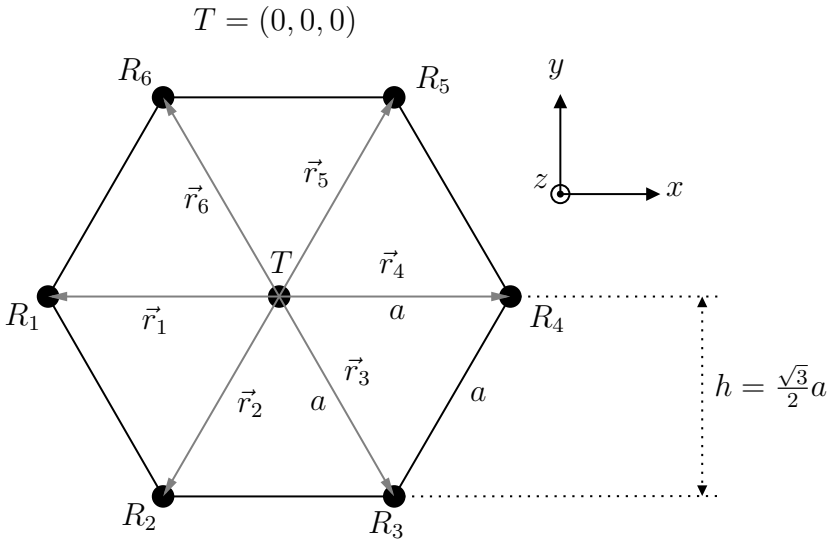


Abbildung 8: Konfiguration der Sender- und Empfängerstationen T und $R_1, \ldots, R_6$, wie sie in einem Testprogramm zur Lösung des nichtlinearen Gleichungssystems (3.11) mit $n = 6$ verwendet werden kann. Die gesamte Konfiguration befindet sich in der x-y-Ebene, wobei ein um die Senderstation T zentriertes topozentrisches Koordinatensystem gewählt wurde.

Zwecks der konkreten Implementierung des in Abschnitt 3.3.2 beschriebenen globalen Lösungsalgorithmus für nichtlineare Gleichungssysteme der Form $F(x) = 0$ in die Programmiersprache Python wurde die Funktion `scipy.optimize.root` des Python-Paketes `scipy` verwendet. Diese Funktion stellt mehrere Lösungsverfahren für nichtlineare Gleichungssysteme der Form $F(x) = 0$ zur Verfügung, welche als Grundlage das in Abschnitt 3.2.4 beschriebene mehrdimensionale Newton-Verfahren mit Line Searching oder Backtracking verwenden. Die Dokumentation dieser Python-Funktion kann online eingesehen werden [**?**].

Der Python-Code des implementierten Lösungsverfahrens gemäss Abschnitt 3.2.4 ist in Kapitel 5.5 gegeben. Das Grundprinzip dieses Testprogrammes sei nun im Folgenden skizziert. Zunächst wird die Geometrie der Messapparatur nach Abbildung 8 initialisiert, wobei der definierende Parameter durch die Seitenlänge a eines der deckungsgleichen gleichseitigen Dreiecke im Hexagon gegeben ist. Danach können dem Testprogramm sechs wahre Koordinaten eines Raumobjektes übergeben werden, also

$$\rho_t = (\vec{\rho}_{S,t}, \dot{\vec{\rho}}_{S,t}) = (x_{S,t}, y_{S,t}, z_{S,t}, \dot{x}_{S,t}, \dot{y}_{S,t}, \dot{z}_{S,t}) \in \mathbb{R}^6, \tag{3.64}$$

wobei das Subskript t für *true* steht. Aus diesen wahren Koordinaten eines Raumobjekts werden dann gemäss Gleichungen (3.50) bis (3.55) die zu erwartenden Dopplerverschiebungen $\delta f_i, i \in \{1, \ldots, 6\}$ an den Empfängerstationen berechnet. Danach werden diese Dopplerverschiebungen δf_i in das Lösungsverfahren des Gleichungsystems (3.49) eingespiesen. Ist das Lösungsverfahren nach erfolgten Iterationen erfolgreich gegen eine Lösung des Gleichungssystems konvergiert, so werden anschliessend die Differenzen des berechneten Lösungsvektors

$$\rho_c = (\vec{\rho}_{S,c}, \dot{\vec{\rho}}_{S,c}) = (x_{S,c}, y_{S,c}, z_{S,c}, \dot{x}_{S,c}, \dot{y}_{S,c}, \dot{z}_{S,c}) \in \mathbb{R}^6 \tag{3.65}$$

zum wahren Lösungsvektor ρ_t berechnet und ausgegeben. Hierbei steht das Subskript c für *calculated*. Ein Plausibilitätstest des berechneten Lösungsvektors, der unabhängig von der Kenntnis der wahren Lösung durchgeführt werden kann, besteht darin, den erhaltenen Lösungsvektor ρ_c in die Gleichung (3.49) einzusetzen und zu prüfen, ob $F(\rho_c)$ ungefähr den Nullvektor gemäss $F(\rho_c) \approx 0$ liefert. Dieser Plausibilitätstest ist allerdings bereits in der verwendete Python-Funktion `scipy.optimize.root` implementiert - erfüllt ein berechneter Lösungsvektor diesen Plausibilitätstest nicht, so wird dieser von der Funktion `scipy.optimize.root` nicht als Lösungsvektor ausgegeben.

3.4.4 Eingrenzung der Lösungsintervalle

Damit mittels ein Algorithmus zur Lösung des betrachteten nichtlinearen Gleichungssystems schneller eine Lösung gefunden werden kann, können die infrage kommenden Lösungsintervalle für die Koordinaten $\rho = (\vec{\rho}_S, \dot{\vec{\rho}}_S) \in \mathbb{R}^6$ durch physikalische Überlegungen eingeschränkt werden. Hierzu sei Abbildung 9 betrachtet. Das Intervall für die z-Position

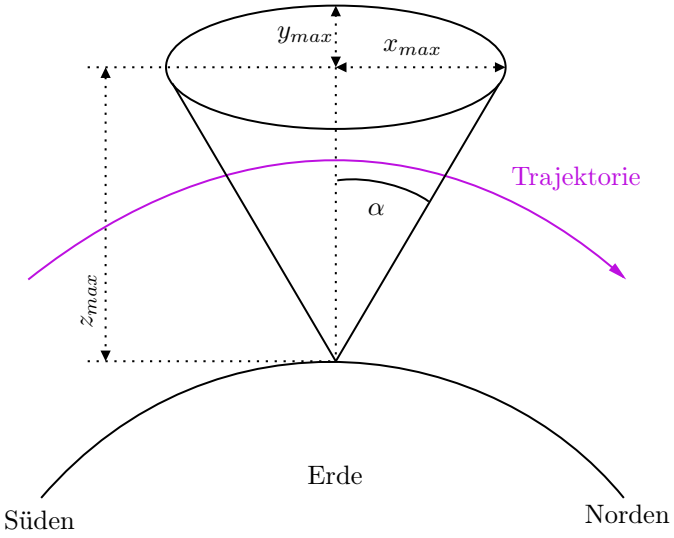


Abbildung 9: Skizze zur Einschränkung der infrage kommenden Raumobjektspositionen $\vec{\rho}_S$.

z_S lässt sich durch typische Flughöhen für Objekte in verschiedenen Erdumlaufbahnen einschränken. Typische Flughöhen von Raumobjekten in der Erdumlaufbahnen reichen von $z_{min} \doteq 160\,\text{km}$ bis ungefähr $z_{max} = 40\,000\,\text{km}$, daher ist das Lösungsintervall für die Flughöhe z_S zunächst auf das Intervall $z_S \in [1.6\text{e}5\,\text{m}, 4\text{e}7\,\text{m}]$ einzuschränken. Setzt man voraus, dass der zentrale Sender einen Sichtkegel mit Öffnungswinkel 2α hat, so ergibt sich eine Abschätzung für die Maximalwerte x_{max}, y_{max} detektierbarer x- und y-Positionen x_S, y_S eines Raumobjektes. Die Werte x_{max}, y_{max} ergeben sich aus der Beziehung

$$\tan(\alpha) = \frac{x_{max}}{z_{max}} = \frac{y_{max}}{z_{max}} \quad \Leftrightarrow \quad x_{max} = y_{max} = z_{max}\tan(\alpha). \tag{3.66}$$

Es ist zu erwarten, dass das Lösungsverfahren sensitiv auf die Länge der in Abbildung 10 ersichtlichen Basislinie a reagieren wird. Denn je kleiner die Basislinie a in Relation zu den wahren Positionskoordinaten eines Raumobjektes wird, desto weniger werden sich die gemessenen Dopplerverschiebungen an den sechs Empfängerstationen voneinander unterscheiden. Um diese erwartete Sensitivität zu berücksichtigen, werden die Werte x_{max} und y_{max} auf $x_{max} = y_{max} = a$, respektive $x_{min} = y_{min} = -a$ angesetzt, wobei für die Basislinie der Messgeometrie $a = 1\text{e}5\,\text{m} = 100\,\text{km}$ gewählt wird, was durchaus

der Grössenordnung von Distanzen zwischen bereits bestehenden Empfängerstationen auf der Erde entspricht. Aus demselben und einem zusätzlichen Grund muss auch die maximale Flughöhe z_{max} weiter eingeschränkt werden. In der vorliegenden Arbeit werden nur beobachtete Raumobjekte im *Low Earth Orbit* (LEO) betrachtet, da die Detektion von Raumobjekten in höheren Umlaufbahnen unrealistisch zu sein scheint, weil die Signalstärke I_{sig} proportional zur vierten Potenz der Flughöhe z_S eines Raumobjektes abnimmt - es gilt also $I_{sig} \propto z_S^{-4}$. Für die Amplitude E des elektrischen Feldes einer von der Sendestation emittierte Kugelwelle gilt

$$E(r) = \frac{E_0}{r} e^{-ikr}, \quad k = \frac{2\pi\nu}{c}. \tag{3.67}$$

Die Intensität I_{obj} der Kugelwelle beim Raumobjekt in der Höhe z_S ist also ungefähr gegeben durch

$$I_{obj} \approx |E(z_S)|^2 = \frac{E_0^2}{z_S^2} = \tilde{E}_0^2 \propto z_S^{-2}, \quad \tilde{E}_0 \doteq \frac{E_0}{z_S}. \tag{3.68}$$

Dass dieses Resultat nur approximativ gilt kommt daher, dass sich das Raumobjekt im Allgemeinen nicht exakt über dem Ursprung des Koordinatensystems befindet. Die von der Sendestation ausgesandte Kugelwelle wird sodann am Raumobjekt reflektiert und zurück zur Erde geworfen. Die Intensität I_{sig} des zu messenden Signals auf der Erde ist somit durch

$$I_{sig} \approx \frac{\tilde{E}_0^2}{z_S^2} = \frac{E_0^2}{z_S^4} \propto z_S^{-4} \tag{3.69}$$

gegeben. Aufgrund dieser Machbarkeitsüberlegung wird die Flughöhe vermuteter Raumobjekte in der vorliegenden Arbeit auf den LEO eingeschränkt, also $z_S \in [z_{min}, z_{max}] = [1.6\text{e}5\,\text{m}, 2.0\text{e}6\,\text{m}]$. Die Einschränkungen an die räumlichen Koordinaten $\vec{\rho}_S = (x_S, y_S, z_S)$ definieren einen Quader von möglichen Raumobjektpositionen, der zentriert über dem Ort der Sendestation «schwebt». Dieser Quader wird durch Abbildung 10 illustriert.

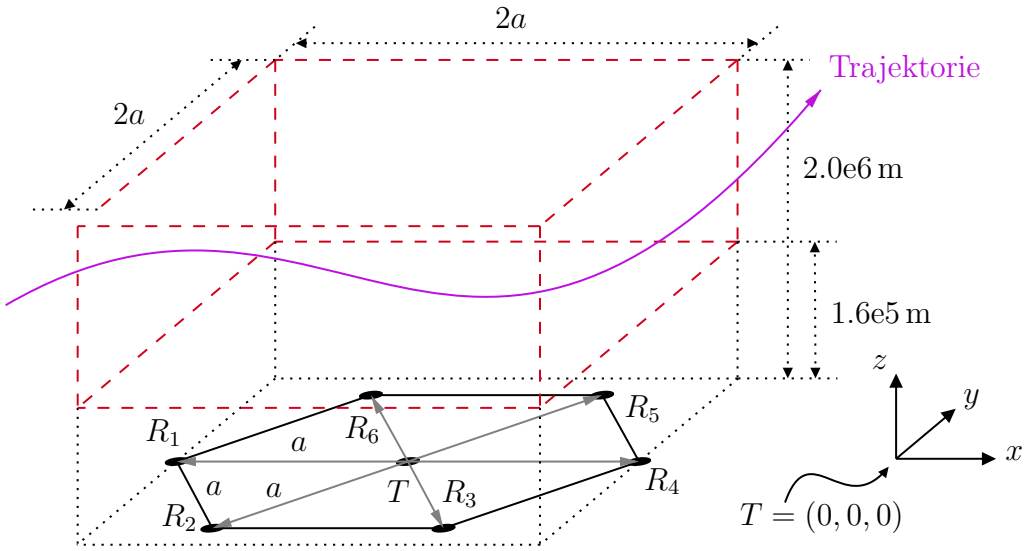


Abbildung 10: Illustration des einschränkenden Quaders (rot) für die räumlichen Koordinaten $\vec{\rho}_S$ eines Raumobjektes.

Aus einer Einschränkung möglicher Raumobjektpositionen $\vec{\rho}_S$ ergibt über das dritte Kepler'sche Gesetz natürlicherweise auch eine Einschränkung der Geschwindigkeitskompo-

nenten $\dot{\vec{\rho}}_S$. Das dritte Kepler'sche Gesetz lautet auf

$$\frac{d^3}{U^2} = \frac{\mu}{(2\pi)^2}, \tag{3.70}$$

wobei d die grosse Halbachse der durch die Relativbewegung zweier Himmelskörper der Massen M_1 und M_2 im Zweikörperproblem beschriebenen Ellipse ist, U die Umlaufszeit des einen Körpers um den Anderen bezeichnet und $\mu = G(M_1 + M_2)$ eine Konstante ist. Zur Abschätzung der Geschwindigkeit eines Raumobjekts sei nun eine Kreisbahn vorausgesetzt, sodass sich die Geschwindigkeit v mittels der Relation $v = \omega d$ berechnen lässt. Ferner sei die Masse des Raumobjekts in Relation zur Erdmasse vernachlässigbar, sodass die Näherung $\mu = GM_E$ mit M_E der Erdmasse zulässig ist und überdies der Schwerpunkt des Systems näherungsweise in den Erdmittelpunkt gelegt werden kann. Die Distanz d ergibt sich aus der Addition des Erdradius R_E mit der aktuellen Flughöhe eines Raumobjektes z_S. Mit diesen Näherungen ergibt sich aus obigem dritten Kepler'schen Gesetz

$$\frac{(2\pi)^2}{U^2} = \omega^2 = \frac{v^2}{d^2} = \frac{\mu}{d^3} \approx \frac{GM_E}{d^3} \quad \Leftrightarrow \quad v = \sqrt{\frac{GM_E}{R_E + z_S}}, \tag{3.71}$$

wobei es sich bei v um den Betrag $v = |\dot{\vec{\rho}}_S|$ des Geschwindigkeitsvektor $\dot{\vec{\rho}}_S$ handelt. Eine Abschätzung des Wertebereiches von v ergibt sich durch

$$v_{min} \doteq \sqrt{\frac{GM_E}{R_E + z_{max}}} \approx 6.9\text{e}3\,\mathrm{m\,s^{-1}}, \quad v_{max} \doteq \sqrt{\frac{GM_E}{R_E + z_{min}}} \approx 7.8\text{e}3\,\mathrm{m\,s^{-1}}. \tag{3.72}$$

Wird eine annähernd zirkuläre Umlaufbahn des Raumobjektes um die Erde angenommen, so ist die z-Komponente der Geschwindigkeit in Relation zu den x- und y-Komponenten der Geschwindigkeit als vernachlässigbar zu betrachten, um aber dennoch eine geringfügig elliptische Umlaufbahn eines Raumobjekts zu erlauben, kann $\dot{z}_S$ im Intervall $[-2.0\text{e}2\,\mathrm{m\,s^{-1}}, 2.0\text{e}2\,\mathrm{m\,s^{-1}}]$ veranschlagt werden.

Zusammengefasst sind die Eingrenzungen der Lösungsintervalle des Vektors $\rho(t) = (\vec{\rho}_S(t), \dot{\vec{\rho}}_S(t)) \in \mathbb{R}^6$ durch

$$\begin{aligned}
x_S(t) &\in [-1.0\text{e}5\,\mathrm{m}, 1.0\text{e}5\,\mathrm{m}],\\
y_S(t) &\in [-1.0\text{e}5\,\mathrm{m}, 1.0\text{e}5\,\mathrm{m}],\\
z_S(t) &\in [1.6\text{e}5\,\mathrm{m}, 2.0\text{e}6\,\mathrm{m}],\\
\dot{x}_S(t) &\in [-7.8\text{e}3\,\mathrm{m\,s^{-1}}, 7.8\text{e}3\,\mathrm{m\,s^{-1}}],\\
\dot{y}_S(t) &\in [-7.8\text{e}3\,\mathrm{m\,s^{-1}}, 7.8\text{e}3\,\mathrm{m\,s^{-1}}],\\
\dot{z}_S(t) &\in [-2.0\text{e}2\,\mathrm{m\,s^{-1}}, 2.0\text{e}2\,\mathrm{m\,s^{-1}}]
\end{aligned} \tag{3.73}$$

gegeben. Diese Einschränkungen definieren einen Quader im Ortsraum gemäss Abbildung 10 und einen weiteren Quader im Geschwindigkeitsraum. Im nachfolgend abgedruckten Quellcode des Programmes `numsolver.py` werden systematisch Punkte dieser beiden Quader als Startwerte des Newton-Verfahrens zur Findung einer Lösung des zu lösenden Gleichungssystemes (3.49) gewählt.

3.4.5 Beschreibung der Implementation

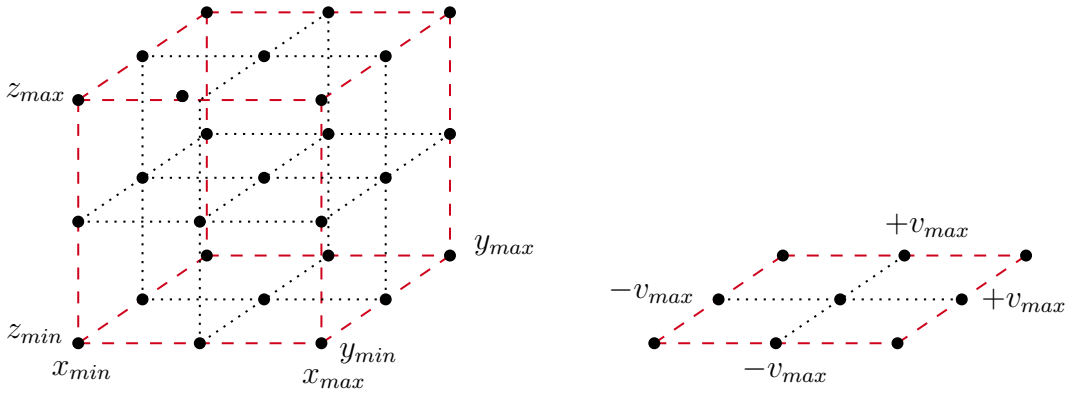
Im vorliegenden Abschnitt soll die Funktionsweise des in Kapitel 5.5 abgedruckten Programmes `numsolver.py` etwas genauer erklärt werden.

In den Zeilen 1-55 werden zunächst verschiedene Python-Pakete importiert und diverse globale Variablen werden initialisiert.

In den Zeilen 56-242 werden sodann Funktionen definiert, die im in den Zeilen 243-457 gegebenen Hauptprogramm verwendet werden.

Die in den Zeilen 60-74 definierte Funktion `eqsyst(f)` implementiert das gesamte Gleichungssystem (3.49), welches es zu lösen gilt. Hierbei ist `f` ein sechskomponentiger Vektor, der als Input der Funktion angenommen wird. Die Funktion `eqsyst(f)` gibt einen sechskomponentigen Vektor zurück, der die mittels Input `f` berechneten Komponenten des Gleichungssystems (3.49) enthält.

Die in den Zeilen 77-114 definierte Funktion `solseeker(z_vel, div)` ist das Herzstück der Lösungsroutine `numsolver.py`. Mittels dieser Funktion wird der durch die Einschränkungen (3.73) gegebene Lösungsraum auf Startwerte des Newton-Verfahrens zur Lösung des Gleichungssystems (3.49) abgetastet, sodass das Verfahren konvergiert. Diese Abtastung ist in Abbildung 11 illustriert. Die Feinheit der Abtastung kann mittels des der Funktion



Abtastung des Ortsraumes mit `div` $= 3$

Abtastung des Geschwindigkeitsraumes mit `div` $= 3$

Abbildung 11: Illustration des von der Funktion `solseeker(z_vel, div)` verwendeten Abtastungsverfahrens. Die schwarzen Punkte markieren Startwerte des Newton-Iterationsverfahrens zur Findung einer Lösung des Gleichungssystems (3.49).

`solseeker(z_vel, div)` zu übergebenden Parameters `div` eingestellt werden - je höher der ganzzahlige Wert dieses Parameters, desto höher die Feinheit der Abtastung und desto höher die Chance, eine plausible Lösung des Gleichungssystems zu finden, desto höher aber auch die benötigte Rechenzeit t_{calc}, die mit $t_{calc} \propto \texttt{div}^5$ geht. Wie in Abbildung 11 ersichtlich ist, wird nur über fünf statt sechs Koordinaten abgetastet. Der Grund hierfür ist praktischer Natur - würde über sechs Parameter abgetastet werden, so würde die benötigte Rechenzeit des Programms aufgrund der Proportionalität $t_{calc} \propto \texttt{div}^6$ stark zunehmen. Ferner wird sich die wahre z-Komponente der Geschwindigkeit eines Raumobjektes auf annähernd zirkulärem Orbit um die Erde in einem kleinen Intervall um null bewegen - daher ist die fehlende Abtastung über die z-Komponente der Raumobjektgeschwindigkeit gerechtfertigt. Werden durch diese Abtastung mehrere Lösungen des Gleichungssystems gefunden, so gibt die Funktion `solseeeker(z_vel, div)` die beste Lösung zurück - das heisst jene Lösung, welche eingesetzt in das Gleichungssystem die kleinste euklidische Norm hat. Ist $\tilde{\rho}_i = (\tilde{\rho}_{i_1}, \tilde{\rho}_{i_2}, \tilde{\rho}_{i_3}, \tilde{\rho}_{i_4}, \tilde{\rho}_{i_5}, \tilde{\rho}_{i_6})$ mit $i \in \mathbb{N}$ eine durch Abtastung des Lösungsraumes gefundene Lösung des zu lösenden Gleichungssystemes $F(\rho) = 0$, so wird diese Lösung in die Funktion $F(\rho)$ eingesetzt und der resultierende Vektor $F(\tilde{\rho}_i) = g_i \approx 0$ berechnet.

Anschliessend wird die euklidische Norm für alle Lösungen $\tilde{\rho}_i$ berechnet, das heisst

$$\|F(\tilde{\rho}_i)\| = \sqrt{\sum_{k=1}^{6} F_k(\tilde{\rho}_i)^2}, \quad i \in \mathbb{N}. \tag{3.74}$$

Mittels der Funktion `bestsol(sols)` gibt `solseeker(z_vel, div)` dann diejenige Lösung $\tilde{\rho}_i$ zurück, welche eingesetzt in das Gleichungssystem von allen gefundenen Lösungen die kleinste euklidische Norm aufweist, also am nächsten bei einer tatsächlichen Nullstelle des Gleichungssystems (3.49) liegt. Die Funktion `solseeker(z_vel, div)` nimmt zwei Parameter als Input entgegen, einen Iterations-Startwert `z_vel` für die z-Komponente der Raumobjektgeschwindigkeit und die Feinheit `div` der Abtastung. Der Wert `z_vel` wird aus obengenannten Gründen standardmässig null gesetzt.

Die in den Zeilen 117-125 definierte Funktion `bestsol(sols)` sucht aus einem Array `sols` möglicher Lösungen des Gleichungssystems den Index der besten darin enthaltenen Lösung und gibt diese ganzzahlige Zahl zurück. Die beste Lösung ist dabei so zu verstehen, dass sie die Norm der Gleichungssytem-Komponenten gemäss Gleichung (3.74) minimiert.

Die in den Zeilen 128-167 definierte Funktion `refiner1(z_vel, sol, diff, div)` tastet die nähere Umgebung einer bereits mittels der Funktion `solseeker(z_vel, div)` erhaltenen Lösung auf weitere - und bessere - Lösungen ab, indem das Lösungsverfahren für verschiedene Startwerte in der Umgebung der bereits bekannten Lösung gestartet wird. Wiederum wird in der Funktion `refiner1(z_vel, sol, diff, div)` nur über fünf anstatt sechs Parameter abgetastet - die z-Komponente der Raumobjektgeschwindigkeit wird als `z_vel` der Funktion übergeben und meist null gesetzt. Der Parameter `sol` ist ein Array und beinhaltet die mittels der Funktion `solseeker(z_vel, div)` erhaltenen ersten Lösung des Gleichungssystems. Die Grösse `diff` beschreibt die Grösse des abzutastenden Bereiches um die Lösung `sol`, wobei `div` die Anzahl der abzutastenden Punkte je Raum- und Geschwindigkeitsrichtung notiert. Wird mittels der Funktion `refiner1(z_vel, sol, diff, div)` eine bessere als die bereits bestehende Lösung `sol` gefunden, so wird diese von `refiner1(z_vel, sol, diff, div)` zurückgegeben.

Die in den Zeilen 170-207 definierte Funktion `refiner2(sol, diff, div)` macht dasselbe wie die Funktion `refiner1(z_vel, sol, diff, div)`, wobei aber nur über die drei räumlichen Koordinaten des Raumobjekts abgetastet wird - diese Ausführung dieser Funktion benötigt somit viel weniger Zeit als eine Exekution der Funktionen `solseeker(z_vel, div)` und/oder `refiner1(z_vel, sol, diff, div)`, denn die Rechenzeit ist geht hierbei mit $t_{calc} \propto \texttt{div}^3$ anstatt mit der fünften Potenz. Durch diese Funktion kann eine im räumlichen Bereich bessere Lösung in kurzer Zeit gefunden werden.

Die in den Zeilen 210-231 definierte Funktion `printer(sol, descr)` gibt die eine erhaltene Lösung `sol` auf dem Bildschirm aus. Der Parameter `descr` ist ein String, welcher der Funktion zur Beschreibung der ausgegebenen Lösung übergeben werden kann.

Die in den Zeilen 234-240 definierte Funktion `examiner(sol, residue)` wird für die Callback-Funktion der im Modul `scipy.optimize.root` implementierten Newton-Lösungsverfahren benötigt. Diese Funktion wird bei jedem Iterationsschritt des Newton-Lösungsverfahrens aufgerufen, wodurch ein Array produziert wird, welches für jeden Iterationsschritt die dazugehörigen iterierten Koordinaten enthält.

In den Zeilen 246-278 werden einige Konstanten und ferner die Geometrie der Messapparatur nach Figur 8 definiert. Die Zeilen 283-291 enthalten die Information zur Testkoordinate des Programms, woraus in den Zeilen 297-302 die zu erwartenden Dopplerverschiebungen bei den Empfängerstationen berechnet werden. In den Zeilen 305-316

werden die Grenzen der Lösungsintervalle gemäss den Einschränkungen (3.73) definiert. Sodann werden in den Zeilen 321-339 alle oben beschriebenen Funktionen in sinnvoller Reihenfolge ausgeführt. Die Zeilen 345-454 enthalten den Quellcode für die Plots 15a-15f. Diese Zeilen müssen zur Erzeugung der entsprechenden Plots unkommentiert werden. Die letzte Zeile 457 gibt schliesslich zum Schluss die benötigte Programmlaufzeit der Routine `numsolver.py` auf dem Bildschirm aus.

3.5 Resultate und Beobachtungen zum numerischen Lösungsverfahren

Das im Kapitel 3.4.5 beschriebene und im Abschnitt 5.5 abgedruckte Testprogramm soll für verschiedene Koordinatenkonfigurationen eines Raumobjekts getestet werden, wobei die Objektpositionen im LEO (*Low Earth Orbit*) angenommen werden. Die Beobachtung von passiven Objekten im MEO (*Medium Earth Orbit*) und im GEO (*Geosynchronous Earth Orbit*) mittels Dopplersignalen ist nicht realitätsnah, da hierzu enorm starke Sendestation(en) erforderlich wären, daher werden die Objektpositionen in den meisten der nachfolgend betrachteten Fällen auf den LEO beschränkt.

Das Subskript c bezeichnet im Folgenden eine durch die numerische Lösungsroutine berechnete Lösung, während das Subskript t die wahren Koordinaten eines Objekts beschreibt.

3.5.1 Testkonfigurationen innerhalb des abgedeckten Iterationsraumes

Die Testkonfiguration des numerischen Lösungsverfahrens für Raumobjektkoordinaten innerhalb[1] des abgedeckten Iterationsraumes ist gegeben durch Abbildung 12, wobei die Basislinie a der Messgeometrie zu $a = 1\text{e}5\,\text{m}$ gewählt wurde. Dabei sind die Raumobjektkoordinaten in der Form

$$\rho_{ti} = (\vec{\rho}_{S,ti}, \dot{\vec{\rho}}_{S,ti}) = (x_{S,ti}, y_{S,ti}, z_{S,ti}, \dot{x}_{S,ti}, \dot{y}_{S,ti}, \dot{z}_{S,ti}), \quad i \in \{1, \ldots, 4\} \tag{3.75}$$

gegeben, wobei die Einheiten für Positionskoordinaten in Metern und für Geschwindigkeitskoordinaten in Metern pro Sekunde angegeben sind. In der nachfolgenden Tabelle 1 sind den benutzten Testkoordinaten ρ_{ti} die Abweichungen

$$\Delta\rho_{ci} = (|x_{S,ti}-x_{S,ci}|, |y_{S,ti}-y_{S,ci}|, |z_{S,ti}-z_{S,ci}|, |\dot{x}_{S,ti}-\dot{x}_{S,ci}|, |\dot{y}_{S,ti}-\dot{y}_{S,ci}|, |\dot{z}_{S,ti}-\dot{z}_{S,ci}|) \tag{3.76}$$

der berechneten Koordinaten ρ_{ci} von den wahren Werten ρ_{ti} für $i \in \{1, \ldots, 4\}$ gegenübergestellt.

[1] Das sind Koordinaten, die innerhalb der einschränkenden Quader im Orts- und Geschwindigkeitsraum liegen, über die das zu lösende Gleichungssystem (3.49) auf Lösungen abgetastet wird. Der einschränkende Quader im Ortsraum ist illustrativ in Abbildung 10 dargestellt - den einschränkenden Quader im Geschwindigkeitsraum kann man sich in ähnlicher Weise vorstellen.

	Position (m)			Geschwindigkeit ($\mathrm{m\,s^{-1}}$)		
ρ_{t1}	2.0e4	1.0e4	1.9e6	7.1e3	−7.5e3	1.3e1
$\Delta\rho_{c1}$	2198.205	1099.102	107746.144	401.536	424.157	3.874
ρ_{t2}	−1.7e4	6.1e4	1.8e5	−7.2e3	−6.9e3	−1.2e1
$\Delta\rho_{c2}$	0.000	0.000	0.000	0.000	0.000	0.000
ρ_{t3}	−6.1e4	−4.8e4	9.0e5	−6.9e3	7.1e3	−1.8e2
$\Delta\rho_{c3}$	0.033	0.026	0.247	0.002	0.002	0.000
ρ_{t4}	1.8e4	−9.4e4	1.3e6	−7.5e3	7.3e3	−1.1e2
$\Delta\rho_{c4}$	113.944	595.049	4110.095	23.701	23.070	3.992

Tabelle 1: Testkoordinaten ρ_{ti} und Abweichungen $\Delta\rho_{ci}$ der berechneten Lösungsvektoren zu den Testkoordinaten. Die Ergebnisse $\Delta\rho_{ci}$ sind auf drei Nachkommastellen gerundet.

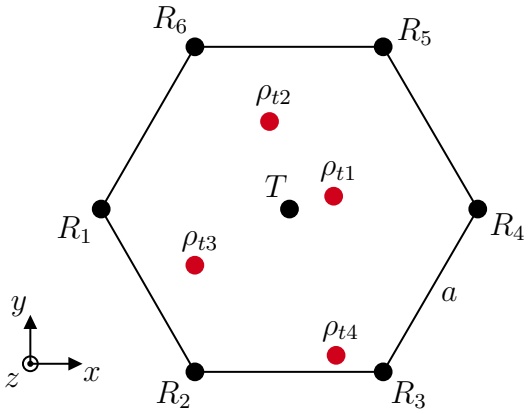


Abbildung 12: Testkonfiguration für das Programm `numsolver.py` mit Raumobjektkoordinaten innerhalb des abgedeckten Iterationsraumes. Die Sendestation T hat die Koordinaten $T = (0, 0, 0, 0, 0, 0)$.

3.5.2 Testkonfigurationen nahe des abgedeckten Iterationsraumes

Die Testkonfiguration des numerischen Lösungsverfahrens für Raumobjektkoordinaten nahe[2] des abgedeckten Iterationsraumes ist gegeben durch Abbildung 13, wobei die Basislinie a der Messgeometrie zu $a = 1\mathrm{e}5\,\mathrm{m}$ gewählt wurde. Dabei sind die Raumobjektkoordinaten in der Form

$$\rho_{ti} = (\vec{\rho}_{S,ti}, \dot{\vec{\rho}}_{S,ti}) = (x_{S,ti}, y_{S,ti}, z_{S,ti}, \dot{x}_{S,ti}, \dot{y}_{S,ti}, \dot{z}_{S,ti}), \quad i \in \{5, \ldots, 8\} \tag{3.77}$$

gegeben, wobei die Einheiten für Positionskoordinaten in Metern und für Geschwindigkeitskoordinaten in Metern pro Sekunde angegeben sind. In der nachfolgenden Tabelle 2 sind den benutzten Testkoordinaten ρ_{ti} die Abweichungen

$$\Delta\rho_{ci} = (|x_{S,ti} - x_{S,ci}|, |y_{S,ti} - y_{S,ci}|, |z_{S,ti} - z_{S,ci}|, |\dot{x}_{S,ti} - \dot{x}_{S,ci}|, |\dot{y}_{S,ti} - \dot{y}_{S,ci}|, |\dot{z}_{S,ti} - \dot{z}_{S,ci}|) \tag{3.78}$$

der berechneten Koordinaten ρ_{ci} von den wahren Werten ρ_{ti} für $i \in \{5, \ldots, 8\}$ gegenübergestellt.

[2]Das sind Koordinaten, die nahe der einschränkenden Quader im Orts- und Geschwindigkeitsraum liegen, über die das zu lösende Gleichungssystem (3.49) auf Lösungen abgetastet wird. Der einschränkende Quader im Ortsraum ist illustrativ in Abbildung 10 dargestellt - den einschränkenden Quader im Geschwindigkeitsraum kann man sich in ähnlicher Weise vorstellen.

	Position (m)			Geschwindigkeit ($\mathrm{m\,s^{-1}}$)		
ρ_{t5}	2.1e5	8.2e4	2.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c5}$	1.083	0.423	5.364	0.017	0.021	0.002
ρ_{t6}	−1.4e4	1.1e5	2.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c6}$	6.824	53.614	535.823	1.608	1.925	0.172
ρ_{t7}	−1.1e5	−7.1e4	2.1e6	−6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c7}$	9.325	6.019	88.869	0.288	0.339	0.007
ρ_{t8}	8.2e4	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c8}$	2.954	4.683	41.322	0.146	0.113	0.023

Tabelle 2: Testkoordinaten ρ_{ti} und Abweichungen $\Delta\rho_{ci}$ der berechneten Lösungsvektoren zu den Testkoordinaten. Die Ergebnisse $\Delta\rho_{ci}$ sind auf drei Nachkommastellen gerundet.

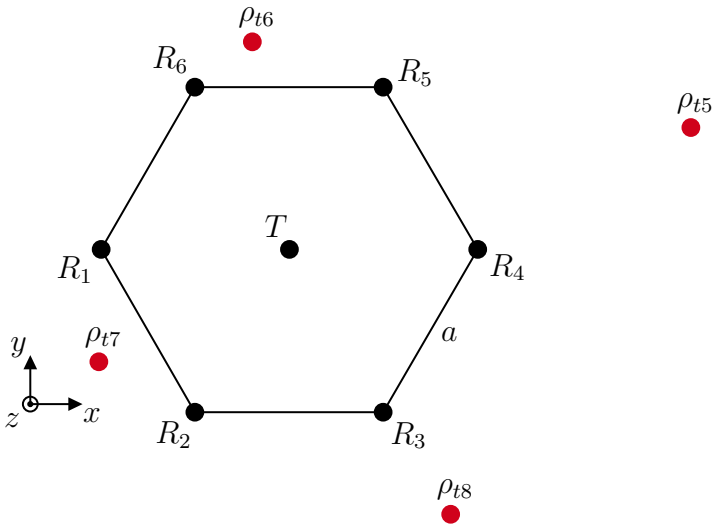


Abbildung 13: Testkonfiguration für das Programm `numsolver.py` mit Raumobjektkoordinaten nahe des abgedeckten Iterationsraumes. Die Sendestation T hat die Koordinaten $T = (0, 0, 0, 0, 0, 0)$.

3.5.3 Testkonfigurationen stark ausserhalb des abgedeckten Iterationsraumes

Die Testkonfiguration des numerischen Lösungsverfahrens für Raumobjektkoordinaten stark ausserhalb[3] des abgedeckten Iterationsraumes ist gegeben durch Abbildung 14, wobei die Basislinie a der Messgeometrie zu $a = 1\mathrm{e}5\,\mathrm{m}$ gewählt wurde. Dabei sind die Raumobjektkoordinaten in der Form

$$\rho_{ti} = (\vec{\rho}_{S,ti}, \dot{\vec{\rho}}_{S,ti}) = (x_{S,ti}, y_{S,ti}, z_{S,ti}, \dot{x}_{S,ti}, \dot{y}_{S,ti}, \dot{z}_{S,ti}), \quad i \in \{9, \ldots, 12\} \tag{3.79}$$

gegeben, wobei die Einheiten für Positionskoordinaten in Metern und für Geschwindigkeitskoordinaten in Metern pro Sekunde angegeben sind. In der nachfolgenden Tabelle 3 sind den benutzten Testkoordinaten ρ_{ti} die Abweichungen

$$\Delta\rho_{ci} = (|x_{S,ti} - x_{S,ci}|, |y_{S,ti} - y_{S,ci}|, |z_{S,ti} - z_{S,ci}|, |\dot{x}_{S,ti} - \dot{x}_{S,ci}|, |\dot{y}_{S,ti} - \dot{y}_{S,ci}|, |\dot{z}_{S,ti} - \dot{z}_{S,ci}|) \tag{3.80}$$

der berechneten Koordinaten ρ_{ci} von den wahren Werten ρ_{ti} für $i \in \{9, \ldots, 12\}$ gegenübergestellt.

[3]Das sind Koordinaten, die stark ausserhalb der einschränkenden Quader im Orts- und Geschwindigkeitsraum liegen, über die das zu lösende Gleichungssystem (3.49) auf Lösungen abgetastet wird. Der einschränkende Quader im Ortsraum ist illustrativ in Abbildung 10 dargestellt - den einschränkenden Quader im Geschwindigkeitsraum kann man sich in ähnlicher Weise vorstellen.

	Position (m)			Geschwindigkeit ($\mathrm{m\,s^{-1}}$)		
ρ_{t9}	7.2e5	2.2e5	4.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c9}$	41.532	12.690	114.336	0.193	0.234	0.042
ρ_{t10}	−1.9e5	9.1e4	5.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c10}$	20613.061	9872.572	289842.819	368.292	440.834	11.163
ρ_{t11}	−3.5e5	−7.1e5	2.1e6	6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c11}$	-	-	-	-	-	-
ρ_{t12}	8.2e5	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c12}$	-	-	-	-	-	-

Tabelle 3: Testkoordinaten ρ_{ti} und Abweichungen $\Delta\rho_{ci}$ der berechneten Lösungsvektoren zu den Testkoordinaten. Die Ergebnisse $\Delta\rho_{ci}$ sind auf drei Nachkommastellen gerundet.

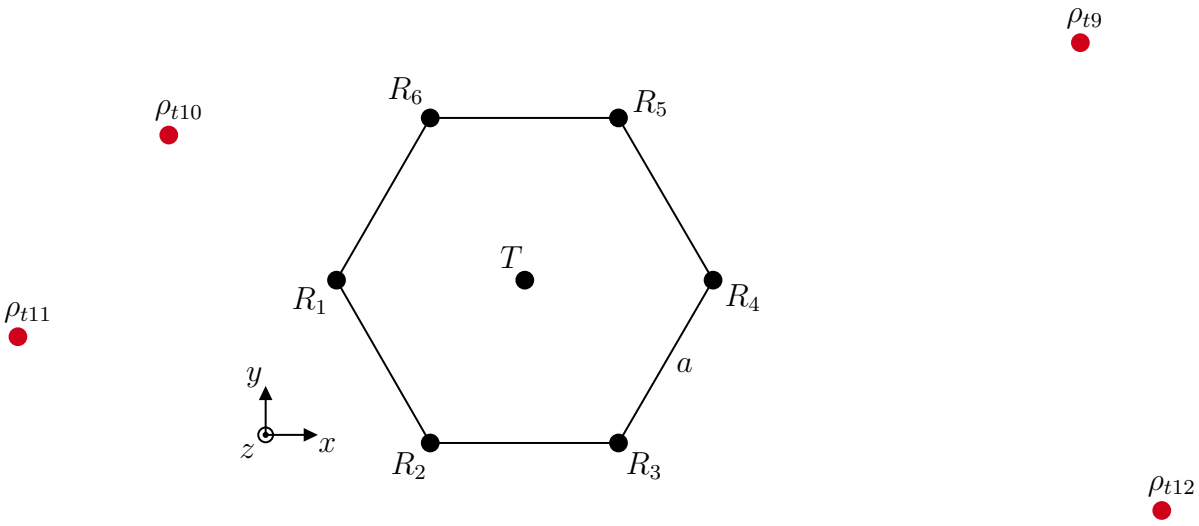


Abbildung 14: Testkonfiguration für das Programm `numsolver.py` mit Raumobjektkoordinaten stark ausserhalb des abgedeckten Iterationsraumes. Die hier abgebildeten Raumobjektpositionen sind aus Platzgründen nicht massstabsgetreu eingezeichnet. Die Sendestation T hat die Koordinaten $T = (0, 0, 0, 0, 0, 0)$.

3.5.4 Testkonfigurationen mit halbierter Basislinie der Messgeometrie

Die Resultate im vorliegenden Unterabschnitt entstammen einer wiederholten Ausführung der Lösungsroutine `numsolver.py` für die Testkonfigurationen $\rho_{ti}, i \in \{1, \ldots, 12\}$, wobei aber die Basislinie a der Messgeometrie gegenüber den Berechnungen in den Abschnitten 3.5.1 bis 3.5.3 halbiert wurde, also $a = 5\mathrm{e}4\,\mathrm{m}$ beträgt. Die nachfolgende Tabelle 4 zeigt die berechneten Ergebnisse.

	Position (m)			Geschwindigkeit ($\mathrm{m\,s^{-1}}$)		
ρ_{t1}	2.0e4	1.0e4	1.9e6	7.1e3	−7.5e3	1.3e1
$\Delta\rho_{c1}$	10197.717	5098.858	570277.334	2129.421	2249.388	17.982
ρ_{t2}	−1.7e4	6.1e4	1.8e5	−7.2e3	−6.9e3	−1.2e1
$\Delta\rho_{c2}$	0.000	0.000	0.000	0.000	0.000	0.000
ρ_{t3}	−6.1e4	−4.8e4	9.0e5	−6.9e3	7.1e3	−1.8e2
$\Delta\rho_{c3}$	1.435	1.129	10.541	0.081	0.084	0.002
ρ_{t4}	1.8e4	−9.4e4	1.3e6	−7.5e3	7.3e3	−1.1e2

$\Delta\rho_{c4}$	2.882	15.052	103.672	0.600	0.584	0.101
ρ_{t5}	2.1e5	8.2e4	2.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c5}$	66.695	26.043	329.795	1.064	1.286	0.112
ρ_{t6}	−1.4e4	1.1e5	2.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c6}$	131.671	1034.559	10300.294	30.964	37.063	3.319
ρ_{t7}	−1.1e5	−7.1e4	2.1e6	−6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c7}$	1303.349	841.253	12437.159	40.405	47.536	1.019
ρ_{t8}	8.2e4	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c8}$	78.585	124.586	1097.972	3.882	3.020	0.618
ρ_{t9}	7.2e5	2.2e5	4.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c9}$	2511.292	767.340	6916.855	11.695	14.138	2.568
ρ_{t10}	−1.9e5	9.1e4	5.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c10}$	26887.653	12877.771	381442.086	484.802	580.293	14.562
ρ_{t11}	−3.5e5	−7.1e5	2.1e6	6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c11}$	-	-	-	-	-	-
ρ_{t12}	8.2e5	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c12}$	-	-	-	-	-	-

Tabelle 4: Testkoordinaten ρ_{ti} und Abweichungen $\Delta\rho_{ci}$ der berechneten Lösungsvektoren zu den Testkoordinaten. Die Ergebnisse $\Delta\rho_{ci}$ sind auf drei Nachkommastellen gerundet. Bei den Berechnungen in vorliegender Tabelle wurde eine gegenüber den Abschnitten Abschnitten 3.5.1 bis 3.5.3 halbierte Basislänge $a/2$ vorausgesetzt.

3.5.5 Testkonfigurationen mit verdoppelter Basislinie der Messgeometrie

Die Resultate im vorliegenden Unterabschnitt entstammen einer wiederholten Ausführung der Lösungsroutine `numsolver.py` für die Testkonfigurationen $\rho_{ti}, i \in \{1, \ldots, 12\}$, wobei aber die Basislinie a der Messgeometrie gegenüber den Berechnungen in den Abschnitten 3.5.1 bis 3.5.3 verdoppelt wurde, also $a = 2\text{e}5\,\text{m}$ beträgt. Die nachfolgende Tabelle 5 zeigt die berechneten Ergebnisse.

	Position (m)			Geschwindigkeit ($\text{m}\,\text{s}^{-1}$)		
ρ_{t1}	2.0e4	1.0e4	1.9e6	7.1e3	−7.5e3	1.3e1
$\Delta\rho_{c1}$	149.288	74.644	3807155.285	26.449	27.939	25.737
ρ_{t2}	−1.7e4	6.1e4	1.8e5	−7.2e3	−6.9e3	−1.2e1
$\Delta\rho_{c2}$	0.000	0.000	0.000	0.000	0.000	0.000
ρ_{t3}	−6.1e4	−4.8e4	9.0e5	−6.9e3	7.1e3	−1.8e2
$\Delta\rho_{c3}$	0.015	0.012	0.115	0.001	0.001	0.000
ρ_{t4}	1.8e4	−9.4e4	1.3e6	−7.5e3	7.3e3	−1.1e2
$\Delta\rho_{c4}$	1199.312	6263.388	43361.588	245.840	239.301	41.856
ρ_{t5}	2.1e5	8.2e4	2.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c5}$	78004.658	30459.619	357044.055	1146.348	1385.807	130.794
ρ_{t6}	−1.4e4	1.1e5	2.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c6}$	0.432	3.394	34.133	0.102	0.122	0.011
ρ_{t7}	−1.1e5	−7.1e4	2.1e6	−6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c7}$	0.090	0.058	0.861	0.003	0.003	0.000

ρ_{t8}	8.2e4	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c8}$	0.087	0.138	1.226	0.004	0.003	0.001
ρ_{t9}	7.2e5	2.2e5	4.1e6	6.7e3	−8.1e3	3.3e2
$\Delta\rho_{c9}$	2323.276	709.902	6401.861	10.801	13.058	2.376
ρ_{t10}	−1.9e5	9.1e4	5.2e6	−6.6e3	−7.9e3	−1.2e1
$\Delta\rho_{c10}$	6.775	3.245	92.694	0.118	0.141	0.004
ρ_{t11}	−3.5e5	−7.1e5	2.1e6	6.8e3	8.0e3	−1.8e2
$\Delta\rho_{c11}$	0.002	0.004	0.005	0.000	0.000	0.000
ρ_{t12}	8.2e5	−1.3e5	2.3e6	8.1e3	−6.3e3	0.1e2
$\Delta\rho_{c12}$	0.005	0.001	0.006	0.000	0.000	0.000

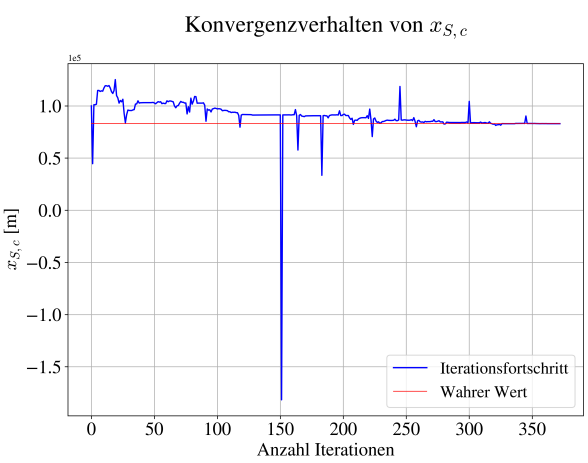
Tabelle 5: Testkoordinaten ρ_{ti} und Abweichungen $\Delta\rho_{ci}$ der berechneten Lösungsvektoren zu den Testkoordinaten. Die Ergebnisse $\Delta\rho_{ci}$ sind auf drei Nachkommastellen gerundet. Bei den Berechnungen in vorliegender Tabelle wurde eine gegenüber den Abschnitten Abschnitten 3.5.1 bis 3.5.3 verdoppelte Basislänge $2a$ vorausgesetzt.

3.5.6 Konvergenzverhalten bei initialer Iteration

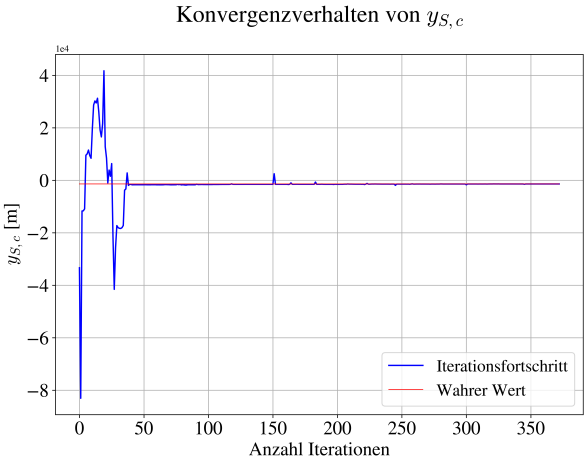
Das Konvergenzverhalten einer initialen mehrdimensionalen Newton-Iteration zur Findung einer Lösung des nichtlinearen Gleichungssystems (3.49) soll nachfolgend mittels einer Testkoordinate innerhalb des abgedeckten Iterationsraumes untersucht werden. Hierbei wird die willkürlich gewählte Testkoordinate

$$\begin{aligned}\rho_{t13} &= (x_{S,t13}, y_{S,t13}, z_{S,t13}, \dot{x}_{S,t13}, \dot{y}_{S,t13}, \dot{z}_{S,t13}) \\ &= (8.3\text{e}4, -1.4\text{e}3, 1.9\text{e}5, 7.8\text{e}3, -6.9\text{e}3, -1.1\text{e}2)\end{aligned} \tag{3.81}$$

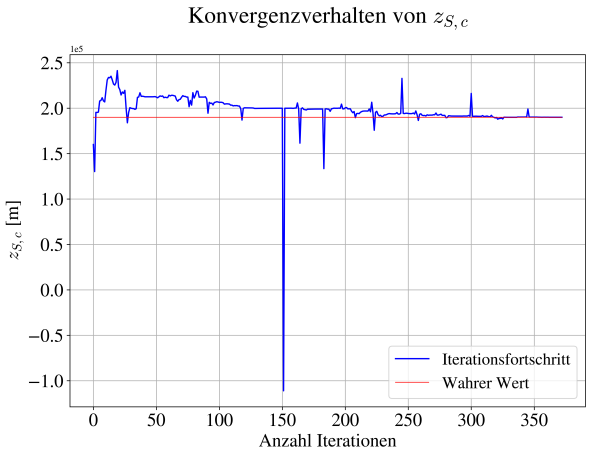
als Input in die Lösungsroutine `numsolver.py` gegeben. Das resultierende Konvergenzverhalten der initialen Iteration für alle sechs Raumobjektkoordinaten ist in den Abbildungen 15a bis 15f ersichtlich.



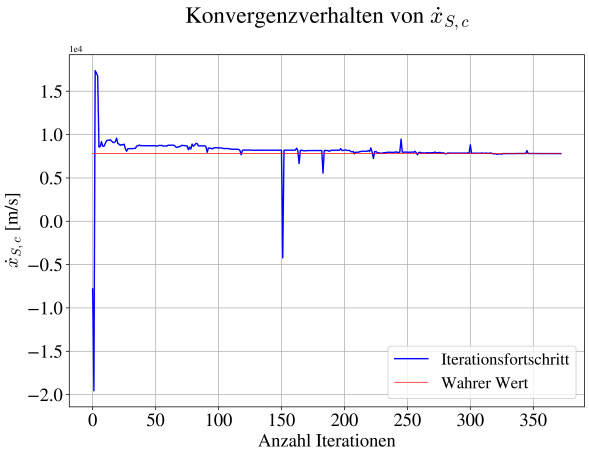
(a) Konvergenzverhalten der Koordinate $x_{S,c}$.



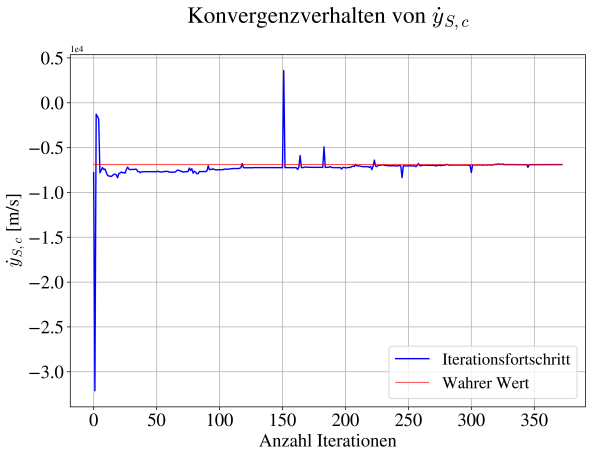
(b) Konvergenzverhalten der Koordinate $y_{S,c}$.



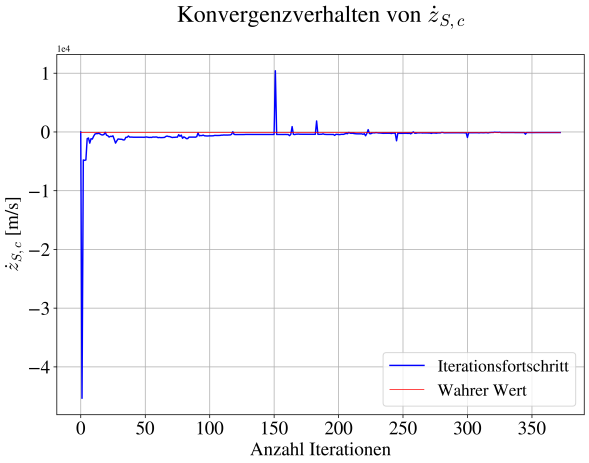
(c) Konvergenzverhalten der Koordinate $z_{S,c}$.



(d) Konvergenzverhalten der Koordinate $\dot{x}_{S,c}$.



(e) Konvergenzverhalten der Koordinate $\dot{y}_{S,c}$.



(f) Konvergenzverhalten der Koordinate $\dot{z}_{S,c}$.

Abbildung 15: Konvergenzverhalten des im Programm `numsolver.py` implementierten Newton-Iterationsverfahrens bei der initialen Lösungsfindung.

Eine Gegenüberstellung der benutzten Testkoordinate ρ_{t13} und der Abweichung $\Delta\rho_{c13}$ für die im initialen Iterationsverfahren berechnete Koordinate ρ_{c13} ist in der nachfolgenden Tabelle 6 zu finden.

Betrachtet man das in den Abbildungen 15a bis 15f dargestellte Konvergenzverhalten des initialen Iterationsverfahrens, so fällt auf, dass das Verfahren relativ langsam gegen eine Nullstelle von $F(\rho) = 0$ konvergiert - über 350 Iterationsschritte sind hierfür nötig. Das liegt vermutlich daran, dass die Funktion $F(\rho)$ in der Nähe einer Lösung von $F(\rho) = 0$ hochgradig nichtlinear ist und deshalb vergleichsweise viele Iterationsschritte notwendig sind, um in den Linearitätsbereich der Funktion $F(\rho)$ um eine Nullstelle $\tilde{\rho} : F(\tilde{\rho}) = 0$ zu gelangen. Die Grafiken wurden erzeugt, indem für die initiale Iteration das Broyden-Verfahren des Python-Moduls `scipy.optimize.root` verwendet wurde, da dieses über eine Callback-Möglichkeit verfügt. Das heisst, dass für jeden Iterationsschritt der entsprechend iterierte Lösungsvektor abgefragt werden kann. Normalerweise wird aber im Python-Modul `scipy.optimize.root` eine Modifikation der Powell-Hybrid-Methode verwendet, welche in der Regel schneller konvergiert als das Broyden-Verfahren, jedoch über keine Callback-Möglichkeit verfügt. Das Broyden-Verfahren ist ein Quasi-Newton-Verfahren,

	Position (m)			Geschwindigkeit ($\mathrm{m\,s^{-1}}$)		
ρ_{t13}	8.3e4	−1.4e3	1.9e5	7.8e3	−6.9e3	−1.1e2
$\Delta\rho_{c13}$	0.037	0.001	0.046	0.002	0.002	0.001

Tabelle 6: Testkoordinaten ρ_{t13} und Abweichung $\Delta\rho_{c13}$ des inital berechneten Lösungsvektors zu der Testkoordinate. Die Ergebnis $\Delta\rho_{c13} = (|x_{S,t13} - x_{S,c13}|, |y_{S,t13} - y_{S,c13}|, |z_{S,t13} - z_{S,c13}|, |\dot{x}_{S,t13} - \dot{x}_{S,c13}|, |\dot{y}_{S,t13} - \dot{y}_{S,c13}|, |\dot{z}_{S,t13} - \dot{z}_{S,c13}|)$ ist auf drei Nachkommastellen gerundet.

basiert also auf dem in Abschnitt 3.2.4 geschilderten Verfahren und verfügt gemäss der SciPy-Community über Backtracking und Line Searching [**?**].

3.5.7 Generelle Erkenntnisse zum numerischen Lösungsverfahren

Eine Betrachtung der Ergebnisse der Routine `numsolver.py` für verschiedene Testkonfigurationen liefert die Erkenntnis, dass generell genauere Ergebnisse erzielt werden, je näher die Testkonfiguration räumlich dem Ursprung des Koordinatensystems ist. Ferner gewinnt man die Erkenntnis, dass `numsolver.py` für eine verdoppelte Basislinie a der Messgeometrie generell die besseren Ergebnisse liefert als für eine halbierte Basislinie. Diese Beobachtungen können dadurch plausibilisiert werden, dass die Unterschiede in den gemessenen Dopplerdaten an den verschiedenen Empfängerorten desto grösser werden, je näher die Testkonfiguration räumlich dem Ursprung des Koordinatensystems ist. Eine grössere Unterschiedlichkeit in den Dopplerdaten liefert erwartungsgemäss eine bessere Determination einer Lösung und folglich bessere und eindeutigere Konvergenzeigenschaften des Newton-Verfahrens - mutmasslich insbesondere dann, wenn eine realistische Situation mit Messunsicherheiten und Messrauschen auf den Dopplersignalen vorliegt. Wie bereits erwähnt, wird die Lösung des Gleichungssystems (3.49) in der vorliegenden Arbeit idealisiert durchgeführt, das heisst ohne Messrauschen und Messunsicherheiten auf den Dopplersignalen.

Dass die Routine `numsolver.py` in den seltensten Fällen eine exakte Lösung des Gleichungssystems (3.49) liefert, kann mehrere Gründe haben. Erstens hat die verwendeten Python-Funktion `scipy.optimize.root` eine gewisse Toleranz ungleich null, die zur Termination des Lösungsverfahren unterschritten werden muss. Ferner kann zweitens der Lösungsraum eine sehr kuriose Geometrie aufweisen, sodass es in unmittelbarer Nähe einer globalen Nullstelle von (3.49) viele lokale Nullstellen geben kann. Als dritte Ursache der Ungenauigkeit kann das Lösungsverfahren `scipy.optimize.root` in einer solchen lokalen Nullstelle landen, wenn der Startwert des Lösungsverfahren nicht nahe genug bei einer globalen Nullstelle gewählt wird. Zu guter Letzt weisen viertens Computer eine begrenzte Maschinengenauigkeit auf, dies kann bei sehr genauen Berechnungen eine Rolle spielen - in der hier vorliegenden Anwendung scheint dies aber unerheblich zu sein, da in einigen Fällen ungeachtet der Maschinengenauigkeit exakte[4] Resultate erzielt werden. Manche der obigen ungenauen Resultaten können verbessert werden, indem die Abtastungsfeinheiten `div` der Funktionen `solseeker(z_vel, div)`, `refiner1(z_vel, sol, diff, div)` und/oder `refiner2(sol, diff, div)` erhöht werden - das geht aber, wie oben erwähnt, einher mit einem starken Anstieg der Rechenzeit.

[4]Exakt im Sinne davon, dass die Lösungsroutine auf drei Nachkommastellen gerundet das exakte Resultat ausgibt - also gewissermassen die Testkoordinaten wieder zurückgibt, sodass $\rho_t = \rho_c$.

4 Fazit

Die Methode 1 mit einem Sender um einem Empfänger scheint zwar zunächst attraktiv zu sein, ist jedoch aufgrund der erforderlichen speziellen Geometrie des Senders nicht anwendbar auf die bereits bestehenden Einrichtungen in Europa - nichtsdestotrotz liefert diese Methode, wie Richards zeigt, erstaunlich genaue Resultate für die Bahnelemente von Raumobjekten [**?**, S.1732]. In der vorliegenden Arbeit wurde jedoch aufgrund des Fehlens passender Sendestationen von einer konkreten Simulation dieser Methode abgesehen. Eine in dieser Arbeit durchgeführte rudimentäre Simulation der Methode 2 mit einem Sender und sechs Empfängern zeigt, dass das theoretisch hergeleitete Verfahren prinzipiell auf die bestehenden Einrichtungen in Europa anwendbar ist und zumindest für Objekte im LEO plausible Resultate liefern kann - sofern die zugrunde liegenden Dopplerdaten nur mit vernachlässigbarem Messrauschen belegt sind. Ein grosser Vorteil der Methode 2 gegenüber der Methode 1 ist dadurch gegeben, dass zur vollständigen Bahnbestimmung eines Raumobjektes nur Dopplerdaten einer einzigen Epoche notwendig sind. Mit Dopplerdaten vieler Epochen kann daher bereits eine Bahnverbesserung vorgenommen werden - was vielversprechende Resultate liefern könnte. In einem weiteren Schritt könnte dies aufbauend auf der in vorliegender Arbeit durchgeführten Simulation untersucht und geprüft werden. Ferner bleibt eine Untersuchung des Einflusses von nicht vernachlässigbarem Messrauschen und nicht vernachlässigbaren Messunsicherheiten auf die Konvergenzeigenschaften des Lösungsverfahrens nach Methode 2 weiterführender Arbeit vorbehalten.

5 Anhang

5.1 Spezielle Relativitätstheorie

5.1.1 Notation

Die Spezielle Relativitätstheorie (SR) kann mittels sogenannten Vierervektoren x^μ formuliert werden. Ein Vierervektor ist definiert durch

$$x^\mu = (x^0 = ct, x^k) = (x^0, x^1, x^2, x^3) = (x^0, \vec{x}), \tag{5.1}$$

wobei c die Lichtgeschwindigkeit und t die Zeit in einem Inertialsystem bezeichnet. Griechische Buchstaben $(\mu, \nu, \dots)$ werden für die Komponenten von Vierervektoren verwendet, lateinische Buchstaben $(i, j, \dots)$ für gewöhnliche Vektoren des euklidischen Raumes.

Eine Matrix A wird als $A_{\mu\nu}$ dargestellt, wobei die Einstein'sche Summenkonvention angewandt wird. Eine Summation ist also stets über zwei identische Indizes zu verstehen, wovon einer tiefliegend und der andere hochliegend sein muss. Beispielsweise kann die Multiplikation einer $n \times m$-Matrix A mit einem $n \times 1$-Vektor v durch

$$w = Av \quad \Leftrightarrow \quad w^\mu = \sum_{\nu=1}^{m} A^\mu{}_\nu v^\nu \equiv A^\mu{}_\nu v^\nu \tag{5.2}$$

dargestellt werden, wobei

(1) die Summation immer über zwei gleiche Indizes läuft, von denen einer tief- und der andere hochgestellt ist,

(2) $(A^\mu{}_\nu)^T = (A^\nu{}_\mu)$ gilt,

(3) $(A^\mu{}_\nu)^{-1} = (A_\mu{}^\nu)$ gilt, sowie

(4) Zeilennummern oben positioniert sind und Spaltennummern unten stehen.

5.1.2 Grundlagen

Ein Inertialsystem ist ein Bezugssystem, welches sich relativ zum Fixsternhimmel mit keiner oder konstanter Geschwindigkeit bewegt, also relativ zum Fixsternhimmel unbeschleunigt ist. Das Relativitätsprinzip von Galilei besagt nach Torsten Fliessbach, dass (1) alle Inertialsysteme gleichwertig[5] sind und (2) die Newton'schen Axiome in allen Inertialsystemen gelten [**?**, S.333]. Das Relativitätsprinzip von Einstein hingegen besagt gemäss Fliessbach, dass (1) alle Inertialsysteme gleichwertig sind und (2) die Lichtgeschwindigkeit in allen Inertialsystemen isotrop konstant und gleich c ist [**?**, S.335].

Die beobachtbaren Eigenschaften des Lichts sind nicht mit dem Galilei'schen Relativitätsprinzip (1) und (2) kompatibel, wohl aber mit dem Einstein'schen Relativitätsprinzip (1) und (2). Matthias Blau gibt eine Möglichkeit an, die korrekte Transformationsvorschrift einer physikalischen Entität von einem Inertialsystem in ein anderes Inertialsystem zu finden, nämlich die Invarianz des Wellenoperators

$$\Box = -\frac{1}{c^2}\frac{\partial^2}{\partial t^2} + \Delta = \eta^{\mu\nu}\partial_\mu\partial_\nu \tag{5.3}$$

[5] Das heisst, in allen Inertialsystemen gelten dieselben physikalischen Gesetzmässigkeiten.

unter der Transformation $x\mu \to \bar{x}^\mu$ zu fordern, wobei $\Delta = \vec{\nabla}^2$ der Laplace-Operator ist [**?**, S.7]. Da Licht durch die Wellengleichung - die mittels des Wellenoperators $\Box$ ausgedrückt werden kann - beschrieben wird, ist dies unter einer Betrachtung von Prinzip (2) eine plausible Forderung. Diese geforderte Invarianz führt als Transformationsvorschrift von physikalischen Entitäten zu den Poincaré-Transformationen, beziehungsweise zu den Lorentz-Transformationen.

5.1.3 Minkowski-Raum

In Analogie zum normalen, euklidischen Raum, ist der Minkowski-Raum ein reeller Raum $\mathbb{R}^4$, ausgestattet mit der Minkowski-Metrik d_M. Die Minkowski-Metrik ist definiert durch

$$d_M : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}, \quad (x^\mu, x^\nu) \mapsto \eta_{\mu\nu} x^\mu x^\nu, \tag{5.4}$$

wobei $\eta_{\mu\nu} = \mathrm{diag}(-1, +1, +1, +1)$ («mostly-plus» Konvention) der Minkowski-Tensor mit $\eta_{\mu\nu}\eta^{\nu\kappa} = \delta_\mu{}^\kappa$ ist. Der Minkowski-Raum ist sodann als Tupel $(\mathbb{R}^4, d_M)$ gegeben.

Das Minkowski-Linienelement $\mathrm{d}s^2$ im Minkowski-Raum ist nach Blau durch

$$\mathrm{d}s^2 = \eta_{\mu\nu}\, \mathrm{d}x^\mu\, \mathrm{d}x^\nu \tag{5.5}$$

gegeben [**?**, S.14].

Der Gradient ∂_μ im Raum der Vierervektoren ist ein sogenannter kovarianter Lorentz-Vektor. Es gilt gemäss obiger Notation

$$\partial_\mu = \frac{\partial}{\partial x^\mu} = \left(\frac{\partial}{\partial x^0}, \frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2}, \frac{\partial}{\partial x^3} \right) = \left(\frac{1}{c}\frac{\partial}{\partial t}, \vec{\nabla} \right). \tag{5.6}$$

Für den entsprechenden kontravarianten Lorentz-Vektor ∂^μ gilt

$$\partial^\mu = \eta^{\mu\nu}\partial_\nu = \left(\frac{1}{c}\frac{\partial}{\partial t}, -\vec{\nabla} \right). \tag{5.7}$$

5.1.4 Poincaré- und Lorentz-Transformationen

Eine Transformationsvorschrift

$$L : (\mathbb{R}, d_M) \to (\mathbb{R}, d_M), \quad x^\mu \mapsto \bar{x}^\mu = L^\mu{}_\nu x^\nu + b^\mu, \quad b_\mu = \text{const.} \in \mathbb{R}^4 \tag{5.8}$$

ist nach Blau genau dann eine Poincaré-Transformation, wenn L das Minkowski-Linienelement $\mathrm{d}s$ invariant lässt [**?**, S.14]:

$$\eta_{\mu\nu}\bar{x}^\mu\bar{x}^\nu = \eta_{\mu\nu}x^\mu x^\nu \quad \Leftrightarrow \quad \eta_{\mu\nu}L^\mu{}_\kappa L^\nu{}_\delta = \eta_{\kappa\delta} \quad \Leftrightarrow \quad L^T\eta L = \eta. \tag{5.9}$$

Diese Transformationen beschreiben Transformationen zwischen Inertialsystemen von beliebigen, in Vierervektoren ausgedrückten physikalischen Entitäten. Die Invarianz des Minkowski-Linienelementes $\mathrm{d}s$ und die Invarianz des Wellenoperators $\Box$ sind gemäss Blau äquivalent [**?**, S.14].

Es seien zwei Intertialsysteme Σ und $\bar{\Sigma}$ gegeben, wobei sich das Inertialsystem $\bar{\Sigma}$ relativ zu Σ mit der Geschwindigkeit $\vec{v} = (v, 0, 0)$ bewege. Ziel ist es nun, eine Transformation zwischen den Inertialsystemen zu finden, welche die Einstein'schen Axiome (1) und (2) erfüllt. Die Erfüllung der Einstein'schen Axiome ist äquivalent mit der Invarianz des Wellenoperators $\Box$ unter den gesuchten Transformationen. Gemäss Ulrich Straumann ergeben sich aus den Einstein'schen Axiomen die folgenden Forderungen an die gesuchte Transformationsvorschrift [**?**, S.9]:

(1) Aus dem Einstein'schen Axiom (1) folgt, dass die gesuchten Transformationsgleichungen linear sein müssen, denn wenn es quadratische Terme in Ort $\vec{x}$ und Zeit t geben würde, so würden Terme, welche Ableitungen nach Ort oder Zeit enthalten, noch von Ort und Zeit abhängig sein. Aufgrund der geforderten Gleichwertigkeit aller Inertialsysteme ist dies jedoch unzulässig - denn Gleichwertigkeit heisst Gleichförmigkeit derselben physikalischen Grundgesetze.

(2) Aus dem Einstein'schen Axiom (2) ergibt sich die Forderung der konstanten Lichtgeschwindigkeit und

(3) ferner die Bedingung, dass die gesuchte Transformationsvorschrift für $v \ll c$ in die Galileitransformation übergeht.

Verwendet man die Notation $x^\mu = (ct, x^1, x^2, x^3)$ für Koordinaten im Inertialsystem Σ und $\bar{x}^\mu = (c\bar{t}, \bar{x}^1, \bar{x}^2, \bar{x}^3)$ für Koordinaten im Inertialsystem $\bar{\Sigma}$, so ergibt sich aus der dritten Forderung die Bedingung

$$(x^1)^2 + (x^2)^2 + (x^3)^2 = c^2 \cdot t^2 \quad \Leftrightarrow \quad (\bar{x}^1)^2 + (\bar{x}^2)^2 + (\bar{x}^3)^2 = c \cdot \bar{t}^2, \tag{5.10}$$

indem man Straumann folgend den Ort und die Position eines Lichtblitzes betrachtet, der sich Kugelförmig ausbreitet [**?**, S.10]. Aus der ersten und zweiten Bedingung ergibt sich der folgende Ansatz für die gesuchte Koordinatentransformation:

$$\bar{x}^\mu = \begin{pmatrix} c\bar{t} \\ \bar{x}^1 \\ \bar{x}^2 \\ \bar{x}^3 \end{pmatrix} = \begin{pmatrix} c(Bx^1 + Dt) \\ A(x^1 - vt) \\ x^2 \\ x^3 \end{pmatrix}, \quad A, B, D \in \mathbb{R}. \tag{5.11}$$

Setzt man diesen Ansatz in die Bedingungsgleichungen (5.10) ein, ergeben sich die Koeffizienten A, B und D zu

$$A = \gamma, \quad B = -\frac{\beta}{c}\gamma, \quad D = \gamma, \tag{5.12}$$

wobei der Beta-Faktor und der Gamma-Faktor durch

$$\beta = \frac{v}{c}, \quad \gamma = \frac{1}{\sqrt{1 - \beta^2}} \tag{5.13}$$

definiert sind. Damit lautet die Transformationgleichung

$$\bar{x}^\mu = \begin{pmatrix} c\bar{t} \\ \bar{x}^1 \\ \bar{x}^2 \\ \bar{x}^3 \end{pmatrix} = \begin{pmatrix} -\beta\gamma x^1 + c\gamma t \\ \gamma(x^1 - vt) \\ x^2 \\ x^3 \end{pmatrix} = \begin{pmatrix} \gamma ct - \beta\gamma x^1 \\ -\beta\gamma ct + \gamma x^1 \\ x^2 \\ x^3 \end{pmatrix}, \tag{5.14}$$

wobei $v = \beta c$ verwendet wurde. Die definierte Transformation $\bar{x}^\mu = L^\mu{}_\nu x^\nu$ kann mittels des (1,1)-Tensors $L^\mu{}_\nu$ - welcher im vorliegenden Fall in Matrixform geschrieben werden kann - ausgeschrieben durch

$$\bar{x}^\mu = \begin{pmatrix} c\bar{t} \\ \bar{x}^1 \\ \bar{x}^2 \\ \bar{x}^3 \end{pmatrix} = L^\mu{}_\nu x^\nu = \underbrace{\begin{pmatrix} \gamma & -\beta\gamma & 0 & 0 \\ -\beta\gamma & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{= L^\mu{}_\nu} \underbrace{\begin{pmatrix} ct \\ x^1 \\ x^2 \\ x^3 \end{pmatrix}}_{= x^\nu} = \begin{pmatrix} \gamma ct - \beta\gamma x^1 \\ -\beta\gamma ct + \gamma x^1 \\ x^2 \\ x^3 \end{pmatrix} \tag{5.15}$$

repräsentiert werden. Es kann gezeigt werden, dass $L^{\mu}{}_{\nu}$ die Relation

$$\eta_{\mu\nu} L^{\mu}{}_{\kappa} L^{\nu}{}_{\delta} = \eta_{\kappa\delta} \tag{5.16}$$

erfüllt, und somit eine Lorentz-Transformation sein muss. Mit der durch $L \doteq L^{\mu}{}_{\nu}$ definierten speziellen Lorentz-Transformation in den Koordinaten $(x^0 = ct, x^1)$ kann nach Christoph Greub jede beliebige Transformation $\mathcal{T}$ zwischen zwei Inertialsystemen dargestellt werden, indem Rotationen R in den Koordinaten $\vec{x} = (x^1, x^2, x^3)$, Translationen T in den Koordinaten $x^{\mu} = (x^0 = ct, x^1, x^2, x^3)$ und spezielle Lorentz-Transformationen L in den Koordinaten $(x^0 = ct, x^1)$ hintereinander ausgeführt werden [**?**, S.137-139]. Es gilt also für eine beliebige Transformation zwischen zwei Inertialsystemen die Definitionsgleichung

$$\mathcal{T} = L \circ R \circ T \quad \Leftrightarrow \quad \bar{x}^{\mu} = \mathcal{T}(x^{\mu}) = (L \circ T \circ R)(x^{\mu}) = L(R(T(x^{\mu}))). \tag{5.17}$$

5.1.5 Eigenzeit und Eigenraum

Es sei mit x^{μ} und $\bar{x}^{\mu}$ dieselbe Weltlinie eines massiven Teilchens in verschiedenen Inertialsystemen IS und $\overline{\text{IS}}$ beschrieben. Im bewegten System $\overline{\text{IS}}$ nimmt das Differential des Vierervektors $\bar{x}^{\mu}$ die Form $\mathrm{d}\bar{x}^{\mu} = (c\,\mathrm{d}\bar{t}, 0, 0, 0)$ an, somit schreibt sich folglich das Minkowski-Linienelement zu $\mathrm{d}\bar{s}^{\,2} = \eta_{\mu\nu}\,\mathrm{d}\bar{x}^{\mu}\,\mathrm{d}\bar{x}^{\nu} = -c^2\,\mathrm{d}\bar{t}^{\,2} \doteq -c^2\,\mathrm{d}\tau^2$. Es gilt aufgrund der Invarianz unter Lorentz-Transformationen

$$\mathrm{d}\bar{s} = \mathrm{d}s \quad \Leftrightarrow \quad -c^2\mathrm{d}\tau^2 = -c^2\mathrm{d}t^2 + (\mathrm{d}x^1)^2 + (\mathrm{d}x^2)^2 + (\mathrm{d}x^3)^2 = -c^2(\mathrm{d}t)^2 + (\mathrm{d}\vec{x}\,)^2. \tag{5.18}$$

Für ein Differential $(\mathrm{d}q)^n$ gilt stets die Schreibweise $\mathrm{d}q^n \equiv (\mathrm{d}q)^n, n \in \mathbb{N}$. Daraus ergibt sich mit

$$\vec{v}\,(t) = \frac{\mathrm{d}\vec{x}\,(t)}{\mathrm{d}t} \quad \Rightarrow \quad \mathrm{d}\tau^2 = \mathrm{d}t^2 - \frac{(\mathrm{d}\vec{x}\,)^2}{c^2} \quad \Leftrightarrow \quad \mathrm{d}\tau = \mathrm{d}t\,\sqrt{1 - \frac{\vec{v}\,(t)^2}{c^2}} = \gamma(\vec{v}\,)^{-1}\mathrm{d}t, \tag{5.19}$$

wobei $\gamma(\vec{v}\,) = (1 - {}^{\vec{v}\,(t)^2}\!/_{c^2})^{-1/2} = (1 - \beta(\vec{v}\,)^2)^{-1/2}$ der Gamma-Faktor ist. Das Integral

$$\tau_{PQ} = \int_P^Q \mathrm{d}\tau = \int_{t_P}^{t_Q} \sqrt{1 - \frac{\vec{v}\,(t)^2}{c^2}}\,\mathrm{d}t \tag{5.20}$$

heisst dann gemäss Blau Eigenzeit [**?**, S.21].

In analoger Weise nehme das Differential des Vierervektors $\bar{x}^{\mu}$ im bewegten System $\overline{\text{IS}}$ die Form $\mathrm{d}\bar{x}^{\mu} = (0, \mathrm{d}\bar{x}^1, \mathrm{d}\bar{x}^2, \mathrm{d}\bar{x}^3)$ an. Somit schreibt sich folglich das Minkowski-Linienelement zu $\mathrm{d}\bar{s}^{\,2} = \eta_{\mu\nu}\,\mathrm{d}\bar{x}^{\mu}\,\mathrm{d}\bar{x}^{\nu} = (\mathrm{d}\bar{x}^1)^2 + (\mathrm{d}\bar{x}^2)^2 + (\mathrm{d}\bar{x}^3)^2 = (\mathrm{d}\bar{\vec{x}}\,)^2$. Es gilt aufgrund der Invarianz unter Lorentz-Transformationen

$$\mathrm{d}\bar{s} = \mathrm{d}s \quad \Leftrightarrow \quad (\mathrm{d}\bar{\vec{x}}\,)^2 = -c^2\mathrm{d}t^2 + (\mathrm{d}x^1)^2 + (\mathrm{d}x^2)^2 + (\mathrm{d}x^3)^2 = -c^2(\mathrm{d}t)^2 + (\mathrm{d}\vec{x}\,)^2. \tag{5.21}$$

Daraus ergibt sich mit

$$\vec{v}\,(t) = \frac{\mathrm{d}\vec{x}\,(t)}{\mathrm{d}t} \quad \Rightarrow \quad \mathrm{d}\bar{s}^{\,2} = -c^2\,\mathrm{d}t^2 + (\mathrm{d}\vec{x}\,)^2 \quad \Leftrightarrow \quad \mathrm{d}\bar{s} = \mathrm{d}t\,\sqrt{\vec{v}\,(t)^2 - c^2}. \tag{5.22}$$

Das Integral

$$s_{PQ} = \int_P^Q \mathrm{d}\bar{s} = \int_{t_P}^{t_Q} \mathrm{d}t\,\sqrt{\vec{v}\,(t)^2 - c^2} \tag{5.23}$$

heisst dann Eigenraum.

5.1.6 Viererposition, Vierergeschwindigkeit, Viererbeschleunigung und Vierer-impuls

Durch die Eigenzeit τ ist ein Lorentz-invarianter Zeitparameter gegeben, der sich deshalb zur Parametrisierung von Weltlinien $x^\mu(t)$ eignet. Die Viererposition ist gegeben durch

$$x^\mu(t) \doteq (ct, \vec{x}\,(t)), \quad x_\mu x^\mu = \eta_{\mu\nu} x^\mu x^\nu = -ct^2 + \vec{x}^{\,2}. \tag{5.24}$$

Die Vierergeschwindigkeit $u^\mu(\tau)$ ist nach Blau durch die Parametrisierung $x^\mu(\tau)$ gemäss

$$u^\mu(\tau) \doteq \frac{\mathrm{d}x^\mu(\tau)}{\mathrm{d}\tau} = (\gamma(\vec{v})c, \gamma(\vec{v})\vec{v}), \quad u_\mu u^\mu = \eta_{\mu\nu} u^\mu u^\nu = -c^2 \tag{5.25}$$

gegeben [**?**, S.34]. Die Viererbeschleunigung $a^\mu(\tau)$ ergibt sich sodann durch

$$a^\mu(\tau) \doteq \frac{\mathrm{d}^2 x^\mu(\tau)}{\mathrm{d}\tau^2}. \tag{5.26}$$

Der Viererimpuls $p^\mu(\tau)$ wird gemäss Blau definiert durch

$$p^\mu(\tau) \doteq m u^\mu(\tau) = m\frac{\mathrm{d}x^\mu(\tau)}{\mathrm{d}\tau} = m(\gamma(\vec{v})c, \gamma(\vec{v})\vec{v}), \quad p_\mu p^\mu = m^2 \eta_{\mu\nu} u^\mu u^\nu = -m^2c^2, \tag{5.27}$$

wobei der Viererimpuls auch zu

$$p^\mu(\tau) = (E/c, \vec{p}) \tag{5.28}$$

geschrieben werden kann, da $mc^2\gamma(\vec{v})$ die relativistische Energie eines massiven Teilchens ist [**?**, S.36]. Durch das Skalarprodukt

$$p_\mu p^\mu = m^2 \eta_{\mu\nu} u^\mu u^\nu = -m^2c^2 = -\frac{E^2}{c^2} + \vec{p}^{\,2} \quad \Leftrightarrow \quad E^2 = \vec{p}^{\,2}c^2 + m^2c^4 \tag{5.29}$$

ergibt sich die bekannte Energie-Impuls-Beziehung, welche auch die «Massenschale» genannt wird.

5.2 Herleitung des Dopplereffektes

5.2.1 Relativistischer Dopplereffekt für elektromagnetische Wellen im Vakuum

Zwecks einer Herleitung des relativistischen Dopplereffektes für elektromagnetische Wellenphänomene sei Abbildung 16 betrachtet. Die im Figur 16 ersichtliche orangen Linien repräsentieren die Wellenfronten einer sich ausgehend vom Ursprung des Inertialsystems Σ in x^1-Richtung ausbreitenden ebenen Welle

$$u(\vec{x}, t) = u_0 e^{i(\vec{k}\cdot\vec{x} - \omega t)} = u_0 e^{i\varphi}, \quad \varphi \doteq \vec{k}\cdot\vec{x} - \omega t, \tag{5.30}$$

die der Herleitung von Straumann folgend angesetzt ist [**?**, S.35]. Die Wellengleichung

$$-\frac{1}{c^2}\frac{\partial^2 u(\vec{x}, t)}{\partial t^2} + \Delta u(\vec{x}, t) = 0 \tag{5.31}$$

$$\Leftrightarrow \quad -\frac{1}{c^2}(i\omega)^2 u(\vec{x}, t) + (i\vec{k}\,)^2 u(\vec{x}, t) = 0 \quad \Leftrightarrow \quad \vec{k}^{\,2} = \left(\frac{\omega}{c}\right)^2 \tag{5.32}$$

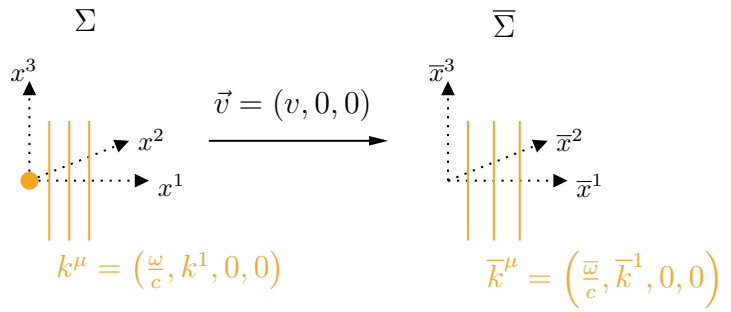


Abbildung 16: Skizze zur Herleitung des relativistischen Dopplereffektes für elektromagnetische Wellen. Orange dargestellt sind die Wellenfronten einer sich ausgehend vom Ursprung des Inertialsystems Σ in x^1-Richtung ausbreitenden ebenen Welle. Dieselbe Welle kann im Inertialsystem $\bar{\Sigma}$, welches sich mit der Geschwindigkeit $\vec{v} = (v, 0, 0)$ in x^1-Richtung bewegt, beobachtet werden.

für $u(\vec{x}, t)$ ergibt die Bedingung $|\vec{k}\,| = \omega/c$ für den Wellenvektor $\vec{k}$. Die Phase $\varphi = \vec{k} \cdot \vec{x} - \omega t$ bleibt gemäss Straumann unter Lorentz-Transformationen invariant - das heisst, dass φ ein Lorentz-Skalar sein muss [**?**, S.35]. Definiert man den Vierer-Wellenvektor k^μ durch

$$k^\mu = \left(\frac{\omega}{c}, \vec{k}\right), \tag{5.33}$$

so resultiert

$$k_\mu k^\mu = \eta_{\nu\mu} k^\mu k^\nu = -\frac{\omega^2}{c^2} + \vec{k}^{\,2} \overset{(5.32)}{=} 0, \tag{5.34}$$

woraus mit der geforderten Invarianz von

$$k_\mu x^\mu = \eta_{\mu\nu} k^\mu x^\nu = \begin{pmatrix} -\omega/c \\ k^1 \\ k^2 \\ k^3 \end{pmatrix} \cdot \begin{pmatrix} ct \\ x^1 \\ x^2 \\ x^3 \end{pmatrix} = \vec{k} \cdot \vec{x} - \omega t = \varphi \overset{!}{=} \bar{\varphi} = \bar{k}_\mu \bar{x}^\mu \tag{5.35}$$

folgt, dass sowohl x^μ als auch k^μ Lorentz-Vektoren sind. Im Bezug auf Figur 16 sind die Wellenvektoren derselben ebenen Welle in den Inertialsystemen Σ und $\bar{\Sigma}$ definiert durch

$$k^\mu = \left(\frac{\omega}{c}, k^1, 0, 0\right), \quad \bar{k}^\mu = L^\mu{}_\nu k^\nu = \left(\frac{\bar{\omega}}{c}, \bar{k}^1, 0, 0\right). \tag{5.36}$$

Der Lorentz-transformierte Wellenvektor $\bar{k}^\mu$ lautet in ausgeschriebener Form auf

$$\bar{k}^\mu = L^\mu{}_\nu k^\nu = \begin{pmatrix} \gamma & -\beta\gamma & 0 & 0 \\ -\beta\gamma & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega/c \\ k^1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \gamma(\omega/c - \beta k^1) \\ \gamma(k^1 - \beta\omega/c) \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \bar{\omega}/c \\ \bar{k}^1 \\ 0 \\ 0 \end{pmatrix}. \tag{5.37}$$

Die Lichtgeschwindigkeit c berechnet sich aus einem Wellenvektor $\vec{k}$ nach Beziehung (5.32) durch $c = \omega/|\vec{k}|$ und ist eine Lorentz-Invariante, das heisst, sie ist in allen Inertialsystemen gleich gross. Deshalb muss die Gleichung

$$c = \frac{\bar{\omega}}{\bar{k}^1} = \frac{\omega}{k^1} \tag{5.38}$$

für die beiden Wellenvektoren $k^\mu = (\omega/c, \vec{k}\,) = (\omega/c, k^1, 0, 0)$ und $\bar{k}^\mu = (\bar{\omega}/c, \vec{\bar{k}}) = (\bar{\omega}/c, \bar{k}^1, 0, 0)$ gelten. Setzt man die aus (5.38) erhaltene Beziehung $k^1 = \omega/c$ in die Gleichung der ersten

Komponente von (5.37) ein, so ergibt sich unter einigen Umformungen die Gleichung

$$\frac{\bar{\omega}}{c} = \gamma(\frac{\omega}{c} - \beta k^1) = \gamma(\frac{\omega}{c} - \beta\frac{\omega}{c}) \quad \Leftrightarrow \quad \bar{\omega} = \gamma(\omega - \beta\omega) = \gamma\omega(1-\beta) = \omega\frac{1-\beta}{(1-\beta^2)^{1/2}}. \quad (5.39)$$

Quadrieren ergibt

$$\bar{\omega}^2 = \omega^2\frac{(1-\beta)^2}{1-\beta^2} = \omega^2\frac{(1-\beta)^2}{(1-\beta)(1+\beta)} = \omega^2\frac{1-\beta}{1+\beta}. \quad (5.40)$$

Nimmt man die Beziehung $\omega = {}^{2\pi}/_{T} = 2\pi f$ mit T der Periodendauer und f der Frequenz der betrachteten elektromagnetischen Welle im Inertialsystem Σ, sowie den Ausdruck für die analoge Grösse $\bar{\omega} = 2\pi\bar{f}$ im Inertialsytem $\bar{\Sigma}$ hinzu, so ergibt sich mit $\beta = {}^{v}/_{c}$ die Gleichung

$$\bar{f} = f\sqrt{\frac{1-\beta}{1+\beta}} = f\sqrt{\frac{1-\frac{v}{c}}{1+\frac{v}{c}}}, \quad v\begin{cases} > 0, & \Sigma \leftarrow\rightarrow \bar{\Sigma} \\ < 0, & \Sigma \rightarrow\leftarrow \bar{\Sigma} \end{cases}, \quad (5.41)$$

welche mit dem relativistischen Dopplereffekt für elektromagnetische Wellen im Vakuum benannt wird. Es sei bemerkt, dass die Geschwindigkeit $\vec{v} = (v, 0, 0)$ des Inertialsystems $\bar{\Sigma}$ gegenüber dem Inertialsystem Σ beliebige Situationen für Betrag und Richtung einer Relativgeschwindigkeit zweier Inertialsysteme beschreibt. Denn jede Relativbewegung zweier Inertialsysteme kann derart beschrieben werden, dass der Geschwindigkeitsvektor der Relativbewegung auf $\vec{v} = (v, 0, 0)$ lautet.

Der relativistische Dopplereffekt elektromagnetischer Wellen im Vakuum kann auch für ein Medium mit Brechungsindex n angegeben werden. Hierzu muss in der Formel des Dopplereffektes die Lichtgeschwindigkeit c im Vakuum durch die Lichtgeschwindigkeit $\tilde{c} = {}^{c}/_{n}$ im Medium ersetzt werden.

5.2.2 Nichtrelativistischer Dopplereffekt für elektromagnetische Wellen im Vakuum

Der nichtrelativistische Ausdruck für den Dopplereffekt bei elektromagnetischen Wellen im Vakuum kann als Grenzfall $v \ll c$ betrachtet werden. Das heisst, der relativistische Ausdruck (5.41) wird um den Entwicklungspunkt ${}^{v}/_{c} \approx 0$ in eine Taylorreihe entwickelt. Hierzu definiert man die variable $x \doteq {}^{v}/_{c}$ und entwickelt $\bar{f}(x)$ um den Punkt $x_0 = 0$:

$$\bar{f}(x) = f\sqrt{\frac{1-x}{1+x}} = \sum_{k=0}^{\infty}\frac{1}{k!}\frac{\mathrm{d}^k f(x)}{\mathrm{d}x^k}\bigg|_{x=x_0}(x-x_0)^k = f - fx + \mathcal{O}(x^2) \approx f(1-x). \quad (5.42)$$

Es gilt also mit der Rücksubstitution für ${}^{v}/_{c} \ll 1$ die Gleichung

$$\bar{f} = f(1-\frac{v}{c}) = \frac{f}{c}(c-v), \quad v\begin{cases} > 0, & \Sigma \leftarrow\rightarrow \bar{\Sigma} \\ < 0, & \Sigma \rightarrow\leftarrow \bar{\Sigma} \end{cases}, \quad (5.43)$$

welche mit dem nichtrelativistischen Dopplereffekt für elektromagnetische Wellen im Vakuum zu identifizieren ist.

Der nichtrelativistische Dopplereffekt elektromagnetischer Wellen im Vakuum kann auch für ein Medium mit Brechungsindex n angegeben werden. Hierzu muss in der Formel des Dopplereffektes die Lichtgeschwindigkeit c im Vakuum durch die Lichtgeschwindigkeit $\tilde{c} = {}^{c}/_{n}$ im Medium ersetzt werden.

5.2.3 Dopplereffekt für nichtrelativistische Wellen

Nachfolgend soll der Dopplereffekt für nichtrelativistische Wellen in einem Medium betrachtet werden. Solche Wellen werden auch als Schallwellen bezeichnet. Deren Geschwindigkeit im Medium sei nachfolgend mit c_w bezeichnet. Der Dopplereffekt soll für zwei Fälle hergeleitet werden: Erstens den Fall, dass sich eine Quelle E relativ zu einem ruhenden Empfänger R bewege und zweitens für den Fall, dass sich ein bewegter Empfänger R relativ zu einer ruhenden Quelle E bewege. Eine entsprechende Skizze der beiden Fälle ist in Abbildung 17 gegeben. Jede beliebige Relativbewegung zweier gleichförmig bewegter Systeme kann

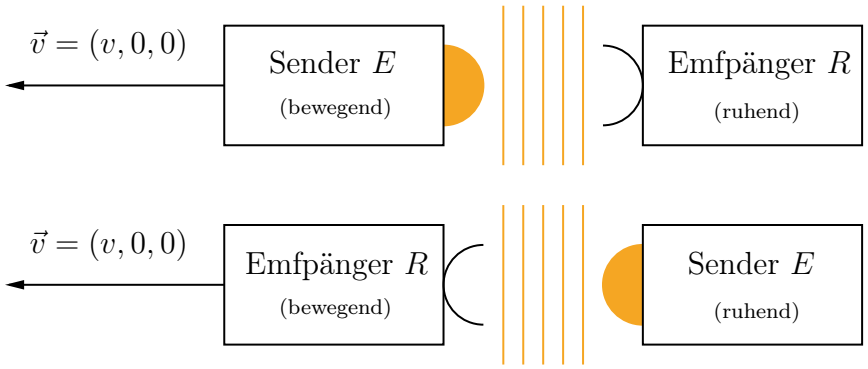


Abbildung 17: Skizze der möglichen Fälle zur Herleitung des Dopplereffektes einer Welle in einem Medium (Schallwelle).

durch eine geeignete Wahl des Koordinatensystems mittels des Geschwindigkeitsvektors $\vec{v} = (v, 0, 0)$ beschrieben werden, daher bildet die Abbildung 17 den allgemeinen Fall einer Relativbewegung zweier Systeme ab. Zunächst sei eine Quelle E betrachtet, welche sich relativ zu einem ruhenden Empfänger R bewege. Für den im Bild 17 gezeigten Fall einer sich vom Empfänger entfernenden Quelle gilt, dass die Wellenlänge λ_R für den Empfänger grösser sein wird, als sie von der Quelle mit Wellenlänge λ_E ausgesandt wird. Es gilt also

$$\lambda_R = \lambda_E + vT_E, \tag{5.44}$$

wobei T_E die Periodendauer der emittierten Welle beim Sender ist. Hieraus folgt für die beim Empfänger gemessene Frequenz

$$f_R = \frac{c_w}{\lambda_E} = \frac{c_w}{\lambda_E + vT_E} = \frac{c_w}{\frac{c_w}{f_E} + \frac{v}{f_E}} = f_E \frac{c_w}{c_w + v}. \tag{5.45}$$

Nun sei ein bewegter Empfänger R betrachtet, welcher sich relativ zu einer ruhenden Quelle E bewege. Für den im Bild 17 gezeigten Fall eines sich von der Quelle entfernenden Empfängers gilt, dass die Periodendauer T_R für den Empfänger grösser sein wird, als sie für die Quelle mit Periodendauer T_E ist. Es gilt also

$$(c_w - v)T_R = c_w T_E \quad \Leftrightarrow \quad T_R = \frac{c_w T_E}{c_w - v}. \tag{5.46}$$

Daraus folgt für die beim Empfänger gemessene Frequenz

$$f_R = \frac{1}{T_R} = \frac{c_w - v}{c_w T_E} = f_E \frac{c_w - v}{c_w}. \tag{5.47}$$

Zusammenfassend kann der Dopplereffekt für Schallwellen also gemäss Paul A. Tipler durch

$$\text{Sender ruhend, Empfänger bewegt:} \quad \bar{f} = f\frac{c_w}{c_w + v}, \quad v \begin{cases} > 0, & \text{Sender} \leftarrow\rightarrow \text{Empfänger} \\ < 0, & \text{Sender} \rightarrow\leftarrow \text{Empfänger} \end{cases}, \tag{5.48}$$

$$\text{Sender bewegt, Empfänger ruhend:} \quad \bar{f} = f\frac{c_w - v}{c_w}, \quad v \begin{cases} > 0, & \text{Sender} \leftarrow\rightarrow \text{Empfänger} \\ < 0, & \text{Sender} \rightarrow\leftarrow \text{Empfänger} \end{cases} \tag{5.49}$$

geschrieben werden, wobei $\bar{f}$ die vom Empfänger empfangene und f die vom Sender ausgesandte Frequenz sind [**?**, S.474-475].

5.3 Koordinatentransformationen

Mittels Translationen

$$T : \mathbb{R}^3 \to \mathbb{R}^3 \quad \vec{x} \mapsto \vec{x} - \vec{u}, \quad \vec{u} \in \mathbb{R}^3 \tag{5.50}$$

und einer Abfolge von Rotationen

$$R : \mathbb{R}^3 \to \mathbb{R}^3 \quad \vec{x} \mapsto R(\psi)\vec{x}, \quad \psi \in \mathbb{R}, \quad R(\psi) \in \text{Mat}(3 \times 3, \mathbb{R}) \tag{5.51}$$

können beliebige galileische Transformationen zwischen Koordinatensystemen durchgeführt werden. Es kann beispielsweise durch die Verwendung von zeitabhängigen Translations- und/oder Rotationsoperatoren von einem zeitunabhängigen Koordinatensystem in ein zeitabhängiges Koordinatensystem transformiert werden, wie dies etwa bei Transformationen topozentrischer Koordinaten in inertiale Koordinaten erfolgt.

5.4 Vektornotation

In der vorliegenden Arbeit werden drei verschiedene Fälle von Dimensionalität unterschieden - eindimensionale Variablen $a \in \mathbb{R}^1$, dreidimensionale Variablen $\vec{a} \in \mathbb{R}^3$ und n-dimensionale Variablen $a \in \mathbb{R}^n$. Wenn aus dem Kontext nicht anders ersichtlich, so ist mit einer Variable der Form a immer eine ein- oder n-dimensionale Grösse gemeint, während mit einer Variable der Form $\vec{a}$ immer auf einen dreidimensionalen Vektor verwiesen wird.

5.5 Python-Quellcode

Nachfolgend ist der Python-Code zur in Abschnitt 3.4.5 beschriebenen Implementation des globalen Lösungsverfahrens für nichtlineare Gleichungssysteme nach Sektion 3.2.4 abgedruckt. Das Programm wird in der vorliegenden Arbeit `numsolver.py` genannt.

```
"""
Author: Daniel Zahnd
Created: 06.07.2022
Last modified: 26.07.2022
"""


"INITIALIZATION"

```

```
# Import python libraries
import math as m
from scipy import optimize
import numpy as np
import time as t
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
import matplotlib

matplotlib.rcParams['mathtext.fontset'] = 'cm'
matplotlib.rcParams['font.family'] = 'STIXGeneral'

# Start program runtime measurement
start_time = t.time()

# Initialize global variables
global x_p_true
global y_p_true # True position coordinates of object
global z_p_true

global x_v_true
global y_v_true # True velocity coordinates of object
global z_v_true

global x_p_min, x_p_max, x_v_min, x_v_max
global y_p_min, y_p_max, y_v_min, y_v_max # Minimum and maximum values
    for coordinates
global z_p_min, z_p_max, z_v_min, z_v_max

global x_1, y_1, z_1
global x_2, y_2, z_2
global x_3, y_3, z_3 # Receiver coordinates
global x_4, y_4, z_4 # Emitter is assumed to be at orignin of coordinate
     system
global x_5, y_5, z_5
global x_6, y_6, z_6

global del_f1
global del_f2
global del_f3 # Frequency shifts at receiver station
global del_f4 # Emitter is assumed to be at orignin of coordinate system
global del_f5
global del_f6

global a # Baseline lenght of measurement facility geometry triangle


"FUNCTION DEFINITIONS"


# Nonlinear equation system
def eqsyst(f):
  x_p = f[0] # x-position of object
  y_p = f[1] # y-position of object
  z_p = f[2] # z-position of object
  x_v = f[3] # x-velocity of object
  y_v = f[4] # y-velocity of object
  z_v = f[5] # z-velocity of object
```

```
  F = np.empty(6)
  F[0] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_1)+y_v*(y_p-y_1)+z_v*(z_p-z_1))*((x_p-x_1)**2+(y_p-y_1)**2+(z_p-z_1)**2)**(-1/2)-del_f1
  F[1] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_2)+y_v*(y_p-y_2)+z_v*(z_p-z_2))*((x_p-x_2)**2+(y_p-y_2)**2+(z_p-z_2)**2)**(-1/2)-del_f2
  F[2] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_3)+y_v*(y_p-y_3)+z_v*(z_p-z_3))*((x_p-x_3)**2+(y_p-y_3)**2+(z_p-z_3)**2)**(-1/2)-del_f3
  F[3] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_4)+y_v*(y_p-y_4)+z_v*(z_p-z_4))*((x_p-x_4)**2+(y_p-y_4)**2+(z_p-z_4)**2)**(-1/2)-del_f4
  F[4] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_5)+y_v*(y_p-y_5)+z_v*(z_p-z_5))*((x_p-x_5)**2+(y_p-y_5)**2+(z_p-z_5)**2)**(-1/2)-del_f5
  F[5] = (x_v*x_p+y_v*y_p+z_v*z_p)/(m.sqrt(x_p**2+y_p**2+z_p**2)) + (x_v*(x_p-x_6)+y_v*(y_p-y_6)+z_v*(z_p-z_6))*((x_p-x_6)**2+(y_p-y_6)**2+(z_p-z_6)**2)**(-1/2)-del_f6
  return F

# Solution seeker function
def solseeker(z_vel, div):
  half = int(div/2)
  x_p = np.linspace(x_p_min, x_p_max, num=div, endpoint=True)
  y_p = np.linspace(y_p_min, y_p_max, num=div, endpoint=True)
  z_p = np.linspace(z_p_min, z_p_max, num=div, endpoint=True)
  x_v_neg = np.linspace(-x_v_max,-x_v_min, num=half, endpoint=True)
  x_v_pos = np.linspace(x_v_min,x_v_max, num=half, endpoint=True)
  x_v = np.hstack((x_v_neg,x_v_pos))
  y_v_neg = np.linspace(-y_v_max,-y_v_min, num=half, endpoint=True)
  y_v_pos = np.linspace(y_v_min,y_v_max, num=half, endpoint=True)
  y_v = np.hstack((y_v_neg,y_v_pos))
  z_v =  z_vel# Should be assumed to be near zero if object orbit is nearly circular
  solutions = np.empty((0,12)) # Solutions array [solution/initial guess]
  it_number = int(0)
  perc_number = int(0)
  print(0, '% progress (initial solving)')
  for i in range(0,len(x_p)):
    for j in range(0,len(y_p)):
      for k in range(0,len(z_p)):
        for l in range(0,len(x_v)):
          for n in range(0,len(y_v)):
            guess = np.array([x_p[i],y_p[j],z_p[k],x_v[l],x_v[n],z_v])
            sol = optimize.root(eqsyst, guess, method='hybr') #, options={'maxiter':10000, 'fatol':1e-9})
            if sol.success == True:
              # print("Solution found")
              sol_entry = np.concatenate((sol.x,guess))
              solutions = np.vstack((solutions,sol_entry))
            it_number = it_number + 1
            if it_number % (int((div**5)/10)) == 0:
              perc_number = perc_number + 1
              print(perc_number*10, '% progress (initial solving)')
  if len(solutions) == 0:
    print("No solutions were found at initial solving.")
```

```
    return
  else:
    ind_bestsol = bestsol(solutions)
    bestsolution = solutions[ind_bestsol,:]
    return bestsolution

# Search for the index of the best solutions in terms of residues
def bestsol(sols):
  (solutions,guesses) = np.hsplit(sols,2)
  res = np.array([])
  rows, columns = solutions.shape
  for i in range(0,rows):
    val = np.linalg.norm(eqsyst(solutions[i,:]))
    res = np.append(res,val)
  index = np.argmin(res)
  return index

# Seek for more exact convergences in environment of best first solution
def refiner1(z_vel, sol, diff, div):
  if sol is None:
    return
  x_p = np.linspace(sol[0]-diff, sol[0]+diff, num=div, endpoint=True)
  y_p = np.linspace(sol[1]-diff, sol[1]+diff, num=div, endpoint=True)
  z_p = np.linspace(sol[2]-diff, sol[2]+diff, num=div, endpoint=True)
  x_v = np.linspace(sol[3]-diff, sol[3]+diff, num=div, endpoint=True)
  y_v = np.linspace(sol[4]-diff, sol[4]+diff, num=div, endpoint=True)
  z_v = z_vel # Should be assumed to be near zero if object orbit is
   nearly circular
  solutions = np.empty((0,12)) # Solutions array [solution/initial guess
   ]
  it_number = int(0)
  perc_number = int(0)
  print(0, '% progress (solution refining)')
  for i in range(0,len(x_p)):
    for j in range(0,len(y_p)):
      for k in range(0,len(z_p)):
        for l in range(0,len(x_v)):
          for n in range(0,len(y_v)):
            guess = np.array([x_p[i],y_p[j],z_p[k],x_v[l],y_v[n],z_v])
            solut = optimize.root(eqsyst, guess, options={'xtol':1e-10})
            if solut.success == True:
              # print("Solution found")
              sol_entry = np.concatenate((solut.x,guess))
              solutions = np.vstack((solutions,sol_entry))
            it_number = it_number + 1
            if it_number % (int((div**5)/10)) == 0:
              perc_number = perc_number + 1
              print(perc_number*10, '% progress (solution refining)')
  if len(solutions) == 0:
    print("No solutions were found at solution refining.")
    return
  else:
    ind_bestsol = bestsol(solutions)
    solu = solutions[ind_bestsol,:]
    ressolu = np.linalg.norm(eqsyst(solu))
    ressol = np.linalg.norm(eqsyst(sol))
    if ressolu < ressol:
      return solu
```

```
    else:
      return sol

# Seek for more exact convergences in environment of best refined
    solution
def refiner2(sol, diff, div):
  if sol is None:
    return
  x_p = np.linspace(sol[0]-diff, sol[0]+diff, num=div, endpoint=True)
  y_p = np.linspace(sol[1]-diff, sol[1]+diff, num=div, endpoint=True)
  z_p = np.linspace(sol[2]-diff, sol[2]+diff, num=div, endpoint=True)
  x_v = sol[3]
  y_v = sol[4]
  z_v = sol[5]
  solutions = np.empty((0,12)) # Solutions array [solution/initial guess
    ]
  it_number = int(0)
  perc_number = int(0)
  print(0, '% progress (solution refining)')
  for i in range(0,len(x_p)):
    for j in range(0,len(y_p)):
      for k in range(0,len(z_p)):
        guess = np.array([x_p[i],y_p[j],z_p[k],x_v,y_v,z_v])
        solut = optimize.root(eqsyst, guess, options={'xtol':1e-10})
        if solut.success == True:
          # print("Solution found")
          sol_entry = np.concatenate((solut.x,guess))
          solutions = np.vstack((solutions,sol_entry))
        it_number = it_number + 1
        if it_number % (int((div**3)/10)) == 0:
          perc_number = perc_number + 1
          print(perc_number*10, '% progress (solution refining)')
  if len(solutions) == 0:
    print("No solutions were found at solution refining.")
    return
  else:
    ind_bestsol = bestsol(solutions)
    solu = solutions[ind_bestsol,:]
    ressolu = np.linalg.norm(eqsyst(solu))
    ressol = np.linalg.norm(eqsyst(sol))
    if ressolu < ressol:
      return solu
    else:
      return sol

# Printer, prints results of iteration
def printer(sol,descr):
  if sol is None:
    return
  else:
    (prin,noprin) = np.hsplit(sol,2)
    totvel = m.sqrt(x_v_true**2 + y_v_true**2 + z_v_true**2)
    print('\n\n')
    print('**************', descr, '**************')
    print('x_{S,c} = ', round(prin[0],3), 'm')
    print('y_{S,c} = ', round(prin[1],3), 'm')
    print('z_{S,c} = ', round(prin[2],3), 'm')
    print('\dot{x}_{S,c} = ', round(prin[3],3), 'm/s')
```

```
222     print('\dot{y}_{S,c} = ', round(prin[4],3), 'm/s')
223     print('\dot{z}_{S,c} = ', round(prin[5],3), 'm/s')
224     print('\Delta x_{S,c} = ', round(abs(prin[0]-x_p_true),3), 'm   or
    ', round((abs(sol[0]-x_p_true)/a)*100,3), '% of baseline')
225     print('\Delta y_{S,c} = ', round(abs(prin[1]-y_p_true),3), 'm   or
    ', round((abs(sol[1]-y_p_true)/a)*100,3), '% of baseline')
226     print('\Delta z_{S,c} = ', round(abs(prin[2]-z_p_true),3), 'm   or
    ', round((abs(sol[2]-z_p_true)/a)*100,3), '% of baseline')
227     print('\Delta \dot{x}_{S,c} = ', round(abs(prin[3]-x_v_true),3), 'm/
    s   or  ', round((abs(sol[3]-x_v_true)/totvel)*100,3), '% of total
    velocity')
228     print('\Delta \dot{y}_{S,c} = ', round(abs(prin[4]-y_v_true),3), 'm/
    s   or  ', round((abs(sol[4]-y_v_true)/totvel)*100,3), '% of total
    velocity')
229     print('\Delta \dot{z}_{S,c} = ', round(abs(prin[5]-z_v_true),3), 'm/
    s   or  ', round((abs(sol[5]-z_v_true)/totvel)*100,3), '% of total
    velocity')
230     print('
    ************************************************************\n')
231     return
232
233 # Callback function to write out iteration information
234 def examiner(sol,residue):
235   global iterstep
236   global iterations
237   iterst = np.array([iterstep])
238   appnd = np.concatenate((iterst,sol))
239   iterations = np.vstack((iterations,appnd))
240   iterstep = iterstep + 1
241
242
243 "MAIN PROGRAM"
244
245
246 # Define Constants
247 M_E = 5.972e24 # Mass of earth
248 G = 6.67430e-11 # Gravitational constant
249 R_E = 6.371e6 # Radius of spherical earth
250
251 # Define geometry of measurement device
252 a = 1*1.0e5 # Edge lenght of triangle (baseline)
253 h = (m.sqrt(3)/2)*a # Height of triangle
254
255 # Position of receiver R_1
256 x_1 = -a
257 y_1 = 0.0
258 z_1 = 0.0
259 # Position of receiver R_2
260 x_2 = -a/2
261 y_2 = -h
262 z_2 = 0.0
263 # Position of receiver R_3
264 x_3 = a/2
265 y_3 = -h
266 z_3 = 0.0
267 # Position of receiver R_4
268 x_4 = a
269 y_4 = 0.0
```

```
z_4 = 0.0
# Position of receiver R_5
x_5 = a/2
y_5 = h
z_5 = 0.0
# Position of receiver R_6
x_6 = -a/2
y_6 = h
z_6 = 0.0

"""*********************  SIMULATED VALUES  **********************"""
"""************************************************************"""
# True position coordinates of object
true_vector = np.array([8.3e4, -1.4e3, 1.9e5, 7.8e3, -6.9e3, -1.1e2])
x_p_true = true_vector[0] # True x-coordinate of object
y_p_true = true_vector[1] # True y-coordinate of object
z_p_true = true_vector[2] # True z-coordinate of object

# True velocity coordinates of object
x_v_true = true_vector[3] # True x-velocity of object
y_v_true = true_vector[4] # True y-velocity of object
z_v_true = true_vector[5] # True z-velocity of object

"""************************************************************"""
"""************************************************************"""

# Calculate expected frequency shifts at receiver stations
del_f1 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_1)+
    y_v_true*(y_p_true-y_1)+z_v_true*(z_p_true-z_1))/(m.sqrt((x_p_true-
    x_1)**2+(y_p_true-y_1)**2+(z_p_true-z_1)**2))
del_f2 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_2)+
    y_v_true*(y_p_true-y_2)+z_v_true*(z_p_true-z_2))/(m.sqrt((x_p_true-
    x_2)**2+(y_p_true-y_2)**2+(z_p_true-z_2)**2))
del_f3 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_3)+
    y_v_true*(y_p_true-y_3)+z_v_true*(z_p_true-z_3))/(m.sqrt((x_p_true-
    x_3)**2+(y_p_true-y_3)**2+(z_p_true-z_3)**2))
del_f4 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_4)+
    y_v_true*(y_p_true-y_4)+z_v_true*(z_p_true-z_4))/(m.sqrt((x_p_true-
    x_4)**2+(y_p_true-y_4)**2+(z_p_true-z_4)**2))
del_f5 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_5)+
    y_v_true*(y_p_true-y_5)+z_v_true*(z_p_true-z_5))/(m.sqrt((x_p_true-
    x_5)**2+(y_p_true-y_5)**2+(z_p_true-z_5)**2))
del_f6 = (x_v_true*x_p_true+y_v_true*y_p_true+z_v_true*z_p_true)/(m.sqrt
    (x_p_true**2+y_p_true**2+z_p_true**2)) + (x_v_true*(x_p_true-x_6)+
    y_v_true*(y_p_true-y_6)+z_v_true*(z_p_true-z_6))/(m.sqrt((x_p_true-
    x_6)**2+(y_p_true-y_6)**2+(z_p_true-z_6)**2))

# Define boundaries of possible values for a solution (dependant upon
    geometry of measurement facility)
x_p_min = -a
x_p_max = a
y_p_min = -a
y_p_max = a
```

```
z_p_min = 1.6e5 # Minimum for LEO (accessible via doppler radar)
z_p_max = 2.0e6 # Maximum for LEO (accessible via doppler radar)
x_v_min = m.sqrt((G*M_E)/(R_E+z_p_max)) # Of order 10^3, approx. 6.9e3
x_v_max = m.sqrt((G*M_E)/(R_E+z_p_min)) # Of order 10^3, approx. 7.8e3
y_v_min = m.sqrt((G*M_E)/(R_E+z_p_max)) # Of order 10^3, approx. 6.9e3
y_v_max = m.sqrt((G*M_E)/(R_E+z_p_min)) # Of order 10^3, approx. 7.8e3
z_v_min = -2.0e2 # Reasonable assumption for almost circular orbit
z_v_max = 2.0e2 # Reasonable assumption for almost circular orbit

"""**********************  FUNCTION CALLS  **********************"""
"""************************************************************"""
# Run solution seeker, returns best initial solution
solution = solseeker(0.0, 4)
sol_plot = solution
printer(solution,'Initial solution')

# Run solution refiner, returns refined solution if better than initial
    solution
solution = refiner1(0.0, solution, 1e3, 4)
printer(solution,'1st ref. solution')

# Run solution refiner, returns refined solution if better than refined
    solution
solution = refiner2(solution, 5e2, 8)
printer(solution,'2nd ref. solution')

# Run solution refiner, returns refined solution if better than refined
    solution
solution = refiner2(solution, 1e2, 10)
printer(solution,'3rd ref. solution')

# Run solution refiner, returns refined solution if better than refined
    solution
solution = refiner1(0.0, solution, 5, 4)
printer(solution,'4th ref. solution')

"""************************************************************"""
"""************************************************************"""

# # Plotting of convergence behaviours for initial solving
# iterstep = 1
# iterations = np.empty((0,7))
# (sol_plot_sol, sol_plot_guess) = np.hsplit(sol_plot,2)
# printsol = optimize.root(eqsyst, sol_plot_guess, method='broyden1',
    callback=examiner)
# firstline = np.concatenate(([0],sol_plot_guess))
# iterations = np.vstack((firstline,iterations))

# itdata = iterations[:,0]
# xpdata = iterations[:,1]
# truexpdata = np.full(len(itdata), x_p_true)
# ypdata = iterations[:,2]
# trueypdata = np.full(len(itdata), y_p_true)
# zpdata = iterations[:,3]
# truezpdata = np.full(len(itdata), z_p_true)
# xvdata = iterations[:,4]
# truexvdata = np.full(len(itdata), x_v_true)
# yvdata = iterations[:,5]
```

```
# trueyvdata = np.full(len(itdata), y_v_true)
# zvdata = iterations[:,6]
# truezvdata = np.full(len(itdata), z_v_true)

# fig1 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $x_{S,c}$\n", fontsize=25)
# plt.plot(itdata, xpdata, 'b', label='Iterationsfortschritt', linewidth=1.5)
# plt.plot(itdata, truexpdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$x_{S,c}$ [m]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_xp.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig1)
#
# fig2 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $y_{S,c}$\n", fontsize=25)
# plt.plot(itdata, ypdata, 'b', label='Iterationsfortschritt', linewidth=1.5)
# plt.plot(itdata, trueypdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$y_{S,c}$ [m]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_yp.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig2)

# fig3 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $z_{S,c}$\n", fontsize=25)
# plt.plot(itdata, zpdata, 'b', label='Iterationsfortschritt', linewidth=1.5)
# plt.plot(itdata, truezpdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$z_{S,c}$ [m]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_zp.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig3)

# fig4 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $\dot{x}_{S,c}$\n", fontsize=25)
# plt.plot(itdata, xvdata, 'b', label='Iterationsfortschritt', linewidth=1.5)
# plt.plot(itdata, truexvdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
```

```
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$\dot{x}_{S,c}$ [m/s]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_xv.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig4)

# fig5 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $\dot{y}_{S,c}$\n", fontsize=25)
# plt.plot(itdata, yvdata, 'b', label='Iterationsfortschritt', linewidth
    =1.5)
# plt.plot(itdata, trueyvdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$\dot{y}_{S,c}$ [m/s]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_yv.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig5)

# fig6 = plt.figure(figsize=(10,7))
# plt.title("\nKonvergenzverhalten von $\dot{z}_{S,c}$\n", fontsize=25)
# plt.plot(itdata, zvdata, 'b', label='Iterationsfortschritt', linewidth
    =1.5)
# plt.plot(itdata, truezvdata, 'r', label='Wahrer Wert', linewidth=0.75)
# plt.legend(loc = 'lower right', fontsize=18)
# plt.xlabel('Anzahl Iterationen', fontsize=20)
# plt.ylabel('$\dot{z}_{S,c}$ [m/s]', fontsize=20)
# plt.ticklabel_format(axis="y", style="sci", scilimits=(0,0))
# plt.xticks(fontsize = 20)
# plt.yticks(fontsize = 20)
# plt.grid(True)
# plt.savefig('conv_zv.png', dpi=600, bbox_inches='tight')
# plt.show()
# plt.close(fig6)

# Evaluate and print runtime of program
print("\nRuntime: %s seconds" % (t.time() - start_time))
```

6 Literatur