# Online-Prüfung im Fach
# Computational Engineering

Für M-TI (FB 6)

WS 2021/22, 2. Prüfungszeitraum

Prüfungstermin am 21.03.2022

Prüfer: Prof. Dr. Marc Kirch

Berliner Hochschule für Technik

**Bearbeitungszeit**: 120 Min. (Prüfungszeit) + 10 Min. (Extrazeit) = 130 Minuten

**Bewertungsrahmen**: Für eine 100%-Wertung benötigen Sie nur 54 der maximal 60 in dieser Klausur zu erreichenden Punkte. Eine Wertung von über 100% ist dabei nicht möglich.
Bestanden haben Sie ab einer Punktzahl von 27.

**Eidesstattliche Erklärung**: Mit meiner Unterschrift bestätige ich mit der Prüfungsform einverstanden zu sein, die rechtzeitig zuvor kenntlich gemachten Regeln zur Durchführung der Prüfung erhalten und verstanden zu haben und die Prüfung eigenständig und ohne Mithilfe von weiteren Personen bearbeitet zu haben:

Name: ________________________ Matrikelnr.: ___________

Unterschrift: ________________________

Hinweis: Wenn Sie keinen Drucker zur Verfügung haben, um das Deckbklatt auszudrucken, dann können Sie diesen Teil auch handschriftlich auf ein Extrablatt kopieren und ausfüllen.

| Aufgabe: | 1 | $2a$ | $2b$ | 3 | $\sum$ |
|---|---|---|---|---|---|
| Punkte: | | | | | |

| | |
|---|---|
| Note: | |
| Berlin, den | |

(Unterschrift Prof. Kirch)

## Weitere Hinweise zur Bearbeitung

- Diese Prüfung ist eine Prüfungsleistung im Sinne der RSPO. Mit dem endgültigen Hochladen Ihrer Lösungen im Prüfungsportal, zählt dies im Rahmen der zum Zeitpunkt der Prüfung geltenden Regelung als zu bewertender Prüfungsversuch.
- Die auf dem Deckblatt angegebene Bearbeitungszeit ist die Zeitspanne zwischen der Verfügbarmachung der Aufgaben und der Schließung der Möglichkeit die Lösungen hochzuladen. Das Portal zum hochladen der Aufgaben schließt automatisch und nach der Schließung sind keine weiteren Abgaben möglich und werden auch auf anderem Wege nicht akzeptiert. Probleme mit der Hochladeprozess müssen innerhalb der Bearbeitungszeit kommuniziert werden.
- Außer der direkten oder indirekten Hilfe durch Zusammenarbeit mit anderen Personen, dürfen Sie alle möglichen Hilfsmittel verwenden.
- Sie können ihre Lösungen als m-Files oder als reine Textfiles hochladen.
- Die Programme müssen für eine volle Punktzahl fehlerfrei laufen und alle in der Aufgabenstellung geforderten Aufgaben abarbeiten.
- Im Sinne der Aufgabenstellung überflüssige Code-Bestandteile führen ggfs. zu Punktabzug. Kommentare sind hiervon ausgenommen.
- Schreiben Sie ihre Programme so einfach und linear wie möglich und nur so komplex wie nötig.

## Aufgabe 1 (15 Punkte)

Geben Sie ein Matlab Skript an, in welchem Sie eine Laufzeitanalyse der Berechnung einer Matrixinversion mittels des Matlabbefehls `inv` durchführen.

Gehen Sie dabei wie folgt vor: Sie füllen eine quadratische Matrix $\mathbf{A}_N = (a_{ij})_N$ der Dimension $N = 2^k$ für $k = 2, 3, \ldots, 10$ mit natürlichen Zufallszahlen $a_{ij} \in \{0, 1, 2 \ldots, 9\}$ . Berechnen Sie für jede Dimension $N$ die Inverse mindestens $R = 20$ mal für eine neu gefüllte Zufallsmatrix. Nehmen Sie dabei die Rechenzeit $T$ der Inversenbildung $\texttt{inv}(\mathbf{A}_N)$ auf.

Bilden Sie den Mittelwert und Standardabweichung der Rechenzeiten über alle $R$ Wiederholungen für jede Dimension $N$. Fertigen Sie eine Graphik an, in der die mittlere Rechenzeit mit Fehlerbalken als Funktion der Dimension dargestellt wird. Approximieren Sie den Verlauf der mittleren Rechenzeiten durch ein Polynom vom Grad 3. Sie dürfen hierzu eine implementierte Matlab-Funktion verwenden. Der Graph der Approximierenden wird ebenfalls in der Graphik dargestellt. Die Graphik ist aussagekräftig formatiert.

---

## Aufgabe 2 (15 + 15 = 30 Punkte)

Sie finden im Moodle ein File mit dem Namen `Signal.mat`. Dieses File enthält einen Vektor $\vec{S} = (S_1, S_2, S_3, \ldots, S_K)^t$ mit den Signalwerten $S_k$ eines $2\pi$-periodischen Signals, die jeweils zu den Zeitpunkten $t_k$ aufgenommen wurden, die im Vektor $\vec{t} = (t_1, t_2, t_3, \ldots, t_K)$ abgelegt sind ($t \in [0, 2\pi]$), der ebenfalls in dem File enthalten ist.

a) Schreiben Sie ein Matlab Skript, welches das File `Signal.mat` einliest. Dann berechnen erechnen Sie in Ihrem Skript die Fourierkoeffizienten $a_n$ und $b_n$ des Signals bis zur Ordnung $n = 100$. Hierzu dürfen Sie die Matlab-Funktion `trapz` verwenden. Mit den Fourierkoeffizienten bilden Sie die Werte der Fourierreihe an den Zeitpunkten $t_k$

$$F_k = F(t_k) = \frac{a_0}{2} + \sum_{n=1}^{100} \left(a_n \cos(nt_k) + b_n \sin(nt_k)\right)$$

Die so erhaltenen Werte der Fourierreihe des Signals seien im Vektor $\vec{F} = (F_1, F_2, F_3, \ldots, F_N)^t$ zusammen gefasst. Geben Sie drei Plots aus: Im ersten Plot stellen Sie die Werte der Fourierkoeffizienten geeignet dar, im zweiten jeweils $\vec{S}$ und $\vec{F}$ über $\vec{t}$ und im dritten die punktweise Abweichung $\vec{S} - \vec{F}$ über $\vec{t}$.

b) In ihrem Skript suchen Sie nun den Fourierkoeffizienten mit dem betragsmäßig größten Wert. Ausgehend von diesem Wert bilden Sie die Fourierreihen $\vec{F}^{(p)}$ für $p = 1, 2, 3, \ldots, 10$, in denen Sie jeweils alle Fourierkoefizienten zu Null setzen, die kleiner als $p\%$ des Wertes des maximalen Fourierkoeffizienten sind.

Sie berechnen für jeden Wert von $p$ die Zahl

$$e_p = |\vec{S} - \vec{F}^{(p)}|$$

Dann geben Sie einen Plot aus, in denen Sie jeweils das ursprüngliche Signal $\vec{S}$ und das geglättete Signal $\vec{F}^{(10)}$ darstellen. Zu letzt erzeugt ihr Skript einen Plot, in dem Sie die Werte von $e_p$ über $p$ auftragen. Alle Graphiken sind passend zu formatieren.

---

## Aufgabe 3 (15 Punkte)

Betrachten Sie einen zweidimensionalen Random Walk auf einer Kreisfläche mit dem Radius $R = 1$. Der Random Walker startet im Mittelpunkt des Kreises. Seien die kartesischen Koordinaten des Walkers im $n$-ten Schritt $(x_n, y_n)$, dann führt ihn der nächste Schritt zu den Koordinaten

$$x_{n+1} = x_n + \delta \cos(\varphi_n), \qquad y_{n+1} = y_n + \delta \sin(\varphi_n)$$

d.h. er springt in die Richtung $\varphi_n$ mit der Schrittlänge $\delta$. Die Richtung $\varphi_n$ wird in jedem Schritt zufällig gleichverteilt im Intervall $[0, 2\pi]$ gewählt. Die Schrittlänge bleibt in einem Lauf jeweils fixiert und wird vorgegeben. Der Walker führt so lange Sprünge aus, bis er in einem Sprung den Rand der Kreisscheibe überschreiten würde.

Fertigen Sie ein Matlab Skript an, in dem Sie für die Werte $\delta = \{10^{-4}, 10^{-3}, 10^{-2}\}$ den Walker jeweils $R$-mal laufen lassen und dabei aufzeichnen, wie viele Sprünge er macht, bis er den Rand der Kreisscheibe überschreiten würde. In Ihrem Skript wird dann für jeden Wert von $\delta$ ein Histogramm ausgegeben, welches die Verteilung der Schrittzahlen zeigt. Als weiteren Plot gibt ihr Skript eine Grafik aus, in der die Mittelwerte der Schrittzahlen über $\delta$ mittels einer halblogarithmischer Skala dargestellt werden.

Zum testen nehmem Sie gern kleinere Werte, für die Simulation nimmt man $R > 100$. Die Plots sind passend zu formatieren.