
Zur Entstehung der S-Sterne um das super-massereiche schwarze Loch im Zentrum der Milchstraße

Christian Franik



München 2011

Zur Entstehung der S-Sterne um das super-massereiche schwarze Loch im Zentrum der Milchstraße

Bachelor-Arbeit

an der

Ludwig–Maximilians–Universität

eingereicht von

Christian Franik

(Matr. Nr.: 10066564)

geboren am 18.05.1987 in D-92637 Weiden

betreut von

Prof. Dr. Andreas Burkert

und

Dr. Marc Schartmann

München, den 20. September 2011

Evaluator: Prof. Dr. Andreas Burkert

Tag der mündlichen Prüfung: _____

Inhaltsverzeichnis

1. Einleitung	1
2. Das Modell	4
2.1. Eigenschaften der Körper	4
2.1.1. Eigenschaften des schwarzen Lochs	4
2.1.2. Eigenschaften des Modellsterns	5
2.1.3. Eigenschaften der Gaswolke	5
2.2. Das Modell in den Programmen	7
3. Die Programme	8
3.1. Benötigte Grundlagen	8
3.2. Das Programm zum 3-Körper-Problem	10
3.3. Das Programm für den Phasenscan	11
4. Ergebnisse und Diskussion	14
4.1. Das Einfangszenario	14
4.2. Das Vorbeiflugszenario	19
Zusammenfassung	21
Danksagung	23
A. Programm 1 (3 Körper)	25
A.1. Code	25
A.2. Format der input-/output-Dateien	27
B. Programm 2 (Phasenscan)	28
B.1. Code	28
B.2. Format der input-/output-Dateien	32
Literaturverzeichnis	33
Abbildungsverzeichnis	34
Selbstständigkeitserklärung	36

1. Einleitung

Das Zentrum unserer Milchstraße ist ein in einzigartiger Weise zugängliches Laboratorium zur Erforschung der Eigenschaften und Entwicklung galaktischer Zentren, da es sich im Vergleich zu anderen Galaxien in sehr kleinem Abstand zu uns befindet (etwa 8 Kiloparsec (kpc)) [Bartko et al., 2009]. Da die Extinktion des Lichts im sichtbaren Bereich viel zu groß ist, um in diesem Spektralbereich mit Teleskopen bis zum Galaktischen Zentrum vorzustößen (Die Extinktion im visuellen Spektralbereich A_V entspricht ungefähr 30 Größenklassen), wird dort hauptsächlich im nahen Infraroten oder im Radiobereich beobachtet. Diverse Beobachtungen (unter vielen anderen [Paumard et al., 2006; Bartko et al., 2009; Eisenhauer et al., 2003; Schödel et al., 2002]) ergänzen in den letzten Jahr(zehnt)en unser Bild davon, wie es innerhalb des ersten Parsec um das Galaktische Zentrum aussieht:

In Entfernungen von 0,03 - 0,5 pc vom Zentrum findet man etwa 100 Wolf-Rayet-Sterne und Sterne des O- und B-Typs, welche sich zu einem großen Teil in einer von zwei Ebenen befinden. Die Sterne in einer der beiden Ebenen bzw. Systeme besitzen jeweils nahezu gleichen Drehimpuls, was für eine gemeinsame Entstehungsgeschichte der Sterne in einem System spricht. Es gibt ein System mit Umlauf im Uhrzeigersinn (CWS: clockwise system) und eines mit Umlauf gegen den Uhrzeigersinn (CCWS: counter clockwise system). Die Sterne des CWS haben im Durchschnitt recht geringe numerische Exzentrizitäten von $0,36 \pm 0,06$, während die Sterne des CCWS höhere Exzentrizitäten aufweisen [Alexander, 2005].

Innerhalb dieses Bereichs findet man die sog. S-Sterne, von denen inzwischen schon etwa 30 bekannt sind. Haupteigenschaften dieser Sterne sind große Exzentrizitäten sowie eine isotrope Verteilung ihrer Bahnen, also keine bevorzugte Umlaufebene [Eisenhauer et al., 2003].

Im Galaktischen Zentrum selbst finden wir schließlich eine starke Radioquelle mit der Bezeichnung Sgr A*. Bahnvermessungen der S-Sterne, welche Keplerbahnen mit nur einigen Jahren Umlaufzeit besitzen [Schödel et al., 2002], zeigen, dass es sich bei Sgr A* um ein Objekt von etwa $4 \cdot 10^6$ Sonnenmassen (M_\odot) handelt. Da sich diese Masse in einem Durchmesser von nur einigen Lichtstunden befinden muss,

ist man sich inzwischen zweifelsfrei sicher, dass es sich um ein (super-massereiches) schwarzes Loch handelt (supermassive black hole, SMBH) [Ghez et al., 2005]. Eine Übersicht des innersten Bereichs um das zentrale SMBH zeigt Abbildung 1.1.

Daher kommt auch das große Rätsel um die S-Sterne: Da sie relativ jung sind, die starken Gezeitenkräfte des schwarzen Lochs eine Entstehung so nahe am Zentrum aber verhindert hätten, muss es andere Erklärungen für ihre Existenz geben, z.B. Transport aus weiter entfernten Bereichen durch ein um das SMBH umlaufendes kleineres schwarzes Loch [Merritt et al., 2009] oder Wechselwirkungen zwischen den Sternen des CWS/CCWS untereinander, was zu großen Exzentrizitäten und beliebigen Umlaufebenen führen kann [Levin, 2007].

In dieser Arbeit soll nun eine weitere Theorie überprüft werden: Einer der Sterne des CWS oder CCWS wird durch Wechselwirkung von einfallenden Gaswolken so abgelenkt, dass seine resultierende Bahn innerhalb der des CWS/CCWS liegt, opti-

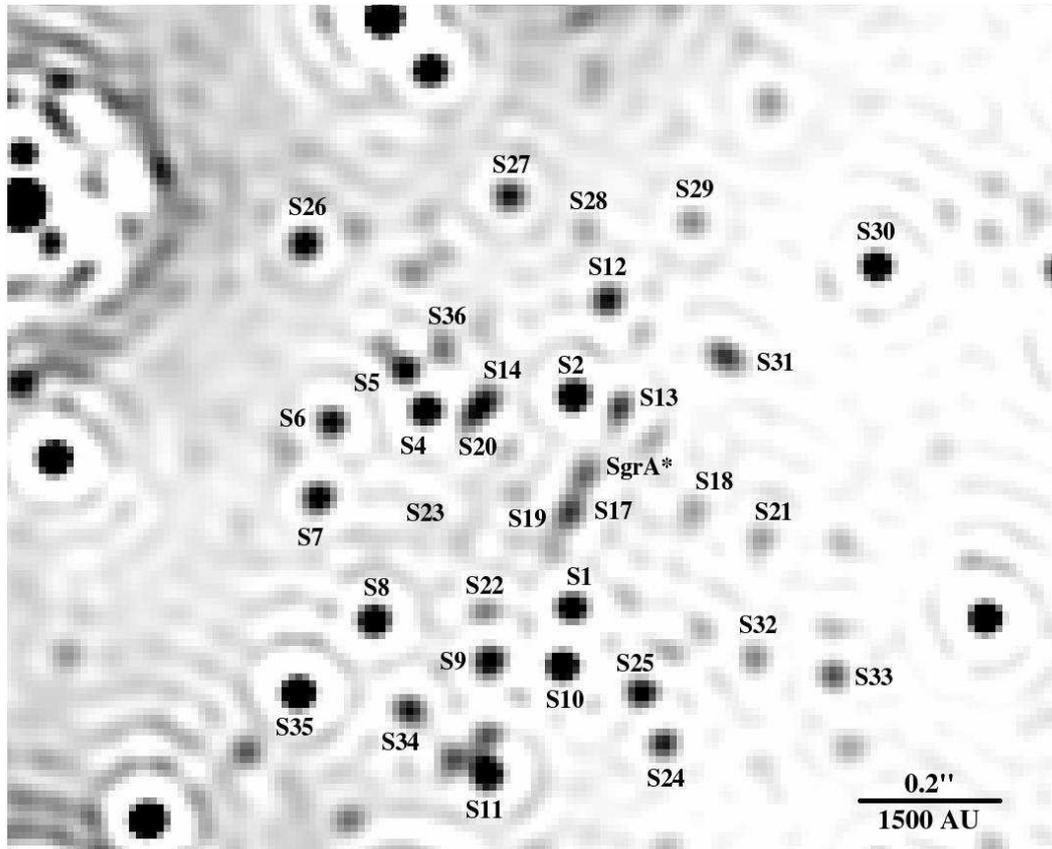


Abb. 1.1.: Die regelmäßig beobachteten S-Sterne. Dieses Bild ist ein Lucy-Richardson-dekonvolviertes K-Band NAOS/CONICA-Bild, aufgenommen im Jahr 2005. Zur Zeit der Belichtung war Sgr A* im nahen Infraroten aktiv und ist deswegen sichtbar. Das Bild wurde ursprünglich in [Trippe et al., 2006] veröffentlicht.

malerweise mit größerer Exzentrizität, und somit zu den S-Sternen gezählt werden muss. Mittels numerischer Simulationen soll festgestellt werden, ob eine Wechselwirkung „Stern - Gaswolke“ ausreichend für eine solche Bahnveränderung ist. Auch wenn schon fertige Programme existieren, die gravitative Wechselwirkungen simulieren können, wurden für diese Arbeit selbst angefertigte Programme verwendet, um einen Einblick in die Programmierung numerischer Simulationen zu bekommen und ein auf das Problem maßgeschneidertes Programm zur Verfügung zu haben.

2. Das Modell

Wie bereits in der Einleitung erwähnt, wird in dieser Arbeit untersucht, ob die Wechselwirkung eines Sterns mit einer Gaswolke ausreicht, um ihn zu einem S-Stern zu machen. In der Realität könnte es z.B. so aussehen, dass aus einer Gaswolke bereits die Sterne des oben beschriebenen CCWS entstanden sind (s. [Alig et al., 2011]). Eine zweite einfallende Gaswolke könnte diese Sterne dann so beeinflussen, dass ihre Bahnen enger gebunden sind. Im weiteren Verlauf würde aus dieser 2. Gaswolke dann die Sterne des CWS werden. In diesem Kapitel sollen nun die Annahmen, Vereinfachungen und Bedingungen erläutert werden, mit denen in den Simulationen gearbeitet wurde. Dazu gehören sowohl die Eigenschaften der Körper selbst als auch Einschränkungen und Bedingungen bezüglich der Simulationen.

2.1. Eigenschaften der Körper

2.1.1. Eigenschaften des schwarzen Lochs

In sämtlichen Simulationen wurde das super-massereiche schwarze Loch als ruhend im Ursprung zur Zeit $t = 0$ angenommen. Laut [Gillessen et al., 2009] besitzt es eine Masse von $(4,40 \pm 0,77) \cdot 10^6 M_{\odot} \approx 7,956 \cdot 10^{36}$ kg. Da andere Publikationen (u.a. [Ghez et al., 2005; Lu et al., 2008]) jedoch meistens eine geringere Masse von $4,0 \cdot 10^6 M_{\odot}$ angeben bzw. verwenden, wurde in dieser Arbeit mit dieser Masse gearbeitet. Des weiteren wurde diese Masse als in einem Punkt konzentriert angenommen, was völlig legitim ist, wenn man in Skalen rechnet, welche weit größer sind als der Ereignishorizont des schwarzen Lochs selbst. Folgende kleine Rechnung zeigt dies:

Der Schwarzschild-Radius r_S , also im Prinzip die Ausdehnung eines schwarzen Lochs oder die Grenze des Ereignishorizonts, einer Masse M ist gegeben durch

$$r_S = \frac{2GM}{c^2}. \quad (2.1)$$

Für ein schwarzes Loch mit Masse $4 \cdot 10^6 M_\odot \approx 7,956 \cdot 10^{36}$ kg ergibt das etwa $r_S \approx 1,181 \cdot 10^{10}$ m $\approx 3,829 \cdot 10^{-7}$ pc $\approx 0,08$ AU (Astronomical Unit/Astronomische Einheit), was viel kleiner ist als die Periapsisdistanz des bisher dem schwarzen Loch nächsten S-Stern S2 von 130 AU [Schödel et al., 2002].

2.1.2. Eigenschaften des Modellsterns

Wie in der Einleitung erwähnt, sind die Sterne der CWS/CCWS hauptsächlich Sterne des Typs O und B, also große junge Sterne mit Massen von über $10 M_\odot$ [Lu et al., 2008]. Andere Arbeiten, welche die bereits existierenden S-Sterne untersuchten (eine Zusammenfassung findet man in [Alexander, 2005]), klassifizierten diese als B0-9 Hauptreihen-Sterne. Was die Bahnen der Sterne angeht, so fand man, dass die Sterne des CWS geringe Exzentrizitäten von $e \approx 0,2 - 0,5$ aufweisen [Paumard et al., 2006; Bartko et al., 2009].

Unter Berücksichtigung dieser Daten wurde für den Modellstern eine Masse von $15 M_\odot$ benutzt, dessen Anfangs-Exzentrizität 3 Werte zwischen den oben genannten Werten besitzt: 0,2881139, 0,3861792 und 0,4769802. Diese 'krummen' Werte erhält man, wenn man in einem 2-Körper-Problem die in Tab. 2.1 angegebenen Werte für Orte und Geschwindigkeit verwendet.

num. Exzentrizität	x_0 [pc]	y_0 [pc]	$v_{x,0}$ [km/s]	$v_{y,0}$ [m/s]
0,2881139	0,1	0,0	0,0	$3,50 \cdot 10^5$
0,3861792	0,1	0,0	0,0	$3,25 \cdot 10^5$
0,4769802	0,1	0,0	0,0	$3,00 \cdot 10^5$

Tab. 2.1.: Die Anfangsbedingungen für die verwendeten Sternbahnen aus Abb. 2.1

Wie man sieht, sind die Anfangsbedingungen, die in den input-Dateien stehen, recht einfach. In Abbildung 2.1 sind die verwendeten Bahnen veranschaulicht.

2.1.3. Eigenschaften der Gaswolke

Bei der Modellierung der Wolke mussten die meisten Vereinfachungen vorgenommen werden, da die Berechnungen ansonsten unverhältnismäßig kompliziert gewesen wären. Ziel dieser Arbeit war es ja zu prüfen, ob eine solche Wolke prinzipiell dazu imstande ist, einen Stern signifikant aus seiner Bahn zu lenken.

Typische Gaswolken der Art, wie sie uns interessieren und wie sie auch im Zentrum

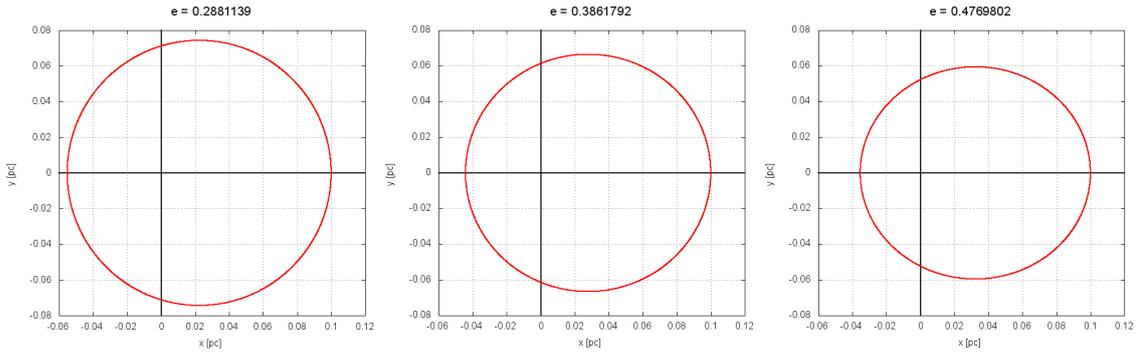


Abb. 2.1.: Die verwendeten Bahnen für den Teststern. Das schwarze Loch befindet sich jeweils am Ursprung. Die Umlaufrichtung ist gegen den Uhrzeigersinn

unserer Milchstraße vorkommen, besitzen Massen von etwa $10^5 M_\odot$ und Ausdehnungen von 1,8-10 pc [Alig et al., 2011]. In den Programmen wurde die Wolke als perfekt kugelförmig angenommen, die ihre Form auch während des gesamten Simulationsverlaufs nicht ändert. Des Weiteren besitzt sie überall eine konstante Dichte. Dieses Modell hat insbesondere den Vorteil, dass damit die Wolke als Punktmasse behandelt werden kann, wenn sich ein Körper außerhalb ihres Radius befindet. Befindet sich ein Körper innerhalb des Radius der Wolke, so kann mit einem einfachen Term $\frac{R^3}{\rho^3}$ (mit ρ = Radius der Wolke und R = Abstand des Körpers vom Zentrum der Wolke) die Kraft auf den Körper korrigiert werden. In [Alig et al., 2011] wird die Bewegung der Wolke über mehrere ihrer Umläufe simuliert, bis sie schließlich als Akkretionsscheibe um das schwarze Loch endet. Hier werde ich diese Bewegung aufteilen: Einmal wird die Wolke in einer Entfernung von knapp über 2 pc vom schwarzen Loch mit Geschwindigkeit 0 km/s 'losgelassen', was in der Folge dazu führt, dass der Schwerpunkt der Wolke durch das schwarze Loch fliegt und im weiteren Verlauf die Wolke vom schwarzen Loch 'eingefangen' wird. Im Folgenden werde ich dies als 'EinfangszENARIO' bezeichnen. Im zweiten Fall wird die Wolke mit einer Anfangsgeschwindigkeit von 20 km/s am schwarzen Loch vorbeifliegen (VorbeiflugszENARIO, s. auch Abb. 2.3)

Da in dieser Arbeit die Wechselwirkung mit der Wolke möglichst groß sein soll, wird sie so dicht wie möglich gewählt: Radius von 2 pc (nahe der Untergrenze von 1,8 pc) und eine Masse von $10^5 M_\odot$ (Obergrenze). Um zu berücksichtigen, dass die Wolke nicht immer aus der gleichen Richtung bezüglich der Bahn des Sterns auf das System zufliegt, wurden Simulationen durchgeführt, in denen die Gaswolke aus positiver und negativer x- und y-Richtung einfliegt (s. Abb. 2.3).

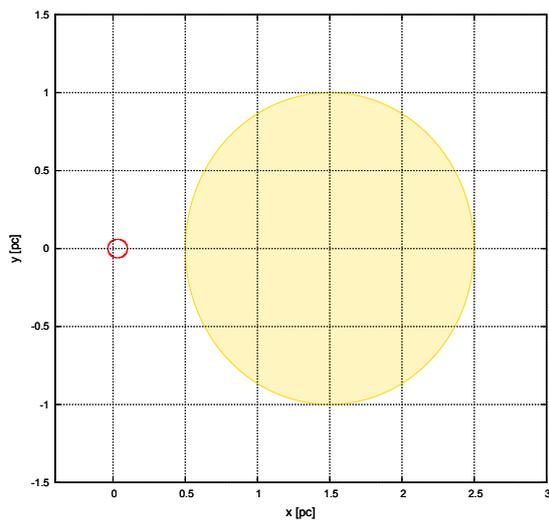


Abb. 2.2.: Maßstabsgetreuer Vergleich der Modellwolke mit der verwendeten Sternbahn mit $e = 0,4769802$.

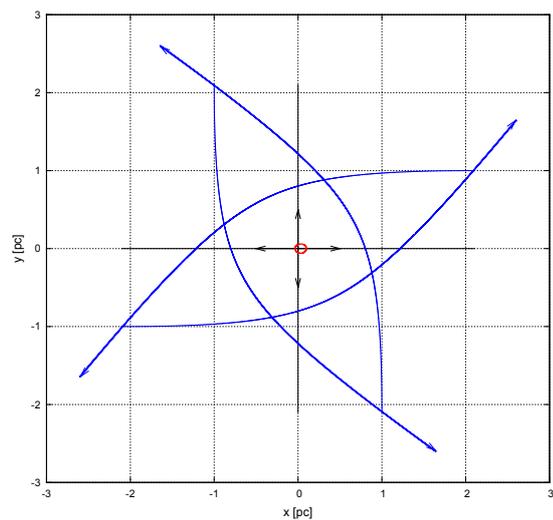


Abb. 2.3.: Die verwendeten Flugbahnen des Schwerpunkts der Wolke: In Schwarz die vier Einflogrichtungen im EinfangszENARIO, welche im verwendeten Koordinatensystem 0° (rechts), 90° (oben), 180° (links) und 270° (unten) entsprechen; in Blau die vier kompletten Flugbahnen für das Vorbeiflugszenario. Die Sternbahn wurde rot gezeichnet

2.2. Das Modell in den Programmen

Auch in der Programmierung selbst wurde eine Vereinfachung vorgenommen: Die Beschränkung auf zwei Dimensionen. Die dritte Dimension ist in den Programmen zwar implementiert, jedoch wurde in allen input-Dateien die z-Komponente sowohl der Positionen als auch der Geschwindigkeiten auf 0 gesetzt.

Als Zeit-Schrittweite wurde in beiden Programmen 100 Tage gewählt, als Simulationsdauer für die 3-Körper-Simulationen (also Programm 1 und der zweite Teil in Programm 2) 9.200.000 Tage, was etwa 25.000 Jahren entspricht.

3. Die Programme

In diesem Kapitel werde ich einen Überblick zur Entwicklung der benutzten Programme geben. Das erste Programm ist in der Lage, ein beliebiges 3-Körper-Problem zu simulieren, welches die kompletten Positions- und Geschwindigkeitsdaten der 3 Körper speichert, während das zweite Programm das eigentliche Arbeitstier dieser Arbeit ist und in einem Durchlauf mehrere tausend 3-Körper-Simulationen durchführt, dabei jedoch nur Bahndaten wie die num. Exzentrizität speichert.

Die 2 Programme wurden in der Programmiersprache C geschrieben. Auch wenn sie mit C++-Compilern (dem C++-Compiler der GNU Compiler Collection (GCC) in einer Linux-Umgebung und dem im Microsoft Visual C++ enthaltenen Compiler in einer Windows-Umgebung) kompiliert wurden, so enthalten sie nur Funktionen der Sprache C. Beispielsweise wurden für Bildschirm- und Dateiausgabe nicht die in C++ üblichen Streams (fstream, ifstream, ofstream, cout, cin) verwendet, sondern die dafür zuständigen C-Funktionen (fscanf, fprintf, printf, scanf).

3.1. Benötigte Grundlagen

In den erstellten Programmen wird die Bewegung von Sternen, Stern-ähnlichen Objekten (hier ein schwarzes Loch) und Gaswolken unter dem Einfluss von gravitativen Kräften untersucht. Da sich unsere Objekte alle mit nicht-relativistischen Geschwindigkeiten bewegen, dürfen wir die klassische Formel von Newtons Gravitationstheorie benutzen:

$$\vec{F}(\vec{r}) = -G \cdot \frac{mM}{r^3} \cdot \vec{r} = m \cdot \dot{\vec{v}} \quad (3.1)$$

($\vec{F}(\vec{r})$ Gravitationskraft am Ort \vec{r} ; G Gravitationskonstante; m,M Massen der Körper; r Entfernung der Körper voneinander; $\dot{\vec{v}}$ Beschleunigung des Körpers mit Masse m). Um nun am Computer damit physikalische Berechnungen durchzuführen,

muss man ein paar Vorarbeiten leisten.

Die Bahn eines Körpers ist zu jedem Zeitpunkt durch seinen Ort sowie seine Geschwindigkeit bestimmt, die wiederum durch Formel (2.1) bestimmt sind, wenn man die Anfangsbedingungen kennt. Für die Lösung von Differentialgleichung (2.1) kann man nun eine Diskretisierung vornehmen:

$$\vec{F} = m \cdot \frac{d^2x}{dt^2} = m \cdot \frac{\Delta\vec{v}}{\Delta t} \quad (3.2)$$

Die Numerik kommt ins Spiel, wenn man nun $\Delta\vec{v}$ durch $\vec{v}_{i+1} - \vec{v}_i$ ersetzt, was auf ein einfaches Iterationsverfahren für die Geschwindigkeiten führt. Die Orte \vec{x}_i lassen sich aus den nun bekannten Geschwindigkeiten ähnlich einfach berechnen:

$$\vec{v}_{i+1} = \vec{v}_i + \frac{\vec{F}_i}{m} \Delta t = \vec{v}_i - GM \frac{\vec{x}_i}{|\vec{x}_i|^3} \Delta t \quad (3.3)$$

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_i \Delta t \quad (3.4)$$

Ein solches Verfahren, in dem einfach zu jedem vollen Zeitschritt Geschwindigkeit und Ort berechnet werden, wird Einschritt- oder Euler-Verfahren genannt. Die ersten Programmversionen arbeiteten nach diesem Prinzip, doch wegen zu großer Fehler in der Energieerhaltung, welche ein Maß für die Qualität der Simulation sind, habe ich mich zur Verwendung eines sog. Leapfrog-Verfahrens entschieden. Dabei werden zuerst die Geschwindigkeiten nach einem halben Zeitschritt berechnet (sog. Kick), dann mithilfe dieser Werte die Orte zum ganzen Zeitschritt (sog. Drift) und schließlich werden die Geschwindigkeiten des ganzen Zeitschritts mithilfe der Orte des Drifts berechnet:

$$\vec{v}_{i+\frac{1}{2}} = \vec{v}_i - GM \frac{\vec{x}_{i-\frac{1}{2}}}{|\vec{x}_{i-\frac{1}{2}}|^3} \cdot \frac{\Delta t}{2} \quad (3.5)$$

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_{i+\frac{1}{2}} \cdot t \quad (3.6)$$

$$\vec{v}_{i+1} = \vec{v}_{i+\frac{1}{2}} - GM \frac{\vec{x}_{i+1}}{|\vec{x}_{i+1}|^3} \cdot \frac{\Delta t}{2} \quad (3.7)$$

Die obigen Formeln sind gültig, wenn man den Körper mit Masse M als ruhend

im Ursprung annimmt. Das ist natürlich nur selten der Fall, auch wenn es in einem 2-Körper-System 'SMBH - Stern' eine recht gute Näherung darstellt. Da sich im Allgemeinen aber alle Körper bewegen, erhält man die endgültigen Formeln, wenn man zu jedem Iterationsschritt den Abstand zwischen den entsprechenden Körpern berechnet und einsetzt:

$$\vec{R}_i = \vec{x}_i^{(2)} - \vec{x}_i^{(1)} \quad (3.8)$$

$$\vec{v}_{i+\frac{1}{2}} = \vec{v}_i - GM \frac{\vec{R}_{i-\frac{1}{2}}}{|\vec{R}_{i-\frac{1}{2}}|^3} \cdot \frac{\Delta t}{2} \quad (3.9)$$

$$\vec{x}_{i+1} = \vec{x}_i + \vec{v}_{i+\frac{1}{2}} \cdot t \quad (3.10)$$

$$\vec{R}_{i+1} = \vec{x}_{i+1}^{(2)} - \vec{x}_{i+1}^{(1)} \quad (3.11)$$

$$\vec{v}_{i+1} = \vec{v}_{i+\frac{1}{2}} - GM \frac{\vec{R}_{i+1}}{|\vec{R}_{i+1}|^3} \cdot \frac{\Delta t}{2} \quad (3.12)$$

($\vec{x}^{(n)}$ = Ort des n-ten Körpers)

3.2. Das Programm zum 3-Körper-Problem

Der vollständige Quellcode zum Programm und das Format seiner input-Dateien steht in Anhang A. Zu gegebenen Anfangsbedingungen (also Positionen und Geschwindigkeiten) für das schwarze Loch, Stern und Wolke, welche durch zwei Dateien „inputStern3Koeper.dat“ und „inputWolke.dat“ (dies ist die selbe „inputWolke.dat“ wie für das zweite Programm, s. unten) in das Programm eingelesen werden, simuliert es einen Vorbeiflug der Wolke am System 'Schwarzes Loch - Stern'. Die Bedingungen und Annahmen, die dabei getroffen werden, wurden in Kapitel 1 besprochen. Auch die Dauer und Schrittweite der Simulation werden mithilfe der input-Dateien eingelesen.

In jedem Iterationsschritt i passiert nun folgendes:

- Zurücksetzen der kinetischen und potentiellen Energie eines jeden Körpers auf 0.
- Berechnung der Geschwindigkeiten und Positionen $\vec{v} = (v_x, v_y, v_z)$ bzw. $\vec{r} = (x, y, z)$ jedes Körpers j durch Wechselwirkung mit Körper k (Kick \rightarrow Drift \rightarrow

Kick). Eine Sprunganweisung 'continue' stellt sicher, dass ein Körper nicht mit sich selbst in Wechselwirkung tritt.

- Neuberechnung der kinetischen und potentiellen Energien der einzelnen Körper sowie des gesamten Systems.
- Schreiben der Geschwindigkeits-, Positions- und Energie-Daten in output-Dateien „3KoerperData.dat“ und „3KoerperEnergies.dat“.

Nach dem kompletten Durchlauf der Schleife (also nach Erreichen der Maximaldauer der Simulation), werden die Dateien gespeichert und geschlossen und das Programm beendet.

Die Stärke dieses Programms ist die vollständige Reproduktion der Bahnen der drei Körper, die man dann auch mithilfe von Plot-Programmen wie gnuplot visuell anschaulich darstellen kann. Die Bahnen in den Abbildungen 2.1, 2.2 und 2.3 wurden mit diesem Programm berechnet. Offensichtlich ist es aber kaum dazu geeignet, eine systematische Untersuchung über einen großen Parameterbereich für die Körper anzustellen, da es nahezu unendlich viele Kombinationsmöglichkeiten gibt. Abhilfe schafft da das zweite Programm.

3.3. Das Programm für den Phasenscan

Das zweite Programm stellt eine Erweiterung des obigen ersten Programms dar. Der vollständige Quellcode sowie das Format seiner input-Dateien steht in Anhang B. In diesem Programm gibt man nur die Anfangsbedingungen für die Wolke an (und natürlich für das schwarze Loch, allerdings sind diese immer die gleichen; s. Kapitel 2). Für den Stern wird nur seine Bahn festgelegt, nicht aber seine Position auf der Bahn. Für jede Position auf seiner Bahn, also für jede Phase, und bei gleichbleibender Startposition der Wolke und des SMBH wird eine 3-Körper-Simulation wie im ersten Programm durchlaufen und dann wird festgestellt, ob die Bahn des Sterns beim Vorbeiflug der Wolke relativ zum schwarzen Loch verändert wurde. Anschließend wird nach Gesetzmäßigkeiten gesucht, unter welchen Bedingungen überhaupt eine „S-Stern-Bahn“ zustande kommt. Die genaue Struktur des Programms ist also dreigeteilt und folgt dem folgenden Algorithmus:

1. Teil: 2-Körper-Simulation der ungestörten Kepler-Bahn des Sterns um das schwarze Loch zur Bahndaten-Bestimmung vor dem Vorbeiflug der Wolke:

- Einlesen der Schrittweite und der Anfangsbedingungen für den Stern von einer Datei „inputStern.dat“ (Geschwindigkeit und Ort zur Zeit $t = 0$ legen eine bestimmte Bahn fest).
 - Berechnung der Bahnpunkte des Sterns, bis ein Umlauf ausgeführt wurde ($0 \leq \text{Phase} \leq 2\pi$) oder der Abstand zwischen Stern und schwarzem Loch mehr als 0,5 pc entspricht, was bedeuten würde, dass er nicht zum CWS/CCWS gezählt werden kann.
 - Berechnung der einfachen Bahnparameter: Periapsisdistanz, Apoapsisdistanz, große und kleine Halbachse, numerische Exzentrizität.
 - Speichern der Position und Geschwindigkeit jedes Bahnpunktes in einer Datei „starPositions.dat“.
- 2. Teil: 3-Körper-Simulationen für jeden in Teil 1 berechneten Bahnpunkt mit Vorbeiflug der Wolke:
 - Einlesen der Schrittweite, Simulationsdauer sowie der Anfangsbedingungen für die Wolke von einer Datei „inputWolke.dat“.
 - Einlesen der Anfangsbedingungen für den Stern aus der Datei „starPositions.dat“: Jeder Bahnpunkt gibt ein mögliches 3-Körper-Szenario vor.
 - Die eigentliche 3-Körper-Simulation, wie sie im ersten Programm abläuft, für jeden in Teil 1 berechneten Bahnpunkt.
 - Speichern der Positionen und Geschwindigkeiten des Sterns nach dem Vorbeiflug der Wolke in einer Datei „starPositionsFinal.dat“ für jedes simulierte 3-Körper-Szenario.
- 3. Teil: 2-Körper-Simulation der ungestörten Kepler-Bahn des Sterns um das schwarze Loch zur Bahndaten-Bestimmung nach dem Vorbeiflug der Wolke:
 - Einlesen der Schrittweite und der Anfangsbedingungen für den Stern aus der Datei „starPositionsFinal.dat“.
 - Neuberechnung der Bahnpunkte des Sterns während eines Umlaufs.
 - Neuberechnung der einfachen Bahnparameter: Perizentrumsdistanz, Apozentrumsdistanz, große und kleine Halbachse, numerische Exzentrizität.
 - Speichern der Bahndaten in einer Datei „data.dat“.

Dieses Programm ergänzt das erste, wie oben schon erwähnt, indem es Simulationen über einen großen Parameterbereich durchführt und Bahndaten zur Verfügung stellt. Es erstellt jedoch nicht für jede Simulation eine Datei mit Positionen, Geschwindigkeiten und Energien der Körper zu jedem Iterationsschritt, da dies zu einem zu großen Datenvolumen führen würde (je nach Schrittweite mehrere tausend Dateien mit jeweils mehreren MB). Einzelne interessante Fälle könnten jedoch durch entsprechendes Umschreiben der input-Dateien mit dem ersten Programm durchlaufen und mit plot-Programmen visualisiert werden.

4. Ergebnisse und Diskussion

4.1. Das Einfangszenario

Wie man aus den Grafiken 4.1 - 4.3 ersehen kann, sind die Störungen der Sternbahn, die die Gaswolke im Einfangszenario hervorruft, minimal und kaum dazu in der Lage, einen solchen Stern signifikant aus seiner Bahn zu lenken, so dass er zu den S-Sternen gezählt werden kann (Tab. 4.2). Man kann jedoch innerhalb dieser kleinen Veränderungen Gesetzmäßigkeiten erkennen, wie sich der kleinste bzw. größte Abstand zum schwarzen Loch (Periapsis- (p) bzw. Apoapsisdistanz (a)) sowie die numerische Exzentrizität (e) ändern. Die Werte in den folgenden Tabellen sind Mittelwerte aus den in den Grafiken 4.1 - 4.3 dargestellten Simulationen:

Wie man erkennen kann, hat die Wolke den 'größten' Einfluss auf die Bahn, wenn sie senkrecht zur Apsidenlinie der Sternbahn einfliegt (90° , 270°) und kaum Einfluss, wenn sie parallel dazu einfliegt (0° , 180°). Die beiden Einflugrichtungen 90° und 270° haben jedoch unterschiedliche Wirkung auf die Bahn: Während bei 90° p um bis zu 1% größer, a um bis zu 0,5% kleiner und e um bis zu 2,2% kleiner wird, ist es bei 270° umgekehrt: p wird kleiner, a wird größer und e wird ebenfalls größer, und zwar um

e_0	$\Delta p(0^\circ)$ [%]	$\Delta p(90^\circ)$ [%]	$\Delta p(180^\circ)$ [%]	$\Delta p(270^\circ)$ [%]
0,2881139	$-0,012 \pm 0,008$	$+0,903 \pm 0,005$	$-0,010 \pm 0,006$	$-0,906 \pm 0,010$
0,3861792	$-0,009 \pm 0,006$	$+0,974 \pm 0,006$	$-0,008 \pm 0,006$	$-0,976 \pm 0,009$
0,4769802	$-0,006 \pm 0,006$	$+1,057 \pm 0,005$	$-0,008 \pm 0,007$	$-1,057 \pm 0,008$
e_0	$\Delta a(0^\circ)$ [%]	$\Delta a(90^\circ)$ [%]	$\Delta a(180^\circ)$ [%]	$\Delta a(270^\circ)$ [%]
0,2881139	$+0,006 \pm 0,006$	$-0,502 \pm 0,005$	$0,000 \pm 0,010$	$+0,496 \pm 0,010$
0,3861792	$+0,004 \pm 0,005$	$-0,434 \pm 0,006$	$-0,005 \pm 0,010$	$+0,428 \pm 0,008$
0,4769802	$+0,002 \pm 0,005$	$-0,378 \pm 0,006$	$-0,005 \pm 0,010$	$+0,370 \pm 0,008$
e_0	$\Delta e(0^\circ)$ [%]	$\Delta e(90^\circ)$ [%]	$\Delta e(180^\circ)$ [%]	$\Delta e(270^\circ)$ [%]
0,2881139	$+0,023 \pm 0,008$	$-2,241 \pm 0,007$	$+0,021 \pm 0,010$	$+2,231 \pm 0,010$
0,3861792	$+0,010 \pm 0,005$	$-1,554 \pm 0,005$	$+0,008 \pm 0,006$	$+1,545 \pm 0,007$
0,4769802	$+0,004 \pm 0,004$	$-1,163 \pm 0,004$	$+0,004 \pm 0,006$	$+1,155 \pm 0,006$

Tab. 4.1.: Die Veränderung der Periapsisdistanz (Δp), der Apoapsisdistanz (Δa) und der numerischen Exzentrizität (Δe) durch die Störung der Wolke, jeweils für die verschiedenen Einflugrichtungen der Wolke.

		notwendig [%]	beobachtet [%]
$e_0 = 0,2881139$	a	-50	-0,502
	e	+38,8 bis +229,7	+2,231
$e_0 = 0,3861792$	a	-50	-0,434
	e	+3,6 bis +146,0	+1,545
$e_0 = 0,4769802$	a	-50	-0,378
	e	-16,1 bis +99,2	+1,155

Tab. 4.2.: Die Veränderungen von a und e, die notwendig wären, um den Stern zu den S-Sternen zu zählen (für diese gilt $0.4 \leq e \leq 0.95$ [Eckart et al., 2002]), sowie deren maximal in den Simulationen beobachtete Veränderung

jeweils fast die gleichen Beträge. Sieht man sich die Geometrie des Systems z.B. in Abb 2.3 an, so erscheint dies völlig logisch und erwartungsgemäß, da beim Übergang von 90° - zu 270° -Einflugrichtung p und a im Grunde genommen vertauscht werden. Sind die Einflugrichtungen parallel zur Apsidenlinie so sind die Änderungen von p und a so gut wie kaum vorhanden: p wird nur um maximal 0,02% kleiner, während a nur um maximal 0,015% um seinen ursprünglichen Wert oszilliert. e vergrößert sich in beiden Fällen, jedoch nur um maximal 0,031%.

Interessant ist auch die Abhängigkeit der Änderungen von der Anfangsexzentrizität: Für den Fall von einer Einflugrichtung senkrecht zur Apsidenlinie werden mit zunehmendem e_0 die Beträge von Δp größer, von Δa jedoch kleiner und damit erwartungsgemäß von Δe ebenfalls kleiner (vergrößert man p mehr als man a verringert, verringert man damit laut $e = \frac{a-p}{a+p}$ automatisch e und umgekehrt). Für eine parallele Einflugrichtung sind die Veränderungen bei Δp und Δa weitaus kleiner, der Betrag von Δe wird jedoch ebenfalls kleiner mit zunehmendem e_0 .

Die oben genannte Symmetrie ist in den Abb. 3.1 - 3.3 auf der jeweils linken Seite gut zu beobachten. Zum einen sieht man die 2π -Periodizität, man kann jedoch auch gut erkennen, dass die Graphen von um 180° gedrehten Einflugrichtungen qualitativ ineinander übergehen, wenn man sie um 180° nach links oder rechts verschiebt.

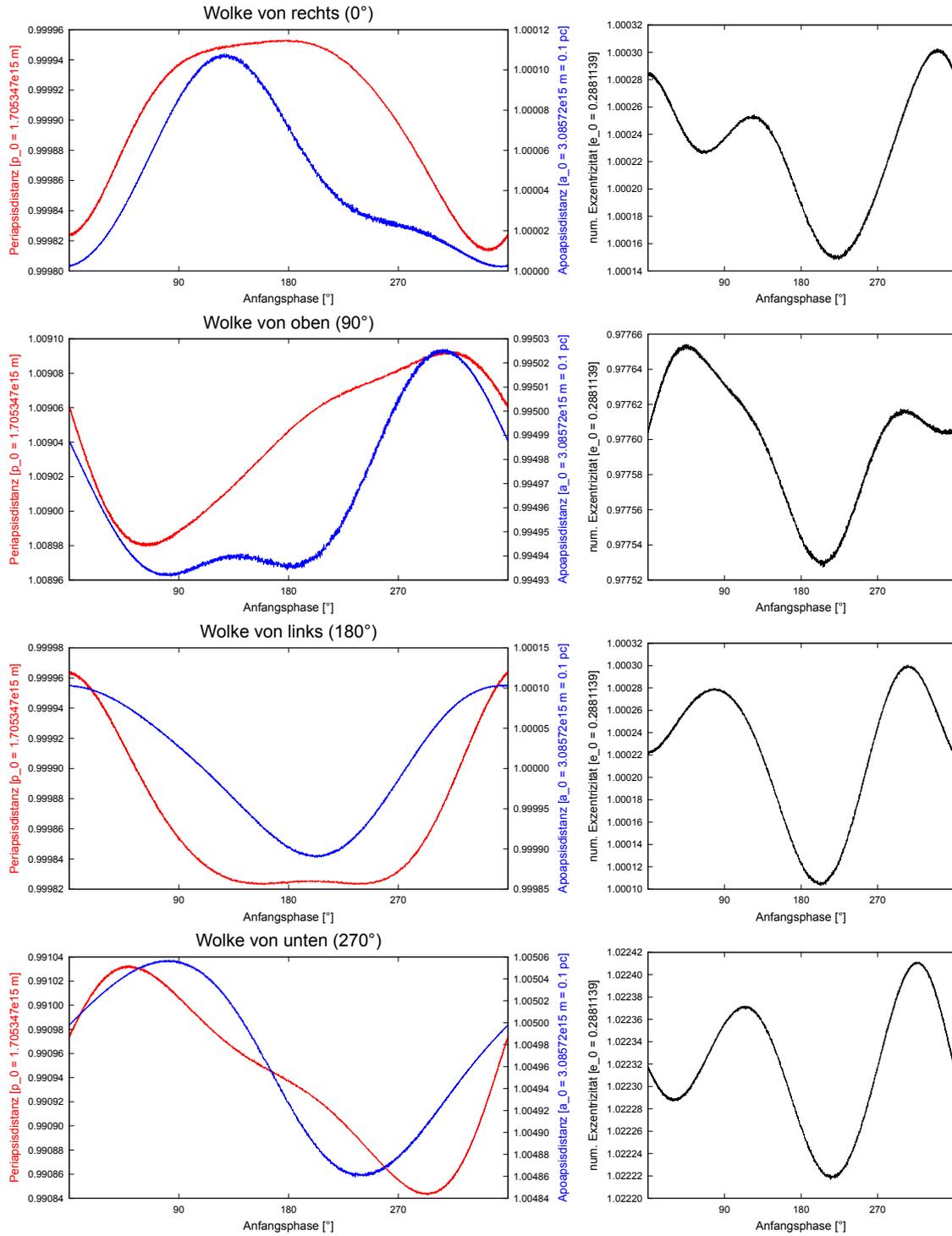


Abb. 4.1.: Periapsis- und Apoapsisdistanzen sowie numerische Exzentrizitäten der Sternbahn, welche ursprünglich eine Exzentrizität von $e = 0,2881139$ hatte. Der rote Graph in den linken Grafiken ist die Periapsisdistanz, der blaue die Apoapsisdistanz.

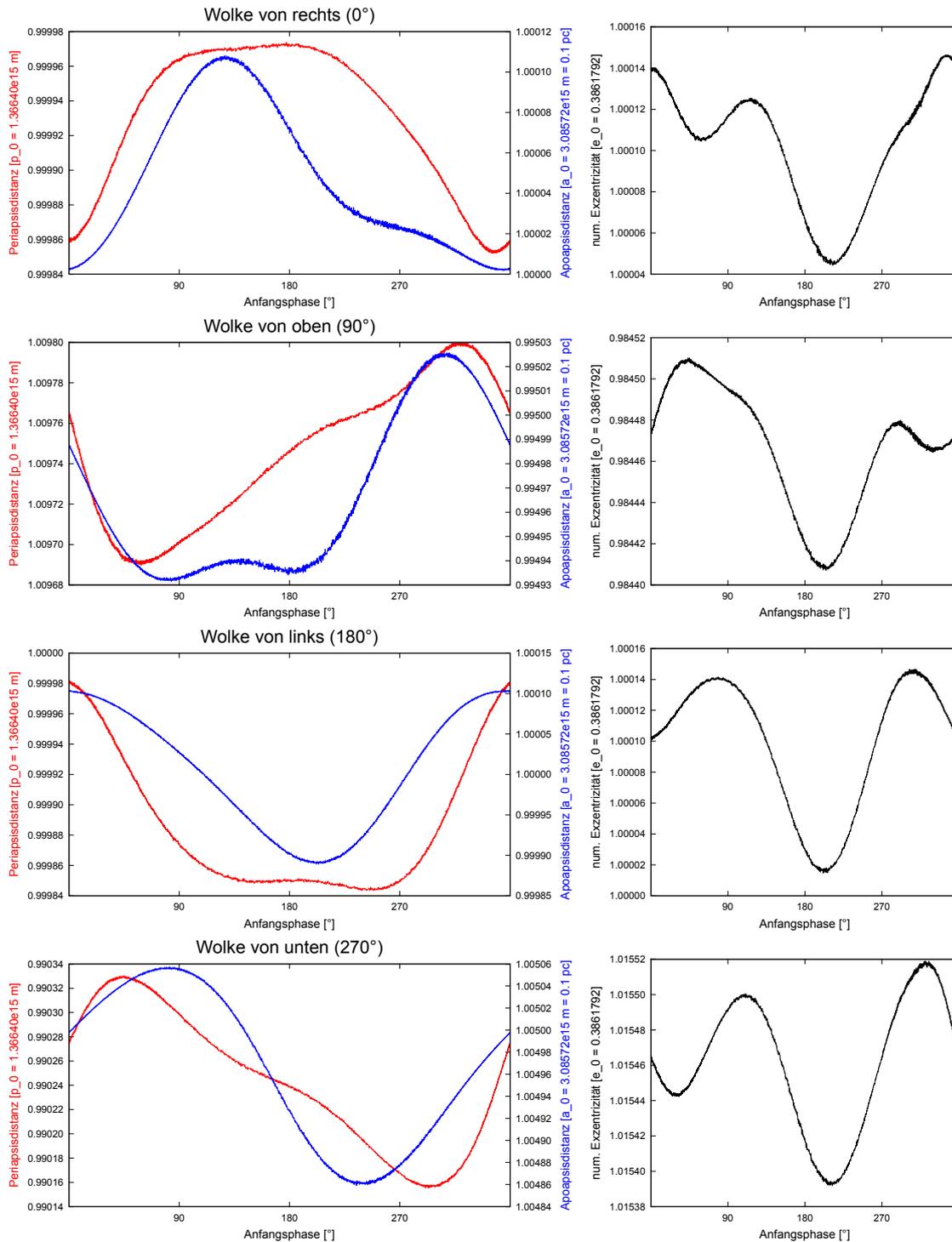


Abb. 4.2.: Periapsis- und Apoapsisdistanzen sowie numerische Exzentrizitäten der Sternbahn, welche ursprünglich eine Exzentrizität von $e = 0,3861792$ hatte. Der rote Graph in den linken Grafiken ist die Periapsisdistanz, der blaue die Apoapsisdistanz.

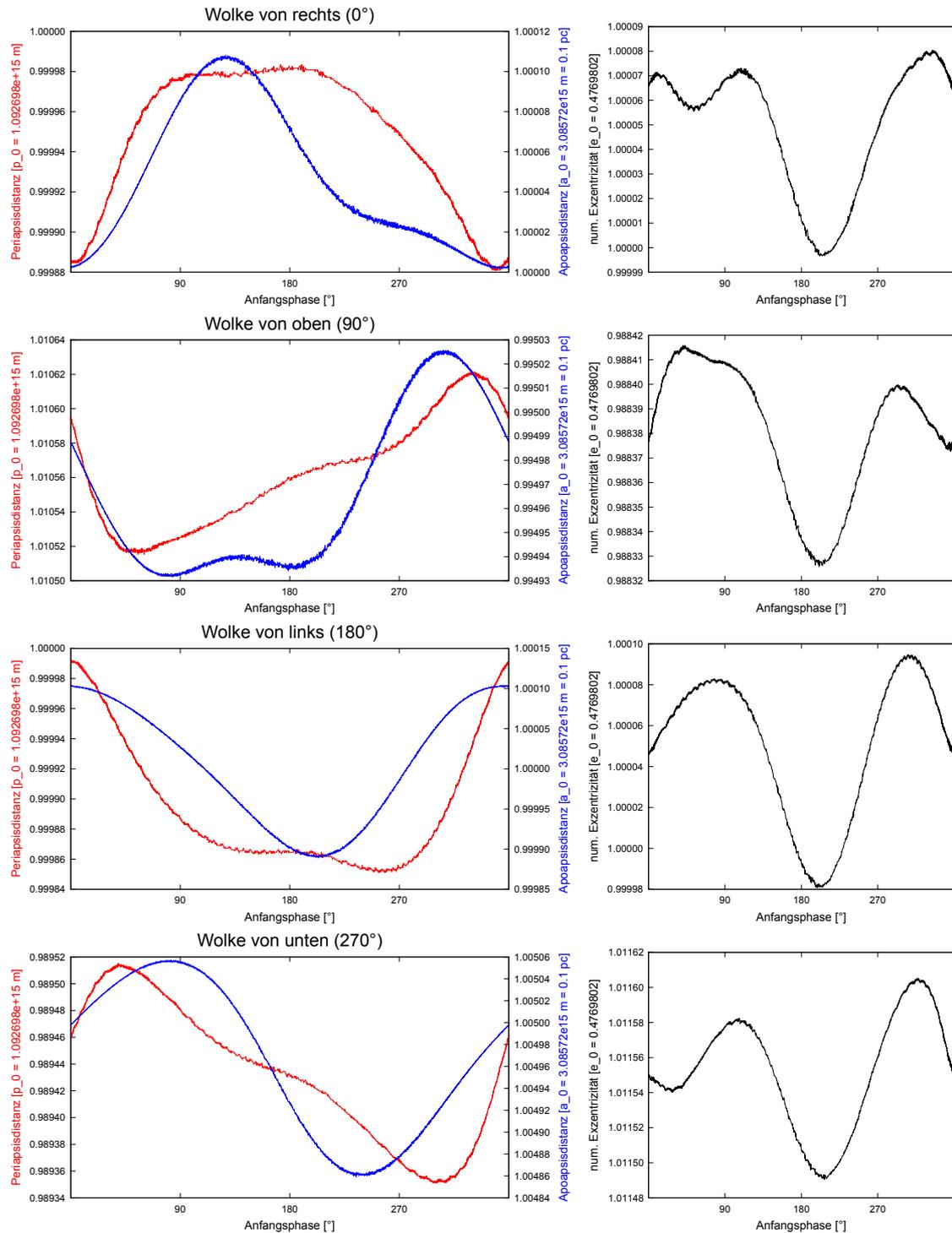


Abb. 4.3.: Periapsis- und Apoapsisdistanzen sowie numerische Exzentrizitäten der Sternbahn, welche ursprünglich eine Exzentrizität von $e = 0,4769802$ hatte. Der rote Graph in den linken Grafiken ist die Periapsisdistanz, der blaue die Apoapsisdistanz.

4.2. Das Vorbeiflugszenario

Da bei einer Anfangsexzentrizität die Änderungen von p und a im Einfangszenario für 270° -Einflugrichtung der Wolke am ehesten die Tendenz in Richtung S-Sternbahn hatten, wurde das Vorbeiflugszenario nur mit dieser Exzentrizität untersucht. Des weiteren wurde erwartet, dass die Veränderungen noch geringer ausfallen als im Einfangszenario, da die Zeit, in der eine erhöhte Masse sich innerhalb der Sternbahn befindet, weitaus kürzer ist als im Einfangszenario. Und in der Tat erkennt man mit dem ersten Blick auf Grafik 4.4, dass im Vorbeiflugszenario die Wolke noch weniger Einfluss auf die Sternbahn hat als im Einfangszenario. Die Abhängigkeiten der Veränderungen von der Einflugrichtung der Wolke sind nur noch schwach zu beobachten. Während bei einer Einflugrichtung von oben mit Offset nach links die Periapsisdistanz um $0,23\%$ kleiner wird, die Apoapsisdistanz um nur $0,07\%$ größer wird und die Exzentrizität um $0,24\%$ größer wird, sind es bei einer Einflugrichtung von links mit Offset nach unten nur noch Veränderungen von $-0,16\%$ für p , $+0,05\%$ für a und $+0,17\%$ für e . Das Phänomen der Spiegelasymmetrie ist noch zu beobachten, da für die um 180° gedrehten Einflugrichtungen die Veränderungen jeweils entgegengesetzt zu den oben genannten sind, mit fast identischen Beträgen. Vergleicht man die Grafiken 4.4 und 4.3, so sieht man, dass die Abhängigkeiten von p , a und e im Vorbeiflugszenario zu denen im Einfangszenario ähnlich sind, nur mit den oben genannten deutlich kleineren Amplituden.

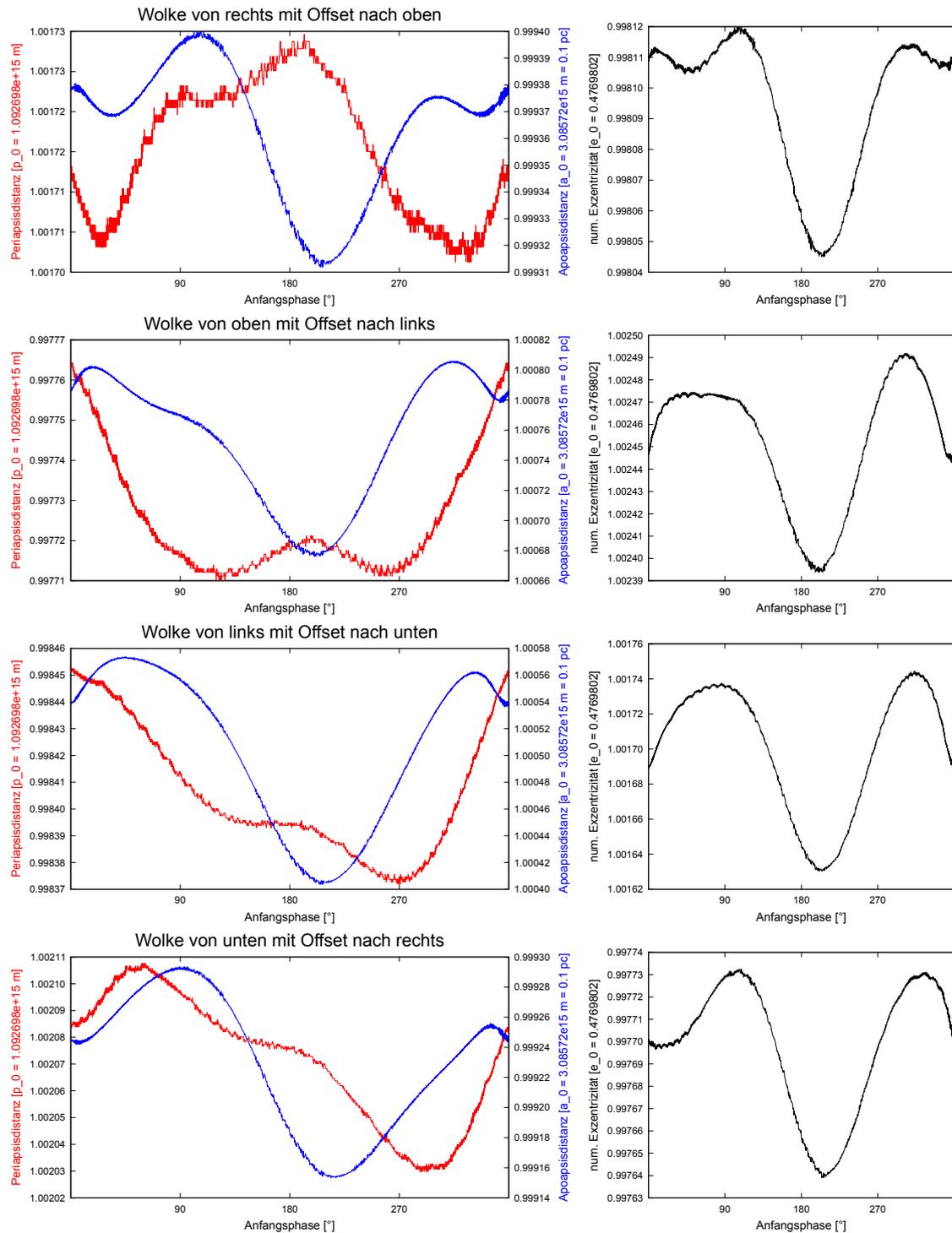


Abb. 4.4.: Periapsis- und Apoapsisdistanzen sowie numerische Exzentrizitäten der Sternbahn, welche ursprünglich eine Exzentrizität von $e = 0,4769802$ hatte, hier im Vorbeiflugzenario. Der rote Graph in den linken Grafiken ist die Periapsisdistanz, der blaue die Apoapsisdistanz.

Zusammenfassung

In dieser Arbeit wurde überprüft, ob ins Galaktische Zentrum einfliegende Gaswolken die Bahnen von Sternen außerhalb des S-Stern-Bereichs alleine so signifikant verändern können, dass diese dann zu den S-Sternen gezählt werden müssen. Dazu wurden selbst verfasste C-Programme verwendet, welche die Bewegungen und Bahn-daten von Modellkörpern (SMBH, Gaswolke, Stern) in numerischen Simulationen berechneten.

Dabei stellte sich heraus, dass die Beeinflussung durch das Potential einer Gaswolke allein nicht ausreicht, die S-Sterne auf diese Weise zu erklären. Sie war nur dazu in der Lage, minimale Bahnveränderungen des Sterns hervorzurufen. Da in der Realität jedoch mehrere Sterne in einem dichten Haufen um das schwarze Loch kreisen, könnten durch eine Gaswolke hervorgerufene Störungen zu engen Begegnungen von zwei oder mehr Sternen führen, in deren Folge signifikantere Bahnveränderungen stattfinden könnten.

Danksagung

Ich möchte gerne Prof. Andreas Burkert sowie Dr. Marc Schartmann vom CAST-Team der USM für ihre Vorschläge zur und Unterstützung während dieser Arbeit danken. Des weiteren möchte ich auch allen weiteren Teilnehmern der wöchentlichen Bachelor-Treffen für die Inspiration und den Austausch von Ideen danken. Besonderer Dank geht noch an Christian Alig, der neben seiner Publikation auch noch seine Zeit während des ein oder anderen Treffens zur Verfügung gestellt hat.

A. Programm 1 (3 Körper)

A.1. Code

Folgender Code ist für die Windows-Implementierung. Für die Verwendung in z.B. einer Unix-Umgebung sind daher die Dateipfade anzupassen (anstatt 'D://input.dat' etwa './input.dat').

```
1  #include "stdio.h"
2  #include "stdlib.h"
3  #include "math.h"
4  #include "string.h"
5
6  const double G=6.674e-11;
7  const double rho=6.17144e16;
8
9  int main()
10 {
11     FILE *data, *inputStern, *inputWolke, *energies;
12     int j,k,N=3;
13     double m[3],x[3],y[3],z[3],vx[3],vy[3],vz[3];
14     double i,R;
15     double Ekintot,Epottot;
16     double Ekin[3],Epot[3];
17     double t,H;
18
19     //oeffnen der Dateien
20     inputStern=fopen("D:\\inputStern3Koerper.dat","r");
21     inputWolke=fopen("D:\\inputWolke.dat","r");
22     data=fopen("D:\\3KoerperData.dat","w+");
23     energies=fopen("D:\\3KoerperEnergies.dat","w+");
24
25     //Einlesen der Anfangsbedingungen der Koerper aus input.txt
26     fscanf(inputWolke,"%lf %lf\n",&t,&H);
27     fscanf(inputWolke,"%lf %lf %lf %lf %lf %lf %lf\n",&m[1],&x[1],&y[1],&z[1],&vx[1],&vy[1],&vz[1]);
28     fscanf(inputStern,"%lf %lf %lf %lf %lf %lf %lf\n",&m[2],&x[2],&y[2],&z[2],&vx[2],&vy[2],&vz[2]);
29     m[0]=7.956e+36;
30     x[0]=y[0]=z[0]=0.000e+00;
31     vx[0]=vy[0]=vz[0]=0.000e+00;
32
33     t=t*24*60*60;
34     H=H*24*60*60;
35
36     //Iteration
37     for (i=0;i<=H;i+=t) {
```

```

38 //Ruecksetzen der totalen kinetischen und potentiellen Energie des Systems auf 0 vor jeder ←
    Iteration
39 Ekintot=0.0;
40 Epottot=0.0;
41 for (j=0;j<3;j++) {
42 //Ruecksetzen der kinetischen und potentiellen Energie jedes Koerpers auf 0
43 Ekin[j]=0.0;
44 Epot[j]=0.0;
45 //Kick 1: Berechnung des Abstandes zwischen Koerper j und k und
46 //Berechnung der Geschwindigkeiten des Koerpers j nach halbem Zeitschritt
47 for (k=0;k<3;k++) {
48     if (k==j) continue;
49     R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
50     if (k==1) {
51         if (R<rho) {
52             vx[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
53             vy[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
54             vz[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
55         else {
56             vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
57             vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
58             vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
59     }
60     else {
61         vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
62         vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
63         vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
64     }
65 //Drift: Berechnung der Positionen des Koerpers j aus obigen Geschwindigkeiten
66 //nach ganzem Zeitschritt
67 x[j] += vx[j]*t;
68 y[j] += vy[j]*t;
69 z[j] += vz[j]*t;
70 //Kick 2: Neuberechnung der Abstände zwischen Koerper j und k und
71 //Berechnung der Geschwindigkeiten des Koerpers j fuer den uebrigen halben Zeitschritt
72 //sowie Berechnung der potentiellen Energie des Koerpers j
73 for (k=0;k<3;k++) {
74     if (k==j) continue;
75     R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
76     if (k==1) {
77         if (R<rho) {
78             vx[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
79             vy[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
80             vz[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
81         else {
82             vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
83             vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
84             vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
85     }
86     else {
87         vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
88         vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
89         vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
90     Epot[j] -= (m[j]*m[k]*G)/R;
91     }
92 //Berechnung der kinetischen Energie des Koerpers j und Aufsummieren der Energien fuer das←
    gesamte System
93 Ekin[j]=(0.5*m[j]*(vx[j]*vx[j]+vy[j]*vy[j]+vz[j]*vz[j]));
94 Ekintot += Ekin[j];
95 Epottot += Epot[j];
96 }
97 //Ausgabe der Daten in Dateien
98 printf("Progress 3-Body-Simulation: %f %\r",i*100/H);

```

```

99     fprintf(data, "%+-15.6e ", i);
100    fprintf(energies, "%+-15.6e %+-15.6e %+-15.6e ", i, Ekintot, Epottot);
101    for (j=0; j<N; j++)
102    {
103        if (j==N-1)
104        {
105            fprintf(data, "%+-15.6e %+-15.6e %+-15.6e\n", x[j], y[j], z[j]);
106            fprintf(energies, "%+-15.6e %+-15.6e\n", Ekin[j], Epot[j]);
107        }
108        else
109        {
110            fprintf(data, "%+-15.6e %+-15.6e %+-15.6e ", x[j], y[j], z[j]);
111            fprintf(energies, "%+-15.6e %+-15.6e ", Ekin[j], Epot[j]);
112        }
113    }
114 }
115 fclose(energies);
116 fclose(data);
117 fclose(inputStern);
118 fclose(inputWolke);
119 return EXIT_SUCCESS;
120 }

```

A.2. Format der input-/output-Dateien

- inputStern3Koerper.dat:
Zeile 1: Masse m , Ort x, y, z und Geschwindigkeit v_x, v_y, v_z des Sterns zum Simulationsanfang
- inputWolke.dat:
Zeile 1: Schrittweite t und Simulationsdauer H
Zeile 2: Masse m , Ort x, y, z und Geschwindigkeit v_x, v_y, v_z der Wolke zum Simulationsanfang
- 3KoerperData.dat:
in jeder Zeile: Zeit i , Ort des schwarzen Lochs $x^{(SMBH)}, y^{(SMBH)}, z^{(SMBH)}$, Ort der Wolke $x^{(Wolke)}, y^{(Wolke)}, z^{(Wolke)}$ und Ort des Sterns $x^{(Stern)}, y^{(Stern)}, z^{(Stern)}$ zu jedem Zeitschritt
- 3KoerperEnergies.dat:
in jeder Zeile: Zeit i , kin. und pot. Energie des ganzen Systems $E_{kin}^{(total)}, E_{pot}^{(total)}$, des schwarzen Lochs $E_{kin}^{(SMBH)}, E_{pot}^{(SMBH)}$, der Wolke $E_{kin}^{(Wolke)}, E_{pot}^{(Wolke)}$ und des Sterns $E_{kin}^{(Stern)}, E_{pot}^{(Stern)}$ zu jedem Zeitschritt

B. Programm 2 (Phasenscan)

B.1. Code

Folgender Code ist für die Windows-Implementierung. Für die Verwendung in z.B. einer Unix-Umgebung sind daher die Dateipfade anzupassen (anstatt 'D://input.dat' etwa './input.dat').

```
1  #define M_PI 3.1415926535897932384626433
2  #include "stdio.h"
3  #include "stdlib.h"
4  #include "math.h"
5  #include "string.h"
6
7  const double G=6.674e-11;
8  const double rho=6.17144e16;
9
10 int main ()
11 {
12     FILE *input , *star_positions , *star_positions_after , *data;
13     int o,c=0,d,j=0,k=0,N=3;
14     double m [3] , x [3] , y [3] , z [3] , vx [3] , vy [3] , vz [3];
15     double i,R,phase , phaseOld , phaseRun , p, apocenter , pericenter ;
16     double Ekintot , Epottot ;
17     double Ekin [3] , Epot [3];
18     double t,H,a,b,e;
19
20     //Oeffnen der Dateien
21     input=fopen("D:\\inputStern.dat","r+"); //t und Anfangsbedingungen fuer Stern
22     star_positions=fopen("D:\\starPositions.dat","w+");
23     data=fopen("D:\\data.dat","w+");
24
25     //Einlesen der Anfangsbedingungen der Koerper aus input.txt
26     fscanf (input , "%lf\n",&t);
27     t=t*24*60*60;
28     fscanf (input , "%lf %lf %lf %lf %lf %lf %lf\n",&m [1] ,&x [1] ,&y [1] ,&z [1] ,&vx [1] ,&vy [1] ,&vz [1] );
29     //Das SMBH befindet sich am Ursprung !
30     m [0]=7.956e+36;
31     x [0]=y [0]=z [0]=0.000e+00;
32     vx [0]=vy [0]=vz [0]=0.000e+00;
33
34     //Teil 1: 2-Koerper-Problem , um die Bahn des Sterns bei einem Umlauf zu simulieren
35     apocenter = pericenter = sqrt ((x [1]-x [0])*(x [1]-x [0])+(y [1]-y [0])*(y [1]-y [0])+(z [1]-z [0])*(z [1]-z [0]));
36     R = sqrt ((x [1]-x [0])*(x [1]-x [0])+(y [1]-y [0])*(y [1]-y [0])+(z [1]-z [0])*(z [1]-z [0]));
37     phaseRun = 0.0;
38     phaseOld = 0.0;
```

```

39 while ((phaseRun < 2*M_PI) && (R < 3.08572e16)) {
40   for (j=0;j<2;j++) {
41     for (k=0;k<2;k++) {
42       if (k==j) continue;
43       R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
44       vx[j] -= G*m[k]*((x[j]-x[k])/(R**R))*(0.5*t);
45       vy[j] -= G*m[k]*((y[j]-y[k])/(R**R))*(0.5*t);
46       vz[j] -= G*m[k]*((z[j]-z[k])/(R**R))*(0.5*t); }
47     x[j] += vx[j]*t;
48     y[j] += vy[j]*t;
49     z[j] += vz[j]*t;
50     for (k=0;k<2;k++) {
51       if (k==j) continue;
52       R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
53       vx[j] -= G*m[k]*((x[j]-x[k])/(R**R))*(0.5*t);
54       vy[j] -= G*m[k]*((y[j]-y[k])/(R**R))*(0.5*t);
55       vz[j] -= G*m[k]*((z[j]-z[k])/(R**R))*(0.5*t);
56     }
57     //Berechnung der Apozentrums- bzw. Perizentrumsdistanz
58     if (R > apocenter) apocenter = R;
59     if (R < pericenter) pericenter = R;
60   }
61   c++;
62   //Bestimmung der Phase des Sterns sowie des Laufparameters "phaseRun"
63   if (x[1] > 0 && y[1] > 0) phase = atan(y[1]/x[1]);
64   else if (x[1] < 0 && y[1] > 0) phase = 0.5*M_PI+atan(y[1]/x[1]) + 0.5*M_PI;
65   else if (x[1] < 0 && y[1] < 0) phase = atan(y[1]/x[1]) + M_PI;
66   else if (x[1] > 0 && y[1] < 0) phase = 0.5*M_PI+atan(y[1]/x[1]) + 1.5*M_PI;
67   if (phaseOld < phase) phaseRun += (phase - phaseOld);
68   else phaseRun += (phase - phaseOld + 2*M_PI);
69   phaseOld = phase;
70   //Variable, um die Phasen der Bahnpositionen zu speichern
71   p = phaseRun;
72   //Ausgabe der Daten in die Datei ".\starPositions.dat"
73   fprintf(star_positions, "%+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e\n",
74           p, m[1], x[1], y[1], z[1], vx[1], vy[1], vz[1]);
75 }
76 //Berechnung der Bahndaten
77 e = (apocenter - pericenter) / (apocenter + pericenter);
78 a = (apocenter + pericenter) / 2;
79 b = sqrt((a*a) * (1 - e*e));
80 fprintf(data, "# Star's orbit before: \n# apocenter-distance: %e ; pericenter distance: %e ; ←
81     half major axis: %e ; half minor axis: %e ; excentricity: %e\n", apocenter, pericenter, a, b, ←
82     , e);
83 fclose(star_positions);
84 fclose(input);
85 //Teil 2: 3-Koerper-Problem mit den Bahnpositionen des Sterns aus der 2-Koerper-Simulation als ←
86 //Anfangsbedingungen
87 //fuer jeweils eine Simulation
88 star_positions=fopen("D:\\starPositions.dat", "r+"); //Anfangsbedingungen fuer Stern
89 star_positions_after=fopen("D:\\starPositionsAfter.dat", "w+");
90
91 for (o=0;o<c;o++) {
92   input=fopen("D:\\inputWolke.dat", "r+"); //neues t, H sowie Anfangsbedingungen fuer Wolke
93   fscanf(input, "%lf %lf %lf %lf %lf %lf %lf %lf\n", &t, &H, &m[1], &x[1], &y[1], &z[1], &vx[1], &vy[1], &vz[1]);
94   t=t*24*60*60;
95   H=H*24*60*60;
96   fclose(input);
97   //Einlesen der im ersten Teil berechneten Bahnpositionen des Sterns
98   fscanf(star_positions, "%lf %lf %lf %lf %lf %lf %lf %lf\n", &p, &m[2], &x[2], &y[2], &z[2], &vx[2], &vy[2], &vz[2]);

```

```

96   for (i=0;i<=H;i+=t) {
97       m[0]=7.956e+36;
98       x[0]=y[0]=z[0]=0.000e+00;
99       vx[0]=vy[0]=vz[0]=0.000e+00;
100      //Ruecksetzen der totalen kinetischen und potentiellen Energie des Systems auf 0 vor ←
           jeder Iteration
101      Ekintot=0.0;
102      Epottot=0.0;
103      for (j=0;j<3;j++) {
104          //Ruecksetzen der kinetischen und potentiellen Energie jedes Koerpers auf 0
105          Ekin[j]=0.0;
106          Epot[j]=0.0;
107          //Kick 1: Berechnung des Abstandes zwischen Koerper j und k und
108          //Berechnung der Geschwindigkeiten des Koerpers j nach halbem Zeitschritt
109          for (k=0;k<3;k++) {
110              if (k==j) continue;
111              R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
112              if (k==1) {
113                  if (R<rho) {
114                      vx[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
115                      vy[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
116                      vz[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
117                  else {
118                      vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
119                      vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
120                      vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); } }
121              else {
122                  vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
123                  vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
124                  vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); } }
125          }
126          //Drift: Berechnung der Positionen des Koerpers j aus obigen Geschwindigkeiten
127          //nach ganzem Zeitschritt
128          x[j] += vx[j]*t;
129          y[j] += vy[j]*t;
130          z[j] += vz[j]*t;
131          //Kick 2: Neuberechnung der Abstaende zwischen Koerper j und k und
132          //Berechnung der Geschwindigkeiten des Koerpers j fuer den uebrigen halben Zeitschritt
133          //sowie Berechnung der potentiellen Energie des Koerpers j
134          for (k=0;k<3;k++) {
135              if (k==j) continue;
136              R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
137              if (k==1) {
138                  if (R<rho) {
139                      vx[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
140                      vy[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
141                      vz[j] -= G*((R*R*R)/(rho*rho*rho))*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); }
142                  else {
143                      vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
144                      vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
145                      vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); } }
146              else {
147                  vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*(0.5*t);
148                  vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*(0.5*t);
149                  vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*(0.5*t); } }
150              Epot[j] -= (m[j]*m[k]*G)/R;
151          }
152          //Berechnung der kinetischen Energie des Koerpers j und Aufsummieren der Energien fuer ←
           das gesamte System
153          Ekin[j]=(0.5*m[j]*(vx[j]*vx[j]+vy[j]*vy[j]+vz[j]*vz[j]));
154          Ekintot += Ekin[j];
155          Epottot += Epot[j];
156      }

```

```

157     printf("Progress 3-Body-Simulation  #d:  %f  %%\r",o+1,i*100/H);
158 }
159 //Ausgabe der Daten in Dateien
160 fprintf(star_positions_after,"%+15.6e %+15.6e %+15.6e %+15.6e %+15.6e %+15.6e %+15.6e
e %+15.6e %+15.6e %+15.6e\n",p,m[2],x[2],y[2],z[2],vx[2],vy[2],vz[2],Ekin[2],Epot
[2]);
161 }
162 fclose(star_positions);
163 fclose(star_positions_after);
164
165 //Teil 3: Unterprogramm zur Berechnung der Bahndaten des Sterns
166 input=fopen("D:\\starPositionsAfter.dat","r");
167
168 fprintf(data,"# Stars orbit after: Phase      apocenter-distance      pericenter-distance      a ←
b      e\n");
169 for (o=0;o<c;o++) {
170     fscanff(input,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf\n",&p,&m[1],&x[1],&y[1],&z[1],&vx←
[1],&vy[1],&vz[1],&Ekin[1],&Epot[1]);
171     m[0]=7.956e+36;
172     x[0]=y[0]=z[0]=0.000e+00;
173     vx[0]=vy[0]=vz[0]=0.000e+00;
174
175     apocenter = pericenter = sqrt((x[1]-x[0])*(x[1]-x[0])+(y[1]-y[0])*(y[1]-y[0])+(z[1]-z[0])*(←
z[1]-z[0]));
176     R = sqrt((x[1]-x[0])*(x[1]-x[0])+(y[1]-y[0])*(y[1]-y[0])+(z[1]-z[0])*(z[1]-z[0]));
177     phaseRun = 0.0;
178     phaseOld = 0.0;
179     d=0;
180     while ((phaseRun < 2*M_PI) && (R < 3.08572e16)) {
181         for (j=0;j<2;j++) {
182             for (k=0;k<2;k++) {
183                 if (k==j) continue;
184                 R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
185                 vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*0.5*t;
186                 vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*0.5*t;
187                 vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*0.5*t;
188             }
189             x[j] += vx[j]*t;
190             y[j] += vy[j]*t;
191             z[j] += vz[j]*t;
192             for (k=0;k<2;k++) {
193                 if (k==j) continue;
194                 R = sqrt((x[j]-x[k])*(x[j]-x[k])+(y[j]-y[k])*(y[j]-y[k])+(z[j]-z[k])*(z[j]-z[k]));
195                 vx[j] -= G*m[k]*((x[j]-x[k])/(R*R*R))*0.5*t;
196                 vy[j] -= G*m[k]*((y[j]-y[k])/(R*R*R))*0.5*t;
197                 vz[j] -= G*m[k]*((z[j]-z[k])/(R*R*R))*0.5*t;
198             }
199         }
200         d++;
201         if (R > apocenter) apocenter = R;
202         if (R < pericenter) pericenter = R;
203         if (x[1] > 0 && y[1] > 0) phase = atan(y[1]/x[1]);
204         else if (x[1] < 0 && y[1] > 0) phase = 0.5*M_PI+atan(y[1]/x[1]) + 0.5*M_PI;
205         else if (x[1] < 0 && y[1] < 0) phase = atan(y[1]/x[1]) + M_PI;
206         else if (x[1] > 0 && y[1] < 0) phase = 0.5*M_PI+atan(y[1]/x[1]) + 1.5*M_PI;
207         if (d==1) phaseOld = phase;
208         if (phaseOld <= phase) phaseRun += (phase - phaseOld);
209         else phaseRun += (phase - phaseOld + 2*M_PI);
210         phaseOld = phase;
211     }
212     e = (apocenter - pericenter) / (apocenter + pericenter);
213     a = (apocenter + pericenter) / 2;
214     b = sqrt((a*a) * (1 - e*e));

```

```

215     printf("2-Body Simulation %d runs\r",o+1);
216     fprintf(data,"%+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e %+-15.6e\n",p,apocenter,←
        pericenter,a,b,e);
217 }
218 return 0;
219 }

```

B.2. Format der input-/output-Dateien

- inputStern.dat:
 - Zeile 1: Zeitschritt für Teil 1 und 3 des Programms t
 - Zeile 2: Masse m , Ort x, y, z und Geschwindigkeit v_x, v_y, v_z des Sterns zum Simulationsanfang
- inputWolke.dat:
 - Zeile 1: Schrittweite t und Simulationsdauer H für Teil 2 des Programms
 - Zeile 2: Masse m , Ort x, y, z und Geschwindigkeit v_x, v_y, v_z der Wolke zum Simulationsanfang
- starPositions.dat:
 - in jeder Zeile: *Phase*, Masse des Sterns m^{Stern} , Ort des Sterns $x^{Stern}, y^{Stern}, z^{Stern}$ und Geschwindigkeit des Sterns $v_x^{Stern}, v_y^{Stern}, v_z^{Stern}$ auf jedem berechneten Punkt seiner Bahn vor dem Vorbeiflug der Wolke
- starPositionsFinal.dat:
 - in jeder Zeile: *Phase*, Masse des Sterns m^{Stern} , Ort des Sterns $x^{Stern}, y^{Stern}, z^{Stern}$, Geschwindigkeit des Sterns $v_x^{Stern}, v_y^{Stern}, v_z^{Stern}$ und kin. und pot. Energie des Sterns $E_{kin}^{Stern}, E_{pot}^{Stern}$ nach dem Vorbeiflug der Wolke
- data.dat:
 - Zeile 1+2: ursprüngliche Bahndaten (Apo- bzw. Perizentrumsdistanz, a, b, e)
 - dann in jeder Zeile: *Phase Apoapsisdistanz Periapsisdistanz a b e*

Literaturverzeichnis

- Alexander, T. (2005). Stellar Processes Near the Massive Black Hole in the Galactic Center. *Physics Reports*, 435:65–142.
- Alig, C., Burkert, A., Johansson, P., and Schartmann, M. (2011). Simulations of direct collisions of gas clouds with the central black hole. *Monthly Notices of the Royal Astronomical Society*, 412:469–486.
- Bartko, H., Martins, F., Fritz, T. K., Genzel, R., Levin, Y., Perets, H. B., Paumard, T., Nayakshin, S., Gerhard, O., Alexander, T., Dodds-Eden, K., Eisenhauer, F., Gillessen, S., Mascetti, L., Ott, T., Perrin, G., Pfuhl, O., Reid, M. J., Rouan, D., Sternberg, A., and Trippe, S. (2009). Evidence for Warped Discs of Young Stars in the Galactic Center. *The Astrophysical Journal*, 697:1741–1763.
- Eckart, A., Genzel, R., Ott, T., and Schödel, R. (2002). Stellar Orbits Near Sagittarius A*. *Monthly Notices of the Royal Astronomical Society*, 331:917–934.
- Eisenhauer, F., Schödel, R., Genzel, R., Ott, T., Tecza, M., Abuter, R., Eckart, A., and Alexander, T. (2003). A Geometric Determination of the Distance to the Galactic Center. *The Astrophysical Journal*, 597:L121–L124.
- Ghez, A., Salim, S., Hornstein, S., Tanner, A., Lu, J., Morris, M., Becklin, E., and Duchêne, G. (2005). Stellar Orbits around the Galactic Center Black Hole. *The Astrophysical Journal*, 620:744–757.
- Gillessen, S., Eisenhauer, F., Fritz, T., Bartko, H., Dodds-Eden, K., Pfuhl, O., Ott, T., and Genzel, R. (2009). The Orbit of the Star S2 around Sgr A* from Very Large Telescope and Keck Data. *The Astrophysical Journal*, 707:L114–L117.
- Levin, Y. (2007). Starbursts near supermassive black holes: young stars in the Galactic Centre, and gravitational waves in LISA band. *Monthly Notices of the Royal Astronomical Society*, 374:515–524.

- Lu, J., Ghez, A., Morris, M., Hornstein, S., and Matthews, K. (2008). Orbits and origins of the young stars in the central parsec of the Galaxy. *Journal of Physics: Conference Series*, 131:012012.
- Merritt, D., Gualandris, A., and Mikkola, S. (2009). Explaining the Orbits of the Galactic Center S-Stars. *The Astrophysical Journal*, 693:L25–L38.
- Paumard, T., Genzel, R., Martins, F., Nayakshin, S., Beloborodov, A., Levin, Y., Trippe, S., Eisenhauer, F., Ott, T., Gillessen, S., Abuter, R., Cuadra, J., Alexander, T., and Sternberg, A. (2006). The two young star disks in the central parsec of the Galaxy: properties, dynamics, and formation. *Journal of Physics: Conference Series*, 54:199.
- Schödel, R., Ott, T., Genzel, R., Hofmann, R., Lehnert, M., Eckart, A., Mouawad, N., Alexander, T., Reid, M., Lenzen, R., Hartung, M., Lacombe, F., Rouan, D., Gendron, E., Rousset, G., Lagrange, A.-M., Brandner, W., Ageorges, N., Lidman, C., Moorwood, A., Spyromilio, J., Hubin, N., and Menten, K. (2002). Closest Star Seen Orbiting the Supermassive Black Hole at the Centre of the Milky Way. *Nature*, 419:694–696.
- Trippe, S., Gillessen, S., Ott, T., Eisenhauer, F., Paumard, T., Martins, F., Genzel, R., Schödel, R., Eckart, A., and Alexander, T. (2006). Stellar orbits around Sgr A*. *Journal of Physics: Conference Series*, 54:288.

Abbildungsverzeichnis

1.1. K-Band-Bild der S-Sterne	2
2.1. Verwendete Sternbahnen	6
2.2. Größenvergleich Stern-Wolke	7
2.3. Einflugrichtungen der Wolke	7
4.1. Diagramme für die 0,2881139 - Bahn im Einfangszenario	16
4.2. Diagramme für die 0,3861792 - Bahn im Einfangszenario	17
4.3. Diagramme für die 0,4769802 - Bahn im Einfangszenario	18
4.4. Diagramme für die 0,4769802 - Bahn im Vorbeiflugszenario	20

Selbstständigkeitserklärung

Hiermit versichere ich,

dass ich diese Bachelorarbeit zum Thema: „Zur Entstehung der S-Sterne um das supermassive schwarze Loch im Zentrum der Milchstraße“ selbstständig verfasst habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht.

Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

München, 20. September 2011

Ort, Datum

Unterschrift