Probleem 1:

(a)

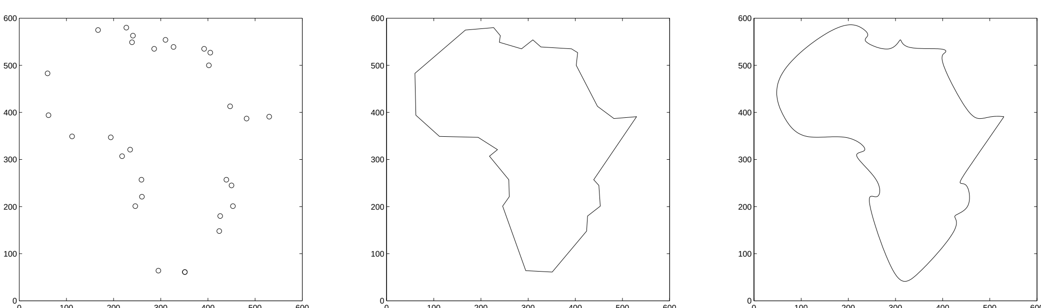
```
>> load AFRIKA
>> plot(x,y,'o'); axis('square');      % sien Figuur (a)
```

(b)

```
>> plot(x,y); axis('square');          % sien Figuur (b)
```

(c)

```
>> t = 1:30;
>> T = linspace(1,30,501);
>> X = spline(t,x,T);
>> Y = spline(t,y,T);
>> plot(X,Y); axis('square');          % sien Figuur (c)
```



(a) Datapunte (b) Stuksgewys lineêr (c) Met `spline`

Probleem 2:

(a)

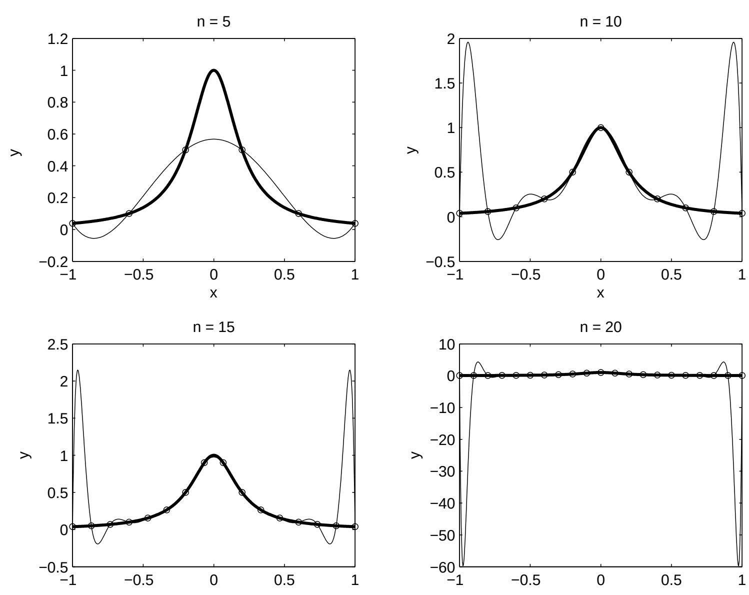
```
% TW324:  MATLAB program wat die Runge-verskynsel illustreer  (Weideman, 2004)

F = inline('1./(1+25*x.^2)');           % Definieer Runge se funksie
X = linspace(-1,1,501);                 % Fyn rooster vir stip
Y = F(X);                               % Evalueer funksie op fyn rooster

for n = 5:5:20
    subplot(2,2,n/5); hold off;

    xk = -1+[0:n]*2/n;                  % Gelykverspreide punte
    yk = F(xk);                         % Evalueer y-waardes
    a = polyfit(xk,yk,n);               % Vind die koeff. van die polinoom ...
    P = polyval(a,X);                   % ... en evalueer dit.

    plot(X,Y,'LineWidth',2); hold on;   % Stip die Runge funksie
    plot(xk,yk,'o','MarkerSize',4);     % Stip die data
    plot(X,P);                          % Stip die interpolant
    xlabel('x'); ylabel('y'); title(['n = ' num2str(n)])
end
```



(b)

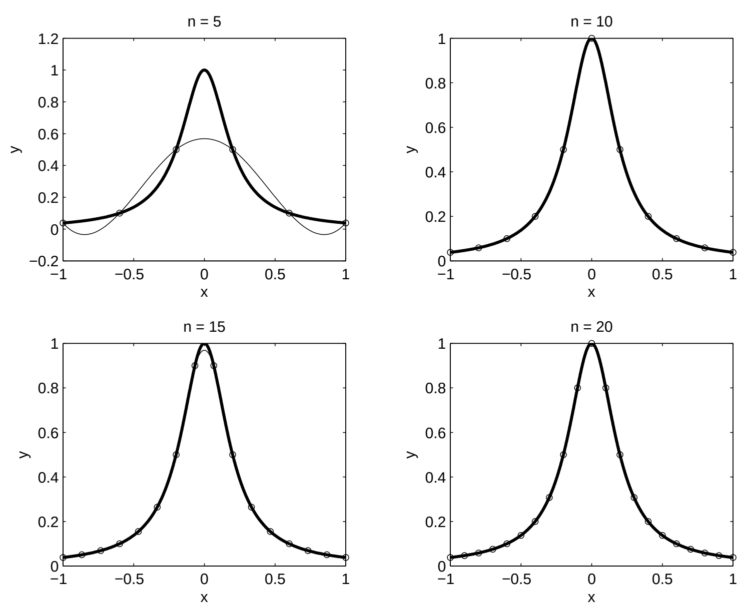
```
% TW324:  MATLAB program wat die Runge-verskynsel illustreer  (Weideman, 2004)

F = inline('1./(1+25*x.^2)');           % Definieer Runge se funksie
X = linspace(-1,1,501);                 % Fyn rooster vir stip
Y = F(X);                               % Evalueer funksie op fyn rooster

for n = 5:5:20
    subplot(2,2,n/5); hold off;

    xk = -1+[0:n]*2/n;                  % Gelykverspreide punte
    yk = F(xk);                         % Evalueer y-waardes
    S = spline(xk,yk,X);                % Evalueer latfunksie interpolant  <---

    plot(X,Y,'LineWidth',2); hold on;   % Stip die Runge funksie
    plot(xk,yk,'o','MarkerSize',4);     % Stip die data
    plot(X,S);                          % Stip die interpolant             <---
    xlabel('x'); ylabel('y'); title(['n = ' num2str(n)])
end
```



Dit lyk asof die Runge-ossillasies nie meer teenwoordig is nie.

(c)

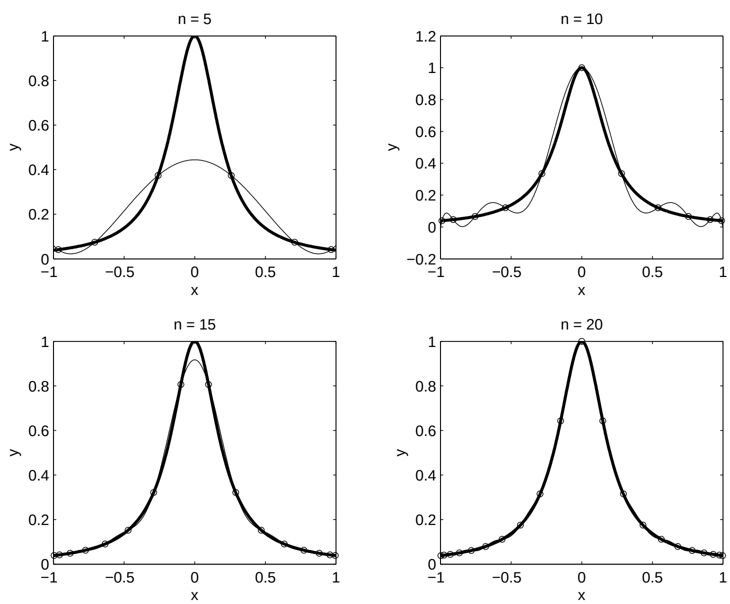
```
% TW324:  MATLAB program wat die Runge-verskynsel illustreer  (Weideman, 2004)

F = inline('1./(1+25*x.^2)');            % Definieer Runge se funksie
X = linspace(-1,1,501);                  % Fyn rooster vir stip
Y = F(X);                                % Evalueer funksie op fyn rooster

for n = 5:5:20
    subplot(2,2,n/5); hold off;

    xk = cos((2*[0:n]+1)*pi/(2*n+2));    % Chebyshev punte                 <---
    yk = F(xk);                          % Evalueer y-waardes
    a  = polyfit(xk,yk,n);               % Vind die koeff. van die polinoom ...
    P  = polyval(a,X);                   % ... en evalueer dit.

    plot(X,Y,'LineWidth',2); hold on;    % Stip die Runge funksie
    plot(xk,yk,'o','MarkerSize',4);      % Stip die data
    plot(X,P);                           % Stip die interpolant
    xlabel('x'); ylabel('y'); title(['n = ' num2str(n)])
end
```



Dit lyk asof die Runge-ossillasies hier ook nie meer teenwoordig is nie.

Probleem 3:

Laat $x \in [0,2]$, en $t \in [-1,1]$, sodat $x = at + b$. Dan

$$0 = -a + b, \qquad 2 = a + b.$$

Los op vir a en b uit bostaande: $a = 1$ en $b = 1$, sodat $x = t + 1$.

(a)

Die Chebyshev nulpunte (vir $n = 3$) van $T_4(x)$ in $[-1,1]$ word gegee deur

$$t_0 = \cos\frac{\pi}{8}, \quad t_1 = \cos\frac{3\pi}{8}, \quad t_2 = \cos\frac{5\pi}{8}, \quad t_3 = \cos\frac{7\pi}{8}.$$

Die interpolasiepunte in $[0,2]$ word dus gegee deur

$$x_0 = 1 + \cos\frac{\pi}{8}, \quad x_1 = 1 + \cos\frac{3\pi}{8}, \quad x_2 = 1 + \cos\frac{5\pi}{8}, \quad x_3 = 1 + \cos\frac{7\pi}{8}.$$

Pas die polinoom m.b.v. `polyfit`:

```
xk = 1+cos([1:2:7]*pi/8);
yk = exp(-xk);
a = polyfit(xk,yk,3);
X = linspace(0,2,501);
P = polyval(a,X);
F = exp(-X);
```

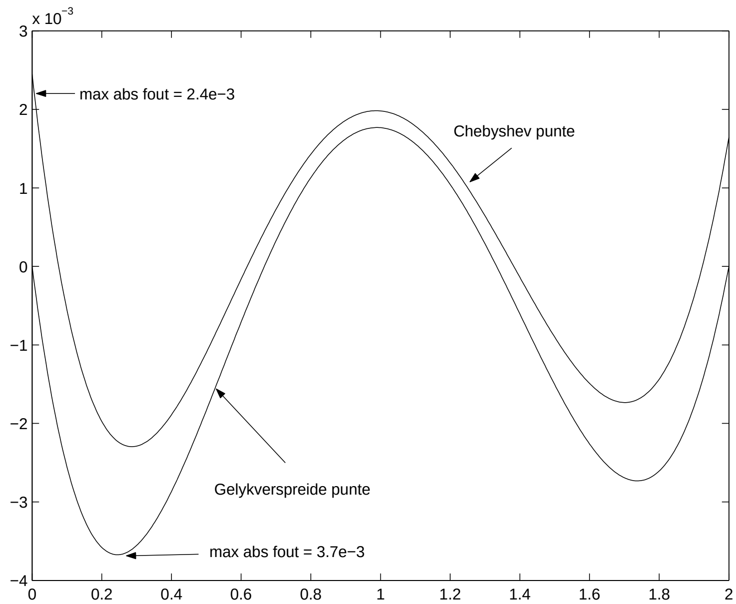
(b)

Vir gelykverspreide punte kan ons dieselfde kode as hierbo gebruik, deur net die definisie van x_k te verander na

```
xk = [0:3]*(2/3);
```

(c)

Onderstaande grafiek toon die foutkrommes. Die maksimum absolute fout vir Chebyshev punte is ongeveer 2.4×10^{-3}, wat kleiner is as die 3.7×10^{-3} wat ons met gelykverspreide punte kry.



(d)

Die vierde afgeleide van $f(x)$ word gegee deur $f^{(4)}(x) = e^{-x}$. Dus

$$M_3 \;=\; \max_{0 \le x \le 2} e^{-x} \;=\; 1 \;\; (\text{by } x = 0).$$

Vir gelykverspreide punte is $h = \frac{2}{3}$. Die foutgrens vir die middelste interval, $\frac{2}{3} < x < \frac{4}{3}$, word dus

$$\max |f(x) - p_3(x)| \;\le\; \frac{3}{128} \cdot \left(\frac{2}{3}\right)^4 \cdot 1 \;=\; 4.6296 \times 10^{-3}.$$

Deur in te zoem op die grafiek van deel (c) sien ons dat die werklike maksimum absolute fout in hierdie interval ongeveer 1.77×10^{-3} is, wat kleiner is as die teoretiese foutgrens (soos verwag).

Die foutgrens vir die buitenste twee intervalle, $0 < x < \frac{2}{3}$ en $\frac{4}{3} < x < 2$, is

$$\max |f(x) - p_3(x)| \;\le\; \frac{1}{24} \cdot \left(\frac{2}{3}\right)^4 \cdot 1 \;=\; 8.2305 \times 10^{-3}.$$

Uit die grafiek van deel (c) sien ons dat die werklike maksimum absolute fout in die interval $0 < x < \frac{2}{3}$ ongeveer 3.67×10^{-3} is, en in die interval $\frac{4}{3} < x < 2$ is dit ongeveer 2.73×10^{-3}. Beide hierdie waardes is kleiner as die teoretiese foutgrens (soos verwag).

(e)

Duidelik is

$$(x - x_0)(x - x_1)(x - x_2)(x - x_3) \;=\; (t - t_0)(t - t_1)(t - t_2)(t - t_3),$$

sodat die maksimum absolute fout van die linkerkant op $[0, 2]$ gelyk is aan die maksimum absolute fout van die regterkant op $[-1, 1]$. Ons pas dus die stelling op p. 361 toe, en kry op $[0, 2]$

$$\max_{0 \le x \le 2} |f(x) - p_3(x)| \;\le\; \frac{1}{2^3 4!} \cdot 1 \;=\; 5.2083 \times 10^{-3}.$$

Deur in te zoem op die grafiek van deel (c) sien ons dat die werklike maksimum absolute fout in hierdie interval ongeveer 2.4×10^{-3} is, wat kleiner is as die teoretiese foutgrens, soos dit hoort.