April / Mai 2015

Splinefunktion als FOURIER-Reihen

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

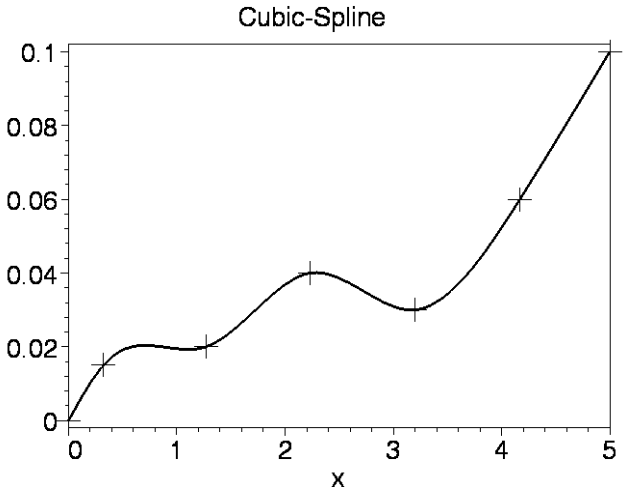
<betten@mmw.rwth-aachen.de>

Im Folgenden werden zunächst experimentelle Daten durch eine *kubische Splinefunktion* interpoliert. Diese Interpolation bietet wesentliche Vorteile beispielsweise gegenüber einer *LAGRANGE Interpolation*, die bei einer Vielzahl von Messpunkten zu starken Schwingungen neigt und somit keine geeigneten Ergebnisse liefert. Die stückweise stetige *Splinefunstion* wird als *FOURIER-Reihe* dargestellt. Dabei wird festgestellt, dass sich die *FOURIER-Reihe* mit steigender Anzahl der Reihenglieder (k = 5, 15, ...) immer stärker an die *Splinefunktion* anschmiegt. Für k = 100 oder 300 etwa sind kaum Unterschiede erkennbar. Im Grenzfall k ---> infinity fällt die *FOURIER-Reihe* mit der *Splinefunktion* zusammen.

```
> restart: with(CurveFitting):
> DATA:=
  [0,0],[0.32,0.015],[1.27,0.020],[2.23,0.040],[3.2,0.03],
  [4.17,0.06],[5,0.1]:
> Sp(x):=Spline([DATA],x,degree=3):
```

Aus Platzgründen ist die kubische *Splinefunktion* (degree=3) nicht ausgedruckt. Man kann sie jedoch ausdrucken, wenn man den Doppelpunkt am Ende der obigen Zeile durch ein Semikolon ersetzt.
Das folgende Bild zeigt die grafische Darstellung der *Splineapproximation* an die gegebenen Messdaten aus *J. BETTEN, Creep Mechanics*, 3rd Edition, Springer-Verlag Berlin 2008.

```
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..5, axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..5,th=3, title="Cubic-Spline",
  style=point,symbol=cross,symbolsize=50,co=black):
> plots[display](seq(p[k],k=1..2));
```



Approximation der Splinefunktion durch FOURIER-Reihen

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

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

```
> 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,%));
```

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

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

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

```
> F(x):=Sp(x):
> A[0]:=value(subs(f(x)=F(x),a[0]));  A[0]:=evalf(%,4);
```

$$A_0 := \frac{0.3585942363}{\pi}$$

$$A_0 := 0.1141$$

```
> A[k]:=simplify(value(subs(f(x)=F(x),a[k]))):
> for i in [seq(k,k=1..5)] do A[i]:=evalf(subs(k=i,A[k]),4) od:
> for i in [seq(k,k=1..15)] do P[i]:=evalf(subs(k=i,A[k]),4) od:
> for i in [seq(k,k=1..100)] do R[i]:=evalf(subs(k=i,A[k]),4) od:
> B[k]:=simplify(value(subs(f(x)=F(x),b[k]))):
> for i in [seq(k,k=1..5)] do B[i]:=evalf(subs(k=i,B[k]),4) od:
> for i in [seq(k,k=1..15)] do Q[i]:=evalf(subs(k=i,B[k]),4) od:
> for i in [seq(k,k=1..100)] do S[i]:=evalf(subs(k=i,B[k]),4) 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):=evalf(A[0]/2+sum(A[k]*cos(k*x)+B[k]*sin(k*x),k=1..5),4);
```

$$Y(x) := 0.05705 + 0.02272\cos(x) - 0.03769\sin(x) + 0.00004041\cos(2.x) - 0.02909\sin(2.x) + 0.003265\cos(3.x) - 0.01347\sin(3.x) - 0.001312\cos(4.x) - 0.01155\sin(4.x) - 0.0003622\cos(5.x) - 0.01004\sin(5.x)$$

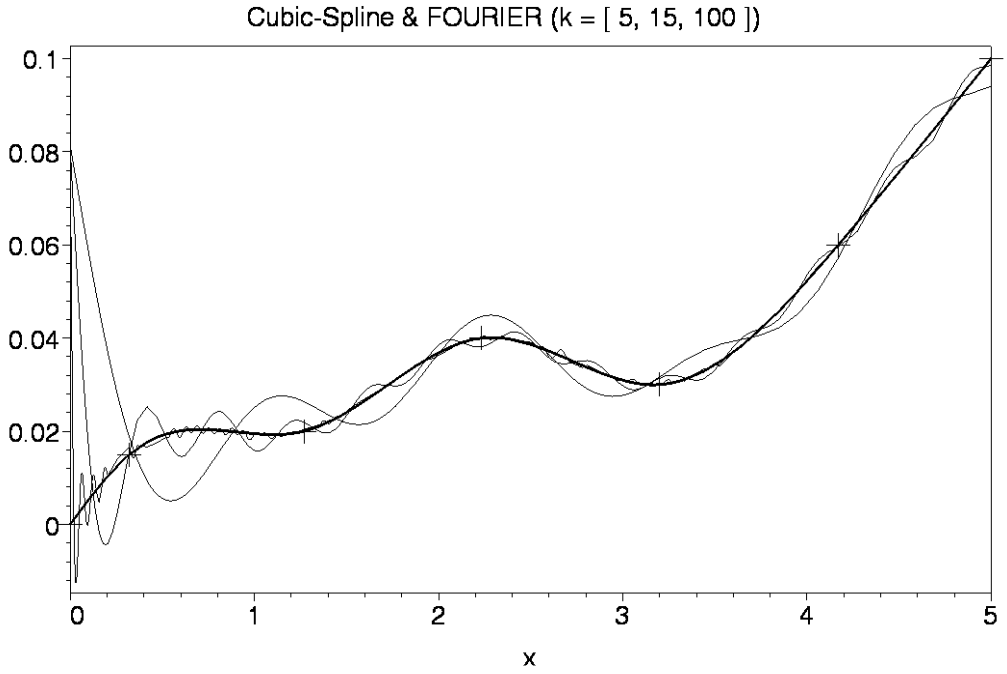
```
> Z(x):=evalf(A[0]/2+sum(P[k]*cos(k*x)+Q[k]*sin(k*x),k=1..15),4);
```

$$Z(x) := -0.0003613\cos(5.x) - 0.001312\cos(4.x) + 0.02262\cos(x) - 0.03770\sin(x) - 0.01004\sin(5.x) - 0.004659\sin(11.x) - 0.02911\sin(2.x) - 0.01156\sin(4.x) + 0.003298\cos(3.x) - 0.01349\sin(3.x) - 0.003652\sin(14.x) - 0.008424\sin(6.x) - 0.004262\sin(12.x) - 0.005625\sin(9.x) - 0.006371\sin(8.x) - 0.0001642\cos(7.x) + 0.00002255\cos(15.x) + 0.05705 + 0.00009\cos(2.x) - 0.0000127\cos(9.x) - 0.0002385\cos(6.x) - 0.00001695\cos(11.x) - 0.007259\sin(7.x) + 0.0000342\cos(14.x) - 0.005112\sin(10.x) - 0.0000556\cos(10.x) - 0.00002594\cos(12.x) - 0.00001293\cos(8.x) + 0.00004663\cos(13.x) - 0.003931\sin(13.x) - 0.003406\sin(15.x)$$

```
> T(x):=evalf(A[0]/2+sum(R[k]*cos(k*x)+S[k]*sin(k*x),k=1..100),4):
```

Auch diese Formel wurde aus Platztgründen nicht ausgedruckt. Das folgende Bild zeigt die Näherung an die *Splinefunktion* durch *FOURIER-Reihen* mit verschiedenen Gliederanzahlen (k = [5, 15, 100]).

```
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..5,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..5,th=3,
  style=point,symbol=cross,symbolsize=50,co=black):
> p[3]:=plot({Y(x),Z(x),T(x)},x=0..5,th=1,co=black,
  title="Cubic-Spline & FOURIER (k = [ 5, 15, 100 ])"):
> plots[display](seq(p[k],k=1..3));
```



Mit zunehmender Anzahl der FOURIER-Glieder (z.B. k = [5, 15, 100]) schmiegt sich die FOURIER-Reihe immer mehr an die Spline-Funktion. Für k = 100 oder auch schon für k = 50 sind kaum noch Unterschiede zur Splinefunktion zu erkennen. Für k ---> infinity ist die FOURIER-Reihe identisch mit der Spline-Funktion.
Um die Güte der Approximationen auch zahlenmäßig beurteilen zu können, wird die L-zwei Fehlernorm eingeführt.

```
> L_zwei[n]:=sqrt((1/5)*Int((F(xi)-y(xi,n))^2,xi=0..5));
```

$$L\_zwei_n := \frac{1}{5}\sqrt{5}\sqrt{\int_0^5 (\mathrm{F}(\xi) - \mathrm{y}(\xi, n))^2 \, d\xi}$$

Darin sind F(x) die *Splinefunktion* und y(x,n) die Formeln Y(x), Z(x), T(x) mit n = [5, 15, 100].

```
> y(x,5):=Y(x):   y(x,15):=Z(x):   y(x,100):=T(x):
> L_zwei[k=5]:=
  evalf(sqrt((1/4.5)*int((F(x)-Y(x))^2,x=0.5..5)),4);
```

$$L\_zwei_{k=5} := 0.005015$$

```
> L_zwei[k=15]:=
  evalf(sqrt((1/4.5)*int((F(x)-Z(x))^2,x=0.5..5)),4);
```

$$L\_zwei_{k=15} := 0.001758$$

```
> L_zwei[k=100]:=
  evalf(sqrt((1/4.5)*int((F(x)-T(x))^2,x=0.5..5)),4);
```

$$L\_zwei_{k=100} := 0.0004581$$

Obige L-zwei Fehlernormen verdeutlichen, dass die *kubische Splinefunktion* zufriedenstellend durch *FOURIER-Reihen* approximiert werden kann.

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*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+sum(g(kappa,Nu)*(A[kappa]*cos(kappa*x)+
  B[kappa]*sin(kappa*x)),kappa=1..n);    #  smoothing function
```

$$G(x, n, N) := 0.05705000000 + \left(\sum_{\kappa=1}^{n} g(\kappa, N)\,(A_\kappa \cos(\kappa\, x) + B_\kappa \sin(\kappa\, x))\right)$$

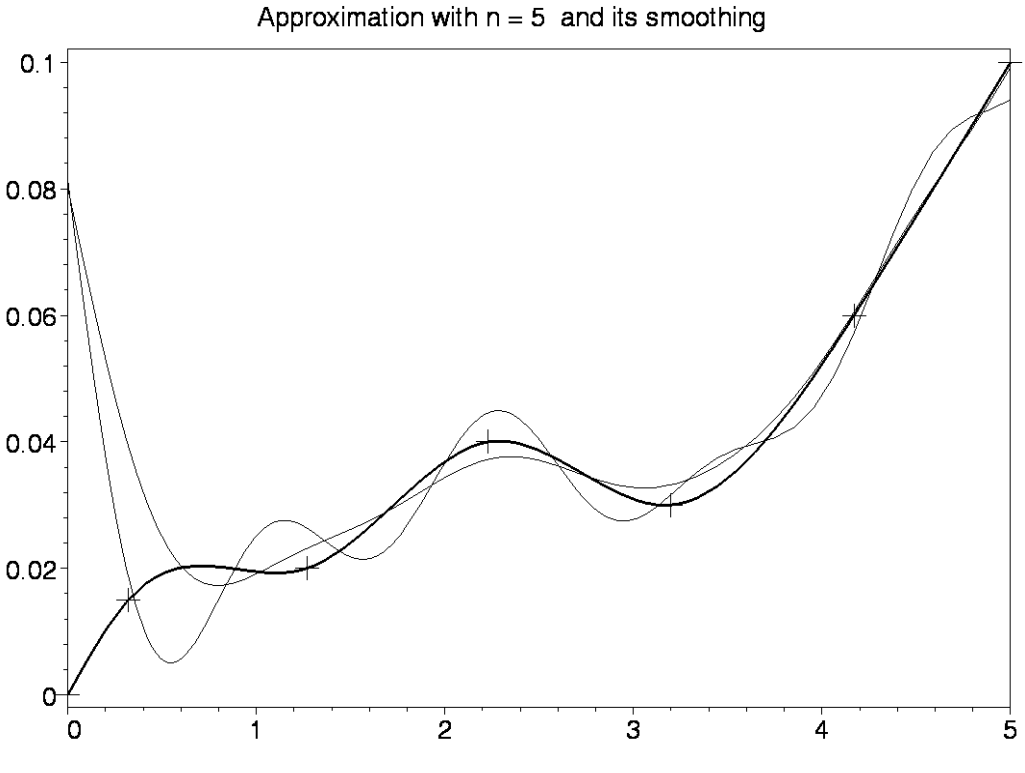
```
>  # Als Beispiel werde die Reihe Y(x) mit n = 5 und N := n +1
  gewählt.
> g(k,6):=subs(N=6,g(k,N));
```

$$g(k, 6) := \frac{6 \sin\left(\frac{\pi k}{6}\right)}{\pi k}$$

```
> G(x,n=5,N=6):= evalf(subs({n=5,Nu=6,kappa=k,g(kappa,Nu)=g(k,6),
  A[kappa]=A[k],B[kappa]=B[k]},G(x,n,N)),4);
```

$$G(x, n = 5, N = 6) := 0.05705 + 0.02164 \cos(x) - 0.03598 \sin(x) + 0.00003256 \cos(2.\, x) - 0.02407 \sin(2.\, x) + 0.002074 \cos(3.\, x) - 0.008578 \sin(3.\, x) - 0.0005851 \cos(4.\, x) - 0.004779 \sin(4.\, x) - 0.00007701 \cos(5.\, x) - 0.001917 \sin(5.\, x)$$

```
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..5,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..5,th=3,
  style=point,symbol=cross,symbolsize=50,co=black):
> p[3]:=plot({Y(x),G(x,n=5,N=6)},x=0..5,th=1,co=black,
  title="Approximation with n = 5  and its smoothing"):
> plots[display](seq(p[k],k=1..3));
```




```
> # Fehlernorm der geglätteten Reihe im Bereich x = 0.5..5:
> L_zwei[n=5]:=
  evalf(sqrt((1/4.5)*int((F(x)-G(x,n=5,N=6))^2,x=0.5..5)),4);
```

$$L\_zwei_{n=5} := 0.001966$$

Die *geglättete FOURIER-Reihe* mit nur n = 5 Gliedern liefert eine Genauigkeit (L-zwei = 0.001966), die sich von der Genauigkeit L-zwei = 0.001752 einer Approximation mit n = 15 Gliedern nur geringfügig unterscheidet. Die Verbesserung durch Glättung beträgt (5020 - 1966)/5020 = 0.6084, also ca. 61%.

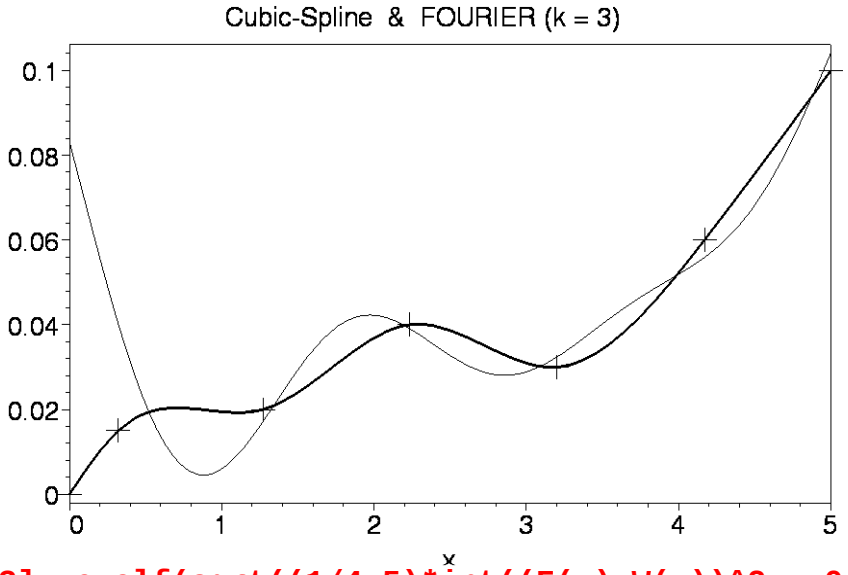
Das folgende Beispiel zeigt eine Approximation mit n = 3.

```
> for i in [seq(k,k=1..3)] do AL[i]:=evalf(subs(k=i,A[k]),4) od:

> for i in [seq(k,k=1..3)] do BE[i]:=evalf(subs(k=i,B[k]),4) od:
> W(x):=evalf(A[0]/2+sum(AL[k]*cos(k*x)+BE[k]*sin(k*x),k=1..3),4);
```

$$W(x) := 0.05705 + 0.02274\cos(x) - 0.03767\sin(x) + 0.00007\cos(2.x) - 0.02911\sin(2.x) + 0.003298\cos(3.x) - 0.01349\sin(3.x)$$

```
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..5,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..5,th=3,
  style=point,symbol=cross,symbolsize=50,co=black):
> p[3]:=plot(W(x),x=0..5,th=1,co=black,
  title="Cubic-Spline  &  FOURIER (k = 3)"):
> plots[display](seq(p[k],k=1..3));
```




```
> L_zwei[k=3]:=evalf(sqrt((1/4.5)*int((F(x)-W(x))^2,x=0.5..5)),4);
```

$$L\_zwei_{k=3} := 0.006560$$

```
> g(k,N):=N*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+sum(g(kappa,Nu)*(AL[kappa]*cos(kappa*x)+
  BE[kappa]*sin(kappa*x)),kappa=1..n);  #  smoothing function
```

$$G(x, n, N) := 0.05705000000 + \left(\sum_{\kappa=1}^{n} g(\kappa, N) \left(AL_{\kappa} \cos(\kappa x) + BE_{\kappa} \sin(\kappa x) \right) \right)$$

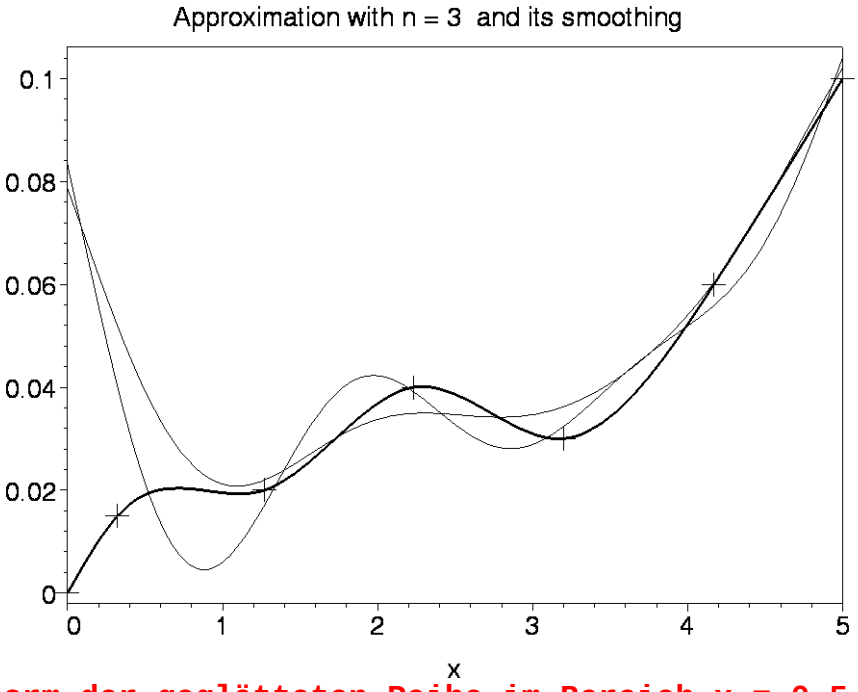
```
> g(k,4):=subs(N=4,g(k,N));
```

$$g(k, 4) := \frac{4 \sin\left(\frac{\pi k}{4}\right)}{\pi k}$$

```
> G(x,n=3,N=4):=evalf(subs({n=3,Nu=4,kappa=k,g(kappa,Nu)=g(k,4),
  AL[kappa]=AL[k],BE[kappa]=BE[k]},G(x,n,N)),4);
```

$$G(x, n = 3, N = 4) := 0.05705 + 0.02047 \cos(x) - 0.03391 \sin(x) + 0.00004456 \cos(2.\, x) - 0.01853 \sin(2.\, x) + 0.0009897 \cos(3.\, x) - 0.004048 \sin(3.\, x)$$

```
> alias(th=thickness,co=color):
> p[1]:=plot(Sp(x),x=0..5,axes=boxed,th=3,co=black):
> p[2]:=plot([DATA],x=0..5,th=3,
  style=point,symbol=cross,symbolsize=50,co=black):
> p[3]:=plot({W(x),G(x,n=3,N=4)},x=0..5,th=1,co=black,
  title="Approximation with n = 3  and its smoothing"):
> plots[display](seq(p[k],k=1..3));
```




```
> # Fehlernorm der geglätteten Reihe im Bereich x = 0.5..5:
> L_zwei[n=3]:=
  evalf(sqrt((1/4.5)*int((F(x)-G(x,n=3,N=4))^2,x=0.5..5)),4);
```

$$L\_zwei_{n=3} := 0.004503$$

```
> # eine Verbesserung gegenüber der ungeglätteten Version
  (0.006551) gemäß (6551-4506)/6551 = 0.31, also 31%.
>
```

Appendix

Im Folgenden sind einige Ergänzungen zum *Glättungsfaktor (smoothingfactor) für FOURIER-Reihen* zusammengestellt.

```
> restart:
> g(k,N):=sin(Pi*k/N)/(Pi*k/N);
```

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

```
> g(k,n+1):=subs(N=n+1,%);
```

$$g(k, n+1) := \frac{\sin\left(\frac{\pi k}{n+1}\right) (n+1)}{\pi k}$$

```
> for i in [seq(j,j=2..5)] do g(k,N=i+1):=subs(n=i,g(k,n+1)) od;
```

$$g(k, N=3) := \frac{3 \sin\left(\frac{\pi k}{3}\right)}{\pi k}$$

$$g(k, N=4) := \frac{4 \sin\left(\frac{\pi k}{4}\right)}{\pi k}$$

$$g(k, N=5) := \frac{5 \sin\left(\frac{\pi k}{5}\right)}{\pi k}$$

$$g(k, N=6) := \frac{6 \sin\left(\frac{\pi k}{6}\right)}{\pi k}$$

```
> g(k,N=infinity):=Limit(g(k,N),N=infinity)=
  limit(g(k,N),N=infinity);
```

$$g(k, N=\infty) := \lim_{N \to \infty} \frac{\sin\left(\frac{\pi k}{N}\right) N}{\pi k} = 1$$

```
> g(n,n+1):=subs(k=n,g(k,n+1));
```

$$g(n, n+1) := \frac{\sin\left(\frac{\pi n}{n+1}\right)(n+1)}{\pi n}$$

```
> for i in [seq(j,j=2..5)] do g(i,i+1):=
  evalf(subs(n=i,g(n,n+1)),4) od;
```

$$g(2, 3) := 0.4132$$

$$g(3, 4) := 0.3001$$

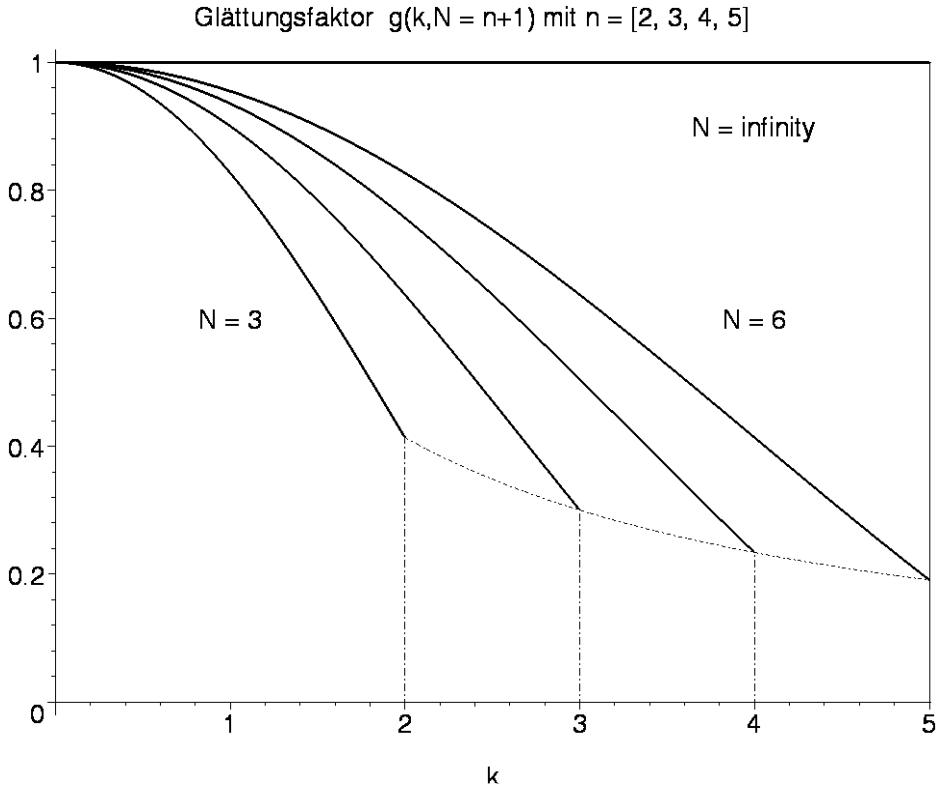
$$g(4, 5) := 0.2336$$

$$g(5, 6) := 0.1909$$

```
> g(k,k+1):=subs(n=k,g(n,n+1));
```

$$g(k, k+1) := \frac{\sin\left(\frac{\pi k}{k+1}\right)(k+1)}{\pi k}$$

```
> alias(H=Heaviside,th=thickness,l=linestyle,co=color):
> p[1]:=plot(g(k,N=3),k=0..2,th=3,co=black):
> p[2]:=plot(g(k,N=4),k=0..3,th=3,co=black):
> p[3]:=plot(g(k,N=5),k=0..4,th=3,co=black):
> p[4]:=plot(g(k,N=6),k=0..5,th=3,co=black):
> p[5]:=plot(1,k=0..5,th=3,co=black):
> p[6]:=plot(0.4132*H(k-2),k=1.99..2.01,l=4,co=black):
> p[7]:=plot(0.3001*H(k-3),k=2.99..3.01,l=4,co=black):
> p[8]:=plot(0.2336*H(k-4),k=3.99..4.01,l=4,co=black):
> p[9]:=plot(H(k-5),k=4.99..5.01,co=black,
  title="Glättungsfaktor  g(k,N = n+1) mit n = [2, 3, 4, 5]"):
> p[10]:=plot(g(k,k+1),k=2..5,l=4,co=black):
> p[11]:=plots[textplot]({[1,0.6,`N = 3`],[4,0.6,`N = 6`],
  [4,0.9,`N = infinity`]}):
> plots[display]({seq(p[m],m=1..11)});
```




```
>
> g(k=infinity,N=infinity):=Limit(g(k,k+1),k=infinity)=
  limit(g(k,k+1),k=infinity);   #  mit N=n+1
```

$$g(k=\infty, N=\infty) := \lim_{k\to\infty} \frac{\sin\left(\frac{\pi k}{k+1}\right)(k+1)}{\pi k} = 0$$

```
> # Im Gegensatz zum obigen Grenzwert
> g(k,N=infinity):=Limit(g(k,N),N=infinity)=
  limit(g(k,N),N=infinity);
```

$$g(k, N=\infty) := \lim_{N\to\infty} \frac{\sin\left(\frac{\pi k}{N}\right) N}{\pi k} = 1$$

```
>
>  # für beliebige k-Werte, wie man auch im obigen Bild erkennt.
>
```