

Module 16 Functies van meer variabelen

Onderwerp	Functies van \mathbb{R}^n naar \mathbb{R}^k ; gradiënt, Jacobi-matrix, Hesse-matrix, Inverse en impliciete functiestelling, Optimalisatie onder nevenvoorwaarden.
Voorkennis	Elementaire vectoranalyse, Lineaire algebra.
Expressies	Nabla, Hessian, Jacobian, SetCoordinates, VectorField, evalVF, About, fieldplot, extrema, Minimize, Maximize
Bibliotheken	LinearAlgebra, VectorCalculus, plots, Optimization
Bestanden	meting1.dat
Zie ook	Module 3, 7, 8, 11, 12 en 13.

16.1 Afgeleiden van functies van meer variabelen

Definities. We herhalen eerst de definities van gradiënt, Jacobi-matrix en Hesse-matrix. Zij $f : \mathbb{R}^n \rightarrow \mathbb{R}$ en $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Als $n = m$ dan heet \mathbf{g} een *vectorveld*. Veronderstel dat f en \mathbf{g} twee keer differentieerbaar zijn.

- De *gradiënt* van f , ook wel geschreven als ∇f (spreek uit: *nabla*), is het vectorveld met componenten

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix},$$

ook vaak genoteerd als

$$\nabla f = \frac{\partial f}{\partial x_1} \hat{\mathbf{e}}_{x_1} + \cdots + \frac{\partial f}{\partial x_n} \hat{\mathbf{e}}_{x_n}.$$

Hierin zijn $\hat{\mathbf{e}}_{x_1}, \dots, \hat{\mathbf{e}}_{x_n}$ de *eenheidsvectoren* in het (Cartesische) coördinatenstelsel. Als $n = 3$, dan komt u voor $\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z$ ook vaak $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ tegen.

- De *Hesse-matrix* van f , aangeduid met $D^2 f$ is de $n \times n$ *symmetrische* matrix

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

- De *Jacobi-matrix* van \mathbf{g} , aangeduid met $D_{\mathbf{x}}\mathbf{g}$, is de $m \times n$ matrix

$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_1} & \dots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}.$$

Hierin zijn g_1, \dots, g_m de componentfuncties van \mathbf{g} .

Kettingregel. Als $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ en $\mathbf{g} : \mathbb{R}^m \rightarrow \mathbb{R}^k$ beide differentieerbaar zijn, dan is de samenstelling $\mathbf{g} \circ \mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ differentieerbaar met Jacobi-matrix

$$D_{\mathbf{x}}(\mathbf{g} \circ \mathbf{f}) = D_{\mathbf{f}(\mathbf{x})}\mathbf{g} D_{\mathbf{x}}\mathbf{f}.$$

Ook in Maple kunnen we met functies van meer dan één variabele werken. We onderscheiden een aantal gevallen, die we bespreken aan de hand van voorbeelden.

16.2 Functies van $\mathbb{R}^n \rightarrow \mathbb{R}$

Zo'n scalaire functie van n variabelen wordt vaak een *scalarveld* genoemd (vooral als $n = 3$). De invoering van een scalaire functie (bijvoorbeeld $f(x_1, \dots, x_n)$) gaat in Maple met

```
f := ( x1, ... , xn) -> functievoorschrift;
```

Andersom kan van een expressie waarin meer variabelen voorkomen een functie gemaakt worden met het commando `unapply`:

```
f := unapply( expressie, x1, ..., xn);
```

In veel gevallen zal men echter ermee kunnen volstaan f als *expressie* te definiëren.

De bibliotheek `VectorCalculus`. Bij het werken met functies van meer variabelen zullen we veelvuldig gebruik maken van procedures uit deze bibliotheek. Het meest opvallende gevolg van het laden van `VectorCalculus` is dat vectoren op een andere manier op het scherm worden vertoond. Bijvoorbeeld: op het commando `<2,3>`; reageert Maple normaliter met

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

maar na `with(VectorCalculus)`; wordt dat

$$2\mathbf{e}_x + 3\mathbf{e}_y$$

Als u `VectorCalculus` samen met `LinearAlgebra` wilt gebruiken (hoeft bijna nooit), dan moet u *eerst* `LinearAlgebra` laden, en *daarna* `VectorCalculus`. Dit komt omdat `VectorCalculus` een aantal procedures uit `LinearAlgebra` ‘overschrijft’ (én andersom, en dat is ongewenst).

Nabla
Hessian

Gradiënt, Hesse-matrix. Van een functie van meer variabelen kunnen we de gradiënt en de Hesse-matrix uitrekenen met de commando’s `Nabla` en `Hessian` uit de bibliotheek `VectorCalculus`. Dit gaat dan volgens

```
Nabla(f(x1,...,xn), [x1,...,xn]);
Hessian(f(x1,...,xn), [x1,...,xn]);
```

als `f` met de `->`-operator of met `unapply` is gemaakt. Als `f` als expressie is gedefinieerd, moeten we natuurlijk `Nabla(f, [x1,...,xn])` invoeren. Tussen de vierkante haken staan de variabelen waarnaar gediifferentieerd moet worden. Dit kunnen er eventueel minder zijn dan n . Inplaats van `Nabla` kan men ook `Gradient` gebruiken.

Gradient
diff

Uiteraard kunnen we met de opdracht `diff` of `D` uit Module 6 ook partiële afgeleiden bepalen. Net als bij `diff` hoeft het eerste argument van `Nabla` en `Hessian` niet per se een *Maple-procedure* te zijn. We geven een voorbeeld.

Voorbeeldopgave

Gegeven $f : \mathbb{R}^3 \rightarrow \mathbb{R}, (x, y, z) \mapsto x^2y^3 + yz^4$. Bereken de gradiënt, de Jacobi-matrix en de Hesse-matrix van f . Bereken de partiële afgeleiden $\frac{\partial f}{\partial y}$ en $\frac{\partial^2 f}{\partial y \partial z}$.

Voorbeeldsessie

```
> with(VectorCalculus):
> f := x^2*y^3 + y*z^4;
```

$$f := x^2 y^3 + y z^4$$

Gradiënt:

```
> gradf := Nabla( f, [x,y,z] );
```

$$\text{gradf} := 2 x y^3 \bar{e}_x + (3 x^2 y^2 + z^4) \bar{e}_y + 4 y z^3 \bar{e}_z$$

Jacobimatrix:

```
> jacf := Jacobian( f, [x,y,z] );
```

```
Error, (in VectorCalculus:-Jacobian) invalid input:
VectorCalculus:-Jacobian expects its 1st argument, f, to be of
type {'Vector'}(algebraic), list(algebraic)}, but received
x^2*y^3+y*z^4
```

```

> jacf := Jacobian( <f>, [x,y,z] );
                jacf := [ 2xy3  3x2y2 + z4  4yz3 ]
> whattype(gradf);
                Vectorcolumn
> whattype(jacf);
                Matrix
Hesse-matrix:
> hesf := Hessian( f, [x,y,z] );
                hesf := [ [ 2y3  6xy2  0 ]
                        [ 6xy2  6x2y  4z3 ]
                        [ 0  4z3  12yz2 ] ]
Partiële afgeleide naar y:
> dfdy := diff( f, y );
                dfdy := 3x2y2 + z4
> F := unapply(f, x,y,z);
                F := (x, y, z) → x2y3 + yz4
> D[2](F)(x,y,z);
                3x2y2 + z4
> d2fdydz := diff( f, y, z );
                d2fdydz := 4z3
> D[2,3](F);
                (x, y, z) → 4z3

```

Toelichting

Kennelijk zorgt Maple er zelf voor dat **gradf** een vectorveld, en **hesf** een matrix wordt van de goede dimensies. De eenheidsvectoren in een vectorveld worden met $\bar{\mathbf{e}}_x, \bar{\mathbf{e}}_y, \bar{\mathbf{e}}_z$ inplaats van $\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z$ genoteerd.⁴¹ De gradiënt en de Jacobi-matrix bevatten dezelfde componentfuncties, de gradiënt is echter een *vectorveld* terwijl de Jacobiaan een 1×3 -matrix is. Men wordt voor dit onderscheid al gewaarschuwd doordat Maple als eerste argument voor de procedure **Jacobian** een lijst of vector (in dit geval met dimensie 1) verlangt.

Voor partiële afgeleiden kunnen we de procedure **diff** voor *expressies* gebruiken; het resultaat is weer een expressie. Voor partiële afgeleiden van functies die (door middel van \rightarrow of **unapply**) als *procedure* zijn gedefinieerd, is daarnaast ook het commando **D** beschikbaar. Tussen vierkante haken moeten we daarbij dan aangeven naar welke variabele(n) gedifferentieerd moet worden. Omdat in een definitie als $\mathbf{f} := (\mathbf{x}, \mathbf{y}, \mathbf{z}) \rightarrow \dots$ de x, y en z als dummies optreden kunnen we

D

⁴¹Waarom het niet $\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z$ is (zonder streepjes), wordt in §16.5 uitgelegd.

de namen x , y , z niet gebruiken. We moeten daarom zeggen dat er naar de *eerste*, *tweede*, en/of *derde* variabele gedifferentieerd moet worden. \diamond

16.3 Functies van \mathbb{R}^n naar \mathbb{R}^k

Voor $x \in \mathbb{R}^n$ is nu $\mathbf{f}(x)$ een vector met k componenten. In Maple kunnen we zo'n functie op verschillende manieren definiëren:

```
f := (x1, ..., xn) -> (expr1, ..., exprk);
f := (x1, ..., xn) -> [expr1, ..., exprk];
f := (x1, ..., xn) -> <expr1, ..., exprk>;
f := VectorField( <expr1, ..., exprk> );
```

In het eerste geval (met de *ronde* haakjes⁴²) is $\mathbf{f}(x_1, \dots, x_n)$ een expressierij. Dat is vooral handig bij samenstellingen van afbeeldingen, dus als $\mathbf{f}(x_1, \dots, x_n)$ zélf als argument van een functie $\mathbf{g}: \mathbb{R}^k \rightarrow \mathbb{R}^m$ optreedt. De tweede en derde mogelijkheid komen in de praktijk vrijwel op hetzelfde neer omdat Maple meestal ook een lijst accepteert als een vector is bedoeld. Op de vierde mogelijkheid (als `VectorField`) gaan we verderop in.

Uiteraard kan van een eerder gedefinieerde of uitgerekende lijst of vector met `unapply` een functie worden gemaakt.

! Van een *vectorveld* — dus bijvoorbeeld het resultaat van `Nabla(F, [x,y,z])` — kun je met `unapply` echter niet een ‘gewone’ functie maken. Dat komt omdat een vectorveld (`VectorField`) zélf een soort functie is. Zie verder §16.5.

Jacobian Van zo'n functie kunnen we de Jacobi-matrix uitrekenen met het commando `Jacobian`:

```
Jacobian( f(x1, ..., xn), [x1, ..., xn] );
```

Als $\mathbf{f}: (x_1, \dots, x_n) \mapsto (f_1(x_1, \dots, x_n), \dots, f_k(x_1, \dots, x_n))$, dan is het resultaat van een aanroep van `Jacobian` een $k \times n$ matrix met in het (i, j) ^{de} element de partiële afgeleide $\partial f_i / \partial x_j$.

⁴²Haakjes niet vergeten! want anders wordt het geïnterpreteerd als een rij, met als eerste element `f := (x1, ..., xn) -> expr1`, als tweede element `expr2`, enzovoort.

Voorbeeldopgave

Gegeven $f : \mathbb{R}^3 \rightarrow \mathbb{R}^4$, $(x, y, z) \mapsto (xy, yz, xz, x + y)$. Bereken de Jacobi-matrix van f ;
 Bepaal de samenstelling $g \circ f$ als $g(t, u, v, w) = \sin(tu) + e^{(w/v)}$.

Voorbeeldsessie

```
> with(VectorCalculus):
f is een functie van  $\mathbb{R}^3$  naar  $\mathbb{R}^4$  :
> f := (x,y,z) -> <x*y, y*z, x*z, x+y>: f(x,y,z);
      xy ex1 + yz ex2 + xz ex3 + (x + y) ex4
> jacf := Jacobian( f(x,y,z), [x,y,z] );
```

$$jacf := \begin{bmatrix} y & x & 0 \\ 0 & z & y \\ z & 0 & x \\ 1 & 1 & 0 \end{bmatrix}$$

Een alternatief is om $f(x, y, z)$ niet als vector, maar als *sequence* te definiëren:

```
> f := (x,y,z) -> (x*y, y*z, x*z, x+y): f(x,y,z);
      xy, yz, xz, x + y
```

Dit is handig als we samenstellingen willen maken, bijvoorbeeld met $g : \mathbb{R}^4 \rightarrow \mathbb{R}$:

```
> g := (t,u,v,w) -> sin(t*u) + exp(w/v): g(t,u,v,w);
      sin(tu) + e( $\frac{w}{v}$ )
> (g@f)(a,b,c,d);
```

$$\sin(a b^2 c) + e^{\left(\frac{a+b}{a c}\right)}$$

Voor de Jacobimatrix van f moeten we natuurlijk van $f(x, y, z)$ een vector maken:

```
> Jacobian( <f(x,y,z)>, [x,y,z] );
```

$$\begin{bmatrix} y & x & 0 \\ 0 & z & y \\ z & 0 & x \\ 1 & 1 & 0 \end{bmatrix}$$

Toelichting

Omdat we niet opgegeven hebben hoe we de eenheidsvectoren in \mathbb{R}^4 willen noemen (zie daarvoor §16.5) verzint Maple zelf de namen e_{x1} enzovoort. De toekenning $f := (x,y,z) \rightarrow \dots$ hebben we met een dubbele punt afgesloten. De uitvoer ervan ziet er namelijk nogal ingewikkeld uit. Dat heeft er onder andere mee te maken dat in bibliotheek **VectorCalculus** vele standaardoperaties (inclusief de gewone vermenigvuldiging $*$!) opnieuw gedefinieerd worden. In de

uitvoer wordt daarom aangegeven dat met de ‘nieuwe’ vermenigvuldiging wordt gewerkt. Dat het goed werkt hebben we gecontroleerd door het resultaat van de aanroep $f(x, y, z)$ wél te tonen. Maple zorgt er zelf voor dat `jacf` een matrix wordt van de goede dimensies. \diamond

16.4 Geparametriseerde krommen

Een bijzonder geval hebben we als $n = 1$ en $k > 1$, dus $g : \mathbb{R} \rightarrow \mathbb{R}^k$. Zo'n functie noemen we een *kromme*, als $k = 2$ een *vlakke kromme* en als $k = 3$ een *ruimtekrumme*. Iets preciezer: eigenlijk is de verzameling $\{g(t) \mid a \leq t \leq b\}$ de kromme, en is de functie g een *parametrisering* van de kromme. Zie ook §9.2 (vlakke krommen) en §10.2 (ruimtekrummen).

Voorbeeldopgave

De vlakke kromme k heeft parametrisering

$$g(t) = \sqrt{t} \cos t \mathbf{e}_x + \sqrt{t} \sin t \mathbf{e}_y, \quad 0 \leq t \leq 2\pi.$$

Bepaal de raaklijn aan k in het punt $g(\frac{\pi}{2}) = \sqrt{\frac{\pi}{2}} \mathbf{e}_y$, en teken deze raaklijn samen met k in één figuur.

Voorbeeldsessie

```
> with(VectorCalculus):
> g := <sqrt(t)*cos(t), sqrt(t)*sin(t)>;
      g := sqrt(t)cos(t) e_x + sqrt(t)sin(t) e_y
> kromme := [g[1], g[2], t=0..2*Pi];
      kromme := [sqrt(t)cos(t), sqrt(t)sin(t), t = 0..2 pi]
> r := diff(g,t);
      r := (1/2 * cos(t)/sqrt(t) - sqrt(t)sin(t)) e_x + (1/2 * sin(t)/sqrt(t) + sqrt(t)cos(t)) e_y
> R := unapply(r,t):
> R(Pi/2); # richtingsvector
      -sqrt(2)*sqrt(pi)/2 e_x + sqrt(2)/2*sqrt(pi) e_y
> G := unapply(g,t):
> G(Pi/2); # steunvector
      sqrt(2)*sqrt(pi)/2 e_y
```

Parametrisering raaklijn:

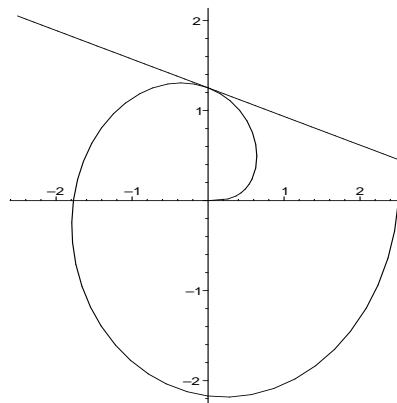
```
> hlp := G(Pi/2) + t*R(Pi/2);
```

$$hlp := -\frac{t\sqrt{2}\sqrt{\pi}}{2} \mathbf{e}_x + \left(\frac{\sqrt{2}\sqrt{\pi}}{2} + \frac{t\sqrt{2}}{2\sqrt{\pi}}\right) \mathbf{e}_y$$

```
> raaklijn := [hlp[1], hlp[2], t=-2..2];
```

$$raaklijn := \left[-\frac{t\sqrt{2}\sqrt{\pi}}{2}, \frac{\sqrt{2}\sqrt{\pi}}{2} + \frac{t\sqrt{2}}{2\sqrt{\pi}}, t = -2..2 \right]$$

```
> plot({kromme,raaklijn}, color=black);
```



Toelichting

We hebben de parametrisering \mathbf{g} als een gewone vector (dus *niet* met een \rightarrow -operator) gemaakt. De twee componentfuncties $g_1(t), g_2(t)$ krijgen we nu met $\mathbf{g}[1]$, $\mathbf{g}[2]$; deze worden samen met de range voor t in de lijst gezet die aan het `plot`-commando gevoerd kan worden.

De vector \mathbf{g} kan gewoon naar t worden gedifferentieerd, en levert de raakvector $\mathbf{g}'(t)$ (dus afhankelijk van t , waarmee een punt op de kromme wordt vastgelegd).

Om geen `subs`-commando te hoeven gebruiken, maken we vervolgens met `unapply` functies van \mathbf{g} en \mathbf{g}' . De raaklijn in het punt $\mathbf{g}(\frac{\pi}{2})$ wordt nu geparametriseerd met $\mathbf{g}(\frac{\pi}{2}) + t\mathbf{g}'(\frac{\pi}{2})$, dat wil zeggen: het is de lijn met de plaatsvector $\mathbf{g}(\frac{\pi}{2})$ als steunvector, en de raakvector $\mathbf{g}'(\frac{\pi}{2})$ als richtingsvector. \diamond

Een *ruimtecurve* tekent men gewoon — na de `plots`-bibliotheek geladen te hebben — met

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

Zelfs de bibliotheek `VectorCalculus` is hierbij niet nodig.

16.5 Plaatsvectoren en vectorvelden

Het zal de oplettende lezer wellicht zijn opgevallen dat Maple soms eenheidsvectoren noteert als $\bar{\mathbf{e}}_x, \bar{\mathbf{e}}_y$, enzovoort, en soms als $\mathbf{e}_x, \mathbf{e}_y$. De betekenis van de eenheidsvectoren met of zonder streepjes (of dakjes) is verschillend. We beperken ons hier even tot twee dimensies, in hogere dimensie gelst precies hetzelfde.

Met $a\mathbf{e}_x + b\mathbf{e}_y$ wordt een ‘vaste’ vector bedoeld, bijvoorbeeld de *plaatsvector* met (Cartesische) coördinaten (a, b) in het vlak. Die coördinaten kunnen eventueel parameters bevatten, en zo kunnen we bijvoorbeeld de functie $\mathbf{g}(t) = a(t)\mathbf{e}_x + b(t)\mathbf{e}_y$ maken met

$$\mathbf{g} := t \rightarrow \langle a(t), b(t) \rangle;$$

dat is de parametrisering van een vlakke kromme, en bijvoorbeeld $\mathbf{g}(4)$ is (de plaatsvector van) een punt op deze kromme.

Met de notatie $a\hat{\mathbf{e}}_x + b\hat{\mathbf{e}}_y$ wordt een *vectorveld* aangeduid, dat is een functie van x en y , die aan elk punt (x, y) in \mathbb{R}^2 een vector (een ‘pijlje’) toekent. Als a en b constanten zijn, dan is dat overal dezelfde vector, namelijk ter grootte a in de x -richting en ter grootte b in de y -richting. Uiteraard zullen de a en b in het algemeen ook afhankelijk van x en y zijn.

In dit geval geven de x en y in $\hat{\mathbf{e}}_x, \hat{\mathbf{e}}_y$ aan dat het vectorveld als functie van x en y moet worden opgevat. Als we in Maple zo’n vectorveld willen maken, dan moeten we daarom eerst opgeven in welke coördinaten we werken, en hoe we de eenheidsvectoren noemen.

SetCoordinates Dat gaat met **SetCoordinates**. Het vectorveld zélf wordt gemaakt met **VectorField**.

evalVF Om de waarde van het veld in een zeker punt uit te rekenen is het speciale commando **evalVF** bedoeld. Het resultaat is een zogenaamde *rooted vector*, die als een ‘gewone’ kolomvector wordt genoteerd. Hoewel alleen de vector zelf op het scherm wordt vertoond, onthoudt Maple wel degelijk het beginpunt van zo’n vector, en met

GetRootPoint **GetRootPoint** kunnen we het weer opvragen.

In **VectorCalculus** worden dus in feite drie ‘soorten vectoren’ gebruikt:

- Plaatsvectoren. Gemaakt met $\mathbf{v} := \langle a, b, c \rangle$; en op het scherm vertoond als $a\mathbf{e}_x + b\mathbf{e}_y + c\mathbf{e}_z$;
- Vectorvelden. Gemaakt met $\mathbf{v} := \text{VectorField}(\langle a, b, c \rangle)$; en op het scherm vertoond als $a\bar{\mathbf{e}}_x + b\bar{\mathbf{e}}_y + c\bar{\mathbf{e}}_z$;

- *Rooted vectors.* Het resultaat van `evalVF(v, <p,q,r>)`; en op het scherm vertoond als een kolomvector.⁴³
- About Wat voor soort vector(veld) \mathbf{v} is, kunnen we opvragen met `About`. (denk aan de hoofdletter A want `about` doet iets anders.)
- fieldplot Tenslotte kan een tweedimensionaal vectorveld worden *getekend* met `fieldplot` uit de `plots`-bibliotheek. Voor driedimensionale vectorvelden is er `fieldplot3d`.

Voorbeeldopgave

Maak het vectorveld $\mathbf{G}(x, y) = y \hat{\mathbf{e}}_x - x \hat{\mathbf{e}}_y$, bereken $\mathbf{G}(1, 2)$ en teken een plaatje.

Voorbeeldsessie

```
> with(VectorCalculus): with(plots):
> SetCoordinates(cartesian[x,y]);
                               cartesian_x, y
> G := VectorField(<y,-x>);
                               G := y e_x - x e_y
> evalVF(G,<1,2>);
                               [ 2 ]
                               [-1 ]
> fieldplot(G, x=-1..1, y=-1..1,
            arrows=slim, grid=[11,11], tickmarks=[[ ],[ ] ] );
```

(zie figuur 11)

Toelichting

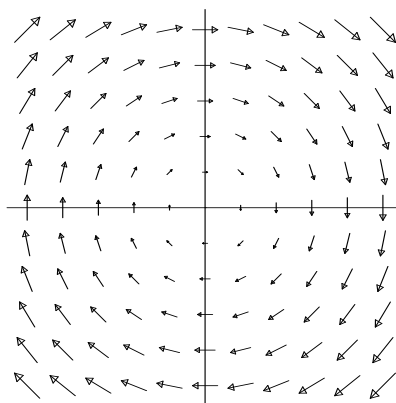
Met de optie `grid` wordt het aantal pijltjes beïnvloed. De pijltjes zijn niet op schaal, maar wel in verhouding getekend. \diamond

De gradiënt is een typisch voorbeeld van een vectorveld. De waarde van de gradiënt in een bepaald punt moet dus met `evalVF` worden berekend.

Voorbeeldopgave

Gegeven $f : \mathbb{R}^3 \rightarrow \mathbb{R}, (x, y, z) \mapsto xy + z$. Bereken de gradiënt van f in het punt $(1, 2, 3)$.

⁴³Uiteraard kan een *rooted vector* ook rechtstreeks worden gemaakt, namelijk met `RootedVector`. Ik heb echter nog geen enkele situatie kunnen bedenken waarin dat zinvol is.



FIGUUR 11. Het vectorveld $y \hat{e}_x - x \hat{e}_y$

Voorbeeldsessie

```
> with(VectorCalculus):
> f := x*y+z;
                                     f := xy + z
> gradf := Nabla(f, [x,y,z]);
                                     gradf := y e_x + x e_y + e_z
> Df123 := evalVF(gradf, <1,2,3>);
```

$$Df_{123} := \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

```
> About(Df123);
```

Type:	Routed Vector
Components:	[2, 1, 1]
Coordinates:	cartesian _{x,y,z}
Root Point:	[1, 2, 3]

Toelichting

Merk op dat het hier geen enkele zin heeft om f als functie te definiëren. In `Nabla` moeten hoe dan ook de variabelen worden gespecificeerd. Omdat deze als `[x,y,z]` worden opgegeven, weet Maple dat de eenheidsvectoren $\hat{e}_x, \hat{e}_y, \hat{e}_z$ zijn. Daarom is het hier zelfs niet eens nodig om `SetCoordinates` te doen. \diamond

In Module 17 gaan we veel uitgebreider in op vectorvelden.

16.6 Inverse functiestelling en impliciete functiestelling

In onderstaande stellingen noteren we met $D_{\mathbf{a}}\mathbf{f}$ de Jacobi-matrix van de functie $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ uitgerekend in het punt $\mathbf{a} \in \mathbb{R}^n$, en met $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{a}, \mathbf{b})$ de Jacobi-matrix van de functie $\mathbf{f} : \mathbb{R}^{n+k} \rightarrow \mathbb{R}^k$ met variabelen $\mathbf{x} \in \mathbb{R}^n$ en $\mathbf{y} \in \mathbb{R}^k$, opgevat als functie van \mathbf{y} , en uitgerekend in het punt $(\mathbf{x} = \mathbf{a}, \mathbf{y} = \mathbf{b})$. Als f een scalarwaardige functie is zullen we ook wel schrijven $f_{\mathbf{y}}(\mathbf{a}, \mathbf{b})$.

Inverse functiestelling. Laat $V \subset \mathbb{R}^n$ open zijn en $\mathbf{a} \in V$. Als

- $\mathbf{f} : V \rightarrow \mathbb{R}^n$ continu differentieerbaar is, en
- $\det D_{\mathbf{a}}\mathbf{f} \neq 0$,

dan is

- (1) \mathbf{f} lokaal rond \mathbf{a} inverteerbaar met continu differentieerbare inverse, en
- (2) de afgeleide van de inverse \mathbf{f}^{-1} in $\mathbf{b} = \mathbf{f}(\mathbf{a})$ is juist de inverse van de afgeleide van \mathbf{f} in \mathbf{a} , ofwel

$$D_{\mathbf{b}}(\mathbf{f}^{-1}) = (D_{\mathbf{a}}\mathbf{f})^{-1}. \quad (16.1)$$

Impliciete functiestelling. Stel dat de afbeelding $f : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^k$ voldoet aan:

- \mathbf{f} is continu differentieerbaar;
- $\mathbf{f}(\mathbf{a}, \mathbf{b}) = \mathbf{0}$ voor zekere $(\mathbf{a}, \mathbf{b}) \in \mathbb{R}^n \times \mathbb{R}^k$;
- $\det \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{a}, \mathbf{b}) \neq 0$,

dan geldt:

- (1) Er is een omgeving U van \mathbf{a} , met daarop *precies één* afbeelding naar \mathbb{R}^k (noem deze \mathbf{y}) zodat:

$$\mathbf{y}(\mathbf{a}) = \mathbf{b} \quad \text{en} \quad \mathbf{f}(\mathbf{x}, \mathbf{y}(\mathbf{x})) = \mathbf{0}, \quad \text{voor alle } \mathbf{x} \in U.$$

Deze afbeelding $\mathbf{y} : U \rightarrow \mathbb{R}^k$ is continu differentieerbaar.

- (2) Voor de Jacobi-matrix in \mathbf{a} van de aldus *impliciet* vastgelegde afbeelding $\mathbf{y}(\mathbf{x})$ geldt bovendien:

$$D_{\mathbf{a}}\mathbf{y} = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}(\mathbf{a}, \mathbf{b})\right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{a}, \mathbf{b}). \quad (16.2)$$

Voorbeeldopgave

Bewijs dat uit het stelsel

$$\begin{aligned}x_1 x_2 - y_1 y_2 &= 2 \\ x_1 y_2 + x_2 y_1 &= 0\end{aligned}$$

in een omgeving van $(x_1, x_2) = (1, 1)$ en $(y_1, y_2) = (1, -1)$ y_1 en y_2 als continu differentieerbare functies van (x_1, x_2) zijn op te lossen. Bepaal $D_{\mathbf{x}}\mathbf{y}$ in $\mathbf{x} = (1, 1)$.

Voorbeeldsessie

```
> with(LinearAlgebra): with(VectorCalculus):
> f := <x1*x2-y1*y2, x1*y2+x2*y1>;
      f := (x1 x2 - y1 y2) e_x + (x1 y2 + x2 y1) e_y
> Dfy := Jacobian( f, [y1,y2] );
```

$$Dfy := \begin{bmatrix} -y_2 & -y_1 \\ x_2 & x_1 \end{bmatrix}$$

```
> DfY := subs( {x1=1, x2=1, y1=1, y2=-1}, Dfy );
```

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

```
> Determinant(DfY);
```

2

Dus ongelijk aan nul.

```
> Dfx := Jacobian( f, [x1,x2] );
```

$$Dfx := \begin{bmatrix} x_2 & x_1 \\ y_2 & y_1 \end{bmatrix}$$

```
> DfX := subs( {x1=1, x2=1, y1=1, y2=-1}, Dfx );
```

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

```
> Dydx := - DfY^(-1).DfX;
```

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

Toelichting

Omdat de determinant van $\frac{\partial f}{\partial \mathbf{y}}(\mathbf{ab}) \neq 0$, is aan de voorwaarden van de impliciete functiestelling voldaan en is het bestaan van de continu

differentieerbare afbeelding $\mathbf{y}(\mathbf{x})$ verzekerd. De partiële afgeleiden van deze functie zijn berekend met (16.2) en

$$\frac{\partial y_1}{\partial x_1}(1, 1) = 0, \quad \frac{\partial y_1}{\partial x_2}(1, 1) = -1, \quad \frac{\partial y_2}{\partial x_1}(1, 1) = 1, \quad \frac{\partial y_2}{\partial x_2}(1, 1) = 0,$$

zoals afgelezen wordt uit de laatste Jacobimatrix $Dydx$. \diamond

Voorals men niet alle partiële afgeleiden van alle impliciet gedefinieerde functies wil bepalen, kunnen ze ook door *impliciet differentiëren* worden gevonden, zie §6.3 (blz. 74)

Voorbeeldopgave

Zie de opgave op blz. 233. Bereken $\frac{\partial y_1}{\partial x_1}$ in het aangegeven punt.

Voorbeeldsessie

```
> f1,f2 := x1*x2-y1*y2=2, x1*y2+x2*y1=0;
      f1, f2 := x1 x2 - y1 y2 = 2, x1 y2 + x2 y1 = 0
> punt := {x1=1,x2=1,y1=1,y2=-1};
      punt := {x1 = 1, x2 = 1, y1 = 1, y2 = -1}
> implicitdiff( {f1,f2}, {y1(x1,x2),y2(x1,x2)}, y1, x1 );
      x1 x2 + y1 y2
      x1 y2 - x2 y1
> eval(% ,punt);
      0
```

Toelichting

We zijn er van uit gegaan dat het punt klopt, en dat aan de voorwaarden van de impliciete functiestelling is voldaan. De procedure `implicitdiff` wordt in dit geval aangeroepen met vier argumenten, achtereenvolgens: de verzameling vergelijkingen, de verzameling impliciet gedefinieerde functies en tenslotte y_1, x_1 omdat we $\frac{\partial y_1}{\partial x_1}$ willen berekenen. \diamond

16.7 Hogere orde afgeleiden van impliciet gedefinieerde functies

Gegeven het stelsel vergelijkingen $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, met $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R}^k$, waarbij in een zeker punt (\mathbf{a}, \mathbf{b}) aan de voorwaarden van de impliciete functiestelling is voldaan. We nemen aan dat \mathbf{f} m keer continu

differentieerbaar is in een omgeving $U \times V$ van (\mathbf{a}, \mathbf{b}) , met $m \in \mathbb{N}$. Dan is de functie $\mathbf{y}(\mathbf{x})$ uit de impliciete functiestelling dit ook. Deze functie voldoet aan $\mathbf{f}(\mathbf{x}, \mathbf{y}(\mathbf{x})) = \mathbf{0}$ voor alle \mathbf{x} in U .

In de vorige paragraaf hebben we de eerste orde partiële afgeleiden van y naar x bepaald. Voor het bepalen van de hogere partiële afgeleiden zijn er verschillende mogelijkheden. We laten deze zien aan de hand van de volgende

Voorbeeldopgave

Gegeven de vergelijking $f(x, y) = 0$, met $f(x, y) = ye^{xy} - 1$. Rond het punt $(0, 1)$ is aan de voorwaarden voor de impliciete functiestelling voldaan, waardoor deze vergelijking een gladde functie $x \mapsto y(x)$ definieert. Bereken $y'(0)$ en $y''(0)$.

Eerste methode: Impliciet differentiëren. Zie §6.9. Voor de onbekende functie $y(x)$ geldt: $y(0) = 1$ en $f(x, y(x)) = 0$ voor alle x in een omgeving van 0, dus ook

$$\frac{d}{dx}f(x, y(x)) = 0 \quad \text{en} \quad \frac{d^2}{dx^2}f(x, y(x)) = 0$$

voor alle x . Uitwerking met Maple geeft dan:

Voorbeeldsessie

```
> f := x -> y(x)*exp(x*y(x))-1;
```

$$f := x \rightarrow y(x) e^{(x y(x))} - 1$$

Differentiëren van $f(x, y(x)) = \text{constant}$ moet de nulfunctie opleveren:

```
> hlp := diff(f(x), x) = 0;
```

$$hlp := \left(\frac{d}{dx} y(x)\right) e^{x y(x)} + y(x) \left(y(x) + x \left(\frac{d}{dx} y(x)\right)\right) e^{x y(x)} = 0$$

In dit geval kan hieruit $\frac{d}{dx} y(x)$, in de bovenstaande expressie aangegeven met $\mathbf{diff}(\mathbf{y}(\mathbf{x}), \mathbf{x})$, worden opgelost:

```
> dydx := solve( hlp, diff(y(x), x) );
```

$$dydx := -\frac{y(x)^2}{1 + x y(x)}$$

De afgeleide van y in $x = 0$ wordt nu berekend door $x = 0$, $y(0) = 1$ te substitueren.

Let op! De *volgorde* waarin wordt gesubstitueerd is van belang; daarom wordt niet $\{x=0, y(0)=1\}$ opgegeven, maar $x=0, y(0)=1$

```
> dy0 := subs( x=0, y(0)=1, dydx );
```

$$dy0 := -1$$

Kijk maar wat er gebeurt als de substituties gelijktijdig worden uitgevoerd:

```
> subs( {x=0, y(0)=1}, dydx );
```

$$-y(0)^2$$

De tweede afgeleide $y''(x)$ verkrijgen we nu eenvoudig door **dydx** te differentiëren:

```
> d2ydx2 := diff(dydx, x);
```

$$d2ydx2 := -\frac{2y(x)\left(\frac{d}{dx}y(x)\right)}{1+xy(x)} + \frac{y(x)^2\left(y'(x) + x\left(\frac{d}{dx}y(x)\right)\right)}{(1+xy(x))^2}$$

...en in het punt $(0, 1)$ is nu alles bekend:

```
> d2y0 := subs( diff(y(x),x)=-1, y(x)=1, x=0, d2ydx2 );
```

$$d2y0 := 3$$

Toelichting

Merk op dat de gevonden uitdrukking voor $y'(x)$, **dydx**, precies overeenkomt met formule (16.2) voor het geval $n = k = 1$, namelijk

$$y'(x) = -\frac{f_x(x, y)}{f_y(x, y)}.$$

We hadden hier ook een uitdrukking voor $y''(x)$ voor willekeurige x (waarbij (x, y) natuurlijk wel moet voldoen aan $ye^{xy} - 1 = 0$) door in de formule voor **d2ydx2** voor **diff(y(x),x)** de eerder gevonden **dydx** te substitueren. Zie ook §4.5 (substitutie). \diamond

Wat we hier ‘stap voor stap’ hebben gedaan is om de procedure **implicitdiff** geautomatiseerd. We berekenen nogmaals de gevraagde $y'(0)$ en $y''(0)$.

Voorbeeldsessie

```
> f := y*exp(x*y)-1;
```

$$f := ye^{xy} - 1$$

```
> punt := {x=0,y=1};
```

```
> dydx := implicitdiff(f,y,x);
```

$$dydx := -\frac{y^2}{1+xy}$$

```
> dy0 := eval(%,punt);
```

$$dy0 := -1$$

```
> d2ydx2 := implicitdiff(f,y,x,x);
```

$$d2ydx2 := \frac{y^3(3+2xy)}{1+3xy+3x^2y^2+x^3y^3}$$

```
> d2y0 := eval( d2ydx2, punt );
```

$$d2y0 := 3$$

Toelichting

Met `implicitdiff` kunnen dus ook hogere afgeleiden worden berekend. \diamond

Tweede methode: Met Taylor-ontwikkelingen. Een methode die ‘met de hand’ meestal zeer bewerkelijk is, maar met Maple goed is te doen, is het berekenen van de n -de orde Taylor-ontwikkeling van de onbekende functie y .

De tweede orde Taylor-ontwikkeling van $f(x, y(x))$ rond 0 is (zonder ordeterm)

$$f(0, 1) + x(f_x + f_y y'(0)) + \frac{1}{2}x^2(f_{xx} + 2f_{xy}y'(0) + f_{yy}y'(0)^2 + f_y y''(0)),$$

waarbij we afkorten $f_x = \frac{\partial f}{\partial x}(0, 1)$ enzovoort. Omdat $f(x, y(x))$ de nulfunctie is, moeten in deze Taylor-ontwikkeling alle coëfficiënten van x^i , $i = 0, 1, 2$ nul zijn. De partiële afgeleiden van f in $(0,1)$ zijn bekend en we hebben voor de diverse afgeleiden van $y(x)$ in $x = 0$ precies genoeg vergelijkingen.

Om deze methode uit te voeren met behulp van Maple, is het gemakkelijk om de tweede orde Taylor-benadering van y rond $x = a$ te schrijven als

$$y_{(2)}(x) = c_0 + c_1(x - a) + \frac{c_2}{2!}(x - a)^2,$$

waarbij we natuurlijk direct voor c_0 de (gegeven!) waarde voor $y(a)$ kunnen invullen. In het algemeen, als we de p^e orde Taylor-benadering willen bepalen van $y(x)$ rond $x = a$, kunnen we schrijven

$$y_{(p)}(x) = c_0 + c_1(x - a) + \dots + \frac{c_p}{p!}(x - a)^p.$$

Toegepast op de opgave krijgen we dan:

Voorbeeldsessie

```
> f := (x,y) -> y*exp(x*y)-1;
```

$$f := (x, y) \rightarrow y e^{(x y)} - 1$$

```
> y := 1 + c1*x + c2/2*x^2;
```

$$y := 1 + c1 x + \frac{1}{2} c2 x^2$$

We hebben nu voor y de tweede orde Taylorontwikkeling van $y(x)$ ingevuld:

```
> f(x,y);
```

$$(1 + c1 x + \frac{1}{2} c2 x^2) e^{(x(1+c1 x+1/2 c2 x^2))} - 1$$

We maken de Taylorontwikkeling van $f(x, y(x))$ rond $x = 0$:

```
> taylor( f(x,y), x=0, 3 ): n := convert( %, polynom );
```

$$n := (1 + c1) x + (\frac{1}{2} + 2 c1 + \frac{c2}{2}) x^2$$

Los $c1$ en $c2$ op uit: $n = 0$ voor alle x

```
> s := solve( identity(n=0, x), {c1,c2} );
```

$$s := \{c1 = -1, c2 = 3\}$$

```
> Y := subs( s, y );
```

$$Y := 1 - x + \frac{3}{2} x^2$$

Nu controleren we of $f(x, y)$ met $y(x) = 1 - x + \frac{3x^2}{2}$ wel van de orde x^3 is:

```
> taylor( f(x,Y), x=0, 3 );
```

$$O(x^3)$$

Dit klopt.

Dus is $y'(0) = -1$ en $y''(0) = 3$.

Toelichting

Zie §5.3 (voorbeeldsessie op blz. 59) voor het oplossen van de identieke vergelijking.

Wanneer we nu $y(x)$ identificeren met haar tweede orde Taylor-ontwikkeling (met coëfficiënten als zojuist berekend), dan zouden de termen van de Taylor-ontwikkeling van f tot op tweede orde nul moeten zijn. Dit klopt inderdaad. \diamond

16.8 Optimalisatieproblemen

Gegeven het optimalisatieprobleem

$$\text{minimaliseer } f(\mathbf{x}), \quad \text{met } \mathbf{x} \in G,$$

waarbij

$$G = \{\mathbf{x} \in \mathbb{R}^n \mid g_j(\mathbf{x}) \leq 0, j = 1, \dots, m; \quad h_i(\mathbf{x}) = 0, i = 1, \dots, k\}.$$

Hierbij zijn de gevallen $m = 0$ en/of $k = 0$ toegestaan. We zullen aannemen dat de functies $g_j, h_i, j = 1, \dots, m, i = 1, \dots, k$ continu differentieerbaar zijn, en dat f twee keer continu differentieerbaar is. Voor $\mathbf{x} \in G$ definiëren we de deelverzameling $J_0(\mathbf{x})$ van $\{1, \dots, m\}$ als

$$J_0(\mathbf{x}) = \{j \in \{1, \dots, m\} \mid g_j(\mathbf{x}) = 0\}.$$

De g_j 's bij een zekere $\mathbf{x} \in G$ met $j \in J_0(\mathbf{x})$ worden wel de *actieve nevenvoorwaarden* genoemd.

Bovendien nemen we aan dat in elk punt $\mathbf{x} \in \partial G$ geldt dat de gradiënten

$$\nabla g_j(\mathbf{x}), j \in J_0(\mathbf{x}), \nabla h_i(\mathbf{x}), i \in \{1, \dots, k\}$$

lineair onafhankelijk zijn.

Dit soort optimalisatieproblemen kunnen we vaak oplossen met het Lagrange-formalisme. Dit gaat in twee stappen.

- (1) Bepaal alle inwendige kritieke punten. Dit zijn oplossingen van de vergelijking $\nabla f(\mathbf{x}) = \mathbf{0}$.
 Bepaal hiervan de aard. Een inwendig kritiek punt $\hat{\mathbf{x}}$ is een **(lokaal) minimum**: als alle eigenwaarden van de Hesse-matrix (zie §16.1) $D^2 f(\hat{\mathbf{x}})$ positief zijn,
(lokaal) maximum: als alle eigenwaarden van $D^2 f(\hat{\mathbf{x}})$ negatief zijn,
zadelpunt: als niet alle eigenwaarden van $D^2 f(\hat{\mathbf{x}})$ hetzelfde teken hebben.
- (2) Bepaal de kritieke punten op de rand. Hiertoe voeren we voor elke deelverzameling $J \subset \{1, \dots, m\}$ in de *Lagrange-functie*:

$$L_J(\mathbf{x}, \mu_j, (j \in J), \lambda_i, (i \in \{1, \dots, k\})) = f(\mathbf{x}) + \sum_{j \in J} \mu_j g_j(\mathbf{x}) + \sum_{i=1}^k \lambda_i h_i(\mathbf{x}).$$

Als nu $\hat{\mathbf{x}}$ een kritiek punt op de rand van G is, dan voldoet $\hat{\mathbf{x}}$ aan het stelsel vergelijkingen $\nabla L_{J_0(\hat{\mathbf{x}})} = \mathbf{0}$, waarbij nu de gradiënt wordt verkregen door $L_{J_0(\mathbf{x})}$ te differentiëren naar \mathbf{x} , naar de μ_j voor $j \in J_0(\hat{\mathbf{x}})$, en naar de λ_i voor $i = 1, \dots, k$.

16.9 Extrema met nevenvoorwaarden in Maple

We bepalen de extrema van een continue functie op een gesloten en begrensde gebied in \mathbb{R}^2 . Aan de voorwaarden van de stelling van Weierstrass is voldaan, zodat we verzekerd zijn van het bestaan van een globaal maximum en een globaal minimum.

Voorbeeldopgave

Gegeven $A = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ en $f : A \rightarrow \mathbb{R}$, met

$$f(x, y) = x^4 + y^4 - 4xy + 8.$$

Bepaal de extrema van f op A , alsmede de aard van de extrema.

Voorbeeldsessie

```
> with(VectorCalculus): with(plots):
> f := (x,y) -> x^4+y^4-4*x*y+8: f(x,y);
      x^4 + y^4 - 4xy + 8
> g := (x,y) -> x^2 + y^2 - 1: g(x,y);
      x^2 + y^2 - 1
Stap 1: Bepaling kritieke punten in het inwendige van A:
> Nabla( f(x,y), [x,y] ): gradf := convert(% ,set);
      gradf := {4x^3 - 4y, 4y^3 - 4x}
> s := solve( gradf, {x,y} );
      s := {x = 0, y = 0}, {x = RootOf(_Z^2 + 1), y = -RootOf(_Z^2 + 1)},
      {x = 1, y = 1}, {x = -1, y = -1},
      {x = RootOf(-RootOf(_Z^2 + 1) + _Z^2),
      y = RootOf(-RootOf(_Z^2 + 1) + _Z^2)RootOf(_Z^2 + 1)}
We verwijderen de niet-reële oplossingen
> Extrema := remove( n -> has(n,I), map(allvalues, [s]) );
      Extrema := [{x = 0, y = 0}, {x = 1, y = 1}, {x = -1, y = -1}]
Controle of ze toegelaten zijn.
> evalb( subs( Extrema[1], g(x,y) ) <= 0 );
      true
> evalb( subs( Extrema[2], g(x,y) ) <= 0 );
      false
> evalb( subs( Extrema[3], g(x,y) ) <= 0 );
      false
Dus alleen de eerste oplossing zit in A.
We bepalen de aard:
> hesf := Hessian( f(x,y), [x,y] );
      hesf :=  $\begin{bmatrix} 12x^2 & -4 \\ -4 & 12y^2 \end{bmatrix}$ 
> hesf := subs( Extrema[1], hesf );
      hesf :=  $\begin{bmatrix} 0 & -4 \\ -4 & 0 \end{bmatrix}$ 
> LinearAlgebra:-Eigenvalues(hesf);
```

$$\begin{bmatrix} 4 \\ -4 \end{bmatrix}$$

Dus het punt (0,0) is zadelpunt.

Stap 2: Nu de randextrema.

```
> lag := f(x,y) + lambda*g(x,y);
      lag := x^4 + y^4 - 4 x y + 8 + lambda (x^2 + y^2 - 1)
> Nabla( lag, [x,y,lambda] ): gradl := convert(%,set);
      gradl := {x^2 + y^2 - 1, 4 y^3 - 4 x + 2 lambda y, 4 x^3 - 4 y + 2 lambda x}
> s := solve( gradl, {x,y,lambda} );
      s := {x = %1, lambda = -3, y = -%1}, {x = %1, y = %1, lambda = 1},
      {x = RootOf(-Z^4 + 1 - Z^2),
      y = RootOf(-Z^4 + 1 - Z^2)^3 - RootOf(-Z^4 + 1 - Z^2),
      lambda = -2}
      %1 := RootOf(2 Z^2 - 1)
> randextrema := remove( n->has(n,I), map(allvalues,[s]) );
      randextrema := [{lambda = -3, x = sqrt(2)/2, y = -sqrt(2)/2},
      {lambda = -3, x = -sqrt(2)/2, y = sqrt(2)/2}, {lambda = 1, x = sqrt(2)/2, y = sqrt(2)/2},
      {lambda = 1, y = -sqrt(2)/2, x = -sqrt(2)/2}]
```

We bepalen de bijbehorende waarden van f .

```
> for i from 1 to 4 do
  '(x,y) =', subs(randextrema[i],[x,y]),
  ' f =', subs( randextrema[i], f(x,y) );
end do;
```

$$(x, y) =, \left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right], \quad f =, \frac{21}{2}$$

$$(x, y) =, \left[-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right], \quad f =, \frac{21}{2}$$

$$(x, y) =, \left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right], \quad f =, \frac{13}{2}$$

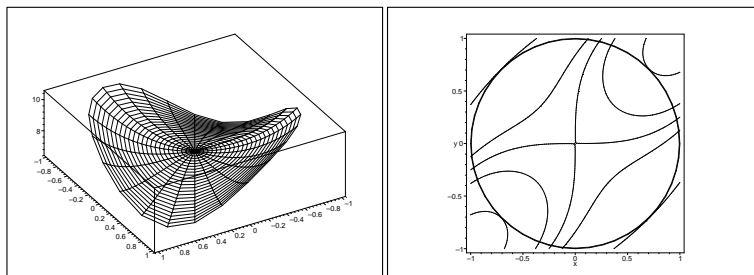
$$(x, y) =, \left[-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right], \quad f =, \frac{13}{2}$$

Dus globaal maximum $\frac{21}{2}$ en globaal minimum $\frac{13}{2}$ (Op grond van de stelling van Weierstrass).

We illustreren dit met de grafiek van f beperkt tot A .

```
> x := r*cos(phi): y := r*sin(phi):
> plot3d( [x,y,f(x,y)], r=0..1, phi=-Pi..Pi,
  style=hidden, color=black, axes=boxed,
  orientation=[60,30] );
> x:='x': y:='y':
```

```
> p1 := contourplot(f(x,y), x=-1..1, y=-1..1,
contours=[7.5,8,8.5,21/2,13/2], grid=[100,100],
color=black):
> p2 := plot( [cos(t),sin(t),t=-Pi..Pi], thickness=3):
> display({p1,p2}, axes=boxed);
```



FIGUUR 12. Grafiek en niveaverzamelingen van $f(x,y) = x^4 + y^4 - 4xy + 8$

Toelichting

Zie het commentaar in de voorbeeldsessie en figuur 12.

In het rechterplaatje is te zien dat in het zadelpunt $(0,0)$ de nivea-krommen snijden en dat in de randextrema de nivea-krommen raken aan de rand van G : de minima liggen linksonder en rechtsboven; de maxima linksboven en rechtsonder. \diamond

extrema

In de Maple-procedure **extrema** is het Lagrange-formalisme ingebouwd. We geven een voorbeeld hoe dit commando gebruikt kan worden aan de hand van de voorbeeldopgave op blz. 240.

Voorbeeldsessie

```
> f := (x,y) -> x^4+y^4-4*x*y+8:
> g := (x,y) -> x^2 + y^2 - 1:
```

De Maple-procedure **extrema** kan ook behulpzaam zijn bij het zoeken naar de randextrema:

```
> extrema( f(x,y), {g(x,y)}, {x,y}, 's');
{13/2, 21/2}
```

De laatste (uitvoer-)variabele bevat de verzameling van kritieke punten, dus de extreempunten:

```
> s;
```

$$\left\{ \left\{ x = -1/2\sqrt{2}, y = -1/2\sqrt{2} \right\}, \left\{ x = -1/2\sqrt{2}, y = 1/2\sqrt{2} \right\}, \right. \\ \left. \left\{ x = 1/2\sqrt{2}, y = -1/2\sqrt{2} \right\}, \left\{ x = 1/2\sqrt{2}, y = 1/2\sqrt{2} \right\} \right\}$$

Toelichting

De procedure `extrema` moet als volgt worden gebruikt:

```
extrema( functie, {nevenvoorwaarden},
        {variabelen}, 'kritieke_punten' )
```

De functie wordt in de vorm van een *expressie* gegeven. De nevenvoorwaarden moeten als een (eventueel lege) *verzameling* van vergelijkingen worden opgegeven. Het derde argument is de verzameling van onafhankelijke variabelen. Het vierde argument (uitvoervariabele) mag eventueel worden weggelaten. De bovenstaande aanroep levert een verzameling die (locale) extrema bevat. \diamond

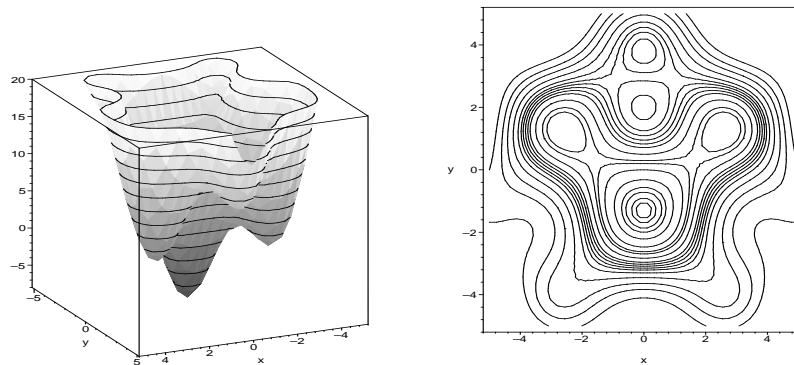
16.10 Numerieke bepaling van extrema

De richting van de gradiënt van een scalarveld in een bepaald punt geeft de richting aan waarin de waarde van het veld vanuit dat punt het meest toeneemt. Van dit idee wordt gebruik gemaakt bij numerieke methoden om (locale) maxima van scalarvelden te zoeken. Daartoe wordt een punt gekozen, liefst in de buurt van het gezochte maximum. In dit punt wordt de gradiënt berekend, en een volgend punt wordt dan gevonden door een klein stukje in de richting van de gradiënt te lopen. Als dat stukje klein genoeg is, dan zal de functiewaarde in het nieuwe punt groter zijn dan die in het oude punt. In dit nieuwe punt wordt weer de gradiënt berekend, en zo voort, totdat de gradiënt ongeveer nul is. In dat geval is er een lokaal maximum gevonden. Het zoeken naar minima gaat precies zo (door in de richting van $-\nabla f$ te lopen, of door een maximum van $-f$ te berekenen). Maple heeft hiervoor de procedures `Minimize` en `Maximize` in de bibliotheek `Optimization`. Deze werken voor één-, twee- en driedimensionale scalarvelden, en zelfs voor functies van meer dan drie variabelen.

`Minimize`
`Maximize`

Voorbeeldopgave

Gegeven de functie $f(x, y) = x^2 + 10 \cos x \sin y + y^2$ (zie figuur 13).
 Bepaal een aantal lokale minima en het globale minimum.
 Bepaal tevens het globale minimum op de cirkelschijf $x^2 + y^2 \leq 1$.



FIGUUR 13. Grafiek van $x^2 + 10 \cos x \sin y + y^2$.

Voorbeeldsessie

```
> f := x^2 + 10*cos(x)*sin(y) + y^2;
      f := x^2 + 10 cos(x) sin(y) + y^2
> with(Optimization):
> Minimize( f, initialpoint=[x=1,y=1] );
      [0.1849, [x = 2.5720, y = 1.2655]]
> Minimize( f, initialpoint=[x=-1,y=3] );
      [8.3156, [x = -0.2870 10^-6, y = 3.8375]]
> Minimize( f, x=-5..5, y=-5..5 );
      [-7.9458, [x = 0.0000, y = -1.3064]]
> Minimize( f, {x^2+y^2<=1} );
      [-7.4147, [x = 0.7230 10^-12, y = -1.0000]]
```

Toelichting

Als we geen voorwaarden aan het domein van de variabelen opleggen, moeten we een beginpunt kiezen met `initialpoint`. In de buurt van $x = 1, y = 1$ vindt Maple een lokaal minimum 0.185 in het punt $x = 2.57, y = 1.27$. Bij een ander startpunt wordt een ander lokaal minimum gevonden.

Figuur 13 suggereert het al, en we kunnen ook gemakkelijk berekenen dat f op \mathbb{R}^2 een globaal minimum heeft (gebruik dat de term $10 \cos x \sin y$ nooit kleiner dan -10 kan worden). Als we grenzen voor x en y geven, dan doet Maple een poging om het globale minimum in het aangegeven gebied te vinden. In dit geval is dat -7.94 bij $x = 0$, $y = -1.31$.

De beperking van f tot de cirkelschijf heeft een globaal minimum op de rand. \diamond

Opgave 16.1

Gegeven $f : \mathbb{R}^4 \rightarrow \mathbb{R}^4$, met

$$f(x, y, z, t) = (xyzt, xyz + yzt, xy, 2 + e^t).$$

Bepaal de Jacobiaan van f in het punt $(0, 1, 4, 17)$.

Opgave 16.2

Gegeven $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, met

$$f(x, y, z) = ye^{\sin(x)} \sin(e^z).$$

en het punt $\mathbf{a} = (x, y, z) = (0, 2, 1)$.

- (a) Bepaal de gradiënt en de Hessiaan van f in het punt \mathbf{a} .
- (b) Bepaal bovendien, gebruikmakend van de berekende gradiënt de lineaire benadering

$$R_{\mathbf{a}}f : \mathbf{x} \mapsto f(\mathbf{a}) + D_{\mathbf{a}}f(\mathbf{x} - \mathbf{a})$$

als *functie*, waarbij \mathbf{a} het gegeven punt is. Vergelijk in een aantal punten in de buurt van \mathbf{a} de waarde van $f(\mathbf{x})$ en $R_{\mathbf{a}}f(\mathbf{x})$.

Aanwijzing bij (b): Gebruik (in dit geval, waarbij f een scalarveld is) voor $D_{\mathbf{a}}f(\mathbf{x} - \mathbf{a})$ het inproduct van de vector $\nabla f(\mathbf{a})$ en de vector $\mathbf{x} - \mathbf{a}$.

Opgave 16.3

De functies $\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ en $\mathbf{g} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ worden gegeven door

$$\begin{aligned} \mathbf{f}(x_1, x_2, x_3) &= (x_1^2 + x_2 + x_3, 2x_1 + x_2 + x_3^2) \\ \mathbf{g}(u_1, u_2, u_3) &= (u_1u_2u_3^2, u_3^2 \sin u_2, u_1^2e^{u_2}) \end{aligned}$$

- (a) Bepaal de Jacobi-matrices $D_{\mathbf{x}}\mathbf{f}$ en $D_{\mathbf{u}}\mathbf{g}$;
- (b) Bepaal $D_{\mathbf{a}}(\mathbf{f} \circ \mathbf{g})$ als $\mathbf{a} = (a_1, 0, a_3)$
 - (i) met de kettingregel, door vermenigvuldiging van twee Jacobi-matrices;
 - (ii) Rechtstreeks, als Jacobi-matrix van $(\mathbf{f} \circ \mathbf{g})(\mathbf{a})$.

Opgave 16.4

Kleinste kwadratenmethode (vervolg van opgave 9.12).

Stel we hebben een serie van n metingen (x_i, y_i) , $i = 1 \dots n$. We veronderstellen dat er tussen de gemeten y -waarden en de x -waarden een verband bestaat van de vorm $y = ax + b$. Gevraagd wordt nu zodanige waarden voor a en b te bepalen dat de meetpunten (x_i, y_i) zo dicht mogelijk bij de grafiek van $ax + b$ liggen.

Bereken daartoe

$$f = \sum_{i=1}^n (y_i - (ax_i + b))^2,$$

de som van de kwadraten van de verticale afstanden van de punten (x_i, y_i) tot de lijn $ax + b$. Omdat alle x_i en y_i bekend zijn, is f alléén een functie van a en b en we zoeken het minimum van deze functie.

Op grond van de probleemstelling is het duidelijk dat er alleen een globaal minimum is (geen maximum) en we (Maple) hoeven alleen a en b op te lossen uit

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} = 0.$$

Met het commando `read("a:meting1.dat");` worden twee lijsten met ‘meetgegevens’ gemaakt (iedere keer als u dit commando geeft een andere lijst). De x -waarden liggen in het interval $[0, 5]$. Let op dat de beide lijsten met *hoofdletters* (X en Y) zijn aangegeven.

- Bepaal de functie $ax + b$ die zo goed mogelijk aansluit bij de meetpunten.
- Bepaal met de methode van de kleinste kwadraten een functie van de vorm $ax^2 + bx + c$ die zo goed mogelijk aansluit bij de meetpunten.
- Maak een tekening met de meetgegevens, de lineaire en de kwadratische benadering.
- Bereken de waarde van f in beide gevallen.

Opgave 16.5

Bepaal of de volgende afbeeldingen lokaal inverteerbaar zijn rond het gegeven punt.

- $f(x) = (\sin(x_1), \cos(x_1x_2))$, in $(\pi, \frac{1}{2})$;
- $f(x) = (e^{x_1x_2}, \log(x_1))$, in $(1, 4)$;
- $f(x) = (x_1x_3, x_1x_2, x_2x_3)$, in $(1, 1, -1)$;

Opgave 16.6

Bewijs voor de volgende functies f dat $f(x, y, z) = 0$ een impliciete functie $z = \phi(x, y)$ oplevert in de aangegeven punten.

- (a) $f(x, y, z) = z^3 - z - xy \sin(z)$, in $(0, 0, 0)$
 (b) $f(x, y, z) = x + y + z - e^{xyz}$, in $(0, \frac{1}{2}, \frac{1}{2})$.

Opgave 16.7

Bewijs dat $f(x, y) = 0$ met $f(x, y) = x^5 + y^5 + xy + 4$ in het punt $(2, -2)$ een impliciete functie $y(x)$ oplevert. Bereken $y'(2)$, $y''(2)$ en $y'''(2)$.

Opgave 16.8

Gegeven is het stelsel vergelijkingen

$$\begin{aligned} -x_1^2 - x_2 + 2y_1^2 + y_2^2 &= 3 \\ -x_1^2 + x_2 + y_1 + 3y_2 &= -2 \end{aligned}$$

- (a) Ga na dat in het punt $(x_1, x_2) = (0, 0)$ en $(y_1, y_2) = (1, -1)$ aan de voorwaarden van de impliciete functiestelling is voldaan, zodat door bovenstaande vergelijkingen impliciet y_1 en y_2 lokaal als functies van x_1 en x_2 gegeven zijn.

We gaan nu de tweede orde partiële afgeleiden van y_1 en y_2 naar x_1 en x_2 bepalen.

- (b) Definieer functies f_1 en f_2 van x_1 , x_2 , y_1 , y_2 door

$$\begin{aligned} f_1(x_1, x_2) &= -x_1^2 - x_2 + 2y_1^2 + y_2^2 - 3 \\ f_2(x_1, x_2) &= -x_1^2 + x_2 + y_1 + 3y_2 + 2 \end{aligned}$$

- (c) Definieer

$$\begin{aligned} y_1 &:= 1 + c_1 x_1 + c_2 x_2 + c_{11} / 2 x_1^2 + c_{12} x_1 x_2 + c_{22} / 2 x_2^2; \\ y_2 &:= -1 + d_1 x_1 + d_2 x_2 + d_{11} / 2 x_1^2 + d_{12} x_1 x_2 + d_{22} / 2 x_2^2; \end{aligned}$$

- (d) Gebruik de procedure `mtaylor` voor het berekenen van de tweede orde Taylor-ontwikkeling van f_1 en f_2 rond $(0, 0)$.
 (e) Construeer de vergelijkingen die moeten worden opgelost (aanzijzing: gebruik de procedure `coeff` of `coeffs`). Los deze vergelijkingen op met `solve`. Wat zijn de variabelen waarnaar moet worden opgelost?
 (f) Controleer de antwoorden door de Taylor-ontwikkelingen van f_1 en f_2 uit te rekenen.

Opgave 16.9

Gegeven zijn de vergelijkingen:

$$x^2 + y^2 - 2 \ln uv + 2 \ln 2 - 10 = 0$$

$$x \ln u + y \ln v - u^2 + 1 - 3 \ln 2 = 0$$

- (a) Bewijs dat door deze vergelijkingen u en v worden gedefinieerd als functies van x en y in een omgeving van $(x, y, u, v) = (1, 3, 1, 2)$.
 (b) Bepaal de partiële afgeleiden van deze functies als uitdrukking in x, y, u en v .

Gebruik formule (16.2) in §16.6.

- (c) Bereken een uitdrukking voor $\frac{\partial^2 v}{\partial x \partial y}$.

Aanwijzing: Vervang (met `subs()`) in de Jacobi-matrix $\frac{\partial(u,v)}{\partial(x,y)}$ de u en v door $u(x, y)$ en $v(x, y)$.

Opgave 16.10

Gegeven is de functie $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ met $f(\mathbf{x}) = x_1 x_2 x_3$.

Onderzoek f op globale extrema onder de nevenvoorwaarden

$$x_1 + x_2 + x_3 = 1 \quad \text{en} \quad x_1^2 + x_2^2 + x_3^2 = 9.$$

Opgave 16.11

Bepaal alle kritieke punten van

$$f(x, y, z) = x^2 + y^2 + z^2 + xyz + x + y + z$$

op de verzameling

$$G = \{(x, y, z) \mid g_1(x, y, z) \leq 0 \text{ en } g_2(x, y, z) \leq 0\},$$

waarin g_1 en g_2 gegeven zijn door

$$g_1(x, y, z) = x^2 + y^2 + z^2 - 4 \quad \text{en} \quad g_2(x, y, z) = x + y + z - 1.$$

Heeft f globale extrema op G ? Zo ja, bepaal deze.

Opgave 16.12

Bepaal alle kritieke punten van $f(x, y, z) = x^2 + y^2 + z^2$ op de verzameling

$$G = \{(x, y, z) \mid g_1(x, y, z) \leq 0 \text{ en } g_2(x, y, z) \leq 0\},$$

waarin g_1 en g_2 gegeven zijn door

$$g_1(x, y, z) = x + y - z \quad \text{en} \\ g_2(x, y, z) = \frac{1}{4}x^2 + \frac{1}{5}y^2 + \frac{1}{25}z^2 - 1.$$

Heeft f globale extrema op G ? Zo ja, bepaal deze.

Opgave 16.13

Gegeven is een blok B met nog onbekende riblengtes. Denk een rechthoekig coördinatensysteem zo aangelegd, dat de oorsprong O samenvalt met een hoekpunt van B en de drie samenkomende ribben in dat hoekpunt op de positieve x -, y - en z -as liggen. Beschouw de lichaamsdiagonaal OP van B . Als bovendien gegeven is dat P een variabel punt is van de bol $x^2 + y^2 + z^2 = a^2$, ($a > 0$, a constant), bereken dan de maximale inhoud van B .

Opgave 16.14

Gegeven de functie $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ met

$$f(\mathbf{x}) = x_1^2 + 2x_1x_2 + x_1 + 2x_2^2 + x_2^3 - 1$$

en

$$A = \{\mathbf{x} \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leq 4\}.$$

gevraagd wordt het *globale maximum* van f op A .

- (a) Bepaal de kritieke punten van f op $\overset{\circ}{A}$ (het inwendige van A), alsmede hun lokale aard;
- (b) Om de kritieke punten van f op ∂A (de rand van A) te bepalen kunt u ∂A parametriseren door $(2 \cos t, 2 \sin t)$, en dan $f(2 \cos t, 2 \sin t)$ bekijken. Gebruik zo nodig `fsolve` voor een numerieke benadering van nulpunten.
- (c) Teken de niveauverzamelingen $\{\mathbf{x} \in \mathbb{R}^2 \mid f(\mathbf{x}) = c\}$. Kies voor c de volgende vier waarden: -1 , $+1$, de functiewaarde die bij het zadelpunt hoort en de maximale waarde van $f(\mathbf{x})$.
Aanwijzing: Veel mooiere plaatjes dan met `contourplot` verkrijgt u door x_1 op te lossen uit $f(\mathbf{x}) = c$ en $[x_1, x_2]$ te tekenen als geparmetriseerde kromme.
- (d) Teken ten slotte de rand van A (in een afwijkende kleur) in hetzelfde plaatje.

