

# Module 17

## Vectoranalyse

---

<b>Onderwerp</b>	Vectoranalyse en herhaald integreren.
<b>Voorkennis</b>	Lijn-, oppervlakte- en volume-integraal, rotatie, divergentie, raaklijn en raakvlak.
<b>Expressies</b>	SetCoordinates, VectorField, Nabla, PathInt, LineInt, ArcLength, ScalarPotential, SurfaceInt, Flux, VectorPotential, MapToBasis, coordplot, coordplot3d, GetCoordinates, Laplacian, AddCoordinates
<b>Bibliotheken</b>	VectorCalculus, plots
<b>Zie ook</b>	Module 6, 8, 10, 13, 16.

---

### 17.1 Begrippen uit de vectoranalyse

Gegeven zijn

- Een differentieerbaar vectorveld  $\mathbf{w} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ ,
- een scalarveld  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ ,
- een regulier geparametriseerd oppervlak

$$S = \{\mathbf{r}(u, v) \mid (u, v) \in H\},$$

met  $H$  een elementair gebied,

- een reguliere geparametriseerde ruimtekromme

$$k = \{\mathbf{r}(t) \mid t \in [a, b]\}.$$

We beschouwen in deze module de volgende begrippen:

- raakvector aan  $k : \mathbf{r}'(t)$ ,
- booglengte van  $k: \int_k ds = \int_a^b |\mathbf{r}'(t)| dt$ ,
- lijnintegraal van het scalarveld  $f$  langs  $k :$

$$\int_k f ds = \int_a^b f(\mathbf{r}(t)) |\mathbf{r}'(t)| dt,$$

- lijnintegraal van  $\mathbf{w}$  langs  $k$ :

$$\int_k \mathbf{w} \cdot d\mathbf{r} = \int_a^b \mathbf{w}(\mathbf{r}(t)) \cdot \mathbf{r}'(t) dt,$$

- raakvlak aan  $S$  met richtingsvectoren  $\mathbf{r}_u$  en  $\mathbf{r}_v$  en normaal  $\pm(\mathbf{r}_u \times \mathbf{r}_v)$ ,

- eenheidsnormaal op  $S$ :

$$\hat{\mathbf{n}} = \pm(\mathbf{r}_u \times \mathbf{r}_v) / |\mathbf{r}_u \times \mathbf{r}_v| ,$$

- oppervlakte van  $S$ :

$$\iint_S dA = \iint_H |\mathbf{r}_u \times \mathbf{r}_v| \, dudv ,$$

- oppervlakte-integraal van het scalarveld  $f$  over  $S$ :

$$\iint_S f \, dA = \iint_H f(\mathbf{r}(u, v)) |\mathbf{r}_u \times \mathbf{r}_v| \, dudv ,$$

- oppervlakte- of flux-integraal van het vectorveld  $\mathbf{w}$  door  $S$ :

$$\iint_S \mathbf{w} \cdot d\mathbf{A} = \iint_S \mathbf{w} \cdot \hat{\mathbf{n}} \, dA = \iint_H \mathbf{w}(\mathbf{r}(u, v)) \cdot (\mathbf{r}_u \times \mathbf{r}_v) \, dudv ,$$

**Differentiaal-operatoren.** Met behulp van de  $\nabla$ -operator die in Cartesische coördinaten is gegeven door

$$\nabla = \hat{\mathbf{e}}_x \frac{\partial}{\partial x} + \hat{\mathbf{e}}_y \frac{\partial}{\partial y} + \hat{\mathbf{e}}_z \frac{\partial}{\partial z} ,$$

wordt gedefinieerd:

- de gradiënt van het scalarveld  $f$  als  $\nabla f(x, y, z)$ ,
- de *rotatie* van het vectorveld  $\mathbf{w}$  als:  $\nabla \times \mathbf{w}$ ,
- de *divergentie* van het vectorveld  $\mathbf{w}$  als:  $\nabla \cdot \mathbf{w}$ .

Er geldt:

$$\nabla \times \nabla f = \mathbf{0} , \quad \nabla \cdot (\nabla \times \mathbf{w}) = 0 .$$

De *Laplace-operator*  $\nabla^2$  (ook vaak genoteerd als  $\Delta$ ), toegepast op een scalarveld  $f$  is gedefinieerd als

$$\nabla^2 f = \nabla \cdot \nabla f .$$

**Conservatieve velden.** Als  $\nabla \times \mathbf{w} = \mathbf{0}$  voor alle  $(x, y, z) \in G \subset \mathbb{R}^3$  dan heet  $\mathbf{w}$  *conservatief*. Er bestaat dan een scalarveld  $\Phi$ , zodat  $\mathbf{w} = \nabla \Phi$  voor alle  $(x, y, z) \in G$ .  $\Phi$  heet een *scalaire potentiaal* van  $\mathbf{w}$ .

**Solenoïdale velden:** Als  $\nabla \cdot \mathbf{w} = 0$  voor alle  $(x, y, z) \in G \subset \mathbb{R}^3$  dan heet  $\mathbf{w}$  *solenoïdaal*. Er bestaat dan een vectorveld  $\mathbf{F}$ , zodat  $\mathbf{w} = \nabla \times \mathbf{F}$  voor alle  $(x, y, z) \in G$ . Het veld  $\mathbf{F}$  heet een *vectorpotentiaal* van  $\mathbf{w}$ .

## 17.2 Herhaalde integralen

Om een dubbele integraal van een tweedimensionaal scalarveld of de volume-integraal van een driedimensionaal scalarveld te berekenen, moet de integraal als *herhaalde* integraal worden geschreven.

Bijvoorbeeld, als  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  een tweedimensionaal scalarveld is, en  $G \subset \mathbb{R}^2$  een elementair gebied:

$$G = \{(x, y) \mid a \leq x \leq b, \quad \phi_1(x) \leq y \leq \phi_2(x)\}$$

dan is

$$\iint_G f \, dA = \int_a^b \int_{\phi_1(x)}^{\phi_2(x)} f(x, y) \, dy \, dx.$$

Om zo'n integraal met Maple te berekenen, moeten we 'de integraal van een integraal' berekenen:

```
int( int( f(x,y), y=phi1(x)..phi2(x) ), x=a..b );
```

eventueel (eerst) met `Int` inplaats van `int`, zie §6.5. De grenzen voor de herhaalde integraal moet men zelf bepalen, meestal is het gebruik van een kladblaadje (inderdaad: gewoon potlood en papier) hierbij onontbeerlijk.

Vaak is het handig om een meervoudige integraal in andere dan Cartesische coördinaten te berekenen. In twee dimensies geldt dan de substitutieregel:

$$\iint_G f(x, y) \, dx \, dy = \iint_H f(x(u, v), y(u, v)) \left| \det \frac{\partial(x, y)}{\partial(u, v)} \right| \, du \, dv.$$

Met  $\frac{\partial(x, y)}{\partial(u, v)}$  is de Jacobimatrix van de transformatie bedoeld.

### Voorbeeldopgave

Gegeven  $f(x, y, z) = x^2 + 2y^2 + 3z^2$ . Bereken  $\iiint_B f \, dV$ , als  $B$  de eenheidsbol is.

### Voorbeeldsessie

```
> f := (x,y,z) -> x^2 + 2*y^2 + 3*z^2;
      f := (x, y, z) -> x^2 + 2y^2 + 3z^2
> with(VectorCalculus):
```

Bolcoördinaten:

```

> x,y,z := r*sin(phi)*cos(theta),
        r*sin(phi)*sin(theta),
        r*cos(phi);
        x, y, z := r sin(phi) cos(theta), r sin(phi) sin(theta), r cos(phi)
> Jf := Jacobian( <x,y,z>, [r,phi,theta] );
        Jf := [ [ sin(phi) cos(theta)  r cos(phi) cos(theta)  -r sin(phi) sin(theta) ]
                [ sin(phi) sin(theta)  r cos(phi) sin(theta)  r sin(phi) cos(theta) ]
                [ cos(phi)             -r sin(phi)             0 ] ]
> df := simplify(LinearAlgebra:-Determinant(Jf));
        df := sin(phi) r^2
> Int( Int( Int( f(x,y,z)*df, r=0..1 ), phi=0..Pi ), theta=0..2*Pi ):
    %=value(%);
        \int_0^{2\pi} \int_0^\pi \int_0^1 (r^2 \sin(\phi)^2 \cos(\theta)^2 + 2 r^2 \sin(\phi)^2 \sin(\theta)^2 + 3 r^2 \cos(\phi)^2)
        \sin(\phi) r^2 dr d\phi d\theta = \frac{8\pi}{5}
    
```

### Toelichting

Als u uit het hoofd weet dat bij bolcoördinaten  $dV = r^2 \sin \phi \, dr d\phi d\theta$ , dan kan dat natuurlijk direct in de herhaalde integraal worden ingevuld. ◇

## 17.3 Rotatie en divergentie

Omdat we nu te maken hebben met vectorvelden, is het nodig de bibliotheek `VectorCalculus` te laden, door middel van de opdracht `with(VectorCalculus)`, en te vermelden in welk coördinatenstelsel we willen werken, voorlopig in de gewone Cartesische coördinaten, via `SetCoordinates(cartesian[x,y,z])`.

Om de onderwerpen genoemd in §17.1 met Maple aan te kunnen, hoeven we in principe alleen nog te weten hoe we de diverse afgeleiden van vectorgrootheden kunnen berekenen. Voor de rotatie en divergentie zijn standaardfuncties in Maple. Stel het vectorveld is

$$\mathbf{w}(x, y, z) = w_1(x, y, z) \hat{\mathbf{e}}_x + w_2(x, y, z) \hat{\mathbf{e}}_y + w_3(x, y, z) \hat{\mathbf{e}}_z,$$

met  $w_1, w_2, w_3$  de (scalaire) componentfuncties. Als het veld  $\mathbf{w}$  als `Vectorfield` is gedefinieerd, dan kunnen we in Maple de rotatie en divergentie met de `Nabla`-operator maken:

Nabla

de rotatie  $\nabla \times \mathbf{w}$ :      `Nabla &x w`  
 de divergentie  $\nabla \cdot \mathbf{w}$ :      `Nabla.w`

Hoe het werkt, laten we zien aan de hand van een voorbeeld.

### Voorbeeldopgave

Gegeven is het oppervlak

$$S = \{(x, y, z) \mid z = x^2 + y^2, \quad 0 \leq z \leq 1\},$$

en het vectorveld  $\mathbf{w} = 2yz \hat{\mathbf{e}}_x + xy \hat{\mathbf{e}}_z$ . Maak een tekening van  $S$  en bereken

$$\iint_S (\nabla \times \mathbf{w}) \cdot \hat{\mathbf{n}} \, dA$$

via een parametrisering van  $S$ . Hierbij is  $\hat{\mathbf{n}}$  de eenheidsnormaal op  $S$  met positieve  $z$ -component.

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
```

Het vectorveld  $\mathbf{w}$ :

```
> w := VectorField(<2*y*z,0,x*y>);
w := 2yz e_x + xy e_z
```

```
> rotw := Nabla &x w;
rotw := x e_x + y e_y - 2z e_z
```

Parametrisering van  $S$ :

```
> S := <r*cos(t), r*sin(t), r^2>;
S := r cos(t) e_x + r sin(t) e_y + r^2 e_z
```

We kunnen een plaatje van  $S$  maken:

```
> plot3d(S, r=0..1, t=0..2*Pi,
orientation=[45,75], axes=BOXED );
```

(resultaat niet opgenomen)

Nu de parametrisering van  $S$  invullen in  $\text{rotw}$  (**N.B.**  $0 \leq r \leq 1, 0 \leq t < 2\pi$ ):

```
> rotw_op_S := evalVF(rotw,S);
rotw_op_S := \begin{bmatrix} r \cos(t) \\ r \sin(t) \\ -2r^2 \end{bmatrix}
```

Normaal uitrekenen.

```
> dSdt := diff(S,t);
dSdt := -r sin(t) e_x + r cos(t) e_y

> dSdr := diff(S,r);
dSdr := cos(t) e_x + sin(t) e_y + 2r e_z

> n := dSdt &x dSdr;
n := 2r^2 cos(t) e_x + 2r^2 sin(t) e_y + (-r sin(t)^2 - r cos(t)^2) e_z
```

$\mathbf{n}$  wijst naar beneden. Verhelpen we dit, dan wordt  $\mathbf{n}$ :

```
> n := simplify(-n);
```

$$n := -2r^2 \cos(t) \mathbf{e}_x - 2r^2 \sin(t) \mathbf{e}_y + r \mathbf{e}_z$$

De integrand:

```
> integr := rotw_op_S . n;
```

$$integr := -2r^3 \cos(t)^2 - 2r^3 \sin(t)^2 - 2r^3$$

vereenvoudigen:

```
> integr := simplify(integr);
```

$$integr := -4r^3$$

integreren:

```
> opp_integraal := Int(Int( integr, r=0..1), t=0..2*Pi);
```

$$opp\_integraal := \int_0^{2\pi} \int_0^1 -4r^3 dr dt$$

```
> value(opp_integraal);
```

$$-2\pi$$

```
> Int(Int( rotw_op_S . (dSdr &x dSdt),
           r=0..1), t=0..2*Pi): %=value(%);
```

$$\int_0^{2\pi} \int_0^1 -2r^3 \cos(t)^2 - 2r^3 \sin(t)^2 - 2r^2 (r \sin(t)^2 + r \cos(t)^2) dr dt = -2\pi$$

## Toelichting

Omdat  $S$  de grafiek van de functie  $x^2 + y^2$  is, waarbij  $x, y$  in de eenheidscircelschijf zitten, is het handig om  $x$  en  $y$  in poolcoördinaten te schrijven voor de parametrisering van  $S$ .

Let er steeds op dat de normaal de goede kant op staat! Verder verdient het aanbeveling om regelmatig tussendoor de uitdrukkingen te vereenvoudigen met behulp van `simplify` – hoewel Maple er vaak ook nog wel uitkomt als dat wordt nagelaten, zie het laatste statement.◊

Bij het uitrekenen van de flux-integraal van dit voorbeeld wordt feitelijk maar op twee momenten een beroep gedaan op het wiskundige inzicht van de gebruiker: bij het opstellen van de parametrisering van het oppervlak  $S$  (inclusief de grenzen van de parameters), en bij het kiezen van de oriëntatie van  $S$ . De rest is invullen van de formules uit §17.1. De bibliotheek `VectorCalculus` bevat daarom procedures voor deze invuloefeningen. In de volgende paragrafen zullen we ze systematisch behandelen.

## 17.4 Lijnintegralen

In deze paragraaf behandelen we het berekenen van booglengten en van lijnintegralen van scalar-en vectorvelden.

**Krommen.** Om een lijnintegraal te kunnen berekenen is hoe dan ook een parametrisering van de kromme nodig. Stel  $k$  wordt geparametriseerd met

$$\mathbf{r}(t) = x(t) \mathbf{e}_x + y(t) \mathbf{e}_y + z(t) \mathbf{e}_z, \quad a \leq t \leq b.$$

Path Zo'n parametrisering geven we als volgt aan Maple door met behulp van de functie<sup>44</sup> `Path`:

$$\mathbf{k} := \text{Path}(\langle x(t), y(t), z(t) \rangle, t=a..b);$$

Voor allerlei 'standaardkrommen' zijn er vereenvoudigde karakteriseringen, bijvoorbeeld

- Line
  - Het lijnstuk van punt  $\mathbf{p}$  naar punt  $\mathbf{q}$  is: `Line(p,q)`, met  $\mathbf{p}$  en  $\mathbf{q}$  de plaatsvectoren van  $\mathbf{p}$  en  $\mathbf{q}$ .
  - De kromme die de punten  $\mathbf{p}, \mathbf{q}, \mathbf{r}, \dots$  door rechte lijnstukken met elkaar verbindt: `LineSegments(p,q,r,...)`. Denk er hierbij aan dat, als  $K$  een *gesloten* kromme is, het laatste opgegeven punt gelijk is aan het eerste.
- LineSegments

Pathint **Lijnintegraal van een scalarveld.** Zo'n lijnintegraal (ook wel *padintegraal* genoemd) wordt berekend met het commando `PathInt`. In cartesische coördinaten:

$$\text{Pathint}(f, [x,y,z]=k);$$

Hierin is  $f$  het scalarveld, in de vorm van een expressie met  $x, y, z$  als parameters, en  $k$  de kromme, in de vorm van een `Path`- `Line`- of `LineSegments`-functie. In twee dimensies gaat het net zo, maar dan natuurlijk met twee variabelen.

inert Als derde argument kan eventueel de optie '`inert`' worden meegegeven. In dat geval is het resultaat een integraal-uitdrukking, die met een `value`-commando kan worden berekend.

LineInt **Lijnintegraal van een vectorveld.** Hiervoor is het commando `LineInt`, in het gebruik zo mogelijk nog eenvoudiger dan de lijnintegraal van een scalarveld:

$$\text{LineInt}(w, k);$$

<sup>44</sup>`Path(...)` is geen maple-commando, maar een 'ongedefinieerde functie-aanroep' (*unevaluated function call*), vergelijk §6.9.

Hierin is  $\mathbf{w}$  een vectorveld en  $k$  als boven. De namen van de variabelen hoeven nu niet te worden vermeld, omdat ze al in het vectorveld gegeven zijn.

**Booglengte.** Uiteraard kunnen we de lengte van de kromme  $k$  berekenen met een `PathInt`, waarbij we voor het scalarveld de constante functie 1 nemen. Korter is:

```
ArcLength          ArcLength( <x(t),y(t),z(t)>, t=a..b );
```

Hierin zijn  $x(t)$ ,  $y(t)$ ,  $z(t)$  de coördinaatfuncties van de parametrisering. In twee dimensies gaat het net zo.

### Voorbeeldopgave

Gegeven het scalarveld  $f(x, y, z) = x^3 + y^2 + z$ , het vectorveld  $\mathbf{w} = y^3 z \hat{\mathbf{e}}_x + x^2 z \hat{\mathbf{e}}_y + (x^2 + y^2) \hat{\mathbf{e}}_z$ . De kromme  $k$  is een horizontale cirkel met straal  $R$  op hoogte  $h$ .

Bereken:  $\int_k ds$ ,  $\int_k f ds$  en  $\int_k \mathbf{w} \cdot d\mathbf{r}$ .

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
> cirkel := <R*cos(t), R*sin(t), h>;
      cirkel := R cos(t) e_x + R sin(t) e_y + h e_z
> f := x^3+y^2+z;
      f := x^3 + y^2 + z
> w := VectorField(<y^3*z,x^2*z,x^2+y^2>);
      w := y^3 z e_x + x^2 z e_y + (x^2 + y^2) e_z
> bl := ArcLength( cirkel, t=0..2*Pi, 'inert' );
      bl := \int_0^{2\pi} \sqrt{R^2 \sin(t)^2 + R^2 \cos(t)^2} dt
> booglengte := value(bl) assuming R>0;
      booglengte := 2 R \pi
> padintegraal := PathInt( f,
      [x,y,z]=Path(cirkel,t=0..2*Pi) ) assuming R>0;
      padintegraal := R^3 \pi + 2 R h \pi
> lijnintegraal := LineInt( w, Path(cirkel,t=0..2*Pi) )
      assuming R>0;
      lijnintegraal := -\frac{3 R^4 h \pi}{4}
```



## Toelichting

Merk op dat de inerte vorm van `ArcLength` precies de integraal van §17.1 geeft:  $\int_0^{2\pi} |\mathbf{r}'(t)| dt$ , waarbij  $\mathbf{r}(t)$  de parametrisering van de kromme is.  $\diamond$

**Lijnintegraal in een gradiëntveld.** Een *conservatief veld*  $\mathbf{w}$  is een gradiëntveld, namelijk van een potentiaalfunctie  $\Phi$  en er geldt

$$\int_k \mathbf{w} \cdot d\mathbf{r} = \int_k \nabla\Phi \cdot d\mathbf{r} = \Phi(\mathbf{b}) - \Phi(\mathbf{a})$$

met  $\mathbf{a}$  en  $\mathbf{b}$  het begin- en eindpunt van  $k$ .

De potentiaal van een conservatief veld kan berekend worden met de `ScalarPotential` opdracht `ScalarPotential`.

## Voorbeeldopgave

Gegeven het vectorveld

$$\mathbf{F}(x, y, z) = (x + y)\hat{\mathbf{e}}_x + (x - z)\hat{\mathbf{e}}_y + (z - y)\hat{\mathbf{e}}_z$$

Ga na dat  $\mathbf{F}$  conservatief is, bepaal een potentiaalfunctie en bereken de  $\int_k \mathbf{F} \cdot d\mathbf{r}$ , waarbij  $k$  een kromme van het punt  $\mathbf{a} = (1, 0, -1)$  naar  $\mathbf{b} = (0, -2, 3)$  is.

## Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
> F := VectorField(<x+y,x-z,z-y>);
      F := (x + y) e_x + (x - z) e_y + (z - y) e_z
> Nabla &x F;
      0 e_x
> ScalarPotential(F); Phi := unapply(%, x,y,z):
      1/2 x^2 + y x - z y + 1/2 z^2
> Phi(0,-2,3) - Phi(1,0,-1);
      19/2
```

Ter controle: rechtstreekse berekening waarbij we met een flinke omweg van  $\mathbf{a}$  naar  $\mathbf{b}$  lopen:

```
> LineInt( F, LineSegments(<1,0,-1>,<4,4,4>,<-2,-9,8>,<0,-2,3>) );
      19/2
```

### Toelichting

De rotatie van  $\mathbf{F}$  is  $\mathbf{0}$ , dus het veld is conservatief. Men gaat gemakkelijk na dat de gradiënt van de gevonden  $\Phi$  inderdaad precies het veld  $\mathbf{F}$  is. De lijnintegraal van punt  $\mathbf{a}$  naar  $\mathbf{b}$  is dus onafhankelijk van de weg, en kan met de potentiaalfunctie worden berekend.  $\diamond$

## 17.5 Oppervlakte- en flux-integralen

Het berekenen van dit type integralen gebeurt in grote trekken op een manier die vergelijkbaar is met pad- of lijnintegralen. Het belangrijkste is hier dat we het oppervlak beschrijven waarover geïntegreerd moet worden.  $\diamond$

**Oppervlakken.** Stel dat het oppervlak  $S$  wordt geparametriseerd met

$$\mathbf{r}(u, v) = x(u, v) \mathbf{e}_x + y(u, v) \mathbf{e}_y + z(u, v) \mathbf{e}_z, \quad a \leq u \leq b, \quad c \leq v \leq d.$$

Voor flux-integralen is nu de *oriëntatie* van  $S$  nog belangrijk. Deze wordt vastgelegd door de normaal op  $S$ , namelijk als  $\mathbf{n} = \mathbf{r}_u \times \mathbf{r}_v$ , dan wel als  $\mathbf{r}_v \times \mathbf{r}_u$ . Met de functie **Surface** leggen we de parametrisering vast:

Surface

$$\text{Surface}(\langle x(u, v), y(u, v), z(u, v) \rangle, u=a..b, v=c..d)$$

De *volgorde* waarin de grenzen van de parameters worden gegeven, moet *dezelfde* zijn als de volgorde waarin de beide raakvectoren worden genomen bij het berekenen van de normaal. Het aardige van deze **Surface**-functie is dat als u het woord **Surface** vervangt door **plot3d**, u meteen een plaatje van het oppervlak krijgt, zodat u precies kunt zien of de parametrisering in orde is.

Daarnaast is er nog een aantal standaardoppervlakken beschikbaar, namelijk

- Een rechthoekig blok, als

Box

$$\text{Box}(a1..a2, b1..b2, c1..c2)$$

om het blok  $[a_1, a_2] \times [b_1, b_2] \times [c_1, c_2]$  aan te geven.

- Een bol met straal  $R$  en middelpunt  $\mathbf{m}$  door

Sphere

$$\text{Sphere}(\langle m1, m2, m3 \rangle, R)$$

waarin  $m_1, m_2, m_3$  de coördinaten van het middelpunt zijn.

De eventuele oriëntatie is automatisch naar buiten. Als u de optie 'inward' als laatste argument in de `Box`- of `Sphere`-functie opneemt, heeft dat het verwachte gevolg voor de oriëntatie.

`SurfaceInt`

**De oppervlakte- en flux-integraal.** Het commando `SurfaceInt` berekent de oppervlakte-integraal van een scalarfunctie:

```
SurfaceInt( f, [x,y,z]=oppervlak );
```

Hierin is `f` een scalarveld in de variabelen `x,y,z` en `oppervlak` een `Surface`-, `Box`- of `Sphere`-functie. Als voor het scalarveld de constante functie 1 wordt genomen, dan wordt uiteraard de oppervlakte van  $S$  berekend.

`Flux`

Voor het berekenen van een flux-integraal van een vectorveld door een (georiënteerd) oppervlak is er het commando met de voor de hand liggende naam `Flux`. Hiermee zullen we de voorbeeldopgave van §17.3 (blz. 254) nog eens oplossen.

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
Het vectorveld w:
> w := VectorField(<2*y*z,0,x*y>):
Parametrisering van S:
> S := <r*cos(t), r*sin(t), r^2>:
> Flux( Nabla &x w, Surface(S, r=0..1, t=0..2*Pi ) );
-2π
Met omgekeerde oriëntatie:
> Flux( Nabla &x w, Surface(S, t=0..2*Pi, r=0..1 ) );
2π
```

### Toelichting

Let op de oriëntatie van het oppervlak; bij de berekening met de 'verkeerde' oriëntatie zijn de parameters in de `Surface`-functie in omgekeerde volgorde gegeven.  $\diamond$

Ook bij de `SurfaceInt`- en `Flux`-opdracht is de 'inert'-optie mogelijk.

## 17.6 Solenoïdale velden; de vectorpotentiaal

Om van een solenoïdaal veld een vectorpotentiaal te berekenen, kunnen we beschikken over het commando met de voor de hand liggende naam `VectorPotential`. Deze is echter volstrekt niet uniek, er kan nog een willekeurig gradiëntveld bij worden opgeteld. Immers als  $\mathbf{F}$  een vectorpotentiaal van  $\mathbf{v}$  is, dan is dus  $\nabla \times \mathbf{F} = \mathbf{v}$ . Als we het gradiëntveld  $\nabla\Phi$  hierbij optellen, dan geldt:

$$\nabla \times (\mathbf{F} + \nabla\Phi) = \nabla \times \mathbf{F} + \nabla \times \nabla\Phi = \mathbf{v} + \mathbf{0},$$

dus  $\mathbf{F} + \nabla\Phi$  is óók een vectorpotentiaal van  $\mathbf{v}$ . Maple berekent altijd een vectorpotentiaal waarvan één van de componenten nul is. Maar met enige moeite kunnen we het krijgen zoals we het hebben willen

### Voorbeeldopgave

Ga na dat het vectorveld

$$\mathbf{v}(x, y, z) = (x^2 + yz) \hat{\mathbf{e}}_x - 2y(x + z) \hat{\mathbf{e}}_y + (xy + z^2) \hat{\mathbf{e}}_z$$

solenoïdaal is, en bepaal een vectorpotentiaal met  $\hat{\mathbf{e}}_y$ -component 0.

### Voorbeeldsessie

```
> restart; with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
> v := VectorField(< x^2+y*z, -2*y*(x+z), x*y+z^2 >):
```

Dit veld is solenoïdaal want de divergentie is 0:

```
> Nabla.v;
```

0

We berekenen een vectorpotentiaal:

```
> F := VectorPotential(v);
```

$$F := (-yz^2 - 2xyz - \frac{1}{2}xy^2) \hat{\mathbf{e}}_x + (-x^2z - \frac{1}{2}yz^2) \hat{\mathbf{e}}_y$$

We willen er nu een hebben met de  $y$ -coördinaat nul.

Dat betekent dat we er een gradiëntveld bij moeten optellen waarvan de  $y$ -coördinaat gelijk is aan

```
> -F[2];
```

$$x^2z + \frac{1}{2}yz^2$$

en dat zou dan de gradient moeten zijn van

```
> Int(-F[2], y); hlp := value(%):
```

$$\int x^2 z + \frac{1}{2} y z^2 dy$$

We kunnen dus bij  $\mathbf{F}$  optellen:

```
> gradientveld := Nabla(hlp, [x,y,z]);
```

$$\text{gradientveld} := 2xy z \bar{\mathbf{e}}_x + (x^2 z + \frac{1}{2} y z^2) \bar{\mathbf{e}}_y + (x^2 y + \frac{1}{2} y^2 z) \bar{\mathbf{e}}_z$$

en dan krijgen we

```
> F_alternatief := F + gradientveld;
```

$$F\_alternatief := (-y z^2 - \frac{1}{2} x y^2) \bar{\mathbf{e}}_x + (x^2 y + \frac{1}{2} y^2 z) \bar{\mathbf{e}}_z$$

en inderdaad:

```
> Nabla &x F_alternatief;
```

$$(x^2 + y z) \bar{\mathbf{e}}_x + (-2 y z - 2 x y) \bar{\mathbf{e}}_y + (x y + z^2) \bar{\mathbf{e}}_z$$

### Toelichting

Helaas is er bij `VectorPotential` geen mogelijkheid om bijvoorbeeld één van de coördinaatfuncties voor te schrijven.  $\diamond$

## 17.7 Locale coördinaten

MapToBasis

Met `SetCoordinates` kunnen we Maple ook de opdracht geven om in andere dan cartesische coördinaten te werken. En met `MapToBasis` kunnen we een vector(veld) in een ander coördinatenstelsel laten omrekenen.

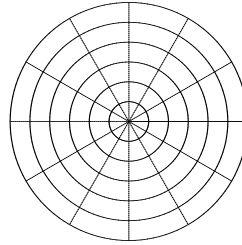
Bij wijze van voorbeeld zullen we een en ander laten zien aan de hand van *poolcoördinaten*. In drie dimensies, bijvoorbeeld met cilinder- of bolcoördinaten gaat het net zo.

**Plaatsvectoren.** In poolcoördinaten wordt een punt in het vlak genoteerd als

$$\mathbf{v} = R \mathbf{e}_r + t \mathbf{e}_\theta,$$

coordplot

dat is het punt met afstand  $r$  tot de oorsprong, terwijl de hoek tussen de positieve  $x$ -as en de vector  $\mathbf{v}$  gelijk aan  $t$  is. Met het commando `coordplot` (in de `plots`-bibliotheek) kan een plaatje worden getekend van een tweedimensionaal coördinatenstelsel. Figuur 14 is het resultaat van `plots:-coordplot](polar)`. Er is een aantal krommen met gelijke  $\mathbf{e}_r$  getekend, namelijk de concentrische cirkels, en een aantal met gelijke  $\mathbf{e}_\theta$ , namelijk de uit de oorsprong vertrekkende halfrechten. Net zoals men in een rechthoekig rooster de (cartesische)  $x$ - en  $y$ -coördinaat van een punt in het vlak kan aflezen, kan men van figuur 14 de (polaire)  $r$ - en  $\theta$ -coördinaat aflezen.



FIGUUR 14. Poolcoördinaten

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(polar[r,theta]);
                                polar_{r,\theta}
> v := <R,t>;
                                v := R e_r + t e_\theta
> MapToBasis( v, cartesian[x,y] );
                                R cos(t) e_x + R sin(t) e_y
```

### Toelichting

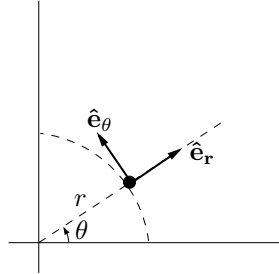
polar

Poolcoördinaten hebben in Maple de naam `polar`, en om aan te geven dat we eenheidsvectoren  $\mathbf{e}_r$  en  $\mathbf{e}_\theta$  willen noemen, zetten we er `[r,theta]` achter (men mag ook andere namen kiezen). Een punt met poolcoördinaten  $(R, t)$  heeft in een Cartesisch assenstelsel inderdaad de coördinaten  $(R \cos t, R \sin t)$ .  $\diamond$

**Vectorvelden,** Een *vectorveld* definiëren we in poolcoördinaten in het algemeen als volgt:

$$\mathbf{v}(r, \theta) = f_1(r, \theta) \hat{\mathbf{e}}_r + f_2(r, \theta) \hat{\mathbf{e}}_\theta. \tag{17.1}$$

Dat wil zeggen: voor een punt met poolcoördinaten  $(r, \theta)$ , dus voor een plaatsvector  $r \mathbf{e}_r + \theta \mathbf{e}_\theta$  (zonder dakjes!) is de waarde van het veld gegeven door (17.1). Hierin vormen de eenheidsvectoren  $\{\hat{\mathbf{e}}_r, \hat{\mathbf{e}}_\theta\}$  een *locaal coördinatenstelsel*, zie figuur 15, en het veld heeft ten opzichte van dit stelsel de coördinaten  $f_1(r, \theta)$  en  $f_2(r, \theta)$ . De eenheidsvector  $\hat{\mathbf{e}}_r$  (met dakje!) wijst in de ‘ $r$ -richting’; daarom heet  $f_1(r, \theta)$  de *radiale component* van  $\mathbf{v}$ . De eenheidsvector  $\hat{\mathbf{e}}_\theta$  wijst in de ‘ $\theta$ -richting’, dat wil zeggen: hij raakt aan de cirkel met straal  $r$ . Daarom wordt  $f_2(r, \theta)$  de *tangentiële component* van  $\mathbf{v}$  genoemd. Deze eenheidsvectoren zijn dus zélf afhankelijk van  $r$  en  $\theta$ , bij het definiëren van



FIGUUR 15. Locale eenheidsvectoren bij poolcoördinaten

een vectorveld moeten we dus voor de parameters dezelfde namen gebruiken als in de namen van de eenheidsvectoren. Het kost enige moeite om Maple te laten zien hoe deze locale eenheidsvectoren er in Cartesische coördinaten uitzien.

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(polar[r,theta]):
De locale eenheidsvector:
> er := VectorField( <1,0> );
                                er := e_r
> erp := evalVF(er,<r,theta>);
                                erp := [ 1 ]
                                       [ 0 ]
> GetRootPoint(erp);
                                re_r + theta e_theta
> MapToBasis(erp, cartesian[x,y]);
                                [ cos(theta) ]
                                [ sin(theta) ]
```

Nu voor  $\hat{e}_\theta$ :

```
> et := VectorField(<0,1>);
                                et := e_theta
> evalVF(et,<r,theta>):
MapToBasis(% , cartesian[x,y]);
                                [ -sin(theta) ]
                                [ cos(theta) ]
```

### Toelichting

De locale eenheidsvectoren moeten we maken als *vectorvelden*, waarvan de ene component 1 is en de andere 0. Deze moeten we dan berekenen in een willekeurig punt  $(r, \theta)$ . We krijgen dan uiteraard dezelfde eenheidsvector, maar dan als *rooted vector* (zie §16.5), met  $(r, \theta)$  (poolcoördinaten!) als beginpunt. En dit moeten we dan weer omzetten in cartesische coördinaten.

Het resultaat hadden we natuurlijk net zo gemakkelijk verkregen door in figuur 15 een paar congruente driehoekjes te tekenen.  $\diamond$

## 17.8 Werken in verschillende coördinatenstelsels

Met `SetCoordinates` legt men vast welk coördinatenstelsel als ‘standaard’ zal worden beschouwd. In drie dimensies worden, naast het Cartesische stelsel, het meest gebruikt *cilindercoördinaten*:

cylindrical `cylindrical[r,theta,zeta]`

en *bolcoördinaten*,

spherical `spherical[r,phi,theta]`

**Vectorvelden tekenen.** Om met `fieldplot` of `fieldplot3d` een plaatje van een in pool- cilinder- of bolcoördinaten gegeven vectorveld te *tekenen* (zie §16.5) kan men het veld het beste eerst omzetten naar cartesische coördinaten. Het kán wel in andere coördinaten, maar het plaatje wordt in het algemeen niet zo mooi. We geven een voorbeeld in twee dimensies (omdat daarvan de plaatjes overzichtelijker zijn dan in drie dimensies). Het vectorveld  $y \hat{e}_x - x \hat{e}_y$  van de voorbeeldsessie op blz. 230 is in poolcoördinaten nóg eenvoudiger te beschrijven, namelijk als  $-r \hat{e}_\theta$ , en we kunnen dan tekenen:

```
plots:-fieldplot( VectorField(<0,-r>), r=0..1,
theta=0..2*Pi );
```

De pijltjes worden nu getekend in de roosterpunten zoals in figuur 14. Met de `grid`-optie kan het aantal roosterpunten (dat wil zeggen: het aantal cirkels en het aantal halfrechten) nog worden beïnvloed. Het (aanbevolen) alternatief is, zoals gezegd, om het veld eerst om te zetten in Cartesische coördinaten.



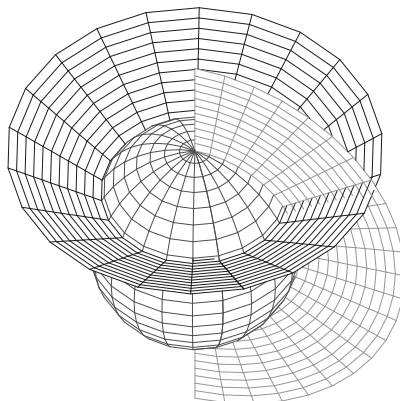
### Voorbeeldsessie

```
> with(VectorCalculus): with(plots):
> SetCoordinates(polar[r,theta]):
> v := VectorField(<0,-r>);
                                v := -r e_θ
> vc := MapToBasis(v, cartesian[x,y]);
                                vc := y e_x - x e_y
> fieldplot( vc, x=-1..1, y=-1..1, arrows=slim, grid=[11,11] );
```

en het resultaat is figuur 11 (blz. 231).

coordplot3d

**Bolcoördinaten.** Met de opdracht `coordplot3d(spherical)` uit de `plots`-bibliotheek maken we een plaatje, zie figuur 16. Nu wordt



FIGUUR 16. Bolcoördinaatvlakken

niet de hele ruimte gevuld met een coördinaten-net (zoals het vlak bij `coordplot`), maar er worden roosters getekend in drie coördinaatoppervlakken, namelijk  $r$  constant (de bol),  $\phi$  constant (de kegel) en  $\theta$  constant (het vlak).<sup>45</sup>

In de volgende voorbeeldsessie laten we een aantal mogelijkheden en valkuilen zien bij het gelijktijdig werken in verschillende coördinatenstelsels.

### Voorbeeldsessie

```
> with(VectorCalculus): with(plots):
> SetCoordinates(spherical[r,phi,theta]):
```

<sup>45</sup>Als figuur 16 in kleur zou zijn, dan waren ze respectievelijk rood, blauw en groen.

```
> vb := VectorField(<0,r,0>);
```

$$vb := r \bar{e}_\phi$$

We berekenen  $|vb|$ :

```
> sqrt(vb.vb) assuming r>0;
```

$$r$$

```
> vc := MapToBasis(vb, cartesian[x,y,z] );
```

$$vc := \frac{xz}{\sqrt{x^2+y^2}} \bar{e}_x + \frac{yz}{\sqrt{x^2+y^2}} \bar{e}_y + \frac{-x^2-y^2}{\sqrt{x^2+y^2}} \bar{e}_z$$

```
> simplify(sqrt(vc.vc));
```

$$\sqrt{x^2+y^2+z^2}$$

Bij het berekenen van het inproduct 'weet' Maple dat het veld  $vc$  in Cartesische coördinaten staat.

```
> GetCoordinates(vb);
```

$$spherical_{r, \phi, \theta}$$

```
> GetCoordinates(vc);
```

$$cartesian_{x, y, z}$$

```
> evalVF(vb, <1,Pi/2,0>);
```

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

```
> evalVF(vc, <1,0,0>);
```

Error, (in VectorCalculus:-evalVF) numeric exception: division by zero

Dat klopt niet, en dat komt omdat niet duidelijk is dat we  $\langle 1,0,0 \rangle$  in Cartesische coördinaten bedoelen.

```
> GetCoordinates(<1,0,0>);
```

$$spherical_{r, \phi, \theta}$$

We maken de juiste plaatsvector:

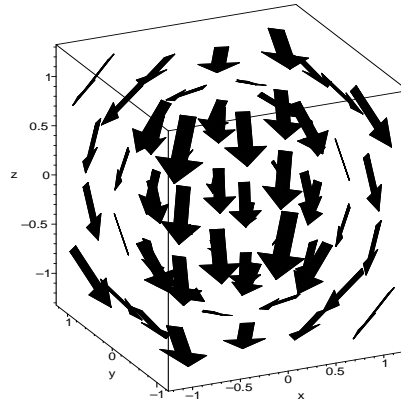
```
> a := Vector( <1,0,0>, attributes=[coords=cartesian[x,y,z]] );
```

$$a := \mathbf{e}_x$$

```
> evalVF(vc,a);
```

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

```
> fieldplot3d(vc, x=-1..1, y=-1..1, z=-1..1,
  grid=[4,4,4], arrows=THICK, axes=boxed,
  color=black, orientation=[-115,70] );
```



```
> w := VectorField( <x,y,z>, cartesian[x,y,z] );
                    w := x e_x + y e_y + z e_z
> MapToBasis( w, spherical[r,phi,theta] ): simplify(%);
                    r e_r
```

### Toelichting

In deze sessie zijn bolcoördinaten het standaard-coördinatenstelsel. Dat betekent dat, als we vectorvelden of plaatsvectoren *maken*, zonder er verder iets bij te vermelden, ze automatisch in bolcoördinaten komen te staan. Dat is dus het geval bij het vectorveld **vb**. Het veld **vc** staat echter in Cartesische coördinaten, daar hebben we immers om gevraagd! Het inproduct wordt in beide gevallen goed berekend, dat wil zeggen: in het juiste coördinatenstelsel. Met **GetCoordinates** kunnen we opvragen in welk stelsel een vector(veld) staat.

**GetCoordinates**

Als we in bolcoördinaten werken, is het *evalueren* van een in afwijkende coördinaten gedefinieerd veld in een bepaald punt een beetje moeizaam. We moeten uiteraard dat punt als plaatsvector opgeven, en dat wordt dan automatisch als een plaatsvector in bolcoördinaten geïnterpreteerd, en krijgen we een foutmelding. Om het punt in Cartesische coördinaten te krijgen, moeten we het maken met **attributes** een **Vector**-commando met een **attributes**-optie. In dit geval was het misschien gemakkelijker geweest om met twee **SetCoordinates**-commando's tijdelijk over te schakelen op Cartesische coördinaten als standaard.

**attributes**

Het definiëren van een *vectorveld* in afwijkende coördinaten is veel gemakkelijker, je kunt gewoon de naam van het coördinatenstelsel als tweede argument meegeven. ◇

Laplacian

**Differentiaaloperatoren in bolcoördinaten.** Ook in bol- (en andere) coördinatenstelsels werkt Nabla naar behoren om de gradiënt, divergentie en rotatie te berekenen. We geven een voorbeeld van het gebruik van de *Laplace-operator*, in Maple: `Laplacian`.

### Voorbeeldopgave

De *Laplace-vergelijking* is:  $\nabla^2 u = 0$ .

Schrijf deze vergelijking uit in cartesische coördinaten.

Geef ook de Laplace-vergelijking in bolcoördinaten in de volgende gevallen:

- (1):  $u$  is een functie van  $r$  alleen;
- (2):  $u$  is een functie van  $r$  en  $\theta$ .

### Voorbeeldsessie

```
> with(VectorCalculus):
> SetCoordinates(cartesian[x,y,z]):
> PDV := Laplacian(u(x,y,z))=0;
      PDV :=  $\frac{\partial^2}{\partial x^2} u(x, y, z) + \frac{\partial^2}{\partial y^2} u(x, y, z) + \frac{\partial^2}{\partial z^2} u(x, y, z) = 0$ 
> SetCoordinates( spherical[r,phi,theta] ):
> PDV1 := expand(Laplacian(u(r))) = 0;
      PDV1 :=  $\frac{2(\frac{d}{dr} u(r))}{r} + \frac{d^2}{dr^2} u(r) = 0$ 
> PDV2 := expand(Laplacian(u(r,theta))) = 0;
      PDV2 :=  $\frac{2(\frac{\partial}{\partial r} u(r, \theta))}{r} + \frac{\partial^2}{\partial r^2} u(r, \theta) + \frac{\partial^2}{\partial \theta^2} u(r, \theta) = 0$ 
```

AddCoordinates

**Eigen coördinatenstelsels.** Naast de tot nu toe behandelde standaard-coördinatenstelsels ken Maple er nog verscheidene. Raadpleeg `?VectorCalculus,Coordinates` voor een overzicht. Met behulp van het commando `AddCoordinates` kunnen nieuwe coördinatenstelsel worden gemaakt. We geven een zeer eenvoudig voorbeeld in twee dimensies.

### Voorbeeldopgave

Definieer de lineaire coördinatentransformatie  $\mathbf{T}$  als

$$x = u - v \quad y = u + v.$$

- (a) Schrijf het vectorveld  $v^2 \hat{\mathbf{e}}_u + u^2 \hat{\mathbf{e}}_v$  en de plaatsvector  $v^2 \mathbf{e}_u + u^2 \mathbf{e}_v$  in Cartesische coördinaten;

- (b) Maak het  $u$ - $v$ -rooster (zoals figuur 14), en teken de grafiek van  $v = 1/u$  in dit rooster.

### Voorbeeldsessie

```
> with(VectorCalculus): with(plots):
> AddCoordinates( T[u,v], [u+v, u-v] );
                                     T
> SetCoordinates( T[u,v] );
> F := VectorField(<v^2,u^2>);
                                     F := v^2 e_u + u^2 e_v
> simplify( MapToBasis( F, cartesian[x,y] ) );
                                     (sqrt(2)x^2 + sqrt(2)y^2) e_x - sqrt(2)xy e_y
                                     4         4         2
> a := <v^2,u^2>;
                                     a := v^2 e_u + u^2 e_v
> MapToBasis( a, cartesian[x,y] );
                                     (v^2 + u^2) e_x + (v^2 - u^2) e_y
> coordplot(T);
```

Error, (in coordplot) Cannot plot coordinate system T

```
> cart := proc(p) # geeft de (x,y)-coördinaten van p
  VectorCalculus:-MapToBasis(p, cartesian[x,y]):
  %[1], %[2]
end proc;
```

Eerst de lijnen met  $u$  constant, namelijk  $u = -4$ ,  $u = -3.5$ ,  $u = -3$  tot en met  $u = 4$ :

```
> plot( [seq( [cart(<u,v>),v=-4..4], u=-4..4, 0.5)],
  color=grey ): p1 := %:
```

Idem met  $v$  constant:

```
> plot( [seq( [cart(<u,v>),u=-4..4], v=-4..4, 0.5)],
  color=grey ): p2 := %:
```

Met het volgende commando krijgen we het plaatje dat we met `coordplot(T)` verwachtten.

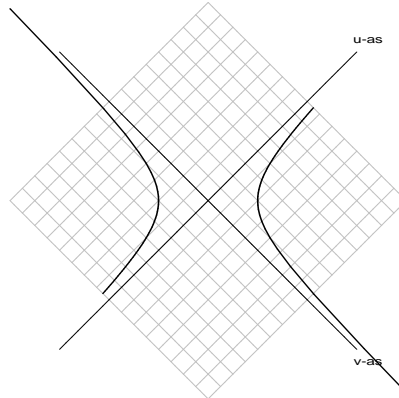
```
> display({p1,p2}, axes=none):
```

Maar nu we toch bezig zijn, kunnen we er ook een  $u$ -as ( $v=0$ ) en een  $v$ -as ( $u=0$ ) in tekenen.

```
> uas := plot( [cart(<u,0>),u=-6..6], color=black ):
> unaam := textplot( [cart(<6.5,0>), "u-as"] ):
> vas := plot( [cart(<0,v>),v=-6..6], color=black ):
> vnaam := textplot( [cart(<0,6.5>), "v-as"] ):
```

Nu de grafiek van  $1/u$ :

```
> grafiek := plot( [cart(<u,1/u>), u=-4..4],
  discont=true, thickness=3, color=black ):
> display({p1,p2,uas,vas,unaam,vnaam,grafiek},
  axes=none, view=[-8..8,-8..8] );
```



### Toelichting

Een nieuw coördinatenstelsel wordt gemaakt met `AddCoordinates` met als eerste argument de naam die we aan het stelsel geven, en als tweede een lijst voor de  $x$  en de  $y$  (en in drie dimensies natuurlijk ook de  $z$ ) in de nieuwe coördinaten. Dat het allemaal goed gegaan is, laat Maple merken door terug te komen met de naam van het nieuwe stelsel. Daar kunnen we verder net zo mee werken als met de bekende coördinatenstelsels.

Helaas is daarop blijkbaar één uitzondering: `coordplot` werkt niet. Dit plaatje moeten we dus zelf maken. Voor de overzichtelijkheid van de benodigde `plot`-commando's gebruiken we een kleine hulp-procedure, `cart`, die de twee Cartesische coördinaten van een in het T-stelsel gegeven vector levert. De roosterlijnen, én de grafiek van  $v = 1/u$  worden als geparametriseerde vlakke krommen getekend (zie §9.2).

De namen “u-as” en “v-as” hebben we aan de *positieve* kant van de assen gezet; merk dus op dat het  $u$ - $v$ -stelsel andersom georiënteerd is als het  $x$ - $y$ -stelsel.  $\diamond$

### Opgave 17.1

Maak voor de volgende integralen een schets van het integratiegebied en bereken de integraal. Bepaal zelf de integratievolgorde.

(a)  $\int_0^1 \int_y^1 e^{x^2} dx dy.$

(b)  $\int_0^1 \int_{y^2}^1 y \frac{\sin(x)}{x} dx dy.$

### Opgave 17.2

Gegeven het gebied  $G$ :

$$G = \{(x, y, z) \mid x^2 + y^2 + z^2 \leq 1, \ x \geq 0, \ y \geq 0, \ z \geq \sqrt{x^2 + y^2}\}.$$

Maak een schets van  $G$  en bereken

$$\iiint_G yz \, dx dy dz.$$

### Opgave 17.3

Gegeven is het gebied

$$G = \left\{ (x, y) \in \mathbb{R}^2 \mid 1 \leq \frac{y^2}{x} \leq 2 \text{ en } 1 \leq \frac{x^2}{y} \text{ en } xy \leq 4 \right\}$$

en de coördinatentransformatie  $\mathbf{T} : (x, y) \mapsto (u, v)$ , gegeven door

$$u = \frac{y^2}{x} \quad v = \frac{x^2}{y}$$

- (a) Maak een schets van  $G$  (in het  $x$ - $y$ -vlak);
- (b) Maak een schets van  $\mathbf{T}(G)$  (in het  $u$ - $v$ -vlak);
- (c) Bereken

$$\iint_G xy^2 \, dx \, dy$$

door over te gaan op  $(u, v)$ -coördinaten. (Los eerst  $x$  en  $y$  op als functies van  $u$  en  $v$ .)

### Opgave 17.4

Bereken de volgende integralen:

(a)

$$\iiint_G (z^2 x^2 + x^2 y^2) \, dx dy dz$$

waarbij  $G = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 \leq 1 \text{ en } -1 \leq z \leq 1\}$ ;

(b)

$$\iiint_G \frac{1}{x^2 + y^2 + z^2} \, dx dy dz$$

waarbij  $G = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 \leq 1 \text{ en } z \geq \frac{1}{2}\}$ .

*Aanwijzingen:*

- Maak de schets op kladpapier (dus niet met Maple). Een verticale doorsnede is genoeg.

- Probeer eerst of Maple de integralen in Cartesische coördinaten kan uitrekenen. Als dat lukt hoeft u natuurlijk geen coördinatentransformatie meer toe te passen.
- Overgang op (bijvoorbeeld) bolcoördinaten: Het eenvoudigst is  $\mathbf{x} := \mathbf{r} * \cos(\phi) * \sin(\theta)$ : enzovoort. Bedenk dat daarna elke  $x$ ,  $y$  en  $z$  automatisch vervangen worden door de juiste uitdrukkingen in  $r$ ,  $\phi$  en  $\theta$ . Bereken de determinant van de Jacobi-matrix van deze transformatie.
- Maple kan ook worden gebruikt om integratiegrenzen te bepalen. Bijvoorbeeld (bij integraal 2): `solve(z=1/2, r)`;
- Als het resultaat niet meteen bevredigend is wil éérst `simplify( Int() )`; nog wel eens helpen.

### Opgave 17.5

De kromme  $k$  is gegeven door  $\mathbf{r}(t) = (t, t^2, \frac{2}{3}t^3)$ , met  $t \in [0, 1]$ .

- (a) Teken de ruimtekromme  $k$ .
- (b) Bereken de lengte van  $k$ .
- (c) Als door een vectorveld  $\mathbf{F}(\mathbf{r}) = (x_1, x_2^2, x_3^3)$  in ieder punt  $\mathbf{r} \in \mathbb{R}^3$  een kracht (uitgedrukt in Newton) naar grootte en richting wordt vastgelegd, bereken dan de arbeid die wordt verricht als een puntmassa van  $(0, 0, 0)$  naar  $(1, 1, \frac{2}{3})$  langs  $k$  beweegt.

### Opgave 17.6

Het oppervlak  $S$  wordt gegeven door

$$S = \{ \mathbf{r} \in \mathbb{R}^3 \mid x_1^2 + x_2^2 - x_3^2 = 4, \text{ en } x_3 \geq 0 \}.$$

Teken  $S$  als geparametriseerd oppervlak, en bepaal in elk punt van  $S$  de eenheidsnormaal met positieve derde component.

### Opgave 17.7

Het oppervlak  $S$  wordt gegeven door

$$S = \left\{ \mathbf{r} \in \mathbb{R}^3 \mid \frac{x_1^2}{9} + \frac{x_2^2}{16} + x_3^2 = 1 \right\}.$$

Teken  $S$  en bepaal in elk punt van  $S$  de naar buiten gerichte eenheidsnormaal.

*Aanwijzing:* Gebruik ellipsoïde-coördinaten:

$$x_1 = 3 \cos(u) \sin(v), \quad x_2 = 4 \sin(u) \sin(v), \quad x_3 = \cos(v).$$



### Opgave 17.8

Teken  $S$  en bereken de oppervlakte-integraal  $\iint_S x_3^2 dA$  als  $S$  de paraboloïde is, gegeven door  $x_3 = x_1^2 + x_2^2$ , met  $0 \leq x_3 \leq 1$ .

### Opgave 17.9

Laat  $S$  de grafiek zijn van de functie  $f : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$  gedefinieerd volgens  $f(u, v) = u + v + 1$ .

Het vectorveld  $\mathbf{w}$  is gegeven door  $\mathbf{w}(\mathbf{r}) = (x_1^2, x_2^2, x_3)$ . Bereken de oppervlakte-integraal  $\iint_S \mathbf{w} \cdot \mathbf{n} dA$ , waarbij  $\mathbf{n}$  de naar boven gerichte eenheidsnormaal is op  $S$ .

### Opgave 17.10

In  $\mathbb{R}^3$  is het vectorveld  $\mathbf{w}$  gegeven door

$$\mathbf{w}(x, y, z) = (zx + z^2y + x, z^3yx + y, z^4x^2),$$

en het oppervlak  $P = P_1 \cup P_2$  met

$$P_1 = \{(x, y, z) \mid x^2 + y^2 + (z - 1)^2 = 1, z \geq 1\}$$

$$P_2 = \{(x, y, z) \mid x^2 + y^2 = 1, 0 \leq z \leq 1\}$$

- Maak een schets van  $P$ .
- Bereken  $\text{rot } \mathbf{w}$ .
- Bereken  $\iint_P \text{rot } \mathbf{w} \cdot \mathbf{n} dA$ , met  $\mathbf{n}$  de eenheidsnormaal op  $P$  met niet-negatieve  $z$ -component welke – op  $P_2$  – van de  $z$ -as ‘afwijst’.

### Opgave 17.11

Gegeven is het vectorveld

$$\mathbf{w}(x, y, z) = (xz + yz^2 + x, xyz^3 + y, x^2z^4)$$

en het oppervlak

$$P = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + (z - 1)^2 = 1, z \geq 1\},$$

dat is de bovenste helft van het boloppervlak met straal 1 en middelpunt  $(0, 0, 1)$ .

Bereken

$$\iint_P \text{rot } \mathbf{w} \cdot \mathbf{n} dA,$$

waarin  $\mathbf{n}$  de naar boven gerichte eenheidsnormaal is

- Met de stelling van Stokes;
- Rechtstreeks.

