Technische Universität München
M2 - Numerische Mathematik
Dr. Laura Scarabosio

1.1. Übung. Einführung in die Programmierung (MA 8003)

Um Matlab zu starten, wählen Sie dies aus dem Menü

`Anwendungen` → `Wissenschaft` → `Matlab`

oder geben Sie in einem Terminal den Befehl `matlab` ein.

Hinweis: Alle Aufgaben auf diesem Blatt können und sollen **ohne** Schleifen gelöst werden!

Aufgabe 1.1.1: Starten Sie Matlab und richten Sie die Oberfläche so ein, dass auch der *Workspace*, sichtbar ist. Legen Sie ein paar Variablen an und löschen Sie mindestens eine wieder aus dem Workspace. Schließen Sie Matlab.

Wiederholen Sie die vorhergehende Teilaufgabe, aber diesmal sollen die angelegten Variablen bei einem erneuten Programmstart wieder verfügbar gemacht werden können. **Tipp:** `help save`

Was macht der Befehl `whos`?

Aufgabe 1.1.2: Welche Ausgabe erwarten Sie bei der Eingabe `b=5, 2*b;`? Stimmt das Ergebnis mit Ihrer Erwartung überein?

Aufgabe 1.1.3: Legen Sie die Variablen $a = 3$, $b = 5$ und $c = -3$ an und verwenden Sie Matlab um folgenden Ausdruck zu berechnen:

$$b - \frac{a}{b + \frac{b+a}{ca}}.$$

Lösung 1.1.3:

```
>> a=3; b=5; c=-3; b-(a/(b+(b+a)/(c*a)))

ans =
    4.2703
```

Aufgabe 1.1.4: Rufen Sie die Hilfe zum Befehl `floor` im Kommandofenster mit `help` auf. Machen Sie das Gleiche mit dem Befehl `ceil`.

Rufen Sie die Hilfe zum Befehl `fix` im Kommandofenster mit `doc` auf.

Geben Sie ein `x` an, bei dem sich die Ausgaben von `floor(x)` und `fix(x)` unterscheiden.

Geben Sie ein `x` an, bei dem sich die Ausgaben von `ceil(x)` und `fix(x)` unterscheiden.

Lösung 1.1.4: z.B. für `x = -1.2` bzw. `x = 1.2`.

Aufgabe 1.1.5: Berechnen Sie $\cos(\pi/5)$, $\sin(58°)$ und $\tan^{-1}(1)$ mit Matlab.
Tipp: Verwenden Sie die Matlabkonstante `pi` und ggf. die Hilfe.

Lösung 1.1.5: `cos(pi/5)`, `sin(58*pi/180)` oder `sind(58)`, `atan(1)`

Aufgabe 1.1.6: Welche Zahl erzeugt die Matlab-Eingabe `9.5e8`? Geben Sie 0.0034 in dieser Schreibweise an.

Lösung 1.1.6: `3.4e-3`

Aufgabe 1.1.7: Legen Sie die Variable `x=sqrt(2)` an. Geben Sie den Befehl `format long` ein, und lassen Sie sich die Variable erneut ausgeben. Was bemerken Sie?

Legen Sie jetzt die Variable `y=pi` an. Nachdem Sie den Befehl `format short` eingegeben haben, wie wird `y` angezeigt?

Benutzen Sie `help format` um mehrere Formate kennenzulernen.

Aufgabe 1.1.8: Erzeugen Sie die komplexe Zahl $z = 3 - i$ auf mindestens drei verschiedene Arten. Stellen Sie die Zahl in Polarkoordinaten $(r, \varphi) \in \mathbb{R}^+ \times [-\pi, \pi)$ dar. Nutzen Sie dazu den Befehl `angle` und ggf. die Hilfe.

Lösung 1.1.8: Beispiele: `3-i, 3-j, complex(3,-1), 3-sqrt(-1)`
`r = abs(3-i), phi = angle(3-i)`

Aufgabe 1.1.9: Erzeugen Sie eine Eins-, Null- und Einheitsmatrix der Dimension 10×5.

Erzeugen Sie eine zufällige 5×4 Matrix `M`, deren Einträge gleichverteilt sind im Intervall $(-0.5, 1.5)$.

Lösung 1.1.9: `ones(10,5), zeros(10,5), eye(10,5)`
`-0.5 + 2*rand(5,4)`

Aufgabe 1.1.10: Benutzen Sie jeweils die Doppelpunkt-Notation sowie den Befehl `linspace`, um folgende Sequenzen zu erzeugen:

a) Alle geraden Zahlen zwischen 2 und 1024,

b) alle ungeraden Zahlen zwischen 1023 und 1,

c) alle Werte zwischen -4 und 4 in 0.25-er Schritten.

Lösung 1.1.10:

a) `2:2:1024`, `linspace(2, 1024, 512)`

b) `1023:-2:1`, `linspace(1023, 1, 512)`

c) `-4:0.25:4`, `linspace(-4, 4, 33)`

Aufgabe 1.1.11: Ersetzen Sie in den folgenden Ausdrücken `?` durch eine Einsmatrix geeigneter Dimension so, dass `C` wohldefiniert ist.

a) `C = [zeros(3,1), eye(3); ?, diag([2 3])]`

b) `C = [diag([5],-2), ?; [1,2; ?], repmat((1:2)',2,2)]`

Lösung 1.1.11:

a) `?` = `ones(2)`

b) `?` = `ones(3,1)`, `?` = `ones(3,2)`

Aufgabe 1.1.12 (*): Zeichnen Sie ein Schachbrett-Muster (abwechselnd 0 und 1) in eine 8×8-Matrix. **Tipp:** `repmat`.

Lösung 1.1.12: `repmat(eye(2),4,4)`

Aufgabe 1.1.13: Erzeugen Sie einen Vektor `y`, in dem die Werte von $\frac{1}{x}\sin^2(x)$ für alle `x` von 1 bis 2 im Abstand 0.1 stehen.

Lösung 1.1.13: `x=1:0.1:2; y=1./x .* sin(x).^2;`

Aufgabe 1.1.14: Das Polynom $x - x^2 + xy + y^2$ soll für verschiedene (x, y) Paare ausgewertet werden (`x=1:2:6`, `y=[3; -2; 4]`). Liefert der folgende Ausdruck das richtige Ergebnis?

```
(x - x.^2)' + x*y + y.^2
```

Lösung 1.1.14: Nein, richtig ist `(x - x.^2)' + x'.*y + y.^2`.

Aufgabe 1.1.15: Erstellen Sie eine 4×4 Matrix A mit Einträgen > 5 aber < 10 und einen Vektor $v \in \mathbb{R}^4$ jeweils mit Zufallswerten.

Bilden Sie Av und $A^T v$ und speichern Sie das Ergebnis in `a` bzw. `b`. Bilden Sie das Standard-Skalarprodukt von `a` und `b`.

Lösung 1.1.15: `A = 5 + 5*rand(4); v = rand(4,1); a = A*v, b = A'*v, a'*b`

Aufgabe 1.1.16 (*): Gegeben sei ein Vektor `x` der Länge 100 mit Zufallswerten.

a) Berechnen Sie die euklidische Norm des Vektors ohne die Funktion `norm` zu verwenden.

b) Berechnen Sie die Summe der Einträge ohne die Funktion `sum` zu verwenden.

Vergleichen Sie Ihr Ergebnis mit der Ausgabe von `norm(x)` bzw. `sum(x)`.

Lösung 1.1.16:

a) `x=rand(100, 1); nrm = sqrt(x'*x)`

b) `x=rand(100, 1); s = ones(1,100)*x`

Aufgabe 1.1.17 (*): Die Funktion $w(x) = a_1 \sin(x) + a_2 \cos(x) + a_3 x^3$ soll für verschiedene Parameter $a \in \mathbb{R}^3$ im Intervall $[0, 2\pi]$ an sieben äquidistanten Punkten ausgewertet werden (inklusive Intervallenden). Verwenden Sie das Matrix-Vektor Produkt wie im Polynombeispiel auf den Folien, um die Funktion für die Parameter $a^1 = (1, 2, 1)^T$ und $a^2 = (-1, 2, 0.04)^T$ auszuwerten. **Tipp:** Welche Ausgabe erzeugt die Funktion `sin`, wenn die Eingabe ein Vektor ist?

Lösung 1.1.17:

```
>> x=linspace(0,2*pi,7)'; X=[sin(x), cos(x), x.^3];
>> X*[1; 2; 1]

ans =
    2.0000
    3.0144
    9.0531
   29.0063
   71.6303
  143.6816
  250.0502

>> X*[-1; 2; 0.04]

ans =
    2.0000
    0.1799
   -1.4985
   -0.7597
    2.8059
    7.6079
   11.9220
```

Oder:

```
>> x = linspace(0,2*pi,7);
>> A = [1 2 1; -1 2 0.04];
>> A * [sin(x); cos(x); x.^3]

ans =
    2.0000    3.0144    9.0531   29.0063   71.6303  143.6816  250.0502
    2.0000    0.1799   -1.4985   -0.7597    2.8059    7.6079   11.9220
```

Aufgabe 1.1.18 (*): Nehmen Sie $N = 10$ und $M = 100$, und erstellen Sie eine $N \times M$ Zufallsmatrix `D`. Stellen Sie sich die Matrix `D`, als M Messungen von N-dimensionalen Daten vor. Zentrieren Sie die N Messungen um den Ursprung, das heisst, subtrahieren Sie von jeder Spalte von D den N-dimensionalen Durchschnittswert.
Tipp: Benutzen Sie den Befehl `repmat`.

Lösung 1.1.18:

```
>> N=10; M=100;
>> D=rand(N,M);
>> mu=sum(D,2)/M; % durchschnitt. Alternative: Befehl 'mean'.
>> D=D-repmat(mu,1,M);
```

Aufgabe 1.1.19 (*): Die zwei Lösungen der Gleichung $ax^2 + bx + c = 0$ können mit der Formel

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1}$$

berechnet werden. Benutzen Sie diese Formel um die Nullstellen für $a = c = 1$ und $b = -10^9$ zu erhalten.

Vergleichen Sie das Ergebnis mit dem Output von `roots([a b c])`. Was beobachten Sie? Versuchen Sie, eine Grund herauszufinden.

Wenn Sie schon $x_1 \neq 0$ haben (z.B. von (1)), wie kann x_2 alternativ berechnet werden? **Tipp**: $x_1 x_2 = \frac{c}{a}$.

Lösung 1.1.19: Verwendet man die Formel (1), so ergibt sich in MATLAB die folgende Ausgabe:

```
>> a = 1.0;
>> b = -1.0e9;
>> c = 1.0;
>>
>> x_1 = (-b+sqrt(b^2-4*a*c))/2/a

x_1 =

1.0000e+09

>> x_2 = (-b-sqrt(b^2-4*a*c))/2/a

x_2 = 0
```

Es ist offensichtlich, dass $x_2 = 0$ keine Nullstelle des Polynoms ist. Man rechnet auch nach, dass $x_1 = 1.0 \cdot 10^9$ keine Nullstelle des Polynoms ist:

$$1.0 \cdot \left(1.0 \cdot 10^9\right)^2 - 1.0 \cdot 10^9 \cdot 1.0 \cdot 10^9 + 1.0 = 1.0 \neq 0.$$

Der Grund für die falschen Nullstellen liegt darin begründet, dass der Ausdruck

$$\sqrt{b^2 - 4ac}$$

auf $1.0 \cdot 10^9$ gerundet wird, denn auf einem Rechner können nicht alle reelle Zahlen genau dargestellt werden. Damit ist die Rechnerarithmetik nicht exakt. Verwendet man den Befehl

`roots([a b c])`,

so ergibt sich:

```
>> format longg
>> a = 1.0;
>> b = -1.0e9;
>> c = 1.0;
>> x = roots([a b c])
```

```
x =

1000000000
1e-09
```

Die Nullstelle x_1 ist identisch mit der Formel, die wir aus (1) erhalten, während die zweite Nullstelle $x_2 = 1.0 \cdot 10^{-9}$ beträgt. Mit dem Satz von Vieta:

$$x_1 + x_2 = 1.0 \cdot 10^9,\ x_1 \cdot x_2 = 1.0$$

sieht man aber sofort, dass auch diese Nullstellen nicht exakt sind. Zumindest die Nullstelle x_2 ist numerisch gleich null, da man Zahlen kleiner als $1.0 \cdot 10^{-16}$ numerisch als gleich Nnull ansieht:

$$1.0 \cdot \left(1.0 \cdot 10^{-9}\right)^2 - 1.0 \cdot 10^9 \cdot 1.0 \cdot 10^{-9} + 1.0 = 1.0 \cdot 10^{-18} \approx 0.$$

Es zeigt sich also, dass der Algorithmus hinter `roots`, der auf einer Berechnung der Eigenwerte der Begleitmatrix beruht, eine zweite Nullstelle berechnet, die zumindest numerisch gleich Null ist. Allerdings bleibt festzuhalten, dass auch in diesem Fall die erste Nullstelle ungenau berechnet wird. Betrachtet man den Verlauf des Polynoms (siehe Abbildung 1), man beachte die Skalierung der Achsen) in der Umgebung von $x = 1.0 \cdot 10^9$, so stellt man fest, dass er so steil verläuft, dass der Verlauf numerisch nicht exakt dargestellt wird.

x_2 kann mit

$$x_2 = \frac{c}{ax_1} \text{ erhalten werden.}$$

Die Formel (1) ist nicht gut, um die Nullstellen mit dem Rechnerarithmetik zu berechnen. Der Grund ist, dass auf einem Rechner nicht alle reelle Zahlen genauer dargestellt werden können, und die Rechnerarithmetik nicht exakt ist.

x_2 kann mit $x_2 = \frac{c}{ax_1}$ erhalten werden.

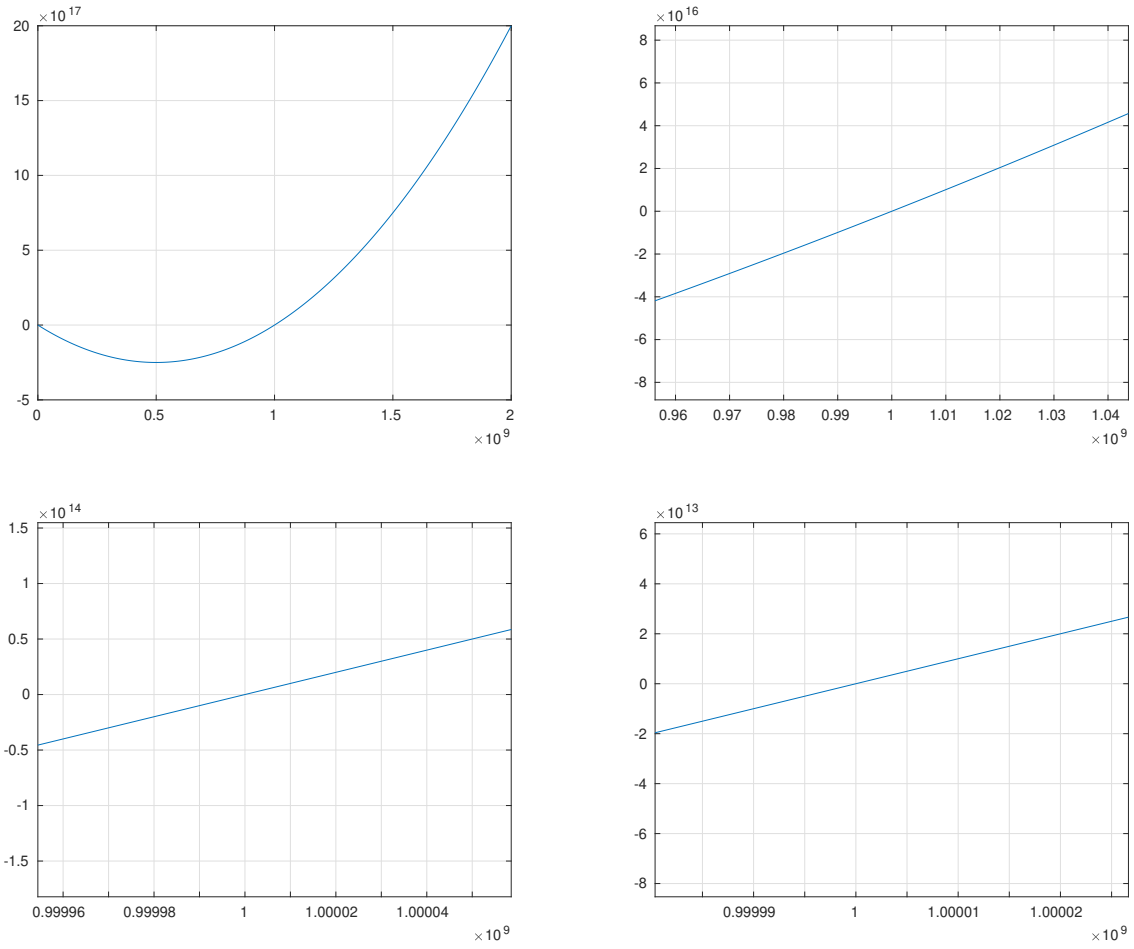

Abbildung 1: Vergrößerung des Polynomverlaufs um $1.0 \cdot 10^9$.