

Übung 04: Das Torkeln des Marsmondes Phobos

Tobias Blesgen und Leonardo Thome

23.06.2021

Im Folgenden wollen wir das Torkeln des Marsmondes Phobos untersuchen, welches durch die Ellipsoideform des Mondes und der elliptischen Umlaufbahn um den Mars entsteht. Dazu beschreiben wir die Situation wie folgt:

Wir nehmen an, dass der Mond Phobos den Mars auf einer festen elliptischen Bahn mit Radius $r(t)$, Polarkwinkel $\phi(t)$, großen Halbachse a und Exzentrizität ϵ in der Umlaufzeit T umkreist. Die Eigenbewegung von Phobos wird durch den Winkel $\theta(t)$ beschrieben. Mit den drei Trägheitsmomenten $I_1 < I_2 < I_3$ von Phobos, kann die Bewegungsgleichung für die Eigenbewegung wie folgt geschrieben werden:

$$I_3 \ddot{\theta}(t) = -\frac{3}{2} \left(\frac{2\pi}{T} \right)^2 (I_2 - I_1) \left(\frac{a}{r(t)} \right)^3 \sin 2(\theta(t) - \phi(t)) \quad (1)$$

Mit der Größe $\alpha = \sqrt{3 \frac{I_2 - I_1}{I_3}}$ vereinfachen wir die Gleichung zu:

$$\ddot{\theta}(t) = -\frac{\alpha^2}{2} \left(\frac{2\pi}{T} \right)^2 \left(\frac{a}{r(t)} \right)^3 \sin 2(\theta(t) - \phi(t)) \quad (2)$$

Dabei lässt sich die Differentialgleichung 2.Ordnung auch als 2 Differentialgleichungen 1.Ordnung schreiben:

$$\frac{d\theta(t)}{dt} = \dot{\theta} \quad (3)$$

$$\frac{d\dot{\theta}(t)}{dt} = -\frac{\alpha^2}{2} \left(\frac{2\pi}{T} \right)^2 \left(\frac{a}{r(t)} \right)^3 \sin 2(\theta(t) - \phi(t)) \quad (4)$$

Da $r(t)$ und $\phi(t)$ selbst zeitabhängig sind, müssen wir erst diese lösen, um die Lösung zu $\theta(t)$ finden zu können.

Nach den Keplerschen Gesetzen erhalten wir die Beziehung (QUELLE):

$$r(\phi) = \frac{p}{1 + \epsilon \cos \phi} = \frac{a(1 - \epsilon^2)}{1 + \epsilon \cos \phi} \quad (5)$$

und für die Ableitung nach dem Winkel:

$$\frac{\partial r(\phi)}{\partial \phi} = \frac{a(1 - \epsilon^2)\epsilon \sin \phi}{(1 + \epsilon \cos \phi)^2} \quad (6)$$

Aus der Drehimpulserhaltung können wir folgern:

$$L = Mr^2 \dot{\phi} = \text{const} \quad (7)$$

$$\Leftrightarrow \dot{\phi} = \frac{L}{M} \frac{1}{r^2} \quad (8)$$

Für die zeitliche Ableitung von r ergibt sich mit Hilfe der Gleichungen (6) & (8):

$$\dot{r} = \frac{\partial r}{\partial \phi} \frac{\partial \phi}{\partial t} = \frac{a(1 - \epsilon^2)\epsilon \sin \phi}{(1 + \epsilon \cos \phi)^2} \frac{L}{M} \frac{1}{r^2} \quad (9)$$

Mit den 4 Differentialgleichung können wir nun $\theta(t)$ und $\dot{\theta}(t)$ bestimmen (r ist dabei in Einheiten von a).

$$\frac{d\phi}{dt} = \frac{L}{M} \frac{1}{r^2} \quad (10)$$

$$\frac{dr}{dt} = \frac{a(1 - \epsilon^2)\epsilon \sin \phi}{(1 + \epsilon \cos \phi)^2} \frac{L}{M} \frac{1}{r^2} \quad (11)$$

$$\frac{d\theta(t)}{dt} = \dot{\theta} \quad (12)$$

$$\frac{d\dot{\theta}(t)}{dt} = -\frac{\alpha^2}{2} \left(\frac{2\pi}{T}\right)^2 \left(\frac{1}{r(t)}\right)^3 \sin 2(\theta(t) - \phi(t)) \quad (13)$$

Für die Implementierung werden wir \tilde{r} in Einheiten von a und \tilde{t} in Einheiten von $T/2\pi$. Es ergibt sich mit dem der Abschätzung des mittleren Radius von a und einer annähernd runden Kreisbahn für das Drehimpuls-Masse-Verhältnis die Näherung:

$$\frac{L}{M} \approx \frac{Ma^2\omega}{M} = \frac{2\pi a^2}{T} \quad (14)$$

In den neu eingeführten Einheiten schreibt sich der Drehimpuls also als 1.

Runge-Kutta 2 Verfahren

Um das Differentialgleichungssysteme auszuwerten, verwenden wir das Runge-Kutta Verfahren nach

$$x_{i+1} = x_i + \frac{h}{2} [f(t_i, x_i) + f(t_i + h, x_i + hf(t_i, x_i))]. \quad (15)$$

Wobei sich unser f aus den vier Anteilen von ϕ , r , θ und $\dot{\theta}$ zusammensetzt.

Dabei bietet uns das Runge-Kutter Verfahren numerische Stabilität (soweit das System selbst Stabile ist) und weist mit einem Verfahrensfehler von $\mathcal{O}(h^2)$ einen kleineren Fehler als das Eulerverfahren auf für kleinen Schrittweiten h .

Implementation des DGS nach dem Runge-Kutta 2 Verfahren

```
#include <Rcpp.h>
#include <stdlib.h>
#include <vector>
#include <algorithm>

using namespace Rcpp;

// Wir verwenden Strukturen, um Funktionsargumente uebersichtlicher zu halten
typedef struct
{
```

```

    double phi, r, theta, thetadot;
} Status;

typedef struct
{
    double epsilon, LM, ktheta;
} Parameter;

// Template zum Zerschneiden der verwendeten Vektoren
template<typename T>
std::vector<T> slice(std::vector<T> const &v, int m, int n)
{
    auto erste = v.cbegin() + m;
    auto letzte = v.cbegin() + n + 1;

    std::vector<T> vektor(erste, letzte);
    return vektor;
}

// Berechnungsschritt der Ableitungen nach dem DGS
void f(Status alterStatus, Parameter parameter, Status& neuerStatus){

    neuerStatus.r = parameter.LM*(1-parameter.epsilon*parameter.epsilon)
        *parameter.epsilon*sin(alterStatus.phi)/(alterStatus.r*alterStatus.r
        *(1+parameter.epsilon*cos(alterStatus.phi))*(1+parameter.epsilon
        *cos(alterStatus.phi)));

    neuerStatus.phi = parameter.LM/(alterStatus.r*alterStatus.r);

    neuerStatus.theta = alterStatus.thetadot;

    neuerStatus.thetadot = parameter.ktheta*sin(2*(alterStatus.theta
        -alterStatus.phi))/(alterStatus.r*alterStatus.r*alterStatus.r);
}

// Ein Intergrationsschritt nach Runge-Kutta
void rkSchritt(Status& status, Parameter parameter, double h){
    Status fStatus;                // Standard Ableitung
    f(status, parameter, fStatus);

    Status f2Status;                // Mischterm Ableitung
    Status gemischt = {.phi=status.phi+h*fStatus.phi, .r = status.r+h*fStatus.r,
        .theta = status.theta + h*fStatus.theta,
        .thetadot = status.thetadot + h*fStatus.thetadot};

    f(gemischt, parameter, f2Status);
    status.phi = status.phi + h/2*(fStatus.phi + f2Status.phi);
    status.r = status.r+ h/2*(fStatus.r + f2Status.r);
    status.theta = status.theta + h/2*(fStatus.theta + f2Status.theta);
    status.thetadot = status.thetadot+h/2*(fStatus.thetadot+f2Status.thetadot);
}

```

```

///[Rcpp::export]
Rcpp::List durchlauf(const int maxSchritte, const double h,
                    const double phi, const double r, const double theta,
                    const double thetadot, const double epsilon,
                    const double LM, const double ktheta){
    // Arrays der Werte zur späteren Ausgabe
    std::vector<double> phiWerte(maxSchritte);
    std::vector<double> rWerte(maxSchritte);
    std::vector<double> thetaWerte(maxSchritte);
    std::vector<double> thetadotWerte(maxSchritte);
    int k = 0;
    // Quelltext
    Status status = {.phi = phi, .r = r, .theta = theta, .thetadot = thetadot};
    Parameter parameter = {.epsilon = epsilon, .LM = LM, .ktheta = ktheta};
    // Schleife bis zur Abbruchsbedingung
    for (int i = 0; i < maxSchritte; i++){
        phiWerte[i] = status.phi;
        rWerte[i] = status.r;
        rkSchritt(status, parameter, h);

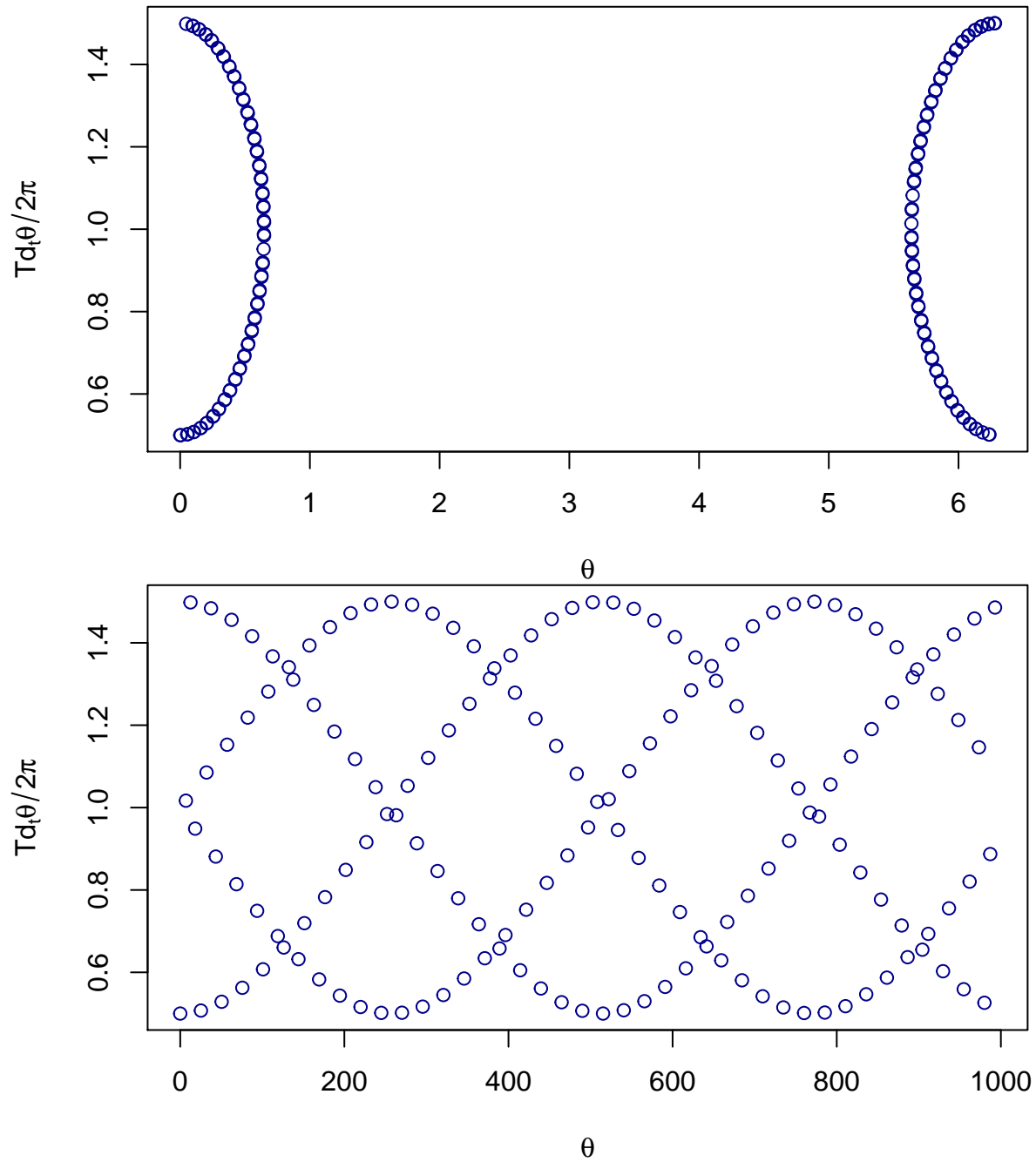
        // theta und thetadot Aufnahme nach jeder Umrundung
        if(status.phi >= 2*3.14159*k){
            k++;
            thetaWerte[k] = status.theta;
            thetadotWerte[k] = status.thetadot;
        }
    }

    // Rückgabe für eine grafische Wiedergabe
    return List::create(Named("phi") = phiWerte, Named("r") = rWerte,
                      Named("theta") = slice(thetaWerte, 1, k-1),
                      Named("thetadot") = slice(thetadotWerte, 1, k-1));
}

```

Kreisbahn

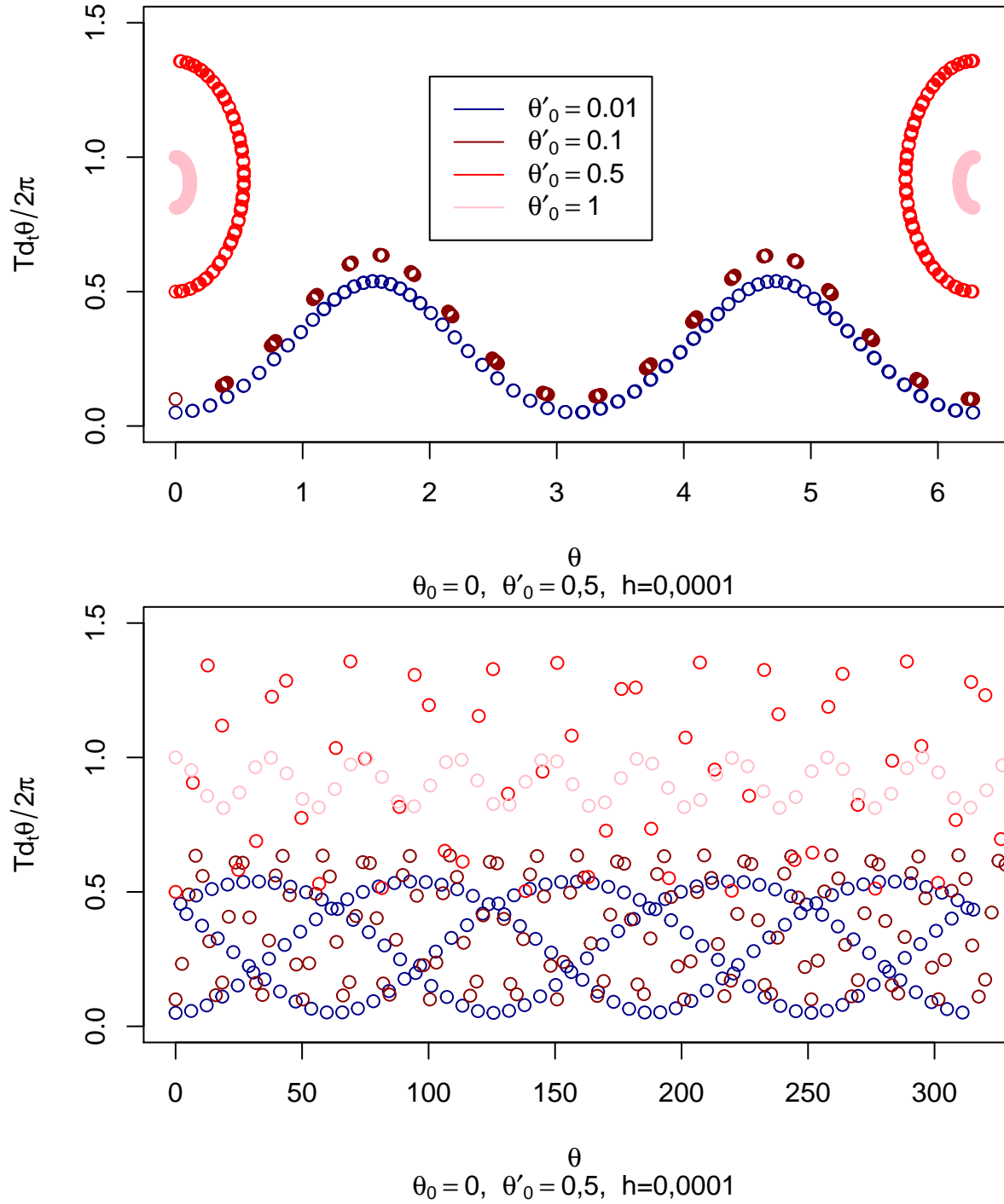
Für eine Exzentrizität von 0 erwarten wir eine gleichmäßige Eigenrotation, also ein Sinusverlauf.



Wir finden in Abb. ?? ein sinusförmiges, regelmäßiges Verhalten der Eigenrotation. Bzw. bilden sich Bögen an den Rändern des Moduloplots.

Phobos

Wir verwenden nun die Werte von Phobos, also $\epsilon = 0,015$ und $\alpha = 0,83$. Wir erwarten, dass der variierende Abstand die Eigenrotation im allgemein Fall stören wird und wir somit recht chaotische Plots erhalten können.



In Abb. ?? können wir sowohl ein periodisches Verhalten, als auch ein chaotisches Verhalten sehen. Offenkundig haben die Startwerte einen deutlichen Einfluss auf das Verhalten des Mondes. Für die große Anfangsrotationsgeschwindigkeiten erhalten wir die rötlich eingetragenen Ränder, welche im Plot ohne Modulo kaum Struktur tragen. Die sehr kleinen Startrotationen sind hingegen am unteren Rand der Grafik in Sinusformen aufzufinden. Wir haben in Abb 1 eine größere Reichweite an Anfangswerten dargestellt um ein Gefühl für die Verteilung zu gewinnen. Das grün gelbliche chaotische Verhalten stellt sich hier bei für θ -Werte in der Größenordnung von $1/7$ ein.

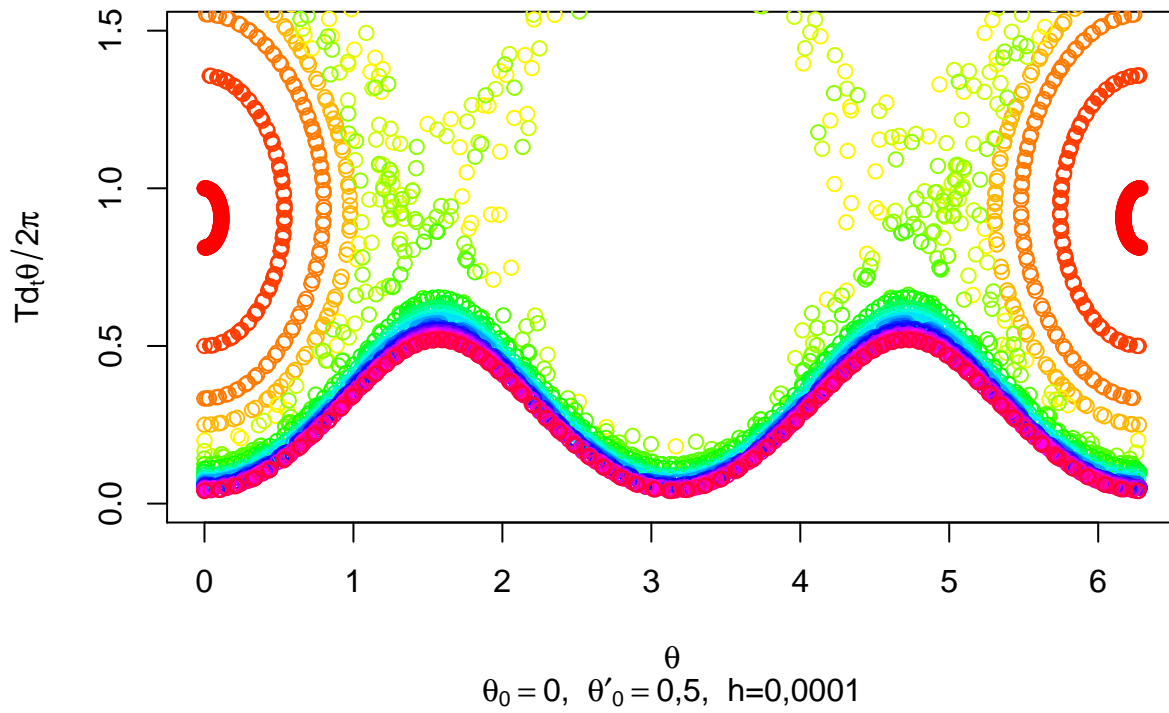


Abbildung 1: Entwicklungskurven für $R_0 = 2.9$

Fazit

Wir erhalten bei spezifischen Werten periodisches Verhalten zwischen θ und $\dot{\theta}$. Dies ist im physikalischen Sinne auch nachvollziehbar da in diesen Fällen die Rotationsfrequenz der Eigenrotation über einen rationalen Faktor mit der der Rotation um Mars übereinstimmt.

Wir können aber sehen, dass dieses Verhalten nicht immer eintritt und wir im Allgemeinen ein chaotisches Verhalten vorfinden.

Literatur