```
>
```

02. / 03. August 2016

Pulsdruckwerte aus Langzeitmessung

Univ.-Prof. Dr.-Ing. habil. Josef BETTEN
RWTH Aachen University
Mathematical Models in Materials Science and Continuum Mechanics
Augustinerbach 4-20
D-52056 A a c h e n , Germany

<betten@mmw.rwth-aachen.de>

Puls Pressure Values. During a period of nearly 24 hours the blood pressure of a patient at a Hospital in Aachen has been measured. Thus, we have a lot of *Systole-*, *Diastole-*, *Pulspressure-, and Pulse-Values* important for a medical doctor treating sick patients. The following worksheet has been concerned with the mathematical analyse of these given "data".

Gemessen werden in einem Zeitraum von 24 h *Systole-*, *Diastole-Werte* [mmHg] und *Herzfrequenzen* [-/min]. Der *Pulsdruck* [mmHg] ergibt sich aus der Differenz von *Systole* und *Diastole*, bisweilen auch als "Amplitude" bezeichnet. Im Folgenden sind diese Daten aufgelistet und statistisch ausgewertet. Unterschieden wird zwischen Tag (von 9 - 22 Uhr) und Nachtruhe (von 22 - 6 Uhr). Zur grafischen Darstellung werden *kubische Splinefunktionen* benutzt. Die *Splineinterpolation* bietet wesentliche Vorteile beispielsweise gegenüber einer *LAGRANGE Interpolation*, die bei einer Vielzahl von Messpunkten zu starken Schwingungen neigt und häufig keine geeigneten Ergebnisse liefert. Die *kubischen Splinefunktionen* werden im Folgenden genähert durch *FOURIER-Reihen* oder durch *nichtlineare Regressionen.*

I) Experimenteller Befund

```
> restart:
> with(Statistics):  with(CurveFitting):
```

Systole-Messwerte DATA [mmHg]

```
> DATA:=
  [9,140],[10,123],[11,112],[12,111],[13,110],[14,107],[15,96],
  [16,111],[17,110],[18,128],[19,131],[20,95],[21,121],[22,113],
  [23,100],[24,112],[25,114],[26,126],[27,131],[28,128],[29,136],
  [30,137]:
```

Diastole-Messwerte data [mmHg]

```
> data:=
  [9,84],[10,88],[11,71],[12,72],[13,80],[14,69],[15,55],[16,74],
```

```
[17,69],[18,78],[19,84],[20,61],[21,83],[22,70],[23,60],[24,70],
[25,67], [26,77], [27,76],[28,78],[29,73],[30,77]:
```

Pulsdruck = (Systole - Diastole) [mmHg]

```
> PD:=
[9,56],[10,35],[11,41],[12,39],[13,30],[14,38],[15,41],[16,37],
[17,41],[18,50],[19,47],[20,34],[21,38],[22,43],[23,40],[24,41],
[25,47], [26,49], [27,55],[28,45],[29,63],[30,60]:
```

Herzfrequenz Hf [- / min]

```
> Hf:=
[9,75],[10,82],[11,62],[12,63],[13,62],[14,60],[15,57],[16,54],
[17,51],[18,45],[19,54],[20,54],[21,66],[22,55],[23,47],[24,50],
[25,44], [26,58], [27,43],[28,45],[29,47],[30,41]:
```

II) Statistische Werte

Mittelwert über alle Systole-Werte (M) und Standardabweichung (S)

```
> DATAY:=array([seq(DATA[i][2],i=1..22)]);
```

$DATAY := [140, 123, 112, 111, 110, 107, 96, 111, 110, 128, 131, 95, 121, 113, 100, 112, 114, 126, 131, 128, 136, 137]$

```
> M:=evalf(Mean(DATAY),4)*mmHg;
```

$$M := 117.8\ mmHg$$

```
> S:=evalf(StandardDeviation(DATAY),4)*mmHg;
```

$$S := 13.10\ mmHg$$

Mittelwert über alle Diastol-Werte (m) und Standardabweichung (s)

```
> datay:=array([seq(data[i][2],i=1..22)]);
```

$datay := [84, 88, 71, 72, 80, 69, 55, 74, 69, 78, 84, 61, 83, 70, 60, 70, 67, 77, 76, 78, 73, 77]$

```
> m:=evalf(Mean(datay),4)*mmHg;
```

$$m := 73.45\ mmHg$$

```
> s:=evalf(StandardDeviation(datay),4)*mmHg;
```

$$s := 8.256\ mmHg$$

Mittelwert über alle Pulsdruck-Werte (pd) und Standardabweichung (pds)

```
> PDy:=array([seq(PD[i][2],i=1..22)]);
```

$PDy := [56, 35, 41, 39, 30, 38, 41, 37, 41, 50, 47, 34, 38, 43, 40, 41, 47, 49, 55, 45, 63, 60]$

```
> pd:=evalf(Mean(PDy),4)*mmHg;
```

$$pd := 44.09\ mmHg$$

```
> pds:=evalf(StandardDeviation(PDy),4)*mmHg;
```

$$pds := 8.557\ mmHg$$

Mittelwert über alle Herzfrequenz-Werte (hf) und Standardabweichung (hfs)

```
> Hfy:=array([seq(Hf[i][2],i=1..22)]);
```

$$Hfy := [75, 82, 62, 63, 62, 60, 57, 54, 51, 45, 54, 54, 66, 55, 47, 50, 44, 58, 43, 45, 47, 41]$$

```
> hf:=evalf(Mean(Hfy),4)/min;
```

$$hf := \frac{55.23}{min}$$

```
> hfs:=evalf(StandardDeviation(Hfy),4)/min;
```

$$hfs := \frac{10.42}{min}$$

Trennung von Tag- und Nachtwerten

```
> Systole[Tag]:=array([seq(DATA[i][2],i=1..14)]);
```

$$Systole_{Tag} := [140, 123, 112, 111, 110, 107, 96, 111, 110, 128, 131, 95, 121, 113]$$

```
> M[Systole][Tag]:=evalf(Mean(%),4)*mmHg;
```

$$M_{Systole_{Tag}} := 114.9\ mmHg$$

```
> Systole[Nacht]:=array([seq(DATA[i][2],i=14..22)]);
```

$$Systole_{Nacht} := [113, 100, 112, 114, 126, 131, 128, 136, 137]$$

```
> M[Systole][Nacht]:=evalf(Mean(%),4)*mmHg;
```

$$M_{Systole_{Nacht}} := 121.9\ mmHg$$

```
> Diastole[Tag]:=array([seq(data[i][2],i=1..14)]);
```

$$Diastole_{Tag} := [84, 88, 71, 72, 80, 69, 55, 74, 69, 78, 84, 61, 83, 70]$$

```
> m[Diastole][Tag]:=evalf(Mean(%),4)*mmHg;
```

$$m_{Diastole_{Tag}} := 74.14\ mmHg$$

```
> Diastole[Nacht]:=array([seq(data[i][2],i=14..22)]);
```

$$Diastole_{Nacht} := [70, 60, 70, 67, 77, 76, 78, 73, 77]$$

```
> m[Distole][Nacht]:=evalf(Mean(%),4)*mmHg;
```

$$m_{Distole_{Nacht}} := 72.\ mmHg$$

```
> Herzfrequenz[Tag]:=array([seq(Hf[i][2],i=1..14)]);
```

$$Herzfrequenz_{Tag} := [75, 82, 62, 63, 62, 60, 57, 54, 51, 45, 54, 54, 66, 55]$$

```
> hf[Tag]:=evalf(Mean(%),4)/min;
```

$$hf_{Tag} := \frac{60.}{min}$$

```
> Herzfrequenz[Nacht]:=array([seq(Hf[i][2],i=14..22)]);
```

$$Herzfrequenz_{Nacht} := [55, 47, 50, 44, 58, 43, 45, 47, 41]$$

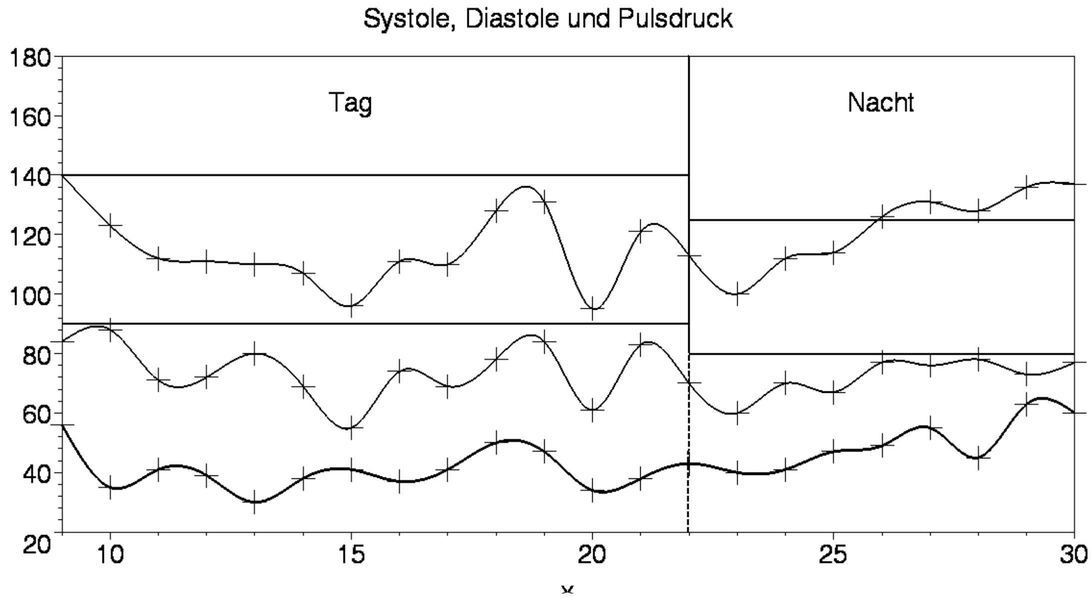
```
> hf[Nacht]:=evalf(Mean(%),4)/min;
```

$$hf_{Nacht} := \frac{47.78}{min}$$

III) Graphische Darstellung der Messwerte

Im folgenden Bild sind die gemessenen *Systole*-, *Diastole*- und *Pulsdruck-Werte* dargestellt und durch *kubische Splinefunktionen* interpoliert. Im Gegensatz dazu werden in der Langzeitmessung die Daten linear interpoliert. Die *kubischen Splinefunktionen* sind gegeben durch:

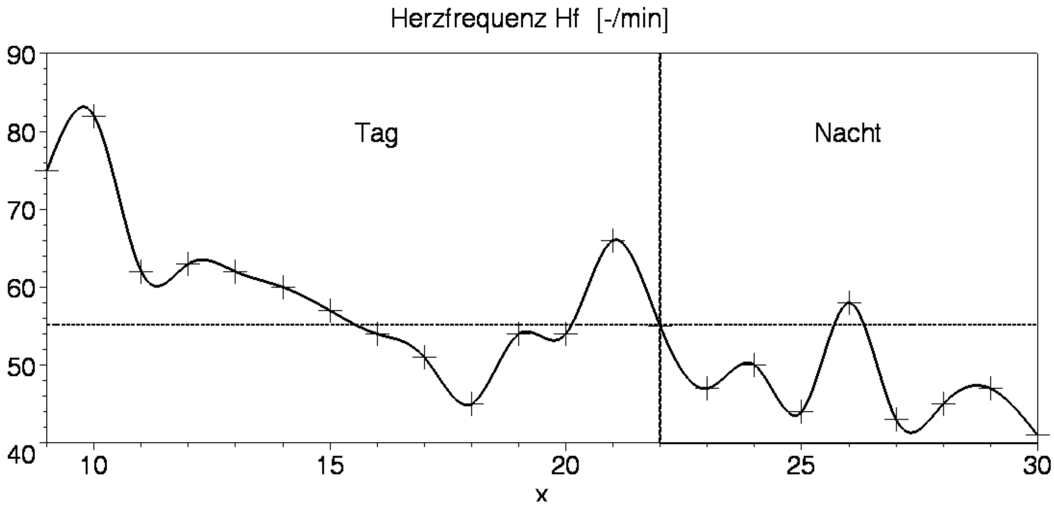
```
> Sp(x):=Spline([DATA],x,degree=3):
> sp(x):=Spline([data],x,degree=3):
> spd(x):=Spline([PD],x,degree=3):
> # Aus Platzgründen sind diese Funktionen nicht ausgedruckt. Man
  kann sie jedoch ausdrucken, wenn man die Doppelpunkte (:) am
  Ende der entsprechenden Befehle durch Semikola (;) ersetzt. Die
  Ergebnisse der Spline-Interpolation sind im folgen Bild
  dargestellt.
> with(CurveFitting):
> alias(H=Heaviside, th=thickness,co=color):
> p[1]:=plot({Sp(x),sp(x)},x=9..30,20..180,th=2,co=black):
> p[2]:=plot(spd(x),x=9..30,axes=boxed,th=3,co=black):
> p[3]:=plot({[DATA],[data],[PD]},x=9..30,co=black,
  style=point,symbol=cross, symbolsize=50,ytickmarks=5):
> p[4]:=plot({140,90},x=9..22,linestyle=4,th=2,co=black):
> p[5]:=plot({125,80},x=22..30,linestyle=4,th=2,co=black):
> p[6]:=plot(180*H(x-21.99)-100*H(x-22.001),x=9..30,linestyle=3,
  th=2,co=black, title="Systole, Diastole und Pulsdruck"):
> p[7]:=plots[textplot]({[15,165,`Tag`],[26,165,`Nacht`]}):
> plots[display]({seq(p[k],k=1..7)});
```



Gestrichelt eingezeichnet sind die Systole-Werte von 140 mmHg (Tag), 125 mmHg (Nacht) und die Diastole-Werte von
 90 mmHg (Tag), 80 mmHg (Nacht), die nicht überschritten werden sollten. Die Systole- (Diastole-) Kurve beginnt mit 140 (84) mmHg um 9 Uhr und endet mit 137 (77) mmHg um 6 Uhr (x =30). Die Puladruck-Werte ergeben sich aus der Differenz von Systole und Diastole mithin beginnt die Pulsdruck-Kurve um 9 Uhr mit 56 mmHg und endet um 6 Uhr bei einem Wert von 60 mmHg. Die Nachtruhe beginnt um 22 Uhr.
  Im nächsten Bild ist die Herzfrequenz [-/min] dargestellt.

```
> SpHf(x):=Spline([Hf],x,degree=3):
```

```
> alias(H=Heaviside,th=thickness,co=color):
> p[1]:=plot([Hf],x=9..30,co=black,
  style=point,symbol=cross,symbolsize =50,axes=boxed):
> p[2]:=plot(55.23,x=9..30,linestyle=4,th=2,co=black):
> p[3]:=plot(SpHf(x),x=9..30,40..90,
  th=3,co=black,title="Herzfrequenz Hf  [-/min]"):
> p[4]:=plot(90*H(x-21.98)-50*H(x-22.01),x=9..30,
  linestyle=3,th=2,co=black):
> p[5]:=plots[textplot]({[16,80,`Tag`],[26,80,`Nacht`]}):
> plots[display]({seq(p[k],k=1..5)});
```



Die gestrichelte Linie im obigen Bild stellt den Mittelwert hf = 55.23 /min dar. Die zwei Bilder zeigen, dass die *Splineinterpolation* glatte Kurven liefert. Aufgrund der Vielzahl der vorliegenden Daten ist eine *LAGRANGE Interpolation* weniger geeignet.
Die Bezeichnung *Spline* bedeutet *dünnes Brett.* Ein leicht biegsames Lineal wird an die Messpunkte gelegt.Somit kann eine glatte Kurve gezeichnet werden. Im *MAPLE Application Center* hat der Autor des vorliegenden Worksheets an Beispielen gezeigt, dass sich die *Spline Kurven* mit wachsendem Grad immer stärker an das "schwingende" *LAGRANGE Polynom* anschmiegen. Im Grenzfall degree ---> infinity sind das *LAGRANGE Polynom* und die *Spline Kurve* deckungsgleich.

IV) Näherung des Pulsdruck-Splines durch nichlineare Regression

Die Langzeitmessung beginnt um 9 Uhr und endet am nächsten Morgen um 6 Uhr. Somit erhalten wir 22 Messwerte innerhalb der Messperiode von 21 Stunden. Zur Erzeugung von *nichtlinearen Regressionen* ist es zweckmäßig, die x-Koordinaten [9...30] auf die Abszisse [0...2*Pi] gemäß X = 2*Pi*(i-1)/21 für i = 1..22 abzubilden.

```
> restart:
> with(Statistics):
> X:=array([seq(2*Pi*(i-1)/21,i=1..22)]);
```

$$X := \left[0, \frac{2\pi}{21}, \frac{4\pi}{21}, \frac{2\pi}{7}, \frac{8\pi}{21}, \frac{10\pi}{21}, \frac{4\pi}{7}, \frac{2\pi}{3}, \frac{16\pi}{21}, \frac{6\pi}{7}, \frac{20\pi}{21}, \frac{22\pi}{21}, \frac{8\pi}{7}, \frac{26\pi}{21}, \frac{4\pi}{3}, \frac{10\pi}{7}, \frac{32\pi}{21}, \frac{34\pi}{21}, \frac{12\pi}{7}, \frac{38\pi}{21}, \frac{40\pi}{21}, 2\pi\right]$$

```
> whattype(X);
```

symbol

```
> Y:=array([56,35,41,39,30,38,41,37,41,50,47,34,38,43,40,41,47,49,
  55,45,63,60]);
```

$$Y := [56, 35, 41, 39, 30, 38, 41, 37, 41, 50, 47, 34, 38, 43, 40, 41, 47, 49, 55, 45, 63, 60]$$

```
> # Y = PDy
> whattype(Y);
```

symbol

```
> # Mittelwert YM = pd [mmHg]
> YM:=evalf(Mean(Y),4)*mmHg;
```

$$YM := 44.09\ mmHg$$

```
> # Data-Pulsdruck = PD [mmHg]
> PD:=seq([X[i],Y[i]],i=1..22);
```

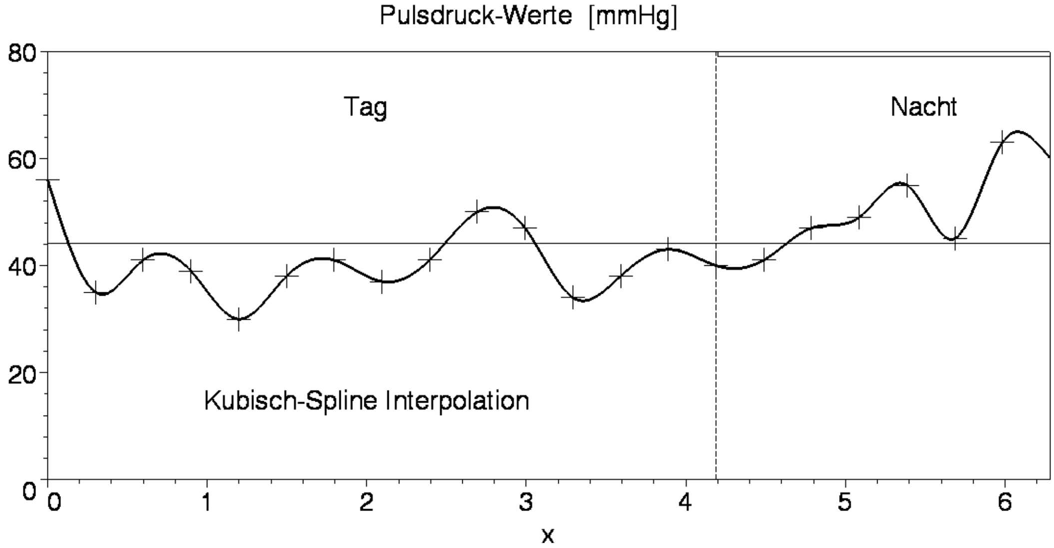
$$PD := [0, 56], \left[\frac{2\pi}{21}, 35\right], \left[\frac{4\pi}{21}, 41\right], \left[\frac{2\pi}{7}, 39\right], \left[\frac{8\pi}{21}, 30\right], \left[\frac{10\pi}{21}, 38\right], \left[\frac{4\pi}{7}, 41\right], \left[\frac{2\pi}{3}, 37\right], \left[\frac{16\pi}{21}, 41\right], \left[\frac{6\pi}{7}, 50\right], \left[\frac{20\pi}{21}, 47\right], \left[\frac{22\pi}{21}, 34\right], \left[\frac{8\pi}{7}, 38\right], \left[\frac{26\pi}{21}, 43\right], \left[\frac{4\pi}{3}, 40\right], \left[\frac{10\pi}{7}, 41\right], \left[\frac{32\pi}{21}, 47\right], \left[\frac{34\pi}{21}, 49\right], \left[\frac{12\pi}{7}, 55\right], \left[\frac{38\pi}{21}, 45\right], \left[\frac{40\pi}{21}, 63\right], [2\pi, 60]$$

```
> whattype(PD);
```

exprseq

```
> # grafische Darstellung des Pulsdrucks [mmHg]
> with(CurveFitting):
> spd(x):=Spline([PD],x,degree=3):
> alias(H=Heaviside,th=thickness,co=color):
> p[1]:=plot([PD],x=0..2*Pi,0..80,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,axes=boxed,
  title="Pulsdruck-Werte  [mmHg]"):
> p[2]:=plot(spd(x),x=0..2*Pi,th=3,co=black):
> p[3]:=plots[textplot]({[2,70,`Tag`],[5.5,70,`Nacht`],
```

```
[2,15,`Kubisch-Spline Interpolation`]}):
> p[4]:=plot(44.09,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plot(80*H(x-4.189)-H(x-4.199),x=0..2*Pi,
  linestyle=3,th=1,co=black,ytickmarks=4):
> plots[display](seq(p[k],k=1..5));
```



In diesem Bild sind die 22 Pulsdruck-Werte (+++) durch einen *kubischen Spline* interpoliert. Die vertikale Linie bei x = 26*Pi/21 = 22 Uhr teilt die Periode [0..2*Pi] in Tag und Nachtruhe. Die horizontale Linie zeigt den Mittelwert pd = 44.09 mmHg an. Bei *normalen* Blutdruckwerten von 120 / 80 ergibt sich ein Pulsdruck-Wert von 40 mmHg.
Im Folgenden wird obige *kubisch Spline Interpolation* durch eine ***nichtlineare Regression*** angenähert.

```
> f:=(t,a,b,c,d,e,g)-> a+b*t*sin(c*t+d)*cos(e*t+g)^2;
```

$$f := (t, a, b, c, d, e, g) \to a + b\, t \sin(c\, t + d) \cos(e\, t + g)^2$$

```
> y:=unapply(evalf(NonlinearFit(f(t,a,b,c,d,e,g),X,Y,t),4),t);
```

$$y := t \to 42.79 + 9.826\, t \sin(0.5155\, t - 14.30) \cos(0.2576\, t + 6.838)^2$$

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

$$\mathrm{y}(x) := 42.79 + 9.826\, x \sin(0.5155\, x - 14.30) \cos(0.2576\, x + 6.838)^2$$

```
> y(0):=evalf(subs(x=0,y(x)))*mmHg;
```

$$\mathrm{y}(0) := 42.79\, mmHg$$

```
> y(0)[data]:=56*mmHg;
```

$$\mathrm{y}(0)_{data} := 56\, mmHg$$

```
> y(2*Pi):=evalf(subs(x=2*Pi,y(x)))*mmHg;
```

$$\mathrm{y}(2\,\pi) := 62.57896379\, mmHg$$

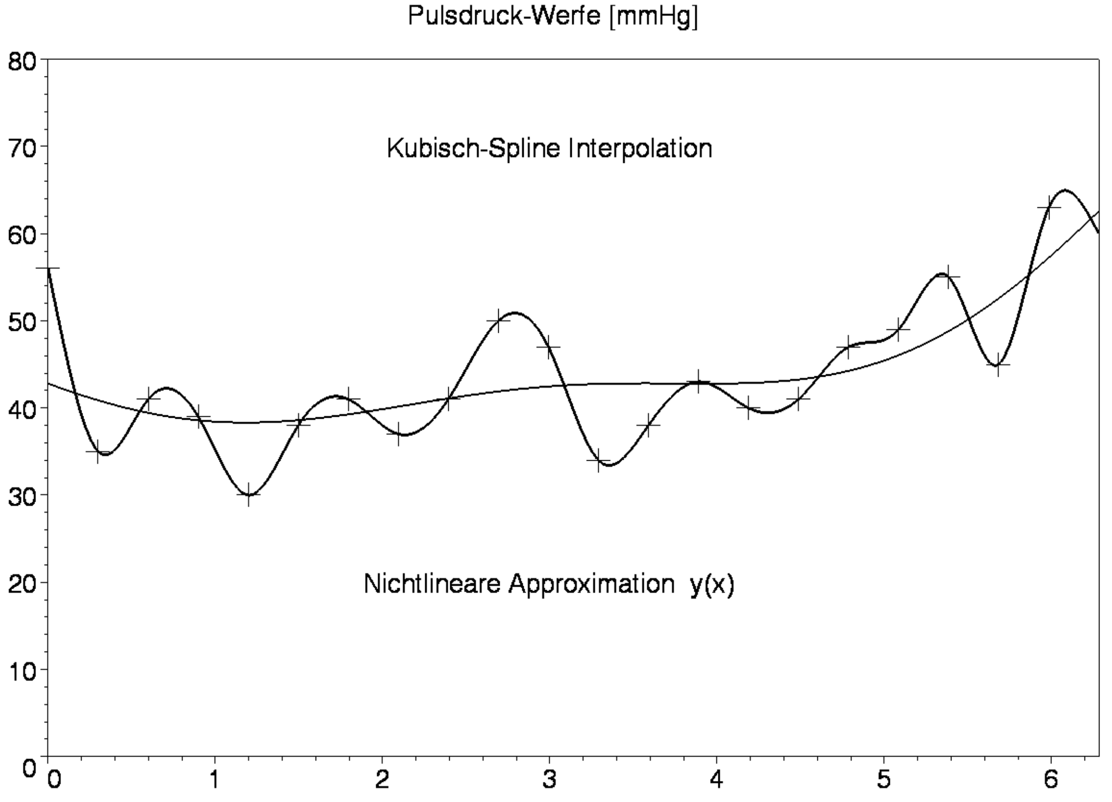
```
> y(2*Pi)[data]:=60*mmHg;
```

$$\mathrm{y}(2\,\pi)_{data} := 60\, mmHg$$

Die **nichlineare Regession y(x) ist im nächste Bild dargestellt.**

```
> alias(th=thickness,co=color):
```

```
> p[1]:=plot([PD],x=0..2*Pi,0..80,th=3,co=black,
  style=point,symbol=cross,symbolsize =50,axes=boxed,
  title="Pulsdruck-Werfe [mmHg]"):
> p[2]:=plot(spd(x),x=0..2*Pi,th=3,co=black):
> p[3]:=plot(y(x),x=0..2*Pi,th=2,co=black):
> p[4]:=plots[textplot]({[3,70,`Kubisch-Spline Interpolation`],
  [3,20,`Nichtlineare Approximation  y(x)`]}):
> plots[display](seq(p[k],k=1..4));
```



Die Güte der *nichtlinearen Approximation* y(x) bezüglich der *Kubisch-Spline Interpolation* und der *Messdaten* [PD] kann mit Hilfe der *error norms* L[2] und l[2] überprüft werden.

```
> L[2]:=
  sqrt((1/(2*Pi))*Int((SPLINE - APPROX)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((spd(x)-y(x))^2,x=0.3..2*Pi))/44.09,4)
  ;
```

$$L_2 := \frac{1}{2}\sqrt{2}\,\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - APPROX)^2\,dx} = 0.09902$$

```
> with(linalg):
> for i from 1 to 22 do
  v[i]:=evalf(subs(x=PD[i][1],y(x))-PD[i][2],4) od:
> V:=vector([seq(v[i],i=1..22)]):
> l[2]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(22))*norm(V,2)/44.09,4);
```

$$l_2 := \frac{\mathrm{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.1227$$

Obige *Fehlernormen* zeigen, dass die *nichtlineare Regression* y(x) geeignet ist, die gegebenen *Pulsdruck-Daten* zu approximieren.

V) Mittlerer artieller Blutdruck MAD [mmHg]

MAD-Werte sind zur Beurteilung von Organdurchblutungen von größter Bedeutung. Man erhält diese Werte näherungsweise als arithmetischen Mittelwert aus den Systole- und Diastole-Werten plus Diastole. Genauere Werte ermittelt man nach folgender Formel:

```
> restart:
> MAD:=Diastole+Pulsdruck/3;
```

$$MAD := Diastole + \frac{Pulsdruck}{3}$$

Beispielweise bei normalem Blutdruck von 120 /80 erhält man näherungsweise: 80 + 40/2 = 100, während nach obiger Formel ein niedrigerer Wert von MAD = 80 + 40/3 = 93.333... folgt.

```
> restart:
> with(Statistics):
> X:=array([seq(2*Pi*(i-1)/21,i=1..22)]);
```

$$X := \left[0, \frac{2\pi}{21}, \frac{4\pi}{21}, \frac{2\pi}{7}, \frac{8\pi}{21}, \frac{10\pi}{21}, \frac{4\pi}{7}, \frac{2\pi}{3}, \frac{16\pi}{21}, \frac{6\pi}{7}, \frac{20\pi}{21}, \frac{22\pi}{21}, \frac{8\pi}{7}, \frac{26\pi}{21}, \frac{4\pi}{3}, \frac{10\pi}{7}, \frac{32\pi}{21}, \frac{34\pi}{21}, \frac{12\pi}{7}, \frac{38\pi}{21}, \frac{40\pi}{21}, 2\pi \right]$$

```
> whattype(X);
```

$$symbol$$

```
> Y:=
  array([103,100,85,89,90,82,74,87,83,95,100,73,96,85,73,84,83,93,
  97,95,94,97]);
```

$$Y := [103, 100, 85, 89, 90, 82, 74, 87, 83, 95, 100, 73, 96, 85, 73, 84, 83, 93, 97, 95, 94, 97]$$

```
> whattype(Y);
```

$$symbol$$

```
> # Mittelwert YM = mad [mmHg]
> YM:=evalf(Mean(Y),4)*mmHg;
```

$$YM := 89.\ mmHg$$

```
> # Data-MAD
> DATA_MAD:=seq([X[i],Y[i]],i=1..22);
```

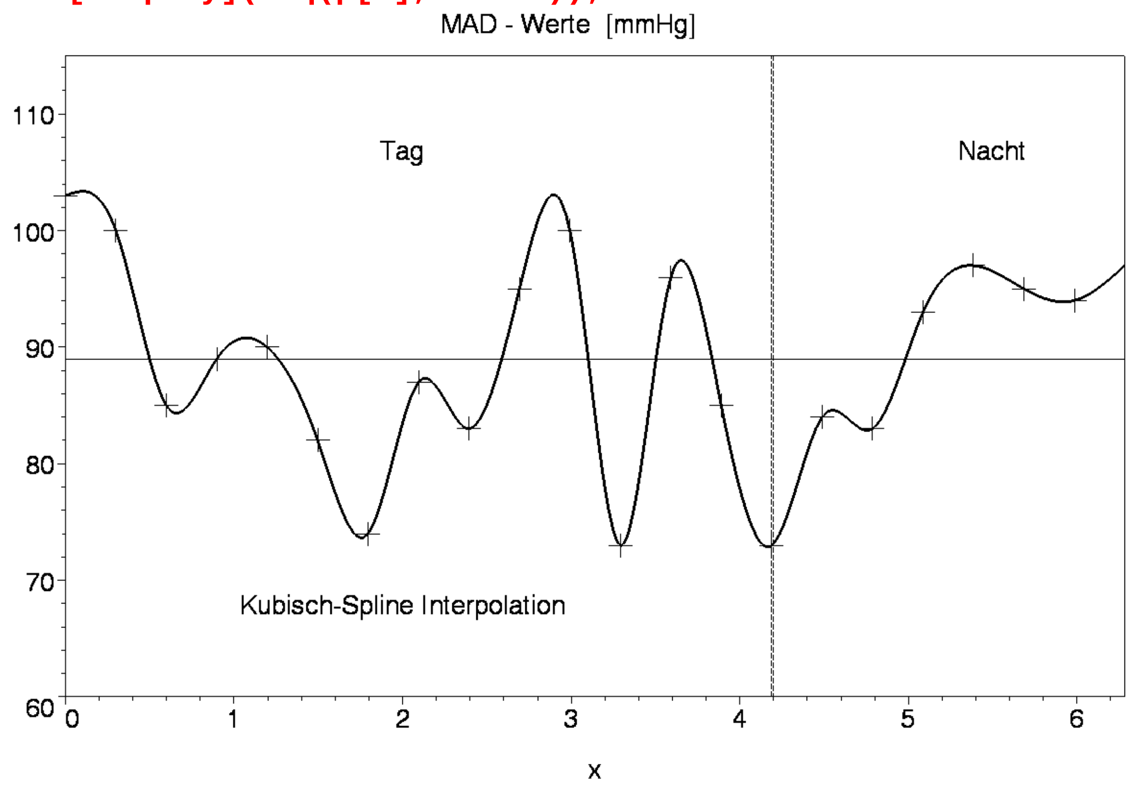
$$DATA\_MAD := [0, 103], \left[\frac{2\pi}{21}, 100\right], \left[\frac{4\pi}{21}, 85\right], \left[\frac{2\pi}{7}, 89\right], \left[\frac{8\pi}{21}, 90\right], \left[\frac{10\pi}{21}, 82\right],$$

$$\left[\frac{4\,\pi}{7}, 74\right], \left[\frac{2\,\pi}{3}, 87\right], \left[\frac{16\,\pi}{21}, 83\right], \left[\frac{6\,\pi}{7}, 95\right], \left[\frac{20\,\pi}{21}, 100\right], \left[\frac{22\,\pi}{21}, 73\right], \left[\frac{8\,\pi}{7}, 96\right],$$
$$\left[\frac{26\,\pi}{21}, 85\right], \left[\frac{4\,\pi}{3}, 73\right], \left[\frac{10\,\pi}{7}, 84\right], \left[\frac{32\,\pi}{21}, 83\right], \left[\frac{34\,\pi}{21}, 93\right], \left[\frac{12\,\pi}{7}, 97\right], \left[\frac{38\,\pi}{21}, 95\right],$$
$$\left[\frac{40\,\pi}{21}, 94\right], [2\,\pi, 97]$$

```
> whattype(DATA_MAD);
```

exprseq

```
> # grafische Darstellung von MAD [mmHg]
> with(CurveFitting):
> sp_mad(x):=Spline([DATA_MAD],x,degree=3):
> alias(H=Heaviside,th=thickness,co=color):
> p[1]:=plot([DATA_MAD],x=0..2*Pi,60..115,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,axes=boxed,
  title="MAD - Werte  [mmHg]"):
> p[2]:=plot(sp_mad(x),x=0..2*Pi,th=3,co=black):
> p[3]:=plots[textplot]({[2,107,`Tag`],[5.5,107,`Nacht`],
  [2,68,`Kubisch-Spline Interpolation`]}):
> p[4]:=plot(89,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plot(115*H(x-4.189)-60*H(x-4.199),x=0..2*Pi,
  linestyle=3,th=1,co=black,ytickmarks=4):
> plots[display](seq(p[k],k=1..5));
```



Die horizontale Linie kennzeichnet den Mittelwert YM = mad = 89 [mmHg]. Der *mittlere artielle Druck* **(MAD)** sollte zwischen **70** und **105** [mmHg]. liegen. Die experimentell ermittelten MAD-Werte liegen in diesen Grenzen, wie man im obigen Bild erkennen kann. Die

Kubisch-Spline Interpolation wird im Folgenden durch eine ***nichtlineare Regression*** angenährt.

```
> f:=(t,a,b,c,d,g,h)->a+b*t*sin(c*t+d)*cos(g*t+h)^2;
```

$$f := (t, a, b, c, d, g, h) \to a + b\, t \sin(c\, t + d) \cos(g\, t + h)^2$$

```
> y:=unapply(evalf(NonlinearFit(f(t,a,b,c,d,g,h),X,Y,t),4),t);
```

$$y := t \to 97.60 + 17.60\, t \sin(0.07667\, t + 5.805) \cos(0.01287\, t - 6.340)^2$$

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

$$\mathrm{y}(x) := 97.60 + 17.60\, x \sin(0.07667\, x + 5.805) \cos(0.01287\, x - 6.340)^2$$

```
> y(0):=evalf(subs(x=0,y(x)),4)*mmHg;
```

$$\mathrm{y}(0) := 97.60\ mmHg$$

```
> y(0)[data]:=103*mmHg;
```

$$\mathrm{y}(0)_{data} := 103\ mmHg$$

```
> y(2*Pi):=evalf(subs(x=2*Pi,y(x)),4)*mmHg;
```

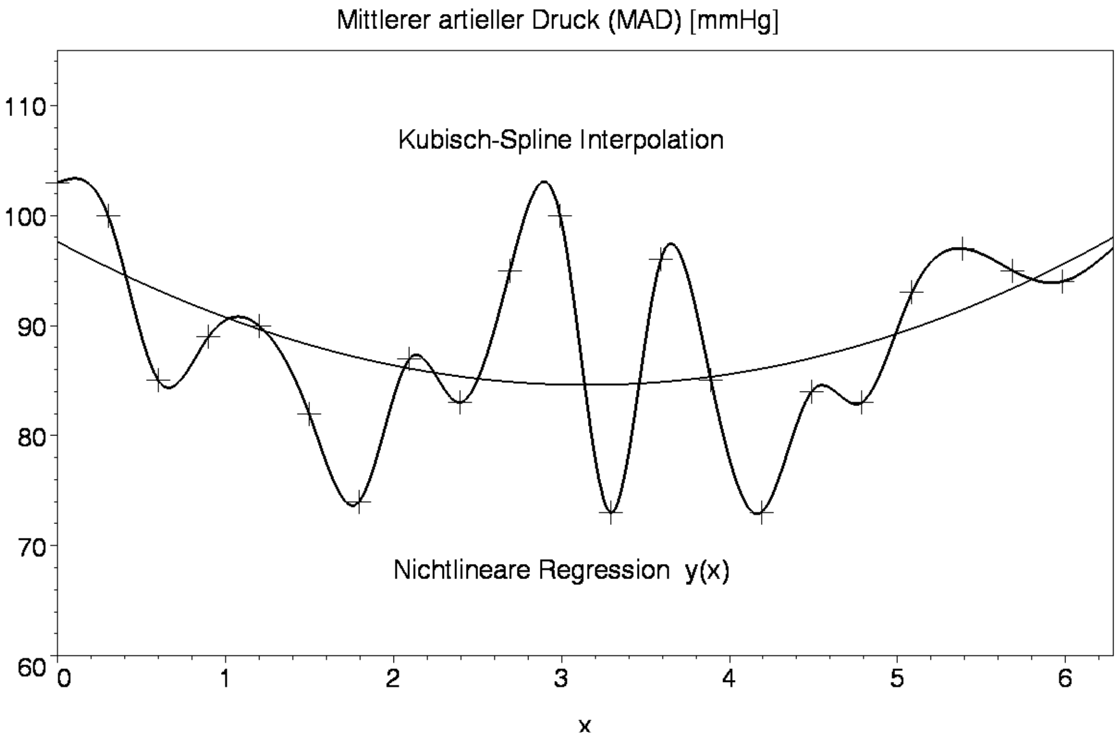
$$\mathrm{y}(2\,\pi) := 98.02\ mmHg$$

```
> y(2*Pi)[data]:=97*mmHg;
```

$$\mathrm{y}(2\,\pi)_{data} := 97\ mmHg$$

Die **nichtlineare Regression** **y(x)** ist im nächstenBild dargestellt.

```
> alias(th=thickness,co=color):
> p[1]:=plot([DATA_MAD],x=0..2*Pi,60..115,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,axes=boxed,
  title="Mittlerer artieller Druck (MAD) [mmHg]"):
> p[2]:=plot(sp_mad(x),x=0..2*Pi,th=3,co=black):
> p[3]:=plot(y(x),x=0..2*Pi,th=2,co=black):
> p[4]:=plots[textplot]({[3,107,`Kubisch-Spline Interpolation`],
  [3,68,`Nichtlineare Regression  y(x)`]}):
> plots[display](seq(p[k],k=1..4));
```



Die Güte der *nichtlinearen Regression* y(x) bezüglich der *Kubisch-Spline Interpolation* und der *Messdaten* [DATA_MAD] kann mit Hilfe der *error norms* L[2] und l[2] überprüft werden.

```
> L[2]:=
  sqrt((1/(2*Pi))*Int((SPLINE-Regression)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((sp_mad(x)-y(x))^2,x=0.3..2*Pi))/89,4)
  ;
```

$$L_2 := \frac{1}{2}\sqrt{2}\,\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - Regression)^2\,dx} = 0.07802$$

```
> with(linalg):
> for i from 1 to 22 do
  v[i]:=evalf(subs(x=DATA_MAD[i][1],y(x))-DATA_MAD[i][2],4) od:
> V:=vector([seq(v[i],i=1..22)]):
> l[2]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(22))*norm(V,2)/89,4);
```

$$l_2 := \frac{\mathrm{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.08278$$

Obige *error norms* zeigen, dass die Approximation y(x) zufriedenstellend ist. Im folgenden Abschnitt wir die *kubische Splinefunktion* sp_mad(x) des *mittleren artiellen Blutdrucks* MAD durch *FOURIER-Reihen* angenähert.

VI) Approximation der Splinefunktion des MAD durch FOURIER-Reihen

```
> restart: with(Statistics):
```

```
> X:=array([seq(2*Pi*(i-1)/21,i=1..22)]);
```

$$X := \left[0, \frac{2\pi}{21}, \frac{4\pi}{21}, \frac{2\pi}{7}, \frac{8\pi}{21}, \frac{10\pi}{21}, \frac{4\pi}{7}, \frac{2\pi}{3}, \frac{16\pi}{21}, \frac{6\pi}{7}, \frac{20\pi}{21}, \frac{22\pi}{21}, \frac{8\pi}{7}, \frac{26\pi}{21}, \frac{4\pi}{3}, \frac{10\pi}{7}, \frac{32\pi}{21}, \frac{34\pi}{21}, \frac{12\pi}{7}, \frac{38\pi}{21}, \frac{40\pi}{21}, 2\pi\right]$$

```
> whattype(X);
```

symbol

```
> Y:=array([103,100,85,89,90,92,74,87,83,95,100,73,96,85,73,84,83,
93,97,95,94,97]);
```

$$Y := [103, 100, 85, 89, 90, 92, 74, 87, 83, 95, 100, 73, 96, 85, 73, 84, 83, 93, 97, 95, 94, 97]$$

```
> whattype(Y);
```

symbol

```
> # Mittelwert YM = mad [mmHg]
> YM:=evalf(Mean(Y),4)*mmHg;
```

$$YM := 89.45\ mmHg$$

```
> DATA_MAD:=seq([X[i],Y[i]],i=1..22);
```

$$DATA\_MAD := [0, 103], \left[\frac{2\pi}{21}, 100\right], \left[\frac{4\pi}{21}, 85\right], \left[\frac{2\pi}{7}, 89\right], \left[\frac{8\pi}{21}, 90\right], \left[\frac{10\pi}{21}, 92\right], \left[\frac{4\pi}{7}, 74\right], \left[\frac{2\pi}{3}, 87\right], \left[\frac{16\pi}{21}, 83\right], \left[\frac{6\pi}{7}, 95\right], \left[\frac{20\pi}{21}, 100\right], \left[\frac{22\pi}{21}, 73\right], \left[\frac{8\pi}{7}, 96\right], \left[\frac{26\pi}{21}, 85\right], \left[\frac{4\pi}{3}, 73\right], \left[\frac{10\pi}{7}, 84\right], \left[\frac{32\pi}{21}, 83\right], \left[\frac{34\pi}{21}, 93\right], \left[\frac{12\pi}{7}, 97\right], \left[\frac{38\pi}{21}, 95\right], \left[\frac{40\pi}{21}, 94\right], [2\pi, 97]$$

```
> whattype(DATA_MAD);
```

exprseq

```
> with(CurveFitting):
> # Die kubische Splinefunktion ist gegeben durch:
> sp_mad(x):=Spline([DATA_MAD],x,degree=3):
> FOURIER_Reihe(x):=
a[0]/2+sum(a[k]*cos(k*x)+b[k]*sin(k*x),k=1..infinity);
```

$$\mathrm{FOURIER\_Reihe}(x) := \frac{1}{2} a_0 + \left(\sum_{k=1}^{\infty} (a_k \cos(k\,x) + b_k \sin(k\,x))\right)$$

```
> a[k]:=(1/Pi)*Int(f(x)*cos(k*x),x=0..2*Pi);
```

$$a_k := \frac{1}{\pi} \int_0^{2\pi} \mathrm{f}(x) \cos(k\,x)\,dx$$

```
> a[0]:=simplify(subs(k=0,a[k]));
```

$$a_0 := \frac{1}{\pi} \int_0^{2\pi} \mathrm{f}(x)\,dx$$

```
> b[k]:=(1/Pi)*Int(f(x)*sin(k*x),x=0..2*Pi);
```

$$b_k := \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(k\,x)\, dx$$

```
> with(CurveFitting): F(x):=sp_mad(x):
> A[0]:=evalf(subs(f(x)=F(x),a[0]));
```

$$A_0 := 177.8868767$$

```
> A[k]:=simplify(value(subs(f(x)=F(x),a[k]))):
> A[k]:=subs({sin(k*Pi)=0.,(cos(k*Pi))^2=1.},%):
> A[k]:=evalf(%):
> for i in [seq(k,k=1..5)] do A[i]:=subs(k=i,A[k]) od:
> for i in [seq(k,k=1..15)] do P[i]:=subs(k=i,A[k]) od:
> for i in [seq(k,k=1..20)] do R[i]:=subs(k=i,A[k]) od:
> B[k]:=simplify(value(subs(f(x)=F(x),b[k]))):
> B[k]:=subs({sin(k*Pi)=0.,(cos(k*Pi))^2=1.},%):
> B[k]:=evalf(%):
> for i in [seq(k,k=1..5)] do B[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..15)] do Q[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..20)] do S[i]:=subs(k=i,B[k]) od:
> # Die Koeffizienten A[i]...S[i] sind aus Platzgründen nicht
  ausgedrucht. Man kann sie jedoch ausdrucken, wenn man am Ende
  der Befehle die Doppelpunkte (:) durch Semikola (;) ersetzt. Die
  FOURIER-Reihen mit k = 5, 15, 20 werden im Folgenden durch y(x),
  z(x), T(x) ausgedrückt.
>
> y(x):=evalf(A[0]/2+sum(A[k]*cos(k*x)+B[k]*sin(k*x),k=1..5));
```

$$\begin{aligned} y(x) := {} & 88.94343835 + 4.605301525\cos(x) + 0.07789561682\sin(x) + 4.940105876\cos(2.\,x) \\ & - 2.579701082\sin(2.\,x) - 2.454419353\cos(3.\,x) + 0.1383960637\sin(3.\,x) \\ & + 0.4626709302\cos(4.\,x) - 0.6075191809\sin(4.\,x) + 3.996710763\cos(5.\,x) \\ & + 2.318502186\sin(5.\,x) \end{aligned}$$

```
> y(0):=evalf(simplify(subs(x=0,y(x))))*mmHg;
```

$$y(0) := 100.4938081\,mmHg$$

```
> y(0)[data]:=103*mmHg;
```

$$y(0)_{data} := 103\,mmHg$$

```
> y(2*Pi):=evalf(simplify(subs(x=2*Pi,y(x))))*mmHg;
```

$$y(2\,\pi) := 100.4938081\,mmHg$$

```
> y(2*Pi)[data]:=95*mmHg;
```

$$y(2\,\pi)_{data} := 95\,mmHg$$

```
> z(x):=evalf(A[0]/2+sum(P[k]*cos(k*x)+Q[k]*sin(k*x),k=1..15));
```

$$\begin{aligned} z(x) := {} & -0.1433531509\cos(13.\,x) - 1.061045630\cos(8.\,x) - 3.560687357\sin(8.\,x) \\ & + 3.744934468\sin(7.\,x) + 1.630242665\cos(7.\,x) + 0.5712651711\sin(6.\,x) \\ & - 0.8013394963\sin(12.\,x) + 0.1026612011\sin(11.\,x) + 0.2591215675\cos(12.\,x) \\ & - 0.5419554867\cos(11.\,x) + 0.2948819430\sin(10.\,x) - 0.7985116891\cos(10.\,x) \\ & + 3.247843343\sin(9.\,x) + 0.1208482973\sin(15.\,x) - 0.9169382052\cos(6.\,x) \end{aligned}$$

$+ 0.7959311515 \cos(9.\, x) - 0.08058756026 \sin(14.\, x) - 0.01452308321 \cos(15.\, x)$
$+ 0.6917959522 \sin(13.\, x) + 0.1110641427 \cos(14.\, x) + 4.6053014 \cos(x)$
$+ 0.0778967 \sin(x) + 2.318502187 \sin(5.\, x) + 3.996710763 \cos(5.\, x)$
$- 0.607519187 \sin(4.\, x) + 0.138395992 \sin(3.\, x) + 0.462670926 \cos(4.\, x)$
$- 2.57970093 \sin(2.\, x) - 2.45441932 \cos(3.\, x) + 4.94010604 \cos(2.\, x) + 88.94343835$

```
> z(0):=evalf(simplify(subs(x=0,z(x))))*mmHg;
```

$$z(0) := 99.81384044\, mmHg$$

```
> z(0)[data]:=103*mmHg;
```

$$z(0)_{data} := 103\, mmHg$$

```
> z(2*Pi):=evalf(simplify(subs(x=2*Pi,z(x))))*mmHg;
```

$$z(2\,\pi) := 99.81384052\, mmHg$$

```
> z(2*Pi)[data]:=95*mmHg;
```

$$z(2\,\pi)_{data} := 95\, mmHg$$

```
>
> T(x):=evalf(A[0]/2+sum(R[k]*cos(k*x)+S[k]*sin(k*x),k=1..20)):
> # Aufgrund der Vielzahl von Reihengliedern ist T(x) nicht
  ausgedruckt.
> T(0):=simplify(subs(x=0,T(x)))*mmHg;
```

$$T(0) := 99.88814293\, mmHg$$

```
> T(0)[data]:=103*mmHg;
```

$$T(0)_{data} := 103\, mmHg$$

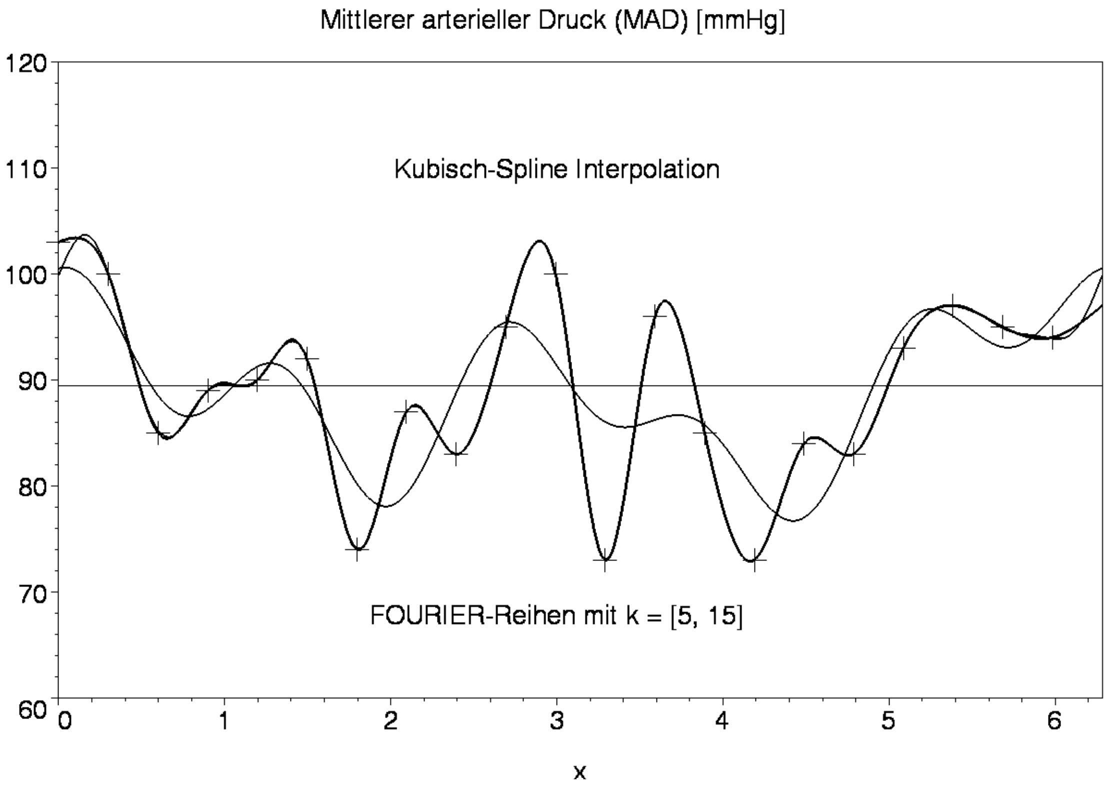
```
> T(2*Pi):=simplify(subs(x=2*Pi,T(x)))*mmHg;
```

$$T(2\,\pi) := 99.88814294\, mmHg$$

```
> T(2*Pi)[data]:=95*mmHg;
```

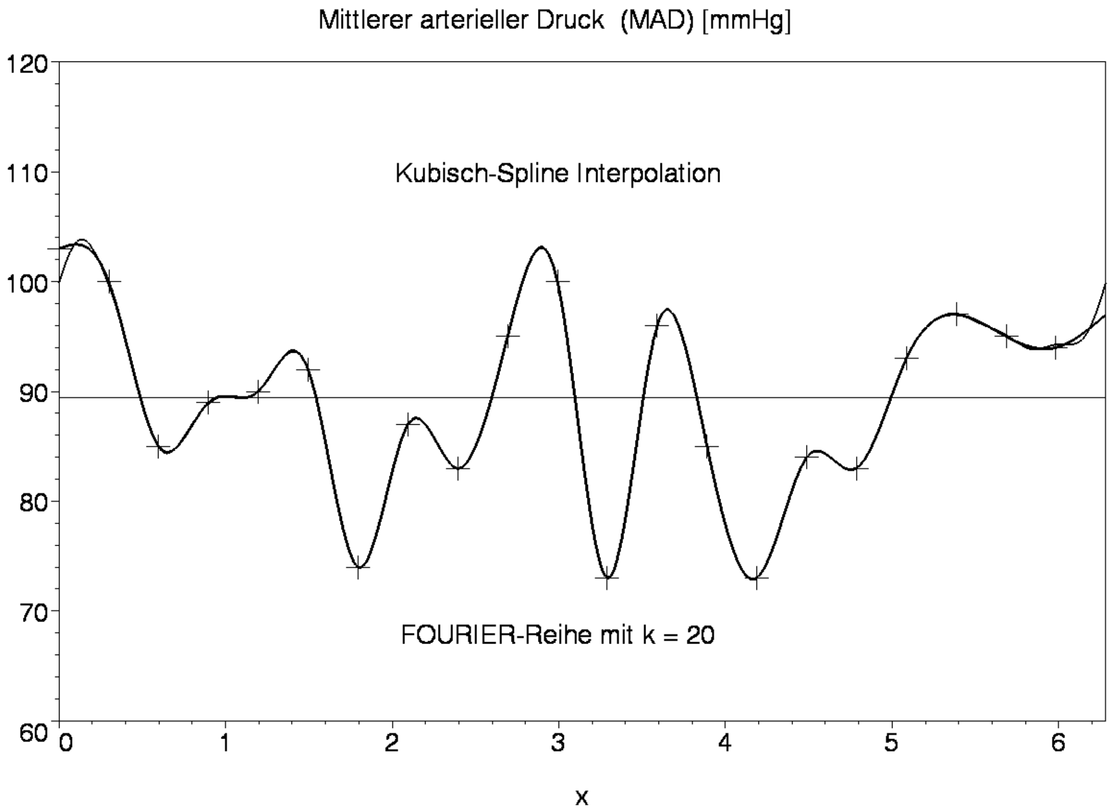
$$T(2\,\pi)_{data} := 95\, mmHg$$

```
> # Im nächsten Bild wird die Approximation der Splinefunktion des
  MAD durch die FOURIER-Reihen mit  k = 5 und 15 gezeigt.
> alias(th=thickness,co=color):
> p[1]:=plot(sp_mad(x),x=0..2*Pi,60..120,
  axes=boxed,th=3,co=black):
> p[2]:=plot([DATA_MAD],x=0..2*Pi,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,
  title="Mittlerer arterieller Druck (MAD) [mmHg]"):
> p[3]:=plot({y(x),z(x)},x=0..2*Pi,th=2,co=black):
> p[4]:=plot(89.45,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plots[textplot]({[3,110,`Kubisch-Spline Interpolation`],
  [3,68,`FOURIER-Reihen mit k = [5, 15]`]}):
> plots[display](seq(p[k],k=1..5));
```



Im Bild erkennt man, dass die FOURIER-Reihe mit fünf Gliedern (k = 5) breichsweise stark von der kubischen Spline Interpolation abweicht, während fünfzehn Glieder (k = 15) eine gute Näherung liefern. Sichtbar sind geringe Randeffekte, die auch aus obigen Werten für z(0) und z(2*Pi) deutlich werden.
Das nächste Bild zeigt den Vergleich zwischen der kubischen Spline Interpolation und einer FOURIER-Reihe mit zwanzig Gliedern (k = 20).

```
> alias(th=thickness,co=color):
> p[1]:=plot(sp_mad(x),x=0..2*Pi,60..120,axes=boxed,th=3,co=black)
  :
> p[2]:=plot([DATA_MAD],x=0..2*Pi,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,
  title="Mittlerer arterieller Druck  (MAD) [mmHg]"):
> p[3]:=plot(T(x),x=0..2*Pi,th=2,co=black):
> p[4]:=plot(89.45,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plots[textplot]({[3,110,`Kubisch-Spline
  Interpolation`],[3,68,`FOURIER-Reihe mit k = 20`]}):
> plots[display](seq(p[k],k=1..5));
```



Dieses Bild zeigt, dass sich die FOURIER-Reihe mit 20 Gliedern (k = 20) bis auf minimale Randstörungen bei x = 0 und x = 2*Pi deckungsgleich der Kubisch-Spline Interpolation anschliesst. Die Güte der Näherungen mit k = [5, 15, 20] kann durch die *Error Normen* L[2] und l[2] ausgedrückt werden.

```
> L[2][k=5]:=
  sqrt((1/(2*Pi))*Int((SPLINE-Ypsilon)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((sp_mad(x)-y(x))^2,x=0..2*Pi))/89.45);
```

$$L_{2_{k=5}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - Ypsilon)^2\,dx} = 0.05323128269$$

```
> L[2][k=15]:=
  sqrt((1/(2*Pi))*Int((SPLINE-ZET)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((sp_mad(x)-z(x))^2,x=0..2*Pi))/89.45);
```

$$L_{2_{k=15}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - ZET)^2\,dx} = 0.003855942826$$

```
> L[2][k=20]:=
  sqrt((1/(2*Pi))*Int((SPLINE-T)^2,x=0..2*Pi))=
  evalf(sqrt((1/(5.7))*int((sp_mad(x)-T(x))^2,x=0.3..6))/89.45);
```

$$L_{2_{k=20}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - T)^2\,dx} = 0.0008997536372$$

```
>
```

Obige Werte zeigen die Fehlernormen L[2] der *FOURIER-Reihen* y(x), z(x) und T(x) bezüglich der *kubischen Splinefunktion* sp_mad(x). Die Fehlernormen l[2] bezüglich der gegebenen Messdaten [DATA_MAD] ermittelt man folgendermassen:

```
> with(linalg):
> for i from 1 to 22 do
  u[i]:=subs(x=DATA_MAD[i][1],y(x))-DATA_MAD[i][2] od:
> U:=evalf(vector([seq(u[i],i=1..22)])):
> l[2][k=5]:=
  (1/sqrt(number_of_points))*Norm(U,2)=
  evalf((1/sqrt(22))*norm(U,2))/89.45;
```

$$l_{2_{k=5}} := \frac{\mathrm{Norm}(U,2)}{\sqrt{number\_of\_points}} = 0.06012266795$$

```
> for i from 1 to 22 do
  v[i]:=subs(x=DATA_MAD[i][1],z(x))-DATA_MAD[i][2] od:
> V:=evalf(vector([seq(v[i],i=1..22)])):
> l[2][k=15]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(22))*norm(V,2))/89.45;
```

$$l_{2_{k=15}} := \frac{\mathrm{Norm}(V,2)}{\sqrt{number\_of\_points}} = 0.01019414720$$

```
> for i from 2 to 21 do
  w[i]:=subs(x=DATA_MAD[i][1],T(x))-DATA_MAD[i][2] od:
> W:=evalf(vector([seq(w[i],i=2..21)])):
> l[2][k=20]:=
  (1/sqrt(number_of_points))*Norm(W,2)=
  evalf((1/sqrt(20))*norm(W,2))/89.45;
```

$$l_{2_{k=20}} := \frac{\mathrm{Norm}(\mathrm{W},2)}{\sqrt{number\_of\_points}} = 0.001402157935$$

Im nächsten Abschnitt wird die *kubische Splinefunktion* spd(x) des *Pulsdrucks* (PD) durch *FOURIER-Reihen* angenähert.

VII) Näherung der Pulsdruck-Splinefunktion PD durch FOURIER-Reihen

```
> restart:  with(Statistics):
> X:=array([seq(2*Pi*(i-1)/21,i=1..22)]);
```

$$X := \left[0, \frac{2\pi}{21}, \frac{4\pi}{21}, \frac{2\pi}{7}, \frac{8\pi}{21}, \frac{10\pi}{21}, \frac{4\pi}{7}, \frac{2\pi}{3}, \frac{16\pi}{21}, \frac{6\pi}{7}, \frac{20\pi}{21}, \frac{22\pi}{21}, \frac{8\pi}{7}, \frac{26\pi}{21}, \frac{4\pi}{3}, \frac{10\pi}{7}, \frac{32\pi}{21}, \frac{34\pi}{21}, \frac{12\pi}{7}, \frac{38\pi}{21}, \frac{40\pi}{21}, 2\pi \right]$$

```
> whattype(X);
```

$$symbol$$

```
> Y:=array([56,35,41,39,30,38,41,37,41,50,47,34,38,43,40,41,47,49,
```

```
55,45,63,60]);
```

$$Y := [56, 35, 41, 39, 30, 38, 41, 37, 41, 50, 47, 34, 38, 43, 40, 41, 47, 49, 55, 45, 63, 60]$$

```
> whattype(Y);
```

$$symbol$$

```
> # Mittelwert YM = pd [mmHg]
> YM:=evalf(Mean(Y),4)*mmHg;
```

$$YM := 44.09\ mmHg$$

```
> # Data-Pulsdruck = PD [mmHg]
> PD:=seq([X[i],Y[i]],i=1..22);
```

$$PD := [0, 56], \left[\frac{2\pi}{21}, 35\right], \left[\frac{4\pi}{21}, 41\right], \left[\frac{2\pi}{7}, 39\right], \left[\frac{8\pi}{21}, 30\right], \left[\frac{10\pi}{21}, 38\right], \left[\frac{4\pi}{7}, 41\right], \left[\frac{2\pi}{3}, 37\right], \left[\frac{16\pi}{21}, 41\right], \left[\frac{6\pi}{7}, 50\right], \left[\frac{20\pi}{21}, 47\right], \left[\frac{22\pi}{21}, 34\right], \left[\frac{8\pi}{7}, 38\right], \left[\frac{26\pi}{21}, 43\right], \left[\frac{4\pi}{3}, 40\right], \left[\frac{10\pi}{7}, 41\right], \left[\frac{32\pi}{21}, 47\right], \left[\frac{34\pi}{21}, 49\right], \left[\frac{12\pi}{7}, 55\right], \left[\frac{38\pi}{21}, 45\right], \left[\frac{40\pi}{21}, 63\right], [2\pi, 60]$$

```
> whattype(PD);
```

$$exprseq$$

```
> with(CurveFitting):
> # Die kubische Splinefunktion zur Interpolation der
  Pulsdruckdaten ist gegeben durch:
> spd(x):=Spline([PD],x,degree=3):
> FOURIER_Reihe(x):=
  a[0]/2+sum(a[k]*cos(k*x)+b[k]*sin(k*x),k=1..infinity);
```

$$\text{FOURIER\_Reihe}(x) := \frac{1}{2} a_0 + \left(\sum_{k=1}^{\infty} (a_k \cos(k\,x) + b_k \sin(k\,x)) \right)$$

```
> a[k]:=(1/Pi)*Int(f(x)*cos(k*x),x=0..2*Pi);
```

$$a_k := \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(k\,x)\,dx$$

```
> a[0]:=simplify(subs(k=0,a[k]));
```

$$a_0 := \frac{1}{\pi} \int_0^{2\pi} f(x)\,dx$$

```
> b[k]:=(1/Pi)*Int(f(x)*sin(k*x),x=0..2*Pi);
```

$$b_k := \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(k\,x)\,dx$$

```
> with(CurveFitting):  F(x):=spd(x):
> A[0]:=evalf(subs(f(x)=F(x),a[0]));
```

$$A_0 := 86.71737673$$

```
> A[k]:=simplify(value(subs(f(x)=F(x),a[k]))):
> A[k]:=subs({sin(k*Pi)=0.,(cos(k*Pi))^2=1.},%):
> A[k]:=evalf(%):
> for i in [seq(k,k=1..5)] do A[i]:=subs(k=i,A[k]) od:
```

```
> for i in [seq(k,k=1..10)] do P[i]:=subs(k=i,A[k]) od:
> for i in [seq(k,k=1..20)] do R[i]:=subs(k=i,A[k]) od:
> B[k]:=simplify(value(subs(f(x)=F(x),b[k]))):
> B[k]:=subs({sin(k*Pi)=0.,(cos(k*Pi))^2=1.},%):
> B[k]:=evalf(%):
> for i in [seq(k,k=1..5)] do B[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..10)] do Q[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..20)] do S[i]:=subs(k=i,B[k]) od:
> # Die Koeffizienten A[i]...S[i] sind aus Platzgründen nicht
  ausgedrückt. Man kann sie jedoch ausdrücken, wenn man am Ende
  der Befehle die Doppelpunkte (:) durch Semikola (;) ersetzt. Die
  FOURIER-Reihen mit k = 5, 10, 20 werden im Folgenden durch y(x),
  z(x), T(x) ausgedrückt.
>
> y(x):=evalf(A[0]/2+sum(A[k]*cos(k*x)+B[k]*sin(k*x),k=1..5));
```

$$y(x) := 43.35868836 + 3.557021015\cos(x) - 4.437235009\sin(x) + 3.065476944\cos(2.\,x) - 5.540187207\sin(2.\,x) + 1.105221979\cos(3.\,x) - 0.02351441881\sin(3.\,x) + 0.8134320229\cos(4.\,x) - 2.085813271\sin(4.\,x) + 1.588286623\cos(5.\,x) + 0.5369991021\sin(5.\,x)$$

```
> y(0):=simplify(subs(x=0,y(x)))*mmHg;
```

$$y(0) := 53.48812695\ mmHg$$

```
> y(0)[data]:=56*mmHg;
```

$$y(0)_{data} := 56\ mmHg$$

```
> y(2*Pi):=simplify(subs(x=2*Pi,y(x)))*mmHg;
```

$$y(2\,\pi) := 53.48812691\ mmHg$$

```
> y(2*Pi)[data]:=60*mmHg;
```

$$y(2\,\pi)_{data} := 60\ mmHg$$

```
>
> z(x):=evalf(A[0]/2+sum(P[k]*cos(k*x)+Q[k]*sin(k*x),k=1..10));
```

$$z(x) := 43.35868836 + 3.55701881\cos(x) - 4.4372348\sin(x) + 3.06547707\cos(2.\,x) - 5.54018728\sin(2.\,x) + 1.105221950\cos(3.\,x) - 0.02351441\sin(3.\,x) + 0.8134320252\cos(4.\,x) - 2.085813272\sin(4.\,x) + 1.588286617\cos(5.\,x) + 0.536999105\sin(5.\,x) - 0.02847930282\cos(6.\,x) - 4.610616098\sin(6.\,x) + 2.264895997\cos(7.\,x) - 1.776484233\sin(7.\,x) + 1.287849799\cos(8.\,x) - 2.120661457\sin(8.\,x) + 0.05089588340\cos(9.\,x) - 0.8741969263\sin(9.\,x) - 0.5431596143\cos(10.\,x) - 0.5292999433\sin(10.\,x)$$

```
> z(0):=simplify(subs(x=0,z(x)))*mmHg;
```

$$z(0) := 56.52012759\ mmHg$$

```
> z(0)[data]:=56*mmHg;
```

$$z(0)_{data} := 56\ mmHg$$

```
> z(2*Pi):=simplify(subs(x=2*Pi,z(x)))*mmHg;
```

$$z(2\,\pi) := 56.52012752\ mmHg$$

```
> z(2*Pi)[data]:=60*mmHg;
```

$$z(2\pi)_{data} := 60\ mmHg$$

```
>
> T(x):=evalf(A[0]/2+sum(R[k]*cos(k*x)+S[k]*sin(k*x),k=1..20)):
> # Aufgrund der Vielzahl von Reihengliedern ist T(x) nicht
  ausgedruckt.
> T(0):=simplify(subs(x=0,T(x)))*mmHg;
```

$$T(0) := 57.08550768\ mmHg$$

```
> T(0)[data]:=56*mmHg;
```

$$T(0)_{data} := 56\ mmHg$$

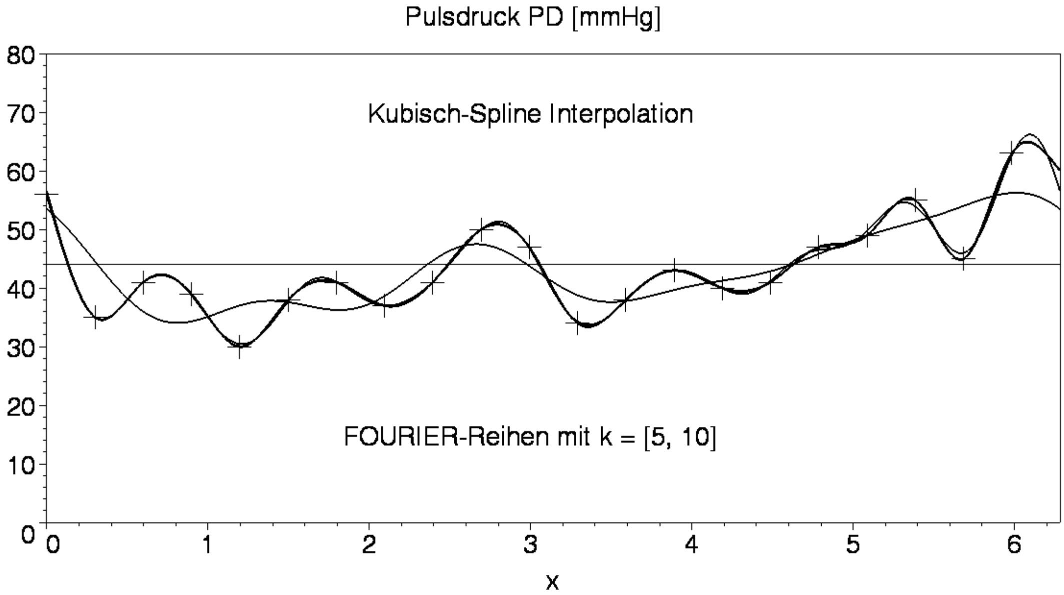
```
> T(2*Pi):=simplify(subs(x=2*Pi,T(x)))*mmHg;
```

$$T(2\pi) := 57.08550761\ mmHg$$

```
> T(2*Pi)[data]:=60*mmHg;
```

$$T(2\pi)_{data} := 60\ mmHg$$

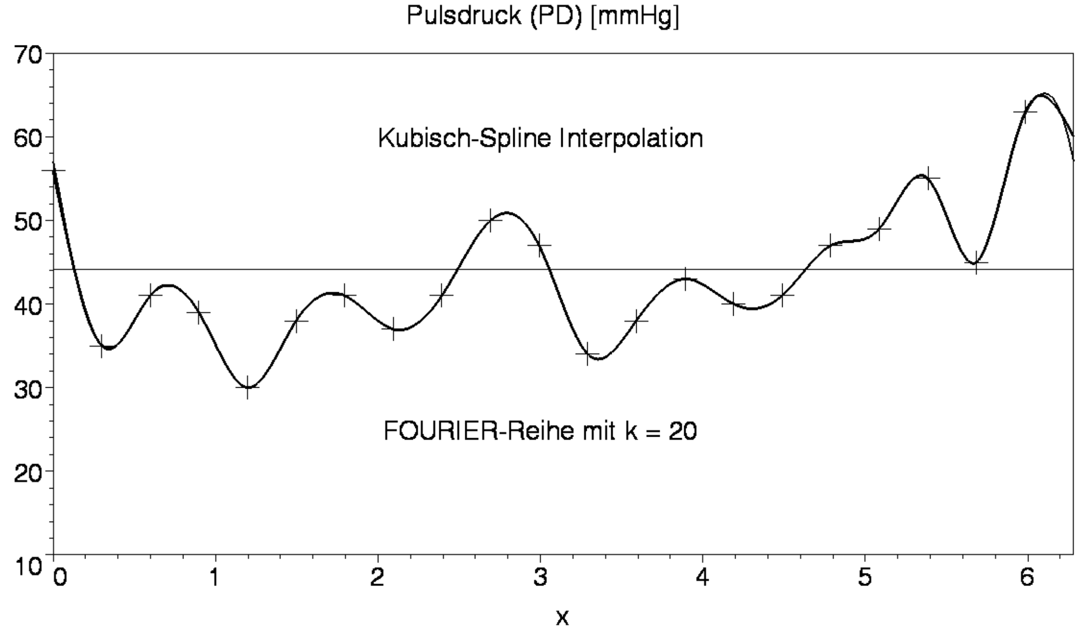
```
> # Im nächsten Bild wird die Approximation der kubischen
  Pulsdruck-Spline Funktion durch die FOURIER-Reihen mit k = 5 und
  10 gezeigt.
> alias(th=thickness,co=color):
> p[1]:=plot(spd(x),x=0..2*Pi,0..80,axes=boxed,th=3,co=black):
> p[2]:=plot([PD],x=0..2*Pi,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,
  title="Pulsdruck PD [mmHg]"):
> p[3]:=plot({y(x),z(x)},x=0..2*Pi,th=2,co=black):
> p[4]:=plot(44.09,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plots[textplot]({[3,70,`Kubisch-Spline
  Interpolation`],[3,15,`FOURIER-Reihen mit k = [5, 10]`]}):
> plots[display](seq(p[k],k=1..5));
```



Im Bild erkennt man, dass die Näherung mit k = 5 bereichsweise stark von der kubischen Splinefunktion abweicht, während die Näherung mit k = 10 Termen bereits eine sehr gute

Anpassung an die *kubische Splinefunktion* liefert. Eine weitere Verbesserung wird erzielt durch Erhöhung der Gliederzahl, z.B. auf k = 20, die jedoch nur als reine "Kosmetik" angesehen werden kann.

```
> alias(th=thickness,co=color):
> p[1]:=plot(spd(x),x=0..2*Pi,10..70,axes=boxed,th=3,co=black):
> p[2]:=plot([PD],x=0..2*Pi,th=3,co=black,
  style=point,symbol=cross,symbolsize=50,
  title="Pulsdruck (PD) [mmHg]"):
> p[3]:=plot(T(x),x=0..2*Pi,th=2,co=black):
> p[4]:=plot(44.09,x=0..2*Pi,linestyle=3,th=1,co=black):
> p[5]:=plots[textplot]({[3,60,`Kubisch-Spline Interpolation`],
  [3,25,`FOURIER-Reihe mit k = 20`]}):
> plots[display](seq(p[k],k=1..5));
```



Die Näherung mit k = 20 stimmt fast *deckungsgleich* mit der *Kubisch-Spline Interpolation* überein. Nur mit einer Lupe kann man vernachlässigbare Störungen bei x = 0 und x = 2*Pi erkennen. Die Güte der Näherungen mit k = [5, 10, 20] kann durch die *Error Normen* L[2] und l[2] ausgedrückt werden.

```
> L[2][k=5]:=
  sqrt((1/(2*Pi))*Int((SPLINE-Ypsilon)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((spd(x)-y(x))^2,x=0..2*Pi))/44.09);
```

$$L_{2_{k=5}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - Ypsilon)^2\,dx} = 0.09820579105$$

```
> L[2][k=10]:=
  sqrt((1/(2*Pi))*Int((SPLINE-ZET)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((spd(x)-z(x))^2,x=0..2*Pi))/44.09);
```

$$L_{2_{k=10}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - ZET)^2\,dx} = 0.01080568207$$

```
> L[2][k=20]:=
  sqrt((1/(2*Pi))*Int((SPLINE-TEE)^2,x=0..2*Pi))=
  evalf(sqrt((1/(2*Pi))*int((spd(x)-T(x))^2,x=0..2*Pi))/44.09);
```

$$L_{2_{k=20}} := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - TEE)^2\,dx} = 0.005069371851$$

```
> with(linalg):
> for i from 1 to 22 do
  u[i]:=evalf(subs(x=PD[i][1],y(x))-PD[i][2]) od:
> U:=vector([seq(u[i],i=1..22)]):
> l[2][k=5]:=
  (1/sqrt(number_of_points))*Norm(U,2)=
  evalf((1/sqrt(22))*norm(U,2)/44.09);
```

$$l_{2_{k=5}} := \frac{\mathrm{Norm}(U, 2)}{\sqrt{number\_of\_points}} = 0.1051411420$$

```
> for i from 1 to 22 do
  v[i]:=evalf(subs(x=PD[i][1],z(x))-PD[i][2]) od:
> V:=vector([seq(v[i],i=1..22)]):
> l[2][k=10]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(22))*norm(V,2)/44.09);
```

$$l_{2_{k=10}} := \frac{\mathrm{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.01942943563$$

```
> for i from 1 to 22 do
  w[i]:=evalf(subs(x=PD[i][1],T(x))-PD[i][2]) od:
> W:=vector([seq(w[i],i=1..22)]):
> l[2][k=20]:=
  (1/sqrt(number_of_points))*Norm(W,2)=
  evalf((1/sqrt(22))*norm(W,2)/44.09);
```

$$l_{2_{k=20}} := \frac{\mathrm{Norm}(\mathrm{W}, 2)}{\sqrt{number\_of\_points}} = 0.01516727799$$

Obige *Fehlernormen* zeigen, dass die *Näherungen* durch *FOURIER-Reihen* an die *kubischen Splinefunktionen* und die gegebenen MAD- und PD-*Daten* zufriedenstellend sind. Diese Ergebnisse wurden erzielt unter Verwendung hochgenauer Gleitkomma Arithmetik [*avaluation as floating point arithmetik (evalf* mit Digits = 10)]. Damit besitzen die FOURIER Reihen y(x), z(x), T(x) 10 Nachkommastellen. Reduziert man die Stellenzahl, z.B. auf Digits = 4, um übersichtlichere Formeln zu erhalten, können die Erbebnisse ungenau oder gar unbrauchbar sein. Das ist in Einzelfällen zu überprüfen. An einigen Stellen sind Zwissenergebnisse aus Platzgründen nicht ausgedruckt. Man kann sie jedoch ausdrucken, wenn man am Ende der entsprechenden Befehle die Doppelpunkte (:) durch Semikola (;) ersetzt. Das gilt insbesondere für die kubischen Splinefunktionen, die je nach gegebener Datenmenge äußerst lang sein können.

>