



Einführung in Matlab

2. Der Editor und Grafikplots

Prof. Dr. Christiane Zarfl, Dipl.-Inf. Willi Kappler

- wie Sie in Matlab Variablen definieren und damit Werte wiederverwenden können.
- wie Sie in Matlab mit Vektoren rechnen und damit effizient auch große Datenmengen verarbeiten können.

Wie kann man mehrere Rechenschritte/Matlab-Befehle, die häufig benötigt werden, "speichern" und zusammenfassen?

Nach diesem zweiten Block...

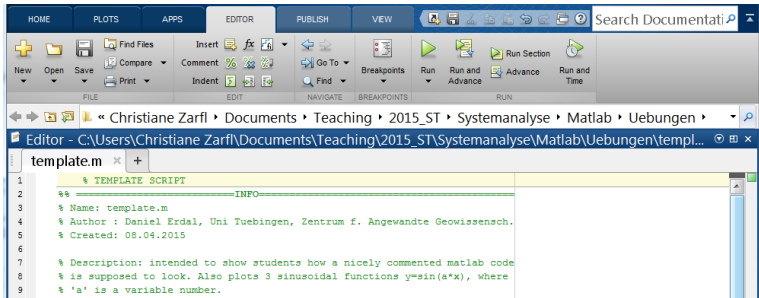
- können Sie eigene Script Files erstellen (Bsp. 1D-Stofftransportgleichung)
- können Sie x-y Plots erstellen, bearbeiten und speichern.



- Matlab ist mehr als nur ein Taschenrechner.
- Ein Script-File ist eine Aneinanderreihung von Matlab-Befehlen.
- Matlab-Scripts haben die Dateiendung “.m”.
- Bsp.: Ihr Programm heißt `MeinScript.m` und befindet sich im Verzeichnis `MyDocuments\ich\Matlab`.
- Dieser Pfad muss das gegenwärtige **Arbeitsverzeichnis** von Matlab sein.
- Dann führt der Befehl `MeinScript` (ohne Dateiendung) im “Command Window” den Inhalt des Scriptes aus.
- Dateinamen dürfen keine Sonderzeichen oder Leerzeichen enthalten und auch nicht mit Ziffern anfangen.
- Script Files/Programme können mit jedem Texteditor oder mit dem Matlab-eigenen Editor erstellt werden.

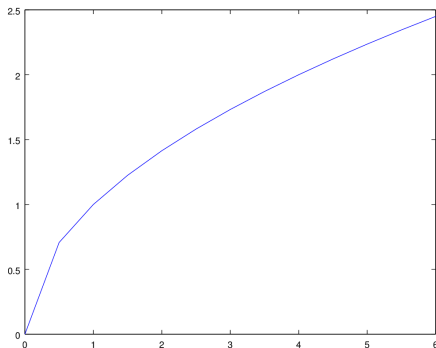


- Legen Sie ein Verzeichnis für die heutige Übung an.
- Machen Sie dies zum aktiven Verzeichnis von Matlab.
- Aufruf des Editors vom prompt: `edit dateiname`.
- **Achtung:** Kommentieren Sie **immer** Ihre Scripte: Kommentare beginnen mit `%` und enden mit dem Zeilenende.





- `plot` trägt numerische Daten gegeneinander auf (`doc plot`).
- `plot(x,y)` erzeugt ein Grafikfenster ("figure") indem die Werte y gegen x aufgetragen sind.
- Beispiel: die Wurzelfunktion im Intervall $[0,6]$ zeichnen:
- `x=0:0.5:6; y=sqrt(x); plot(x,y)`





Schreiben Sie ein Script (`Wurzel.m`), welches die auf der letzten Folie gezeigte Grafik erzeugt.

Tipp: Der erste Befehl eines Scripts sollte `clear all` sein, um den Arbeitsspeicher zu leeren

Ändern Sie die Auflösung der x-Achse.



Die eindimensionale Stoffverteilung bei pulsartiger Zugabe des Stoffes hat die analytische Lösung:

$$c(t, x) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(-\frac{(x - vt)^2}{4Dt}\right)$$

Schreiben Sie ein Script, welches die räumliche Stoffverteilung graphisch darstellt. (Fortsetzung nächste Folie)



Tips:

- Löschen sie zuerst den Arbeitsspeicher (`clear all`)
- Definieren sie die Parameter:
 - v : Geschwindigkeit ($10^{-5} m/s$)
 - D : Diffusionskoeffizient ($10^{-7} m^2/s$)
 - t : Zeit ($1 d = 86400 s$)
 - x : Ortsvektor (0-2 m in 1cm-Schritten)
- Berechnung der Konzentration (c , siehe Formel oben)
- Plotten des Profils (`plot(x, c)`)

Erinnerung: Bitte kommentieren Sie jede Zeile! Kommentare beginnen mit “`%`” und enden mit dem Zeilenende.



- Stilparameter in Hochkommata übergeben: `plot(x,c,'k:x');`
 - `k` \Rightarrow schwarz; `:` \Rightarrow gepunktete Linie; `x` \Rightarrow x-Zeichen als Marker.
 - Informieren Sie sich mit `doc plot` über weitere Stilparameter.
- Achsenbereiche festsetzen:
`xlim([minx maxx]); ylim([miny maxy])`
- Achsenbeschriftung: `xlabel('x [m]');` und `ylabel('c [mol/L]');`.
- Titel hinzufügen: `title('Titeltext');`
- Legende hinzufügen: `legend('c');`
- Matlab kennt rudimentäre LaTeX-Syntax für Strings:
`xlabel('c [mol/m^3]')` ergibt hoch gestellte “3”,
`title('c_{cons}')` ergibt tief gestelltes “cons”



- Variante 1: mehrere Parameterpärchen übergeben:
`plot(t1,c1,'r-',t2,c2,'b:')`
- Variante 2: mit `hold on` in Hinzufügmodus wechseln (und mit `hold off` wieder raus):
 - `plot(t1,c1,'r-');`
 - `hold on;plot(t2,c2,'b:');hold off`
- Ein neues Grafikfenster wird mit dem Befehl `figure` erzeugt:
 - `figure(1)`, erzeugt das Grafikfenster 1
 - Man kann somit zwischen den Grafikfenstern “springen”.



- Verschönern Sie Ihren Plot von vorhin:
- Zeichnen Sie die Konzentration in rot!
- Beschriften Sie die Achsen!
- Geben Sie dem Plot einen Titel!
- Fügen sie eine Legende hinzu!
- Plotten Sie mehrere Konzentrationsprofile in Ihrer Grafik, für $t=1, 2, 3$ Tage
- ... vergessen Sie das **Kommentieren** nicht!



- Konzentrationsverteilung bei stufenartiger Anfangsverteilung:

$$c(x, t) = \frac{c_{ini}}{2} \operatorname{erfc} \left(\frac{x - vt}{\sqrt{4Dt}} \right)$$

- `erfc` ist die komplementäre Fehlerfunktion (in etwa das Integral der Gauß-Funktion)
- Vermutung: könnte in Matlab `erfc` heißen (Testen Sie: `help erfc`)



- Plotten Sie mehrere Konzentrationsprofile bei stufenartiger Anfangsverteilung:
- v : Geschwindigkeit ($10^{-5} m/s$)
- D : Dispersionskoeffizient ($10^{-7} m^2/s$)
- c_{ini} : Anfangskonzentration (1 mg/L)
- t : Zeit (0, 1, 2, 3 Tage)
- x : Ortsvektor (-2 bis 5 m in 1cm-Schritten)
- Speichern Sie Ihr Script und kommentieren Sie es!

- Problem: Sie brauchen für einen Bericht einen guten Plot Ihrer Daten, und zwar genau 9cm breit, 6 cm hoch
- Schon mal mit Excel probiert?
- Matlab-Lösung: Abbildung in Grafikdatei “drucken” (siehe `doc print`)
- `print -djpeg100 -r300 my_plot.jpg`
- `djpeg100`: Format und Qualität
- `r300`: Auflösung (DPI)
- `my_plot.jpg`: Dateiname



- Die Größe der Grafik ist eine Eigenschaft der Abbildung:
- `set(gcf, 'paperunits', 'centimeters', 'paperposition', [1 1 9 6])`
- `set`: verändere eine Eigenschaft
- `gcf`: “get current figure” = handle auf die aktuelle Abbildung
- Eigenschaft “paperposition”: Koordinaten der linken unteren Ecke auf dem Papier, Breite, Höhe
- Abfrage einer Eigenschaft: `get(gcf, 'paperposition')`



- `gca`: “get current axis” = handle auf die aktuelle Achse
- Einige Achseneigenschaften:
 - `'linewidth'`
 - `'fontsize'`
 - `'xgrid', 'ygrid'` (Wert `'on'` oder `'off'`)
 - `'xscale', 'yscale'` (Wert `'linear'` oder `'log'`)
 - `'position'` (Position innerhalb der Abbildung)
- `gco`: “get current object” = handle auf das aktuelle Objekt, z.B. die Linie der Grafik



- `subplot(2,2,1)` : unterteile Abbildung in 2×2 Unterabbildungen
nehme die erste als aktive Abbildung
- `subplot(2,2,2)` : nehme die zweite Unterabbildung als aktive
Abbildung
- Matlab erzeugt dann mehrere Achsen, die jeweils “Kinder” der
Abbildung sind



- **Hausaufgaben zur Vertiefung:**

- Sie können verschiedene Rechnungen mit Matlab ausführen und kennen die Funktionsweise der in Matlab implementierten Funktionen.
- Sie können Funktionen in 2D-Grafiken darstellen.

- **Nächste Kurseinheit - noch eine Dimension:**

- Rechnen mit Matrizen
- verschiedene 3D-Darstellungen