Module 21 Gewone differentiaalvergelijkingen

Onderwerp	Differentiaalvergelijkingen; fasevlak-analyse.
Voorkennis	Gewone differentiaalvergelijkingen.
Expressies	`dsolve`, `DEplot`, `numeric`, `procedurelist`, `listprocedure`, `odeplot`
Bibliotheken	`plots`, `DEtools`
Zie ook	Module 5, 22.

21.1 Stelsels gewone differentiaalvergelijkingen

Een gewone, n^{de} orde (scalar-) differentiaalvergelijking is een vergelijking waarin een functie $x(t)$ en de eerste tot en met de n^{de} afgeleiden van x als onbekenden voorkomen.
Zonder verdere voorwaarden heeft zo'n vergelijking in veel gevallen een algemene oplossing waarin nog n willekeurige constanten voorkomen.
Als van de onbekende functie x in een punt $a \in \mathbb{R}$ de waarden $x(a)$, $x'(a)$ tot en met $x^{(n-1)}(a)$ zijn gegeven, dan spreekt men van een *beginwaardeprobleem*.

Een *stelsel eerste orde differentiaalvergelijkingen* is van de vorm

$$\mathbf{x}'(t) = \mathbf{G}(\mathbf{x}(t), t), \quad \text{met } \mathbf{G} : \mathbb{R}^{n+1} \to \mathbb{R}^n \quad \text{of} \tag{21.1}$$

$$\mathbf{x}'(t) = \mathbf{F}(\mathbf{x}(t)), \quad \text{met } \mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n \tag{21.2}$$

Hierin is $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))$. Het stelsel (21.2), waarin de onafhankelijke variabele t niet expliciet in het rechterlid voorkomt, heet *autonoom* en het stelsel (21.1) is niet-autonoom. Ook hier spreekt men van een beginwaardeprobleem als bovendien $\mathbf{x}(t_0) = \mathbf{x}_0$ is gegeven.

21.2 Het eenvoudigste geval: één eerste orde vergelijking

`dsolve`

Voor het oplossen van differentiaalvergelijkingen maken we gebruik van het commando `dsolve`. Zoals de naam al suggereert, vertoont deze opdracht in de toepassing veel overeenkomsten met `solve` (zie Module 5).

Voorbeeldopgave

Bepaal de oplossing van de differentiaalvergelijking $y'(t) = k\,y(t)$, die voldoet aan de beginvoorwaarde $y(0) = a$.
Teken een grafiek van de oplossing voor verschillende waarden van k en een geschikte waarde van a.

Voorbeeldsessie

```
> DV := diff(y(t),t) = k*y(t);
```

$$DV := \frac{d}{dt}\,\mathrm{y}(t) = k\,\mathrm{y}(t)$$

```
> dsolve( DV, y(t) );
```

$$\mathrm{y}(t) = \_C1\; e^{(k\,t)}$$

Nu met beginvoorwaarde:

```
> opl := dsolve( {DV, y(0)=a}, y(t) );
```

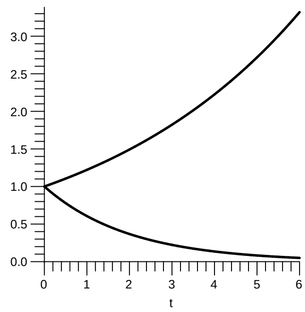
$$opl := \mathrm{y}(t) = a\,e^{(k\,t)}$$

We kiezen de waarde 1 voor de beginwaarde.

```
> a:=1:
> Y1 := subs( {k=-1/2}, rhs(opl) );
```

$$Y1 := e^{(-\frac{t}{2})}$$

```
> Y2 := subs( {k=0.2}, rhs(opl) ):
> plot( {Y1,Y2}, t=0..6, color=black );
```



Toelichting

Hoewel het gebruikelijk is om de differentiaalvergelijking kortweg te noteren als $y' = ky$, is het voor `dsolve` nodig om voluit `y(t)` te schrijven. Maple weet dan meteen dat `y` de naam van de functie

is (de 'afhankelijke variabele'), en `t` de naam van de parameter (de 'onafhankelijke variabele').
Als geen beginwaarde wordt meegegeven, geeft Maple de *algemene oplossing*, met een willekeurige constante (`_C1`) er in. Omdat deze met een *underscore*-teken (`_`) begint, kunnen we zien dat hij door Maple is geïntroduceerd.
De beginvoorwaarde wordt gegeven in de vorm van een tweede vergelijking. We laten Maple dan een stelsel (verzameling) van twee vergelijkingen oplossen: de differentiaalvergelijking zélf en de beginvoorwaarde.
De oplossing komt, net als bij `solve`, in de vorm van een *vergelijking*. De gewenste expressie krijgen we dan door het rechterlid (`rhs`) van de oplossing te nemen. Met `unapply` kunnen we er desgewenst een finctie van maken, maar dat is hier niet nodig omdat we alleen een plaatje willen tekenen. ◇

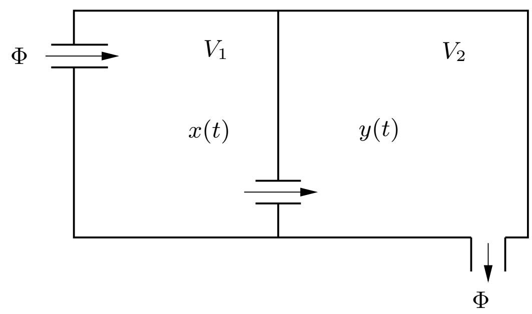
> **!** **Nooit doen:** `assign(opl)`. Hierdoor wordt namelijk de waarde $a\,e^{kt}$ toegekend aan een variabele met de naam `y(t)`. Een aanroep als `y(4)` of `y(x)` is dan niet gedefinieerd. Zie ook §3.2.

21.3 Een inleidend voorbeeld

Aan de hand van een eenvoudig voorbeeld laten we in vogelvlucht diverse mogelijkheden zien om Maple te gebruiken bij het analyseren van een stelsel differentiaalvergelijkingen.

Voorbeeldopgave

Een bak van V liter is door een tussenschot met een niet al te grote doorstroomopening verdeeld in twee compartimenten van respectievelijk V_1 en V_2 liter. Zie figuur 39. Op het tijdstip $t = 0$ zijn beide compartimenten gevuld met een zoutoplossing in een concentratie van c kg/l.
Men voert nu per seconde Φ liter schoon water toe in het linkercompartiment en men voert een gelijke stroom zoutoplossing uit het rechtercompartiment af. De zoutconcentratie in beide compartimenten kan worden beschreven als functie van de tijd: in het linkercompartiment $x(t)$ en in het rechtercompartiment $y(t)$. Deze functies



FIGUUR 39. In twee compartimenten verdeelde waterbak

zullen voldoen aan de volgende differentiaalvergelijkingen:

$$\left\{\begin{aligned} V_1\,x'(t) &= -\Phi\,x(t) \\ V_2\,y'(t) &= \Phi\,x(t) - \Phi\,y(t)\,. \end{aligned}\right. \tag{21.3}$$

Gevraagd wordt om uit deze vergelijkingen x en y als functies van t op te lossen.

Als we dit 'met de hand' zouden doen, dan zouden we de eerste vergelijking van (21.3) apart oplossen, omdat alleen $x(t)$ in deze vergelijking voorkomt. Het resultaat kunnen we dan in de tweede vergelijking invullen, zodat dat een vergelijking wordt waarin alleen $y(t)$ als onbekende functie voorkomt. Met Maple kunnen we echter ook direct een *stelsel* van differentiaalvergelijkingen oplossen. Eerst zonder beginvoorwaarden.

Voorbeeldsessie

We voeren de differentiaalvergelijkingen in als een *sequence* van *vergelijkingen*:

```
> restart;
> stelsel := V1*diff(x(t),t) = -Phi*x(t),
             V2*diff(y(t),t) = Phi*x(t) - Phi*y(t);
```

$$stelsel := V1\,(\tfrac{d}{dt}\,\mathrm{x}(t)) = -\Phi\,\mathrm{x}(t),\ V2\,(\tfrac{d}{dt}\,\mathrm{y}(t)) = \Phi\,\mathrm{x}(t) - \Phi\,\mathrm{y}(t)$$

De functies die we willen weten zijn:

```
> functies := x(t), y(t);
```

$$functies := \mathrm{x}(t),\ \mathrm{y}(t)$$

```
> dsolve( {stelsel}, {functies} );
```

$$\left\{\mathrm{y}(t) = \_C1\,e^{(-\frac{\Phi\,t}{V2})} + \frac{V1\,e^{(-\frac{\Phi\,t}{V1})}\,\_C2}{-V2+V1},\ \mathrm{x}(t) = \_C2\,e^{(-\frac{\Phi\,t}{V1})}\right\}$$

Toelichting

Het resultaat van het commando `dsolve` komt in dezelfde vorm als het resultaat van `solve`, namelijk als een *verzameling* van *vergelijkingen*, zie §5.4. Dat is een vorm die direct als invoer voor de procedure `subs` kan worden gebruikt; we zullen dat verderop toepassen.
Merk op dat er twee nieuwe constanten in de oplossing staan, $\_C_1$ en $\_C_2$. We zouden deze constanten kunnen bepalen door de gegevens die we nog niet hebben gebruikt in te vullen, namelijk $x(0) = c$ en $y(0) = c$. Dit zijn zogenaamde *beginvoorwaarden*, en we hadden ze ook direct aan `dsolve` kunnen meegeven. We doen dat in de volgende sessie. ◇

Voorbeeldsessie

We hebben weer hetzelfde stelsel als zojuist:

```
> stelsel := V1*diff(x(t),t) = -Phi*x(t),
             V2*diff(y(t),t) = Phi*x(t) - Phi*y(t):
> functies := x(t), y(t):
```

De beginvoorwaarden zijn voor Maple gewoon twee extra vergelijkingen waaraan de oplossingen moeten voldoen:

```
> beginvoorwaarden := x(0)=c, y(0)=c;
```

$$beginvoorwaarden := \mathrm{x}(0) = c,\ \mathrm{y}(0) = c$$

```
> opl := dsolve( {stelsel,beginvoorwaarden}, {functies} );
```

$$opl := \left\{\mathrm{y}(t) = -\frac{c\,V2\,e^{\left(-\frac{\Phi\,t}{V2}\right)}}{-V2 + V1} + \frac{V1\,e^{\left(-\frac{\Phi\,t}{V1}\right)}\,c}{-V2 + V1},\ \mathrm{x}(t) = c\,e^{\left(-\frac{\Phi\,t}{V1}\right)}\right\}$$

We hadden al gezien dat er onbepaalde constanten in de oplossing verschijnen als er *te weinig* extra voorwaarden zijn. Als er *te veel* voorwaarden zijn zal er geen oplossing mogelijk zijn:

```
> opl2 := dsolve( {stelsel,beginvoorwaarden,x(1)=10}, {functies} );
```

$$opl2 :=$$

Toelichting

We hebben hier te maken met een (eerste orde) beginwaardeprobleem. De *toestand van het systeem* wordt beschreven door de grootheden $x(t)$ en $y(t)$. Het stelsel differentiaalvergelijkingen (21.3) beschrijft hoe het systeem in de tijd evolueert. Dat betekent dat de functies $x(t)$ en $y(t)$ eenduidig vastliggen zodra de toestand op een zeker tijdstip t_0 gegeven is, dus door de beginwaarden $x(t_0)$ en $y(t_0)$. ◇

In de zojuist berekende `opl` zijn x en y nog niet beschikbaar als functies (procedures). De veiligste manier hiervoor is het gebruik van `subs`, eventueel gecombineerd met `unapply` om er een procedure van te maken.

Voorbeeldsessie

Vervolg van de vorige sessie (`opl` is nog bekend):

```
> X := unapply( subs(opl,x(t)), t);
```

$$X := t \to c\, e^{\left(-\frac{\Phi\, t}{V1}\right)}$$

```
> Y := unapply( subs(opl,y(t)), t);
```

$$Y := t \to -\frac{c\, V2\, e^{\left(-\frac{\Phi\, t}{V2}\right)}}{-V2 + V1} + \frac{V1\, e^{\left(-\frac{\Phi\, t}{V1}\right)} c}{-V2 + V1}$$

Toelichting

We hebben hier `opl`, dat is: {`x(t)=`*expr1*`,` `y(t)=`*expr2*} gesubstitueerd in `x(t)`. Dit resulteert in *expr1*, en hiervan hebben we met `unapply` weer een functie gemaakt die we de naam `X` geven. Op dezelfde manier maken we de functie `Y`. We hebben de functies nieuwe namen gegeven (hoofdletter `X` en `Y`) zodat `x(t)` en `y(t)` nog steeds ongedefinieerd zijn. Hierdoor kunnen we hetzelfde stelsel nog eens oplossen, bijvoorbeeld met andere beginvoorwaarden, zonder dat we `x` en `y` dan eerst 'vrij' hoeven te maken.
Deze methode lijkt nogal omslachtig, maar heeft diverse voordelen. We hoeven ons er bijvoorbeeld niet om te bekommeren in welke *volgorde* `x(t)` en `y(t)` in de oplossingsverzameling verschijnen. ◇

Het verdient aanbeveling om zo lang mogelijk met symbolen te blijven werken.[49] De concrete gegevens kunnen het beste worden ingevuld als we écht numerieke resultaten nodig hebben, bijvoorbeeld voor een grafiek. Dat gaat dan als volgt:

Voorbeeldsessie

Vervolg van de vorige sessie (`X` en `Y` nog bekend van de vorige sessie):

```
> with(plots):
> gegevens := {Phi=1.2, V1=8, V2=3, c=0.1};
```

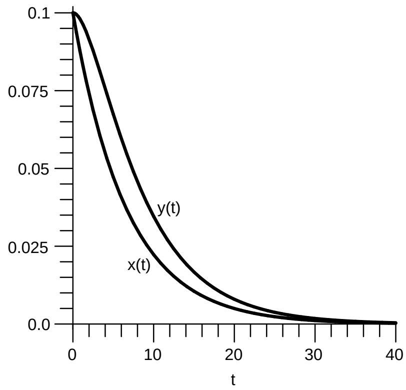
$$gegevens := \{\Phi = 1.2,\ V1 = 8,\ V2 = 3,\ c = 0.1\}$$

```
> p1 := plot( subs(gegevens,X(t)), t=0..40,
              color=black, thickness=2 ):
  p2 := plot( subs(gegevens,Y(t)), t=0..40,
              color=black, thickness=2 ):
```

[49] In sommige gevallen is het wél nuttig om een en ander te veronderstellen omtrent de gebruikte parameters, bijvoorbeeld `assume(c>0)`, als de oplossing niet 'mooi' is. In dit voorbeeld was dat blijkbaar niet nodig.

```
> t1 := textplot( [10,subs(gegevens,X(10)),"x(t)"],
                  align={BELOW,LEFT} ):
  t2 := textplot( [10,subs(gegevens,Y(10)),"y(t)"],
                  align={ABOVE,RIGHT} ):
> display({p1,p2,t1,t2});
```

(zie figuur 40.)



FIGUUR 40. De plot bij de voorbeeldsessie

Toelichting

We hebben hier met `textplot` aangegeven welke grafiek bij $x(t)$ hoort en welke bij $y(t)$. Op een beeldscherm is het natuurlijk eenvoudiger om de grafieken verschillende kleuren te geven. ◇

Speciale gevallen. We bepalen de oplossing met een nieuwe set gegevens, waarin $V_1 = V_2$:

Voorbeeldsessie

Vervolg (`X` en `Y` nog bekend van de vorige sessie):

```
> gegevens2 := {Phi=1.2, V1=5.5, V2=5.5, c=0.1}:
> subs(gegevens2,Y(t));
```

```
Error, numeric exception: division by zero
```

Dit resultaat is niet verwonderlijk als we even kijken naar $\mathrm{Y}(t)$:

```
> Y(t);
```

$$-\frac{c\,V2\,e^{(-\frac{\Phi\,t}{V2})}}{-V2+V1}+\frac{V1\,e^{(-\frac{\Phi\,t}{V1})}\,c}{-V2+V1}$$

Er staat $V1 - V2$ in de noemer en dit is 0 als we `gegevens2` invullen. Voor dit geval moeten we het stelsel apart oplossen:

```
> stelsel2 := op(subs( {V2=V1}, {stelsel} ));
```

$$\mathit{stelsel2} := \mathit{V1}\,(\tfrac{d}{dt}\,\mathrm{x}(t)) = -\Phi\,\mathrm{x}(t),\ \mathit{V1}\,(\tfrac{d}{dt}\,\mathrm{y}(t)) = \Phi\,\mathrm{x}(t) - \Phi\,\mathrm{y}(t)$$

```
> opl2 := dsolve( {stelsel2,beginvoorwaarden}, {functies} ):
> X2 := unapply( subs(opl2,x(t)), t);
> Y2 := unapply( subs(opl2,y(t)), t);
```

$$\mathit{X2} := t \to c\,e^{(-\frac{\Phi\,t}{\mathit{V1}})}$$

$$\mathit{Y2} := t \to \frac{(\Phi\,c\,t + c\,\mathit{V1})\,e^{(-\frac{\Phi\,t}{\mathit{V1}})}}{\mathit{V1}}$$

Toelichting

We moeten van (het oorspronkelijke) `stelsel` eerst een verzameling maken om het in zijn geheel als tweede argument van `subs` te kunnen gebruiken. Van het resultaat ‘verwijderen we de accolades’ door middel van het commando `op` (zie §8.1).
Uiteraard heeft het nieuwe stelsel ook een oplossing; deze is alleen niet te verkrijgen door $V_1 = V_2$ in te vullen in de eerst gevonden oplossing. De grafieken van `X2` en `Y2` zijn niet getekend. Ze hebben dezelfde gedaante als die in figuur 40; `X2` en `Y2` dalen iets sneller. ◇

21.4 Hogere orde differentiaalvergelijkingen

Ook hogere orde differentiaalvergelijkingen kunnen (soms) met `dsolve` exact worden opgelost. We beginnen met een eenvoudig voorbeeld, een lineaire tweede orde vergelijking.met constante coëfficiënten.

Voorbeeldopgave

Bepaal voor verschillende waarden van a, b en c, en voor verschillende beginvoorwaarden $x(0) = x_0, x'(0) = v_0$ de oplossing van de vergelijking $a\,x'' + 2b\,x' + c\,x = 0$.

Voorbeeldsessie

```
> restart;
> DV := a*diff(x(t),t,t) + 2*b*diff(x(t),t) + c*x(t) = 0;
```

$$DV := a\,(\tfrac{d^2}{dt^2}\,\mathrm{x}(t)) + 2\,b\,(\tfrac{d}{dt}\,\mathrm{x}(t)) + c\,\mathrm{x}(t) = 0$$

```
> dsolve( DV, x(t) );
```

$$\mathrm{x}(t) = \_\mathit{C1}\,e^{(-\frac{(b-\sqrt{b^2-c\,a})\,t}{a})} + \_\mathit{C2}\,e^{(-\frac{(b+\sqrt{b^2-c\,a})\,t}{a})}$$

```
> gegevens := {a = 1, b = 1, c = 5}:
> dsolve( {subs(gegevens,DV), x(0)=1, D(x)(0)=0}, x(t) );
```

$$\mathrm{x}(t) = \frac{1}{2}\, e^{(-t)} \sin(2\,t) + e^{(-t)} \cos(2\,t)$$

```
> gegevens := {a = 1, b = 4, c = 7}:
> dsolve( {subs(gegevens,DV), x(0)=6, D(x)(0)=0}, x(t) );
```

$$\mathrm{x}(t) = 7\, e^{(-t)} - e^{(-7\,t)}$$

Toelichting

Een beginvoorwaarde als $x'(0) = 0$ kan worden opgegeven in de vorm `D(x)(0)=0`. ◇

Vooral bij niet-lineaire vergelijkingen kunnen er weleens complicaties optreden.

Voorbeeldopgave

Bereken de oplossing van het beginwaardeprobleem

$$y'' = k^2 \sqrt{1 + (y')^2}, \quad y(0) = 1,\ y'(0) = 0. \tag{21.4}$$

Voorbeeldsessie

```
> restart;
> DV := diff(y(x),x$2) = k^2*sqrt(1+diff(y(x),x)^2);
```

$$DV := \frac{d^2}{dx^2}\,\mathrm{y}(x) = k^2 \sqrt{1 + (\frac{d}{dx}\,\mathrm{y}(x))^2}$$

```
> s := dsolve( {DV, y(0)=1, D(y)(0)=0}, {y(x)} );
```

$$s := \mathrm{y}(x) = \frac{\cosh(x\,k^2) + k^2 - 1}{k^2},\ \mathrm{y}(x) = \frac{-\cosh(x\,k^2) + k^2 + 1}{k^2}$$

Controle eerste oplossing:

```
> subs(s[1],DV);
```

$$\frac{\partial^2}{\partial x^2}\,(\frac{\cosh(x\,k^2) + k^2 - 1}{k^2}) = k^2 \sqrt{1 + (\frac{\partial}{\partial x}\,(\frac{\cosh(x\,k^2) + k^2 - 1}{k^2}))^2}$$

```
> simplify(%, assume=real);
```

$$\cosh(x\,k^2)\,k^2 = \cosh(x\,k^2)\,k^2$$

Dat klopt.
Controle tweede oplossing:

```
> simplify( subs(s[2],DV), assume=real );
```

$$-\cosh(x\,k^2)\,k^2 = \cosh(x\,k^2)\,k^2$$

Klopt dus niet!

In feite heeft Maple opgelost:

```
> DV2 := map(`^`,DV,2);
```

$$DV2 := (\frac{d^2}{dx^2}\,\mathrm{y}(x))^2 = k^4\,(1 + (\frac{d}{dx}\,\mathrm{y}(x))^2)$$

```
> s2 := dsolve( {DV2, y(0)=1, D(y)(0)=0}, {y(x)} );
```

$$s2 := \mathrm{y}(x) = \frac{1}{2}\,\frac{e^{(x\,\sqrt{k^4})}}{k^2} + \frac{1}{2}\,\frac{1}{e^{(x\,\sqrt{k^4})}\,k^2} + \frac{k^2-1}{k^2},$$
$$\mathrm{y}(x) = -\frac{1}{2}\,\frac{e^{(x\,\sqrt{k^4})}}{k^2} - \frac{1}{2}\,\frac{1}{e^{(x\,\sqrt{k^4})}\,k^2} + \frac{k^2+1}{k^2}$$

```
> map( convert, {s2}, trigh ):
  normal(simplify(%)) assuming real; s2 := %:
```

$$\{\mathrm{y}(x) = -\frac{\cosh(x\,k^2) - k^2 - 1}{k^2},\ \mathrm{y}(x) = \frac{\cosh(x\,k^2) + k^2 - 1}{k^2}\}$$

```
> s2 minus normal(simplify({s}));
```

$$\{\}$$

Toelichting

Oplossingen van niet-lineaire vergelijkingen moeten altijd met enige argwaan worden bekeken. Het kan nooit kwaad om de oplossingen te controleren door ze te substitueren in de differentiaalvergelijking.
De oplossing van de gekwadrateerde versie van de differentiaalvergelijking wordt met e-machten gegeven. Om te kunnen vergelijken met de eerder gevonden oplossingen, herschrijven we deze met een `convert`-commando in hyperbolische functies (sinh en cosh). ◇

Een hogere orde differentiaalvergelijking kan altijd worden herschreven in een stelsel van eerste orde vergelijkingen. Als we $y'(x)$ schrijven als $v(x)$, dan gaat (21.4) over in het stelsel

$$\begin{cases} y'(x) = v(x) \\ v'(x) = k^2\sqrt{1+v(x)^2} \end{cases} \qquad \begin{cases} y(0) = 1 \\ v(0) = 0 \end{cases}$$

Om $y(x)$ te vinden, hoeven we in dit geval alleen de oplossing van de tweede vergelijking te integreren. Maple laten we meteen maar het hele stelsel oplossen.

Voorbeeldsessie

```
> stelsel := diff(y(x),x) = v(x),
  diff(v(x),x) = k^2*sqrt(1+v(x)^2):
> s := dsolve( {stelsel,y(0)=1,v(0)=0}, {y(x),v(x)} );
```

$$s := \Big\{v\,(x) = \sinh\left(xk^2 + i\pi\ (1 + 2\,\_Z1)\right),$$
$$y\,(x) = \frac{\sqrt{1+\left(\sinh\left(xk^2+i\pi\ (1+2\,\_Z1)\right)\right)^2}}{k^2} + \frac{k^2-1}{k^2}\Big\},$$
$$\Big\{v\,(x) = \sinh\left(xk^2 + 2\,i\pi\,\_Z1\right),\, y\,(x) = \frac{\sqrt{1+\left(\sinh\left(xk^2+2\,i\pi\,\_Z1\right)\right)^2}}{k^2} + \frac{k^2-1}{k^2}\Big\}$$

```
> eval([s],_Z1=0):  op( simplify(%) assuming real );
```

$$\left\{v(x) = -\sinh\left(xk^2\right), y(x) = \frac{\cosh\left(xk^2\right) + k^2 - 1}{k^2}\right\},$$
$$\left\{v(x) = \sinh\left(xk^2\right), y(x) = \frac{\cosh\left(xk^2\right) + k^2 - 1}{k^2}\right\}$$

Toelichting

In eerste instantie krijgen we oneindig veel (complexe) oplossingen: die `_Z1` in het antwoord (op het scherm waarschijnlijk als `_Z1~` vertoond) is een willekeurig geheel getal. We nemen $\_Z1 = 0$ om reële oplossingen te krijgen. Daarmee hebben we voor $y(x)$ hebben we twee keer dezelfde (goede) oplossing, maar de eerste oplossing voor $v(x)$ is fout.[50] ◇

21.5 Richtingsveld; grafieken van oplossingen

Voor de *eerste orde scalar differentiaalvergelijking* $x'(t) = f(t, x)$, kunnen we het *richtingsveld* tekenen. In een rechthoek in het (t, x)-vlak tekenen we op een grid vectoren met de richtingscoëfficiënt $f(t, x)$. Een curve $t \mapsto x(t)$ is een oplossing van de differentiaalvergelijking als op ieder tijdstip de afgeleide gelijk is aan $f(t, x)$ en dus de raaklijn aan de curve $(t, x(t))$ de richtingscoëfficiënt $f(t, x)$ heeft. Anders gezegd, als een curve overal raakt aan het getekende richtingsveld dan is het een oplossing van de differentiaalvergelijking.

Voorbeeldopgave

Teken het richtingsveld van de differentiaalvergelijking

$$\frac{dx}{dt} = \sin(t - x)$$

voor $\quad -\pi \le t \le \pi, \quad -\pi \le x \le \pi$.
Teken de grafieken van de oplossingen $x(t)$ die op $t = 0$ gelijk zijn aan respectievelijk -1, 0 en 2.

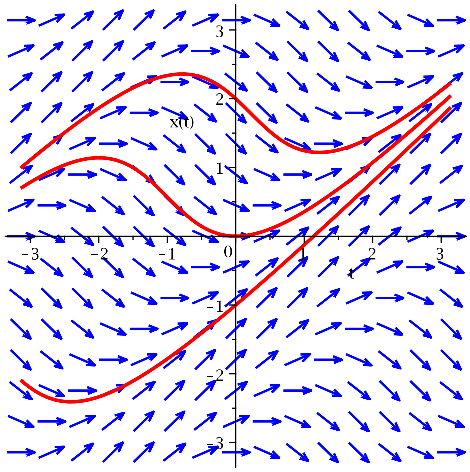
Voorbeeldsessie

```
> restart: with(DEtools):
```

[50]Blijkbaar worstelen de makers van Maple nogal met vergelijkingen van het type (21.4). Tot nu toe gaf zo'n beetje elke volgende versie van Maple een ander resultaat.

```
> DEplot( diff(x(t),t)=sin(t-x(t)), x, t=-Pi..Pi,
         {[0,0],[0,2],[0,-1]}, x=-Pi..Pi, stepsize=0.1,
         arrows=smalltwo, color=blue, linecolor=red,
         dirfield=[15,15] );
```

(zie figuur 41.)



FIGUUR 41. Richtingsveld en enkele benaderde oplossingen van $x' = sin(t - x)$

Toelichting

DEplot

We gebruiken de procedure `DEplot` uit de bibliotheek `DEtools`. Deze procedure maakt een plaatje van een of meer benaderde oplossingen van een (stelsel) differentiaalvergelijking(en). Hij moet worden aangeroepen met ten minste vier of vijf argumenten:

Eerste orde scalar DV: Gebruik van `DEplot` voor het tekenen van een richtingsveld en/of enkele oplossingen:

(1) Het eerste argument is de differentiaalvergelijking.
(2) Het tweede argument is de afhankelijke variabele.
(3) Het derde argument geeft het bereik van de onafhankelijke variabele.
(4) Het vierde argument is een verzameling met begincondities, in dit geval elk van de vorm `[t0, x(t0)]`. Als het vierde argument wordt weggelaten, dan wordt alleen een richtingsveld getekend.
(5) Het vijfde argument geeft het bereik van de afhankelijke variabele.
(6) Ten slotte volgt een reeks opties in de bekende vorm: trefwoord=waarde. Als alléén een aantal oplossingskrommen

(dus zonder richtingsveld) getekend moet worden, dan moet de optie `arrows=NONE` worden opgegeven.

Maple berekent de (benaderde) oplossing $x(t)$ voor de waarden $t = 0 \pm h,\ t = 0 \pm 2h, \ldots$, waarin h de stapgrootte (`stepsize`) is. In grote trekken[51] komt het er op neer dat vanuit de gegeven beginwaarde. dus in dit geval het punt $(0,0)$, $(0,2)$ of $(0,-1)$ een stukje h in de richting van de pijl loopt. In het punt waar je dan uitkomt, staat weer een pijl, die dan weer een stukje h wordt gevolgd, enzovoort. ⋄

Ook voor *hogere orde* scalar differentiaalvergelijkingen kunnen we met `DEplot` grafieken van benaderde oplossingen tekenen. Een richtingsveld is dan (uiteraard!) niet meer mogelijk.

Voorbeeldopgave

Gegeven de differentiaalvergelijking $y'' - y' - 6y = e^{2t} \cos 3t$. Teken grafieken van de oplossingen met beginwaarden $y(0) = 1, y'(0) = -1$, respectievelijk $y(0) = 0, y'(0) = 2$.

Voorbeeldsessie

```
> restart; with(DEtools):
> DV := diff(y(t),t,t) - diff(y(t),t) - 6*y(t) = exp(2*t)*cos(3*t);
```

$$DV := (\frac{d^2}{dt^2}\,\mathrm{y}(t)) - (\frac{d}{dt}\,\mathrm{y}(t)) - 6\,\mathrm{y}(t) = e^{(2\,t)}\cos(3\,t)$$

```
> DEplot(DV, y(t), t=-1..1,
         {[y(0)=1, D(y)(0)=-1],[y(0)=0, D(y)(0)=2]},
         y=-1..3, linecolor=black );
```

(zie figuur 42.)

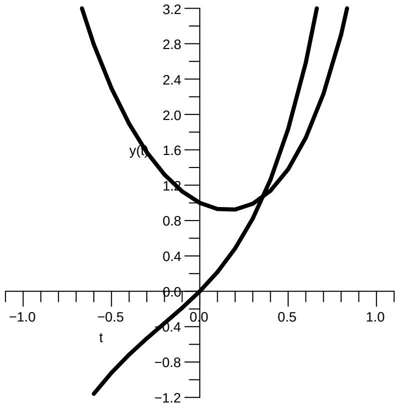
Toelichting

De beginvoorwaarden moeten in dit geval op een iets andere manier worden opgegeven dan bij een eerste orde vergelijking, namelijk in de vorm

```
[y(t0)=c1, D(y)(t0)=c2]
```

(hogere afgeleiden met `(D@@2)(y)(x0)=c3` enzovoort.)
Omdat er toch geen richtingsveld kan worden getekend zal een eventuele optie `arrows=...` door Maple worden genegeerd. ⋄

[51] In feite is de methode veel verfijnder: er wordt bijvoorbeeld óók gekeken naar de richtingen van andere pijlen in de buurt.



FIGUUR 42. Enkele benaderde oplossingen van de vergelijking $y'' - y' - 6y = e^{2t} \cos 3t$

21.6 Richtingsveld en oplossingskrommen bij stelsels van twee autonome vergelijkingen

Ook voor het stelsel van *twee autonome differentiaalvergelijkingen* kunnen we het *richtingsveld* tekenen. Als het stelsel gegeven wordt door

$$\begin{cases} \frac{dx}{dt} &=& f(x,\, y) \\ \frac{dy}{dt} &=& g(x,\, y) \end{cases} \tag{21.5}$$

dan tekenen we op een grid in een rechthoek in het $(x,\, y)$-vlak vectoren met richtingscoëfficiënt

$$\frac{g(x,y)}{f(x,y)}\,.$$

Een curve $t \mapsto (x(t),\, y(t))$ is een oplossing van het stelsel differentiaalvergelijkingen als op ieder punt van de kromme de raaklijn richtingscoëfficiënt

$$\frac{g(x(t),y(t))}{f(x(t),y(t))}$$

heeft. Anders gezegd, als de curve overal raakt aan het richtingsveld dan is het een oplossing van het stelsel differentiaalvergelijkingen. De informatie over de snelheid waarmee de oplossing over een curve loopt gaat verloren. De tijd t wordt ‘weggedeeld’ door in feite te kijken naar de differentiaalvergelijking

$$\frac{dy}{dx} = \frac{g(x,y)}{f(x,y)}\,.$$

In de punten waar $f(x,y) = 0$ én $g(x,y) = 0$ kan het richtingsveld niet getekend worden. Dit zijn de zogeheten *singuliere punten* van het vectorveld en dit zijn de *evenwichtspunten* van het stelsel differentiaalvergelijkingen.
Een tekening in het (x,y)-vlak van een aantal representatieve banen $t \mapsto (x(t), y(t))$ wordt wel een *faseportret* of faseplaatje van het stelsel (21.5) genoemd.

Voorbeeldopgave

Teken het richtingsveld van de *Volterra-Lotka-vergelijkingen*

$$\begin{aligned} \frac{dx}{dt} &= x(1-y) \\ \frac{dy}{dt} &= \frac{3\,y\,(x-1)}{10}, \end{aligned} \tag{21.6}$$

op het gebied $-1/2 \le x \le 5/2, \quad -1/2 \le y \le 5/2$.
Schets de oplossingscurven die gaan door de punten $(x,\, y) = (0.5,\, 1)$ en $(x,\, y) = (0.3,\, 1)$

Voorbeeldsessie

```
> restart:  with(DEtools):
> DEplot( {diff(x(t),t)=x(t)*(1-y(t)),
          diff(y(t),t)=.3*y(t)*(x(t)-1)}, [x,y],
     t=0..15, {[0,0.5,1],[0,0.3,1]},
     x=-0.5..2.5, y=-0.5..2.5,
     stepsize=0.05, arrows=SLIM, color=blue,
     linecolor=red, dirfield=[16,16],
     title="Lotka-Volterra model");
```

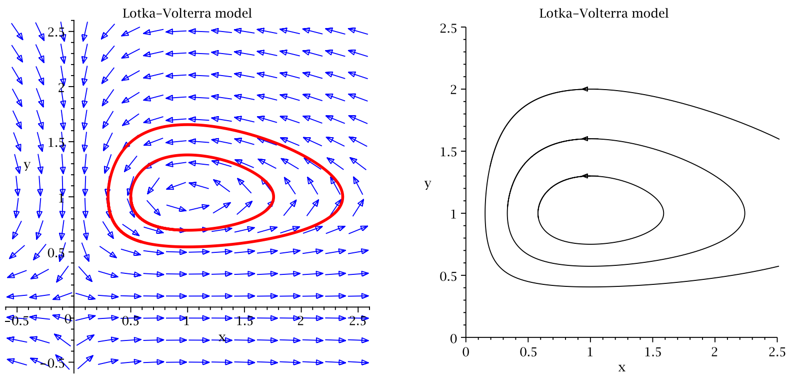
(zie figuur 43, links.)

```
> DEplot( {diff(x(t),t)=x(t)*(1-y(t)),
          diff(y(t),t)=.3*y(t)*(x(t)-1)}, [x,y],
     t=0..15, {[0,1,1.3],[0,1,1.6],[0,1,2]},
     x=0..2.5, y=0..2.5,
     stepsize=0.05, arrows=medium, color=black,
     linecolor=black, dirfield=[[1,1.3],[1,1.6],[1,2]],
     title="Lotka-Volterra model", obsrange=false);
```

(zie figuur 43, rechts.)

Toelichting

Na het vorige voorbeeld moet het eerste deel van de Maple-invoer wel duidelijk zijn; met de `dirfield`-optie geven wea an dat in horizontale en in verticale richting 16 pijltjes getekend moeten worden.
In het tweede deel hebben we drie banen getekend, zónder richtingsveld. Als `dirfield` hebben we nu een lijst met drie punten opgegeven.



FIGUUR 43. Faseportret van de Lotka-Volterra-vergelijkingen

Dat betekent dat er een 'richtingsveld' wordt getekend dat slechts uit drie pijltjes bestaat. En omdat deze drie pijltjes precies op de drie banen liggen krijgen we banen met een richting.
Verder loopt één van de drie banen 'het beeld uit'. Normaliter houdt
obsrange Maple dan op met rekenen, maar met `obsrange=false` kunnen we dat voorkomen. ◇

We geven voor de volledigheid nog een overzicht:

Stelsel eerste orde differentiaalvergelijkingen. Argumenten voor `DEplot` voor het tekenen van een faseportret:

(1) Het eerste argument is het stelsel differentiaalvergelijkingen als verzameling;
(2) Het tweede argument is de lijst met afhankelijke variabelen.
(3) Het derde argument geeft het bereik van de onafhankelijke variabele.
(4) Het vierde argument is een verzameling met begincondities, in dit geval elk van de vorm `[t0, x(t0), y(t0)]`. Als het vierde argument wordt weggelaten, dan wordt alleen een richtingsveld getekend.
(5) Het vijfde argument geeft het bereik van de eerste afhankelijke variabele. Deze wordt langs de horizontale as getekend.
(6) Het zesde argument geeft het bereik van de tweede afhankelijke variabele. Deze wordt langs de verticale as getekend.
(7) Ten slotte volgt een reeks opties in de bekende vorm: trefwoord=waarde. Als alléén een aantal banen (dus zonder richtingsveld) getekend moet worden, dan moeten we `arrows=NONE` als optie meegegeven.

In Module 22 gaan we verder in op de analyse van het faseportret.

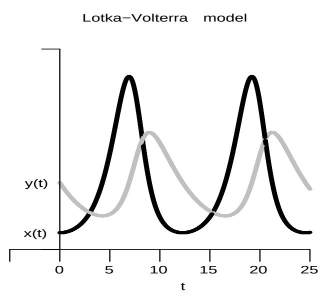
scene

Grafieken van $x(t)$ en $y(t)$. De procedure `DEplot` heeft nog een interessante optie, namelijk de `scene`-optie. Hiermee kunnen de grafieken van de oplossingen, dus als functie van t, worden getekend. We demonstreren dat aan de hand van het Lotka-Volterra-voorbeeld.

Voorbeeldsessie

```
> restart:  with(DEtools):  with(plots):
> stelsel := {diff(x(t),t)=x(t)*(1-y(t)),
              diff(y(t),t)=.3*y(t)*(x(t)-1)}:
> px := DEplot( stelsel, [x,y], t=0..25, {[0,0.25,1]},
     stepsize=0.05, linecolor=black, scene=[t,x] ):
> tx := textplot( [0,0.25,"x(t)  "], align=LEFT ):
> py := DEplot( stelsel, [x,y], t=0..25, {[0,0.25,1]},
     stepsize=0.05, linecolor=grey, scene=[t,y] ):
> ty := textplot( [0,1,"y(t)  "], align=LEFT ):
> display({px,tx,py,ty}, view=[-5..25,0..3],
     tickmarks=[[0,5,10,15,20,25],[]], labels=["t",""],
     title="Lotka-Volterra model" );
```

(zie figuur 44.)



FIGUUR 44. Oplossingen van de Lotka-Volterra-vergelijkingen

Toelichting

We hebben de oplossingen met beginwaarden $x(0) = 0.25$, $y(0) = 1$ getekend. Het *bereik* van de afhankelijke variabelen hoeven we nu niet op te geven, de (beide) *namen* natuurlijk wél. ◇

DEplot3d

Tenslotte is er nog de procedure `DEplot3d` waarmee de oplossing van een stelsel van twee (of meer) differentiaalvergelijkingen als ruimtekromme wordt getekend. Men kan dan respectievelijk t, $x(t)$ en $y(t)$ langs de assen uitzetten. U heeft waarschijnlijk `?DEplot3d` niet eens nodig om het meteen goed te doen.

21.7 Numerieke benadering

Vaak lukt het niet om analytische oplossingen van een (stelsel) differentiaalvergelijking(en) te berekenen. In dat geval moeten we onze toevlucht nemen tot een numerieke methode. Bij het tekenen van allerlei plaatjes met `DEplot` worden natuurlijk ook numerieke benaderingen berekend, maar de resultaten daarvan worden direct 'vertaald' in 'punten van de tekening'.
In deze paragraaf behandelen we verschillende manieren om numerieke benaderingen van de oplossing in de vorm van getallen te verkrijgen. Uiteraard mogen er dan geen onbepaalde constanten in de vergelijkingen meer voorkomen. We laten zien hoe dat gaat aan de hand van het stelsel (21.3) dat we in §21.3 hebben opgelost:

$$\begin{cases} V_1\, x'(t) &= -\Phi\, x(t) \\ V_2\, y'(t) &= \Phi\, x(t) - \Phi\, y(t)\,. \end{cases}$$

We vullen eerst de gegevens in in het oude stelsel, en lossen dit stelsel op met het commando `dsolve` met als extra optie: `numeric`.

Voorbeeldsessie

```
> restart;
> DVset := {V1*diff(x(t),t) = -Phi*x(t),
            V2*diff(y(t),t) = Phi*x(t) - Phi*y(t)}:
> BWset := {x(0)=c, y(0)=c}:
> gegevens := {Phi=1.2, V1=8, V2=3, c=0.1}:
> stelsel := op( subs(gegevens,DVset) );
```

$$\mathit{stelsel} := 8\,(\tfrac{d}{dt}\,\mathrm{x}(t)) = -1.2000\,\mathrm{x}(t),$$
$$3\,(\tfrac{d}{dt}\,\mathrm{y}(t)) = 1.2000\,\mathrm{x}(t) - 1.2000\,\mathrm{y}(t)$$

```
> beginvoorwaarden := op( subs(gegevens,BWset) );
```

$$\mathit{beginvoorwaarden} := \mathrm{x}(0) = 0.1000,\ \mathrm{y}(0) = 0.1000$$

```
> functies := x(t), y(t):
```

We vragen Maple om een numerieke benadering van het stelsel te berekenen:

```
> numopl := dsolve( {stelsel,beginvoorwaarden}, {functies}, numeric );
```

$$\mathit{numopl} := \mathbf{proc}(\mathit{x\_rkf45})\ \ldots\ \mathbf{end\ proc}$$

We zien dat de uitvoer een *procedure* is, die blijkbaar met één argument moet worden aangeroepen.
Om te kijken wat `numopl` eigenlijk voor een soort ding is, laten hem eens werken op een (willekeurig) argument:

```
> numopl(4);
```

$$[t = 4.0000,\ \mathrm{x}(t) = 0.0549,\ \mathrm{y}(t) = 0.0757]$$

dit zijn blijkbaar de benaderde waarden van x(4) en y(4). De functiewaarde x(4) kunnen we dus krijgen met de volgende `subs`-opdracht:

```
> subs(numopl(4),x(t));
```

0.0549

Als we snel plaatjes willen hebben is er de speciale procedure `odeplot` die met `numopl` uit de voeten kan:

```
> plots:-odeplot( numopl, [[t,x(t)],[t,y(t)]], 0..40 ):
```

(zie figuur 40 op blz. 333.)

Om verder te werken kunnen we bijvoorbeeld een lijst van benaderde waarden maken

```
> Xlijst := [seq( subs(numopl(i),[t,x(t)]), i=0..40,0.1 )]:
```

We bekijken het begin van deze lijst:

```
> seq( Xlijst[i], i=1..5 );
```

[0.0000, 0.1000], [0.1000, 0.0985], [0.2000, 0.0970], [0.3000, 0.0956], [0.4000, 0.0942]

Toelichting

We hebben eerst (via het Tools → Options-menu) de numerieke uitvoer naar het scherm op 4 decimalen gezet (zie §3.5). De uitgevoerde berekeningen zijn veel nauwkeuriger, maar de getoonde resultaten worden op vier cijfers achter de komma afgerond.

`numeric` Het commando `dsolve`, aangeroepen met de optie `numeric`, geeft een procedure als output. Deze procedure kan worden aangeroepen met een waarde van de onafhankelijke variabele als argument. Het resultaat van een dergelijke aanroep is een *lijst* van vergelijkingen met de opgegeven waarde van de onafhankelijke variabele en de benaderde waarden van de afhankelijke variabelen. Dit heet:

`procedurelist` `odeplot` `output=procedurelist`. Zo'n `procedurelist` kan dienen als argument in het commando `odeplot` uit de bibliotheek `plots`. Als er alleen een plaatje nodig is, dan biedt het gebruik van `DEplot` (zoals gezien) meer mogelijkheden. ◇

`output=Array` Een iets andere manier om de resultaten in een handzame vorm beschikbaar te krijgen, is de `output`-optie `Array` te gebruiken. We vervolgen de vorige sessie met

Voorbeeldsessie

```
> numopl := dsolve( {stelsel,beginvoorwaarden}, {functies},
             numeric, output=Array([0,1,1.5,3]) );
```

$$numopl := \begin{bmatrix} [t,\ \mathrm{x}(t),\ \mathrm{y}(t)] \\ \begin{bmatrix} 0.0000 & 0.1000 & 0.1000 \\ 1.0000 & 0.0861 & 0.0975 \\ 1.5000 & 0.0799 & 0.0948 \\ 3.0000 & 0.0638 & 0.0839 \end{bmatrix} \end{bmatrix}$$

```
> vars := numopl[1,1];
```

$$vars := [t,\ \mathrm{x}(t),\ \mathrm{y}(t)]$$

```
> res := numopl[2,1];
```

$$res := \begin{bmatrix} 0.0000 & 0.1000 & 0.1000 \\ 1.0000 & 0.0861 & 0.0975 \\ 1.5000 & 0.0799 & 0.0948 \\ 3.0000 & 0.0638 & 0.0839 \end{bmatrix}$$

Toelichting

In de `output`-optie geven we de waarden van de onafhankelijke variabele t aan waarin we de afhankelijke variabelen $x(t)$ en $y(t)$ berekend willen hebben. Het resultaat ziet er een beetje raar uit: `numopl` is nu een 2×1-matrix (dus eigenlijk een kolomvector). Het eerste element van `numopl` is een Array van de variabelen; het tweede element is een Matrix, met de waarden van t, $x(t)$ en $y(t)$ als kolommen. ◇

Vaak is het handig om de (numerieke benadering van) de oplossing als functie te hebben, bijvoorbeeld on de volgende

Voorbeeldopgave

Zie de voorbeeldsessie op blz. 344. Bereken het tijdstip waarop de concentratie $x(t)$, respectievelijk $y(t)$ tot de helft van de beginconcentratie is gereduceerd.

Voorbeeldsessie

Vervolg van de vorige sessie

```
> numopl := dsolve( {stelsel,beginvoorwaarden}, {functies},
             numeric, output=listprocedure );
```

$$numopl := [t = (\mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}),$$
$$\mathrm{x}(t) = (\mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}),$$
$$\mathrm{y}(t) = (\mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc})]$$

```
> X := subs(numopl,x(t));
```

$$X := \mathbf{proc}(t)\ \ldots\ \mathbf{end\ proc}$$

```
> X(0), X(1), X(2);
```

$$0.1000,\ 0.0861,\ 0.0741$$

```
> t_xhalf := fsolve( X(t)=X(0)/2, t );
```

$$t\_xhalf := 4.6210$$

```
> Y := subs(numopl,y(t)):
> t_yhalf := fsolve( Y(t)=Y(0)/2, t );
```

$$t\_yhalf := 7.3432$$

Toelichting

`listprocedure` Met de optie `output=listprocedure`, levert `dsolve, numeric` een *lijst van procedures*, die direct in een `subs`-commando gebruikt kan worden. Dit gaat op dezelfde manier als de analytische oplossing van een (stelsel) differentiaalvergelijking(en). In de voorbeeldsessie is het resultaat van zo'n substitutie dat bijvoorbeeld `X` een procedure wordt, waarmee benaderingen van $x(t)$ voor willekeurige waarden van t berekend kunnen worden. Dergelijke uitdrukkingen kunnen gewoon in een vergelijking worden opgenomen, die met `fsolve` (zie §5.6) kan worden opgelost. ◇

Opgave 21.1

Gebruik `dsolve` om de algemene oplossingen te bepalen van de volgende differentiaalvergelijkingen; Controleer uw antwoord door het in te vullen in de vergelijking.

(a) $y' - y = xe^x$
(b) $(x^2+1)y' - xy = 2x + \sqrt{x^2+1}$
(c) $y' + xe^{-x}y = xe^{-x}$
(d) $y'' - y' - 6y = e^{2x}\cos(3x)$
(e) $y'' + (1+a)y' + ay = e^{-x}$ (welke a moet u apart bekijken?)

Opgave 21.2

Bepaal de oplossing van het volgende randwaardeprobleem:

$$y'' + 2y' + y = x, \quad y(0) = 0 \text{ en } y(1) = 1.$$

Opgave 21.3

Schets richtingsveld en enkele oplossingscurven van de differentiaalvergelijking $\frac{dx}{dt} = x^2 + t$.
Probeer ook eens of Maple deze vergelijking analytisch kan oplossen:

```
dsolve( diff(x(t),t)=x(t)^2 + t, x(t) )
```

Zoek op wat het antwoord betekent. Vergelijk de baan die gaat door het punt (0, 0) met de analytische oplossing.

Opgave 21.4

Schets snelheidsveld en enkele oplossingscurven van de differentiaalvergelijking

$$\frac{dx}{dt} = \frac{2\,t - x}{t - 2\,x}.$$

Verklaar waarom het resultaat er zo beroerd uit komt te zien.
Vergelijk dit nu eens met het stelsel autonome differentiaalvergelijkingen:

$$\frac{dx}{ds} = 2\,t - x$$
$$\frac{dt}{ds} = t - 2\,x.$$

Verklaar overeenkomst en verschil tussen de twee plaatjes.

Opgave 21.5

Beschouw de Volterra-Lotka-vergelijkingen (21.6) op blz. 341.

(a) Gebruik `DEplot3d` om een ruimtekromme $(t, x(t), y(t))$ te tekenen. Kies zelf geschikte beginvoorwaarden.
Draai (met de muis) het plaatje zó, dat u achtereenvolgens een baan zoals in figuur 43 en de beide grafieken zoals in figuur 44 te zien krijgt.

(b) De oplossingen $x(t)$ en $y(t)$ zijn periodiek. Bereken numerieke oplossingen, en bepaal daarmee de periode van de oplossing $x(t)$ en van $y(t)$. (Waarom) zijn deze periodes gelijk?
Is de periode afhankelijk van de beginwaarden?