Numerische Methoden Serie 3

Tim Launer

1. Gauss-Legendre Quadratur und Golub-Welsch Algorithmus

a) 3-Term-Rekursion für Legendre Polynome $P_n(x)$, wobei das Slakarprodukt gegeben ist durch:

$$\langle p, q\rangle = \int_{-1}^{1} p(x)q(x)dx \tag{1}$$

und $P_0(x) = 1$ und $P_1(x) = x$ gilt.
Nun wollen wir die ersten 4 Polynome herleiten, indem wir die Formel aus der Vorlesung

$$P_{k+1}(x) = (x - \beta_{k+1})P_k(x) - \gamma_{k+1}^2 P_{k-1}(x) \tag{2}$$

benutzen, wobei $\beta_{k+1} = \frac{\langle xP_k, P_k\rangle}{\langle P_k, P_k\rangle}$ und $\gamma_{k+1}^2 = \frac{\langle P_k, P_k\rangle}{\langle P_{k-1}, P_{k-1}\rangle}$ gilt. Es folgt:

$$P_2(x) = (x - \frac{\langle x^2, x\rangle}{\langle x, x\rangle}) \cdot x - \frac{\langle x, x\rangle}{\langle 1, 1\rangle} \cdot 1$$

Also rechnen wir die Skalarprodukte aus:

$$\langle x^2, x\rangle = \int_{-1}^{1} x^3 dx = \left[\frac{1}{4}x^4\right]_{-1}^{1} = 0$$

$$\langle x, x\rangle = \int_{-1}^{1} x^2 dx = \left[\frac{1}{3}x^3\right]_{-1}^{1} = \frac{2}{3}$$

$$\langle 1, 1\rangle = \int_{-1}^{1} 1 dx = [x]_{-1}^{1} = 2$$

Daraus ergibt sich für das dritte Legendre Polynom:

$$P_2(x) = x^2 - \frac{\frac{2}{3}}{2} \cdot 1 = x^2 - \frac{1}{3}$$

Dieses Ergebnis können wir nun verwenden, um auch das vierte Legendre Polynom mithilfe von Gleichung (2) zu berechenen:

$$P_3(x) = (x - \frac{\langle x^3 - \frac{1}{3}x, x^2 - \frac{1}{3}\rangle}{\langle x^2 - \frac{1}{3}, x^2 - \frac{1}{3}\rangle}) \cdot (x^2 - \frac{1}{3}) - \frac{\langle x^2 - \frac{1}{3}, x^2 - \frac{1}{3}\rangle}{\langle x, x\rangle} \cdot x$$

Wieder müssen wir die Integrale lösen:

$$\langle x^3 - \frac{1}{3}x, x^2 - \frac{1}{3}\rangle = \int_{-1}^{1} x^5 - \frac{2}{3}x^3 + \frac{x}{9} dx = \left[\frac{1}{6}x^6 - \frac{1}{6}x^4 + \frac{x^2}{18}\right]_{-1}^{1} = 0$$

$$\langle x^2 - \frac{1}{3}, x^2 - \frac{1}{3}\rangle = \int_{-1}^{1} x^4 - \frac{2}{3}x^2 + \frac{1}{9} dx = \left[\frac{1}{5}x^5 - \frac{2}{9}x^3 + \frac{x}{9}\right]_{-1}^{1} = \frac{8}{45}$$

Daraus ergibt sich nun für das vierte Legendre Polynom:

$$P_3(x) = x^3 - \frac{1}{3}x - \frac{\frac{8}{45}}{\frac{2}{3}} \cdot x = x^3 - \frac{1}{3}x - \frac{24}{90}x = x^3 - (\frac{5}{15}x + \frac{4}{15}x) = x^3 - \frac{3}{5}x$$

Zuletzt müssen wir alle Polynome noch normieren, denn es soll $P_k(1) = 1$ gelten (Wikipedia):

$$P_2(1) = C1^2 - \frac{C}{3} \cdot 1 = 1 \;\Rightarrow\; C = \frac{3}{2} \;\Rightarrow\; P_2(x) = \frac{3}{2}x^3 - \frac{1}{2}$$

$$P_3(1) = \lambda 1^3 - \frac{3\lambda}{5} \cdot 1 = 1 \;\Rightarrow\; \lambda = \frac{5}{2} \;\Rightarrow\; P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$$

Die Polynome müssen normiert werden, da es sich bei ihnen ja um eine orthogonale Familie von Polynomen handeln soll. Das bedeutet, dass immmer $\langle P_n(x), P_m(x)\rangle = 0$, wenn $n \neq m$ gelten soll (Definition "orthogonal"). Das erklärt auch, warum $P_0(x) = 1$ gilt. Weiter wissen wir aus der Uebungsstunde, dass dies bez. einem Skalarpodukt $\langle \cdot, \cdot \rangle$ auch folgendermassen formuliert werden kann:

$$\langle P_n(x), P_m(x)\rangle = \frac{2}{2n+1} \cdot \delta_{n,m}$$

Wir bemerken, dass man daraus zusammen mit Gleichung (1) die 3-Term-Rekursionsformel erhält:

$$P_{k+1}(x) = \frac{2k+1}{k+1} x P_k(x) - \frac{k}{k+1} P_{n-1}(x) \tag{3}$$

Alternativ kann man von Wikipedia die Formel $(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$ verwenden zsm mit einer Substitution $n = n+1$.
Wir berechnen:

$$\begin{aligned} (n+1)P_{n+1}(x) &= (2n+1)xP_n(x) - nP_{n-1}(x) \\ nP_n(x) &= (2(n-1)+1)xP_{n-1}(x) - (n-1)P_{n-1}(x) \\ \Rightarrow P_n(x) &= \frac{2(n-1)}{n} xP_{n-1}(x) - \frac{n-1}{n} P_{n-1}(x) \end{aligned}$$

Das ist genau wieder Glaichung (3).

b) Begründe die Wahl der **Einträge der Jacobi Matrix J im Golub-Welsch Algorithmus**, wobei

$$J = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & \cdots & \cdots & \beta_{n-1} & \alpha_n \end{bmatrix}$$

mit $\alpha_k = -\frac{b_k}{a_k}$, $\beta_k = \sqrt{\frac{c_{k+1}}{a_k \cdot a_{k+1}}}$.

Aus der Uebungsstunde wissen wir, dass die n Nullstellen von einem Legendre Polynom $P_n(x)$ genau den Eigenwerten der Matrix J entsprechen. Für die Legendre Polynome wissen wir ausserdem auch der Uebungsstunde, dass zusätzlich zur oben gefundenen Gleichung (3) auch gilt:

$$P_n(x) = (a_n x + b_n)P_{n-1}(x) - c_n P_{n-2}(x)$$

Die Koeffizienten a_n, b_n, c_n kann man nun daraus über die Rekursionsformel erhalten, indem man in Gleichung (3) $k+1=n$ setzt:

$$P_n(x) = \frac{2(n-1)}{n} x P_{n-1}(x) - \frac{n-1}{n} P_{n-1}(x)$$

Daraus ergibt sich $\frac{2n-1}{n}x = a_n x + b_n \Rightarrow b_n = 0$, $a_n = \frac{2n-1}{n}$ und $c_n = \frac{n-1}{n}$. Eingesetzt in die Gleichungen für α_k und β_b ergibt sich schlussendlich:

$$b_k = 0, \forall k \;\Rightarrow\; \alpha_k = 0$$

$$\beta_k = \sqrt{\frac{\frac{k}{k+1}}{\frac{2k-1}{k} \cdot \frac{2(k+1)-1}{k+1}}} = \frac{k}{\sqrt{4k^2-1}}$$

Das sind ja genau die Einträge im Code für den Golub-Welsch Algorithmus 1.3.3!