Wiskundige modellering in de ingenieurswetenschappen: Bordoefeningenles 2

Oefening 1

```
> restart: with(plots):with(LinearAlgebra):with(plottools):
  oorsprong := <0,0,0>:
```

Constructie matrix A en vector y:

```
> K1 := <1,2,3>;
  K2 := <1,4,9>;
  A  := <K1|K2>;
  y  := <10.1,7.4,-5.2>;
```

$$K1 := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$K2 := \begin{bmatrix} 1 \\ 4 \\ 9 \end{bmatrix}$$

$$A := \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \end{bmatrix}$$

$$y := \begin{bmatrix} 10.1 \\ 7.4 \\ -5.2 \end{bmatrix} \quad (1.1)$$

Heeft dit stelsel een oplossing?

```
> solve(A.(<v0, -g/2>)=y, {v0,g});
> Determinant(<K1|K2|y>);
```

$$5.8 \quad (1.2)$$

We bepalen de kleinste kwadraten benadering en fit:

De onbekenden x bepalen kan op 2 manieren:

- met een stelsel (meest efficiënt)

```
> solve((A^%T.A).(<v0,-g/2>)=A^%T.y,{v0,g});
```

$$\{g = 11.42631579, v0 = 15.35526316\} \quad (1.3)$$

- met behulp van de matrix inverse

```
> x     := MatrixInverse(A^%T.A).A^%T.y;
```

```
v0    := x[1];
g     := -2*x[2];
```

$$x := \begin{bmatrix} 15.3552631578947 \\ -5.71315789473684 \end{bmatrix}$$

$$v0 := 15.3552631578947$$

$$g := 11.4263157894737 \quad (1.4)$$

De kleinste kwadraten benadering vinden we als

```
> y_kk  := A.x;
  y_com := y-y_kk:
```

$$y\_kk := \begin{bmatrix} 9.64210526315790 \\ 7.85789473684211 \\ -5.35263157894736 \end{bmatrix} \quad (1.5)$$

Visualisatie kolomruimte K(A) + nulruimte N(A^T)

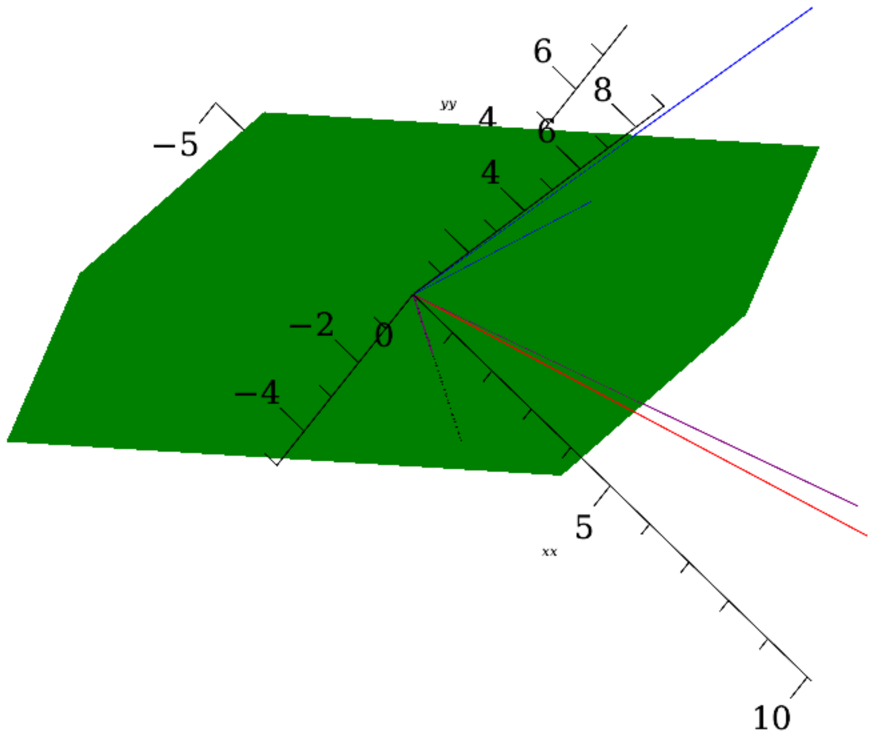
```
> KA1        := plot3d(h*K1+v*K2, h=-8..8,v=-9..9,
                color=green, numpoints=20,style=surface, axes=
  normal):
  KA2        := implicitplot3d((<xx,yy,zz>).y_com=0, xx=-5..5,
  yy=-5..5,zz=-5..5, color=green, numpoints=20,style=surface,
  axes=normal):

  NAT        := plot3d(t*y_com,t=-5..5, linestyle="dot"):

  K1_lijn    := line(oorsprong,K1, color=blue):
  K2_lijn    := line(oorsprong,K2, color=blue):

  y_lijn     := line(oorsprong,y, color=red):
  y_kk_lijn  := line(oorsprong,y_kk, color=purple):
  y_com_lijn := line(oorsprong,2*y_com, color=purple):


  display(KA2, NAT, K1_lijn, K2_lijn, y_lijn, y_kk_lijn,
  y_com_lijn, scaling=constrained);
```



Ten slotte visualiseren we de gevonden fit oplossingen met de datapunten:
We beginnen met de gemeten en geprojecteerde data punten te visualiseren:

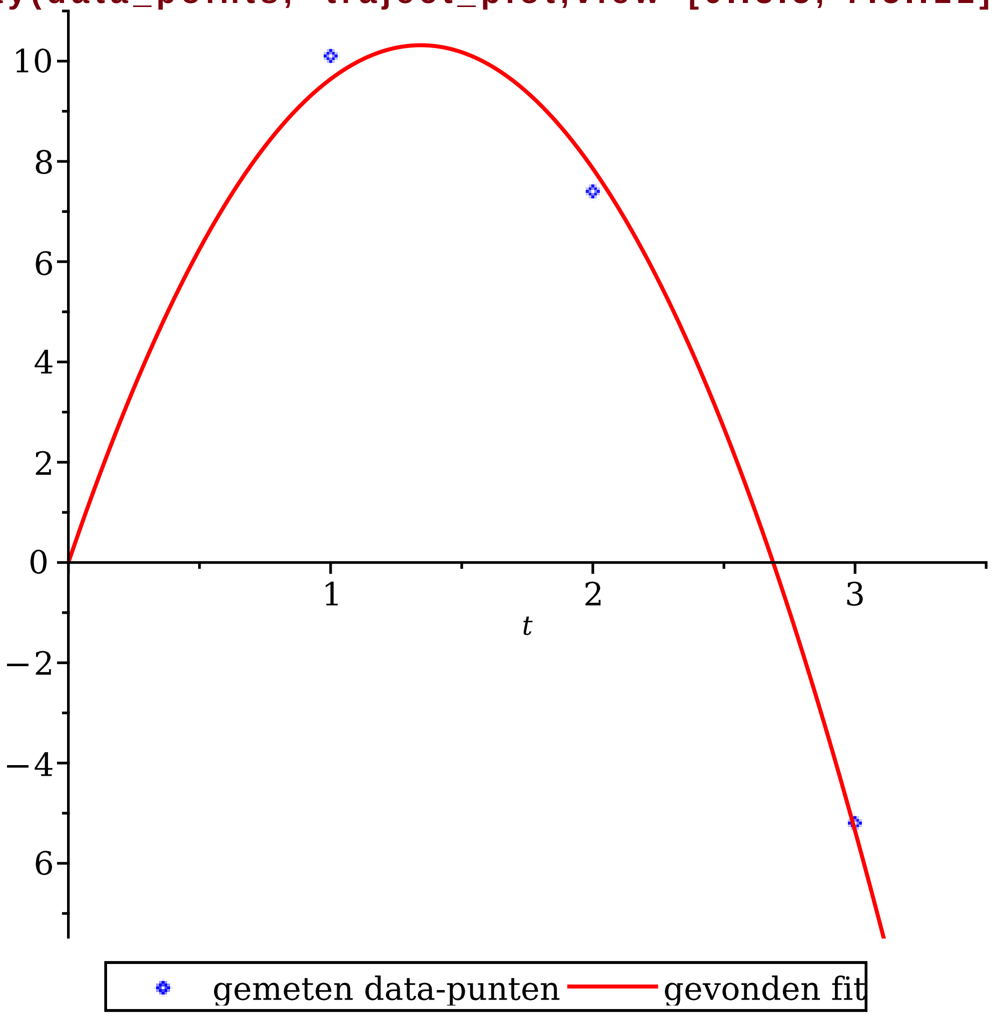
```
> data_points := pointplot(K1, y, color=blue, legend=
  "gemeten data-punten"):
```

Daarna plotten we het traject (=de gevonden fit):

```
> traject := t->v0*t-g/2*t**2:
```

```
> traject_plot := plot(traject(t),t=0..4.5,    color=red, legend=
  "gevonden fit"):
```

```
> display(data_points,  traject_plot,view=[0..3.5,-7.5..11]);
```



▶ Oefening 5

▶ Oefening 2

▶ Oefening 3

▶ Oefening 4