

**REPUBLIKA E SHQIPERISË
UNIVERSITETI POLITEKNIK, TIRANË
FAKULTETI I INXHINIERISË MATEMATIKE DHE INXHINIERISË FIZIKE
DEPARTAMENTI I INXHINIERISË MATEMATIKE**

DISERTACION

**PËR MARRJEN E GRADËS SHKENCORE
DOKTOR**

**“STUDIM MBI TEKNIKAT E PËRAFRIMIT JO UNIFORM ME ZBATIME NË
METODAT INTERPOLUESE DHE PSEUDOSPEKTRALE”**

Paraqitur nga:

M.Sc. Alvin ASIMI

Udhëheqës shkencor:

Prof. Dr. Lulëzim HANELLI

Tiranë, 2017

Falenderime

Si fillim dëshiroj të shpreh mirënjohjen dhe falenderimet më të thella për udhëheqësin tim shkencor Prof. Dr. Lulëzim Hanelli, për udhëheqjen, bashkëpunimin e tij të pa kursyer dhe mbështetjen që më ka ofruar gjatë gjithë kohës së përgatitjes së këtij disertacioni.

Faleminderit Prof. Luli!

Gjithashtu falenderoj familjen time, bashkëshorten Megi, për qetësinë, dashurinë dhe mbështetjen që më kanë ofruar përgjatë këtij rrugëtimi. Ju kërkoj ndjesë që gjatë gjithë kësaj kohe kam qënë mjaft i zënë e nuk kam patur në dispozicion, kohën dhe mundësinë që tju përkushtohesha mjaftueshëm.

Një falenderim i singertë shkon për të gjithë ata kolegë të Departamentit të Inxhinierise Matematike dhe Inxhinierisë Fizike, të cilët nuk kanë hezituuar të më ndihmojnë dhe të më qëndrojnë pranë në çdo moment. Nëse ky studim është realizuar, një pjesë shumë e rëndësishme i takon sugjerimeve dhe shtysave tuaja.

Së fundmi, dua të falenderoj të gjithë mësuesit dhe profesorët nga klasa e parë deri në përfundimin teorik të shkollës së doktoraturës, të cilët, kush më shumë e kush më pak, kanë lënë gjurmë të thella në formimin tim human dhe profesional.

Ju faleminderit!

Abstrakt

Manipulimi me një numër shumë të madh të dhënash është një nga problemet kryesore në shkencë dhe inxhinieri. Metodat klasike numerike shpesh herë nuk mundësojnë zgjidhjen e këtyre problemeve, me përjashtim të rasteve standarte. Në të kundërt Metodat Meshfree mund të përshtaten më mirë me situata më të ndërlikuara (psh. sipërfaqe me deformime të ndryshme) sesa metodat klasike si: metoda e diferencave të fundme, elementëve të fundëm ose vëllimeve të fundme.

Metodat përafruese për të dhënat jo uniforme studiohen nga shumë kërkues. Ato ekzistojnë në shumë fusha të ndryshme të shkencës dhe njihen me emra të ndryshëm, p.sh metoda e elementëve-free Galerkin, metoda e elementeve të fundme të përgjithshuar, hp-clouds, metoda meshless lokale Petrov-Galerkin, metoda e katrorëve më të vegjël lëvizëse; metoda e funksioneve me bazë radiale; metoda sheshuese e partikelave hidrodinamike.

Në këtë punim jemi përqendruar më shumë në metodën e funksioneve me bazë radiale dhe metodën e katrorëve më të vegjël të lëvizshme. Do të shikohen disa aspekte teorike dhe praktike të ndërtimit të modeleve përafrues (interpolues) të metodave jo uniforme. Qëllimi do të jetë përcaktimi i një baze teorike të mjaftueshme për të siguruar ekzistencën dhe unicitetin e zgjidhjes. Gjithashtu, do të kryhen eksperimente të ndryshme numerike për të studiuar dhe testuar saktësinë dhe qëndrueshmërinë e metodave përafruese dhe interpoluese.

Për të bërë implementimin praktik të metodave të studiuar do të përdoret software-i inxhinierik MATLAB, në mjedisin e të cilit janë shkruar të gjitha kodet dhe janë marrë të gjitha paraqitjet grafike dhe numerike të këtij punimi.

Fjalë kyçe: Metodat Meshfree, Funksionet me Bazë Radiale, Metoda e Katrorëve më të Vegjël e Lëvizshme, Interpolim dhe Përafrim Shumëmasorë.

Tabela Përmbledhse

Lista e Figurave	ix
Lista e Tabelave	xi
1 Hyrje	1
1.1 Motivimi: Interpolimi i të dhënave jo-uniforme në $\mathbb{R}^s$	2
1.1.1 Problemi i interpolimi i të dhënave jo-uniforme	2
1.1.2 Shembull: Interpolimi me Matricat Disatancë	3
1.2 Metodat Meshfree në kohë	9
2 Interpolimi me funksionet me bazë radiale	11
2.1 Funksionet me bazë radiale	11
2.2 Funksionet Pozitivisht të Përcaktuar	13
2.2.1 Funksionet Radialë Pozitivisht të Përcaktuar	17
2.2.2 Funksionet totalisht monotonë	20
2.3 Interpolimi me funksione baze Gaussianët	22
2.4 Interpolimi Stacionar dhe Jo-Stacionar	25
2.4.1 Interpolimi Jo-Stacionar	26
2.4.2 Interpolimi Stacionar	26
2.5 Përcaktimi i vlerës optimale të parametrut të shkallëzimit	30
2.5.1 Përcaktimi i vlerës optimale të parametrut me anë të metodës " <i>Provë-Gabim</i> "	31
2.5.2 Përcaktimi i vlerës optimale të parametrut me anë të metodës " <i>Përrjashtoj një të dhënë</i> "	32
2.5.3 Disa Eksperimente Numerike	33
3 Përafrimi i të dhënave jo uniforme me funksionet me bazë radiale: Zbatime dhe eksperimente numerike	39
3.1 Përafrimi i normalizuar me katrorë më të vegjël	41

3.2	Përafrimi i katrorëve më të vegjël kur qendrat ndryshojnë nga të dhënat . . .	42
3.3	Koncepte teorike për përafrimin me katrorët më të vegjël.	45
3.4	Katrorët më të vegjël me përshtatje	47
3.4.1	Algoritmi Shto Nyje	47
3.4.2	Algoritmi Largo Nyje	50
3.4.3	Disa eksperimente numerike	52
3.5	Përafrimi me metodën e katrorëve me të vegjël të lëvizshme	55
3.5.1	Formulimi Backus-Gilbert	56
3.5.2	Implementimi Standart i Metodës së Katrorëve më të Vegjël të Lëvizshme	58
3.5.3	Metoda Shepard	60
3.5.4	Përafrimi me metodat e rindërtimit polinomial	63
3.5.5	Një rast zbatimi	66
4	Zbatime të funksioneve me bazë radiale në metodat pseudospektrale	69
4.1	Matricat Diferenciale	70
4.2	Metodat Pseudospektrale në EDDP me kushte kufitare	72
4.3	Një Metodë Jo-Simetrike PS me bazë radiale	73
4.4	Një Metodë Simetrike PS me bazë radiale	76
4.5	Zbatime të metodave PS me funksione me bazë radiale	79
4.5.1	Zgjidhja e ekuacionit të transportit 1-D	79
4.5.2	Zgjidhja e ekuacionit Allen Cahn	81
4.5.3	Zgjidhja e ekuacionit Helmholtz 2D	83
4.6	Metodat Galerkin dhe funksionet me bazë radiale	85
4.6.1	Një problemë zbatimi	86
	Referenca	101

Lista e Figurave

1.1	289 pika Halton në katrorin njësi në $\mathbb{R}^s$	4
1.2	Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 5$ pikave nyje me Matricën e Distancave	7
1.3	Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 289$ pikave nyje me Matricën e Distancave. Rasti dy dimensional	8
1.4	Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 729$ pikave nyje me Matricën e Distancave. Rasti dy dimensional	8
2.1	Majtas: Gaussian me parametër $\varepsilon = 1$ dhe Djathtas: Gaussian me parametër $\varepsilon = 4$ të qendëruar në origjinë në $\mathbb{R}^2$	12
2.2	Majtas: Gaussianët Laguerre $s = 1, n = 1$ dhe Djathtas: Gaussianët Laguerre $s = 2, n = 2$ të qendëruar në origjinë në $\mathbb{R}^2$	19
2.3	Majtas: Multiquadrics Invers $\beta = 1/2$ dhe Djathtas: Multiquadrics Invers $\beta = 1$ të qendëruar në origjinë në $\mathbb{R}^2$	20
2.4	Grafiku i funksionit Franke në katrorin $[0, 1] \times [0, 1]$	23
2.5	Majtas: Interpolimi i $N = 289$ pikave halton me anë të gaussianëve me parametrin $\varepsilon = 5$ dhe Djathtas: Paraqitja grafike e gabimit të interpolimit	25
2.6	Interpolimi me Gaussian dhe parametër $\varepsilon = 19.8$ i $N = 4225$ pikave halton.	27
2.7	Gabimet maksimale dhe mesatare kundrejt ε për $N = 81$ pika halton	27
2.8	Gabimet maksimale dhe mesatare kundrejt ε për $N = 289$ pika halton	28
2.9	Gabimet maksimale dhe mesatare kundrejt ε për $N = 1089$ pika halton	28
2.10	Gabimet maksimale dhe mesatare kundrejt ε për $N = 4225$ pika halton	28
2.11	Kurbat e gabimit si funksion i parametrin ε për interpolimin e N -të dhënave 1D. Metoda "Provë-Gabim"	34
2.12	Kurbat e gabimit si funksion i parametrin ε për interpolimin e N -të dhënave 2D. Metoda "Provë-Gabim"	35
2.13	Kurbat e gabimit si funksion i parametrin ε për interpolimin e N -të dhënave 1D. Metoda "Përrjashtoj një të dhënë"	36

2.14	Kurbat e gabimit si funksion i parametrin ε për interpolimin e N -të dhënave 2D. Metoda "Përrjashtoj një të dhënë"	37
3.1	$N = 1089$ pika Halton dhe $M = 289$ qendra	44
3.2	Majtas : Përafrimi i $N = 1089$ pikave halton dhe $M = 289$ qendrash me anë të gaussianëve me parametrin $\varepsilon = 5$ dhe Djathtas : Paraqitja grafike e gabimit të përafrimit	45
3.3	Grumbullimi i pikave rreth qendrave të shpërndara mirë	46
3.4	Majtas : Përafrimi i $N = 289$ pikave halton dhe $M = 155$ qendrash, <i>algoritmi shtoj një</i> dhe Djathtas : $M = 155$ qendra të shtuara në $N = 289$ pika nyje për të arritur tolerancën $tol = 10^{-4}$	53
3.5	Majtas : Përafrimi i $N = 289$ pikave halton dhe $M = 153$ qendrash, <i>algoritmi largo një</i> dhe Djathtas : $M = 153$ qendra të shtuara në $N = 289$ pika nyje për të arritur tolerancën $tol = 0.5$	54
3.6	Majtas : Përafrimi i $N = 4225$ pikave halton dhe $M = 164$ qendrash, <i>algoritmi largo një</i> dhe Djathtas : $M = 164$ qendra të shtuara në $N = 4225$ pika nyje për të arritur tolerancën $tol = 10^{-4}$	55
3.7	Lart Majtas : Përafrimi me metodën Shepard për $N = 1089$ të dhëna me "zhurmë" me parametër shkallëzimi $\varepsilon = 48$, Lart Djathtas , Gabimi i mesatarizuar dhe Poshtë Majtas : Përafrimi me metodën Shepard $N = 1089$ të dhëna me "zhurmë" me parametër shkallëzimi $\varepsilon = 12$, Poshtë Djathtas : Grafiku i gabimeve të mesatarizuar	62
3.8	Lart : Përafrimi në rrjetën 60×60 bazuar në 361201 pika nyje dhe Poshtë : përafrimi në rrjetën 100×100 bazuar në 361201 pika nyje. Metoda e katorëve më të vegjël me lëvizje	68
4.1	Zgjidhja numerike e problemit të transportit 1D me anë të metodave PS me funksione me bazë radiale.	80
4.2	Zgjidhja numerike e problemit Allan Chan me anë të metodave PS me funksione me bazë radiale.	83
4.3	Zgjidhja numerike e problemit Helmholtz me anë të metodave PS me funksione me bazë radiale.	84
4.4	Majtas : Zgjidhja numerike me metodën Galerkin me Gaussianin si funksion me bazë radiale dhe Djathtas : Gabimi maksimal	89

Lista e Tabelave

1.1	Interpolimi me Matricën e Distancave në kubin njësi $[0, 1]^s$, $s = 1, 2, 3$. . .	9
2.1	Gausaianët Laguerre për vlera të ndryshme n dhe s	19
2.2	Interpolimi i funksioni <i>sinc</i> duke përdorur Gaussianin me $\varepsilon = 19.8$	26
2.3	Rezultatet e Interpolimit Stacionar të funksionit <i>Sinc</i> për velra të ndryshme të N —së	29
2.4	Vlerat optimale e parametrut ε për interpolimin e të dhënave 1D	35
2.5	Vlerat optimale e parametrut ε për interpolimin e të dhënave 2D	36
3.1	Rezultate të përafrimit 2D me gaussianët për N pika nyje dhe M qendra . . .	44
3.2	Gabimi i normuar dhe gabimi maksimal i përafrimit të N —pika nyje dhe M —qendrave.	55
3.3	Rezultate të përafrimit 2D me funksione peshuese gaussianë për N pika nyje	63

Kapitulli 1

Hyrje

Metodat Meshfree kanë marrë shumë rëndësi vitet e fundit si në komunitetin e matematikanëve ashtu dhe atë të inxhinierëve. Për pasojë, pjesa më e madhe e punës me metoda të tilla është mbi probleme shumdisiplinore, ku ndërthuret matematika me fusha të tjera inxhinierike.

Manipulimi me një numër shumë të madh të dhënash është një nga problemet kryesore në shkencë dhe inxhinieri. Metodat klasike numerike shpesh herë nuk mundësojnë zgjidhjen e këtyre problemeve, me përjashtim të rasteve standarte. Në të kundërt Metodat Meshfree mund të përshtaten më mirë me situata më të ndërlikuara (psh. sipërfaqe me deformime të ndryshme) sesa metodat klasike si: metoda e diferencave të fundme, elementëve të fundëm ose vëllimeve të fundme.

Një avantazh tjetër shumë i madh është pavarësia e tyre nga shpërndarja e të dhënave. Nuk është i nevojshëm optimimi ose diskretizimi i zonës së shqyrtuar në zona ose element të caktuar, përderisa këto metoda bazohen në një bashkësi pikash të pavarura. Kjo ul ndjeshën edhe koston llogaritse numerike në zgjidhjen e problemeve me zona të ndryshme.

Metodat klasike si metoda e diferencave të fundme, elementëve të fundëm ose vëllimeve të fundme, zgjidhin probleme me dimensione të kufizuara në dy ose tre parametra (dimensionet e hapsirës). Në ndryshim, me anë të Metodave Meshfree mund të zgjidhen probleme me më shumë se tre përmasa.

Zbatime të tyre mund të gjenden:

- Në shumë fusha të inxhinierisë ku kërkohet *modelimi i të dhënave jouniforme- scattered data modeling* (psh. modelimi i sipërfaqeve të energjisë potenciale në kimi; problemet e mapping-ut në gjeodezi, gjeofizikë, meteorologji)

- Në disa fusha të inxhinierisë ku kërkohet *zgjidhja e ekuacioneve diferenciale me derivate të pjesshme* (psh. në dinamikën e gazeve; ekuacionet Boltzmann dhe Fokker-Planck në hapsirën gjashtë-dimensionale; aplikime në nanoteknologji)
- Në imazheri mjeksore (psh. *non-uniform sampling*)
- Në matematikë financiare
- Në rrjeta neurale artificiale
- Në optimizim

Meqënëse pjesa më e madhe e këtyre zbatimeve ka të bëjë me problemin e përafrimit të një funksioni mbi disa të dhëna, më poshtë paraqitet problemi i **Interpolimit të të dhënave jo-uniforme** në $\mathbb{R}^s$.

1.1 Motivimi: Interpolimi i të dhënave jo-uniforme në $\mathbb{R}^s$

Një nga problemet kryesore në teorinë e përafrimit është procesi i modelimit të të dhënave jo-uniforme (*scattered data*). Ndërtimin e një modeli të qendrueshëm dhe të mirë-përcaktuar do ta fillojmë me një shembull ku përdoren të ashtuquajturat "*matricat distancë*" (*distance matrices*). Më pas do përgjithësohet kjo qasje me futjen e konceptit të funksioneve me bazë radiale (*radial basis function*).

1.1.1 Problemi i interpolimit i të dhënave jo-uniforme

Në shumë disiplina shkencore haset problemi: Jepet një bashkësi të dhënash (matje dhe pozicionet ku janë kryer ato) dhe kërkohet të gjejmë një mënyrë për të marrë informacion për "rregullin" nga i cili janë gjeneruar këto të dhëna. Gjithashtu, kërkohet informacion për vlerën në pozicione të tjera të ndryshme nga ato të matjeve. Praktikisht, po kërkohet një funksion P_f , i cili i modelon më mirë këto të dhëna. Fjala "modelim i mirë" mund të shikohet në shumë mënyra, ne fillimisht si kriter matës po kërkohet që ky funksion P_f të marrë vlerën e matjeve të dhëna në pozicionet e tyre. Kjo lloj përqasje quhet *interpolim* dhe nëse të dhënat e marra nuk gjenden në një sipërfaqe uniforme ose të rregullt, kemi të bëjmë me *interpolim të dhënash jo-uniforme*.

Nëse shënojmë me x_j , $j = 1, \dots, N$ pozicionet e matjeve, $y_j = f(x_j)$ vlerat e matura në këto pozicione dhe χ bashkësinë e të dhënave, ku $\chi \in \Omega$ për ndonjë zonë $\Omega \in \mathbb{R}^s$, atëherë mund të formuljmë problemin e interpolimit si më poshtë:

Problemi 1.1 *Interpolimi i të dhënave jo-uniforme.* Jepen të dhënat (x_j, y_j) , $j = 1, \dots, N$, ku $x_j \in \mathbb{R}^s$, $y_j \in \mathbb{R}$, të gjendet një funksion i vazhdueshëm P_f i tillë që $P_f(x_j) = y_j$, $j = 1, \dots, N$.

Përcaktimi i x_j në hapsirë s -dimensionale, bën të mundur që formulimi i problemit 1.1 të jetë i vlefshëm në fusha të ndryshme të shkencës. Nëse $s = 1$ të dhënat mund të jenë matje të kryera gjatë një intervali të caktuar. Nëse $s = 2$ të dhënat mund të jenë matje në një zonë plane, psh. matjet e rreshjeve në zona të caktuara në të gjithë territorin. Rasti $s = 3$ mund të paraqesi rastin e temperaturës në brëndësinë e një trupi tre dimensional. Në shumë zbatime të tjera si në rrjetat neurale artificiale, optimizim, statistikë etj mund të gjenden shembuj me më shumë parametra. Kërkojmë funksionin interpolues P_f të ndërtuar si kombinim linear i disa funksioneve të quajtura *funksione baze* B_k , psh:

$$P_f(x) = \sum_{k=1}^N c_k B_k(x), \quad x \in \mathbb{R}^s. \quad (1.1)$$

Në këtë mënyrë, problemi i interpolimit kalon në zgjidhjen e sistemit të ekuacioneve lineare të mëposhtëm:

$$A\mathbf{c} = \mathbf{y},$$

ku elementët e matricës së interpolimit A jepen në formën $A_{jk} = B_k(x_j)$, $j, k = 1, \dots, N$; $\mathbf{c} = [c_1, \dots, c_N]^T$; dhe $\mathbf{y} = [y_1, \dots, y_N]^T$. Problemi 1.1 do të jetë i "përcaktuar drejt", pra do të ketë zgjidhje të vetme, atëhere dhe vetëm atëhere kur matrica e interpolimit A do të jetë jo-singulare. Për këtë problem do të flasim në kapitullin tjetër. Tani do të vazhdojmë me një shëmbull interpolimi me anë të *matricave distancë*.

1.1.2 Shëmbull: Interpolimi me Matricat Disatancë

Si funksion "të panjohur", nga i cili rrjedhin vlerat për të dhënat tona do të marrim funksionin

$$f_s(x) = 4^s \prod_{d=1}^s x_d(1 - x_d), \quad \mathbf{x} = (x_1, \dots, x_s) \in [0, 1]^s.$$

Ky funksion është zero në kufijtë e kubit njësi në $\mathbb{R}^s$ dhe ka vlerën maksimale një në qendër të kubit. Funksioni MATLAB për funksionin f_s gjendet në Aneksin si Programi [(4.5), Fq.95].

Në këtë shëmbull, ashtu si edhe në raste të tjera, do të përdorim pikat *Halton* (*Halton points*). Këto janë pika të shpërndara njëtrajtësisht në $(0, 1)^s$. Në figurën 1.1 paraqiten 289 pika Halton në katrorin njësi në $\mathbb{R}^2$. Për më shumë detaje mbi pikat Halton shikoni [23])

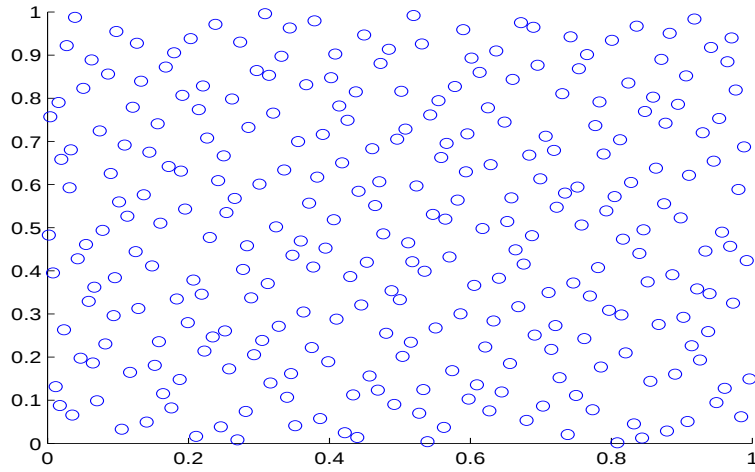


Figura 1.1 289 pika Halton në katrorin njësi në $\mathbb{R}^s$

dhe [43]). Programi MATLAB për gjenerimin e pikave Halton mund të shkarkohet nga "MATLAB Central File Exchange" (MCFE) me emrin *haltonseq.m*.

Pra, qellimi është ndërtimi i një funksioni P_f , i cili do të interpolojë të dhënat e përftuara nga funksioni f_s në pikat Halton, dmth.

$$P_f(x_j) = f_s(x_j),$$

ku x_j është një pikë Halton.

Nëse $s = 1$ kemi të bëjmë me interpolim për të dhëna njëpërmasore. Ky problem zgjidhet me anë të interpolimit polinomial, nëse numri i pikave është relativisht i vogël, ose me saktësi më të madhe me anë të interpolimit me splajne (funksione "pjesë-pjesë polinom"). Një ide e mirë për ndërtimin e funksioneve "pjesë-pjesë polinom" për interpolim është qendërzimi i tyre në një bashkësi pikash në $[0, 1]$, pra interpolanti ynë do të ketë pamjen:

$$P_f(x) = \sum_{k=1}^N c_k |x - x_k|, \quad x \in [0, 1]$$

dhe koeficientët c_k do të gjendeshin nga plotësimi i kushteve të interpolimit

$$P_f(x_j) = f_1(x_j), \quad j = 1 \dots, N.$$

Pikat ku qendërzohen të dhënat, quhen "*qendra*" dhe në përgjithësi zgjidhen të njëjta me pikat nyje x_j . Meqënëse funksionet B_k janë simetrike në lidhje me qendrat e tyre x_k , kjo

përbën një shembull mbi përdorimin e "funksioneve me bazë radiale" (ang. *radial basis functions*). (shiko kapitullin në vijim)

Rikthehemi në problemin e gjetjes së koeficientëve c_k . Gjetja e tyre vjen nga zgjidhja e sistemit të ekuacioneve lineare

$$\begin{bmatrix} |x_1 - x_1| & |x_1 - x_2| & \dots & |x_1 - x_N| \\ |x_2 - x_1| & |x_2 - x_2| & \dots & |x_2 - x_N| \\ \vdots & \vdots & \ddots & \vdots \\ |x_N - x_1| & |x_N - x_2| & \dots & |x_N - x_N| \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (1.2)$$

Matica e sistemit 1.2 është një shembull i *Matricave Distancë*. Këto matarica janë studiuar në gjeometri dhe analizë për një kohë të gjatë (shiko [7]; [10]; [30])

Ideja e mësipërme mund të përgjithësohet edhe për rastet kur pikat nyje kanë dimension $s > 1$. Është treguar se matrica distancë bazuar në normën euklidiane ndërmjet pikave të ndryshme në $\mathbb{R}^s$, është një matricë jo-singulare (shikoni kapitullin 2). Prandaj, nëse rishkruajmë funksionin tonë interpolues për të dhënat s -dimensionale në $(0, 1)^s$:

$$P_f(x) = \sum_{k=1}^N c_k \|x - x_k\|_2, \quad x \in [0, 1] \quad (1.3)$$

ku koeficientët c_k llogariten nga zgjidhja e sistemit

$$\begin{bmatrix} \|x_1 - x_1\|_2 & \|x_1 - x_2\|_2 & \dots & \|x_1 - x_N\|_2 \\ \|x_2 - x_1\|_2 & \|x_2 - x_2\|_2 & \dots & \|x_2 - x_N\|_2 \\ \vdots & \vdots & \ddots & \vdots \\ \|x_N - x_1\|_2 & \|x_N - x_2\|_2 & \dots & \|x_N - x_N\|_2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix}.$$

Funksioni MATLAB për llogaritjen e matricës distancë mbi bazën e normës Euklidiane jepet në Aneksin B si programi [(4.6), Fq.95] me emrin ***DistanceMatrix.m*** marrë nga <http://www.math.unipd.it/demarchi/TAA2010/>. Programi MATLAB 1.1 ***DistanceMatrix-Fit.m*** kryen interpolimin me matricat distancë.

Programi 1.1 *DistanceMatrixFit.m*

```
lineskip
1 % DistanceMatrixFit
2 % Script qe perdor Matricen distance Euklidiane
3 % per interpolimin e te dhenave jo uniforme me dimensione te ndryshme s
4 % Therret funksionet: DistanceMatrix, CreatePoints,
5 % PlotSurf, PlotError2D, PlotSlices, PlotErrorSlices
```

```

6   s = 2;
7   k = 2; N = (2^k+1)^s;
8   neval = 50; M = neval^s;
9   % Perdor Halton points per te dhena dhe qendra
10  dsites = haltonseq(N,s);
11  ctrs = dsites;
12  % Krijom neval^s pika te barazlanguara
13  % ne kubin njesi s-dimensional
14  epoints = MakeSDGrid(s,neval);
15  % gjenerohet krahu i djathte , funksioni i "panjohur",
16  rhs = testfunction(s,dsites);
17  % Llogarisim matricen e distancave
18  IM = DistanceMatrix(dsites , ctrs );
19  % Llogarisim matricen distance ndermjet nyjeve ku do kryhen llogaritjet dhe
20  % qendrave
21  EM = DistanceMatrix(epoints , ctrs );
22  %% Kryejme interpolimin me Matricat Distance
23  % (Matrica distance * zgjidhjen e sistemit te interpolimit)
24  Pf = EM * (IM\rhs);
25  % Llogarisim zgjidhjen e sakte
26  exact = testfunction(s,epoints);
27  % Llogaritim gabimin maksimal dhe rms
28  maxerr = norm(Pf-exact,inf);
29  rms_err = norm(Pf-exact)/sqrt(M);
30  fprintf('RMS error:      %e\n', rms_err)
31  fprintf('Maximum error: %e\n', maxerr)
32  switch s
33      case 1
34          plot(epoints , Pf)
35          set(gca,'FontSize',14)
36          xlabel('x','FontSize',14);
37          ylabel('y','FontSize',14,'Rotation',0);
38          figure; plot(epoints , abs(Pf-exact))
39          set(gca,'FontSize',14)
40          xlabel('x','FontSize',14);
41          ylabel('Error','FontSize',14);
42      case 2
43          fview = [-30,30];
44          xe = reshape(epoints(:,2),neval,neval);
45          ye = reshape(epoints(:,1),neval,neval);
46          PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
47          PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview)
48      case 3
49          xe = reshape(epoints(:,2),neval,neval,neval);
50          ye = reshape(epoints(:,1),neval,neval,neval);
51          ze = reshape(epoints(:,3),neval,neval,neval);
52          xslice = .25:.25:1; yslice = 1; zslice = [0,0.5];
53          PlotSlices(xe,ye,ze,Pf,neval,xslice,yslice,zslice);

```

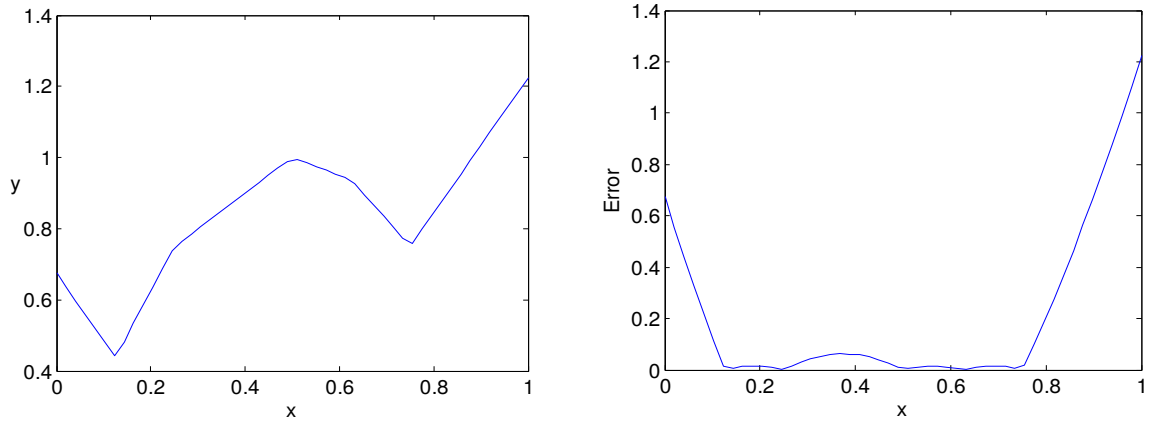


Figura 1.2 Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 5$ pikave nyje me Matricën e Distanave

```

54     PlotErrorSlices(xe, ye, ze, Pf, exact, neval, xslice, yslice, zslice);
55     otherwise
56         disp('Cannot display plots for s > 3')
57     end

```

Në rreshtin 1 dhe 2 përcaktohen dimensionin s dhe numri i pikave nyje N . Do të përdoren pikat Halton (rreshti 10) dhe qëndrat përputhen me të dhënat. Funkcioni **MakeSD-Grid.m**, i cili ndërton pikat e barazlanguara në kubin njësi s -dimensional jepet në Aneksin B si programi ([(4.7), Fq.96]). Aty gjenden dhe programet për paraqitjen grafike të rastit 2D (rreshtat 46-47) dhe rastit 3D (rreshtat 53-54).

Llogaritja e matricës së distanave kryhet në rreshtin 18 për pikat nyje dhe qendrat, kurse në rreshtin 21 llogaritet për pikat ku do të llogaritet interpolanti dhe qendrat. Interpolimi i të dhënave kryhet në rreshtin 24 me anë të komandës $Pf = EM * (IM \backslash rhs)$.

Në rreshtat 28 dhe 29 llogariten përkatësisht gabimi maksimal, duke e llogaritur si normën infinit të vektorit të gabimeve, si dhe gabimi mesatar katror (ang. root-mean-square) rms_{err} i dhënë nga formula 1.4.

$$RMS_{err} = \sqrt{\frac{1}{M} \sum_{j=1}^M [Pf(\xi_j) - f(\xi_j)]^2} = \frac{1}{\sqrt{M}} \|Pf - f\|_2, \quad (1.4)$$

ku me ξ_j , $j = 1, \dots, M$ janë pikat ku është llogaritur interpolanti.

Në figurën 1.2 paraqiten përkatësisht interpolimi i 5 të dhënave për rastin $s = 1$ majtas dhe gabimi absolut djathtas. Nga figura me 5 pika nyje vihet re qartë natyra "pjesë-pjesë" lineare e interpolantit.

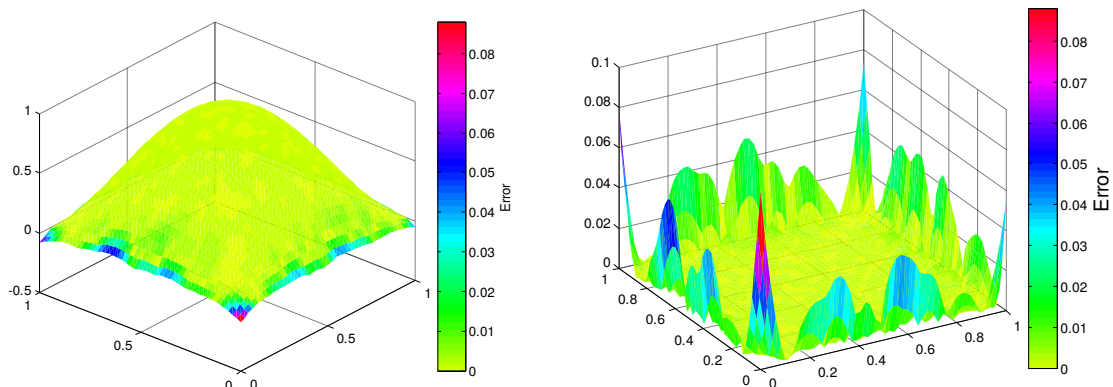


Figura 1.3 Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 289$ pikave nyje me Matricën e Distancave. Rasti dy dimensional

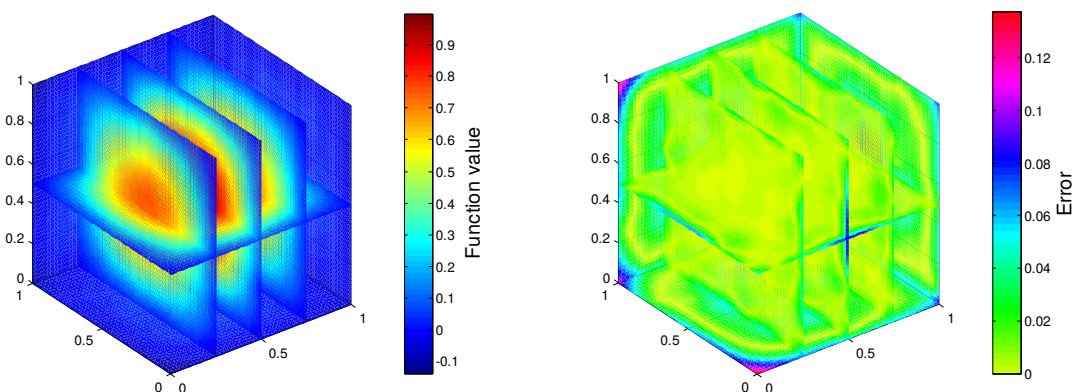


Figura 1.4 Interpolimi (majtas) dhe gabimi absolut (djathtas) i $N = 729$ pikave nyje me Matricën e Distancave. Rasti dy dimensional

Interpolimi i $N = 289$ pikash nyje për rastin dy dimensional sëbashku me gabimin në katrorin njësi $(0, 1)^2$ paraqiten në figurën 1.3.

Kurse në figurën 1.4 paraqitet interpolimi i $N = 729$ të dhënave tre dimensionale sëbashku me gabimin e kryer. Ajo që vihet re nga të gjitha figurat është fakti që gabimi maksimal merret në skajet e zonës. Kjo vjen për shkak se pikat nyje ndodhën më në qendër të zonës. Por, duhet thënë edhe nëse do të kishim të dhëna në kufij, gabimi më i madh do të ishte përsëri aty kur përdoren funksionet me bazë radiale.

Për të pasur një ide më të qartë për interpolimin me matricat distancë, në tabelën 1.1 paraqiten gabimet mesatare katore për disa vlera të ndryshme të N —së për secilin nga rastet një, dy dhe tre dimensionale. Ajo që vihet re është fakti që saktësia rritet me rritjen e pikave nyje dhe për më tepër zvogëlohet me një faktor të përafërt me dy. Bazuar në të dhënat $N = (2^k + 1)^s$ shohim se konvergjenca është e rendit $\mathcal{O}(h)$, ku si h paraprakisht (do t'i

Tabela 1.1 Interpolimi me Matricën e Distancape në kubin njësi $[0, 1]^s$, $s = 1, 2, 3$

k	N	1D		2D		3D	
		Gab-mes	N	Gab-mes	N	Gab-mes	N
1	3	$6.207567e-01$	9	$1.912296e-01$	27	$8.239817e-02$	
2	5	$3.862568e-01$	25	$6.170345e-02$	125	$4.196932e-02$	
3	9	$1.303455e-01$	81	$2.239244e-02$	729	$1.410127e-02$	
4	17	$4.977443e-02$	289	$9.684326e-03$			
5	33	$2.107198e-02$	1089	$3.859039e-03$			
6	65	$9.978796e-03$	4225	$1.561265e-03$			
7	129	$4.964727e-03$					
8	257	$2.476383e-03$					
9	513	$1.236720e-03$					
10	1025	$6.179952e-04$					
11	2049	$3.089068e-04$					
12	4097	$1.544307e-04$					

rikthehemi më pas përkufizimit të saj) quajmë distancën mesatare të të dhënave në zonën në shqyrtim.

Nga ky shembull që ndërtuam mund të nxjerrim këto përfundime, ku disa prej tyre vijnë për shkak të përdorimit të funksioneve me bazë radiale. Së pari, funksionet bazë të përdorura $B_k = \|\cdot - x_k\|_2$, kanë simetri radiale.

Së dyti, duke parë kodin MATLAB vihet re thjeshtësia në implementimin e metodës për vlera të ndryshme dimensionit s . Gjenerimi i zonës së interpolimit është një problem i rëndësishëm kur punojmë me dimensione të larta me metodat me bazë polinomiale si spline-et ose elementët e fundme. Në metodën tonë mjafton të llogariten çiftet e distancave mes të dhënave për të kryer interpolimin.

Së fundmi, kryerja e interpolimit me matricën e distancave, ashtu si me funksionet me bazë radiale më vonë, ka nevojë për zgjidhjen e një sistemi me matricë $N \times N$. Ky përbën një problem kur numri i të dhënave rritet shumë, për shkak të kompleksitetit të veprimeve. Gjithashtu, do të shohim se në disa raste matrica e interpolimit shfaq probleme keqkushtëzimi.

1.2 Metodat Meshfree në kohë

Fillimisht përdorimi i Metoda Meshfree erdhi si zbatim në gjeodezi, gjeofizikë ose meteorologji. Më pas, aplikimet u zgjeruan edhe në fushën e ekuacioneve diferenciale me derivate të pjesshme, inteligjencë artificiale, procesim imazhi dhe sinjali etj.

Në literaturën e përafrimit numerik dhe aplikimeve të tij në shumë fusha inxhinierike kontribut të veçantë kanë dhënë :

- Donald Shepard, i cili gjatë kohës së studimeve në Universitetin e Harvardit dha idenë e përdorimit të atyre që sot quhen "*Funksionet Shepard*" në fundin e viteve 60. Në publikimin e tij (shiko [34], Shepard, 1968) Shepard, diskuton metodën inverse të distancave të peshuar dhe disa modifikime të saj. Kjo metodë në atë kohë u bë pjesë e një programi kompjuterik, SYMAP, për gjenerimin e hartave.
- Rolland Hardy, i cili ishte një gjeodet në universitetin "Iowa State", paraqiti funksionet "*multiquadrics*" (MQs) në fillimin e viteve 1970 (shiko [24], Hardy, 1971). Punimi i tij ishte i bazuar kryesisht në aplikim në gjeodezi dhe harta.
- Peter Lancaster dhe Kes Salkauskas, matematicien në universitetin "Calgary", Kanada. Punimi i tyre (shiko [28], Lancaster and Salkauskas, 1981) prezanton "*metodën e lëvizshme të katrorëve më të vegjël*" (ang. *moving least squares method*) si përgjithësim i metodës së Shepard.
- Richard Franke, një matematikan në Naval Postgraduate School në Monterey, Kaliforni, në punimin (shiko [20], Franke, 1982a) ka kryer një krahasim metodash të ndryshme për interpolimin e të dhënave meshfree (të dhëna jo-uniforme). Franke, gjithashtu, hodhi supozimin se matrica interpoluese për funksionet MQs është e kthyeshme.
- Charles Micchelli, një matematikan në qendrën kërkimore IBM Watson Research Center, publikoi artikullin ([30], Micchelli, 1986). Gjithashtu, ai provoi supozimin e Franke. Vërtetimet e tij bazohen në punimet e ([8], Bochner, 1932; [9], Bochner, 1933) mbi funksionet pozitivisht të përcaktuar dhe totalisht monotone (ang. Completely monotone).

Kapitulli 2

Interpolimi me funksionet me bazë radiale

Në këtë kapitull do të prezantohet interpolimi i të dhënave jo-uniforme duke u bazuar në funksionet me bazë radiale. Fillimisht do jepet një prezantim i parë i tyre. Më pas, do të vazhdohet me problemin e interpolimit me këto funksione, duke paraqitur edhe një koncentrat të përfundimeve teorike ku ky problem mbështetet për të siguraur ekzistencën dhe unicitetin e zgjidhjes së tij. Në fund do të prezantohen disa eksperimente numerike për të pasur një ide më të qartë për saktësinë, qendrueshmërinë dhe aplikimin e metodave interpoluese meshfree.

2.1 Funksionet me bazë radiale

Si shembull të parë marrim një funksion të mirënjohur në shumë fusha të matematikës si funksioni *Gaussian*

$$\varphi(r) = e^{-(\varepsilon r)^2}, \quad r \in \mathbb{R}.$$

Ku, parametri i shkallëzimit ε është i lidhur me variancën e funksionit të shpërndarjes normale σ^2 me anë të barazimit $\varepsilon = \frac{1}{(2\sigma^2)}$. Nëse kompozomë funksioni gaussian me normën Euklidiane $\|\cdot\|_2$ për ndonjë *qendër* të fiksuar $x_k \in \mathbb{R}^s$, marrim një funksion *shumëpërmasor*

$$\phi(x) = e^{-\varepsilon^2 \|x - x_k\|_2^2}, \quad x \in \mathbb{R}^s.$$

Pra, shohim se lidhja midis ϕ_k dhe φ jepet nga

$$\phi_k(x) = \varphi(\|x - x_k\|_2).$$

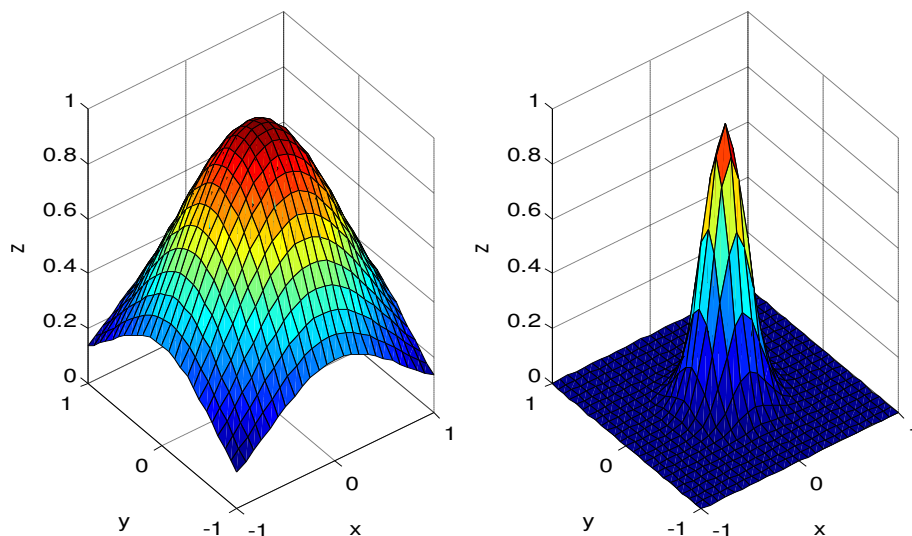


Figura 2.1 **Majtas**: Gaussian me parametër $\varepsilon = 1$ dhe **Djathtas**: Gaussian me parametër $\varepsilon = 4$ të qendëruar në origjinë në $\mathbb{R}^2$

Kjo është lidhja që i jep emrin *funksion me bazë radiale*. Më poshtë jepet një përkufizim formal i një funksioni me bazë radiale.

Përkufizim 2.1 Një funksion $\phi : \mathbb{R}^s \rightarrow \mathbb{R}$ quhet me bazë radiale, nëse ekziston një funksion njëpërmasor $\varphi : [0, \infty) \rightarrow \mathbb{R}$, i tillë që

$$\phi(x) = \varphi(r), \quad \text{ku } r = \|x\|$$

dhe $\|\cdot\|$ është një funksion normë në $\mathbb{R}^s$ – zakonisht norma Euklidiane.

Siaps përkufizimit 2.1, për një funksion radial mund të shkruajmë:

$$\|x_1\| = \|x_2\| \implies \phi(x_1) = \phi(x_2), \quad x_1, x_2 \in \mathbb{R}^s.$$

Me fjalë të tjera, vlera e funksionit ϕ një një distancë të fiksuar nga origjina, ose qendër tjetër të fiksuar, është e njëjtë. Tani mund të themi se, rasti i funksionit normë Euklidiane të përdorur në kapitullin e parë, është një rast i veçantë i funksioneve me bazë radiale i fituar nga barazimi $\varphi(r) = r$. Në figurën 2.1 paraqitet grafiku i funksionit gaussian me parametër $\varepsilon = 1$ (majtas) dhe $\varepsilon = 4$ (djathtas), të dy të qendëruar në origjinën e $\mathbb{R}^2$.

Më vonë, do të shohim se vlera e parametrut ε luan një rol shumë të rëndësishëm në saktësinë dhe qendrueshmërinë e interpolimit me funksionet gaussian. Momentalisht, themi që për të vlera të vogla të tij, gaussiani bëhet më i lakuar dhe me rritjen e vlerës të parametrut ε ai bëhet më i "thepisur".

Aplikimi i funksioneve me bazë radiale në interpolimin e të dhënave jo-uniforme, ka avantazhin e të mos qënit i varur nga dimensionin s i hapsirës ku janë marrë të dhënat. Në vend që të kemi të bëjmë me një funksion shumpërmasorë ϕ , i cili bëhet më kompleks kur rritet dimensionin i të dhënave, ne merremi me funksionin njëpërmasor φ pavarësisht dimensioneve s të hapsirës së shqyrtuar. I rikthehemi edhe njëherë problemit [(1.1), Fq.3] të interpolimit të përcaktuar në kapitullin e parë. Trajta e tij duke zëvendësuar funksionin normë Euklidiane me një funksion të përgjithshëm me bazë radiale merr formën:

$$P_f(x) = \sum_{k=1}^N c_k \varphi(\|x - x_k\|_2), \quad x \in \mathbb{R}^s \quad (2.1)$$

ku koeficientët c_k llogariten nga zgjidhja e sistemit

$$\begin{bmatrix} \varphi(\|x_1 - x_1\|_2) & \varphi(\|x_1 - x_2\|_2) & \dots & \varphi(\|x_1 - x_N\|_2) \\ \varphi(\|x_2 - x_1\|_2) & \varphi(\|x_2 - x_2\|_2) & \dots & \varphi(\|x_2 - x_N\|_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|x_N - x_1\|_2) & \varphi(\|x_N - x_2\|_2) & \dots & \varphi(\|x_N - x_N\|_2) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_N) \end{bmatrix} \quad (2.2)$$

Kuptohet se, që problemi interpolimit të ketë zgjidhje të vetme duhet që matrica e sistemit 2.2 të ketë të anasjelltë. Për këtë problem do të jepen disa rezultate teorike në vazhdimin e këtij kapitulli.

2.2 Funksionet Pozitivisht të Përcaktuar

Në këtë pjesë do të prezantojmë disa rezultate teorike për problemin e zgjidhjes së vetme të sistemit 2.2. Fillojmë me përkufizimin e matricës pozitivisht të përcaktuar.

Përkufizim 2.2 *Në matricë reale simetrike A quhet pozitivisht gjysëm e përcaktuar nëse forma koadratike e saj është jo negative:*

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k A_{jk} \geq 0 \quad (2.3)$$

për $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^s$.

Nëse forma kuadratike 2.3 është zero vetëm për $\mathbf{c} \equiv \mathbf{0}$, atëherë A quhet *pozitivisht e përcaktuar*.

Një karakteristikë e matricave pozitivisht të përcaktuara është që vlerat vetjake të tyre janë të gjitha pozitive, për rrjedhojë ato kanë të anasjelltë. Prandaj, na intereson vendosja

e disa kritereve teorike në mënyrë që funksionet bazë të përdorura në ekuacionin (2.1) të gjenerojnë matrica të tilla. Për këtë arsye përkufizojmë *funksionet pozitivisht të përcaktuara*

Përkufizim 2.3 *Një funksion i vazhdueshëm me vlera komplekse $\phi : \mathbb{R}^s \rightarrow \mathbb{C}$ quhet pozitivisht i përcaktuar në $\mathbb{R}^s$ nëse,*

$$\sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \phi(x_j - x_k) \geq 0 \quad (2.4)$$

për çdo N pika të ndryshme $x_1, \dots, x_N \in \mathbb{R}^s$ dhe $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{C}^N$.

Funksioni ϕ quhet *rigorozisht pozitivisht i përcaktuar në $\mathbb{R}^s$* , nëse forma e tij kuadratike (2.4) bëhet zero vetëm për $\mathbf{c} \equiv 0$.

Në përkufizimin 2.2 shohim se kemi të bëjmë edhe me koeficientë kompleks, edhe pse në problemet e botës reale kemi të bëjmë me funksione me vlera reale, por ky përkufizim na ndihmon në paraqitjen e disa vetive të funksioneve (rigorozisht) pozitivisht të përcaktuara të paraqitura në Teoremën 2.1. Një përshtatje e përkufizimit 2.2, për funksionet me vlera reale jepet nga Teorema 2.2.

Shembull 2.1 *Nëse shënojmë me $x \cdot y$ prodhimin skalar të x dhe y në $\mathbb{R}^s$, atëherë funksioni $\Phi(x) = e^{ix \cdot y}$, për ndonjë $y \in \mathbb{R}^s$ të fiksuar, është pozitivisht i përcaktuar në $\mathbb{R}^s$ përderisa forma kuadratike në përkufizimin 2.3 bëhet:*

$$\begin{aligned} \sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} \Phi(x_j - x_k) &= \sum_{j=1}^N \sum_{k=1}^N c_j \overline{c_k} e^{i(x_j - x_k) \cdot y} \\ &= \sum_{j=1}^N c_j e^{ix_j \cdot y} \sum_{k=1}^N \overline{c_k} e^{-ix_k \cdot y} \\ &= \left| \sum_{j=1}^N c_j e^{ix_j \cdot y} \right|^2 \geq 0. \end{aligned}$$

Nga përkufizimi 2.3 dhe diskutimi i kryer deri tani, shohim se si funksione bazë në 1.1, shërbejnë funksione rigoroz pozitivisht të përcaktuara, pra $B_k(x) = \Phi(x - x_k)$ ose

$$P_f(x) = \sum_{k=1}^N c_k \Phi(x - x_k), \quad x \in \mathbb{R}^s. \quad (2.5)$$

Teoremë 2.1 *Disa veti të funksioneve pozitivisht të përcaktuara janë:*

- 1) *Kombinimi linear i fundëm jo-negativ i funksioneve pozitivisht të përcaktuara, është pozitivisht i përcaktuar. Nëse $\phi_1, \dots, \phi_n$ janë pozitivisht të përcaktuara në $\mathbb{R}^s$, $c_j \geq 0$*

dhe $j = 1, \dots, n$, atëhere

$$\phi(x) = \sum_{j=1}^n c_j \phi_j(x), \quad x \in \mathbb{R}^s,$$

është gjithashtu pozitivisht i përcaktuar. Për më tepër, nëse të paktën njëri nga ϕ_j është rigorozisht pozitivisht i përcaktuar dhe koeficienti përkatës $c_j > 0$, atëhere edhe ϕ është rigorozisht pozitivisht i përcaktuar.

2) $\phi(0) \geq 0$.

3) $\phi(-x) = \overline{\phi(x)}$.

4) Çdo funksion pozitivisht i përcaktuar është i kufizuar. Për më tepër $|\phi(x)| \leq \phi(0)$.

5) Nëse ϕ është pozitivisht i përcaktuar dhe $\phi(0) = 0$, atëhere $\phi \equiv 0$.

6) Prodhimi i funksioneve (rigorozisht) pozitivisht të përcaktuar, është funksion (rigorozisht) pozitivisht i përcaktuar.

Vërtetim: Vetitë (1) dhe (2) rrjedhin direkt nga përkufizimi 2.3.

Për të vërtetuar (3) marrim $N = 2$, $x_1 = 0$, $x_2 = x$ dhe zgjedhim $c_1 = 1$, $c_2 = c$. Atëhere forma kuadratike nga përkufizimi 2.3 merr pamjen:

$$\sum_{j=1}^2 \sum_{k=1}^2 c_j \overline{c_k} \phi(x_j - x_k) = (1 + |c|^2) \phi(0) + c \phi(x) + \overline{c} \phi(-x) \geq 0$$

për çdo $c \in \mathbb{C}$. Duke marrë $c = 1$ dhe $c = i$, shohim se përkatësisht $\phi(x) + \phi(-x)$ dhe $i(\phi(x) + \phi(-x))$ duhet të jenë numra realë. Kjo është e mundur vetëm nëse $\phi(-x) = \overline{\phi(x)}$.

Për të vërtetuar vetinë (4) marrim $N = 2$, $x_1 = 0$, $x_2 = x$ dhe zgjedhim $c_1 = |\phi(x)|$, $c_2 = -\overline{\phi(x)}$. Atëhere forma kuadratike nga përkufizimi 2.3 merr pamjen:

$$\sum_{j=1}^2 \sum_{k=1}^2 c_j \overline{c_k} \phi(x_j - x_k) = 2\phi(0)|\phi(x)|^2 - \phi(-x)\phi(x)|\phi(-x)| - \phi^2(x)|\phi(x)| \geq 0.$$

Përderisa $\phi(-x) = \overline{\phi(x)}$, nga vetia (3) kemi:

$$2\phi(0)|\phi(x)|^2 - 2|\phi(x)|^3 \geq 0.$$

Nëse $|\phi(x)| > 0$, pjesetojmë me $|\phi(x)|^2$ dhe barazimi rrjedh menjëherë. Nëse $|\phi(x)| = 0$, atëhere barazimi qendron në formë triviale.

Vetia (5) rrjedh direkt nga (4). Vetia (6) është një rrjedhim i teoremës së *Schur* në fushën e algjebërës lineare, sipas të cilës thuhet se prodhimi i dy matricave pozitive (gjysëm-) të përcaktuara është matricë pozitivisht (gjysëm-) e përcaktuar. Për detaje të mëtejshme shikoni ([12]; [41]).

Shembull 2.2 Funksioni $f(x) = \cos(x)$ është pozitivisht i përcaktuar në $\mathbb{R}$, përderisa për $x \in \mathbb{R}$ kemi $\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix})$. Tani mjafton të zbatojmë vetinë (1) dhe shembulli 2.1.

Teoremë 2.2 Një funksion i vazhdueshëm me vlera reale ϕ është pozitivisht i përcaktuar në $\mathbb{R}^s$, atëherë dhe vetëm atëherë kur është çift dhe

$$\sum_{j=1}^N \sum_{k=1}^N c_j c_k \phi(x_j - x_k) \geq 0 \quad (2.6)$$

për N pika të ndryshme $x_1, \dots, x_N \in \mathbb{R}^s$ dhe $\mathbf{c} = [c_1, \dots, c_N]^T \in \mathbb{R}^N$.

Funksioni ϕ është rigoroz pozitivisht i përcaktuar në $\mathbb{R}^s$ nëse forma kuadratike (2.6) është zero vetëm për $\mathbf{c} \equiv \mathbf{0}$.

Për funksionet pozitivisht të përcaktuar mund të bëhet edhe një karakterizim integral. Një nga rezultatet më të njohura mbi karakterizimin e funksioneve pozitivisht të përcaktuar në terma të transformimit Fourier është vendosur nga Bochner në vitin 1932 ($s = 1$) dhe në 1933 (për çdo s).

Teoremë 2.3 (Bochner) Një funksion (me vlera komplekse) $\Phi \in C(\mathbb{R}^s)$ është pozitivisht i përcaktuar në $\mathbb{R}^s$, atëherë dhe vetëm atëherë kur është një transformim Fourier i një mase μ jo-negative Borel-i në $\mathbb{R}^s$, psh.

$$\Phi(x) = \hat{\mu}(x) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} e^{-ix \cdot y} d\mu(y), \quad x \in \mathbb{R}^s.$$

Vërtetim: Ka disa vërtetime për këtë teoremë. Vërtetimi origjinal i Bochner mund të gjendet në [9], Bochner, (1933)].

Në mënyrë që të arrijmë qellimin tonë për një problem interpolimi të "mirë-përcaktuar", duhet të shtrimë rezultatet e teoremës Bochner në funksionet rigoroz pozitivisht të përcaktuar. Fillojmë me një kusht të mjaftueshëm që një funksion të jetë rigoroz pozitivisht i përcaktuar në $\mathbb{R}^s$.

Paraqesim fillimish nocionin e një mbajtësi të një mase (jo-negative) Boreli në ndonjë hapsirë topologjike X . Kjo bashkësi jepet si:

$$X \setminus \cup \{O : \text{e hapur dhe } \mu(O) = 0\}.$$

Teoremë 2.4 *Le të jetë μ një masë jo-negative Boreli në $\mathbb{R}^s$, mbajtëse e së cilës është një bashkësi masash jozero të Lebegut. Atëhere transformimi Fourier i μ është rigoroz pozitivisht i përcaktuar.*

Së fundmi, një kriter për të kontrolluar nëse një funksion është rigorozisht pozitivisht i përcaktuar jepet nga Wendland në [41].

Teoremë 2.5 *Le të jetë Φ një funksion i vazhdueshëm në $L_1(\mathbb{R}^s)$. Φ është rigorozisht pozitivisht i përcaktuar atëhere dhe vetëm atëhere kur Φ është i kufizuar, transformimi Fourier i tij nuk është negativ dhe jo identikisht zero.*

2.2.1 Funkzionet Radialë Pozitivisht të Përcaktuar

Në përkufizimin (2.3) u karakterizuan funksionet pozitivisht të përcaktuar në funksionet me shumë variabla Φ . Por, nëse merremi me funksione me bazë radiale, psh. $\Phi(x) = \varphi(\|x\|)$, do të ishte me leverdi t'i referohemi funksionit me një variabël φ si *një funksion radial pozitivisht të përcaktuar*.

Që këtë rrjedh menjëherë:

Lema 2.1 *Në qoftë se $\Phi = \varphi(\|\cdot\|)$ është (rigoroz) pozitivisht i përcaktuar dhe radial në $\mathbb{R}^s$, atëhere Φ është gjithashtu (rigoroz) pozitivisht i përcaktuar dhe radial në $\mathbb{R}^\sigma$ për $\forall \sigma \leq s$.*

Ashtu si në pjesën e mëparshme, mund të bëhet një karakterizim integral për funksionet radiale rigoroz pozitivisht të përcaktuar. Një karakterizim i tillë mund të gjendet në [41], Wendland, (2005a)]

Teoremë 2.6 *Një funksion i vazhdueshëm $\varphi : [0, +\infty) \rightarrow \mathbb{R}$, i tillë që $r \mapsto r^{s-1}\varphi(r) \in L_1[0, +\infty)$ është rigorozisht pozitivisht i përcaktuar dhe radial në $\mathbb{R}^s$, atëhere dhe vetëm atëhere kur Transformimi Fourier s -dimensional*

$$\mathcal{F}_s \varphi(r) = \frac{1}{\sqrt{r^{s-2}}} \int_0^\infty \varphi(t) t^{\frac{s}{2}} J_{(s-2)/2}(rt) dt$$

është jo-negativ dhe jo identikisht zero.

Duke u rikthyer tani në Lemën 2.1, themi se funksionet që janë (rigorozisht) pozitivisht të përcaktuar në $\mathbb{R}^s$, janë gjithashtu (rigorozisht) pozitivisht të përcaktuar në $\mathbb{R}^\sigma$, $\forall \sigma < s$. Me interes të veçantë janë funksionet rigorozisht pozitivisht të përcaktuar në $\mathbb{R}^s$, $\forall s$.

Më poshtë jepen disa shembuj funksionesh rigoroz pozitivisht të përcaktuar:

Shembull 2.3 *Gaussianët.*

Mund të tregohet se Gaussiani

$$\Phi(X) = e^{-\varepsilon^2 \|x\|^2}, \quad \varepsilon > 0 \quad (2.7)$$

është rigoroz pozitivisht i përcaktuar në $\mathbb{R}^s$ për të gjitha s -të. Kjo vjen për shkak të faktit se transformimi Fourier i një gaussiani është në thelb Gaussian,

$$\widehat{\Phi}(\omega) = \frac{1}{(\sqrt{2}\varepsilon)^2} e^{-\frac{\|\omega\|^2}{4\varepsilon^2}}$$

dhe është pozitiv pavarësisht dimensionit s të hapsirës. Në veçanti, nëse marrim $\varepsilon = \frac{1}{\sqrt{2}}$, do të kemi $\widehat{\Phi} = \Phi$. Paraqitje grafike e Gaussianit u dha në figurën 2.1.

Shembull 2.4 *Gaussianët Laguerre*

Fillojmë me *polinomet e përgjithësuara Laguerre* $L_n^{s/2}$ të fuqisë n dhe rendit $s/2$ të përcaktuar nga formula e tyre (shiko formulën (6.2.1) në [1], Andrews and Roy, 1999)

$$L_n^{s/2}(t) = \frac{e^t t^{-s/2}}{n!} \frac{d^n}{dt^n} \left(e^{-t} t^{n+s/2} \right), \quad n = 1, 2, 3, \dots$$

Një formulë eksplicite për polinomet e përgjithsuara Laguerre është

$$L_n^{s/2}(t) = \sum_{k=0}^n \frac{(-1)^k}{k!} \binom{n+s/2}{n-k} t^k.$$

Në këtë mënyrë përcaktojmë Gaussianët Laguerre:

$$\Phi(x) = e^{-\|x\|^2} L_n^{s/2}(\|x\|^2), \quad (2.8)$$

dhe transformimin Fourier

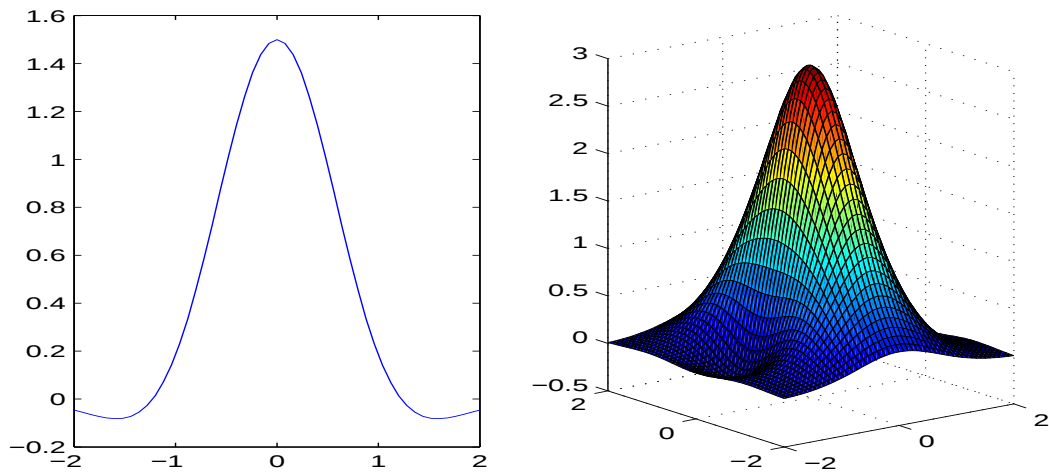
$$\widehat{\Phi}(\omega) = \frac{e^{-\frac{\|\omega\|^2}{4}}}{\sqrt{2^s}} \sum_{j=0}^n \frac{\|\omega\|^{2j}}{j! 4^j} \geq 0 \quad (2.9)$$

Vëmë re se, përkufizimi i Gaussianëve Laguerre varet nga dimensionit s i hapsirës. Prandaj, ata janë rigoroz pozitivisht të përcaktuar në $\mathbb{R}^s$ dhe në bazë të lems 2.1 do të jenë edhe në $\mathbb{R}^\sigma$, për çdo $\sigma < s$. Funksionet Gaussianë-Laguerre për disa vlera të s dhe n jepen në tabelën 2.1. Në figurën 2.2 paraqiten grafikisht dy raste: majtas $s = 1, n = 1$ dhe djathtas $s = 2, n = 2$.

Shembull 2.5 *Funksionet e përgjithshme Inverse Multiquadrics*

Tabela 2.1 Gausaianët Laguerre për vlera të ndryshme n dhe s

s	$n = 1$	$n = 2$
1	$\left(\frac{3}{2} - x ^2\right)e^{- x ^2}$	$\left(\frac{15}{8} - \frac{5}{2} x ^2 + \frac{1}{2} x ^4\right)e^{- x ^2}$
2	$\left(2 - \ x\ ^2\right)e^{-\ x\ ^2}$	$\left(3 - 3\ x\ ^2 + \frac{1}{2}\ x\ ^4\right)e^{-\ x\ ^2}$
3	$\left(\frac{5}{2} - \ x\ ^2\right)e^{-\ x\ ^2}$	$\left(\frac{35}{8} - \frac{7}{2}\ x\ ^2 + \frac{1}{2}\ x\ ^4\right)e^{-\ x\ ^2}$

Figura 2.2 **Majtas**: Gaussianët Laguerre $s = 1, n = 1$ dhe **Djathtas**: Gaussianët Laguerre $s = 2, n = 2$ të qendëruar në origjinë në $\mathbb{R}^2$

Tregohet se funksionet e përgjithshme Inverse Multiquadrics

$$\Phi(x) = (1 + \|x\|^2)^{-\beta}, \quad \beta > \frac{s}{2}, \quad (2.10)$$

janë rigorozisht pozitivisht të përcaktuar në $\mathbb{R}^s$ për të gjitha $s < 2\beta$. Gjithashtu janë dhe pafundësisht të derivueshme. Funksioni "origjinal" invers multiquadrics u prezantua nga Hardy në fillimin e viteve 1970 dhe i korrespondon vlerës $\beta = 1/2$. Zgjedhja e veçantë $\beta = 1$ u quajt si *kuadratike inverse* në punime të ndryshme të Fornberger dhe bashkpuntorëve të tjerë. (shiko [18]) Me anë të kodit të mëposhtëm MATLAB ndërtohen grafikët e figurës 2.3.

lineskip

```
1 % Script qe nderton grafikisht
2 % funksionin multiquadrics invers
3 % Phi(x)=(1+||x||^2)^{-B}
```

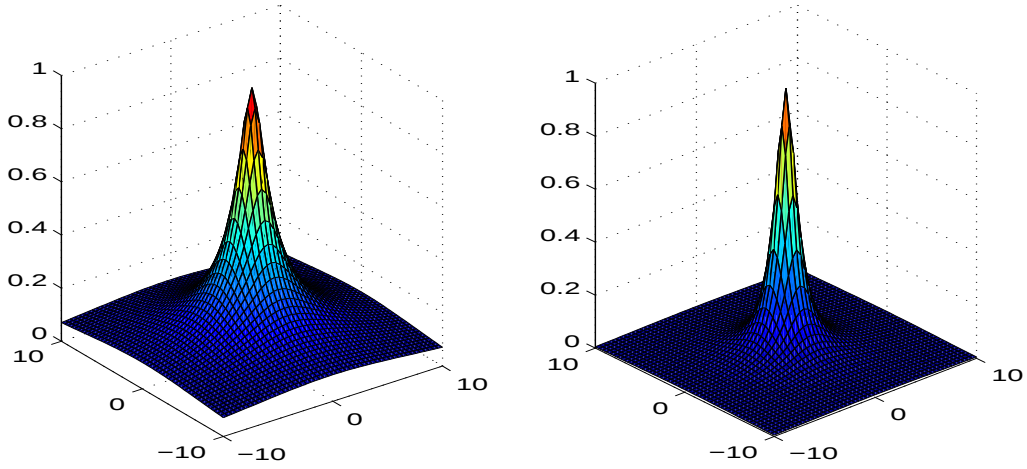


Figura 2.3 **Majtas**: Multiquadrics Invers $\beta = 1/2$ dhe **Djathtas**: Multiquadrics Invers $\beta = 1$ të qendëruar në origjinë në $\mathbb{R}^2$

```

4 x=linspace(-10,10,51);
5 y=x;
6 [xx,yy]=meshgrid(x,y);
7 z1=(1+xx.^2+yy.^2).^(-1/2); % Beta=1/2
8 z2=(1+xx.^2+yy.^2).^(-1); % Beta=1
9 % Paraqitja grafike
10 subplot(1,2,1)
11 surf(x,y,z1)
12
13 subplot(1,2,2)
14 surf(x,y,z2)

```

2.2.2 Funksionet totalisht monotone

Transformimet Fourier jo gjithmonë gjenden me lehtësi. Prandaj, në këtë paragraf do të jepet një mënyrë tjetër për të kontrolluar nëse një funksion është pozitivisht i përcaktuar rigoroz dhe radial në $\mathbb{R}^s$.

Fillojmë me një klasë funksionesh që është shumë afër me funksionet pozitivisht të përcaktuar rigoroz dhe radial në $\mathbb{R}^s$.

Përkufizim 2.4 Një funksion $\varphi : [0, \infty) \rightarrow \mathbb{R}$ që është në $C[0, \infty) \cap C^\infty(0, \infty)$ dhe plotëson kushtin

$$(-1)^l \varphi^{(l)}(r) \geq 0, \quad r > 0, \quad l = 1, 2, 3, \dots,$$

quhet funksion totalisht monoton në $[0, \infty)$.

Shembull 2.6 Funksioni $\varphi(r) = \varepsilon$, $\varepsilon > 0$, është totalisht monoton në $[0, \infty)$.

Shembull 2.7 Funksioni $\varphi(r) = e^{-\varepsilon r}$, $\varepsilon \geq 0$, është totalisht monoton në $[0, \infty)$ përderisa

$$(-1)^l \varphi^{(l)}(r) = \varepsilon^l e^{-\varepsilon r} \geq 0, \quad l = 1, 2, 3, \dots$$

Shembull 2.8 Funksioni $\varphi(r) = \frac{1}{(1+r)^\beta}$, $\beta \geq 0$, është totalisht monoton në $[0, \infty)$ përderisa

$$(-1)^l \varphi^{(l)}(r) = (-1)^{2l} \beta(\beta+1) \cdots (\beta+l-1) (1+r)^{-\beta-l} \geq 0, \quad l = 1, 2, 3, \dots$$

Disa veti të funksioneve totalisht monoton, të cilat mund të gjenden në ([12]; [17]; [42]) janë:

- 1) Kombinimi linear i fundëm jonegativ i funksioneve totalisht monoton është funksion totalisht monoton.
- 2) Prodhimi i dy funksioneve totalisht monoton është funksion totalisht monoton.
- 3) Nëse funksioni φ është totalisht monoton dhe funksioni ψ është monoton absolut (p.sh, $\psi^{(l)} \geq 0$ për të gjitha l -të), atëherë $\psi \circ \varphi$ është totalisht monoton.
- 4) Nëse φ është totalisht monoton dhe ψ është një funksin pozitiv, i tillë që derivati i tij të jetë totalisht monoton, atëherë $\psi \circ \varphi$ është totalisht monoton.

Vëmë re se dy shembujt e dytë të mësipërm, me ndryshim të një variabli $r \rightarrow r^2$, janë të ngjashëm me Gaussianin dhe Multiquadrics Invers. Në mënyrë që të shohim lidhjen e funksioneve totalisht monoton me funksionet rigorozë pozitivisht të përcaktuar dhe radial, duhet të bëjmë një karakterizim integral të funksioneve totalisht monoton.

Teoremë 2.7 (Hausdorff-Bernstein-Widder). Një funksion $\varphi [0, \infty) \rightarrow \mathbb{R}^s$ është totalisht monoton në $[0, \infty)$ atëherë dhe vetëm atëherë kur është transformimi Laplasit i një mase të fundme jonegative μ të Borelit në $[0, \infty)$, pra φ është i formës

$$\varphi(r) = \mathcal{L}_\mu(r) = \int_0^\infty e^{-rt} d\mu(t).$$

Vërtetim: Vërtetimi i Widder-it për teoremën mund të gjendet në [42], Widder, (1941). Një vërtetim tjetër i detajuar mund të gjendet në ([12] Cheney and Light, 1999) dhe ([41] Wendland, (2005)).

Më poshtë jepet lidhja midis funksioneve pozitivisht të përcaktuar dhe radial në $\mathbb{R}^s$ me funksionet totalisht monoton. Ky rezultat u përcaktua nga Schoenberg në 1938.

Teoremë 2.8 *Një funksion φ është totalisht monoton në $[0, \infty)$ atëhere dhe vetëm atëhere kur $\Phi = \varphi(\|\cdot\|^2)$ është pozitivisht i përcaktuar dhe radial në $\mathbb{R}^s$ për çdo s .*

Vëmë re se funksioni Φ tani përcaktohet me *katrorin* e normës. Kjo ndryshon nga përkufizimi i dhënë për funksionet me bazë radiale. (shiko përkufizimin [(2.1), Fq.12]).

Mund të themi se nëse masa μ nuk është e përqendruar në origjinë, atëhere Φ është rigorozisht pozitivisht i përcaktuar dhe radial në $\mathbb{R}^s$ për çdo s . Ky kusht për masën μ është ekuivalent me faktin që φ të mos jetë konstant. Me këtë kusht shtesë për φ mund të përdorim funksionet totalisht monoton në interpolimin e të dhënave jouniforme. Teorema e mëposhtme, e prezantuar në punën e Schoenberg (shiko [33], Schoenberg, (1938a)), i cili tregoi se një funksion totalisht monoton do të thotë funksion rigoroz pozitivisht i përcaktuar, përcakton një test shumë të thjeshtë për të verifikuar "mirpërcaktimin" (zgjidhje të vetme) e një problemi interpolimi të dhënash jo-uniforme.

Teoremë 2.9 *Një funksion $\varphi : [0, \infty) \rightarrow \mathbb{R}$ është totalisht monoton, por jo konstant atëhere dhe vetëm atëhere kur $\varphi(\|\cdot\|^2)$ është rigorozisht pozitivisht i përcaktuar dhe radial në $\mathbb{R}^s$ për çdo s .*

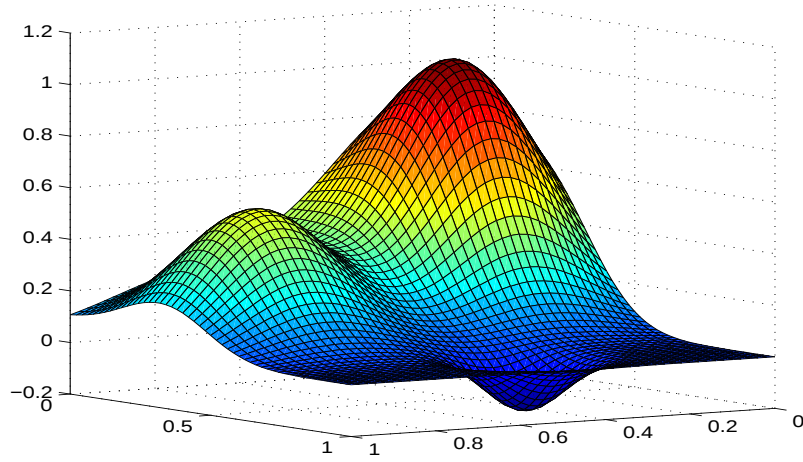
Shembull 2.9 *Më parë pamë se funksionet $\varphi(r) = e^{-\varepsilon r}$, $\varepsilon > 0$ dhe $\varphi(r) = 1/(1+r)^\beta$, $\beta \geq 0$ janë totalisht monotone në $[0, \infty)$ dhe për më tepër nuk janë konstant, atëhere në saj të teoremës 2.9 mund të themi se funksioni Gaussian $\Phi(x) = \varphi(\|x\|^2) = e^{-\varepsilon^2 \|x\|^2}$, $\varepsilon > 0$ dhe funksioni Invers Multiquadric $\Phi = \varphi(\|x\|^2) = 1/(1+\|x\|^2)^\beta$, $\beta \geq 0$ janë rigoroz pozitivisht të përcaktuar dhe radial në $\mathbb{R}^s$.*

Në këtë mënyrë jo vetëm që nuk është e nevojshme llogaritja e transformimit Fourier për të testuar nëse një funksion është rigorozisht pozitivisht i përcaktuar, por u tregua edhe se funksioni Invers Multiquadric është rigoroz pozitivisht i përcaktuar për çdo s në β .

2.3 Interpolimi me funksione baze Gaussianët

Më sipër u treguan disa rezultate teorike për problemin e interpolimit të të dhënave uniforme me anë të funksioneve me bazë radiale. Në këtë pjesë do të shohim disa zbatime, si dhe disa metoda për të rritur saktësinë dhe qendrueshmërinë e interpolimit.

Rikthehemi në barazimin [(2.1), Fq.13], ku shprehet trajta e funksioni interpolues P_f . Nga sa u tha në kapitullin e mëparshëm është e udhës të përdorim si funksione bazë interpoluese Gaussianët, sepse duke qenë funksione totalisht monoton dhe jo konstant dhe për rrjedhojë (në bazë të teoremës 2.9) funksione rigoroz pozitivisht të përcaktuar, atëhere

Figura 2.4 Grafiku i funksionit Franke në katrorin $[0, 1] \times [0, 1]$

matrica e sistemit [(2.2), Fq.13] do të jetë jo singulare. Për pasojë, sistemi do të ketë zgjidhje të vetme c_k për $k = 1, 2, \dots, n$.

Nëse zëvendësojmë Gaussianët në formulën e funksionit interpolues marrim trajtën:

$$P_f(x) = \sum_{k=1}^N c_k e^{-\varepsilon^2 \|x - x_k\|_2^2}, \quad x \in \mathbb{R}^s \quad (2.11)$$

Programi MATLAB **Interpolimi_gaus_2d.m** 2.1, kryen interpolimin me funksione baze Gaussianët. Si funksion të panjohur nga janë gjeneruar të dhënat dy dimensionale kemi marrë *funksionin Franke*

$$f(x, y) = \frac{3}{4} e^{-1/4((9x-2)^2 + (9y-2)^2)} + \frac{3}{4} e^{-1/49((9x+1)^2 - 1/10(9y+1)^2)} + \\ + \frac{1}{2} e^{-1/4((9x-7)^2 + (9y-3)^2)} - \frac{1}{5} e^{-((9x-4)^2 - (9y-7)^2)}, \quad (2.12)$$

i cili është një funksion standart që përdoret shpesh për të testuar problemet e interpolimit dy dimensionale. Grafiku i tij në katorin njësi në $\mathbb{R}^2$ paraqitet në figurën 2.4.

Për këtë shembull do të përdoren të dhëna të cilat ruhen në skedar të jashtëm. Thirrja e të dhënave në programin 2.1 bëhet në rreshtin 15, me anë të komandës **name=sprintf**(...). Në këtë rast të dhënat janë ruajtur në file—in **Data2D_h** dhe kanë të bëjnë me pika halton në palnin $\mathbb{R}^2$.

Përcaktimi i funksionit testues Franke bëhet në rreshtat 7 – 11. Në përgjithësi struktura e këtij programi është e ngjashme me atë të programit [(1.1), Fq.5] të ndërtuar në kapitullin e

parë. Vëmë re se matrica e diferencave e llogaritur në rrshtin 21, i kalohet si parametër hyrës (rreshti 25) funksionit Gaussian të përcaktuar në rreshtin 5. Funksionet për paraqitjet grafike *PlotSurf.m* dhe *PlotError2D.m*, të rreshtave 41 – 42 jepen në Aneksin B.

Programi 2.1 *Interpolimi_gaus_2d.m*

```

• lineskip
1  % Interpolimi_gaus_2d
2  % Skript qe realizon interpolimin RBF 2D
3  % Therret funksionin DistanceMatrix
4      % Percaktojme Gaussianin RBF dhe parametrin
5  gaussian=@(e,r) exp(-(e*r).^2); ep=1.3576;
6  %Percaktojme funksionin Franke si funksion testues
7  f1=@(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
8  f2=@(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
9  f3=@(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
10 f4=@(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
11 testfunction=@(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
12
13 N =289; gridtype = 'h';
14 % Therrasim pikat nyje
15 name = sprintf('Data2D_%d%s',N,gridtype);
16 load(name)
17 ctrs = dsites;
18 neval=40;grid=linspace(0,1,neval);
19 [xe,ye] = meshgrid(grid);epoints = [xe(:) ye(:)];
20 % Llogarisim vlerat e funksionit ne pikat nyje
21 rhs = testfunction(dsites(:,1),dsites(:,2));
22 % Llogarisim matricen e distancave
23 DM_data = DistanceMatrix(dsites,ctrs);
24 % LLogarisim matricen interpoluese
25 IM=gaussian(ep,DM_data);
26 % Llogarisim matricen distance ndermjet nyjeve ku do kryhen llogaritjet dhe
27 % qendrave
28 DM_eval = DistanceMatrix(epoints,ctrs);
29 % Matrica e llogaritjeve
30 EM=gaussian(ep,DM_eval);
31 % Kryejme interpolimin RBF
32 Pf=EM*(IM\rhs);
33 % Llogarisimm zgjidhjet e sakta
34 exact=testfunction(epoints(:,1),epoints(:,2));
35 % LLogarisim Gabimet
36 maxerr = norm(Pf - exact,inf);
37 rms_err = norm(Pf-exact)/neval;
38 fprintf('RMS error: %e\n',rms_err)
39 fprintf('Max error: %e\n',maxerr)
40 fview = [160,20];

```

```

41 PlotSurf(xe, ye, Pf, neval, exact, maxerr, fview);
42 PlotError2D(xe, ye, Pf, exact, maxerr, neval, fview)

```

Në figurën 2.5 paraqitet interpolimi i $N = 289$ pikave halton sëbashku me grafikun e gabimeve, duke përdorur si funksione bazë Gaussianët.

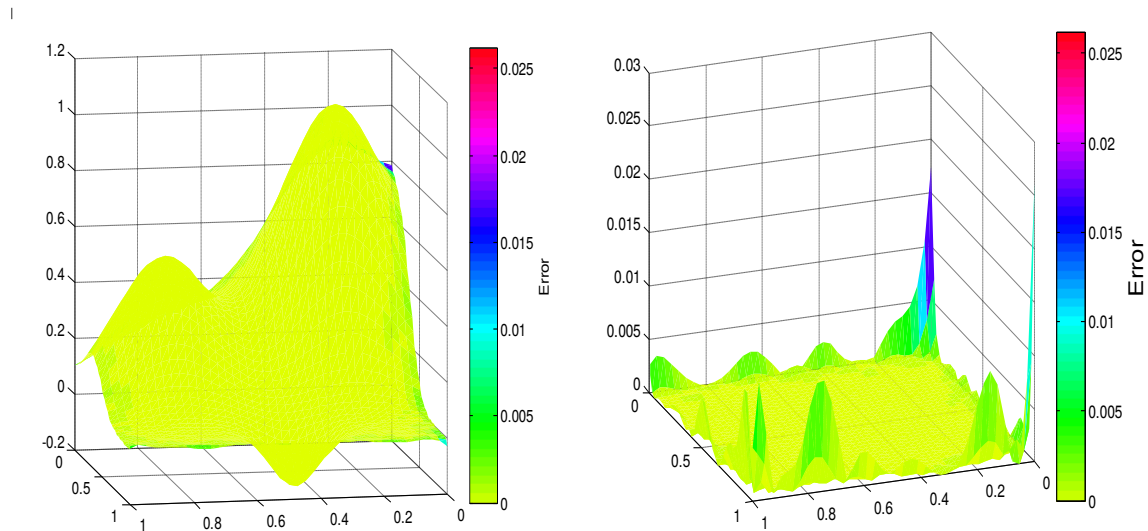


Figura 2.5 **Majtas**: Interpolimi i $N = 289$ pikave halton me anë të gaussianëve me parametrin $\varepsilon = 5$ dhe **Djathtas**: Paraqitja grafike e gabimit të interpolimit

2.4 Interpolimi Stacionar dhe Jo-Stacionar

Në këtë pjesë do të studiohet efekti i parametrin të shkallëzimit ε në qendrueshmërinë numerike dhe saktësinë e zgjidhjes së problemit të interpolimit të të dhënave jo-uniforme (rasti dy dimensional). (shiko [5], Asimi and Hanelli, (2016a)). Si funksione bazë do të përdorim Gaussianët. Për këtë qellim do të ndërtojmë një sërë eksperimentesh, në të cilat do të ndërtohen funksionet interpolues Gaussianë me një vlerë të fiksuar të parametrin ε . Kjo quhet interpolim stacionar. Nga ana tjetër, do të kryhen disa eksperimente ku do të shkallëzohet parametri ε duke u bazuar në dendësinë e të dhënave. Ky quhet interpolim jo stacionar. Në fund do të kryhen krahasimet e dy përjasjeve, në mënyrë që të nxirren përfundimet përkatëse.

Në kapitullin e dytë, pamë (figura [(2.1), Fq.12]) se për vlera të mëdha të parametrin ε funksioni Gaussian bëhet më i "thepisur" dhe më i hollë, në të kundërt për vlera më të vogla të tij bëhet më i "lëmuar" dhe më i "gjerë". Do të shohim se zgjedhja e parametrin ε ndikon në qendrueshmërinë dhe saktësinë e problemit të interpolimit.

Tabela 2.2 Interpolimi i funksioni *sinc* duke përdorur Gaussianin me $\varepsilon = 19.8$

N	Gab-Norm	Gab-Maks
9	$4.29293E - 01$	$1.00000E + 00$
25	$3.79072E - 01$	$1.00000E + 00$
81	$2.41838E - 01$	$9.96995E - 01$
289	$8.35899E - 02$	$9.22630E - 01$
1089	$1.64278E - 02$	$4.46650E - 01$
4225	$3.55230E - 04$	$1.41152E - 02$

Si funksione testuese f për eksperimentet tona do të përdorim funksionin:

$$F(x) = \text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0 \end{cases}$$

Si pika nyje dhe pika ku do të llogaritet interpolantin do të përdorim pikat halton. Pikat nyje do të shërbejnë edhe si qendra.

2.4.1 Interpolimi Jo-Stacionar

Në këtë pjesë do të prezantohen rezultatet e eksperimenteve të kryera me interpolimin me anë të funksioneve Gaussian të $N = 9, 25, 81, 289, 1089, 4225$ pikave nyje duke marrë si parametër $\varepsilon = 19.8$. Marrim këtë vlerë të lartë të parametrut, sepse interpolimi i $N = 4225$ pikave nyje me një vlerë të vogël të tij do të sillte nga MATLAB mesazhe keqkushtëzimi të matricës së sistemit.

Prandaj, presim që interpolimi me këtë parametër i të dhënave të vogla në numër nuk do të jetë shumë i kënaqshëm (shikoni figurën 2.6 dhe tabelën 2.2)

Nga tabela 2.2 shohim se, për rastet $N = 9, 25, 81$ gabimi i normuar është i rendit 10^{-1} . Për rastin $N = 4225$ gabimi maksimal është i rendit 10^{-2} dhe gabimi i normuar është i rendit 10^{-4} .

2.4.2 Interpolimi Stacionar

Fokusohe mi tani në interpolimin stacionar. Dallimi kryesor me interpolimin jo-stacionar, qendron në shkallëzimin e parametrut ε në varësi të dendësisë h të të dhënave. Kjo do të thotë përdorim të funksioneve të thepisur (ε i madh) për numër të madh pikash dhe vlere të vogla të ε për numër të vogël pikash nyje.

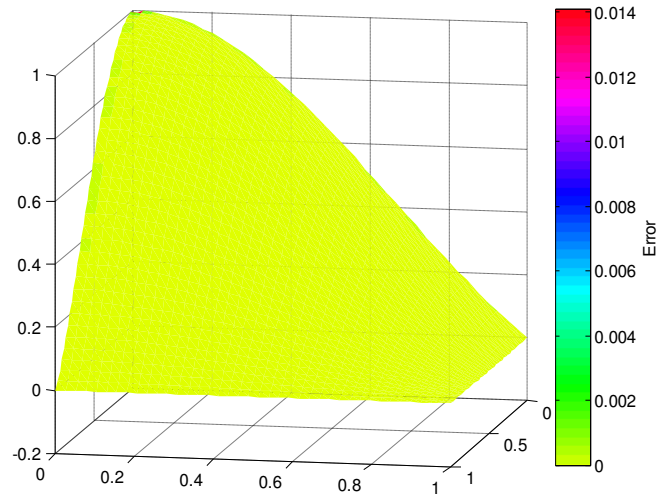


Figura 2.6 Interpolimi me Gaussian dhe paramet er $\varepsilon = 19.8$ i $N = 4225$ pikave halton.

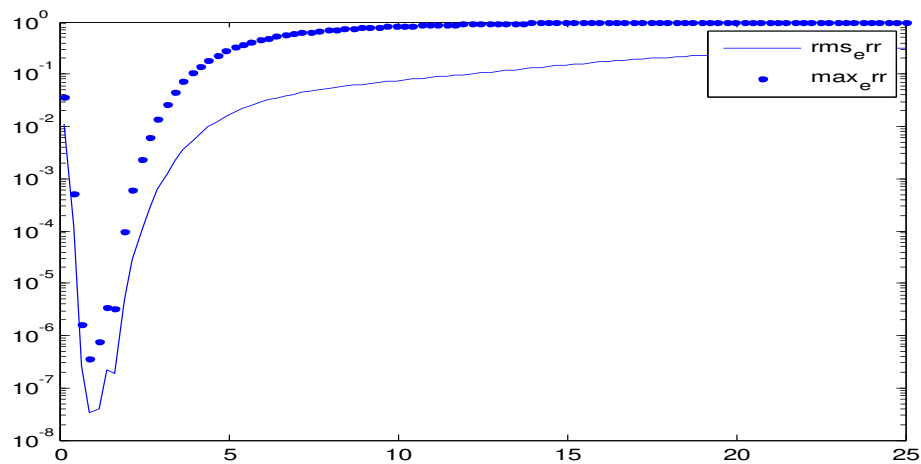


Figura 2.7 Gabimet maksimale dhe mesatare kundrejt ε për $N = 81$ pika halton

Zakonisht dendësia h e të dhënave përcaktohet si:

$$h = \sup_{x \in \Omega} \min_{x_j \in \mathcal{X}} \|x - x_j\|_2,$$

dhe tregon se sa "mirë" mbushet hapsira Ω nga bashkësia e të dhënave $\mathcal{X}$. Një përkufizim gjeometik i saj, është ai i rrezes më të madhe të rrethit që mund të vendoset mes të dhënave.

Për këtë qëllim në Figurat e 2.7, 2.8, 2.9, 2.10 paraqiten grafikët e gabimeve mesatarë dhe atyre maksimal, si funksione të parametrit të shkallëzimit ε për katër bashkësi pikas halton $N = 81, N = 289, N = 1089, N = 4225$.

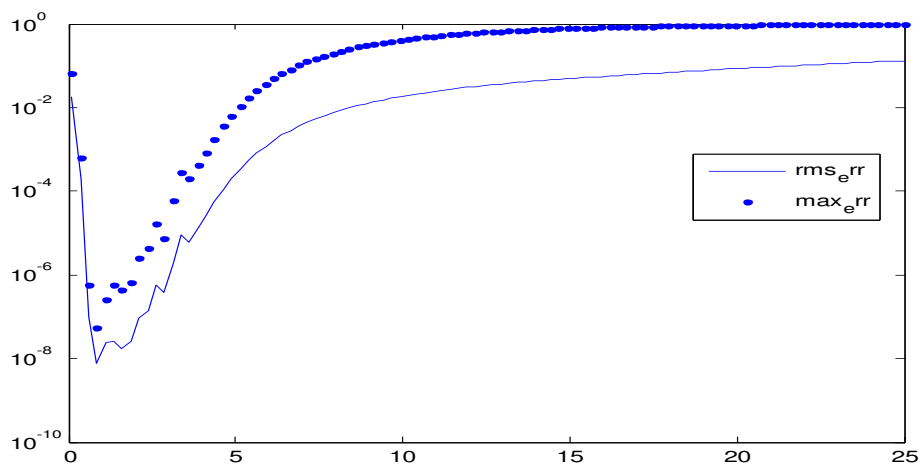


Figura 2.8 Gabimet maksimale dhe mesatare kundrejt ε për $N = 289$ pika halton

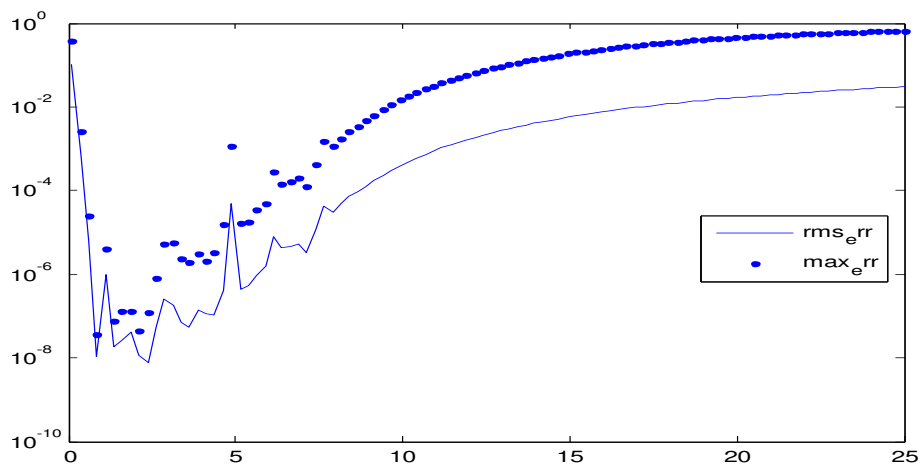


Figura 2.9 Gabimet maksimale dhe mesatare kundrejt ε për $N = 1089$ pika halton

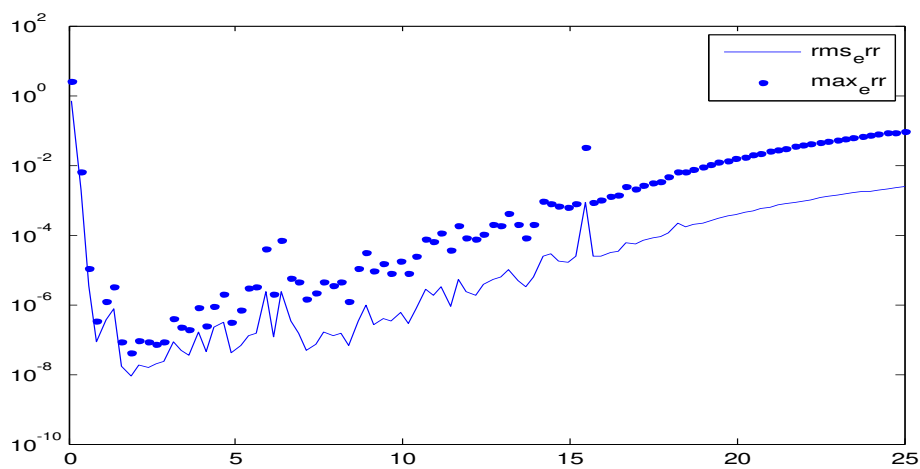


Figura 2.10 Gabimet maksimale dhe mesatare kundrejt ε për $N = 4225$ pika halton

Tabela 2.3 Rezultatet e Interpolimit Stacionar të funksionit *Sinc* për vlera të ndryshme të N —së

vlera min "sigurtë"				gabimi minimal		
N	ϵ	Gab-Normuar	Gabimi Maksimal	ϵ	Gab-Normuar	Gabimi Maksimal
81	1.3576	$3.002E-07$	$5.304E-06$	0.8545	$3.278E-08$	$3.651E-07$
289	4.3758	$5.472E-05$	$1.771E-03$	0.8545	$7.918E-09$	$5.378E-08$
1089	9.9091	$3.925E-04$	$1.449E-02$	2.3636	$7.695E-09$	$1.238E-07$
4225	19.9697	$3.880E-04$	$1.542E-02$	1.8606	$9.100E-09$	$4.278E-08$

Nga figurat shohim se kurbat e gabimeve nuk janë monotone. Gjithashtu mund të shohim një vlerë optimale të parametrin të shkallëzimit për secilin rast. Një gjë tjetër shumë e rëndësishme që mund të shihet nga figurat është që vlera optimale e parametrin, në disa raste, gjendet në një zonë të "çrregullt" të kurbës së gabimeve. Për vlerat e parametrin ϵ nga kjo zonë, MATLAB jep mesazhe për keqkushtëzim të matricës interpoluese.

Në këtë mënyrë ne mund të përcaktojmë parametrin "me të mirë" në dy mënyra: njëherë si parametri për të cilin arrihet gabimi minimal (shiko tabelën 2.3 kolonat 4 – 6) dhe e dyta si vlera minimale e parametrin për të cilin nuk kemi mesazh për keqkushtëzim të matricës interpoluese (tabela 2.3 kolonat 4 – 6). Në këtë rast parametrin ϵ e quajmë "të sigurt".

Nga tabela 2.3 shohim se vlera optimale e parametrin ϵ ndodhet në zonën "jo të sigurt" të grafikut. Diferenca e dy vlerave të parametrin optimal sipas të dy mënyrave, rritet shumë me rritjen e numrit të pikave nyje N . Këto eksperimente në MATLAB kryen me një modifikim të programit (2.1). Kodi paraqitet si programi (2.2).

Programi 2.2 *Interpolimi_gaus_2d_eps.m*

```

lineskip
1 % Interpolimi_gaus_eps
2 % Skript qe realizon interpolimin 2D
3 % per te llogaritur parametrin eps optimal
4 % Therret funksionin DistanceMatrix
5 % Percaktojme Gaussianin RBF dhe parametrin
6 rbf=@(e,r) exp(-(e*r).^2);
7 minep=0.1; maxep=25; nep=100;
8 eps=linspace(minep,maxep,nep);
9
10 % Percaktojme funksionin Franke si funksion testues
11 testfunction=@(x,y) sinc(x).*sinc(y);
12 N = 289; gridtype = 'h';
13 % Therrasim pikat nyje
14 name = sprintf('Data2D_%d%s',N,gridtype);
15 load(name)
16 ctrs = dsites;
```

```

17 neval=40; grid=linspace(0,1,neval);
18 [xe,ye] = meshgrid(grid); epoints = [xe(:) ye(:)];
19 % Llogarisim vlerat e funksionit ne pikat nyje
20 rhs = testfunction(dsites(:,1),dsites(:,2));
21 % Llogarisim matricen e distancave
22 DM_data = DistanceMatrix(dsites, ctrs);
23 for i=1:length(eps)
24     % Llogarisim matricen interpoluese
25     IM=rbf(eps(i),DM_data);
26     % Llogarisim matricen distance ndermjet nyjeve ku do kryhen llogaritjet dhe
27     % qendrave
28     DM_eval = DistanceMatrix(epoints, ctrs);
29     % Matrica e llogaritjeve
30     EM=rbf(eps(i),DM_eval);
31     % Kryejme interpolimin RBF
32     Pf=EM*(IM\rhs);
33     % Llogarisim zgjidhjet e sakta
34     exact=testfunction(epoints(:,1),epoints(:,2));
35     % Llogarisim Gabimet
36     maxerr(i) = norm(Pf - exact, inf);
37     rms_err(i) = norm(Pf-exact)/neval;
38     eps(i)
39 end
40 semilogy(eps, rms_err)
41 hold on
42 semilogy(eps, maxerr, 'r')
43 legend('rms_err', 'max_err')
44 hold off

```

Si përfundim mund të nxjerrim disa konkluzione. Së pari, parametri ε ndikon shumë në cilësinë dhe qendrueshmërinë e interpolimit me funksione bazë gaussianët. Së dyti, për të punuar me vlera të mëdha pikash nyje duhet të merret vlerë e madhe e parametrut ε , sepse përndryshe do të kemi keqkushtëzim të matricës interpoluese.

Si përfundim, eksperimentet numerike për interpolimin stacionar dhe jo-stacionar, ndihmojnë në njohjen e ndikimit të parametrut ε në interpolimin e të dhënave jo-uniforme, por kemi nevojë për njohejen e funksioneve gjenerues të të dhënave, gjë kjo që në praktikë nuk është e mundur. Për këtë problem do të filtet në vazhdim të këtij kapitulli.

2.5 Përcaktimi i vlerës optimale të parametrut të shkallëzimit

Fillojmë duke konsideruar problemin e zgjedhjes së parametrut të shkallëzimit ε për një grup të fiksuar të dhënash jo-uniforme. Kjo është situata që haset më shpesh në praktikë.

Në këtë pjesë do të krahasojmë dy metoda për të përcaktuar vlerën optimale të parametrut ε gjatë interpolimit të një bashkësie të dhënash jo-uniforme (shiko [4], (Asimi and Hanelli), (2016b)). Fillimisht do prezantojmë metodat "Provë Gabim" ("Trial and Error") dhe "Përkashtojë një të dhënë" ("Leave one Out").

Së dyti, për të testuar dhe krahasuar këto dy mënyra, do të kryhen disa eksperimente me të dhëna një dhe dy dimensionale me N-vlera të ndryshme të dhënash. Gjatë ndërtimit të funksionit interpolues, vlerat e parametrut të shkallëzimit ε do të mbahen të fiksuara për të gjitha funksionet bazë. Një përjashtim tjetër e këtij problemi mund të jetë përcaktimi i një parametri për çdo k , si dhe ndryshimi i funksioneve bazë për çdo k . Për kërkime të kryera në këtë drejtim shikoni ([27], [11]).

Pra mund të formulojmë problemin e mëposhtëm:

- Çfarë vlere e parametrut të shkallëzimit ε do të merret në funksionet bazë

$$\varphi(\|x - x_j\|_2) = e^{-\varepsilon^2 \|x - x_j\|_2^2}, \quad x \in \mathbb{R}^s ?$$

2.5.1 Përcaktimi i vlerës optimale të parametrut me anë të metodës "Provë-Gabim"

Strategjia më e thjeshtë është të kryhen një numër i caktuar interpolimesh me vlera të ndryshme të parametrut ε dhe më pas të zgjidhet ai me gabimin më të vogël duke krahasuar rezultatet e interpolimit me ato reale. Kuptohet, që për këtë përjashtim duhet të njihen funksionet "*e panjohur f*", të cilët kanë gjeneruar të dhënat. Por, nëse ne njohim funksionet që kanë gjeneruar të dhënat, atëherë nuk do të ishte nevoja për interpolim të dhënash. Në këtë kuptim metoda e mësipërme mbetet vetëm teorike dhe paraqet interes vetëm në studime krahasuese të metodave të tjera.

Konkretisht ne do të përdorim rezultatet e arritura me këtë metodë si rezultate krahasuese për metodën tjetër që do të shqyrtojmë. Si funksion të "*panjohur f*" do të përdorim funksionin:

$$F(x) = \text{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0 \end{cases}$$

Si pika nyje dhe pika ku do të kryhen llogaritjet do të merren N-pika Halton.

Modelet interpoluese do të llogariten me anë të programit [(2.1), Fq.24]. Ky program modifikohet mjaft lehtë për të fituar një kod të ngjashëm për rastin një dimensional.

2.5.2 Përcaktimi i vlerës optimale të parametrut me anë të metodës "Përrjashto një të dhënë"

Strategjia tjetër që do të përdoret për përcaktimin e vlerës optimale të parametrut ε është metoda "Përrjashto një të dhënë". (Shiko [32]). Kjo metodë përdoret edhe në statistikë me emrin "Parashikimi i mbetjes së shumës së katrorëve". Sipas kësaj mënyre, vlera optimale e parametrut ε merret ajo vlerë, për të cilën arrihet vlera minimale e gabimit (në sensin e katrorëve më të vegjël) të përfutur nga interpolimi duke hequr njërin nga nyjet. Ky proces kryhet duke hequr me rradhë një e nga një të gjitha nyjet. Më pas gabimi llogaritet midis vlerës së interpolantit të ndërtuar me vlerën e skatë në nyjen e hequr. Në këtë mënyrë ndërtohet një vektor gabimesh me gjatësi të njëjtë me atë të të dhënave.

Më specifikuohet kemi:

P_f^k është interpoluesi i të dhënave $\{f_1, \dots, f_{k-1}, f_{k+1}, \dots, f_N\}$, kështu:

$$P_f^{[k]}(x) = \sum_{j=1, j \neq k}^N c_j^{[k]} \varphi(\|x - x_j\|), \quad \mathbf{x} \in \mathbb{R}^s$$

i tillë që:

$$P_f^{[k]}(x_i) = f_i, \quad i = 1, \dots, k-1, k+1, \dots, N$$

dhe nëse shënojmë me E_k gabimin

$$E_k = f_k - P_f^{[k]}(x)$$

në pikë x_k që nuk është përdorur në ndërtimin e interpolantit, atëherë cilësia e interpolimit jepet me anë të normës së vektorit të gabimeve $E = [E_1, E_2, \dots, E_N]$ të krijuar duke hequr me rradhë secilën nga pikat nyje. Për të përcaktuar vlerën optimale të parametrut ε , mund të vendoset një cikël për vlera të ndryshme të parametrut. Implementimi i kësaj metode në këtë formë ka shumë kosto llogaritëse, si pasojë e rendit N^4 operacione. Në punimin e tij Rippa tregoi se llogaritja e komponentes së gabimit jepet me anë të formulës së vetme:

$$E_k = \frac{c_k}{A_{kk}^{-1}}$$

ku c_k është koeficienti i k -të i interpolantit P_f të ndërtuar në të gjitha të dhënat dhe A_{kk}^{-1} është elementi i k -të i diagonales së matricës të anasjelltë të matricës interpoluese A . Kjo përjasje e metodës e ul numrin e operacioneve numerike në N^3 , sepse c_k dhe A^{-1} nevojiten të llogariten vetëm njëherë për çdo vlerë të ε . Elementët e vektorit të gabimeve mund të llogariten në

MATLAB vetëm me anë të një instruksioni (rreshti 26 në programin 2.3) duke vektorizuar formulën e mësipërme.

Programi 2.3 *PerjashtoTeDhene.m*

```

lineskip
1 % Perjashto nje te dhene
2 % PerjashtoTeDhene.m
3 % Skript qe realizon Perjashto nje te dhene
4 % Therret funksionin DistanceMatrix
5 % Percaktojme Gaussianin RBF dhe parametrin
6 rbf=@(e,r) exp(-(e*r).^2);
7 minep=0; maxep=20; nep=100;
8 eps=linspace(minep,maxep,nep);
9 testfunction=@(x,y) sinc(x).*sinc(y);
10 N = 289; gridtype = 'h';
11 % Therrasim pikat nyje
12 name = sprintf('Data2D_%d%s',N,gridtype);
13 load(name)
14 ctrs = dsites;
15 neval=40; grid=linspace(0,1,neval);
16 [xe,ye] = meshgrid(grid); epoints = [xe(:) ye(:)];
17 % Llogarisim vlerat e funksionit ne pikat nyje
18 rhs = testfunction(dsites(:,1),dsites(:,2));
19 % Llogarisim matricen e distancave
20 DM_data = DistanceMatrix(dsites,ctrs);
21 for i=1:length(eps)
22     % Llogarisim matricen interpoluese
23     IM=rbf(eps(i),DM_data);
24     % Llogarisim funksionin e gabimit
25     invIM=pinv(IM);
26     EF=(invIM*rhs)./diag(invIM);
27     % Llogarisim Normen e gabimit
28     maxEF(i)=norm(EF(:),inf);
29 end
30 fprintf('Vlera me e vogel e gabimit: %e\n',min(maxEF))
31 fprintf('merret per epsilon: %f\n',eps(maxEF==min(maxEF)))
32 figure;
33 semilogy(eps,maxEF)

```

2.5.3 Disa Eksperimente Numerike

Në këtë pjesë do të kryhen disa eksperimente numerike për të krahasuar metodat e paraqitur më lart. Do shohim nëse ka ngjashmëri mes rezultateve të përfutuara me to.

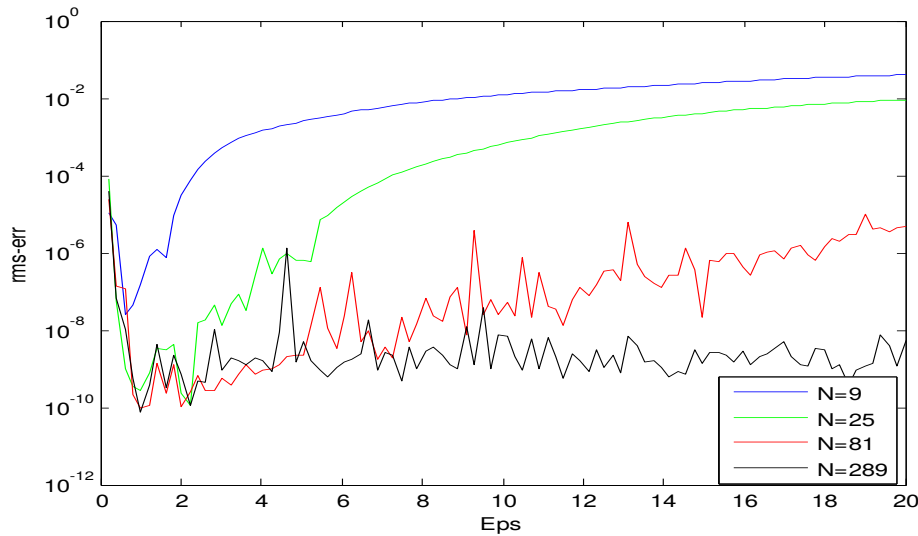


Figura 2.11 Kurbat e gabimit si funksion i parametrut ε për interpolimin e N -të dhënave 1D. Metoda "Provë-Gabim"

Siç përmendëm edhe më sipër për eksperimentet tona do të përdorim si funksion "të panjohur", nga i cili kanë ardhur të dhënat, funksionin $F(x) = \text{sinc}(x)$. Si pika njëje $x_i \in \mathbb{R}^s$, $i = 1, 2, \dots, N$, për $s = 1$ dhe $s = 2$, do të merren N — pika halton ($N = 9, 25, 81, 259$).

Fillimisht do të llogariten gabimet e përfutuara sipas metodës së parë, pra "Provë-Gabim", duke ndërtuar me rradhë për çdo $\varepsilon \in [0, 20]$ interpolantin përkatës. Për këtë qëllim do të përdoret programi i ndërtuar më parë [(2.2), Fq.29].

Në mënyrë të ngjashme ndërtohet dhe programi për rastin një dimensional. Programi 2.2 ndërton kurbat e gabimit maksimal dhe të normalizuar për vlera të ndryshme të parametrut të shkallëzimit ε .

Në figurën 2.11 dhe 2.12 paraqiten kurbat e gabimit si funksion i parametrut ε përkatësisht për rastin 1D dhe 2D.

Siç mund të shikojmë nga figura 2.11, për rastin e interpolimit 1D, gabimi minimal është i të njëjtit rend për $N=25$, $N=61$, $N=289$ dhe arrihet afërsisht në të njëjtën vlerë të parametrut ε . Në rastin kur $N=9$ gabimi është më i madh se rastet e tjera, por minimumi i tij arrihet afërsisht në të njëjtën vlerë të parametrut ε . Gabimi minimal për rastet $N=81$ dhe $N=289$ është i të njëjtit rend 10^{-8} . Për rastet $N=9$ dhe $N=25$ është përkatsisht 10^{-3} dhe 10^{-4} .

Kryejmë tani eksperimentet me metodën "Përjashto një të dhënë". Në figurën 2.13 dhe 2.14 paraqiten kurbat e gabimit si funksion i parametrut ε përkatësisht për rastin 1D dhe 2D me metodën "Përjashto një të dhënë".

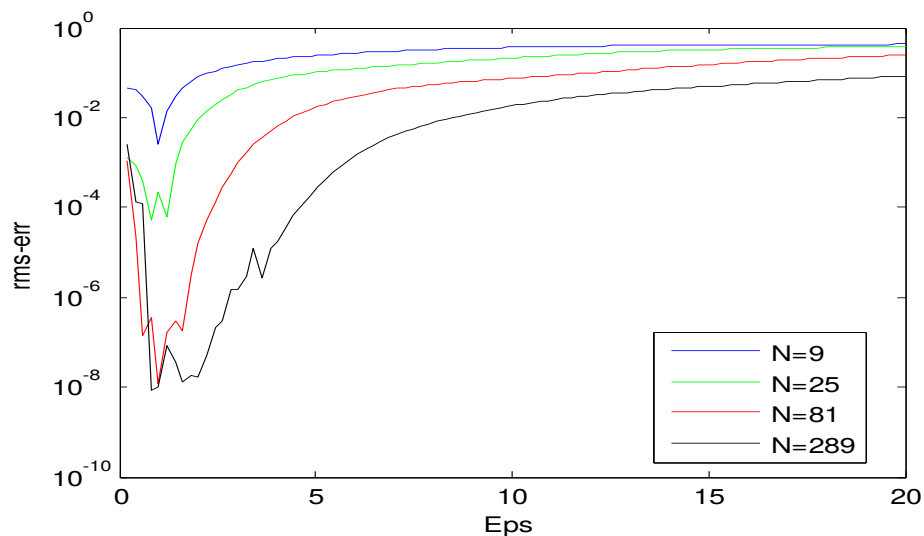


Figura 2.12 Kurbat e gabimit si funksion i parametrut ϵ për interpolimin e N-të dhënave 2D. Metoda "Provë-Gabim"

Shikojmë nga figura e mësipërme se gabimi minimal për rastin $N=81$ dhe $N=28$ është i rendit 10^{-8} . Për rastin e $N=9$ dhe $N=25$ pikash një interpolimi është më i dobët dhe gabimi minimal është përkatësisht i rendit 10^{-2} dhe 10^{-4} . Për të kryer interpolimin njëpërmasor me metodën "Përfashto një të dhënë" (Leave one out) mund të modifikohet shumë kollaj programi 2.3.

Në rastin e interpolimit të të dhënave 2D shohim se gabimi minimal për $N=289$ është i rendit 10^{-6} dhe për $N=9$ gabimi optimal është i rendit 10^{-2} . Vihet re se nëse numri i nyjeve zvogëlohet gabimi rritet.

Në tabelën e mëposhtme (Tabela 2.4), paraqiten vlerat optimale të parametrut ϵ për të gjitha rastet e interpolimit të të dhënave një dimensionale me të dyja metodat sëbashku me gabimet përkatëse.

Tabela 2.4 Vlerat optimale e parametrut ϵ për interpolimin e të dhënave 1D

Metoda "Provë-Gabim"			Metoda "Përfashto një të dhënë"	
N	eps	Gabimi	eps	Gabimi
9	0.6	$2.5159E-08$	0.81	$1.8399E-06$
25	2.22	$1.1627E-10$	1.82	$1.7462E-07$
81	1.01	$9.8934E-11$	2.02	$1.5463E-06$
289	1.01	$7.5343E-11$	15.96	$6.8118E-06$

Vihet re se në rastin e interpolimit të $N=9$ dhe $N=25$ të dhënash një dimensionale, vlerat optimale për parametrin arrihen në vlera të përafërta. Diferenca më e madhe ndodh në rastin e interpolimit të $N=289$ të dhënave, përkatësisht $\epsilon = 1.01$ për metodën "Provë gabim" dhe

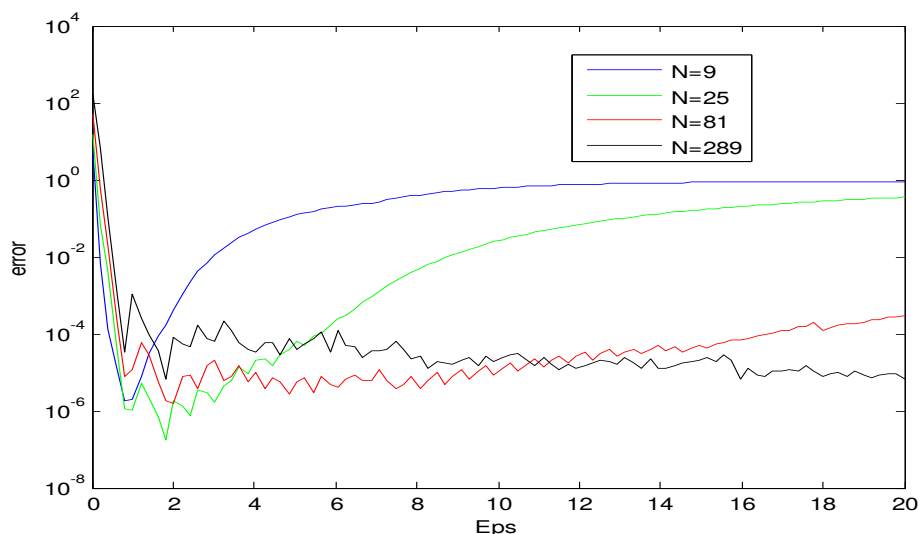


Figura 2.13 Kurbat e gabimit si funksion i parametrut ε për interpolimin e N-të dhënave 1D. Metoda "Përrjashto një të dhënë"

$\varepsilon = 15.96$ për metodën "Përrjashto një të dhënë", por nëse i referohemi figurës 2.13 shohim se marrim një vlerë të kënaqshme gabimi, për ε të rendit të metodës "Provë Gabim".

Një situatë e ngjashme ndodh edhe në rastin e interpolimit dypërmasor (shiko Tabelën 2.5).

Tabela 2.5 Vlerat optimale e parametrut ε për interpolimin e të dhënave 2D

Metoda "Provë-Gabim"			Metoda "Përrjashto një të dhënë"	
N	eps	Gabimi	eps	Gabimi
9	1.01	$2.5000E-03$	1.01	$1.3787E-02$
25	0.81	$5.0979E-05$	1.21	$4.3946E-04$
81	1.01	$1.1968E-08$	1.01	$4.5232E-06$
289	0.81	$8.3476E-09$	1.81	$1.5268E-06$

Shohim se për secilin rast marrim vlera shumë të përafërta të ε optimal. Në rastin $N=9$ dhe $N=81$ parametri ε është i njëjtë 1.01.

Si përfundim mund të themi se, metoda "Provë Gabim" edhe pse nuk mund të përdoret në një situatë reale (ka nevojë të njihen funksionet që gjenerojnë të dhënat), na jep një mjet shumë të mirë krahasimi me metoda të tjera për përcaktimin e vlerës optimale të parametrut ε .

Metoda "Përrjashto një të dhënë" jep rezultate të përafërta me metodën "Provë Gabim", edhe pse nuk ka nevojë për njohjen e funksioneve që gjenerojnë të dhënat. Rezultatet e përfutura ishin të përafërta në të gjitha rastet e interpolimit 1D dhe 2D për vlera të ndryshme të N-së. Ndryshimi i vetëm i madh ndodhi në rastin 1D për $N=289$ të dhëna. Duhet theksuar

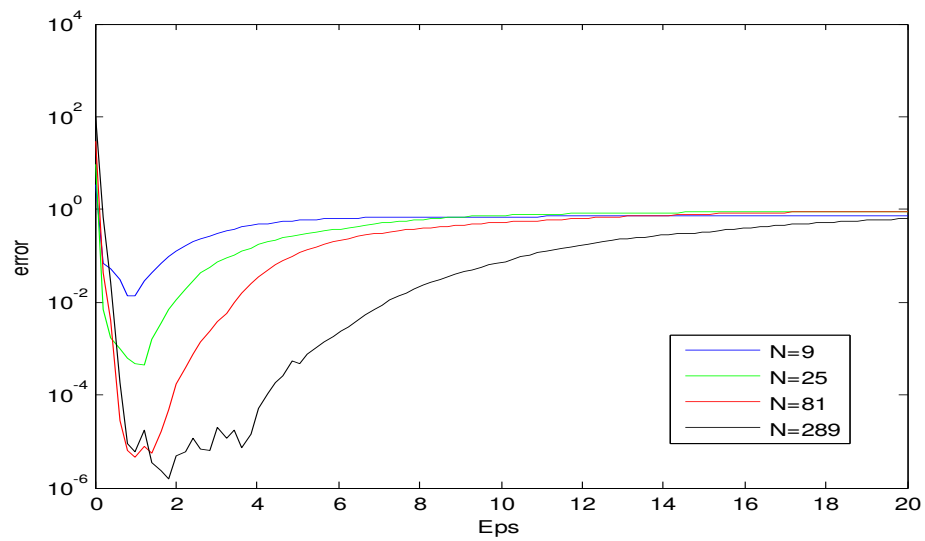


Figura 2.14 Kurbat e gabimit si funksion i parametrut ε për interpolimin e N -të dhënave 2D. Metoda "Përrjashtoj një të dhënë"

(bazuar ne figurën 2.13) që gabimi në këtë rast ishte i të njëjtit rend për të gjitha vlerat e parametrut ε .

Kapitulli 3

Përafrimi i të dhënave jo uniforme me funksionet me bazë radiale: Zbatime dhe eksperimente numerike

Deri tani është trajtuar problemi i interpolimit të të dhënave jo uniforme. Por, shpesh është më e përshtatshme përafrimi i të dhënave me anë të metodës së katrorëve më të vegjël. Kjo gjë është e nevojshme në rastet kur të dhënat nuk vijnë nga një ligj i fiksuar matematik, por janë të përziera edhe me vlera rastësore (zhurma), ose në rastin kur numri i të dhënave është shumë i madh.

Tregohet (shiko [16]) se problemi i përafrimit me funksionet me bazë radiale mund të trajtohet si problem optimizimi.

Shtrojmë problemin e gjetjes së një funksioni P_f të formës

$$P_f(x) = \sum_{j=1}^M c_j \Phi(x, x_j), \quad x \in \mathbb{R}^s$$

ku numri M i funksioneve bazë në përgjithësi nuk e kalon numrin N të të dhënave. Që këtë na duhet të përcaktojmë koeficientët $\mathbf{c} = [c_1, \dots, c_M]^T$, në mënyrë të tillë që të minimizohet forma kuadratike

$$\frac{1}{2} \mathbf{c}^T Q \mathbf{c} \tag{3.1}$$

me disa matrica simetrike pozitivisht të përcaktuara Q që u nënshtrohen kufizimeve lineare

$$A \mathbf{c} = \mathbf{f} \tag{3.2}$$

ku A është një matricë me përmasa $N \times M$ me rang të plotë dhe krahu i djathtë $\mathbf{f} = [f_1, \dots, f_N]^T$ jepet. Një problem minimizimi me këto kushte mund të kthehet në një sistem ekuacionesh lineare duke futur konceptin e *shumëzuesve Lagranzh* $\lambda = [\lambda_1, \dots, \lambda_N]^T$, p.sh konsiderojmë gjetjen e minimumit të

$$\frac{1}{2}c^T Qc - \lambda^T [Ac - f] \quad (3.3)$$

si funksion të c dhe λ . Përderisa Q është një matricë pozitivisht e përcaktuar, dihet se funksionali që po minimizohet është konveks dhe kështu ka një minimum të vetëm. Kushti i nevojshëm (që në këtë rast është edhe i mjaftueshëm) i gjetjes së tij, është zgjidhja e sistemit të formuar nga barazimi më zero i derivateve në lidhje me c dhe λ . Në këtë mënyrë marrim sistemin

$$\begin{aligned} Qc - A^T \lambda &= 0 \\ Ac - f &= 0 \end{aligned}$$

që në trajtë matricore shkruhet

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} c \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}$$

Meqënëse matrica Q është pozitivisht e përcaktuar, ajo është e kthyeshme. Zgjidhim sistemin e mësipërm me metodën e Gausit (në bllok) dhe marrim zgjidhjen

$$\lambda = (AQ^{-1}A^T)^{-1}f \quad (3.4)$$

$$c = Q^{-1}A^T(AQ^{-1}A^T)^{-1}f. \quad (3.5)$$

Më konkretisht, nëse forma kuadratike përfaqëson normën në hapsirën e Interpolantit $P_f(x) = \sum_{j=1}^M c_j \Phi(\cdot, x_j)$, p.sh

$$\|P_f\|_{N_\Phi(\omega)}^2 = \sum_{i=1}^M \sum_{j=1}^M c_i c_j \Phi(x_i, x_j) = c^T Qc$$

me $Q_{ij} = \Phi(x_i, x_j)$ dhe $\mathbf{c} = [c_1, \dots, c_M]^T$, si dhe me kushte lineare kushtet e interpolimit

$$Ac = f \iff P_f(x) = f_i, \quad i = 1, \dots, M,$$

ku $A = A^T = Q$ (simetrike), të njëjtën c si më sipër dhe vektorin e të dhënave $\mathbf{f} = [f_1, \dots, f_M]^T$, atëhere shumëzuesit Lagranzh (3.4) bëhen

$$\lambda = A^{-1}f$$

dhe koeficientët jepen nga

$$c = \lambda$$

nga barazimi (3.5). Prandaj, interpolanti me normë më të vogël gjendet duke zgjidhur më vete ekuacionet e interpolimit

3.1 Përafrimi i normalizuar me katrorë më të vegjël

Meqënëse u fokusuam në një këndvështrim të gjerë, atë që P_f gjenerohet nga M funksione baze dhe N kushte lineare, formulimi i problemit të mësipërm mbulon njëkohësisht rastet e përafrimit me katrorë më të vegjël të mbi- dhe nën-përcaktuar ku forma kuadratike $c^T Qc$ paraqet një term *sheshues* (ose *normalizues*). Ky term nuk është i nevojshëm për të përfutur një zgjidhje të vetme të sistemit $Ac = f$ në rastin e sistemit të mbi-përcaktuar ($N \geq M$), por duhet në rastin e sistemit të nën-përcaktuar.

Zakonisht problemi i përafrit të normalizuar me katrorë më të vegjël formulohet si minimizim i

$$\begin{aligned} & \frac{1}{2}c^T Qc + \omega \sum_{j=1}^N \left(P_f(x_j) - f_j \right)^2 \\ \iff & \frac{1}{2}c^T Qc + \omega (Ac - f)^T (Ac - f). \end{aligned} \quad (3.6)$$

Forma kuadratike $c^T Qc$ kontrollon lëmueshmërinë e funksionit përafrues dhe "afërsinë" me të dhënat në termin e katrorëve më të vegjël. Parametri ω kontrollon bashkveprimin midis këtyre të dyjave, ku vlera më e madhe e ω e kthen peshoren nga afërsia me pikat.

Formulimi i problemit sipas 3.6 përdoret në *teorinë e normalizimit* (ang. regularization theory, shiko [13] dhe [22]). I njëjti formulim përdoret gjithashtu në *përafrimin e kushtëzuar me katrorë më të vegjël* (shiko [38]), në literaturën e "splaineve lëmues". Në praktikë, një përafrim i kushtëzuar katrorësh më të vegjël është shumë i dobishëm në rastin kur të dhënat nuk janë shumë të "besueshme", p.sh në to ka edhe zhurma. Problemi i minimizimit të 3.6 njihet gjithashtu si *regres i ngurtë* në literaturat statistike.

3.2 Përafrimi i katrorëve më të vegjël kur qendrat ndryshojnë nga të dhënat

Shtrojmë problemin e përafrimit të një grupi të dhënash $\chi = \{x_1, \dots, x_N\}$. Paraqesim tani një bashkësi tjetër të dhënash $\Xi = \{\xi_i\}_{i=1}^M$ ku qendërzohen funksionet bazë. Zakonisht kemi $M \leq N$ dhe rasti $M = N$, $\Xi = \chi$ përbën problemin e interpolimit të të dhënave. Prandaj mund të kërkojmë që funksioni përafrues të jetë i formës

$$Q_f(x) = \sum_{j=1}^M c_j \Phi(x, \xi_j), \quad x \in \mathbb{R}^s. \quad (3.7)$$

Koeficientët c_j mund të gjenden si zgjidhje e katrorëve më të vegjël të $Ac = f$, duke minimizuar $\|Q_f - f\|_2^2$ ku norma l_2

$$\|f\|_2^2 = \sum_{i=1}^N [f(x_i)]^2, \quad x_i \in \chi$$

rrjedh prej prodhimit numerik

$$\langle f, g \rangle = \sum_{i=1}^N f(x_i)g(x_i), \quad x_i \in \chi \quad (3.8)$$

Problemi i përafrimit do të ketë zgjidhje të vetme nëse matrica e sistemit A me elementë

$$A_{jk} = \Phi(x_j, \xi_k), \quad j = 1, \dots, N, \quad k = 1, \dots, M$$

do të kishte rang të plotë.

Nëse qendrat Ξ zgjidhen të tilla që të jenë një nënbashkësi e pikave nyje χ , atëherë nga sa thamë për problemin e interpolimit me funksionet me bazë radiale, matrica jonë do të ketë rang të plotë. Kjo është e vërtetë, përderisa në këtë rast matrica A do të ketë një nënmatricë të rendit $M \times M$ e cila nuk është singulare (ka vetinë që është matricë interpolimi nga funksione me bazë radiale). Sistemi i mbi-përcakuar $Ac = f$ mund të zgjidhet me anë të teknikave a algjibrës lineare, psh me anë të faktorizimit QR . Prandaj, kodi i programit për përafrimin dy dimensional me funksione me bazë radiale është ngjashëm me programin [(2.1), Fq.24].

Programi 3.1 *Perafrim_gaus_2D.m*

```

lineskip
1 % Perafrimi_gaus_2D
2 % Perafrimi me ane te funksioneve me baze radiale

```

```

3 % Therret funksionet: DistanceMatrix, PlotSurf, PlotError2D
4 clear all;clc;
5 rbf = @(e,r) exp(-(e*r).^2); ep = 9.68;
6
7
8 f1=@(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
9 f2=@(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
10 f3=@(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
11 f4=@(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
12 testfunction=@(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
13
14 N =1089; gridtype = 'h';
15 M = 289; grid2type = 'h';
16 neval = 40;
17 % Thirren qendrat
18 name=sprintf('Data2D_%d%s',M,grid2type);
19 load(name)
20 ctrs = dsites;
21 % Pikat nyje
22 name=sprintf('Data2D_%d%s',N,gridtype);load(name)
23 % Matrica e distancave mes nyjeve dhe qendrave
24 DM_data = DistanceMatrix(dsites,ctrs);
25
26 CM = rbf(ep,DM_data);
27 % Krah i djathte i sistemit
28 rhs = testfunction(dsites(:,1),dsites(:,2));
29 % Krijohen pikat e llogaritjes se perafrimit
30 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
31 epoints = [xe(:) ye(:)];
32 % Matrica e distancave mes pikave te vleresimit dhe qendrave
33 DM_eval = DistanceMatrix(epoints,ctrs);
34 EM = rbf(ep,DM_eval);
35 % Perafrimi me funksionet me baze radiale
36 Pf = EM * (CM\rhs);
37 % Zgjidhja e sakte
38 exact = testfunction(epoints(:,1),epoints(:,2));
39 % Vleresohen gabimet
40 maxerr = norm(Pf-exact,inf);
41 rms_err = norm(Pf-exact)/neval;
42 % Paraqitjet grafike
43 figure; fview = [100,30];
44 caption = sprintf('%d nyje dhe %d qendra',N,M);
45 title(caption);
46 plot(dsites(:,1),dsites(:,2),'bo',ctrs(:,1),ctrs(:,2),'r+');
47 legend('nyje','qendra')
48 sprintf('Gab_rms: %d',rms_err)
49 sprintf('Gab_max: %d',maxerr)
50 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);

```

51 `PlotError2D(xe, ye, Pf, exact, maxerr, neval, fview);`

Në rrshtat 4 dhe 6 – 7 të programit 3.1 bëhet marrja e M të dhënave që do të përdoren si qendra. Piakt nyje thirren në rreshtin 22. Ndërsa, zgjidhja e problemit të përafrimit bëhet në rreshtin 38. Dy rreshtat e fundit të kodit bëjnë të mundur paraqitjen grafike të modelit dhe gabimit.

Në figurën 3.1 paraqiten $N = 1089$ pika nyje me $M = 289$ qendra. Përafrimi duke

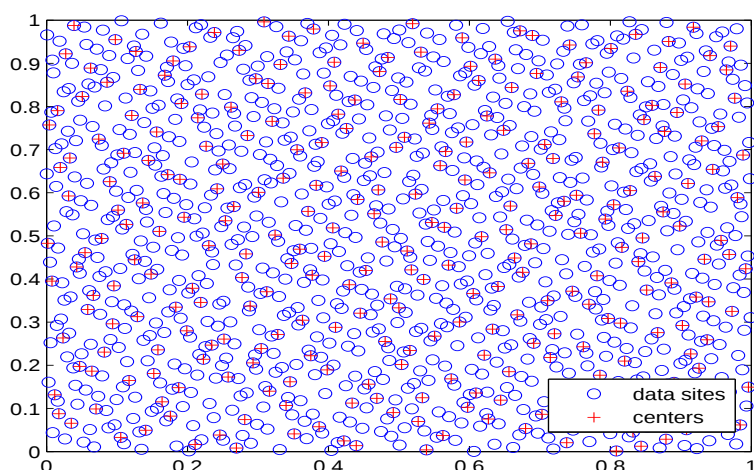


Figura 3.1 $N = 1089$ pika Halton dhe $M = 289$ qendra

përdorur Gaussianët me $\varepsilon = 9.68$ për $N = 1089$ pika nyje me $M = 289$ qendra paraqitet në figurën 3.2. Gjithashtu, disa të dhëna për përafrimin e vlera të ndryshme N pikash nyje dhe M qendrash paraqiten në tabelën 3.1, sëbashku me gabimet e mesatare.

Tabela 3.1 Rezultate të përafrimit 2D me gaussianët për N pika nyje dhe M qendra

N	M	Gab-maks	Gab-mes
81	25	$1.467057e-01$	$7.734880e-01$
289	81	$4.341447e-02$	$5.573211e-01$
1089	289	$6.553595e-03$	$1.897826e-01$
4225	1089	$1.993233e-05$	$7.907976e-04$

Mund të bëhet edhe këtu një anailzë e gabimit për përafrimin stacionar dhe jostacionar e ngjashme me atë të kryer në kapitullin e dytë për interpolimin. Rezultatet eksperimentale janë paraqitur në punimin ([3], Asimi and Hanelli).

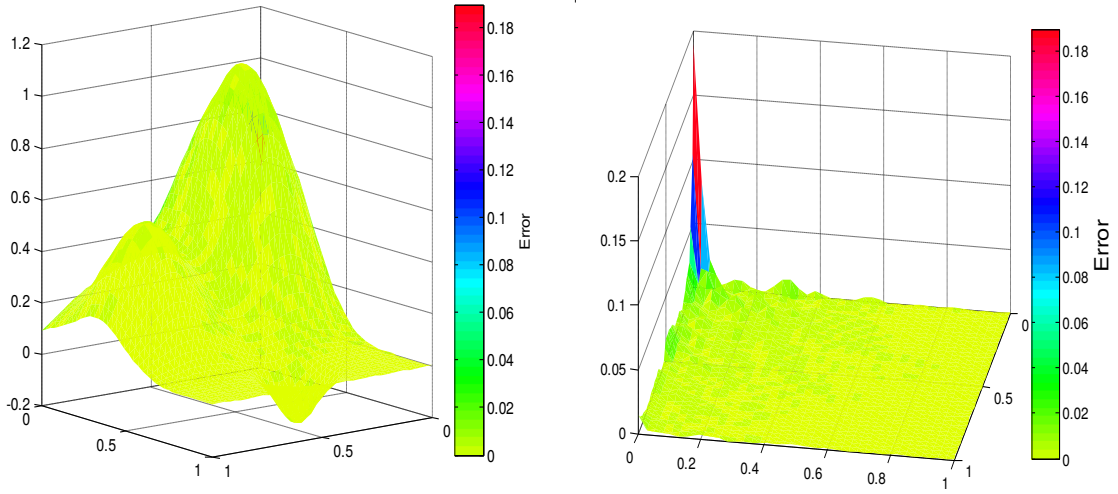


Figura 3.2 **Majtas**: Përafrimi i $N = 1089$ pikave halton dhe $M = 289$ qendrash me anë të gaussianëve me parametrin $\varepsilon = 5$ dhe **Djathtas**: Paraqitja grafike e gabimit të përafrimit

3.3 Koncepte teorike për përafrimin me katrorët më të vegjël.

Rezultatet e paraqitura në këtë pjesë janë bazuar tek punimet e Quak, Sivakumar dhe Ward (shiko [35]; dhe [31]). Në punimin e parë trajtohet përafrimi me katrorët më të vegjël rasti diskret dhe në të dytin ai i rastit të vazhduar. Në të dy punimet, autorët nuk diskutojmë mbi matricën e përafrimit A , por bazojnë rezultatet e tyre në kthyeshmërinë e matricës të fituar nga një sistem normal.

Në rastin diskret ata përdorin prodhimin numerik (3.8), i cili përfshin normën l_2 dhe më pas diskutojnë nëse matrica Gram G që del na sistemi i mëposhtëm i *ekuacioneve normale*, është ose jo singulare

$$Gc = \omega \quad (3.9)$$

ku elementët e saj janë prodhimi numerik l_2 i funksioneve me bazë radiale, p.sh

$$G_{jk} = \langle \Phi(\cdot, \xi_j), \Phi(\cdot, \xi_k) \rangle = \sum_{i=1}^N \Phi(x_i, \xi_j) \Phi(x_i, \xi_k), \quad j, k = 1, \dots, M,$$

dhe termat e lira ω të sistemit (krahu i djathtë) janë

$$\omega_j = \langle \Phi(\cdot, \xi_j), f \rangle = \sum_{i=1}^N \Phi(x_i, \xi_j) f(x_i), \quad j = 1, \dots, M.$$

Mund të vërehet se në rastin kur $M = N$ dhe $\Xi = \chi$ (pra qendrat përputhen me të dhënat) kemi

$$\langle \Phi(\cdot, x_j), \Phi(\cdot, x_k) \rangle = \langle \Phi(\cdot, x_j), \Phi(\cdot, x_k) \rangle_{N_\Phi(\Omega)} = \Phi(x_j, x_k).$$

Pra matrica G në këtë rast është matrica interpoluese A . Nga ky përfundim mund të themi se matrica interpoluese A , është gjithashtu matrica e sistemit të ekuacioneve normale në rastin e përafrimit më të mirë duke u bazuar në normën e hapsirës vetjake (*native norm space*).

Thelbi i rezultatit të punimeve të tyre mund të përshkruhet si në vijim. Në thelb, autorët treguan se *matrica Gramin* për disa funksione me bazë radiale (norma, multiquadrics inverse, gaussianët) nuk është singulare nëse qendrat $\Xi = \xi_k, k = 1, \dots, M$ janë të shpërndara mjaftueshmërisht mirë dhe të dhënat janë grumbulluar "mirë" rreth qendrave, në mënyrë të tillë që diametri i këtij "grumbullimi" të jetë relativisht i vogël në krahasim me distancën e pikave. Figura 3.3 e jep më mirë idenë.

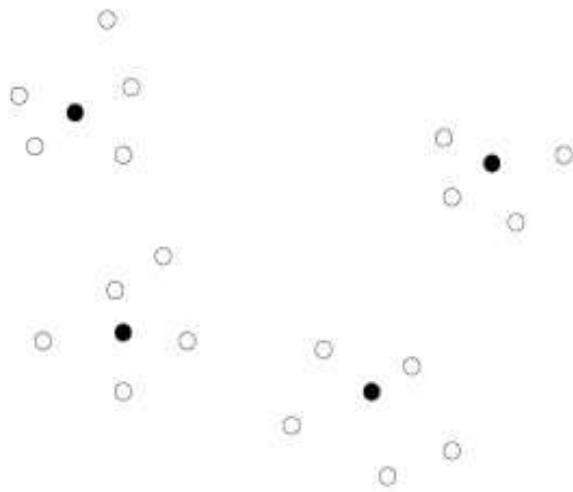


Figura 3.3 Grumbullimi i pikave rreth qendrave të shpërndara mirë

3.4 Katrorët më të vegjël me përshtatje

Në këtë pjesë do të trajtohen dhe krahasohen dy algoritme me përshtatje të katrorëve më të vegjël për problemin e përafrimit të të dhënave jo uniforme. Përkatësisht algoritmi *Shto Nyje* dhe *Largo Nyje*. Në fund do të kryen disa eksperimente numerike për të krahasuar rezultatet e dy algoritmeve.

3.4.1 Algoritmi Shto Nyje

Me qellimin e vetëm për të rritur cilësinë e përafrimit numerik me anë të disa funksioneve baze, një teknikë e përdorur është ajo e rritjes në mënyrë të përshtatshme të numrit të funksioneve bazë. Kjo gjë mund të bëhet duke shtuar me rradhë nyje të reja në model. Kjo përjasje u eksplorua nga [19], për funksionet multiquadrics dhe më pas nga [14] për funksionet me bazë radiale.

Formulojmë problemin: Supozojmë se kemi një grup të dhënash prej N pikash nyje dhe kërkojmë të ndërtojmë një model përafrues me anë të funksioneve me bazë radiale me një tolerancë të dhënë. Fillimisht përafrojmë të dhënat me një model përafrues të bazuar në një numër të vogël nyjesh. Më pasë, evidentohet e dhëna "*më e keqe*" dhe shtohet në model si nyje e re. Kjo procedurë përsëritet derisa të arrihet toleranca e kërkuar.

Algoritmi 1- Shto Nyje

- 1) Jepen të dhënat $\chi = \{x_1, \dots, x_N\}$, f_i , $i = 1, \dots, N$ dhe toleranca tol
- 2) Zgjedhim M nyjet fillestare $\Xi = \{\xi_1, \dots, \xi_M\}$.
- 3) Ndërto modelin përafrues

$$Q_f(x) = \sum_{j=1}^M c_j \Phi(x, \xi_j)$$

- 4) Llogarit gabimin

$$e = \sum_{i=1}^N [f_i - Q_f(x_i)]^2$$

• **Derisa $e > tol$ bëj:**

- 5) Pesho të gjitha të dhënat $x_1, \dots, x_N$ duke u bazuar në komponenten e gabimit

$$\omega_i = |f_i - Q_f(x_i)|, \quad i = 1, \dots, N.$$

6) Përcaktohet e dhëna $x_v \notin \Xi$ "më e keqe" në saj të peshës së vendosur më sipër.

$$\Xi = \Xi \cup \{x_v\}, \quad M = M + 1.$$

7) Rillogaritet modeli dhe gabimi.

Modeli përfundimtar përafrues llogaritet në fund pasi është arritur toleranca e kërkuar. Në secilin hap të algoritmit zgjidhet një sistem ekuacionesh lineare, matrica e të cilëve ka rang të plotë, përderisa nyjet fillestare dhe ato që shtohen janë nga të dhënat. Diçka tjetër me rëndësi për algoritmin është fakti që nuk shtohen dy herë të njëjtat nyje (në të kundërt matrica nuk do të kishte rang të plotë). Implementimi në matlab i këtij algoritmi jepet në programin 3.2.

Programi 3.2 *ShtoNyje2D.m*

```

lineskip
1 % ShtoNyje2D
2 % Ky kod ndderton funksionin përafrues me katrore me te vegjel
3 % me ane te algoritmit Shto Nyje
4
5 tic
6 rbf=@(e,r) exp(-(e*r).^2); ep=5.5;
7
8 % Përcaktohet funksioni Franke
9 f1=@(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
10 f2=@(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
11 f3=@(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
12 f4=@(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
13
14 testfunction=@(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
15 N = 4225; gridtype = 'h';
16 M= 1; % Numri i nyjeve fillestare
17 neval = 40;
18 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
19 epoints = [xe(:) ye(:)];
20 tol= 1e-4; % Toleranca
21
22 % Te dhenat
23 name = sprintf('Data2D_%d%s',N,gridtype);
24 load(name)
25 % Merren M pikat e para si qendra
26 ctrs = dsites(1:M,:);
27 % Llogarit zgjidhjet e sakta
28 exact = testfunction(epoints(:,1),epoints(:,2));
29 % Krijohet ana e djathte e sistemit

```

```

30 rhs = testfunction(dsites(:,1),dsites(:,2));
31 rms_res = 999999;
32 while (rms_res > tol)
33     % Llogarit funksionin perafrues
34     DM_data = DistanceMatrix(dsites, ctrs);
35     CM = rbf(ep,DM_data);
36     coef = CM\rhs;
37     % Llogarit gabimin
38     residual = abs(CM*coef - rhs);
39     [sresidual,idx] = sort(residual);
40     lres = length(residual);
41     rms_res = norm(residual)/sqrt(lres);
42     % Shto Nyje
43     if (rms_res > tol)
44         addpoint = idx(lres); % Kjo eshte pika e s-te e shtuar
45         % Nese eshte perdorur si pike kalo tek tjetra
46         while any(ismember(ctrs,dsites(addpoint,:), 'rows'))
47             lres = lres -1; addpoint = idx(lres);
48         end
49         ctrs = [ctrs; dsites(addpoint,:)];
50     end
51     M=M+1;
52 end
53 % Matrica per llogaritjen pasi eshte arritur toleranca
54 DM_eval = DistanceMatrix(epoints, ctrs);
55 EM = rbf(ep,DM_eval);
56 Pf = EM*coef; % Perafrimi me funksionet me baze radiale
57 maxerr = max(abs(Pf - exact)); rms_err = norm(Pf-exact)/neval;
58 fprintf('Gabimi RMS: %e\n',rms_err)
59 fprintf('Gabimi Maksimal: %e\n',maxerr)
60 fprintf('%d te dhena and %d qendra:\n',N,M)
61 figure; % Paraqitjet grafike
62 plot(dsites(:,1),dsites(:,2), 'bo', ctrs(:,1), ctrs(:,2), 'r+');
63 legend('Data sites', 'Centers')
64 PlotSurf(xe, ye, Pf, neval, exact, maxerr, [160,20]);
65 toc

```

Në këtë program kontrollohet me anë të komandës MATLAB *ismember* (rreshti) nëse një nyje futet disa herë. Kjo, siç thamë, do të sillte që matrica e sistemit të mos kishte rang të plotë.

Programi përdor edhe komandat *tic* dhe *toc* për të llogaritur kohën e ekzekutimit. Vihet re që funksioni përafrues përfundimtar ndërtohet pasi është arritur toleranca e kërkuar (rreshtat 54 deri 56). Gjithashtu, në çdo hap zgjidhet një sistem ekuacionesh lineare, i cili rritet në përmasa në secilin hap. Është treguar (shiko [16] dhe [21]) se toleranca e lejuar arrihet për pika të shtuara, gjë që e bën algoritmin relativisht të shpejtë.

3.4.2 Algoritmi Largo Nyje

Ideja e algoritmit "Largo Nyje" erdhi si pasojë e nevojës për reduktim të dhënash, por mund të përdoret edhe në metodën e përafrimit me përshtatje (shiko [29])

Formulojmë problemin: Zotërojmë një sasi prej N pikash nyje dhe kërkojmë të ndërtojmë një model përafrues me anë të funksioneve me bazë radiale me një tolerancë të dhënë. Fillojmë me një model të mirë, p.sh interpolojmë të dhënat tona. Më pasë, evidentohet e dhëna "më e keqe" dhe largohet nga grupi i nyjeve të përdorura. Kjo procedurë përsëritet derisa të arrihet toleranca e kërkuar. Nyja që largohet, është nyja që ka më pak peshë tek modeli. Algoritmi i mëposhtëm (shiko [14]) jep më mirë idenë:

Algoritmi 2- Largo Nyje

- 1) Jepen të dhënat $\chi = \{x_1, \dots, x_N\}$, f_i , $i = 1, \dots, N$ dhe toleranca tol
- 2) Zgjidh M nyjet fillestare $\Xi = \{\xi_1, \dots, \xi_M\}$.
- 3) Ndërto modelin përafrues të katrorëve më të vegjël

$$Q_f(x) = \sum_{j=1}^M c_j \Phi(x, \xi_j)$$

- 4) Llogarit gabimin

$$e = \sum_{i=1}^N [f_i - Q_f(x_i)]^2$$

• **Derisa $e > tol$ bëj:**

- 5) Pesho të gjitha të dhënat $x_1, \dots, x_N$ duke u bazuar në komponenten e gabimit,

$$\Xi^* = \Xi \setminus \{\xi_j\},$$

dhe llogarit peshat

$$\omega_j = \sum_{i=1}^N [f_i - Q_f^*(x)]^2,$$

ku

$$Q_f^*(x) = \sum_{i=1}^N c_j \Phi(x, \xi_j^*)$$

është përafrimi bazuar në $\Xi^* = \Xi \setminus \{\xi_j\}$,

6) Përcakto dhe largo nyjen $\xi_\mu \notin \Xi$ "më e keqe" me peshë më të vogël ω_μ .

$$\Xi^* = \Xi \setminus \{\xi_j\}, \quad M = M - 1.$$

7) Rillogarit modelin dhe gabimin.

Implementimi në matlab i këtij algoritmi jepet në programin 3.3.

Programi 3.3 *LargoNyje2D.m*

```

lineskip
1 % LargoNyje2D.m
2 % Ky kod ndnderton funksionin perafgues me katroreve me te vegjel
3 % me ane te algoritmit Shto Nyje
4
5 tic
6 rbf = @(e,r) exp(-(e*r).^2); ep = 5.5;
7 % Përcaktohet funksioni Franke
8 f1 = @(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
9 f2 = @(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
10 f3 = @(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
11 f4 = @(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
12 testfunction = @(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
13 N = 4225; gridtype = 'h';
14 M = 4225; % Numri i qendrave fillestare
15 neval=40;
16 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
17 epoints = [xe(:) ye(:)];
18 tol = 0.5; % Toleranca
19 % Merr te dhenat
20 name=sprintf('Data2D_%d%s',N,gridtype); load(name)
21 % M te dhenat e para si qendra
22 ctrs=dsites(1:M,:);
23 % Llogarit zgjidhjet e sakta
24 exact = testfunction(epoints(:,1),epoints(:,2));
25 % Krijohet ana e djathte e sistemit
26 rhs= testfunction(dsites(:,1),dsites(:,2));
27 minres = 0;
28
29 while (minres < tol)
30     % Llogarit funksionin perafgues
31     DM_data = DistanceMatrix(dsites, ctrs);
32     CM=rbf(ep,DM_data);
33     % Llogarit gabimin
34     invCM=pinv(CM); EF= (invCM*rhs)./diag(invCM);
35     residual = abs(EF);

```

```

36     [sresidual,idx] = sort(residual); minres = residual(1);
37     % Largo Nyje
38     if (minres < tol)
39         ctrs = [ctrs(1:idx(1)-1,:); ctrs(idx(1)+1:M,:)];
40         M = M-1;
41     end
42 end
43
44 % Llogarit modelin e perfundimtar perafgues
45 DM_data = DistanceMatrix(dsites, ctrs);
46 CM = rbf(ep, DM_data);
47 DM_eval = DistanceMatrix(epoints, ctrs);
48 EM = rbf(ep, DM_eval);
49 Pf = EM*(CM\rhs);
50 maxerr = max(abs(Pf - exact)); rms_err = norm(Pf-exact)/neval;
51 fprintf('Gabimi RMS:: %e\n', rms_err)
52 fprintf('Gabimi Maks: %e\n', maxerr)
53 fprintf('%d te dhena dhe %d qendra:\n', N, M)
54 figure; % paraqitjet grafike
55 plot(dsites(:,1), dsites(:,2), 'bo', ctrs(:,1), ctrs(:,2), 'r+');
56 caption=sprintf('%d te dhena dhe %d qendra', N, M);
57 title(caption);
58 PlotSurf(xe, ye, Pf, neval, exact, maxerr, [160, 20]);
59 toc

```

Shihet se modeli përfundimtar përafgues llogaritet pasi është arritur toleranca e kërkuar. Edhe në këtë procedurë, në çdo hap zgjidhet një sistem ekuacionesh lineare në secilin hap. Kjo gjë e rrit shumë numrin e veprimeve kur numri i nyjeve filestare është shumë i madh. Matrica e sistemit ashtu si në metodën e parë është jo singulare, përderisa qendrat e përdorura janë nga të dhënat. Në këtë rast nuk është e nevojshme të kontrollohet se mos një nyje hiqet dy herë, ashtu sic bëhej në algoritmin e parë.

3.4.3 Disa eksperimente numerike

Në eksperimentet tona do të përdorim funksionin Franke

$$\begin{aligned}
 f(x,y) = & \frac{3}{4}e^{-1/4((9x-2)^2+(9y-2)^2)} + \frac{3}{4}e^{-1/49((9x+1)^2-1/10(9y+1)^2)} + \\
 & + \frac{1}{2}e^{-1/4((9x-7)^2+(9y-3)^2)} - \frac{1}{5}e^{-((9x-4)^2-(9y-7)^2)}.
 \end{aligned}$$

Modelet përafgues do të ndërtohen mbi një rrjet 40×40 pikash të barazlanguara në katrorin njësi (shiko [2], Asimi). Do të përdoren dy programet e paraqitura *ShtoNyje2D.m* dhe *LargoNyje2D.m* për përafrimin e $N = 289,4225$ pikave Halton. Këto eksperimente do të

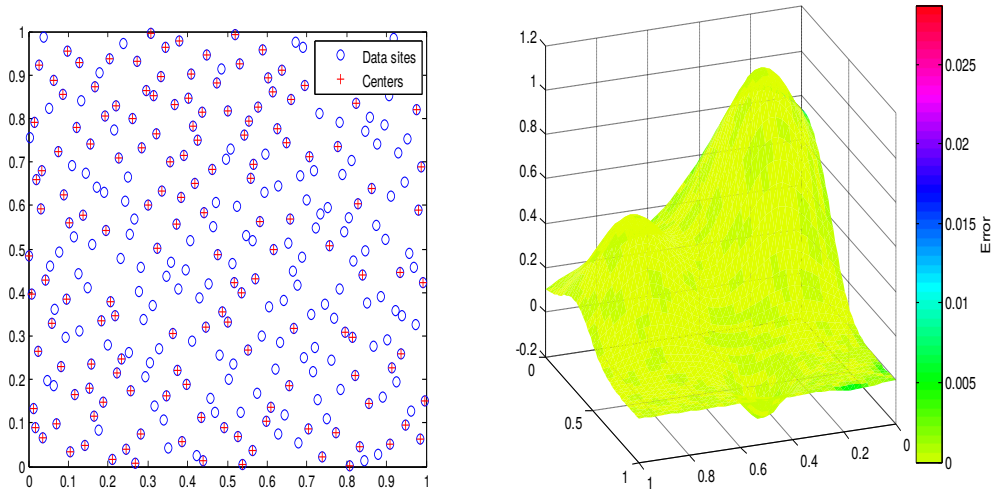


Figura 3.4 **Majtas**: Përafrimi i $N = 289$ pikave halton dhe $M = 155$ qendrash, *algoritmi shto nyje* dhe **Djathtas**: $M = 155$ qendra të shtuara në $N = 289$ pika nyje për të arritur tolerancën $tol = 10^{-4}$

kryhen për të krahasuar dy algoritmet e paraqitura më sipër. Krahassimi do të bëhet me anën e gabimit maksimal dhe atij të normuar.

$$Max_{err} = \|Q_f - f\|_{\infty}$$

$$RMS = \frac{\|Q_f - f\|_{\infty}}{40}$$

ku: Q_f — vlerat e funksionit përafrues; f — vlerat e sakta (funksioni Franke) dhe 40— përmasa e rrjetës së nyjeve.

Eksperimenti i parë:

Ndërtohet funksioni përafrues me funksione baze Gaussianët me $\varepsilon = 5.5$ për përafrimin e $N = 289$ pikave nyje, duke përdorur algoritmin shto nyje. Toleranca e zgjedhur është $tol = 10^{-4}$. Figura 3.4 paraqet përafrimin e funksionit Franke (djathtas) duke filluar me një nyje të rastit në katrorin njësi dhe (majtas) qendrat e përdorura $M = 155$ nga $N = 289$ nyje, për të arritur tolerancën e kërkuar. Gjithashtu do të krahasohet koha e ekzekutimit për secilin rast.

Si eksperiment të dytë, do të përdoren $N = 289$ pika Halton me funksione baze Gaussianët me $\varepsilon = 5.5$ për përafrimin më anë të algoritmit largo një nyje. Do të fillohet me interpolim të të dhënave dhe më pas do të hiqen me radhë nyjet derisa të arrihet toleranca 0.5. Figura 3.5 paraqet përafrimin e funksionit Franke (djathtas) duke filluar me interpolim të dhënash dhe

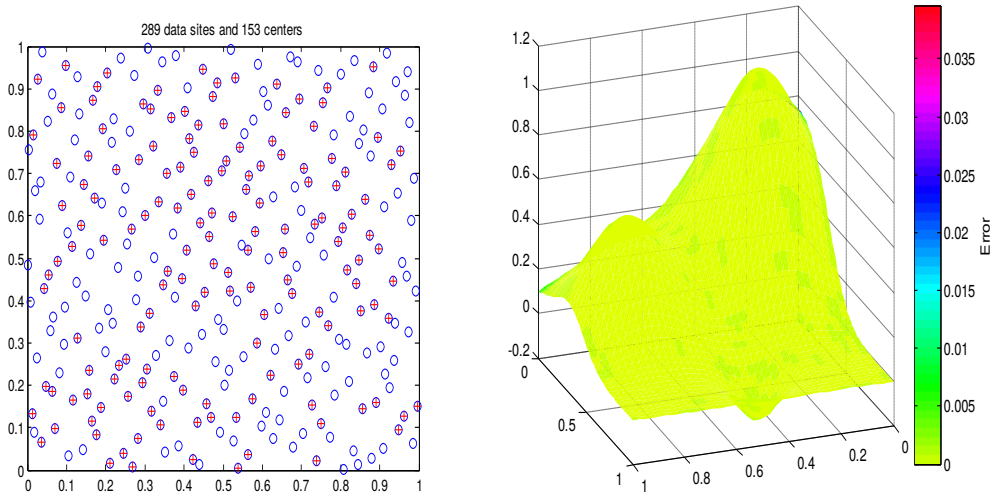


Figura 3.5 **Majtas**: Përafrimi i $N = 289$ pikave halton dhe $M = 153$ qendrash, *algoritmi largo nyje* dhe **Djathtas**: $M = 153$ qendra të shtuara në $N = 289$ pika nyje për të arritur tolerancën $tol = 0.5$

duke hequr me rradhë nyjet, si dhe (majtas) qendrat e përdorura $M = 153$ nga $N = 289$ nyje, për të arritur tolerancën e kërkuar.

Si eksperiment të fundit, ndërtojmë përafrimin e $N = 4225$ pikave Halton, duke përdorur si funksione baze Gaussianët me parametër shkallëzimi $\varepsilon = 5.5$. Do të përdoret algoritmi shto nyje, duke filluar nga një qendër e çfardoshme me tolerancë 10^{-4} . Rezultatet grafike paraqiten në figurën 3.6.

Ajo që vihet re në figurën 3.6, është fakti që nyjet e shtuara (edhe pse ka shumë qendra kandidate për t'u shtuar) shtohen në pozicionet ku funksioni ka maksimume dhe minimume lokale, sëbashku me konturet e zonës. Një përfundim tjetër i rëndësishëm merret nëse shohim saktësinë e përafrimit (shiko figurën 3.2) për rastin e përafrimit të $N = 4225$ me atë të $N = 289$ pikash Halton me anë të algoritmit shto nyje. Në rastin e përafrimit të $N = 4225$ të dhënash saktësia e përafrimit është me e lartë se rasti $N = 289$ edhe pse arrihet për një numër të përafërt nyjesh (përkatesisht, $M = 155$ dhe $M = 164$), por koha e ekzekutimit ndryshon shumë. Megjithatë, kjo kohë ekzekutimi është e përafërt me ekzekutimin e algoritmit largo nyje për rastin $N = 289$ pika nyje. Implementimi i algoritmit përjashtojt një nyje në rastin $N = 4225$ pika, do të ishte jo eficient.

Në dy rreshtat e para të tabelës 3.2 mund të shohim se rendi i gabimeve është i njëjtë për të dy algoritmet për përafrimin e $N = 289$ pikave, por koha e implementimit të metodës largo nyje, është rreth dy herë më e madhe se ekzekutimi i shto një të dhënë. Kjo kohë është e krahasueshme me kohën e ekzekutimit të algoritmit shto nyje për përafrimin e $N = 4225$ nyje.

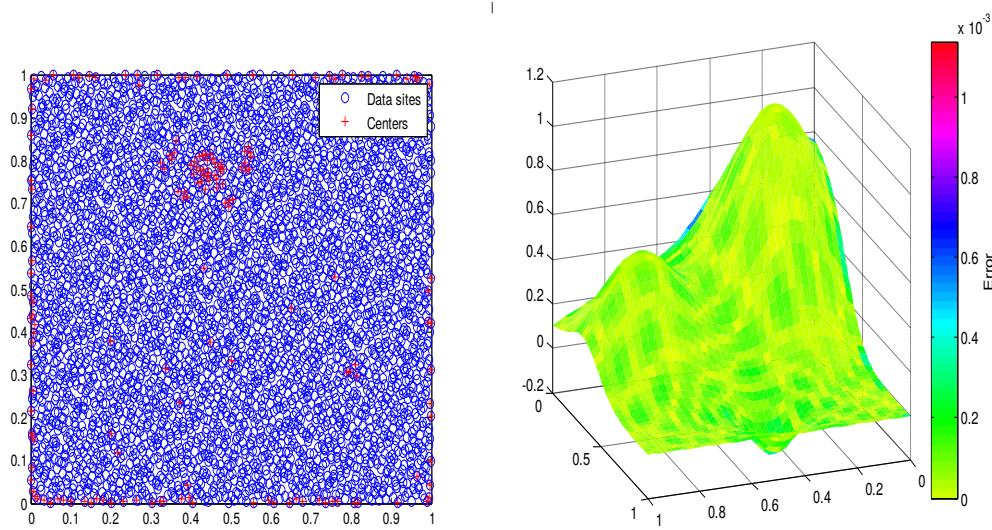


Figura 3.6 **Majtas**: Përafrimi i $N = 4225$ pikave halton dhe $M = 164$ qendrash, *algoritmi largo nyje* dhe **Djathtas**: $M = 164$ qendra të shtuara në $N = 4225$ pika nyje për të arritur tolerancën $tol = 10^{-4}$

Tabela 3.2 Gabimi i normuar dhe gabimi maksimal i përafrimit të N –pika nyje dhe M –qendrave.

<i>Algoritmi</i>	N	M	$Gab - RMS$	$gab - max$	<i>koha</i>
<i>ShtoNyje</i>	289	155	$1.3346E - 03$	$2.8715E - 02$	1.9816s
<i>LargoNyje</i>	289	153	$1.4246E - 03$	$3.9616 - 02$	5.0296s
<i>ShtoNyjet</i>	4225	164	$1.1987E - 04$	$1.1379E - 03$	10.9525s

Theksojmë se implementimi në MATLAB i algoritmit largo nyje duke filluar me interpolimin e $N = 4225$ pikave nyje do të ishte jo eficient për tu ekzekutuar.

3.5 Përafrimi me metodën e katrorëve me të vegjël të lëvizshme

Metoda e katrorëve më të vegjël me lëvizje është një alternativë e interpolimit me funksionet me bazë radiale. Në ndryshim nga metoda e interpolimit me funksionet me bazë radiale, në këtë metodë funksioni përafrues P_f i funksionit të "panjohur" f merret si zgjidhje e shumë sistemeve të vegjël lineare. Pra, jo duke zgjidhur një sistem të vetëm të madh linear ashtu si në metodën e mëparshme.

3.5.1 Formulimi Backus-Gilbert

Konsiderojmë një kuazi-interpolant të formës

$$P_f(x) = \sum_{i=1}^N f(x_i) \Psi_i(x), \quad (3.10)$$

ku $f = [f(x_1), \dots, f(x_N)]^t$ përfaqëson të dhënat. Kuazi-interpolanti që minimizon gabimin, merret në rastin kur *funksionet gjeneruese* Ψ_i janë funksione kardinale, p.sh $\Psi_i(x_j) = \delta_{ij}$, $i, j = 1, \dots, N$.

Në metodën e katrorëve më të vegjël me lëvizje, kërkohen të gjenden vlerat e funksioneve gjenerues $\Psi_i(x) = \Psi(x, x_i)$ duke minimizuar

$$\frac{1}{2} \sum_{i=1}^N \Psi_i^2(x) \frac{1}{W(x, x_i)} \quad (3.11)$$

subjekt i kushteve të ndërtimit polinomial (*Polynomial Reproduction*).

$$\sum_{i=1}^N p(x_i) \Psi_i(x) = p(x), \quad \forall p \in \Pi_d^s, \quad (3.12)$$

ku Π_d^s është hapsira e polinomeve s -dimensional me rend të shumtën d , e cila ka dimension $\binom{s+d}{d}$.

Funksionet peshuese $W(\cdot, x_i)$ zvogëlohen me rritjen e largesës nga origjina. Funksionet me bazë radiale mund të përdoren si funksione peshues. Megjithatë, kushti i të qënurit rigoroz pozitivisht i përcaktuar nuk është i nevojshëm, për pasojë si funksione peshuese mund të përdoren edhe *B-splinet*. Kushtet e ndërtimit polinomial shtohen në mënyrë që kuazi-interpolanti të arrijë një rend përafrimi të kënaqshëm.

Funksioni sheshues 3.11 i përcaktuar mësipër ka përdorim praktik. Në teorinë e formulimit Backus-Gilbert, e cila është zhvilluar në këndvështrimin gjeofizik (shiko [6]), kërkohet që funksionet gjenerues Ψ_i të jenë sa më afër të jetë e mundur me funksionet kardinale ideale (p.sh funksionet delta). Kushtet e ndërtimit polinomial korrespondojnë me *kushte diskrete momenti* për funksionin $\Psi = \Psi(x, \cdot)$.

Nëse merret x si një pikë (llogaritjeje) e fiksuar, atëherë shtohet një tjetër problem kuadratik minimizimi i kushtëzuar. Të panjohurat gjenden në "vektorin me koeficient"

$\Psi(x) = [\Psi(x, x_1), \dots, \Psi(x, x_N)]^T$. Funkzionali sheshues 3.11

$$\frac{1}{2} \Psi(x)^T Q(x) \Psi(x)$$

jepet me anë të matricës diagonale

$$Q(x) = \text{diag} \left(\frac{1}{(x, x_1)}, \dots, \frac{1}{(x, x_N)} \right), \quad (3.13)$$

ku $W(\cdot, x_i)$ janë funksione peshe pozitive (që këtej për çdo x matrica $Q(x)$ është pozitivisht e përcaktuar).

Kushtet e ndërtimit polinomial 3.12 mund të shkruhen në trajtë matricore

$$A\Psi(x) = p(x),$$

ku A është një matricë $m \times N$ me elementë $A_{ij} = p_j(x_i)$, $i = 1, \dots, N$, $j = 1, \dots, m$, dhe $p = [p_1, \dots, p_m]^T$ është një vektor që përmban një bazë për hapësirën Π_d^s të polinomeve me fuqi d .

Nëse i rikthehemi edhe njëherë shumëzuesve Lagranzh (barazimet [(3.4), Fq.40] dhe [(3.5), Fq.40]) mund të shkruajmë

$$\lambda(x) = (AQ^{-1}(x)A^T)^{-1} p(x) \quad (3.14)$$

$$\Psi(x) = Q^{-1}(x)A^T \lambda(x). \quad (3.15)$$

Nga ekuacionit 3.14 rrjedh se shumëzuesit Lagranzh përftohen si zgjidhje e një sistemi Gram

$$G(x)\lambda(x) = p(x),$$

ku elementët e G janë prodhime numerike l_2 të peshuar

$$G_{jk}(x) = \langle p_j, p_k \rangle_{w(x)} = \sum_{i=1}^N p_j(x_i) p_k(x_i) w(x, x_i), \quad j, k = 1, \dots, m. \quad (3.16)$$

I veçantë këtu është fakti që peshat variojnë me pikën e llogaritjes x .

Ka vend për dy komente. Së pari, matrica Gram është simetrike dhe pozitivisht e përcaktuar përderisa bazat polinomiale janë linearisht të pavarura dhe peshat pozitive. Së dyti, në praktikë polinomet do të jenë të zhvendosur, p.sh të qendëruar në pikën e llogaritjes x , në mënyrë që $p_1(x) \equiv 1 \neq 0$.

Ekuacioni 3.15 mund të shkruhet në trajtën

$$\Psi_i(x) = W(x, x_i) \sum_{j=1}^N \lambda_j(x) p_j(x_i), \quad i = 1, \dots, N. \quad (3.17)$$

Ekuacioni i mësipërm përcakton funksionet gjeneruese të funksionit përafrues. Pra, pasi llogariten vlerat e shumzuesve Lagranzh $\lambda_j(x)$, $j = 1, \dots, N$, mund të përcaktojmë një formulë eksplicite për vlerat e funksioneve gjenerues. Në përgjithësi, llogaritja e shumzuesve Lagranzh kërkon zgjidhjen e një sistemi relativisht të vogël, i cili ndryshon me ndryshimin e pikës llogaritëse x .

3.5.2 Implementimi Standart i Metodës së Katrorëve më të Vegjël të Lëvizshme

Konsiderojmë problemin e mëposhtëm të përafrimit. Supozojmë se na jepen vlerat $f(x_i)$, $i = 1, \dots, N$, në një bashkësi pikash të ndryshme $\chi = \{x_1, \dots, x_N\} \subset \mathbb{R}^s$, në një hapsirë përafrimi $U = \text{span}\{u_1, \dots, u_n\}$ (me $m < N$) sëbashku me prodhimin e brendshëm të peshuar l_2

$$\langle f, g \rangle_{W(x)} = \sum_{i=1}^N f(x_i) g(x_i) W_i(x), \quad x \in \mathbb{R}^s \text{ i fiksuar.} \quad (3.18)$$

Peshat pozitive W_i , $i = 1, \dots, N$ varen nga pikat e llogaritjes x . Do të interpretojmë funksionet peshë në mënyrë të ngjashme me funksionet me bazë radiale, p.sh $W_i(x) = W(x, x_i)$, ku pika x_i është pikë e bashkësisë χ .

Kështu, kërkohet të gjendet përafrimi më i mirë nga U i f -së në pikën x bazuar në normën e përkthuar në barazimin (3.18). Kjo do të thotë se funksioni përafrues, në pikën x do të ketë pamjen

$$P_f(x) = \sum_{j=1}^m c_j(x) u_j(x), \quad (3.19)$$

ku koeficientët $c_j(x)$ janë të tillë që të minimizohet

$$\sum_{i=1}^N [P_f(x_i) - f(x_i)]^2 W_i(x) \quad (3.20)$$

Bazuar në përcaktimin e prodhimit të brendshëm (3.18), funksionet peshuese të së cilit "lëvizin" me ndryshimin e pikës së llogaritjes x , edhe koeficientët c_j në barazimin (3.19) varen gjithashtu nga pika x . Për rrjedhojë, duhen zgjidhur ekuacionet normale

$$\sum_{j=1}^m c_j(x) \langle u_j, u_k \rangle_{W(x)} = \langle f, u_k \rangle_{W(x)}, \quad k = 1, \dots, m, \quad (3.21)$$

çdo herë që pika x ndryshon. Në trajtë matricore barazimi (3.21) shkruhet

$$G(x)c(x) = f_u(x), \quad (3.22)$$

ku matrica Gram pozitive $G(x) = \left(\langle u_j, u_k \rangle_{W(x)} \right)_{j,k=1}^m$, vektori me koeficientët c_j dhe vektori i krahut të djathtë $f_u(x)$ varen nga x .

Zakonisht, në metodën e katrorëve më të vegjël të lëvizshëm hapsira U merret ajo e polinomeve, p.sh

$$P_f(x) = \sum_{j=1}^m c_j(x) u_j(x), \quad x \in \mathbb{R}^s \quad (3.23)$$

ku $\{p_1, \dots, p_m\}$ është bazë e hapsirës Π_d^s të polinomeve s -përmasore të rendit d .

Atëhere, sistemi Gram (3.22) bëhet tani

$$G(x)c(x) = f_p(x), \quad (3.24)$$

ku matrica Gram $G(x)$ ka si elementë

$$G_{j,k}(x) = \langle p_j, p_k \rangle_{W(x)} = \sum_{i=1}^N p_j(x_i) p_k(x_i) (x, x_i), \quad j, k = 1, \dots, m \quad (3.25)$$

dhe vektori i krahut të djathtë konsiston në projeksionet e të dhënave f në funksionet e bazës, p.sh

$$f_p(x) = [\langle f, p_1 \rangle_{W(x)}, \dots, \langle f, p_m \rangle_{W(x)}]^T.$$

Tregohet se dy formulimet e trajtuara për metodën e katrorëve më të vegjël të lëvizshme janë ekuivalente (shiko [15]). Gjithashtu mund të përcaktohen disa veti të kësaj metode:

- Riprodhon çdo polinom me rend të shumtën d me ekzaktësisht s variabla.
- Prodhon modelin lokal të peshuar më të mirë përafrues.

- Në këndvështrimin e një kuazi-interpolanti, funksionet gjeneruese Ψ_i , janë afër sa më shumë të jetë e mundur me funksionet bazë cardinale optimale në sensin e minimizimit të funksionit (3.11).
- Derisa polinomet janë pafundësisht të lëmuar, të dyja përcaktimet e P_f tregojnë se lëmueshmëria e tij përcaktohet nga lëmueshmëria e funksioneve peshues $W_i(x) = W(x, x_i)$.

3.5.3 Metoda Shepard

Metoda e katrorëve më të vegjël e lëvizshme në rastin kur $m = 1$ me $p_1(x) \equiv 1$ njihet me emrin Metoda Shepard. Duke përdorur simbolikën tonë mund të shkruajmë

$$P_f(x) = c_1(x).$$

Matrica Gram në këtë rast do të përbëhet vetëm nga një element

$$G(x) = \langle p_1, p_1 \rangle_{W(x)} = \sum_{i=1}^N W(x, x_i)$$

kështu që $G(x)c(x) = f_p(x)$ sjell

$$c_1(x) = \frac{\sum_{i=1}^N f(x_i) W(x, x_i)}{\sum_{i=1}^N W(x, x_i)}.$$

Vlerat e shumëzuesve Lakranzh mund të gjenden nga barazimi $G(x)\lambda(x) = p(x)$ kështu që

$$\lambda_1(x) = \frac{1}{\sum_{i=1}^N W(x, x_i)}.$$

Përfundimisht, funksionet gjeneruese përcaktohen si

$$\Psi_i(x) = W(x, x_i)\lambda_1(x)p_1(x_i) = \frac{W(x, x_i)}{\sum_{i=1}^N W(x, x_i)}.$$

Që ketej mund të shkruhet formula kuazi–interpoluese e Metodës Shepard

$$P_f(x) = \sum_{i=1}^N f(x_i)\Psi_i(x) = \sum_{i=1}^N f(x_i) \frac{W(x, x_i)}{\sum_{k=1}^N W(x, x_k)}. \quad (3.26)$$

Interesant është fakti që funksionet polinomiale bazë dhe shumëzuesit Lagranzh janë ortogonal në të dhënat, p.sh $\langle \lambda_1, p_1 \rangle_{W(x)} = 1$. Me të vërtetë

$$\begin{aligned} \langle \lambda_1, p_1 \rangle_{W(x)} &= \sum_{i=1}^m \lambda_1(x_i) W(x, x_i) = \\ &= \sum_{i=1}^N \frac{W(x, x_i)}{\sum_{k=1}^N W(x_i, x_k)}, \end{aligned}$$

ku shprehja e fundit është e barabartë me 1 nëse x —i është një element i bashkësisë χ .

Implementimi i metodës Shepard në MATLAB jepet në programin 3.4. Si funksione testuese (rreshtat 4 – 9) merret funksioni Franke. Nëse duam që të dhënat tona të jenë të përziera me "zhurmë" pas rreshtit 16 shkruajmë komandën matlab

f=f+0.03*rand(size(f));

që bën të mundur shtimin e një numri të rastit në secilën nga të dhënat. Në pjesën e sipërme të figurës 3.7 paraqitet përafrimi me metodën Shepard sëbashku me grafikun e gabimeve për $N = 1089$ me parametër shkallëzimi $\varepsilon = 48$ ku gabimi i mesatarizuar është **1.809402e-02**. Në pjesën e poshtme paraqitet përafrimi me metodën Shepard sëbashku me grafikun e gabimeve për $N = 1089$ me parametër shkallëzimi $\varepsilon = 12$ ku gabimi i mesatarizuar është **3.100888e-02**. Vihet re që ndryshimi i parametrut të shkallëzimit ε nga 48 në 12 ndikion në lëmueshmërinë e funksionit përafrues edhe pse gabimi i mesatarizuar i përafrimit në rastin e dytë është më i madh.

Programi 3.4 *MetShepard2D.m*

```

lineskip
1 % MetShepard2D
2 % Skript qe nderton modelin përafrues me funksione peshuese gaussianet
3 rbf = @(e,r) exp(-(e*r).^2); ep = 67.34;
4 % Funksioni Franke si funksion testues
5 f1 = @(x,y) 0.75*exp(-((9*x-2).^2+(9*y-2).^2)/4);
6 f2 = @(x,y) 0.75*exp(-((9*x+1).^2/49+(9*y+1).^2/10));
7 f3 = @(x,y) 0.5*exp(-((9*x-7).^2+(9*y-3).^2)/4);
8 f4 = @(x,y) 0.2*exp(-((9*x-4).^2+(9*y-7).^2));
9 testfunction = @(x,y) f1(x,y)+f2(x,y)+f3(x,y)-f4(x,y);
10 N = 4225; gridtype = 'h';
11 neval = 40;
12 % Therrit te dhenat
13 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
14 ctrs = dsites;
```

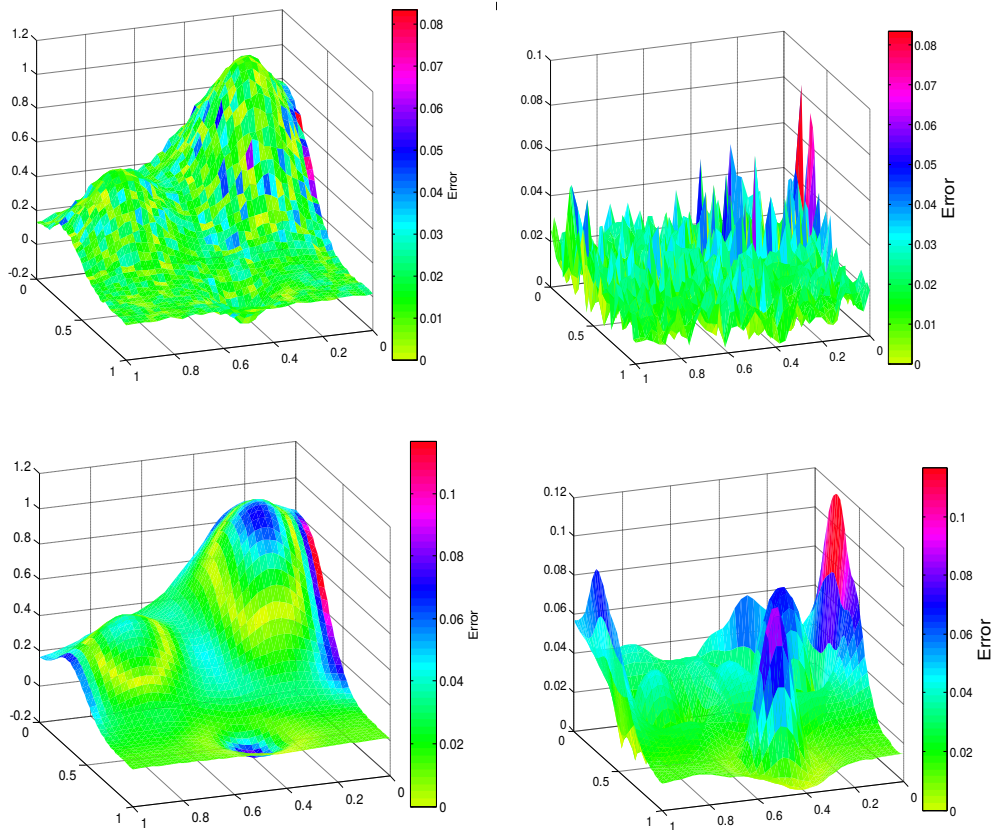


Figura 3.7 **Lart Majtas**: Përafrimi me metodën Shepard për $N = 1089$ të dhëna me "zhurmë" me parametër shkallëzimi $\varepsilon = 48$, **Lart Djathtas**, Gabimi i mesatarizuar dhe **Poshtë Majtas**: Përafrimi me metodën Shepard $N = 1089$ të dhëna me "zhurmë" me parametër shkallëzimi $\varepsilon = 12$, **Poshtë Djathtas**: Grafiku i gabimeve të mesatarizuar

```

15 % Krijo vektorin e te dhenave f
16 f = testfunction(dsites(:,1),dsites(:,2));
17 % Krijohet rrjeta e pikave ku do te vleresohet modeli
18 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
19 epoints = [xe(:) ye(:)];
20 % Matrica e distancave midis pikave nyje dge qendrave
21 DM_eval = DistanceMatrix(epoints, ctrs);
22 % Matrica Llogaritjes
23 EM = rbf(ep, DM_eval);
24 EM = EM./ repmat(EM*ones(N,1), 1, N); % Normalizimi Shepard
25 % Llogaritet kuazi-interpolanti
26 Pf = EM*f;
27 % Zgjidhjet e sakta
28 exact = testfunction(epoints(:,1),epoints(:,2));
29 % Llogariten Gabimet
30 maxerr = norm(Pf-exact, inf);
31 rms_err = norm(Pf-exact)/neval;

```

```

32  fprintf('Gabimi Mes:      %e\n', rms_err)
33  fprintf('Gabimi max: %e\n', maxerr)
34  fview = [160,20];
35  PlotSurf(xe, ye, Pf, neval, exact, maxerr, fview);
36  PlotError2D(xe, ye, Pf, exact, maxerr, neval, fview)

```

Disa rezultate për përafrimin me metodën Shepard për N — pika nyje të ndryshme në katorin njësi $[0, 1]^T$ paraqiten në tabelën 3.3, sëbashku me vlerat e gabimit të mesatarizuar dhe gabimit makismal. Vlerat e parametrut të shkallëzimit janë marrë bazuar në dendësinë e pikave. Për vlerat më të mëdha të N —së merren vlera të mëdha të parametrut të shkallëzimit ε dhe anasjelltas. Mund të bëhet një analizë e ngjashme me atë të kryer në kapitullin e dytë për interpolimin stacionar (Në këtë mënyrë janë përcaktuar edhe këtu vlerat optimale të parametrut ε)

Tabela 3.3 Rezultate të përafrimit 2D me funksione peshuese gaussianë për N pika nyje

N	eps	Gab-maks	Gab-mes
9	7.06	$1.380241e-01$	$4.788692e-01$
25	7.06	$6.822044e-02$	$2.110542e-01$
289	19.18	$1.670186e-02$	$1.037791e-01$
1089	37.74	$7.508205e-03$	$6.437136e-02$
4225	67.34	$3.677352e-04$	$3.744388e-02$

3.5.4 Përafrimi me metodat e rindërtimit polinomial

Pamë se duke zgjidhur sistemet Gram mund të përcaktohen formula implicite për të llogaritur funksionet gjeneruese Ψ_i , $i = 1, \dots, N$. Një nga këto metoda ishte metoda Shepard e prezantuar më sipër, e cila është e vlefshme pavarësisht dimensioneve s të hapsirës. Megjithatë, nëse fuqia d e rindërtimit polinomial (*ang.* Polynomial Reproduction) është jotriviale (dmth $d > 0$), atëherë përmasa e sistemit Gram është $m = \binom{s+d}{d}$ dhe për pasojë formulat për llogaritjen e funksioneve gjenerues do të varen nga dimensionin s të hapsirës.

Por, shpesh me interes është përcaktimi i disa formulave eksplicite për llogaritjen e funksioneve gjenerues Ψ_i . Në këtë pjesë do të prezantohen tre formula eksplicite për funksionet gjenerues të metodës së katrorëve më të vegjël me zhvendosje. Në formulimet e mëposhtme do të përdoren *momentet* e përcaktuara sipas

$$\mu_\alpha = \sum_{i=1}^N (x_i - x)^\alpha W(x_i, x), \quad x \in \mathbb{R}^s, \quad |\alpha| \leq 2d,$$

me α si indeks. Këto momente shërbejnë si elemente të matricës Gram $G(y)$, nëse si baza përdoren monome dhe me y referohen vlerat e llogaritjes x .

Shembull 3.1

Nëse marrim $s = 1$ dhe $d = 1$, atëherë $m = 2$. Baza e të dhënave jepet si $\chi = \{x_1, \dots, x_N\}$. Zgjedhim bazën standarte

$$U = \text{span}\{p_1(x) = 1, p_2(x) = x\}.$$

Atëherë, matrica Gram do të jetë

$$\begin{aligned} G(x) &= \begin{bmatrix} \langle p_1(\cdot - x), p_1(\cdot - x) \rangle_{W(x)} & \langle p_1(\cdot - x), p_2(\cdot - x) \rangle_{W(x)} \\ \langle p_2(\cdot - x), p_1(\cdot - x) \rangle_{W(x)} & \langle p_2(\cdot - x), p_2(\cdot - x) \rangle_{W(x)} \end{bmatrix} = \\ &= \begin{bmatrix} \sum_{i=1}^N W(x_i, x) & \sum_{i=1}^N (x_i - x) W(x_i - x) \\ \sum_{i=1}^N (x_i - x) W(x_i, x) & \sum_{i=1}^N (x_i - x)^2 W(x_i, x) \end{bmatrix} = \\ &= \begin{bmatrix} \mu_0 & \mu_1 \\ \mu_1 & \mu_2 \end{bmatrix}, \end{aligned}$$

ndërsa krahu i djathtë i sistemit Gram është

$$p(x - x) = \begin{bmatrix} p_1(0) \\ p_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Përdorim metodën Kramer mbi sistemin Gram të mësipërm dhe marrim zgjidhjen

$$\begin{aligned} \lambda_1(x, x) &= \frac{\mu_2}{\mu_0\mu_2 - \mu_1^2} \\ \lambda_2(x, x) &= \frac{-\mu_1}{\mu_0\mu_2 - \mu_1^2} \end{aligned}$$

Bazuar tani në ekuacionin (3.17) të formulimit Backus–Gilbert mund të shkruajmë formulën eksplicite për llogaritjen e funksioneve gjenerues të kuazi interpolantit

$$\Psi_i(x, x) = W(x_i, x) [\lambda_1(x, x) + \lambda_2(x, x)(x_i - x)], \quad i = 1, \dots, N,$$

ku funksionet $W(x_i, \cdot)$ janë funksione peshuese (pozitive) arbitrare.

Shembull 3.2

Vazhdojmë të qendrojmë në rastin $s = 1$, por tani rritim rendin $d = 2$ të rindërtimit polinomial dhe për pasojë $m = 3$. Si bazë marrim bazën standarte

$$U = \text{span}\{p_1(x) = 1, p_2(x) = x, p_3(x) = x^2\}.$$

Sistemi Gram në këtë rast do të jetë

$$\begin{bmatrix} \mu_0 & \mu_1 & \mu_2 \\ \mu_1 & \mu_2 & \mu_3 \\ \mu_2 & \mu_3 & \mu_4 \end{bmatrix} \begin{bmatrix} \lambda_1(x, x) \\ \lambda_2(x, x) \\ \lambda_3(x, x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Përdorim metodën Kramer mbi sistemin Gram të mësipërm dhe marrim zgjidhjen

$$\begin{aligned} \lambda_1(x, x) &= \frac{\mu_2\mu_4 - \mu_3^2}{|G|} \\ \lambda_2(x, x) &= \frac{\mu_2\mu_3 - \mu_1\mu_4}{|G|} \\ \lambda_3(x, x) &= \frac{\mu_1\mu_3 - \mu_2^2}{|G|} \end{aligned}$$

ku $|G| = 2\mu_1\mu_2\mu_3 - \mu_0\mu_3^2 - \mu_2^3 - \mu_1^2\mu_4 + \mu_0\mu_2\mu_4$. Që këtej funksionet gjenerues jepen nga

$$\Psi_i(x, x) = W(x_i, x) [\lambda_1(x, x) + \lambda_2(x, x)(x_i - x) + \lambda_3(x, x)(x_i - x)^2], \quad i = 1, \dots, N.$$

Shembull 3.3

Rindërtimi polinomial në 2D, çon gjithashtu në një sistem Gram 3×3 (përderisa $s = 2$, $d = 1$ dhe $m = 3$). Si bazë në këtë rastë do të jetë

$$U = \text{span}\{p_1(x, y) = 1, p_2(x, y) = x, p_3(x, y) = y\}.$$

ku $\mathbf{x} = (x, y) \in \mathbb{R}^2$ dhe sistemi Gram do të jetë

$$\begin{bmatrix} \mu_{00} & \mu_{10} & \mu_{01} \\ \mu_{10} & \mu_{20} & \mu_{11} \\ \mu_{01} & \mu_{11} & \mu_{02} \end{bmatrix} \begin{bmatrix} \lambda_1(\mathbf{x}, \mathbf{x}) \\ \lambda_2(\mathbf{x}, \mathbf{x}) \\ \lambda_3(\mathbf{x}, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Sumëzuesit Lagranzh në këtë rast do të jenë

$$\begin{aligned}\lambda_1(\mathbf{x}, \mathbf{x}) &= \frac{\mu_{11}^2 - \mu_{20}\mu_{02}}{|G|} \\ \lambda_2(\mathbf{x}, \mathbf{x}) &= \frac{\mu_{10}\mu_{02} - \mu_{01}\mu_{11}}{|G|} \\ \lambda_3(\mathbf{x}, \mathbf{x}) &= \frac{\mu_{20}\mu_{01} - \mu_{10}\mu_{11}}{|G|}\end{aligned}$$

ku $|G| = \mu_{10}^2\mu_{02} + \mu_{20}\mu_{01}^2 - \mu_{00}\mu_{20}\mu_{02} - 2\mu_{10}\mu_{01}\mu_{11} + \mu_{00}\mu_{11}^2$. Funksionet gjenerues $\Psi_i, i = 1, \dots, N$ jepen nga

$$\Psi_i(\mathbf{x}, \mathbf{x}) = W(\mathbf{x}_i, \mathbf{x}) [\lambda_1(\mathbf{x}, \mathbf{x}) + \lambda_2(\mathbf{x}, \mathbf{x})(x_i - x) + \lambda_3(\mathbf{x}, \mathbf{x})(y_i - y)].$$

Perdorimi i formulave eksplicite për llogaritjen e funksioneve gjenerues dhe shumëzuesve Lagranzh jepet në Aneksin B si programi **LinearScaling2D_CS.m** [(4.8), Fq.96], i cili mund të shkarkohet në <http://www.math.unipd.it/~demarchi/TAA2010/>

3.5.5 Një rast zbatimi

Në këtë pjesë do të prezantohet një zbatim i metodës së katrorëve më të vegjël të lëvizshme në një bazë të dhënash të përbërë prej $601 \times 601 = 361201$ matjesh. Matjet konsistojnë në lartësitë gjeografike të zonës verilindore Iowa dhe janë në formatin DEM (ang. Digital Elevation Model) në shkallë 1 : 250000. Ato mund të shkarkohen nga <http://edcs9.cr.usgs.gov>. Vlerat e lartësive variojnë nga 178 metra në 426 metra.

Për të konvertuar të dhënat DEM është përdorur utiliteti *DEM2XYZN*, i cili mund të shkarkohet në <http://data.geocomm.com/dem/dem2xyzn/>. Të dhënat e përfutuara ruhen në një format *VeriLindje_Iowa2D.dat* që mund të thërren në Matlab. Në programin **MetKatVegjLev.m** të prezantuar si programi 3.5, jepet një kod i cili thërret programin [(4.8), Fq.96], **LinearScaling2D_CS.m** që jepet në Aneksin e këtij punimi.

Programi 3.5 *MetKatVegjLev.m.m*

```
lineskip
1 % MetKatVegjLev
2 % Perdoren matricat e rralla
3 % Therret: LinearScaling2D_CS
4
5
6 % Percaktojme Gaussianin
7 rbf = @(e,r) exp(-(e*r).^2);
```

```

8   ep = 70; npik = 100;
9
10  % Merren te dhenat
11  load('VeriLindje_Iowa2D');
12  ctrs = dsites; % Qendrat perputhen me te dhenat
13  % Krijohet rrejtja me pikat ku do te kryhen perafimet
14  grid = linspace(0,1,npik); [xe,ye] = meshgrid(grid);
15  epoints = [xe(:) ye(:)]; % pikat llogaritese
16
17  % Therritet funksioni LinearScaling2D_CS
18  EM = LinearScaling2D_CS(epoints, ctrs, rbf, ep);
19  % Përafrimi me Met e Katroreve me te Vegjel te Levizshme
20  Pf = EM*rhs;
21  figure % Paraqitjet grafike
22  surfplot = surf(xe,ye, reshape(Pf,npik,npik));

```

Thirrja e të dhënave bëhet në rreshtin 11. Matjet janë të ndara në dy pjesë: *dsite* mban koordinatat x,y dhe *rhs* matjet në këto pika. Në rreshtat 7 – 8 përcaktohet Gausiani si funksione me bazë radiale me parametër shkallëzimi $\varepsilon = 70$. Kjo vlerë e madhe e tij merret për arsye të dëndësisë së pikave (shiko kapitullin 2). Për të kryer llogaritjet është marrë rrjeta me 100×100 nyje (rreshtat 13 – 15).

Në rreshtin 18 të programit, thirret funksioni ***LinearScaling2D_CS***. Në ndërtimin e tij janë implementuar funksionet *kdtree.m* dhe *kdrangequery.m*. Këto funksione të ndërtuara nga Guy Shechter mund të shkarkohen nga *Matlab Central File Exchange* dhe janë të nevojshme në rastin e punimit më shumë të madh të dhënash. Qellimi i një *kd-tree* është dekompozimi i një bashkësie prej N pikash në $\mathbb{R}^n$ në një grup relativisht të vogël nënbashkësishë, të tilla që çdo nënbashkësi të ketë numër të përafërt pikash. Përpara se të mund të përdoren këto funksione duhet të kompilohen njëherë me anë të kompiluesit *mex* në Matlab. Kjo bëhet e mundur me anë të komandave

```

» cd kdtree/lib
» edit ../src/kdtree_common.h
» mex ../src/kdtree.cc
» mex ../src/kdtreeidx.cc
» mex ../src/kdrangequery.cc

```

Kuptohet se, në secilën komandë duhet të përcaktohet path-i ku janë vendosur funksionet në Matlab. Më pas funksionet e kompiluara merren dhe ruhen në direktorinë aktive të Matlab. Më pas mund të ekzekutohet programi 3.5. Rezultatet e përafrimit për dy rrjeta pikash 60×60 dhe 100×100 paraqiten në figurën 3.8. Koha e manipulimit të këtyre të dhënave bazuar në komandat Matlab *tic* dhe *toc* është 8.169403sek.

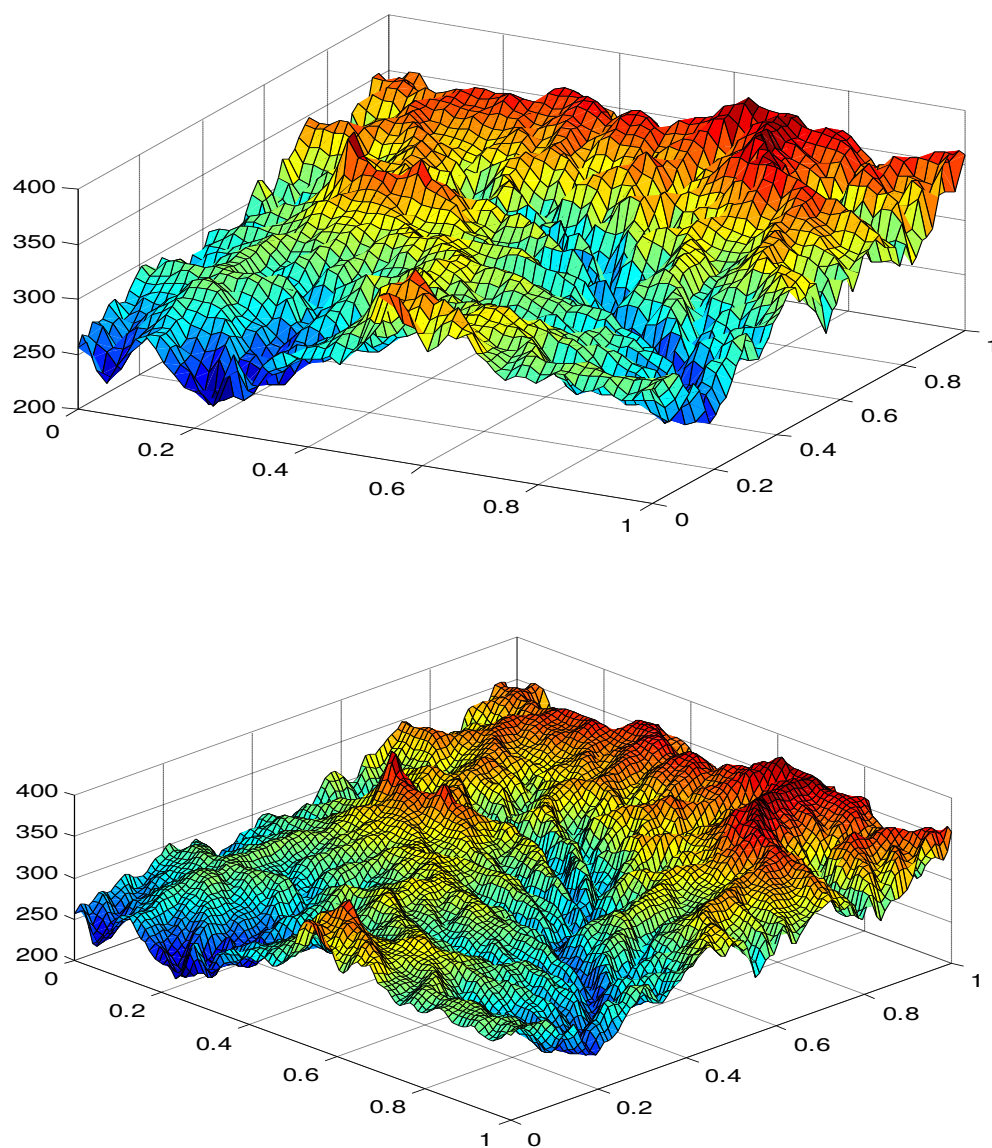


Figura 3.8 **Lart**: Përafrimi në rrjetën 60×60 bazuar në 361201 pika nyje dhe **Poshtë**: përafrimi në rrjetën 100×100 bazuar në 361201 pika nyje. Metoda e katrorëve më të vegjël me lëvizje

Kapitulli 4

Zbatime të funksioneve me bazë radiale në metodat pseudospektrale

Metodat pseudospektrale (PS) njihen si metoda shumë të sakta për zgjidhjen e ekuacioneve diferenciale me derivate të pjesshme (EDP)(shiko [37]). Ideja qendron në përdorimin e funksioneve shumë të lëmuar baze, siç janë polinomet, për përcaktimin e zgjidhjes së përafërt të EDP, sipas:

$$\hat{u}(x) = \sum_{j=1}^N c_j B_j(x), \quad x \in \mathbb{R}. \quad (4.1)$$

Fillimisht, nuk do të konsiderojmë variablin kohë, i cili mund të jetë pjesë e formulës për llogaritjen e $\hat{u}$. Më pas, do të përdorim procedurë për të marrë parasysh edhe pjesën (shtrirjen) kohore të zgjidhjes. Gjithashtu, meqë metodat PS janë ndërtuar për rastet njëpërmasore, fillimisht do të merremi me funksionet me një variabël, derisa të përgjithësohet problemi me ndihmën e funksioneve me bazë radiale.

Një nga mënyrat e implementimit të metodave PS, është përdorimi i *matricave të diferencimit*, dmth gjetja e një matrice D , të tillë që në pikat nyje x_i të kemi:

$$\mathbf{u}' = D\mathbf{u} \quad (4.2)$$

ku me $\mathbf{u} = [\hat{u}(x_1), \dots, \hat{u}(x_N)]^T$ shënohet vektori me vlerat e përafërta të zgjidhjes $\hat{u}$ në pikat nyje.

Shembull 4.1 *Për të pasur një ide të përdorimit të matricës diferencimit, konsiderojmë problemin një dimensional të transportit si më poshtë*

$$\begin{aligned} u_t(x, t) + cu_x(x, t) &= 0, \quad x > -1, t > 0, \\ u(-1, t) &= 0; \quad u(x, 0) = f(x) \end{aligned} \quad (4.3)$$

Për të zgjidhur problemin e mësipërm coptojmë zonën me pikat nyje x_i , $i = 1, \dots, N$, në mënyrë të tillë që në një moment të fiksuar t_n të kemi si vektor përafrimesh të zgjidhjes $\mathbf{u}^{(n)} = [\hat{u}(x_1, t_n), \dots, \hat{u}(x_N, t_n)]^T$. Për të vazhduar në kohë do të përdoren diferencat për përafrimin e derivatit në lidhje me kohën,

$$u_t(x, t_n) \approx \frac{u(x, t_{n+1}) - u(x, t_n)}{\Delta t}, \quad (4.4)$$

ku $\Delta t = t_{n+1} - t_n$. Zgjidhja e problemit të transportit (4.3) duke aplikuar matricën diferenciale në pikat nyje, sëbashku me përafrimin e derivatit në lidhje me kohën sipas formulës së diferencave (4.4), mund të shprehet në formë vektoriale si

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} - c\Delta t D\mathbf{u}^{(n)}. \quad (4.5)$$

Interesi ynë qendron në përdorimin e funksioneve me bazë radiale pafundësisht të lëmuar në funksionet e bazës në zbërthimin (4.1), dmth $B_j(x) = \varphi(\|x - x_j\|)$, ku φ është një nga funksionet me bazë radiale rigorozisht pozitivisht të përcaktuar.

4.1 Matricat Diferenciale

Rikthehemi edhe një herë në ekuacionin (4.1) dhe le të jenë B_j , $j = 1, \dots, N$ funksione arbitrare të lëmuara linearisht të pavarura. Këto funksione do të shërbejnë si funksione baze të zgjidhjes së përafërt.

Nëse llogaritet (4.1) në pikat nyje x_i , $i = 1, \dots, N$, atëherë kemi

$$\hat{u}(x_i) = \sum_{j=1}^N c_j B_j(x_i), \quad i = 1, \dots, N,$$

ose në formë matricore $\mathbf{u} = A\mathbf{c}$, ku $\mathbf{c} = [c_1, \dots, c_N]^T$, elementët e matricës A janë $A_{ij} = B_j(x_i)$ dhe $\mathbf{u}$ është si më parë.

Duke u bazuar në linearitetin e funksioneve bazë, mund të llogarisim derivatin e $\hat{u}$ si:

$$\frac{d}{dx} \hat{u}(x) = \sum_{j=1}^N c_j \frac{d}{dx} B_j(x).$$

Në trajtë matricore në pikat x_i , formula e mësipërme bëhet

$$\mathbf{u}' = A_x \mathbf{c}, \quad (4.6)$$

ku $\mathbf{u}$ dhe $\mathbf{c}$ janë si më sipër dhe *matrica derivat* ka elementë $\frac{d}{dx} B_j(x_i)$, ose në rastin e funksioneve me bazë radiale $\frac{d}{dx} \varphi(\|x - x_i\|)|_{x=x_i}$.

Për të përfutur matricën diferenciale D , duhet të sigurohet që matrica A të jetë jo singulare. Kjo gjë varet nga zgjedhja e funksioneve të bazës dhe të pikave nyje x_i . Duke u bazuar në ato që janë thënë në kapitullin 2, nëse përdorim funksionet me bazë radiale rigorozisht pozitivisht të përcaktuar sigurohet josingulariteti i matricës A (edhe për pika të shërndara jouniformisht në $\mathbb{R}^s$, $s > 1$).

Prandaj, formalisht mund të shkruajmë $\mathbf{c} = A^{-1} \mathbf{u}$ dhe që këtëj (4.6) shkruhet

$$\mathbf{u}' = A_x A^{-1} \mathbf{u}.$$

Që këtëj *matrica diferenciale* D jepet si

$$D = A_x A^{-1}.$$

Në të njëjtën linjë mund të vazhdohet edhe për operatorë diferencialë më kompleks $\mathcal{L}$, për të përfutur diskretizimin e operatorit diferencial (matricën diferenciale)

$$L = A_{\mathcal{L}} A^{-1}, \quad (4.7)$$

ku matrica $A_{\mathcal{L}}$ ka për element $(A_{\mathcal{L}})_{ij} = \mathcal{L} B_j(x_i)$. Në rastin e funksioneve me bazë radiale, elementët e matricës janë $(A_{\mathcal{L}})_{ij} = \mathcal{L} \varphi(\|x - x_j\|)|_{x=x_i}$. Në kontekstin e metodave pseudospektrale, matrica diferenciale D mund të përdoret për të zgjidhur të gjitha tipet e ekuacioneve diferenciale me derivate të pjeshshme (të varura nga koha ose jo).

Shembull 4.2 Zgjidhja e një problemi të keq–kushtëzuar

Kërkojmë zgjidhjen e një problemi linear EDP eliptik të keq–përcaktuar të formës $\mathcal{L}u = f$, duke mos marrë parasysh kushtet kufitare. Një zgjidhje e përafërt në pikat nyje x_i mund të merret duke zgjidhur sistemin linear diskret

$$L\mathbf{u} = \mathbf{f},$$

ku $\mathbf{f}$ përmban vlerat e f në pikat nyje dhe L është si më sipër. Me fjalë të tjera, zgjidhja në pikat nyje jepet si

$$\mathbf{u} = L^{-1}\mathbf{f} = A(A_{\mathcal{L}})^{-1}\mathbf{f}.$$

Në këtë rast duhet që matrica L (dhe rrjedhimisht $A_{\mathcal{L}}$) të jetë jo singulare. Nëse si funksione baze përdoren funksione me bazë radiale rigorozisht pozitivisht të përcaktuar, atëherë matrica $A_{\mathcal{L}}$ është e kthyeshme dhe operatori diferencial është eliptik (shiko [16] kap 36). Prandaj, matrica standarte diferenciale L për metodat PS të bazuara në funksionet me bazë radiale, është e kthyeshme. Kjo siguron një zgjidhje edhe për probleme të keq-përcaktuar!

Një program Matlab për llogaritjen e matricës diferenciale paraqitet në Anekisn B si programi *DRBF.m* [(4.9), Fq.98], i cili mund të gjendet në [16] si programi 43.1.

4.2 Metodat Pseudospektrale në EDDP me kushte kufitare

Shtrojmë problemin e zgjidhjes së EDDP eliptik

$$\mathcal{L}u = f$$

me kushte kufitare Dirichlet

$$u = g \text{ në } \Gamma = \partial\Omega$$

me anë të metodave pseudospektrale. Ndonjëherë mund të gjenden funksione bazë, të cilat plotësojnë menjëherë kushtet kufitare. Megjithatë, nëse funksionet bazë nuk plotësojnë kushtet kufitare mund të ndiqet një procedurë e thjeshtë që këto kushte të plotësohen (p.sh, shiko Programin 36 në [37]). Thjesht në matricën diferenciale L , ndryshohen rreshtat që u korrespondojnë pikave nyje në kufij, me vektorë njësi që kanë element të barabartë me 1 në pozicionin korrespondues të diagonales të L -së. Kështu, kushti $u = g$ do të plotësohet menjëherë nëse në krahun e djathtë do të zëvendësohen vlerat korresponduese të g -së.

Duke kryer një riorganizim të rreshtave dhe të shtyllave, përftohet një matricë me blloqe e formës

$$L_{\Gamma} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix} \quad (4.8)$$

ku blloqet jo-zero M dhe I kanë përmasa respektivisht $(N - N_B) \times (N - N_B)$ dhe $N_B \times N_B$. Simboli N_B përcakton numrin e pikave nyje në konturin Γ .

Në këtë mënyrë, zgjidhja e përafërt në pikat e zonës Ω merret duke zgjidhur sistemin e ekuacioneve lineare

$$L_{\Gamma} \mathbf{u} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \quad (4.9)$$

ku në vektorët $\mathbf{f}$ dhe $\mathbf{g}$ janë grupuar respektivisht vlerat e f dhe g në pikat nyje të brëndshme dhe kufitare. Vektori $\mathbf{u}$ mban vlerat e përafërta në pikat nyje. këto të dhëna janë rigrupuar në trajtën $\mathbf{u} = [\mathbf{u}_I, \mathbf{u}_B]^T$. Ku $\mathbf{u}_I$ përfshin pikat e brëndshme dhe rrjedhimisht $\mathbf{u}_B$ vlerat në pikat kufitare.

Duke zgjidhur sistemin 4.9 për $\mathbf{u}_B = \mathbf{g}$ dhe më pas duke zëvendësuar më pas në pjesën e sipërme marrim zgjidhjen

$$\mathbf{u}_I = M^{-1}(\mathbf{f} - P_{\mathbf{g}}),$$

ose në rastin e kushteve kufitare homogjene

$$\mathbf{u}_I = M^{-1}\mathbf{f}.$$

Kuptohet që problemi tani qëndron në faktin nëse matrica M është e kthyeshme ose jo.

4.3 Një Metodë Jo-Simetrike PS me bazë radiale

Nisemi nga metoda Kansa (shiko [26])

$$\hat{u}(x) = \sum_{j=1}^N c_j \Phi_j(\mathbf{x}), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^s. \quad (4.10)$$

njëlloj si në barazimin (4.1). Vektori $\mathbf{c}$ llogaritet duke zëvendësuar (4.10) në ekuacioni diferencial dhe kushtet kufitare për pikat nyje x_i . Zgjidhja në këtë rast merret duke zgjidhur sistemin e ekuacioneve lineare

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} \mathbf{c} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \quad (4.11)$$

ku $\mathbf{f}$ dhe $\mathbf{g}$ janë si më sipër, ndërsa matricat $\tilde{A}_{\mathcal{L}}$ dhe $\tilde{A}$ kanë për elemente

$$\begin{aligned} (\tilde{A}_{\mathcal{L}})_{ij} &= \mathcal{L}\Phi_j(x_i) = \mathcal{L}\varphi(\|x - x_j\|)|_{x=x_i}, \quad i = 1, \dots, N - N_B, \quad j = 1, \dots, N \\ \tilde{A}_{ij} &= \Phi_j(x_i) = \varphi(\|x_i - x_j\|), \quad i = N - N_B + 1, \dots, N, \quad j = 1, \dots, N. \end{aligned}$$

Nëse matrica e sistemit (4.11) është e kthyeshme, atëhere zgjidhja e përafërt merret duke zëvendësuar vektorin $\mathbf{c}$ në barazimin (4.10). Mund të ndodhë që për rrjeta të ndryshme pikash nyje të mos sigurohet josingulariteti i matricës së sistemit (sis:zgjidhjekans).

Meqë jemi të interesuar për zgjidhjen e bazuar në funksionet me bazë radiale, duke supozuar akoma që matrica e sistemit është jo singulare atëhere mund të shkruajmë

$$\mathbf{u} = \mathbf{A}\mathbf{c} = A \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$$

ku A ka element $A_{ij} = \Phi_j(x_i)$. Që këtë mund të themi se diskretizimi i operatorit diferencial L bazuar në pikat nyje $x_i, i = 1, \dots, N$ dhe funksionet e bazës $\Phi_j, j = 1, \dots, N$ jepen nga

$$L_{\Gamma} = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1}.$$

Në fakt, mund të shkruajmë

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix}$$

ku blloqet $M, P, 0$ dhe I janë si më parë. Për të treguar këtë futim shënimet e mëposhtme:

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} \text{ dhe } A^{-1} = [\mathbf{b}_1, \dots, \mathbf{b}_N]$$

ku vektorët $\mathbf{a}_i$ dhe $\mathbf{b}_i$ janë të tillë që $\mathbf{a}_i^T \mathbf{b}_j = \delta_{ij}$. Matrica e metodës Kansa pas këtyre shënimeve do të jetë

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{\mathcal{L},1}^T \\ \vdots \\ \mathbf{a}_{\mathcal{L},N-N_B}^T \\ \mathbf{a}_{N-N_B+1}^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix}.$$

Që këtë operatori diferencial sipas metodës jo-simetrike do të jetë

$$\begin{aligned} \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1} &= \begin{bmatrix} \mathbf{a}_{\mathcal{L},1}^T \\ \vdots \\ \mathbf{a}_{\mathcal{L},N-N_B}^T \\ \mathbf{a}_{N-N_B+1}^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} [\mathbf{b}_1, \dots, \mathbf{b}_N] = \\ &= \begin{bmatrix} \tilde{A}_{\mathcal{L}} A_I^{-1} & \tilde{A}_{\mathcal{L}} A_B^{-1} \\ \underbrace{\tilde{A} A_I^{-1}}_{=0} & \underbrace{\tilde{A} A_B^{-1}}_{=1} \end{bmatrix} \end{aligned}$$

Këtu matrica A^{-1} është ndarë në dy blloqe: blloku A_I^{-1} me $N - N_B$ kolona që u korrespondojnë pikave të brendëshme dhe blloku A_B^{-1} me N_B kolona që i korrespondojnë pikave të kufirit.

Kjo është e njëjta gjë me (shiko (4.8))

$$\begin{bmatrix} M & P \\ 0 & I \end{bmatrix} = L_{\Gamma}$$

ku M dhe P merren nga operatori diferencial diskret (4.7)

$$L = A_{\mathcal{L}} A^{-1} = [A_{\mathcal{L}} A_I^{-1} \quad A_{\mathcal{L}} A_B^{-1}]$$

duke ndryshuar disa rreshta me vektorë njësi, siç treguam në fillim, për të plotësuar kushtet kufitare Dirichlet.

Si përfundim, mund të themi se metoda jo simetrike bazuar në funksionet me bazë radiale për zgjidhjen e një EDDP eliptik me kushte kufitare Dirichlet është e ngjashme me përfaqsimin e një metode pseudospektrale. Për më tepër, asnjëra nga metodat nuk është në përgjithësi e mirë-përcaktuar, përderisa nuk sigurohet jo singulariteti i matricës së sistemit të formuar.

Nga ana tjetër, treguam se mund të ndërtohet gjithmonë operatori diferencial diskret

$$L_{\Gamma} = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix}$$

edhe në ato raste kur matrica e sistemit nuk është e kthyeshme. Kjo do të thotë se metoda jo simetrike pseudospektrale me funksione me bazë radiale, mund të përdoret në të gjitha rastet kur nuk kërkohet inversi i operatorit diferencial diskret.

4.4 Një Metodë Simetrike PS me bazë radiale

Për të njëtin problem kufitar eliptik me kushte kufitare Dirichlet fillojmë me përcaktimin e zgjidhjes së përafërt si

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N-N_B} c_j \mathcal{L}_j^{\varepsilon} \Phi(\mathbf{x}) + \sum_{j=N-N_B+1}^N c_j \Phi_j(\mathbf{x}) \quad (4.12)$$

Meqënëse Φ_j supsozohen të jenë funksione me bazë radiale, pra $\Phi_j(x) = \varphi(\|x - x_j\|)$, atëhere funksionalët $\mathcal{L}^{\varepsilon}$ mund të shikohen si aplikim i $\mathcal{L}$ në φ , të parë si funksione të variablit të dytë. Koeficientët $\mathbf{c} = [c_I, c_B]^T$ merren duke zgjidhur sistemin e ekuacioneve lineare

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}\mathcal{L}^{\varepsilon}} & \tilde{A}_{\mathcal{L}} \\ \tilde{A}_{\mathcal{L}^{\varepsilon}} & \tilde{A} \end{bmatrix} \begin{bmatrix} c_I \\ c_B \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (4.13)$$

Blloqet $\tilde{A}_{\mathcal{L}\mathcal{L}^{\varepsilon}}$ dhe $\tilde{A}$ janë matrica katrore të përfutuara respektivisht nga bashkëveprimi i pikave të brendëshme me njëra tjetrën dhe pikave të konturit. Elementët e tyre janë

$$(\tilde{A}_{\mathcal{L}\mathcal{L}^{\varepsilon}})_{ij} = \left[\mathcal{L}[\mathcal{L}^{\varepsilon} \varphi(\|x - \varepsilon\|)]_{\varepsilon=x_j} \right]_{x=x_i}, \quad i, j = 1, \dots, N - N_B,$$

$$\tilde{A}_{ij} = \varphi(\|x_i - x_j\|), \quad i, j = N - N_B + 1, \dots, N.$$

Dy blloqet e tjera janë matrica drejtëkëndëshe dhe merren si bashkëveprim i pikave të brendëshme me ato të kufirit dhe anasjelltas. Elementët e tyre janë

$$\begin{aligned} (\tilde{A}_{\mathcal{L}})_{ij} &= [\mathcal{L}\varphi(\|x - x_j\|)]_{x=x_i}, \quad i = 1, \dots, N - N_B, j = N - N_B, \dots, N \\ (\tilde{A}_{\mathcal{L}^e})_{ij} &= [\mathcal{L}^e\varphi(\|x_i - x\|)]_{x=x_j}, \quad i = N - N_B + 1, \dots, N, j = 1, \dots, N - N_B. \end{aligned}$$

Në rastin e funksioneve me bazë radiale rigorozisht pozitivisht të përcaktuar, matrica e sistemit është e kthyeshme. Kjo do të thotë, që zgjidhja e përafërt në çdo pikë $\mathbf{x}$ merret duke zëvendësuar zgjidhjen $\mathbf{c}$ të sistemit (4.13) në ekuacionin (4.12).

Mund të bëhet një diskutim mes dy përjasjeve, asaj jo simetrike dhe simetrike. Në të dyja përjasjet mund të mendojmë zgjidhjen si një kombinim linear i disa funksioneve baze të përshtatshme. Në trajtë vektoriale kjo mund të shkruhet

$$\hat{u}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{c}, \quad (4.14)$$

ku vektori $\mathbf{p}(\mathbf{x})$ mban vlerat e funksioneve të bazës në pikat $\mathbf{x}$. Në rastin e metodës jo simetrike, këto funksione janë vetëm Φ_j , $j = 1, \dots, N$, ndërsa në rastin e përjasjes simetrike përbëhen nga funksione të tipëve Φ_j dhe $\mathcal{L}_j^e\Phi$ (barazimi (4.12)).

Shënojmë me $\mathcal{D}$ operatorin linear që kombinon sëbashku operatorin diferencial $\mathcal{L}$ dhe operatorin kufitar (në rastin e kushteve kufitare Dirichlet është thjesht identiteti I). Në këtë mënyrë do të kemi

$$\mathcal{D}\hat{u}(\mathbf{x}) = \mathcal{D}\mathbf{p}^T(\mathbf{x})\mathbf{c} = \mathbf{q}^T(\mathbf{x})\mathbf{c} \quad (4.15)$$

për një vektor $\mathbf{q}(\mathbf{x})$ të përcaktuar në mënyrë të përshtatshme. Që këtë problem i vlerës kufitare për zgjidhjen e përafërt shkruhet si

$$\mathcal{D}\hat{u}(x) = f(x),$$

ku me f përcaktohet një funksion pjesë-pjesë, i cili përcakton funksionet si në pjesën e pikave të brendshme ashtu edhe të konturit.

Nëse llogarisim (4.14) dhe (4.15) në rrjetën e pikave x_i , $i = 1, \dots, N$ marrim

$$\mathbf{u} = P\mathbf{c} \text{ dhe } \mathbf{u}_{\mathcal{D}} = Q\mathbf{c}$$

ku rreshtat e matricave P dhe Q u korrespondojnë respektivisht vlerësimeve të vektorëve $\mathbf{p}^T(x)$ dhe $\mathbf{q}^T(x)$ në pikat x_i . Në këtë mënyrë problemi i diskretizuar i vlerës kufitare është

$$\mathbf{u}_D = \mathbf{f} \iff Q\mathbf{c} = \mathbf{f}, \quad (4.16)$$

ku me $\mathbf{f}$ shënohet vlera e funksionit f në pikat nyje të rrjetës.

Në rastin e përqasjes jo simetrike matrica P është matrica standarte e interpolimit me funksionet me bazë radiale dhe matrica e derivateve Q është matrica e Kansa's. Në rastin e përqasjes simetrike, matrica P jepet nga matrica e transpozuar e Kansa's.

Qëllimi është gjetja e zgjidhjes së përafërt $\mathbf{u}$ në pikat nyje. Për këtë qellim mund të ndiqen dy rrugë.

- (1) Fillimisht zgjidhet $Q\mathbf{c} = \mathbf{f}$ në lidhje me $\mathbf{c}$, pra

$$\mathbf{c} = Q^{-1}\mathbf{f}.$$

Më pas, në bazë të barazimit (4.14) mund të llogarisim zgjidhjet $\mathbf{u}$ si

$$\mathbf{u} = PQ^{-1}\mathbf{f}.$$

- (2) Fillimisht transformojmë formalisht koeficientët $\mathbf{c}$

$$\mathbf{c} = P^{-1}\mathbf{u}.$$

Më pas, problemi kufitar (4.16) kthehet

$$QP^{-1}\mathbf{u} = \mathbf{f}$$

dhe që këtë, $\mathbf{u}$ mund të merret duke zgjidhur sistemin.

Përqasja e parë i korespondon metodës që përdor funksionet me bazë radiale, ndërsa e dyta metodave pseudospektrale. Megjithatë, të dyja përqasjet janë ekuivalente për aq kohë sa matricat e përfshira në to janë të kthyeshme. Fatkeqsisht kjo gjë nuk është gjithmonë e mundur, sepse ka disa konfigurime pikash ku matrica e Kansa's nuk është e kthyeshme. Kjo do të thotë se në rastin e metodës jo simetrike (ku Q është matrica Kansa), përqasja e parë nuk është e sigurt që funksionon në përgjithësi dhe përqasja e dytë mund të përdoret nëse operatori diferencial diskret aplikohet drejtpërdrejt, por jo i anasjelltë. Në rastin e metodës simetrike (ku P është matrica Kansa), përqasja e parë funksionon gjithmonë, kurse përqasja e dytë mund të mos jetë gjithmonë e mirë-përcaktuar.

Përpara se sa të kalojmë në zbatime të metodave PS me bazë radiale në EDP, po bëjmë një përmbledhje diskutimeve të bëra. Në rastin e metodës jo simetrike mund të formulohet gjithmonë operatori diferencial diskret

$$L_T = \begin{bmatrix} \tilde{A}_g \\ \tilde{A} \end{bmatrix} A^{-1}.$$

Megjithatë, nuk mund të sigurohet gjithmonë jo singulariteti i operatorit L_T . Që këtë rrjedh se, përjasja jo simetrike PS me funksione me bazë radiale justifikohet për EDDP të varur nga koha (duke përdorur metoda eksplicite për avancimin në kohë (*ang.* time-stepping)). Në krahun tjetër, rasti i përjasjes simetrike në përgjithësi siguron zgjidhjen e problemit $\mathcal{L}u = f$ me anë të funksioneve me bazë radiale. Megjithëse, në përgjithësi nuk është i mundur as formulimi i operatorit diferencial diskret, i cili nevojitet në rastin e metodave PS. Kjo na sugjeron përdorimin e përjasjes simetrike në rastin e EDDP të pavarur nga koha dhe mundësisht në rastin e EDDP të varur nga koha me metoda implicite për avancimin në kohë. Vështirësitë në të dyja përjasjet lidhjen me singularitetin e matricës Kansa, e cila në rastin e përjasjes jo simetrike shfaqet si operator diferencial diskret dhe si matrica e llogaritjeve në përjasjen simetrike. Në shumicën e rasteve preferohet të përdoret përjasja jo simetrike sepse është më e lehtë për tu përdorur dhe është treguar nga shumë kërkues (shiko [25]) që rrjetat e pikave që sjellin singularitet të matricës Kansa janë shumë të rralla.

4.5 Zbatime të metodave PS me funksione me bazë radiale

Bazuar në diskutimet e mësipërme mund të themi se mund të aplikohen funksionet me bazë radiale në metodat pseudospektrale. Në veçanti, në zgjidhjen e ekuacioneve diferenciale me derivate të pjesshme që varen nga koha. Tre probleme diferenciale tipike do të konsiderohen më poshtë për të ilustruar përdorimin e metodave PS me funksionet me bazë radiale. Zgjidhja e këtyre 3 tre problemeve me anë të metodave standarte PS gjenden në [36].

4.5.1 Zgjidhja e ekuacionit të transportit 1-D

Konsiderojmë problemin e transportit të prezantuar në shembullin (4.1).

$$\begin{aligned} u_t(x, t) + cu_x(x, t) &= 0, \quad x > -1, t > 0, \\ u(-1, t) &= 0; \quad u(x, 0) = f(x). \end{aligned}$$

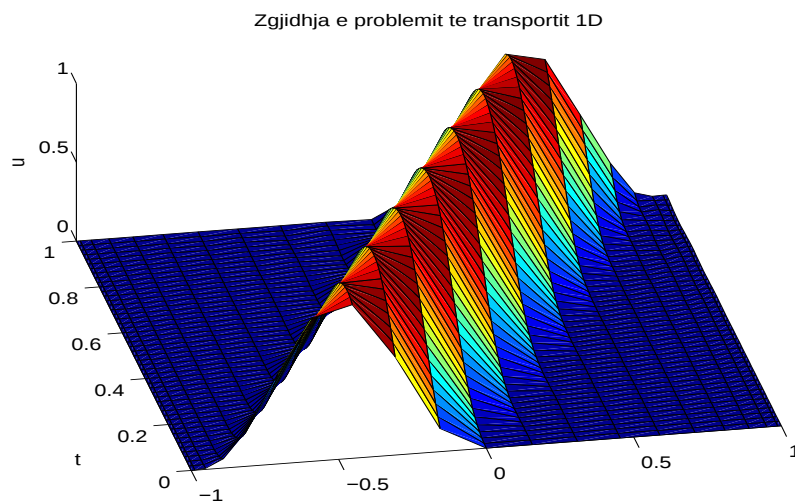


Figura 4.1 Zgjidhja numerike e problemit të transportit 1D me anë të metodave PS me funksione me bazë radiale.

Zgjidhja e të cilit ka trajtën:

$$u(x, t) = f(x - ct).$$

Për implementimin e zgjidhjes së përafërt të tij në Matlab, duhet hartuar kodi MATLAB **DRBF.m** [(4.9), Fq.98], i cili llogarit matricën diferenciale. Për përcaktimin e vlerës optimale të parametrut të shkallëzimit ε është përdorur metoda "largo nyje" (shiko Asimi and Hanelli, 2016b [4]). Vlera optimale rezulton $\varepsilon = 1.874764$. Funksioni i përdorur është funksioni Gaussian. Gabimi maksimal në çastin $t = 1$ është 0.0416. Programi matlab që përafron zgjidhjen e problemit të transportit 1D jepet më poshtë, në programin (4.1) dhe zgjidhja numerike paraqitet në figurën 4.1. Llogaritjet janë kryer në një rrjetë me 21 pika.

Programi 4.1 *TransportDRBF.m*

```

lineskip
1
2 % TransportDRBF
3 rbf= @(e, r) exp(-(e*r).^2); % Funksioni Gaussian.
4 dxrbf= @(e, r, dx) -2*dx*e.^2.*exp(-(e*r).^2);
5 N=20;
6 [D, x]=DRBF(N, rbf, dxrbf);
7 x=flipud(x); dt=0.001; t=0; c=-1;
8 v=64*(-x).^3.*(1+x).^3;
9 v(find(x>0))=zeros(length(find(x>0)),1);
10 % Ndarja e kohes nga formula Euler
11 tmax=1; tplot=.02; plotgap= round(tplot/dt);

```

```

12 dt=tplot/plotgap; nplots=round(tmax/tplot);
13 data=[v'; zeros(nplots,N+1)]; tdata=t;
14 I=eye(size(D));
15 for i=1:nplots
16     for n=1:plotgap
17         t=t+dt;
18         vv=v(end-1);
19         v=v-dt*c*(D*v); % Formula Euler
20         v(1)=0; v(end)=vv;
21     end
22     data(i+1,:)=v'; tdata=[tdata; t];
23 end
24 surf(x,tdata,data), view(10,70), colormap('default');
25 axis([-1 1 0 tmax 0 1]), ylabel('t'), zlabel('u'), grid off
26 % Zgjidhja e sakte dhe gabimi i kryer
27 xx=linspace(-1,1,101);
28 vone=64*(1-xx).^3.*xx.^3;
29 vone(find(xx<0))=zeros(length(find(xx<0)),1);
30 w=interp1(x,v,xx);
31 maxErr=norm(w-vone,inf)

```

4.5.2 Zgjidhja e ekuacionit Allen Cahn

Konsiderojmë problemin e zgjidhjes së ekuacionit jolinear reaksion-difuzion. Problemi specifik që do të zgjidhim është i formës

$$u_t = \mu u_{xx} + u - u^3, \quad x \in (-1, 1), \quad t \geq 0$$

me koeficient μ , kusht fillestar

$$u(x, 0) = 0.53x + 0.47 \sin\left(-\frac{3}{2}\pi x\right), \quad x \in [-1, 1]$$

dhe kushte kufitare jo homogjene (të varura nga koha) $u(-1, t) = -1$ dhe $u(1, t) = \sin^2(t/5)$. Zgjidhja e këtij ekuacioni paraqitet në tre gjëndje ($u = (-1, 0, 1)$), ku dy zgjidhjet jozero janë të qendrueshme. Kalimi midis këtyre gjëndjeve rregullohet nga koeficienti μ . Në llogaritjet në vazhdim do të merret $\mu = 0.01$ dhe gjëndja jo e qendrueshme do të zhduket rreth $t = 30$.

Problemi ynë diferencial është i rendit të dytë. Kjo do të thotë që duhet llogaritur matrica diferenciale e rendit të dytë. Në rastin e matricave diferenciale polinomiale, llogaritja e derivatit të rendit k kryhet duke marrë fuqinë e k -të të matricës diferenciale të rendit të parë, pra $D^{(k)} = D^k$. Fatkeqësisht, në rastin e funksioneve me bazë radiale kjo veti nuk qendron. Për pasojë, duhet ndërtuar kod shtesë në Matlab për llogaritjen e matricave diferenciale

të rendeve të larta. Kodi për rastin e matricës diferenciale të rendit të dytë paraqitet në aneksin B si programi **D2RBF.m** [(4.12), Fq.99]. Programi Matlab **AllenChan.m** i paraqitur si programi (4.2), jep zgjidhjen numerike të problemit tonë diferencial. Paraqitja grafike e zgjidhjes jepet në figurën (4.2). Parametri i shkallëzimi optimal për funksionin Gaussian të përdorur, është $\varepsilon = 1.873402$ dhe merret duke u bazuar në algoritmin "Përjashto një të dhënë". Funkzioni **CostEpsilonDRBF.m** [(4.11), Fq.99] për llogaritjen e parametrin optimal mund të modifikohet shumë lehtë për rastin e matricës diferenciale të rendit të dytë.

Programi 4.2 AllenChan.m

```

lineskip
1 % Skript qe zgjidh ekuacionin Allen-Chan me kushte kufitare
2 % me ane te metodes pseudospektrale te bazuar ne
3 % funksionet me baze radiale
4 % Therret funksioni: D2RBF
5 % Si funksion me baze radiale perdorim Gaussianin
6 rbf = @(e,r) exp(-(e*r).^2);
7 d2rbf = @(e,r) 2*e^2*(2*(e*r).^2-1).*exp(-(e*r).^2);
8 N = 20;
9 [D2,x] = D2RBF(N,rbf,d2rbf);
10
11 mu = 0.01; dt = min([.01,50*N^(-4)/mu]);
12 t = 0; v = .53*x + .47*sin(-1.5*pi*x);
13 % Perdoret metoda Euler
14 tmax = 100; tplot = 2; nplots = round(tmax/tplot);
15 plotgap = round(tplot/dt); dt = tplot/plotgap;
16 xx = -1:.025:1; vv = polyval(polyfit(x,v,N),xx);
17 plotdata = [vv; zeros(nplots,length(xx))]; tdata = t;
18 for i = 1:nplots
19     for n = 1:plotgap
20         t = t+dt; v = v + dt*(mu*D2*v + v - v.^3); % Euler
21         v(1) = 1 + sin(t/5)^2; v(end) = -1; % Kushti kufitar
22     end
23     vv = polyval(polyfit(x,v,N),xx);
24     plotdata(i+1,:) = vv; tdata = [tdata; t];
25 end
26 % Paraqitet zgjidhja grafikisht
27 surf(xx,tdata,plotdata), grid on
28 axis([-1 1 0 tmax -1 2]), view(-40,55)
29 colormap('default'); xlabel x, ylabel t, zlabel u
30 title ('Zgjidhja e ekuacionit Allen-Cahn')

```

Për avancimin në kohë përdoret metoda Euler në rreshtin 20 të kodit. Gjithashtu, shumë i rëndësishëm është fakti që zgjidhja e sistemit linear kërkohet vetëm në gjetjen matricës derivat dhe jo në ciklin për avancimin në kohë në rreshtat 18-25. Në këtë cikël kërkohen vetëm shumëzime matricore.

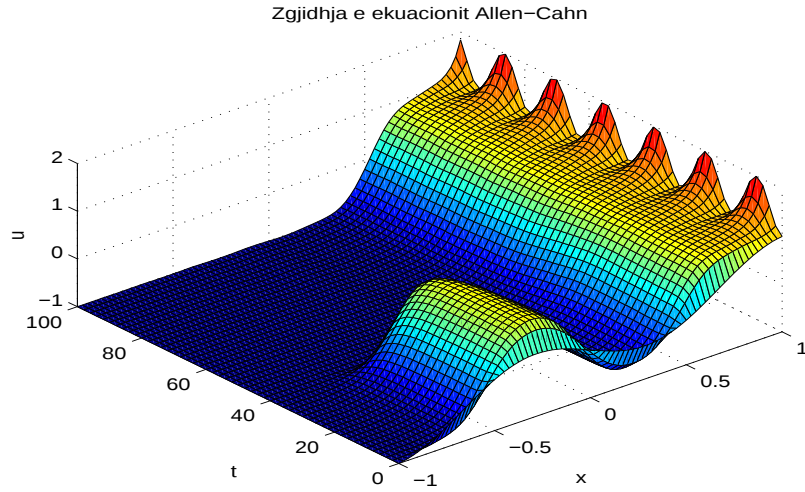


Figura 4.2 Zgjidhja numerike e problemit Allan Chan me anë të metodave PS me funksione me bazë radiale.

4.5.3 Zgjidhja e ekuacionit Helmholtz 2D

Fillojmë me përcaktimin e ekuacionit Helmholtz për rastin 2D

$$u_{xx} + u_{yy} + k^2 u = f(x, y), \quad x, y \in (-1, 1)^2,$$

me kushte kufitare $u = 0$ dhe

$$f(x, y) = \exp\left(-10\left[(y-1)^2 + \left(x - \frac{1}{2}\right)^2\right]\right).$$

Do të zgjidhim këtë ekuacion me anë të funksioneve me bazë radiale, me përfaqësues Gaussianin, duke përdorur funksioni *kron* për të shprehur Laplasianin diskret në një rrjet prodhimi tenzorësh $(N+1) \times (N+1)$ si

$$L = I \otimes D2 + D2 \otimes I, \quad (4.17)$$

ku $D2$ është matrica diferenciale e rendit të dytë, I është matrica identike me përmasa $(N+1) \times (N+1)$ dhe $\otimes$ është *shumëzimi-tenzor Kronecker*. Thamë se në rastin e matricave PS për të llogaritur matricën diferenciale të rendit të dytë, mjaftonte të merrnim D^2 . Në rastin e metodave PS me funksione baze radiale $D^{(2)} \neq D^2$, prandaj duhet të përdorim përsëri funksionin Matlab ***D2RBF.m*** [(4.12), Fq.99].

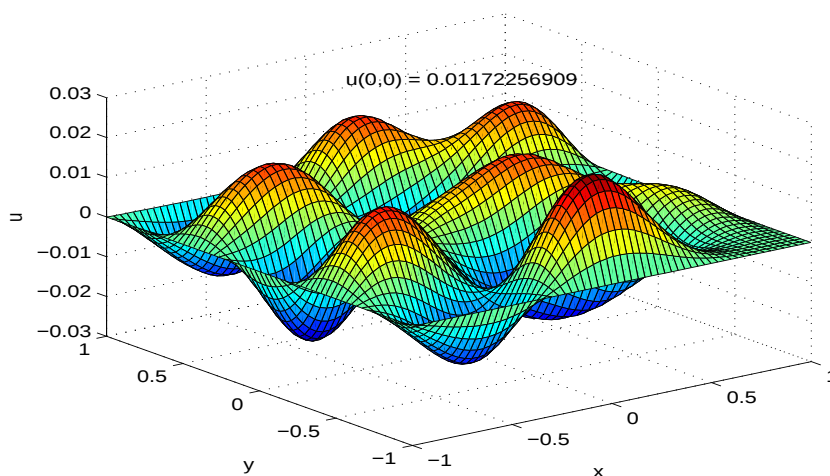


Figura 4.3 Zgjidhja numerike e problemit Helmholtz me anë të metodave PS me funksione me bazë radiale.

Barazimi (4.17) implementohet në rreshtat 10 dhe 15 të programit (4.3). Zgjidhja numerike e ekuacionit për $k = 9$ dhe $N = 24$ (pra në total 625 pika) paraqitet në figurën (4.3). Parametri optimal i shkallëzimit rezulton $\varepsilon = 2.548968$.

Programi 4.3 *Helmholtz2D.m*

```

lineskip
1 % Skript qe zgjidh ekuacionin Helmholtz
2 %  $u_{xx} + u_{yy} + (k^2)u = f$  ne  $[-1,1] \times [-1,1]$ 
3 % me ane te metodes pseudospektrale te bazuar ne
4 % funksionet me baze radiale
5 % Calls on: D2RBF
6 % Si funksion me baze radiale perdorim Gaussianin
7 rbf = @(e,r) exp(-(e*r).^2);
8 d2rbf = @(e,r) 2*e^2*(2*(e*r).^2 - 1).*exp(-(e*r).^2);
9 N = 24;
10 [D2,x] = D2RBF(N,rbf,d2rbf); y = x;
11 [xx,yy] = meshgrid(x,y);
12 xx = xx(:); yy = yy(:);
13 I = eye(N+1);
14 k = 9;
15 L = kron(I,D2) + kron(D2,I) + k^2*eye((N+1)^2);
16 % Percaktohen kushtet kufitare duke ndryshuar rreshtat e L
17 b = find(abs(xx)==1 | abs(yy)==1); % Pikat kufitare
18 L(b,:) = zeros(4*N,(N+1)^2); L(b,b) = eye(4*N);
19 f = exp(-10*((yy-1).^2+(xx-.5).^2));
20 f(b) = zeros(4*N,1);
21 % Zgjidhe ne lidhje me u, kthe ne 2D dhe nderto grafikun:

```

```

22  u = L\f;
23  uu = reshape(u,N+1,N+1);
24  [xx,yy] = meshgrid(x,y);
25  [xxx,yyy] = meshgrid(-1:.0333:1,-1:.0333:1);
26  uuu = interp2(xx,yy,uu,xxx,yyy,'cubic');
27  figure, clf, surf(xxx,yyy,uuu),
28  xlabel x, ylabel y, zlabel u
29  text(.2,1,.022,sprintf('u(0,0) = %13.11f',uu(N/2+1,N/2+1)))

```

4.6 Metodat Galerkin dhe funksionet me bazë radiale

Konsiderojmë ekuacioni Helmholtz me kushte kufitare të zakonshme, pra

$$\begin{aligned} -\Delta u + u &= f \quad \text{ne } \Omega \\ \frac{\partial}{\partial \mathbf{n}} u &= 0 \quad \text{ne } \partial\Omega \end{aligned}$$

ku me $\mathbf{n}$ shënohet vektori njësi normal.

Formulimi klasik Galerkin konsiston në gjetjen e një funksioni $u \in H^1(\Omega)$ të tillë që

$$a(u, v) = (f, v)_{L_2(\Omega)} \quad v \in H^1(\Omega),$$

ku $(f, v)_{L_2(\Omega)}$ është prodhimi skalar në L_2 dhe forma bilineare a për ekuacionin Helmholtz është

$$a(u, v) = \int_{\Omega} (\nabla u \cdot \nabla v + uv) d\mathbf{x}.$$

Për të përfutur një zgjidhje numerike të problemit, hapsira me dimensione të pafundme $H^1(\Omega)$ zëvendësohet me ndonjë nënhapsirë me dimensione të fundme $\mathbb{S}_{\chi} \subseteq H^1(\Omega)$, ku me χ shënohet rrjeta e pikave që do të përdoren për të llogaritur zgjidhjen. Në kontekstin e funksioneve me bazë radiale $\mathbb{S}_{\chi}$ merret si

$$\mathbb{S}_{\chi} = \text{span}\{\varphi(\|\cdot - x_j\|), x_j \in \varphi\}.$$

Kjo sjell ndërtimin e një sistemi ekuacionesh lineare për koeficientët e $\hat{u} \in \mathbb{S}_{\chi}$ të përcaktuar nga

$$a(\hat{u}, v) = (f, v)_{L_2(\Omega)} \quad \forall v \in \mathbb{S}_{\chi}. \quad (4.18)$$

Në mënyrë më specifike, nëse $\chi = \{x_1, \dots, x_N\}$, atëhere

$$\hat{u} = \sum_{j=1}^N c_j \varphi(\|\cdot - x_j\|),$$

dhe sistemi (4.18) jepet si

$$\int_{\Omega} [\nabla \hat{u}(x) \cdot \nabla \varphi(\|x - x_i\|) + \hat{u} \varphi(\|x - x_i\|)] dx = \int_{\Omega} f(x) \varphi(\|x - x_i\|) dx, \quad i = 1, \dots, N.$$

Bazuar në linearitetin dhe përkufizimin e mësipërm të $\hat{u}$, barazimi i mësipërm kthehet

$$\begin{aligned} \sum_{j=1}^N c_j \left\{ \int_{\Omega} [\nabla \varphi(\|x - x_i\|) \cdot \nabla \varphi(\|x - x_j\|) + \varphi(\|x - x_i\|) \cdot \varphi(\|x - x_j\|)] dx \right\} = \\ = \int_{\Omega} f(x) \varphi(\|x - x_i\|) dx, \quad i = 1, \dots, N. \end{aligned}$$

Në trajtë matricore mund të shkruajmë $\mathbf{A} \mathbf{c} = \mathbf{f}$, ku elementët e matricës *stiffness* A janë

$$A_{ij} = \int_{\Omega} [\nabla \varphi(\|x - x_j\|) \cdot \nabla \varphi(\|x - x_i\|) + \varphi(\|x - x_j\|) \cdot \varphi(\|x - x_i\|)] dx$$

dhe termat e lira të krahut të djathtë

$$\mathbf{f}_i = \int_{\Omega} f(x) \varphi(\|x - x_i\|) dx.$$

Logaritja e këtyre integraleve është problemi më i madhë në metodat Galerkin me funksione me bazë radiale (llogaritjet e shumta dhe koha e madhe për tu llogaritur). Për më tepër, (shiko [39] dhe [40]) metodat Galerkin me funksionet me bazë radiale kanë vështirësi në rastin e kushteve kufitare Dirichlet. Të dyja këto lloj vështirësisë hasen edhe në shumë metoda meshfree.

4.6.1 Një problemë zbatimi

Konsiderojmë problemin Helmholtz

$$\begin{aligned} -\Delta u(x, y) + u(x, y) &= \cos(\pi x) \cos(\pi y), \quad (x, y) \in \Omega = [-1, 1]^2, \\ \frac{\partial}{\partial \mathbf{n}} u(x, y) &= 0, \quad n \partial \Omega, \end{aligned}$$

ku $(x, y) \in \mathbb{R}^2$ dhe $\mathbf{n}$ është vektori i jashtëm normal njësi. Mund të konstatohet se zgjidhja e saktë është

$$u(x, y) = \frac{\cos(\pi x) \cos(\pi y)}{2\pi^2 + 1}$$

Në programin (4.4) jepet një implementim i Metodës Galerkin në Matlab për zgjidhjen numerike të problemit të mësipërm. Ashtu si dhe thamë më sipër, pjesa më e vështirë dhe më pak efiçente është llogaritja e integraleve për gjenerimin e matricës *stiffness* A , elementët e të cilës janë

$$A_{ij} = \int_{[-1,1]^2} \nabla \varphi(\|x - x_i\|) \cdot \nabla \varphi(\|x - x_j\|) dx + \int_{[-1,1]^2} \varphi(\|x - x_i\|) \cdot \varphi(\|x - x_j\|) dx$$

dhe termat e lira

$$\int_{[-1,1]^2} f(x) \varphi(\|x - x_i\|) dx.$$

Për të llogaritur këto integrale përdorim programin për inegrim numerik *dblquad* (rreshtat 33 – 39 të programit 4.4). Për ndërtimin e matricës A , fillimisht shfrytëzohet simetria e saj në ciklin *for* dhe më pasë ndërtohet ajo në rreshtin 44.

Programi 4.4 *Galerkin2D.m*

```

lineskip
1 % Galerkin2D
2 % Zgjidh problemin Helmholtz 2 dimensional
3 % - u_xx - u_yy + u = f
4
5 % Funksioni me baze radiale dhe gradienti , Funksioni Wendland C2
6 rbf=@(e,x,y,xi,yi) exp(-(e*sqrt((x-xi).^2+(y-yi).^2)).^2);
7 dxrbf=@(e,x,y,xi,yi) -2*e.^2*(x-xi).*exp(-(e*sqrt((x-xi).^2+(y-yi).^2)).^2);
8 dyrbf=@(e,x,y,xi,yi) -2*e.^2*(y-yi).*exp(-(e*sqrt((x-xi).^2+(y-yi).^2)).^2);
9 evalrbf=@(e,r) exp(-(e*r).^2);
10 % Shumezimet per integralet
11 rp = @(e,x,y,xi,yi,xj,yj) rbf(e,x,y,xi,yi).*rbf(e,x,y,xj,yj);
12 gp = @(e,x,y,xi,yi,xj,yj) dxrbf(e,x,y,xi,yi).*...
13 dxrbf(e,x,y,xj,yj)+dyrbf(e,x,y,xi,yi).*dyrbf(e,x,y,xj,yj);
14 % Parametri i shkallezimit
15 ep = 1.7;
16 % Funksioni i krahut te djathte ne ekuacionin Helmholtz

```

```

17 f = @(x,y) cos(pi*x).*cos(pi*y);
18 % Zgjidhja e sakte
19 u = @(x,y) cos(pi*x).*cos(pi*y)/(2*pi^2+1);
20 % Përcaktohen qendrat dhe tipi i tyre (pika uniforme)
21 N = 81; gridtype = 'u';
22 % Permasat e rrjetës së pikëve për të llogaritur dhe paraqitur grafikisht
23 neval = 40;
24 % Thërriten të dhënat 2D
25 name = sprintf('Data2D_%d%s', N, gridtype); load(name)
26 % Shndërrojmë qendrat nga [0,1]^2 në [-1,1]^2
27 ctrs = 2*dsites-1;
28 % Matrica e siftness dhe krahu i djathtë i sistemit
29 A = zeros(N,N); rhs = zeros(N,1);
30 for i=1:N
31     for j=1:i
32         A(i,j) = dblquad(@(x,y) gp(ep,x,y, ctrs(i,1), ctrs(i,2), ...
33                               ctrs(j,1), ctrs(j,2)), -1,1, -1,1) + ...
34               dblquad(@(x,y) rp(ep,x,y, ctrs(i,1), ctrs(i,2), ...
35                               ctrs(j,1), ctrs(j,2)), -1,1, -1,1);
36     end
37     rhs(i) = dblquad(@(x,y) f(x,y).*...
38                     rbf(ep,x,y, ctrs(i,1), ctrs(i,2)), -1,1, -1,1);
39 end
40 % Ndërtohet matrica simetrike
41 A = A + A' - diag(diag(A));
42 % Gënden koeficientet me zgjidhjen e sistemit
43 c = A\rhs;
44 % Kryen Llogaritjet
45 grid = linspace(-1,1,neval); [xe, ye] = meshgrid(grid);
46 epoints = [xe(:) ye(:)];
47 exact = u(epoints(:,1), epoints(:,2));
48 DM_eval = DistanceMatrix(epoints, ctrs);
49 EM = evalrbf(ep, DM_eval);
50 Pf = EM * c;
51 % Llogariten Gabimet
52 maxerr = norm(Pf-exact, inf); rms_err = norm(Pf-exact)/neval;
53 fprintf('Gabimi i Mesatsarizuar (RMS): %e\n', rms_err)
54 fprintf('Gabimi Maksimal: %e\n', maxerr)
55 % Paraqitjet grafike
56 fview = [-30,30];
57 titulli = 'Zgjidhja Numerike. Ekuacioni Helmholtz';
58 PlotSurf(xe, ye, Pf, neval, exact, maxerr, fview, titulli);
59 titulli = 'Gabimi Maksimal';
60 PlotError2D(xe, ye, Pf, exact, maxerr, neval, fview, titulli)

```

Për këtë problem është përdorur Gausiani si funksioni me bazë radiale $\varphi_{3,1}(r) = (1 - r)_4^+(4r + 1)$ dhe parametër shkallëzimi $\varepsilon = 1.7$. Zgjidhja numerike sëbashku me gabimin

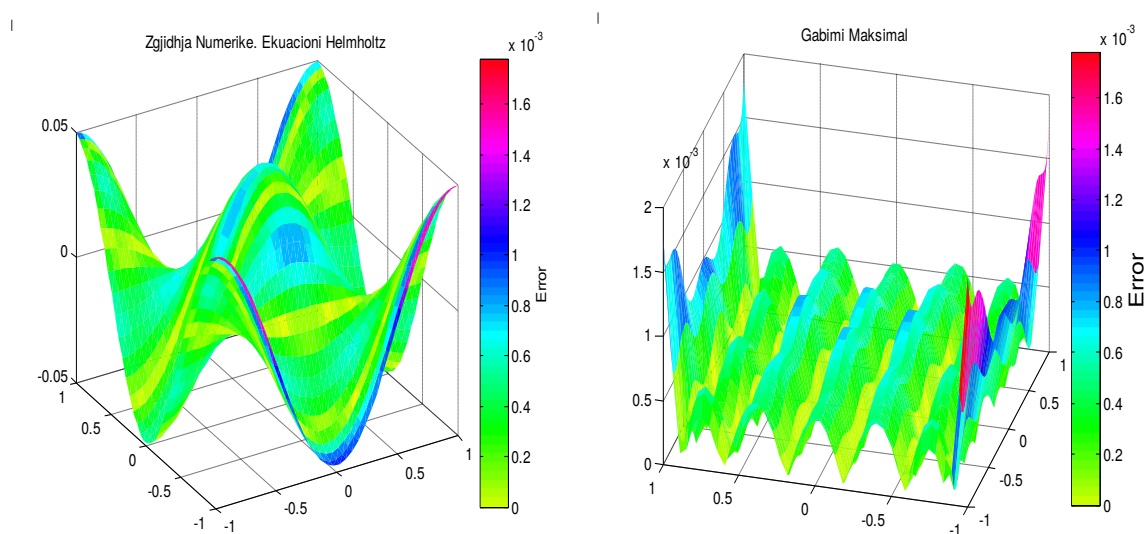


Figura 4.4 **Majtas**: Zgjidhja numerike me metodën Galerkin me Gaussianin si funksion me bazë radiale dhe **Djathtas**: Gabimi maksimal

për $N = 81$ pika nyje, paraqitet në figurën (4.4). Vlerat e llogaritura të gabimit në matlab janë: gabimi maksimal: $1.780074e-03$ dhe gabimi i mesatarizuar: $5.101441e-04$.

Përfundime

Manipulimi me një numër të konsiderueshëm të dhënash është një nga problemet kryesore në shkencë dhe inxhinieri. Metodat klasike numerike shpesh herë nuk mundësojnë zgjidhjen e këtyre problemeve, me përjashtim të rasteve standarte. Në të kundërt Metodat Meshfree mund të përshtaten më mirë me situata më të ndërlikuara. Metodat përafruese për të dhënat jo uniforme studiohen nga shumë kërkues. Ato ekzistojnë në shumë fusha të ndryshme të shkencës dhe njihen me emra të ndryshëm, p.sh metoda e elementëve-free Galerkin, metoda e elementeve të fundme të përgjithshuar, hp-clouds, metoda meshless lokale Petrov-Galerkin, metoda e katrorëve më të vegjël lëvizëse; metoda e funksioneve me bazë radiale.

Një avantazh tjetër është pavarësia e tyre nga shpërndarja e të dhënave. Nuk është i nevojshëm optimimi ose diskretizimi i zonës së shqyrtuar në zona ose element të caktuar, përderisa këto metoda bazohen në një bashkësi pikash të pavarura. Kjo ul ndjeshën edhe koston llogaritse numerike në zgjidhjen e problemeve me zona të ndryshme.

Si shembull i parë i interpolimit me anë të funksioneve me bazë radiale u përdorën si funksione bazë $B_k = \|\cdot - x_k\|_2$. Vihet re thjeshtësia në implementimin e metodës për vlera të ndryshme dimensionale s . Kryerja e interpolimit me matricën e distancave ka nevojë për zgjidhjen e një sistemi me matricë $N \times N$. Ky përbën një problem kur numri i të dhënave rritet shumë, për shkak të kompleksitetit të veprimeve.

Siç thamë, interpolimi me anë të funksioneve me bazë radiale kërkon zgjidhjen e një sistemi ekuacionesh lineare. Nga përfundimet e paraqitura në kapitullin e dytë, mund të themi se funksionet me bazë radiale (më konkretisht Gaussianët), duke qënë funksione rigoroz pozitivisht të përcaktuar dhe totalisht monotone, nëse përdoren si funksione bazë në problemin e interpolimit sigurojnë që matrica interpoluese të jetë jo singulare. Për pasojë, problemi i interpolimit ka zgjidhje të vetme në $\mathbb{R}^s$ për $\forall s$.

Për të pasur një model sa më të mirë është me shumë rëndësi të studiohet efekti i parametrut ε në qendrueshmërinë dhe saktësinë e zgjidhjes së problemit të interpolimit. Gjatë eksperimenteve numerike të kryera për interpolimin stacionar dhe jostacionar u vu re se kurbat e gabimit nuk janë monotone dhe und të identifikojmë vlerën optimale të parametrut. Por, në disa raste u vu re se vlera optimale arrihet në një zonë ku kurba e gabimit nuk

është e njëtrajtshme. Për vlera të parametrut ε nga kjo zonë, MATLAB jep mesazhe për keqkushtëzim. Prandaj, mund të përcaktohet vlera optimale e parametrut në dy mënyra: si vlerë minimale gabimi ose si vlera minimale e parametrut ku nuk shfaqet mesazhi për keqkushtëzim.

Disa eksperimente të tjera numerike u kryen me modele interpolimi 1 dhe 2 dimensional për të përcaktuar vlerën optimale të parametrut ε . Si metoda krahasimi u përdorën metodat "Provë Gabim" dhe "Përfashtoj një të dhënë". Metoda "Provë Gabim" ka nevojë për njohjen e funksioneve që gjenerojnë të dhënat, por mund të përdoret si mjet krahasues për metoda të tjera. Bazuar në eksperimentet e kryera, u vu re se metoda "Përfashtoj një të dhënë" jep rezultate të kënaqshme në llogaritjen e parametrut optimal të shkallëzimit. Rezultatet e marra prej saj janë të përafërta me metodën "Provë-Gabim" edhe pse kjo e fundit ka nevojë për njohjen e funksionit që gjeneron të dhënat.

Shpesh është më e përshtatshme përafrimi i të dhënave me anë të metodeës së katrorëve më të vegjël. Kjo gjë është e nevojshme në rastet kur të dhënat nuk kanë ardhur nga një ligj i fiksuar matematik, por janë të përziera edhe me vlera rastësore (zhurma), ose në rastin kur numri i të dhënave është shumë i madh. Në këtë rasti numri M i qendrave të zgjedhura është më i vogël se numri i N i nyjeve. Nëse qendrat zgjidhen që të përputhen me pikat nyje, atëherë matrica e sistemit e gjeneruar nga funksionet me bazë radiale (si funksione bazë të funksionit përafrues) është jo singulare dhe për rrjedhojë problemi i përafrimit ka zgjidhje të vetme.

Për të ndërtuar modelin përafrues me anë të funksioneve me bazë radiale mund të përdoren edhe dy metoda me përshtatje që janë "Shto nyje" dhe "Largo nyje". Implementimi në MATLAB i metodës shto nyje është shumë më i shpejtë se algoritmi largo nyje, përderisa fillon me përafrimin e një numri të vogël pikash nyje. Më pas në çdo hap shtohet një nyje e re derisa gabimi maksimal të arrijë një tolerancë të caktuar. Në secilin hap duhet të zgjidhet një sistem ekuacionesh lineare, i cili do të ketë gjithmonë zgjidhje nëse qendrat përputhen me pikat nyje. Në rastin kur qendrat nuk përputhen me pikat nyje nuk ka përfundime të mirëfillta teorike për singularitetin e matricës së sistemit.

Një zbatim tjetër i funksioneve me bazë radiale në është në metodat pseudospektrale. Në veçanti, në zgjidhjen e ekuacioneve diferenciale me derivate të pjesshme që varen nga koha. Nga diskutimet e kryera në kapitullin e fundit mund të themi se nuk mund të sigurohet gjithmonë jo singulariteti i operatorit L_T . Që këtë rrjedh se, përfaqësja jo simetrike PS me funksione me bazë radiale justifikohet për EDP të varur nga koha (duke përdorur metoda eksplite për avancimin në kohë (time-stepping)). Në krahun tjetër, rasti i përfaqësjes simetrike në përgjithësi siguron zgjidhjen e problemit $\mathcal{L}u = f$ me anë të funksioneve me bazë radiale. Megjithatë, në përgjithësi nuk është i mundur as formulimi i operatorit diferencial diskret,

i cili nevojitet në rastin e metodave PS. Kjo na sugjeron përdorimin e përqasjes simetrike në rastin e EDP të pavarur nga koha dhe mundësisht në rastin e EDP të varur nga koha me metoda implicite për avancimin në kohë.

Aneksi A

Programi 4.5 *testfunction.m*

```
lineskip
1 % tf=testfunction(s,points)
2 % Llogarit testfunction
3 % prod_{d=1}^s x_d*(1-x_d) {Normalizuar qe maximumi i tij te jete 1}
4 % ne pika s-dimensionale
5 function tf=testfunction(s,points)
6 tf=4^s*prod(points.*(1-points),2);
```

Programi 4.6 *DistanceMatrix.m*

```
lineskip
1 % DM = DistanceMatrix(dsites, ctrs)
2 % Forms the distance matrix of two sets of points in R^s,
3 % i.e., DM(i,j) = || datasite_i - center_j ||_2.
4 % Input
5 %   dsites: Mxs matrix representing a set of M data sites in R^s
6 %           (i.e., each row contains one s-dimensional point)
7 %   ctrs:   Nxs matrix representing a set of N centers in R^s
8 %           (one center per row)
9 % Output
10 %   DM:     MxN matrix whose i,j position contains the Euclidean
11 %           distance between the i-th data site and j-th center
12 function DM = DistanceMatrix(dsites, ctrs)
13 [M,s] = size(dsites); [N,s] = size(ctrs);
14 DM = zeros(M,N);
15 % Accumulate sum of squares of coordinate differences
16 % The ndgrid command produces two MxN matrices:
17 %   dr, consisting of N identical columns (each containing
18 %       the d-th coordinate of the M data sites)
19 %   cc, consisting of M identical rows (each containing
20 %       the d-th coordinate of the N centers)
21 for d=1:s
22     [dr,cc] = ndgrid(dsites(:,d), ctrs(:,d));
23     DM = DM + (dr-cc).^2;
```

```

24     end
25     DM = sqrt(DM);

```

Programi 4.7 *MakeSDGrid.m*

```

lineskip
1  % gridpoints = MakeSDGrid(s,neval)
2  % Produces matrix of equally spaced points in s-dimensional unit cube
3  % (one point per row)
4  % Input
5  %   s:      space dimension
6  %   neval: number of points in each coordinate direction
7  % Output
8  %   gridpoints: neval^s-by-s matrix (one point per row,
9  %                               d-th column contains d-th coordinate of point)
10 function gridpoints = MakeSDGrid(s,neval)
11 if (s==1)
12     gridpoints = linspace(0,1,neval)';
13     return;
14 end
15 % Mimic this statement for general s:
16 % [x1, x2] = ndgrid(linspace(0,1,neval));
17 outputarg = 'x1';
18 for d = 2:s
19     outputarg = strcat(outputarg, ',x',int2str(d));
20 end
21 makegrid = strcat('[',outputarg,'] = ndgrid(linspace(0,1,neval));');
22 eval(makegrid);
23 % Mimic this statement for general s:
24 % gridpoints = [x1(:) x2(:)];
25 gridpoints = zeros(neval^s,s);
26 for d = 1:s
27     matrices = strcat('gridpoints(:,d) = x',int2str(d), '(:);');
28     eval(matrices);
29 end

```

Programi 4.8 *LinearScaling2D_CS.m*

Ky program është ndërtuar në mënyrë që të punojmë me funksionet peshues *compactly supported*, të cilët gjenerojnë matrica të rralla. Për këtë arsye duhen disa teknika që të përshpejtojnë zgjidhjen e sistemeve me matrica të tilla (me shumë zero). Në këtë program është përdorur struktura *kd-trees*, e cila është implementuar në MATLAB në një bashkësi MEX-file nga Guy Shchter dhe mund të shkarkohet nga MATLAB Central File Exchange [MCFE].

Në këtë program ndërtohet një *kd-tree* për funksionet peshues, për secilën pikë të llogaritjes. Si të dhëna hyrëse ka nevojë për **epoints** (një matricë $N \times s$, N të dhëna s —dimensionale), **ctrs** (një matricë $M \times s$, M qendra për funksionet peshuese në $\mathbb{R}^s$), **rbf** (emrin e një funksioni që përcakton funksionet peshues me bazë radiale) dhe **ep** (parametrin që përcakton supportin e funksioneve peshuese, ku suporti i funksioneve peshues jepet si $\frac{1}{ep}$). Si e dhënë dalëse merret një matricë **Phi** të rrallë $N \times M$ që përmban funksionet gjeneruese në pikat e dhëna nga **center** dhe të llogaritura në pikat **epoints**.

```

lineskip
1 % Phi = LinearScaling2D_CS(epoints , ctrs , rbf , ep)
2 % Forms a sparse matrix of scaled generating functions for MLS
3 % approximation with linear reproduction .
4 % Uses:          k-D tree package by Guy Shechter from
5 %               MATLAB Central File Exchange
6 function Phi = LinearScaling2D_CS(epoints , ctrs , rbf , ep)
7 [N,s] = size(epoints); [M,s] = size(ctrs);
8 alpha = [0 0; 1 0; 0 1; 1 1; 2 0; 0 2];
9 % Build k-D tree for centers
10 [tmp,tmp,Tree] = kdtree(ctrs ,[]);
11 % For each eval. point, find centers whose support overlap it
12 support = 1/ep; mu = zeros(6);
13 % Modify the following line for optimum performance
14 veclength = round(support*N*M/4);
15 rowidx = zeros(1,veclength); colidx = zeros(1,veclength);
16 validx = zeros(1,veclength);
17 istart = 1; iend = 0;
18 for i = 1:N
19     [pts ,dist ,idx] = kdrangequery(Tree ,epoints(i ,:) , support);
20     newlen = length(idx);
21     % Vector of basis functions
22     Phi_i = rbf(ep ,dist ');
23     % Compute all 6 moments for i-th evaluation point
24     for j=1:6
25         x_to_alpha = 1;
26         for coord=1:s
27             x_to_alpha = x_to_alpha .*( ctrs(idx ,coord) -...
28                 repmat(epoints(i ,coord),newlen ,1)).^ alpha(j ,coord);
29         end
30         mu(j) = Phi_i*x_to_alpha;
31     end
32     L1=(mu(4)^2-mu(5)*mu(6)); L2=(mu(2)*mu(6)-mu(3)*mu(4));
33     L3=(mu(5)*mu(3)-mu(2)*mu(4));
34     scaling = L1*repmat(1,newlen ,1) + ...
35         L2*(ctrs(idx ,1)-repmat(epoints(i ,1),newlen ,1))+...
36         L3*(ctrs(idx ,2)-repmat(epoints(i ,2),newlen ,1));
37     denom = mu(2)^2*mu(6)+mu(5)*mu(3)^2-mu(1)*mu(5)*mu(6)-...

```

```

38         2*mu(2)*mu(3)*mu(4)+mu(1)*mu(4)^2;
39     if (denom ~= 0)
40         scaling = scaling/denom;
41         iend = iend + newlen;
42         rowidx(istart:iend) = repmat(i,1,newlen);
43         colidx(istart:iend) = idx';
44         validx(istart:iend) = Phi_i.*scaling';
45         istart = istart + newlen;
46     end
47 end
48 filled = find(rowidx); % only those actually filled
49 Phi = sparse(rowidx(filled), colidx(filled), validx(filled), N, M);
50 % Free memory
51 clear rowidx colidx validx; kdtree([], [], Tree);

```

Për të llogaritur matricën diferenciale të rendit të parë, duhet pasur parasysh rregulli i derivimit zinxhir. Në këtë rast derivati i funksioneve me bazë radiale do të jetë i formës

$$\frac{\partial}{\partial \mathbf{x}} \varphi(\|\mathbf{x}\|) = \frac{x}{r} \frac{d}{dr} \varphi(r).$$

Programi 4.9 DRBF.m Matrica Diferenciale në Matlab

Për të llogaritur matricën diferenciale të rendit të parë, duhet pasur parasysh rregulli i derivimit zinxhir. Në këtë rast derivati i funksioneve me bazë radiale do të jetë i formës

$$\frac{\partial}{\partial \mathbf{x}} \varphi(\|\mathbf{x}\|) = \frac{x}{r} \frac{d}{dr} \varphi(r).$$

```

lineskip
1 % [D,x]=DRBF(N,rbf,dxrbf)
2 % Llogarit matricen diferenciale D per derivatet 1-D
3 % duke perdorur pika Chebyshev dhe Leave-one-out algoritim
4 % per te llogaritur nje vlere optimale te epsilon
5 % Vlera hyrese: N, krijo N+1 pika collation
6 %           rbf, dxrbf emrat e funksioneve per rbf dhe derivatin e tij
7 % Therret:   DistanceMatrix, DifferenceMatrix
8 % Kerkon:    CostEpsilonDRBF
9 function [D,x]=DRBF(N,rbf,dxrbf)
10 if N==0, D=0; x=1; return, end
11 x=cos(pi*(0:N)/N)'; % Pikat Chebyshev
12 mine=.1; maxe=10; % Intervali ku do merret eps optimal
13 r=DistanceMatrix(x,x);
14 dx=DifferenceMatrix(x,x);
15 ep = fminbnd(@(ep) CostEpsilonDRBF(ep,r,dx,rbf,dxrbf),mine,maxe);

```

```

16 fprintf('Eshte perdorur epsilon = %f\n',ep)
17 A=rbf(ep,r);
18 Ax=dxbf(ep,r,dx);
19 D=Ax/A;

```

Prandaj, përveç distancave, r , na duhen gjithashtu edhe diferencat në x , ku me x shënohet komponentja e parë e $\mathbf{x}$. Llogaritja e tyre kryhet përkatësisht në rreshtin 13 dhe 14. Funksioni *DifferenceMatrix.m* jepet në programin 4.10. Llogaritja e matricës diferenciale kryhet sipas barazimit $D = A_x A^{-1}$ dhe implementohet në kodin e mëposhtëm në rreshtat 17 – 19.

Programi 4.10 *DifferenceMatrix.m*

```

lineskip
1 % DM= DifferenceMatrix(datacoord,centercoord)
2 % Formon matricen diferenciale te dy vargjesh pikash ne R
3 % Shemb: DM(j,k)=datacoord_j-centercoord_k
4 function DM= DifferenceMatrix(datacoord,centercoord)
5 % Komanda ndgrid prodhon dy MxN matrica:
6 % dr, me N- shtylla identike (secila mban M te dhenat)
7 % cc, me M- rreshta identike (secili permбан N qendrat)
8 [dr,cc]=ndgrid(datacoord(:),centercoord(:));
9 DM=dr-cc;

```

Llogaritja e parametrin optimal të shkallëzimit ϵ , kryhet sipas metodës "Përjashtoj një të dhënë" të trajtuar në Kapitullin II. Kjo procedurë implementohet në programin 4.11 *CostEpsilonDRBF.m*, i cili mund të shkarkohet nga <http://www.math.unipd.it/~demarchi/TAA2010/>.

Programi 4.11 *CostEpsilonDRBF.m*

```

lineskip
1 function ceps=CostEpsilonDRBF(ep,r,dx,rbf,dxbf)
2 N=size(r,2);
3 A=rbf(ep,r);
4 rhs=dxbf(ep,r,dx)';
5 invA=pinv(A);
6 EF=(invA*rhs)./ repmat(diag(invA),1,N);
7 ceps=norm(EF(:));

```

Programi 4.12 *DRBF.m*

Ky program është i ngjashëm me programin 4.9 dhe llogarit matricën diferenciale të rendit të dytë.

```

lineskip
1 % [D2,x] = D2RBF(N,rbf,d2rbf)
2 % Computes the second-order differentiation matrix D2 for 1-D
3 % derivative using Chebyshev points and LOOCV for optimal epsilon
4 % Input: N, number of points -1
5 %         rbf, d2rbf, function handles for rbf and its derivative
6 % Calls on: DistanceMatrix, DifferenceMatrix
7 % Requires: CostEpsilonD2RBF
8 function [D2,x] = D2RBF(N,rbf,d2rbf)
9 if N==0, D2=0; x=1; return, end
10 x = cos(pi*(0:N)/N)'; % Chebyshev points
11 mine = .1; maxe = 10; % Shape parameter interval
12 r = DistanceMatrix(x,x);
13 ep = fminbnd(@(ep) CostEpsilonD2RBF(ep,r,rbf,d2rbf),mine,maxe);
14 fprintf('Using epsilon = %f\n', ep)
15 A = rbf(ep,r);
16 AD2 = d2rbf(ep,r);
17 D2 = AD2/A;

```

Referenca

- [1] Andrews, G. Askey, R. and Roy, R. (1999). Special functions. *Cambridge University Pres.*
- [2] Asimi (2016). Adaptive least square approximation for fitting scattered data. *IJSET - International Journal of Innovative Science, Engineering and Technology*, Vol. 3 Issue 2, ISSN 2348 – 7968.
- [3] Asimi, A. and Hanelli, L. (2016a). Scattered data least approximation using gaussians. stationary and non stationary approximation. 1. *The 5-th Advanced Research in Scientific Areas (ARSA-2016) 7 - 11 november, 2016.*
- [4] Asimi, A. and Hanelli, L. (2016b). Some numerical experiments for determining the optimal value of the scale parameter for meshfree interpolation. *International Journal of Innovative Research in Engineering and Management (IJIERM) Volume 3, Issue-3, May – 2016, ISSN: 2350-0557.*
- [5] Asimi, A. and Hanelli, L. (2016c). Stationary and non-stationary meshfree interpolation using radial basis functions. 2. *24th Conference on Applied and Industrial Mathematics - CAIM 2016, Craiova, Romania, September 15-18, 2016.*
- [6] Backus, G. Gilbert, F. (1968). The resolving power of gross earth data. *Geophys. J. R. Astr. Soc. (16), pp. 169-205.*
- [7] Baxter, B. J. C. (1991). Conditionally positive functions and p -norm distance matrices. *Constructive Approximation*, vol. 7, p. 427-440.
- [8] Bochner, S. (1932). Vorlesungen uber fouriersche integrale. *Akademische Analyse, Math. Ann. 108, pg. 378-410.*
- [9] Bochner, S. (1933). Monotone funktionen, stieltjes integrale und harmonische analyse. *Ann. of Math, 108, pg. 378-410.*
- [10] Bochner, S. (1941). Hilbert distances and positive definite functions. *Ann. of Math, vol. 2, pg. 647-656.*
- [11] Bozzini, M. Lenarduzzi, L. and Schaback, R. (2002). Adaptive interpolation by scaled multiquadrics. *Adv. in Comp. Math. 16, pp. 375-387.*
- [12] Cheney, E. W. and Light, W. (1999). A course in approximation theory, Brooks/Cole (Pacific Grove, CA).

- [13] Evgeniou, T. P. M. and Poggio, T. (2000). Regularization networks and support vector machines. *Adv. Comput. Math.* 13 1, pp.1-50.
- [14] Fasshauer, G. (1995a). Adaptive least square fitting with radial basis functions on the sphere. *Mathematical Methods for Curves and Surfaces*, M. Dehlen, T. Lyche, and L. Schumaker (eds), Vanderbilt University Press (Nashville), pp. 141-150.
- [15] Fasshauer, G. (2006). Meshfree approximation methods with matlab. *Interdisciplinary Mathematical Sciences*, pp. 198-203.
- [16] Fasshauer, G. (2007). Meshfree approximation methods with matlab. *World Scientific Publishing Co. Pte. Ltd*, pp.159-165.
- [17] Feller, W. (1966). An introduction to probability theory and its applications, vol. 2. *Wiley and Sons*, New York.
- [18] Fornberg, B. Wright, G. (1997). *Stable computation of multiquadrics interpolants for all values of the shape parameter*. *Comput. Math. Appl* 47, pp.497-523.
- [19] Franke, R. Salkauskas, K. (1985). Localization of multivariate interpolation and smoothing methods. *Paper presented at SIAM Geometric Modeling Conference*, Nashville.
- [20] Franke, R. (1982a). Scattered data interpolation: tests of some methods. *Math. Comp.* 48, pg. 181-200.
- [21] Franke, R. (1995). Paper presented at. *SIAM Geometric Modeling Conference in Nashville Nov. 1995*.
- [22] Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computations* 10, pp. 1455-1480.
- [23] Halton (1960). *On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals*. *Numer. Math.* 2, pp.84-90.
- [24] Hardy, R. (1971). Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res*, 76, pg. 1905-1915.
- [25] Hon, Y.C Schaback, R. (1997). *On nonsymmetric collocation by radial basis functions*. *Appl. Math. Comput.* 119, pp.177-186.
- [26] Kansa, E. (1992). Multiquadrics, a scattered data approximation scheme with applications to computational fluid dynamics, ii: Solution to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.* 19, pp.147-161.
- [27] Kansa, E. and Carlson, R. (1992). Improved accuracy of multiquadric interpolation using variable shape parameters. *Comput. Math. Appl.* 24, pp.99-120.
- [28] Lancaster, P. and Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Math. Comp.* 37, pg. 141-158.
- [29] Lyche, T. (1992). Knot removal for spline curves and surfaces,. *in Approximation Theory VII*, E. W. Cheney, C. Chui, and L. Schumaker (eds.), Academic Press (New York), pp. 207-226.

- [30] Micchelli (1986). Interpolation of scattered data: distane matrices and conditionally pozitivite definite functions. *Constr. Approx. Vol. 2*, pg. 11-22.
- [31] Quak, E. Sivakumar, N. and Ward, J. (1993). Least squares approximation by radial functions. *SIAM J. Numer. Anal. 24*, pp. 1043-1066.
- [32] Rippa, S. (1999). An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. in Comp. Math. 11*, pp. 193-210.
- [33] Schoenberg, I. J. (1938). Metric spaces and completely monotone functions. *Ann. of Math. 39*, pp. 811-841.
- [34] Shepard, D. (1968). A two dimensional interpolation function for irregularly spaced data. *Proc. 23rd Nat. Conf ACM*, pg. 517-524.
- [35] Sivakumar, N. and Ward, J. (1993). On the least squares fit by radial functions to multidimensional scattared data. *Numer. Math. 65*, pp. 219-243.
- [36] Trefethen, L. (2000a). *Spectral Methods in MATLAB*. SIAM Philadelphim PA.
- [37] Trefethen, L. N. (2000b). Spectral methods in matlab. *SIAM (Philadelphia, PA)*.
- [38] von Golitschek, M. and Schumaker, L. (1990). Data fitting by penalized least squares, in algorithms for approximation ii. *M.G Cox and J. C. Mason (eds.), Chapman Hall (London)*, pp. 210-227.
- [39] Wendland, H. (1999a). Meshless galerkin approximation using radial basis functions. *Math. Comp. 68*, pp 1521-1531.
- [40] Wendland, H. (1999b). Numerical solutions of variational problems by radial basis functions. *Approximation Theory IX, Vol.II Computational Aspects, Charles K. Chui and L. L. Schumaker, Vanderbilt University Press*, pp. 361-368.
- [41] Wendland, H. (2005a). Scattered data approximation. *Cambridge University Press*.
- [42] Widder, D. V. (1941). The laplace transform. *Princeton University Press*.
- [43] Wong, Luk W.S, H. P. (1997). *Samplingwith Hammersley and Halton points*. J. Graphics Tools 2, pp 9-24.

