

Übungsblatt 6

Ausgabe: 22.11.2017

Die Tabellenumgebung `uitable`

Das Darstellen von Tabellen im Command Window von Matlab ist meist sehr unübersichtlich. Um Abhilfe zu schaffen, bietet Matlab die Umgebung `uitable`, mit der es möglich ist, Tabellen in separaten `figure`-Fenstern anzeigen zu lassen. Das `ui` in `uitable` ist eine Abkürzung für den englischen Begriff „user interface“ (zu dt.: „Benutzerschnittstelle“). Mit `uitable` ist es möglich, interaktive Tabellen zu erstellen, deren Einträge man mit Buttons oder Schieberegler beeinflussen kann. Auf diesem Übungsblatt werden wir uns jedoch nicht mit diesen interaktiven Features, sondern mit der rein tabellarischen Darstellung und Wiedergabe von Daten beschäftigen.

Bei der Erstellung eines `uitable` wird eine zweidimensionale Tabelle angelegt. Dabei kann man einige Argumente direkt bei der Erzeugung an `uitable` übergeben. Ein standardmäßiger Aufruf von `uitable` sieht wie folgt aus:

```
uitable('Argument1', Argumentwert1, 'Argument2', Argumentwert2,...)
```

Die wichtigste und elementarste Übergabe an `uitable` ist die Übergabe der darzustellenden Daten. Hierzu wird das Argument `'data'` verwendet und der Argumentwert (in unserem Fall sind dies Zahlenwerte) kann in Form von $m \times n$ -Matrizen übergeben werden.

Aufgabe 23 (Die erste Tabelle)

Erstellen Sie eine 9×3 -Matrix mit beliebigen Einträgen. Lassen Sie sich diese Matrix in einem `uitable` ausgeben.

Wie man in Aufgabe 23 erkennt, kann ein `uitable` relativ einfach erzeugt werden, doch besitzt die wiedergegebene Tabelle augenscheinlich eine dürftige Formatierung. Um dieser Sache Herr zu werden, gehen wir zunächst auf die Einstellung der Größe der Tabelle ein. Hierbei ist zu beachten, dass wir im Allgemeinen zwei verschiedene Größen betrachten – einmal die Größe des äußeren `figure`-Fensters und einmal die Größe der inneren Tabelle. Sowohl `uitable` als auch `figure` verwenden die für die Größe verantwortlichen Argumente `'position'` und `'unit'`.

Mit `'position'` wird die Position des jeweiligen Tochterobjekts innerhalb des jeweiligen Elternobjekts festgelegt. Für das Tochterobjekt `figure` ist das Elternobjekt der Computerbildschirm, für `uitable` ist das Elternobjekt das `figure`-Fenster. Das Eingabeargument von `'position'` besteht aus dem vierdimensionalen Vektor `[left bottom width height]`. Die Bedeutungen sind wie folgt:

<code>left</code>	Abstand zwischen linkem Rand des Elternobjekts und linkem Rand des Tochterobjekts
<code>bottom</code>	Abstand zwischen unterem Rand des Elternobjekts und unterem Rand des Tochterobjekts
<code>width</code>	Distanz zwischen der linken und rechten äußeren Kante des Objekts (Breite)
<code>height</code>	Distanz zwischen der oberen und der unteren äußeren Kante des Objekts (Höhe)

Mit `'unit'` wird die Maßeinheit festgelegt, in der die in `'position'` übermittelten Werte interpretiert werden sollen. Die Messung von Matlab erfolgt für alle diese Einheiten von der unteren linken Ecke des Elternobjekts bis zur oberen rechten Ecke. Für `'unit'` werden folgende Argumentwerte akzeptiert:

<code>'pixels'</code>	Die Größe eines Objekts in Pixeln ist abhängig von den Bildeinstellungen und der Auflösung (Standardeinstellung).
<code>'normalized'</code>	Diese Einstellung kartiert die untere linke Ecke des Elternobjekts mit (0,0) und die obere rechte Ecke mit (1,1). Dazwischen erfolgt eine gleichmäßige Einteilung.
<code>'inches'</code> , <code>'centimeters'</code> , <code>'points'</code>	Diese Größen sind die bekannten absoluten Maßeinheiten.
<code>'characters'</code>	Buchstaben-Units werden durch die Standardschrift des Computersystems beeinflusst.

Wichtig:

- Beachten Sie, dass 'unit' in der Reihenfolge der Argumente vor 'position' angegeben werden muss, da andernfalls eine Umstellung der Maßeinheit von Matlab zu spät erkannt wird.
- Falls Sie mehrere (initialisierte) figure-Fenster nutzen, können Sie für uitable über das Argument 'parent' das Fenster auswählen, in dem Sie Ihre Tabelle angezeigt haben möchten. Als Argumentwert verwenden Sie als String den Namen des entsprechenden figure-Fensters.

Aufgabe 24 (Die erste Tabelle – neu formatiert)

Wir möchten weiter mit einer 9×3 -Matrix arbeiten. Schreiben Sie ein Skript, in dem Sie eine solche Matrix erzeugen und in einem uitable darstellen. Erzeugen Sie diesmal jedoch vorab ein figure-Fenster und stellen Sie sowohl für dieses als auch uitable die Maßeinheit 'normalized' ein. Tragen Sie als Argumentwert für die 'position' des figure-Fensters den Vektor [0.3 0.3 0.4 0.4] ein und für die 'position' der Tabelle den Vektor [0.33 0.28 0.34 0.44].

Was passiert, wenn Sie die 'normalized'-Einstellung durch eine der anderen oben angegebenen Einstellungen ersetzen?

Wir können nun die Größe und Position einer Tabelle nach unseren Wünschen einstellen. Bei einer solchen Einstellung ist es oft sinnvoll, die verwendeten Abstände in Abhängigkeit von eigenen position-Variablen zu setzen, um bestimmte Relationen für die Eltern- und Tochterobjekte beizubehalten.

Möchte man eine „passende“ Größe für einen uitable schnell finden, so kann es leider oft zu nervigem Rumprobieren mit den Größenargumenten kommen. Damit dies nicht sein muss, berechnet uitable automatisch nach der Übergabe des Datensatzes einen Vektor, der die Breiten- (width) und Höhenwerte (height) enthält, um den Datensatz passend zu umrahmen. Dieser Vektor heißt extent und wird in der Maßeinheit berechnet, die vom Nutzer im entsprechenden uitable-Argument festgelegt wurde. extent ist (wie position) ein vierdimensionaler Vektor, von dem die ersten beiden Einträge jedoch Nulleinträge sind, so dass nur der dritte und vierte Eintrag relevant sind. Um auf extent zuzugreifen, muss der entsprechende uitable vorab mit einer Variable initialisiert werden, um handhabbar zu sein. Ein Beispiel zur Festlegung der neuen Größen sieht wie folgt aus und zeigt auch insbesondere die Verwendung des Punktoperators:

```
table = uitable('data', data,...)
table.position(3:4) = table.extent(3:4)
```

Ein wichtiges Feature von uitable ist die Einstellung von Zeilen- und Spaltenüberschriften. Die entsprechenden Argumente hierfür sind 'rowname' und 'columnname'. Als Argumentwerte sind nur Cell-Arrays zugelassen, die Strings enthalten. Cell-Arrays lassen sich mittels geschweifter Klammern {String1, String2, ...} erzeugen. Die Eingabe eines leeren Cell-Arrays {} hat zur Folge, dass die gesamte Tabelle keine Zeilen- oder Spaltenüberschriften enthält; die Eingabe einer leeren Matrix [] im Cell-Array bewirkt, dass die Tabelle an der entsprechenden Stelle keine Zeilen- oder Spaltenüberschrift enthält. Standardmäßig ist der Argumentwert 'numbered' eingestellt, der eine Nummerierung beginnend bei 1 bis zu der maximalen Zeilen- bzw. Spaltenanzahl vornimmt.

Die Breite einer Spalte wird grundsätzlich an die Breite der jeweiligen Spaltenüberschrift angepasst. Möchten Sie die Breite Ihrer Spalten manuell beeinflussen, so ist dies über das Argument 'columnwidth' möglich. Als Argumentwert ist abermals ein Cell-Array erforderlich. Diesmal ist jedoch keine Angabe im String-Format notwendig, sondern die Angabe von Zahlen, die im Pixel-Maß interpretiert werden. Möchten Sie, dass eine Spalte sich wieder an ihrer Standardbreite orientiert, so tragen Sie einfach den String 'auto' ein.

Aufgabe 25 (Berechnung von Normen)

Gegeben seien die Vektoren

$$\begin{aligned}x_1 &= (2, 4, -5, 4, 8)^T, \\x_2 &= (4, 1, 9, 7, -2)^T, \\x_3 &= (3, 5, -9, 1, 0)^T.\end{aligned}$$

Erstellen Sie eine Tabelle, in der Sie für die Vektoren $x_i, i \in \{1,2,3\}$, die 1-Norm, die 2-Norm und die ∞ -Norm mittels der Befehle $\text{norm}(x,1)$, $\text{norm}(x,2)$ und $\text{norm}(x,\text{inf})$ berechnen. Achten Sie auf eine angemessene Beschriftung der Zeilen und Spalten und benutzen Sie zur passenden Umrahmung Ihrer Tabelle den extent-Vektor.

Möchten Sie einen Beschreibungstext zu Ihrer Tabelle hinzufügen, so funktioniert dies mit der Umgebung uicontrol. uicontrol bietet gleich zu uitable eine ganze Bandbreite an Interaktionsmöglichkeiten an, doch möchten wir uns auf den Gebrauch von uicontrol als statische Textbox beschränken. uicontrol ist syntaktisch identisch zu uitable und die Verwendung der Argumente 'parent', 'position', 'unit' und 'extent' ist übertragbar.

Möchte man eine statische Textbox erzeugen, so sieht ein beispielhafter Aufruf wie folgt aus:

```
uicontrol('style', 'text', 'string', 'Hallo Welt')
```

In diesem Beispiel lernen wir zwei neue Argumente kennen – 'style' und 'string'. Durch 'style' wird die Verwendung von `uicontrol` festgelegt. In unserem Fall haben wir 'text' ausgewählt, da damit eine statische Textbox erzeugt wird. Durch das darauffolgende Argument 'string' wird kenntlich gemacht, dass der nächste Argumentwert der darzustellende String ist (hier: 'Hallo Welt').

Man kann in Matlab Strings durch die Funktion `sprintf` auch separat abspeichern. Durch diese Funktion ist auch das Einpflegen von Variablen in einen String möglich. `sprintf` hat folgende Syntax:

```
str = sprintf('ausgabertext', A1, ..., An)
```

Hier erkennt man, dass `sprintf` einmal den String 'ausgabertext' fordert, der den abzuspeichernden Text enthält, sowie die Variablen `A1, ..., An`, auf die sich in 'ausgabertext' bezogen wird. Die Variablen sind in der Reihenfolge aufzulisten, in der Sie auftreten sollen. Ein Beispiel:

```
A = 1/eps;  
str_e = sprintf('Exponentielle Darstellung: %0.5e', A)  
str_f = sprintf('Gleitkommadarstellung: %0.5f', A)  
str_g = sprintf('Exp. Darstellung ohne angehangene Nullen: %0.5g', A)
```

Die Ausgabe der Strings schaut wie folgt aus:

```
str_e = Exponentielle Darstellung: 4.50360e+15  
str_f = Gleitkommadarstellung: 4503599627370496.00000  
str_g = Exp. Darstellung ohne angehangene Nullen: 4.5036e+15
```

In allen drei Darstellungen wurde die Anzahl der Nachkommastellen auf 5 festgelegt. Innerhalb des Strings 'ausgabertext' kann man mit `\n` eine neue Zeile anlegen. Durch `\t` kann man einen horizontalen Tab erzeugen, durch `\v` einen vertikalen Tab. `sprintf` bietet noch eine Vielzahl an weiteren Einstellungs- und Darstellungsmöglichkeiten mit denen wir uns jetzt aber nicht beschäftigen möchten. Sie können in der entsprechenden Matlab-Dokumentation eingesehen werden.

Es sei hier auch insbesondere die Funktion `fprintf` erwähnt, die exakt wie `sprintf` verwendet wird, aber den erzeugten String direkt im Command Window ausgibt.

Aufgabe 26 (Textboxen mit `uitable` und `sprintf`)

Schreiben Sie eine Funktion `table_func(m,n)`, in der Sie eine Tabelle mit `m` Zeilen und `n` Spalten sowie zufälligen Einträgen anlegen. Fügen Sie oberhalb der Tabelle mittels `uicontrol` eine Textbox hinzu, die angeben soll, wie viele Zeilen und Spalten die Tabelle enthält. Probieren Sie, mithilfe der `extent`-Vektoren von `uicontrol` und `uitable` passende Größen für Ihre Ausgaben festzulegen.