

Allgemeine Informationen: Dieses Aufgabenblatt enthält schriftliche und/oder Programmieraufgaben. Bitte kombinieren Sie alle Lösungen zu den schriftlichen Aufgaben zu einem einzelnen PDF Dokument, welches Sie nach folgendem Schema benennen: `{lastname}-written.pdf`. Sie können Ihre Lösungen auch scannen oder fotografieren. Achten Sie in diesem Fall auf die Lesbarkeit. Es werden JPEG/PNG Bilddateien akzeptiert welche wie folgt benannt werden müssen: `{exercisenummer}-{lastname}-written.{jpeg/png}`. Stellen Sie sicher, dass alle Rechenschritte nachvollziehbar sind und kombinieren Sie nicht zu viele kleine Schritte zu einem einzelnen. Die Programmieraufgaben müssen in *Julia* gelöst sein und Ihr Quellcode sollte nach folgendem Schema benannt sein: `{exercisenummer}-{lastname}.jl`.

Informationen zu diesem Übungsblatt: Mit den folgenden Aufgaben machen Sie sich vertraut mit der Programmiersprache [Julia](#) und dem package [Plots](#).

Folgende Ressourcen könnten hilfreich sein:

<https://benlauwens.github.io/ThinkJulia.jl/latest/book.html>

<https://docs.julialang.org/en/v1/>

<https://docs.juliaplots.org/stable/>

<https://docs.juliaplots.org/latest/generated/gr/>

Bitte beachten Sie, dass zu jeder Aufgabe in diesem Übungsblatt eine Vorlage in Form einer *Julia* Datei zur Verfügung steht.

- (1) (3.5 Punkte) In der Linearen Algebra unterscheiden wir zwischen Skalaren, Vektoren und Matrizen. Dabei gibt es verschiedene Operationen zwischen Instanzen solcher. Für diese Aufgabe implementieren Sie eine Untermenge dieser Operationen:

- a) Skalar/Skalar Multiplikation
- b) Skalar/Vektor Multiplikation
- c) Vektor/Matrix Multiplikation
- d) Matrix/Matrix Multiplikation
- e) Das Skalarprodukt zweier Vektoren (zu engl. dot-product)
- f) Matrix/Matrix Addition
- g) Das Transponieren einer Matrix

Nutzen Sie die zur Verfügung gestellte *Julia* Vorlage in OLAT und implementieren Sie jede Funktionalität in der respektiven Funktionsdefinition. Ihre Implementierung sollte im Optimalfall alle Tests bestehen! Nutzen Sie Folgendes **nicht**:

- 1. Jedwede *Julia packages* (im speziellen **LinearAlgebra**).
- 2. Die Multiplikations- (`*`) und Transpositions- (`'`) Operatoren.

Im weiteren Verlauf des Semesters dürfen Sie darauf zurückgreifen.

- (2) (3.5 Punkte) Bei einer Gitarre entsteht ein musikalischer Ton durch das Schwingen einer Saite – die periodischen Stöße gegen die Luft erzeugen Schallwellen. Wir können einen *perfekten* Ton mathematisch durch eine Sinusfunktion ausdrücken, welche die Druckabweichung zu einem bestimmten Zeitpunkt t beschreibt:

$$y(t) = A \sin(2\pi f t)$$

wobei A die Amplitude (Lautstärke) und f die Frequenz der Schwingung in Hertz (Hz) ist. Eine Note in der Musiktheorie ist dabei einer gewissen Frequenz zugeordnet, z.B. hat der **Kammerton A** 440 Hz.

- a) Stellen Sie die obige Funktion für die Noten **E (329.63 Hz)**, **G# (415.30 Hz)** und **B (493.88 Hz)** mit Amplitude 1 über einen Zeitraum von 12ms graphisch dar (mit GR's plot Funktion). Achten Sie darauf dass sie kleine Zeitschritte für die Achsen verwenden, damit die Schwingungen auch sichtbar sind.
- b) Wenn zwei oder mehr Wellen sich gleichzeitig in demselben Raum ausbreiten, kommt es zu einer Überlagerung der akustischen Signale. Das Superpositionsprinzip besagt, dass der Gesamteffekt mehrerer gleichzeitig auftretender Einzeleffekte durch die Summe der Einzeleffekte gegeben ist (gilt für lineare Wellengleichungen). Für einen Dreiklang, welcher aus drei Tönen besteht können wir somit dessen Funktion als die Summe der einzelnen Ton Funktionen y_0 , y_1 , und y_2 bilden:

$$y(t) = y_0(t) + y_1(t) + y_2(t)$$

Bilden Sie die Funktion des Dreiklangs basierend auf den Noten **E**, **G#** und **B** und stellen Sie diese graphisch im selben Plot dar. Der Graph sollte mit einer dickeren Linie hervorgehoben werden.

- c) Beschriften Sie ihren Plot, d.h. mit einen entsprechenden Titel, einer Legende und Achsenbeschriftung. Das schlußendliche Resultat sollte dem folgenden entsprechen:

