Im Rahmen dieses Aufgabenblattes sollen Sie sich mit dem Computer-Algebra-System `Mathematica` *beschäftigen und sich – neben den grundsätzlichen syntaktischen Konventionen einige Werkzeuge für die Behandlung und Darstellung von Matrizen und linearen Abbildungen erarbeiten. Als Teil Ihrer Lösung können Sie gerne ein (ausgedrucktes!) Mathematica-Notebook abgeben, dies ist aber nicht erforderlich.*

Mathematica: Grundlagen

Nachdem man Mathematica gestartet und ein neues Notebook geöffnet hat, können Kommandos eingegeben werden. Mehrere Kommandos werden durch Zeilenumbruch oder ein Semikolon getrennt. Das Vorhandensein eines Semikolons verhindert die Ausgabe des Ergebnisses eines Befehls.

```
a = 3+5; b = Log[2]; a + b
```

Wenn man Mathematica mit Hilfe von `Shift+Enter` anweist, die eingegebenen Befehle auszuführen, erhält man

```
 In[1]:=  a = 3+5; b = Log[2]; a + b
Out[1] =  8 + Log[2]
```

wobei Mathematica sowohl die Eingabezuweisung `In [1] :=` als auch die Ausgabemarkierung `Out[1] =` automatisch hinzufügt; diese müssen also *nicht* mit eingegeben werden. Hilfe zu jeder Funktion erhält man mittels

```
 In[2]:=  ? Simplify
          Simplify[expr] performs a sequence of algebraic and other transformations
 ↪  on expr and returns the simplest form it finds.
```

Grundsätzliche Arithmetik sowie die grundlegenden Funktionen funktionieren genau wie erwartet. Multiplikation kann durch ein Leerzeichen ersetzt werden. Funktionen, die nur ein Argument erwarten, wie zum Beispiel `Simplify[]` und `Expand[]`, können auch mit Hilfe von `//` auf einen Ausdruck angewendet werden:

```
 In[3]:=  a = 3+5
          b = Log[2]
          b + Log[4] - Log[a]
          Simplify[b + Log[4] - Log[a]]
          Log[2]+Log[4] - Log[a] // Simplify
          Exp[a] Exp[b] // Simplify
          (c + d)^4 // Expand
Out[3] =  8
Out[4] =  Log[2]
Out[5] =  Log[2] + Log[4] - Log[8]
Out[6] =  0
Out[7] =  0
Out[8] =  2 Exp[8]
Out[9] =  c^4 + 4 c^3 d + 6 c^2 d^2 + 4 c d^3 + d^4
```

Ausdrücke können mit Hilfe der Kommandos `Expand[]`, `Simplify[]`, `FullSimplify[]`, `Factor[]`, `Together[]`, `Apart[]`, `ExpToTrig[]`, `TrigToExp[]` in eine passende algebraische Form gebracht werden.

```
 In[10]:= Cos[x] + I Sin [x] // FullSimplify
          a/8 + b/4
          a/8 + b/4 // Together
          % // Apart
          (x - a1) (x - a2) (x - a3) // Expand
          % // Factor
          Log[z + Sqrt[z^2 + 1]] // ExpToTrig
Out[10] = Exp[I x]
Out[11] = 1 + Log[2]/4
Out[12] = 1/4 (4 + Log[2])
Out[13] = 1 + Log[2]/4
Out[14] = -a1 a2 a3 + a1 a2 x + a1 a3 x + a2 a3 x - a1 x^2 - a2 x^2 - a3 x^2 + x^3
Out[15] = - (a1 - x) (a2 - x) (a3 - x)
Out[16] = ArcSinh[z]
```

Jedes Objekt in Mathematica wird intern als Liste dargestellt. Auf die interne Darstellung kann man mit dem Kommando `FullForm[]` zugreifen. Jede Liste hat einen Kopf (`Head[]`), der angibt, welcher Natur die gelisteten Objekte sind oder was mit ihnen geschehen soll:

```
In [17]:= FullForm[(c+d)^2]
Out[17] = Power[Plus[c, d], 2]
In [18]:= Head[c+d]
Out[18] = Plus
```

Listen können mit `Join[]` verknüpft werden und auf einzelne Listenelemente kann man mit Hilfe von `Part[]` oder der dazugehörigen Kurzform `[[ ]]` direkt zugreifen:

```
In [19]:= lst = {a, b, 3, Exp[I Pi]}
Out[19] = {8, Log[2], 3, -1}
In [20]:= Head[lst]
Out[20] = List
In [21]:= Exp[lst[[2]]]
Out[21] = 2
```

Für komplexe Zahlen gibt es einige grundsätzliche Funktionen, die sich hoffentlich durch das folgende Beispiel gut erschließbar sind:

```
In [22]:= z = 1/2 - 0.3 I
          Re[z]
          Im[z]
          {Re[z],Im[z]}
          ReIm[z]
          Complex[1/2, -0.3]
Out[22] = 0.5 - 0.3 I
Out[23] = 0.5
Out[24] = -0.3
Out[25] = {0.5, -0.3}
Out[26] = {0.5, -0.3}
Out[27] = 0.5 - 0.3 I
```

Matrizen werden in Mathematica als verschachtelte Listen dargestellt. Die Unterlisten sind dabei die Zeilen der Matrix. Eine Bildschirmdarstellung als Matrix erhält man mit dem Befehl `MatrixForm[]`.

```
In [28]:= AA = {{7, 2, 1}, {0, 3, -1}, {-3, 4, -2}}; AA // MatrixForm
In [29]:= AAInv = Inverse[AA]; AAInv // MatrixForm
```

Weitere Matrixoperationen (die Sie unbedingt ausprobieren sollten) sind `Det[]`, `Inverse[]`, `Transpose[]`. Eine Einheitsmatrix erzeugt man mittels `IdentityMatrix[]`. Einzelne Elemente in einer Matrix kann man mittels `ReplacePart[]` verändern.

```
In [30]:= BB = ReplacePart[AA, {1, 1} -> 9]; BB // MatrixForm
```

Mathematica bietet die Möglichkeit, Prozeduren zu definieren, damit Operationen nicht mehrfach eingegeben werden müssen. Hier sollen – als Beispiel – die Elementarmatrizen in Fischers Notation definiert werden. Die Objekte in eckigen Klammern, die mit einem Unterstrich enden, sind dabei die *Parameter*, die der Prozedur übergeben werden.

```
In [31]:= EE[i_ , j_] := ReplacePart[Table[0, {dim}, {dim}], {i, j} -> 1];
          FiS[i_ , lambda_] := ReplacePart[IdentityMatrix[dim], {i, i} -> lambda];
          FiQ[i_ , j_] := IdentityMatrix[dim] + EE[i, j];
          FiQl[i_ , j_ , lambda_] := IdentityMatrix[dim] + lambda EE[i, j];
          FiP[i_ , j_] := ReplacePart[IdentityMatrix[dim], {{i, i} -> 0, {j, j} -> 0,
↪  {i, j} -> 1, {j, i} -> 1}]
```

Zunächst werden die Matrizen E_{ij} definiert, die an der Stelle $\{i, j\}$ gerade eine Eins und sonst Nullen aufweisen. Mit Hilfe dieser Hilfsprozedur kann man dann die anderen Elementarmatrizen zusammenbauen. Die Zuweisung `:=` besagt (im Vergleich zum bloßen Gleichheitszeichen), dass bei jedem Aufruf die rechte Seite neu berechnet werden soll.

Die Inversion der Matrix `AA` läßt sich dann schrittweise – dem in der Vorlesung vorgestellten Schema folgend – wie folgt berechnen:

```
In [32]:= dim = 3;
          BB[1] = FiP[3, 2].AA; BB[1] // MatrixForm
          BB[2] = FiQl[2, 1, 3/7].BB[1]; BB[2] // MatrixForm
          BB[3] = FiQl[3, 2, -21/34].BB[2]; BB[3] // MatrixForm
          BB[4] = FiS[1, 1/7].FiS[2, 7/34].FiS[3, -34].BB[3]; BB[4] // MatrixForm
          (* Gesamttransformation GT *)
          GT = FiS[1, 1/7].FiS[2, 7/34].FiS[3, -34].FiQl[3, 2, -21/34].FiQl[2, 1,
↪  3/7].FiP[3, 2]; GT // MatrixForm
          Det[GT]
          (* Endergebnis *)
          GT.AA // MatrixForm
```

Natürlich beherrscht Mathematica auch die Berechnung von Eigenwerten und die Diagonalisierung von Matrizen. Die dazugehörigen Befehle lauten `Eigensystem[]`, `Eigenvalues[]` und `Eigenvectors`. Zusätzlich gibt es noch eine Abfragefunktion, `DiagonalizableQ[]`, die mitteilt, ob eine Matrix diagonalisierbar ist oder nicht.

Bei den folgenden Aufgaben ist Mathmatica mit Sicherheit ein hilfreiches Arbeitsmittel. Sie sind aber alle auch allein mit Zettel und Papier lösbar.

$\Longrightarrow$

1. Simultane Diagonalisierung **[4 Punkte]**

Bestimmen Sie eine Matrix $S \in M(4 \times 4; \mathbb{R})$ so, dass SAS^{-1} und SBS^{-1} diagonal sind, wobei

$$A = \begin{pmatrix} -1 & 2 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ -2 & 2 & 2 & 1 \\ 2 & -2 & -1 & 0 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} -1 & 4 & 2 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix}.$$

Welche Bedingung müssen die Matrizen A und B erfüllen, damit sie gleichzeitig diagonalisiert werden können? Überprüfen Sie diese Bedingung explizit!

2. Diagonalisierung **[5 Punkte]**

a) Überprüfen Sie, ob die Matrix

$$A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 5 & 1 \\ 0 & -1 & 3 \end{pmatrix}$$

diagonalisierbar ist! Bestimmen Sie dazu sowohl das charakteristische Polynom als auch die zu den Eigenwerten korrespondierenden Eigenräume und vergleichen Sie algebraische und geometrische Vielfachheiten. Bestimmen Sie den Rang der Abbildung(en) $(A - \lambda_i E_3)$ und setzen Sie Ihr Ergebnis in Beziehung zu den eben bestimmten Vielfachheiten.

b) Sei eine Abbildung $x \mapsto Bx$, $\mathbb{C}^3 \to \mathbb{C}^3$ durch die Matrix

$$B = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

gegeben. Bestimmen Sie das charakteristische Polynom und die drei Eigenwerte $\lambda_i \in \mathbb{C}$ von B! Bestimmen Sie zu jedem Eigenwert λ_i einen Eigenvektor v_i und zeigen Sie, dass mit der aus den Eigenvektoren gebildeten Matrix S^{-1} die Matrix SBS^{-1} diagonal ist.

$\Longrightarrow$

3. Verschobene Zeilen und Spalten **[5 Punkte]**

Gegeben sei eine Matrix $A_n \in M(n \times n; \mathbb{C})$ der Form

$$A_n = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_{n-1} & & a_2 \\ \vdots & a_1 & a_0 & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{pmatrix}.$$

a) Der Beweis der Gültigkeit der allgemeinen Formel für Eigenwerte und Eigenvektoren dieser Matrizen ist langwierig. Zeigen Sie daher an den Beispielen $n = 2, 3$ und 4, dass sich die Eigenwerte λ_j und die Eigenvektoren v_j der Matrizen A_n zu

$$\lambda_j = a_0 + a_{n-1}\rho_j + a_{n-2}\rho_j^2 + \cdots + a_1\rho_j^{n-1}$$

und

$$v_j = (1, \rho_j, \rho_j^2, \ldots, \rho_j^{n-1}), \quad j = 0, 1, \ldots, n-1,$$

ergeben, wobei $\rho_j = \exp(\frac{2\pi i j}{n})$ die n-ten Einheitswurzeln sind.

b) Sei p_k der k-te Spaltenvektor von A_n. Finden Sie diejenigen Matrizen P_n, die die Verschiebung der Elemente um eine Position vermitteln, für die also gilt:

$$P_n \cdot p_k = p_{k+1}\,,$$

wobei der Index des Vektors modulo n verstanden werden soll.

c) Stellen Sie die Matrix A_n als Polynom über der Matrix P_n dar.

4. Spuren **[6 Punkte]**

a) Sei K ein Körper. Zeigen Sie, dass die in der Vorlesung definierte *Spur* eine lineare Abbildung

$$M(n \times n; K) \to K$$

ist.

b) Sei $A \in M(n \times n; \mathbb{C})$. Beweisen Sie: falls die Matrix A Rang 1 hat, ist der einzige evtl. von Null verschiedene Eigenwert gerade $\mathrm{Sp}\, A$.

c) Seien nun $A_i \in M(n \times n; K)$. Beweisen Sie die Zyklizitätseigenschaft der Spur:

$$\mathrm{Sp}(A_1 A_2 \cdots A_n) = \mathrm{Sp}(A_n A_1 A_2 \cdots A_{n-1})\,.$$

d) Zeigen Sie, in Analogie zur Endomorphismendeterminante, wie man über eine darstellende Matrix $M_{\mathcal{B}}(F)$ auch eine mit a) kompatible allgemeinere Abbildung

$$\mathrm{Sp}\colon \mathrm{End}_K(V) \to K$$

definieren kann. Zeigen Sie unter Zuhilfenahme von c), dass die gesuchte Definition nicht von der Basis $\mathcal{B}$ abhängt.