Interpolatie

Academiejaar 2020-2021

Systeem

```
> restart;
  print(`Hardware float machine epsilon` = evalf(2^(-52)));
  Digits := 16:
```

$$\textit{Hardware float machine epsilon} = 2.220446049\ 10^{-16} \quad (1.1)$$

```
> with(plots):
  with(Interpolation):
  with( Student[NumericalAnalysis] ):
```

Opgave:

Gegeven is $f(x) = \dfrac{1}{25\,x^2 + 1}$ op het interval $[-1, 1]$ voor x. deze functie wordt ook wel de Runge-functie genoemd.
Bereken de interpolerende veelterm $P(x)$ door equidistante interpolatiepunten. Bereken tevens de interpolerende veelterm $Q(x)$ door de Chebyshev nulpunten.
Bereken tenslotte de lineare en kubische spline $S(x)$ door equidistante punten. Welke keuze maak je voor de randvoorwaarden?

Gegeven zijn de datapunten

x = [1 , 2 , 3 , 4 , 5 , 6]
y = [16, 18, 21, 17, 15, 12]

Vergelijk de interpolerende veeltern en de kubische spline. Welke keuze kan je maken voor de randvoorwaarden?

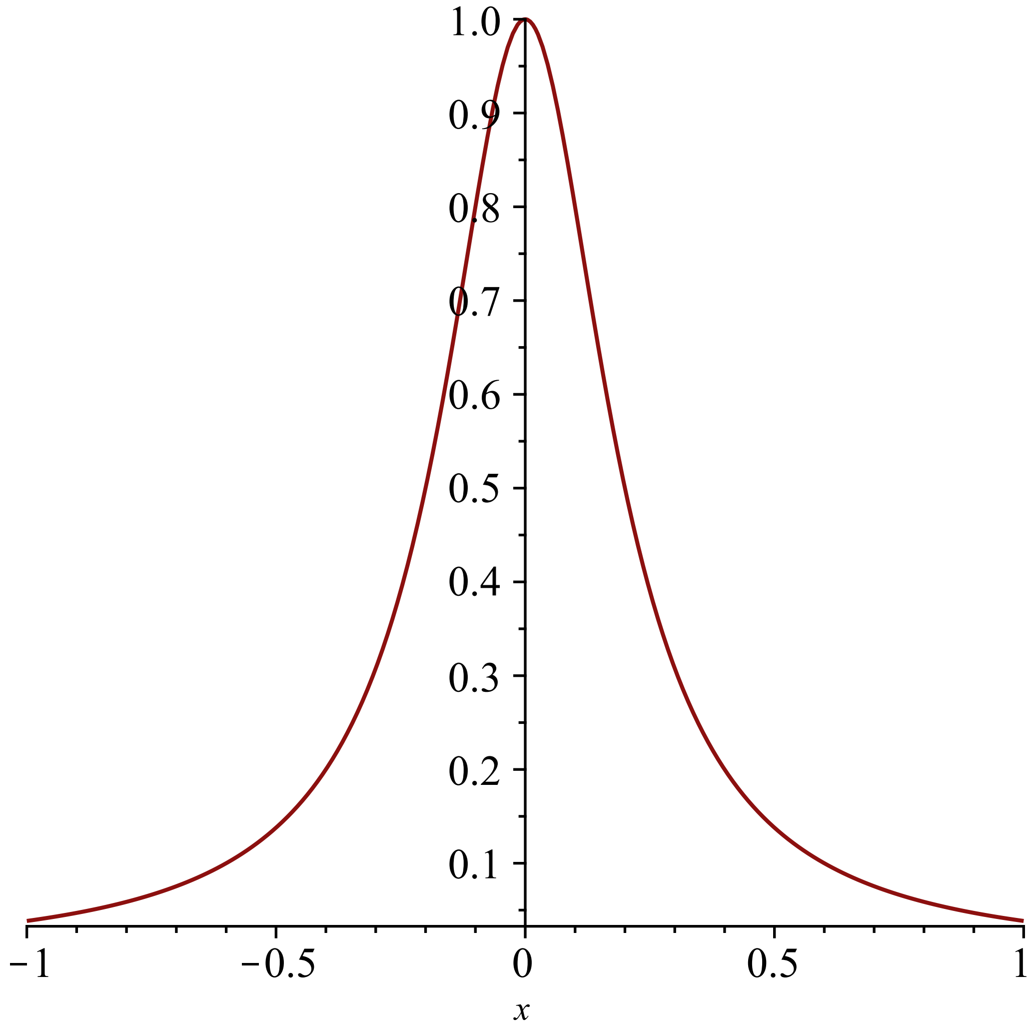
```
> f := x -> 1/(1+25*x^2):
  print( 'f'(x) = f(x) );
```

$$f(x) = \frac{1}{25\,x^2 + 1} \quad (1)$$

```
> a := -1:
  b :=  1:
  print( ['a','b'] = [a,b] );
```

$$[a, b] = [-1, 1] \quad (2)$$

```
> plot( f(x), x=a..b );
```



3.1. Veelterminterpolatie

Veelterminterpolatie met equidistante punten

Een veelterm van graad n heeft $n+1$ coëfficiënten en dus hebben we $n+1$ interpolatievoorwaarden nodig om al deze coëfficiënten te kunnen bepalen.
Bijgevolg hebben we dus de functiewaarden in $n+1$ verschillende punten in het interval $[a, b]$ nodig om de interpolerende veelterm uit te rekenen.

Equidistante punten

Neem $n+1$ equidistante punten in $[a, b]$ zodat $x_i = a + ih$ met $h = \frac{b-a}{n}$:

```
> h := (b-a)/n;
```

(2.1.1.1)

$$h := \frac{2}{n} \tag{2.1.1.1}$$

```
> x := unapply( a+i*h, i );
```

$$x := i \mapsto \frac{2\,i}{n} - 1 \tag{2.1.1.2}$$

Lagrange voorstelling

```
> n := 6;
```

$$n := 6 \tag{2.1.2.1}$$

```
> [seq( x(i), i=0..n )];
  evalf(%);
```

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

$$[-1., -0.6666666667, -0.3333333333, 0., 0.3333333333, 0.6666666667, 1.] \tag{2.1.2.2}$$

```
> [seq( f(x(i)), i=0..n )];
  evalf(%);
```

$$\left[\frac{1}{26}, \frac{9}{109}, \frac{9}{34}, 1, \frac{9}{34}, \frac{9}{109}, \frac{1}{26}\right]$$

$$[0.03846153846, 0.08256880734, 0.2647058824, 1., 0.2647058824, 0.08256880734, 0.03846153846] \tag{2.1.2.3}$$

```
> xy := [ seq( [x(i),f(x(i))], i=0..n ) ];
```

$$xy := \left[\left[-1, \frac{1}{26}\right], \left[-\frac{2}{3}, \frac{9}{109}\right], \left[-\frac{1}{3}, \frac{9}{34}\right], [0, 1], \left[\frac{1}{3}, \frac{9}{34}\right], \left[\frac{2}{3}, \frac{9}{109}\right], \left[1, \frac{1}{26}\right]\right] \tag{2.1.2.4}$$

```
> L := PolynomialInterpolation( xy, independentvar=x, method=
  lagrange ):
> Interpolant( L );
```

$$\frac{81\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{2080} \tag{2.1.2.5}$$
$$-\frac{2187\,(x+1)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{4360}$$
$$+\frac{2187\,(x+1)\left(x+\frac{2}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{544}$$
$$-\frac{81\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{4}$$
$$+\frac{2187\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{2}{3}\right)(x-1)}{544}$$

$$-\frac{2187\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)(x-1)}{4360}$$

$$+\frac{81\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)}{2080}$$

```
> BL := identify( BasisFunctions( L ) ):
  for j from 1 to nops(BL) do
    print( beta[j-1]('x') = BL[j] );
  end;
```

$$\beta_0(x)=\frac{81\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{80}$$

$$\beta_1(x)=-\frac{243\,(x+1)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{40}$$

$$\beta_2(x)=\frac{243\,(x+1)\left(x+\frac{2}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{16}$$

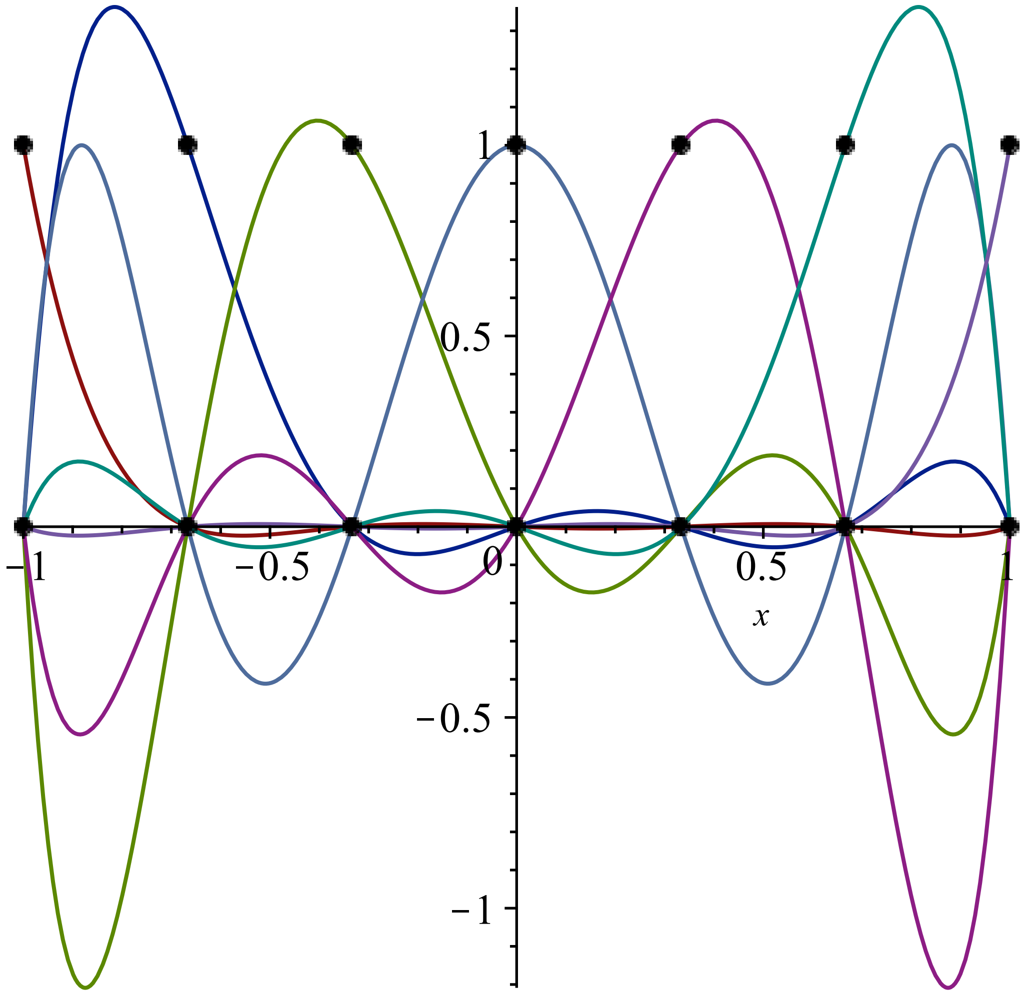
$$\beta_3(x)=-\frac{81\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)(x-1)}{4}$$

$$\beta_4(x)=\frac{243\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{2}{3}\right)(x-1)}{16}$$

$$\beta_5(x)=-\frac{243\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)(x-1)}{40}$$

$$\beta_6(x)=\frac{81\,(x+1)\left(x+\frac{2}{3}\right)\left(x+\frac{1}{3}\right)x\left(x-\frac{1}{3}\right)\left(x-\frac{2}{3}\right)}{80} \qquad \textbf{(2.1.2.6)}$$

```
> p1 := pointplot( [seq([x(i),1],i=0..n)], color=blue, symbol=
  solidcircle, symbolsize = 15 , color = black):
  p2 := pointplot( [seq([x(i),0],i=0..n)], color=blue, symbol=
  solidcircle, symbolsize = 15 , color = black):
  p3 := Draw( L, objects=[BasisFunctions] ):
  display(p3,p1,p2);
```

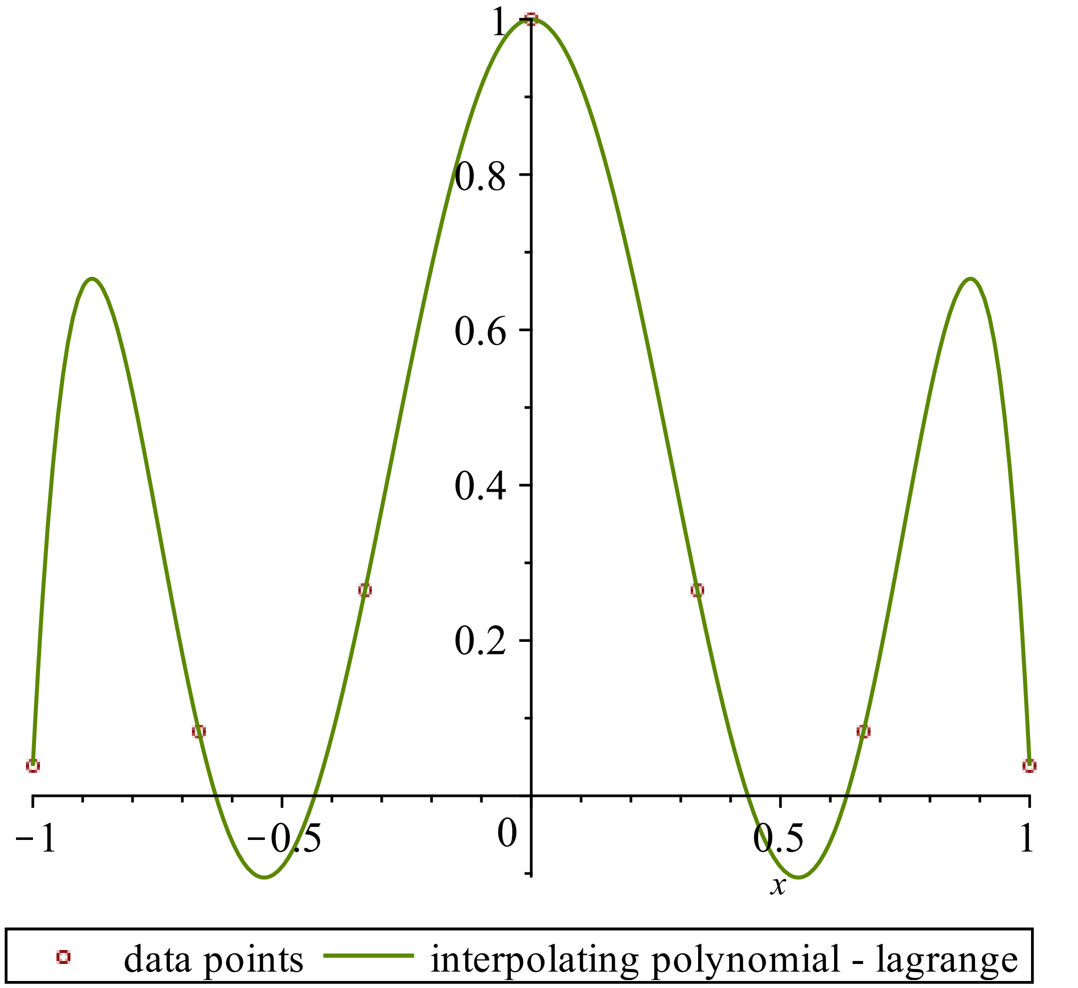


Polynomial interpolation. The basis functions.

```
> sort( expand( Interpolant( L ) ), x, ascending );
```

$$1-\frac{211600}{24089}x^2+\frac{2019375}{96356}x^4-\frac{1265625}{96356}x^6 \tag{2.1.2.7}$$

```
> Draw( L );
```

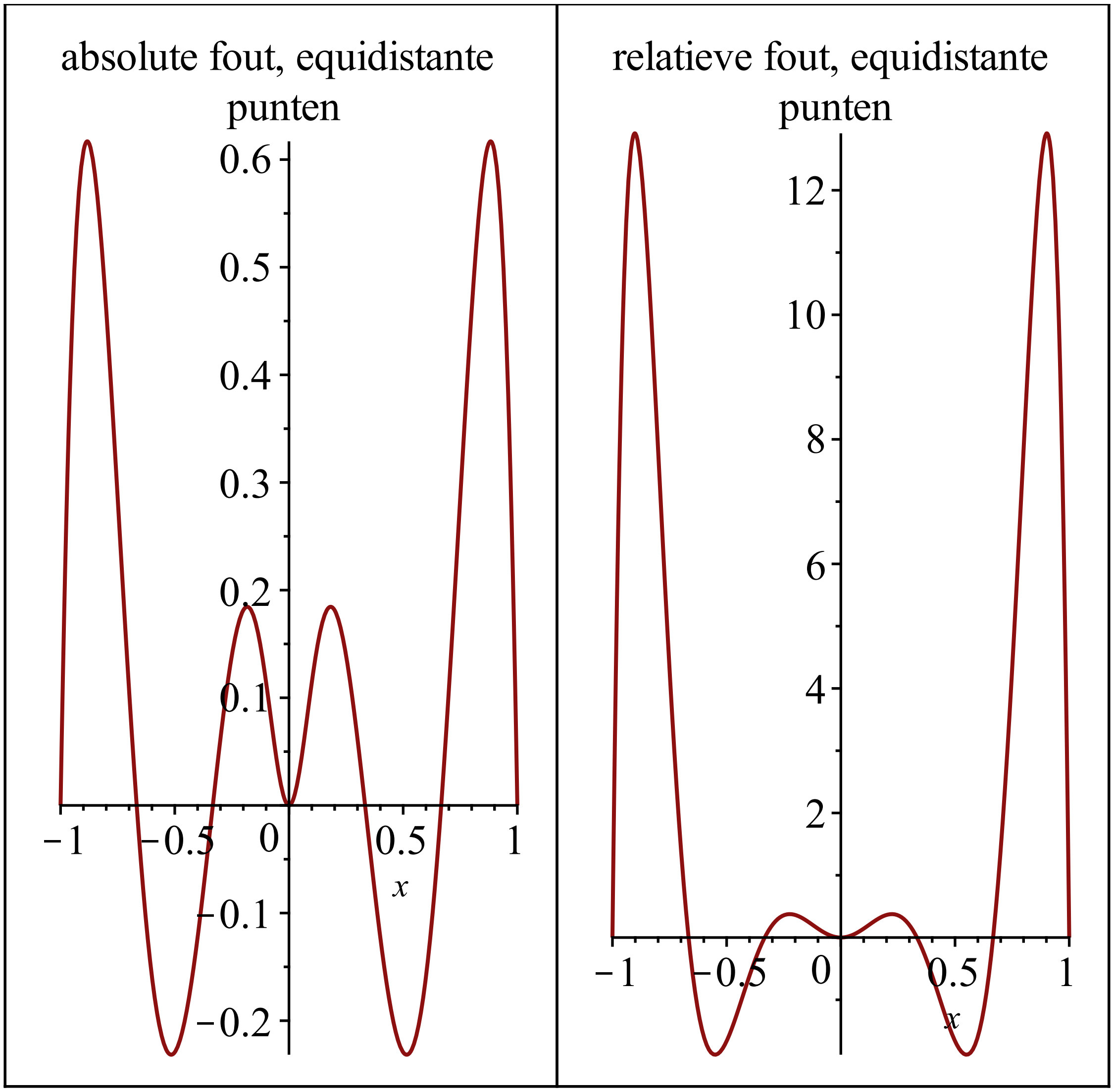


Polynomial interpolation.

Foutenkromme

We beschouwen zowel de absolute als de relatieve fout. Maar vergeet niet, uit de les over computer aritmetiek, dat de relatieve fout in praktijk veelzeggender is en dus ook nuttiger.

```
> P := unapply( sort( expand( Interpolant( L ) ) ), x ):
  t := Matrix(1,2):
  t(1,1) := plot( P(x)-f(x), x=a..b, title="absolute fout,
  equidistante punten" ):
  t(1,2) := plot( (P(x)-f(x))/f(x), x=a..b, title="relatieve
  fout, equidistante punten"):
  display(t);
```



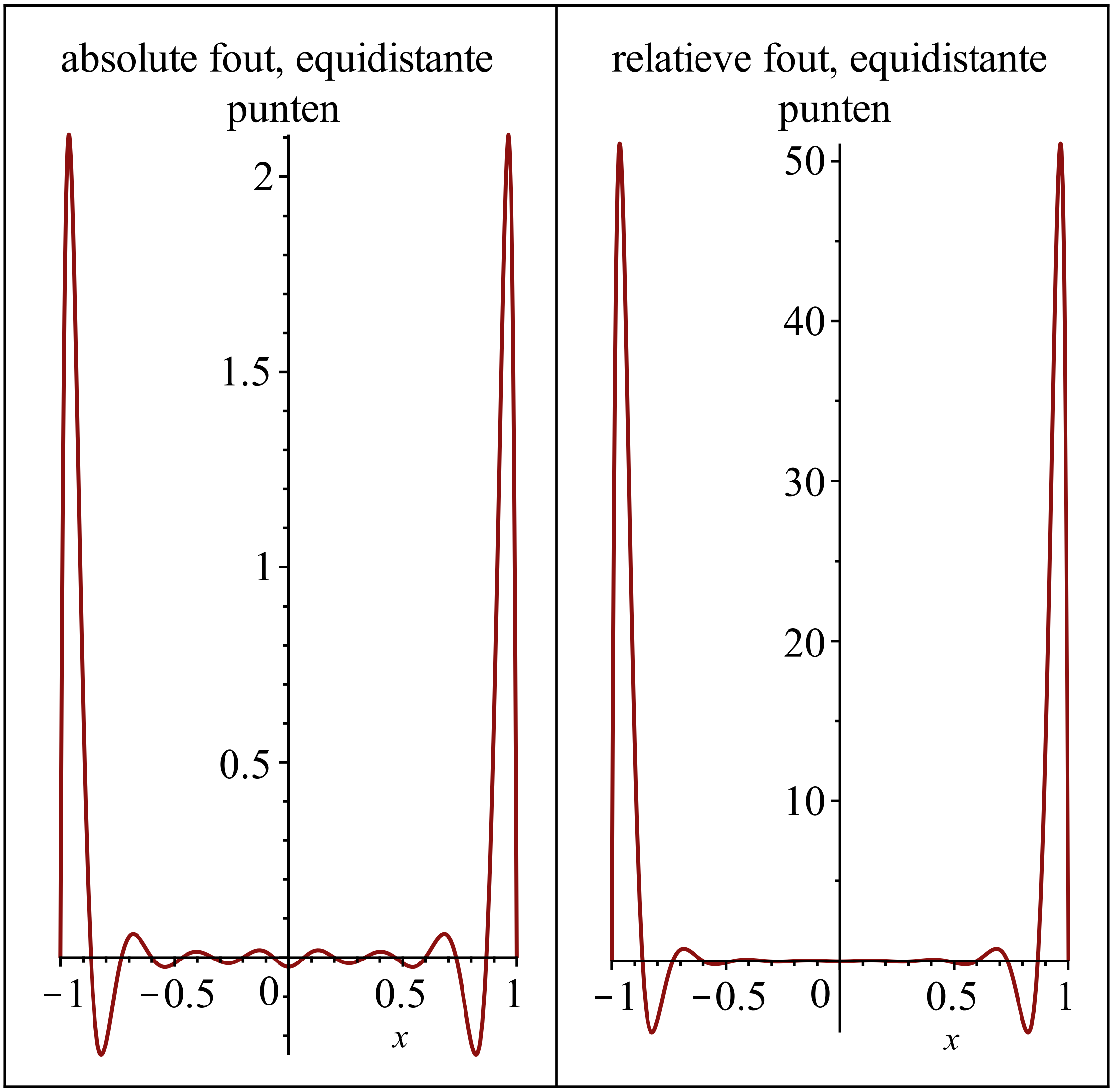
We stellen vast dat de foutenkromme bij veelterminterpolatie met equidistante punten opvallend groter is in de buurt van de grenspunten van het interval $[a, b]$. Dit wordt ook wel het Runge-effect genoemd.

Verhogen graad

Kunnen we dit probleem oplossen door gewoon extra punten toe te voegen?

```
> n := 15:
  xy := [ seq( [x(i),f(x(i))], i=0..n ) ]:
  P := unapply( sort( expand( Interpolant(
  PolynomialInterpolation( xy, independentvar=x, method=newton
  ) ) ) ), x ):
> t := Matrix(1,2):
  t(1,1) := plot( P(x)-f(x), x=a..b, title="absolute fout,
  equidistante punten" ):
  t(1,2) := plot( (P(x)-f(x))/f(x), x=a..b, title="relatieve
  fout, equidistante punten"):
```

```
display(t);
```



We stellen vast dat het verhogen van het aantal interpolatiepunten of de graad van de veelterm zeker geen oplossing biedt. In tegendeel zelfs, de relatieve fout groeit alleen maar, in plaats van een daling.

Veelterminterpolatie met Chebyshev nulpunten

Kunnen we het probleem oplossen door over te schakelen naar andere interpolatiepunten?

Chebyshev nulpunten

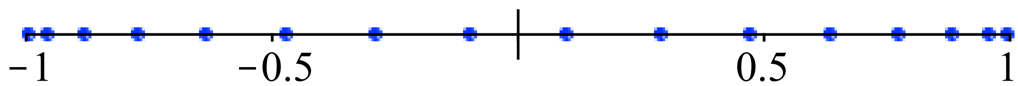
De Chebyshev nulpunten worden gegeven door : $x_i = \cos\left(\frac{(2\,i+1)\pi}{2\,(n+1)}\right)$, $i = 0\,..n$.

```
> x := unapply( cos( 2*(i+1/2)/(2*(n+1))*Pi ), i );
```

$$x := i \mapsto \cos\left(\frac{\left(i+\frac{1}{2}\right)\pi}{16}\right) \tag{2.2.1.1}$$

De Chebyshev nulpunten liggen meer 'geclusterd' aan de eindpunten van het interval $[a, b]$:

```
> pointplot( [ seq([x(i),0], i=0..n ) ], color=blue, symbol=
solidcircle, symbolsize=20, view=[a..b,-0.1..0.1], size=
[default,40], axis[1]=[tickmarks=5], axis[2]=[tickmarks=0] )
;
```

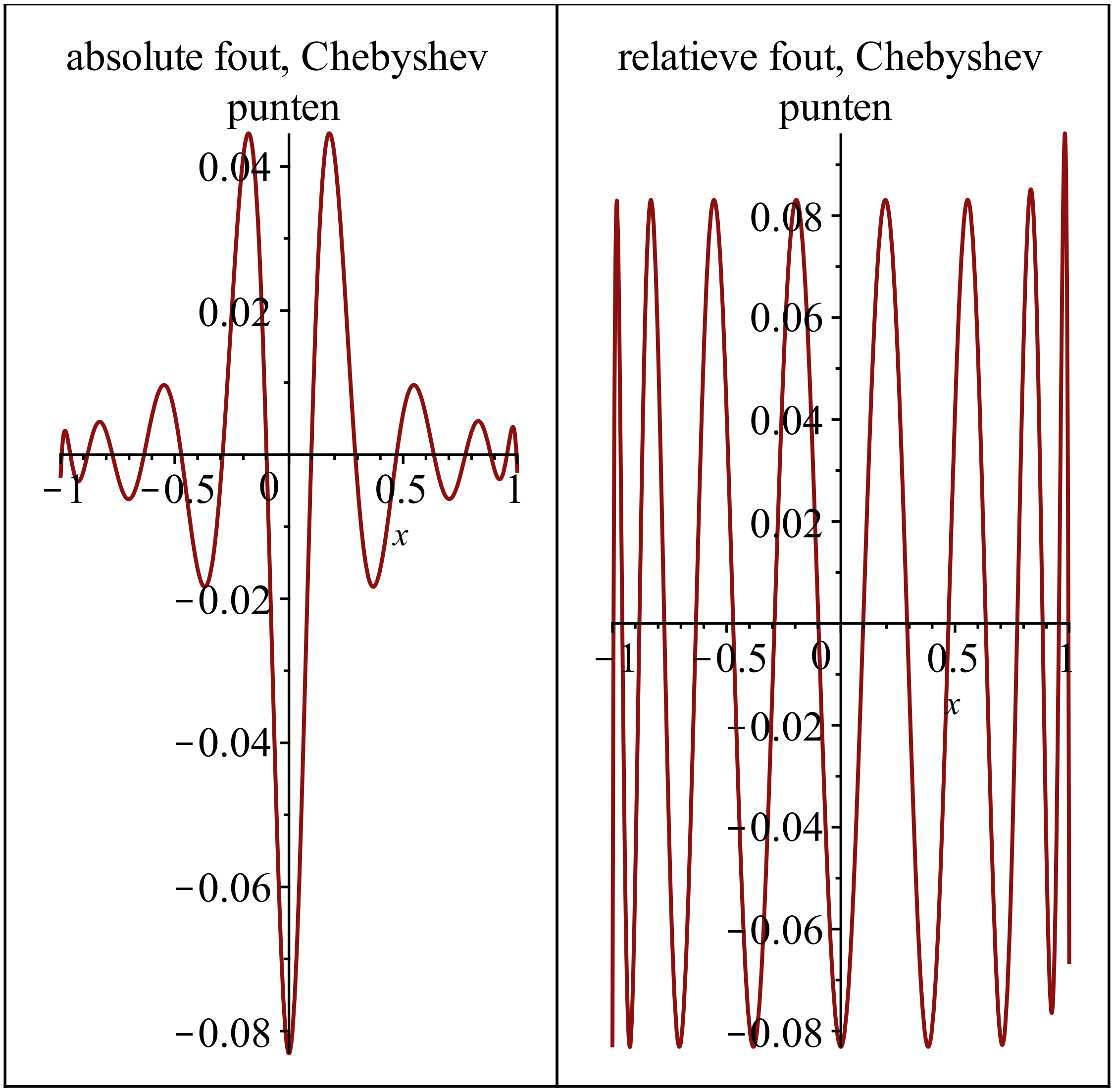


Foutenkromme

```
> xy := [ seq( [x(i),f(x(i))], i=0..n ) ]:
Q := unapply( sort( expand( Interpolant(
PolynomialInterpolation( xy, independentvar=x, method=
lagrange ) ) ) ), x );
```

$$Q := x \mapsto 1.622\ 10^{-6}\, x^{15} - 108.9300710\, x^{14} - 2.00\ 10^{-6}\, x^{13} + 440.0775232\, x^{12} + 0.00012706\, x^{11} - 725.6483883\, x^{10} + 0.00014148\, x^{9} + 628.1414168\, x^{8} + 0.000041093\, x^{7} - 305.9609753\, x^{6} + 5.3249\ 10^{-6}\, x^{5} + 83.72379929\, x^{4} + 2.4213\ 10^{-7}\, x^{3} - 12.28462176\, x^{2} - 9.8249\ 10^{-9}\, x + 0.9168929533 \tag{2.2.2.1}$$

```
> t := Matrix(1,2):
t(1,1) := plot( Q(x)-f(x), x=a..b, title="absolute fout,
Chebyshev punten" ):
t(1,2) := plot( (Q(x)-f(x))/f(x), x=a..b, title="relatieve
fout, Chebyshev punten" ):
display(t);
```



We stellen vast dat :

- In vergelijking met veelterminterpolatie met equidistante interpolatiepunten is de absolute fout gelijkmatiger verdeeld over het interval $[a, b]$.
- Door de 'clustering' van de Chebyshev nulpunten in de buurt van de grenspunten van het interval $[a, b]$ wordt de fout in die grenspunten ook beperkt.
- De Chebyshev nulpunten zijn optimaal voor veelterminterpolatie.

3.2. Stuksgewijze lineaire interpolatie

Om het Runge-effect te vermijden kunnen we overschakelen naar stuksgewijze interpolatie. Op deze manier vermijden we het oscillerend gedrag van veeltermen van hoge graad.
De meest eenvoudige manier om stuksgewijs te interpoleren is natuurlijk stuksgewijze lineaire interpolatie: het verbinden van alle punten met een rechte lijn.

Equidistante punten

```
> unassign('i'); unassign('n');
```

We nemen terug $n+1$ equidistante punten:

```
> h := (b-a)/n;
  x := unapply( a+i*h, i );
```

$$h := \frac{2}{n}$$

$$x := i \mapsto \frac{2\,i}{n} - 1 \tag{3.1.1}$$

Spline graad 1

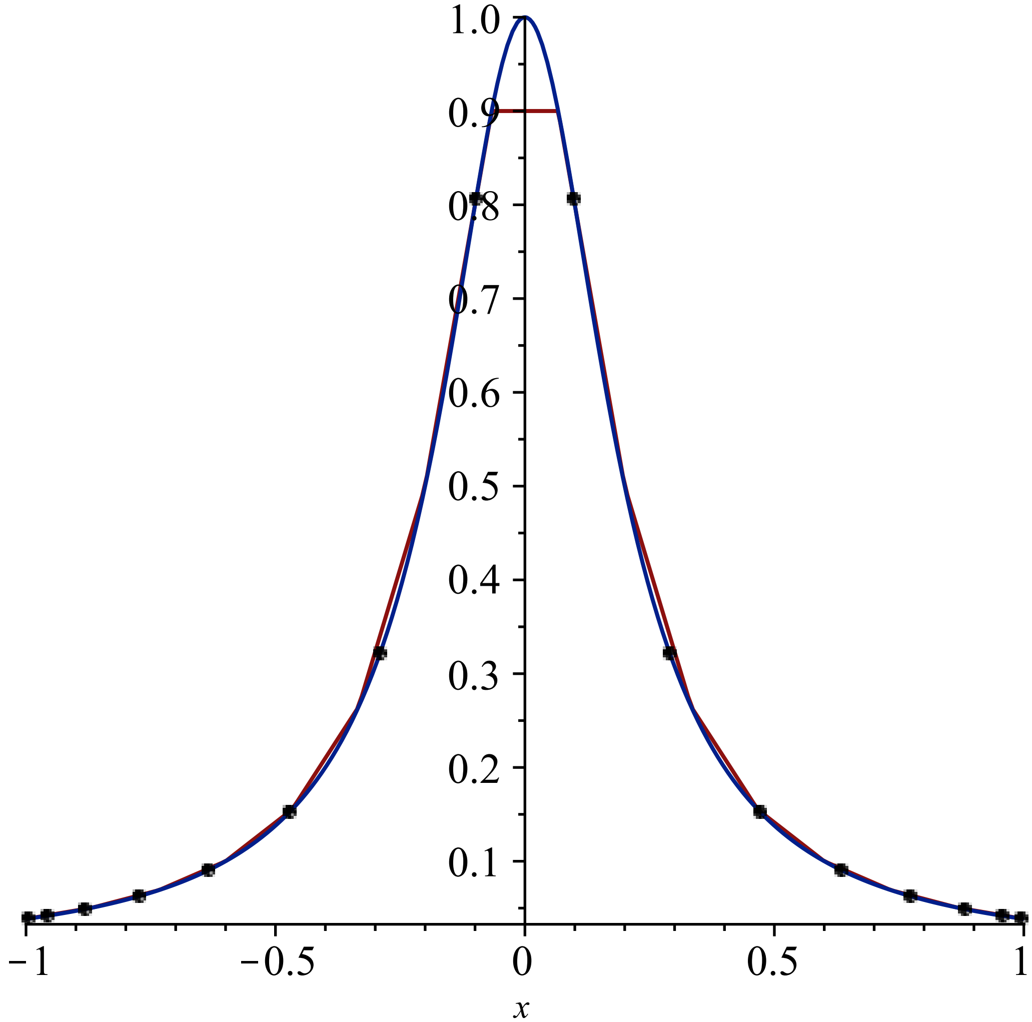
```
> n := 15;
  S := spline( [seq(x(i),i=0..n)],[seq(f(x(i)),i=0..n)],x,1);
```

$$n := 15$$

(3.2.1)

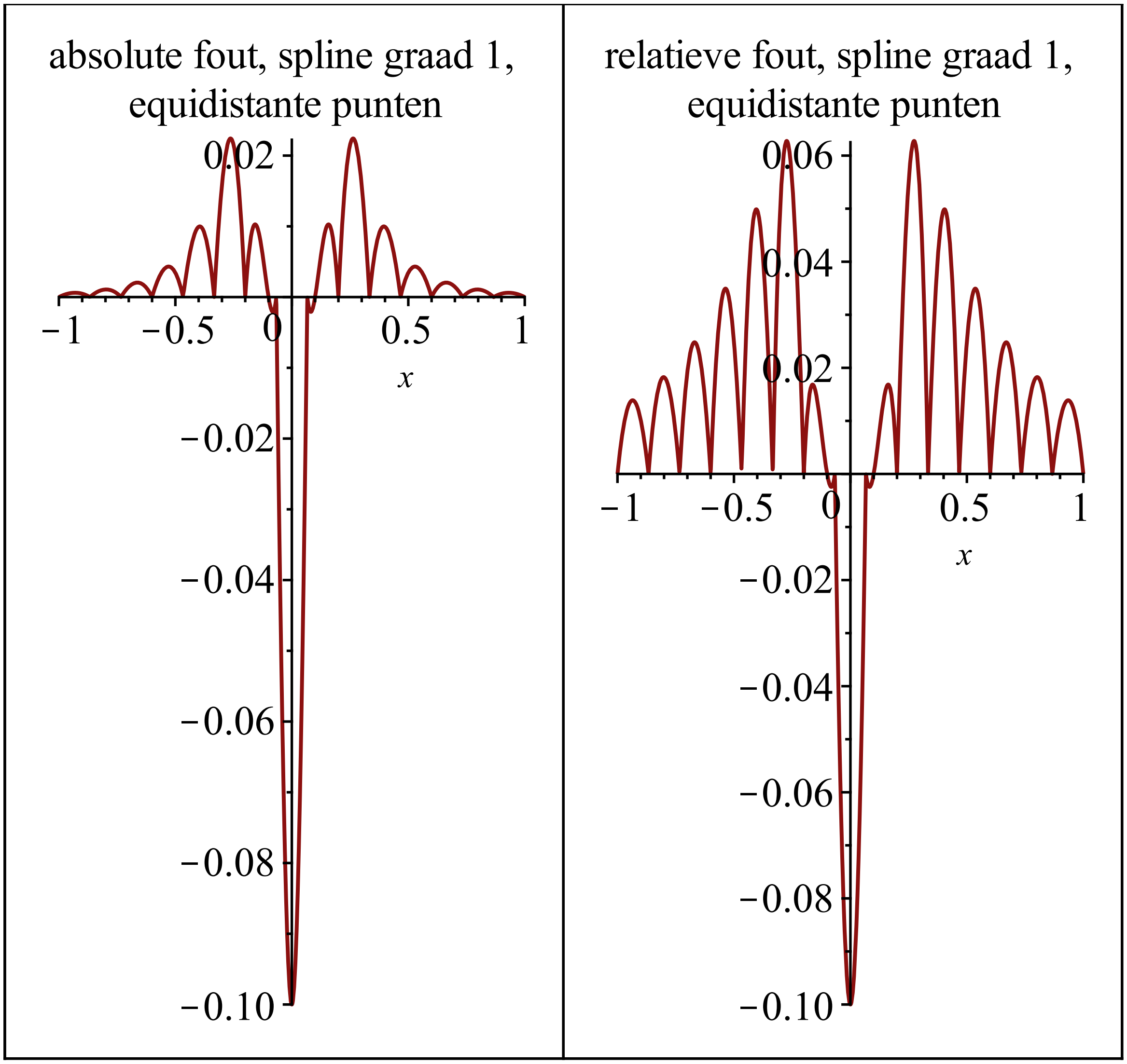
$$S := \begin{cases} \frac{23}{178} + \frac{105\,x}{1157} & x < -\frac{13}{15} \\ \frac{153}{890} + \frac{162\,x}{1157} & x < -\frac{11}{15} \\ \frac{31}{130} + \frac{3\,x}{13} & x < -\frac{3}{5} \\ \frac{101}{290} + \frac{12\,x}{29} & x < -\frac{7}{15} \\ \frac{531}{986} + \frac{405\,x}{493} & x < -\frac{1}{3} \\ \frac{29}{34} + \frac{30\,x}{17} & x < -\frac{1}{5} \\ \frac{11}{10} + 3\,x & x < -\frac{1}{15} \\ \frac{9}{10} & x < \frac{1}{15} \\ \frac{11}{10} - 3\,x & x < \frac{1}{5} \\ \frac{29}{34} - \frac{30\,x}{17} & x < \frac{1}{3} \\ \frac{531}{986} - \frac{405\,x}{493} & x < \frac{7}{15} \\ \frac{101}{290} - \frac{12\,x}{29} & x < \frac{3}{5} \\ \frac{31}{130} - \frac{3\,x}{13} & x < \frac{11}{15} \\ \frac{153}{890} - \frac{162\,x}{1157} & x < \frac{13}{15} \\ \frac{23}{178} - \frac{105\,x}{1157} & \mathit{otherwise} \end{cases} \tag{3.2.1}$$

```
> p1 := plot( [S,f(x)], x=a..b ):
  p2 := pointplot(xy,symbol = solidcircle):
  display(p1,p2);
```



We merken op dat dit een 'hoekig' verloop heeft. Dit betekent dat de eerste afgeleide niet meer continu is!

```
> t := Matrix(1,2):
  t(1,1) := plot( S-f(x), x=a..b, title="absolute fout, spline
  graad 1, equidistante punten" ):
  t(1,2) := plot( (S-f(x))/f(x), x=a..b, title="relatieve fout,
  spline graad 1, equidistante punten" ):
  display(t);
```



3.5. Stuksgewijze kubische interpolatie

We kunnen het vorige resultaat verbeteren door de graad in elk interval te verhogen van 1 naar 3. We kunnen bijgevolg ook de eerste twee afgeleides fitten, wat een visueel mooier resultaat oplevert.

$n+1$ interpolatiepunten => n intervallen, met elk een kubische veelterm => $4\,n$ onbekenden

Maar hoeveel voorwaarden hebben we?

- Elk interval interpolatie door 2 punten => $2\,n$ voorwaarden
- Eerste afgeleide fitten voor binnenste knooppunten => $n-1$ voorwaarden
- Tweede afgeleide fitten voor binnenste knooppunten => $n-1$ voorwaarden

2 voorwaarden te weinig! We voegen dus nog 2 randvoorwaarden toe aan de tweede afgeleide.

Kubische spline met gegeven randvoorwaarden

```
> b1 := eval( diff( f(x), x$2 ), x=a );
```

$$b1 := \frac{925}{4394} \tag{4.1.1}$$

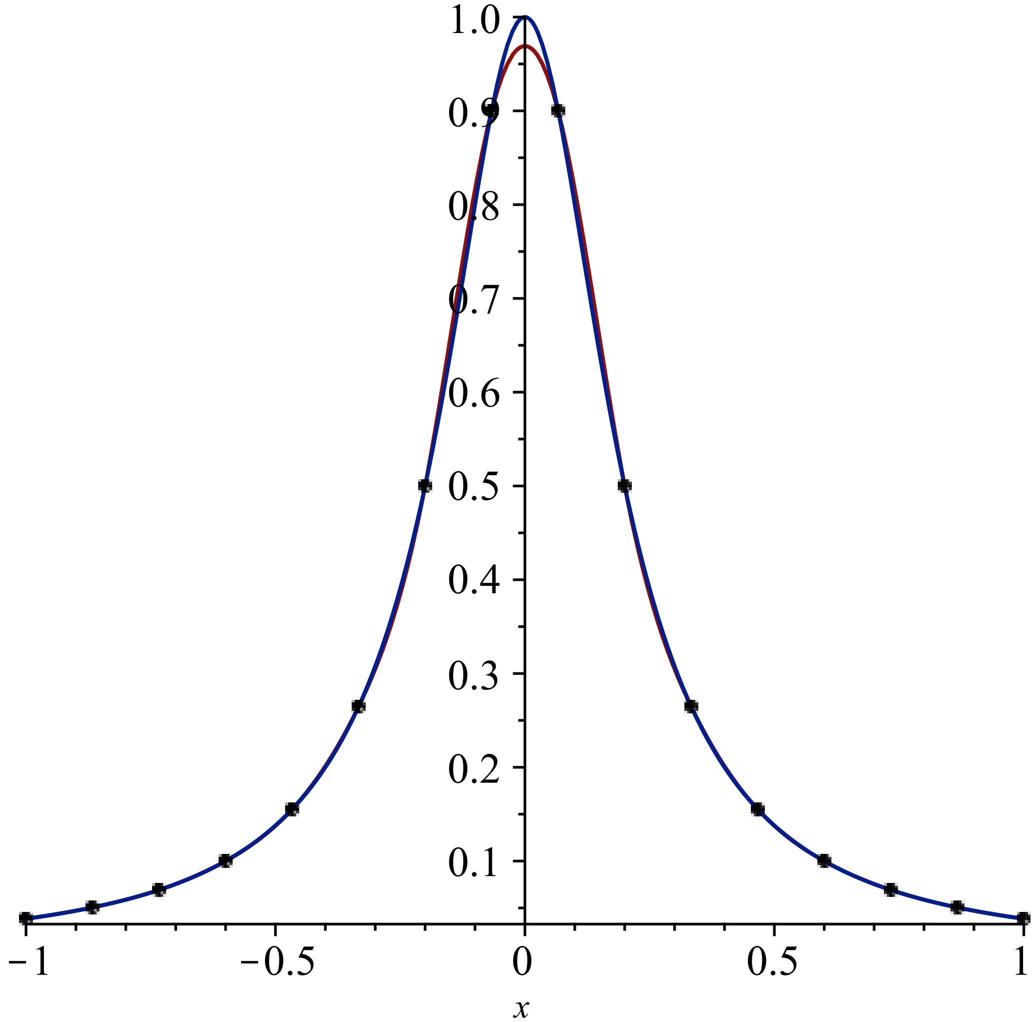
```
> b2 := eval( diff( f(x), x$2 ), x=b );
```

$$b2 := \frac{925}{4394} \tag{4.1.2}$$

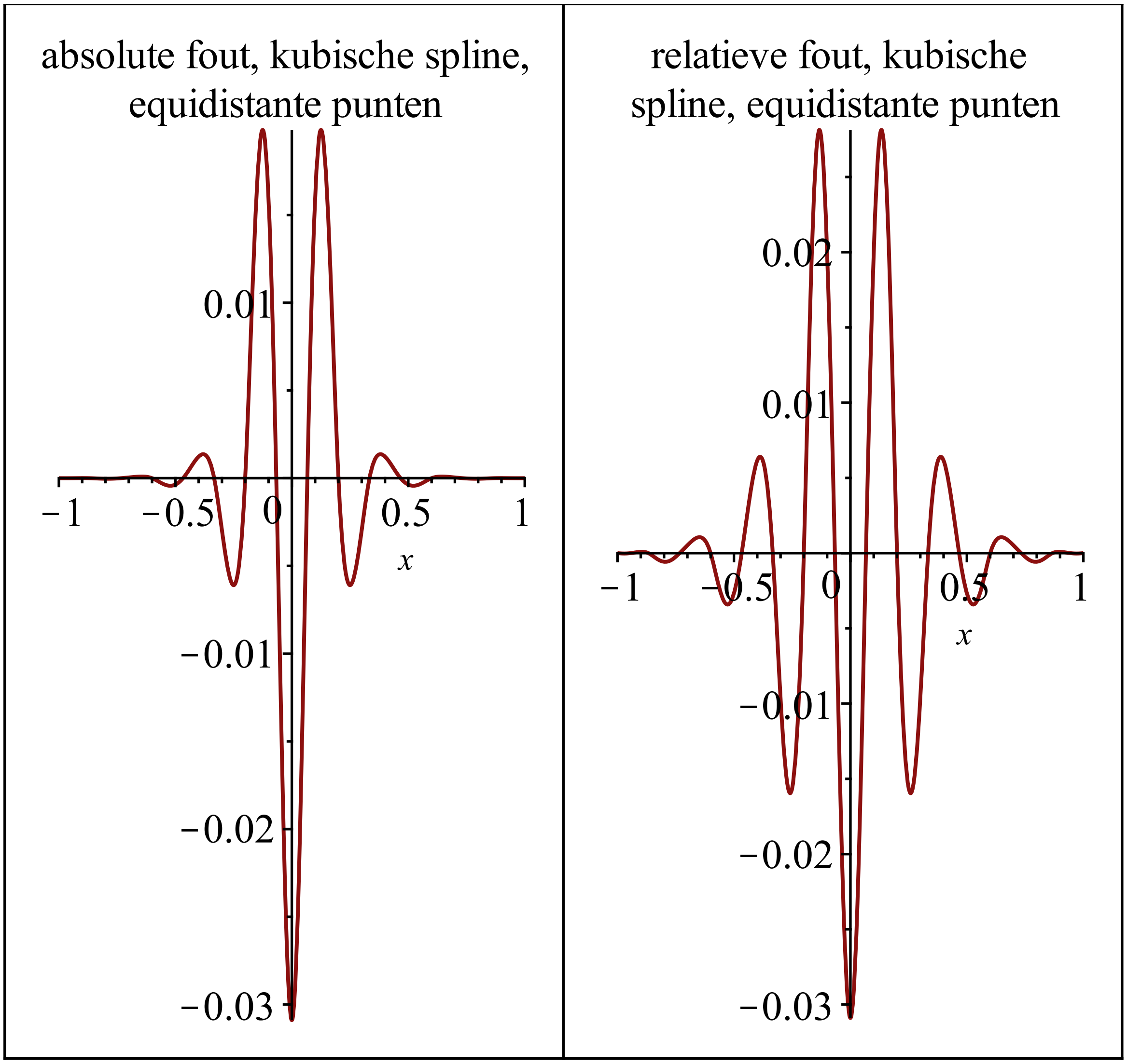
```
> xy := [ seq( [x(i),f(x(i))], i=0..n ) ]:
> G := Matrix( 2, 13, {(1,3)=1,(1,13)=b1, (2,10)=1, (2,13)=b2} )
  :
  S := CurveFitting[Spline]( xy, x, endpoints=G );
```

$$S := \begin{cases}
\frac{16777068213}{106230341438}x^3+\frac{123025419253}{212460682876}x^2+\frac{2013642459047}{2655758535950}x+\frac{9026207989}{24034013900} & x \\
\frac{1645192260}{4085782363}x^3+\frac{1291073290487}{1062303414380}x^2+\frac{1739096273264}{1327879267975}x+\frac{9834149286547}{18386020633500} & x \\
\frac{3159175713}{5591070602}x^3+\frac{1670558765801}{1062303414380}x^2+\frac{245524465761}{156221090350}x+\frac{15905539432901}{26557585359500} & x \\
\frac{134247278976}{53115170719}x^3+\frac{187000783279}{36631152220}x^2+\frac{4901305920936}{1327879267975}x+\frac{2089456252367}{2042891181500} & x \\
\frac{208587171645}{106230341438}x^3+\frac{4584319306793}{1062303414380}x^2+\frac{8824124532191}{2655758535950}x+\frac{46153511838281}{47803653647100} & x \\
\frac{1048497820284}{53115170719}x^3+\frac{23468403996023}{1062303414380}x^2+\frac{12280430886608}{1327879267975}x+\frac{8625220702259}{5311517071900} & x \\
-\frac{6851203069317}{106230341438}x^3-\frac{30220788263287}{1062303414380}x^2-\frac{2283734356439}{2655758535950}x+\frac{19}{20} & x \\
-\frac{16518382124653\,x^2}{1062303414380}+\frac{231634823536603}{239018268235500} & x \\
\frac{6851203069317}{106230341438}x^3-\frac{30220788263287}{1062303414380}x^2+\frac{2283734356439}{2655758535950}x+\frac{19}{20} & \\
-\frac{1048497820284}{53115170719}x^3+\frac{23468403996023}{1062303414380}x^2-\frac{12280430886608}{1327879267975}x+\frac{8625220702259}{5311517071900} & \\
-\frac{208587171645}{106230341438}x^3+\frac{4584319306793}{1062303414380}x^2-\frac{8824124532191}{2655758535950}x+\frac{46153511838281}{47803653647100} & x \\
-\frac{134247278976}{53115170719}x^3+\frac{187000783279}{36631152220}x^2-\frac{4901305920936}{1327879267975}x+\frac{2089456252367}{2042891181500} & \\
-\frac{3159175713}{5591070602}x^3+\frac{1670558765801}{1062303414380}x^2-\frac{245524465761}{156221090350}x+\frac{15905539432901}{26557585359500} & x \\
-\frac{1645192260}{4085782363}x^3+\frac{1291073290487}{1062303414380}x^2-\frac{1739096273264}{1327879267975}x+\frac{9834149286547}{18386020633500} & x \\
-\frac{16777068213}{106230341438}x^3+\frac{123025419253}{212460682876}x^2-\frac{2013642459047}{2655758535950}x+\frac{9026207989}{24034013900} & o
\end{cases}$$

```
> p1 := plot( [S,f(x)], x=a..b ):
  p2 := pointplot(xy,symbol = solidcircle):
  display(p1,p2);
```




```
> t := Matrix(1,2):
  t(1,1) := plot( S-f(x), x=a..b, title="absolute fout, kubische
  spline, equidistante punten" ):
  t(1,2) := plot( (S-f(x))/f(x), x=a..b, title="relatieve fout,
  kubische spline, equidistante punten" ):
  display(%);
```



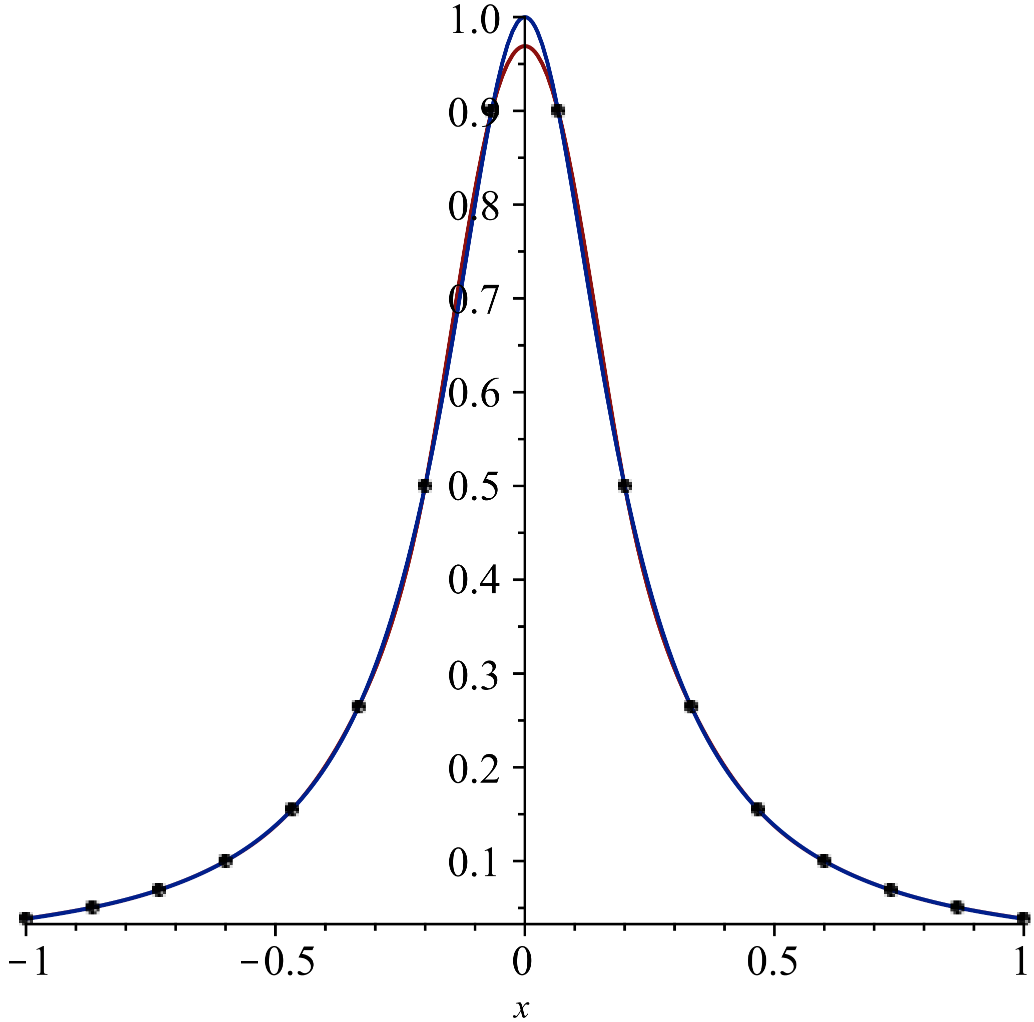
Natuurlijke kubische spline

Wat als we de randvoorwaarden niet kennen? Een eenvoudige optie is om de tweede afgeleide op de randen gelijk te stellen aan 0. Dit noemen we de **natuurlijke kubische spline**.

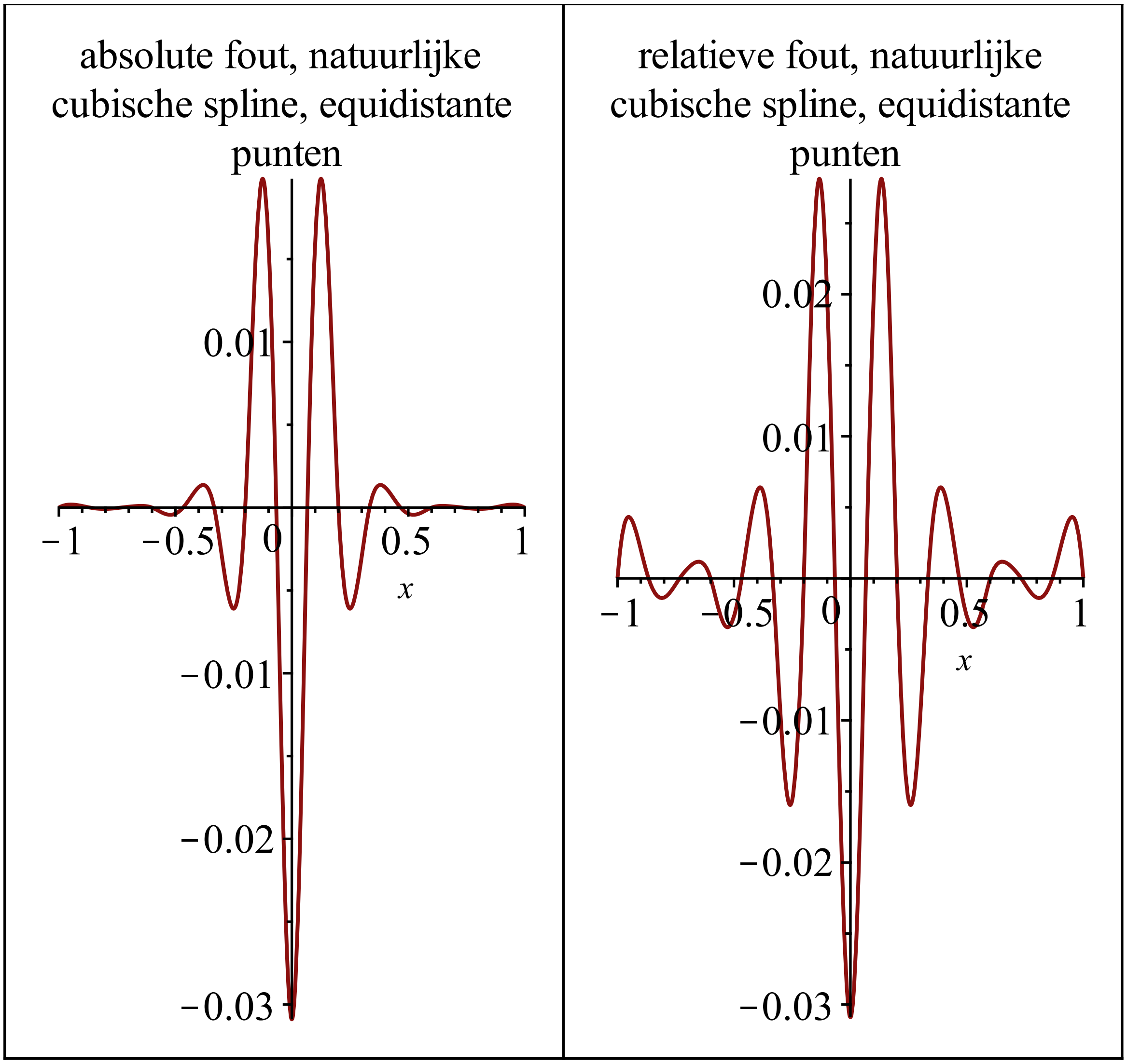
```
> S := CubicSpline( xy, independentvar=x, boundarycondition=
  natural ):
> S := Interpolant(S):
  S := sort( expand( % ) );
```

$$S := \begin{cases} \frac{618000219}{1257163804} x^3 + \frac{1854000657}{1257163804} x^2 + \frac{48927601161}{31429095100} x + \frac{87042541}{142213100} & x < -\frac{13}{15} \\ \frac{98455095}{314290951} x^3 + \frac{3177832689}{3142909510} x^2 + \frac{9074700891}{7857273775} x + \frac{1498854063}{3022028375} & x < -\frac{11}{15} \\ \frac{38971719}{66166516} x^3 + \frac{10168730469}{6285819020} x^2 + \frac{2957649543}{1848770300} x + \frac{95028591869}{157145475500} & x < -\frac{3}{5} \\ \frac{60949638}{24176227} x^3 + \frac{21239388}{4168315} x^2 + \frac{2228093118}{604405675} x + \frac{6177284299}{6044056750} & x < -\frac{7}{15} \\ \frac{190049895}{96704908} x^3 + \frac{2087528409}{483524540} x^2 + \frac{8034477741}{2417622700} x + \frac{2334352317}{2417622700} & x < -\frac{1}{3} \\ \frac{477229617}{24176227} x^3 + \frac{5340935637}{241762270} x^2 + \frac{5589595629}{604405675} x + \frac{981474323}{604405675} & x < -\frac{1}{5} \\ -\frac{6236860167}{96704908} x^3 - \frac{13755464631}{483524540} x^2 - \frac{2078953389}{2417622700} x + \frac{19}{20} & x < -\frac{1}{15} \\ -\frac{1879651116\, x^2}{120881135} + \frac{5857351323}{6044056750} & x < \frac{1}{15} \\ \frac{6236860167}{96704908} x^3 - \frac{13755464631}{483524540} x^2 + \frac{2078953389}{2417622700} x + \frac{19}{20} & x < \frac{1}{5} \\ -\frac{477229617}{24176227} x^3 + \frac{5340935637}{241762270} x^2 - \frac{5589595629}{604405675} x + \frac{981474323}{604405675} & x < \frac{1}{3} \\ -\frac{190049895}{96704908} x^3 + \frac{2087528409}{483524540} x^2 - \frac{8034477741}{2417622700} x + \frac{2334352317}{2417622700} & x < \frac{7}{15} \\ -\frac{60949638}{24176227} x^3 + \frac{21239388}{4168315} x^2 - \frac{2228093118}{604405675} x + \frac{6177284299}{6044056750} & x < \frac{3}{5} \\ -\frac{38971719}{66166516} x^3 + \frac{10168730469}{6285819020} x^2 - \frac{2957649543}{1848770300} x + \frac{95028591869}{157145475500} & x < \frac{11}{15} \\ -\frac{98455095}{314290951} x^3 + \frac{3177832689}{3142909510} x^2 - \frac{9074700891}{7857273775} x + \frac{1498854063}{3022028375} & x < \frac{13}{15} \\ -\frac{618000219}{1257163804} x^3 + \frac{1854000657}{1257163804} x^2 - \frac{48927601161}{31429095100} x + \frac{87042541}{142213100} & \textit{otherwise} \end{cases} \quad \textbf{(4.2.1)}$$

```
> p1 := plot( [S,f(x)], x=a..b ):
  p2 := pointplot(xy,symbol = solidcircle):
  display(p1,p2);
```




```
> t := Matrix(1,2):
  t(1,1) := plot( S-f(x), x=a..b, title="absolute fout,
  natuurlijke cubische spline, equidistante punten" ):
  t(1,2) := plot( (S-f(x))/f(x), x=a..b, title="relatieve fout,
  natuurlijke cubische spline, equidistante punten" ):
  display(t);
```



We stellen vast dat de fout op de randen iets groter is. Dit is logisch aangezien we eigenlijk de verkeerde waardes fitten voor de tweede afgeleide op de rand.

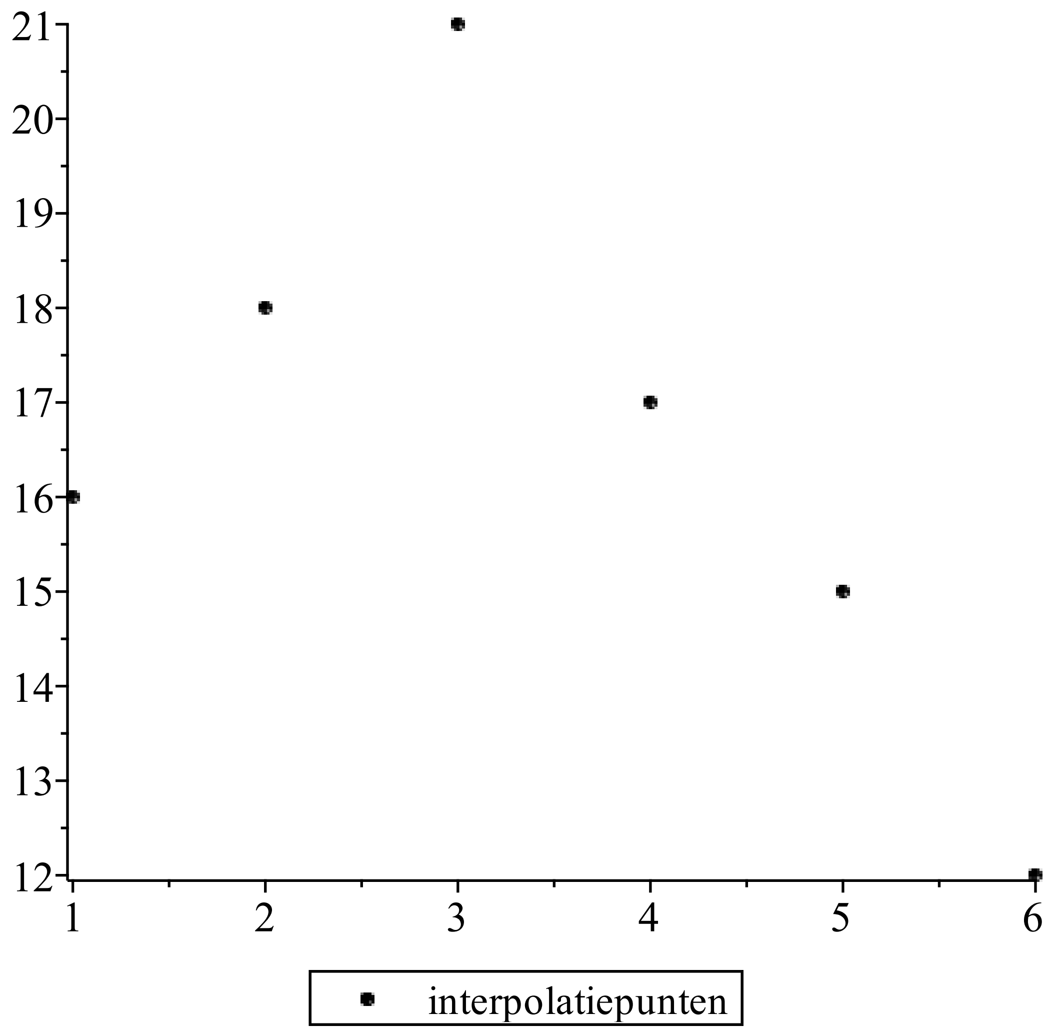
Vergelijking interpolatie en spline

```
> a := 1:
  b := 6:
> y := [16, 18, 21, 17, 15, 12];
  xy := [seq([i,y[i]],i=1..6)];
```

$$y := [16, 18, 21, 17, 15, 12]$$

$$xy := [[1, 16], [2, 18], [3, 21], [4, 17], [5, 15], [6, 12]] \quad \textbf{(5.1)}$$

```
> pp := pointplot(xy,symbol = solidcircle, legend =
  "interpolatiepunten");
```



3.1. Veelterminterpolant

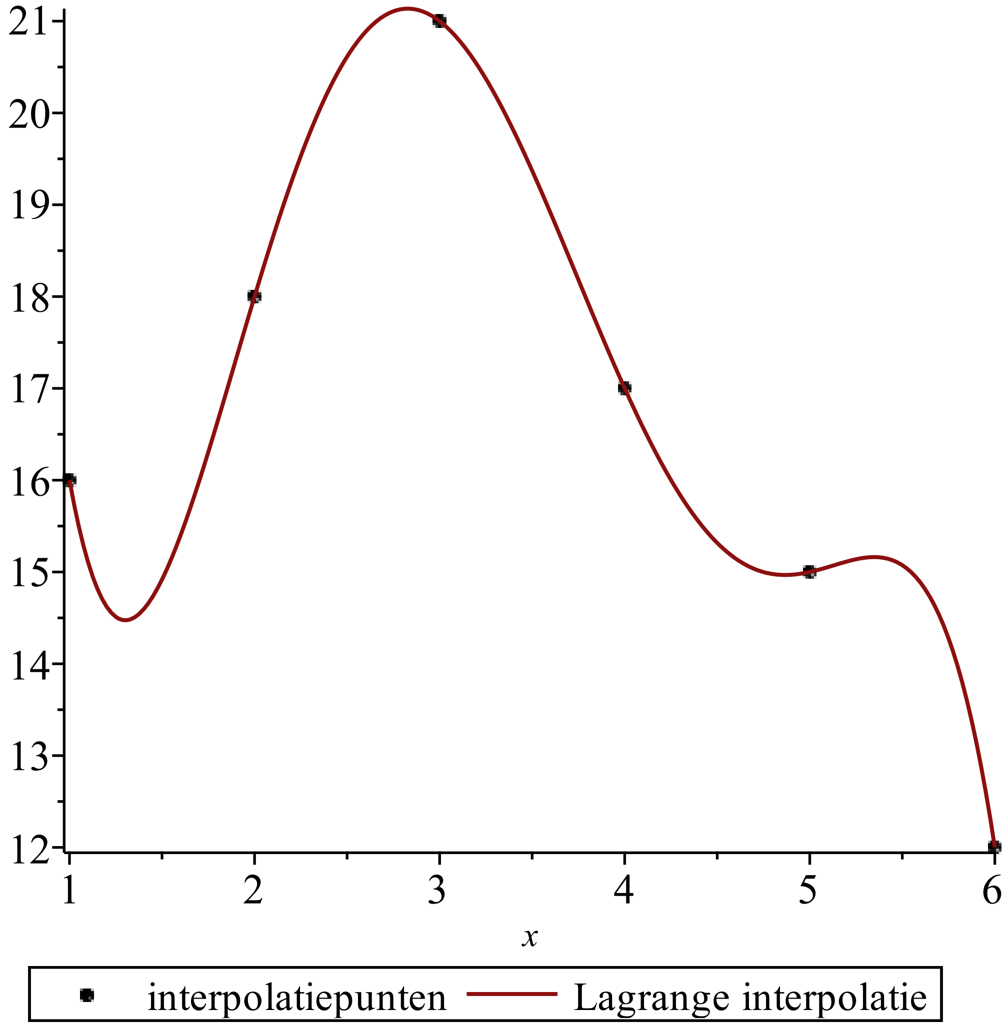
```
> L := PolynomialInterpolation( xy, independentvar=x, method=
  lagrange ):
  L := Interpolant( L );
```

$$L := -\frac{2\,(x-2)\,(x-3)\,(x-4)\,(x-5)\,(x-6)}{15} + \frac{3\,(x-1)\,(x-3)\,(x-4)\,(x-5)\,(x-6)}{4} - \frac{7\,(x-1)\,(x-2)\,(x-4)\,(x-5)\,(x-6)}{4} + \frac{17\,(x-1)\,(x-2)\,(x-3)\,(x-5)\,(x-6)}{12} - \frac{5\,(x-1)\,(x-2)\,(x-3)\,(x-4)\,(x-6)}{8} + \frac{(x-1)\,(x-2)\,(x-3)\,(x-4)\,(x-5)}{10} \tag{5.1.1}$$

```
> sort( expand( L ), x, ascending );
```

$$69 - \frac{579}{5} x + \frac{263}{3} x^2 - \frac{695}{24} x^3 + \frac{13}{3} x^4 - \frac{29}{120} x^5 \quad \textbf{(5.1.2)}$$

```
> pL := plot(L,x=a..b, legend = "Lagrange interpolatie"):
  display([pp,pL]);
```



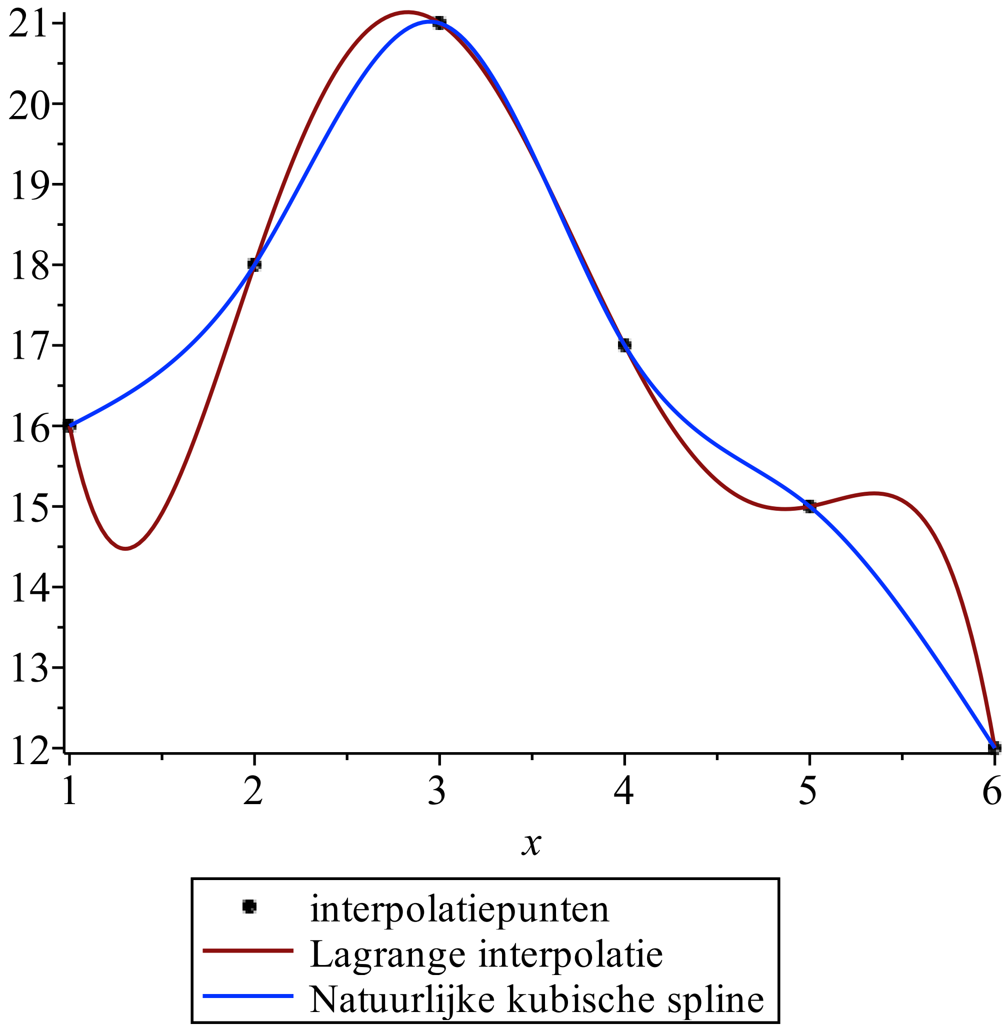
3.5. Kubische spline

We hebben geen informatie, behalve de interpolatiepunten. We moeten dus zelf nog de randvoorwaarden opstellen. We kiezen ervoor om de natuurlijke kubische spline te berekenen.

```
> S := CubicSpline( xy, independentvar=x, boundarycondition=
  natural ):
> S := Interpolant(S):
  S := sort( expand( % ) );
```

$$S := \begin{cases} \frac{170}{209}x^3 - \frac{510}{209}x^2 + \frac{758}{209}x + 14 & x < 2 \\ -\frac{641}{209}x^3 + \frac{396}{19}x^2 - \frac{8974}{209}x + \frac{9414}{209} & x < 3 \\ \frac{38}{11}x^3 - \frac{7911}{209}x^2 + \frac{27827}{209}x - \frac{27387}{209} & x < 4 \\ -\frac{366}{209}x^3 + \frac{5145}{209}x^2 - \frac{24397}{209}x + \frac{42245}{209} & x < 5 \\ \frac{115}{209}x^3 - \frac{2070}{209}x^2 + \frac{11678}{209}x - \frac{17880}{209} & \mathit{otherwise} \end{cases} \tag{5.2.1}$$

```
> pS := plot(S,x=a..b, legend = "Natuurlijke kubische spline",
  color = "blue"):
  display([pp,pL,pS]);
```



We merken op dat de natuurlijke kubische spline veel minder oscilleert. Dit is namelijk de interpolant die de kromming minimaliseert op het interval $[a, b]$.