

Angewandte Numerik 2

Abgabetermin: Freitag, 18.12.2015, vor der Übung

Für dieses Übungsblatt gibt es 11 Theorie- und 24 Matlab-Punkte, sowie 4 Theorie- und 17 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 9) bei 75,5 Theoriepunkten und 87 Matlabpunkten.

Aufgabe 35 (Programmieraufgabe: Runge-Kutta-Verfahren mit Schrittweitensteuerung)

(3T+2M+4M+2T*+6M+3M+1T*+6M*+1T*+4M*+3M* Punkte)

In dieser Aufgabe wollen wir das *restringierte Dreikörperproblem* betrachten, das die Bewegung von drei Körpern im Weltall unter dem Einfluss ihres gemeinsamen Gravitationsfeldes beschreibt.

Dabei gehen wir von vereinfachenden Annahmen aus: Zwei Massen m_1 (Erde) und m_2 (Mond) bewegen sich auf Kreisbahnen um ihren gemeinsamen Schwerpunkt. Die dritte Masse m_3 (Satellit) sei im Verhältnis zu m_1 und m_2 so klein, dass sie die Kreisbewegungen der beiden Körper mit den Massen m_1 und m_2 nicht beeinflusst. Ausserdem verlaufe die Bewegung aller drei Massen in einer Ebene.

Mit μ bezeichnen wir die relative Mondmasse, also das Verhältnis der Mondmasse zur Gesamtmasse von Erde und Mond. $\hat{\mu}$ bezeichne analog die relative Erdmasse, also

$$\mu = \frac{m_2}{m_1 + m_2} \quad \text{und} \quad \hat{\mu} = \frac{m_1}{m_1 + m_2} = 1 - \mu.$$

Mit obigen Vereinfachungen lässt sich die Bewegung der drei Himmelskörper in einem (y_1, y_2) -Koordinatensystem beschreiben. Dieses Koordinatensystem rotiere um den gemeinsamen Schwerpunkt von Erde und Mond, und zwar so, dass die beiden Himmelskörper Erde und Mond stets auf der y_1 -Achse liegen. Bei geeigneter Längenskalierung befindet sich dann die Erde im festen Punkt $(-\mu, 0)$ und der Mond im festen Punkt $(1 - \mu, 0)$.

Die Bewegung des Satelliten lässt sich in diesem Koordinatensystem durch das folgende System von Differentialgleichungen beschreiben. Dabei bezeichne $(y_1(t), y_2(t))$ den Ort, an dem sich der Satellit zum Zeitpunkt t befindet.

$$y_1'' = y_1 + 2y_2' - (1 - \mu) \frac{y_1 + \mu}{A} - \mu \frac{y_1 - (1 - \mu)}{B},$$
$$y_2'' = y_2 - 2y_1' - (1 - \mu) \frac{y_2}{A} - \mu \frac{y_2}{B}.$$

A und B sind gegeben durch

$$A := ((y_1 + \mu)^2 + y_2^2)^{3/2},$$
$$B := ((y_1 - (1 - \mu))^2 + y_2^2)^{3/2}.$$

Mit den Anfangswerten (zum Zeitpunkt $t_0 = 0$)

$$y_1(0) = 0.994, \quad y_1'(0) = 0, \quad y_2(0) = 0 \quad \text{und} \quad y_2'(0) = -2.0015851063790825$$

ergibt sich für $\mu = 0.012277471$ als Lösung ein geschlossener sogenannter (vierblättriger) Arenstorf-Orbit mit einer Periode von $T = 17.06521656015796255889$ (Monaten).

- a) Transformieren Sie obiges System von Differentialgleichungen auf ein System erster Ordnung. Geben Sie auch die Anfangswerte im transformierten System an.
- b) Schreiben Sie eine Matlabfunktion `dydt = f3KProblem(t, y, pMu)`, die $f(t, y)$ für das Dreikörperproblem zu einem Zeitpunkt t und einem Wert y berechnet. Der Parameter `pMu` gibt dabei die relative Mondmasse an.
- c) Schreiben Sie ein Matlabskript `dreiKoerperProblem`, das das Dreikörperproblem mit den oben angegebenen Anfangswerten und dem oben angegebenen Parameter $\mu = 0.012277471$ löst. Verwenden Sie dazu das klassische Runge-Kutta-Verfahren der Ordnung 4, das Sie in Aufgabe 32 vom letzten Übungsblatt 8 in Ihren Matlabfunktionen `yk = rungeKutta(f, y0, tk, bt)` und `bt = rk4()` bereits implementiert haben. Testen Sie mit unterschiedlichen (äquidistanten) Schrittweiten, insbesondere mit 0.1, 0.01 und 0.001. Stellen Sie die berechnete Bahn des Satelliten in der (y_1, y_2) -Ebene graphisch dar. Hinweis zur Implementierung: Übergeben Sie an `rungeKutta` eine anonyme Funktion `f = @(t,y)`, die Ihre Funktion `f3KProblem(t, y, pMu)` mit dem Parameter μ aufruft.
- d) Was beobachten Sie bei unterschiedlichen Schrittweiten? Woran liegt das?
- e) Schreiben Sie eine Matlabfunktion, die ein Anfangswertproblem mit dem eingebetteten Runge-Kutta-Verfahren 3(4) aus Beispiel 3.5.4 des Skripts löst. Verwenden Sie zur Schrittweitensteuerung den Algorithmus aus Abbildung 3.5 des Skripts. Überlegen Sie sich, welche Parameter Ihre Funktion benötigt und welche Werte sie zurück geben muss.
- Trennen Sie dabei den Algorithmus zur Schrittweitensteuerung sorgfältig vom Algorithmus des eingebetteten Runge-Kutta-Verfahrens. Ferner soll Ihre Funktion, die den Algorithmus für das eingebettete Runge-Kutta-Verfahren realisiert, das Butcher-Tableau für die beiden Runge-Kutta-Verfahren unterschiedlicher Ordnung (analog zu Aufgabe 32 a) vom letzten Übungsblatt) als Parameter erhalten.
- f) Erweitern Sie Ihr Matlabskript `dreiKoerperProblem` und lösen Sie das Dreikörperproblem auch mit Ihrer Matlabfunktion aus der letzten Teilaufgabe. Verwenden Sie für die Genauigkeit ϵ aus dem Algorithmus zur Schrittweitensteuerung den Wert 10^{-8} und geeignete Schranken für die Schrittweite. Geben Sie die Anzahl der benötigten Schritte sowie die minimale und die maximale verwendete Schrittweite aus. Stellen Sie wieder die berechnete Bahn des Satelliten in der (y_1, y_2) -Ebene graphisch dar und plotten Sie zusätzlich die verwendete Schrittweite über der Zeit.
- g) Was beobachten Sie, wenn Sie die Genauigkeit verkleinern oder vergrößern?
- h) Schreiben Sie analog zu Ihrer Matlabfunktion aus Teilaufgabe e) eine Matlabfunktion, die ein Anfangswertproblem mit dem Dormand-Prince-Verfahren DoPri 5(4) löst. Das Butcher-Tableau finden Sie in der angegebenen Literatur oder beispielsweise auch unter https://www-user.tu-chemnitz.de/~benner/Lehre/NumerikODE/Folie_embRK.pdf. Integrieren Sie analog zu Teilaufgabe f) auch diese Matlabfunktion in Ihr Matlabskript `dreiKoerperProblem`.
- i) Was verändert sich gegenüber dem Verfahren aus Teilaufgabe e)?
- j) Variieren Sie den Anfangswert für $y_2'(0) = -2.0015851063790825$ und testen Sie auch $y_2'(0) = -2.01$, $y_2'(0) = -2.02$, $y_2'(0) = -2.03$ und $y_2'(0) = -2.031732630$. Vergleichen Sie die Orbits.
- k) Testen Sie Ihre Programme auch mit den folgenden Anfangswerten:

$$y_1(0) = 1.2, \quad y_1'(0) = 0, \quad y_2(0) = 0 \quad \text{und} \quad y_2'(0) = -1.049357510.$$

Damit erhalten Sie für $\mu = 1/82.45$ als Lösung eine periodische Bahn des Satelliten mit Periode $T = 6.192169331$.

Aufgabe 36 (Programmieraufgabe: Steife Anfangswertaufgaben) (3M+2M+8T+2M+2M+4M* Punkte)

In dieser Aufgabe betrachten wir das skalare Modellproblem aus Beispiel 3.6.4 der Vorlesung, also das Anfangswertproblem

$$y' = \lambda y, \quad y(0) = 1,$$

dessen exakte Lösung durch $y(t) = e^{\lambda t}$ gegeben ist. Dabei sind wir insbesondere am Abklingverhalten der Funktion y interessiert, also am Verhalten der Funktion y für große Werte von t .

- a) Schreiben Sie ein Matlaskript `steif`, das Näherungslösungen dieses Anfangswertproblems für $\lambda = -20$ berechnet und visualisiert. Berechnen Sie die Näherungslösungen zunächst mit dem expliziten Euler-Verfahren im Intervall $[0, 1]$ und mit der Schrittweite $h_1 = 10^{-3}$. Sie können hierzu die Matlabfunktion `yk = eulerExplizit(f, y0, tk)` aus Aufgabe 26 verwenden. Plotten Sie Ihre Näherungslösung und die exakte Lösung gemeinsam in ein Schaubild.
- b) Da Sie am Abklingverhalten der Funktion interessiert sind, vergrößern Sie nun die Schrittweite. Wählen Sie insbesondere $h_2 = 10^{-2}$ und $h_3 = 10^{-1}$. Plotten Sie auch hier jeweils die Näherungslösung und die exakte Lösung in ein Schaubild. Was stellen Sie fest?
- c) Woran liegt das?
 - i) Überlegen Sie sich, welche Iterationswerte y^{k+1} das explizite Euler-Verfahren für dieses Anfangswertproblem ($y' = \lambda y$, $y^0 = y(0) = 1$, $\lambda < 0$) bei der Schrittweite h liefert. Für welche Werte von h verhält sich die Näherungslösung $(y^0, y^1, \dots, y^k, \dots)$ wie die exakte Lösung? Wie würde sich die Näherungslösung $(y^0, y^1, \dots, y^k, \dots)$ für andere Werte von h verhalten?
 - ii) Erklären Sie die numerischen Ergebnisse aus den Aufgabenteilen a) und b) mit Ihren Überlegungen aus Teil ci). Verifizieren Sie Ihre Überlegungen mit den Schrittweiten $h_4 = 0,08$ und $h_5 = 0,11$.
 - iii) Welche Iterationswerte y^{k+1} würden Sie mit dem impliziten Euler-Verfahren erhalten? Welche Bedingungen müssen Sie beim impliziten Euler-Verfahren an h stellen?
- d) Überprüfen Sie nun Ihre letzten Überlegungen numerisch. Berechnen Sie dazu die Näherungslösungen des Anfangswertproblems mit dem impliziten Euler-Verfahren. Sie können wieder die Matlabfunktion `yk = eulerImplizit(f, y0, tk, tol)` aus Aufgabe 26 verwenden. Wählen Sie zunächst die Schrittweiten $h_1 = 10^{-3}$ und $h_2 = 10^{-2}$ und plotten Sie wiederum die Lösungen.
- e) Vergrößern Sie nun wieder die Schrittweite. Verwenden Sie insbesondere auch die Schrittweite $h_3 = 10^{-1}$. Erhalten Sie eine Näherungslösung? Falls nicht, woran liegt das?
- f) Implementieren Sie das implizite Euler-Verfahren nun so, dass Sie auch für größere Schrittweiten, zumindest für $h_3 = 10^{-1}$ und $h_4 = 1$, eine Näherungslösung erhalten. Dafür gibt es mehrere Möglichkeiten (die Implementierung einer dieser Möglichkeiten genügt):
 - i) Ihre Matlabfunktion `eulerImplizit` erhält als weiteren Parameter eine Funktion, mit der Sie einen „guten“ Startwert für die Fixpunktiteration berechnen können. Diese Funktion müssen Sie dann im aufrufenden Matlaskript `steif` in Abhängigkeit vom konkreten Anfangswertproblem definieren.
 - ii) Sie implementieren das implizite Euler-Verfahren speziell für das Anfangswertproblem aus dieser Aufgabe.

Vergrößern Sie jetzt auch das Intervall, zumindest auf $[0, 5]$, und plotten Sie wiederum Ihre Lösungen. Interpretieren Sie Ihre Ergebnisse.

Hinweise:

Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Matlab-Files in einer E-mail mit dem Betreff **Loesung-Blatt09** an angewandte.numerik@uni-ulm.de (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.