2 Interpolations- und Ausgleichsfunktionen

Für die Beschreibung von Kurven und Flächen ist ein arithmetischer Ausdruck, mit dem die Koordinaten aller Punkte (im interessierenden Bereich) bestimmt werden, eher die Ausnahme als die Regel. Stellvertretend für viele Probleme der Kurven- und Flächenapproximation sollen die beiden in der technischen Praxis wohl häufigsten Fälle dieser Art stehen:

- Der Ingenieur, der **Meßreihen** aufnimmt, hat in der Regel damit noch keine anschauliche Kurve, die den Zusammenhang der untersuchten Parameter verdeutlicht, sondern nur diskrete Punkte. Auch für die weitere Bearbeitung (vielfach auch nur für eine "geglättete" graphische Darstellung) ist ein funktionaler Zusammenhang $\boldsymbol{y = f(x)}$ wünschenswert.
- Der Konstrukteur, der sogenannte **Freiformflächen** beschreiben muß (Tragflügel, Turbinenschaufeln, Karosserien, Schiffsrümpfe, Gußteile, ...), kann diese im allgemeinen nur punktweise definieren. Für die graphische Darstellung (bevorzugt als ebene Kurven in Schnittdarstellungen) ist es sehr hilfreich, wenn die Einzelpunkte nicht durch Polygonzüge verbunden werden müssen.

Gegeben ist also eine Menge diskreter Punkte ("Stützstellen"), die auf sinnvolle Art durch stetige Funktionen approximiert werden sollen. **Interpolation** und **Ausgleichsrechnung** sind die mathematischen Hilfsmittel für solche Probleme. Während bei der Interpolation eine Funktion gesucht ist, die durch sämtliche Stützstellen verläuft, ist das Ziel der Ausgleichsrechnung eine möglichst "glatte" Funktion, die nur einige (oder keinen) der vorgegebenen Punkte berührt und in einem exakt zu definierenden Sinne "optimal" ist.

In allen Fällen sind Polynomfunktionen für die Approximation besonders beliebt, weil diese sich auf bequeme Art weiterverarbeiten lassen. Wenn allerdings für einen gemessenen Sachverhalt der theoretische Zusammenhang bekannt ist (z. B.: Abkühlung verläuft nach einer Exponentialfunktion), sollte der "passende" Funktionstyp bevorzugt werden.

2.1 Interpolation durch Polynome

Bei $\boldsymbol{n+1}$ vorgegebenen Funktionswerten

$$y_i(x_i) \quad , \qquad i = 0,\, 1,\, \ldots ,\, n$$

kann stets **eindeutig** ein Polynom $\boldsymbol{n}$-ten Grades der Form

$$y(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

bestimmt werden, das alle vorgegebenen Punkte berührt (z. B. kann durch zwei Punkte eindeutig eine Gerade, durch 3 Punkte eine quadratische Parabel gelegt werden usw.). Auf dieses Problem wird hier nicht weiter eingegangen, weil es als Sonderfall der "Ausgleichung durch Polynome" (behandelt im Abschnitt 2.3) anfällt.

2.2 Spline-Interpolation

Wenn man ein elastisches Lineal (engl.: "SPLINE" - biegsame Feder, Latte) so verformt, daß es durch $n+1$ diskrete Punkte verläuft, so ergibt sich eine Kurve, über die nach den Regeln der Technischen Mechanik folgende Aussagen möglich sind:

- Da die Belastung nur an den Stützstellen eingeleitet wird, wird die "Biegelinie" in jedem der n Bereiche durch eine Polynomfunktion dritten Grades beschrieben (gilt eigentlich nur für "kleine Verformungen", wird hier jedoch allgemein verwendet), so daß $4n$ Koeffizienten zu bestimmen sind, die sämtliche Funktionen beschreiben.
- Jedes (jeweils nur in einem Bereich gültige) Polynom muß die beiden "Verschiebungsrandbedingungen" an beiden Rändern erfüllen ($2n$ Bedingungen).
- Benachbarte Polynome stoßen an den Übergangsstellen tangential aneinander ($n-1$ "Übergangsbedingungen").
- Die Gleichheit der Biegemomente an den Übergangsstellen wird durch Gleichheit der zweiten Ableitungen der jeweils beiden Polynome an diesen Punkten garantiert ($n-1$ "Übergangsbedingungen").
- An den beiden freien Rändern ist das Biegemoment (und damit die zweite Ableitung) gleich Null (**2** Bedingungen).

Funktionen, die auf die beschriebene Weise die Stützstellen miteinander verbinden, heißen **natürliche Splines**. Sie haben mit der Polynom-Interpolation (Abschnitt 2.1) den Vorteil gemeinsam, alle Stützstellen zu "treffen", vermeiden aber den wesentlichen Nachteil der Polynom-Interpolation, zwischen den Stützstellen sehr weit auszuschwingen.

Während sich bei der Polynom-Interpolation stets eindeutige Funktionen ergeben (zu jedem x-Wert gehört genau ein y-Wert), können mit der Spline-Interpolation beliebige (auch sich mehrfach selbst schneidende und geschlossene) Kurven nachgebildet werden. Man beachte, daß bei der Definition der Punktmenge, die die Kurve beschreibt, die Reihenfolge der Punkte bei der Spline-Interpolation (im Gegensatz zur Polynom-Interpolation) eine Rolle spielt.

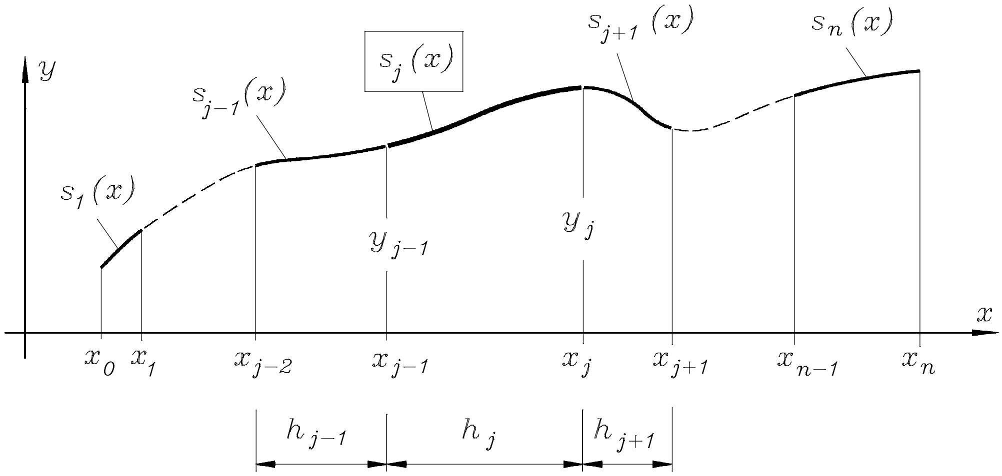
Neben den natürlichen Splines sind für die **Approximation geschlossener Kurven** noch die **periodischen Splines** bedeutsam, deren Definition sich nur in den letzten beiden Bedingungen von der Definition der natürlichen Splines unterscheidet: Da kein freier Rand mehr vorhanden ist, gibt es auch keine Momentenfreiheit mehr, dafür müssen Tangentenneigung und Biegemoment gleich sein, und damit werden mit der Gleicheit der beiden ersten Ableitungen am (identischen) Anfangs- bzw. Endpunkt ebenfalls zwei Bedingungen gefunden.

Nachfolgend wird ein effektiver Algorithmus zur Ermittlung aller Koeffizienten der **kubischen Polynome, die die natürliche Spline-Interpolation beschreiben**, entwickelt.

Für jeden Abschnitt $j = 1, \dots, n$ wird ein Polynomansatz

$$s_j(x) = y(x) = a_j + b_j (x - x_{j-1}) + c_j (x - x_{j-1})^2 + d_j (x - x_{j-1})^3 \qquad \textbf{(2.1)}$$

mit den unbekannten Parametern a_j, b_j, c_j, d_j gewählt (x_{j-1} ist die Abszisse des linken Randpunktes des Abschnitts j). Von den verschiedenen Möglichkeiten, die $4n$ Parameter zu ermitteln, wird hier folgende Strategie verfolgt: Die a_j, b_j und d_j werden durch die gegebenen Größen (x_j und y_j) und die unbekannten Werte c_j ausgedrückt, wobei ein lineares Gleichungssystem für die c_j entsteht, nach dessen Lösung auch a_j, b_j und d_j berechnet werden können.



- Die Forderung, daß die Funktion s_j am linken Rand x_{j-1} den Funktionswert y_{j-1} haben muß, liefert entsprechend

$$s_j(x_{j-1}) = y_{j-1}$$

die erste der gesuchten Beziehungen:

$$a_j = y_{j-1} \quad . \tag{2.2}$$

- Die geforderte Gleichheit der 2. Ableitungen benachbarter Funktionen wird für den rechten Rand des Abschnitts j (an der Stelle x_j) formuliert. Man erhält aus

$$s_j''(x_j) = s_{j+1}''(x_j)$$

die zweite der gesuchten Beziehungen:

$$d_j = \frac{c_{j+1} - c_j}{3\,h_j} \tag{2.3}$$

mit

$$h_j = x_j - x_{j-1} \quad . \tag{2.4}$$

- Am rechten Rand des Abschnitts j (an der Stelle x_j) muß die Funktion s_j den Funktionswert y_j annehmen. Man erhält aus

$$s_j(x_j) = y_j$$

mit (2.1) und unter Verwendung der bereits ermittelten Beziehungen (2.2), (2.3) und (2.4):

$$b_j = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{3}\left(2\,c_j + c_{j+1}\right) \quad . \tag{2.5}$$

- Schließlich wird die geforderte Gleichheit der ersten Ableitungen benachbarter Funktionen für den linken Randpunkt (an der Stelle x_{j-1}) formuliert:

$$s_{j-1}'(x_{j-1}) = s_j'(x_{j-1}) \quad .$$

Diese Forderung liefert unter Verwendung der Formeln (2.1) bis (2.5) eine Beziehung, die die "***c***-Parameter" der Ansatzfunktion s_j und ihrer beiden Nachbarn enthält:

$$h_{j-1}\, c_{j-1} + 2\left(h_{j-1} + h_j\right) c_j + h_j\, c_{j+1} = g_j \qquad \textbf{(2.6)}$$

mit

$$g_j = 3\left(\frac{y_j - y_{j-1}}{h_j} - \frac{y_{j-1} - y_{j-2}}{h_{j-1}}\right) \quad . \qquad \textbf{(2.7)}$$

♦ Die Gleichungen (2.6) können nur für $j = 2, 3, \ldots , n$ aufgeschrieben werden (Funktion s_1 hat keinen linken Nachbarn für die Forderung nach Gleichheit der ersten Ableitung), sie enthalten aber $n + 1$ Unbekannte $c_1, c_2, \ldots , c_{n+1}$. Dieses Defizit wird für die **natürlichen Splines** durch die Forderung nach dem Verschwinden der zweiten Ableitungen an den Rändern behoben. Aus

$$s_1^{\prime\prime}(x_0) = 0 \qquad \textit{und} \qquad s_n^{\prime\prime}(x_n) = 0$$

folgt:

$$c_1 = 0 \qquad ; \qquad c_{n+1} = 0 \quad . \qquad \textbf{(2.8)}$$

Aufgabe 2.1: Für **periodische Splines** gelten die Gleichungen (2.1) bis (2.7) ebenfalls. Wie lauten die beiden Gleichungen, die an Stelle von (2.8) verwendet werden müssen (sie müssen die Gleicheit der ersten beiden Ableitungen für die Punkte **0** und ***n*** repräsentieren)?

Mit (2.8) können die Gleichungen (2.6) zu folgendem Gleichungssystem zusammengefaßt werden:

$$\begin{bmatrix} 2\,(h_1 + h_2) & h_2 & 0 & \ldots & & & \\ h_2 & 2\,(h_2 + h_3) & h_3 & 0 & & & \\ 0 & h_3 & \ldots & & & & \\ & \ldots & & & & & \\ & & \ldots & & & & \\ & & & & 0 & h_{n-1} & 2\,(h_{n-1} + h_n) \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ . \\ . \\ . \\ c_n \end{bmatrix} = \begin{bmatrix} g_2 \\ g_3 \\ . \\ . \\ . \\ g_n \end{bmatrix} \quad . \qquad \textbf{(2.9)}$$

Dieses Gleichungssystem hat eine symmetrische Bandmatrix (Tridiagonalmatrix). Da die h_j–Werte positiv, negativ und auch Null sein können, ist die Matrix weder garantiert positiv definit (es können negative Hauptdiagonalelemente auftreten, Cholesky-Zerlegung scheidet also aus) noch garantiert regulär. Wenn z. B. nur zwei h_j–Werte benachbarter Abschnitte Null sind (gleiche x_j–Werte für drei benachbarte Punkte), ergibt sich in der Matrix eine Null-Zeile. Man muß also bei der Lösung des Gleichungssystems diese Sonderfälle ausschließen.

Wenn allerdings mit (2.9) die c_j–Werte berechnet werden können, sind mit (2.2), (2.3), (2.5) und (2.8) auch alle übrigen Ansatzparameter der ***n*** Polynome 3. Grades bekannt, und das Interpolationsproblem ist gelöst.

Für den **Spezialfall äquidistanter Stützstellen** mit

$$h_j = h = konst. \qquad \textit{für} \qquad j = 1, 2, \ldots, n$$

vereinfachen sich die Beziehungen folgendermaßen:

$$\begin{bmatrix} 4 & 1 & 0 & \ldots & & & \\ 1 & 4 & 1 & 0 & \ldots & & \\ 0 & 1 & 4 & 1 & 0 & \ldots & \\ & \ldots & & & & & \\ & & \ldots & & & & \\ & & & 0 & 1 & 4 & 1 \\ & & & & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} c_2 \\ c_3 \\ \cdot \\ \cdot \\ \cdot \\ c_n \end{bmatrix} = \frac{3}{h^2} \begin{bmatrix} y_0 - 2\,y_1 + y_2 \\ y_1 - 2\,y_2 + y_3 \\ \cdot \\ \cdot \\ \cdot \\ y_{n-2} - 2\,y_{n-1} + y_n \end{bmatrix} , \qquad \textbf{(2.10)}$$

$$\begin{aligned} c_1 &= 0 \quad , \\ c_{n+1} &= 0 \quad , \\ a_j &= y_{j-1} \quad , \\ b_j &= \frac{1}{h}\left(y_j - y_{j-1}\right) - \frac{h}{3}\left(2\,c_j + c_{j+1}\right) \quad , \\ d_j &= \frac{1}{3\,h}\left(c_{j+1} - c_j\right) \quad . \end{aligned}$$

Das lineare Gleichungssystem, aus dem die c_j–Werte berechnet werden, hat in diesem Fall immer eine Lösung, weil die Koeffizientenmatrix positiv definit ist (die sehr effektive Cholesky-Zerlegung kann als Lösungsverfahren verwendet werden). Der gesamte Formelsatz (2.10) versagt nur für $\boldsymbol{h = 0}$. Dieser Ausnahmefall muß also in einem Rechenprogramm abgefangen werden, kann aber mühelos einer "Sonderbehandlung" zugeführt werden: Da bei $\boldsymbol{h = 0}$ sämtliche Stützstellen die gleiche x_j–Koordinate haben, degenerieren alle Spline-Funktionen zu Parallelen zur $\boldsymbol{y}$-Achse mit der Gleichung: $\boldsymbol{x = x_j =}$ **konst.**

Der Formelsatz (2.10) ist im CAMMPUS-Programm MCALCU für das Definieren der sogenannten "Funktionen vom File" realisiert, um damit z. B. die Übernahme von Ergebnissen von Diskretisierungsverfahren (Differenzenverfahren) zu ermöglichen. Die dafür sinnvolle Beschränkung auf äquidistante Stützstellen stellt keine generelle Einschränkung der Allgemeinheit dar, weil in MCALCU jede Funktion in Abhängigkeit von einer anderen Funktion dargestellt werden kann.

Die in MCALCU realisierte Lösung (mit äquidistanten Stützstellen), die ($\boldsymbol{h > 0}$ vorausgesetzt) keinerlei Spezialfälle zu fürchten hat, kann sogar als sinnvolle Alternative zum allgemeinen Fall (2.9) empfohlen werden, um das bei Verwendung von (2.9) recht mühsame Abfangen aller Sonderfälle zu umgehen: Man ersetzt alle Wertepaare $\boldsymbol{(x_j, y_j)}$ durch jeweils zwei Wertepaare $\boldsymbol{(x_j, t_j)}$ und $\boldsymbol{(y_j, t_j)}$, wobei dem Parameter $\boldsymbol{t}$ eine (beliebige positive) äquidistante Schrittweite zugewiesen wird:

$$h = t_j - t_{j-1} > 0 \quad .$$

Der Formelsatz (2.10) ist einmal auf die Wertepaare $\boldsymbol{(x_j, t_j)}$ anzuwenden (und liefert die nachfolgend mit $\boldsymbol{a_{xj}}$, $\boldsymbol{b_{xj}}$, $\boldsymbol{c_{xj}}$, $\boldsymbol{d_{xj}}$ bezeichneten Parameter) und zum anderen auf die Wertepaare

(y_j, t_j) (liefert die nachfolgend mit a_{yj}, b_{yj}, c_{yj}, d_{yj} bezeichneten Parameter). Man erhält auf diesem Wege (bei Umgehung aller numerischen Probleme) die Spline-Funktionen in Parameter-Darstellung:

$$\begin{aligned} s_{xj}(t) &= x(t) = a_{xj} + b_{xj}(t - t_{j-1}) + c_{xj}(t - t_{j-1})^2 + d_{xj}(t - t_{j-1})^3 \quad , \\ s_{yj}(t) &= y(t) = a_{yj} + b_{yj}(t - t_{j-1}) + c_{yj}(t - t_{j-1})^2 + d_{yj}(t - t_{j-1})^3 \quad . \end{aligned} \tag{2.11}$$

Aufgabe 2.2: Von der Bahnkurve, die der Punkt eines Getriebes beschreibt, wurden folgende Koordinaten gemessen:

t	0	0,1	0,2	0,3	0,4	0,5	0,6
s_x	0	− 1	0,1	0,1	− 0,2	0,5	1,2
s_y	3	1,1	− 0,2	− 1	− 0,6	0,8	2,5

Man schreibe zwei Files **SX.DAT** und **SY.DAT**, die zeilenweise die Koordinaten s_x bzw. s_y enthalten, und erzeuge mit diesen Files jeweils zwei "Funktionen vom File" mit dem CAMMPUS-Programm MCALCU, wobei einmal die Auswahl "Spline-Interpolation" und einmal die Auswahl "Polygonzug" zu wählen ist. Für beide Varianten sind die graphischen Darstellungen der Funktion $s_y(t)$ und der Bahnkurve $s_y(s_x)$ zu erzeugen.

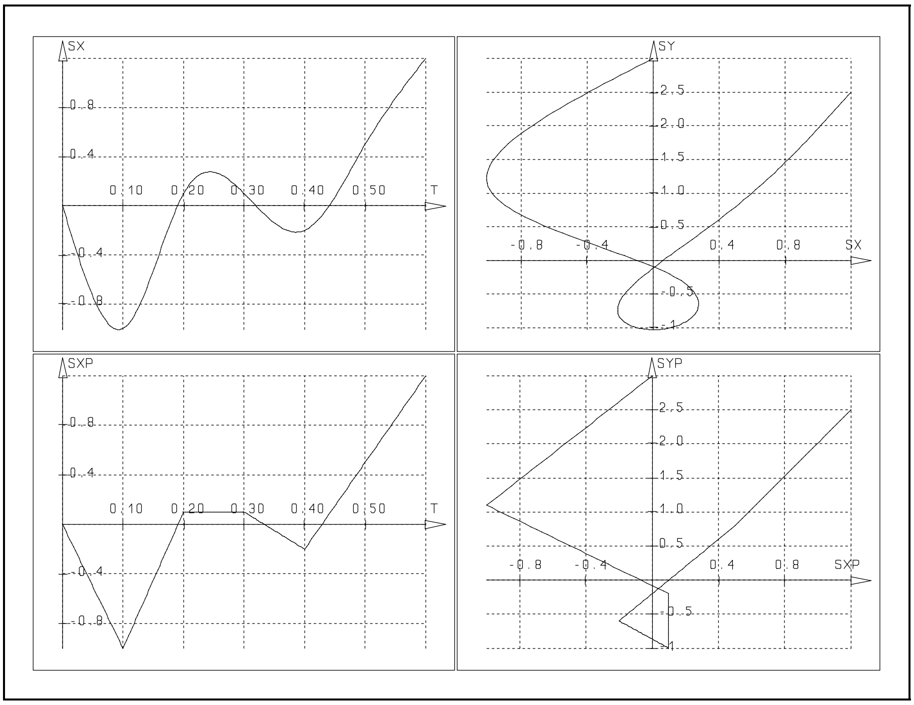
```
 0
-1
 0.1
 0.1
-0.2
 0.5
 1.2
```

File SX.DAT

```
 3
 1.1
-0.2
-1
-0.6
 0.8
 2.5
```

File SY.DAT

Das folgende Bild zeigt das Ergebnis der Aufgabe 2.2. In beiden Fällen ergeben die Spline-Approximationen im Vergleich mit den Polygon-Darstellungen sehr schöne Kurven trotz der geringen Stützstellen-Anzahl.

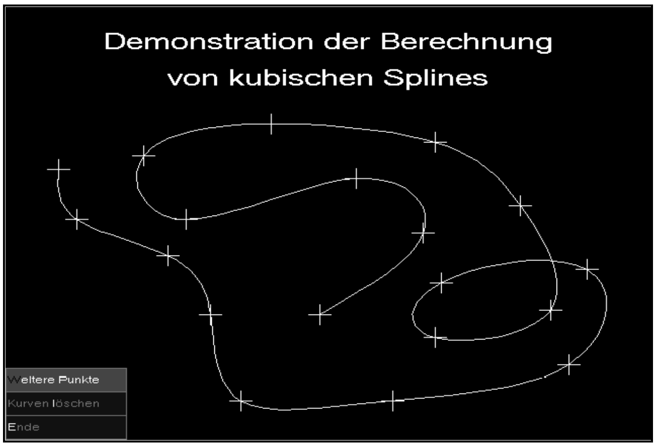


Aufgabe 2.2: Spline-Approximationen (oben) und Polygonzüge (Programm MCALCU)

In der MATHLIB-Toolbox existiert ein Unterprogramm **SPLINP_M**, das den Formelsatz (2.10) realisiert und aus $\boldsymbol{n+1}$ äquidistanten Funktionswerten die Koeffizienten der natürlichen Spline-Funktionen für die $\boldsymbol{n}$ Abschnitte berechnet. Das im Quellcode auf dem NOVELL-Server verfügbare Programm SPLINE01.FOR verwendet SPLINP_M (zweimal, um nicht äquidistante Stützstellen durch Erzeugen der Splines in Parameterdarstellung realisieren zu können) und demonstriert die Berechnung von natürlichen Splines mit beliebigen (mit der Maus einzugebenden) Punkten.

Aufgabe 2.3: Man compiliere und starte das Programm SPLINE01 und gebe beliebige Punktsequenzen ein, um ein "Gefühl" für die Spline-Approximation zu bekommen (nebenstehender Bildschirm-Schnappschuß zeigt die graphische Ausgabe der berechneten Spline-Funktionen).

Man beachte die Veränderungen, die sich in bereits approximierten Bereichen ergeben, wenn ein weiterer Punkt hinzugefügt wird.



2.3 Ausgleichung nach der "Fehlerquadrat"-Methode

Für eine zu bestimmende Funktion eines vorgegebenen Typs (Gerade, Parabel, Exponentialfunktion, ...) sind mehr Wertepaare (Punkte) vorgegeben, als für die eindeutige Bestimmung der Funktion erforderlich sind (Beispiel: Es sind 5 Punkte gegeben, die eine Gerade bestimmen sollen). Dann muß (bei Forderung nach einer eindeutigen Lösung) ein "Maß" definiert werden, mit dem von allen möglichen Funktionen des vorgegebenen Typs die "beste Funktion" identifiziert werden kann.

In den technischen und naturwissenschaftlichen Fachgebieten (aber auch in der mathematischen Statistik) wird für das beschriebene Problem vornehmlich die **Gaußsche Methode der kleinsten Fehlerquadratsumme** verwendet (große Ausnahme: Beschreibung beliebiger Kurven und Flächen in CAD-Systemen und bei der NC-Programmierung, vgl. Abschnitt 2.4), nach der das oben genannte "Maß" wie folgt definiert wird:

Gegeben sind $\boldsymbol{n}$ Wertepaare (x_1,y_1), (x_2,y_2), ... , (x_n,y_n), die eine Funktion $y = f(x)$ definieren ($\boldsymbol{n}$ ist größer als die Anzahl der unbestimmten Parameter in f). Die Abweichungen der Funktionswerte $f(x_i)$ von den vorgegebenen Werten y_i werden als "Fehler"

$$v_i = y_i - f(x_i)$$

bezeichnet, und es wird gefordert, daß die "Summe der Fehlerquadrate" minimal wird:

$$S = \sum_{i=1}^{n} v_i^2 = \sum_{i=1}^{n} \left[y_i - f(x_i) \right]^2 \quad \Rightarrow \quad \textit{Minimum}$$

Beispiel: Es sollen $n > 3$ Meßpunkte (x_1,y_1), (x_2,y_2), ... , (x_n,y_n) durch eine quadratische Parabel

$$y = a\,x^2 + b\,x + c$$

ausgeglichen werden.

Notwendige Bedingung für die Erfüllung der Minimalforderung

$$S = \sum_{i=1}^{n} v_i^2 = \sum_{i=1}^{n} \left[y_i - (a\,x_i^2 + b\,x_i + c) \right]^2 \quad \Rightarrow \quad Minimum$$

ist das Verschwinden der drei partiellen Ableitungen von S nach den zu bestimmenden Freiwerten a, b und c:

$$\frac{\partial S}{\partial a} = 0 \; : \qquad -2 \sum_{i=1}^{n} \left[y_i - (a\,x_i^2 + b\,x_i + c) \right] x_i^2 = 0 \quad ,$$

$$\frac{\partial S}{\partial b} = 0 \; : \qquad -2 \sum_{i=1}^{n} \left[y_i - (a\,x_i^2 + b\,x_i + c) \right] x_i = 0 \quad ,$$

$$\frac{\partial S}{\partial c} = 0 \; : \qquad -2 \sum_{i=1}^{n} \left[y_i - (a\,x_i^2 + b\,x_i + c) \right] = 0 \quad .$$

Dies ist ein lineares Gleichungssystem, die sogenannten **Normalgleichungen** zur Betimmung von a, b und c:

$$\begin{bmatrix} \sum x_i^4 & \sum x_i^3 & \sum x_i^2 \\ \sum x_i^3 & \sum x_i^2 & \sum x_i \\ \sum x_i^2 & \sum x_i & n \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i^2\, y_i \\ \sum x_i\, y_i \\ \sum y_i \end{bmatrix} \quad .$$

Nach der Lösung dieses Gleichungssystems ist die gesuchte quadratische Parabel bekannt.

Das am Beispiel der Ausgleichung durch eine quadratische Parabel demonstrierte Vorgehen läßt sich verallgemeinern. Speziell führt die **Ausgleichung von n Meßpunkten durch ein Polynom m-ten Grades ($m \le n - 1$)** stets auf ein lineares Gleichungssystem der Form

$$\begin{bmatrix} \sum x_i^{2m} & \sum x_i^{2m-1} & \dots & \sum x_i^{m} \\ \sum x_i^{2m-1} & \sum x_i^{2m-2} & \dots & \sum x_i^{m-1} \\ \dots & & & \dots \\ \dots & & & \dots \\ \dots & & \dots & \sum x_i \\ \sum x_i^{m} & \dots & \sum x_i & n \end{bmatrix} \begin{bmatrix} a_m \\ a_{m-1} \\ \cdot \\ \cdot \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum x_i^{m}\, y_i \\ \sum x_i^{m-1}\, y_i \\ \cdot \\ \cdot \\ \sum x_i\, y_i \\ \sum y_i \end{bmatrix} \quad . \qquad \textbf{(2.12)}$$

Die Koeffizientenmatrix des Gleichungssystems (2.12) ist symmetrisch und positiv definit. Das Gleichungssystem kann also z. B. mit dem Cholesky-Verfahren gelöst werden (CAMMPUS-Programm MLINEQ). Mit der Lösung von (2.12) ist dann das Ausgleichspolynom

$$y = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 \qquad \textbf{(2.13)}$$

bekannt.

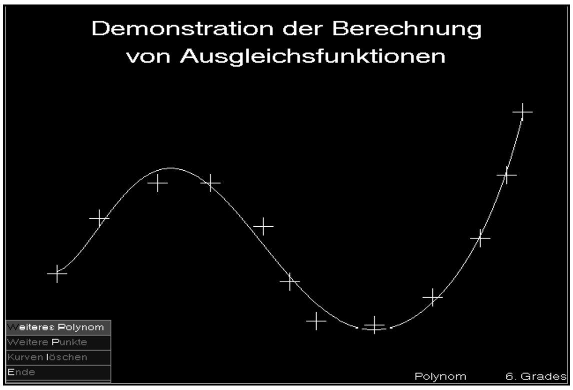
Mit dem Spezialfall $\boldsymbol{m = n - 1}$ (Polynom $\boldsymbol{m}$-ten Grades durch $\boldsymbol{m + 1}$ Stützstellen) ist das Interpolationsproblem (vgl. Abschnitt 2.1) als Sonderfall im Ausgleichsproblem enthalten.

Die Ausgleichung einer beliebigen Wertepaar-Menge durch eine Polynomfunktion entsprechend (2.12) und (2.13) wird von dem MATHLIB-Unterprogramm **AUSPOL_M** erledigt. Dies wird in den Programmen **AUSGL01.FOR** und **AUSGL02.FOR** verwendet, die beide im Quellcode auf dem NOVELL-Server zu finden sind.

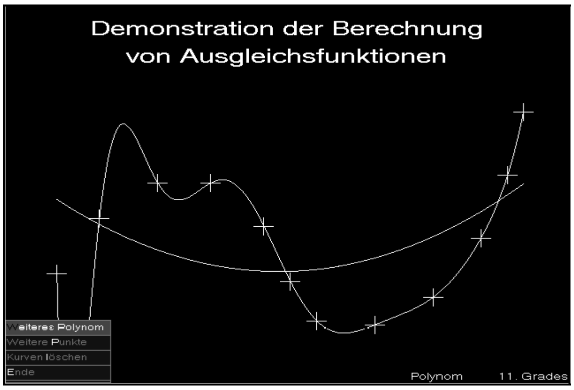
AUSGL01 gestattet die Eingabe einer Wertetabelle vom File und berechnet die Koeffizienten eines Ausgleichspolynoms (Grad des gewünschten Polynoms wird abgefragt, vgl. nachfolgende Aufgabe 2.4).

AUSGL02 demonstriert die Approximation einer (durch Eingabe mit der Maus) zu definierenden Punktmenge durch Ausgleichsfunktionen unterschiedlichen Grades.

Die nachfolgenden Bildschirm-Schnappschüsse zeigen die Ergebnisse der Berechnung von Ausgleichspolynomen unterschiedlichen Grades für 12 vorgegebene Punkte:



Ausgleichspolynom 6. Grades



Ausgleichspolynome 2. und 11. Grades

Während eine Ausgleichsfunktion 6. Grades (linkes Bild) zwar keinen der vorgegebenen Punkte trifft, erweist sie sich doch als sehr gute Approximation mit glättendem Charakter. Im Gegensatz dazu ist eine Funktion 2. Grades (rechtes Bild) bei der vorgegebenen Lage der Punkte als Ausgleichsfunktion untauglich. Eine Funktion 11. Grades (rechtes Bild) ist für 12 Punkte eine Interpolationsfunktion, trifft sämtliche Punkte exakt, zeigt aber (besonders deutlich im linken Bereich) das typische weite Ausschwingen der Polynom-Interpolation.

Die Ausgleichung mit beliebigen anderen Funktionen kann auf wesentlich größere Probleme stoßen, meist entsteht ein nichtlineares Gleichungssystem für die Ansatzparameter. Für einen wichtigen Spezialfall der Ingenieur-Mathematik, der Ausgleichung durch eine Exponentialfunktion

$$y = c\, e^{a x} \quad , \qquad \textbf{(2.14)}$$

hilft ein kleiner Trick. Man logarithmiert (2.14) auf beiden Seiten:

$$\ln y = \boldsymbol{a}\, x + \ln c \quad .$$

Dann kann nach Einführen der Abkürzungen

$$z = \ln y \qquad , \qquad \boldsymbol{b} = \ln c$$

das einfache Ausgleichsproblem (Ausgleichung durch eine Gerade)

$$z = \boldsymbol{a}\, x + \boldsymbol{b}$$

gelöst werden, wobei die $\boldsymbol{y_i}$–Werte der Stützstellen durch $\boldsymbol{z_i = \ln y_i}$ zu ersetzen sind. Die Lösung liefert den Parameter $\boldsymbol{a}$ für (2.14) direkt, der andere Parameter wird nach

$$c = \boldsymbol{e}^{b}$$

berechnet.

Aufgabe 2.4: Die Auswertung von Videoaufnahmen der Flugbahn einer Kugel beim Kugelstoßen ergab folgende Punktkoordinaten:

***x*[*m*]**	**0**	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**	**20**
***y*[*m*]**	**2,00**	**3,65**	**4,93**	**5,84**	**6,38**	**6,56**	**6,37**	**5,81**	**4,88**	**3,58**	**1,91**

Da der Luftwiderstand vernachlässigt werden kann, ist die Flugbahn eine quadratische Parabel. Man ermittle mit Hilfe des Programms AUSGL01 die Gleichung der Parabel und stelle die Kurve graphisch dar (z. B. mit dem CAMMPUS-Programm MCALCU).

Das Programm AUSGL01 liest die Eingabewerte von einem File XY.DAT (jeweils ein Wertepaar pro Zeile) und liefert folgende Ergebnisse:

```
 0  2
 2  3.65
 4  4.93
 6  5.84
 8  6.38
10  6.56
12  6.37
14  5.81
16  4.88
18  3.58
20  1.91
```

File XY.DAT

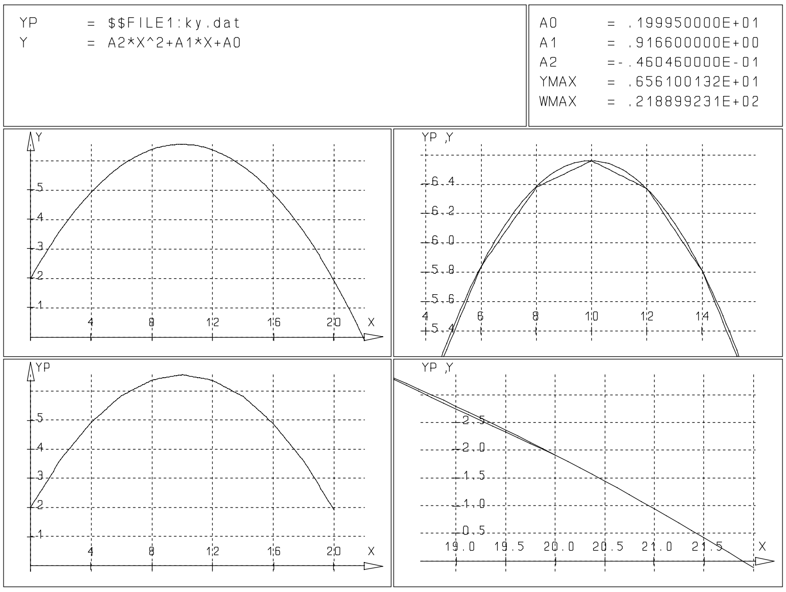
```
Ausgleichspolynom m-ten Grades
==============================

Es wurden    n =   11    Werte eingelesen.
Gewünschter Grad des Polynoms (m < n):    m =  2

Es wurde ein Polynom 2. Grades erzeugt.
Koeffizienten des Ausgleichspolynoms
y = a(m) x^(m) + a(m-1) x^(m-1) + ... + a(0):

a(2) =  -.46046E-01
a(1) =   .91660E+00
a(0) =   .19995E+01
```

Ausgabe des Programms AUSGL01

Es ergibt sich also die Ausgleichsparabel

$$y = -0{,}046046\, x^2 + 0{,}91660\, x + 1{,}9995 \quad .$$

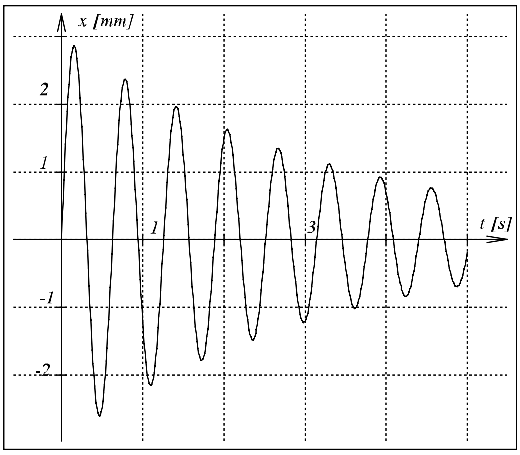
Alle eventuell interessierenden Aussagen über den Kurvenverlauf (Scheitelpunkt, Auftreffpunkt der Kugel, ...) können mit dieser Funktion berechnet werden. Die nachfolgend gezeigte Darstellung (CAMMPUS-Programm MCALCU) zeigt, daß die Ausgleichsfunktion (Bild links oben) nur geringfügig vom Polygonzug (links unten) abweicht. In der Zoom-Darstellung des Bereichs in der Nähe des Scheitelpunktes (rechts oben) werden die Abweichungen sichtbar. In der Zoom-Darstellung rechts unten (Umgebung des Auftreffpunktes auf dem Boden) zeigt sich ein weiterer Vorteil der Ausgleichsfunktion gegenüber der Wertetabelle: Die Funktion kann über den Bereich der Wertetabelle hinaus extrapoliert werden.



Auswertung der Ausgleichsfunktion mit dem CAMMPUS-Programm MCALCU

In dem kleinen Fenster rechts oben sind u. a. die mit der Ausgleichsfunktion von MCALCU berechneten Werte YMAX (***y***-Koordinate des Scheitels) und WMAX ("Wurfweite") zu sehen.

Aufgabe 2.5: Eine freie gedämpfte Schwingung mit $\omega = \mathbf{10}\ s^{-1}$ (Eigenkreisfrequenz des ungedämpften Schwingers) ist durch den nebenstehenden Meßschrieb gegeben, dem näherungsweise die Amplituden der Schwingung entnommen wurden, die in der nachfolgenden Tabelle zusammengestellt sind.



Mit der Phasenverschiebung $\alpha = \pi/2$ kann der zeitliche Ablauf theoretisch durch folgende Funktion beschrieben werden, die als Grundlage für die Ermittlung einer Ausgleichsfunktion verwendet werden soll (***D*** ist das dimensionslose Lehrsche Dämpfungsmaß):

$$x(t) = C\, e^{-D\,\omega\, t} \cos\left(\sqrt{1 - D^2}\,\omega\, t - \alpha\right)$$

t **[*s*]**	**0,15**	**0,47**	**0,78**	**1,10**	**1,41**	**1,73**	**2,04**	**2,35**
x **[*mm*]**	**2,85**	**− 2,60**	**2,35**	**− 2,15**	**1,95**	**− 1,80**	**1,65**	**− 1,50**

Man ermittle die Werte ***C*** und ***D*** mit Hilfe der Ausgleichsrechnung, stelle die Funktion $\boldsymbol{x(t)}$ mit dem Programm MCALCU graphisch dar und vergleiche mit dem Meßschrieb.