

Numerieke Modelling & Benadering Practicum

Dennis Debree and Jonas Bertels

KU Leuven Departement Computerwetenschappen

11 april 2019

1 Bivariate Veelterm Benadering

1.1 Inleiding

Voor het vak numerieke modellering en benadering aan de KU Leuven werd de opdracht gegeven een bivariate veelterm benadering te implementeren in MATLAB. Deze benadering werd dan gebruikt om enkele 2 dimensionale oppervlakken te benaderen. Dit verslag geeft een visueel overzicht van de benaderingen en hun oppervlak en van de nauwkeurigheid van de benadering afhankelijk van de graad van de benaderende veelterm.

1.2 De basisfunctie

```
function C = kkb(x, y, F, m, n)
Afull = fliplr(vander(x)); % we do it this way because it saves us from
Bfull = fliplr(vander(y)); % calculating the inverse twice
C = reshape(kron(Afull(:, 1:m), Bfull(:, 1:n))\F(:), [n,m]);
end
```

Deze functie creëert eerst de A en B Vandermonde matrices om hieruit de x en y waarden te halen. Daarna wordt uit deze A en B matrices de F matrix berekend door eerst de graad van de A en B matrices te verkorten tot m en n en hierna C te bereken door de formule $C = B^+ F (A^+)^T$ om te zetten. Dit gebeurt als volgt:

$$C = B^+ F (A^+)^T \quad (1)$$

We weten ook dat:

$$(A \otimes B)^+ = A^+ \otimes B^+ \quad (2)$$

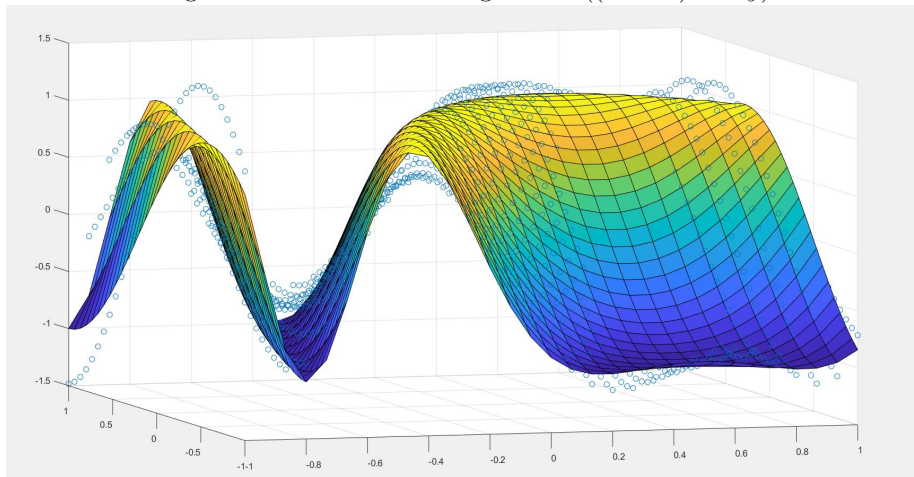
$$(A \otimes B) \text{vec}(F) = \text{vec}(BFA^T) \quad (3)$$

Uit (2) en (3) kunnen we dan afleiden dat

$$\text{vec}(C) = \text{vec}(B^+ F (A^+)^T) = (A^+ \otimes B^+) \text{vec}(F) = (A \otimes B)^+ \text{vec}(F) \quad (4)$$

De reden dat oorspronkelijk (4) werd gebruikt in plaats van (1) was dat we dachten dat (4) ons toeliet om slechts 1 inverse matrix uit te rekenen in plaats van 2 en zo de hoeveelheid benodigd werk doet dalen. Dit was natuurlijk fout

Figuur 1: Functiebenadering van $\sin((2x - 1)^2 + 2y)$



omdat $A \otimes B$ dimensies heeft die het product zijn van de A en B matrices apart. Toen de functiebenadering van de Etna moest gegenereerd worden bleek al snel dat het gebruik van (4) helemaal geen optimalisatie was in termen van geheugen. Het Kronecker product zorgde er namelijk voor dat matrices zo groot werden dat ze niet meer in het geheugen pasten, en dat bij kleinere waarden van m en n het verschillende minuten duurde voordat de matrix gegenereerd werd (dit is logisch want bij een $m = n = 25$ is $\dim(A) = 1425 \times 25$ en $\dim(B) = 1425 \times 25$ waardoor het Kronecker product een dimensie van 2030625×625 had).

1.3 2 functiebenaderingen

We gebruiken nu de kkbtest functie om enkele 2-D oppervlakken te benaderen. De gegeven functie polyval laat ons toe aan de hand van het resultaat van onze kkb-functie de benaderde waarden voor x en y te evalueren. Op de 2 figuren is de functie voorgesteld als een oppervlak en de benaderde waarden als punten (de blauwe cirkels) die boven of onder dit oppervlak verschijnen.

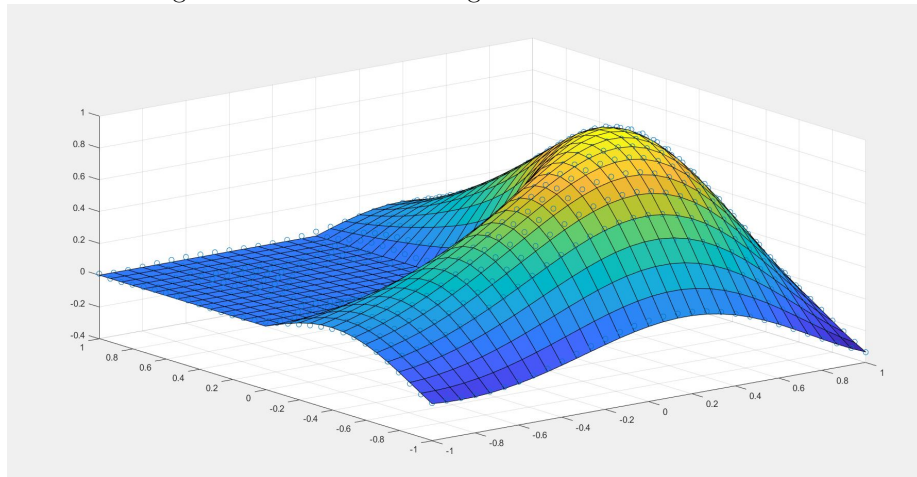
1.4 De kostfunctie afhankelijk van de graad van de benaderende veelterm

De volgende stap is het effect bepalen van de graad van de benaderende veelterm op de nauwkeurigheid van de benadering. Om dit te doen gebruikten we de volgende formule: $\sum_{i=1}^M \sum_{j=1}^N (f_{ij} - z(x_i, y_j))^2$.

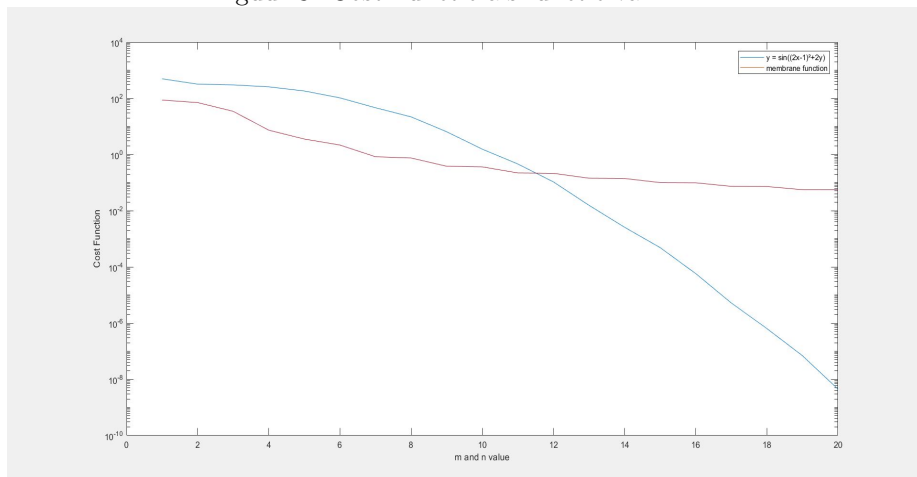
1.5 Veeltermbenadering Etna

De veeltermbenadering van de Etna werd op praktisch dezelfde manier geïmplementeerd als de functiebenaderingen in oefening 1.2. Zoals al eerder vermeld kwam het door deze functiebenadering dat we begrepen dat onze oorspronkelijk implementatie van kkb niet optimaal was.

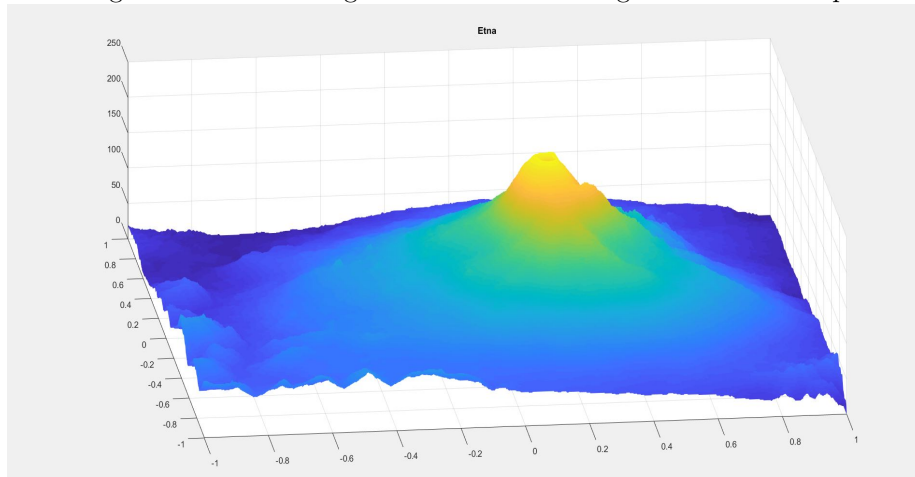
Figuur 2: Functiebenadering van de membraan functie



Figuur 3: Cost Function als functie van $n=m$



Figuur 4: De Etna volgens ASTER Global Digital Elevation Map



Figuur 5: Functiebenadering van de Etna met $m = n = 25$

