Numerische Methoden D-PHYS

Nachname	
Vorname	
Leginummer	
Computername	slabhg _ _ _
Datum	6. August 2020

1	2	3	4	5	6	Punkte

- Bitte füllen Sie zuerst dieses Deckblatt aus.
- Schalten Sie Ihr Handy aus. Legen Sie Ihre Legi auf den Tisch.
- *Hilfsmittel:* 10 Seiten A4 eigenhändig verfasste Zusammenfassung, handgeschrieben. Vorlesungsunterlagen stehen elektronisch während der Prüfung zur Verfügung. Sonst keine Hilfsmittel.

Vor der Prüfung:

(i) Befolgen Sie die Anweisungen auf dem Bildschirm.

(ii) Kontrollieren Sie, dass die Verzeichnisse

- `questions/Aufgabe1/`,
- `questions/Aufgabe2/`,
- `questions/Aufgabe3/`,
- `questions/Aufgabe4/`,
- `questions/Aufgabe5/`,
- `questions/Aufgabe6/`

vorhanden sind. Diese Verzeichnisse enthalten die `Python`-Templates für die Aufgaben.

(iii) Im Verzeichnis `documentation` liegt das Vorlesungsskript `lnev.pdf`, der Aufschrieb 2019 `AufschriebNumPhysFS2019.pdf` und der Aufschrieb 2020 `AufschriebNumPhys2020live.pdf`.

Bitte wenden!

Während der Prüfung:

- Verwenden Sie die ersten 10 Minuten um **alle** Aufgaben in Ruhe zu **lesen**. Schreiben und Tippen sind in dieser Zeit verboten. Die Prüfungszeit beginnt erst nach dieser Lesezeit.
  Tipp: Beginnen Sie mit der Lösung der Aufgaben, bei der Sie sich am sichersten fühlen.
- Nicht mit Bleistift, rot oder grün schreiben.
- Stellen Sie den Lösungsweg auf Prüfungspapier klar dar und implementieren Sie danach sorgfältig.
- Speichern Sie Ihre Resultate *ausschliesslich* im Verzeichnis `questions`. Nur von diesen Dateien wird regelmässig ein Backup erstellt. Nur diese Dateien erscheinen in Ihrer elektronischen Abgabe. (Speichern Sie insbesondere keine Dateien in `documentation` oder `questions_backup`!)
- Die Benutzung der Templatedateien ist *nicht obligatorisch*, aber sehr empfohlen.
- Unter `questions_backup` finden Sie die unmodifizierten Templates. Vorsicht: Änderungen an Dateien in `questions_backup` werden in Ihrer Abgabe nicht berücksichtigt!
- Der `Python` Code in den Templates ist konsistent mit *4 Spaces* eingerückt. Falls Ihr Editor eine andere Einstellung hat, sollten Sie das umstellen. Im `gedit` Editor in der Fusszeile unten rechts: *Use Spaces* und *Tab Width* auf 4 stellen.
- Sie können das Tastaturlayout mit dem Shellkommando `setxkbmap us` und `setxkbmap ch` ändern.
- Melden Sie sich **NIE** ab. Schalten Sie den Computer **NIE** aus!

Viel Erfolg!

Am Ende der Prüfung:

(i) Alle Dateien speichern, alle Editoren und Terminals schliessen.

(ii) Melden Sie sich **nicht** ab. Schalten Sie den Computer **nicht** aus!

(iii) Bleiben Sie bitte sitzen. Ein Assistent wird Ihnen sagen, wann Sie den Prüfungsraum verlassen können. Wir bitten Sie um Geduld.

Weitere Hinweise:

- Im Falle einer Fehlermeldung: Bitte schliessen Sie die Fehlermeldung nicht und wenden Sie sich unverzüglich an einen Examinator.
- Melden Sie technische Probleme sofort. Extra Zeit wird Ihnen ab dem Zeitpunkt der Meldung angerechnet.
- Aus rechtlichen Gründen und als technische Absicherung wird Ihr Bildschirm während der gesamten Prüfung aufgezeichnet.

Notenskala: Die maximal erreichbare Punktzahl ist 48. Für eine Note 6 braucht man circa 42 und für eine 4 braucht man circa 22 Punkte.

Siehe nächstes Blatt!

ETHZ, D-MATH

Prüfung

Numerische Methoden D-PHYS, FS 2020

Dr. V. Gradinaru

06.08.2020

Prüfungsdauer : 180 Minuten.

1. Ausgleichsrechnung mit linearen Nebenbedingungen (8 Punkte)

Die Messungen (x, y) einer unbekannten Funtkion $y = f(x)$ sind im Template `1_Ausgleichsrechnung_template.py` gegeben.

a) Schreiben Sie auf Papier ein lineares Ausgleichsproblem um f mit einem Polynom $P_n(x)$ vom Grad n via gegebener Messungen zu approximieren.

b) Plotten Sie P_6, das f im Sinne der kleinsten Quadrate approximiert.

c) Die Messungen mit den Indizes $[0, 2, 4]$ werden als besonders wichtig gesehen; darum werden diese als Nebenbedingungen betrachtet. Schreiben Sie ein Ausgleichsproblem mit diesen Nebenbedingungen um f mit einem Polynom Q_n vom Grad n zu approximieren.

d) Plotten Sie Q_4, das f im Sinne dieses Ausgleichsproblems mit Nebenbedingungen approximiert, zusammen mit P_6 aus b).

(**Schriftliche Lösung nur hier**)

Siehe nächstes Blatt!

(Schriftliche Lösung nur hier)

Bitte wenden!

2. Potenzmethoden (8 Punkte)

Im Template `2_Potenzmethoden_template.py` ist eine Matrix $\mathbf{A}$ definiert. Die Eigenwerte von $\mathbf{A}$ sind im Folgenden so geordnet:

$$\lambda_1 < \lambda_2 < \ldots < \lambda_{n-1} < \lambda_n.$$

a) Geben Sie die Eigenwerte und entsprechenden Eigenvektoren von $\mathbf{A}$ aus.

b) Verwenden Sie die Potenzmethoden, um die folgenden Eigenwerte mit maximal 10000 Iterationen und einer absoluten Toleranz $|\rho - \rho_{\text{old}}|/|\rho|$ von 10^{-8} zu berechnen. Ihre Funktionen sollten jeweils den Eigenwert, den entsprechenden Eigenvektor und die benötigte Anzahl Iterationen ausgeben.

(i) λ_n, den grössten Eigenwert von $\mathbf{A}$;

(ii) λ_1, den kleinsten Eigenwert von $\mathbf{A}$, wobei Sie **keine** Gleichungssysteme lösen dürfen;
Hinweis: Verwenden Sie eine geeignet verschobene (Englisch:shifted) Matrix und Ihre Funktion von (i).

(iii) λ_1, den kleinsten Eigenwert von $\mathbf{A}$, wobei Sie lineare Gleichungssysteme mit einer **einzigen** LU-Zerlegung lösen sollen;

(iv) Den Eigenwert von $\mathbf{A}$, welcher am nächsten an 42 liegt.

c) Erklären Sie die beobachteten/erwarteten Unterschiede in der Anzahl der Iterationen und in der Laufzeit bei den Aufgaben in b) für diese Matrix.

(Schriftliche Lösung nur hier)

Bitte wenden!

(**Schriftliche Lösung nur hier**)

Siehe nächstes Blatt!

3. Nichtlineare Algebraische Gleichung (8 Punkte)

a) Berechnen Sie numerisch den Schnittpunkt der Ellipsoide

$$\begin{aligned}
\frac{x^2}{4} + \frac{y^2}{9} + \frac{z^2}{25} &= 1 \\
(x-2)^2 + \frac{(y-1)^2}{2} + \frac{(z-1)^2}{2} &= 1 \\
\frac{(x+3)^2}{75} + \frac{(y+1)^2}{3} + \frac{(z+1)^2}{3} &= 1
\end{aligned}$$

mit einer Funktion aus `scipy.optimize` und mit einer eigenen Implementierung des Newton-Verfahrens im Template `3_Newton_template.py`.

b) Finden Sie die vier Fehler im folgenden Newton-Code (schreiben Sie direkt auf dieses Papier):

```
def newton(f, x0, df, tol=1e-8, N=1000):
    error = np.linalg.norm(f(x0))
    i=0
    x=x0
    while error > tol and i > N:
        fprime = df(x)
        fx = f(x)
        x = x + np.linalg.solve(fprime, fx)
        error = np.linalg.norm(f(x))
        i = 1
        return x
```

c) Wahr oder Falsch? (Schreiben Sie Ihre Antwort und Ihre Begründung direkt auf dieses Papier.)

(c.1) Die Newton-Iteration konvergiert immer quadratisch.

(c.2) Das Newton-Verfahren funktioniert nicht in der Raumfahrt, da es dort die Relativitätstheorie gilt.

Bitte wenden!

(Schriftliche Lösung nur hier)

Siehe nächstes Blatt!

(Schriftliche Lösung nur hier)

Bitte wenden!

4. Splitting separabler Hamiltonsysyteme (8 Punkte)

a) Leiten Sie die Teilschritte des Splitting-Verfahrens für Hamiltonsche Differentialgleichungen mit separabler Hamilton-Funktion $H(\mathbf{p}, \mathbf{q}) = T(\mathbf{p}) + V(\mathbf{q})$ im Detail her.

b) Schreiben Sie die oberen Schritte für $V(\mathbf{q}) = M(\|\mathbf{q}\|)$ mit dem Morse-Potential

$$M(x) = V_0 + V_0 \left(e^{-2\beta(x-s)} - 2e^{-\beta(x-s)}\right)$$

und $T(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{p}$ auf Papier.

(Das ist ein Model in der Molekulardynamik mit Zentralkraft aus dem Morse-Potential.)

c) Wir nehmen $\mathbf{p}, \mathbf{q} \in \mathbb{R}^2$; die Modelparameter V_0, β, s, die Startwerte, die Endzeiten (früh und spät) und die jeweils nötige Anzahl an Schritten für die folgenden Simulationen sind im Templatefile `4_HamiltonSplitting_template.py` angegeben. Das Templatefile generiert automatisch numerische Ergebnisse und Bilder passend zu den folgenden Aufgaben.

Implementieren Sie die Teilschritte des Splitting-Verfahrens und berechnen Sie die Lösungen mit der Runge–Kutta–Nystrom-Methode der Ordnung 4 und 6 (d.h. `BM42` und `BM63`).

d) Lösen Sie diese Differentialgleichung mit dem standard adaptiven Integrator aus `scipy.integrate`, der die Runge–Kutta-Methode nach Dormand–Prince verwendet; setzen Sie hier die Toleranzen `rtol = atol =` 10^{-6}.

e) Wenden Sie die drei Methoden auch für die Langzeitpropagation an. Bechreiben Sie die Vor- und Nachteile jeder dieser drei Ansätze für Kurzzeit- und für Langzeitintegration.

Siehe nächstes Blatt!

(**Schriftliche Lösung nur hier**)

Bitte wenden!

(Schriftliche Lösung nur hier)

Siehe nächstes Blatt!

(Schriftliche Lösung nur hier)

Bitte wenden!

5. Runge–Kutta-Verfahren (8 Punkte)

Das Butcher-Schema der Runge–Kutta-3/8-Regel ist

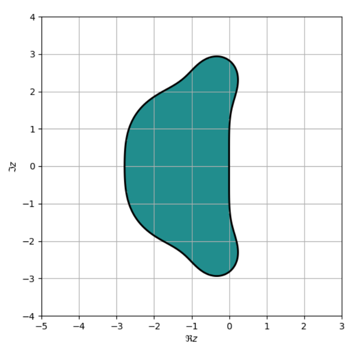
$$\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\
\frac{2}{3} & -\frac{1}{3} & 1 & 0 & 0 \\
1 & 1 & -1 & 1 & 0 \\
\hline
 & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8}
\end{array}.$$

Verwenden Sie für diese Aufgabe das Template `5_RK38_template.py`.

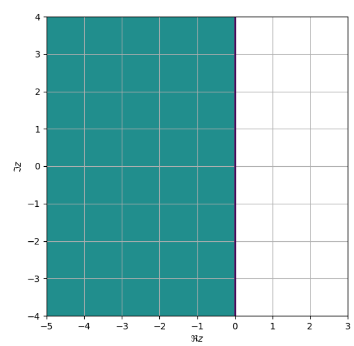
a) Ist die Runge–Kutta-3/8-Regel eine explizite oder eine implizite Methode? Finden Sie die Stabilitätsfunktion dieser Regel.
Hinweis: Sie dürfen die Funktionen `symbols, Matrix, pprint` *aus* `sympy` *verwenden. Die Inverse einer* `Matrix U` *kann mit* `U.inv()` *berechnet werden;* `*` *multipliziert zwei* `Matrix`*-Objekte im mathematischen Sinn.*

b) Finden Sie die Konvergenzordnung dieses Verfahrens, indem Sie entsprechene numerische Experimente für die logistische Differentialgleichung mit $\alpha = \beta = 10$, Startwert $y(0) = 0.01$ und Endzeit $T = 1$ durchführen.

c) Eines der folgenden Bilder entspricht dem Stabilitätsgebiet der Runge–Kutta-3/8-Regel. Geben Sie an welches und begründen Sie Ihre Antwort.



a) Der grüne Bereich



b) Die ganze linke Halbebene

d) Für jedes der Stabilitätsgebiete aus c) wählen Sie eine Zeitschrittweite h, so dass die Methode für das Testproblem $\dot{y} = \lambda y$ mit $\lambda = -3-3i$ (wobei $i^2 = -1$) konvergiert.

(Schriftliche Lösung nur hier)

Bitte wenden!

(Schriftliche Lösung nur hier)

Siehe nächstes Blatt!

6. Quadraturformeln (8 Punkte)

Sei die Quadraturformel auf $[-1, 1]$ gegeben durch die Knoten und Gewichte im File `6_Quadrature_template.py`.

a) Welche Ordnung hat diese Quadraturformel? Schreiben Sie ein Programm, mit dem Sie Ihre Antwort begründen.

b) Zu welcher Klasse von Quadraturformeln gehört diese Quadraturformel? Begründen Sie Ihre Aussage.

c) Schreiben Sie ein Programm, das eine auf diesen Knoten und Gewichten basierte zusammengestetzte Quadraturformel berechnet. Wenden Sie dies für die Funktionen

$$f_1(x) = \frac{1}{1+5x^2}, \qquad f_2(x) = \sqrt{x}$$

auf dem Intervall $[0, 1]$ an. Plotten Sie die Fehler in doppellogarithmischer Skala. Die exakten Werte der zwei Integrale sind:

$$\int_0^1 f_1(x)\,dx = \frac{\arctan(\sqrt{5})}{\sqrt{5}}, \qquad \int_0^1 f_2(x)\,dx = \frac{2}{3}.$$

Welche Ordnungen beobachten Sie für die zwei Fehler? Plotten Sie auf dem Bild mit den Fehlern die Graphen der Polynome, die diesen Ordnungen entsprechen.

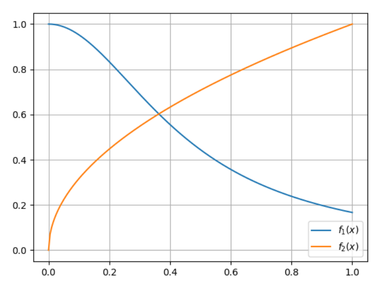


Abbildung 1 – Funktionen f_1 und f_2.

d) Begründen Sie auf Papier das beobachtete oder erwartete Verhalten der Fehler in den zwei Fällen.

Bitte wenden!

(Schriftliche Lösung nur hier)

Siehe nächstes Blatt!

(Schriftliche Lösung nur hier)

Bitte wenden!

Weitere leere Seiten: (**Schriftliche Lösung nur hier**)

(Schriftliche Lösung nur hier)

Bitte wenden!

(Schriftliche Lösung nur hier)

Siehe nächstes Blatt!

(Schriftliche Lösung nur hier)

Bitte wenden!