Aufgabe 25.3

Lösen Sie die folgenden Aufgaben mit `MATLAB`. Protokollieren Sie Ihr Vorgehen in einer `diary`-Datei und speichern Sie erstellte Plots ab.

1. Es sei
$$A = \begin{pmatrix} 1 & 2 \\ 3 & -4 \end{pmatrix}, B = \begin{pmatrix} 0 & -2 \\ 2 & 1 \end{pmatrix}, \vec{w} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \text{ und } \alpha = 3.$$
   a) Berechnen Sie $A+B$, $A-B$, αA, $A\vec{w}$, $AB\vec{w}$, $\vec{w}^\top$, $A^\top$, $B^\top$, $A^\top\vec{w}$ und $\left(A^\top\right)^\top$.
   b) Berechnen Sie AB, BA, $(A+B)^\top$, $A^\top+B^\top$, $(\alpha A)^\top$, $\alpha A^\top$, $(AB)^\top$, $B^\top A^\top$ sowie $A^\top B^\top$. Überzeugen Sie sich für dieses Beispiel von der Nichtkommutativiät der Multiplikation und den Rechenregeln für das Transponieren.
2. Lösen Sie Aufgabe 6.64 mit Hilfe von `MATLAB`. Geben Sie dabei alle Ausdrücke von a) bis h) ein.
3. Es sei
$$C = \begin{pmatrix} 3 & -1 & 8 \\ -4 & 3 & 9 \\ 5 & -2 & -6 \end{pmatrix}, \vec{p} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ und } P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$
   Berechnen Sie EC, CE, $E\vec{p}$, $C\vec{p}$, $\vec{p}^\top C$, CP und PC. (Überlegen Sie sich vor der Befehlseingabe, welche Ergebnisse zu erwarten sind.)
4. Weiter sei
$$\vec{s} = \begin{pmatrix} 15 \\ 8 \end{pmatrix}, \vec{t} = \begin{pmatrix} -8 \\ 16 \end{pmatrix}.$$
   a) Berechnen Sie $\|\vec{s}\|$, $\|\vec{t}\|$, $\|\vec{s}+\vec{t}\|$, $\|\vec{s}\|+\|\vec{t}\|$. Überzeugen Sie sich davon, dass die Dreiecksungleichung erfüllt ist.
   b) Berechnen Sie den Winkel Zwischen $\vec{s}$ und $\vec{t}$. Geben Sie den Winkel sowohl in Bogenmaß als auch in Grad an.
   Hinweis: Die `arccos`-Funktion heißt in `MATLAB` `acos`.
5. Es sei
$$\vec{u} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \vec{v} = \begin{pmatrix} -2 \\ 3 \\ 0 \end{pmatrix}.$$
   Stellen Sie die lineare Hülle grafisch dar. Plotten Sie dazu die neun Linearkombinationen $\alpha\vec{u}+\beta\vec{v}$ für $\alpha,\beta \in \{-1, 0, 1\}$ in einen Plot. Drehen Sie die Ansicht so, dass zu erkennen ist, dass alle Vektoren in einer Ebene liegen. Geben Sie dem Plot einen Titel, der Ihren Namen enthält.

Öffnen Sie die erstellte `diary`-Datei (vorher mit $\gg$ `diary off` die Protokollierung abschließen) und entfernen Sie ggf. überflüssige Zeilen (z.B. Fehleingaben). Drucken Sie anschließend die bearbeitete `diary`-Datei und eventuell angefertigte Plots möglichst sparsam (d.h. nach Möglichkeit duplex, mehrere Seiten pro Blatt, kleine Schriftgröße) aus.

Hinweise zur MATLABaufgabe

Matrizen und Vektoren

In MATLAB wird nicht zwischen einem Vektor und einer Matrix unterschieden. Spalten-/Zeilenvektoren sind Matrizen mit nur einer Spalte/Zeile. Matrizen werden in eckigen Klammern zeilenweise eingegeben. Hierbei wird das Zeilenende mit einem Semikolon angezeigt. Die Einträge in einer Zeile können mit einem Komma abgetrennt werden. Der folgende Befehl erzeugt Matrizen *A* und *B*.

```
≫ A=[2 1 3; -1 0 1]
≫ B=[1, -1, 2; -4, 2, 0; 1, 0, 3]
```

Analog ergibt sich die Eingabe eines Spaltenvektors als Eingabe einer Matrix mit nur einer Spalte. Beispiel:

```
≫ v=[-3; 4; 7]
≫ w=[-2; -3; 1]
```

Vektoren können genutzt werden, um daraus Matrizen zu erzeugen. Beispiel:

```
≫ C=[v w [1; -1; 4]]
```

Ein Zugriff auf einen Eintrag geschieht durch die Angabe der Zeilen- und Spaltennummer. Beispiel:

```
≫ A(2,3)
≫ v(3)
≫ v(3)=5
≫ v
```

Um auf eine ganze Zeile/Spalte zugreifen, verwendet man einen Doppelpunkt. Beispiel:

```
≫ A(:,3)
≫ A(1,:)
```

Die Einheitsmatrix der Größe 3×3 und die Nullmatrix der Größe 2×3 lassen sich wie folgt erzeugen.

```
≫ E=eye(3)
≫ Z=zeros(2,3)
```

Das Rechnen mit Matrizen funktioniert mit den „natürlichen“ Rechenzeichen. Beispiel:

```
≫ B+C
≫ 2*A
≫ B*v
≫ A*B
```

Das Transponieren einer Matrix wird mit einem Hochkomma realisiert. Beispiel:

```
≫ A'
≫ B'
≫ v'
≫ B'*A'
```

Die Norm (Länge) eines Vektors kann mit der Funktion `norm` berechnet werden. Beispiel:

```
≫ norm(v)
```

Um einen 2- bzw. 3-dimensionalen Vektor darzustellen, kann man beispielsweise die Verbindungslinie vom Koordinatenursprung $\vec{0}$ nach $\vec{v}$ mit dem Befehl `plot` bzw. `plot3` zeichnen. Der Befehl

```
≫ plot3( [0, v(1)] , [0, v(2)] , [0, v(3)] )
```

bildet den Vektor `v` ab. Mit `≫ grid on` kann man die Darstellung verbessern.

Lösung:

```
% ------------------------------------------------------------------------
% Aufgabe 1
% ------------------------------------------------------------------------

A=[1 2; 3 -4]
A =
     1     2
     3    -4

B=[0 -2; 2 1]
B =
     0    -2
     2     1

w=[1; 2]
w =
     1
     2

alpha=3
alpha =
     3

% a) ------------------------------------------------------------------------

A+B
ans =
     1     0
     5    -3

A-B
ans =
     1     4
     1    -5

alpha*A
ans =
     3     6
     9   -12

A*w
ans =
     5
    -5

A*B*w
ans =
     4
   -28

w'
ans =
     1     2

A'
ans =
     1     3
     2    -4

B'
ans =
     0     2
    -2     1

A'*w
ans =
     7
    -6

(A')'
ans =
     1     2
     3    -4

% b) ------------------------------------------------------------------------

A_mal_B=A*B
A_mal_B =
     4     0
    -8   -10

B_mal_A=B*A
B_mal_A =
    -6     8
     5     0

fprintf('\nFaktoren '), if A_mal_B==B_mal_A, fprintf('vertauschbar\n\n'),
                          else fprintf('nicht vertauschbar\n\n'),
                         end

Faktoren nicht vertauschbar

A_plus_B__transp=(A+B)'
A_plus_B__transp =
     1     5
     0    -3

A_transp_plus_B_transp=A'+B'
A_transp_plus_B_transp =
     1     5
     0    -3

fprintf('\nAddition und Transposition '),
if A_plus_B__transp==A_transp_plus_B_transp, fprintf('vertauschbar\n\n'),
 else fprintf('nicht vertauschbar\n\n'),
end

Addition und Transposition vertauschbar

alpha_mal_A__transp=(alpha*A)'
alpha_mal_A__transp =
     3     9
     6   -12

alpha_mal_A_transp=alpha*A'
alpha_mal_A_transp =
     3     9
     6   -12
```

```
fprintf('\nMultiplikation mit Skalar und Transposition '),
if alpha_mal_A__transp==alpha_mal_A_transp, fprintf('vertauschbar\n\n'),
 else fprintf('nicht vertauschbar\n\n'),
end

Multiplikation mit Skalar und Transposition vertauschbar

A_mal_B__transp=(A*B)'
A_mal_B__transp =
     4    -8
     0   -10

B_transp_mal_A_transp=B'*A'
B_transp_mal_A_transp =
     4    -8
     0   -10

A_transp_mal_B_transp=A'*B'
A_transp_mal_B_transp =
    -6     5
     8     0

fprintf('\nMatrizenmultiplikation und Transposition '),
if A_mal_B__transp==A_transp_mal_B_transp, fprintf('vertauschbar\n\n'),
 else fprintf('nicht vertauschbar\n\n'),
end

Matrizenmultiplikation und Transposition nicht vertauschbar

fprintf('\nMatrizenmultiplikation und Transposition '),
if A_mal_B__transp==B_transp_mal_A_transp, fprintf('vertauschbar'),
 else fprintf('nicht vertauschbar'),
end, fprintf(', wenn auch Faktoren vertauscht werden\n\n')

Matrizenmultiplikation und Transposition vertauschbar, wenn auch Faktoren
vertauscht werden
```

```
% ------------------------------------------------------------------------
% Aufgabe 2
% ------------------------------------------------------------------------

A=[3 3 -1; 2 0 1]
A =
     3     3    -1
     2     0     1

x=[1; -1; 1]
x =
     1
    -1
     1

y=[-2; 3]
y =
    -2
     3

A*x+y
ans =
    -3
     6

y'*A+x
??? Error using ==> plus
Matrix dimensions must agree.

y*A*x
??? Error using ==> mtimes
Inner matrix dimensions must agree.

y'*A*x
ans =
    11

x'*A*y
??? Error using ==> mtimes
Inner matrix dimensions must agree.

x'*(y'*A)'
ans =
    11

A*x*y'
ans =
     2    -3
    -6     9

A'*y*x'
ans =
     0     0     0
    -6     6    -6
     5    -5     5
```

```
% ------------------------------------------------------------------------
% Aufgabe 3
% ------------------------------------------------------------------------
C=[3 -1 8; -4 3 9; 5 -2 -6]
C =
     3    -1     8
    -4     3     9
     5    -2    -6

p=[0; 1; 0]
p =
     0
     1
     0

E=eye(3)
E =
     1     0     0
     0     1     0
     0     0     1

P=[0 1 0; 1 0 0; 0 0 -1]
P =
     0     1     0
     1     0     0
     0     0    -1

E*C
ans =
     3    -1     8
    -4     3     9
     5    -2    -6

C*E
ans =
     3    -1     8
    -4     3     9
     5    -2    -6

E*p
ans =
     0
     1
     0

C*p
ans =
    -1
     3
    -2

p'*C
ans =
    -4     3     9

C*P
ans =
    -1     3    -8
     3    -4    -9
    -2     5     6

P*C
ans =
    -4     3     9
     3    -1     8
    -5     2     6


% ------------------------------------------------------------------------
% Aufgabe 4
% ------------------------------------------------------------------------

s=[15; 8]
s =
    15
     8

t=[-8; 16]
t =
    -8
    16

% a) -------------------------------------------------------------------------

norm(s)
ans =
    17

norm(t)
ans =
   17.8885

Norm__s_plus_t=norm(s+t)
Norm__s_plus_t =
    25

Norm_s_plus_Norm_t=norm(s)+norm(t)
Norm_s_plus_Norm_t =
   34.8885

fprintf('\nDreiecksungleichung '),
if Norm__s_plus_t<=Norm_s_plus_Norm_t, fprintf('erfüllt\n\n'),
 else fprintf('nicht erfüllt\n\n'),
end

Dreiecksungleichung erfüllt

% b) -------------------------------------------------------------------------

arc=acos(s'*t/(norm(s)*norm(t)))
arc =
    1.5445

Winkel_in_Grad=arc*180/pi
Winkel_in_Grad =
   88.4926
```

```
% ------------------------------------------------------------------------
% Aufgabe 5
% ------------------------------------------------------------------------

u=[1; 2; -1]
u =
     1
     2
    -1

v=[-2; 3; 0]
v =
    -2
     3
     0

z1=-u-v; z2=-u;  z3=-u+v;
z4=  -v; z5=0*v; z6=   v;
z7= u-v; z8= u;  z9= u+v;

hold on;

plot3([0,z1(1)],[0,z1(2)],[0,z1(3)],'LineWidth',3);
plot3([0,z2(1)],[0,z2(2)],[0,z2(3)],'LineWidth',3);
plot3([0,z3(1)],[0,z3(2)],[0,z3(3)],'LineWidth',3);
plot3([0,z4(1)],[0,z4(2)],[0,z4(3)],'LineWidth',3);
plot3([0,z5(1)],[0,z5(2)],[0,z5(3)],'LineWidth',3);
plot3([0,z6(1)],[0,z6(2)],[0,z6(3)],'LineWidth',3);
plot3([0,z7(1)],[0,z7(2)],[0,z7(3)],'LineWidth',3);
plot3([0,z8(1)],[0,z8(2)],[0,z8(3)],'LineWidth',3);
plot3([0,z9(1)],[0,z9(2)],[0,z9(3)],'LineWidth',3);

plot3([z1(1),z3(1)],[z1(2),z3(2)],[z1(3),z3(3)],'-.k');
plot3([z3(1),z9(1)],[z3(2),z9(2)],[z3(3),z9(3)],'-.k');
plot3([z9(1),z7(1)],[z9(2),z7(2)],[z9(3),z7(3)],'-.k');
plot3([z7(1),z1(1)],[z7(2),z1(2)],[z7(3),z1(3)],'-.k');

xlabel('x'); ylabel('y'); zlabel('z');
grid on

% ggf. von Hand in gewünschte Lage drehen,
% mit der folgenden Anweisung wird Betrachterstandpunkt vorgegeben:
view(66,36)

viewparam=get(gca(),'view');
title(['Plot von Vorname1 Name1 und Vorname2 Name2, azimuth=',num2str(
viewparam(1)),' elevation=',num2str(viewparam(2))]);
print -depsc ak3_zusatz_5a.eps

view(8,51)

viewparam=get(gca(),'view');
title(['Plot von Vorname1 Name1 und Vorname2 Name2, azimuth=',num2str(
viewparam(1)),' elevation=',num2str(viewparam(2))]);
print -depsc ak3_zusatz_5b.eps

diary off
```