Lineare Algebra 1

WS 2000 / 2001 - Prof. Plesken

Lösungen der Übungen

2. Gruppenübung Lineare Algebra I

Aufgabe 4:

Zunächst definieren wir die Funktionen wie auf dem Aufgabenblatt

```
> f:=x->x^2+x+1;
```

$$f := x \to x^2 + x + 1$$

```
> g:=x->x^3+x-1;
```

$$g := x \to x^3 + x - 1$$

Berechne die Verkettung der Ausdrücke folgendermaßen:

```
> (f@g)(x);
```

$$(x^3 + x - 1)^2 + x^3 + x$$

Überführe das letzte Ergebnis nun in Summendarstellung (d.h. multipliziere den Term aus):

```
> expand(%);
```

$$x^6 + 2\,x^4 - x^3 + x^2 - x + 1$$

```
> (g@f)(x);
```

$$(x^2 + x + 1)^3 + x^2 + x$$

```
> expand(%);
```

$$x^6 + 3\,x^5 + 6\,x^4 + 7\,x^3 + 7\,x^2 + 4\,x + 1$$

Entsprechendes funktioniert auch bei dem zweiten Paar von Funktionen:

```
> f3:=(x,y,z)->(x+y+z,2*x+z,-z);
```

$$f3 := (x, y, z) \to (x + y + z,\ 2\,x + z,\ -z)$$

```
> g3:=(x,y,z)->(2*x,x+y,3*x-y+2*z);
```

$$g3 := (x, y, z) \to (2\,x,\ x + y,\ 3\,x - y + 2\,z)$$

```
> (f3@g3)(x,y,z);
```

$$6\,x + 2\,z,\ 7\,x - y + 2\,z,\ -3\,x + y - 2\,z$$

```
> (g3@f3)(x,y,z);
```

$$2\,x + 2\,y + 2\,z,\ 3\,x + y + 2\,z,\ x + 3\,y$$

```
>
```

3. Gruppenübung zur Linearen Algebra I

Um Matrixrechnungen ausführen zu können, müssen wir zuerst die Befehle zur linearen Algebra in den Speicher laden:

```
> with(linalg):
```

Aufgabe 1:

Jetzt definieren wir die vorkommenden Matrizen in MAPLE:

```
> A:=matrix(3,3,[[1,1,1],[2,0,1],[0,0,-1]]);
```

$$A := \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

```
> B:=matrix(3,3,[[2,0,0],[1,1,0],[3,-1,2]]);
```

$$B := \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 2 \end{bmatrix}$$

```
> C:=matrix(3,4,[[1,-1,3,-2],[2,0,1,0],[0,-4,-1,2]]);
```

$$C := \begin{bmatrix} 1 & -1 & 3 & -2 \\ 2 & 0 & 1 & 0 \\ 0 & -4 & -1 & 2 \end{bmatrix}$$

```
> D_:=matrix(3,1,[[1],[-7],[9]]);
```

$$D\_ := \begin{bmatrix} 1 \\ -7 \\ 9 \end{bmatrix}$$

Das Produkt von Matrizen berechnet sich nun folgendermaßen:

```
> evalm(A&*A);
```

$$\begin{bmatrix} 3 & 1 & 1 \\ 2 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> evalm(A&*B);
```

$$\begin{bmatrix} 6 & 0 & 2 \\ 7 & -1 & 2 \\ -3 & 1 & -2 \end{bmatrix}$$

```
> evalm(A&*C);
```

$$\begin{bmatrix} 3 & -5 & 3 & 0 \\ 2 & -6 & 5 & -2 \\ 0 & 4 & 1 & -2 \end{bmatrix}$$

```
> evalm(A&*D_);
```

$$\begin{bmatrix} 3 \\ 11 \\ -9 \end{bmatrix}$$

```
> evalm(B&*A);
```

$$\begin{bmatrix} 2 & 2 & 2 \\ 3 & 1 & 2 \\ 1 & 3 & 0 \end{bmatrix}$$

```
> evalm(B&*B);
```

$$\begin{bmatrix} 4 & 0 & 0 \\ 3 & 1 & 0 \\ 11 & -3 & 4 \end{bmatrix}$$

```
> evalm(B&*C);
```

$$\begin{bmatrix} 2 & -2 & 6 & -4 \\ 3 & -1 & 4 & -2 \\ 1 & -11 & 6 & -2 \end{bmatrix}$$

```
> evalm(B&*D_);
```

$$\begin{bmatrix} 2 \\ -6 \\ 28 \end{bmatrix}$$

```
> evalm(A&*B&*C);
```

$$\begin{bmatrix} 6 & -14 & 16 & -8 \\ 5 & -15 & 18 & -10 \\ -1 & 11 & -6 & 2 \end{bmatrix}$$

```
> evalm(B&*A&*C);
```

$$\begin{bmatrix} 6 & -10 & 6 & 0 \\ 5 & -11 & 8 & -2 \\ 7 & -1 & 6 & -2 \end{bmatrix}$$

```
> evalm(A&*B&*D_);
```

$$\begin{bmatrix} 24 \\ 32 \\ -28 \end{bmatrix}$$

```
> evalm(B&*A&*D_);
```

$$\begin{bmatrix} 6 \\ 14 \\ -20 \end{bmatrix}$$

Aufgabe 3:

Hier geben wir nun zuerst die Matrix und den Lösungsvektor vor:

```
> M:=matrix(8,8,[[-1,0$7],[0,1,0$6],[-1,0,-sqrt(3)/2,1/2*sqrt(2),0$4
```

```
],[0,0,1/2,1/2*sqrt(2),0$4],[0$3,-1/2*sqrt(2),-1,0$3],[0$3,-1/2*sq
rt(2),0,0,0,1],[0,0,sqrt(3)/2,0,1,1,0,0],[0,-1,-1/2,0$3,1,0]]);
```

$$M := \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -\frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2}\sqrt{2} & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2}\sqrt{2} & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2}\sqrt{3} & 0 & 1 & 1 & 0 & 0 \\ 0 & -1 & \frac{-1}{2} & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> b:=vector(8,[1,0$7]);
```

$$b := [1, 0, 0, 0, 0, 0, 0, 0]$$

Wir lösen das lineare Gleichungssystem in einem Schritt:

```
> sol:=linsolve(M,b);
```

$$sol := \left[-1, 0, \sqrt{3} - 1, -\frac{1}{2}\sqrt{3}\sqrt{2} + \frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{3} - \frac{1}{2}, -1, \frac{1}{2}\sqrt{3} - \frac{1}{2}, -\frac{1}{2}\sqrt{3} + \frac{1}{2}\right]$$

Stelle das Ergebnis nun auf verschiedene Arten dar:

```
> map(x->expand(-x*(1+sqrt(3))),sol);
```

$$[1 + \sqrt{3}, 0, -2, \sqrt{2}, -1, 1 + \sqrt{3}, -1, 1]$$

```
> map(x->-x,sol);
```

$$\left[1, 0, -\sqrt{3} + 1, \frac{1}{2}\sqrt{3}\sqrt{2} - \frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{3} + \frac{1}{2}, 1, -\frac{1}{2}\sqrt{3} + \frac{1}{2}, \frac{1}{2}\sqrt{3} - \frac{1}{2}\right]$$

```
> map(x->evalf(x),sol);
```

[-1., 0, .732050808, -.5176380910, .3660254040, -1., .3660254040, -.3660254040]

```
>
```

4. Gruppenübung Lineare Algebra I

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

In diesem Worksheet benötigen wir eine zusätzliche, nicht im MAPLE-Paket enthaltene Prozedur, die i der Datei `lasupp.txt` bereitgestellt wird. Um diese einbinden zu können, müssen wir zunächst den Pfad angeben, in dem die Datei auf der Festplatte gespeichert ist, oder aber einfacher die Datei gleich in das Verzeichnis speichern, in dem MAPLE nach ihr suchen wird.

```
> currentdir();
```

"/usb/gehrt"

Wenn die Datei in diesem Verzeichnis steht, können wir sie wie folgt einlesen, sonst müssen wir in der nächsten Zeile den gesamten Pfad vor dem Dateinamen ergänzen, z.B. "C://dokumente//maple//lasupp.txt" oder enstsprechend (Hierbei wird in der Windows-Umgebung der Backslash (\) durch den Doppelslash (//) ersetzt).

```
> read‘lasupp.txt‘;
```

Aufgabe 1:

Wir definieren die Matrix wie gehabt:

```
> A:=matrix(4,4,[[1,-1,2,0],[1,0,1,0],[-1,3,-3,1],[-1,-1,1,3]]);
```

$$A := \begin{bmatrix} 1 & -1 & 2 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 3 & -3 & 1 \\ -1 & -1 & 1 & 3 \end{bmatrix}$$

und invertieren sie folgendermaßen:

```
> inverse(A);
```

$$\begin{bmatrix} -5 & 5 & \frac{-3}{2} & \frac{1}{2} \\ 4 & -3 & \frac{3}{2} & \frac{-1}{2} \\ 5 & -4 & \frac{3}{2} & \frac{-1}{2} \\ -2 & 2 & \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$$

```
> B:=matrix(3,4,[[1,-1,-3,0],[1,0,-4,0],[1,1,-4,1]]);
```

$$B := \begin{bmatrix} 1 & -1 & -3 & 0 \\ 1 & 0 & -4 & 0 \\ 1 & 1 & -4 & 1 \end{bmatrix}$$

Um die Rechtsinverse zu bestimmen, gehen wir vor wie in der Vorlesung: Wir schreiben die Einheitsmatrix rechts neben die Matrix:

```
> B1:=augment(B,array(1..3,1..3,identity));
```

```
>
```

$$B1 := \begin{bmatrix} 1 & -1 & -3 & 0 & 1 & 0 & 0 \\ 1 & 0 & -4 & 0 & 0 & 1 & 0 \\ 1 & 1 & -4 & 1 & 0 & 0 & 1 \end{bmatrix}$$

...und bringen sie dann auf strikte Stufenform:

```
> B2:=gaussjord(B1);
```

$$B2 := \begin{bmatrix} 1 & 0 & 0 & 4 & 4 & -7 & 4 \\ 0 & 1 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 & 1 & -2 & 1 \end{bmatrix}$$

Nun ergänzen wir die vierte Zeile:

```
> B3:=stackmatrix(B2,[0,0,0,1,a,b,c]);
```

$$B3 := \begin{bmatrix} 1 & 0 & 0 & 4 & 4 & -7 & 4 \\ 0 & 1 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & a & b & c \end{bmatrix}$$

und bestimmen erneut die strikte Stufenform:

```
> B4:=gaussjord(B3);
```

$$B4 := \begin{bmatrix} 1 & 0 & 0 & 0 & 4-4a & -7-4b & 4-4c \\ 0 & 1 & 0 & 0 & -a & -1-b & 1-c \\ 0 & 0 & 1 & 0 & 1-a & -2-b & 1-c \\ 0 & 0 & 0 & 1 & a & b & c \end{bmatrix}$$

Daraus können wir nun die Rechtsinverse der Ursprungsmatrix extrahieren:

```
> B5:=submatrix(B4,1..4,5..7);
```

$$B5 := \begin{bmatrix} 4-4a & -7-4b & 4-4c \\ -a & -1-b & 1-c \\ 1-a & -2-b & 1-c \\ a & b & c \end{bmatrix}$$

überprüfe ihre rechtsinverse Eigenschaft:

```
> evalm(B&*B5);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

setze die freien Parameter gleich Null:

```
> subs(a=0,b=0,c=0,copy(B5));
```

$$\begin{bmatrix} 4 & -7 & 4 \\ 0 & -1 & 1 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

...und nun führen wir das gleiche Verfahren auch für das zweite Beispiel durch:

```
> C:=matrix(2,3,[[1,2,3],[2,4,-1]]);
```

$$C := \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -1 \end{bmatrix}$$

```
> C1:=gaussjord(augment(C,array(1..2,1..2,identity)));
```

$$C1 := \begin{bmatrix} 1 & 2 & 0 & \frac{1}{7} & \frac{3}{7} \\ 0 & 0 & 1 & \frac{2}{7} & \frac{-1}{7} \end{bmatrix}$$

```
> C2:=gaussjord(stackmatrix(row(C1,1),[0,1,0,a,b],row(C1,2)));
```

$$C2 := \begin{bmatrix} 1 & 0 & 0 & \frac{1}{7} - 2\,a & \frac{3}{7} - 2\,b \\ 0 & 1 & 0 & a & b \\ 0 & 0 & 1 & \frac{2}{7} & \frac{-1}{7} \end{bmatrix}$$

```
> C3:=submatrix(C2,1..3,4..5);
>
```

$$C3 := \begin{bmatrix} \frac{1}{7} - 2\,a & \frac{3}{7} - 2\,b \\ a & b \\ \frac{2}{7} & \frac{-1}{7} \end{bmatrix}$$

```
> evalm(C&*C3);
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> subs(a=0,b=0,copy(C3));
```

$$\begin{bmatrix} \frac{1}{7} & \frac{3}{7} \\ 0 & 0 \\ \frac{2}{7} & \frac{-1}{7} \end{bmatrix}$$

Aufgabe 3:

Wir definieren zuerst die Funktion vom Aufgabenblatt:

```
> f:=x->a*x^3+b*x^2+c*x+d;
```

$$f := x \to a\,x^3 + b\,x^2 + c\,x + d$$

Wir können das gegebene Gleichungssystem entweder sofort und direkt lösen lassen:

```
> solve({f(1)=1,f(2)=2,f(3)=5});;
```

$$\{d = d,\ b = d - 1,\ c = -\frac{11}{6}\,d + \frac{5}{3},\ a = -\frac{1}{6}\,d + \frac{1}{3}\}$$

oder wir gehen vor, wie in der Vorlesung gelernt: Stelle zuerst die Koeffizientenmatrix des linearen Gleichungssystems auf (wir benutzen hier die oben aus dem File bereitgestellte Funktion `coeffmatrix`):

```
> F:=coeffmatrix([f(1),f(2),f(3)]);
```

$$F := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{bmatrix}$$

und bringe die Matrix dann auf strikte Stufenform:

```
> gaussjord(augment(F,vector(3,[1,2,5])));
```

$$\begin{bmatrix} 1 & 0 & 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & \frac{11}{6} & \frac{5}{3} \end{bmatrix}$$

...oder aber wir lösen das System wieder direkt und geben die Paramter vor, nach denen wir auflösen wollen:

```
> solve({f(1)=1,f(2)=2,f(3)=5},{b,c,d});
```

$$\{d = -6\,a + 2,\ c = 11\,a - 2,\ b = -6\,a + 1\}$$

```
> solve({f(1)=1,f(2)=2,f(3)=5},{a,c,d});
```

$$\{a = -\frac{1}{6}\,b + \frac{1}{6},\ d = b + 1,\ c = -\frac{11}{6}\,b - \frac{1}{6}\}$$

```
> solve({f(1)=1,f(2)=2,f(3)=5},{a,b,d});
```

$$\{d = -\frac{6}{11}\,c + \frac{10}{11},\ a = \frac{1}{11}\,c + \frac{2}{11},\ b = -\frac{6}{11}\,c - \frac{1}{11}\}$$

```
> solve({f(1)=1,f(2)=2,f(3)=5},{a,b,c});
```

$$\{a = -\frac{1}{6}\,d + \frac{1}{3},\ b = d - 1,\ c = -\frac{11}{6}\,d + \frac{5}{3}\}$$

Das gleiche Verfahren wenden wir nun auch für die zweite Gleichung an:

```
> g:=x->a*sin(x)+b*cos(x)+c*cos(2*x);
```

$$g := x \to a\sin(x) + b\cos(x) + c\cos(2\,x)$$

Zunächst berechnen wir einige interessante Funktionswerte:

```
> g(0);
```

$$b + c$$

```
> g(Pi/4);
```

$$\frac{1}{2}\,a\sqrt{2} + \frac{1}{2}\,b\sqrt{2}$$

```
> g(Pi/2);
```

$$a - c$$

Die direkte Lösung liefert keinen Output, also legt das schon nahe, dass keine Lösungen für das

Gleichungssystem existieren...

```
> solve({g(0)=1,g(Pi/4)=1,g(Pi/2)=-1});
> G:=coeffmatrix([g(0),g(Pi/4),g(Pi/2)]);
```

$$G := \begin{bmatrix} 1 & 1 & 0 \\ \frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} \\ 0 & -1 & 1 \end{bmatrix}$$

...Entsprechndes zeigt uns auch die gaußreduzierte Koeffizientenmatrix:

```
> gaussjord(augment(G,vector(3,[1,1,-1])));
```

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Gruppenübung Lineare Algebra I

Aufgabe 2:

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Die erste Matrix können wir wie gewohnt angeben und invertieren:

```
> A:=matrix(3,3,[[-3,-2+2*I,-I],[5+5*I,5,2*I],[2+2*I,2,I]]);
```

$$A := \begin{bmatrix} -3 & -2+2I & -I \\ 5+5I & 5 & 2I \\ 2+2I & 2 & I \end{bmatrix}$$

```
> inverse(A);
```

$$\begin{bmatrix} 1 & -2I & 1+4I \\ -1-I & -1+2I & 1-5I \\ 0 & 2I & -5I \end{bmatrix}$$

Bei den Matrizen über Restklassenkörpern wird es allerdings ein klein wenig aufwendiger: Wir definieren sie zunächst wie gehabt:

```
> B:=matrix(5,5,[[0,0,1,1,1],[0,0,1,1,0],[0,1,0,1,1],[1,0,1,0,0],[1,
  1,0,1,0]]);
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

```
> C:=matrix(4,4,[[2,1,1,1],[1,2,0,2],[0,0,0,2],[0,1,0,1]]);
```

$$C := \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Zum Invertieren geben wir nun die Rechnung modulo der Primzahl explizit an (Beachte: der Invertierungsbefehl lautet nun `Inverse` statt `inverse`, da wir jetzt die inerte Funktion benutzen müssen, d.h. es wird zuerst die gesamte Befehlszeile inerpretiert, bevor das Kommando ausgeführt wir

```
> BI:=Inverse(B) mod 2;
```

$$BI := \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

```
> CI:=Inverse(C) mod 3;
```

$$CI := \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

Um die inversen Matrizen zu überprüfen, wenden wir auch wieder einen kleinen Trick an, um modulo der Primzahl zu rechnen: Wir wandeln die Matrix zuerst in eine verschachtelte Liste um, berechnen dann die Einträge, und konvertieren sie schließlich zurück in eine Matrix:

```
> convert(Expand( convert(evalm(B &* BI),listlist) ) mod 2,matrix);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> convert(Expand( convert(evalm(C &* CI),listlist) ) mod 3,matrix);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
>
```

6. Gruppenübung Lineare Algebra I

```
> with(linalg):
```

Aufgabe 1:

Wenn wir in MAPLE einen Bruch eingeben, wird er bei der Ausgabe automatisch vollständig gekürzt.

```
> 14568917340/14568917350;
```

$$\frac{1456891734}{1456891735}$$

Zur Kontrolle berechnen wir noch den ggT der beiden Zahlen:

```
> igcd(14568917340,14568917350);
```

10

Für die folgenden Beispiele rufen wir nun den erweiterten Euklidischen Algorithmus auf, der uns sowohl den ggT als auch die beiden Faktoren α und β liefert mit der Eigenschaft ggT $(a, b) = \alpha\, a + \beta\, b$.

```
> igcdex(17,91,'alpha','beta'); alpha; beta;
```

1

-16

3

```
> igcdex(42,101,'alpha','beta'); alpha; beta;
```

1

-12

5

```
> igcdex(668,1001,'alpha','beta'); alpha; beta;
```

1

-499

333

Das Invertieren von Zahlen im Restklassenring läuft analog ab (vgl. Vorlesung):

```
> igcdex(17,101,'alpha','beta'); alpha; beta;
```

1

6

-1

Schließlich rechnen wir die Ergebnisse noch im Restklassenkörper nach:

```
> 6*17 mod 101;
```

1

```
> igcdex(39,113,'alpha','beta'); alpha; beta;
```

1

29

-10

```
> 29*39 mod 113;
```

$$1$$

```
> igcdex(83,167,'alpha','beta'); alpha; beta;
```

$$1$$

$$-2$$

$$1$$

```
> 165*83 mod 167;
```

$$1$$

Aufgabe 2, Teil 4:

Hier wollen wir nur unsere Ergebnisse aus den theoretischen Überlegungen nachprüfen, auch hier wied
mit Matrizen über dem $\mathbf{F}_{2}$

```
> C:=matrix([[1,0,1,0,1,0,1],[0,1,1,0,0,1,1],[0,0,0,1,1,1,1]]);
```

$$C := \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> evalm(C&*transpose(C));
```

$$\begin{bmatrix} 4 & 2 & 2 \\ 2 & 4 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

```
> convert(Expand( convert(evalm(C&*transpose(C)),listlist) ) mod
  2,matrix);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
> A:=matrix([[1,0,0,0,0,1,1],[0,1,0,0,1,0,1],[0,0,1,0,1,1,0],[0,0,0,
  1,1,1,1]]);
```

$$A := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

```
> evalm(A&*transpose(C));
```

$$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 4 \end{bmatrix}$$

```
> convert(Expand( convert(evalm(A&*transpose(C)),listlist) ) mod
  2,matrix);
```

```
[
[
[
[
[ >
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

7.Gruppenübung Lineare Algebra I

Aufgabe 1:

Definiere zunächst die Polynome:

```
> f:=x^7+x^5+x^3+1; g:=x^5+x^4+x^2+1;
```

$$f := x^7 + x^5 + x^3 + 1$$

$$g := x^5 + x^4 + x^2 + 1$$

Nun führen wir die Polynomdivision durch, indem wir zuerst den Quotienten berechnen (auch hier rechnen wir wieder modulo einer Primzahl)...

```
> Quo(f,g,x,'r') mod 2;
```

$$x^2 + x$$

und zusätzlich noch den Rest modulo g ausgeben:

```
> r;
```

$$x^4 + x^2 + x + 1$$

```
> f:=x^8+x^5+x^2+1; g:=x^3+x+1;
```

$$f := x^8 + x^5 + x^2 + 1$$

$$g := x^3 + x + 1$$

Wir können den Rest natürlich auch direkt ausrechnen:

```
> Rem(f,g,x) mod 13;
```

$$11\,x$$

Ebenso bestimmen wir den ggT der beiden Polynome folgendermaßen:

```
> Gcd(f,g) mod 13;
```

$$1$$

Aufgabe 2:

```
> with(linalg):
```

Hier definieren wir zuerst die gegebenen Matrizen:

```
> A:=matrix([[0,1,0,1,1],[1,1,0,1,1],[1$5],[0$3,1,1],[1,1,0,1,1]]);
```

$$A := \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

```
> B:=matrix([[0,1,0,1],[1,0,1,1],[1$4],[0,0,1,1],[1,1,0,1]]);
```

$$B := \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

```
> C:=matrix([[1,2,0],[2,1,1],[2$3],[1,0,1]]);
```

$$C := \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 0 & 1 \end{bmatrix}$$

Der Kern lässt sich folgendermaßen berechnen (oder auch mit einem expliziten Gaußalgorithmus)

```
> Nullspace(A,'d') mod 2;
```

$$\{[0, 0, 0, 1, 1]\}$$

mit der zugehörigen Dimension

```
> d;
```

$$1$$

```
> Gaussjord(A) mod 2;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Zur Bestimmung des Bildes führen wir nun einen Gaußalgorithmus auf den Spalten der Matrix durch (oder transponieren erst, arbeiten dann wie gewohnt mit den Zeilen und transponieren anschließend zurück):

```
> transpose(Gaussjord(transpose(A))mod 2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Wir können den Spaltenraum auch direkt berechnen, benötigen bei endlichen Körpern dann jedoch eine weitere Umformung...

```
> colspace(A);
```

$$\{[0, 0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 1, 0, 0, 1], [1, 0, 0, 0, 0]\}$$

```
> Nullspace(B,'d') mod 2;
```

$$\{\}$$

```
> d;
```

$$0$$

```
> Gaussjord(B) mod 2;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> transpose(Gaussjord(transpose(B))mod 2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

```
> colspace(B);
```

$$\{[1, 0, 0, 0, 1], [0, 0, 0, 1, -1], [0, 0, 1, 0, 0], [0, 1, 0, 0, 1]\}$$

```
> map(i->convert(i,list) mod 2,colspace(B));
```

$$\{[0, 0, 0, 1, 1], [1, 0, 0, 0, 1], [0, 1, 0, 0, 1], [0, 0, 1, 0, 0]\}$$

```
> Nullspace(C,'d') mod 7;
```

$$\{\ \}$$

```
> d;
```

$$0$$

```
> Gaussjord(C) mod 7;
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

```
> transpose(Gaussjord(transpose(C))mod 7);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 3 & 4 & 2 \end{bmatrix}$$

```
> colspace(C);
```

$$\{\begin{bmatrix} 1, 0, 0, \frac{-1}{2} \end{bmatrix}, \begin{bmatrix} 0, 0, 1, \frac{1}{4} \end{bmatrix}, \begin{bmatrix} 0, 1, 0, \frac{1}{2} \end{bmatrix}\}$$

```
> map(i->convert(i,list) mod 7,colspace(C));
```

$$\{[0, 0, 1, 2], [0, 1, 0, 4], [1, 0, 0, 3]\}$$

Aufgabe 3:

```
> Quo(x^6-1,x^5-1,x,'r') mod 5;
```

$$x$$

```
> r;
```

$$x + 4$$

```
>
```

8.Gruppenübung Lineare Algebra I

Aufgabe 1:

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Um die Dimension der angegebenen Vektorräume zu bestimmen, wenden wir den Gaußalgorithmus auf die zugehörige Matrix an:

```
> A:=matrix([[2,3,4],[3,2,0],[0,0,1]]);
```

$$A := \begin{bmatrix} 2 & 3 & 4 \\ 3 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Zunächst führen wir einen Gaußalgorithmus ohne Divisionen durch, um zu sehen, wie sich die Matrix für d verschiedenen Körper verhält:

```
> ffgausselim(A);
```

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & -5 & -12 \\ 0 & 0 & -5 \end{bmatrix}$$

Wir können die gegebene Matrix aber auch direkt über den rationalen Zahlen auf strikte Stufenform bringe und die Dimension ablesen

```
> gaussjord(A);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Das gleichen machen wir nun über dem $\mathbf{F}_{-5}$:

```
> Gaussjord(A) mod 5;
```

$$\begin{bmatrix} 1 & 4 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Bei der zweiten Matrix führen wir die Umformungen über den komplexen Zahlen durch:

```
> B:=matrix([[0,sqrt(2),0],[0,1,1],[0,0,2],[1,0,0]]);
```

$$B := \begin{bmatrix} 0 & \sqrt{2} & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 1 & 0 & 0 \end{bmatrix}$$

```
> gaussjord(B);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Aufgabe 4:

An dieser Stelle kommt nun wieder der euklidische Algorithmus zum Einsatz, um die Kombination der

beiden Polynome zu ihrem ggT zu bestimmen:

```
> gcdex(x^4+x^3+x^2+x+1,x^2+1,x,'p','q');
```

$$1$$

```
> p; q;
```

$$1$$

$$-x^2 - x$$

```
> Gcdex(x^4+x^2+2,x^3+x^2+1,x,'p','q') mod 3;
```

$$1$$

```
> p;
```

$$x$$

```
> q;
```

$$2\,x^2 + x + 1$$

```
>
```

9.Gruppenübung Lineare Algebra I

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Aufgabe 1:

Zuerst definieren wir die vorgegebene Matrix:

```
> A:=matrix([[1,5,3],[2,-2,2],[-1,7,1]]);
```

$$A := \begin{bmatrix} 1 & 5 & 3 \\ 2 & -2 & 2 \\ -1 & 7 & 1 \end{bmatrix}$$

Um den Rang der Matrix zu bestimmen, bringen wir sie zunächst mit dem Gaußalgorithmus auf strikte Stufenform:

```
> gaussjord(A);
```

$$\begin{bmatrix} 1 & 0 & \frac{4}{3} \\ 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix}$$

Nun können wir den Rang direkt ablesen.
Über den rationalen oder komplexen Zahlen könen wir den Rang aber auch direkt angeben:

```
> rank(A);
```

$$2$$

```
> B:=matrix([[1,3,0,2],[2,5,2,1],[6,7,1,0],[$3..6],[1$4]]);
```

$$B := \begin{bmatrix} 1 & 3 & 0 & 2 \\ 2 & 5 & 2 & 1 \\ 6 & 7 & 1 & 0 \\ 3 & 4 & 5 & 6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Über einem Restklassenkörper geben wir den Gauß-Algorithmus folgendermaßen an...

```
> Gaussjord(B) mod 7;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

und können auch hier anschließend noch den Rang direkt ausgeben lassen (bzw. direkt ablesen).

```
> rank(%);
```

$$4$$

```
> C:=matrix([[1,1,1,0,1,0],[0,1,1,1,0,1],[0,1,0,0,1,0],[0,1,0,1,0,1]
```

```
  ,[1$4,0,0],[0,1,0,0,1,0]]);
```

$$C := \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

```
> Gaussjord(C) mod 2;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> rank(%);
```

5

Aufgabe 2:

Um den Zassenhaus-Algortihmus durchzuführen, stellen wir zuerst die Matrix auf wie in der Vorlesung angegeben:

```
> A:=matrix([[1,2,3,4,1,2,3,4],[3,1,2,0,3,1,2,0],[1$8],[0,1,2,3,0$4]
  ,[2,1,2,1,0$4]]);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 0 & 3 & 1 & 2 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Danach bringen wir sie auf strikte Stufenform und können die gesuchten Basen direkt ablesen:

```
> B:=Gaussjord(A) mod 5;
```

$$B := \begin{bmatrix} 1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \end{bmatrix}$$

Die Basis von $U + V$ entspricht den Zeilen der folgenden Teilmatrix...

```
> submatrix(B,1..3,1..4);
```

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

und die Basis des Schnittes von U und V enstpricht den Zeilen dieser Matrix:

```
> submatrix(B,4..5,5..8);
```

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

Wie man sieht, ist die erste Zeile der beiden Teilmatrizen identisch...

Aufgabe 3:

Wir definieren zunächst die gegebene Differentialgleichung:

```
> ode:=diff(y(x),x$2)+diff(y(x),x)+y(x)=x^4+2*x+3;
```

$$ode := \left(\frac{\partial^2}{\partial x^2} \mathrm{y}(x)\right) + \left(\frac{\partial}{\partial x} \mathrm{y}(x)\right) + \mathrm{y}(x) = x^4 + 2\,x + 3$$

Nun können wir die vollständige Lösung (der inhomogenen Gleichung) explizit berechnen:

```
> dsolve(ode);
```

$$\mathrm{y}(x) = -23 + 26\,x - 4\,x^3 + x^4 + \_C1\ \mathbf{e}^{(-1/2\,x)} \cos\left(\frac{1}{2}\sqrt{3}\ x\right) + \_C2\ \mathbf{e}^{(-1/2\,x)} \sin\left(\frac{1}{2}\sqrt{3}\ x\right)$$

Wir überprüfen das Ergebnis durch Einsetzen...

```
> odetest(%,ode);
```

$$0$$

Betrachten wir nun die zugehörige homogene Differentialgleichung mit ihrer Lösung:

```
> dsolve(lhs(ode)=0);
```

$$\mathrm{y}(x) = \_C1\ \mathbf{e}^{(-1/2\,x)} \cos\left(\frac{1}{2}\sqrt{3}\ x\right) + \_C2\ \mathbf{e}^{(-1/2\,x)} \sin\left(\frac{1}{2}\sqrt{3}\ x\right)$$

Wie erwartet, bleibt beim Einsetzen die rechte Seite übrig.

```
> odetest(%,ode);
```

$$-x^4 - 2\,x - 3$$

Aufgabe 4:

Für die gegebene Matrix...

```
> phi:=matrix([[1,2,3,4],[5,6,7,8],[9,10,11,12]]);
```

$$\phi := \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

können wir zunächst den Kern auf zwei Arten berechnen sowie (zeilenweise) als Matrix schreiben:

```
> nullspace(phi);
```

$$\{[2, -3, 0, 1], [1, -2, 1, 0]\}$$

```
> BB:=matrix([[2,-3,0,1],[1,-2,1,0]]);
```

$$BB := \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \end{bmatrix}$$

```
> B:=gaussjord(phi);
```

$$B := \begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Das gleiche gilt auch für das Bild der Matrix:

```
> colspace(phi);
```

$$\{[0, 1, 2], [1, 0, -1]\}$$

```
> CC:=matrix([[1,0,-1],[0,1,2]]);
```

$$CC := \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

```
> C:=gaussjord(transpose(phi));
```

$$C := \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Nun können wir entweder die berechneten Basen für Kern und Bild so ergänzen, dass die Transformationsmatrizen den Vorgaben genügen, oder aber wir wählen einen intuitiveren Zugang: Wir bringen die Matrix (durch Zeilenoperationen) auf strikte Stufenform und berechnen dabei gleichzeitig die Transformationsmatrix:

```
> A1:=augment(phi,array(1..3,1..3,identity));
```

$$A1 := \begin{bmatrix} 1 & 2 & 3 & 4 & 1 & 0 & 0 \\ 5 & 6 & 7 & 8 & 0 & 1 & 0 \\ 9 & 10 & 11 & 12 & 0 & 0 & 1 \end{bmatrix}$$

```
> gaussjord(A1);
```

$$\begin{bmatrix} 1 & 0 & -1 & -2 & 0 & \frac{-5}{2} & \frac{3}{2} \\ 0 & 1 & 2 & 3 & 0 & \frac{9}{4} & \frac{-5}{4} \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

Aus der gaußreduzierten Matrix extrahieren wir die Transformationsmatrix.

```
> C:=submatrix(%,1..3,5..7);
```

$$C := \begin{bmatrix} 0 & \frac{-5}{2} & \frac{3}{2} \\ 0 & \frac{9}{4} & \frac{-5}{4} \\ 1 & -2 & 1 \end{bmatrix}$$

Wie man sieht, liegen die ersten beiden Spaltenvektoren ihrer Inversen in Bild(ϕ)...

```
> CI:=inverse(%);
```

$$CI := \begin{bmatrix} 1 & 2 & 1 \\ 5 & 6 & 0 \\ 9 & 10 & 0 \end{bmatrix}$$

und sie hat die folgenden Eigenschaften:

```
> evalm(CI&*gaussjord(phi));
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

```
> phi1:=evalm(C&*phi);
```

$$\phi 1 := \begin{bmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Mit dieser Matrix in Stufenform fahren wir nun fort, indem wir sie (durch Spaltenoperationen) auf Einheitsgestalt bringen, und auch hier wieder die Transformationsmatrix bestimmen. (Wir arbeiten hier allerdings mit den transponierten Matrizen und Zeilenoperationen)

```
> gaussjord(augment(transpose(phi1),array(1..4,1..4,identity)));
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 3 & -2 \\ 0 & 1 & 0 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 & 0 & -3 & 2 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

Auch hier extrahieren wir wieder die Transformationsmatrix..

```
> B:=transpose(submatrix(%,1..4,4..7));
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3 & 2 & -3 & -2 \\ -2 & -1 & 2 & 1 \end{bmatrix}$$

... und es gilt die folgende Eigenschaft:

```
> evalm(phi1&*B);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Hieraus erhalten wir zusammen mit der obigen Beziehung die folgende Matrixgleichung, aus der wir erkennen, dass *C* und *B* die gesuchten Basiswechselmatrizen sind, um ϕ in die gewünschte Form zu bringen.

```
> evalm(C&*phi&*B);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Weiterhin gilt auch noch (vgl. obige Matrix):

```
> rank(phi);
```

2

10. Gruppenübung zur Linearen Algebra I

```
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
> phi:=(x1,x2,x3,x4)->(x1+3*x2,x1+x2+x3+x4,x2+x4,x1);
```

$$\phi := (x1, x2, x3, x4) \to (x1 + 3\,x2, x1 + x2 + x3 + x4, x2 + x4, x1)$$

```
> C:=[[1,-1,0,0],[1,0,-1,0],[1,1,-1,1],[2,-3,2,2]];
```

$$C := [[1, -1, 0, 0], [1, 0, -1, 0], [1, 1, -1, 1], [2, -3, 2, 2]]$$

```
> DD:=[[0,-1,0,0],[0,-1,-1,0],[3,1,2,1],[2,-1,1,1]];
```

$$DD := [[0, -1, 0, 0], [0, -1, -1, 0], [3, 1, 2, 1], [2, -1, 1, 1]]$$

Wie man sieht, sind C und D Basen des Q^4:

```
> rank(convert(C,matrix));
```

$$4$$

```
> rank(convert(DD,matrix));
```

$$4$$

Wir wenden nun zunächst die Abbildung auf die Basis an...

```
> map(i->[phi(op(i))],C);
```

$$[[-2, 0, -1, 1], [1, 0, 0, 1], [4, 2, 2, 1], [-7, 3, -1, 2]]$$

```
> pC:=transpose(convert(%,matrix));
```

$$pC := \begin{bmatrix} -2 & 1 & 4 & -7 \\ 0 & 0 & 2 & 3 \\ -1 & 0 & 2 & -1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

...und transformieren das Ergebnis dann von der Standardbasis in die Basis C

```
> evalm(transpose(inverse(convert(C,matrix)))&*pC);
```

$$\begin{bmatrix} -19 & 1 & 34 & -36 \\ 16 & -1 & -31 & 27 \\ -7 & 1 & 15 & -12 \\ 4 & 0 & -7 & 7 \end{bmatrix}$$

Jetzt stellen wir die Basiswechselmatrizen bzgl. der Standardbasis auf (beachte: Spaltenkonvention)

```
> S:=[[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]];
```

$$S := [[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]$$

```
> SPhiS:=transpose(convert(map(i->[phi(op(i))],S),matrix));
```

$$SPhiS := \begin{bmatrix} 1 & 3 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

```
> SidC:=transpose(convert(C,matrix));
```

$$SidC := \begin{bmatrix} 1 & 1 & 1 & 2 \\ -1 & 0 & 1 & -3 \\ 0 & -1 & -1 & 2 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

```
> CidS:=inverse(SidC);
```

$$CidS := \begin{bmatrix} 5 & 4 & 5 & -4 \\ -4 & -4 & -5 & 3 \\ 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$

```
> SidD:=transpose(convert(DD,matrix));
```

$$SidD := \begin{bmatrix} 0 & 0 & 3 & 2 \\ -1 & -1 & 1 & -1 \\ 0 & -1 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

```
> DidS:=inverse(DidS);
```

$$DidS := \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 3 & 1 & 2 & 1 \\ 2 & -1 & 1 & 1 \end{bmatrix}$$

```
> evalm(SPhiS&*SidC);
```

$$\begin{bmatrix} -2 & 1 & 4 & -7 \\ 0 & 0 & 2 & 3 \\ -1 & 0 & 2 & -1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$

Damit erhalten wir nun ${}_{-}{}^{C}\phi^{C}$ (vgl. oben):

```
> evalm(CidS&*SPhiS&*SidC);
```

$$\begin{bmatrix} -19 & 1 & 34 & -36 \\ 16 & -1 & -31 & 27 \\ -7 & 1 & 15 & -12 \\ 4 & 0 & -7 & 7 \end{bmatrix}$$

entsprechend ist ${}_{-}{}^{C}\phi^{D}$

```
> evalm(CidS&*SPhiS&*SidD);
```

$$\begin{bmatrix} -24 & -28 & 56 & -1 \\ 21 & 25 & -53 & -2 \\ -10 & -12 & 27 & 2 \\ 5 & 6 & -12 & 0 \end{bmatrix}$$

und ${}_{-}{}^{D}\phi^{C}$:

```
> evalm(DidS&*SPhiS&*SidC);
```

$$\begin{bmatrix} 0 & 0 & -2 & -3 \\ 1 & 0 & -4 & -2 \\ -7 & 4 & 19 & -18 \\ -4 & 3 & 9 & -16 \end{bmatrix}$$

sowie schließlich $\underline{}^{D}\phi^{D}$

```
> evalm(DidS&*SPhiS&*SidD);
```

$$\begin{bmatrix} 1 & 2 & -7 & -3 \\ 2 & 3 & -9 & -3 \\ -12 & -13 & 32 & 2 \\ -6 & -5 & 10 & -3 \end{bmatrix}$$

Aufgabe 2:

```
> phi:=matrix([[1,0,0,1,a],[1,0,1,0,b],[0,1,0,0,c],[1,1,0,1,d],[1,1,
  0,1,e]]);
```

$$\phi := \begin{bmatrix} 1 & 0 & 0 & 1 & a \\ 1 & 0 & 1 & 0 & b \\ 0 & 1 & 0 & 0 & c \\ 1 & 1 & 0 & 1 & d \\ 1 & 1 & 0 & 1 & e \end{bmatrix}$$

```
> Gaussjord(phi,4)mod 2;
```

$$\begin{bmatrix} 1 & 0 & 0 & 1 & a \\ 0 & 1 & 0 & 0 & c \\ 0 & 0 & 1 & 1 & b+a \\ 0 & 0 & 0 & 0 & d+a+c \\ 0 & 0 & 0 & 0 & e+a+c \end{bmatrix}$$

Aufgabe 3:

```
> A:=matrix([[4,0,1,1],[0,-1,0,0],[1,0,4,0],[0,0,0,1]]);
```

$$A := \begin{bmatrix} 4 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
> minpoly(A,x);
```

$$-15 + 8x + 14x^2 - 8x^3 + x^4$$

```
> factor(%);
```

$$(x-5)(x-1)(x-3)(x+1)$$

```
> eigenvectors(A);
```

$$[5, 1, \{[1, 0, 1, 0]\}], [3, 1, \{[-1, 0, 1, 0]\}], [-1, 1, \{[0, 1, 0, 0]\}], [1, 1, \{[-3, 0, 1, 8]\}]$$

```
>
```

11. Gruppenübung zur Linearen Algebra I

```
> restart;
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Aufgabe 1:

Zuerst derfinieren wir uns die Hilfsfunktion `linindep`, um zu überprüfen, ob eine Liste von Vektoren linear unabhängig (Ausgabe *true*) oder linear abhängig (Ausgabe *false*) ist. Ein optionales zweites Argument gibt den Modul beim Rechnen in endlichen Körpern an.

```
> linindep:=proc(vecs)
  local r:
  if nargs>1 then
    Gaussjord(matrix(vecs),'r') mod args[2]:
    else
    r:=rank(matrix(vecs))
  fi:
  RETURN(evalb(nops(vecs)=r))
  end;
```

linindep := **proc**(*vecs*)

local *r*;

if 1 < nargs **then** Gaussjord(matrix(*vecs*), '*r*') mod args[2]

else *r* := rank(matrix(*vecs*))

fi;

RETURN(evalb(nops(*vecs*) = *r*))

end

Definieren wir nun die gegebene Matrix:

```
> A1:=matrix([[1,0,1,1,0],[0,1,1,0,1],[0,0,1,0,0],[1,0,1,0,0],[1,0
  ,0,0,1]]);
```

$$A1 := \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Zur Bestimmung des Minimalpolynoms beginnen wir mit einem beliebigen Vektor (vgl. Vorlesung)...

```
> V1:=vector([1,0,0,0,0]);
```

$$V1 := [1, 0, 0, 0, 0]$$

...wenden die durch obige Matrix gegebene Abbildung an und erhalten so einen neuen Vektor:

```
> V11:=map('mod',evalm(A1&*V1),2);
```

$$V11 := [1, 0, 0, 1, 1]$$

Mit Hilfe der oben definierten Funktion können wir jetzt die beiden Vektoren auf lineare Unabhängigkeit überprüfen:

```
> linindep([V1,V11],2);
```

true

Da die Vektoren linear unabhängig sind, wenden wir die Abbildung nun zweimal auf den Ausgangsvektor an und nehmen das Ergebnis zu den bisherigen Vektoren hinzu.

```
> V12:=map('mod',evalm(A1&*V11),2);
```

$$V12 := [0, 1, 0, 1, 0]$$

Dieses Verfahren setzen wir solange fort, bois wir einen Vektor erhalten, der von den anderen linear abhängig ist...

```
> linindep([V1,V11,V12],2);
```

true

```
> V13:=map('mod',evalm(A1&*V12),2);
```

$$V13 := [1, 1, 0, 0, 0]$$

```
> linindep([V1,V11,V12,V13],2);
```

true

```
> V14:=map('mod',evalm(A1&*V13),2);
```

$$V14 := [1, 1, 0, 1, 1]$$

```
> linindep([V1,V11,V12,V13,V14],2);
```

false

Nachdem wir nun eine lineare Abhängigkeit erkannt haben, bestimmen wir die entsprechende Linearkombination (Gaußalgorithmus oder scharfes Hinsehen):

```
> Nullspace(transpose(matrix([V1,V11,V12,V13,V14])))mod 2;
```

$$\{[1, 1, 0, 1, 1]\}$$

Diese schreiben wir nun in ein Polynom um, das den ersten Teiler unseres Minimalpolynoms darstellt.

```
> p1:=evalm(matrix([[1,x,x^2,x^3,x^4]])&*op(%))[1];
```

$$p1 := 1 + x + x^3 + x^4$$

Da wir bisher nur vier linear unabhängige Vektoren gefunden haben, der Raum aber fünfdimension ist, müssen wir nun einen weiteren Vektor wählen, der von den bisherigen linear unabhängig ist:

```
> V2:=vector([0,0,1,0,0]);
```

$$V2 := [0, 0, 1, 0, 0]$$

```
> linindep([V1,V11,V12,V13,V2],2);
```

true

Nun verfahren wir genause wie oben, um einen zweiten Teiler des Minimalpolynoms zu erhalten...

```
> V21:=map('mod',evalm(A1&*V2),2);
```

$$V21 := [1, 1, 1, 1, 0]$$

```
> linindep([V2,V21],2);
```

true

```
> V22:=map('mod',evalm(A1&*V21),2);
```

$$V22 := [1, 0, 1, 0, 1]$$

```
> linindep([V2,V21,V22],2);
```

true

```
> V23:=map('mod',evalm(A1&*V22),2);
```

$$V23 := [0, 0, 1, 0, 0]$$

```
> linindep([V2,V21,V22,V23],2);
```

false

```
> Nullspace(transpose(matrix([V2,V21,V22,V23])))mod 2;
```

$$\{[1, 0, 0, 1]\}$$

```
> p2:=evalm(matrix([[1,x,x^2,x^3]])&*op(%))[1];
```

$$p2 := 1 + x^3$$

Da wir nun fünf linear unabhängige Vektoren betrachtet haben (v_1, v_{11}, v_{12}, v_{13} und v_2), treten keine weiteren Faktoren auf und das Minimalpolynom ergibt sich als kleinstes gemeinsames Vielfaches der beiden berechneten Teiler:

```
> mu:=expand(lcm(p1,p2));
```

$$\mu := 1 + x + x^3 + x^4$$

```
> Factor(mu) mod 2;
```

$$(x + 1)^2 (x^2 + x + 1)$$

Wir erkennen also, dass die Matrix nicht diagonalisierbar ist und 1 als einzigen Eigenwert hat.

```
> A2:=matrix([[2,0,0],[2,2,1],[-2,0,1]]);
```

$$A2 := \begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 1 \\ -2 & 0 & 1 \end{bmatrix}$$

Bei der zweiten Matrix können wir genauso vorgehen wie oben, rechnen nun allerdings über den reellen Zahlen:

```
> V1:=vector([1,0,0]);
```

$$V1 := [1, 0, 0]$$

```
> V11:=evalm(A2&*V1);
```

$$V11 := [2, 2, -2]$$

```
> linindep([V1,V11]);
```

true

```
> V12:=evalm(A2&*V11);
```

$$V12 := [4, 6, -6]$$

```
> linindep([V1,V11,V12]);
```

false

Also haben wir auch hier wieder eine Linearkombination gefunden:

```
> nullspace(transpose(matrix([V1,V11,V12])));
```

$$\{[2, -3, 1]\}$$

```
> p1:=evalm(matrix([[1,x,x^2]])&*op(%))[1];
```

$$p1 := 2 - 3\,x + x^2$$

Da wir erst zwei linear unabhängige Vektoren haben, geht es weiter...

```
> V2:=vector([0,1,0]);
```

$$V2 := [0, 1, 0]$$

```
> V21:=evalm(A2&*V2);
```

$$V21 := [0, 2, 0]$$

wie man sofort sieht...

```
> linindep([V2,V21]);
```

false

```
> nullspace(transpose(matrix([V2,V21])));
```

$$\{[-2, 1]\}$$

```
> p2:=evalm(matrix([[1,x]])&*op(%))[1];
```

$$p2 := -2 + x$$

```
> mu:=expand(lcm(p1,p2));
```

$$\mu := 2 - 3\,x + x^2$$

```
> factor(mu);
```

$$(x - 1)\,(-2 + x)$$

Wir erkennen, dass die Matrix diagonalisierbar ist, können die Diagonalgestalt jedoch noch nicht am Minimalpolynom ablesen.
Berechen also die Eigenvektoren zum Eigenwert 2:

```
> nullspace(A2-2*array(1..3,1..3,identity));
```

$$\{[1, 0, -2], [0, 1, 0]\}$$

Der Eigenraum hat also die Dimension 2, und somit hat die Diagonalmatrix folgende Gestalt:

```
> diag(2,2,1);
```

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Wir hätten bei dieser Aufgabe auch das Minimalpolynom direkt berechnen können:

```
> minpoly(A2,x);
```

$$2 - 3\,x + x^2$$

und die Eigenwerte mit zugehörigen Eigenvektoren:

```
> eigenvectors(A2);
```

$$[1, 1, \{[0, -1, 1]\}], [2, 2, \{[1, 0, -2], [0, 1, 0]\}]$$

```
> A3:=matrix([[-1,0,0],[5,2,1],[4,0,1]]);
```

$$A3 := \begin{bmatrix} -1 & 0 & 0 \\ 5 & 2 & 1 \\ 4 & 0 & 1 \end{bmatrix}$$

Bei der dritten Matrix sei der ausführliche Weg dem Leser überlassen und wir beschränken uns auf direkte Berechnung:

```
> mu:=minpoly(A3,x);
```

$$\mu := 2 - x - 2\,x^2 + x^3$$

```
> factor(mu);
```

$$(x - 1)\,(-2 + x)\,(x + 1)$$

```
> eigenvectors(A3);
```

$$[1, 1, \{[0, -1, 1]\}], [-1, 1, \{[-1, 1, 2]\}], [2, 1, \{[0, 1, 0]\}]$$

Damit hat die Matrix also die Diagonalgestalt

```
> diag(-1,1,2);
```

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Aufgabe 2:

gegeben ist die folgende Matrix mit Werten in F_5:

```
> A:=matrix([[4,0,4,1,0],[1,2,2,4,3],[2,0,2,2,4],[1,1,1,0,3],[2,2,
  4,3,3]]);
```

$$A := \begin{bmatrix} 4 & 0 & 4 & 1 & 0 \\ 1 & 2 & 2 & 4 & 3 \\ 2 & 0 & 2 & 2 & 4 \\ 1 & 1 & 1 & 0 & 3 \\ 2 & 2 & 4 & 3 & 3 \end{bmatrix}$$

Bestimme zunächst das Minimalpolynom wie oben angegeben:

```
> V1:=vector([0,1,0,0,0]);
```

$$V1 := [0, 1, 0, 0, 0]$$

```
> V11:=map('mod',evalm(A&*V1),5);
```

$$V11 := [0, 2, 0, 1, 2]$$

```
> V12:=map('mod',evalm(A&*V11),5);
```

$$V12 := [1, 4, 0, 3, 3]$$

```
> linindep([V1,V11,V12],5);
```

$$true$$

```
> V13:=map('mod',evalm(A&*V12),5);
```

$$V13 := [2, 0, 0, 4, 3]$$

```
> linindep([V1,V11,V12,V13],5);
```

$$true$$

```
> V14:=map('mod',evalm(A&*V13),5);
```

$$V14 := [2, 2, 4, 1, 0]$$

```
> linindep([V1,V11,V12,V13,V14],5);
```

$$true$$

```
> V15:=map('mod',evalm(A&*V14),5);
```

$$V15 := [0, 3, 4, 3, 2]$$

```
> Nullspace(transpose(matrix([V1,V11,V12,V13,V14,V15])))mod 5;
```

$$\{[4, 3, 1, 3, 4, 1]\}$$

```
> mu:=evalm(matrix([[1,x,x^2,x^3,x^4,x^5]])&*op(%))[1];
```

$$\mu := 4 + 3\,x + x^2 + 3\,x^3 + 4\,x^4 + x^5$$

```
> Factor(mu) mod 5;
```

$$(x^2 + x + 1)\,(x^2 + x + 2)\,(x + 2)$$

Damit erhalten wir das Minimalpolynom wie in der Aufgabenstellung vorgegeben.
Berechne entsprechend:

```
> Nullspace(evalm(A^2+A+array(1..5,1..5,identity)))mod 5;
```

$$\{[3, 2, 1, 0, 1], [0, 2, 0, 1, 0]\}$$

```
> Nullspace(evalm(A^2+A+2*array(1..5,1..5,identity)))mod 5;
```

$$\{[4, 2, 1, 0, 0], [0, 0, 0, 2, 1]\}$$

```
> Nullspace(evalm(A+2*array(1..5,1..5,identity)))mod 5;
```

$$\{[0, 4, 1, 1, 1]\}$$

Aufgabe 3:

gegeben zunächst das Polynom:

```
> p1:=x^5-3*x^4-4*x+12;
```

$$p1 := x^5 - 3\,x^4 - 4\,x + 12$$

Nun können wir die zugehörige Begleitmatrix folgendermaßen ageben:

```
> companion(p1,x);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -12 \\ 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix}$$

```
> factor(p1);
```

$$(x-3)\,(x^2-2)\,(x^2+2)$$

wir sehen: Die Matrix ist nicht diagonalisierbar.

```
> p2:=x^5-4*x^4+x^3+8*x^2-6*x;
```

$$p2 := x^5 - 4\,x^4 + x^3 + 8\,x^2 - 6\,x$$

```
> companion(p2,x);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 6 \\ 0 & 1 & 0 & 0 & -8 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

```
> factor(p2);
```

$$x\,(x-1)\,(x-3)\,(x^2-2)$$

```
> factor(p2,sqrt(2));
```

$$(x-\sqrt{2})\,(x+\sqrt{2})\,(x-3)\,x\,(x-1)$$

Das Polynom zerfällt also nicht über den rationalen Zahlen, wohl aber über den reellen Zahlen vollständig in Linearfaktoren, also ist auch nur dort die Matrix diagonalisierbar.

Aufgabe 4:

Für die gegebene Matrix berechnen wir Minimalpolynom, Eigenwerte und Eigenräume:

```
> alpha:=matrix([[0,1],[-1,0]]);
```

$$\alpha := \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

```
> mu:=minpoly(alpha,x);
```

$$\mu := x^2 + 1$$

```
> eigenvalues(alpha);
```

$$I, -I$$

```
> eigenvectors(alpha);
```

$$[I, 1, \{[1, I]\}], [-I, 1, \{[1, -I]\}]$$

```
>
```

12. Gruppenübung zur Linearen Algebra I

```
> restart;
> with(linalg):
Warning, new definition for norm
Warning, new definition for trace
```

Zuerst definieren wir uns die Hilfsfunktion `linindep` (vgl. 11. Übung), um zu überprüfen, ob ein Liste von Vektoren linear unabhängig (Ausgabe: true) oder linear abhängig (Ausgabe: false) ist. Ein optionales zweites Argument gibt den Modul beim Rechnen in endlichen Körpern an.

```
> linindep:=proc(vecs)
  local r:
  if nargs>1 then
    Gaussjord(matrix(vecs),'r') mod args[2]:
    else
    r:=rank(matrix(vecs))
  fi:
  RETURN(evalb(nops(vecs)=r))
  end;
```

linindep := **proc**(*vecs*)

local *r*;

if 1 < nargs **then** Gaussjord(matrix(*vecs*), '*r*') mod args[2]

else *r* := rank(matrix(*vecs*))

fi;

RETURN(evalb(nops(*vecs*) = *r*))

end

Aufgabe 1:

Zuerst definiere die vorgegebene Matrix:

```
> A:=matrix([[0,-3,0,1],[-1,2,0,-1],[1,3,1,-1],[1,3,0,0]]);
```

$$A := \begin{bmatrix} 0 & -3 & 0 & 1 \\ -1 & 2 & 0 & -1 \\ 1 & 3 & 1 & -1 \\ 1 & 3 & 0 & 0 \end{bmatrix}$$

Nun berechnen wir systematisch das Minimalpolynom nach dem schon in der letzten Übung vorgestellten Verfahren:

```
> V1:=vector([1,0,0,0]);
```

$$V1 := [1, 0, 0, 0]$$

```
> V2:=evalm(A&*V1);
```

$$V2 := [0, -1, 1, 1]$$

```
> linindep([V1,V2]);
```

$$true$$

```
> V3:=evalm(A&*V2);
```

$$V3 := [4, -3, -3, -3]$$

```
> linindep([V1,V2,V3]);
```

true

```
> V4:=evalm(A&*V3);
```

$$V4 := [6, -7, -5, -5]$$

```
> linindep([V1,V2,V3,V4]);
```

false

```
> nullspace(transpose(matrix([V1,V2,V3,V4])));
```

$$\{[2, -1, -2, 1]\}$$

```
> p1:=evalm(matrix([[1,x,x^2,x^3]])&*op(%))[1];
```

$$p1 := 2 - x - 2\,x^2 + x^3$$

```
> V5:=vector([0,0,0,1]);
```

$$V5 := [0, 0, 0, 1]$$

```
> V6:=evalm(A&*V5);
```

$$V6 := [1, -1, -1, 0]$$

```
> linindep([V5,V6]);
```

true

```
> V7:=evalm(A&*V6);
```

$$V7 := [3, -3, -3, -2]$$

```
> linindep([V5,V6,V7]);
```

false

```
> nullspace(transpose(matrix([V5,V6,V7])));
```

$$\{[2, -3, 1]\}$$

```
> p2:=evalm(matrix([[1,x,x^2]])&*op(%))[1];
```

$$p2 := 2 - 3\,x + x^2$$

Damit erhalten wir also das Minimalpolynom:

```
> mu:=expand(lcm(p1,p2));
```

$$\mu := 2 - x - 2\,x^2 + x^3$$

und können die Eigenwerte ablesen:

```
> factor(%);
```

$$(x - 1)\,(x - 2)\,(x + 1)$$

Mit einer Kurzschreibweise für die Einheitsmatrix

```
> I4:=array(1..4,1..4,identity);
```

$$I4 := \mathrm{array}(identity, 1 .. 4, 1 .. 4, [\ ])$$

berechnen wir nun die zugehörigen Eigenräume:

```
> E[1]:=nullspace(A-I4);
```

$$E_1 := \{[0, 0, 1, 0], [1, -1, 0, -2]\}$$

```
> E[2]:=nullspace(A-2*I4);
```

$$E_2 := \{[-1, 1, 1, 1]\}$$

```
> E[-1]:=nullspace(A+I4);
```

$$E_{-1} := \{[-1, 0, 1, 1]\}$$

Wie wir sehen, bilden die Eigenvektoren eine Basis des Raumes:

```
> linindep([op(E[1]),op(E[2]),op(E[-1])]);
```

true

Wir können in diesem Fall das Ergebnis natürlich auch direkt ausrechnen:

```
> minpoly(A,x);
```

$$2 - x - 2\,x^2 + x^3$$

```
> eigenvectors(A);
```

[-1, 1, {[-1, 0, 1, 1]}], [2, 1, {[-1, 1, 1, 1]}], [1, 2, {[0, 0, 1, 0], [-1, 1, 0, 2]}]

Aufgabe 2:

Schreiben wir die gegebene Bilinearform zunächst in der Form für Monome auf:

```
> phi:=(i,j)->int(x^(i-1)*x^(j-1),x=0..1);
```

$$\phi := (i, j) \to \int_0^1 x^{(i-1)}\, x^{(j-1)}\, dx$$

Damit ergibt sich die Grammatrix folgendermaßen:

```
> Phi:=matrix(3,3,phi);
```

$$\Phi := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

Nun berechnen wir unter Benutzung der angegebenen Basis $\Phi(B_1 + 2\,B_2 + B_3, B_1 + B_3)$ und erhalten

```
> evalm([1,2,1]&*Phi&*vector([1,0,1]));
```

$$\frac{101}{30}$$

Um eine Orthogonalbasis bezüglich Φ zu bestimmen, führen wir zuerst das Verfahren der simultanen Zeilen- und Spaltenumformungen durch (hier nur Zeilenumformungen durchgeführt, da die Spaltenumformungen die Basiswechselmatrix nicht mehr verändern)

```
> concat(Phi,array(1..3,1..3,identity));
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & 0 & 1 & 0 \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & 0 & 0 & 1 \end{bmatrix}$$

```
> gausselim(%,3);
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & 1 & 0 & 0 \\ 0 & \frac{1}{12} & \frac{1}{12} & \frac{-1}{2} & 1 & 0 \\ 0 & 0 & \frac{1}{180} & \frac{1}{6} & -1 & 1 \end{bmatrix}$$

Wir erhalten als Basiswechselmatrix zur Orthogonalbasis:

```
> BidBs:=transpose(submatrix(%,1..3,4..6));
```

$$BidBs := \begin{bmatrix} 1 & \frac{-1}{2} & \frac{1}{6} \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

Ausgehend von der Standardbasis können wir das gleiche Ergebnis auch sukzessiv mit dem Gram-Schmidtschen Orthogonalisierungsverfahren berechnen: (Achtung: die Ergebnisvektoren sind hier noch nicht normiert!)

```
> B:=[[1,0,0],[0,1,0],[0,0,1]];
```

$$B := [[1, 0, 0], [0, 1, 0], [0, 0, 1]]$$

```
> Bs:=[],[],[]]:
> Bs[1]:=B[1];
```

$$Bs_1 := [1, 0, 0]$$

```
> Bs[2]:=B[2]-evalm(Bs[1]&*Phi&*B[2])*Bs[1];
```

$$Bs_2 := \left[\frac{-1}{2}, 1, 0\right]$$

```
> Bs[3]:=B[3]-evalm(Bs[1]&*Phi&*B[3])*Bs[1]-(evalm(Bs[2]&*Phi&*B[3
  ]))/evalm(Bs[2]&*Phi&*Bs[2])*Bs[2];
```

$$Bs_3 := \left[\frac{1}{6}, -1, 1\right]$$

```
> Bs;
```

$$\left[[1, 0, 0], \left[\frac{-1}{2}, 1, 0\right], \left[\frac{1}{6}, -1, 1\right]\right]$$

Daraus ergibt sich nun folgende Orthonormalbasis

```
> Bn:=map(i->i/sqrt(evalm(i&*Phi&*i)),Bs);
```

$$Bn := \left[[1, 0, 0], 2\left[\frac{-1}{2}, 1, 0\right]\sqrt{3}, 6\left[\frac{1}{6}, -1, 1\right]\sqrt{5}\right]$$

mit der Eigenschaft (als Überprüfung der Orthonormalität):

```
> matrix(3,3,(i,j)->evalm(Bn[i]&*Phi&*Bn[j]));
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Aufgabe 3:

Zur gegebenen Matrix

```
> phi:=matrix([[-1,1,-1,0,0],[-1,1,-1,1,0],[0,0,-1,1,0],[0$4,1],[-
  2,3,-1,0,-2]]);
```

$$\phi := \begin{bmatrix} -1 & 1 & -1 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -2 & 3 & -1 & 0 & -2 \end{bmatrix}$$

bestimmen wir zunächst das Minimalpolynom:

```
> minpoly(phi,x);
```

$$-1 - 3\,x - 2\,x^2 + 2\,x^3 + 3\,x^4 + x^5$$

```
> mu:=factor(%);
```

$$\mu := (x-1)\,(x+1)^4$$

Der Eigenraum zum Eigenwert 1 ergibt sich dann zu

```
> nullspace(phi-array(1..5,1..5,identity));
```

$$\{[1, 3, 1, 2, 2]\}$$

wobei wir die zweite Komponente der Zerlegung nun sowohl als Kern als auch als Bild bestimmen können (vgl. Vorlesung)

```
> V2:=colspace(phi-array(1..5,1..5,identity));
```

$$V2 := \{[1, 0, 0, 0, 3], [0, 1, 0, 0, -4], [0, 0, 0, 1, -3], [0, 0, 1, 0, 1]\}$$

```
> V3:=nullspace(evalm((phi+array(1..5,1..5,identity))^4));
```

$$V3 := \{[1, 0, 0, 0, 3], [0, 1, 0, 0, -4], [0, 0, 0, 1, -3], [0, 0, 1, 0, 1]\}$$

und erkennen, dass wir den gleichen Raum erhalten:

```
> intbasis(V2,V3);
```

$$\{[1, 0, 0, 0, 3], [0, 1, 0, 0, -4], [0, 0, 0, 1, -3], [0, 0, 1, 0, 1]\}$$

Aufgabe 4:

```
> A:=matrix([[1$5],[0,0,1,0,1],[0,1,0,0,0],[1,1,0,0,0],[0,1,1,0,0]
  ]);
```

$$A := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Zuerst berechnen wir wie gehabt das Minimalpolynom der gegebenen Matrix (vgl. 11. Übung):

```
> V1:=vector([1,0,0,0,0]);
```

$$V1 := [1, 0, 0, 0, 0]$$

```
> V11:=map('mod',evalm(A&*V1),2);
```

$$V11 := [1, 0, 0, 1, 0]$$

```
> V12:=map('mod',evalm(A&*V11),2);
```

$$V12 := [0, 0, 0, 1, 0]$$

```
> linindep([V1,V11,V12],2);
```

$$false$$

```
> Nullspace(transpose(matrix([V1,V11,V12])))mod 2;
```

$$\{[1, 1, 1]\}$$

```
> p1:=evalm(matrix([[1,x,x^2]])&*op(%))[1];
```

$$p1 := 1 + x + x^2$$

```
> V2:=vector([0,1,0,0,0]);
```

$$V2 := [0, 1, 0, 0, 0]$$

```
> V21:=map('mod',evalm(A&*V2),2);
```

$$V21 := [1, 0, 1, 1, 1]$$

```
> V22:=map('mod',evalm(A&*V21),2);
```

$$V22 := [0, 0, 0, 1, 1]$$

```
> linindep([V2,V21,V22],2);
```

$$true$$

```
> V23:=map('mod',evalm(A&*V22),2);
```

$$V23 := [0, 1, 0, 0, 0]$$

```
> linindep([V2,V21,V22,V23],2);
```

$$false$$

```
> Nullspace(transpose(matrix([V2,V21,V22,V23])))mod 2;
```

$$\{[1, 0, 0, 1]\}$$

```
> p2:=evalm(matrix([[1,x,x^2,x^3]])&*op(%))[1];
```

$$p2 := 1 + x^3$$

```
> linindep([V1,V11,V2,V21,V22],2);
```

$$true$$

```
> mu:=expand(Lcm(p1,p2) mod 2);
```

$$\mu := 1 + x^3$$

```
> Factor(mu) mod 2;
```

$$(1 + x + x^2)\,(x + 1)$$

Damit ergibt sich nun die folgende Zerlegung des Raumes mit Hilfe des Eigenraumes bzw. erweiterten Eigenraumes (den wir auch wiederum als Bild berechnen können).

```
> I5:=array(1..5,1..5,identity);
```

$$I5 := \mathrm{array}(identity, 1 .. 5, 1 .. 5, [\ ])$$

```
> V_1:=Nullspace(evalm(A-I5)) mod 2;
```

$$V\_1 := \{[1, 1, 1, 0, 0]\}$$

```
> V_2:=Nullspace(evalm(A^2+A+I5)) mod 2;
```

$$V\_2 := \{[0, 0, 0, 1, 0], [0, 0, 1, 0, 0], [0, 1, 0, 0, 1], [1, 0, 0, 0, 0]\}$$

```
> transpose(Gaussjord(transpose(evalm(A-I5))) mod 2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Konstruieren wir nun die Basis wie im Beweis angegeben...

```
> B:=[[]$5];
```

$$B := [[\ ], [\ ], [\ ], [\ ], [\ ]]$$

```
> B[1]:=op(V_1);
```

$$B_1 := [1, 1, 1, 0, 0]$$

```
> B[2]:=V_2[1];
```

$$B_2 := [1, 0, 0, 0, 0]$$

```
> B[3]:=map('mod',evalm(A&*B[2]),2);
```

$$B_3 := [1, 0, 0, 1, 0]$$

```
> B[4]:=V_2[3];
```

$$B_4 := [0, 0, 1, 0, 0]$$

```
> B[5]:=map('mod',evalm(A&*B[4]),2);
```

$$B_5 := [1, 1, 0, 0, 1]$$

mit der Basiswechselmatrix...

```
> T:=transpose(matrix(B));
```

$$T := \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

und erhalten schließlich als Darstellung unserer Matrix in der Ausgangsbasis:

```
> map('mod',evalm((Inverse(T) mod 2)&*A&*T),2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

```
>
```

13. Gruppenübung zur Linearen Algebra I

```
> with(linalg):
```

Aufgabe 1:

Gegeben sind die folgenden Matrizen:

```
> Phi:=matrix([[2,1,1],[1,2,1],[1,1,2]]);
```

$$\Phi := \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

```
> phi:=matrix([[-1,-1,-2],[-1,-1,-2],[3,3,10]]);
```

$$\phi := \begin{bmatrix} -1 & -1 & -2 \\ -1 & -1 & -2 \\ 3 & 3 & 10 \end{bmatrix}$$

```
> psi:=matrix([[-1,-2,-1],[3,10,3],[-1,-2,-1]]);
```

$$\psi := \begin{bmatrix} -1 & -2 & -1 \\ 3 & 10 & 3 \\ -1 & -2 & -1 \end{bmatrix}$$

Rechne zunächst direkt nach, obϕ und ψ selbstadjungiert sind:

```
> phi_ad:=evalm(inverse(Phi)&*transpose(phi)&*Phi);
```

$$phi\_ad := \begin{bmatrix} -1 & -1 & -2 \\ -1 & -1 & -2 \\ 3 & 3 & 10 \end{bmatrix}$$

```
> psi_ad:=evalm(inverse(Phi)&*transpose(psi)&*Phi);
```

$$psi\_ad := \begin{bmatrix} -1 & -2 & -1 \\ 3 & 10 & 3 \\ -1 & -2 & -1 \end{bmatrix}$$

oder berechne einfacher (also ohne Matrixinversion):

```
> evalm(Phi&*phi=transpose(phi)&*Phi);
```

$$\begin{bmatrix} 0 & 0 & 4 \\ 0 & 0 & 4 \\ 4 & 4 & 16 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 \\ 0 & 0 & 4 \\ 4 & 4 & 16 \end{bmatrix}$$

```
> evalm(Phi&*psi=transpose(psi)&*Phi);
```

$$\begin{bmatrix} 0 & 4 & 0 \\ 4 & 16 & 4 \\ 0 & 4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 4 & 0 \\ 4 & 16 & 4 \\ 0 & 4 & 0 \end{bmatrix}$$

...was zum gleichen Ergebnis führt.

Aufgabe 2:

Gegeben die Matrix:

```
> A:=matrix([[3,-2,-1],[-2,3,-1],[-1,-1,9/2]]);
```

$$A := \begin{bmatrix} 3 & -2 & -1 \\ -2 & 3 & -1 \\ -1 & -1 & \frac{9}{2} \end{bmatrix}$$

Berechne zunächst Minimalpolynom und Eigenwerte mit zugehörigen Eigenräumen:

```
> minpoly(A,x);
```

$$\frac{5}{2} - \frac{11}{2}x + x^2$$

```
> ER:=eigenvectors(A);
```

$$ER := \left[\frac{1}{2}, 1, \left\{\left[1, 1, \frac{1}{2}\right]\right\}\right], [5, 2, \{[0, 1, -2], [1, 0, -2]\}]$$

Nun stellen wir eine Orthogonalbasis auf: den ersten Eigenvektor können wir gleich übernehmen, den zweiten Eigenraum müssen wir noch m,it dem Gram-Schmidt-Verfahren orthogonalisieren:

```
> v1:=convert(op(ER[1,3]),list);
```

$$v1 := \left[1, 1, \frac{1}{2}\right]$$

```
> v2,v3:=op(GramSchmidt(ER[2,3]));
```

$$v2, v3 := [1, 0, -2], \left[\frac{-4}{5}, 1, \frac{-2}{5}\right]$$

Durch Normieren erhalten wir nun eine Orthonormalbasis:

```
> v:=map(w->expand(w/norm(w,2)),[v1,v2,v3]);
```

$$v := \left[\left[\frac{2}{3}, \frac{2}{3}, \frac{1}{3}\right], \left[\frac{1}{5}\sqrt{5}, 0, -\frac{2}{5}\sqrt{5}\right], \left[-\frac{4}{15}\sqrt{5}, \frac{1}{3}\sqrt{5}, -\frac{2}{15}\sqrt{5}\right]\right]$$

mit der dazugehörigen Transformationsmatrix:

```
> U:=transpose(convert(v,matrix));
```

$$U := \begin{bmatrix} \frac{2}{3} & \frac{1}{5}\sqrt{5} & -\frac{4}{15}\sqrt{5} \\ \frac{2}{3} & 0 & \frac{1}{3}\sqrt{5} \\ \frac{1}{3} & -\frac{2}{5}\sqrt{5} & -\frac{2}{15}\sqrt{5} \end{bmatrix}$$

Wie wir sehen, ist die Matrix orthogonal

```
> evalm(U&*transpose(U));
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

und diagonalisiert die Ausgangsmatrix.

```
> evalm(transpose(U)&*A&*U);
```

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

>

14. Gruppenübung zur Linearen Algebra I

```
> with(linalg):
```

Aufgabe 1:

```
> B:=matrix([[2,1,0],[1,0,2],[0,2,1]]);
```

$$B := \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

Wie wir sehen, ist *B* symmetrisch und damit selbstadjungiert, d.h. wir können eine Hauptachsentransformation durchführen (vgl. 13. Übung, Aufgabe 2):

Hierzu berechne zunächst Minimalpolynom, Eigenwerte und Eigenräume:

```
> minpoly(B,x);
```

$$9 - 3x - 3x^2 + x^3$$

```
> ER:=eigenvectors(B);
```

$$ER := [3,\ 1,\ \{[1,\ 1,\ 1]\}],\left[\sqrt{3},\ 1,\ \left\{\left[-\frac{1}{2}-\frac{1}{2}\sqrt{3},\ \frac{1}{2}\sqrt{3}-\frac{1}{2},\ 1\right]\right\}\right],$$
$$\left[-\sqrt{3},\ 1,\ \left\{\left[\frac{1}{2}\sqrt{3}-\frac{1}{2},\ -\frac{1}{2}-\frac{1}{2}\sqrt{3},\ 1\right]\right\}\right]$$

Damit erhalten wir die Orthogonalbasis

```
> OG:=[ER[1,3,1],ER[2,3,1],ER[3,3,1]];
```

$$OG := \left[[1,\ 1,\ 1],\left[-\frac{1}{2}-\frac{1}{2}\sqrt{3},\ \frac{1}{2}\sqrt{3}-\frac{1}{2},\ 1\right],\left[\frac{1}{2}\sqrt{3}-\frac{1}{2},\ -\frac{1}{2}-\frac{1}{2}\sqrt{3},\ 1\right]\right]$$

die sich zu folgender Orthonormalbasis normiert:

```
> ON:=(map(v->simplify(evalm(1/norm(v,2)*v)),OG));
```

$$ON := \left[\left[\frac{1}{3}\sqrt{3},\ \frac{1}{3}\sqrt{3},\ \frac{1}{3}\sqrt{3}\right],\left[-\frac{1}{6}\sqrt{3}\,(1+\sqrt{3}),\ \frac{1}{6}\sqrt{3}\,(-1+\sqrt{3}),\ \frac{1}{3}\sqrt{3}\right],\right.$$
$$\left.\left[\frac{1}{6}\sqrt{3}\,(-1+\sqrt{3}),\ -\frac{1}{6}\sqrt{3}\,(1+\sqrt{3}),\ \frac{1}{3}\sqrt{3}\right]\right]$$

mit folgender orthogonaler Transformationsmatrix:

```
> U:=transpose(matrix(ON));
```

$$U := \begin{bmatrix} \frac{1}{3}\sqrt{3} & -\frac{1}{6}\sqrt{3}\,(1+\sqrt{3}) & \frac{1}{6}\sqrt{3}\,(-1+\sqrt{3}) \\ \frac{1}{3}\sqrt{3} & \frac{1}{6}\sqrt{3}\,(-1+\sqrt{3}) & -\frac{1}{6}\sqrt{3}\,(1+\sqrt{3}) \\ \frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3} \end{bmatrix}$$

```
> simplify(evalm(U&*transpose(U)));
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

und der Hauptachsentransformation

```
> DI:=simplify(evalm(transpose(U)&*B&*U));
```

$$DI := \begin{bmatrix} 3 & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & -\sqrt{3} \end{bmatrix}$$

Setze nun:

```
> DA:=map('abs',DI);
```

$$DA := \begin{bmatrix} 3 & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$$

Damit erfüllen nun die folgenden Matrizen die gewünschten Bedingungen:

```
> A:=simplify(evalm(U&*DA&*transpose(U)));
```

$$A := \begin{bmatrix} 1+\frac{2}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} \\ 1-\frac{1}{3}\sqrt{3} & 1+\frac{2}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} \\ 1-\frac{1}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} & 1+\frac{2}{3}\sqrt{3} \end{bmatrix}$$

ist symmetrisch und (nach Konstruktion von *DA*) positiv definit, und

```
> g:=evalm(inverse(A)&*B);
```

$$g := \begin{bmatrix} \frac{1}{3}\sqrt{3}+\frac{1}{3} & \frac{1}{3} & \frac{1}{3}-\frac{1}{3}\sqrt{3} \\ \frac{1}{3} & \frac{1}{3}-\frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3}+\frac{1}{3} \\ \frac{1}{3}-\frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3}+\frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

ist orthogonal:

```
> simplify(evalm(g&*transpose(g)));
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Damit gilt nun also:

```
> simplify(evalm(A&*g));
```

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

```
> simplify(evalm(g&*A));
```

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

2. Verfahren: Berechne eine positive Quadratwurzel der Matrix $B\,B^{tr}$ (vgl. 13. Übung, Aufgabe 3)

```
> C:=evalm(B&*transpose(B));
```

$$C := \begin{bmatrix} 5 & 2 & 2 \\ 2 & 5 & 2 \\ 2 & 2 & 5 \end{bmatrix}$$

Berechne nun eine Orthonormalbasis wie oben:

```
> minpoly(C,x);
```

$$27 - 12\,x + x^2$$

```
> ER:=eigenvectors(C);
```

$$ER := [9, 1, \{[1, 1, 1]\}], [3, 2, \{[-1, 1, 0], [-1, 0, 1]\}]$$

```
> v1:=ER[1,3,1];
```

$$v1 := [1, 1, 1]$$

```
> v2,v3:=op(GramSchmidt(ER[2,3]));
```

$$v2, v3 := \left[\frac{-1}{2}, \frac{-1}{2}, 1\right], [-1, 1, 0]$$

```
> ON:=map(v->evalm(v/norm(v,2)),[v1,v2,v3]);
```

$$ON := \left[\left[\frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}\right], \left[-\frac{1}{6}\sqrt{6}, -\frac{1}{6}\sqrt{6}, \frac{1}{3}\sqrt{6}\right], \left[-\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}, 0\right]\right]$$

Daraus ergibt sich wieder die orthogonale Matrix

```
> U:=transpose(matrix(ON));
```

$$U := \begin{bmatrix} \frac{1}{3}\sqrt{3} & -\frac{1}{6}\sqrt{6} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{3}\sqrt{3} & -\frac{1}{6}\sqrt{6} & \frac{1}{2}\sqrt{2} \\ \frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{6} & 0 \end{bmatrix}$$

```
> evalm(U&*transpose(U));
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Die Ausgangsmatrix wird folgendermaßen diagonalisiert:

```
> DI:=evalm(transpose(U)&*C&*U);
```

$$DI := \begin{bmatrix} 9 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Aus den Diagonaleinträgen können wir nun die Wurzeln ziehen und erhalten:

```
> DW:=map('sqrt',DI);
```

$$DW := \begin{bmatrix} 3 & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$$

Also setzen wir damit nun die Matrix *A* wir folgt fest (durch Rücktransformation der Wurzelmatrix):

```
> AA:=evalm(U&*DW&*transpose(U));
```

$$AA := \begin{bmatrix} 1+\frac{2}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} \\ 1-\frac{1}{3}\sqrt{3} & 1+\frac{2}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} \\ 1-\frac{1}{3}\sqrt{3} & 1-\frac{1}{3}\sqrt{3} & 1+\frac{2}{3}\sqrt{3} \end{bmatrix}$$

Wir erkennen, dass wir die gleiche Matrix erhalten haben wie im ersten Verfahren, und damit gilt auch das gleiche *g*.

```
> evalm(AA-A);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Aufgabe 3:

```
> alpha:=matrix([[1,1,0,0],[2,1,2,1],[0,1,1,0]]);
```

$$\alpha := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 2 & 1 & 2 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Nach Aufgabenstellung hat die Basiswechselmatrix die Gestalt

```
> A:=matrix([[1,2,1],[1,1,1],[1,2,0]]);
```

$$A := \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

```
> rank(Gaussjord(A) mod 3);
```

$$3$$

Da die Matrix vollen Rang hat, ist wieder eine Basis entstanden, und die gesuchte Matrix ergibt sich als

```
> map(‘mod‘,evalm(transpose(alpha)&*A),3);
```

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 2 \\ 0 & 1 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

```
>
```

15. Gruppenübung zur Linearen Algebra I

```
> with(linalg):
```

Aufgabe 1:

Definiere zunächst die gegebenen Matrizen:

```
> A:=matrix([[x+2,1,0],[2,x,1],[1,1,x+1]]);
```

$$A := \begin{bmatrix} x+2 & 1 & 0 \\ 2 & x & 1 \\ 1 & 1 & x+1 \end{bmatrix}$$

Nun können wir die Determinante berechnen (beachte den jeweiligen Restklassenkörper):

```
> Det(A) mod 3;
```

$$x^3 + 2\,x$$

```
> B:=matrix([[0,0,1,0,1],[1,0,0,1,0],[0,1$4],[1,1,0,1,1],[0,1,1,1,
  0]]);
```

$$B := \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

```
> Det(B) mod 2;
```

$$1$$

```
> C:=matrix([[2,1,1],[1,2,1],[1,1,2]]);
```

$$C := \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

analog für eine Determinante über den reellen (bzw. komplexen) Zahlen

```
> det(C);
```

$$4$$

```
>
```