

Glättung und Darstellung von Messwertkurven (Splines)

Inhalt

1. Einlesen und Sortieren der Messwerte	2
2. Splinefunktionen berechnen und ausgeben	4
3. Button anlegen und einbinden	6
4. Diagramm anlegen	7
5. Sicherung	10
6. Nachschlag	11

Die Fähigkeit zur schnellen Erzeugung von Diagrammen teilt Excel mit allen anderen Office-Anwendungen. Wie bei diesen stehen Diagramme im Vordergrund, wie sie Journalisten oder Politiker lieben (sollen). Der Techniker oder Wissenschaftler kommt zu kurz weg. Da aber Excel bevorzugtes Werkzeug für Messwertaufnahme und -verarbeitung ist, soll die folgende Arbeitsanweisung die Grafikfähigkeiten demonstrieren, denen mittels Makroprogrammierung etwas auf die Sprünge geholfen wird.

Um das Vorhaben genau zu beschreiben, es soll zu einer Reihe von Messwertpaaren ein Diagramm erzeugt werden, das die Messpunkte mittels einer Glättungsfunktion verbindet. Diese Glättungsfunktion wird durch kubische Splinefunktionen erzeugt und sichert neben gutem Aussehen die Differenzierbarkeit der Messwertfunktion.

Achtung: Wer Splines benutzt, sollte sich zuvor darüber informieren, was er tut ! Der Autor lehnt jegliche Verantwortung für missbräuchliche Interpretation der Ergebnisse ab!

Die Einzelaufgaben sind

- Einlesen und Sortieren der Messwerte,
- Berechnung der Splinefunktionen,
- Ausgabe der berechneten Glättungskurve,
- Diagramm mit Messpunkten und Glättungskurve,
- Aufruf der Routine mit einem Button,
- Schutz der Tabelle.

1. Einlesen und Sortieren der Messwerte

Das Eintragen oder Einfügen einer Reihe von Messwertpaaren bereitet keine Probleme. Eine solche Messwertreihe muss für die Verarbeitung mittels vba (= "Visual Basic for Applications") in Variable eingelesen werden. Die folgende Spline-Berechnung benötigt die Messwertpaare mit aufsteigender Reihenfolge der x-Werte. Da man sich auf Pisa-geschädigte User diesbezüglich kaum verlassen kann -Quatsch, weil die Sortierung eine zusätzliche Last ist, die man dem teuren Computer getrost aufbürden kann, sollen die Messwertpaare vorsorglich sortiert werden.

Zunächst sollen die Messwertpaare in die Tabelle eingetragen werden. Keine schlechte Idee, dafür einen speziellen Platz auf der Tabelle zu reservieren. So etwa könnte dies aussehen:

	A	B	C	D
1		Spline-Berechnung mit Darstellung		
2				
3		Meßpunkt	x	y
4		1	3,4	4
5		2	4,5	6
6		3	7,22	8
7		4	10	7
8		5	12,3	3
9		6	13,5	0
10		7	14,8	-0,5
11		8	17	6
12		9	20	12
13		10	25	9
14				
15				

Wenn die Wertepaare da so schön stehen, können sie für die interne Bearbeitung übernommen werden. Das geht etwa so:

In den vba-Modus wechseln: <Extras/><Makro>/<Visual Basic-Editor>, es erscheint die "Integrierte Entwicklungsumgebung" (IDE) in der man mit <Einfügen>/<Modul> eine neuen - was wohl? - einen Modul anlegt. Dies ist ein Behälter für Makros. Das hier einzige wird "glaetten" heißen. Die folgende Tabelle zeigt den zu schreibenden Code:

```
Sub glaetten()  
Dim ROff%, COff%, n%  
Dim x(25) As Single, y(25) As Single  
'Linke obere Ecke des Datenbereiches ist C4  
    ROff = 4  
    COff = 3  
'Sortieren  
    Range("C4:D28").Select  
    Selection.Sort Key1:=Range("C4:D28"), Order1:=xlAscending, Header:=xlGuess  
'Wertepaare einlesen  
    n = 0  
    While Cells(ROff + n, COff) <> Empty  
        n = n + 1  
        x(n) = Cells(ROff + n - 1, COff).Value  
        y(n) = Cells(ROff + n - 1, COff + 1).Value  
    Wend  
End Sub
```

Was haben wir getan?

Ein bisschen dimensioniert, die Festlegung auf 25 Messwerte maximal ist willkürlich. Diese Wahl ist mit der

Reservierung des Eingabefeldes verknüpft und kann mit diesem zusammen leicht verändert werden. Es folgt die Sortierung. Wie kommt man auf diesen Befehl, bzw. die erforderlichen Anhängsel? Einfach mal dem Makrorecorder einsetzen, dann das Makro ansehen. Überflüssigen Code löschen. Den Rest stehen lassen. Danach folgt die Übernahme der Werte in die Variablen $x(i)$ und $y(i)$, die so lange fortgesetzt wird, bis es keine mehr gibt. An dieser Stelle wird es später, nachdem Fehlermeldungen aufgetreten sind, noch eine Nachbesserung geben. Jetzt könnte man die Glättungsroutine starten. Die haben wir aber nicht. Also was?

2. Splinefunktionen berechnen und ausgeben

Zur Theorie und zum Algorithmus hier nur so wenig: Es werden für jeweils drei Messwerte Polynome dritten Grades so angepasst, dass sich über die ganze Kurve gesehen ein "glatter" Kurvenzug ergibt.

Wir nutzen den folgenden Algorithmus:

```
Sub Splines(n As Integer, x() As Single, y() As Single)
'Kubische Splines f(x), fixiert an den Stützstellen x(1) < x(2) < ... < x(n),
'wobei f"(x(1)) = f"(x(n)) = 0
'Eingaben: n, x(1), x(2), ..., x(n), y(1), y(2), ..., y(n)
'Koeffizienten werden in a(j), b(j), c(j), d(j) for j = 1, 2, ..., n abgelegt
'es gilt: f(x) = a(j) + b(j) * (x - x(j)) + c(j) * (x - x(j))^2 + d(j) * (x - x(j))^3,
für x(j) <= x < x(j + 1)
Dim A(100) As Single, B(100) As Single, C(100) As Single, D(100) As Single
Dim t(100) As Single, Ta(100) As Single, Tl(100) As Single, Tu(100) As Single, Tz(100)
As Single
Dim i%, j%, m%
  For i = 1 To n
    A(i) = y(i)
  Next
  m = n - 1
'Tridiagonale Matrix erstellen und lösen
  For i = 1 To m
    t(i) = x(i + 1) - x(i)
  Next
  For i = 2 To m
    Ta(i) = 3 * (A(i + 1) * t(i - 1) - A(i) * (x(i + 1) - x(i - 1)) + A(i - 1) *
t(i)) / (t(i) * t(i - 1))
  Next
  Tl(1) = 1: Tu(1) = 0: Tz(1) = 0
  For i = 2 To m
    Tl(i) = 2 * (x(i + 1) - x(i - 1)) - t(i - 1) * Tu(i - 1)
    Tu(i) = t(i) / Tl(i)
    Tz(i) = (Ta(i) - t(i - 1) * Tz(i - 1)) / Tl(i)
  Next
  Tl(n) = 1: Tz(n) = 0: C(n) = Tz(n)
  For i = 1 To m
    j = n - i
    C(j) = Tz(j) - Tu(j) * C(j + 1)
    B(j) = (A(j + 1) - A(j)) / t(j) - t(j) * (C(j + 1) + 2 * C(j)) / 3
    D(j) = (C(j + 1) - C(j)) / (3 * t(j))
  Next
  ausgabe x, A, B, C, D, n
End Sub
```

Richtig erkannt, es muss noch einen Prozedur namens "ausgabe" geben, die nämlich Punktweise die geglättete Kurve berechnet. Kommt gleich. Sie will wissen, wieviel Schritte wir errechnen sollen. Das teilen wir in einer Konstanten mit, die gleich hinter "Option Explicit" festgelegt wird (Const schritte = 100), und vielleicht später auf dem Tabellenblatt wählbar gemacht werden kann.

Die Ausgabe erfolgt in einen Tabellenbereich, der hier dicht neben dem Eingabebereich liegt, damit man ihn in der Programmierphase im Blick behält. Später könnte er auch versteckt werden.

```
Sub ausgabe(x() As Single, A() As Single, B() As Single, C() As Single, D() As Single,
n%)
Dim i%, m%, dx As String
Dim Steps As Single
Dim aROff%, aCOff%
'Offsets der Ausgabespalte
  aROff = 4
  aCOff = 12
'Vorbereitungen
  Steps = (x(n) - x(1)) / schritte
  m = 0
  i = 1
  dx = x(1)
'Fortsetzung nächste Seite!
```

```

'eigentliche Rechenschleife
Do
  Do
    Cells(aROff + m, aCOff).Value = CSng(dx)
    Cells(aROff + m, aCOff + 1).Value = CSng(A(i) + B(i) * (dx - x(i)) + C(i) *
(dx - x(i)) ^ 2 + D(i) * (dx - x(i)) ^ 3)
    dx = dx + Steps
    m = m + 1
  Loop Until dx > x(i + 1)
  i = i + 1
Loop Until i > n
End Sub

```

Aufmerksame Betrachtung fördert Merkwürdiges zu Tage: Wieso müssen die berechneten x-, und y-Werte hier in Single-Zahlen umgewandelt werden, wenn es doch schon welche waren? Antwort: Weil Excel, was Formate angeht, durchaus seinen eigenen Kopf hat und hier meine dx-Werte, trotz formatierter Zielzellen, partout als Text verstehen wollte, was dann die Darstellung in einem Diagramm zuverlässig verhindert.

Nachdem nun die Mathematik "steht", muss noch der Aufruf bewerkstelligt (schönes Wort!) werden. Dazu muss in das Makro "glaetten" genau zwischen "Wend" und "End Sub" noch der Aufruf mit Argumenten eingetragen werden. Er lautet hier: "Splines n, x, y"

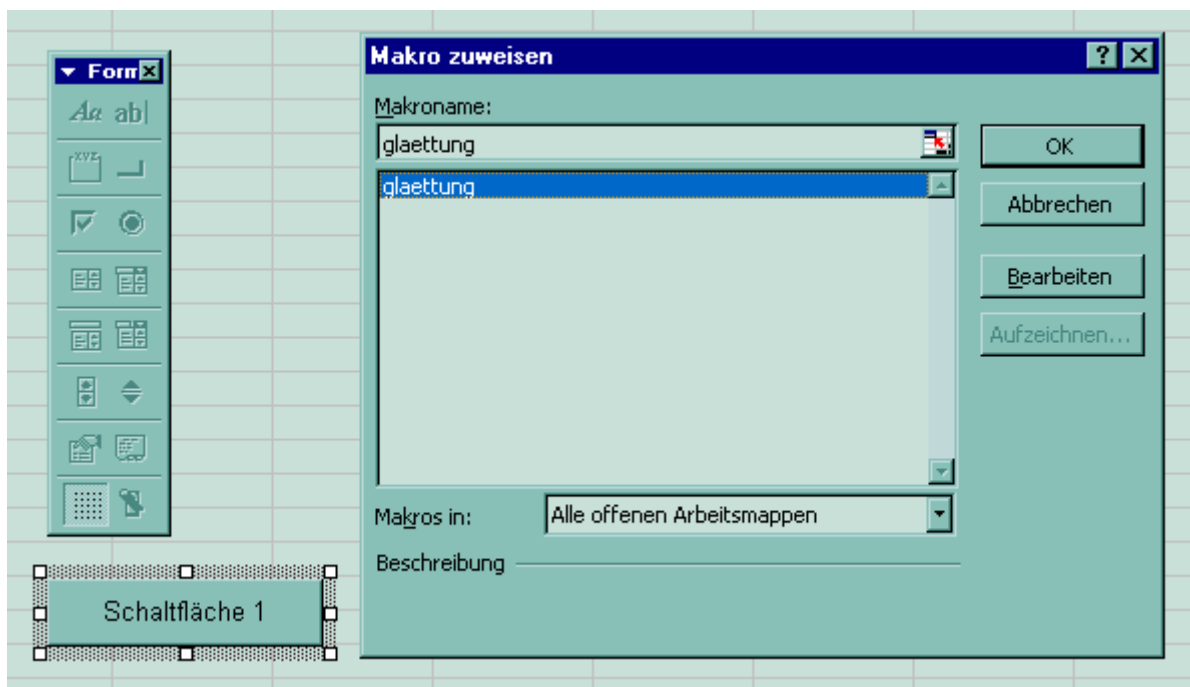
So, jetzt gäbe es nach Ausführung des Makros "glaetten" eine Doppelspalte mit 100 berechneten Kurvenpunkten. Hier der (Teil-)Beweis:

3,400	4,000
3,616	4,421
3,832	4,836
4,048	5,237
4,264	5,617
4,480	5,969
4,696	6,288
4,912	6,573
5,128	6,827
5,344	7,050
5,560	7,246
5,776	7,415
5,992	7,559
6,208	7,681
6,424	7,781
6,640	7,863
6,856	7,927
7,072	7,975
7,288	8,009
7,504	8,031
7,720	8,038


Nun könnte man zum Darstellen schreiten, erhöbe sich nicht zuvor die Frage, wie hat er eigentlich das Makro gestartet? Natürlich mit einem Button, den es schnell nachzuholen gilt.

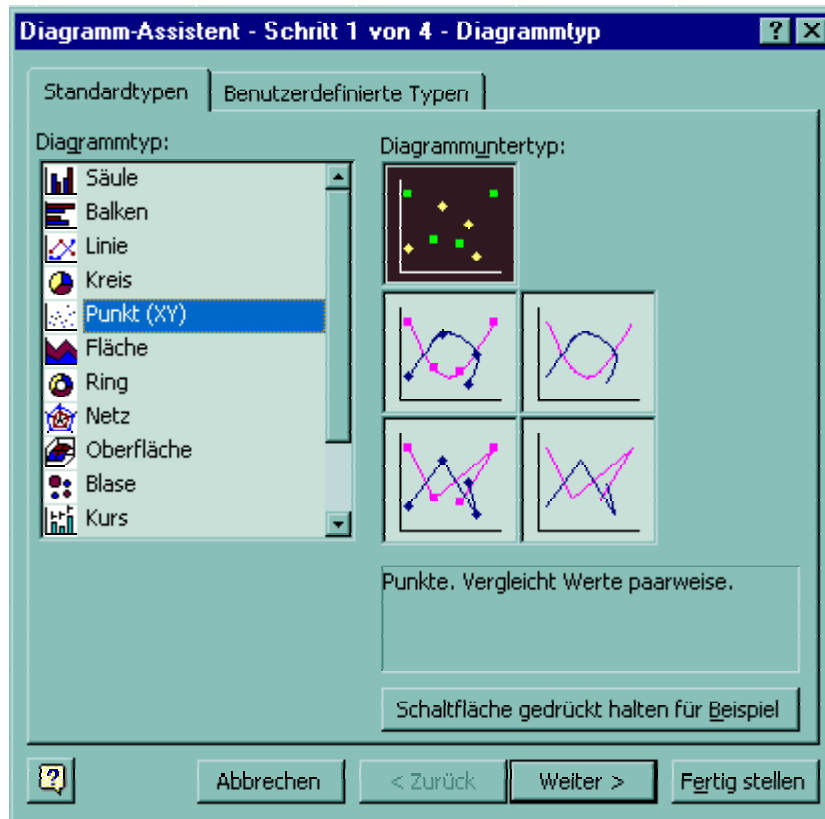
3. Button anlegen und einbinden

Es wird ein **Button** auf der Tabelle benötigt. Den holt man, indem man mit `<Ansicht>/<Symbolleisten>/<Formular>` einen Werkzeugkasten holt. Daraus übernimmt man einen Button, den man mit der Maus auf der Tabelle platziert. Dabei öffnet sich das Fenster mit der Übersicht vorhandener Makros, interessanterweise erscheinen hier nur die Subs, die keine Argumente erwarten. Das zu klickende Makro heißt "glaetten" und schon sind wir fast fertig. Fast, weil die Taste natürlich nicht "Schaltfläche 1" heißen soll. Deshalb, ach, wie lieben wir es: Rechte Maustaste auf den Button, "Steuerelement formatieren...", leider keine Möglichkeit, die Aufschrift zu ändern, deshalb "Abbrechen", und - oh Wunder - schon haben wir den Cursor im Button, um endlich eine poetischere Bezeichnung eintragen zu können.

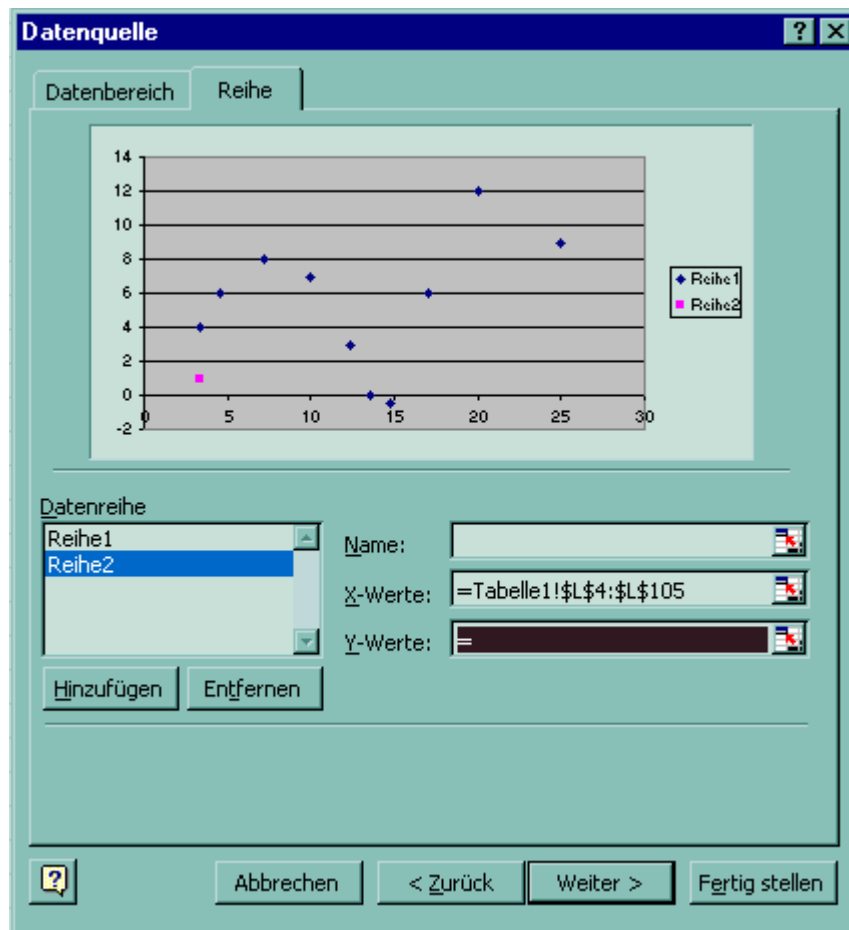


4. Diagramm anlegen

Jetzt kehren wir zur reinen Excel-Routine zurück, Nach Markierung des gesamten Eingabebereiches der Messwerte (also aller 25 Reihen) wird der Diagramm-Button  gewählt. Im Assistenten erfolgt die Wahl des Punktdiagramms:



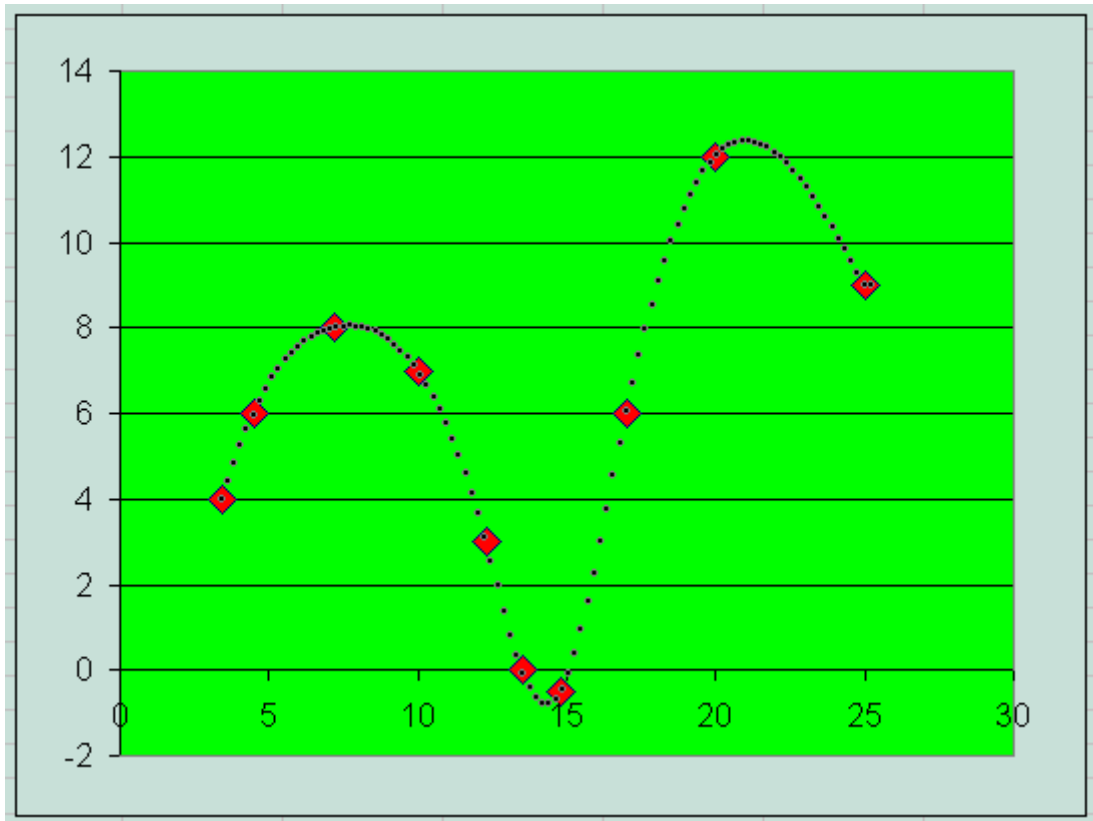
dann "Weiter". Auf der folgenden Karte erscheint schon ein Diagramm der Messpunkte, um das man sich nicht kümmern muß. Stattdessen die zweite, zunächst verborgene Karte "Reihe" anklicken. Auf ihr hat man Gelegenheit, eine neue Datenreihe hinzuzufügen. Dazu müssen die Bereiche für die x- und die y-Werte eingetragen werden. Das geschieht durch Markierung der betreffenden Spalte mit der Maus.



Danach erscheint in anderer Farbe die zweite, dichtere Punktreihe im Diagramm. Jetzt kann man "Fertig stellen".

Das wär es gewesen, wenn man jetzt noch die Messpunkte erkennen könnte. Ist aber nicht. Deshalb ist nun etwas Formatierung erforderlich.

Mit rechter Maustaste auf die Punktfolge klicken und "Datenreihen formatieren..." wählen. Selbstverständlich erhält man Zugang nicht zu den Datenreihen, sondern nur zur Reihe 2. Der sollte man möglichst kleine, farblich etwas unauffällige Punkte zuordnen. Danach hat man die Chance, die nun wieder sichtbaren größeren Punkte der Reihe1 mit der Maus zu erwischen. Gleich wenn das gelungen ist, kann man auch diese Punkte nach Geschmack zurichten.



Was nun noch zu tun bleibt, ist die Sicherung des Projektes. Dazu gleich mehr.

5. Sicherung

Letzter Bearbeitungsschritt ist die Sicherung des Excelprojektes.

Hier ist der einzige Teil, der beschreibbar bleiben sollte, der Eingabebereich für die sog. Messwerte. Den Rest, also den Bereich der gerechneten Wertereihen und des Diagramms, sollte man vor Zugriffen schützen.

Der Schutz wird im Menü `<Extras>/<Schutz>/<Blattschutz ...>` gesetzt oder aufgehoben. Die Bereiche, die zugänglich bleiben sollen, müssen zuvor aus dem Schutz ausgenommen. Dazu müssen sie markiert werden. Dann kann man unter `<Format>/<Zellen>/<Sicherheit>` diese Einstellung vornehmen. Allerdings ergeben sich nun Probleme. Excel selbst kann in den geschützten Bereichen auch nicht tätig werden. Deshalb muss während der Schreibaktivitäten der "Sub ausgabe" der Blattschutz außer Kraft gesetzt werden. Also muss in der "Sub ausgabe" unmittelbar vor dem doppelten "Do" die Aufhebung des Blattschutzes erfolgen. Das geschieht mit `"ActiveSheet.Unprotect"` und nach erfolgten Ausgaben, die ihrerseits die Aktualisierung des Diagramms bewirken, kann hinter den beiden Loop-Zeilen wieder eingeschaltet werden durch: `"ActiveSheet.Protect"`.

Hat jemand nach dem Test eine Fehlermeldung erhalten? Die Sortierung im Makro "glaetten" wird verweigert, obwohl die betroffenen Zellen vom Schutz ausgenommen sind, selbst, wenn es gar nichts zu sortieren gibt. Deshalb wird die Sort-Zeile ebenfalls von den Protection-Zeilen umrahmt und dann klappt alles.

Hallo, wir sind fertig!

Viel Erfolg beim Verschlimmbessern, das nun hoffentlich einsetzt. Erst dabei lernt man richtig, worauf es ankommt!

6. Nachschlag

Irgendwann wird es ja doch bemerkt.

Der Spline-Algorithmus kann es auf den Tod nicht ausstehen, wenn ein x -Wert zweimal vertreten ist (Ausprobieren!) Um ihn nicht unnötig zu ärgern, kann man dem ja vorbeugen, indem man dies, natürlich nach der Sortierung, beim Einlesen, überprüft. Im Falle des Falles übernimmt man einfach nur den ersten Wert, oder den letzten, schon ist das Problem gelöst.

Wie das im Programm aussieht? Das darf sich jeder, als Hausaufgabe, selbst ausdenken.