Module 14 Lineaire algebra

Onderwerp	Stelsels lineaire vergelijkingen: Gauss(-Jordan)-eliminatie; Gram-Schmidt; Eigenwaarden, eigenvectoren.
Voorkennis	Lineaire algebra.
Expressies	`GenerateMatrix`, `LinearSolve`, `ReducedRowEchelonForm`, `GaussianElimination`, `Rank`, `LUDecomposition`, `NullSpace`, `RowSpace`, `ColumnSpace`, `Basis`, `SumBasis`, `IntersectionBasis`, `GramSchmidt`, `CharaceristicPolynomial`, `Eigenvalues`, `Eigenvectors`, `JordanForm`
Bibliotheken	`LinearAlgebra`
Zie ook	Module 5, 13, 15.

14.1 Enige definities, notaties en stellingen

In deze paragraaf wordt een overzicht gegeven van die begrippen uit de lineaire algebra die in de voorbeelden en opgaven van deze module worden gebruikt.

Lineaire deelruimte, lineaire variëteit. De vector $\mathbf{v}$ is een *lineaire combinatie* van de vectoren $\mathbf{v}_1, \ldots, \mathbf{v}_n$ als

$$\mathbf{v} = \alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n\,,$$

met $\alpha_i \in \mathbb{R}$, $i = 1 \ldots n$.
$D \subset \mathbb{R}^n$ is een *lineaire deelruimte* als alle lineaire combinaties van vectoren in D weer element van D zijn. Een lineaire deelruimte $D \subset \mathbb{R}^n$ wordt voortgebracht door $\mathbf{v}_1, \ldots, \mathbf{v}_k$ als elke $\mathbf{v} \in D$ te schrijven is als lineaire combinatie van $\mathbf{v}_1, \ldots, \mathbf{v}_k$. We noteren $D = \langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$.
De verzameling

$$\mathbf{a} + D = \left\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \mathbf{a} + \mathbf{v},\ \text{met } \mathbf{v} \in D \right\}$$

is een *lineaire variëteit* als D een lineaire deelruimte is.
Als $D_1 \subset \mathbb{R}^n$ en $D_2 \subset \mathbb{R}^n$ lineaire ruimten zijn, dan is de *somruimte* $D_1 + D_2$ als de verzameling van alle $\mathbf{u} + \mathbf{v}$ met $\mathbf{u} \in D_1$ en $\mathbf{v} \in D_2$.
De *kolomruimte* van matrix A, aangegeven met $\mathrm{Kol}(A)$, is de lineaire ruimte die wordt voortgebracht door de kolommen van A; de *rijruimte*, aangegeven met $\mathrm{Rij}(A)$, is de lineaire ruimte die wordt voortgebracht door de rijen van A.

Een *basis* van een lineaire ruimte is een verzameling van een minimaal aantal voortbrengers van deze ruimte. De *dimensie* is gedefinieerd als het aantal elementen in een basis. De representatie van een vector $\mathbf{v}$ ten opzichte van een basis E geven we aan met $[\mathbf{v}]_E$. De *standaardbasis* van $\mathbb{R}^n$ is $\{(1, 0, 0, \dots),\ (0, 1, 0, \dots),\ \dots\}$.

Lineaire afbeeldingen. De afbeelding $F : \mathbb{R}^n \to \mathbb{R}^k$ is *lineair* als

$$F(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha F(\mathbf{x}) + \beta F(\mathbf{y})$$

voor alle $\mathbf{x}$ en $\mathbf{y} \in \mathbb{R}^n$ en alle α en $\beta \in \mathbb{R}$. Bij elke lineaire afbeelding bestaat een *representatiematrix*[36] A zodat $F(\mathbf{x}) = A\mathbf{x}$. De matrix A wordt meestal genoteerd als $[F]$.

Stelsels lineaire vergelijkingen. Een stelsel van k lineaire vergelijkingen met n onbekenden kan worden geschreven in de vorm $A\mathbf{x} = \mathbf{b}$. Hierin is A de $k \times n$-coëfficiëntenmatrix, $\mathbf{b} \in \mathbb{R}^k$ de vector van rechterleden en $\mathbf{x} \in \mathbb{R}^n$ de vector van onbekenden. De *aangevulde matrix* A^{a} wordt gegeven door $A^{\mathrm{a}} = [A \,|\, \mathbf{b}]$, dat is de matrix A waaraan $\mathbf{b}$ als laatste kolom is toegevoegd.
Er geldt dat het stelsel een oplossing heeft als $\mathrm{rang}(A) = \mathrm{rang}(A^{\mathrm{a}})$; de dimensie van de oplossingsvariëteit is dan $n - \mathrm{rang}(A)$. Als $\mathrm{rang}(A) \neq \mathrm{rang}(A^{\mathrm{a}})$ dan heeft het stelsel geen oplossing.
De *kern* (of *nulruimte*) van een matrix, aangegeven met $\mathrm{Ker}\, A$, is de oplossingsverzameling van het stelsel $A\mathbf{x} = \mathbf{0}$.

Gauss(-Jordan)-eliminatie. Het stelsel $A\mathbf{x} = \mathbf{b}$ heeft dezelfde oplossing als het stelsel $C\mathbf{x} = \mathbf{c}$ als de matrix $[C \,|\, \mathbf{c}]$ door elementaire rijoperaties (zie §13.4) is verkregen uit $[A \,|\, \mathbf{b}]$. Een matrix is in *standaardrijvorm* als elke rij met méér nullen begint dan de rij erboven (Gauss-eliminatie van het bijbehorende stelsel). Het eerste getal in een niet-nulrij heet het *hoofdelement* van die rij. Als bovendien elke kolom die een hoofdelement bevat een standaardbasisvector is (zo'n kolom heet een *elementaire kolom*), dan is de matrix in *canonieke rijvorm* (Gauss-Jordan-eliminatie).

LU-ontbinding. Elke $k \times n$-matrix A kan worden geschreven als het product van een *permutatiematrix* P, een $k \times k$-onderdriehoeksmatrix L en een standaardrijvorm U van A. Een permutatiematrix is een vierkante matrix, met (allemaal verschillende) eenheidsvectoren als kolommen, PA zet dus de rijen van A in een andere volgorde.

Eigenwaarden, eigenvectoren. Zij A een $n \times n$-matrix. Een $\lambda \in \mathbb{C}$ waarvoor geldt dat $A\mathbf{v} = \lambda\mathbf{v}$ voor een $\mathbf{v} \neq \mathbf{0}$ is een *eigenwaarde* van

[36] Als niet vermeld wordt ten opzichte van welke basis F door A wordt gerepresenteerd, dan wordt de standaardbasis bedoeld.

A. Een bijbehorende vector $\mathbf{v}$ heet *eigenvector* van A. De verzameling van alle eigenvectoren bij eigenwaarde λ is een lineaire ruimte en wordt genoteerd als E_λ.
Eigenwaarden zijn nulpunten van $\det(A - \lambda I_n)$ (opgevat als functie van λ), de *karakteristieke polynoom* van A. De multipliciteit van λ als nulpunt van de karakteristieke polynoom heet *algebraïsche multipliciteit*; de dimensie van de bijbehorende eigenruimte E_λ heet de *geometrische multipliciteit* van de eigenwaarde λ. Een matrix die minstens één eigenwaarde heeft met geometrische multipliciteit kleiner dan de algebraïsche heet *defect*.

De Jordan-normaalvorm. Een *Jordan-normaalvorm* van een vierkante matrix A is een matrix met zogenaamde *Jordanblokken* op de diagonaal. Een $n \times n$-Jordanblok B is een matrix met een eigenwaarde $\lambda \in \mathbb{C}$ op de hoofddiagonaal, en 1 op de *superdiagonaal*, dat wil zeggen: $b_{k,k+1} = 1$ voor $k = 1, \ldots n-1$ (dus de 'diagonaal' boven de hoofddiagonaal). Een 1×1-Jordanblok bevat dus alleen het element λ. Een 2×2-, respectievelijk een 3×3-Jordanblok hebben de gedaante

$$\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix}, \quad \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}, \quad \text{enzovoort.}$$

Bij elke vierkante matrix is er een inverteerbare matrix Q, zodat $J = Q^{-1}AQ$ een Jordan-normaalvorm is. Als A *niet* defect is, dan zijn alle Jordanblokken 1×1-matrices, en J is een diagonaalmatrix van eigenwaarden, en Q een matrix met eigenvectoren als kolommen.

14.2 Oplossen van een stelsel lineaire vergelijkingen

Voor het oplossen van een stelsel vergelijkingen kunnen we het commando `solve` gebruiken, zie §5.4. Als het stelsel *lineair* is kan beter het commando `LinearSolve` uit de `LinearAlgebra`-bibliotheek worden gebruikt. Hierbij geeft `LinearSolve(A,b)` de oplossingsverzameling van het stelsel $A\mathbf{x} = \mathbf{b}$. Als er geen oplossing is geeft `LinearSolve`, net als `solve`, `NULL` (dus niets) terug.

`LinearSolve`

Voorbeeldopgave

Geef de oplossingsvariëteit van het stelsel

$$\begin{cases} 3x_1 + 8x_2 - 18x_3 + x_4 = 7 \\ x_1 + 2x_2 - 4x_3 = 5 \\ x_1 + 3x_2 - 7x_3 - x_4 = 4 \end{cases}$$

Voorbeeldsessie

```
> with(LinearAlgebra):
```

Het stelsel vergelijkingen:

```
> stelsel := {3*x1 + 8*x2 - 18*x3 + x4 = 7,
              x1 + 2*x2 -  4*x3      = 5,
              x1 + 3*x2 -  7*x3 - x4 = 4};
```

$$stelsel := \{3\,x1 + 8\,x2 - 18\,x3 + x4 = 7,\ x1 + 2\,x2 - 4\,x3 = 5,\ x1 + 3\,x2 - 7\,x3 - x4 = 4\}$$

```
> solve( stelsel, {x1,x2,x3,x4} );
```

$$\{x2 = -3 + 3\,x3,\ x4 = -2,\ x1 = 11 - 2\,x3,\ x3 = x3\}$$

Druk **x1**, **x3** en **x4** uit in **x2**:

```
> solve( stelsel, {x1,x3,x4} );
```

$$\{x4 = -2,\ x3 = 1 + \frac{x2}{3},\ x1 = 9 - \frac{2\,x2}{3}\}$$

De coëfficientenmatrix en de rechterleden:

```
> A,b := GenerateMatrix( stelsel, [x1,x2,x3,x4] );
```

$$A, b := \begin{bmatrix} 3 & 8 & -18 & 1 \\ 1 & 2 & -4 & 0 \\ 1 & 3 & -7 & -1 \end{bmatrix}, \begin{bmatrix} 7 \\ 5 \\ 4 \end{bmatrix}$$

```
> LinearSolve(A,b);
```

$$\begin{bmatrix} 11 - 2\,\_t_3 \\ -3 + 3\,\_t_3 \\ \_t_3 \\ -2 \end{bmatrix}$$

```
> LinearSolve(A,b, free='alpha');
```

$$\begin{bmatrix} 11 - 2\,\alpha_3 \\ -3 + 3\,\alpha_3 \\ \alpha_3 \\ -2 \end{bmatrix}$$

Toelichting

Blijkbaar heeft het stelsel een oplosvariëteit van dimensie 1: als we Maple vragen x_1, x_2, x_3 en x_4 op te lossen, worden x_1, x_2 en x_4 in x_3 uitgedrukt. Als we vragen om drie van de vier onbekenden op te lossen, dan worden deze (zo mogelijk) in de vierde uitgedrukt. Merk op dat de oplossing de gedaante heeft van een *verzameling van vergelijkingen*.

`GenerateMatrix` De coëfficiëntenmatrix van het stelsel kunnen we met `GenerateMatrix` maken. Denk er aan dat u de variabelen als een *lijst* opgeeft (dus niet als verzameling), want anders kunnen de kolommen van de matrix door elkaar raken. Maar wees niet verbaasd als de *rijen* van de

matrix in een ander volgorde komen als in het stelsel is opgegeven. Dit stelsel is immers een *verzameling* van vergelijkingen, dus zonder voorgeschreven volgorde van de elementen.
Als we `LinearSolve` gebruiken is direct aan het *aantal* door Maple geïntroduceerde parameters (in dit geval één, namelijk $\_t_3$)[37] te zien wat de dimensie van de oplossingsvariëteit is. Deze wordt dus gegeven door $(11, -3, 0, -2) + \langle(-2, 3, 1, 0)\rangle$. Merk op dat de oplossing wordt gegeven in de vorm van een *vector* met eventueel een of meer parameters. De namen van deze parameters kunnen nog worden gekozen

`free` met de optie `free`. ◇

We geven nog een voorbeeld van een probleem waarbij een stelsel vergelijkingen moet worden opgelost. Het is hetzelfde probleem als in §13.5 (blz. 175) met `solve` is aangepakt.

Voorbeeldopgave

Schrijf de vector $\mathbf{a} = (7, 3)$ als *lineaire combinatie* van de vectoren $\mathbf{v}_1 = (3, 1)$ en $\mathbf{v}_2 = (2, 3)$.

Voorbeeldsessie

```
> with(LinearAlgebra):
> v1 := <3,1>:  v2 := <2,3>:    a := <7,3>:
> A := <v1|v2>;
```

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

```
> v := LinearSolve(A,a);
```

$$v := \begin{bmatrix} \frac{15}{7} \\ \frac{2}{7} \end{bmatrix}$$

Controle:

```
> A.v;
```

$$\begin{bmatrix} 7 \\ 3 \end{bmatrix}$$

[37]Namen die met een *underscore* (_) beginnen, zijn altijd door Maple zélf 'verzonnen'. Voor de gebruiker is het onverstandig om namen te gebruiken die met zo'n underscore beginnen, omdat ze wel eens in conflict kunnen raken met door Maple gehanteerde interne variabelen.

Toelichting

In §13.5 hebben we de vectorvergelijking $\alpha\mathbf{v}_1 + \beta\mathbf{v}_2 = \mathbf{a}$ omgezet in een stelsel van twee (lineaire) vergelijkingen.
Een andere mogelijkheid is dus om $\mathbf{v}_1$ en $\mathbf{v}_2$ op te vatten als de kolommen van de matrix A. De gevraagde (α, β) is dan de oplossing van de vergelijking $A\mathbf{x} = \mathbf{a}$. $\diamond$

14.3 Gauss(-Jordan)-eliminatie

`ReducedRow...` `GaussianEli...` `FractionFree`

Na het commando `ReducedRowEchelonForm` brengt Maple een matrix op de *canonieke rijvorm*. Voor een *standaardrijvorm* kan gebruik gemaakt worden van `GaussianElimination`. Als men per se niet wil dat er breuken in de standaardrijvorm komen, moet bij `GaussianElimination` de optie `method=FractionFree` worden gebruikt. Zo'n standaardrijvorm is vooral nuttig bij het analyseren van een stelsel vergelijkingen waarin parameters in de coëfficiënten voorkomen.

Voorbeeldopgave

Geef voor alle waarden van α, β en λ de oplossingsverzameling van het stelsel

$$\begin{cases} 3x_1 + x_2 + 7x_3 = 1 \\ x_1 + 2x_3 + \beta x_4 = 1 \\ 2x_1 - x_2 + \alpha x_3 - 5x_4 = \lambda \end{cases}$$

Voorbeeldsessie

```
> with(LinearAlgebra):
> A := Matrix([[3,1,7,0],[1,0,2,beta],[2,-1,alpha,-5]]);
```

$$A := \begin{bmatrix} 3 & 1 & 7 & 0 \\ 1 & 0 & 2 & \beta \\ 2 & -1 & \alpha & -5 \end{bmatrix}$$

```
> b := <1,1,lambda>;
```

$$b := \begin{bmatrix} 1 \\ 1 \\ \lambda \end{bmatrix}$$

Canonieke rijvorm van de aangevulde matrx:

```
> ReducedRowEchelonForm(<A|b>);
```

$$\begin{bmatrix} 1 & 0 & 0 & \dfrac{10+7\beta+\beta\,\alpha}{\alpha-3} & -\dfrac{-\alpha-5+2\lambda}{\alpha-3} \\ 0 & 1 & 0 & -\dfrac{3\beta\,\alpha-14\beta-5}{\alpha-3} & -\dfrac{2\alpha-10+\lambda}{\alpha-3} \\ 0 & 0 & 1 & -\dfrac{5(1+\beta)}{\alpha-3} & \dfrac{\lambda-4}{\alpha-3} \end{bmatrix}$$

```
> GaussianElimination(<A|b>);
```

$$\begin{bmatrix} 3 & 1 & 7 & 0 & 1 \\ 0 & \dfrac{-1}{3} & \dfrac{-1}{3} & \beta & \dfrac{2}{3} \\ 0 & 0 & \alpha-3 & -5-5\beta & \lambda-4 \end{bmatrix}$$

```
> GaussianElimination(<A|b>, method=FractionFree);
```

$$\begin{bmatrix} 3 & 1 & 7 & 0 & 1 \\ 0 & -1 & -1 & 3\beta & 2 \\ 0 & 0 & -\alpha+3 & 5+5\beta & -\lambda+4 \end{bmatrix}$$

```
> GaussianElimination( subs({alpha=3,beta=-1}, <A|b>) );
```

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

Eerste geval: $\alpha = 3$, $\beta = -1$ en $\lambda = 4$:

```
> LinearSolve( subs({alpha=3,beta=-1}, A),
  subs({lambda=4}, b) );
```

$$\begin{bmatrix} 1-2\_t_3+\_t_4 \\ -2-\_t_3-3\_t_4 \\ \_t_3 \\ \_t_4 \end{bmatrix}$$

Tweede geval: $\alpha = 3$, $\beta = -1$ en $\lambda \neq 4$:

```
> opl := LinearSolve( subs({alpha=3,beta=-1}, A), b );
```

```
Error, (in LinearAlgebra:-LA_Main:-LinearSolve) inconsistent
system
```

(inderdaad geen oplossingen)

Overige gevallen:

```
> LinearSolve(A,b);
```

$$\begin{bmatrix} \dfrac{1}{5}\,\dfrac{-10\_t1_3-7\_t1_3\beta-\beta\,\alpha\_t1_3+\beta+\beta\lambda+5}{1+\beta} \\ -\dfrac{1}{5}\,\dfrac{5\_t1_3+14\_t1_3\beta-3\beta\,\alpha\_t1_3-2\beta+3\beta\lambda+10}{1+\beta} \\ \_t1_3 \\ -\dfrac{1}{5}\,\dfrac{-\alpha\_t1_3+3\_t1_3-4+\lambda}{1+\beta} \end{bmatrix}$$

Toelichting

De aangevulde coëfficiëntenmatrix A^{a} wordt met `<A|b>` gemaakt. Na het commando `ReducedRowEchelonForm(<A|b>)` lijkt op het eerste gezicht de rang van A^{a} 3 te zijn: er zijn drie elementaire kolommen in de canonieke rijvorm. Het commando `Rank(<A|b>)` (niet uitgevoerd) zou trouwens ook het antwoord 3 hebben opgeleverd. Een nadere beschouwing leert echter dat er in de overige kolommen expressies voorkomen met $\alpha - 3$ in de noemer. Voor $\alpha = 3$ kan het antwoord dus niet juist zijn.

`Rank`

Het resultaat van `GaussianElimination` en biedt meer informatie. Hieruit is direct af te lezen dat:

- $\mathrm{rang}(A) = \mathrm{rang}(A^{\mathrm{a}}) = 2$ als $\alpha = 3,\ \beta = -1,\ \lambda = 4$, dus in dat geval heeft het stelsel een oplossing met dimensie 2;
- $\mathrm{rang}(A) < \mathrm{rang}(A^{\mathrm{a}})$ als $\alpha = 3,\ \beta = -1,\ \lambda \neq 4$, dus in dat geval is het stelsel niet oplosbaar;
- in alle overige gevallen is $\mathrm{rang}(A) = \mathrm{rang}(A^{\mathrm{a}}) = 3$, en heeft het stelsel dus een oplossing met dimensie 1.

Met `LinearSolve` is het antwoord voor de verschillende gevallen berekend. ◇

14.4 LU-ontbinding

`LUDecomposition`

Hiervoor hebben we het commando `LUDecomposition`. We geven een voorbeeld om te laten zien hoe het werkt.

Voorbeeldsessie

```
> with(LinearAlgebra):

> A := Matrix([[3,8,-18,1],[1,2,-4,0],
              [1,3,-7,-1],[2,5,-11,-1]]);
```

$$A := \begin{bmatrix} 3 & 8 & -18 & 1 \\ 1 & 2 & -4 & 0 \\ 1 & 3 & -7 & -1 \\ 2 & 5 & -11 & -1 \end{bmatrix}$$

```
> P,L,U := LUDecomposition(A);
```

$$P, L, U := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{3} & \frac{-1}{2} & 1 & 0 \\ \frac{2}{3} & \frac{1}{2} & 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 3 & 8 & -18 & 1 \\ 0 & \frac{-2}{3} & 2 & \frac{-1}{3} \\ 0 & 0 & 0 & \frac{-3}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> GaussianElimination(A);
```

$$\begin{bmatrix} 3 & 8 & -18 & 1 \\ 0 & \frac{-2}{3} & 2 & \frac{-1}{3} \\ 0 & 0 & 0 & \frac{-3}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
> B := A^%T;
```

$$B := \begin{bmatrix} 3 & 1 & 1 & 2 \\ 8 & 2 & 3 & 5 \\ -18 & -4 & -7 & -11 \\ 1 & 0 & -1 & -1 \end{bmatrix}$$

```
> LUDecomposition(B);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{8}{3} & 1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \\ -6 & -3 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 3 & 1 & 1 & 2 \\ 0 & \frac{-2}{3} & \frac{1}{3} & \frac{-1}{3} \\ 0 & 0 & \frac{-3}{2} & \frac{-3}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Toelichting

Het commando `LUDecomposition(A)` levert een rijtje van *drie* matrices: de permutatiematrix (die in het eerste geval I_4 is, dus er hoeft *niet* te worden gepermuteerd), de 4×4-onderdriehoeksmatrix L en de standaardrijvorm U. We zien dat `GaussianElimination(A)` dezelfde standaardrijvorm geeft.
Voor de LU-ontbinding van A^{T} is er wél een permutatie nodig. ◇

14.5 Kern, Rijruimte, Kolomruimte

NullSpace
RowSpace
ColumnSpace

Voor de *kern*, *rijruimte* en de *kolomruimte* van een matrix hebben we respectievelijk `NullSpace`, `RowSpace` en `ColumnSpace`.

Voorbeeldopgave

Bepaal de kern, rijruimte en kolomruimte van de matrix

$$A = \begin{bmatrix} -7 & 6 & 6 & 4 \\ 2 & -18 & 3 & 2 \\ 5 & 7 & -17 & 6 \\ 0 & 5 & 8 & -12 \end{bmatrix}$$

Voorbeeldsessie

```
> with(LinearAlgebra):
> A := <<-7,2,5,0>|<6,-18,7,5>|<6,3,-17,8>|<4,2,6,-12>>:
> NullSpace(A);
```

$$\left\{\begin{bmatrix} \frac{728}{359} \\ \frac{572}{1077} \\ \frac{1258}{1077} \\ 1 \end{bmatrix}\right\}$$

```
> RowSpace(A);
```

$$\left[\left[1,\ 0,\ 0,\ \frac{-728}{359}\right],\ \left[0,\ 1,\ 0,\ \frac{-572}{1077}\right],\ \left[0,\ 0,\ 1,\ \frac{-1258}{1077}\right]\right]$$

```
> ColumnSpace(A);
```

$$\left[\begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix},\ \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix},\ \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}\right]$$

Toelichting

Met `NullSpace(A)` wordt een *basis* voor de kern berekend, in de vorm van een *verzameling* basisvectoren. Ven de rijruimte en kolomruimte wordt eveneens een basis bepaald, evenwel nu in de vorm van een *lijst*. ◇

Basis
SumBasis
Intersect...

Basis lineaire ruimte. Verwant hiermee zijn de commando's om een basis ven een lineaire ruimte te bepalen. Er zijn er drie: `Basis(V)` om een van een stelsel voortbrengers V te bepalen, `SumBasis(V,W)` geeft een basis van de *somruimte* $V+W$ en `IntersectionBasis(V,W)` van de *doorsnede* $V \cap W$. De lineaire ruimten V en W moeten als lijst of als verzameling van vectoren worden gegeven. Het resultaat komt in dezelfde vorm, dat wil zeggen als lijst óf als verzameling.

14.6 Numerieke en exacte berekeningen

Als er elementen van de gebruikte matrices of vectoren zijn gegeven als decimale getallen (dus bijvoorbeeld 0.5 inplaats van $\frac{1}{2}$), dan gebruikt Maple automatisch benaderende (numerieke) methoden om bijvoorbeeld stelsels vergelijkingen op te lossen. We geven een voorbeeld.

Voorbeeldsessie

```
> with(LinearAlgebra):
> A:=Matrix([[-0.35,0.3,0.3,0.2],  [0.1,-0.9,0.15,0.1],
            [0.25,0.35,-0.85,0.3],[0,0.25,0.4,-0.6]]);
```

$$A := \begin{bmatrix} -0.35 & 0.3 & 0.3 & 0.2 \\ 0.1 & -0.9 & 0.15 & 0.1 \\ 0.25 & 0.35 & -0.85 & 0.3 \\ 0 & 0.25 & 0.4 & -0.6 \end{bmatrix}$$

```
> b:=Vector(4):      # nulvector
> Opl:=LinearSolve(A,b);
```

$$Opl := \begin{bmatrix} -0. \\ -0. \\ 0. \\ 0. \end{bmatrix}$$

```
> NullSpace(A);  kern1 := %[1]:
```

$$\left\{ \begin{bmatrix} 0.780023231905231484 \\ 0.204291798832322652 \\ 0.449299096033324852 \\ 0.384654313535684356 \end{bmatrix} \right\}$$

```
> kern1.kern1;
```

$$1.00000000000000024$$

```
> Rank(A), Determinant(A);
```

$$3,\ 0.$$

```
> Ar := convert(A,rational);
```

$$Ar := \begin{bmatrix} \frac{-7}{20} & \frac{3}{10} & \frac{3}{10} & \frac{1}{5} \\ \frac{1}{10} & \frac{-9}{10} & \frac{3}{20} & \frac{1}{10} \\ \frac{1}{4} & \frac{7}{20} & \frac{-17}{20} & \frac{3}{10} \\ 0 & \frac{1}{4} & \frac{2}{5} & \frac{-3}{5} \end{bmatrix}$$

```
> kern2 := LinearSolve(Ar,b, free='t');
```

$$kern2 := \begin{bmatrix} \frac{728}{359}\,t_4 \\ \frac{572}{1077}\,t_4 \\ \frac{1258}{1077}\,t_4 \\ t_4 \end{bmatrix}$$

Is er een waarde van t_4, zodat kern2 = kern1?

Dat kunnen we onderzoeken door kern2 te normeren.

```
> kern3 := Normalize(evalf(subs(t[4]=1,kern2)),2);
```

$$kern3 := \begin{bmatrix} 0.780023231957438057 \\ 0.204291798875302661 \\ 0.449299095982731376 \\ 0.384654313600000019 \end{bmatrix}$$

```
> kern1 -kern3;
```

$$\begin{bmatrix} -0.522065723984610486\ 10^{-10} \\ -0.429800084411624540\ 10^{-10} \\ 0.505934738548319274\ 10^{-10} \\ -0.643156639057451685\ 10^{-10} \end{bmatrix}$$

Toelichting

Het stelsel vergelijkingen $A\mathbf{x} = \mathbf{0}$ wordt *numeriek* opgelost. Dat betekent dat er hoe dan ook maar één oplossing wordt geleverd, in dit geval de triviale oplossing **0**. Om Ker A te bepalen wordt blijkbaar een andere methode gebruikt, want Maple vindt een `NullSpace` van dimensie 1; de basisvector ervan is blijkbaar genormeerd. Dat de dimensie > 0 is, volgt ook uit de berekening van de rang en de determinant.
In de matrix A_r zijn de decimale getallen omgezet in breuken. Het stelsel wordt nu *exact* opgelost. Dat betekent in dit geval dat er een oplossing komt met een parameter er in. ◇

Voor grotere berekeningen zijn numerieke benaderingen veel efficiënter dan exacte berekeningen. We laten dat zien aan de hand van een voorbeeld.

Voorbeeldopgave

Bepaal de oplossing van het stelsel $A\mathbf{x} = \mathbf{b}$, waarbij A een 500×500-matrix is en $\mathbf{b} \in \mathbb{R}^{500}$. Vul A en $\mathbf{b}$ met willekeurige reële getallen.

Voorbeeldsessie

```
> with(LinearAlgebra):
> A := RandomMatrix(500);
```

$$A := \begin{bmatrix} \text{500 x 500 Matrix} \\ \text{Data Type : anything} \\ \text{Storage : rectangular} \\ \text{Order : Fortran\_order} \end{bmatrix}$$

```
> b := RandomVector(500);
```

$$b := \begin{bmatrix} \text{500 Element Column Vector} \\ \text{Data Type : anything} \\ \text{Storage : rectangular} \\ \text{Order : Fortran\_order} \end{bmatrix}$$

```
> st := time(): x := LinearSolve(A,b):
  `benodigde tijd` = time()-st;
```

$$\mathit{benodigde\ tijd} = 32.81$$

```
> evalf(x[108]);
```

$$-0.3522746977$$

```
> A1 := copy(evalf(A));
```

$$A1 := \begin{bmatrix} \text{500 x 500 Matrix} \\ \text{Data Type : anything} \\ \text{Storage : rectangular} \\ \text{Order : Fortran\_order} \end{bmatrix}$$

```
> st := time(): x1 := LinearSolve(A1,b):
  `benodigde tijd` = time()-st;
```

$$\mathit{benodigde\ tijd} = 0.088$$

```
> A2 := Matrix(A, datatype=float[8]);
  b2 := Vector(b, datatype=float[8]):
```

$$A2 := \begin{bmatrix} \text{500 x 500 Matrix} \\ \text{Data Type : float[8]} \\ \text{Storage : rectangular} \\ \text{Order : Fortran\_order} \end{bmatrix}$$

```
> st := time(): x2 := LinearSolve(A2,b2):
  `benodigde tijd` = time()-st;
```

$$\mathit{benodigde\ tijd} = 0.056$$

```
> evalf(x[108]), x1[108], x2[108];
```

$$-0.3522746977,\ -0.352274697663909453,$$
$$-0.352274697663909453$$

Toelichting

De matrix A en de vector **b** worden gevuld met willekeurige gehele getallen tussen -99 en $+99$. Zo'n grote matrix wordt door Maple niet op het scherm vertoond, het geeft alleen een soort samenvatting. Als we met `LinearSolve` het stelsel oplossen dan blijkt Maple daar ruim een halve minuut voor nodig te hebben. (Dat is nog overkomelijk, maar neem maar eens een 5000×5000-matrix; reken maar op minstens een half uur!) Omdat A en **b** gehele getallen bevatten, rekent Maple in dit geval exact, dat wil zeggen in grote trekken met Gauss-eliminatie. De elementen van de oplossingsvector **x** zijn dan ook breuken met honderden cijfers in de teller en de noemer.
Met `evalf(A)` maken we er een matrix met decimale getallen van. Het oplossen van het stelsel gaat nu in minder dan eentiende seconde. Hierbij hebben we echter nog niet eens ten volle gebruikgemaakt van de numerieke capaciteiten van `LinearAlgebra`. In matrix `A1` staat: `Data Type: anything`, en dat betekent dat Maple niets weet over wat voor soort dingen de de elementen zijn. Er zouden bijvoorbeeld expressies met variabelen erin tussen kunnen zitten. Als we er voor zorgen dat Maple weet dat de getallen in de matrix allemaal van het type `float[8]` zijn, en dat hebben we voor matrix `A2` gedaan, dan gaat het nog eens bijna twee keer zo snel.
In dit geval hebben we de nieuwe matrix `A2` gemaakt met het commando `Matrix`, zoals in §13.1,[38] waarbij we de optie
`datatype` `datatype=float[8]` hebben meegegeven. Dit komt overeen met het type *double* in MATLAB. ◇

Inlezen van matrices uit een bestand. Dergelijke grote matrices en vectoren zult u vaak uit een bestand willen inlezen. Hiervoor
`ImportMatrix` is het commando `ImportMatrix` bedoeld. Zeker als u dat nog niet zo vaak gedaan hebt, is het handig om daarvoor een *assistant* te gebruiken: Tools → Assistants → Import Data.... Hiermee kan zelfs op een eenvoudige manier een foto (in de vorm van een *.jpg- of *.bmp-bestand) als matrix worden ingelezen. Op zo'n matrix kunnen dan allerlei bewerkingen worden toegepast om het plaatje te veranderen. De bibliotheek `ImageTools` bevat een aantal standaardfuncties die hierbij gebruikt kunnen worden. Ook geluidsfragmenten (als *.wav-bestand) kunnen worden ingelezen als matrix: voor elk kanaal een kolom. Voor bewerking is de bibliotheek `AudioTools` bedoeld..

[38] Het argument van `Matrix` hoeft dus niet per se een lijst van lijsten te zijn. Het mag zelfs een matrix zijn.

14.7 Constructie van een orthonormale basis

GramSchmidt

Bij een gegeven basis van een lineaire deelruimte kan de procedure volgens Gram-Schmidt worden gebruikt om een *orthonormale basis* te construeren. De Maple-procedure heet `GramSchmidt` en het argument is een lijst van vectoren. Ook het resultaat is weer een lijst van vectoren.

Voorbeeldopgave

Gebruik Gram-Schmidt om uitgaande van de basis

$$\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\} = \{(1,1,1), (0,1,1), (0,0,1)\}$$

een orthonormale basis van $\mathbb{R}^3$ te construeren.

Voorbeeldsessie

```
> with(LinearAlgebra):
> v1,v2,v3 := <1,1,1>,<0,1,1>,<0,0,1>:
> GramSchmidt([v1,v2,v3]);
```

$$\left[\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} \frac{-2}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix}\right]$$

```
> GramSchmidt([v2,v1,v3]);
```

$$\left[\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{-1}{2} \\ \frac{1}{2} \end{bmatrix}\right]$$

```
> b := GramSchmidt([v1,v2,v3], normalized);
```

$$b := \left[\begin{bmatrix} \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} \end{bmatrix}, \begin{bmatrix} -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{bmatrix}, \begin{bmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix}\right]$$

Controle of `b` een orthonormale basis is:

```
> B := Matrix(b);
```

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

```
> B.B^%T;
```

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

Toelichting

`normalized`

De volgorde waarin de basisvectoren worden opgegeven is van belang: bij een andere volgorde vindt `GramSchmidt` een ander stelsel. Het gevonden stelsel is niet genormeerd. Met de extra optie `normalized` krijgen we wél een orthonormale basis.
De controle of het stelsel inderdaad orthonormaal is voeren we uit door de gevonden vectoren als kolommen in een matrix te plaatsen en na te gaan of deze matrix orthogonaal is. ◇

14.8 Eigenwaarden en eigenvectoren

`Characterist...`

Met het commando `Eigenvalues` berekent Maple de *eigenwaarden* van een $n \times n$-matrix en met `CharacteristicPolynomial` de *karakteristieke polynoom.*

`Eigenvalues`

Het commando `Eigenvalues` levert een *vector* van eigenwaarden; meervoudige eigenwaarden komen even vaak voor in deze vector als hun multipliciteit.

`Eigenvectors`

Het commando `Eigenvectors` levert de vector met eigenwaarden af en een $n \times n$-matrix, waarvan elke kolom een eigenvector is. In deze matrix staan de eigenvectoren staan in dezelfde volgorde als de eigenwaarden in de vector van eigenwaarden. Voor elke eigenwaarde waarvan de geometrische multipliciteit kleiner is dan de algebraïsche multipliciteit komen er een of meer nulkolommen in de matrix met eigenvectoren.

Door `Eigenvectors` aan te roepen met de optie `output=list` kunnen deze gegevens ook in een andere vorm verkregen worden. Het resultaat is dan namelijk een *lijst*, waarvan elk element weer een lijst is met drie elementen. Het eerste element is steeds de eigenwaarde, het tweede de algebraïsche multipliciteit van die eigenwaarde, en het

derde element is een *verzameling* van lineair onafhankelijke eigenvectoren behorende bij deze eigenwaarde en is dus een basis van de *eigenruimte*. Als de matrix in kwestie niet defect is, bevat deze verzameling dus net zoveel eigenvectoren als de algebraïsche multipliciteit van de betreffende eigenwaarde. De afzonderlijke eigenvectoren zijn uit deze lijst te halen door middel van de [ ]-notatie.
Na `ev := Eigenvectors(A, output=list)` is `ev[i]` de i-de lijst met eigenwaarde, multipliciteit en verzameling eigenvectoren, `ev[i,3]` de bijbehorende *verzameling* van eigenvectoren en `ev[i,3,1]` de eerste (eventueel enige) eigenvector uit deze verzameling.

Voorbeeldopgave

Gegeven de matrix

$$A = \begin{bmatrix} 3 & -2 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

Bereken de eigenwaarden en eigenvectoren van A. Bepaal een matrix P en een diagonaalmatrix D zodat $D = P^{-1}AP$.

Voorbeeldsessie

```
> with(LinearAlgebra):
> A := Matrix([[3,-2,0],[-2,3,0],[0,0,5]]):
> eigenw := Eigenvalues(A);
```

$$eigenw := \begin{bmatrix} 1 \\ 5 \\ 5 \end{bmatrix}$$

```
> eigenw, P := Eigenvectors(A);
```

$$eigenw, P := \begin{bmatrix} 1 \\ 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

```
> P^(-1).A.P = DiagonalMatrix(eigenw);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

Ordenen:

```
> eig := Eigenvectors(A, output=list);
```

$$eig := \left[\left[5, 2, \left\{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}\right\}\right], \left[1, 1, \left\{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}\right\}\right]\right]$$

```
> eigenv := op( sort( eig, (x,y)->x[1]<y[1] ) );
```

$$eigenv := \left[1,\ 1,\ \left\{\begin{bmatrix}1\\1\\0\end{bmatrix}\right\}\right],\ \left[5,\ 2,\ \left\{\begin{bmatrix}0\\0\\1\end{bmatrix},\ \begin{bmatrix}-1\\1\\0\end{bmatrix}\right\}\right]$$

```
> er1 := op(eigenv[1][3]);
```

$$er1 := \begin{bmatrix}1\\1\\0\end{bmatrix}$$

```
> er2 := op(eigenv[2][3]);
```

$$er2 := \begin{bmatrix}0\\0\\1\end{bmatrix},\ \begin{bmatrix}-1\\1\\0\end{bmatrix}$$

P wordt de matrix met de lineair onafhankelijke eigenvectoren als kolommen

```
> P := Matrix([er1,er2]);
```

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

```
> P^(-1).A.P;
```

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

Toelichting

Als de algebraïsche multipliciteit van een eigenwaarde groter dan één is, dan komt deze ook even vaak in de vector van eigenwaarden voor; in dit geval wordt de eigenwaarde 5 dus twee keer genoemd. Omdat A blijkbaar niet defect is, geeft `Eigenvectors(A)` als vector precies de diagonaal van de gevraagde D, de matrix met eigenvectoren is precies de gevraagde P.
De *volgorde* in de vector en matrix die door `Eigenvectors` worden afgeleverd is volstrekt willekeurig. Als het commando voor een tweede keer wordt gegeven, komt er vaak een andere volgorde uit. Soms wil men de eigenwaarden bijvoorbeeld in opklimmende volgorde hebben. In dat geval is het het handigst op de optie `output=list` te gebruiken. Voor het ordenen hebben we de methode van §8.3 gebruikt. ◇

14.9 De Jordan-normaalvorm

`JordanForm`

Hiervoor is het commando `JordanForm` bedoeld. We volstaan met een voorbeeld.

Voorbeeldopgave

Gegeven

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

Bepaal een matrix T zo dat $T^{-1}AT$ in Jordan-normaalvorm staat.

Voorbeeldsessie

```
> with(LinearAlgebra):
> A := Matrix([[2,0,0,1,1],[0,2,0,2,2],
  [0,0,3,0,1],[-2,1,1,2,1],[0,0,-1,0,1]]):
> factor(CharacteristicPolynomial(A,'lambda'));
```

$$(\lambda - 2)^5$$

```
> Eigenvectors(A);
```

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

```
> J,T := JordanForm(A, output=['J','Q']);
```

$$J, T := \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, \begin{bmatrix} -2 & 0 & 1 & 0 & 0 \\ -4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 \end{bmatrix}$$

```
> T^(-1).A.T;
```

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

Toelichting

Kennelijk bezit A twee onafhankelijke eigenvectoren bij $\lambda_1 = 2$, dus de geometrische multipliciteit is 2. Dit komt overeen met het aantal Jordan-blokjes. In de `output`-optie kunnen we aangeven of we de Jordan-vorm (`'J'`), de transformatiematrix (`'Q'`) of allebei willen hebben. ◇

Opgave 14.1

Gegeven is de matrix

$$A = \begin{bmatrix} 1 & x & x^2 \\ 1 & (x+3) & (x+3)^2 \\ 1 & (x+5) & (x+5)^2 \end{bmatrix} \qquad (x \in \mathbb{R}).$$

Toon aan dat A voor alle waarden van x inverteerbaar is en bepaal A^{-1}.

Opgave 14.2

Laat $A = \begin{bmatrix} 2 & 3 & 1 \\ 1 & -1 & 0 \end{bmatrix}$ en beschouw de afbeelding $F : \mathbb{R}^3 \to \mathbb{R}^2$, gegeven door $F(\mathbf{x}) = A\mathbf{x}$.

(a) Maak de Maplefunctie `F` waarmee u `F(u1,u2,u3)` kunt berekenen;

(b) Bepaal alle vectoren $\mathbf{u} \in \mathbb{R}^3$ waarvoor $F(\mathbf{u}) = \mathbf{0}$;

(c) Laat ℓ de lijn $(1,1,0) + \langle (0,1,2) \rangle$ in $\mathbb{R}^3$ zijn. Toon aan dat F de lijn ℓ op zekere lijn m afbeeldt en schrijf m in de vorm $\mathbf{p} + \langle \mathbf{u} \rangle$.

Opgave 14.3

Bepaal de representatiematrices van F en G waarbij

$$\begin{aligned} F(u_1,u_2) &= (2u_1 + u_2,\ u_1 + u_2,\ u_1 - u_2,\ u_1 - 2u_2) \\ G(u_1,u_2,u_3,u_4) &= (3u_1 + 2u_2 - u_3 + u_4,\ u_2 + u_3,\ -u_1) \end{aligned}$$

Bepaal $G \circ F : \mathbb{R}^2 \to \mathbb{R}^3$ alsmede $[G \circ F]$. Ga na of $F \circ G$ bestaat.
Aanwijzing: Definieer bijvoorbeeld de functie G als

```
G := u -> vector(3, [3*u[1] + ...
```

Het is in zo'n geval niet nodig eerst aan te geven dat `u` een vector is. Bereken de beelden van de standaard-eenheidsvectoren.

Opgave 14.4

Gegeven zijn de matrices

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \quad \text{en} \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

(1) bepaal een matrix X, zodat $AX = B$;

(2) bepaal een matrix Y, zodat $YA = B$.

Aanwijzing: Zie `?LinearSolve`. Denk eraan dat $AX \neq XA$. Controleer uw antwoorden.

Opgave 14.5

Van een lineaire afbeelding $F : \mathbb{R}^3 \to \mathbb{R}^2$ zijn de volgende gegevens bekend: $F(1,2,1) = (1,1)$, $F(3,1,0) = (-1,0)$ en $F(0,1,0) = (0,0)$. Bepaal $[F]$ door de gegevens in de vorm $[F]A = B$ te zetten, met A en B matrices.

Opgave 14.6

Gegeven de matrix $A = \begin{bmatrix} 2 & 3 & -1 \\ 0 & 1 & 2 \end{bmatrix}$.
Ga na of het stelsel $A\mathbf{x} = \mathbf{b}$ oplosbaar is voor elke $\mathbf{b} \in \mathbb{R}^2$. Bepaal de oplossingsvariëteit als $\mathbf{b} = (4,3)$.

Opgave 14.7

Gegeven de matrix

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}.$$

Ga na of $\mathbf{v} \in \mathrm{Kol}(A)$ als
(a) $\mathbf{v} = (2,0,0)$; (b) $\mathbf{v} = (1,1,1)$.

Opgave 14.8

Bepaal voor alle α een basis, respectievelijk de dimensie van

$$D = \langle(\alpha,0,0,0),\ (2,1,1,\alpha),\ (2,1,\alpha,1),\ (2,\alpha,1,1)\rangle.$$

Opgave 14.9

Gegeven is het stelsel van vier vergelijkingen met vier onbekenden:

$$\begin{cases} x_1 - 3x_2 - 5x_3 - x_4 = 2 \\ x_1 + \alpha x_2 + 6x_3 + 5x_4 = -1 \\ 2x_1 - 2x_2 + x_3 + 4x_4 = 1 \\ 5x_1 - 11x_2 - 14x_3 + \alpha x_4 = \beta \end{cases}$$

(a) Bepaal de waarden van α en β waarvoor het stelsel oplosbaar is; geef voor alle mogelijke gevallen de dimensie van de oplossingsverzameling.
(b) Bepaal die oplossingsverzamelingen met dimensie groter dan nul.
(c) Bepaal de oplossingsverzameling als $\alpha = 2, \beta = 100$.

Variant: Dezelfde vragen, maar met "$+x_4$" in plaats van "$-x_4$" in de eerste vergelijking.

Opgave 14.10

$A = \frac{1}{7}\begin{bmatrix} 2 & 3 & \kappa \\ 3 & -6 & \lambda \\ 6 & 2 & \mu \end{bmatrix}$ is een orthogonale matrix. Bepaal $\kappa,\ \lambda$ en μ.

Opgave 14.11

Gegeven een lijst E van vectoren in $\mathbb{R}^n$, en een vector $\mathbf{v} \in \mathbb{R}^n$. Gevraagd wordt om na te gaan of E een basis is van $\mathbb{R}^n$ en zo ja, de coördinaten van $[\mathbf{v}]_E$ te berekenen. Neem als voorbeeld:

```
v1 := <1,2,3>:  v2 := <8,-1,7>:  v3 := <0,4,-3>:
E := [v1, v2, v3]:       v := <1,1,1>:
```

Onderdeel (a) en (b) van deze opdracht zijn bedoeld om te onderzoeken hoe het probleem het best kan worden aangepakt. Uw bevindingen kunnen dan worden verwerkt in de procedure die in onderdeel (c) wordt gevraagd.

(a) Maak een matrix A met de elementen van `E` als kolommen. Ga na hoe u Maple kunt laten besluiten of E een basis is van $\mathbb{R}^n$. Denk na over het algemene geval, dus uw werkwijze moet ook werken bij $n = 2,\ n = 4,\ n = 5$ enzovoort.

(b) Bepaal $[\mathbf{v}]_E$, dus bereken de coördinaten van $\mathbf{v}$ ten opzichte van de basis E.

(c) Schrijf een procedure met de naam *coord*, zodat het commando `coord(v, E);` geeft:
(1) als E een basis is: de lijst met de coördinaten van $[\mathbf{v}]_E$;
(2) als E geen basis is: de tekst "E is geen basis".

(d) Test uw procedure met de gegeven $\mathbf{v}$ en E, en met andere, zelfbedachte $\mathbf{v}$'s en E's (ten minste óók een E die géén basis is: geen lineair onafhankelijk stelsel (l.o.s.) of te weinig vectoren).

Opgave 14.12

D is de lineaire deelruimte van $\mathbb{R}^4$ voortgebracht door de vectoren $(1, 1, 1, 1)$, $(1, 2, -1, 0)$ en $(2, 1, 0, -1)$. Verder is $\mathbf{p} = (1, 2, 3, 4)$.

(a) Bepaal een orthonormale basis van D.
(b) Bepaal de orthogonale projectie van $\mathbf{p}$ op D.
(c) Bepaal een basis van $D^{\perp}$.

Opgave 14.13

Ga van de volgende matrices A na of ze defect of diagonaliseerbaar zijn. Indien A diagonaliseerbaar is, bepaal dan een matrix P die A diagonaliseert, en bepaal $P^{-1}AP$.

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

Opgave 14.14

We schrijven $V = D_1 \oplus D_2$ als geldt: $V = D_1 + D_2$ én $D_1 \cap D_2 = \{\mathbf{0}\}$. Bepaal in de volgende gevallen die waarden van κ, λ en μ waarvoor $\mathbb{R}^4 = D_1 \oplus D_2$:

(a) $D_1 = \langle (1,1,0,1),\ (0,-3,3,\kappa) \rangle$
$D_2 = \langle (1,-1,4,2),\ (2,-5,10,5),\ (0,-3,2,\lambda) \rangle$

(b) $D_1 = \langle (1,1,0,0),\ (1,\mu,1,2) \rangle$
$D_2 = \langle (0,1,1,1),\ (0,3,\mu,3),\ (\mu-3,2,-1,0) \rangle$

Opgave 14.15

Stelling: Als $L = \mathbf{u}_1 + D_1$ en $M = \mathbf{u}_2 + D_2$ lineaire variëteiten in $\mathbb{R}^n$ zijn, dan geldt:

$$L \cap M \neq \varnothing \iff (\mathbf{u_1} - \mathbf{u_2}) \in (D_1 + D_2)$$

Laat nu $D_1 = \langle (1,1,0,0), (0,0,1,0) \rangle$, $D_2 = \langle (1,1,1,1), (0,0,0,1) \rangle$, $\mathbf{u_1} = (2,1,1,1)$ en $\mathbf{u_2} = (-1,0,0,-1)$.

(a) Onderzoek of $\mathbb{R}^4 = D_1 \oplus D_2$;
(b) Ga na of in dit geval $L \cap M \neq \varnothing$;
(c) Bepaal $D_1 \cap D_2$.

Opgave 14.16

Breng de onderstaande kwadratische vorm via een orthonormale basistransformatie in de canonieke gedaante (hoofdassengedaante). Geef de canonieke gedaante aan en geef tevens de overgangsformules die de kentallen y_1, y_2, y_3 van een vector ten opzichte van de nieuwe basis uitdrukken in zijn kentallen x_1, x_2, x_3 ten opzichte van de oude basis en omgekeerd.

$$8x_1^2 - 4x_1x_2 + 5x_2^2 - 4x_1x_3 - 8x_2x_3 + 5x_3^2.$$

Opgave 14.17

Bereken van elk van de volgende matrices A de karakteristieke polynoom. Bereken bovendien getallen a, b, c zodanig dat

$$aI_3 + bA + cA^2 + A^3 = 0.$$

$$A = \begin{bmatrix} 1 & -2 & 2 \\ -2 & -2 & 4 \\ 2 & 4 & -2 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & -2 & 0 \\ -2 & 3 & 0 \\ 0 & 0 & 5 \end{bmatrix}.$$

Opgave 14.18

Beschouw de lineaire ruimte $C[0, 1]$, met het inproduct

$$(\mathbf{f}, \mathbf{g}) = \int_0^1 f(x)g(x)\, dx$$

$P_2 \subset C[0, 1]$ is de verzameling van alle polynomen met graad ≤ 2.

(a) Construeer een orthonormale basis van de lineaire deelruimte P_2 in $C[0, 1]$;

(b) Bepaal de beste benadering van de functie $x \mapsto e^x$ door een functie van de vorm $x \mapsto \alpha_0 + \alpha_1 x + \alpha_2 x^2$;

(c) Teken grafieken van e^x en de berekende kwadratische benadering ervan in één plaatje. Gebruik verschillende kleuren. Neem als interval op de x-as $[-1/2, 3/2]$.

Aanwijzing: Werk met Gram-Schmidt. De procedure `GramSchmidt` van Maple werkt alleen in $\mathbb{R}^n$, met het 'gewone' inproduct en is daarom hier niet te gebruiken.

Opgave 14.19

Maak een procedure met de naam `diagonalisator`, zodat het resultaat van de aanroep `diagonalisator(A);` met `A` een vierkante matrix, is

- het woord *Defect* als A defect is;
- een matrix P waarmee $P^{-1}AP$ een diagonaalmatrix is.

Test uw procedure met de matrices van opgave 14.13.

Opgave 14.20

Op P_2 is het volgende inproduct gedefinieerd

$$(p,q) = \int_0^1 p(x)q(x)\,dx$$

Gegeven de lineaire afbeelding $F : P_2 \to P_2$, bepaald door:

$$F(ax^2 + bx + c) = ax^2 - (2a + b)x + (a + b + c).$$

(a) Maak een Maple-procedure `F( )`, die als *input* een, als een functie gedefinieerde, polynoom p heeft en als *output* de functie $F(p)$. `F()` moet dus als volgt werken:

```
> p := x -> a*x^2 + b*x + c;
```

$$p := x \to a\,x^2 + b\,x + c$$

```
> F(p);
```

$$x \to a\,x^2 - (2\,a + b)\,x + a + b + c$$

```
> F( y -> (y + 6)^2 - 2 );
```

$$x \to x^2 - 14\,x + 47$$

(b) Bereken voor twee verschillende (willekeurige) polynomen p en q de inproducten (Fp, Fq) en (p, q), laat zien dat deze gelijk zijn, en bewijs daarmee dat F een orthogonale afbeelding is.

Opgave 14.21

In de lineaire ruimte $C[-1, 1]$ wordt een inproduct gegeven door

$$(f,g) = \int_{-1}^1 f(x)g(x)\,dx.$$

(a) Maak procedures `IP( )` en `NRM( )` zodat `IP(f,g);` het inproduct (f, g) aflevert en `NRM(f);` de norm $\|f\|$ aflevert als `f` en `g` *procedures* zijn;

(b) Maak een procedure `GS( )`, zodat `GS(L);` een orthonormale basis van $V = \langle f_1, \ldots, f_n \rangle$ aflevert in de vorm van een *lijst* van *procedures* als `L` de lijst van functies $f_1, \ldots, f_n$ is. U mag ervan uitgaan dat $\{f_1, \ldots, f_n\}$ een l.o.s. is.

(c) Test uw procedure met `L:=[x->1,x->x,x->x^2,x->x^3];`.

(d) Bepaal de orthogonale projectie van $f : x \mapsto \sin x$ op P_3. Teken een plaatje van de grafieken van f en $Proj_{P_3} f$ op het interval $[-2, 2]$.

(e) Wat zou er in de procedure `GS( )` veranderd moeten worden als niet gegeven is dat $f_1, \ldots, f_n$ een l.o.s. is?

Opgave 14.22

Maak een procedure die het resultaat van `LinearSolve(A,b)` bewerkt tot de vorm $\mathbf{u} + D$, dat wil zeggen: een steunvector `u` aflevert en een lijst met voortbrengers van D.
Aanwijzing: Gebruik de procedure `indets` om het lijstje parameters te krijgen.
Voor enthousiaste puzzelaars: Zorg dat de procedure ook uitvoer geeft van de vorm: $\mathbf{u} + \langle \mathbf{v}_1, \ldots, \mathbf{v}_k \rangle$.
(Gebruik `convert(..., string)`, `substring` en `cat`.)

Opgave 14.23

Bepaal de Jordan-normaalvorm en de transformatiematrices in de volgende gevallen:

(a) $A = \begin{bmatrix} 3 & 6 \\ 8 & 16 \end{bmatrix}$

(b) $A = \begin{bmatrix} 3 & 6 & 1 \\ 8 & 16 & 0 \\ 2 & 3 & 4 \end{bmatrix}$

(c) $A = \begin{bmatrix} -103 & 132 & 292 \\ 4 & 0 & 1 \\ -41 & 50 & 110 \end{bmatrix}$

Opgave 14.24

Bereken de reële en complexe Jordan-normaalvorm van de matrix

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

Opgave 14.25

Bereken de Jordan-normaalvorm, alsmede een Jordan-basis van $\mathbb{R}^3$, respectievelijk $\mathbb{R}^4$ van de volgende matrices:

$$\begin{bmatrix} 4 & -1 & 1 \\ 1 & 6 & 0 \\ 0 & 0 & 5 \end{bmatrix}, \quad \begin{bmatrix} 4 & -1 & -1 \\ 1 & 6 & 1 \\ 0 & 0 & 5 \end{bmatrix}, \quad \begin{bmatrix} 7 & 1 & 1 & -1 \\ 1 & 10 & 0 & 2 \\ 0 & 0 & 12 & 0 \\ 2 & -1 & -3 & 10 \end{bmatrix}.$$