

Blutdruckwerte 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>

Blood Pressure Values. During a period of 24 hours the blood pressure of a patient at the University Hospital Aachen has been measured. Thus, we have a lot of *Systole*-, *Diastole*-, and *Pulse-Values* important for a medical doctor treating sick patients. To analyse these "data" the Maple Program 16 is very useful.

Blutdruckwerte aus Langzeitmessungen unter alltäglichen Belastungen sind von grundlegende Bedeutung zur Behandlung von Herzpatienten. Gemessen werden in einem Zeitraum von 24 h *Systole*-, *Diastole*-Werte und *Herzfrequenzen*. Die entsprechenden Messdaten wurden von einer Arztpraxis in Aachen dem Autor zur Verfügung gestellt.

Im Folgenden sind diese Daten aufgelistet und statistisch ausgewertet. Unterschieden wird zwischen Tag (von 9 - 23 Uhr) und Nachtruhe (von 23 - 6 Uhr). Zur grafischen Darstellung werden *kubische Splinefunktionen* benutzt. Die *Splineinterpolation* bietet wesentliche Vorteile gegenüber einer *LAGRANGE Interpolation*, die bei einer Vielzahl von Messpunkten zu starken Schwingungen neigt und somit keine geeigneten Ergebnisse liefert.

> **restart:**

> **with(stats): with(CurveFitting):**

Systole-Messwerte DATA

```
> DATA:=[9,116],[10,140],[11,135],[12,115],[13,132],[14,105],
[15,107],[16,105],[17,124],[18,128],[19,132],[20,113],[21,108],
[22,116],[23,123],[24,111],[25,126],[26,123],[27,112],
[28,128],[29,132],[30,132]:
```

Diastole-Messwerte data

```
> data:=[9,75],[10,85],[11,71],[12,78],[13,74],[14,69],[15,69],
[16,70],[17,70],[18,75],[19,82],[20,76],[21,72],[22,84],
[23,75],[24,68],[25,67],[26,72],[27,62],[28,69],[29,81],
[30,80]:
```

Herzfrequenz Hf

```
> Hf:=[9,68],[10,54],[11,62],[12,65],[13,68],[14,68],[15,59],
[16,67],[17,61],[18,59],[19,57],[20,57],[21,57],[22,64],[23,47],
[24,48],[25,51],[26,52],[27,50],[28,52],[29,44],[30,47]:
```

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

```
> DATAY:=[116,140,135,115,132,105,107,105,124,132,113,108,
116,123,111,126,123,112,128,132,132]:
```

```
> M:=evalf(describe[mean](DATAY),4)*mmHg;
```

$$M := 120.7 \text{ mmHg}$$

```
> S:=evalf(describe[standarddeviation](DATAY),4)*mmHg;
```

$$S := 10.55 \text{ mmHg}$$

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

```
> datay:=[75,85,71,78,74,69,69,70,70,82,76,72,84,75,68,
67,72,62,69,81,80]:
```

```
> m:=evalf(describe[mean](datay),4)*mmHg;
```

$$m := 73.76 \text{ mmHg}$$

```
> s:=evalf(describe[standarddeviation](datay),4)*mmHg;
```

$$s := 5.952 \text{ mmHg}$$

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

```
> Hfy:=[68,54,62,65,68,68,59,67,61,57,57,57,64,47,48,
51,52,50,52,44,47]:
```

```
> hf:=evalf(describe[mean](Hfy),4)/min;
```

$$hf := \frac{57.05}{\text{min}}$$

```
> hfs:=evalf(describe[standarddeviation](Hfy),4)/min;
```

$$hfs := \frac{7.605}{\text{min}}$$

Zur Erzeugung von *Splinefunktionen* beliebigen Grades ist die MAPLE-Software unverzichtbar. Die folgenden *kubischen Splinefunktionen* für Systole [Sp(x)] und Diastole [sp(x)] sind aus Platzgründen nicht ausgedruckt. Man kann sie jedoch ausdrucken, wenn man die Doppelpunkte durch Semikola ersetzt.

```
> Sp(x):=Spline([DATA],x,degree=3):
```

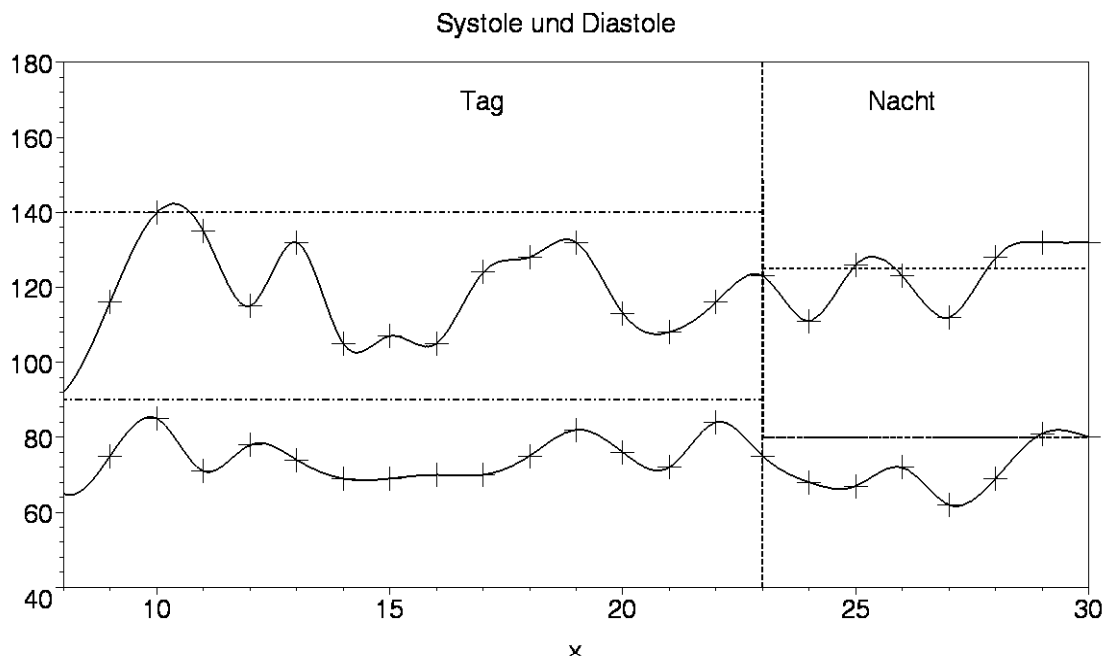
```
> sp(x):=Spline([data],x,degree=3):
```

Grafische Darstellung der Ergebnisse

```

[ > with(CurveFitting):
[ > alias(H=Heaviside,th=thickness,co=color):
[ > p[1]:=plot({Sp(x),sp(x)},x=8..30,40..180,co=black,th=2):
[ > p[2]:=plot([DATA],[data],x=8..30,axes=boxed,
[ style=point,symbol=cross,symbolsize=50,co=black):
[ > p[3]:=plot({140,90},x=8..23,linestyle=4,th=2,co=black):
[ > p[4]:=plot({125,80},x=23..30,linestyle=4,th=2,co=black,
[ ytickmarks=4):
[ > p[5]:=plot(180*H(x-22.99)-100*H(x-23.001),x=8..30,
[ linestyle=3, th=2,co=black,title="Systole und Diastole"):
[ > p[6]:=plots[textplot]([17,170,`Tag`],[26,170,`Nacht`]):
[ > plots[display]({seq(p[k],k=1..6)});

```



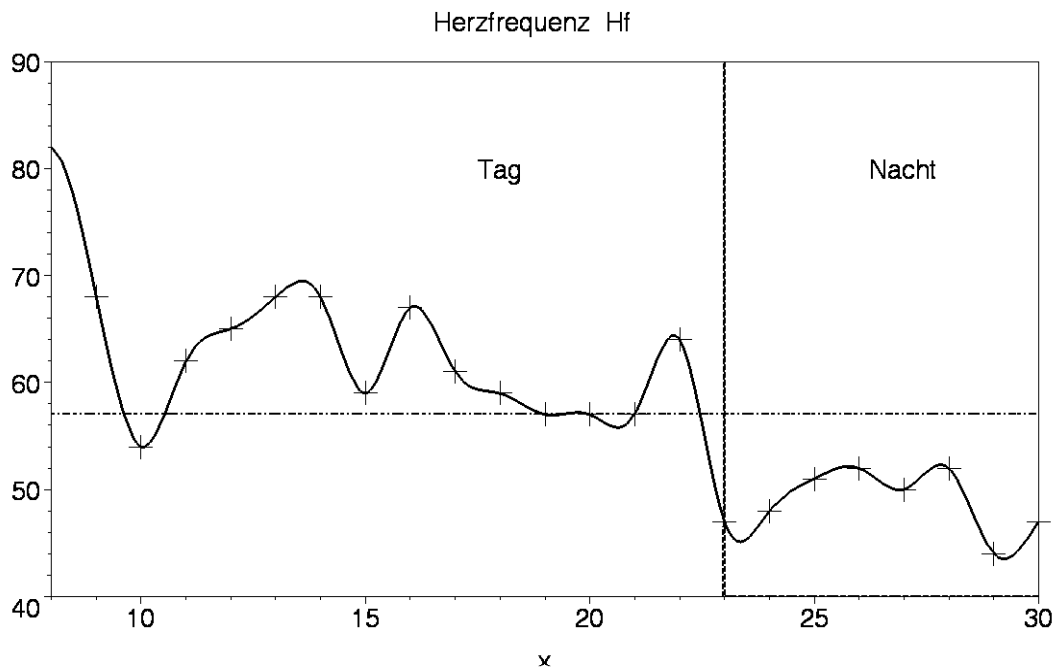
Gestrichelt eingezeichnet sind die Systole-Werte von 140 mmHg (Tag), 125 mmHg (Nacht), und die Diastole-Werte von 90 mmHg (Tag), 80 mmHg (Nacht), die nicht überschritten werden sollten.

Im nächsten Bild ist die Herzfrequenz dargestellt.

```

[ > SpHf(x):=Spline([Hf],x,degree=3):
[ > alias(H=Heaviside,th=thickness,co=color):
[ > p[1]:=plot([Hf],x=8..30,co=black,
[ style=point,symbol=cross,symbolsize=50,axes=boxed):
[ > p[2]:=plot(hf*min,x=8..30,linestyle=4,th=2,co=black):
[ > p[3]:=plot(SpHf(x),x=8..30,40..90,
[ th=3,co=black,title="Herzfrequenz Hf"):
[ > p[4]:=plot(90*H(x-22.98)-50*H(x-23.01),x=8..30,
[ linestyle=3,th=2,co=black):
[ > p[5]:=plots[textplot]([18,80,`Tag`],[27,80,`Nacht`]):
[ > plots[display]({seq(p[k],k=1..5)});

```



Die Nachtruhe (23 - 6 Uhr) ist in obigen Bildern durch $x = 23 - 30$ auf der Abszisse gekennzeichnet. Die gestrichelte Linie im letzten Bild stellt den Mittelwert $hf = 57.38$ [1/min] dar. Obige 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 unter dem Stichwort *LAGRANGE Interpolation* wird an einem Beispiel mit fünf Messpunkten gezeigt, dass sich die *Spline* Kurven mit wachsendem Grad immer stärker an das schwingende *LAGRANGE Polynom* anschmiegen. Im Grenzfall $\text{degree} \rightarrow \text{infinity}$ sind das *LAGRANGE*sche Interpolations Polynom und die *Spline* Kurve deckungsgleich.

Approximation der Splinefunktionen durch FOURIER-Reihen

```
> restart:
```

```
> with(stats): with(CurveFitting):
```

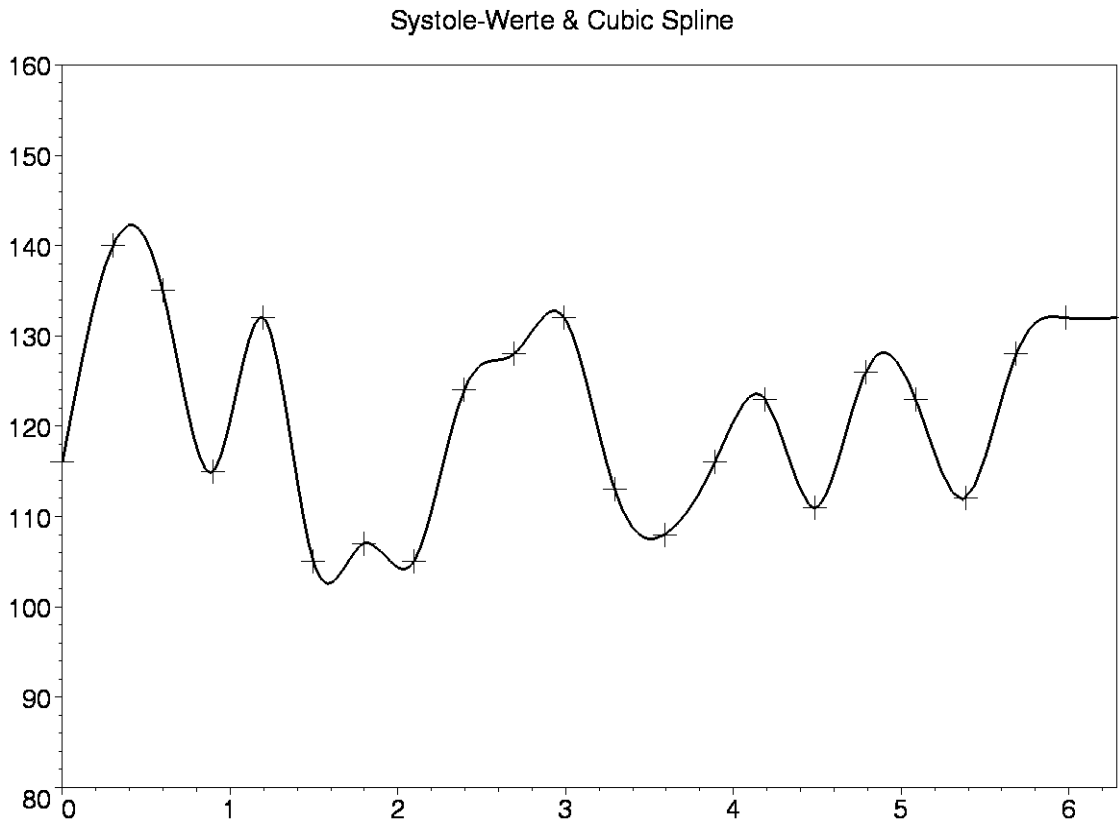
Systole-Messwerte DATA mit $x = [0..2\pi]$ im Gegensatz zu $X = [9..39]$ in oberen Bildern. Die lineare Abbildung der Abszisse [X] auf [x] erfolgt gemäß $x := -6\pi/7 + 2\pi \cdot X/21$.

```
> DATA:=[0,116],[2*Pi/21,140],[4*Pi/21,135],[2*Pi/7,115],[8*Pi/21,132],
10*Pi/21,105],[4*Pi/7,107],[2*Pi/3,105],[16*Pi/21,124],[6*Pi/7,128],
20*Pi/21,132],[22*Pi/21,113],[8*Pi/7,108],[26*Pi/21,116],[4*Pi/3,123],
10*Pi/7,111],[32*Pi/21,126],[34*Pi/21,123],[12*Pi/7,112],[38*Pi/21,128],
40*Pi/21,132],[2*Pi,132]:
```

```

[ > Sp(x):=Spline([DATA],x,degree=3):
[ > alias(H=Heaviside,th=thickness,co=color):
[ > p[1]:=plot(Sp(x),x=0..2*Pi,80..160,th=3,co=black):
[ > p[2]:=plot([DATA],x=0..2*Pi,axes=boxed,th=3,co=black,style=point
, symbol=cross, symbolsize=50,title="Systole-Werte & Cubic
Spline"):
[ > plots[display](seq(p[k],k=1..2));

```



```

[ >
[ > FOURIER_series(x):=
a[0]/2+sum(a[k]*cos(k*x)+b[k]*sin(k*x),k=1..infinity);

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

[ > F(x):=Sp(x):
[

```

```

> A[0]:=value(subs(f(x)=F(x),a[0]));
                                
$$A_0 := \frac{3639533959}{15036735}$$

> A[0]:=evalf(%);
                                
$$A_0 := 242.0428344$$

> 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..100)] 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},%):
> 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..100)] do S[i]:=subs(k=i,B[k]) od:
> # Die Koeffizienten A[i]...S[i] sind aus Platzgründen nicht
ausgedruckt.Man kann sie jedoch ausdrucken, wenn man jeweils an
den entsprechenden Zeilenenden die Doppelpunkte durch Semikola
ersetzt.
> Y(x):=A[0]/2+sum(A[k]*cos(k*x)+B[k]*sin(k*x),k=1..5);

$$Y(x) := 121.0214172 + 5.932903474 \cos(x) - 0.6527600356 \sin(x) + 6.686564454 \cos(2. x)$$


$$+ 0.4037944346 \sin(2. x) - 0.7406301146 \cos(3. x) + 6.358904279 \sin(3. x)$$

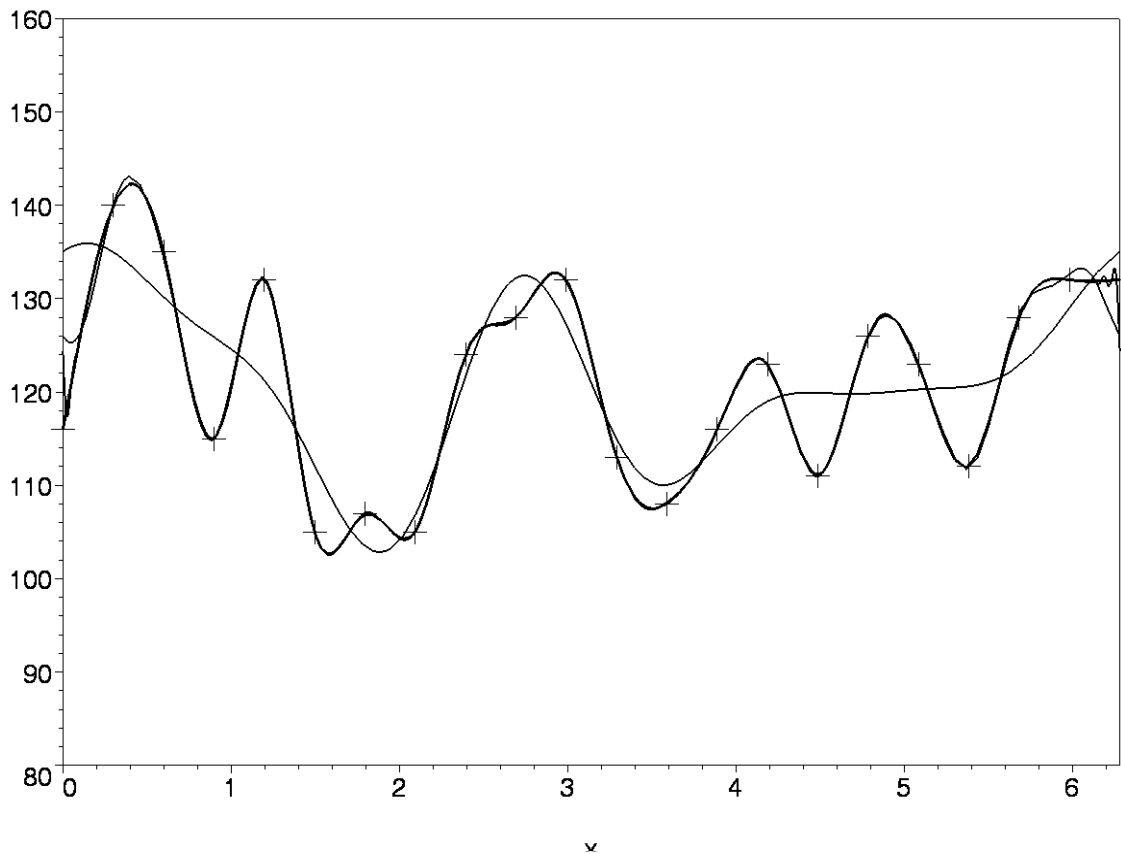

$$+ 0.2772992763 \cos(4. x) - 4.143256787 \sin(4. x) + 1.868783848 \cos(5. x)$$


$$+ 1.845916662 \sin(5. x)$$

> Z(x):=A[0]/2+sum(P[k]*cos(k*x)+Q[k]*sin(k*x),k=1..15):
> T(x):=A[0]/2+sum(R[k]*cos(k*x)+S[k]*sin(k*x),k=1..100):
> # Auch diese Formeln wurden aus Platzgründen nicht ausgedruckt.
Das folgende Bild zeigt die Näherung an die Splinefunktion durch
FOURIER-Reihen(k=[5,7,10]).
> alias(H=Heaviside,th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..2*Pi,80..160,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..2*Pi,th=3,co=black,style=point,symbol=cross,
symbolsize=50,title="Cubic-Spline & FOURIER mit k = [5, 15,
100]"):
> p[3]:=plot({Y(x),Z(x),T(x)},x=0..2*Pi,th=2,co=black):
> plots[display](seq(p[i],i=1..3));

```

Cubic-Spline & FOURIER mit k = [5, 15, 100]



Mit zunehmender Anzahl der FOURIER-Glieder (z.B. $k = [5, 15, 100]$) schmiegt sich die FOURIER-Reihe immer mehr an die Splinefunktion. Für $k = 15$ etwa sind kaum Unterschiede zur Splinefunktion zu erkennen. Für $k \rightarrow \infty$ ist die FOURIER Reihe identisch mit der Splinefunktion.

Obige Genauigkeit wird erzielt durch eine Vielzahl von Gliedern in der FOURIER-Reihe. Diese Anzahl kann jedoch reduziert werden durch Glättung (smoothing), ohne eine Genauigkeit der Approximation einzubüßen. Dazu wird ein *smoothing factor* $g(k,N)$ definiert, der in die FOURIER-Reihe eingebaut wird. Man erhält somit eine geglättete Reihe mit weniger Gliedern.

> $g(k,N) := N \cdot \sin(\pi k/N) / \pi k$; # smoothing factor

$$g(k, N) := \frac{N \sin\left(\frac{\pi k}{N}\right)}{\pi k}$$

> $G(x, n, N) := A[0]/2 + \text{sum}(g(\text{kappa}, N) * (A[\text{kappa}] * \cos(\text{kappa} * x) + B[\text{kappa}] * \sin(\text{kappa} * x)), \text{kappa}=1..n)$; # smoothing function

$$G(x, n, N) := 121.0214172 + \left(\sum_{\kappa=1}^n g(\kappa, N) (A_{\kappa} \cos(\kappa x) + B_{\kappa} \sin(\kappa x)) \right)$$

> # Als Beispiel werde die Reihe $Z(x)$ mit $n = 15$ und $N := n+1$ gewählt.

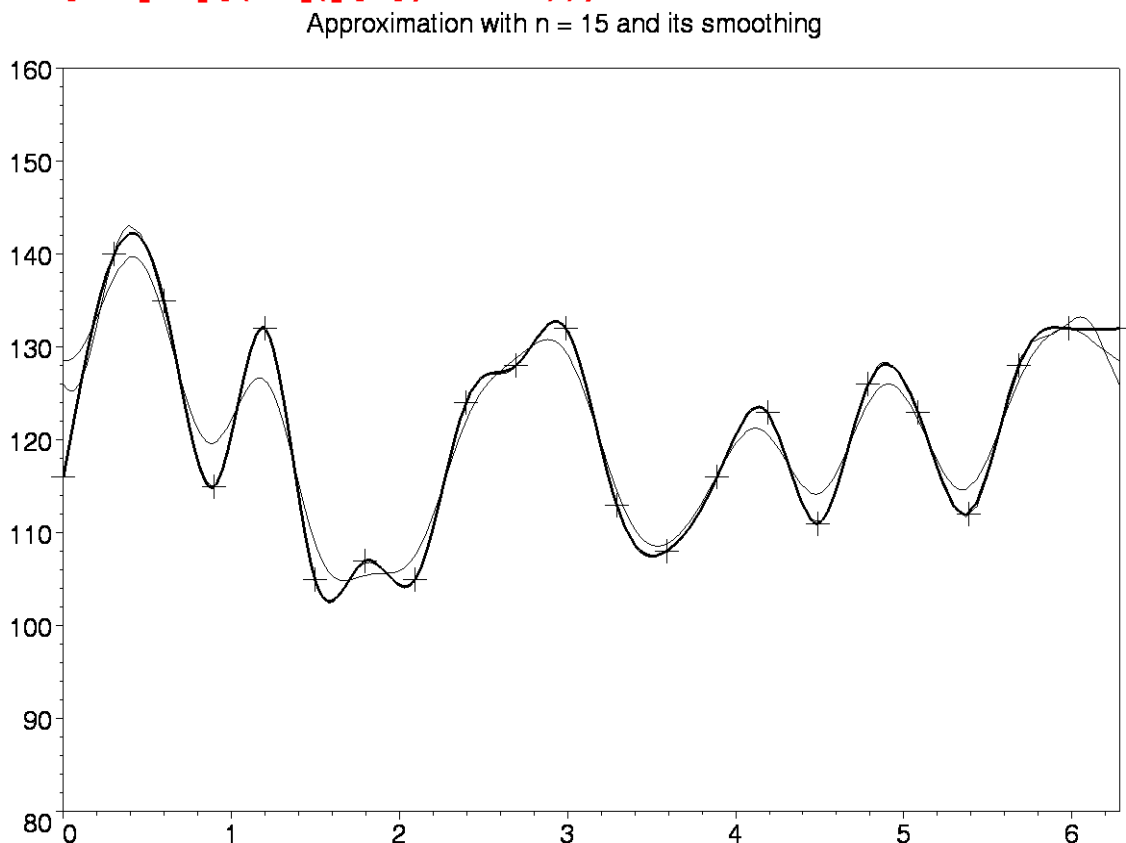
> $g(k, 16) := \text{subs}(N=16, g(k, N))$;

$$g(k, 16) := \frac{16 \sin\left(\frac{\pi k}{16}\right)}{\pi k}$$

```

> G(x,n=15,N=16):=evalf(subs({n=15,Nu=16,kappa=k,g(kappa,Nu)=g(k,16),
A[kappa]=A[k],B[kappa]=B[k]},G(x,n,N))):
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..2*Pi,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..2*Pi,80..160,th=3,
style=point,symbol=cross,symbolsize=50,co=black):
> p[3]:=plot({Z(x),G(x,n=15,N=16)},x=0..2*Pi,th=1,co=black,
title="Approximation with n = 15 and its smoothing"):
> plots[display](seq(p[k],k=1..3));

```



In diesem Worksheet wird eine *kubische Splinefunktion* zu Grunde gelegt und durch *FOURIER-Reihen* approximiert. Es wird gezeigt, dass mit zunehmender Anzahl der Glieder sich die *FOURIER-Reihen* immer mehr an die kubische Splinefunktion anschmiegen und im Grenzfall ($k \rightarrow \infty$) mit der kubischen Splinefunktion identisch sind. In einem anderen Worksheet im Application Center "Comparison between LAGRANGE and Spline Interpolation" wird von der *LAGRANGE-Interpolation* ausgegangen und gezeigt, dass mit zunehmendem Grad sich die Splinefunktion immer mehr an die betrachtete *LAGRANGE-Funktion* anschmiegt. Für $\text{grad} \rightarrow \infty$ geht die Splinefunktion in *LAGRANGE* über.

>