Comparison between LAGRANGE and Spline Interpolation

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

In this worksheet a comparison between the *LAGRANGE* and the *Spline Interpolation* is discused based upon eleven interpolating points. In this case we arrive at a LAGRANGE polynomial of degree ten. High-degree polynomials have an oscillatory character and are therefore not ever suitable as interpolation functions.

An alternative approach is given by a spline interpolation - a piecewise approximation. In order to arrive at a ***smooth*** interpolation a cubic spline is often used. In the following it has been shown that high-degree splines are similar to LAGRANGE polynomials. A spline of degree = infinity is identical to the LAGRANGE approximation.

LAGRANGE Interpolation

Given n + 1 distinct points x[k], k = 0,1,...,n and the corresponding values f(x[k]) the LAGRANGE interpolation polynomial is defined as:

```
> restart:
> P(x):=Sum(f(x[k])*L[n,k](x),k=0..n);
```

$$P(x) := \sum_{k=0}^{n} f(x_k)\, L_{n,k}(x)$$

Where for every k = 0,1,...,n we introduce a LAGRANGE basic function:

```
> restart:
> L[n,k](x):=Product((x-x[i])/(x[k]-x[i]),i=0..n), k<>i;
```

$$L_{n,k}(x) := \prod_{i=0}^{n} \frac{x - x_i}{x_k - x_i},\ k \neq i$$

```
> L[n,k](x[i]):=delta[ik]=piecewise(i=k,1, i<>k,0);
```

$$L_{n,k}(x_i) := \delta_{ik} = \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases}$$

Example: Given the following experimental data

```
> restart:
> DATA:=[x[k],y[k]]=[[0,120],[1,139],[2,134],[3,149],
  [4,124],[5,145],[6,118],[7,112],[8,127],[9,125],[10,113]];
```

$DATA := [x_k, y_k] = [[0, 120], [1, 139], [2, 134], [3, 149], [4, 124], [5, 145], [6, 118], [7, 112], [8, 127], [9, 125], [10, 113]]$

```
> L[10,k](x):=Product((x-x[i])/(x[k]-x[i]),i=0..10); k<>i;
```

$$L_{10,k}(x) := \prod_{i=0}^{10} \frac{x - x_i}{x_k - x_i}$$

$$k \neq i$$

```
> L[10,k](x):=value(%%);
```

$L_{10,k}(x) := (x - x_0)(x - x_1)(x - x_2)(x - x_3)(x - x_4)(x - x_5)(x - x_6)(x - x_7)(x - x_8)(x - x_9)(x - x_{10}) \big/ ((x_k - x_0)(x_k - x_1)(x_k - x_2)(x_k - x_3)(x_k - x_4)(x_k - x_5)(x_k - x_6)(x_k - x_7)(x_k - x_8)(x_k - x_9)(x_k - x_{10}))$

```
> L[10,k](x[k]):=1;
```

$$L_{10,k}(x_k) := 1$$

example for k = 4

```
> L[10,4](x):=(%%)*(x[k]-x[4])/(x-x[4]);
```

$L_{10,4}(x) :=$

$$\frac{(x - x_0)(x - x_1)(x - x_2)(x - x_3)(x - x_5)(x - x_6)(x - x_7)(x - x_8)(x - x_9)(x - x_{10})}{(x_k - x_0)(x_k - x_1)(x_k - x_2)(x_k - x_3)(x_k - x_5)(x_k - x_6)(x_k - x_7)(x_k - x_8)(x_k - x_9)(x_k - x_{10})}$$

```
> L[10,4](x):=
  subs({x[k]=4,seq(x[k]=k,k=0..3),seq(x[k]=k,k=5..10)},%);
```

$$L_{10,4}(x) := \frac{x(x-1)(x-2)(x-3)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)}{17280}$$

```
> L[10,4](4):=subs(x=4,%);
```

$$L_{10,4}(4) := 1$$

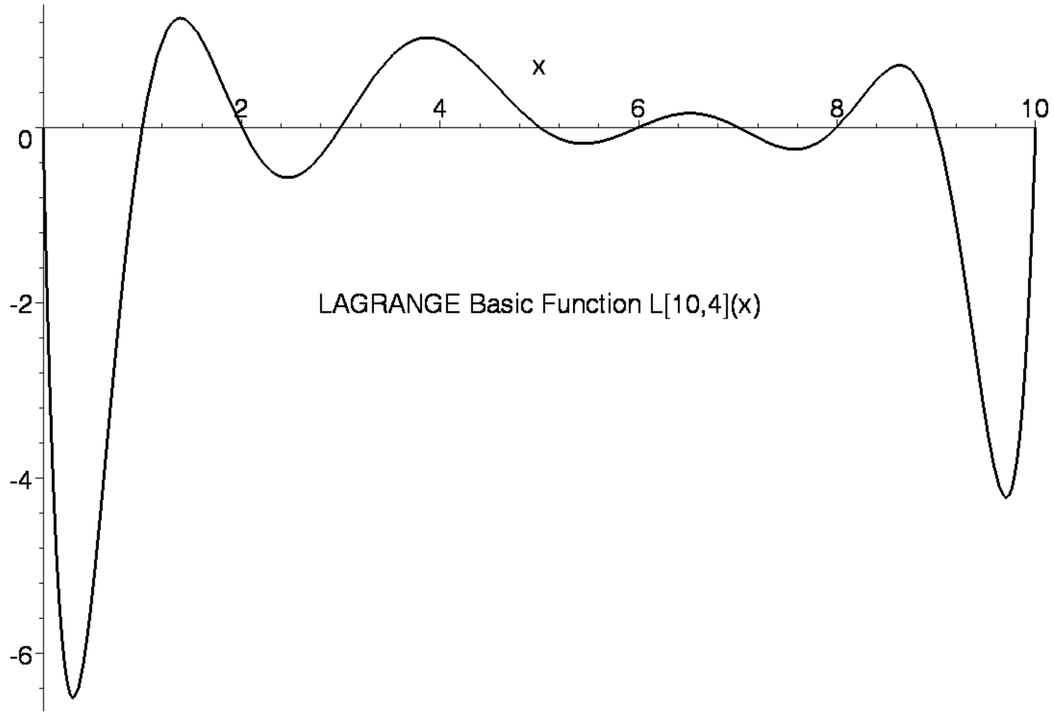
```
> L[10,4](x):=expand(%%);
```

$$L_{10,4}(x) := \frac{1}{17280}x^{10} - \frac{17}{5760}x^9 + \frac{31}{480}x^8 - \frac{2281}{2880}x^7 + \frac{34343}{5760}x^6 - \frac{163313}{5760}x^5 + \frac{728587}{8640}x^4 - \frac{71689}{480}x^3 + \frac{6751}{48}x^2 - \frac{105}{2}x$$

Graphical representation of this LAGRANGE function

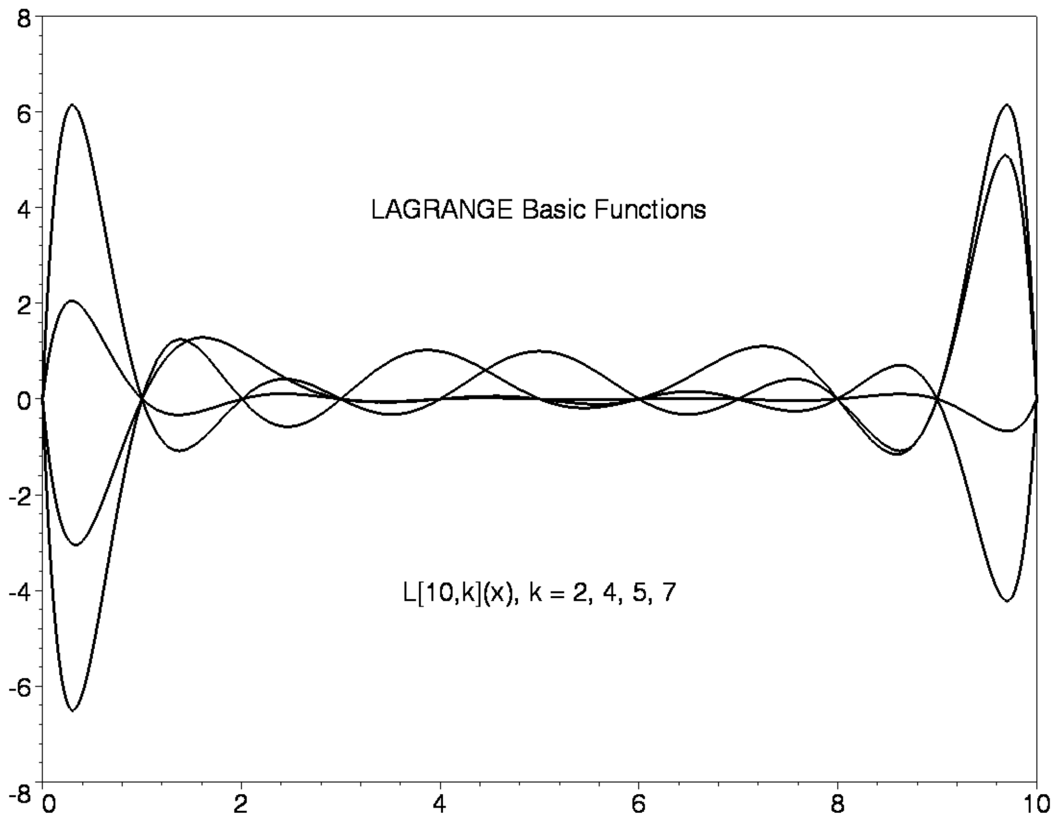
```
> alias(th=thickness,co=color):
> p[1]:=plot(L[10,4](x),x=0..10,th=3,co=black):
```

```
> p[2]:=plots[textplot]([5,-2,`LAGRANGE Basic Function
  L[10,4](x)`]):
> plots[display](seq(p[k],k=1..2));
```



Further LAGRANGE basic functions are illustrated in the next Figure.

```
> for i from 0 to 10 do
  L[10,i](x):=L[10,k](x)*(x[k]-x[i])/(x-x[i]) od:
> # These functions can be printed, if the doloop
  is ending with a semicolon. A colon can be used
  instead of a semicolon if we do not want to see
  the output,e.g. if you do not need to look at an
  intermediate result or if an output will be very
  large.
> # With values x[k],k=0..10 we find:
> for i from 0 to 10 do
  L[10,i](x):=subs({x[k]=x[i]},{seq(x[k]=k,k=0..10)},
  L[10,i](x)) od:
> # furthermore
> for i from 0 to 10 do L[10,i](x):=expand(L[10,i](x)) od:
> # graphical representation:
> alias(th=thickness,co=color,sc=scaling):
> p[1]:=plot({L[10,2](x),L[10,4](x),L[10,5](x),L[10,7](x)},
  x=0..10,-8..8,th=3,co=black,axes=boxed):
> p[2]:=plots[textplot]({[5,4,`LAGRANGE Basic Functions`],
  [5,-4,`L[10,k](x), k = 2, 4, 5, 7`]}):
> plots[display](seq(p[k],k=1..2));
```



With these *LAGRANGE functions* and the values f(x[k]) = y[k] we arríve at the following *interpolation polynom* of **degree ten**:

```
> P[10](x):=Sum(y[kappa]*L[10,kappa](x),kappa=0..10);
```

$$P_{10}(x) := \sum_{\kappa=0}^{10} y_\kappa L_{10,\kappa}(x)$$

```
> # We read from the data-set
> with(linalg):
```

Warning, the protected names norm and trace have been redefined and unprotected

```
> y[k]:=
  matrix(1,11,[120,139,134,149,124,145,118,112,127,125,113]);
```

$$y_k := [120 \quad 139 \quad 134 \quad 149 \quad 124 \quad 145 \quad 118 \quad 112 \quad 127 \quad 125 \quad 113]$$

```
> mean_value:=127.8;
```

$$mean\_value := 127.8$$

```
> # The LAGRANGE functions can be posted in a matrix
  with 11 rows and one column,i.e. a columnvector:
> L[10,k](x):=matrix(11,1,[seq(L[10,k](x),k=0..10)]):
> # Thus, we find the polynomial P[10](x) by the matrix product:
> P[10](x):=multiply(y[k],L[10,k](x));
```

$P_{10}(x) :=$

$$\left[120 + \frac{5001539}{2520} x - \frac{37488433}{7200} x^2 + \frac{496694509}{90720} x^3 + \frac{8922667}{8640} x^5 - \frac{18774473}{86400} x^6 + \frac{218563}{7560} x^7\right.$$

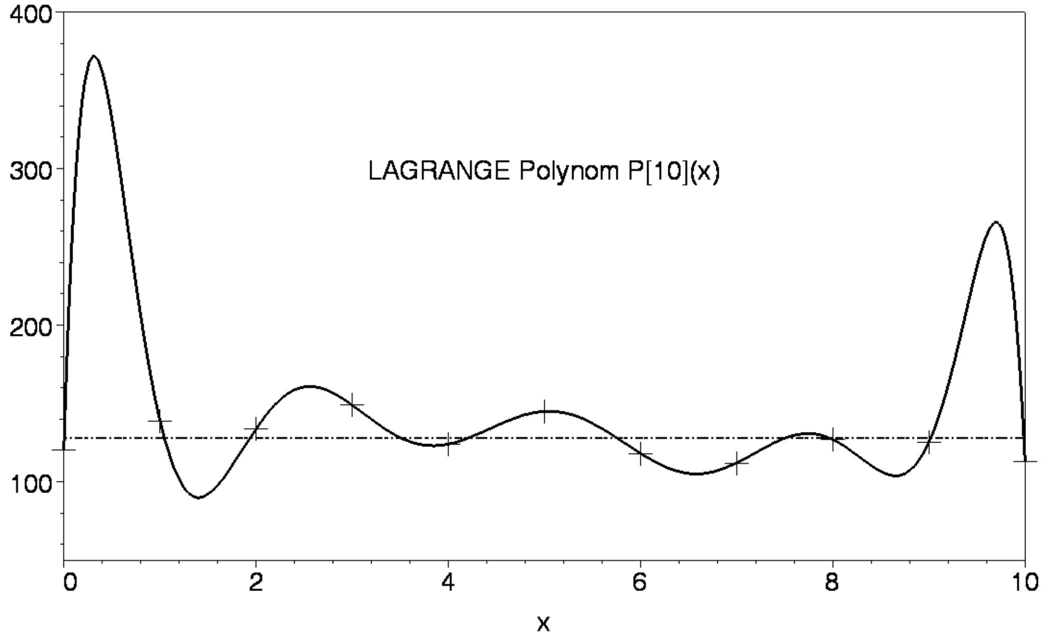
$$-\frac{40811}{17280}x^8+\frac{3923}{36288}x^9-\frac{3851}{1814400}x^{10}-\frac{1116259589}{362880}x^4\Big]$$

```
> P[10](x):=120+(5001539/2520)*x-(37488433/7200)*x^2+
  (496694509/90720)*x^3-(1116259589/362880)*x^4+
  (8922667/8640)*x^5-(18774473/86400)*x^6+(218563/7560)*x^7-
  (40811/17280)*x^8+(3923/36288)*x^9-(3851/1814400)*x^10;
```

$$P_{10}(x) := 120+\frac{5001539}{2520}x-\frac{37488433}{7200}x^2+\frac{496694509}{90720}x^3+\frac{8922667}{8640}x^5-\frac{18774473}{86400}x^6$$
$$+\frac{218563}{7560}x^7-\frac{40811}{17280}x^8+\frac{3923}{36288}x^9-\frac{3851}{1814400}x^{10}-\frac{1116259589}{362880}x^4$$

This polynomial together with the experimental data is represented in the next Figure.

```
> alias(th=thickness,co=color):
> p[1]:=plot(P[10](x),x=0..10,th=3,co=black,axes=boxed):
> p[2]:=plot(rhs(DATA),x=0..10,50..400,ytickmarks=4,
  style=point,symbol=cross,symbolsize=50,th=3,co=black):
> p[3]:=plot(127.8,x=0..10,linestyle=4,th=2,co=black):
> p[4]:=plots[textplot]([5,300,`LAGRANGE Polynom P[10](x)`]):
> plots[display](seq(p[k],k=1..4));
```



The dotted line in this Figure characterizes the mean value 127.8 of the experimental data. Because of its oscillatory character the polynomial P[10](x) is less suitable as an interpolation function. In order to arrive at a ***smooth*** interpolation a ***cubic spline*** should be prefered.

Cubic Spline in Comparison with the LAGRANGE Interpolation Polynom P[10](x)

It is very comfortable to arrive at spline functions by utilizing the Maple progamm *Curve*

Fitting as we can see in the following.

```
> restart:  with(stats):  with(CurveFitting):
> data:=[0,120],[1,139],[2,134],[3,149],[4,124],
  [5,145],[6,118],[7,112],[8,127],[9,125],[10,113];
```

$data := [0, 120], [1, 139], [2, 134], [3, 149], [4, 124], [5, 145], [6, 118], [7, 112], [8, 127], [9, 125], [10, 113]$

```
> datay:=[120,139,134,149,124,145,118,112,127,125,113];
```

$$datay := [120, 139, 134, 149, 124, 145, 118, 112, 127, 125, 113]$$

```
> mean_value:=evalf(describe[mean](datay),4);
```

$$mean\_value := 127.8$$

```
> # cubic spline S[3](x)
> S[3](x):=Spline([data],x,degree=3);
```

$$S_3(x) := \begin{cases} 120 + \frac{4228519}{151316}x - \frac{1353515}{151316}x^3 & x < 1 \\ \frac{6834207}{75658} + \frac{931423}{7964}x - \frac{6734259}{75658}x^2 + \frac{3135991}{151316}x^3 & x < 2 \\ \frac{37508351}{75658} - \frac{74325395}{151316}x + \frac{16271349}{75658}x^2 - \frac{238555}{7964}x^3 & x < 3 \\ -\frac{4706209}{3439} + \frac{207764503}{151316}x - \frac{809043}{1991}x^2 + \frac{5915229}{151316}x^3 & x < 4 \\ \frac{140718485}{37829} - \frac{369695849}{151316}x + \frac{20719455}{37829}x^2 - \frac{6115195}{151316}x^3 & x < 5 \\ -\frac{370878155}{75658} + \frac{413082301}{151316}x - \frac{36838905}{75658}x^2 + \frac{4321847}{151316}x^3 & x < 6 \\ \frac{174871333}{75658} - \frac{132667187}{151316}x + \frac{8640219}{75658}x^2 - \frac{731389}{151316}x^3 & x < 7 \\ \frac{144451013}{37829} - \frac{230407781}{151316}x + \frac{7810845}{37829}x^2 - \frac{73489}{7964}x^3 & x < 8 \\ -\frac{106791995}{37829} + \frac{7708249}{7964}x - \frac{360561}{3439}x^2 + \frac{566545}{151316}x^3 & x < 9 \\ -\frac{6916961}{6878} + \frac{54791785}{151316}x - \frac{2839845}{75658}x^2 + \frac{189323}{151316}x^3 & otherwise \end{cases}$$

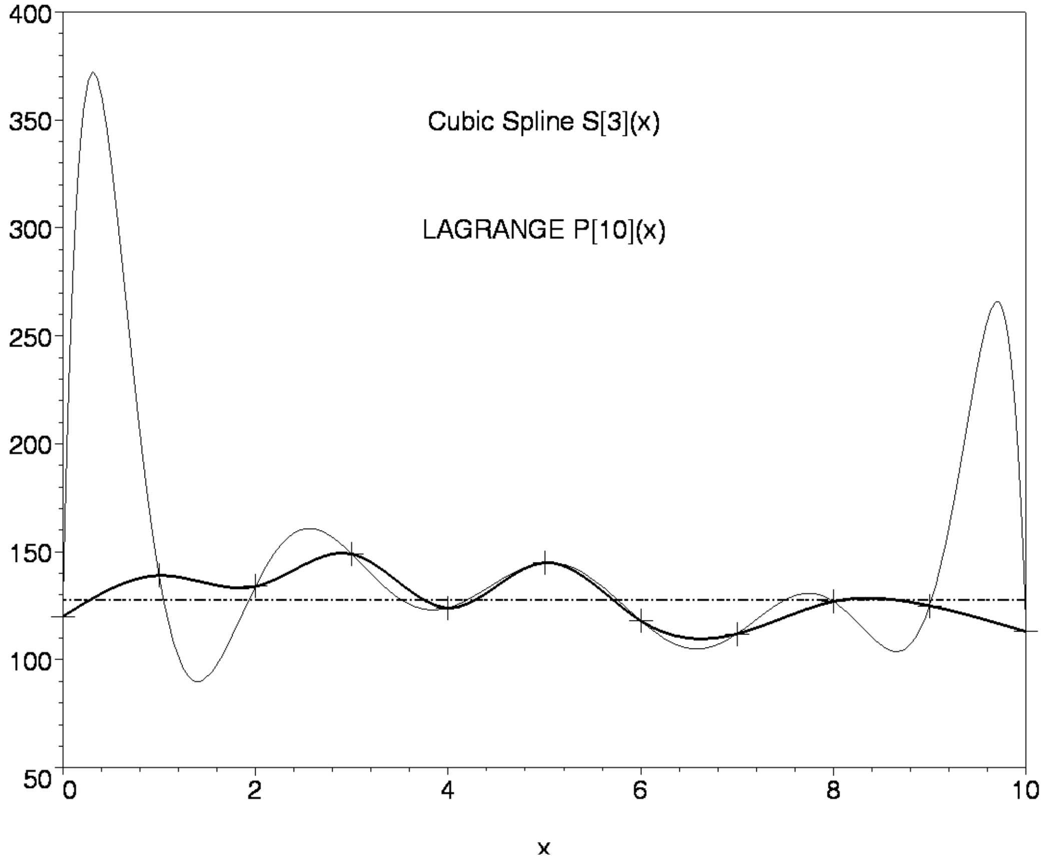
The above LAGRANGE interpolation polynom is written as:

```
> P[10](x):=120+(5001539/2520)*x-(37488433/7200)*x^2+
  (496694509/90720)*x^3-(1116259589/362880)*x^4+
  (8922667/8640)*x^5-(18774473/86400)*x^6+
  (218563/7560)*x^7-(40811/17280)*x^8+(3923/36288)*x^9-
  (3851/1814400)*x^10;
```

$$P_{10}(x) := 120 + \frac{5001539}{2520}x - \frac{37488433}{7200}x^2 + \frac{496694509}{90720}x^3 - \frac{1116259589}{362880}x^4 + \frac{8922667}{8640}x^5$$

$$-\frac{18774473}{86400}x^6+\frac{218563}{7560}x^7-\frac{40811}{17280}x^8+\frac{3923}{36288}x^9-\frac{3851}{1814400}x^{10}$$

```
> # The cubic spline, the LAGRANGE polynom, and the data values
> # are represented in the following Figure.
> alias(th=thickness,co=color):
> p[1]:=plot(S[3](x),x=0..10,50..400,th=3,co=black):
> p[2]:=plot(P[10](x),x=0..10,th=1,co=black,axes=boxed):
> p[3]:=plot([data],style=point,symbol=cross,
  symbolsize=50,th=3,co=black):
> p[4]:=plot(127.8,x=0..10,linestyle=4,th=2,co=black):
> p[5]:=plots[textplot]({[5,350,`Cubic Spline S[3](x)`],
  [5,300,`LAGRANGE P[10](x)`]}):
> plots[display](seq(p[k],k=1..5));
```



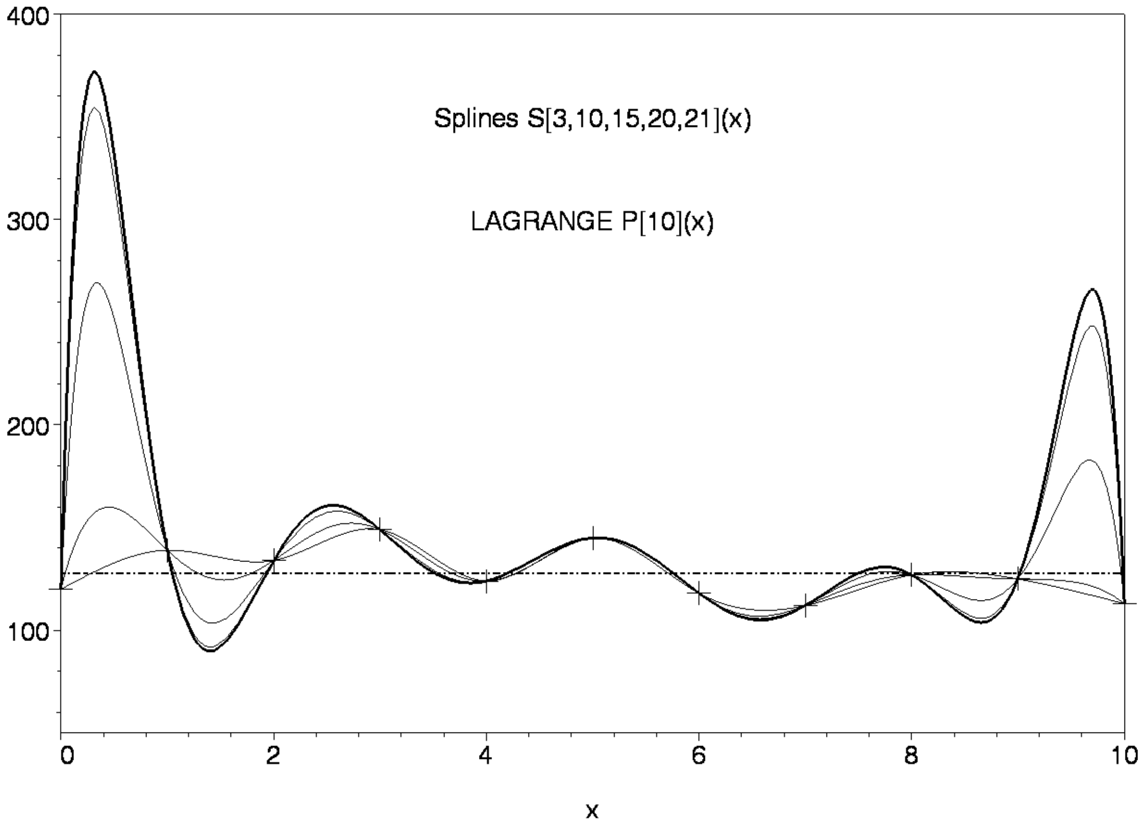
The dotted line characterizes the mean value 127.8 of the experimental data. This Figure shows the smooth cubic spline and the more or less oscillating LAGRANGE polynomial, which is not suitable as an intepolating function.

High-Degree Splines S[n](x) in Comparison with the LAGRANGE Polynom P[10](x)

As mentioned obove spline functions of arbitrary degree can be easyly found by utilizing the MAPLE program *Curve Fitting*.

```
> for i in [3,10,15,20,21] do
```

```
  S[i](x):=Spline([data],x,degree=i) od:
> S[3](x):=Spline([data],x,degree=3):
> S[10](x):=Spline([data],x,degree=10):
> S[15](x):=Spline([data],x,degree=15):
> S[20](x):=Spline([data],x,degree=20):
> S[21](x):=Spline([data],x,degree=21):
>
> # graphical representation in the next Figure
> alias(th=thickness,co=color):
> p[1]:=plot({S[3](x),S[10](x),S[15](x),S[20](x),S[21](x)},
  x=0..10,50..400,th=1,co=black):
> p[2]:=plot(P[10](x),x=0..10,th=3,co=black):
> p[3]:=plot([data],style=point,symbol=cross,symbolsize=50,
  th=3,co=black,axes=boxed,ytickmarks=4):
> p[4]:=plot(127.8,x=0..10,linestyle=4,th=2,co=black):
> p[5]:=plots[textplot]({[5,350,`Splines S[3,10,15,20,21](x)`],
  [5,300,`LAGRANGE P[10](x)`]}):
> plots[display](seq(p[k],k=1..5));
```



This Figure illustrates that by increasing the degree the splines are more and more similar to the LAGRANGE polynom (thick line). A spline of degree = infinity is identical to the LAGRANGE approximation as can be proved by introducing the **L-two error norm:**

```
> L[2]:=sqrt((1/10)*Int((P[10](xi)-S[n](xi))^2,xi=0..10));
```

$$L_2 := \frac{1}{10}\sqrt{10}\sqrt{\int_0^{10} (P_{10}(\xi) - S_n(\xi))^2 \, d\xi}$$

```
> for i in [3,10,15,20,21] do L[2][i]:=
  evalf(sqrt((1/10)*int((P[10](x)-S[i](x))^2,x=0..10)),4) od;
```

$$L_{2_3} := 61.64$$

$$L_{2_{10}} := 55.30$$

$$L_{2_{15}} := 27.92$$

$$L_{2_{20}} := 5.206$$

$$L_{2_{21}} := 0.$$

```
> L[2][infinity]:=
  Limit(sqrt((1/10)*Int((P[10](xi)-S[n](xi))^2,x=0..10)),
  n=infinity)=0;
```

$$L_{2_\infty} := \lim_{n \to \infty} \frac{1}{10}\sqrt{10}\sqrt{\int_0^{10} (P_{10}(\xi) - S_n(\xi))^2 \, dx} = 0$$

We see for degree = 21 the spline curve is already identical to the LAGRANGE polynom.

Summary

This paper is concerned with both the *LAGRANGE* and the *spline interpolation.*
Based upon a given set of eleven experimental data we arrive at a LAGRANGE interpolation polynom of degree ten. Because of its oscillation property the LAGRANGE polynomial is not suitable to interpolate the given experimental data. Thus, the *spline interpolation* has been discused as an alternative approach. Especially, the common *cubic spline* leads to a ***smooth*** interpolation.

Furthermore, it has been illustrated that high-degree splines are approaching to LAGRANGE polynomials. A spline of degree = infinity is identical to the LAGRANGE approximation.


```
>
```