>

January 2016

Global Temperature Anomaly

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>

Abstract

The *temperature anomaly* or *temperature index* is defined as the change from a reference temperature or a long-term mean value. A positive (negative) anomaly indicates that a measured temperuture is warmer (cooler) than the reference value. In this worksheet anomalies have been based upon the period between 1951 to 1980. We consider especially temperature change from 1980 to 2015. Since the global temperatur is increasing in the last years, i.e., the temperature anomaly is positive in the following. One can differentiate between *global land-ocean* or *surface temperature index (°C*), for instance.

Global Land-Ocean Temperature Index (°C)

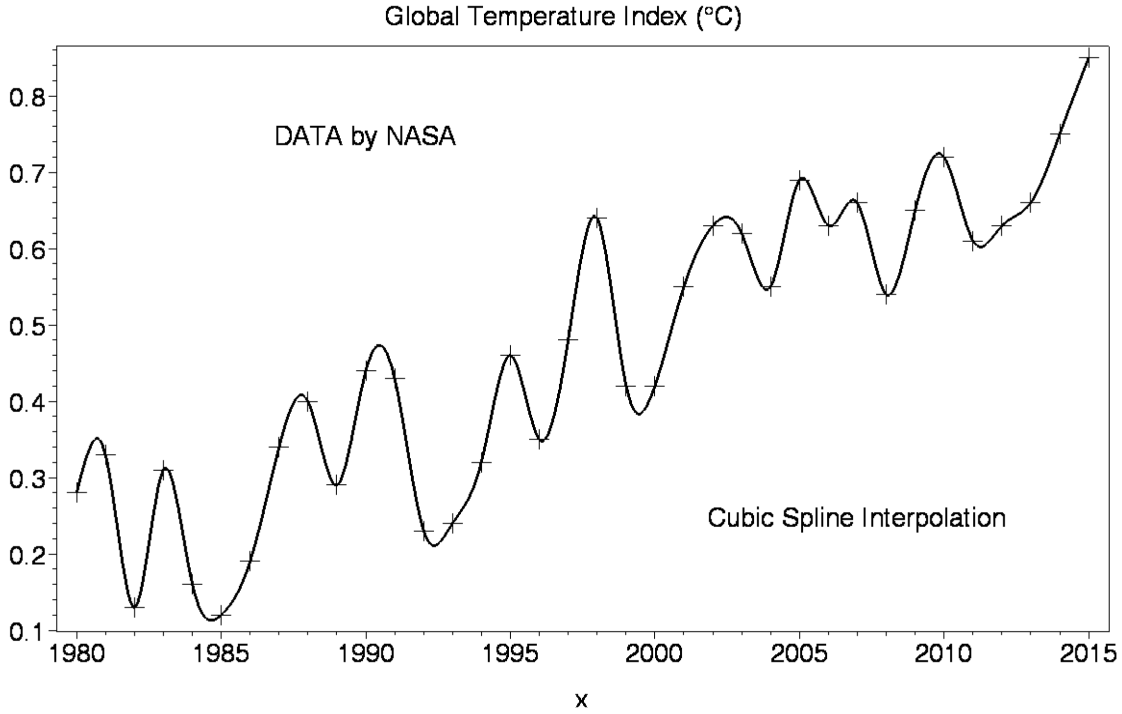
The following DATA are based on experiments do to Nasa (Anomaly with base: 1951 - 1980). These experimental values have been *interpolated* by a *cubic spline*, which is approximated by *FOURIER*-series.

```
> restart:
> with(stats):
> with(CurveFitting):
> DATA:=
  [1980,0.28],[1981,0.33],[1982,0.13],[1983,0.31],[1984,0.16],
  [1985,0.12],[1986,0.19],[1987,0.34],[1988,0.40],[1989,0.29],
  [1990,0.44],[1991,0.43],[1992,0.23],[1993,0.24],[1994,0.32],
  [1995,0.46],[1996,0.35],[1997,0.48],[1998,0.64],[1999,0.42],
  [2000,0.42],[2001,0.55],[2002,0.63],[2003,0.62],[2004,0.55],
  [2005,0.69],[2006,0.63],[2007,0.66],[2008,0.54],[2009,0.65],
  [2010,0.72],[2011,0.61],[2012,0.63],[2013,0.66],[2014,0.75],
  [2015,0.85];
```

DATA := [1980, 0.28], [1981, 0.33], [1982, 0.13], [1983, 0.31], [1984, 0.16], [1985, 0.12],
[1986, 0.19], [1987, 0.34], [1988, 0.40], [1989, 0.29], [1990, 0.44], [1991, 0.43],
[1992, 0.23], [1993, 0.24], [1994, 0.32], [1995, 0.46], [1996, 0.35], [1997, 0.48],
[1998, 0.64], [1999, 0.42], [2000, 0.42], [2001, 0.55], [2002, 0.63], [2003, 0.62],

[2004, 0.55], [2005, 0.69], [2006, 0.63], [2007, 0.66], [2008, 0.54], [2009, 0.65],
[2010, 0.72], [2011, 0.61], [2012, 0.63], [2013, 0.66], [2014, 0.75], [2015, 0.85]

```
> # Cubic Spline Function:
> Sp(x):=Spline([DATA],x,degree=3):
> # This function is not printed because of its length. One can
  print it, if we insert a semicolon (;) instead of the doppel
  point (:) at the end of the spline  comand. The following Figure
  shows the result of the Cubic Spline Interpolation.
> alias(th=thickness,co=color):
> p[1]:=plot([DATA],x=1980..2015,th=3,co=black,
  axes=boxed,style=point,symbol=cross,symbolsize=40):
> p[2]:=plot(Sp(x),x=1980..2015,th=3,co=black,
  title="Global Temperature Index (°C)"):
> p[3]:=plots[textplot]({[1990,0.75,`DATA by NASA`],
  [2007,0.25,`Cubic Spline Interpolation`]}):
> plots[display]({seq(p[k],k=1..3)});
```




```
>
```

Approximation of the Splinefunction by *FOURIER*-Series

First let us map linear the annual abscissa X = [1980...2015] to the period x = [0...2*Pi] according to:

```
> restart:
> x:=2*Pi*(X-1980)/35;
```

$$x := \frac{2\,\pi\,(X-1980)}{35}$$

```
> # The experimental data are  expressed as:
> for i from 1980 to 2015 do x[i]:=[data]=2*Pi*(i-1980)/35 od:
> datax:=array([seq(2*Pi*(i-1980)/35,i=1980..2015)]);
```

$$datax := \left[0, \frac{2\pi}{35}, \frac{4\pi}{35}, \frac{6\pi}{35}, \frac{8\pi}{35}, \frac{2\pi}{7}, \frac{12\pi}{35}, \frac{2\pi}{5}, \frac{16\pi}{35}, \frac{18\pi}{35}, \frac{4\pi}{7}, \frac{22\pi}{35}, \frac{24\pi}{35}, \frac{26\pi}{35}, \frac{4\pi}{5}, \frac{6\pi}{7}, \frac{32\pi}{35}, \frac{34\pi}{35}, \frac{36\pi}{35}, \frac{38\pi}{35}, \frac{8\pi}{7}, \frac{6\pi}{5}, \frac{44\pi}{35}, \frac{46\pi}{35}, \frac{48\pi}{35}, \frac{10\pi}{7}, \frac{52\pi}{35}, \frac{54\pi}{35}, \frac{8\pi}{5}, \frac{58\pi}{35}, \frac{12\pi}{7}, \frac{62\pi}{35}, \frac{64\pi}{35}, \frac{66\pi}{35}, \frac{68\pi}{35}, 2\pi\right]$$

```
> datax:=evalf(array([seq(2*Pi*(i-1980)/35,i=1980..2015)]),3);
```

datax := [0., 0.179, 0.358, 0.537, 0.719, 0.898, 1.08, 1.26, 1.43, 1.61, 1.79, 1.98, 2.15, 2.33, 2.51, 2.69, 2.87, 3.05, 3.23, 3.42, 3.58, 3.77, 3.96, 4.11, 4.30, 4.49, 4.68, 4.84, 5.02, 5.21, 5.37, 5.56, 5.75, 5.93, 6.09, 6.28]

```
> whattype(datax);
```

symbol

```
> datay:=array([0.28,0.33,0.13,0.31,0.16,0.12,0.19,0.34,
  0.40,0.29, 0.44,0.43,0.23,0.24,0.32,0.46,0.35,0.48,
  0.64,0.42,0.42,0.55, 0.63,0.62,0.55,0.69,0.63,0.66,
  0.54,0.65,0.72,0.61,0.63,0.66,0.75,0.85]);
```

datay := [0.28, 0.33, 0.13, 0.31, 0.16, 0.12, 0.19, 0.34, 0.40, 0.29, 0.44, 0.43, 0.23, 0.24, 0.32, 0.46, 0.35, 0.48, 0.64, 0.42, 0.42, 0.55, 0.63, 0.62, 0.55, 0.69, 0.63, 0.66, 0.54, 0.65, 0.72, 0.61, 0.63, 0.66, 0.75, 0.85]

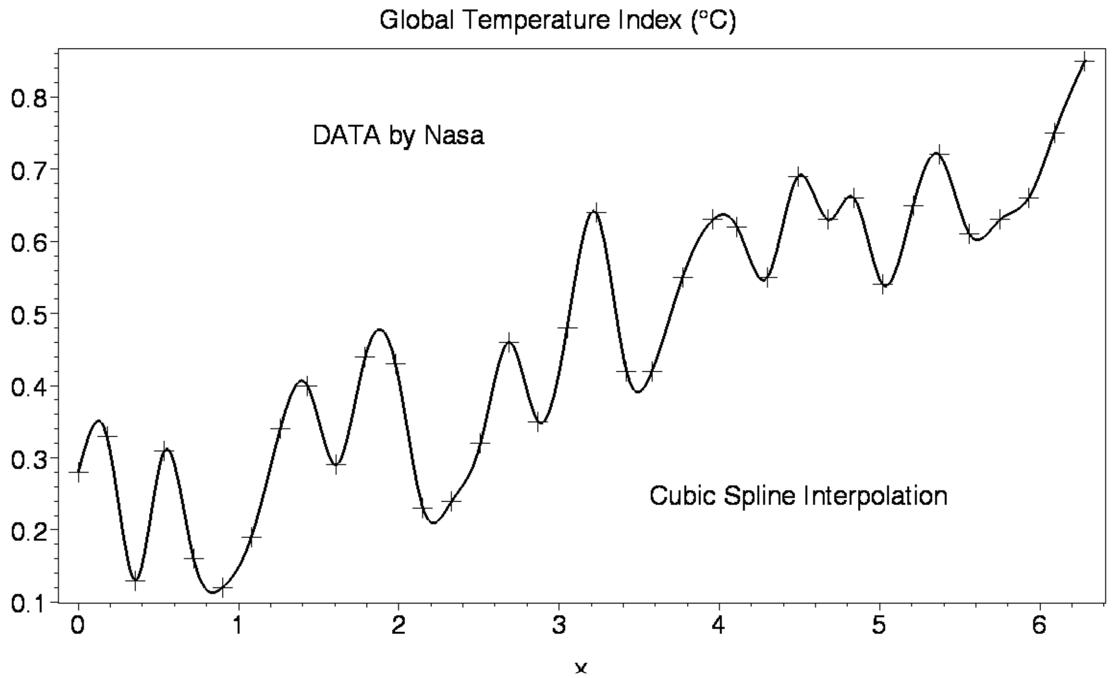
```
> whattype(datay);
```

symbol

```
> DATA:=seq([datax[i],datay[i]],i=1..36);
```

DATA := [0., 0.28], [0.179, 0.33], [0.358, 0.13], [0.537, 0.31], [0.719, 0.16], [0.898, 0.12], [1.08, 0.19], [1.26, 0.34], [1.43, 0.40], [1.61, 0.29], [1.79, 0.44], [1.98, 0.43], [2.15, 0.23], [2.33, 0.24], [2.51, 0.32], [2.69, 0.46], [2.87, 0.35], [3.05, 0.48], [3.23, 0.64], [3.42, 0.42], [3.58, 0.42], [3.77, 0.55], [3.96, 0.63], [4.11, 0.62], [4.30, 0.55], [4.49, 0.69], [4.68, 0.63], [4.84, 0.66], [5.02, 0.54], [5.21, 0.65], [5.37, 0.72], [5.56, 0.61], [5.75, 0.63], [5.93, 0.66], [6.09, 0.75], [6.28, 0.85]

```
> with(stats):   with(CurveFitting):
> # The cubic splinefunction is given by:
> Sp(x):=Spline([DATA],x,degree=3):
> # The results are represented in the following Figure:
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=40,
  title="Global Temperature Index (°C)"):
> p[3]:=plots[textplot]({[2,0.75,`DATA by Nasa`],
  [4.5,0.25,`Cubic Spline Interpolation`]}):
> plots[display](seq(p[k],k=1..3));
```




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

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

```
> 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]:=subs(k=0,%):
> a[0]:=subs(cos(0)=1,%);
```

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

```
> 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]));
```

$$A_0 := 0.9238343800$$

```
> 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.},%):
```

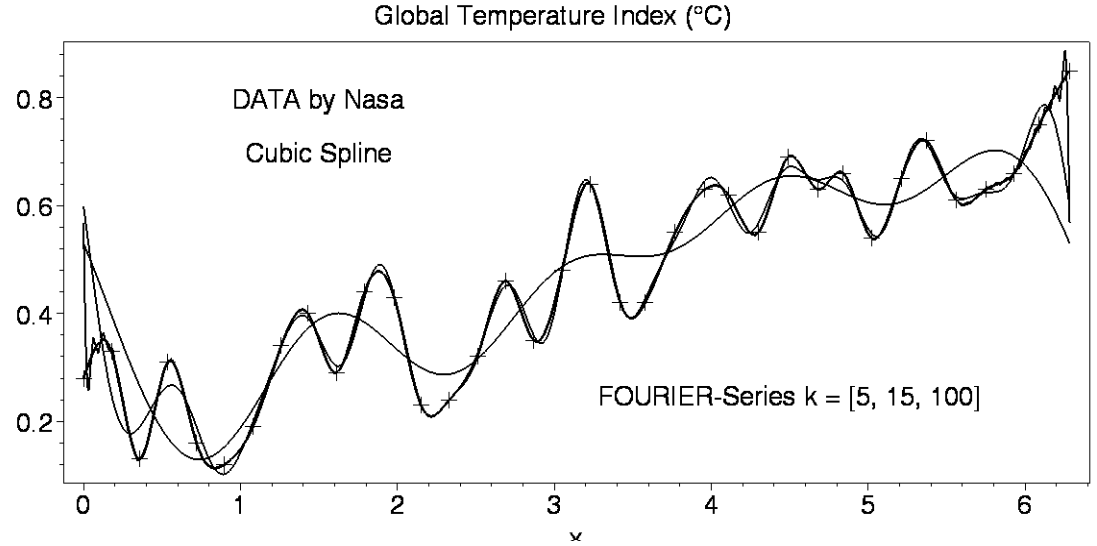
```
> for i in [seq(k,k=1..5)] do B[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..15)] do Q[i]:=subs(k=i,B[k]) od:
> for i in [seq(k,k=1..100)] do S[i]:=subs(k=i,B[k]) od:
> # The coefficients A[i]...S[i] are not printed because of their
  length. However, it can be done by inserting semicolons (;)
  instead of doppelpoints (:) at the ends of the comands. The
  several FOURIER-Series with k = 5, 15, 100 are expressed,
  respectively, by Y(x), Z(x), T(x) as follows:
>
> Y(x):=evalf(A[0]/2+sum(A[k]*cos(k*x)+B[k]*sin(k*x),k=1..5));
```

$$Y(x) := 0.4619171900 - 0.00126839 \cos(x) - 0.1994095199 \sin(x) - 0.002112290000 \cos(2.\,x) - 0.06689752450 \sin(2.\,x) + 0.02246139506 \cos(3.\,x) - 0.07400420268 \sin(3.\,x) + 0.05560480578 \cos(4.\,x) - 0.03867736827 \sin(4.\,x) - 0.006746665347 \cos(5.\,x) + 0.003879916000 \sin(5.\,x)$$

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

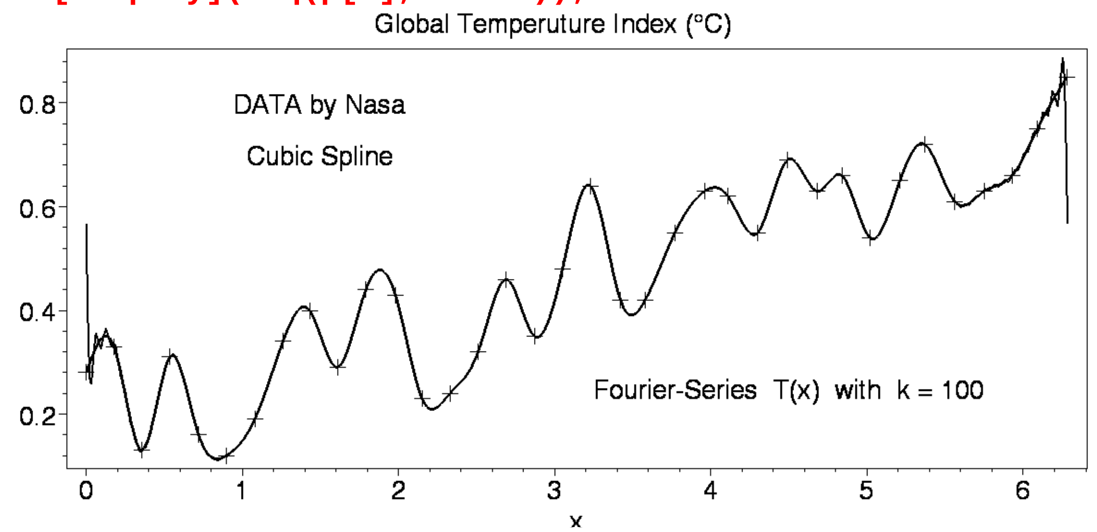
$$Z(x) := 0.00387991576 \sin(5.\,x) - 0.006746665105 \cos(5.\,x) - 0.03867736861 \sin(4.\,x) + 0.0556048050 \cos(4.\,x) - 0.0740042028 \sin(3.\,x) + 0.0224613933 \cos(3.\,x) - 0.0668975234 \sin(2.\,x) - 0.002112284 \cos(2.\,x) - 0.001268369784 \cos(x) - 0.1994095720 \sin(x) + 0.4619171900 + 0.002949052345 \sin(14.\,x) + 0.01792042605 \cos(14.\,x) - 0.01146441471 \sin(13.\,x) + 0.006900361173 \cos(13.\,x) + 0.001562043141 \sin(12.\,x) + 0.009029515667 \cos(12.\,x) - 0.006726948932 \sin(11.\,x) - 0.008730614792 \cos(11.\,x) + 0.01098689589 \sin(10.\,x) - 0.01324355714 \sin(15.\,x) - 0.01926639980 \cos(15.\,x) + 0.03359667303 \cos(10.\,x) - 0.07154219204 \sin(9.\,x) - 0.007604331623 \cos(9.\,x) - 0.02814642451 \sin(8.\,x) + 0.01793538830 \cos(8.\,x) - 0.01755966185 \sin(7.\,x) + 0.006262061080 \cos(7.\,x) - 0.02997603416 \sin(6.\,x) + 0.01299643979 \cos(6.\,x)$$

```
> T(x):=evalf(A[0]/2+sum(R[k]*cos(k*x)+S[k]*sin(k*x),k=1..100)):
> # This equation is not printed because of its length with more
  than 100 terms. However, printing is possible,if we insert a
  semicolon (;) instead of the doppel point (:). The following
  Figure illustrades the approximations of the spline function by
  FOURIER-Series with  k = [5, 15, 100]):
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35,
  title="Global Temperature Index (°C)"):
> p[3]:=plot({Y(x),Z(x),T(x)},x=0..2*Pi,th=2,co=black):
> p[4]:=plots[textplot]({[1.5,0.8,`DATA by Nasa`],
  [1.5,0.7,`Cubic Spline`],
  [4.5,0.25,`FOURIER-Series k = [5, 15, 100]`]}):
> plots[display](seq(p[k],k=1..4));
```



In this Figure we see that the *FOURIER*-Series with k = [5, 15], i.e., the functions Y(x) and Z(x) are not suitable approximations to the given DATA, because they do not increase in the period x = [5.5.. 6.28] = [2010..2015] like the cubic spline. The best approximation is the series with k = 100, which is very close to the cubic spline. With increasing the value of k one can improve the approximation. Finaly, with k = infinity, we arrive exactly at the splinefunction.

```
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35,
  title="Global Temperuture Index (°C)"):
> p[3]:=plot(T(x),x=0..2*Pi,th=2,co=black):
> p[4]:=plots[textplot]({[1.5,0.8,`DATA by Nasa`],
  [1.5,0.7,`Cubic Spline`],
  [4.5,0.25,`Fourier-Series  T(x)  with  k = 100`]}):
> plots[display](seq(p[k],k=1..4));
```



In this Figure we cannot observe a difference between the FOURIER-Series (k = 100) and the Cubic Spline.

The *L[2] error norms* can be expressed as:

```
> L[2][k=5]:=
  sqrt((1/Pi)*Int((Sp(xi)-Y(xi))^2,xi=Pi..2*Pi))=
  sqrt((1/Pi)*int((Sp(x)-Y(x))^2,x=Pi..2*Pi));
```

$$L_{2_{k=5}} := \sqrt{\frac{1}{\pi}\int_{\pi}^{2\pi}(\mathrm{Sp}(\xi)-\mathrm{Y}(\xi))^2\,d\xi} = 0.08181416648$$

```
> L[2][k=15]:=
  sqrt((1/Pi)*Int((Sp(xi)-Z(xi))^2,xi=Pi..2*Pi))=
  sqrt((1/Pi)*int((Sp(x)-Z(x))^2,x=Pi..2*Pi));
```

$$L_{2_{k=15}} := \sqrt{\frac{1}{\pi}\int_{\pi}^{2\pi}(\mathrm{Sp}(\xi)-\mathrm{Z}(\xi))^2\,d\xi} = 0.02918338151$$

```
> L[2][k=100]:=
  sqrt((1/Pi)*Int((Sp(xi)-T(xi))^2,xi=Pi..2*Pi))=0.01278068452;
```

$$L_{2_{k=100}} := \sqrt{\frac{1}{\pi}\int_{\pi}^{2\pi}(\mathrm{Sp}(\xi)-\mathrm{T}(\xi))^2\,d\xi} = 0.01278068452$$

```
>
```

The three values above show the *L[2] error norms* of the *FOURIER-Series* Y(x), Z(x), and T(x) relative to the *cubic spline function.* Furthermore one should discuss the *error norms* with respect to the given experimental data, as follows:

```
> with(linalg):
> for i from 1 to 36 do
  u[i]:=subs(x=DATA[i][1],Y(x))-DATA[i][2] od:
> U:=evalf(vector([seq(u[i],i=1..36)]),4);
```

U := [0.2498, 0.06659, 0.1357, -0.1408, -0.03049, 0.03266, 0.03307, -0.03157, -0.02774, 0.1102, -0.05623, -0.09228, 0.06943, 0.04757, -0.004570, -0.08647, 0.08845, 0.006887, -0.1322, 0.08777, 0.08558, -0.03127, -0.07473, -0.02643, 0.08638, -0.03500, 0.01490, -0.03613, 0.06447, -0.04337, -0.09017, 0.06089, 0.07040, 0.03150, -0.1087, -0.3180]

```
> l[2][k=5]:=
  (1/sqrt(number_of_points))*Norm(U,2)=
  evalf((1/sqrt(36))*norm(U,2),4);
```

$$l_{2_{k=5}} := \frac{\mathrm{Norm}(U,2)}{\sqrt{number\_of\_points}} = 0.09762$$

```
> for i from 1 to 36 do
  v[i]:=subs(x=DATA[i][1],Z(x))-DATA[i][2] od:
> V:=evalf(vector([seq(v[i],i=1..36)]),4);
```

V := [0.3189, -0.0822, 0.0593, -0.0449, 0.0308, -0.0179, 0.0089, -0.0001, -0.00907, 0.0131, -0.01048, 0.00222, 0.0001, -0.0017, 0.0051, -0.008643, 0.0084, -0.00375, 0.0007, -0.00197, 0.00754, -0.01251, 0.0163, -0.0136, 0.01712, -0.0185, 0.0182, -0.0142, 0.00834, 0.0011, -0.0041, 0.0067, -0.0054, -0.0018, 0.0292, -0.2444]

```
> l[2][k=15]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(36))*norm(V,2),4);
```

$$l_{2_{k=15}} := \frac{\mathrm{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.07042$$

```
> for i from 1 to 36 do
  w[i]:=subs(x=DATA[i][1],T(x))-DATA[i][2] od:
> W:=evalf(vector([seq(w[i],i=1..36)]),4);
```

W := [0.2869, 0.0060, -0.0010, -0.0029, -0.0022, -0.0015, 0., 0.0011, 0.00127, 0., -0.00070, -0.00037, -0.0007, -0.0002, 0.0006, 0.0009498, 0.0012, 0.00019, -0.0004, -0.00036, -0.00016, -0.00070, -0.0003, 0.0009, 0.00017, 0.0007, 0.0013, -0.0006, -0.00194, -0.0006, 0.0003, 0.0027, 0.0028, 0.0022, -0.0089, -0.2270]

```
> l[2][k=100]:=
  (1/sqrt(number_of_points))*Norm(W,2)=
  evalf((1/sqrt(36))*norm(W,2),4);
```

$$l_{2_{k=100}} := \frac{\mathrm{Norm}(W, 2)}{\sqrt{number\_of\_points}} = 0.06102$$

Note: Because of its "piecevise character" of the spline function the precision of the L[2] error norm cannot be arrived by *Digits*, for instance, equal to four. Therefore, we have calculated the L[2] norms above with the default value of ten. However, the calculation of l[2] norms with *Digits = 4* is possible without loss of *precision.*
The *error norms* may sometimes be improved without changing the number of terms, if we utilize the method of *smoothing.* As an example let us apply this method to the *FOURIER*-Series Y(x) with k = 5 terms. Introducing the *smoothing factor* g(k,N) we arrive at the *modified FOURIER-Siries* G(x,n,N) as:

```
> g(k,N):=N*sin(Pi*k/N)/Pi/k;   # smoothing factor
```

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

```
> G(x,n,N):= A[zero]/2+sum(g(kappa,Nu)*(A[kappa]*cos(kappa*x)+
  B[kappa]*sin(kappa*x)),kappa=1..n);   # smoothing function
```

$$\mathrm{G}(x, n, N) := \frac{1}{2} A_{zero} + \left(\sum_{\kappa=1}^{n} \mathrm{g}(\kappa, \mathrm{N})\,(A_\kappa \cos(\kappa\, x) + B_\kappa \sin(\kappa\, x))\right)$$

```
> # Example for Y(x) with n = 5 and N = n + 1:
> g(k,6):=subs(N=6,g(k,N));
```

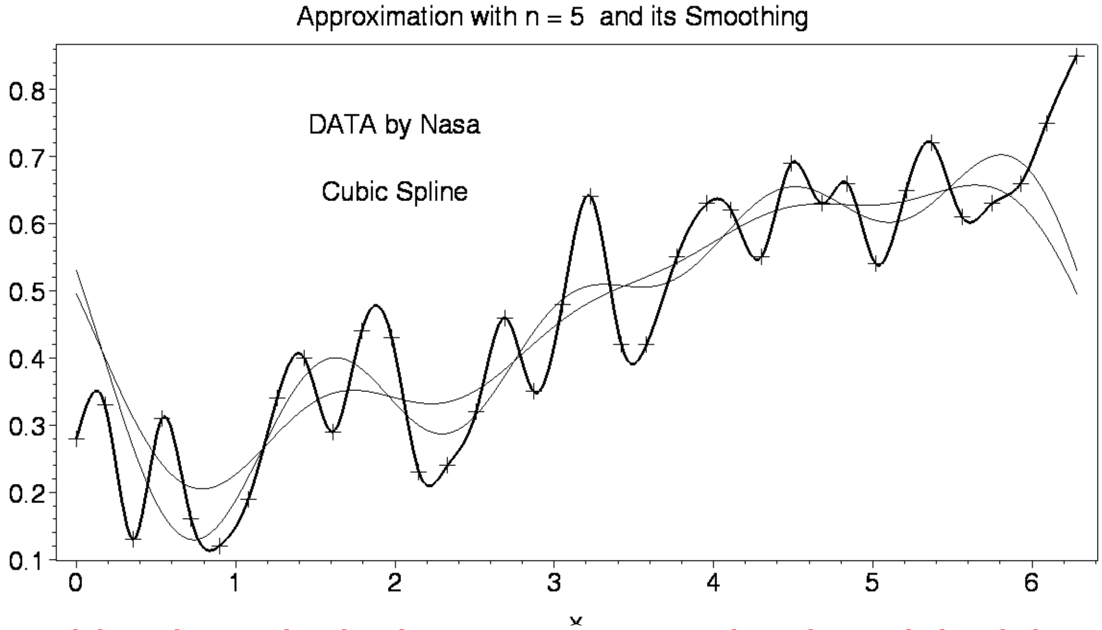
$$\mathrm{g}(k, 6) := \frac{6 \sin\left(\frac{\pi k}{6}\right)}{\pi k}$$

```
> G(x,n=5,N=6):=evalf(subs({n=5,Nu=6,kappa=k,g(kappa,Nu)=g(k,6),
```

```
A[zero]=A[0],A[kappa]=A[k],B[kappa]=B[k]},G(x,n,N)));
```

$G(x, n=5, N=6) := 0.4619171900 - 0.001211159545\cos(x) - 0.1904220876\sin(x) - 0.001746849536\cos(2.\,x) - 0.05532380917\sin(2.\,x) + 0.01429936731\cos(3.\,x) - 0.04711253834\sin(3.\,x) + 0.02299240155\cos(4.\,x) - 0.01599296323\sin(4.\,x) - 0.001288518137\cos(5.\,x) + 0.0007410093358\sin(5.\,x)$

```
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35):
> p[3]:=plot({Y(x),G(x,n=5,N=6)},x=0..2*Pi,th=1,co=black,
  title="Approximation with n = 5  and its Smoothing"):
> p[4]:=plots[textplot]({[2,0.75,`DATA by Nasa`],
  [2,0.65,`Cubic Spline`]}):
> plots[display](seq(p[k],k=1..4));
```




```
> # In this Figure both the FOURIER approximation Y(x) with n = 5
  and its smoothing are represented by thin lines and differ from
  the cubic spline (thick line) in the period x = [5..2*Pi].The
  error Norm of the smoothing with respect to the experimental
  data is given as follows:
> for i from 1 to 36 do
  m[i]:=subs(x=DATA[i][1],G(x,n=5,N=6))-DATA[i][2] od:
> M:=evalf(vector([seq(m[i],i=1..36)]),4);
```

$M := [0.2150, 0.07059, 0.1803, -0.06808, 0.04792, 0.09122, 0.05340, -0.05277, -0.07569, 0.05742, -0.08861, -0.08804, 0.1033, 0.09430, 0.03175, -0.07687, 0.07087, -0.02378, -0.1560, 0.08535, 0.1008, -0.008413, -0.06330, -0.03211, 0.06107, -0.06475, -0.0007975, -0.03190, 0.08772, -0.01613, -0.07586, 0.04567, 0.02404, -0.03168, -0.1693, -0.3533]$

```
> l[2][G_k=5]:=
  (1/sqrt(number_of_points))*Norm(M,2)=
  evalf((1/sqrt(36))*norm(M,2),4);
```

$$l_{2_{G\_k=5}} := \frac{\mathrm{Norm}(M, 2)}{\sqrt{number\_of\_points}} = 0.1038$$

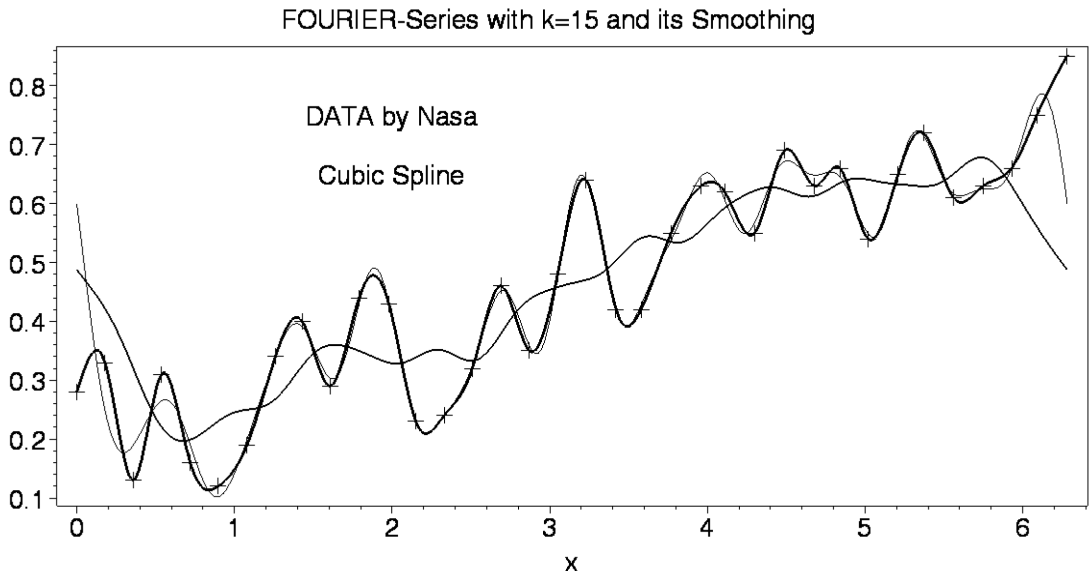
```
> # Another Example,i.e., the smoothing of the FOURIER-Series Z(x)
  with k = 15 and N = 16  has been illustrated, as follows:
> g(k,16):=subs(N=16,g(k,N));    # smoothing factor
```

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

```
> G(x,n=15,N=16):=
  evalf(subs({n=15,Nu=16,kappa=k,g(kappa,Nu)=g(k,6),
  A[zero]=A[0],A[kappa]=A[k],B[kappa]=B[k]},G(x,n,N)));
```

$$\mathrm{G}(x, n = 15, N = 16) := 0.0007410093308 \sin(5.\, x) - 0.001288518136 \cos(5.\, x) - 0.01599296325 \sin(4.\, x) + 0.02299240155 \cos(4.\, x) - 0.04711253825 \sin(3.\, x) + 0.01429936729 \cos(3.\, x) - 0.05532380937 \sin(2.\, x) - 0.001746849128 \cos(2.\, x) - 0.001211174497 \cos(x) - 0.1904220903 \sin(x) + 0.4619171900 + 0.0003484066628 \sin(14.\, x) + 0.002117153272 \cos(14.\, x) - 0.0008421315087 \sin(13.\, x) + 0.0005068738156 \cos(13.\, x) + 0.0005839784577 \sin(11.\, x) + 0.0007579202581 \cos(11.\, x) - 0.001817217951 \sin(10.\, x) - 0.001686222065 \sin(15.\, x) - 0.002453074210 \cos(15.\, x) - 0.005556844983 \cos(10.\, x) + 0.01518172469 \sin(9.\, x) + 0.001613689340 \cos(9.\, x) + 0.005819226452 \sin(8.\, x) - 0.003708111709 \cos(8.\, x) + 0.002395463120 \sin(7.\, x) - 0.0008542610962 \cos(7.\, x)$$

```
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35):
> p[3]:=plot(Z(x),x=0..2*Pi,th=1,co=black,
  title="FOURIER-Series with k=15 and its Smoothing"):
> p[4]:=plot(G(x,n=15,N=16),x=0..2*Pi,th=2,co=black):
> p[5]:=plots[textplot]({[2,0.75,`DATA by Nasa`],
  [2,0.65,`Cubic Spline`]}):
> plots[display](seq(p[k],k=1..5));
```




```
>
> # Error Norm of the smoothing with respect to the experimental
  data:
> for i from 1 to 36 do
  q[i]:=subs(x=DATA[i][1],G(x,n=15,N=16))-DATA[i][2] od:
> Q:=evalf(vector([seq(q[i],i=1..36)]),4);
```

Q := [0.2074, 0.09166, 0.1904, -0.0892, 0.0390, 0.1107, 0.0599, -0.0718, -0.07558, 0.0695, -0.08998, -0.1003, 0.1086, 0.1100, 0.0129, -0.08104, 0.08700, -0.02201, -0.1698, 0.08279, 0.1213, -0.01499, -0.07510, -0.0276, 0.07182, -0.0671, -0.0170, -0.0253, 0.1006, -0.0172, -0.0907, 0.0397, 0.0478, -0.0305, -0.1896, -0.3615]

```
> l[2][G_k=15]:=
  (1/sqrt(number-of-points))*Norm(Q,2)=
  evalf((1/sqrt(36))*norm(Q,2),4);
```

$$l_{2_{G\_k=15}} := \frac{\mathrm{Norm}(Q, 2)}{\sqrt{number - of - points}} = 0.1110$$

All Error norms calculated above proof that we have found suitable approximations to the experimental data. The following part of this Worksheet is concerned with a simple model function, the parameters of which have been determined by the

Nonlinear Regression **with Maple**

```
> restart:
> with(Statistics):   with(CurveFitting):
> # The experimental data are expressed as:
> x:=2*Pi*(xi-1980)/35;
```

$$x := \frac{2\,\pi\,(\xi - 1980)}{35}$$

```
> for i from 1980 to 2015 do x[i]:=[data]=2*Pi*(i-1980)/35 od:
> datax:=array([seq(2*Pi*(i-1980)/35,i=1980..2015)]);
```

$$datax := \left[0, \frac{2\pi}{35}, \frac{4\pi}{35}, \frac{6\pi}{35}, \frac{8\pi}{35}, \frac{2\pi}{7}, \frac{12\pi}{35}, \frac{2\pi}{5}, \frac{16\pi}{35}, \frac{18\pi}{35}, \frac{4\pi}{7}, \frac{22\pi}{35}, \frac{24\pi}{35}, \frac{26\pi}{35}, \frac{4\pi}{5}, \frac{6\pi}{7}, \frac{32\pi}{35}, \frac{34\pi}{35}, \frac{36\pi}{35}, \frac{38\pi}{35}, \frac{8\pi}{7}, \frac{6\pi}{5}, \frac{44\pi}{35}, \frac{46\pi}{35}, \frac{48\pi}{35}, \frac{10\pi}{7}, \frac{52\pi}{35}, \frac{54\pi}{35}, \frac{8\pi}{5}, \frac{58\pi}{35}, \frac{12\pi}{7}, \frac{62\pi}{35}, \frac{64\pi}{35}, \frac{66\pi}{35}, \frac{68\pi}{35}, 2\pi\right]$$

```
> X:=evalf(array([seq(2*Pi*(i-1980)/35,i=1980..2015)]),3);
```

$X := [0., 0.179, 0.358, 0.537, 0.719, 0.898, 1.08, 1.26, 1.43, 1.61, 1.79, 1.98, 2.15, 2.33, 2.51, 2.69, 2.87, 3.05, 3.23, 3.42, 3.58, 3.77, 3.96, 4.11, 4.30, 4.49, 4.68, 4.84, 5.02, 5.21, 5.37, 5.56, 5.75, 5.93, 6.09, 6.28]$

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

symbol

```
> Y:=array([0.28,0.33,0.13,0.31,0.16,0.12,0.19,0.34,
  0.40,0.29,0.44,0.43,0.23,0.24,0.32,0.46,0.35,0.48,0.64,
  0.42,0.42,0.55,0.63,0.62,0.55,0.69,0.63,0.66,0.54,0.65,
  0.72,0.61,0.63,0.66,0.75,0.85]);
```

$Y := [0.28, 0.33, 0.13, 0.31, 0.16, 0.12, 0.19, 0.34, 0.40, 0.29, 0.44, 0.43, 0.23, 0.24, 0.32, 0.46, 0.35, 0.48, 0.64, 0.42, 0.42, 0.55, 0.63, 0.62, 0.55, 0.69, 0.63, 0.66, 0.54, 0.65, 0.72, 0.61, 0.63, 0.66, 0.75, 0.85]$

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

symbol

```
> DATA:=seq([X[i],Y[i]],i=1..36);
```

$DATA := [0., 0.28], [0.179, 0.33], [0.358, 0.13], [0.537, 0.31], [0.719, 0.16], [0.898, 0.12], [1.08, 0.19], [1.26, 0.34], [1.43, 0.40], [1.61, 0.29], [1.79, 0.44], [1.98, 0.43], [2.15, 0.23], [2.33, 0.24], [2.51, 0.32], [2.69, 0.46], [2.87, 0.35], [3.05, 0.48], [3.23, 0.64], [3.42, 0.42], [3.58, 0.42], [3.77, 0.55], [3.96, 0.63], [4.11, 0.62], [4.30, 0.55], [4.49, 0.69], [4.68, 0.63], [4.84, 0.66], [5.02, 0.54], [5.21, 0.65], [5.37, 0.72], [5.56, 0.61], [5.75, 0.63], [5.93, 0.66], [6.09, 0.75], [6.28, 0.85]$

```
> # The cubic splinefunction is given by:
> Sp(x):=Spline([DATA],x,degree=3):
> # A suitable model function to the DATA has been assumed as:
> f:=(t,A,B,a,b,c,d,e) -> a+A*exp(B*t)*(b*cos(c*t)+d*sin(e*t));
```

$$f := (t, A, B, a, b, c, d, e) \rightarrow a + A\,\mathbf{e}^{(B\,t)}\,(b\cos(c\,t) + d\sin(e\,t))$$

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

$$y := t \rightarrow 0.3148 + 0.003582\,\mathbf{e}^{(0.9353\,t)}\,(-2.897\cos(1.239\,t) - 3.728\sin(0.9736\,t))$$

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

$$\mathrm{y}(x) := 0.3148 + 0.003582\,\mathbf{e}^{(0.9353\,x)}\,(-2.897\cos(1.239\,x) - 3.728\sin(0.9736\,x))$$

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

$$\mathrm{y}(0) := 0.304422946$$

```
> y(0)[data]:=0.28;
```

$$\mathrm{y}(0)_{data} := 0.28$$

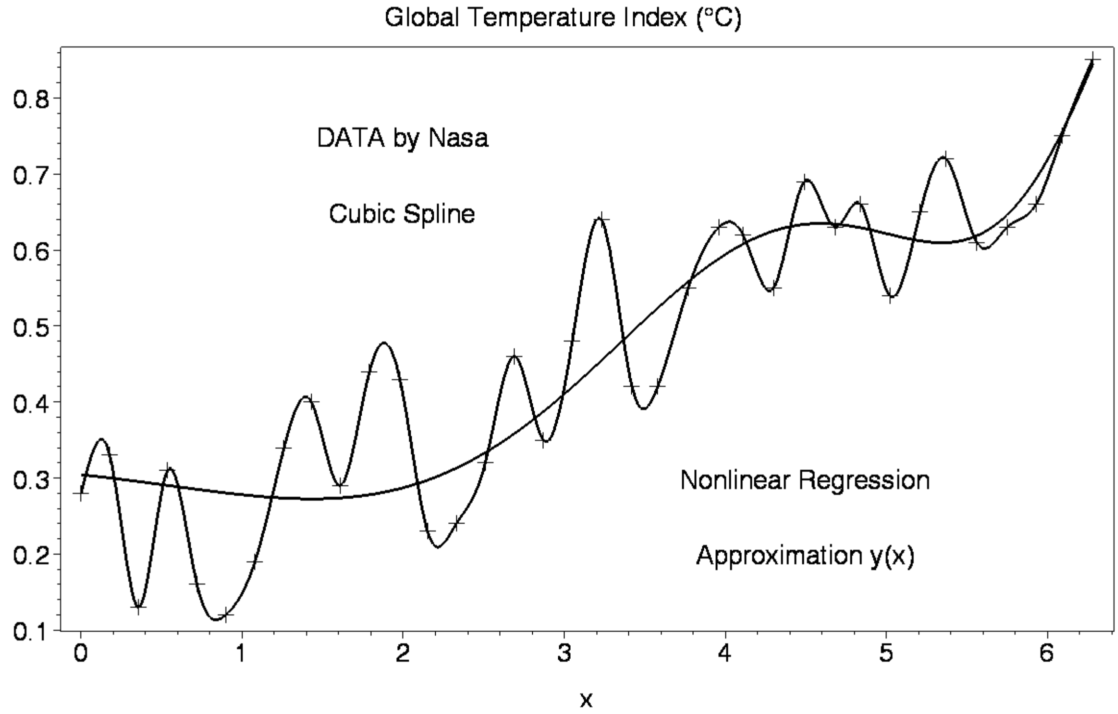
```
> y(2*Pi):=simplify(subs(x=2*Pi,y(x)));
```

$$y(2\,\pi) := 0.8455447604$$

```
> y(2*Pi)[data]:=0.85;
```

$$y(2\,\pi)_{data} := 0.85$$

```
> # The results are representd in the next Figure:
>
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35,
  title="Global Temperature Index (°C)"):
> p[3]:=plot(y(x),x=0..2*Pi,th=3,co=black):
> p[4]:=plots[textplot]({[2,0.75,`DATA by Nasa`],
  [2,0.65,`Cubic Spline`],[4.5,0.3,`Nonlinear Regression`],
  [4.5,0.2,`Approximation y(x)`]}):
> plots[display](seq(p[k],k=1..4));
```



In this Figure, we see, the approximation y(x) is verry close to the cubic spline. The *L[2] error norm* between the cubic spline and the approximation y(x) can be expressed as:

```
> L[2]:=
  sqrt((1/(2*Pi))*Int((SPLINE-APPROXIMATION)^2,xi=0..2*Pi))=
  sqrt((1/2/Pi)*int((Sp(x)-y(x))^2,x=0..2*Pi));
```

$$L_2 := \frac{1}{2}\sqrt{2}\,\sqrt{\frac{1}{\pi}\int_0^{2\,\pi}(SPLINE - APPROXIMATION)^2\,d\xi} = 0.08181141513$$

The *l[2] error norm* of the approximation y(x) with respect to the experimental data is given by:

```
> with(linalg):
> for i from 1 to 36 do
  u[i]:=subs(x=DATA[i][1],y(x))-DATA[i][2] od:
> U:=vector([seq(u[i],i=1..36)]);
```

U := [0.024422946, -0.02990521279, 0.1653248282, -0.01970833595, 0.1251637815, 0.1604439126, 0.08639540586, -0.06637435981, -0.1273699943, -0.01615192083, -0.1620153257, -0.1438610605, 0.06719203757, 0.07311090985, 0.01353535615, -0.1015990652, 0.03735430600, -0.0602825065, -0.1855268186, 0.0722679495, 0.1036078669, 0.0084059714, -0.0414537161, -0.0122427551, 0.0750205663, -0.0563249393, 0.0040344509, -0.0308672925, 0.0804514501, -0.0380588973, -0.1103774758, 0.0085249307, 0.0162382905, 0.0340846772, 0.0047840012, -0.0060401312]

```
> l[2]:=
  (1/sqrt(number_of_points))*Norm(U,2)=
  evalf((1/sqrt(36))*norm(U,2),4);
```

$$l_2 := \frac{\mathrm{Norm}(U, 2)}{\sqrt{number\_of\_points}} = 0.08415$$

Both the L[2] and the l[2] error norms show that the assumed function y(x) is a suitable approximation to the experimental data.

Note: Because of its "piecevise character" of the splinefunction the high precision of the L[2] norm cannot be arrived by *Digits*, for instance, equal to four. Thus, we have calculated the L[2] error norms above with "default" value of ten. However, the calculation of l[2] error norms with *Digits = 4* is possible without loss of *precision.*

Another example within the category ***Nonlinear Regression*** should be discussed in the following:

```
> h:=(t,A,B,C,D,a,b,c) ->
  a+A*exp(B*t)*cos(b*t)+C*exp(D*t)*sin(c*t);
```

$$h := (t, A, B, C, \mathrm{D}, a, b, c) \rightarrow a + A\, \mathbf{e}^{(B\, t)} \cos(b\, t) + C\, \mathbf{e}^{(\mathrm{D}\, t)} \sin(c\, t)$$

```
> z:=unapply(evalf(NonlinearFit(h(t,A,B,C,D,a,b,c),X,Y,t),4),t);
```

$$z := t \rightarrow 0.2925 - 0.008539\, \mathbf{e}^{(0.9705\, t)} \cos(1.001\, t) - 0.001584\, \mathbf{e}^{(1.267\, t)} \sin(0.7948\, t)$$

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

$$\mathrm{z}(x) := 0.2925 - 0.008539\, \mathbf{e}^{(0.9705\, x)} \cos(1.001\, x) - 0.001584\, \mathbf{e}^{(1.267\, x)} \sin(0.7948\, x)$$

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

$$\mathrm{z}(0) := 0.283961$$

```
> z(0)[data]:=0.28;
```

$$\mathrm{z}(0)_{data} := 0.28$$

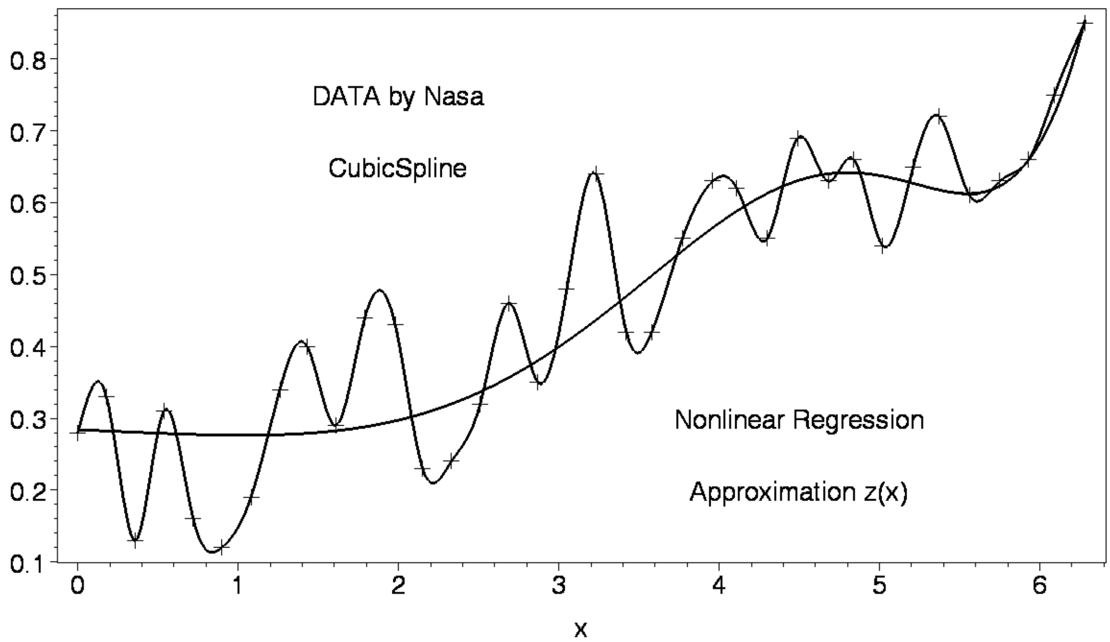
```
> z(2*Pi):=simplify(subs(x=2*Pi,z(x)));
```

$$\mathrm{z}(2\,\pi) := 0.855255878$$

```
> z(2*Pi)[data]:=0.85;
```

$$z(2\pi)_{data} := 0.85$$

```
> # The results of the approximation z(x) are represented in the
  next Figure:
> 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,th=3,co=black,
  style=point,symbol=cross,symbolsize=35,
  title="Global Temperature Index (°C)"):
> p[3]:=plot(z(x),x=0..2*Pi,th=3,co=black):
> p[4]:=plots[textplot]({[2,0.75,`DATA by Nasa`],
  [2,0.65,`CubicSpline`], [4.5,0.3,`Nonlinear Regression`],
  [4.5,0.2,`Approximation z(x)`]}):
> plots[display](seq(p[k],k=1..4));
```



In this Figure, we see, the approximation z(x) is very close to the cubic spline, similar to the approimation y(x) in one Figure before. The *M[2] error norm* between the cubic spline and the approximation z(x) can be expressed as:

```
> M[2]:=
  sqrt((1/2/Pi)*Int((SPLINE-APPROXIMATION)^2,xi=0..2*Pi))=
  sqrt((1/2/Pi)*int((Sp(x)-z(x))^2,x=0..2*Pi));
```

$$M_2 := \frac{1}{2}\sqrt{2}\sqrt{\frac{1}{\pi}\int_0^{2\pi}(SPLINE - APPROXIMATION)^2\, d\xi} = 0.08037261461$$

The *m[2] error norm* of the approximation z(x) with respect to the experimatal data is given by:

```
> with(linalg):
> for i from 1 to 36 do
  v[i]:=subs(x=DATA[i][1],z(x))-DATA[i][2] od:
```

```
> V:=vector([seq(v[i],i=1..36)]);
```

V := [0.003961, -0.04777816197, 0.1504815197, -0.03114638782, 0.1174672875, 0.1565585479, 0.08633369230, -0.06291983162, -0.1210486070, -0.007505475104, -0.1519988774, -0.1336479029, 0.07636888008, 0.07989929510, 0.01667348684, -0.1031763805, 0.03033314104, -0.07300329763, -0.2036168606, 0.04959397336, 0.07849631705, -0.01732185911, -0.06490280976, -0.03178133171, 0.0627937730, -0.0594905968, 0.0101314769, -0.0183286679, 0.0969226508, -0.0231649540, -0.102295258, 0.002491299, -0.006965774, -0.000732068, -0.026510733, 0.002523863]

```
> m[2]:=
  (1/sqrt(number_of_points))*Norm(V,2)=
  evalf((1/sqrt(36))*norm(V,2),4);
```

$$m_2 := \frac{\text{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.08255$$

Both the M[2] and the m[2] error norms show that the assumed function z(x) on pages 14/15 like y(x) on pages 12/13 is a suitable approximation to the experimental data. The error norms M[2], m[2] of the last approximation z(x) are smaler, respectively, than the error norms L[2], l[2] of the approximation y(x).

Note: Because of its "piecevise character" of the splinefunction the high precision of the M[2] error norm cannot be arrived by *Digits*, for example, equal to four. Thus, we have calculated the M[2] error norm with "default" value of ten. However, the calculation of the m[2] error norm with *Digits = 4* is possible without loss of *precision*.

Five-Year Mean Temperature Change with the Base Period 1950...2015

```
> restart:
> with(Statistics):  with(CurveFitting):
> dataxl:=[seq(k,k=1950..2015,5)];
```

dataxl :=

[1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015]

```
> whattype(dataxl);
```

list

```
> X:=array([seq(k,k=1950..2015,5)]);
```

X := [1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985, 1990, 1995, 2000, 2005, 2010, 2015]

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

symbol

```
> datay[1946..1950]:=[-0.04,-0.04,-0.10,-0.10,-0.18];
```

$$datay_{1946\,..\,1950} := [-0.04, -0.04, -0.10, -0.10, -0.18]$$

```
> m[1950]:=evalf(Mean(%),3);
```

$$m_{1950} := -0.0920$$

```
> datay[1951..1955]:=[-0.07,0.01,0.09,-0.12,-0.14];
```

$$datay_{1951\,..\,1955} := [-0.07, 0.01, 0.09, -0.12, -0.14]$$

```
> m[1955]:=evalf(Mean(%),3);
```

$$m_{1955} := -0.0460$$

```
> datay[1956..1960]:=[-0.19,0.04,0.06,0.03,-0.03];
```

$$datay_{1956\,..\,1960} := [-0.19, 0.04, 0.06, 0.03, -0.03]$$

```
> m[1960]:=evalf(Mean(%),3);
```

$$m_{1960} := -0.0180$$

```
> datay[1961..1965]:=[0.05,0.03,0.06,-0.21,-0.10];
```

$$datay_{1961\,..\,1965} := [0.05, 0.03, 0.06, -0.21, -0.10]$$

```
> m[1965]:=evalf(Mean(%),3);
```

$$m_{1965} := -0.0340$$

```
> datay[1966..1970]:=[-0.05,-0.02,-0.07,0.06,0.03];
```

$$datay_{1966\,..\,1970} := [-0.05, -0.02, -0.07, 0.06, 0.03]$$

```
> m[1970]:=evalf(Mean(%),3);
```

$$m_{1970} := -0.0100$$

```
> data[1972..1975]:=[-0.09,0.01,0.15,-0.08,-0.01];
```

$$data_{1972\,..\,1975} := [-0.09, 0.01, 0.15, -0.08, -0.01]$$

```
> m[1975]:=evalf(Mean(%),3);
```

$$m_{1975} := -0.00400$$

```
> datay[1976..1980]:=[-0.11,0.18,0.07,0.17,0.28];
```

$$datay_{1976\,..\,1980} := [-0.11, 0.18, 0.07, 0.17, 0.28]$$

```
> m[1980]:=evalf(Mean(%),3);
```

$$m_{1980} := 0.118$$

```
> datay[1981..1985]:=[0.33,0.13,0.31,0.16,0.12];
```

$$datay_{1981\,..\,1985} := [0.33, 0.13, 0.31, 0.16, 0.12]$$

```
> m[1985]:=evalf(Mean(%),3);
```

$$m_{1985} := 0.210$$

```
> datay[1986..1990]:=[0.19,0.34,0.40,0.29,0.44];
```

$$datay_{1986\,..\,1990} := [0.19, 0.34, 0.40, 0.29, 0.44]$$

```
> m[1990]:=evalf(Mean(%),4);
```

$$m_{1990} := 0.3320$$

```
> datay[1991..1995]:=[0.43,0.23,0.24,0.32,0.46];
```

$$datay_{1991\,..\,1995} := [0.43, 0.23, 0.24, 0.32, 0.46]$$

```
> m[1995]:=evalf(Mean(%),4);
```

$$m_{1995} := 0.3360$$

```
> datay[1996..2000]:=[0.35,0.48,0.64,0.42,0.42];
```

$$datay_{1996\,..\,2000} := [0.35, 0.48, 0.64, 0.42, 0.42]$$

```
> m[2000]:=evalf(Mean(%),4);
```

$$m_{2000} := 0.4620$$

```
> datay[2001..2005]:=[0.55,0.63,0.62,0.55,0.69];
```

$$datay_{2001\,..\,2005} := [0.55, 0.63, 0.62, 0.55, 0.69]$$

```
> m[2005]:=evalf(Mean(%),4);
```

$m_{2005} := 0.6080$

```
> datay[2006..2010]:=[0.63,0.66,0.54,0.65,0.72];
```

$datay_{2006\,..\,2010} := [0.63, 0.66, 0.54, 0.65, 0.72]$

```
> m[2010]:=evalf(Mean(%),3);
```

$m_{2010} := 0.640$

```
> datay[2011..2015]:=[0.61,0.63,0.66,0.75,0.85];
```

$datay_{2011\,..\,2015} := [0.61, 0.63, 0.66, 0.75, 0.85]$

```
> m[2015]:=evalf(Mean(%),3);
```

$m_{2015} := 0.700$

```
> datayl:=[seq(m[i],i=1950..2015,5)];
```

datayl := [-0.0920, -0.0460, -0.0180, -0.0340, -0.0100, -0.00400, 0.118, 0.210, 0.3320, 0.3360, 0.4620, 0.6080, 0.640, 0.700]

```
> whattype(datayl);
```

list

```
> Y:=array([seq(m[i],i=1950..2015,5)]);
```

Y := [-0.0920, -0.0460, -0.0180, -0.0340, -0.0100, -0.00400, 0.118, 0.210, 0.3320, 0.3360, 0.4620, 0.6080, 0.640, 0.700]

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

symbol

```
> DATAL:=[seq([i,m[i]],i=1950..2015,5)];
```

DATAL := [[1950, -0.0920], [1955, -0.0460], [1960, -0.0180], [1965, -0.0340], [1970, -0.0100], [1975, -0.00400], [1980, 0.118], [1985, 0.210], [1990, 0.3320], [1995, 0.3360], [2000, 0.4620], [2005, 0.6080], [2010, 0.640], [2015, 0.700]]

```
> whattype(DATAL);
```

list

```
> DATA:=seq([i,m[i]],i=1950..2015,5);
```

DATA := [1950, -0.0920], [1955, -0.0460], [1960, -0.0180], [1965, -0.0340], [1970, -0.0100], [1975, -0.00400], [1980, 0.118], [1985, 0.210], [1990, 0.3320], [1995, 0.3360], [2000, 0.4620], [2005, 0.6080], [2010, 0.640], [2015, 0.700]

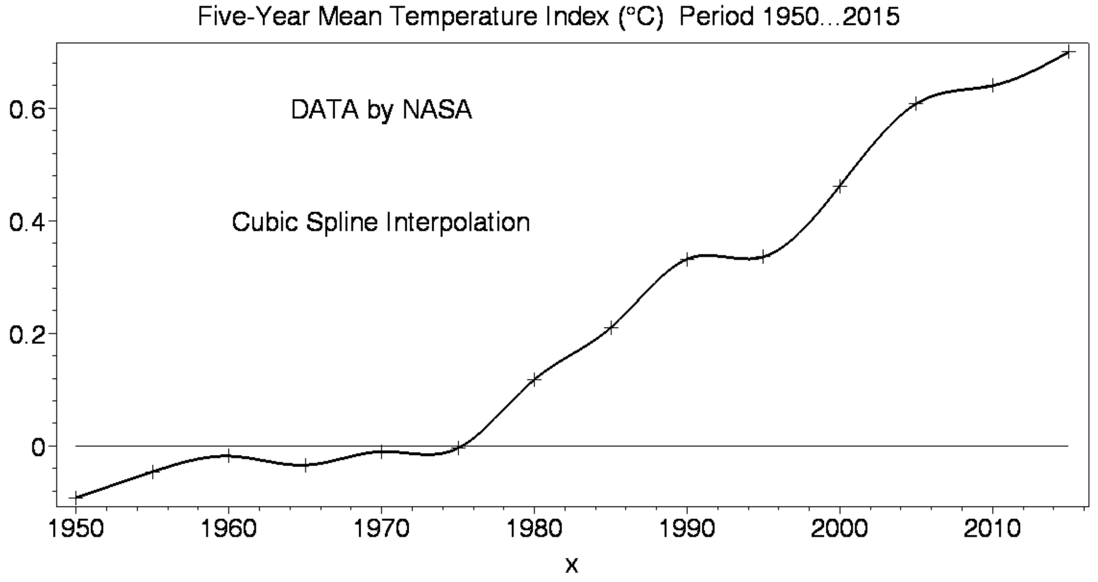
```
> whattype(DATA);
```

exprseq

Cubic Spline:

```
> Sp(x):=Spline([DATA],x,degree=3):
> alias(th=thickness,co=color):
> p[1]:=plot([DATA],x=1950..2015,axes=boxed,th=3,co=black,
  style=point,symbol=cross,symbolsize=30):
> p[2]:=plot(Sp(x),x=1950..2015,th=3,co=black,
  title="Five-Year Mean Temperature Index (°C)  Period
  1950...2015"):
> p[3]:=plot(0,x=1950..2015,th=1,co=black):
```

```
> p[4]:=plots[textplot]({[1970,0.6,`DATA by NASA`],
  [1970,0.4,`Cubic Spline Interpolation`]}):
> plots[display](seq(p[k],k=1..4));
```




```
>
```

Nonlinear Regression: Cubic Polynom

```
> with(Statistics):
> f:=(t,a,b)->-0.092+a*((t-1950)/65)+b*((t-1950)/65)^2+
  (0.792-a-b)*((t-1950)/65)^3;
```

$$f := (t, a, b) \to -0.092 + a\left(\frac{1}{65}t - 30\right) + b\left(\frac{1}{65}t - 30\right)^2 + (0.792 - a - b)\left(\frac{1}{65}t - 30\right)^3$$

```
> y:=unapply(evalf(NonlinearFit(f(t,a,b),X,Y,t)),t);
```

$$y := t \to 5.22835990112961 - 0.00272838969288698\, t + 1.84083434192959\,(0.01538461538\, t - 30.)^2 - 0.871489011891941\,(0.01538461538\, t - 30.)^3$$

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

$$\mathrm{y}(x) := 5.22835990112961 - 0.00272838969288698\, x + 1.84083434192959\,(0.01538461538\, x - 30.)^2 - 0.871489011891941\,(0.01538461538\, x - 30.)^3$$

```
> y(1950):=evalf(subs(x=1950,y(x)));    y(1950)[data]:=-0.0920;
```

$$\mathrm{y}(1950) := -0.0920000000000003$$

$$\mathrm{y}(1950)_{data} := -0.0920$$

```
> y(2015):=evalf(subs(x=2015,y(x)));    y(2015)[data]:=0.700;
```

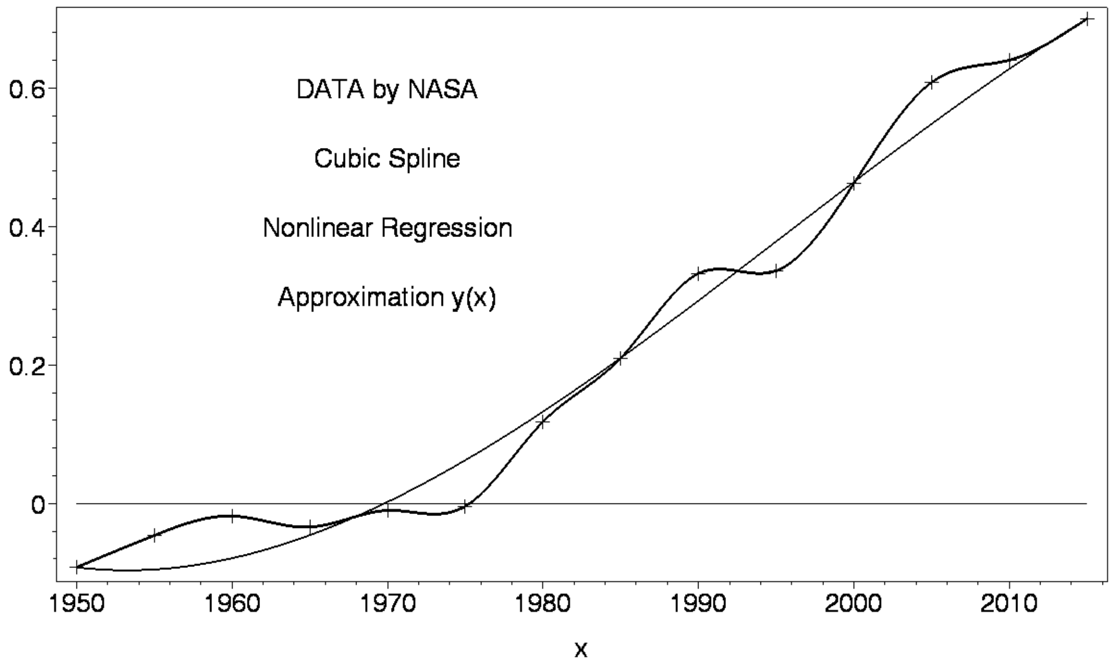
$$\mathrm{y}(2015) := 0.699999989327983$$

$$\mathrm{y}(2015)_{data} := 0.700$$

The Boundary Conditions are fulfilled exactly.

```
> alias(th=thickness,co=color):
> p[1]:=plot([DATA],x=1950..2015,axes=boxed,th=3,co=black,
```

```
style=point,symbol=cross,symbolsize=30):
> p[2]:=plot(Sp(x),x=1950..2015,th=3,co=black,
title="Five-Year Mean Temperature Index (°C)  Period
1950...2015"):
> p[3]:=plot(0,x=1950..2015,th=1,co=black):
> p[4]:=plot(y(x),x=1950..2015,th=2,co=black):
> p[5]:=plots[textplot]({[1970,0.6,`DATA by NASA`],
[1970,0.5,`Cubic Spline`],
[1970,0.4,`Nonlinear Regression`],
[1970,0.3,`Approximation y(x)`]}):
> plots[display](seq(p[k],k=1..5));
```




```
>
```

The L[2] *error norm* between the cubic spline and the approximation y(x) can be expressed as:

```
> L[2]:=
sqrt((1/65)*Int((SPLINE-Z(x))^2,Z=1950..2015))=
evalf(sqrt((1/65)*int((Sp(x)-y(x))^2,x=1950..2015)),4);
```

$$L_2 := \frac{1}{65}\sqrt{65}\sqrt{\int_{1950}^{2015}(SPLINE - Z(x))^2\, dZ} = 0.03621$$

The l[2] *error norm* of the approximation y(x) with respect to the experimental data is given by:

```
> with(linalg):
> for i in [seq(1950..2015,5)] do v[i]:=subs(x=i,y(x))-m[i] od:
> V:=evalf(vector([seq(v[i],i=1950..2015,5)]),4);
```

V := [-0.3469 10^{-15}, -0.04915, -0.06089, -0.01160, 0.01233, 0.06652, 0.01460, 0.0001808, -0.03911, 0.04234, 0.002159, -0.06004, -0.01263, -0.1067 10^{-7}]

```
> l[2]:=
```

```
(1/sqrt(number_of_points))*Norm(V,2)=
evalf((1/sqrt(14))*norm(V,2),4);
```

$$l_2 := \frac{\mathrm{Norm}(V, 2)}{\sqrt{number\_of\_points}} = 0.03600$$

```
>
> # Example ->  exp(...) and tanh(...)  combined:
> h:=
(t,a,b)-> -0.092+
(0.792/(tanh(1)*exp(b)))*exp(b*(t-1950)/65)*tanh((t-1950)/65);
```

$$h := (t, a, b) \to -0.092 + \frac{0.792\, \mathbf{e}^{(1/65\, b\, (t-1950))} \tanh\left(\frac{1}{65} t - 30\right)}{\tanh(1)\, \mathbf{e}^{b}}$$

```
> r:=unapply(evalf(NonlinearFit(h(t,a,b),X,Y,t)),t);
```

$$r := t \to$$
$$-0.092 + 0.328984296303067\, \mathbf{e}^{(0.0177060437280216\, t - 34.5267852696422)} \tanh(0.01538461538\, t - 30.)$$

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

$$\mathrm{r}(x) :=$$
$$-0.092 + 0.328984296303067\, \mathbf{e}^{(0.0177060437280216\, x - 34.5267852696422)} \tanh(0.01538461538\, x - 30.)$$

```
> r(1950):=evalf(subs(x=1950,r(x)));  r(1950)[data]:=-0.0920;
```

$$\mathrm{r}(1950) := -0.0920000032898430$$

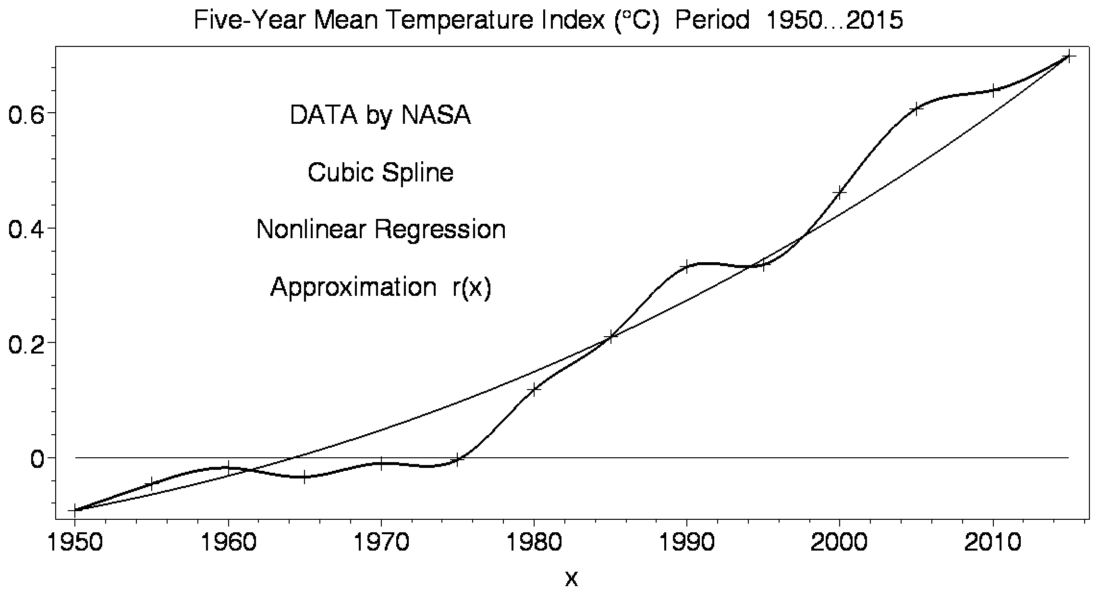
$$\mathrm{r}(1950)_{data} := -0.0920$$

```
> r(2015):=evalf(subs(x=2015,r(x)));  r(2015)[data]:=0.700;
```

$$\mathrm{r}(2015) := 0.699999995279239$$

$$\mathrm{r}(2015)_{data} := 0.700$$

```
> # The boundary conditions are exactly fulfilled.
> alias(th=thickness,co=color):
> p[1]:=plot([DATA],x=1050..2015,axes=boxed,th=3,co=black,
style=point,symbol=cross,symbolsize=30):
> p[2]:=plot(Sp(x),x=1950..2015,th=3,co=black,
title="Five-Year Mean Temperature Index (°C)  Period
1950...2015"):
> p[3]:=plot(0,x=1950...2015,th=1,co=black):
> p[4]:=plot(r(x),x=1950..2015,th=2,co=black):
> p[5]:=plots[textplot]({[1970,0.6,`DATA by NASA`],
[1970,0.5,`Cubic Spline`],
[1970,0.4,`Nonlinear Regression`],
[1970,0.3,`Approximation  r(x)`]}):
> plots[display](seq(p[k],k=1..5));
```



>

The N[2] *error norm* between the cubic spline and the approximation r(x) can be expressed as:

```
> N[2]:=sqrt((1/65)*Int((SPLINE-R(x))^2,x=1950..2015))=
  evalf(sqrt((1/65)*int((Sp(x)-r(x))^2,x=1950..2015)));
```

$$N_2 := \frac{1}{65}\sqrt{65}\sqrt{\int_{1950}^{2015} (SPLINE - \mathrm{R}(x))^2\, dx} = 0.04940573100$$

```
> for i in [seq(1950..2015,5)] do s[i]:=subs(x=i,r(x))-m[i] od:
> S:=evalf(vector([seq(s[i],i=1950..2015,5)]));
```

$S := [-0.328984296303065\ 10^{-8}, -0.0184053718475058, -0.0140553325780236, 0.0392932303190068, 0.0578536888142986, 0.0998181630225897, 0.0313688694424441, -0.00130869094603342, -0.0580118795428445, 0.00949270969804600, -0.0385181108309168, -0.0997129538157974, -0.0396962268990830, -0.462286897651154\ 10^{-8}]$

```
> n[2]:=
  (1/sqrt(number_of_points))*Norm(S,2)=
  evalf((1/sqrt(14))*norm(S,2),4);
```

$$n_2 := \frac{\mathrm{Norm}(S, 2)}{\sqrt{number\_of\_points}} = 0.04842$$

Résumé: The *error norms* relativ to the spline interpolations or alternatively with respect to experimental data have
shown that the selected nonlinear model functions are suitable to fit the given data.

>