Numerieke Modellering en Benadering: Practicum Gedeelte Benaderingstheorie

Academiejaar 2019-2020

1 Een continue kleinste-kwadratenbenadering

In deze opgave zullen we een continue functie benaderen met behulp van orthogononale veeltermen, gebruikmakend van het continue kleinste-kwadratencriterium. De integralen die hiervoor berekend moeten worden, zullen we numeriek benaderen met behulp van de discrete cosinustransformatie.

1.1 Chebyshev-veeltermen en Chebyshev-benadering

In de cursus werd de algemene formule afgeleid voor het benaderen van een functie $f(x)$ op $[a, b]$:

$$f(x) \approx y_n(x) = \sum_{k=0}^{n} a_k \phi_k(x) \quad \text{met } a_k = \frac{(f, \phi_k)}{\|\phi_k\|^2}. \tag{1}$$

De functies $\phi_k(x)$ vormen een rij orthogonale veeltermen en $(f, g) = \int_a^b w(x)f(x)g(x)dx$. Het is de bedoeling in dit deel van het practicum deze formules te implementeren voor het concrete geval waarbij $[a, b] = [-1, 1]$ en $w(x) = 1/\sqrt{1 - x^2}$. De veeltermen $\phi_k(x)$ noemt men dan *Chebyshev-veeltermen* en men stelt ze voor als $T_k(x)$. Dit zijn wellicht de best bestudeerde veeltermen in de wiskunde. Men kan bijvoorbeeld aantonen dat de drietermsrecursiebetrekking vereenvoudigt tot

$$T_0 = 1, \quad T_1 = x \quad \text{en} \quad T_{k+1} = 2xT_k - T_{k-1} \text{ voor } k > 0.$$

Er bestaat zelfs een expliciete formule voor deze veeltermen en voor hun norm:

$$T_k(x) = \cos(k \arccos(x)), \quad \|T_0\|^2 = \pi \quad \text{en} \quad \|T_k\|^2 = \frac{\pi}{2} \text{ voor } k > 0.$$

Gebruikmakend van deze eigenschappen kan men de formule voor a_k, met $k > 0$, verder uitwerken:

$$a_k = \frac{(f, T_k)}{\|T_k\|^2} = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x)\cos(k \arccos(x))}{\sqrt{(1 - x^2)}} dx = \frac{2}{\pi} \int_0^{\pi} f(\cos(\theta)) \cos(k\theta) d\theta.$$

Gebruiken we diezelfde formule ook voor a_0, dan krijgen we een waarde die eigenlijk met een factor twee te groot is. In de lineaire combinatie (1) wordt de eerste term door twee gedeeld om dit te compenseren, wat we als volgt noteren:

$$f(x) \approx y_n(x) = \sum_{k=0}^{n}{}' a_k T_k(x). \tag{2}$$

De integraal in de berekening van a_k zal vaak niet analytisch uit te werken zijn. Daarom stellen we een numerieke benadering op aan de hand van de *trapeziumregel*. Dit geeft de uitdrukking

$$a_k \approx \frac{2}{N} \sum_{n=0}^{N}{}'' f\left(\cos\left(\frac{\pi n}{N}\right)\right) \cos\left(\frac{\pi k n}{N}\right),$$

waarbij de notatie $\sum''$ erop duidt dat zowel de eerste als de laatste term in de som gehalveerd moeten worden. Als we deze formule vergelijken met die van de discrete cosinustransformatie waarbij $f(x)$ wordt geëvalueerd in de punten $\cos(\pi n/N)$, dan blijken ze op een constante factor na hetzelfde! Dit laat ons toe de coëfficiënten a_k te berekenen met het snelle FFT-algoritme. Uiteraard moeten na het uitvoeren van het FFT-algoritme de eerste en laatste termen niet meer gehalveerd worden (ga dit na in Hoofdstuk 4 van de cursus).

1.2 Opgaven

1. Schrijf een functie `function` [`T`] = `cheb(n)` die met behulp van de recursiebetrekking de rij van Chebyshev-veeltermen opstelt van graad nul tot graad n. De functie geeft als uitvoer een matrix met daarin de coëfficiënten van de termen in de veelterm. Elke opeenvolgende rij daarin hoort bij een veelterm T_k van de volgende graad, de i-de kolom geeft de coëfficiënt die hoort bij de i-de graadsterm van T_k:

$$T = \begin{bmatrix} 1 & & & \\ 0 & 1 & & \\ -1 & 0 & 2 & \\ & & & \ddots \end{bmatrix}$$

   Schrijf een functie `function` [`c`] = `poly(a, T)` die voor een gegeven vector a met a_k-waarden de klassieke veeltermcoëfficiënten van de veelterm $y_n(x)$ in (2) berekent.

   Test de correctheid van deze functies en teken ter illustratie enkele van die Chebyshev-veeltermen.

2. Schrijf nu een functie `function` [`a`] = `chebcoeff(f_handle, N)` voor het bepalen van de coëfficiënten a_k in benadering (2). Het argument `f_handle` is een function handle (bvb.: `f = @(x) x - 1`) die kan worden gebruikt om de functie te evalueren in de nodige punten en N is de maximale graad van de Chebyshev veeltermen. Merk op dat het MATLAB-commando `dct` niet precies overeenkomt met de trapeziumregel zoals hierboven beschreven. We maken daarom een even uitbreiding van de bemonsteringen: $[f_0, f_1, \ldots, f_{N-1}, f_N, f_{N-1}, \ldots, f_1]$ en gebruiken het commando `fft`.

3. Benader met je routine de functie $f_1(x) = \frac{x-1}{1+6x^2}$ Illustreer de convergentie-snelheid van de benadering naar $f(x)$ in functie van de graad van de benadering en plot een aantal benaderingen van verschillende graad. Doe hetzelfde voor de functies $f_2(x) = \ln(x+2)\sin(10x)$ en $f_3(x) = x^5 - x^4 + x^3 - x^2 + x - 1$. Vergelijk en verklaar de convergentiesnelheid van de benadering voor de drie functies.

2 Bivariate kleinste-kwadratenveeltermbenadering

Een discrete kleinste-kwadratenbenadering kan op twee manieren berekend worden: met het normaalstelsel uit de benaderingstheorie of met het overgedetermineerde stelsel uit de numerieke lineaire algebra. In de cursus toonden we de equivalentie van beide methodes aan voor ééndimensionale benaderingen. In deze opgave werken we nu de tweede methode verder uit voor een tweedimensionaal benaderingsprobleem.

2.1 De methode van het overgedetermineerde stelsel

We wensen een oppervlak te benaderen waarvan de functiewaarden f_{ij} gegeven zijn op een rechthoekig puntenrooster (x_i, y_j), met $i = 1, \ldots, M$ en $j = 1, \ldots, N$. Als benaderende functie gebruiken we een bivariate veelterm $z_{mn}(x, y)$ van graad m in x en van graad n in y, voorgesteld als

$$z_{mn}(x,y) = \sum_{k=0}^{m}\sum_{l=0}^{n} c_{kl}\phi_{kl}(x,y) \quad \text{met} \quad \phi_{kl} = p_k(x)q_l(y).$$

We zullen bij de berekeningen verder de klassieke monomiale basis nemen: $p_k(x) = x^k$ en $q_l(y) = y^l$. Als benaderingscriterium kiezen we dat van de kleinste-kwadraten. De coëfficiënten c_{kl} worden dan zo gekozen dat de benadering de volgende uitdrukking minimaliseert:

$$\|f - z_{mn}\|_2^2 = \sum_{i=0}^{M}\sum_{j=0}^{N} \left(f(x_i, y_j) - z_{mn}(x_i, y_j)\right)^2 .$$

We bekomen een overgedermineerd stelsel door te eisen dat de benadering de functie in alle MN roosterputen interpoleert. De vergelijking horende bij het punt (x_i, y_j) is dan van de vorm

$$\sum_{k=0}^{m}\sum_{l=0}^{n} c_{kl}p_k(x_i)q_l(y_j) = f_{ij}.$$

Ordenen we deze MN vergelijkingen onder elkaar, dan kunnen we het overgedetermineerde stelsel noteren als $\mathbf{\Gamma c} = \mathbf{f}$. De rechthoekige matrix $\mathbf{\Gamma}$ heeft MN rijen en mn kolommen. Elke rij van het stelsel komt overeen met een interpolatievoorwaarde in één van de roosterpunten. Elke kolom van $\mathbf{\Gamma}$ komt overeen met één van de onbekenden c_{kl} uit de vector $\mathbf{c} \in \mathbb{R}^{(m+1)(n+1)}$. In het rechterlid staat de vector $\mathbf{f} \in \mathbb{R}^{MN}$ met de functiewaarden f_{ij} . De matrix $\mathbf{\Gamma}$ krijgt een heel eenvoudige vorm als we de roosterpunten en de coëfficiënten op een goede manier ordenen. De volgorde van de vergelijkingen volgt uit onderstaande voorstelling voor de vector $\mathbf{f}$:

$$\mathbf{f} = [f_{11} \quad \cdots \quad f_{M1} \quad f_{12} \quad \cdots \quad f_{M2} \quad \cdots \quad f_{1N} \quad \cdots \quad f_{MN}]^{\mathrm{T}},$$

die de vectorisatie $\mathrm{vec}(\mathbf{F})$ voorstelt van de matrix $\mathbf{F}$, gedefinieerd als:

$$\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ f_{21} & f_{22} & \cdots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \cdots & f_{MN} \end{bmatrix},$$

waarvan de kolommen dus onder elkaar geplaatst worden. Ook de kolomvector $\mathbf{c}$ kan gezien worden als de vectorisatie van een matrix $\mathbf{C} \in \mathbb{R}^{(m+1)\times(n+1)}$ van coëfficiënten c_{ij}, dus $\mathbf{c} = \mathrm{vec}(\mathbf{C})$. Schrijf nu zelf eens de matrix neer voor een benaderingsprobleem op een klein rooster en controleer zodoende dat de matrix $\mathbf{\Gamma}$ heel compact kan geschreven worden als $\mathbf{\Gamma} = \mathbf{B} \otimes \mathbf{A}$, met

$$\mathbf{A} = \begin{bmatrix} p_0(x_1) & p_1(x_1) & \cdots & p_m(x_1) \\ p_0(x_2) & p_1(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_0(x_M) & p_1(x_M) & \cdots & p_m(x_M) \end{bmatrix} \quad \text{en} \quad \mathbf{B} = \begin{bmatrix} q_0(y_1) & q_1(y_1) & \cdots & q_n(y_1) \\ q_0(y_2) & q_1(y_2) & \cdots & q_n(y_2) \\ \vdots & \vdots & \ddots & \vdots \\ q_0(y_N) & q_1(y_N) & \cdots & q_n(y_N) \end{bmatrix}.$$

In deze notatie maken we gebruik van het zogenaamde Kroneckerproduct. Het Kroneckerproduct van twee matrices $\mathbf{P} \in \mathbb{R}^{s\times t}$ en $\mathbf{Q} \in \mathbb{R}^{u\times v}$ is de blokmatrix

$$\mathbf{P} \otimes \mathbf{Q} = \begin{bmatrix} p_{11}\mathbf{Q} & p_{12}\mathbf{Q} & p_{1t}\mathbf{Q} & \\ p_{21}\mathbf{Q} & p_{22}\mathbf{Q} & p_{2t}\mathbf{Q} & \\ \vdots & \vdots & \ddots & \vdots \\ p_{s1}\mathbf{Q} & p_{s2}\mathbf{Q} & p_{st}\mathbf{Q} & \end{bmatrix} \in \mathbf{R}^{su\times tv},$$

waarbij p_{ij} het element van $\mathbf{P}$ op de i-de rij en de j-de kolom voorstelt. Enkele eigenschappen die belangrijk zijn voor deze opgave zijn:

$$(\mathbf{P} \otimes \mathbf{Q})^{\dagger} = \mathbf{P}^{\dagger} \otimes \mathbf{Q}^{\dagger} \quad \text{en} \quad (\mathbf{P} \otimes \mathbf{Q})\,\mathrm{vec}(\mathbf{F}) = \mathrm{vec}(\mathbf{QFP}^{\mathrm{T}}),$$

waarbij $(.)^{\dagger}$ de Moore–Penrose pseudoinverse voorstelt. Het bivariate kleinste-kwadratenprobleem kan nu geherformuleerd worden als volgt:

$$\text{Vind } \mathbf{c} \in \mathbb{R}^{(m+1)(n+1)} \quad \text{zodat} \quad \mathbf{c} = \mathrm{argmin}\|\mathbf{\Gamma c} - \mathbf{f}\|_2^2.$$

De oplossing kan geschreven worden als $\mathbf{c} = \mathbf{\Gamma}^{\dagger}\mathbf{f}$. Gebruikmakend van de eigenschappen van het Kroneckerproduct, vinden we nu in enkele stappen de veeltermcoëfficiënten, in vector- en matrixvorm:

$$\mathbf{c} = (\mathbf{B} \otimes \mathbf{A})^{\dagger}\,\mathbf{f} = \left(\mathbf{B}^{\dagger} \otimes \mathbf{A}^{\dagger}\right)\mathrm{vec}(\mathbf{F}) = \mathrm{vec}\left(\mathbf{A}^{\dagger}\mathbf{F}\mathbf{B}^{\dagger^{\mathrm{T}}}\right), \quad \text{of nog:} \quad \mathbf{C} = \mathbf{A}^{\dagger}\mathbf{F}\mathbf{B}^{\dagger^{\mathrm{T}}}.$$

2.2 Opgave

1. Schrijf een functie `function [C] = kkb2d(x,y,F,m,n)` die de coëfficiëntenmatrix $\mathbf{C}$ berekent, gegeven de vectoren $x = [x_1 \ldots x_M]$ en $y = [y_1 \ldots y_N]$, waarvan het cartesisch product het puntenrooster vormt, de matrix $\mathbf{F}$ met functiewaarden en de veeltermgraden m en n. Welke testen heb je uitgevoerd om de correctheid van je code te controleren?

2. We gebruiken $M = N = 31$ equidistante roosterpunten in de x- en y-richting om twee verschillende functies $f(x, y)$ op het vierkant $[-1, 1] \times [-1, 1]$ te benaderen:

   (a) $f(x, y) = \sin((2x - 1)^2 + 2y)$,

(b) $f(x, y) =$ de MATLAB-functie waarvoor de functiewaarden op het puntenrooster gegenereerd worden door $\mathbf{F} =$ `membrane(1)`.

Stel voor elk de veeltermbenadering op van graad m=n=7. Teken per dataset één figuur, waarin je de punten plot tesamen met het benaderende oppervlak. Enkele nuttige MATLAB-functies zijn `meshgrid`, `polyval2`, `scatter3` en `surf`. De functie `polyval2` vind je op Toledo.

3. Plot de opeenvolgende waarden van de benaderingsfout $\|f - z_{mn}\|^2$ voor $m = n = 1, \ldots, 20$ voor de twee functies op eenzelfde figuur. Voor welke dataset zijn veeltermoppervlakken een geschikte benadering? Hoe zou je het andere oppervlak beter kunnen benaderen?

4. Benader de functie van Runge $f(x, y) = \frac{1}{1+25(x^2+y^2)}$ en ga de invloed na van de ligging van de roosterpunten (x_i, y_j). Gebruik eerst de equidistante verdeling: `linspace(−1, 1, 30)` voor zowel x en y, en vergelijk dit daarna met de niet-equidistante verdeling:

   [`linspace(−1, −0.20, 5)`, `linspace(−0.18, 0.18, 20)`, `linspace(0.20, 1, 5)`].

   Merk op dat we in beide gevallen evenveel roosterpunten gebruiken. Welk rooster is te verkiezen en waarom? Wat is de maximale afwijking tussen de benadering en de opgegeven functie in de roosterpunten voor beide roosters?

5. De ASTER Global Digital Elevation Map (`https://gdex.cr.usgs.gov/gdex/`) is een product van METI en NASA en brengt de hele wereld in kaart onder de vorm van discrete hoogtegegevens. Het bestand `etna.jpg` dat je op Toledo kan vinden, bevat een kaart van een rechthoekig gebied rond de vulkaan Etna op Sicilië. Elke pixel van de afbeelding bevat een hoogtewaarde. Je kan de data inlezen als een matrix $\mathbf{F}$ via het commando `F = double(imread('etna.jpg'))`.

   Stel nu een veeltermbenadering op van de Etna. Gebruik hiertoe `x = linspace(−1, 1, M)` en `y = linspace(−1, 1, N)` waarbij M en N de gepaste waarden hebben. Na uitvoering van het commando `[X, Y] = meshgrid(x, y)`, kan je de data mooi weergeven met

   ```
   surf(X,Y,F,'EdgeColor','none','LineStyle','none','FaceLighting','phong');
   xlim([-1,1]);
   ylim([-1,1]);
   zlim([0,250]);
   title('Etna');
   ```

   Met het commande `imagesc(F)` genereer je een bovenaanzicht. Doe hetzelfde voor het benaderende veeltermoppervlak met $m = n = 25$ en voeg de resultaten toe aan het verslag.

3 Praktische schikkingen

- Het practicum wordt gemaakt in groepjes van twee. Desnoods wordt er één groepje van drie gevormd. Op het examen kan het practicum aan bod komen tijdens het mondelinge deel. Het is dus niet de bedoeling dat je de opdrachten verdeelt en slechts van een deel van de oplossing op de hoogte bent.
- Het verslag kan je doorsturen via mail. De deadline is vrijdag 29 mei 2020.
- Bespreek je resultaten bondig en illustreer met verduidelijkende figuren. Probeer de lengte onder de tien pagina's te houden. Lever ook je MATLAB-code (de m-files) in van de functies en van de scripts die je gebruikt hebt om de resultaten te genereren. Als alternatief kan je ook een MATLAB live script (.mlx) indienen dat het verslag en de code combineert. Zorg ervoor dat je ook in dat geval genoeg duiding geeft bij de bekomen resultaten.

Succes!

Michiel Vandecappelle (A329)
michiel.vandecappelle@kuleuven.be