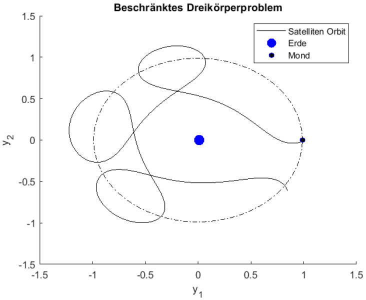
Satelliten

Universität Rostock

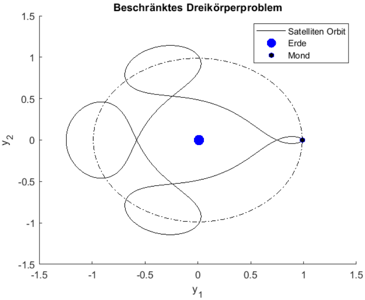
Till Nestke

14.01.2025

Einführung Thematik



(a) Berechnung 1



(b) Berechnung 2

Abbildung 1: Vergleich zweier Satelliten-Umlaufbahnen numerisch ermittelt

Gliederung

Allgemeines

Eine Gewöhnliche Differentialgleichung (kurz DGL) hat die Form

$$y^{(n)} = f(t, y, y', ...y^{(n-1)}), \tag{1}$$

wobei $y = y(t)$ Lösung der Differentialgleichung sei. Gleichung (1) wird Differentialgleichung n-ter Ordnung genannt.
Entsprechend haben Differentialgleichungen 1. Ordnung die Form:

$$y' = f(t, y), \tag{2}$$

welche erst durch Definition eines Anfangswertes $y(t_0) = y_0$ eindeutig bestimmt werden kann (**Anfangswertproblem**).

Ordinary **D**ifferential **E**quation (ODE)

- Ableitungen von y nach nur 1 Variablen

Partial **D**ifferential **E**quation (PDE)

- Ableitungen von y nach mehr als 1 Variablen
- z.B. vereinfachte Wärmeleitungsgleichung:

$$\frac{\partial y(x,t)}{\partial t} - \kappa \frac{\partial^2 y(x,t)}{\partial x^2} = 0 \tag{3}$$

- Vortrag von Eric zu PDE's

DGL 2. Ordnung

Eine DGL 2. Ordnung hat die Form

$$y'' = f(t, y, y'). \tag{4}$$

Hier werden zum Lösen zwei Anfangswerte benötigt:

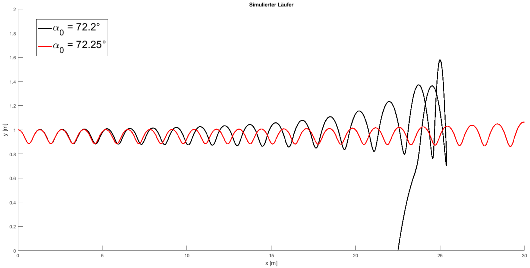
$$y(t_0) = y_0, \qquad y'(t_0) = y'_0. \tag{5}$$

Häufig muss in ein DGL-System 1. Ordnung übertragen werden. Mit $y_1 := y$ und $y_2 := y'$ wird Gleichung (4) zu:

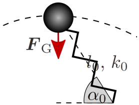
$$\begin{aligned} y'_1 &= y_2 & \qquad y_1(t_0) &= y_0 \\ y'_2 &= f(t, y_1, y_2) & \qquad y_2(t_0) &= y'_0. \end{aligned} \tag{6}$$

Weiteres

- Anfangswertprobleme sind sensibel
- Bsp. simuliertes Laufen: $y'' = \frac{k}{lm}(l_0 - l)y - g$



(a) Laufsimulation mit unterschiedlichen Beinwinkeln



(b) Läufer Modell

Abbildung 2: Laufsimulationsmodell

Weiteres

Randwertproblem:

- Vorgabe von Randwerten für $y(t)$
- z.B. sei $y' = f(t, y)$ und $t \in [t_0, t_{end}]$:

$$y(t_0) = y_0, \; y(t_{end}) = y_{end} \tag{7}$$

- Periodisches Randwertproblem:

$$y(t_{end}) - y(t_0) = y(t_0 + T) - y(t_0) = 0 \tag{8}$$

Restringiertes Dreikörperproblem

- Massen m_1, m_2 und m_3 bewegen sich unter Masseanziehung
- m_3 vernachlässichbar klein
- m_1 und m_2 laufen auf Kreisbahnen um gemeinsamen Masseschwerpunkt

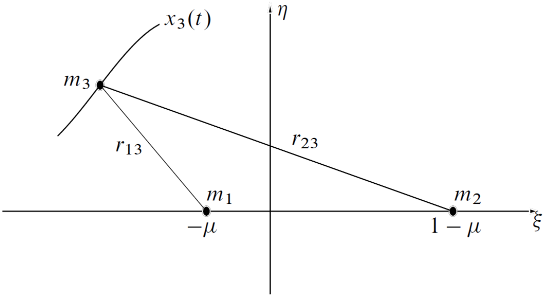


Abbildung 3: Restringiertes Dreikörperproblem (Deuflhard und Bornemann, 2013)

Für Masse 3 (z.B. ein Satellit) gelten folgende Bewegungsgleichungen:

Bewegungsgleichungen von m_3

$$\begin{aligned} y_1'' &= y_1 + 2y_2' - \mu' \frac{y_1 + \mu}{r_{13}} - \mu \frac{y_1 - \mu'}{r_{23}}, \\ y_2'' &= y_2 + 2y_1' - \mu' \frac{y_2}{r_{13}} - \mu \frac{y_2}{r_{23}}, \end{aligned} \tag{9}$$

wobei r_{12} und r_{13} den Abstand von m_3 zu m_1 bzw. m_2 darstellen.
Die Position von m_3 ist hierbei:

$$y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}. \tag{10}$$

Für das numerische Lösen wird in DGL-System 1. Ordnung umgeformt. Dabei gilt nun:

$$\begin{aligned} y_3 &= y_1', & \qquad y_5 &= y_1'' = y_3' \\ y_4 &= y_2', & \qquad y_6 &= y_2'' = y_4' \end{aligned} \tag{11}$$

und somit folgt:

Bewegungsgleichungen von m_3 als DGL-System

$$y' = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}' = \begin{pmatrix} y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} y_3 \\ y_4 \\ f_1(y.y') \\ f_2(y.y') \end{pmatrix} = \begin{pmatrix} y_3 \\ y_4 \\ f_1(y_1, y_2.y_3, y_4) \\ f_2(y_1, y_2.y_3, y_4) \end{pmatrix}. \tag{12}$$

Dabei sind $f_1(y, y')$ und $f_2(y, y')$ die durch (9) gegebenen rechten Seiten.

Was brauchen wir, um das Problem mit Matlab zu lösen?

- Anfangswerte für Position und Geschwindigkeit: $\vec{y_0}$, $\vec{y_0'}$
- Parameterwerte für μ und μ'
- Numerisches Verfahren zum Lösen der DGL

Solver	Problem Type	Accuracy	When to Use
`ode45`	Nonstiff	Medium	Most of the time. `ode45` should be the first solver you try.
`ode23`		Low	`ode23` can be more efficient than `ode45` at problems with crude tolerances, or in the presence of moderate stiffness.
`ode113`		Low to High	`ode113` can be more efficient than `ode45` at problems with stringent error tolerances, or when the ODE function is expensive to evaluate.
`ode78`		High	`ode78` can be more efficient than `ode45` at problems with smooth solutions that have high accuracy requirements.
`ode89`		High	`ode89` can be more efficient than `ode78` on very smooth problems, when integrating over long time intervals, or when tolerances are especially tight.
`ode15s`	Stiff	Low to Medium	Try `ode15s` when `ode45` fails or is inefficient and you suspect that the problem is stiff. Also use `ode15s` when solving differential algebraic equations (DAEs).
`ode23s`		Low	`ode23s` can be more efficient than `ode15s` at problems with crude error tolerances. It can solve some stiff problems for which `ode15s` is not effective.

Abbildung 4: Übersicht Matlab ODE-Solver

- stiff vs. nonstiff
- auf zu Matlab!

Periodische Lösung finden

Wir wollen folgendes periodisches Randwertproblem (siehe (8)) lösen:

$$y(t_{end}) - y(t_0) = y(t_0 + T) - y(t_0) = 0$$

Da die Bewegungsgleichungen (9) unabhängig von t sind (autonome DGL), können wir $t_0 = 0$ setzen.

Wie können wir das lösen?

- Residuen minimieren!
- aus Nullestellenapproximation wissen wir: ist x^* Nullstelle und x approximierte Nullstelle, gilt:

$$\begin{aligned} r &= f(x^*) - f(x) = -f(x) \\ |r| &= |f(x)| \end{aligned} \tag{13}$$

Übertragen auf das Dreikörperproblem bedeutet das:

$$|r| = |y(t_{end}) - y(0)| \overset{!}{=} 0 \tag{14}$$

Wie lässt sich $|r|$ minimieren?

- aus Quadraturmittelprobleme wissen wir: Kleinste Quadrate minimieren!

$$||r||_2^2 = ||y(t_{end}) - y(0)||_2^2 = \sum_{i=1}^{m} (y_i(t_{end}) - y_i(0))^2 , \tag{15}$$

wobei m die Anzahl der Einträge von y, also 4, entspricht.

Achtung: t_{end} muss auch mithilfe der Minimierung ermittelt werden!

Matlab Solver für Minimierungsproblem

- Was bietet Matlab an, um Residuum zu minimieren? Suchmaschine nutzen
- Matlab: Optimization Toolbox / Nonlinear Optimization / Solver-Based Nonlinear Optimization

Functions

`fminbnd`	Find local minimum of single-variable function on fixed interval
`fmincon`	Find minimum of constrained nonlinear multivariable function
`fminsearch`	Search for local minimum of unconstrained multivariable function using derivative-free method
`fminunc`	Find minimum of unconstrained multivariable function
`fseminf`	Find minimum of semi-infinitely constrained multivariable nonlinear function
`checkGradients`	Check first derivative function against finite-difference approximation *(Since R2023b)*
`optim.coder.infbound`	Infinite bound support for code generation *(Since R2022b)*

Abbildung 5: Übersicht Matlab Solver-Based Nonlinear Optimization

- Anfangswertproblem ist sehr sensibel
- wir wollen sinnige Ergebnisse und möglichst wenig Rechenaufwand
- Bereich der Anfangswerte einschränken: fmincon
- auf zu Matlab!

Arenstorf Orbits

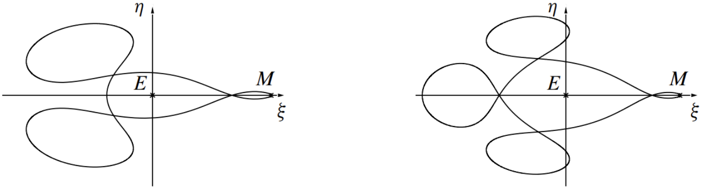


Abbildung 6: Arenstorf Orbits mit 3 bzw. 4 Schlaufen (Deuflhard und Bornemann, 2013)

Arenstorf Orbits

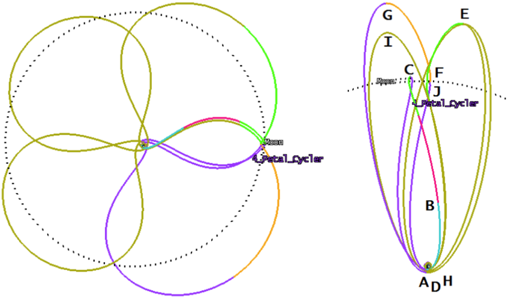


Abbildung 7: Arenstorf Orbits mit 4 Schlaufen, erdnah (Genova und Aldrin, 2015)

Literatur

Moler, C. B., Numerical Computing with Matlab, SIAM, 2004.

Deuflhard, P. & Bornemann, F. Numerische Mathematik 2: Gewöhnliche Differentialgleichungen, Berlin, Boston: De Gruyter, 2013.

Hairer, E., Nørsett, S., Wanner, G. Solving Ordinary Differential Equations I, Springer-Verlag, 1993.

Genova, A. L.,, Aldrin, B. A Free-Return Earth-Moon Cycler Orbit for an Interplanetary Cruise Ship, 2015. [online] https://ntrs.nasa.gov/api/citations/20150018049/downloads/20150018049.pdf [zuletzt abgerufen am 10.01.2024]

The MathWorks Inc. Optimization Toolbox (R2024b), Natick, Massachusetts: The MathWorks Inc., 2025. [online] `https://de.mathworks.com/help/optim/solver-based-nonlinear-optimization.html?s_tid=CRUX_lftnav` [zuletzt abgerufen am 13.01.2024]

Frenzlein, T. Numerisches Seminar - Vortrag Quadraturmittelprobleme, 2024.

Hessenmüller, M. Numerisches Seminar - Vortrag Nullstellen, 2024.

Westendorf, E. Numerisches Seminar - Vortrag PDEs, 2024.