Hochschule München Fakultät 04EI Redaktion: Kahl/Paster/Rauh Stand SS 2010

Praktikum Numerik Versuch 2: "Approximation, Interpolation, Ausgleichsrechnung"

Bearbeiter: Dennis Kunz

Datum: 15.05.2012

1 Interpolation

Gegeben sind **n+1** *Datenpunkte als "Stützwerte" der Interpolation. Die Stützwerte sind hier zur Demonstration aus einer elementaren Funktion* **f** *entnommen, die Sie von Ihrem Dozenten erhalten:*

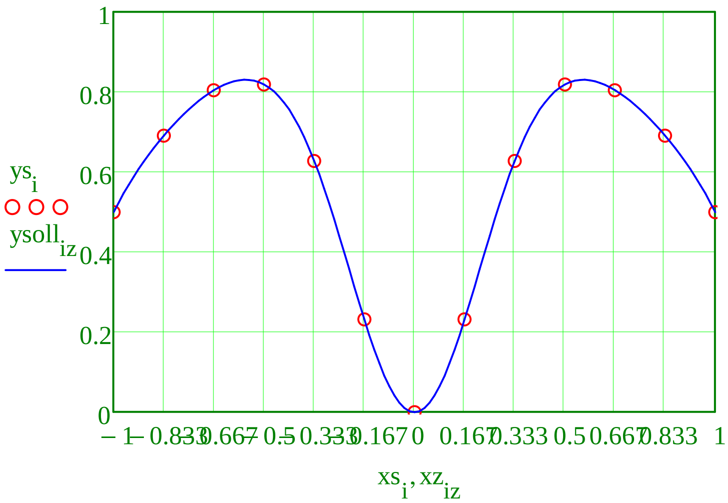
$f(x) := \sqrt{1 - 0.75x^2} - e^{-10x^2}$ nach Angabe Ihres Dozenten (1)

$n := 12$ äquidistant im Bereich [-1, +1]

Wir speichern die Koordinaten der Stützpunkte in den Vektoren **xs** *und* **ys,** *berechnen auch Zwischenpunkte* **(xz, ysoll)** *der Sollwerte und stellen die Daten grafisch dar:*

$hr := \frac{2}{n}$ $i := 0 .. n$ $xs_i := -1 + i \cdot hr$ $ys_i := f(xs_i)$

$nz := 10 \cdot n$ $hz := \frac{hr}{10}$ $iz := 0 .. nz$ $xz_{iz} := -1 + iz \cdot hz$ $ysoll_{iz} := f(xz_{iz})$



1.1 Ein Polynom vom Grad n

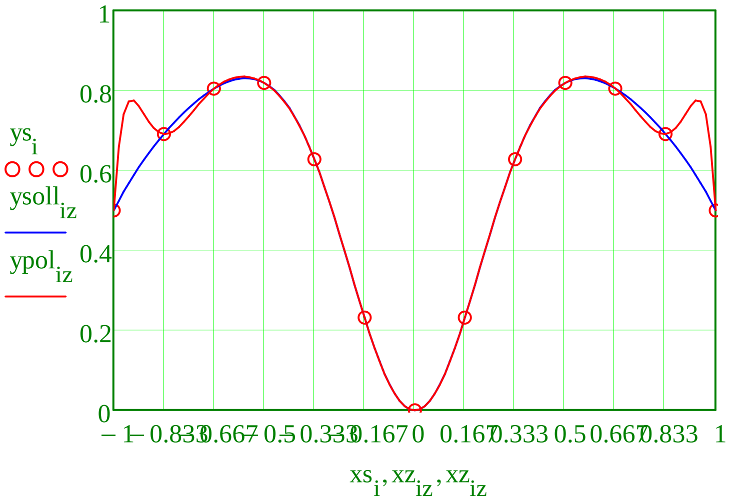
Durch die **n+1** *Datenpunkte kann man bekanntlich eindeutig ein Polynom vom Grad* **<=n** *legen. Die Koeffizienten* **c** *sind Lösungen eines LGS mit Vandermonde-Matrix:*

Gleichungssystem aufstellen $ze := 0 .. n$ $sp := 0 .. n$ $amp_{ze,sp} := (xs_{ze})^{sp}$

$condi(amp) = 5.915 \times 10^5$

Koeffizienten berechnen $c := amp^{-1} \cdot ys$

Formel für Polynom $p(x) := \sum_{i=0}^{n} \left(c_i \cdot x^i\right)$ interpolieren $ypol_{iz} := p(xz_{iz})$



Wir merken uns für Kapitel 3
die exakten Werte
der Stützstellen xs, ys,
der Koeffizientenmatrix A,
der Polynomkoeffizienten c:
xse, yse, Ae, ce

$xse := xs$

$yse := ys$

$Ae := amp$

$ce := c$

Wie beurteilen Sie das Ergebnis? Warum war das zu erwarten?

Gute Näherung im mittleren Bereich. Überschwingen im Außenbereich aufgrund der Eigenart des Polynoms (Streben nach unendlich und minus-unendlich). Funktion f(x) hingegen ist polynomuntypisch.

*Worauf weisen absolute Größe und Vorzeichen der Koeffizienten **c** hin ?*

Sie weisen auf den Wertebereich von y und numerische Instabilität hin.

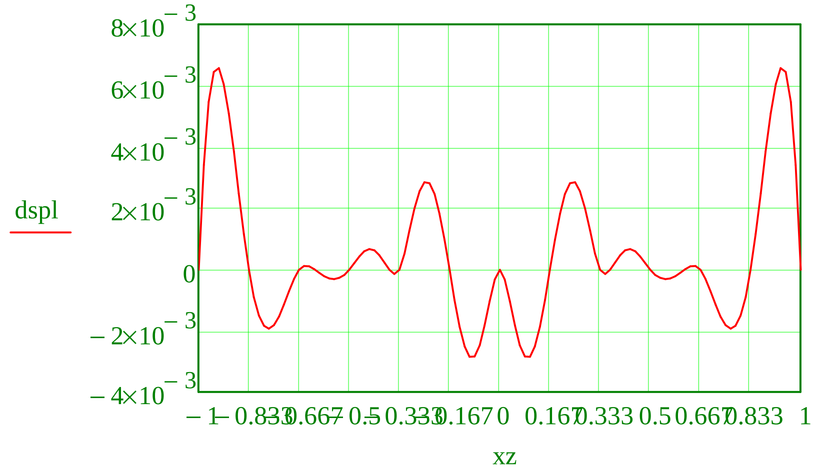
1.2 Kubische Spline-Funktion

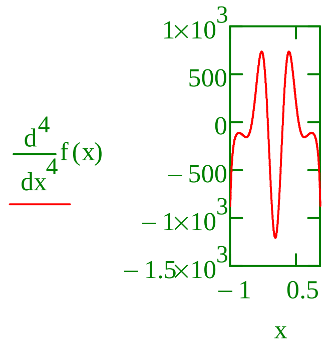
*In Ihrer **Vorbereitung** haben Sie den Aufbau und die Eigenschaften von kubischen Spline-Funktionen studiert.Die Mathcad-Bibliothek enthält dazu einige Prozeduren. Für eine natürliche k.S.F. verwenden wir zunächst die Prozedur **lspline** (=linear am Rand), die einen Vektor **k2** berechnet, der u.a. die Koeffizienten der quadratischen Glieder der Teilpolynome enthält. Eine weitere Prozedur **interp** verwendet anschließend diesen Vektor, um zu der gegebenen Abszisse **xz** eines Zwischenpunktes die interpolierte Ordinate **yspl** zu berechnen: (Alternativ kspline oder ispline für andere Randrestriktionen)*

$k2 := lspline(xs, ys)$ $yspl_{iz} := interp(k2, xs, ys, xz_{iz})$

Um den Unterschied zur Sollkurve sichtbar zu machen, stellen wir direkt die Differenz dar:

$dspl := ysoll - yspl$





Vergleichen Sie die maximale Differenz mit der (theoretischen) oberen Schranke 5/384 * max |f""| * h^4.

Maxium an Rand: 5/384 * 750 * (2/12)^4 = 0.007535 > 0.006 ist erfüllt

$\max(dspl) = 6.579 \times 10^{-3}$ $\quad f\_4(x) := \frac{d^4}{dx^4} f(x)$ $\quad \frac{5}{384} hr^4 \cdot |f\_4(0)| = 0.012$

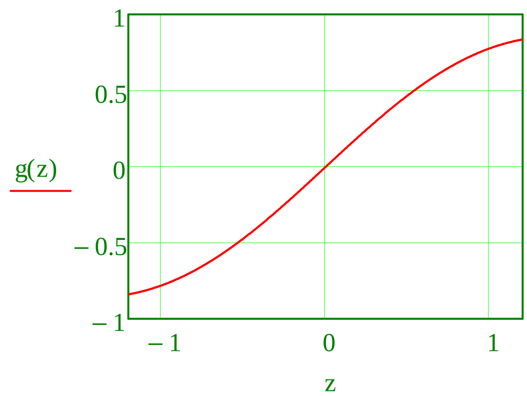
2 Approximation und Ausgleichsrechnung

Von Ihrem Dozenten erhalten Sie eine Funktion **g(x)**, *die im Bereich* **D** *durch eine einfacher zu berechnende Funktion* **p(x)** *(z.B. ein Polynom) angenähert werden soll. Die Abweichung zwischen* **g(x)** *und* **p(x)** *soll dabei nirgends größer sein als eine gegebene Schranke* **eps**.

$g(x) := x \cdot e^{-x^2 \div 4}$

nach Angabe Ihres Dozenten (2)

$eps := 10^{-5}$ im Bereich D: -1 <= x <= +1

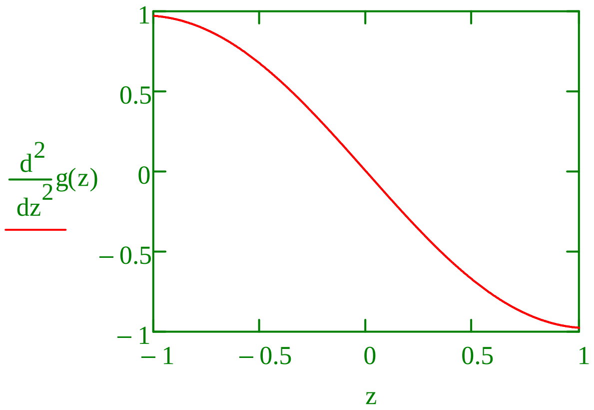


Funktion ist punktsymmetrisch:
die ungerade Funktion hat nur ungerade Tylorkoeffizienten

Für alternierende Reihen gilt das Leibnitz Krit:
Beträge müssen eine Nullfolge bilden
Für den Abrruchfehler gilt |s-sn|<an+1

2.1 Approximation durch stückweise Interpolation

Um **g(x)** *zu approximieren, berechnen wir zunächst einen Satz von Stützpunkten dieser Funktion. Die Werte der approximierenden Funktion* **p(x)** *zwischen den Stützpunkten erhalten wir dann wie in Kap. 1 durch Interpolation, in der Praxis oft stückweise durch Polynome dritten Grades. In Ihrer* ***Vorbereitung*** *haben Sie Formeln zur Abschätzung des max. Fehlers, abhängig vom Abstand* **h** *des Gitters der Abszissen* **x,** *bereitgestellt. Wie lautet die Formel für den Grad 3 (= kubische Interpolation), wenn nur das mittlere Teilstück verwendet wird? (Die benötigte Ableitung können Sie hier mit Mathcad abschätzen.)*



Betragsmax. kleiner als

$M := 1$

*Berechnen Sie nun die für die Toleranz **eps** max. zulässige Stützweite **hmax** für die Funktion **g** im gegebenen Definitionsbereich. Aus **hmax** ergibt sich dann die Mindestzahl der Stützintervalle **nmin**.*

$hmax := \left(\frac{128 \cdot eps}{3 \cdot M}\right)^{\frac{1}{4}}$ $hmax = 0.144$ $nmin := \frac{2}{hmax}$ $nmin = 13.916$

*Wir wählen eine runde Zahl **ns** und berechnen die zugehörige Stützweite **hs**.*

$ns := 10 \cdot ceil\left(\frac{nmin}{10}\right)$ $hs := \frac{2}{ns}$ $ns = 20$ $hs = 0.1$

*Zur einfachen kubischen Interpolation stellen wir Ihnen eine Prozedur **kubint** zur Verfügung. Sie arbeitet nach den Formeln von Newton und berechnet zu jeweils 4 Stützstellen Zwischenwerte, wobei sie immer nur das mittlere Teilstück der entsprechenden Interpolationsparabel benutzt. Deshalb benötigt sie an den Rändern des Nutzbereiches je eine zusätzliche Stützstelle.*

Eingangsgrößen:

- ***ys** ist ein Vektor von Stützordinaten, die zu äquidistanten Abszissen gehören, incl. Zusatzwerte an den Rändern, z.B. zu x = -0.1, 0, 0.1, ...1.0, 1.1 für einen Nutzbereich 0 <= x <= 1. Die x-Werte werden von der Prozedur nicht benötigt.*
- ***nz** ist die Zahl der gewünschten Interpolations-Intervalle zwischen je zwei Stützpunkten; berechnet werden also je nz-1 Zwischenpunkte, aber nur im Nutzbereich.*

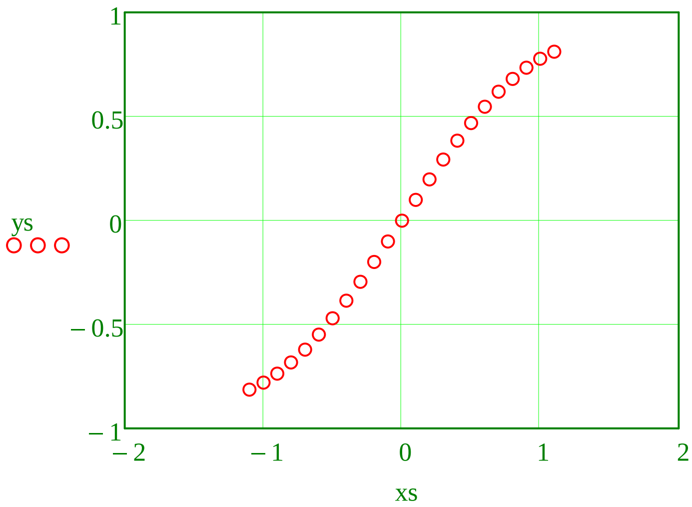
Ausgangsgröße:

- ***y** ist ein Vektor von interpolierten Ordinaten über dem Nutzbereich von x. Also erhalten wir für obiges Beispiel mit nz=5 insgesamt 51 Werte, die zu den Abszissen x = 0, 0.02, 0.04, ... 1.0 gehören.*

```
kubint(ys, nz) := | ns ← length(ys) − 3
                  | for k ∈ 0 .. ns − 1
                  |   | d1 ← ys[k+1] − ys[k]
                  |   | d2 ← ys[k+2] − 2·ys[k+1] + ys[k]
                  |   | d3 ← ys[k+3] − 3·ys[k+2] + 3·ys[k+1] − ys[k]
                  |   | for kz ∈ 0 .. nz − 1
                  |   |   | u ← 1 + kz ÷ nz
                  |   |   | yt ← ys[k] + u·d1 + u·(u − 1)·d2 ÷ 2
                  |   |   | y[k·nz+kz] ← yt + u·(u − 1)·(u − 2)·d3 ÷ 6
                  | y[ns·nz] ← ys[ns+1]
                  | return y
```

*Wir stellen **g** an den Stützpunkten der Interpolation (incl. Zusatzpunkte an den Rändern) grafisch dar:*

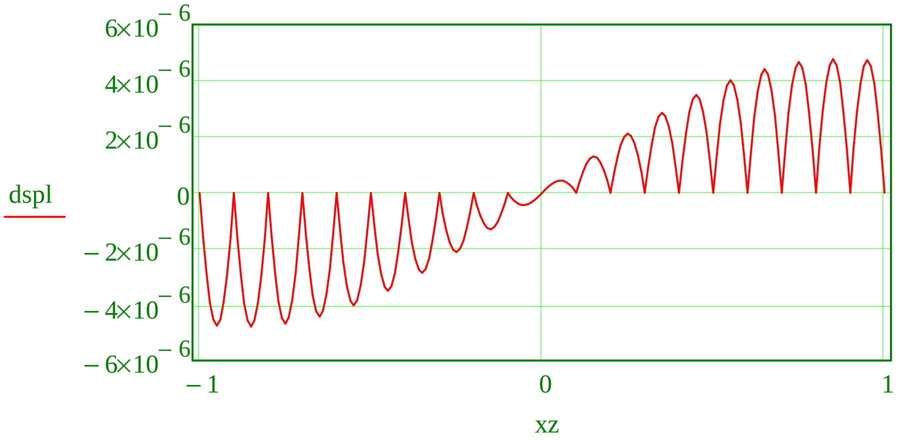
$xs := 0$ $ys := 0$ rücksetzen $i := 0 .. ns + 2$ $xs_i := -1 + (i - 1) \cdot hs$ $ys_i := g(xs_i)$



Und jetzt die Interpolation mit kubint im Nutzbereich ***D****:*

$nz := 10 \quad nzg := ns \cdot nz \quad hz := hs \div nz \quad iz := 0 .. nzg \quad xz_{iz} := -1 + iz \cdot hz$

$ysoll_{iz} := g(xz_{iz}) \quad ykub := kubint(ys, nz) \quad dspl := ysoll - ykub$



Wird unsere Toleranz ***eps*** *eingehalten?*

*Ja, 5*10^-6 < 10^-5*

2.2 Endliche Taylorreihe

Zur Entwicklung von ***g(x)*** *in eine Taylorreihe um Null geben Sie die Formel für die Koeffizienten* ***koef*** *an und berechnen Sie deren Zahlenwerte bis zum Index 15. Entscheiden Sie dann, welche Glieder der Reihe weggelassen werden dürfen, ohne die Schranke* ***eps*** *zu überschreiten.*
Antwort mit Begründung:

$k := 0 .. 7 \qquad koef_{2k+1} := \frac{(-1)^k}{4^k \cdot k!}$

Notwendig ist eine Reihe vom Grad $\quad m := 9$

Summenformel für das Polynom: $\quad p(x) := \sum_{k=0}^{m} \left(koef_k \cdot x^k\right)$

Wir berechnen Abtastpunkte von ***g*** *und* ***p*** *und stellen die Differenz dar:*

$xz := 0 \quad ysoll := 0$ rücksetzen:

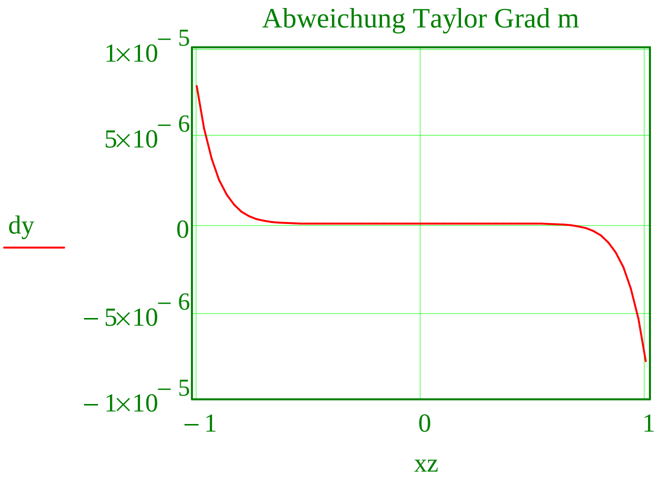
$nz := 60 \quad i := 0 .. nz \quad xz_i := \frac{2 \cdot i}{nz} - 1$

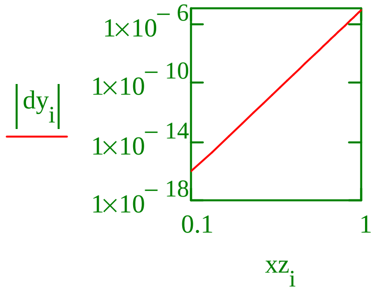
koef =

	0
0	0
1	1
2	0
3	-0.25
4	0
5	0.031
6	0
7	$-2.604 \cdot 10^{-3}$
8	0
9	$1.628 \cdot 10^{-4}$
10	0

$ysoll_i := g(xz_i)$ $ytay_i := p(xz_i)$ $dy := ysoll - ytay$ *Wird unsere Toleranz* ***eps*** *eingehalten? Ja wird sie!*

12	0
13	$3.391 \cdot 10^{-7}$
14	0
15	$-1.211 \cdot 10^{-8}$





2.3 Ausgleichspolynom

Wir legen wieder einen Satz von Stützwerten der Funktion ***g(x)*** *zugrunde, suchen aber jetzt ein Polynom von gegebenem Grad* ***m****, das den Ordinaten der Stützpunkte "möglichst nahe" kommt. In Ihrer* ***Vorbereitung*** *haben Sie sich eine mathematische Formulierung der zu erfüllenden Forderung überlegt:*

Zurch zusätzliche Überlegungen (oder einfach durch Probieren) lässt sich der notwendige Grad ***m*** *zum Einhalten der Toleranz* ***eps*** *bestimmen. In unserem Fall ergibt sich:*

$m := 7$ *Als Sollwerte verwenden wir* ***ns+1*** *Punkte:* $ns := 50$ nach Angabe Ihres Dozenten (3)

$xs := 0$ $ysoll := 0$ rücksetzen: $i := 0..ns$ $xs_i := \frac{2 \cdot i}{ns} - 1$ $ysoll_i := g(xs_i)$

*Stellen Sie nun das überbestimmte System der Bedingungsgleichungen (****ns+1*** *x* ***m+1****) auf und überführen Sie es in das System (****m+1*** *x* ***m+1****) der Normalgleichungen.*

$k := 0..m$ $As_{i,k} := (xs_i)^k$ Bedingungen $Aq := As^T \cdot As$ $yq := As^T \cdot ysoll$ Normalgleichungen

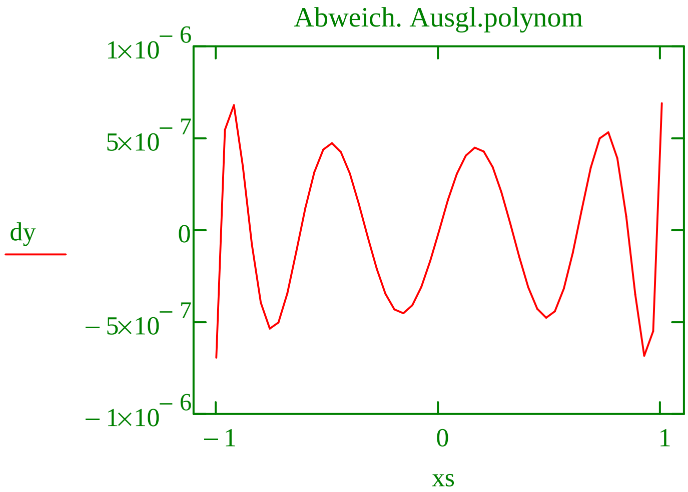
$condi(Aq) = 9.052 \times 10^4$ noch akzeptabel

Als Lösung der Normalgleichungen erhalten wir den Vektor ***c*** *der Polymomkoeffizienten:*

$c := Aq^{-1} \cdot yq$ $c^T = (0 \;\; 1 \;\; -0 \;\; -0.25 \;\; 0 \;\; 0.031 \;\; -0 \;\; -0.002)$

Durch Einsetzen der optimalen Lösung ***c*** *in die Bedingungsgleichungen ergeben sich die Funktionswerte* ***yp*** *des Ausgleichspolynoms an den Stützpunkten, daraus dann die Abweichungen* ***dy****.*

$yp := As \cdot c$ $dy := ysoll - yp$ Abweichung Soll-Ist



Diskutieren Sie ausführlich die Unterschiede zur Approximation nach Taylor.

1) Grad des Polynoms: Taylorreihe: Grad 9
Ausgleichspolynom: Grad 7 trotz kleineren Fehlers (+)

2) Fehlerverteilung: Taylorreihe: gegen Null im mitteleren Bereich und gegen obere Schranke am Rand
Ausgleichspolynom insgesamt sehr klein und annährend gleichmäßig verteilt (+)

3) Rechenaufwand: Taylorreihe: umständliches Berechnen der Ableitungen
Ausgleichspolynom: schnelles Lösen des LSG ohne nicht-lineare Formeln (+)

3 Fehlerabschätzungen bei linearen Gleichungssystemen

Wir haben in der Vorlesung Abschätzungen für die Auswirkung von Fehlern in der rechten Seite und in der Koeffizientenmatrix auf die Lösung linearer Gleichungssysteme betrachtet. Die Abhängigkeit der Lösung von Fehlern der rechten Seite soll hier für das Gleichungssytem bei der Polynominterpolation untersucht werden.

Wir addieren zu den Stützwerten yse statistisch verteilte Störungen dys um maximal p*yse. Wir erhalten die gestörten Stützwerte ysg.

$n := 12 \quad i := 0..n \quad p := 0.01 \quad dys_i := yse_i \cdot (-p + 2 \cdot rnd(p)) \quad ysg_i := yse_i + dys_i$

$yse^T =$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.5	0.691	0.805	0.819	0.628	0.232	0	0.232	0.628	0.819	0.805	...

$ysg^T =$

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.495	0.687	0.806	0.817	0.632	0.231	0	0.231	0.623	0.814	0.813	...

Mit den gestörten Stützwerten ysg berechnen wir die gestörten Polynomkoeffizienten cg und die Differenz dc zu den ungestörten Koeffizienten.

$cg := Ae^{-1} \cdot ysg \qquad dc := ce - cg$

$ce^T =$

	0	1	2	3	4
0	$2.603 \cdot 10^{-13}$	$1.37 \cdot 10^{-12}$	9.59	$-5.56 \cdot 10^{-12}$	...

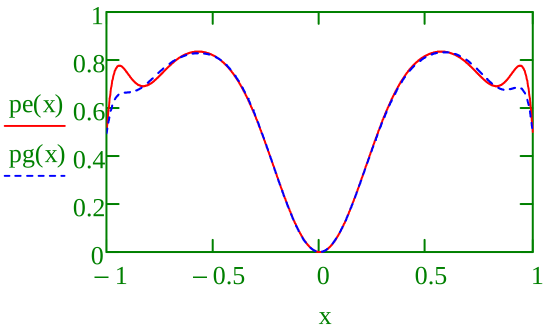
$cg^T =$

	0	1	2	3	4	5
0	$2.595 \cdot 10^{-13}$	0.016	9.511	-0.6	-46.587	...

Das Polynom pg zu den gestörten Stützstellen unterscheidet sich deutlich vom Polynom pe zu den exakten Stützstellen.

$$pe(x) := \sum_{i=0}^{n} \left(ce_i \cdot x^i\right) \qquad pg(x) := \sum_{i=0}^{n} \left(cg_i \cdot x^i\right)$$

Zu den Fehlerabschätzungen verwenden wir die unendlich-Norm. Die unendlich-Norm einer Matrix kann mit der Mathcad-Funktion normi bestimmt werden. Zur Berechnung der unendlich-Norm eines Vektors schreiben wir die Funktion normiv.



$$\mathrm{normiv}(v) := \left| \begin{array}{l} \text{for } i \in 0..\mathrm{length}(v)-1 \\ \quad vv_i \leftarrow |v_i| \\ \text{return } \max(vv) \end{array} \right.$$

$\mathrm{normi}(Ae) = 13$

$\mathrm{normi}\left(Ae^{-1}\right) = 4.55 \times 10^{4}$

$\mathrm{condi}(Ae) = 5.915 \times 10^{5}$

$\mathrm{normiv}(dys) = 7.863 \times 10^{-3}$ $\quad \mathrm{normiv}(dc) = 41.334$

$\dfrac{\mathrm{normiv}(dys)}{\mathrm{normiv}(yse)} = 9.597 \times 10^{-3}$ $\quad \dfrac{\mathrm{normiv}(dc)}{\mathrm{normiv}(ce)} = 0.179$

Tragen Sie die in der Vorlesung diskutierten Relationen (<, >, ...) zwischen den Größen ein. Sind die Relationen erfüllt? entfällt!

$\mathrm{normiv}(dc) = 41.334 \quad < \quad \mathrm{normi}\left(Ae^{-1}\right)\cdot \mathrm{normiv}(dys) = 357.753$

$\dfrac{\mathrm{normiv}(dc)}{\mathrm{normiv}(ce)} = 0.179 \quad < \quad \mathrm{condi}(Ae)\cdot\dfrac{\mathrm{normiv}(dys)}{\mathrm{normiv}(yse)} = 5.677 \times 10^{3}$