Richard Huber
richard.huber@uni-graz.at

11. Aufgabenblatt

Definition (Least Squares Approximation)**.** *Wie in früheren Beispielen gesehen sind Interpolationspolynome nur bedingt eine gute Approximation für eine gegebene Funktion $f\colon I \to \mathbb{R}$ auf einem Intervall $I \neq \emptyset$. Alternativ kann man das Polynom f^* suchen das eine möglichst gute Approximation bietet indem man das Least Square Approximation Problem*

$$f^* \in \operatorname*{argmin}_{g \in \mathcal{P}_N} \|f-g\|_{L^2}^2 = \int_I (f-g)^2 \,\mathrm{d}x \tag{1}$$

löst, wobei $\mathcal{P}_N$ die Menge der Polynome vom Grad kleiner gleich N ist und wir erneut die Darstellung $g(x) = \sum_{i=0}^N a_i x^i$ nutzen. Insbesondere hat (1) *immer eine eindeutige Lösung.*
In analogem Sinne kann man wenn nur endlich viele Datenpunkte $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^M$ (also Stellen $\mathbf{x}_i \in I$ und entsprechende Werte $f(\mathbf{x}_i)$) vorhanden sind, eine diskrete Approximation $\overrightarrow{f}^*$ bestimmen. Wenn man die Vektoren $\overrightarrow{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_M))^T$ und $\overrightarrow{g} = (g(\mathbf{x}_1), \dots, g(\mathbf{x}_M))^T$ betrachtet, kann diese Approximation wie folgt beschrieben werden:

$$\overrightarrow{f}^* \in \operatorname*{argmin}_{g \in \mathcal{P}_N} \|\overrightarrow{f} - \overrightarrow{g}\|_{l^2}^2 = \sum_{i=1}^M |f(x_i) - g(x_i)|^2. \tag{2}$$

Insbesondere hat (2) *eine eindeutige Lösung falls $M \geq N$ und $x_i \neq x_j$ für $i \neq j \in \{1, \dots, M\}$.*

Aufgabe 1 (Legendre Approximation)**.** *Wir betrachten die Legendre Polynome die iterativ für $n \in \mathbb{N}$ via*

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \qquad P_0(x) = 1 \quad \text{und} \quad P_1(x) = x \tag{3}$$

definiert sind. Die Legendre Polynome sind orthogonal zueinander bezüglich dem Skalarprodukt $\langle g, h\rangle_{L^2} = \int_{-1}^1 g(x)h(x)\,\mathrm{d}x$, d.h. $\langle P_n, P_j\rangle_{L^2} = 0$ für $n \neq j \in \mathbb{N}_0$. Es ist damit leicht zu zeigen, dass die Approximation f^* einer Funktion $f\colon [-1,1] \to \mathbb{R}$ durch Folgendes gegeben ist:

$$f^*(x) = \sum_{n=0}^N \frac{\langle f, P_n\rangle_{L^2}}{\langle P_n, P_n\rangle_{L^2}} P_n(x). \tag{4}$$

a) *Bestimmen Sie das 2. Legendre Polynom P_2, und überprüfen Sie rechnerisch die Orthogonalität von P_0, P_1, P_2, d.h. zeigen Sie $\langle P_i, P_j\rangle_{L^2} = 0$ für $i \neq j \in \{0,1,2\}$.*

b) *Gegeben sei die Funktion*

$$f\colon [-1,1] \to \mathbb{R} \qquad \text{mit} \qquad f(x) = \begin{cases} -1 & x \in [-1,0) \\ 1 & x \in [0,1]. \end{cases} \tag{5}$$

Nutzen Sie (4) *um die Approximation f^* von Grad $N = 2$ für die in* (5) *gegebene Funktion zu berechnen. Skizzieren Sie darüber hinaus die Funktion f sowie die errechnete Approximation f^* in $[-1,1]$.*

Hinweis. *Für die Berechnung der Koeffizienten $\langle f, P_n\rangle_{L^2}$ kann es nützlich sein das Integral in zwei Integrale auf $[-1,0]$ und $[0,1]$ aufzuspalten.*

Bemerkung. *Das Beispiel zeigt, dass Least Squares Approximation sich durch die Nutzung der Legendre Polynome einfach und algorithmisch vornehmen lässt, insbesondere wenn die Legendre Polynome bereits vorberechnet wurden. Allerdings ist die Berechnung der klassischen Koeffizientenschreibweise des Polynoms wieder aufwändig, da die Legendre Polynome viele nicht-Null Koeffizienten haben.*

Aufgabe 2 (Approximation diskreter Punkte)**.** *Gegeben seien Punkte* $\mathbf{x} = [-2, -1, 0, 1, 2]$.

a) *Es sei ein Polynom* $g(x) = a_0 x^0 + a_1 x^1 + a_2 x^2$ *mit zugehörigen Koeffizientenvektor* $a = [a_0, a_1, a_2]^T$ *gegeben. Bestimmen Sie eine Matrix* B *sodass* $\overrightarrow{g} = Ba$ *gilt, wobei* $\overrightarrow{g} = (g(\mathbf{x}_1), \ldots, g(\mathbf{x}_5))^T$.

b) *Das diskrete Approximationsproblem* (2) *ist damit äquivalent zu dem Ausgleichsproblems* $\operatorname{argmin}_{a \in \mathbb{R}^{N+1}} \|\overrightarrow{f} - Ba\|_{l^2}^2$ *in dem Sinne dass für die Approximation* $\overline{f}^*$ *die Darstellung* $\overline{f}^*(x) = \sum_{i=0}^N a_i^* x^i$ *gilt wenn* a^* *Lösung des Augleichsproblems ist. Falls* B *injektiv ist, gilt dass* $B^T B a^* = B^T \overrightarrow{f}$ *eine notwendige und hinreichende Bedingung dafür ist, dass* a^* *das Ausgleichsproblem löst.*

Gegeben sei die Funktionswerte $\overrightarrow{f} = [-3, 2, 3, 0, 3]$ *zu* $\mathbf{x}$ *wie oben. Bestimmen Sie den Koeffizientenvektor* a^* *und das dazugehörige Polynom* $\overline{f}^*$*, das das Approximationsproblem* (2) *für* $N = 2$ *löst.*

Bemerkung. *Dieses Beispiel zeigt, wie sich das diskrete Approximationsproblem als Ausgleichsproblem schreiben lässt, welches wiederum dank den Gauß Gleichungen* $B^T B a^* = B^T \overrightarrow{f}$ *algebraisch lösbar ist. Insbesonders lässt sich dieser Ansatz für beliebige* M, N *mit* $M \geq N$ *veralgemeinern.*

Matlab-Aufgabe 3 (Approximation diskreter Punkte Fortsetzung)**.** *Wir wollen das Rechenbeispiel 11.2 zu Computercode für algemeine* N *und* M *weiterentwickeln.*

a) *Veralgemeinern Sie den Ansatz aus Aufgabe 11.2: Wie sieht für algemeines* $\mathbf{x}$ *die zugehörige Matrix* B *aus. Schreiben Sie den Pseudocode eines Computeprogrammes [a*]=my_approximation*$(\mathbf{x}, \mathbf{y}, N)$*, welcher für die Eingabe von* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_M)$ *und* $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_M)$ *das Approximationsproblem* (2) *in* $\mathcal{P}_N$ *löst, und den entsprechenden Koeffizientenvektor* a^* *wieder zurückgibt.*

b) *Implementieren Sie [a*]=my_approximation*$(\mathbf{x}, \mathbf{y}, N)$ *als Matlab-Funktion.*

c) *Schreiben Sie ein Matlab-Skript main11_2.m, welches die Approximation für* $N = 0, 2, 4, 6, 8, 10$ *und* $M = 51$ *mit äquidistanten* $(\mathbf{x}_i)_{i=1}^M$ *in [-5,5] und* $f(x) = \frac{1}{\sqrt{1+x^2}}$ *berechnet, und plotten Sie die Funktion* f *und Approximation gegeneinander, siehe Abbildung unten.*

Hinweis. *Sie können den Backslash Operator nutzen um die resultierenden Gauß-Gleichungen zu lösen. Außerdem können Sie die Funktion evaluate_polynom.m verwenden.*

Bemerkung. *Wie Aufgabe b) zeigt erhält man mittels Approximationsansätzen im Gegensatz zur Interpolation in Beispiel 11.3c hier sehr hohe Approximationgüte, welche mit zunehmendem Polynomgrad nur besser wird.*

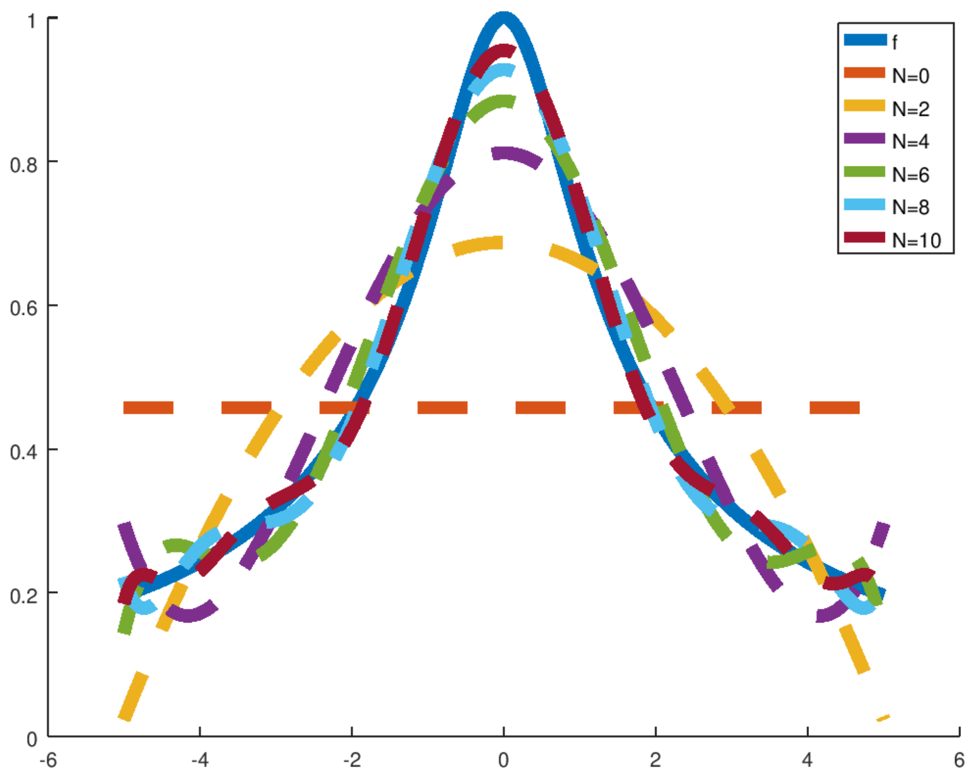

Abbildung 1: Funktion f und Least Squares Approximationen $\overline{f}^*$ für $N \in \{0, 2, 4, 6, 8, 10\}$