

Name des Studierenden :

_____ Dieses Dokument muss ergänzt und am Ende des Moduls dem/der Dozierenden zurückgegeben werden _____

Arbeitsblatt für den Workshop Octave

Lernziele und Forderungen des Moduls

Dieses Dokument enthält die Liste der Lernziele, die von den Studierenden am Ende des Moduls erreicht werden müssen (Dauer: zweimal 3 Perioden). Die Lernziele sind in "Kompetenzen zu erwerben" beschrieben. Der/die Studierende muss in der Lage sein, **selber durchzuführen**, was verlangt ist. In gewissen Fällen muss er/sie nur die erwähnten Abschnitte lesen. Ein Kreuz kann in der rechten Spalte eingetragen werden, aber nur wenn das entsprechende Lernziel erreicht wird, d.h.: **die Aufgaben sind abgeschlossen und die Programme laufen und geben die erwarteten Resultate**. Das Kontrollblatt muss am Ende des Moduls dem/der Dozierenden zurückgegeben werden.

Alle von dem/der Studierenden geschriebene Programme müssen mindestens eine Kommentarzeile enthalten mit dem Namen und dem Datum. Sie müssen in ein Verzeichnis `name.vorname/` gelegt werden, welches schlussendlich archiviert und dem/der Dozierenden durch Email zugeschickt wird. Zum Archivieren kann man den Unix Befehl

```
> tar cfz name.vorname.taz name.vorname
```

oder den Befehl `zip` anwenden.

Das Modul wird erfüllt, wenn der/die Dozierende-r das ergänzte Kontrollblatt und das archivierte Verzeichnis erhalten hat, und wenn der Inhalt des Verzeichnisses mit den Forderungen übereinstimmt, die im Kurs gegeben werden.

1 Octave-Befehle Zeile für Zeile

- | | |
|---|--------------------------|
| 1. Die mathematischen Grundbefehle: Abschnitt 3 lesen, S. 3 | <input type="checkbox"/> |
| 2. Die Befehle auf Seite 3 in Octave selbst schreiben | <input type="checkbox"/> |
| 3. Die Informationen über die Konstanten <code>e</code> , <code>i</code> und <code>pi</code> mit Hilfe des Befehls <code>help</code> finden | <input type="checkbox"/> |
| 4. Die Rolle des Separators <code>;</code> verstehen | <input type="checkbox"/> |

2 Skript-Files

- | | |
|--|--------------------------|
| 1. Der Begriff "Script-File": Abschnitt 4, S. 3-4 lesen | <input type="checkbox"/> |
| 2. Das Octave Programm <code>triangle.m</code> selber schreiben und das entsprechende Script-File erzeugen | <input type="checkbox"/> |
| 3. Das Script-File <code>triangle.m</code> in Octave ausführen | <input type="checkbox"/> |

3 Vektoren und Matrizen

1. Vektoren und Matrizen: Abschnitt 5, S. 4-5 lesen
2. Die Befehle auf Seite 5 selber schreiben
3. Verstehen des Unterschieds zwischen einem Zeilenvektor und einen Spaltenvektor.
Verstehen der Bedeutung des Operators '
4. Das Skalarprodukt zweier Vektoren ausführen
5. Eine Matrix deklarieren, z.B. mit den Dimensionen 3×3
6. Die Determinante einer Matrix berechnen
7. Eine Matrix mit einem Vektor multiplizieren
8. Ein lineares Gleichungssystem in ein Matrixensystem transformieren
9. Ein lineares Gleichungssystem lösen

4 Funktions-Files

1. Der Begriff "Funktions-File": Abschnitt 6, S. 6 lesen
2. Den Code der Funktion `f01.m` selber schreiben und das entsprechende Funktions-File erzeugen
3. Die Funktion für einen einzigen Wert der Variable evaluieren
4. Die Funktion für eine Liste Argumente mit einem einzigen Befehl evaluieren
5. Wie kann eine Funktion ohne Verwendung eines Skripts-Files evaluiert werden ? Vorteile, Nachteile?

5 2D und 3D Graphik

1. Graphische Darstellung einer Funktion: Abschnitt 7, S. 6-7 lesen
2. Den Code für die Zeichnung der Funktion `f01.m` selber schreiben
3. Den Code der Funktion `f02.m` selber schreiben und das entsprechende Funktions-File erzeugen
4. Den Code für den Graphen der Funktion `f02.m` mit zwei Variablen selber schreiben

6 Aufgaben

1. Aufgabe 10.1, S. 13
2. Aufgabe 10.2, S. 13
3. Aufgabe 10.3, S. 13

7 Lineare Regression für den Widerstand als Funktion der Temperatur

1. Die Aufgabenstellung in Abschnitt 11.1 lesen und verstehen
2. Das Datenfile `copper.dat` von der Webseite in Ihr Arbeitsverzeichnis kopieren
3. Den Code um die Daten zu lesen und visualisieren verstehen und selber schreiben
4. Den Befehl `polyfit()` verwenden um die Parameter der Regressionsgerade zu bestimmen
5. Den Code schreiben um eine gute Graphik zu erzeugen, Abschnitt 11.4
6. Das Funktionsfile `LinearRegression.m` von der Webseite in Ihr Arbeitsverzeichniss kopieren
7. Den Befehl `LinearRegression()` verwenden um die Parameter der Regressionsgerade und deren Standardabweichung zu bestimmen
8. Den Code für eine Regressionsparabel durchsehen und das Resultat verstehen, Abschnitt 11.6

8 Regression angewandt auf ein Wärmeleitungsproblem

1. Die Aufgabenstellung in Abschnitt 12.1 lesen und verstehen
2. Das Datenfile `Messdaten.prn` und das Scriptfile `ReadHeatData.m` von der Webseite in Ihr Arbeitsverzeichnis kopieren
3. Den Code um die Daten zu lesen von verstehen, Abschnitt 12.2
4. Den Algorithmus für die Vorbereitung der Daten verstehen und den Code schreiben, Abschnitt 12.3
5. Die lineare Regression ausführen und die Resultate visualisieren, Abschnitt 12.4

_____ Dieses Dokument muss ergänzt und am Ende des Moduls dem/der Dozierenden zurückgegeben werden _____

Datum: Unterschrift:
