Computerphysik I: Swing-by Manöver

Aurel Müller-Schoenau, Leon Oleschko und Moritz Schröer

30. September 2022

1 Einleitung

Es wurde die Ulysses Raumfahrtmission gewählt, da nur ein Swing-by durchgeführt wurde und die Flugbahn verfügbar[1] ist.

2 Trajektorie

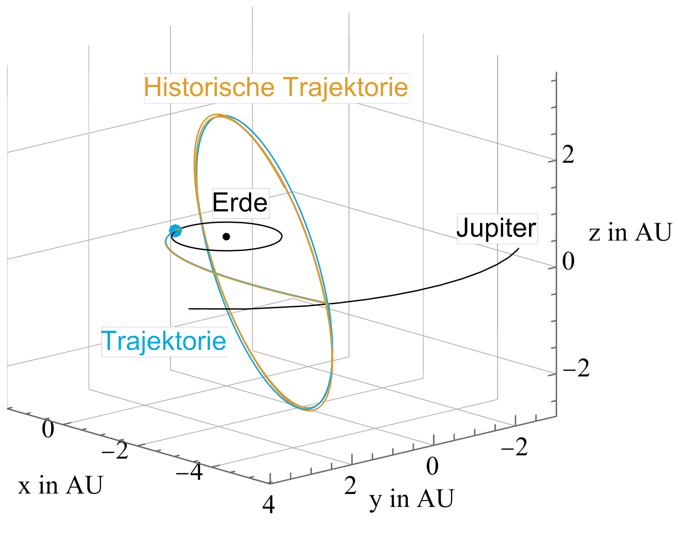


Abbildung 1: Abbildung der historischen Trajektorie, der simulierten Flugbahn und der relevanten Himmelskörpern

[1] `https://www.cosmos.esa.int/web/ulysses/mission-trajectory`

Um die Flugbahn zu simulieren, wird die wirkende Gravitationskraft benötigt. Zu Vereinfachung werden nur die Sonne, die Erde und der Jupiter berücksichtigt. Die Planeten Daten wurden aus Mathematica[2] in täglicher Auflösung exportiert, um die Datenmenge zu reduzieren. Diese Koordinaten werden in der Simulation linear über den Tag interpoliert. In Abbildung 1 sind die Positionen der Planeten, die Historische Trajektorie und ein simulierte Flugbahn dargestellt.

Für die Simulation wurde 2800 Tage betrachtet, was ungefähr 474 Tage für der Transfer zum Jupiter und dann einen Vollständigen Orbit nach dem Swing-by entspricht. Jeder dieser Tage wurde in 1000 Schritten aufgeteilt, da bei mehr Schritten sich die Ergebnisse nicht mehr verändert haben. Für jeden dieser Schritte wurde ein Leap-Frog schritt durchgeführt, um die Position und Geschwindigkeit zu berechnen.

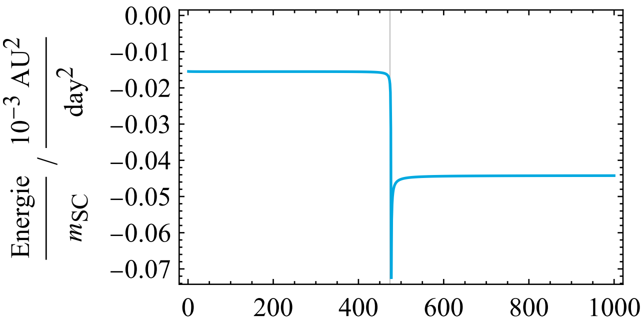


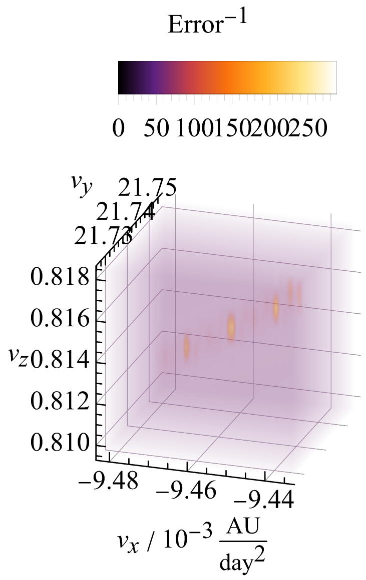
Abbildung 2: Energie und Geschwindigkeit während des Swing-by Manövers

3 Fehlerfunktionen

Damit ein gewünschter Orbit erreicht werden kann, muss zunächst definiert werden, wie dieser aussehen soll. Der Starpunkt des Satelliten liegt natürlich auf der Erde. Den genauen Endzustand kennen wir im Fall der historischen Raumfahrtmission, ansonsten bleibt nur übrig, den gewünschten Zielorbit auf andere Weise zu definieren. In beiden Fällen handelt es sich um ein Randwertproblem, welches sich mithilfe der Shooting-Methode lösen lässt.

Dazu definieren wir eine Fehlerfunktion, die genau dann zu null wird, wenn der Satellit sich am Ende der Simulationszeit auf dem gewünschten Kurs befindet. Um den Satelliten in unserer Simulation auf den finalen Orbit des Ulysses-Satelliten zu schicken, haben wir verschiedene Fehlerfunktionen ausprobiert.

[2]Wolfram Research, Inc., Wolfram|Alpha Notebook Edition, Champaign, IL (2022).



3.1 Vergleich der Endgeschwindigkeit und des Ortes

Um einen Orbit zu erhalten, der dem historischen ähnelt, wurde zunächst *zu einem festen Zeitpunkt* nach dem Swing-By der *Ort und die Geschwindigkeit* des Satelliten mit den historischen Daten verglichen. Da diese keine Geschwindigkeitsinformationen enthalten, wurde dafür die mittlere Geschwindigkeit des Tages angenommen.
Der Ort allein als Vergleichswert reichte nicht aus, da das Programm sonst eine Abkürzungslösung fand, bei der der Satellit an der (im Vergleich zu den historischen Daten) entgegengesetzten Seite des Jupiter vorbeiflog, dadurch jedoch beim Swing-By in die falsche Richtung, und somit nicht auf den gewünschten Orbit abgelenkt wurde.

3.2 Mehre Endpunkte zu festen Zeiten

Eine andere Methode ist der Vergleich der *Orts*koordinaten nach dem Swing-By *zu mehreren Zeitpunkten*, dafür aber *ohne Vergleich der Geschwindigkeiten*. Der Zielorbit wird nun also durch eine Reihe an Wegpunkten definiert, die zu bestimmter Zeit zu erreichen sind.

3.3 Integral des Ort-Fehler-Quadrates

Bei dieser Fehlerfunktion wurde über ein Zeitintervall der Abstand zu den entsprechenden Wegpunkten des Zielorbits errechnet, und die Abstände wurden diesmal quadratisch aufaddiert. Als diskrete Annäherung des Integrals des Fehlerquadrats liefert dies eine

Bewertung des Orbits, indem die Fläche zwischen errechnetem und Zielorbit betrachtet wird, das Quadrieren dient der stärkeren Gewichtung einzelner größerer Abweichungen gegenüber vieler leichter Abweichungen.

3.4 Vergleich von Zielpunkten bei nächster Zeit

Um die Tatsache, dass in der Realität bei der Planung eines Swing-By-Manövers wahrscheinlich keine passenden Daten vorlägen, nicht völlig außer Acht zu lassen, wurden bei der vierten Fehlerfunktion einige Punkte auf dem Wunsch-Orbit ausgewählt - in unserem Fall genau solche, die in den historischen Daten vorlagen, aber es hätten beliebige Punkte sein können. Und dann wurde beim Berechnen der Trajektorie bei jedem Zeitschritt der aktuelle Abstand zu jedem der ausgewählten Punkte gemessen. Der Wert wurde mit einem gespeicherten Wert verglichen, und dieser überschrieben, falls der aktuelle Wert kleiner war. So wurde zu jedem der ausgewählten Punkte der nächstgelegene Punkt der aktuellen Trajektorie gefunden.
Bei dieser Methode muss also der Zeitpunkt des Manövers nicht im Voraus bekannt sein, jedoch besteht die Möglichkeit, dass ein korrekter Orbit nicht erkannt wird, wenn die ausgewählten Punkte des gewünschten Orbits gerade zwischen denen des berechneten Orbits liegen. Solange die Zeitschritte klein genug sind, wird dann aber der Fehler immernoch sehr klein. Es wurde deshalb *bei jedem Sub-Step*, also pro tausendstel Tag, der Abstand verglichen.

Vergleich der Endgeschwindigkeit und Ortes

Mehre Endpunkte zu festen Zeiten

Vergleich von Zielpunkten bei nächster Zeit

Integral des Ort-Fehler-Quadrates

4 Optimierung

Als Optimierungsparameter wurde die Anfangsgeschwindigkeit gewählt, da diese nur 3 Dimensionen hat und somit leichter zu optimieren ist, als z.B. der Transferorbit. Als initialer Wert wurde die echte gemessene Geschwindigkeit verwendet. Wichtig war vor allem, eine Anfangsgeschwindigkeit zu wählen, für welche der Satellit möglichst nahe am Jupiter vorbeifliegt, um die Laufzeit des Suchalgorithmus zu verringern.

Die historischen Daten liegen aber in Form von Kooridinaten in einem sphärischen heliozentrischen Koordinatensystem vor und haben eine Auflösung von 0.01°. Daraus lässt sich eine Unsicherheit der Position von $4\text{ AU} \cdot \sin 0.01° \approx 7 \cdot 10^{-4}$ AU und daraus eine Unsicherheit der Geschwindigkeit von $\sqrt{2} \cdot 7 \cdot 10^{-4} \approx 10^{-3}$ AU/day abschätzen. Solche Unsicherheit ist viel zu groß für die nötige Genauigkeit von 10^{-6} bis 10^{-8} AU/day, denn so viele Stellen muss der Suchalgorithmus optimieren, um die Nullstelle der Fehlerfunktion möglichst genau zu finden. Eine sehr kleine Änderung der Startgeschwindigkeit kann bereits zu einem völlig anderen Orbit führen. Dies dämpft die Erwartungen an die Ergebnisse.
Zur Nullstellensuche wurden verschiedene Verfahren ausprobiert.

4.1 Newton-Verfahren

Die Nullstelle einer stetig differenzierbaren Funktion lässt sich mithilfe des Newton-Verfahrens finden. Dabei wird die Funktion lokal linear durch ihre Ableitung angenähert. Ist die Determinante der Ableitung ungleich Null, so lässt sich das entsprechende Gleichungssystem für jedes gewünschte Ergebnis eindeutig lösen. Die Nullstelle der Taylor-Entwicklung bis zur ersten Ordnung an einem beliebigen Startpunkt wird auf diese Weise bestimmt, und die Lösung wird als neuer Startwert für die nächste Iteration desselben Vorgehens verwendet.
Das Newton-Verfahren konvergiert unter bestimmten Bedingungen (die unsere Fehlerfunktionen erfüllen) lokal quadratisch gegen eine Nullstelle der untersuchten Funktion (ohne Beweis).

4.2 Newton-Verfahren der Optimierung

Um das Newton-Verfahren anwenden zu können, ist es also notwendig, dass die Fehlerfunktion genau so viele Ausgabewerte besitzt, wie es anzupassende Parameter gibt. Deshalb sind die Beschreibungen der Fehlerfunktionen immer Komponentenweise zu verstehen.
In der Optimierung gibt es ein gebäuchliches Verfahren, um *lokale Extremstellen* einer Fehlerfunktion mit eindimensionalem Output zu finden. Dazu wird zunächst der Gradient berechnet, und darauf dann das Newton-Verfahren angewendet. Man rechnet also zunächst die Hesse-Matrix aus, und versucht dann wie im vorigen Abschnitt das Gleichungssystem zu lösen, um um sich iterativ einer Nullstelle zu nähern.
Wir haben im Rahmen dieses Projekts auch dieses Verfahren ausprobiert, indem es auf die quadratische Summe der Komponenten einer unserer Fehlerfunktionen angewendet

wurde. Dann ist nämlich die Nullstelle der vorherigen Fehlerfunktion ein globales, und somit auch ein lokales Minimum der resultierenden Funktion.
Ein Problem dieses Verfahrens ist die lange Laufzeit, denn das numerische Berechnen der Hesse-Matrix läuft in quadratischer Zeit, im Gegensatz zur einfachen Ableitung, bei der der Aufwand nur linear ansteigt. Und da der Einfluss der Startparameter auf den End-Orbit des Satelliten untersucht wurde, musste hier für jeden Optimierungsschritt viel gerechnet werden, weshalb das Programm sehr lange lief, auch unter Verwendung der OMP-Library.

4.3 Gradienten-Verfahren

Ebenfalls zum finden lokaler Minima solcher Funktionen verwendet man das Gradienten-Verfahren. Dabei wird ebenfalls iterativ vorgegangen, jedoch bestehen die Iterationsschritte darin, dem Gradienten ein Stück weit zu folgen. Hier muss allerdings nur einmal abgeleitet werden, was die Rechenzeit wieder erträglich macht.

Es wurden verschiedene Verfahren verwendet, weil manchmal das eine, manchmal das andere Probleme mit der Konvergenz (bei gleichen Anfangsbedingungen) lieferte.

5 Diskussion der Ergebnisse und Probleme

Bei den Berechnungen der Trajektorie konnten mit keiner der benutzen Methoden eine exakte Reproduktion der historischen Daten erreicht werden. Es gibt mehrere mögliche Fehlerquellen, die die Genauigkeit der Berechnungen einschränken. Eines der Probleme, welches alle bei diesem Projekt implementierten numerischen Optimierungsverfahren haben ist, dass diese nur lokal konvergent sind. Das entscheidende Problem der lokalen Konvergenz kann dabei mittels der bei dem Projekt implementierten „Grid-Search-Methode“ gezeigt werden. Bei der „Grid-Search-Methode“ werden die Werte der verwendeten Fehlerfunktionen für ein feines Gitter von Anfangsgeschwindigkeiten berechnet. Das Gitter von Anfangsgeschwindigkeiten umfasst dabei einen gewissen Geschwindigkeitsbereich welcher sich um die gesuchte Anfagsgeschwindigkeit erstreckt. Dabei hat sich gezeigt, dass die untersuchten Fehlerfunktionen mehrere lokale Minima im Bereich der gesuchten Lösung besitzen, welche nur sehr geringe Abstände zueinander haben. Das hat zur Folge, dass die lokal konvergenten Optimierungsverfahren, nicht immer gegen die gesuchte Lösung konvergieren. Besonders beim Newton-Verfahren zur Suche von Minima (in Kapitel 4.2 beschrieben) hat sich gezeigt, dass dieses aufgrund der vielen lokalen Minima, für diese Art von Problemstellung eher ungeeingnet ist.

6 Fazit