

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

DISERTACION

PËR MARRJEN E GRADËS SHKENCORE

DOKTOR

Nënhapësirat Krylov dhe zbatime

Paraqitur nga:

M.Sc. Amanda Zeqiri

Udhëheqës shkencor:

Prof. Asoc. Eglantina Kalluçi

Tiranë, 2021

**UNIVERSITETI POLITEKNIK I TIRANËS
FAKULTETI I INXHINIERISË MATEMATIKE DHE
INXHINIERISË FIZIKE
DEPARTAMENTI I INXHINIERISË MATEMATIKE**

Disertacion
I
Paraqitur nga

M.Sc. Amanda Zeqiri

Për marrjen e gradës shkencore

DOKTOR

Programi i studimit : Analizë Numerike dhe Ekuacione Diferenciale

Tema : Nënhapësirat Krylov dhe zbatime

Udhëheqës Shkencor: Prof. Asoc. Eglantina Kalluçi

Mbrohet më datë 29. 09. 2021 para jurisë:

- | | |
|--|-------------------------|
| 1. Prof. Dr. Shkëlqim Kuka | Kryetar |
| 2. Prof. Dr. Lulzim Hanelli | Anëtar (oponent) |
| 3. Prof. Dr. Fatmir Hoxha | Anëtar (oponent) |
| 4. Prof. Dr. Akli Fundo | Anëtar |
| 5. Prof. Asoc. Majlinda Godolja | Anëtare |

PARATHËNIE

*Së pari, dua të shpreh mirënjohje të thellë për udhëheqësin shkencor **Prof. Asoc. Eglantina Kalluçi**, e cila ka qenë një burim konstant ndihme gjatë gjithë punës time kërkimore. Ajo më ka treguar mbështetje që në hapat e parë si studente dhe më pas si pedagoge. Faleminderit për kohën dhe energjinë që ke shpenzuar për ta bërë këtë disertacion sa më dinjitoz.*

Së dyti, dua të falënderoj kolegët në Departamentin e Inxhinierisë Matematike (UPT) dhe në Departamentin e Matematikës së Aplikuar (UT) që kanë kontribuar në formimin tim akademik.

Së fundmi, një falënderim i veçantë i shkon familjes time për dashurinë, kujdesin dhe sakrificat e tyre për edukimin tim dhe përgatitjen për të ardhmen. Prindërit dhe vëllai im kanë qenë gjithmonë model frymëzimi për mua.

PËRMBLEDHJE

Ky disertacion vë në dukje rëndësinë dhe përparësitë e përdorimit të nënhapësirave Krylov në fushën e Analizës Numerike. Disertacioni është menduar si një udhëzues për t'u njohur me konceptet bazë dhe të avancuara të metodave që bazohen në nënhapësirat Krylov si dhe me zbatimin e tyre në një gamë të gjerë problemesh nga jeta reale.

Në Kapitullin 1 trajtohen disa nocione që lidhen me nënhapësirën Krylov dhe prezantohen algoritmet e metodave më të rëndësishme për zgjidhjen e sistemeve lineare të rralla dhe të mëdha. Grupi i metodave Krylov bazuar në dy qasjet kryesore Arnoldi dhe Lanczos, janë trajtuar në versionet e tyre standarde, si dhe është paraqitur koncepti për rifillimin e tyre.

Konvergjencia e metodave iterative Krylov jo gjithmonë garantohet në varësi të specifikave të veçanta të problemeve ku ato zbatohen. Matrica e koeficienteve të sistemit përfaqësues të problemeve të mëdha reale, në shumicën e rasteve është e keq-kushtëzuar. Për këto arsye, lind nevoja e parakushtëzimit të këtyre sistemeve. Në Kapitullin 2 jepen përkufizimet e nevojshme për sigurimin e ekzistencës dhe ndërtimin e parakushtëzuesëve efektivë. Dy lloje parakushtëzimi janë trajtuar, përkatësisht ILU dhe IC, të cilat zbatohen duke përdorur teknika të ndryshme.

Në Kapitullin 3 bëhet analiza empirike e grupit të metodave të përmendura në Kapitullin 1, duke marrë në konsideratë katër raste. Secili rast përfaqësohet nga matrica me përmasa dhe rrallësi/dendësi të ndryshme. Sistemet e përdorura janë simetrike ose jo, të mirë-kushtëzuara ose jo. Kjo pjesë e punimit ofron një paketë të algoritmeve të metodave Krylov të implementuara në Matlab. Këto algoritme (kode) janë modifikuar në mënyrë që përdoruesi të bëjë një analizë sa më të gjerë dhe ta ketë sa më të lehtë zgjedhjen e një teknike parakushtëzimi të duhur bazuar në vetitë e sistemit përkatës.

Për të vënë në dukje gamën e gjerë të zbatimeve të nënhapësirave Krylov, Kapitulli 4 paraqet disa zbatime në probleme me të dhëna reale të mëdha, si analiza e rrjetave komplekse (rrjetat biologjike etj.) apo zvogëlimi i përmasave në fushën e përpunimit të imazheve apo vlerësimit të gjeneve të rëndësishme në biomjekësi. Në këtë kapitull gjithashtu vihet në dukje epërsia e metodave të nënhapësirave Krylov në implementim dhe në menaxhim e kujtesës, krahasuar me metodat klasike ekzistuese në të njëjtat fusha.

Së fundmi, në këtë disertacion paraqitet një tjetër avantazh i metodave të nënhapësirave Krylov, i cili është lehtësia e paralelizimit të tyre. Kjo lehtësi vjen nga ndërtimi i algoritmeve të tyre bazuar në një grup veprimesh matematikore lehtësisht të paralelizueshme si prodhimi matricë-vektor, përditësimi i vektorëve, veprime vektor me vektor etj. Kapitulli 5 jep bazat e një pune të mëtejshme që mund të bëhet për përpunimin e shpejtë në paralel të të dhënave të rralla e të mëdha me anë të metodave Krylov.

Fjalë kyçe: *Nënhapësira Krylov; Sisteme lineare; Matrica të rralla; Parakushtëzimi; Iteracioni Arnoldi; Iteracioni Lanczos; Rrjeta komplekse; Përpunim imazhi; Paralelizimi.*

ABSTRACT

This dissertation paper emphasizes the importance and advantages of using the Krylov subspaces in the field of Numerical Analysis. The dissertation is intended as a handbook to introduce the basic and advanced concepts on Krylov subspace-based methods, as well as their application to a wide range of real-life problems.

Chapter 1 presents some notions related to the Krylov subspace and the algorithms of the most important methods for solving rare and large linear systems. The group of Krylov methods based on the two main approaches Arnoldi and Lanczos, are used in their standard version, and also the concept for the restart process is presented.

The convergence of Krylov iterative methods is not always guaranteed depending on the specific properties of the problems where they are applied. The coefficient matrix of the representative system for large real problems, in most cases is ill-conditioned. For these reasons, it is needed to apply preconditioning to these systems. Chapter 2 provides the necessary definitions to ensure the existence and construction of effective preconditioners. Two types of preconditioning have been addressed, respectively ILU and IC, which are applied using different techniques.

In Chapter 3 the empirical analysis is conducted for the group of methods mentioned in Chapter 1, considering four cases. Each case is represented by matrices of different sizes and different sparsity/density. The systems used are symmetric or asymmetric, well-conditioned or ill-conditioned. This part of the paper provides a package of the algorithms for Krylov methods, implemented in Matlab. These algorithms (source codes) are modified so that the user can make a wider analysis and make the decision of choosing a proper preconditioning easier based on the properties of the corresponding system.

To highlight the wide range of Krylov subspace applications, Chapter 4 presents some applications on large real data problems, such as complex network analysis (biological network etc.) or dimension reduction in image processing or evaluation of important genes in bio-medicine. This chapter also highlights the predominance of Krylov subspace methods in implementation and memory storage, compared to existing classical methods in the same fields.

Finally, in this dissertation another advantage of Krylov subspace methods is presented, i.e., they can be easily parallelized. This advantage comes from the construction of their algorithms based on a set of easily parallelizable mathematical operations such as matrix-vector product, vector update, vector-to-vector operations, and so on. Chapter 5 provides the basis for further work on faster parallel processing of sparse and large data using Krylov methods.

Keywords: *Krylov subspace; Linear systems; Sparse matrices; Preconditioning; Arnoldi iteration; Lanczos iteration; Complex networks; Image Processing; Parallelization.*

PËRMBAJTJA E LËNDËS

PARATHËNIE	iii
PËRMBLEDHJE	iv
LISTA E FIGURAVE	vii
LISTA E TABELAVE	ix
HYRJE: Nënhapësirat Krylov	x
KAPITULLI 1	
Metoda për zgjidhjen e sistemeve lineare.....	1
1.1 Iteracioni Arnoldi.....	1
1.2 Iteracioni Lanczos për matricat simetrike.....	4
1.3 Iteracioni Lanczos për matricat jo-simetrike.....	7
1.4 Procesi i rifillimit.....	11
KAPITULLI 2	
Parakushtëzimi.....	13
2.1 Faktorizimi ILU.....	15
2.1.1 ILU(0) me plotësim zero.....	18
2.1.2 ILU(p) me nivel plotësimi p.....	19
2.2 Faktorizimi Cholesky jo i plotë (IC).....	20
KAPITULLI 3	
Analiza empirike e metodave të nënhapësirës Krylov.....	22
3.1 Rasti I.....	23
3.2 Rasti II.....	25
3.3 Rasti III.....	26
3.4 Rasti IV.....	29
KAPITULLI 4	
Zbatime të nënhapësirave Krylov në të dhëna reale.....	32
4.1 Analiza e rrjetave komplekse.....	33
4.2 Zvogëlimi i përmasave.....	40
4.2.1 Vlerësimi i sfondit në imazhe.....	41
4.2.2 Vlerësimi i gjeneve të rëndësishme.....	43
KAPITULLI 5	
Puna e mëtejshme.....	45
PËRFUNDIME	52
BIBLIOGRAFIA	55
SHTOJCA A	58
SHTOJCA B	68

LISTA E FIGURAVE

Figura 3.1	Matricë me 1666 elemente të ndryshme nga zero nga 244,036.....	23
Figura 3.2	Matricë me 298710 elemente të ndryshme nga zero.....	25
Figura 3.3	Matricë me 28831 elemente të ndryshme nga zero nga 18,088,009.....	26
Figura 3.4	GMRES me rifillim për Rastin III.....	27
Figura 3.5	Saktësia e metodave me teknika të ndryshme parakushtëzimi.....	27
Figura 3.6	Mbetjet relative për teknika të ndryshme parakushtëzimi.....	28
Figura 3.7	Rrjeta për nivel 5 rafinimi, me 4096 trekëndësha dhe 2113 nyje.....	30
Figura 3.8	Rrjeta për nivel 7 rafinimi, me 65536 trekëndësha dhe 33025 nyje.....	30
Figura 3.9	Majtas matrica 2113×2113 me 13769 elemente të ndryshme nga zero për nivel 5 rafinimi dhe në të djathtë matrica 33,025×33,025 me 227081 elemente të ndryshme nga zero për nivel 7 rafinimi.....	31
Figura 4.1	Rrjeta Bio-SC-Protein.....	36
Figura 4.2	Matrica e fqinjësisë për Bio-SC-Protein.....	36
Figura 4.3	Shpërndarja probabilitare e kulmeve të rrjetës FB Harward Friendship.....	39
Figura 4.4	Matrica e Laplasianit për rrjetën e bashkëpunimit shkencor Sc_collab.....	39
Figura 4.5	Bashkësia e të dhënave CaVignal.....	42
Figura 4.6	Bashkësia e të dhënave Foliage.....	42
Figura 4.7	Bashkësia e të dhënave HighwayII.....	42
Figura 4.8	Komponentët e rëndësishëm Scree Plot.....	43
Figura 4.9	Përqindja e gjeneve në dy komponentët kryesorë, për nivele të ndryshme pavarësie.....	44
Figura 5.1	Procesor me njësi funksionale të shumëfishta.....	46
Figura 5.2	Teknika Pipeline.....	46
Figura 5.3	Procesori vektorial.....	47
Figura 5.4	Hapësira e përbashkët e të dhënave.....	48
Figura 5.5	Skema paralele e përditësimit të vektorëve.....	49
Figura 5.6	Skema paralele e prodhimit skalar dhe normës matricore $\ \cdot \ _2$	49
Figura 5.7	Skema paralele e veprimeve vektor me vektor.....	49

Figura 5.8	Prodhimi matricë-vektor paralel duke përdorur prodhimin e brendshëm.....	50
Figura 5.9	Caktimi i ndryshoreve në procesorët përkatës sipas një grafi ndarës.....	51

LISTA E TABELAVE

Tabela 3.1	Metodat e rifilluara me parametër 1 deri në 10.....	24
Tabela 3.2	Teknika parakushtëzimi për Rastin I.....	24
Tabela 3.3	Tabelë përmbledhëse për Rastin III.....	28
Tabela 3.4	Gabimet për metodën QMR me parakushtëzim.....	31
Tabela 4.1	Informacion i përgjithshëm mbi rrjetat.....	35
Tabela 4.2	Koeficientet e korrelacionit për të gjithë matësit e qendërsisë që janë përdorur për rrjetën Bio-SC-Protein.....	37
Tabela 4.3	Koeficientet e korrelacionit për secilën rrjetë dhe për listën e të gjithë kulmeve, vlerësuar nëpërmjet elementeve të diagonales dhe shumës së rreshtave të e^A	37
Tabela 4.4	Koha e përafrimit të e^A me metodën e rifilluar të Lanczos kundrejt metodës SSM.....	38
Tabela 4.5	Indeksi Estrada i normalizuar, lidhshmëria totale e normalizuar dhe $e^{\lambda_{max}}$	38
Tabela 4.6	Informacion i përgjithshëm në lidhje me bashkësinë e imazheve SBI...	41
Tabela 4.7	PSNR për secilën metodë dhe për secilën bazë të të dhënave.....	42

HYRJE

Nënhapësirat Krylov

Bazuar në raportin “Zhvillimi dhe zbatimi i shkencës dhe inxhinierisë në shekullin e 20-të”, metodat e nënhapësirës Krylov konsiderohen si një nga 10-të klasat më të rëndësishme të metodave numerike për zgjidhjen e sistemeve lineare algjebrike [1]. Në vitin 1931 një punim nga shkencëtari rus Aleksei Nikolaevich Krylov, paraqiti mënyrën se si vargjet e trajtës $\{v, Av, A^2v, \dots\}$, mund të përdoren për të ndërtuar polinomin karakteristik të një matrice. Kujtojmë që për një matricë katrore A vlera vetjake λ (skalarë) do të quhen rrënjët e polinomit karakteristik $\det(A - \lambda I) \equiv p(\lambda) = 0$ dhe vektorë vetjakë përkatës $v \neq 0$ të tillë që $Av = v\lambda$. Matrica I është matrica identike $n \times n$ dhe $\det(\cdot)$ është përcaktori i matricës. Polinomi karakteristik $p(\lambda)$ është një polinom i shkallës n dhe bashkësia e vlerave vetjake të A do të quhet spektër i saj $\sigma(A) := \{\lambda \in \mathbb{C} : \lambda \text{ është një vlerë vetjake e } A\}$. Gjithashtu, supozohet pavarësia lineare e n vektorëve vetjakë.

Nënhapësirat Krylov kanë nisur fillimisht si zgjidhës të problemeve të mëdha të vlerave vetjake pasi shumë probleme të inxhinierisë dhe jo vetëm, fokusohen dhe kërkojnë llogaritjen vetëm të vlerës vetjake më të madhe apo vlerës vetjake më të vogël të një matrice katrore. Shpesh në praktikë kërkohen të vlerësohen k -vlerat vetjake më të mëdha apo më të vogla, së bashku me vektorët vetjakë përkatës. Po të marrim shembullin e një salle koncertesh, vlera vetjake më e vogël e ekuacionit akustik zbulon frekuencën më të ulët të rezonancës.

Këto metoda janë veçanërisht të përshtatshme edhe në zgjidhjen e sistemeve lineare të rralla dhe të mëdha $Ax = b$, të cilat hasen në mjaft zbatime në shkencë dhe inxhinieri. Të rralla do të thotë se shumica e elementeve të matricës janë zero. Ndërsa probleme të përmasave shumë të mëdha fitohen zakonisht nga diskretizimi i EDDP-ve (Ekuacione diferenciale me derivate të pjesshme) me Metodën e Elementeve të Fundme (Finite Element Method) ose me Metodën e Diferencave të Fundme (Finite Difference Method). Në rastin kur problemi origjinal është jolinear, linearizimi mund të bëhet me anë të metodës së Njutonit ose një metodë të tipit Njuton. Sistemet e rralla mund të zgjidhen me të ashtuquajturit zgjidhës të drejtpërdrejtë të rrallë, që janë varianti i modifikuar i eliminimit të Gauss-it për rastet e sistemeve të rralla, ose me anë të metodave iterative. Metodot iterative zëvendësojnë sistemin e dhënë me një sistem të përafërt që është më i thjeshtë për tu zgjidhur. Pra, në vend të $Ax = b$, zgjidhim një sistem më të thjeshtuar $Kx_0 = b$, ku x_0 është një përafrim fillestar për zgjidhjen reale x të sistemit. Numri i iteracioneve rritet në mënyrë sistematike me qëllim që përafrimi të mund të përmirësohet. Qëllimi është të përcaktojmë vektorin z që plotëson kushtin $A(x_0 + z) = b$. Nga këtu, krijohet një sistem i ri linear $Az = b - Ax_0$. Përsëri, zgjidhim këtë sistem në bazë të një sistemi të përafërt, ku K përafrohet prej $Kz_0 = B - Ax_0$ nga ku marrim përafrim $x_1 = x_0 + z_0$. Procedura e përmirësimit përsëritet për x_1 duke krijuar kështu një metodë iterative.

Për të zgjidhur një sistem linear të përmasave të mëdha, metodat e njohura klasike si eliminimi Gauss ose iteracione të tjera të thjeshta si ai i Jakobit nuk janë aq të dobishme. Për më tepër, në shumë raste matricat përfaqësuese të sistemit mund të jenë të papërcaktuara dhe josimetrike. Algoritmet e nënhapësirës Krylov nuk kufizohen vetëm në përafrimin e zgjidhjes së sistemeve të ekuacioneve lineare apo vlerësimin e vlerave vetjake të problemeve me matrica të përmasave të mëdha. Ato mund të përdoren dhe për përafrimin e funksioneve eksponencialë apo racionalë të matricave të rralla e të mëdha. Këto metoda janë një klasë e madhe e algoritmeve iterative që punojnë me matrica dhe vektorë të fundmë në aritmetikë reale dhe komplekse.

Nga ana matematikore, këto metoda bazohen në metodat e projekcionit. Ideja kryesore e nënhapësirës Krylov dhe në përgjithësi e një procesi projekcioni, është gjetja e një zgjidhje të përafërt të një sistemi potencialisht shumë të madh $Ax = b$ duke zgjidhur një sistem me përmasa shumë më të vogla. E njëjta ide përdoret edhe për hartimin e një numër algoritmesh efektive për problemet e mëdha të vlerave vetjake. Në të gjitha trajtat e algoritmeve Krylov, është e nevojshme të llogaritet prodhimi i brendshëm i vektorëve të përmasave të mëdha dhe prodhimi matricë-vektor, i cili shpesh herë mund të përfshijë zbatimin e parakushtëzuesit, i cili gjithashtu mund të rezultojë në zgjidhjen e një sistemi linear me përmasa të mëdha.

Le të jetë $Ax = b$ një sistem linear ku $A \in C^{n \times n}$ matrica e përgjithshme e sistemit, $b \in C^n$ vektori i termave të lirë, x_0 një përafrim fillestar i zgjidhjes së këtij sistemi dhe $r_0 = b - Ax_0$ përafrimi fillestar i mbetjes së tij. Procesi iterativ bazuar në nënhapësirën Krylov prodhon vargun e zgjidhjeve të përafërta x_m me anë të:

$$x_m \in x_0 + \mathcal{K}_m(A, r_0) \quad (1)$$

ku $\mathcal{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}$ është nënhapësira e m -të Krylov e prodhuar nga matrica A prej vektorit r_0 dhe $m < \text{rank}(A)$.

Meqenëse $\mathcal{K}_m \subseteq C^n$ ose në rastin e të dhënave reale $\mathcal{K}_m \subseteq R^n$ dhe bazuar në (1), për vektorin e m -të të mbetjes $r_m = b - Ax_m$ kemi:

$$r_m \in r_0 + A\mathcal{K}_m(A, r_0) \subset \mathcal{K}_{m+1}(A, r_0) \quad (2)$$

Qëllimi është të gjendet zgjidhja e përafëruar me një numër sa më të vogël iteracionesh dhe sa më afër zgjidhjes reale x , duke qenë se kujtesa kompjuterike në dispozicion është e kufizuar [2]. Gjithashtu synohet të përafrohet r_m sa më i vogël që të jetë e mundur nëpërmjet elementeve të $A\mathcal{K}_m$. Kjo nënkupton që duhet të zgjedhim r_m si pingulja r_0 në projektimin ortogonal të saj në $A\mathcal{K}_m$, bazuar në qasjen e mbetjes me normë minimale [3] pra $\min \|b - Ax_m\|_2$. Kjo qasje është baza e disa metodave Krylov për sistemet Hermitiane ($A = A^H$) dhe jo-Hermitiane (si GMRES) dhe r_m përafrohet bazuar në:

$$r_m \in r_0 + A\mathcal{K}_m(A, r_0) \text{ të tilla që } r_m \perp A\mathcal{K}_m \quad (3)$$

Disa metoda të tjera të bazuara në nënhapësirën Krylov formulohen në varësi të llojeve të tjera projektimesh ortogonale apo jo të drejtpërdrejta. Si për shembull:

$$r_m \in r_0 + A\mathcal{K}_m(A, r_0) \text{ të tilla që } r_m \perp \mathcal{K}_m \quad (4)$$

për metodën e Gradientit të Konjuguar (CG). Apo bazuar në projektionin:

$$r_m \in r_0 + A\mathcal{K}_m(A, r_0), \text{ të tilla që } r_m \perp \widetilde{\mathcal{K}}_m \quad (5)$$

në rastin e metodës së Gradientit të Bi-konjuguar (BiCG), ku $\tilde{r}_0 \in \mathcal{C}^n$ është një vektor i zgjedhur në mënyrë arbitrare dhe $\widetilde{\mathcal{K}}_m = \mathcal{K}_m(A^H, \tilde{r}_0)$ është nënhapësira e m -të Krylov formuar nga A^H me vektorin mbetje \tilde{r}_0 .

Procedura themelore e ndërtimit të bazave ortogonale në nënhapësirat Krylov është algoritmi i njohur Gram-Schmidt (klasik apo i modifikuar) ose algoritme të tjera si algoritmi Householder apo algoritmi Givens, për një matrice A me përmasa $n \times n$ në formën e një matrice Hessenberg.

Dy qasjet bazë për metodat e përgjithshme të nënhapësirës Krylov janë iteracioni i Arnoldit (dhe variacione të tij) dhe iteracioni Lanczos (për matrice Hermitiane dhe jo-Hermitiane) [4].

Iteracioni Arnoldi është më i përdoruri në hapësirën Krylov, i cili u prezantua si një mjet për të zvogëluar një matricë të dendur me një transformim unitar në formën e matricës Hessenberg. Ky iteracion përdor projektionin ortogonal mbi \mathcal{K}_m për matricat e përgjithshme jo Hermitiane dhe zgjedh elementet e kësaj matrice h_{ij} në mënyrë që baza të jetë ortonormale. Vlerat vetjake të matricës Hessenberg mund të sigurojnë përafrime të sakta me vlerat vetjake të matricës origjinale. Më poshtë është paraqitur Iteracioni Arnoldi:

- Hapi 1. Zgjidhet vektori fillestar v_1 i tillë që $\|v_1\|^2 = 1$
- Hapi 2. Për $j = 1, 2, \dots, m$ kryej
- Hapi 3. $w_j = Av_j$
- Hapi 4. Për $i = 1, 2, \dots, j$
 - Hapi 5. $h_{ij} = (w_j, v_i)$
 - Hapi 6. $w_j = w_j - h_{ij}v_i$
- Hapi 7. Fund
- Hapi 8. $h_{j+1,j} = \|w_j\|_2$
- Hapi 9. Në qoftë së $h_{j+1,j} = 0$ atëherë Ndalo
- Hapi 10. $v_{j+1} = w_j / h_{j+1,j}$
- Hapi 11. Fund

Ndërtimi i një baze ortonormale $\{v_1, \dots, v_m\}$ të \mathcal{K}_m bëhet duke përdorur procedurën e modifikuar të ortogonalizimit Gram-Schmidt ose ndonjë procedurë tjetër ortonormalizimi ekuivalente mbi vektorët Krylov. Vektorët ortonormal v_j të fituar nga iteracioni i Arnoldit mbajnë emrin e tij. Në secilin hap, algoritmi shumëfishon vektorin e mëparshëm Arnoldi v_j nga A dhe më pas ortonormalizon w_j kundrejt të gjitha v_j –ve të mëparshme me një procedurë të modifikuar të Gram-Schmidt. Nëse në hapin e m -të kemi $w_m = 0$ algoritmi ndalet. Në këtë rast, rreshti i fundit i matricës Hessemberg H_m është zero [5].

Iteracioni Lanczos, prezantuar nga Cornelius Lanczos, është adaptim i metodës së fuqisë për të gjetur përmasën m të nënhapësirës Krylov me prirje drejt vlerave vetjake

ekstremale dhe vektorëve vetjakë përkatës, të një matrice Hermitiane $n \times n$. Ky iteracion është një përshtatje e iteracionit Arnoldi për rastin e matricave simetrike, ku matrica Hessenberg H_m shndërrohet në një matricë tridiagonale simetrike. Iteracioni Lanczos ndërton një bazë ortonormale Krylov për sistemet Hermitiane. Më poshtë është paraqitur Iteracioni Lanczos [5]:

Hapi 1. Zgjidhet vektori fillestar $v_1 = r_0 / \|r_0\|_2$, vendos $\beta_1 = 0, v_0 = 0$

Hapi 2. Për $j = 1, 2, \dots, m$ kryej

Hapi 3. $w_j = Av_j - \beta_j v_{j-1}$

Hapi 4. $\alpha_j = t_{jj} = (w_j, v_j)$

Hapi 5. $w_j = w_j - \alpha_j v_j$

Hapi 6. $\beta_{j+1} = t_{j,j-1} = \|w_j\|_2$

Hapi 7. Në qoftë se $\beta_{j+1} = 0$ atëherë Ndal

Hapi 8. $v_{j+1} = w_j / \beta_{j+1}$

Hapi 9. Fund

Iteracioni Lanczos e transformon problemin e vlerave vetjake të A në problemin e vlerave vetjake të T_m , i cili zgjidhet duke zbatuar zbërthimin QR mbi matricën tridiagonale simetrike. Në këtë mënyrë, vlera vetjake më e vogël e T_m korrespondon me vlerën vetjake më të vogël të matricës A .

KAPITULLI 1

Metoda për zgjidhjen e sistemeve lineare

1.1 Iteracioni Arnoldi

Algoritmi i iteracionit Arnoldi i paraqitur në hyrje të punimit mund të përdoret për të përafuar vlerat vetjake, duke e kthyer problemin fillestar në atë të gjetjes të vlerave vetjake të një matrice të sipërme Hessenberg. Çifti (λ_j, u_j) përmban përkatësisht vlerat vetjake dhe vektorët vetjakë për matricat e reduktuara $H_m = V_m^T A V_m$, kështu më pas zgjidhen r çiftet Ritz (λ_j, x_j) ku $X = V u$, si përafrimet e çiftit vetjak të dëshiruar.

I njëjti algoritëm përshtatet edhe për zgjidhjen e sistemeve të rralla e të mëdha duke përdorur teknikën Ritz-Galerkin e cila merr vektorin $v_1 = r_0/\beta$ dhe duke vendosur $\beta = \|r_0\|_2$, ku $r_0 = b - Ax_0$. Për rrjedhojë kemi $V_m^T r_0 = V_m^T (\beta v_1) = \beta e_1$ dhe $y_m = H_m^{-1}(\beta e_1)$. Tani zgjidhja e sistemit linear përafrohet nga (1.1):

$$x_m = x_0 + V_m y_m \quad (1.1)$$

Një nga metodat e para bazuar në këtë qasje është Metoda e Ortogonalizimit të Plotë (Full Orthogonalization Method) ose siç do ti referohemi shkurt FOM. Algoritmi i kësaj metode jepet më poshtë [6].

Algoritmi FOM:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, dhe $v_1 = r_0 / \beta$

Hapi 2. Vendoset $H_m = 0$

Hapi 3. Për $j = 1, 2, \dots, m$ kryej

Hapi 4. $w_j = A v_j$

Hapi 5. Për $i = 1, 2, \dots, j$ kryej

Hapi 6. $h_{i,j} = (w_j, v_i)$

Hapi 7. $w_j = w_j - h_{i,j} v_i$

Hapi 8. Fund

Hapi 9. $h_{j+1,j} = \|w_j\|_2$

Hapi 10. Në qoftë se $h_{j+1,j} = 0$ atëherë $m = j$ dhe kalo në hapin 13

Hapi 11. $v_{j+1,j} = w_j / h_{j+1,j}$

Hapi 12. Fund

Hapi 13. $y_m = H_m^{-1}(\beta e_1)$ dhe zgjidhja e përafërt $x_m = x_0 + V_m y_m$

Vektori i mbetjes r_m për përafrimin x_m me FOM është:

$$r_m = -h_{m+1,m} e_m^T y_m v_{m+1} \quad (1.2)$$

dhe norma dy e tij është:

$$\|r_m\|_2 = h_{m+1,m} |e_m^T y_m| \quad (1.3)$$

Sipas FOM rreshti i $(i + 1)$ –të i matricës trekëndore të sipërme H_m nuk merret parasysh. Ka disa metoda të tjera të cilat janë matematikisht të njëjta me FOM si për shembull metoda Orthores dhe metoda GENCG. Për këto metoda ashtu si FOM, zgjidhja e përafërt ndërtohet në mënyrë të tillë që të rezultojë në mbetje ortogonale. Metoda FOM është e lidhur ngushtë me metodat e ekstrapolimit vektorial kur ajo zbatohet mbi vargje vektoriale të formuara në mënyrë lineare.

Kosto e veprimeve të FOM është përafërsisht $2(m \times NZ(A) + m^2n)$ ku $NZ(A)$ është numri i elementeve të ndryshëm nga zero për matricën A dhe m numri i iteracioneve të Arnoldit për \mathcal{K}_m . Procedura Arnoldi kërkon në çdo hap një prodhim matricë-vektor dhe zbaton një ortogonalizim të Gram-Schmidt, me kosto në total të veprimeve, përkatësisht $2m \times NZ(A)$ dhe $2m^2n$. Duke qenë se m është shumë herë më e vogël se n , kosto e veprimeve dhe e hapësirës kompjuterike është më e lartë në iteracionin e parë. Procesi i ruajtjes në kujtesë për metodën FOM përfshin ruajtjen e shtyllave të matricës së vektorëve ortogonalë V_m , vektorët e përafërimeve të përkohshme të zgjidhjes dhe termave të lirë, vektorëve të prodhimeve matricë-vektor dhe si përfundim ruajtja e matricës Hessenberg H_m . Përafërsisht kosto totale për sa u përmendën është $\frac{m^2}{2} + n(m + 3)$.

Metoda e Përgjithshme e Mbetjeve Minimale (Generalized Minimum Residual Method) ose shkurt GMRES u zhvillua nga Y. Saad dhe M. H. Schultz në vitin 1986. Kjo metodë është më e njohura e nënhapësirës Krylov dhe trajta më e njohur e saj bazohet në algoritmin e modifikuar të Gram-Schmidt. Metoda *GMRES* është e njohur për zgjidhjen e një sistemi të rrallë të ekuacioneve lineare $Ax = b$, veçanërisht, kur një zgjidhje e përafërt është e përshtatshme. Për zgjidhjen e sistemeve lineare jo-simetrike, kjo metodë konsiderohet të jetë e qëndrueshme, sidoqoftë numri i veprimeve rritet mjaft me numrin e iteracioneve. *GMRES* përfshin një matricë trekëndore të matricës Hessenberg të gjeneruar nga procesi Arnoldi, e cila lejon marrjen e kufijve që tregojnë se norma e mbetjes zvogëlohet në zero kur vlera më e vogël singulare e kësaj matricë trekëndore shkon në zero [5].

Metoda përafëron zgjidhjen $x_m = x_0 + V_m y_m$ me anë vektorëve të nënhapësirës Krylov me mbetje minimale $r_m = b - Ax_m = V_{m+1}(\beta e_1 - \bar{H}_m y_m)$ ku \bar{H}_m është matrica Hessenberg me përmasa $(m + 1) \times m$. Duke qenë se vektorët e matricës V_{m+1} janë ortonormalë atëherë:

$$\|r_m\|_2 = \|b - Ax_0\|_2 = \|\beta e_1 - \bar{H}_m y_m\|_2 \quad (1.4)$$

Për të zgjidhur problemin e minimizimit të (1.4) përdoret zakonisht rrotullimi i Givens që transformon matricën Hessenberg në një matricë trekëndore të sipërme [6]. Më pas përafërimit x_m mund të përftohen të tilla që:

$$y_m = \arg \min_y \|\beta e_1 - \bar{H}_m y\|_2 \quad (1.5)$$

ku minimizuesi y_m kërkon zgjidhjen e një problemi të katrorëve më të vegjël me përmasa $(m + 1) \times m$. Më pas matrica katrore H_m fitohet prej \bar{H}_m duke rrotulluar rreshtin e fundit të saj në një vektor zero me anë të procedurës.

Algoritmi GMRES:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ dhe $v_1 = r_0 / \beta$

Hapi 2. Për $j = 1, 2, \dots, m$ kryej

Hapi 3. $w_j = Av_j$

Hapi 4. Për $i = 1, 2, \dots, j$ kryej

Hapi 5. $h_{ij} = (w_j, v_i)$

Hapi 6. $w_j = h_{ij}v_i$

Hapi 7. Fund

Hapi 8. $h_{j+1,j} = \|w_j\|_2$

Hapi 9. Në qoftë se $h_{j+1,j} = 0$, vendos $m = j$ dhe kalo në hapin 12

Hapi 10. $v_{j+1} = w_j / h_{j+1,j}$

Hapi 11. Fund

Hapi 12. Përcaktohet $\bar{H}_m = \{h_{ij}\}$ me përmasa $(m + 1) \times m$
për $1 \leq i \leq m + 1$ dhe $1 \leq j \leq m$

Hapi 13. Zgjidh problemin e katrorëve më të vegjël $\min \|\beta e_1 - \bar{H}_m y\|_2$

Hapi 14. Llogarit zgjidhjen e përafërt $x_m = x_0 + V_m y_m$ ku $y_m = H_m^{-1}(\beta e_1)$

Nëse shënojmë me $\|r_k^{FOM}\|_2$ dhe $\|r_k^{GMRES}\|_2$ normat e mbetjeve të metodave përkatëse FOM dhe GMRES në hapin k , si dhe $\|r_{min}^{FOM}\|_2$ normën më të vogël të mbetjeve të marra në k hapat e parë të FOM, është i vërtetë relacioni i mëposhtëm [3]:

$$\|r_k^{GMRES}\|_2 \leq \|r_{min}^{FOM}\|_2 \leq \sqrt{k} \|r_k^{GMRES}\|_2 \quad (1.6)$$

Nëse në hapin k të GMRES $x_k^{GMRES} = x_{k-1}^{GMRES}$ atëherë H_k është unitare dhe FOM ndërpritet pra nuk përditëson x_k^{FOM} . Algoritmi GRMES ndërpritet në hapin k për një matricë A të kthyeshme të sistemit, që do të thotë $h_{k+1,k} = 0$ atëherë dhe vetëm atëherë kur x_k është zgjidhja e saktë x . Nëse matrica A është Hermitiane por jo detyrimisht pozitivisht e përcaktuar atëherë matrica H_m reduktohet në një matricë tridiagonale. Pavarësisht përafrimit të një zgjidhjeje me mbetje minimale, GMRES ka mangësi faktin që me rritjen e numrit të iteracioneve rritet në mënyrë lineare edhe numri i veprimeve [7]. Për më tepër të gjithë vektorët e bazës së nënhapësirës Krylov duhet të ruhen në kujtesë. Kosto e ekzekutimit dhe e kujtesës për GMRES janë përkatësisht $\mathcal{O}(k^2n)$ dhe $\mathcal{O}(kn)$ për një numër k iteracionesh. Versione të tjera të GMRES janë propozuar por që rezultojnë në procedura shtesë ndihmëse dhe si përfundim kanë të njëjtën kosto për

iteracion [8]. Varianti i thjeshtuar i saj i paraqitur nga H. F. Walker and L. Zhou (1994) ka përparësi ndaj algoritmit të mësipërm vetëm në rastin kur m është shumë e vogël.

GMRES mund të përdoret edhe për sistemet komplekse duke përdorur të njëjtën qasje si më sipër. Shpeshherë zbatime të ekuacioneve diferenciale të ndryshme çojnë në nevojën e zgjidhjes së sistemeve komplekse, si në rastin e ekuacioneve Maxwell. Algoritmi GMRES për rastin e sistemeve komplekse ndryshon vetëm nga rrotullimi kompleks i Givensit që zbatohet në hapin 12 të algoritmit standard.

1.2 Iteracioni Lanczos për matricat simetrike

Në rastin e matricave reale simetrike problemi i zgjidhjes së sistemeve lineare bazohet në iteracionin Lanczos të paraqitur në hyrjen e këtij punimi. Në rastin e procesit Lanczos matrica Hessenberg H_m e fituar është një matricë tridiagonale dhe simetrike, të cilës do i referohemi si T_m . Elementet e diagonales kryesore të T_m do i shënojmë me $\alpha_j = h_{j,j}$ dhe jashtë diagonales kryesore me $\beta_j = h_{j-1,j}$. Për gjetjen e vlerave vetjake të një matrice simetrike A , bashkohet algoritmi Lanczos për ndërtimin e nënhapësirës Krylov me procedurën Rayleigh-Ritz. Konkretisht ndërtohet matrica ortogonale $V_m = [v_1, v_2, \dots, v_m]$ e vektorëve ortogonalë Lanczos dhe përafrohen vlerat vetjake të A nga vlerat Ritz, vlerat vetjake të matricës simetrike tridiagonale T_m .

Në mënyrë të ngjashme me iteracionin Arnoldi për zgjidhjen e sistemeve të rralla dhe të mëdha, duke qenë se $T_m = V_m^T A V_m$ dhe $y_m = T_m^{-1}(\beta e_1)$ atëherë zgjidhja e sistemit linear përafrohet nga:

$$x_m = x_0 + V_m y_m \quad (1.7)$$

Algoritmi i iteracionit Lanczos për zgjidhjen e ekuacioneve lineare jepet më poshtë:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, dhe $v_1 = r_0 / \beta$

Hapi 2. Për $j = 1, 2, \dots, m$ kryej

Hapi 3. Në qoftë se $j = 1$ vendos $\beta_1 v_0 = 0$

Hapi 4. $w_j = Av_j - \beta_j v_{j-1}$

Hapi 5. $\alpha_j = t_{jj} = (w_j, v_j)$

Hapi 6. $w_j = w_j - \alpha_j v_j$

Hapi 7. $\beta_{j+1} = t_{j,j-1} = \|w_j\|_2$

Hapi 8. Në qoftë se $\beta_{j+1} = 0$, vendos $m = j$ dhe kalo në hapin 10

Hapi 9. $v_{j+1} = w_j / \beta_{j+1}$

Hapi 10. Fund

Hapi 11. Përcakto $T_m = tridiagonale(\beta_i, \alpha_i, \beta_{i+1})$ për $1 \leq i \leq m$

Hapi 12. Llogarit zgjidhjen e përafërt $x_m = x_0 + V_m y_m$ ku $y_m = T_m^{-1}(\beta e_1)$

Vektori i mbetjes r_m për përaftrimin x_m me anë të qasjes Lanczos është:

$$r_m = -\beta_{m+1} h_{m+1,m} e_m^T y_m v_{m+1} \quad (1.8)$$

Gjatë zbatimit të iteracionit Lanczos vetia e ortogonalitetit midis vektorëve v_j të nënhapësirës Krylov sa vjen dhe zbehet.

Metoda e Gradientit të Konjuguar (Conjugate Gradient Method) ose shkurt CG, zbatohet vetëm për sisteme që janë simetrike dhe pozitivisht të përcaktuara (përftuar nga FOM) ose Hermitiane dhe pozitivisht të përcaktuara (përftuar nga GMRES). CG ndërtohet duke zbatuar vetitë e ortogonalitetit të nënhapësirës Krylov dhe të konjugimit duke kënaqur kushtin që gabimi është minimal në të ashtuquajturën normë energji të vektorit të gabimit $x - x_{direkte}$, që shënohet si $\|x - x_{direkte}\|_A$ dhe $\|x - x_{direkte}\|_A^2 = (x - x_{direkte}, A(x - x_{direkte}))$. Kjo lloj norme ka rëndësi fizike për shumë probleme të modeluara me EDDP, që pas diskretizimit kthehen në probleme të minimizimit të një funksioni kuadratik f i cili është konveks dhe ka një minimum të vetëm:

$$f(x) = \frac{1}{2}x^T Ax - b^T x \quad (1.9)$$

me gradient $\nabla f(x) = Ax - b = -r$ ku r vektori mbetje. Vlera minimale e $f(x)$ arrihet duke marrë $x_{direkte} = A^{-1}b$ sa zgjidhja e drejtpërdrejtë (direkte) e sistemit dhe është konkretisht $-\frac{1}{2}b^T A^{-1}b$.

Me fjalë të tjera nënkupton minimizimin e normës energji të vektorit të gabimit $\|x - x_{direkte}\|_A^2$ për sistemin linear $Ax = b$ ku minimizuesi është $x_{direkte}$ [9].

Vektori p_1 i parë i bazës është gradienti i f në x_0 , pra $p_1 = Ax_0 - b$. Vektorët e tjerë të bazës të fituar me CG janë të konjuguar me gradientin. Çdo vektor p_{k+1} merret i tillë që të jetë drejtimi më i afërt me gradientin r_k :

$$p_{k+1} = r_{k+1} + \beta_k p_k \quad (1.10)$$

Ky drejtim merret nëpërmjet projektimit të r_k në hapësirën ortogonale me p_k në lidhje me produktin e brendshëm të A :

$$(Ap_k, r_k) = (Ap_k, p_k - \beta_{k-1}p_{k-1}) = (Ap_k, p_k) \quad (1.11)$$

Zgjidhjet e përafërta të metodës CG merren si $x_{k+1} = \arg \min f(x)$ pra $x_{k+1} = x_k + \alpha_k p_k$ me vektor të mbetjeve $r_{k+1} = r_k - \alpha_k Ap_k$. Për të fituar vargun r_j , për $j = 1, 2, \dots, m$ në mënyrë që të jenë ortogonale duhet të plotësohet:

$$(r_k + \alpha_k Ap_k, r_k) = 0 \quad (1.12)$$

Për rrjedhojë bazuar në (1.11) dhe (1.12) ndërtohen α_k dhe β_k si [10]:

$$\alpha_k = (r_k, r_k)/(Ap_k, p_k) \text{ dhe } \beta_k = (r_{k+1}, Ap_k)/(p_k, Ap_k) = (r_{k+1}, r_{k+1})/(r_k, r_k) \quad (1.13)$$

meqenëse $Ap_k = -(r_{k+1} - r_k)/\alpha_k$.

Për të gjetur një kufi të sipërm të gabimit të CG, problemi formulohet nëpërmjet trajtave të ndryshme polinomiale të CG si për shembull përafrimi nëpërmjet polinomeve të Chebyshev-it të cilët përshpejtojnë procesin e projektimit. Prandaj një kufi i sipërm i gabimit të kësaj metode do të ishte:

$$\|x - x_{direkte}\|_A \leq \sqrt{2} \left(\frac{\sqrt{\lambda_{max}/\lambda_{max}-1}}{\sqrt{\lambda_{max}/\lambda_{max}+1}} \right)^i \|x_0 - x_{direkte}\|_A \quad (1.14)$$

Ky kufi i sipërm përshkruan më së miri sjelljen e konvergjencës për matricat A , vlerat vetjake të së cilës janë të shpërndara në mënyrë homogjene. Pra konvergjenca e metodave të tipit gradient i konjuguar varet nga spektri i matricës A .

Fillimisht CG u prezantua si një metodë e drejtpërdrejtë e cila kërkon të shumtën n hapa. Në praktikë kjo metodë me rritjen e gabimit të rrumbullakimit mund të kërkojë më shumë se n iteracione ose të dështojë.

Në rastin kur matrica e sistemit është simetrike dhe jo pozitivisht e përcaktuar, metoda MINRES (Minimal Residual Method) është më e përshtatshme se CG. Metoda MINRES është një rast i veçantë i GMRES për matricat simetrike dhe formulohet në parim njësoj, pra duke zbatuar rrotullimet e Givens, matrica Hessenberg shndërrohet në një matricë tridiagonale simetrike T_m . MINRES krahasuar me GMRES, ul numrin e veprimeve [3]. Së pari, zbatohet iteracioni Lanczos për fitimin e bazës ortogonale dhe matricës që ka këta vektorë si shtylla të saj. Së dyti zbatohet faktorizimi QR për matricën tridiagonale të fituar \bar{T}_m me përmasa $(m + 1) \times m$. Më pas në ngjashmëri me GMRES, zgjidhet problemi i katrorëve më të vegjël me përmasa $(m + 1) \times m$ që minimizon mbetjet $\|r_m^{MINRES}\| = \|\beta e_1 - \bar{T}_m y\|_2$ dhe përafrimet e zgjidhjes fitohen si $x_m = x_0 + V_m y_m$.

Algoritmi MINRES:

Hapi 1. Jepen vlerësimet fillestare $v_1 = b - Ax_0$

Hapi 2. $\beta_1 = \|v_1\|_2$; $\eta = \beta_1$

Hapi 3. $\gamma_1 = \gamma_0 = 1$; $\sigma_1 = \sigma_0 = 0$

Hapi 4. $v_0 = 0$; $w_0 = w_{-1} = 0$

Hapi 5. Për $j = 1, 2, \dots, m$

Hapi 6. $v_j = \frac{1}{\beta_j} v_j$; $\alpha_j = v_j^T A v_j$

Hapi 7. $v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}$

Hapi 8. $\beta_{j+1} = \|v_{j+1}\|_2$

Hapi 9. $\delta_j = \gamma_j \alpha_j - \gamma_{j-1} \sigma_j \beta_j$

Hapi 10. $\rho_1 = \sqrt{\delta_j^2 + \beta_{j+1}^2}$

Hapi 11. $\rho_2 = \sigma_j \alpha_j + \gamma_{j-1} \gamma_j \beta_j$; $\rho_3 = \sigma_{j-1} \beta_j$

Hapi 12. $\gamma_{j+1} = \delta/\rho_1$; $\sigma_{j+1} = \beta_{j+1}/\rho_1$

Përditësohet zgjidhja me $W_j = V_j R_{j,j}^{-1}$

Hapi 13. $w_j = (v_j - \rho_3 w_{j-2} - \rho_2 w_{j-1})/\rho_1$

Hapi 14. $x_j = x_{j-1} + \gamma_{j+1} \eta w_j$

Hapi 15. $\|r_1\|_2 = |\sigma_{j+1}| \|r_{j-1}\|_2$

Kontrollohet konvergjenca dhe vazhdohet në qoftë se është e nevojshme

Hapi 16. $\eta = -\sigma_{j+1} \eta$

Hapi 17. Fund

Shtylla e fundit e matricës W në hapin 13 përftohet nga tre shtyllat e fundit të matricës V .

Kosto e formimit të bazës me anë të iteracionit Lanczos është e njëjtë si në metodën CG por në hapat në vazhdim MINRES rrit pak më tepër se CG numrin e veprimeve dhe koston e ruajtjes në kujtesë [10]. Avantazhi i MINRES qëndron pikërisht në fitimin e r_m^{MINRES} gjatë procesit dhe $\|r_m^{MINRES}\|_2$ zvogëlohet monotonisht.

1.3 Iteracioni Lanczos për matricat jo-simetrike

Në shumicën e rasteve të sistemeve jo-simetrike të mëdha nuk është i mundur reduktimi i matricës së koeficienteve të sistemit në një matricë simetrike të përmasave më të vogla, duke përdorur projektimin ortogonal. Madje edhe kur reduktimi është i mundur, qasja klasike e Lanczos që përdor ortogonalizimin e modifikuar Gram-Schmidt për matricat jo-simetrike, rrit tej mase numrin e veprimeve dhe hapësirën që zë procesi në kompjuter. Ajo që kërkohet të bëhet në këtë rast është përfitim i një baze jo ortogonale më anë të trajtës së CG me tre terma rekursivë [10], duke vënë si kusht që baza e fituar të jetë ortogonale me një bazë tjetër.

Pra, në rastin e matricave jo-simetrike formohen dy baza bi-ortogonale konkretisht për dy nënhapësirat Krylov: $\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, A^2v_1, \dots, A^{m-1}v_1\}$ dhe $\mathcal{K}_m(A^T, w_1) = \text{span}\{w_1, A^T w_1, (A^T)^2 w_1, \dots, (A^T)^{m-1} w_1\}$.

Algoritmi i Bi-ortogonalizimit të Lanczos:

Hapi 1. Zgjidhen v_1 dhe w_1 të tillë që $v_1^T w_1 = 1$

Hapi 2. Vendosen $\beta_1 = \delta_1 = 0$, $v_0 = w_0 = 0$

Hapi 3. Për $j = 1, 2, \dots, m$ kryej

Hapi 4. $\alpha_j = w_j^T A v_j$

Hapi 5. $\hat{v}_{j+1} = A v_j - \alpha_j v_j - \beta_j v_j - 1$

Hapi 6. $\hat{w}_{j+1} = A w_j - \alpha_j w_j - \beta_j w_j - 1$

Hapi 7. $\delta_{j+1} = \sqrt{|\hat{v}_{j+1}^T \hat{w}_{j+1}|}$;

Hapi 8. Në qoftë se $\delta_{j+1} = 0$ atëherë Ndalo

Hapi 9. $\beta_{j+1} = \hat{v}_{j+1}^T \hat{w}_{j+1} / \delta_{j+1}$

Hapi 10. $v_{j+1} = \hat{v}_{j+1}/\delta_{j+1}$

Hapi 11. $w_{j+1} = \hat{w}_{j+1}/\beta_{j+1}$

Hapi 12. Fund

Vektorët δ_{j+1} dhe β_{j+1} janë faktorë shkallëzimi përkatësisht për vektorët v_{j+1} të nënhapësirës $\mathcal{K}_m(A, v_1)$ dhe w_{j+1} të nënhapësirës $\mathcal{K}_m(A^T, w_1)$. Këta vektorë zgjidhen në mënyra të ndryshme në mënyrë të tillë që $(v_{j+1}, w_{j+1}) = 1$. Ky algoritëm ndërpritet nëse $\hat{v}_{j+1}^T \hat{w}_{j+1} = 0$, që në rastin më të mirë kur $\hat{v}_{j+1} = 0$ është ndërtuar nënhapësira invariante dhe zgjidhja e përafëruar është e mirë. Kur $\hat{w}_{j+1} = 0$ lind nevoja e rifillimit të procesit (procesi i rifillimit trajtohet në çështjen 1.4).

Algoritmi i Bi-ortogonalizimit mund të përshtatet edhe për zgjidhjen e sistemeve lineare duke marrë vlerësimet fillestare $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, $v_1 = r_0 / \beta$ dhe w_1 të tilla që $(v_1, w_1) = 1$.

Mbasi fitohen vektorët v_m , w_m dhe matrica tridiagonale T_m me elemente α_j dhe β_j si në algoritmin e bi-ortogonalizimit të Lanczos, duke u bazuar në teknikën Ritz-Galerkin, fitohet zgjidhja e përafërt $x_m = x_0 + V_m y_m$ ku $y_m = T_m^{-1}(\beta e_1)$ si në iteracionin e thjeshtë të Lanczos.

Procesi i bi-ortogonalizimit është me leverdi nëse kërkohet edhe zgjidhja e sistemit dual $A^T x^* = b^*$ në të njëjtën kohë me atë primar $Ax = b$, që duke zbatuar teknikën Ritz-Galerkin për matricën tridiagonale T_m^T , fiton zgjidhjet $x_m^* = x_0^* + W_m y_m^*$ ku $y_m^* = (T_m^T)^{-1}(\beta e_1)$.

Duke qenë se vektorët e të dyja bazave të gjeneruara nga procesi i bi-ortogonalizimit nuk janë gjithmonë ortogonale, përafërimi i një zgjidhjeje me mbetje minimum si në rastin e MINRES nuk është i mundur. Për më tepër vetë gjenerimi i dy bazave bi-ortogonale është i kushtueshëm dhe mundësia për dështim të algoritmit është e lartë [11]. Nga dy bazat e ndërtuara vetëm njëra përdoret për përafërimin e zgjidhjes së sistemit, tjetra luan thjeshtë rol ndihmës për të kryer prodhimin e brendshëm që ndërton bazën bi-ortogonale.

Metoda e Gradientit të Bi-Konjuguar (Bi-Conjugate Gradient Method) ose shkurt BiCG është metoda që përdor qasjen e paraqitur më sipër dhe është e ngjashme me metodën CG. Ndryshe nga CG kjo metodë mund të përdoret edhe në rastin e matricave jopozitivisht të përcaktuara.

Mbi matricat tridiagonale T_m dhe T_m^T të fituara nga algoritmi i bi-ortogonalizimit të Lanczos, zbatohet zbërthimi LU duke shprehur $T_m = L_m U_m$ dhe $T_m^T = U_m^T L_m^T$. Kështu fitojmë matricat $P_m = V_m U_m^{-1}$ dhe $P_m^* = W_m (L_m^T)^{-1}$. Zgjidhja e sistemeve primare dhe duale do të fitohet përkatësisht:

$$x_m = x_0 + P_m L_m^{-1}(\beta e_1) \text{ dhe } x_m^* = x_0^* + P_m^* (U_m^T)^{-1}(\beta e_1) \quad (1.15)$$

Algoritmi BiCG:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$ dhe r_0^* të tilla që $r_0^T r_0^* \neq 0$

Hapi 2. Vendos $p_0 = r_0$, $p_0^* = r_0^*$

Hapi 3. Për $j = 1, 2, \dots, m - 1$ bëj

Hapi 4. $\alpha_j = (r_j^T r_j^*) / ((p_j^*)^T A p_j)$

Hapi 5. $x_{j+1} = x_j + \alpha_j p_j$

Hapi 6. $r_{j+1} = r_j - \alpha_j A p_j$

Hapi 7. $r_{j+1}^* = r_j^* - \alpha_j A p_j^*$

Hapi 8. $\beta_j = (r_{j+1}^T r_{j+1}^*) / (r_j^T r_j^*)$

Hapi 9. $p_{j+1} = r_{j+1} + \beta_j p_j^*$

Hapi 10. $p_{j+1}^* = r_{j+1}^* + \beta_j p_j^*$

Hapi 11. Fund

Nga algoritmi i mësipërm në hapin 5, shohim që numerikisht zgjidhjet x_{j+1} përditësohen njëllor si në algoritmin CG. Siguria që vektorët r dhe r^* janë ortogonalë lind nga vetë fakti se v dhe w janë bi-ortogonalë. Ndërsa vetia e konjugimit për vektorët p dhe p^* sigurohet pasi $P_m^* A P_m = I$.

Konvergjencia e BiCG është zakonisht jo e rregullt, pasi normat e mbetjeve pasuese mund të sillen në mënyrë të çrregullt. Variante të BiCG janë sugjeruar të cilat shmangin përdorimin e A^T dhe zvogëlojnë më tej normën e mbetjeve, duke shprehur mbetjet si $r_m = p_m(A)r_0$ dhe $r_m^* = p_m(A^T)r_0^*$ për p_m një polinom të rendit m .

Në rast se duam të përdorim qasjen e mbetjeve minimum dhe vetinë e bi-ortogonalizimit, në mënyrë të ngjashme me metodën GMRES apo MINRES, ndërtohet Metoda e Mbetjeve Thuajse-Minimum (Quasi Minimal Residual) ose shkurt QMR. Ky term vjen nga fakti i përmendur më sipër që vektorët e të dyja bazave të gjeneruara nga procesi i bi-ortogonalizimit jo gjithmonë janë ortogonalë. Për rrjedhojë përafrimi i një zgjidhjeje me mbetje skatësisht minimum, jo gjithmonë është i mundur.

Duke marrë $v_1 = \beta^{-1} r_0$ përafrohen zgjidhjet e sistemit si $x_m = x_0 + V_m y_m$ në mënyrë të tillë që minimizojnë normën e mbetjeve $\|\beta e_1 - \bar{T}_m y\|_2$, ku \bar{T}_m është matrica tridiagonale $(m + 1) \times m$ e fituar nga iteracioni klasik Lanczos (si në MINRES) [12].

Algoritmi QMR:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$, $\gamma_1 = \|r_0\|_2$, $v_1 = r_0/\gamma_1$ dhe $w_1 = v_1$

Hapi 2. Për $j = 1, 2, \dots, m$ deri sa algoritmi të konvergjojë kryej

Hapi 3. Vlerësohen α_j, δ_{j+1} dhe v_{j+1}, w_{j+1} sipas algoritmit të bi-ortogonalizimit

Hapi 4. Kryhet faktorizimi QR i \bar{T}_m

Hapi 5. $\gamma_{j+1} = -s_j \gamma_j$

Hapi 6. $\gamma_j = c_j \gamma_j$ dhe $\alpha_j = c_j \alpha_j + s_j \delta_{j+1} = \sqrt{\delta_{j+1}^2 + \alpha_j^2}$

Hapi 7. $p_j = (v_j - \sum_{k=j-2}^{j-1} t_{kj} p_k) / t_{jj}$

Hapi 8. $x_j = x_{j-1} + \gamma_j p_j$

Hapi 9. Në qoftë se $|\gamma_{j+1}|$ është mjaftueshëm e vogël atëherë Ndalo

Hapi 10. Fund

QMR është më e qëndrueshme se metoda e gradientit të konjuguar, duke qenë se CG mund të rregullojë konvergjencën në rast dështimi, duke zbatuar zbërthimin LU të shtjellur mbi sistemin e reduktuar tridiagonal, zbërthim ky që jo gjithmonë ekziston. QMR ka konvergjencë më uniforme se CG dhe BiCG, megjithëse nuk konvergjon më shpejt se kjo e fundit. Lidhja (1.16) midis mbetjeve të tyre e tregon këtë:

$$\|r_m^{BiCG}\|_2 = \|r_m^{QMR}\|_2 / \sqrt{1 - (\|r_m^{QMR}\|_2 / \|r_{m-1}^{QMR}\|_2)^2} \quad (1.16)$$

QMR edhe pse nuk minimizon normën dy të mbetjeve, ka shpesh konvergjencë të njëjtë me GMRES por me dyfishin e prodhimeve matricë-vektor edhe pse kosto e prodhimit të vektorëve të bazave në QMR është e ulët. Gjithashtu e ulët është edhe kujtesa e përdorur për veprimet e prodhimit të këtyre vektorëve.

Ashtu siç përmendëm në rastin e BiCG, për të shmangur përdorimin e A^T , norma e mbetjes shprehet si $r_m = g_m(A)r_0$ për sistemin primar dhe si $r_m^* = g_m(A^T)r_0^*$ për sistemin dual $A^T x^* = b^*$ ku g_m është një polinom i rendit m . Nga këtu kemi që $(g_m(A)r_0, g_s(A^T)r_0^*) = 0$ për $s < m$ dhe është e mundur që parametrat e BiCG të fitohen nga një bashkësi tjetër polinomesh \tilde{g}_s të rendit s për të cilët $(\tilde{g}_s(A)g_m(A)r_0, r_0^*) = 0$. Me fjalë të tjera, meqenëse $g_m(A)r_0$ është pingul (perpendikular) me nënhapësirën $\mathcal{K}_m(A^T, r_0^*) = \text{span}\{r_0^*, A^T r_0^*, \dots, (A^T)^{m-1} r_0^*\}$ parametrat e BiCG mund të fitohen duke marrë r_m perpendikular me $\tilde{g}_s g_m^T r_0$ [3].

Duke marrë polinomet \tilde{g}_s në trajtën $q_s(x) = (1 - \omega_j x)q_{s-1}(x)$ dhe duke marrë ω_j të tillë që minimizojnë r_m (në lidhje me ω_j) për mbetjet r_s që mund të shkruhen si $r_s = q_s(A)g_s(A)r_0$, fitojmë atë që quhet Metoda e Stabilizuar e Gradientit të Bi-Konjuguar (Bi-conjugate Gradient Stabilized Method) apo shkurt BiCGStab [13]. BiCGStab ka nevojë për dy prodhime të brendshme, më shumë se BiCG për çdo iteracion dhe përdor saktësisht dy prodhime matricë-vektor për iteracion.

Algoritmi BiCGStab:

Hapi 1. Jepen vlerësimet fillestare $r_0 = b - Ax_0$ dhe $r^* = r_0$

Hapi 2. Për $j = 1, 2 \dots m$ kryej

Hapi 3. $\rho_{j-1} = (r^*)^T r_{j-1}$

Hapi 4. Në qoftë se $\rho_{j-1} = 0$ metoda dështon

Hapi 5. Në qoftë se $j = 1$

Hapi 6. $p_j = r_{j-1}$

- Hapi 7. Përndryshe $\beta_{j-1} = (\rho_{j-1}/\rho_{j-2})(\alpha_{j-1}/\omega_{j-1})$
Hapi 8. $p_j = r_{j-1} + \beta_{j-1}(p_{j-1} - \omega_{j-1}v_{j-1})$
Hapi 9. Fund
Hapi 10. $v_j = Ap_j$
Hapi 11. $\alpha_j = \rho_{j-1}/(r^*)^T v_j$
Hapi 12. $s = r_{j-1} - \alpha_j v_j$
Hapi 13. Në qoftë se $\|s\|_2$ është mjaftueshëm e vogël: $x_j = x_{j-1} + \alpha_j p_j$ dhe Ndalo
Hapi 14. $t = As$, $\omega_j = t^T s / t^T t$
Hapi 15. $x_j = x_{j-1} + \alpha_j p_j + \omega_j s$
Hapi 16. $r_j = s - \omega_j t$
Hapi 17. Nëse $\omega_i \neq 0$ atëherë vazhdo
Hapi 18. Fund

Lidhja që ekziston midis vektorit të mbetjeve r_m dhe vektorit p_m është e njëjtë si në metodën BiCG, të cilat në trajtë polinomiale shkruhen si $r_m = g_m(A)r_0$ (ku $g_m(0) = 1$) dhe $p_m = t_{m-1}(A)r_0$. Njësoj edhe për sistemin dual kemi $r_m^* = g_m(A^T)r_0^*$ dhe $p_m^* = t_{m-1}(A^T)r_0^*$. Nga këtu lind lidhja e dy trajtave polinomiale me njëra-tjetrën:

$$g_m(A)r_0 = (g_{m-1}(A) - \alpha_{m-1}A t_{m-1}(A))r_0 \quad (1.17)$$

dhe

$$t_{m-1}(A)r_0 = (g_{m-1}(A) + \beta_{m-1}t_{m-2}(A))r_0 \quad (1.18)$$

Atëherë, prodhimet $q_m(A)g_m(A)r_0$ dhe $q_m(A)t_m(A)r_0$ jepen nga:

$$q_m(A)g_m(A)r_0 = [q_{m-1}(A)g_{m-1}(A) - \alpha_{m-1}A q_{m-1}(A) t_{m-1}(A)] r_0 \quad (1.19)$$

dhe

$$q_m(A)t_m(A)r_0 = [q_m(A)g_m(A) + \beta_m q_{m-1}(A)t_{m-1}(A) - \omega_i \beta_m q_{m-1}(A) t_{m-1}(A)] \quad (1.20)$$

Së fundmi ndërtohen parametrat α_j dhe β_j si në rastin e BiCG dhe duke riformuluar r_j dhe p_j për $q_j(A)g_j(A)r_0$ dhe $q_{j-1}(A)t_{j-1}(A)r_0$, fitojmë formulat e algoritmit të mësipërm BiCGStab.

Koha e ekzekutimit të algoritmit BiCGStab dhe kërkesa për kujtesë nuk rriten me rritjen e numrit të iteracioneve, ndryshe nga metoda GMRES. BiCGStab përdor për të një iteracion të njëjtën hapësirë në kujtesë që përdor GMRES për dy iteracione. Sjellja e konvergencës së kësaj metode është zakonisht më pak e qëndrueshme se GMRES (bazuar në bi-ortogonalizim).

1.4 Procesi i rifillimit

Procesi i rifillimit nënkupton rifillimin e hapave të metodave të nënhapësirave Krylov mbas një numri të caktuar k iteracionesh, duke marrë si përafrim fillestar të zgjidhjes, përafrimin e k -të të metodës. Ky proces përsëritet derisa norma e mbetjes të jetë

mjaftueshëm e vogël dhe përshpejton konvergjencën. Në rastin e metodave që bazohen në iteracionin Arnoldi, rifillimi ul koston e ruajtjes së vektorëve ortogonalë të bazës me rritjen e numrit të iteracioneve (duke qenë se kosto e ruajtjes për shtyllat e V_m për procesin pa rifillim është $m \times n$). Gjithashtu në problemin e vlerave vetjake për probleme të mëdha dhe të rralla, gjatë procesit Arnoldi nuk kemi shumë kontroll se cilin çift vetjak përafron ky proces. Procesi i rifillimit ndihmon edhe në këtë rast. Konkretisht nuk ka një rregull për të përcaktuar mbas sa hapash duhet të rifillojë metoda. Në përgjithësi, për veti të ndryshme të sistemeve, konvergjenca mbas zbatimit të procesit të rifillimit nuk është e njëjtë. Prandaj është e nevojshme që të kryen disa teste me vlera të ndryshme të parametrin të rifillimit. Ideja është që fillohet nga parametri i rifillimit me vlerë 1 dhe rritet derisa të arrijë një vlerë maksimale. Kur vlera maksimale arrihet, ky parametër rimerr vlerën një ose vazhdon ruhet sa vlera maksimale. Procesi i rifillimit çalon kur matrica nuk është pozitivisht e përcaktuar.

Duke zbatuar rifillimin qëndrueshmëria e metodave Krylov çënohet, pasi me rritjen shumë të parametrin të rifillimit (për probleme me përmasa shumë të mëdha) konvergjenca e metodave Krylov ngadalësohet ose nuk është më e garantuar. Metoda e nënhapësirave Krylov më efikase kur zbatohet procesi i rifillimit është GMRES (dhe më pas FOM). Për shumë metoda të tjera të nënhapësirës Krylov ky proces nuk sjell asnjë fitim, prandaj lind nevoja e zbatimit të teknikave të tjera që çojnë në uljen e numrit të veprimeve, koston së kujtesës apo përshpejtimin të konvergjencës. Një ndër to është procesi i ndërprerjes (trungëzimit) të këtyre metodave apo zbatimi i parakushtëzuesëve të dobishëm.

KAPITULLI 2

Parakushtëzimi

Bazuar në hapësirën e limituar kompjuterike dhe në faktin se metodat e nënhapësirave Krylov jo gjithmonë garantojnë një përafrim të kënaqshëm të zgjidhjes brenda një numri të duhur veprimesh apo hapash, kërkohet të gjendet një operator K i tillë që $K^{-1}A$ të ketë veti spektrale më të mira se A . Ashtu siç pamë më sipër për arsye se disa metoda të këtij grupi nuk janë praktike apo dështojnë, disa janë specifike vetëm për një grup të caktuar problemesh dhe konvergjenca e tyre varet nga vetitë spektrale të matricës së sistemit, informacion ky që jo gjithmonë është në dispozicion. Prandaj operatori K sa më afër A merret si një parakushtëzues dhe nëse zgjidhet me kujdes, kur zbatohen metodat Krylov, do të duhen vetëm disa iteracione që të çojnë në një përafrim mjaftueshëm të mirë të zgjidhjes së sistemit. Për shembull, shkallëzimi i të gjithë rreshtave të një sistemi linear për t'i bërë elementet e diagonales kryesore të barabarta me një është një formë e qartë parakushtëzimi. Ideja është se duke marrë $K = A$ do të kishim sistemin ideal $K^{-1}Ax = Ix = K^{-1}b$ dhe metodat Krylov do të jepnin zgjidhjen e saktë me një iteracion të vetëm.

Vështirësia për gjetjen e një parakushtëzuesi të mirë qëndron në faktin se sa afër A -së merret ai, sa është kosto e ndërtimit të tij, në mangësinë e informacionit rreth sjelljes së zgjidhjes së sistemit si dhe në analizën e vetive spektrale të A [14]. Prandaj përzgjedhja dhe ndërtimi i një parakushtëzuesi të mirë bëhet në mënyrë intuitive ose në mënyrë eksperimentale, duke qenë se nuk ka baza teorike të mirëfillta. Metodat e nënhapësirave Krylov kanë mjaft liri në përcaktimin dhe ndërtimin e parakushtëzuesëve pasi nuk është e nevojshme që të njihet secili element i matricës A dhe nuk është i nevojshëm modifikimi i pjesëve të saj. Duke qenë se K merret i tillë që sistemi $Ky = z$ të zgjidhet më lehtë se sistemi $Ax = b$, për një vektor të dhënë y mund të gjenerohet vektori z si $z = Ay$. Përveç se në raste të veçanta matrica $K^{-1}A$ nuk llogaritet shprehimisht si në rastin e sistemeve të rralla edhe në atë të sistemeve të dendura. Në shumicën e rasteve kjo rezulton në matrica të dendura, prandaj për çdo aplikim të $K^{-1}A$ ndaj një vektori y , duhet fillimisht të llogaritet rezultati w duke zbatuar operatorin A ndaj vektorit y . Së fundmi llogaritet rezultati z i zbatimit të K^{-1} ndaj vektorit w duke zgjidhur sistemin $Kz = w$.

Parakushtëzuesit në trajtë matricash diagonale janë ndër të vetmit parakushtëzues që zbatohen direkt ndaj matricës A dhe mund të aplikohen para dhe mbas një parakushtëzuesi tjetër. Ka mënyra të ndryshme për të zbatuar parakushtëzimin që pavarësisht teknikës së përdorur do të rezultojë në të njëjtat vlera vetjake të matricës së parakushtëzuar. Por sjellja e konvergjencës gjithashtu varet nga komponentet e mbetjes fillestare në drejtimet e vektorëve vetjakë të cilët ndryshojnë për teknika të ndryshme parakushtëzimi të përdorura.

Teknikat e mundshme të parakushtëzimit që mund të përdoren janë [14]:

- Parakushtëzimi i majtë: i cili zbaton metodën iterative ndaj $K^{-1}Ax = K^{-1}b$. Vetia e simetrisë për A dhe K nuk nënkupton simetri edhe për $K^{-1}A$ dhe nëse K është simetrike dhe pozitivisht e përcaktuar atëherë $[x, y] \equiv (x, Ky)$ përcakton një prodhim të brendshëm të përshtatshëm. Për metodat si GMRES ose MINRES në këtë rast minimizohen mbetjet e parakushtëzuara $K^{-1}(b - Ax)$, që ndryshojnë nga mbetjet $(b - Ax)$. Kjo gjë sjell pasoja në kriterin e ndalimit të këtyre algoritmeve.
- Parakushtëzimi i djathtë: i cili zbaton metodën iterative ndaj $AK^{-1}y = b$, dhe me $x = K^{-1}y$. Vetia e simetrisë për A dhe K nuk nënkupton simetri edhe për AK^{-1} . Parakushtëzimi nga e djathta ka ndikim vetëm mbi operatorin dhe jo mbi anën e djathtë të sistemit të parakushtëzuar, gjë që favorizon hartimin e softeve për aplikacione të ndryshme.
- Parakushtëzimi i dyanshëm: Për një parakushtëzues $K = K_1K_2$, metoda iterative mund të zbatohet ndaj $K_1^{-1}AK_2^{-1}z = K_1^{-1}b$ ku $x = K_2^{-1}z$. Kjo teknikë parakushtëzimi mund të përdoret për ata parakushtëzues që vijnë në formë faktorësh. Në rastet kur K nuk mund të përdoret për të përcaktuar një prodhim të brendshëm, parakushtëzimi i dyanshëm mund të ndihmojë në përfitim të një operatori simetrik.

Në hapësirat Krylov të gjeneruara nuk mund të nxjerrin informacion edhe për matricën origjinale të pa parakushtëzuar. Zgjedhja e operatorit K varet nga teknika të ndryshme algebrë që zbatohen mbi matricën fillestare dhe që bazohen në vetitë e veçanta të një grupi të caktuar problemesh. Por lind nevoja për teknika parakushtëzimi të qëndrueshme dhe për grupe më të mëdha problemesh jo aq specifike. Nuk ka teori të mirëfilltë që tregon sjelljen e metodave iterative mbas zbatimit të parakushtëzimit, sidomos për matricat jopozitivisht të përcaktuara. Supozojmë se A është e papërcaktuar, atëherë duhet të sigurohet se matrica e parakushtëzuar $K^{-1}A$ të ketë vlerat e saj vetjake sa më afër njëshit që të jetë e mundur. Më tej të supozojmë se është e mundur të përmirësohet parakushtëzuesi K nga $K = I$ në $K = A$, ku të gjitha vlerat vetjake janë saktësisht një, prandaj vlerat vetjake do të duhet të lëvizin drejt njëshit teksa parakushtëzuesi përmirësohet. Për sa i përket vlerave vetjake negative, me përmirësimin e parakushtëzuesit sa vijnë e afrohen më shumë drejt origjinës duke u munduar të afrohen sa më shumë drejt njëshit.

Shpesh disa lloje parakushtëzuesish siç janë parakushtëzuesit jo të plotë, një nga mënyrat më të thjeshta të përcaktimit të një parakushtëzuesi duke kryer një faktorizim jo të plotë të matricës origjinale A , rrisin numrin e iteracioneve nëse rritet niveli i plotësimin të tyre. Madje edhe në disa raste kur parakushtëzuesit reduktojnë një numër të konsiderueshëm të iteracioneve, duhet parë se sa e lartë është kosto e ndërtimit dhe zbatimit të tyre.

Në vazhdim janë paraqitur disa nga llojet më të përdorura të parakushtëzimit për rastin e matricave të rralla dhe të mëdha.

2.1 Faktorizimi ILU

Faktorizimi jo i plotë i matricës origjinale A të sistemit përfshin një zbrëthim të formës $A = LU - R$ ku L dhe U kanë përkatësisht pjesët e poshtme dhe të sipërme të strukturës jo zero të A dhe R është matrica e mbetjes ose gabimi i faktorizimit. Ky faktorizim jo i plotë i njohur si $ILU(0)$ është mjaft i lehtë për t'u llogaritur por që shpesh çon në një përafrim të papërpunuar, i cili mund të rezultojë në rritjen shumë të numrit të iteracioneve. Për këtë janë zhvilluar disa faktorizime alternative jo të plota që lejojnë më shumë plotësim në L dhe U por kosto e llogaritjes së këtyre faktorëve është e lartë. Në përgjithësi, mund të specifikohet një model për L dhe U ku ato kërkohen në mënyrë që të përmbushin disa kushte të caktuara. Kjo çon në klasën e përgjithshme të teknikave të faktorizimit jo të plotë.

Marrim një matricë të rrallë të përgjithshme A elementet e së cilës janë a_{ij} , për $i = 1, \dots, n$ dhe $j = 1, \dots, n$. Një proces i përgjithshëm jo i plotë i faktorizimit LU (ILU) llogarit një matricë të rrallë trekëndore të poshtme L dhe një matricë të rrallë trekëndore të sipërme U në mënyrë që matrica e mbetjes $R = LU - A$ të përmbushë disa kufizime të caktuara, si të ketë elemente zero në disa pozicione të caktuara. Fillimisht, përshkruhet një parakushtëzues i përgjithshëm ILU dhe më pas, diskutohet faktorizimi $ILU(0)$ si trajta më e thjeshtë e parakushtëzuesve ILU .

Një algoritëm i përgjithshëm për ndërtimin e faktorizimeve jo të plota LU mund të përftohet duke zbatuar eliminimin e Gauss-it dhe duke hequr disa elemente në pozicione të paracaktuara jashtë diagonales kryesore. Për të analizuar këtë proces dhe për të provuar ekzistencën e M –matricave i referohemi teoremës 2.1 [15].

Teorema 2.1.

Le të jetë A një M –matricë dhe le të jetë A_1 matrica e përftuar nga hapi i parë i eliminimit të Gauss-it. Atëherë edhe A_1 është një M – matricë. Një M –matricë është një Z – matricë (pra me elemente jashtë diagonales kryesore më të vegjël ose të barabartë me zero) me pjesën reale të vlerave vetjake jonegative.

Matrica $(n - 1) \times (n - 1)$ e përftuar nga A_1 , duke eliminuar rreshtin e parë dhe shtyllën e parë, është gjithashtu një M –matricë.

Duke supozuar që disa elemente janë eliminuar nga rezultati i eliminimit të Gauss-it jashtë diagonales kryesore, çdo element që eliminohet është jopozitiv, i cili shndërrohet në zero. Prandaj, matrica që përftohet \tilde{A}_1 është e tillë që $\tilde{A}_1 = A_1 + R$, ku elementet e R janë të tillë që $r_{ii} = 0$ dhe $r_{ij} \geq 0$. Prandaj, $A_1 \leq \tilde{A}_1$ dhe elementet jashtë diagonales së \tilde{A}_1 janë jopozitivë. Meqenëse A_1 është një M –matricë, atëherë dhe \tilde{A}_1 është një M –matricë. Procesi tani mund të përsëritet në matricën $\tilde{A}(2:n, 2:n)$, dhe më pas vazhdohet derisa përftohet faktorizimi jo i plotë i A .

Arsyetimi i mësipërm tregon se në çdo hap të këtij procesi përftojmë një M –matricë dhe se procesi nuk paraqet vështirësi. Elementet që do të eliminohen në çdo hap nuk janë specifikuar ende. Specifikimi i tyre mund të bëhet në mënyrë statistikore, duke

zgjedhur paraprakisht një model jo-zero pra një strukturë të pozicioneve të elementeve të ndryshme nga zero. Për sa i përket strukturës së pozicioneve të elementeve zero që po e quajmë modeli zero, i vetmi kufizim është që ky model duhet të përjashtojë elementet e diagonales kryesore. Prandaj, për çdo bashkësi P të modelit zero, të tillë që $P \subset \{(i, j) \mid i \neq j; 1 \leq i, j \leq n\}$, një faktorizim jo i plotë LU , ILU_p , mund të kryhet si më poshtë.

Algoritmi i Faktorizimit të Përgjithshëm ILU , Versioni KIJ :

Hapi 0. Për çdo $(i, j) \in P$ bashkësia $a_{ij} = 0$

Hapi 1. Për $k = 1, \dots, n - 1$ kryej

Hapi 2. Për $i = k + 1, \dots, n$ dhe nëse $(i, k) \notin P$ kryej

Hapi 3. $a_{ik} := a_{ik} / a_{kk}$

Hapi 4. Për $j = k + 1, \dots, n$ dhe për $(i, j) \notin P$ kryej

Hapi 5. $a_{ij} := a_{ij} - a_{ik} * a_{kj}$

Hapi 6. Fund

Hapi 7. Fund

Hapi 8. Fund

Nëse A është një M –matricë dhe P një model zero i dhënë, i përcaktuar nga $P \subset \{(i, j) \mid i \neq j; 1 \leq i, j \leq n\}$ atëherë algoritmi i faktorizimit të përgjithshëm ILU nuk dështon dhe prodhon një faktorizim jo të plotë, $A = LU - R$ që është një ndarje e rregullt e A .

Një faktorizim ILU i bazuar në algoritmin e faktorizimit të përgjithshëm ILU (versioni KIJ) është i vështirë për t'u zbatuar sepse në secilin hap k , të gjitha rreshtat nga $k + 1$ deri tek rreshti n janë në proces modifikimi. Sidoqoftë, faktorizimet ILU varen nga zbatimi i eliminimit Gauss. Njihen disa variante të eliminimit Gauss që varen nga renditja e tre cikleve që lidhen me ndryshoret e kontrollit i, j dhe k në këtë algoritëm. Kështu, algoritmi i parë rrjedh nga ajo që njihet si varianti k, i, j . Në kontekstin e faktorizimit jo të plotë të LU , varianti që përdoret më shpesh për një strukturë të dhënave rresht të njëpasnjëshëm është varianti i, k, j , i përshkruar më poshtë për matricat e dendura.

Algoritmi i Eliminimit Gauss - Versioni IKJ

Hapi 1. Për $i = 2, \dots, n$ kryej

Hapi 2. Për $k = 1, \dots, i - 1$ kryej

Hapi 3. $a_{ik} := a_{ik} / a_{kk}$

Hapi 4. Për $j = k + 1, \dots, n$ kryej

Hapi 5. $a_{ij} := a_{ij} - a_{ik} * a_{kj}$

Hapi 6. Fund

Hapi 7. Fund

Hapi 8. Fund

Algoritmi i paraqitur më lart nënkupton që rreshti i -të i A mund të mbishkruhet nga rreshtat e i -të të matricave të faktorizimit pra L dhe U . Njëkohësisht, hapi i i algoritmit gjeneron rreshtin e i -të të matricës L dhe rreshtin e i -të të matricës U . Rreshtat $1, 2, \dots, i - 1$ të L dhe U përdoren në hapin i por nuk modifikohen.

Për matricat e rralla, përshtatja e këtij versioni është e lehtë sepse rreshtat e L dhe U gjenerohen në mënyrë të njëpasnjëshme dhe grumbullohen në një strukturë të dhënash në trajtë rreshti të formatit CSR (Compressed Sparse Row). Gjë kjo që përbën një avantazh të rëndësishëm, pasi bazuar në këtë fakt, faktorizimi i përgjithshëm ILU merr trajtën e mëposhtme.

Algoritmi i Faktorizimit të Përgjithshëm ILU, Versioni IKJ

- Hapi 1. Për $i = 2, \dots, n$ kryej
- Hapi 2. Për $k = 1, \dots, i - 1$ dhe nëse $(i, k) \notin P$ kryej
- Hapi 3. $a_{ik} := a_{ik}/a_{kk}$
- Hapi 4. Për $j = k + 1, \dots, n$ dhe për $(i, j) \notin P$ kryej
- Hapi 5. $a_{ij} := a_{ij} - a_{ik}a_{kj}$.
- Hapi 6. Fund
- Hapi 7. Fund
- Hapi 8. Fund

Shihet qartë që ky variant praktik IKJ i ILU është i njëvlershëm me versionin KIJ.

Pohim 2.1

Le të jetë P një model zero që plotëson $P \subset \{(i, j) \mid i \neq j; 1 \leq i, j \leq n\}$. Më pas, faktorët ILU të prodhuar nga Algoritmi i Faktorizimit të Përgjithshëm ILU i bazuar në KIJ dhe Algoritmi i Faktorizimit të Përgjithshëm ILU i bazuar në IKJ janë identikë nëse të dy mund të llogariten.

Vërtetim

IKJ përftohet nga versioni KIJ duke ndryshuar rendin e cikleve k dhe i . Për tu bindur për këtë fakt, riformulohen dy ciklet e para të versionit KIJ si më poshtë:

për $k = 1:n$ kryej
 për $i = 1:n$ kryej
 nëse $k < i$ dhe për $(i, k) \notin P$ kryej
 $ope(row(i), row(k))$

ku $ope(row(i), row(k))$ janë hapat nga 3 deri në 6 të versionit KIJ dhe IKJ. Është e qartë se ciklet k dhe i mund të ndërrojnë vend pa problem. Saktësojmë faktin që, kjo është e vërtetë vetëm për një model të pandryshueshëm ILU. Nëse modeli përcaktohet në mënyrë të zgjuar ndërkohë që ndodh algoritmi i eliminimit Gauss, atëherë modelet e përfutuara nga versione të ndryshme të eliminimit Gauss mund të jenë të ndryshme nga njëra-tjetra.

Algoritmi ILU versioni IKJ prodhon faktorët L dhe U të tillë që $A = LU - R$ ku $-R$ është matrica e elementeve që janë eliminuar gjatë procesit jo të plotë të eliminimit. Kur $(i, j) \in P$, elementi i R është $r_{ij} = -a_{ij}$, që përftohet nga përfundimi i ciklit k , përndryshe është $r_{ij} = 0$.

2.1.1 ILU(0) me plotësim zero

Teknika e faktorizimit jo të plotë LU pa plotësim, ILU(0), konsideron modelin zero P që të ketë pikërisht strukturën e elementeve zero të A . Matrica trekëndore e poshtme L që ka të njëjtën strukturë si pjesa e poshtme e A , dhe matrica U ka të njëjtën strukturë me atë të pjesës së sipërme të A . Në përgjithësi, është e pamundur të përputhet A me këtë prodhim për çdo L dhe U . Kjo ndodh për shkak të diagonaleve shtesë në prodhim, elementet e të cilave quhen elemente plotësuese. Gjithsesi, nëse këto elemente plotësuese nuk merren parasysh, atëherë është e mundur të gjejmë L dhe U në mënyrë që prodhimi i tyre të jetë i barabartë me A në diagonalet e tjera.

Pra për faktorizimin ILU(0) kemi elementet e $R = LU - A$ që janë zero në pozicionet e elementeve të ndryshme nga zero të A pra $NZ(A)$. Algoritmi standard ILU(0) përcaktohet duke përdorur ILU versioni IKJ me modelin P të barabartë me strukturën e elementeve zero të A [6].

Algoritmi i Faktorizimit Standard ILU(0)

- Hapi 1. Për $i = 2, \dots, n$ kryej
 - Hapi 2. Për $k = 1, \dots, i - 1$ dhe për $(i, k) \in NZ(A)$ kryej
 - Hapi 3. $a_{ik} = a_{ik}/a_{kk}$
 - Hapi 4. Për $j = k + 1, \dots, n$ dhe për $(i, j) \in NZ(A)$ kryej
 - Hapi 5. $a_{ij} := a_{ij} - a_{ik}a_{kj}$.
 - Hapi 6. Fund
 - Hapi 7. Fund
 - Hapi 8. Fund

Në disa raste, është e mundur të shprehim faktorizimin ILU(0) në trajtën $K = (D - E)D^{-1}(D - F)$, ku $-E$ dhe $-F$ janë përkatësisht, matrica trekëndore e poshtme dhe trekëndore e sipërme e A , dhe D është një matricë diagonale e caktuar, në përgjithësi e ndryshme nga matrica diagonale e A -së, elementet e të cilës përcaktohen nga një rekursion i tillë që diagonalja e prodhimit të matricave $(D - E)$, D^{-1} dhe $(D - F)$ të jetë i barabartë me diagonalen e A .

Një avantazh i qartë është se kërkohet vetëm një diagonale shtesë për të grumbulluar elementet. Kjo formë e faktorizimit ILU(0) është e njëvlershme me faktorizimet jo të plota të marra nga Algoritmi i Faktorizimit ILU(0), ku prodhimi i pjesës saktësisht të poshtme dhe të pjesës saktësisht të sipërme të A , përbëhet vetëm nga elementet e diagonales dhe elementet plotësuese. Të dyja matricat L dhe U në ILU(0) kanë të njëjtin numër të elementeve të ndryshme nga zero si matrica origjinale A .

Ashtu si ILU(0) dhe parakushtëzuesit SSOR me $\omega = 1$ mund të paraqiten në formën $K = (D - E)D^{-1}(D - F)$ por ndryshojnë në mënyrën e përcaktimit të matricës diagonale D , pasi për SSOR me $\omega = 1$, D është diagonalja e vetë matricës A .

2.1.2 ILU(p) me nivel plotësimi p

Saktësia e faktorizimit jo të plotë ILU(0) ndonjëherë mund të jetë e pamjaftueshme për të dhënë një shpejtësi të përshtatshme të konvergjencës. Faktorizimet më të sakta jo të plota LU , shpesh janë më efikase, si dhe më të besueshme, të cilat ndryshojnë nga ILU(0) duke lejuar disa plotësime. Kështu për shembull, ILU(1) nënkupton plotësimet me nivel një, term i cili do të shpjegohet në vazhdim.

Për të ilustruar ILU(p) me të njëjtin shembull si më parë, faktorizimi ILU(1) rezulton nga marrja e P si modeli zero i prodhimit LU të faktorëve L dhe U të marrë nga ILU(0). Supozohet se matrica origjinale ka këtë model të shtuar $NZ_1(A)$, që do të thotë se pozicionet e plotësimi të krijuara në këtë prodhim, i përkasin modelit të shtuar $NZ_1(A)$, por vlerat e tyre në fakt janë zero. Faktorët $L1$ dhe $U1$ të faktorizimit ILU(1) merren duke kryer një faktorizim ILU(0) në këtë matricë modeli të shtuar. Matrica e re ka dy diagonale shtesë në pjesët e sipërme dhe të poshtme.

Ndërtimi i përcaktuar në këtë mënyrë, nuk zbatohet edhe për matricat e rralla të përgjithshme por mund të përgjithësohet duke parqitur konceptin e nivelit të plotësimi (mbushjes). Një nivel i plotësimi i atribuohet çdo elementi që përftohet nga eliminimi Gauss dhe eliminimi do të bazohet në vlerën e nivelit të plotësimi. Do të referohemi algoritmit të eliminimit Gauss paraqitur më lart por mund të përdoret edhe çdo formë tjetër e tij. Niveli i mbushjes duhet të jetë tregues i përmasës që do të thotë sa më i lartë niveli plotësimi, aq më të vegjël janë elementet. Një madhësi ϵ^k i atribuohet çdo elementi, niveli i mbushjes së të cilit është k , ku $\epsilon < 1$. Fillimisht, niveli plotësimi merret zero $nivel_{ij} = 0$ për element të ndryshëm nga zero $a_{ij} \neq 0$ ose $i = j$ dhe një element zero $a_{ij} = 0$ ka një nivel plotësimi $nivel_{ij} = \infty$. Elementet a_{ij} përditësohet në algoritmin e eliminimit të Gauss-it sipas $a_{ij} = a_{ij} - a_{ik} \times a_{kj}$. Nëse $nivel_{ij}$ është niveli aktual i elementit a_{ij} , atëherë modeli ynë tregon se madhësia e elementit të përditësuar duhet të jetë

$$madhësia(a_{ij}) := \epsilon^{nivel_{ij}} - \epsilon^{nivel_{ik}} \times \epsilon^{nivel_{kj}} = \epsilon^{nivel_{ij}} - \epsilon^{nivel_{ik} + nivel_{kj}}.$$

Prandaj, madhësia e a_{ij} do të jetë sa maksimumi i dy madhësive $\epsilon^{nivel_{ij}}$ dhe $\epsilon^{nivel_{ik} + nivel_{kj}}$, dhe nga këtu përcaktohet niveli i ri i plotësimi si më poshtë,

$$nivel_{ij} = \min\{nivel_{ij}, nivel_{ik} + nivel_{kj} + 1\}.$$

Vëmë re se niveli i plotësimi të një elementi nuk do të rritet kurrë gjatë eliminimit. Për rrjedhojë, nëse $a_{ij} \neq 0$ në matricën origjinale A , atëherë elementi në pozicionin i, j do të ketë një nivel plotësimi të barabartë me zero gjatë gjithë procesit të eliminimit.

Një përkufizim alternativ për përditësimin e nivelit të plotësimi në mënyrën rekursive është:

$$nivele_{ij} = \min\{nivele_{ij}, \max\{nivele_{ik}, nivele_{kj}\} + 1\}$$

Një trajtë e përgjithshme e algoritmit të faktorizimit ILU me nivel plotësimi p jepet në vazhdim [6].

Algoritmi i Faktorizimit ILU me nivel plotësimi p

Hapi 1. Për $a_{ij} \neq 0$, vendos $nivele(a_{ij}) = 0$

Hapi 2. Për $i = 2, \dots, n$ kryej

Hapi 3. Për $k = 1, \dots, i - 1$ dhe për $nivele(a_{ik}) \leq p$ kryej

Hapi 4. $a_{ik} := a_{ik}/a_{kk}$

Hapi 5. $a_{i*} := a_{i*} - a_{ik}a_{k*}$ rreshtin e i -të të matricës A .

Hapi 6. përditëso nivelet e plotësimit $\min\{nivele_{ij}, nivele_{ik} + nivele_{kj} + 1\}$

Hapi 7. Fund

Hapi 8. zëvendëso çdo element në rreshtin i ku $nivele(a_{ij}) > p$, me zero

Hapi 9. Fund

Ekzistojnë një numër pengesash në algoritmin e mësipërm. Së pari, sasia e punës plotësuese dhe llogaritëse për përfundimin e faktorizimit ILU(p) nuk është e parashikueshme për $p > 0$. Së dyti, kosto e përditësimit të niveleve mund të jetë e lartë. Për më tepër, niveli i plotësimit për matricat e papërcaktuara mund të mos jetë një tregues i saktë i madhësisë së elementeve që po eliminohen. Kështu, algoritmi mund të eliminojë elemente të mëdha dhe si rezultat, përftojme një faktorizim jo të plotë jo të saktë. Për pasojë, $R = LU - A$ nuk është e vogël. Parakushtëzimi me anë të faktorizimeve jo të plota është zhvilluar fillimisht me qëllim që të përdoret për matrica të rralla me një strukturë të rregullt (për shembull një matricë me një numër të vogël diagonalesh) dhe më pas është praktikuar edhe për matricat e rralla me strukturë çfarëdo.

Për një matricë të rrallë A me strukturë jo të rregullt, mund të përcaktohet një parakushtëzim i trajtës $K = (D - E)D^{-1}(D - F)$ duke kërkuar që vetëm elementet e diagonales së K të përputhen me ato të A . Sidoqoftë, kjo nuk do të japë të njëjtin faktorizim ILU si ai i bazuar në eliminimin e IKJ të Gauss-it. Prodhimi i L dhe U nuk i ndryshon vlerat e elementeve ekzistuese në pjesën e sipërme, përveç atyre të diagonales.

Kjo qasje tani mund të zgjerohet për të përcaktuar faktorizimin ILU(1) si dhe faktorizimet me nivele më të larta plotësimi.

2.2 Faktorizimi Cholesky jo i plotë (IC)

Faktorizimi Cholesky jo i plotë përdoret për të zgjidhur në mënyrë iterative një sistem ekuacionesh lineare $Ax = b$ ku A është një matricë simetrike dhe pozitivisht e përcaktuar. Faktorizimi IC mbi një matricë simetrike pozitivisht të përcaktuar, është version i faktorizimit Cholesky për rastin e matricave apo sistemeve të rralla. IC përdoret shpesh si një parakushtëzues për algoritme të tilla si Metoda e Gradientit të Konjuguar (CG).

Nëse A është një matricë pozitivisht e përcaktuar atëherë faktorizimi Cholesky është i trajtës $A = LL^T$ ku L është një matricë trekëndore e poshtme [16]. Në mënyrë të ngjashme faktorizimi jo i plotë Cholesky jepet me anë të një matrice të rrallë trekëndore të poshtme K sa më afër L pra parakushtëzuesi përkatës është KK^T [14].

Ekzistojnë tre strategji të përgjithshme, të cilat përdoren për të garantuar ekzistencën e faktorizimit Cholesky jo të plotë, të cilat sipas rendit rritës të kompleksitetit janë:

- Rritja e peshës së diagonales kryesore të A me ndërrime para ose pas parakushtëzimit të diagonaleve lokale apo globale.
- Zëvendësimi i A me një M -matricë të përafërt \hat{A} , kryerja e faktorizimit IC për $\hat{A} \approx \bar{L}\bar{L}^T$ dhe përdorimi i saj si një parakushtëzues për sistemin origjinal.
- Mos modifikimi i A -së por rimodelimi i faktorizimit në një formë që shmang dështimin e metodave iterative.

Koncepti i plotësimit që është shpjeguar në rastin e faktorizimit ILU vlen edhe për faktorizimin $A = LL^T$, duke rezultuar në faktorizimin IC pa plotësim ndryshe IC(0). Për disa matrica me struktura të veçanta, parakushtëzimi IC(0) implementohet me të njëjtën kosto për iteracion si zbatimi i metodës iterative pa parakushtëzim (një shembull është metoda CG).

Parakushtëzimet ILU(0) dhe IC(0) janë shumë të thjeshtë për t'u implementuar, ndërtimi dhe llogaritja e tyre nuk është proces i kushtueshëm sidomos në rastin e diskretizimit të EDDP-ve eliptikë, duke rezultuar në M -matrica dhe matrica me diagonale dominante. Megjithatë, për probleme më të vështira dhe reale, faktorizimet jo të plota pa plotësim japin një përafrim shumë më të papërpunuar të A . Për këtë arsye lind nevoja e përdorimit të parakushtëzime më të sakta, siç janë ato me nivele.

Një teknikë tjetër e njohur që përdor IC është faktorizimi kompakt IC [17], propozuar nga M.A. Ajiz dhe A. Jennigs (1984). Kjo teknikë bazohet në idenë e eliminimit të stabilizuar. Sa herë që një element l_{ij} i faktorit jo të plotë Cholesky eliminohet, vetë $|l_{ij}|$ ose një version i shkallëzuar i tij i shtohet elementeve përkatëse të diagonales kryesore a_{ii} dhe a_{jj} . Në këtë mënyrë, faktori jo i plotë Cholesky L është ekzakësisht faktori Cholesky i një matrice të turbulluar $\hat{A} = A + C$ ku C është matrica e eliminimit dhe gjysmë pozitivisht e përcaktuar. Në këtë mënyrë shmangen dështimet. Për më tepër, ndarja e A si $\hat{A} - C$ është konvergjente, në kuptimin që rrezja spektrale $\rho(I - \hat{A}^{-1}A) < 1$, prandaj matrica trekëndore e faktorizimit jo të plotë dhe e transpozuar e saj nuk mund të bëhen të paqëndrueshme. Kjo strategji krahasuar me parakushtëzimet standarde jo të plota Cholesky (nëse ata krijohen pa dështuar) jep rezultate me cilësi më të ulët, pasi me uljen e nivelit të plotësimit për faktorët, rritet numri i modifikimeve që duhen zbatuar mbi diagonalen.

KAPITULLI 3

Analiza empirike e metodave të nënhapësirës Krylov

Për të analizuar sjelljen dhe vetitë e metodave Krylov për zgjidhjen e sistemeve me ekuacione të rralla dhe të mëdha, të përmendura në Kapitullin 1 është kryer një analizë empirike bazuar në këto metoda dhe të zbatuara në kategori të ndryshme problemesh. Për analizën empirike janë marrë parasysh përmasat e sistemit, rrallësia/dendësia e sistemit (pra numri i elementeve të ndryshme nga zero), simetria/jo-simetria e sistemit, janë konsideruar sisteme të mirëkushtëzuara apo keqkushtëzuara si dhe sisteme të rastit. Janë aplikuar dy llojet e parakushtëzimit pa plotësim, Incomplete Cholesky Factorization IC(0) dhe Incomplete LU Factorization ILU(0) duke alternuar disa teknika parakushtëzimi si parakushtëzimi i majtë, i djathtë dhe i dyanshëm. Kjo e bën më të lehtë marrjen e një vendimi për të zgjedhur një teknikë të përshtatshme bazuar në vetitë e A , krahasuar me variante të tjera të cilat përdorin vetëm një teknikë specifike [19]. Në secilin rast rezultatet empirike tregojnë teknikën më efektive bazuar në vetitë e sistemit përkatës dhe mbështeten nga informacioni teorik lidhur me metodat e nënhapësirës Krylov [18].

Më poshtë është paraqitur informacioni i përgjithshëm në lidhje me gjashtë nga metodat e implementuara në Matlab të cilat janë përshtatur në trajtën e një pakete ndihmëse që përdor versionet bazë e të rifilluara të metodave si dhe për të zbatuar parakushtëzime të ndryshme. Skripti i metodave dhe përshkrimi i detajuar i ndryshoreve hyrëse dhe dalëse për secilin rast janë paraqitur në Shtojcën A.

```
gmres=modified_GMRES(x0,MaxIt,tol,A,b,method,K1,K2,restart);
```

Algoritmi GMRES është modifikuar që të implementohet me ose pa parametrin e rifillimit duke përdorur $restart = 1$ për GMRES standarde, si dhe të konsiderojë parakushtëzim ose jo duke përdorur treguesit (inpute) 'K1' & 'K2'. Treguesi 'method' përdoret për të zgjedhur midis dy versioneve të GMRES përkatësisht $method = 1$ i përket GMRES me ortogonalizimin e modifikuar të Gram-Schmidt dhe $method = 2$ i përket GMRES për sistemet komplekse.

```
[x,error,fl,iter]=MINRES(A,b,tol,MaxIt,K1,K2,x0);
```

Algoritmi MINRES është modifikuar që të konsiderojë parakushtëzim ose jo duke përdorur treguesit (inpute) 'K1' & 'K2'.

```
[X,R,H,Q]=FOM(A,b,x0);
```

Algoritmi FOM pa parakushtëzim i cili përdor procedurën e modifikuar të Gram-Schmidt dhe një numër maksimal iteracionesh sa N . Gjithashtu FOM është modifikuar që të implementohet me ose pa parametrin e rifillimit.

```
[x,error,fl,iter]=BiCG(x0,M,tol,A,b,K,restart);
```

Algoritmi BiCG është modifikuar që të konsiderojë parakushtëzim ose jo duke përdorur treguesin (input) 'K'. Mund të implementohet me ose pa parametrin e rifillimit.

```
[x,error,fl,iter]=QMR(A,b,tol,MaxIt,K1,K2,x0);
```

Algoritmi QMR pa parakushtëzim i cili përdor iteracionin Lanczos të dyanshëm (algoritmi i Lanczos që përdor bi-ortogonalizimin për sistemet lineare) është modifikuar të konsiderojë parakushtëzim ose jo duke përdorur treguesit 'K1' & 'K2'.

```
[x,error,fl,iter]=BiCGStab(A,b,tol,MaxIt,K1,K2,x0);
```

Algoritmi BiCGStab pa parakushtëzim gjithashtu është modifikuar të konsiderojë parakushtëzim ose jo duke përdorur treguesit 'K1' & 'K2'.

Së pari, janë marrë në konsideratë dy nivele të ndryshme densiteti për matricën e koeficienteve A , përkatësisht më pak se 1% dhe 30% të elementeve të ndryshme nga zero. Është përdorur toleranca $\varepsilon = 0.0001$ dhe numri maksimal i iteracioneve 2500. Si vektor b gjenerohet një vektor i rastit me gjatësi N dhe si përafrim fillestar i zgjidhjes x_0 është marrë vektori $\mathbf{1}$ i të njëjtës gjatësi. Për të krahasuar metodat më njëra-tjetrën është vlerësuar ndryshimi i zgjidhjes së përafërt nga ajo e saktë (e vlerësuar drejtpërdrejt).

3.1 Rasti I

Së pari është marrë një matricë simetrike dhe pozitivisht e përcaktuar, me përmasa 494×494 , elementet e së cilës përfaqësojnë ndryshoret e një problemi të rrjetave nervore. Matrica A ka densitet 0.6827% të elementeve të ndryshme nga zero dhe është e keqkushtëzuar:

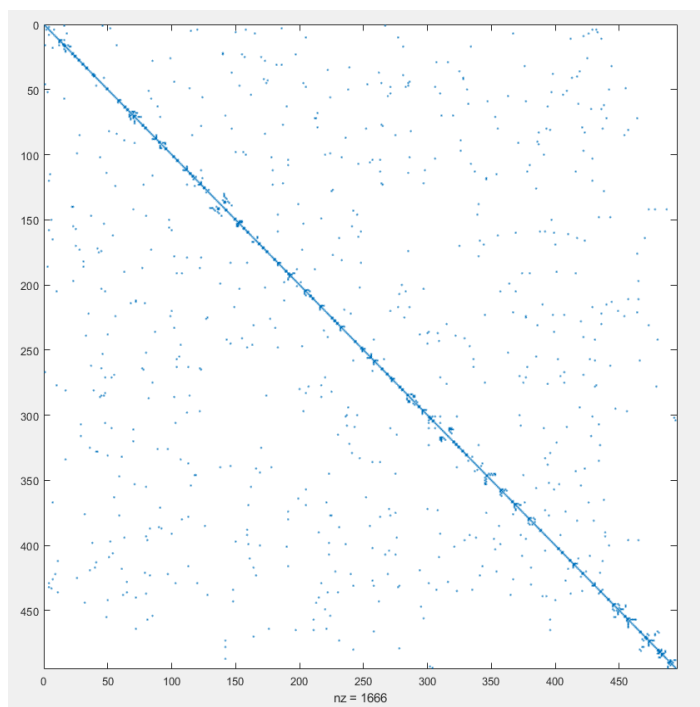


Figura 3.1 Matricë me 1666 elemente të ndryshme nga zero nga 244,036.

Duke zbatuar të gjitha metodat bazike pa rifillim dhe pa parakushtëzim vërehet se metoda që bën një diferencë të dukshme në gabimin e përafërimit krahasuar me metodat e tjera është FOM e cila përdor maksimalisht $N = 494$ iteracione dhe jep një gabim $0.3888e-06$. FOM zakonisht nuk është një metodë e qëndrueshme pasi është e ndjeshme ndaj vetive të sistemit $Ax = b$. Ndërsa BiCGStab është një metodë më e qëndrueshme e cila përsëri jep një përafërim mjaft të mirë të zgjidhjes 0.000694 me numër iteracionesh 1034 . Edhe pse QMR, GMRES and MINRES nuk ndryshojnë shumë nga njëra tjetra për sa i përket zgjidhjes së përafërt (me gabim $0.0012-0.0013$), GMRES bën ndryshim duke konvergjuar më shpejt bazuar në numrin më të vogël të iteracioneve 250 . BiCG konvergjon mbas 880 iteracionesh me gabim 0.0088 .

Tabela 3.1. Metodatat e rifilluara me parametër 1 deri në 10.

Metoda	GMRES		MINRES		BiCG		QMR		BiCGStab		FOM
	Gabimi	It.	Gabimi	It.	Gabimi	It.	Gabimi	It.	Gabimi	It.	Gabimi
1	0.0012	250	0.0013	908	0.0088	880	0.0012	1026	6.94e-04	1034	3.888e-07
2	0.0011	1	0.0013	1	0.0099	7	0.0012	1	6.94e-04	1	2.00e-10
3	0.0011	1	0.0013	1	0.0083	3	0.0012	1	6.94e-04	1	2.00e-10
4	0.0011	1	0.0013	1	0.0078	3	0.0012	1	6.94e-04	1	2.00e-10
5	0.0011	1	0.0013	1	0.0099	14	0.0012	1	6.94e-04	1	2.00e-10
6	0.0011	1	0.0013	1	0.0072	3	0.0012	1	6.94e-04	1	2.00e-10
7	0.0011	1	0.0013	1	0.0068	3	0.0012	1	6.94e-04	1	2.00e-10
8	0.0011	1	0.0013	1	0.0053	3	0.0012	1	6.94e-04	1	2.00e-10
9	0.0011	1	0.0013	1	0.0063	3	0.0012	1	6.94e-04	1	2.00e-10
10	0.0011	1	0.0013	1	0.0084	2	0.0012	1	6.94e-04	1	2.00e-10

Tabela 3.1 ashtu siç pritej, tregon se rritja e numrit të rifillimit nuk ndikon në rritjen e saktësisë së përafërimit dhe shumica e metodave konvergjojnë mbas iteracionit të parë. Ndryshim bën vetëm metoda BiCG duke qenë se është një nga metodat që është shumë e ndjeshme nga zgjedhja e përafërimit fillestar. Vazhdohet duke zbatuar teknika të ndryshme parakushtëzimi me faktorizimin ILU dhe IC. Përdoret parametri K_1 për të kryer parakushtëzimin ILU nga e djathta, parametri K_2 për të kryer parakushtëzimin ILU nga e majta, parametri K për parakushtëzimin IC nga e djathta dhe parametri K' për parakushtëzimin IC nga e majta.

Tabela 3.2 Teknika parakushtëzimi për Rastin I.

Metoda	ILU dyanshëm	ILU i djathtë	ILU i majtë	IC i djathtë	IC i majtë
GMRES	-	-	-	-	-
MINRES	0.0011, 80	17.8680	18.1915	-	-
BiCG	-	-	-	-	-
QMR	0.0011, 80	0.0012, 865	0.0012, 113	9.6295e-04, 254	0.0010, 231
BiCGStab	8.5883e-04, 59	8.0051e-04, 823	3.5014e-04, 72	0.0011, 269	0.0011, 269

Konvergenca e GMRES dhe BiCG nuk përmirësohen aspak me të dy llojet e parakushtëzimit ILU dhe IC, gabimi rritet shumë. Konvergenca e MINRES përmirësohet me parakushtëzimin ILU të dyanshëm duke zvogëluar numrin e iteracioneve në 80 nga 908. Parakushtëzimi K1 (i majtë) dhe K2 (i djathtë) nuk është i suksesshëm. Jo gjithmonë është e mundur të përmirësohet njëherësh konvergenca dhe gabimi përafrit. Metoda QMR ka të njëjtën saktësi përafrit si në rastin kur nuk është përdorur parakushtëzim por me një reduktim të dukshëm të numrit të iteracioneve kur përdor parakushtëzimin ILU të dyanshëm. Ndërsa parakushtëzimi IC nga e djathta për QMR shkakton një zvogëlim drastik të gabimit të përafrit. Teknika më e dobishme për metodën BiCGStab, rezulton parakushtëzimi ILU nga e majta pasi ndikon si në saktësinë e metodës dhe në shpejtësinë e saj.

3.2 Rasti II

Është gjeneruar një matricë e rastit duke rritur përmasat deri në $N = 1000$ dhe dendurinë deri në 30%:

```
A = sprand(n, n, density);
A = 100*A - 50*spones(A);
```

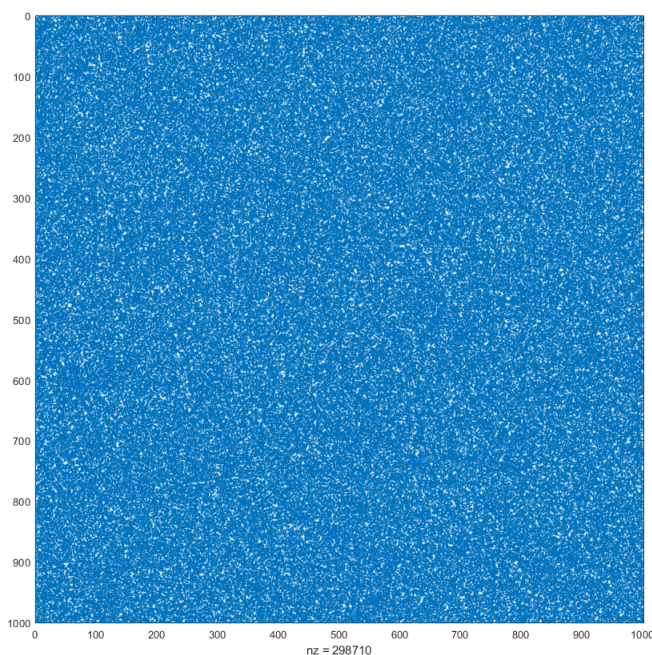


Figura 3.2 Matricë me 298710 elemente të ndryshme nga zero.

Duke qenë se matrica është jo-simetrike, jopozitivisht e përcaktuar, e rastit dhe ka një numër kushtëzimi të madh $\kappa(A) = 6.1951e + 04$, metodat tregojnë një sjellje tjetër. GMRES është e vetmja metodë që tregon stabilitet me një gabim përafrit të kënaqshëm $2.3017e - 13$ dhe numër iteracionesh 1000. Procesi i rifillimit nuk përmirëson as saktësinë dhe as shpejtësinë e metodave. Parakushtëzimi ILU dhe IC, duke përdorur të njëjtat teknika si në Tabelën 3.2 nuk i përmirëson rezultatet, duke hapur në këtë mënyrë rrugën për të kërkuar për lloje të tjera parakushtëzimi që mund të jenë të suksesshëm.

3.3 Rasti III

Në këtë rast është konsideruar një matricë reale jo-simetrike e rendit $N = 4253$ e fituar nga analiza e stabilitetit të zgjidhjeve të ekuacioneve Navier-Stokes. Këto ekuacione përfaqësojnë rrymat mbi turbina dhe matrica është marrë nga The Matrix Market (National Institute of Standards & Technology). Matrica turbullohet duke rritur densitetin e elementeve të ndryshme nga zero rreth diagonales kryesore dhe një paraqitje e saj jepet në Figurën 3.3.

```
load airfoil
x = pow2(x,-32);
y = pow2(y,-32);
n = max(max(i),max(j));
A = sparse(i,j,-1,n,n);
A = A + A';
d = abs(sum(A)) + 1;
A = A + diag(sparse(d));
```

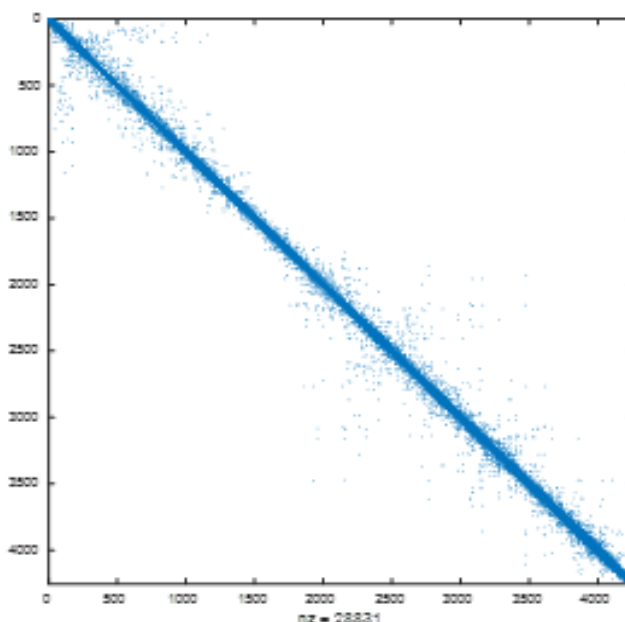


Figura 3.3 Matricë me 28831 elemente të ndryshme nga zero nga 18,088,009.

Në këtë rast toleranca e përdorur është $\varepsilon = 0.001$ dhe numri maksimal i iteracioneve vënë në dispozicion është 1000. Si vektor b është marrë vektori y dhe numri i kushtëzimit në këtë rast është 19.

Gabimi i metodës GMRES është i qëndrueshëm kur procesi i rifillimit përdor parametër mbi 6, me gabim mesatar 3% dhe normë mesatare të mbetjes relative 0.08%. Përafrimi më i mirë arrihet duke rifilluar procesin mbas 4 iteracioneve me gabim 1.6% dhe normë të mbetjes relative 0.05%, pavarësisht nëse merret në konsideratë numri maksimal i iteracioneve ose jo. Zvogëlimi i tolerancës me 10% çon në zvogëlimin e normës së mbetjes relative nga 10^{-3} në 10^{-5} , por përsëri do të ruhej i njëjti raport në lidhje me tolerancën në secilin rast. Ndërsa gabimi i përafrimit nuk ndryshon.

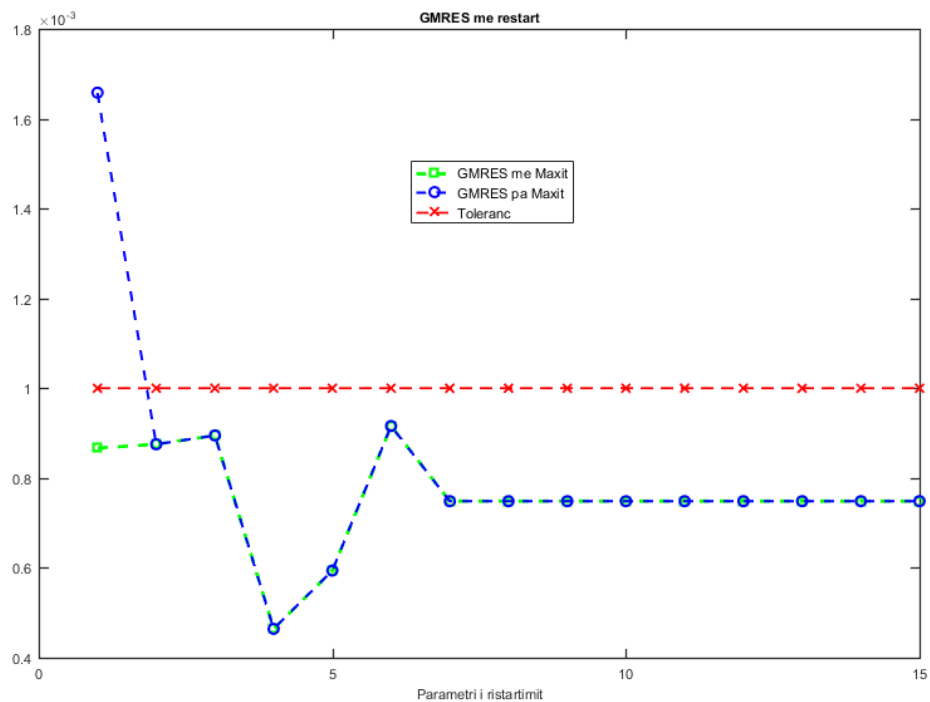


Figura 3.4 GMRES me rifillim për Rastin III.

QMR dhe MINRES kanë të dyja gabim dhe normë të mbetjeve përkatësisht 2.58% dhe 0.07%. BiCG krahasuar me metodat e tjera ka gabimin më të lartë 3.09% dhe mbetjen relative më të lartë 0.09%. Ndërsa, BiCGStab ka gabimin më të vogël 1.95% dhe normën e mbetjes relative 0.06%, më të vogël.

Duke zbatuar teknika të ndryshme parakushtëzimi për rastin e tretë, Figura 3.5 dhe Figura 3.6 tregojnë gabimet e përafritimit nga zgjidhja e saktë dhe normën e mbetjes relative për të gjitha metodat.

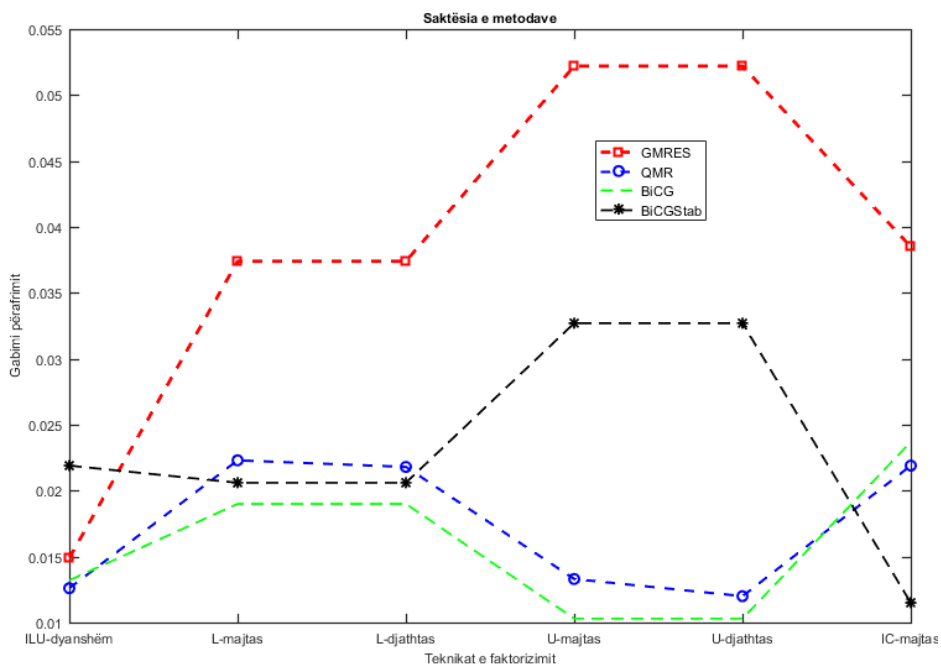


Figura 3.5 Saktësia e metodave me teknika të ndryshme parakushtëzimi, Rasti III.

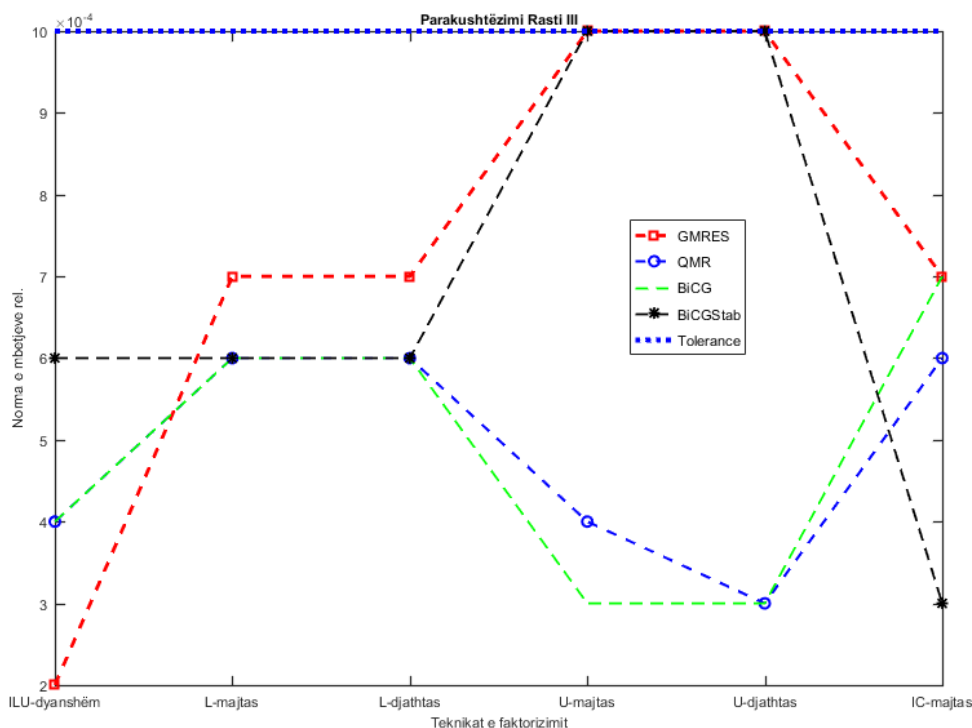


Figura 3.6 Mbetjet relative për teknika të ndryshme parakushtëzimi, Rasti III.

Metoda GMRES me parakushtëzim ILU të dyanshëm redukton gabimin në 1.49% dhe normën e mbetjes relative në 0.02%. Parakushtëzimi me U nga e djathta e metodës QMR zvogëlon gabimin deri në 1.20% dhe normën e mbetjes relative në 0.03%. Parakushtëzimi me ILU të dyanshëm për metodën MINRES zvogëlon gabimin deri në 1.25% dhe normën e mbetjes relative në 0.04%. Të dy llojet e parakushtëzimit ILU me U nga e majta dhe e djathta për metodën BiCG rezultojnë në zvogëlimin e gabimit deri në 1.03% dhe normën e mbetjes relative në 0.03%. BiCGStab e parakushtëzuar jep gabimin më të vogël 1.15% dhe normën e mbetjes relative më të vogël 0.03% me faktorizimin IC.

Tabela 3.3 Tabelë përmbledhëse për Rastin III.

Gabimi	Standard	Rifillim	Parakush.	Gabimi mes.	Mbetjet Rel.	Standard	Rifillim	Parakush.	M.Rel. mes
GMRES	2.58%	1.60%	1.49%	1.89%	GMRES	0.07%	0.05%	0.02%	0.05%
QMR	2.58%	2.58%	1.20%	2.12%	QMR	0.07%	0.07%	0.03%	0.06%
MINRES	2.58%	2.58%	1.25%	2.14%	MINRES	0.07%	0.07%	0.04%	0.06%
BiCG	3.09%	3.09%	1.03%	2.40%	BiCG	0.09%	0.09%	0.03%	0.07%
BiCGStab	1.95%	1.95%	1.15%	1.68%	BiCGStab	0.06%	0.06%	0.03%	0.05%

Parakushtëzimi me skema të ndryshme u jep të gjitha metodave, përveç BiCG dhe QMR, performacë të ngjashme dhe luhatje të dukshme të gabimit dhe mbetjeve relative. BiCG dhe QMR janë metodat më të qëndrueshme mbas të gjitha llojeve të parakushtëzimit të aplikuar. Nga Tabela 3.3 shihet se GMRES është metoda që jep përmirësim të rezultateve nga versioni standard, në versionin me rifillim dhe me

parakushtëzim të dyanshëm ILU. Ndërsa, BiCGStab ka gabimin mesatar më të vogël dhe mbetjen relative mesatare më të vogël.

3.4 Rasti IV

Në këtë rast është shqyrtuar ekuacioni eliptik dypërmasor i difuzion-konveksionit:

$$-K(u_{xx} + u_{yy}) + v_1 u_x + v_2 u_y + 2R^2 u = S \quad (3.1)$$

me kushte kufitare të tipit Dirichlet $u = g$ në Γ , i cili shpjegon fenomenin fizik sesi temperatura, energjia apo madhësi të tjera fizike transferohen në një sistem fizik Ω si pasojë procesit të përhapjes (difuzionit) dhe procesit të konveksionit (lëvizja e një grupi molekulash në lëngje ose gazra). K -koeficienti i difuzionit (termal ose lëngut); (v_1, v_2) -komponentet e shpejtësisë për drejtimet x dhe y ; R^2 -koeficienti i njësisë së prodhimit dhe S - burimi i prodhimit të u .

Trajtimi i këtij ekuacioni gjen zbatime në problemet e sistemeve stokastike: rrymat e ajrit dhe deteve; transportit të ndotjes atmosferike (parashikime); parashikimit të motit apo transportit të nxehtësisë në përcjellëse apo lëngjeve në trupa me pore.

Konkretisht merret në konsideratë ekuacionin:

$$-(u_{xx} + u_{yy}) + u_x + u_y = -2y^2 - 2x^2 + 2xy^2 + 2yx^2 \quad (3.2)$$

i tillë që në kufirin Γ të zonës $\Omega = [0,1]^2$ merr vlerat e zgjidhjes së saktë:

$$u(x, y) = x^2 y^2 \quad (3.3)$$

Pra $K = v_1 = v_2 = 1$; $R^2 = 0$; $S = -2y^2 - 2x^2 + 2xy^2 + 2yx^2$ dhe $g = x^2 y^2$.

Duke zbatuar Metodën e Elementeve të Fundme (MEF), ekuacioni (3.2) diskretizohet nëpërmjet hapave në vazhdim [20].

Së pari bëhet diskretizimi i zonës Ω me elemente të fundme (treëndësha ose katërkëndësha), ku fillimisht fillohet me një rrjetë të ‘rrallë’ dhe më pas rafinohet duke ndarë secilin element në 4 elemente më të vegjël. Si nyje merren kulmet, qendrat apo meset e faqeve të elementit.

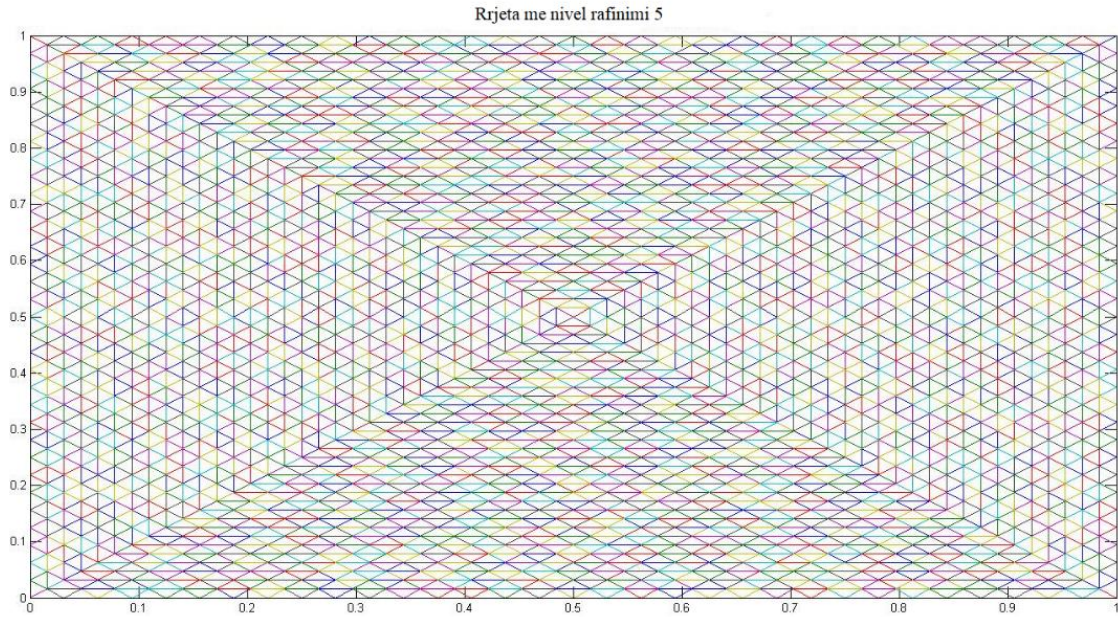


Figura 3.7 Rrjeta për nivel 5 rafinimi, me 4096 trekëndësja dhe 2113 nyje.

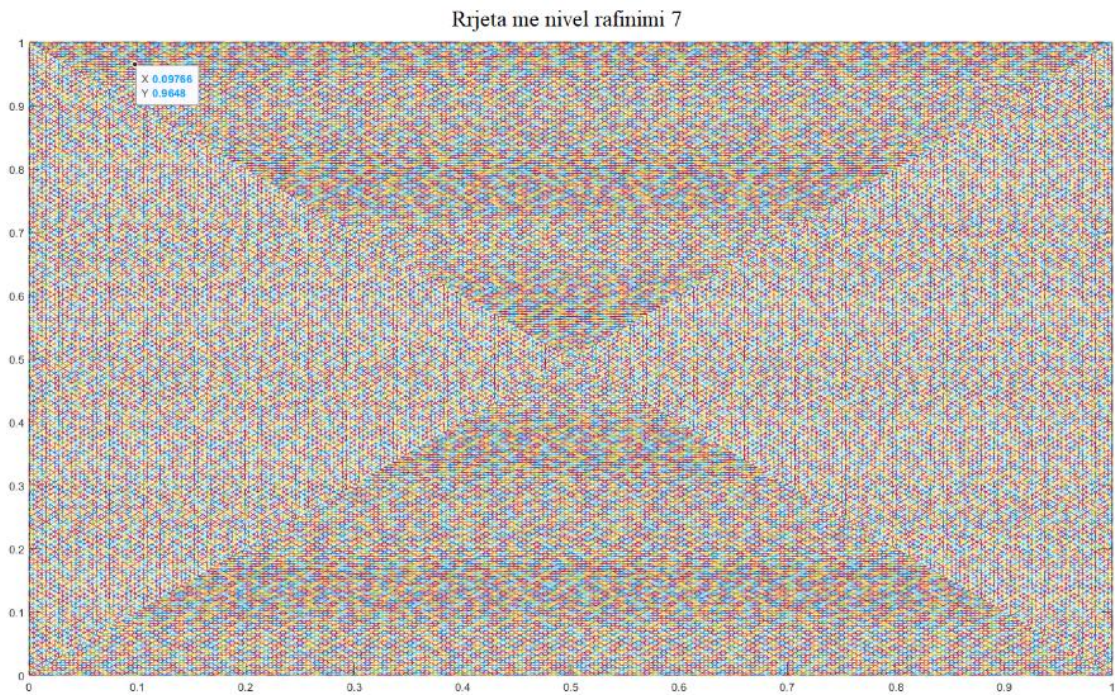


Figura 3.8 Rrjeta për nivel 7 rafinimi, me 65536 trekëndësja dhe 33025 nyje.

Hapi i radhës është formimi i ekuacioneve të MEF pra zgjidhja përafrohet me një funksion që është kombinimi linear i një sistemi funksionesh formë linearisht të pavarura, që janë përafime të secilës nyje të brendshme të Ω . Bashkësia e këtyre funksioneve është bashkësi pjesë-pjesë polinomesh me fuqi fikse (Polinome lineare me fuqi 1 në rastin tonë, pasi zona ndahet në trekëndësja). Për secilën nyje të brendshme llogaritet matrica e një sistemi 3×3 ose 4×4 nisur nga funksioni përkatës i formës. Këto nënsisteme të secilës nyje duke u bashkangjitur dhe nyjet kufitare formojnë një

sistem linear global që përfaqëson më së miri gjithë zonën Ω . Në Figurën 3.9 jepet trajta e matricave të sistemeve lineare të fituara me dy nivelet e rafinimit, 5 dhe 7.

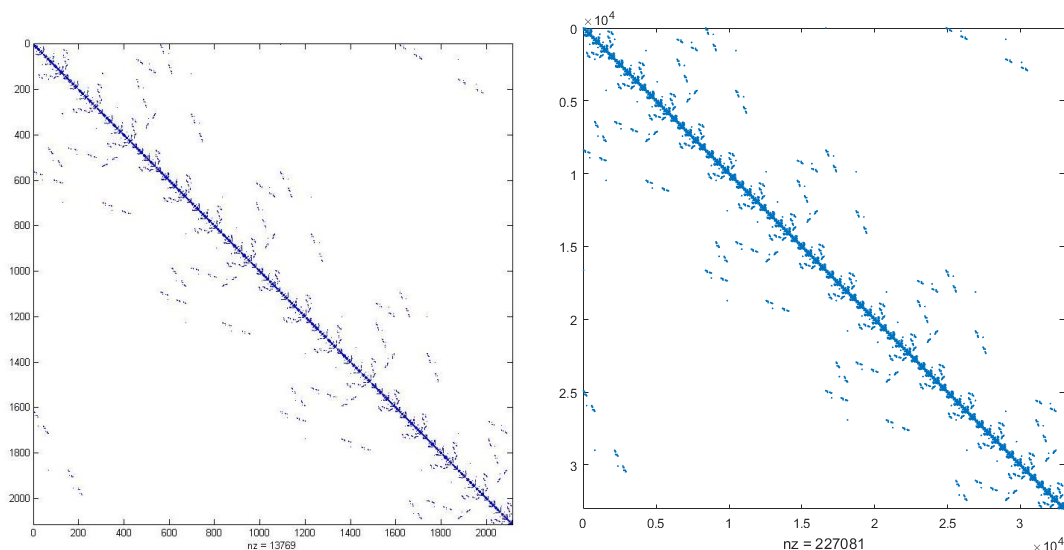


Figura 3.9 Majtas matrica 2113×2113 me 13769 elemente të ndryshme nga zero për nivel 5 rafinimi dhe në të djathtë matrica $33,025 \times 33,025$ me 227081 elemente të ndryshme nga zero për nivel 7 rafinimi.

Për një nivel optimal rafinimi 7 fitohet një matricë e rrallë e përmasave $33,025 \times 33,025$ me $\kappa(A) = 6.1951e + 04$. Me rritjen e niveleve të rafinimit më shumë se 7 nuk ndikon në përmirësimin e zgjidhjes numerike krahasuar me atë analitike në (3.3). Duke zbatuar të gjitha metodat në versionin e tyre standard vërehet se BiCG jep zgjidhjen më të mirë numerike që është më afër asaj reale (3.3) dhe gabim përafrimi më të vogël 0.0016 krahasuar me metodat e tjera. Gjithsesi ky përafrim nuk është mjaftueshëm i mirë dhe lind nevoja e parakushtëzimit të sistemit ashtu si dhe është tipike e modeleve të diskretizuar që lindin nga probleme reale të ekuacioneve diferenciale me derivate të pjesshme. Teknika më efikase e parakushtëzimit për të gjitha metodat ka rezultuar ILU i dyanshëm që jep të njëjtën saktësi përafrimi për zgjidhjen numerike mjaftueshëm afër asaj reale. QMR është metoda që jep gabim përafrimi $1.5692e-13$ edhe gabim absolut $2.1608e-04$ më të mirë. Metoda që tregon dukshëm përmirësim të konvergencës mbas zbatimit të parakushtëzimit është GMRES, krahasuar me atë pa parakushtëzimit.

Tabela 3.4 Gabimet për metodën QMR me parakushtëzim.

Gabimi	Norma mbetjeve	Norma rel. mbetjeve	Gab. abs
QMR	$1.5692e-13$	$9.5730e-15$	$2.1608e-04$

KAPITULLI 4

Zbatime të nënhapësirave Krylov në të dhëna reale

Problemet reale të botës moderne gjenerojnë një sasi shumë të madhe të dhënash çdo ditë. Shkencëtarët mund të japin një kontribut mjaft të mirë me anë të punës kërkimore në fusha të ndryshme si studimi i rrjetave komplekse reale, përpunimi i sinjaleve apo imazheve dhe gjenetikë, duke përdorur pajisje teknologjike të avancuara. Përdorimi i këtyre pajisjeve vjen nga ana tjetër edhe me një mangësi, pasi këto të dhëna të pa përpunuara janë shumë të vështira për t'u manipuluar, për të nxjerrë informacion të vlefshëm prej tyre apo për t'u ruajtur në kujtesë. Përpunimi i të dhënave shndërron pikërisht këto të dhëna të papërpunuara në informacion lehtësisht të kuptueshëm duke përdorur teknika të ndryshme. Qasjet e zakonshme të cilat përdorin bashkësinë e plotë të të dhënave nuk janë praktike teksa përmasat dhe ndryshoret e problemit rriten shumë. Prandaj, lind nevoja e marrjes në konsideratë vetëm të ndryshoreve më të rëndësishme nga gjithë bashkësia apo e transformimit të bashkësisë së të dhënave në një nënbashkësi më të vogël që gjithashtu ruan informacionin e rëndësishëm apo vetitë më të rëndësishme të bazës së të dhënave reale. Mbasi kjo fazë parapërgatitore e të dhënave është zbatuar atëherë është i mundur zbatimi i strategjive dhe metodave më komplekse mbi këto të dhëna.

Bashkësi gjigande të të dhënave shpesh fitohen nga:

- studimi i rrjetave reale biologjike, sociale, biomjekësore
- përpunimi i sinjaleve, imazheve dhe videove.
- analiza gjenetike (veçanërisht shprehja e gjeneve)

Zakonisht, në këto lloje të dhënash numri i ndryshoreve p është më i madh se numri i vrojttimeve n .

Të dyja qasjet e zakonshme, si për renditjen e kulmeve sipas rëndësisë në rrjeta gjigande dhe për gjetjen e nënhapësirës k -përmasore mbizotëruese për të dhëna me përmasa shumë të mëdha, janë shumë të kushtueshme.

Nënhapësirat Krylov janë një mjet i mirë për analizimin e rrjetave komplekse dhe vlerësimin e nënhapësirës mbizotëruese (të rëndësishme) në probleme të reduktimit të përmasave bazuar në vetitë e tyre të ortogonalitetit, minimizimit të mbetjeve, konvergjencës së shpejtë dhe menaxhimit të kujtesës. Këto nënhapësira mund të përdoren për vlerësimin e matësve të qendërsisë, komunikueshmërisë, Indeksit Estrada e jo vetëm. Analiza spektrale gjithashtu mund të kryhet duke përdorur nënhapësirat Krylov për të përafëruar k -vlerat vetjake më të mëdha dhe vektorët vetjakë përkatës.

4.1 Analiza e rrjetave komplekse

Rrjetat janë një mënyrë për të përfaqësuar këto të dhëna gjigande dhe të rralla. Ato paraqesin marrëdhënien (brinjët E) midis entiteteve (kulmet V) të një strukture të organizuar, nëpërmjet të dhënave diskrete $G(V, E)$. Duke përdorur funksionin $f(A)$ të matricës së fqinjësisë së rrjetës A , mund të vlerësohet rëndësia e elementeve (kulmeve) të saj. Kulmet e rëndësishëm japin informacion mbi ndikimin lokal apo global që ato kanë në të gjithë rrjetën. Ky lloj informacioni merret nga vlerësimi i qendrësive të ndryshme të kulmeve të rrjetës. Elementet e diagonales së funksionit të matricës së fqinjësisë $[f(A)]_{ii}$ si dhe shuma e rreshtave të saj $[f(A)\mathbf{1}]_i$ mund të përdoren gjithashtu si vlerësues të rëndësisë së kulmeve, kur duhet të merret parasysh si rëndësia e tyre lokale edhe ajo globale. Në këtë rast mund të përdoren qendrësia eksponenciale e nëngraftit dhe komunikueshmëria totale e nëngraftit të cilat përdorin përkatësisht funksionin eksponencial të matricës së fqinjësisë A . Duke qenë se rrjetat e përdorura për zbatime janë të gjitha të pa orientuara, konceptet teorike të prezantuara në këtë kapitull i përkasin vetëm këtij lloji rrjetash.

Qendërsi eksponenciale të nëngraftit për një kulm i do të quajmë $SC(i) = [e^A]_{ii}$ që jep numrin e udhëve të mbyllura (cikleve) që nisin dhe përfundojnë në kulmin i , duke peshuar me $\frac{1}{k!}$ udhët me gjatësi k . Me fjalë të tjera, sa më të gjata këto cikle, aq më pak konsiderohen të rëndësishme në vlerësimin e qendrësisë së kulmit i . Peshat nevojiten për të “penalizuar” udhët më të gjata dhe që seria e fuqisë të konvergjojnë. Elementet jashtë diagonales kryesore janë gjithashtu të rëndësishme, ato mund të vlerësojnë komunikueshmërinë midis një çifti kulmesh i dhe j . Komunikueshmëria e nëngraftit $[e^A]_{ij}$ midis dy kulmeve numëron numrin total të udhëve midis kulmeve i dhe j , duke peshuar me $\frac{1}{k!}$ udhët me gjatësi k . Pra $[e^A]_{ij}$ mat se sa lehtë shkëmbejnë informacion kulmet e gjithë rrjetës me njëri tjetrin. Informacion me vlerë mbi rëndësinë e kulmeve mund të nxirret edhe nga elementet e secilit rresht të e^A dhe është shumë më pak e kushtueshme të përafrohet shuma e këtyre elementeve me metoda iterative sesa elementet e diagonales apo të gjithë matricës. Shuma e rreshtave $\sum_{j=1}^n [e^A]_{ij}$, $1 \leq i \leq n$ mund të përdoret si një matës i qendrësisë dhe jep një vlerësim se sa mirë komunikon një kulm me të gjithë të tjerët [21].

Kështu kur numërohen të gjitha udhët midis kulmit i dhe të gjithë kulmeve të tjerë përfshirë këtu edhe vetë kulmin i , duke peshuar me faktor penalizimi $\frac{1}{k!}$ udhët me gjatësi k , kemi atë që quhet komunikueshmëria totale e nëngraftit për kulmit i pra $TSC(i) = [e^A\mathbf{1}]_i$. Më pas, duke mbledhur $TSC(i)$ për secilin kulm i , marrim komunikueshmërinë totale të rrjetës, e cila mat se sa efikas është komunikimi përgjatë gjithë rrjetës:

$$TC(A) = \sum_{i=1}^n [e^A\mathbf{1}]_i = \sum_{i=1}^n \sum_{k=1}^n e^{\lambda_k} (V_i^T \mathbf{1}) v_{ki} = \mathbf{1}^T e^A \mathbf{1}$$

ku v_{ik} është elementi i i -të i vektorit vetjak të normalizuar v_k që i përket vlerës së k -të vetjakë λ_k për matricën A . Pra komunikueshmëria totale e rrjetës përafrohet shumën e të gjithë elementeve të A , duke mos qenë nevoja të njihet secili element në mënyrë individuale.

Qendërsia eksponenciale e nëngrafit dhe komunikueshmëria totale e nëngrafit pritet të japin të njëjtën renditje sipas rëndësisë për kulmet e rrjetës kur distanca spektrale e saj $|\lambda_{max1} - \lambda_{max2}|$ është mjaftueshëm e madhe, ku λ_{max1} vlera më e madhe vetjake dhe λ_{max2} vlera e dytë me e madhe vetjake e A [22]. Nënhapësirat Krylov përdoren për përafrimin e këtyre vlerave vetjake dhe vektorëve vetjakë përkatës duke na dhënë kështu informacion të vlefshëm mbi vetitë spektrale të matricës që mund të na shërbejë më pas për të krahasuar saktësinë e matësve të qendërsisë dhe për ti krahasuar edhe ndërmjet tyre.

Përveç funksionit eksponencial e^A mund të përdoret edhe $(I - \alpha A)^{-1}$ për vlerësimin e qendërsive të përmendura më lart por e para jep vlerësime më të mira në lidhje me renditjen e kulmeve sipas rëndësisë (vlera më të mëdha tregojnë rëndësi më të madhe të kulmit), bazuar në koeficientet e korrelacionit Pearson. Gjithashtu nëse përdorim komunikueshmërinë totale të nëngrafit bazuar në funksionin e anasjelltë $\sum_{i=1}^n \sum_{j=1}^n [(I - \alpha A)^{-1}]_{ij}$, ndryshe nga ai në trajtë eksponenciale, nuk bën ndryshimin midis rrjetave të lidhura dhe të lidhura fort, në rastin e rrjetave të orientuara.

Duhet pasur parasysh se renditjet e kulmeve sipas qendërsisë eksponenciale të nëngrafit dhe qendërsisë së komunikueshmërisë totale konvergjojnë shpejt drejt qendërsisë së vektorit vetjak (një vlerësues ky i ndikimit global të kulmeve) kur distanca spektrale e A është mjaftueshëm e madhe. Prandaj, qendërsia e vektorit vetjak, e cila për kulmin i , përdor vektorin vetjak $v_i = \frac{1}{\lambda_i} \sum_{j \in N(i)} x_j = \frac{1}{\lambda_i} \sum_j A_{ij} x_j$ që i përket vlerës vetjake λ_i dhe kulmeve të tij fqinjë $N(i)$, mund të përdoret në vend të dy matësve të parë në këtë rast [22]. Për të përcaktuar se kur distanca spektrale $|\lambda_{max1} - \lambda_{max2}|$ është mjaftueshëm e madhe është një procedurë komplekse më vete.

Metodat e nënhapësirave Krylov janë metoda efikase për llogaritjen e eksponencialit të matricave të rralla dhe të mëdha. Përafrimi i elementeve të diagonales kryesore kur përdoren metodat Krylov bëhet me anë të përafrimit $exp(A) \approx V_m * exp(T_m) * V_m^T$ ku $V_m = [v_1 \ v_2 \ \dots \ v_n]$ bazë ortogonale e \mathcal{K}_m bazuar në iteracionin standard të Lanczos [23] dhe trajta e algoritmit të përdorur jepet në Shtojcën B. Përafrimi i shumës së rreshtave bëhet nëpërmjet prodhimit të eksponencialit të matricës me një vektor, pra $[e^A \mathbf{1}]_i \approx \|\mathbf{1}\|_2 * V_m * e^T * e_1$ ku $\mathbf{1} = [1, 1, 1, \dots, 1]$ dhe $e_1 = [1, 0, 0, \dots, 0]$.

Përafrimi i e^A për rrjetat e paorientuara varet shumë nga diametri $d(G)$ që jepet si maksimumi i largësive për çdo dy kulme të rrjetës. Sa më e madhe gjatësia e shtegut më të shkurtër në rrjetë, aq më pak kohë nevojitet për përafrimin e elementeve të e^A .

Për shkak të kujtesës së kufizuar të pajisjeve kompjuterike, ruajtja e bazës ortonormale të gjeneruar nga procesi standard i Lanczos për nënhapësirën e m -të Krylov përbën një sfidë të vërtetë. Për më tepër gjatë këtij procesi vetia e ortogonalitetit të vektorëve të

bazës sa vjen e zbehet. Procesi i ri-ortogonalizimit të vektorëve siç e kemi përmendur mund të përmirësohet duke rifilluar iteracionin standard të Lanczos disa herë. Procesi i rifillimit prodhon vargun e zërthimeve Lanczos $AV_m^{(k)} = V_m^{(k)}T_m^{(k)} + \beta_m^{(k)}v_{m+1}^{(k)}e_m^T$ ku shtyllat e matricës $V_m^{(k)} = [v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)}]$ formojnë një bazë ortonormale të \mathcal{K}_m , $T_m^{(k)}$ matricë tridiagonale dhe e_m vektor njësi [24].

Janë marrë në konsideratë katër rrjeta reale të paorientuara me veti të ndryshme nga njëra tjetra dhe në Tabelën 4.1 është dhënë informacioni i përgjithshëm lidhur me to, ku përfshihet numri i kulmeve, numri i elementeve të ndryshme nga zero, fuqia minimale dhe maksimale e kulmeve, numri i komponenteve të lidhura, diametri dhe koeficienti i klasterizimit $C(i)$ që mat prirjen e kulmeve të rrjetit për të formuar cikle me gjatësi tre (pra trekëndësha).

Tabela 4.1 Informacion i përgjithshëm mbi rrjetat.

Rrjeta	Sc_collab	Bio-SC-Protein	Erdos972	FB Harward Friendship
Kulmet	143	1458	5488	15126
Elem. jozero	1246	3896	14170	1649234
d_{max}	42	56	61	1183
d_{min}	1	1	0	1
Nr. komp. lidhura	1	1	754	20
Diametri	8	19	13	11
λ_{max1}	13.4293663	7.53500469	14.4481963	247.368562
λ_{max2}	11.9062337	7.11050035	11.8863344	155.061763
C(G)	0.433907	0.0708304	0.0825312	0.2123338

Burimi i rrjetave të marra për eksperimente është <http://networkrepository.com/> dhe përfshihen rrjeta të llojeve të ndryshme si rrjeta bashkëpunimi (rrjeta e bashkëpunimit në punime shkencore), biologjike (rrjeta e ndërveprimit midis proteinave të majasë), sociale (rrjeta e miqësisë në FB për Universitetin Harward) dhe të tipit Erdos (rrjeta të rastit me shpërndarje binomiale të fuqisë së kulmeve).

Figura 4.1 jep përfaqësimin e bazës së të dhënave Bio-SC-Protein nëpërmjet një rrjete. Rrjetat e ndërveprimit midis proteinave (PPI), në këtë rast të majasë, përmbajnë informacion në lidhje me mënyrën sesi proteinat veprojnë me njëra tjetrën për të mundësuar proceset biologjike brenda qelizave. Rrjetat PPI janë rrjeta me shpërndarje fuqi të fuqisë së kulmeve për këtë arsye proteinat me shkallë të lartë lidhshmërie janë ato që kanë gjasa të kenë rëndësi më të madhe.

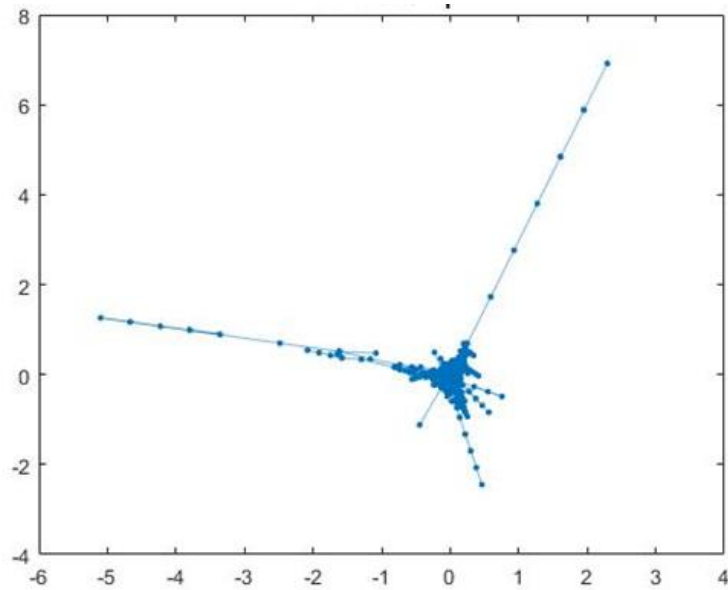


Figura 4.1 Rrjeta Bio-SC-Protein.

Matrica përkatëse e fqinjësisë A për rrjetin Bio-SC-Protein jepet në Figurën 4.2 dhe është një matricë e madhe, simetrike dhe e rrallë me përmasa 1458×1458 . Kjo matricë ka 3896 elemente të ndryshme nga zero.

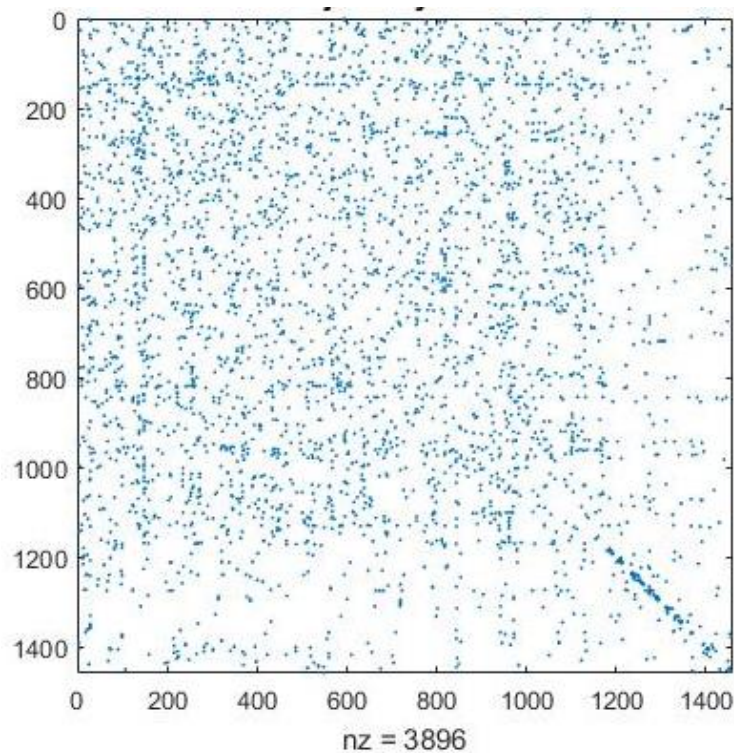


Figura 4.2 Matrica e fqinjësisë për Bio-SC-Protein.

Për të renditur kulmet sipas rëndësisë vlerësohen të gjithë matësit e qendërsisë (shih Shtojcën B për matës të tjerë qendërsie). Mund të zgjidhet të renditen të gjithë kulmet e rrjetave, k -kulmet më të rëndësishëm apo k - përqind kulmet më të rëndësishëm (kjo e fundit parapëlqehet në shumicën e rasteve) dhe më pas të vlerësohen koeficientet e

korrelacionit të Pearson për të parë përputhshmërinë e listave të renditura nga secili matës (Tabela 4.2). Rikujtojmë se vlerat e qendrësishë eksponenciale të nëngraftit $ESC(i)$ dhe të komunikueshmërisë totale të nëngraftit $TSC(i)$ janë përafuar duke përdorur qasjen Lanczos.

Tabela 4.2 Koeficientet e korrelacionit për të gjithë matësit e qendrësishë që janë përdorur për rrjetën Bio-SC-Protein.

Matësit	Page rank	Fuqi	Vek. vetjak	Afërsi	Ndërmjetës	ESC	TSC
Page rank	1	0.979526	0.399938	0.332179	0.806247	0.679343	0.626720
Fuqi	0.979526	1	0.337945	0.423184	0.846312	0.629219	0.664686
Vek. vetjak	0.399938	0.337945	1	0.176210	0.272121	0.761052	0.723954
Afërsi	0.332179	0.423184	0.176210	1	0.437560	0.192252	0.566521
Ndërmjetës	0.806247	0.846312	0.272121	0.437560	1	0.523874	0.686216
ESC	0.679343	0.629219	0.761052	0.192252	0.523874	1	0.747351
TSC	0.626720	0.664686	0.723954	0.566521	0.686216	0.747351	1

Janë renditur 10% kulmet më të rëndësishëm (sipas vlerave të $ESC(i)$ dhe $TSC(i)$) për secilën nga katër rrjetat e marra në shqyrtim dhe vihet re një përqindje e lartë përputhshmërie e këtyre listave deri në 100%, si në rastin e rrjetës Facebook Harward Friendship. Duke marrë në konsideratë listat e të gjithë kulmeve të rrjetave arrihet përsëri një përputhshmëri e lartë deri në 90% të kulmeve.

Tabela 4.3 Koeficientet e korrelacionit për secilën rrjetë dhe për listën e të gjithë kulmeve, vlerësuar nëpërmjet elementeve të diagonales dhe shumës së rreshtave të e^A .

Rrjeta	$\rho(x, y)$
Sc_collab	0.917812888
Bio-Yeast	0.74735124
Erdos972	0.870254696
FB Harward Friendship	0.885172606

Nga Tabela 4.3 vërehet se listat e renditura të kulmeve të rrjetës Bio-SC-Protein, bazuar në vlerat e $ESC(i)$ dhe $TSC(i)$, ndryshojnë dukshëm duke qenë se distanca spektrale $|\lambda_{max1} - \lambda_{max2}| = 0.4245043$, për dy vlerat veltjake më të mëdha $\lambda_{max1} = 7.5350046$ dhe $\lambda_{max2} = 7.1105003$ të përafuara me Krylov, nuk është domethënëse. Ndërsa për rrjetën FB Harward Friendship distanca spektrale është mjaftueshëm e madhe $|\lambda_{max1} - \lambda_{max2}| = 92.306798466308010$ për $\lambda_{max1} = 247.3685621101465$ dhe $\lambda_{max2} = 155.061763643838$, prandaj përputhshmëria është me e lartë krahasuar me rrjetat e tjera të përmasave të mëdha.

Për të krijuar një ide mbi komunikimin e kulmeve nëpër të gjithë rrjetën llogaritet komunikueshmëria totale e grafit për secilin rast. Në rastin e rrjetës FB Harward Friendship komunikueshmëria totale është $TC(A) = 8.898587734948889 e + 110$ që tregon shkëmbim të lartë informacioni midis kulmeve të rrjetës.

Në Tabelën 4.4 duket qartë avantazhi i metodës së rifilluar të Lanczos, krahasuar me metoda klasike si përafrimi Padé dhe metoda Scaling & Squaring të kombinuara [25] në funksionin $\text{expm}()$ në Matlab.

Tabela 4.4 Koha e përafrimit të e^A me metodën e rifilluar të Lanczos kundrejt metodës SSM.

Rrjeta	Metoda Lanczos e rifilluar	Metoda Scaling & Squaring $\text{expm}()$
Sc_collab	0.022819 sek	0.014745 sek
Bio-Yeast	0.504860 sek	0.682185 sek
Erdos972	17.674911 sek	31.668211 sek
FB Harward Friendship	612.854085 sek	891.016200 sek

Avantazhi i iteracionit të rifilluar të Lanczos është më i dukshëm me rritjen e përmasave gjithmonë e më shumë dhe nuk kufizohet vetëm në vlerësimin e kulmeve të rëndësishëm në rrjet [21]. Qasja e iteracionit Lanczos mund të shërbejë edhe për analizimin e vetive të grafit të Laplacit që i korrespondon rrjetës, për vlerësimin e shpërndarjes probabilitare të kulmeve, Indeksit Estrada etj.

Indeksi Estrada llogaritet si shuma e të gjitha qendrësive eksponenciale të nëngraftit pra $EE(A) = \sum_{i=1}^n [e^A]_{ii}$. Në Tabelën 4.5 jepen kufijtë e poshtëm dhe të sipërm për komunikueshmërisë totale të normalizuar të rrjetave, duke përdorur gjithashtu Indeksit Estrada të normalizuar nga numri n i kulmeve të rrjetave (normalizimi mund të bëhet edhe duke përdorur numrin m të brinjëve) dhe vlerën vetjake më të madhe të matricës së fqinjësisë përkatëse të vlerësuar me anë të nënhapësirave Krylov. Për rrjetat e paorientuara kemi $EE(A) \leq C(A) \leq ne^{\lambda_{max}}$ dhe $\lambda_{max} = \|A\|_2$ [22].

Tabela 4.5 Indeksi Estrada i normalizuar, lidhshmëria totale e normalizuar dhe $e^{\lambda_{max}}$.

Rrjeta	$EE(A)/n$	$TC(A)/n$	$e^{\lambda_{max}}$
Sc_collab	3.5105614156e+04	3.0616245890e+05	6.7967252920e+05
Bio-Yeast	127.07864384	263.37040034	1.8724531583e+03
Erdos972	6.6894503602e+04	1.3308050194e+05	1.8826602840e+06
FB Harward Friendship	1.4600284778 e+106	8.5434275152e+106	2.6965068059e+107

Shpërndarja probabilitare e kulmeve llogaritet si $p(i) = ESC(i)/EE(A)$ dhe më poshtë jepet Figura 4.3 e shpërndarjes $p(i)$ për rrjetën FB Harward Friendship.

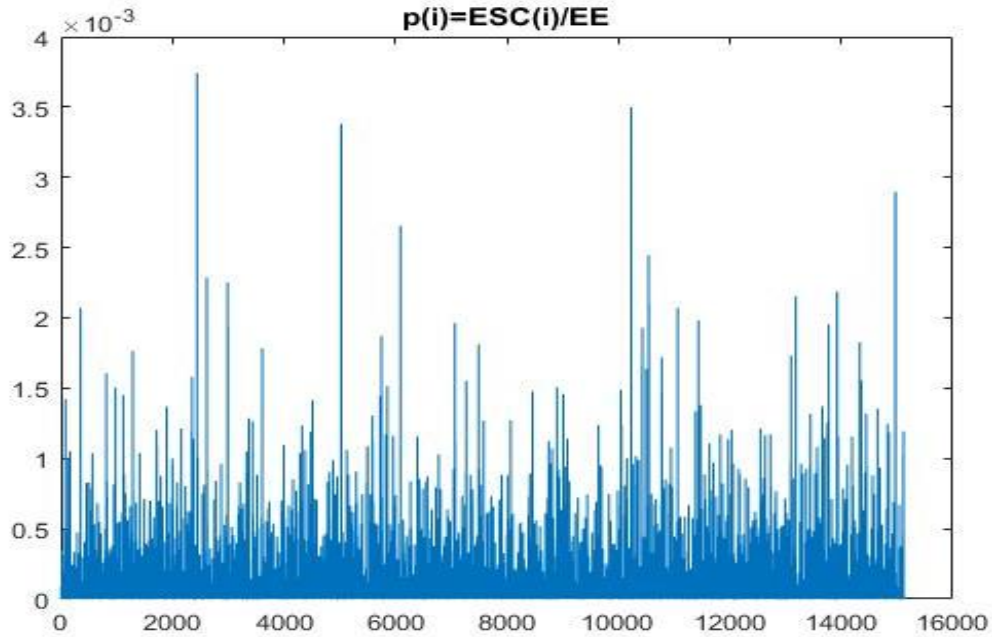


Figura 4.3 Shpërndarja probabilitare e kulmeve të rrjetës FB Harward Friendship.

Laplasiani L i një rrjete është gjithashtu një matricë e madhe dhe e rrallë, prandaj metodat e nënhapësirës Krylov mund të përdoren për të zbatuar mbi këtë matricë teknika të klasterizimit spektral, përafrimin e lidhshmërisë algjebrike (λ_{nim2} vlera vetjake e dytë më e vogël e L) dhe përafrimin e vektorit Feidler (vektori vetjak përkatës i λ_{nim2}).

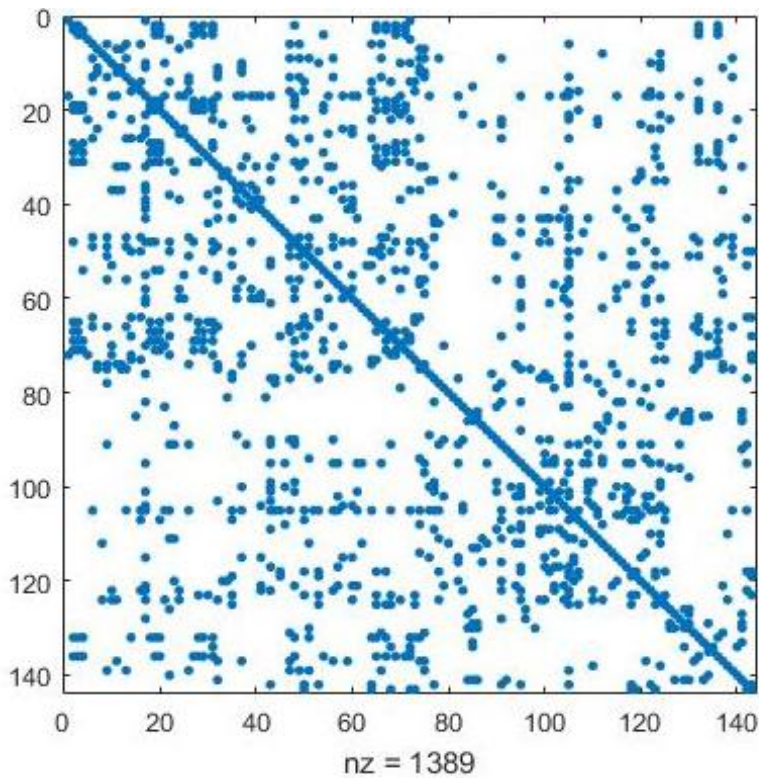


Figura 4.4 Matrica e Laplasianit për rrjetën e bashkëpunimit shkencor Sc_collab.

4.2 Zvogëlimi i përmasave

Zvogëlimi i përmasave është shumë i dobishëm në mjaft fusha si përpunimi i imazheve, sinjaleve apo videove. Sistemet e sigurisë dhe mbikëqyrjes kanë nevojë për përpunimin e imazheve dhe videove në mënyrë që të evidentojnë, përcaktojnë apo numërojnë personat ose objektet e ndryshme që lëvizin në to. Për të identifikuar këto elemente është e nevojshme të veçohet sfondi. Me fjalë të tjera të ndahen elementet statikë në imazhe që quhen *Sfond* nga elementet e tjerë jo statikë që quhen *Plan i parë*, duke ju referuar një imazhi ku sfondi është mjaft i qartë. Objektet në plan të parë nuk kanë kontribut të rëndësishëm në vlerësimin e sfondit për aq kohë sa nuk kanë përmasa të mëdha dhe lëvizin në momente të ndryshme [26].

Zvogëlimi i përmasave gjithashtu sjell përfitime të mëdha edhe në fushën e biomjekësisë, ku të dhënat me përmasa shumë të mëdha përfaqësohen nga mijëra ndryshore (zakonisht gjene) dhe qindra vrojtme. Të dhënat e papërpunuara të quajtura ‘microarray’ të marra nga laboratorët mjekësore janë vargje informacioni në lidhje me ADN, proteinat apo indet. Ato përdoren zakonisht për të studiuar nivelet e shprehjes së gjeneve të një kondite (gjendje) të caktuar shëndetësore në një organizëm. Zvogëlimi i numrit të ndryshoreve mund të zbatohet në këtë rast për të vënë në dukje gjenet më të rëndësishme duke qenë se vetëm disa prej tyre nga gjithë bashkësia e gjeneve të një organizmi luajnë një rol të rëndësishëm në shfaqjen e konditës (gjendje) [27].

Teknika më e njohur lineare e zvogëlimit të ndryshoreve është Analiza e Komponentëve Kryesorë (PCA). PCA shpreh secilën ndryshore të re si një kombinim linear të ndryshoreve origjinale në mënyrë të tillë që ndryshoret e reja të maksimizojnë dispersionin. Një mënyrë për të gjetur nënhapësirën k -përmasore të komponentëve të rëndësishëm është duke përdorur zbërthimin SVD (Singular Value Decomposition) mbi matricën e qendëruar apo të standardizuar të matricës së të dhënave [28]. PCA garanton humbje minimale të informacionit të rëndësishëm përgjatë transformimit dhe ndryshoret e fituara janë të pakorreluara por meqenëse secila nga ndryshoret është një kombinim linear i të gjitha ndryshoreve origjinale, bëhet shumë i vështirë interpretimi i grupeve të reja. Gjithashtu, në rastin e bazave të të dhënave të mëdha që fitohen gjatë përpunimit të imazheve apo të dhënave ‘microarray’, ku numri i ndryshoreve p është shumë herë më i madh se numri i vrojtmeve n , teknika PCA nuk ka rezultate të qëndrueshme [29].

Metoda inovative të nënhapësirës Krylov apo nënhapësirave Krylov në bllok $\mathbb{K}^{(m)}(A, V) = \text{span}\{V, AV, \dots, A^{m-1}V\}$ për një matricë simetrike A dhe V një matricë e rastit, mund të përdoren si në rastin e veçimit të sfondit në imazhe apo video ashtu edhe në përcaktimin e gjeneve më të rëndësishme për një konditë (gjendje) të caktuar.

4.2.1 Vlerësimi i sfondit në imazhe

Komponentët kryesorë të një video janë elementet e matricës së fituar nga n imazhe të veçuara nga filmimi, me përmasa $a \times b$ (gjatësi dhe gjerësi) që janë thuhajse gjithmonë konstant përgjatë këtyre imazheve. Për të vënë në dukje sfondin do të thotë që elementet e planit të parë (jo statik), të cilët mund të shihen si një zhurmë që i shtohet sfondit real (truth background apo shkurt GT), duhen fshirë. Imazhi që përdoret si referencë për sfondin real mund të zgjidhet nga bashkësia e imazheve të prera nga video aty ku nuk ka objekte në lëvizje (zakonisht imazhi i parë i vargut).

Eksperimentet e veçimit të sfondit janë zbatuar në disa bashkësi imazhesh që ju përkasin tre videove të marra nga kamerat e vrojtimit në skenarë të ndryshëm (me pak apo shumë lëvizje). Baza e të dhënave për videot e marra në shqyrtim është bashkësia e të dhënave SBI, marrë nga <https://sbmi2015.na.icar.cnr.it/SBIdataset.html>.

Dy metoda të nënhapësirave Krylov në bllok janë përdorur përkatësisht Metoda LMSVD [30] dhe Krylov PCA [31], (algoritmet e të cilave janë dhënë në Shtojcën B), së bashku me një nga variantet e teknikës PCA të përshtatshme për matricat e mëdha dhe të rralla e quajtur SPCA [32]. SPCA mund të gjendet në paketën *elasticnet* në software-in R dhe në Matlab në paketën *SpaSM*. Këto paketa mund të shkarkohen përkatësisht në:

<https://cran.r-project.org/web/packages/elasticnet/>
<https://www.jstatsoft.org/article/viiv/v084i10>

Rezultatet e këtyre metodave janë krahasuar me sfondin referencë GT. Informacioni i përgjithshëm në lidhje me bashkësinë e imazheve të përzgjedhura jepet në Tabelën 4.6.

Tabela 4.6 Informacion i përgjithshëm në lidhje me bashkësinë e imazheve SBI.

Bashkësia Im.	Rezolucioni	Nr. imazheve (n)	Nr. ndryshoreve (p)
CaVignal	136x200	258	27200
Foliage	144x200	394	28800
HighwayII	240x320	500	76800

Fillimisht, secili imazh nga vargu i secilës video transformohet nga një strukturë trepërmasore $a \times b \times c$ për $c = 1, 2, 3$ (që i përket kanalit të ngjyrave RGB) në tre matrica dypërmasore $a \times b$, duke mbajtur c konstante. Më pas, elementet e secilës matricë kthehen në vektorë rresht me gjatësi $p = a \times b$. Për secilën bashkësi të dhënash duke shqyrtuar n imazhe të marra nga vargu, formohen rrjedhimisht tre matrica të dhënash me përmasa $n \times p$. Të gjitha matricat e të dhënave janë standardizuar para se të zbatohen metodat përkatëse.

Mbas përafrimit të komponentëve kryesorë Y_i për $i = 1:3$ dhe ruajtjen e pikave të projektuara në matricën Z_i për $i = 1:3$ për secilin varg imazhesh [28], të dhënat e projektuara duhet t'i nënshtrohen procesit të destandardizimit duke qenë se të dhënat fillestare u standardizuan. Si hap mbyllës, imazhi i përfutur rindërtohet në trajtën trepërmasore të tij. Sfondet e vlerësuara nëpërmjet metodave SPCA, LMSVD dhe Krylov PCA jepen më poshtë së bashku me GT e tyre.



Figura 4.5 Bashkësia e të dhënave CaVignal.



Figura 4.6 Bashkësia e të dhënave Foliage.



Figura 4.7 Bashkësia e të dhënave HighwayII.

Në videon e parë CaVignal objekt i planit të parë është vetëm një njeri, i cili shfaqet në të njëjtin pozicion pa lëvizur në më shumë se gjysmat e imazheve të bashkësisë dhe më pas zhvendoset ngadalë nëpër imazhe. Prania konstante e objektit edhe pse një i vetëm, ndikon më mënyrë të konsiderueshme në cilësinë e sfondit të vlerësuar (Figura 4.5). Në videon e dytë (Foliage) ndodhen më shumë objekte në plan të parë (gjethe) të cilat pengojnë sfondin por nuk janë aq konstante përgjatë vargut të imazheve krahasuar me videon CaVignal. Në këtë rast të tria metodat japin një rezultat të pranueshëm për vlerësimin e sfondit (Figura 4.6). Ndërsa në videon e fundit HighwayII ashtu si dhe pritej, metodat japin një vlerësim shumë të mirë të sfondit (Figura 4.7) edhe pse në këtë video ka shumë aktivitet dhe lëvizje të objekteve (makina). Kjo vjen si pasojë e lëvizjes së shpejtë të objekteve duke bërë që sfondi të jetë i ekspozuar gjatë gjithë kohës.

Parametri PSNR (peak signal-to-noise ratio) për secilën metodë është llogaritur në Tabelën 4.7, për të bërë krahasimin midis sfondit real dhe sfondeve të fituar me metodat Krylov.

Tabela 4.7 PSNR për secilën metodë dhe për secilën bazë të të dhënave.

Metoda	SPCA	LMSVD	Krylov PCA
Video			
CaVignal	20.01	23.64	20.24
Foliage	12.26	28.05	12.26
HighwayII	32.45	33.18	32.39

Sipas figurave të mësipërme dhe Tabelës 4.7 , LMSVD mbizotëron mbi metodat e tjera për vlerësimin e sfondit pavarësisht llojit të aktivitetit të objekteve në video. Ndërsa, SPCA dhe Krylov PCA kanë saktësi thujtë të njëjtë në përafrim [26]. Në rastin kur objektet në plan të parë lëvizin në mënyrë konstante dhe sfondi është mjaft i dukshëm, SPCA jep një vlerësim më të mirë se sa Krylov PCA. Duhet theksuar se në këtë rast (pra video HighwayII) të gjitha metodat japin PSNR më të lartë krahasuar me videot e tjera. Kur objektet e planit të parë janë më shumë të qëndrueshëm por nuk mbulojnë shumicën e sfondit të imazheve, Krylov PCA jep rezultate më të mira se SPCA. Në rastet kur fondi në shumicën e imazheve nga bashkësia nuk është i ekspozuar, metodat SPCA dhe Krylov PCA nuk bëjnë dallim.

4.2.2 Vlerësimi i gjeneve të rëndësishme

Lloji më i njohur i të dhënave ‘microarray’ është vargu i të dhënave të ADN ku çdo element përmban miliona kopje të molekulave unike të ADN që i përkasin një gjeni. Për të kryer matjet e niveleve të shprehjes së gjeneve krahasohen dy kampionë të të njëjtit gjen, ku njëri është kampioni referencë i mbajtur në kushte normale dhe tjetri trajtohet me teknika apo mjekime të ndryshme. Këto lloje të dhënash mund të përfaqësohen nga një matricë e përmasave të mëdha e cila quhet matrica e shprehjes së gjeneve. Secila shtyllë e kësaj matrice i përket një lloji të ndryshëm gjeni dhe secili rresht i përket vrojtimeve të konditës (gjendjes) shëndetësore.

Për këtë zbatim është marrë baza e të dhënave që përmban nivelet e shprehjes së 22,284 gjeneve për sëmundjen e Leucemisë të marra nga 64 pacientë [27]. Burimi i të dhënave është <https://www.kaggle.com/>.

Për të gjetur fillimisht komponentët statistikisht të rëndësishëm (për $k \leq n$), kryhet Testi Scree siç tregohet në Figurën 4.8. Vlerat vetjake mbizotëruese janë llogaritur duke përdorur mbi matricën e qendëruar të shprehjes së gjeneve algoritmin Krylov–Schur në Matlab, i cili është një përgjithësim i procedurës së rifilluar për problemet jo-Hermitiane [33].

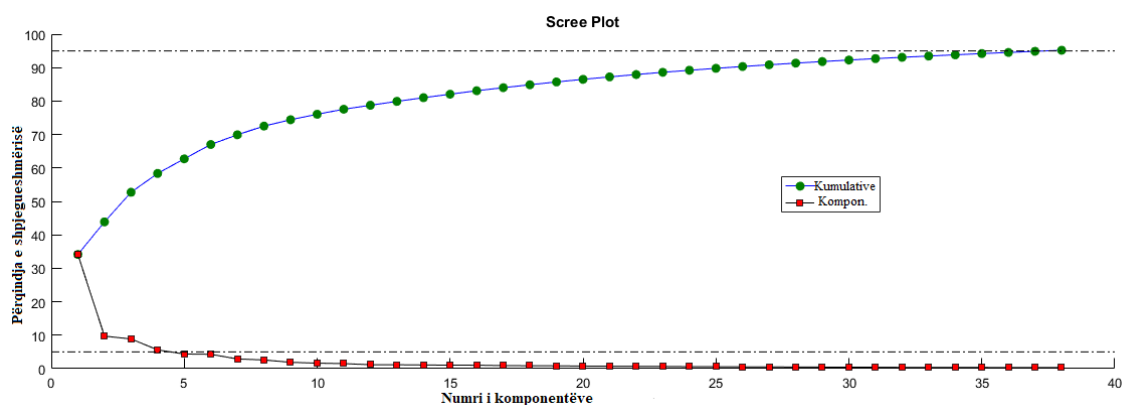


Figura 4.8 Komponentët e rëndësishëm Scree Plot.

Nga Figura 4.8 vërehet se numri i komponentëve kryesorë është i barabartë me katër. Secili nga komponentët kryesorë shpjegon më shumë se 5% të dispersionit dhe të katërta së bashku shpjegojnë 58% të dispersionit total. Në vazhdim zbatojmë algoritmin

LMSVD dhe llogariten pikët (scores) e katër komponentëve kryesorë. Më pas testojmë nivele të ndryshme të pavarësisë midis gjeneve dhe komponentëve kryesorë në mënyrë që të vlerësojmë përqindjen e gjeneve që bëjnë pjesë në komponentin e parë dhe të dytë. Në Figurën 4.9 tregohet se si ndryshon shpërndarja e gjeneve në dy komponentët kryesorë, për nivele besimi nga 50% deri në 95%.

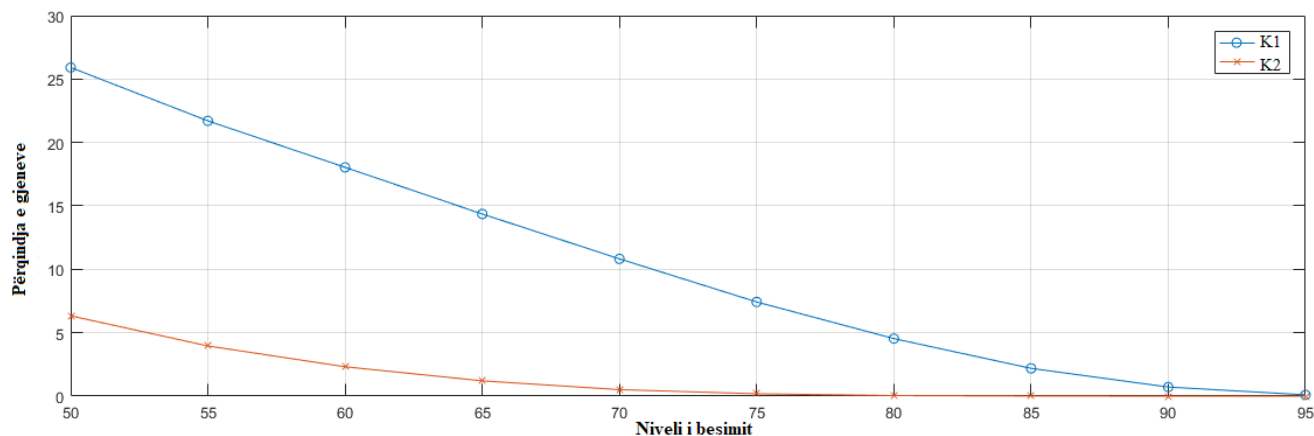


Figura 4.9 Përqindja e gjeneve në dy komponentët kryesorë, për nivele të ndryshme pavarësie.

Për një nivel optimal besimi 80%, në komponentin e parë kryesor bëjnë pjesë 4.5% e gjeneve nga 22,284 lloje, me 34% të shpjgueshmërisë. Ndërsa një përqindje e vogël prej 0.05% e këtyre 22,284 lloje gjenesh bëjnë pjesë në komponentin e dytë kryesor, me thujse 10% të shpjgueshmërisë.

KAPITULLI 5

Puna e mëtejshme

Pamë se si rritja e kompleksitetit të problemeve ka sjellë nevojën e modelimit të tyre në hapësira shumëpërmasore, për të cilat është shumë e vështirë të zgjidhen me anë të metodave direkte. Për këtë arsye, metodat iterative kanë marrë një zhvillim të madh dhe luajnë një rol të rëndësishëm në zgjidhjen e gati të gjitha problemave zbatuese në matematikë. Një ndër avantazhet më të rëndësishme të metodave iterative është edhe aftësia e tyre për t'u paralelizuar në kompjutera paralelë. Puna e mëtejshme konsiston në implementimin paralel të metodave të shpjeguara në Kapitullin 1 dhe në përdorimin e teknikave të suksesshme për paralelizimin e parakushtëzuesve. Kujdesi në paralelizimin e parakushtëzuesve duhet të bëhet në mbajtjen e një ekuilibri midis rritjes së nivelit të paralelizimit dhe rritjes së numrit të përlllogaritjeve. Në këtë kapitull trajtohen disa nga koncepte dhe teknika bazë të paralelizimit, si dhe disa veprime kryesore matematikore lidhur me metodat e nënhapësirës Krylov që mund të favorizojnë paralelizimin e tyre.

Disa nga format më të përdorura në ditët e sotme të paralelizimit janë:

- Njësitë funksionale të shumëfishta
- Pipelining
- Përpunimi vektorial
- Pipeline me vektorë të shumëfishtë
- Përpunimi i shumëfishtë
- Përlllogaritjet e shpërndara

Njësitë funksionale të shumëfishta janë forma më e vjetër e paralelizimit e cila bazohet në shumëfishimin e njësive funksionale si mbledhëse dhe shumëzues. Në këtë mënyrë njësitë e kontrollit dhe regjistrat ndahen nga njësitë funksionale.

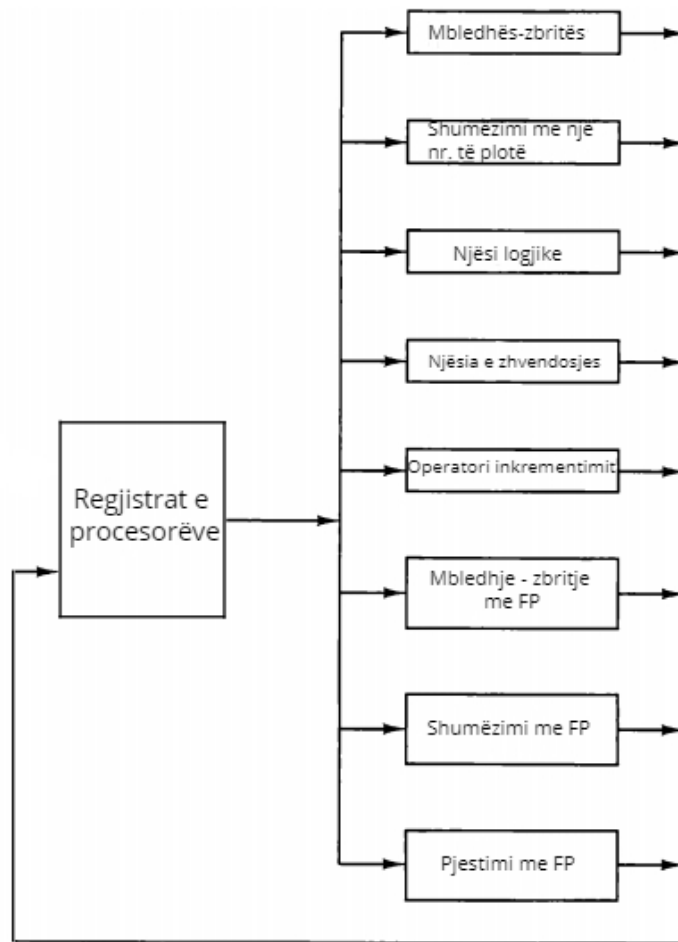


Figura 5.1 Procesor me njësi funksionale të shumëfishta.

Pipelining është një teknikë ku instruksione të shumta mbivendosen në ekzekutim. Pipeline është i ndarë në faza. Secila nga fazat kryen një pjesë të instruksionit në paralel. Faza paraardhëse është e lidhur më atë pasardhëse me anë të një instruksioni që quhet pipe; komandat hyjnë në njërin skaj, përparojnë nëpër faza dhe rezultatet përftohen në skajin tjetër [34].

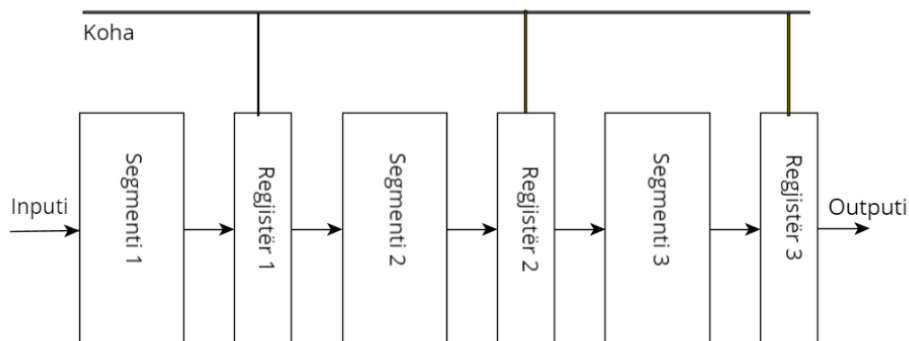


Figura 5.2 Teknika Pipeline.

Përpunim vektorial quhet procesi i përdorimit të vektorëve për ruajtjen e një numri të madh të dhënash dhe përpunimi i tyre me intensitet të lartë, si për shembull: të dhënat dhe veprimet llogaritëse për parashikimin e motit apo të dhënat GIS. Një procesor vektorial është një njësi qendrore (CPU) në një kompjuter me disa procesor paralelë, të cilët mund të kryejnë disa detyra njëkohësisht [35]. Disa nga instruksionet vektoriale më të përdorura janë:

- *VLOAD*- Ngarkon një vektor nga kujtesa në një regjistër vektorial.
- *VADD*- Mbledh përmbajtjen e dy regjistrave vektorial
- *VMUL*- Shumëzon përmbajtjen e dy regjistrave vektorial.

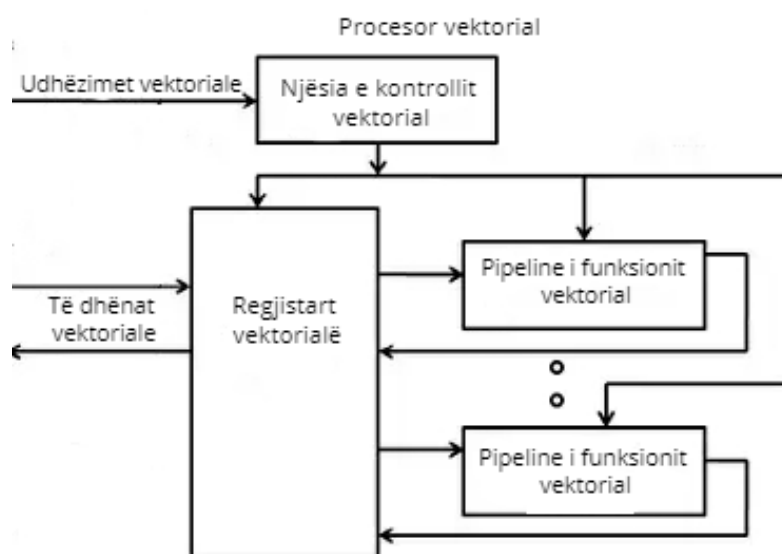


Figura 5.3 Procesori vektorial.

Pipeline vektorial lejon një vektor instruksionit në pipeline të përdorë rezultatin e marrë nga vektori instruksionit i mëparshëm. Ka shumë veprime vektoriale që mund të përfitojnë nga përdorimi i pipeline me vektorë të shumëfishtë. Një sistem me përpunim të shumëfishtë është një kompjuter, ose një bashkësi kompjuterash të cilët përbëhen nga disa elemente përpunues, ku secila prej tyre përmban një CPU, një kujtesë dhe një nënsistem të hyrje/daljes (input/output). Ndërsa përllogaritjet e shpërndara janë një përgjithësim i formës së përpunimit të shumëfishtë, në të cilën procesorët janë kompjutera të lidhura përmes rrjetave LAN.

Në përgjithësi, në varësi të arkitekturës së kujtesës, ekzistojnë dy qasje themelore të shkëmbimit të të dhënave midis proceseve paralele:

- Qasja me kujtesë të përbashkët e të dhënave

Kujtesë e përbashkët do të quhet kujtesa në të cilën mund të shkruajnë ose të lexojnë të dhëna të gjithë procesorët. Adresa e qelizave të kujtesës është e vetme për të gjithë procesorët. Komunikimi i procesorëve bëhet nëpërmjet qelizave të kujtesës duke modifikuar objektet e ruajtura në kujtesë [36].

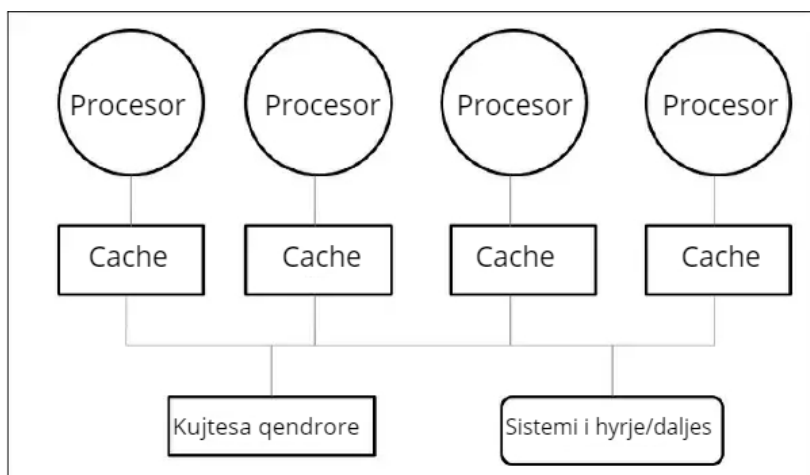


Figura 5.4 Hapësira e përbashkët e të dhënave.

Avantazh i kësaj lloji kujtese është se çdo proces ka akses në të gjitha të dhënat. Për këtë arsye një kod në seri mund të kalojë në kompjutera paralelë me kujtesë të përbashkët dhe zakonisht kjo sjell një rritje të performancës me një numër të vogël procesorësh në përdorim. Nga ana tjetër, me rritjen e numrit të procesorëve rriten edhe problemet me aksesin e kujtesës duke qenë se në një moment të caktuar mund të jenë shumë procesor që kërkojnë të aksesojnë të njëjtën adresë në kujtesë. Një mënyrë për të zvogëluar këto probleme është përdorimi i sistemit bus, por kjo sjell rritje të kostos.

- Qasja e shkëmbimit të mesazheve me anë të kujtesës së shpërndarë

Kjo klasë arkitekturash shmang problemin e konflikteve në aksesimin e kujtesës, duke parashikuar për çdo procesor, kujtesën e vet lokale ku nuk të lexohet nga procesorët e tjerë. Meqenëse, procesorët nuk mund të komunikojnë me anë të kujtesës së përbashkët, ato mund të lidhen ndërmjet tyre me ndonjë rrjet. Çdo procesor lexon drejtpërdrejtë të dhënat në kujtesën e tij lokale [34]. Komunikimi ndërmjet procesorëve bëhet me anë të kalimit të mesazheve ose me mekanizmin e kujtesës së përbashkët virtuale. Komunikimi jo i drejtpërdrejtë i një procesori me të dhënat në kujtesën e një procesori tjetër kërkon që të shtohen komunikime të veçanta dhe kjo sjell që kodet në seri të mos ekzekutohen dhe duhen patjetër algoritme paralelë. Një mangësi e këtij modeli është koha e lartë që shpenzon një procesor për aksesimin e të dhënave nga kujtesa lokale e një procesori tjetër. Gjerësia e kanalit dhe shkalla e transmetimit të të dhënave përbën një tjetër disavantazh të kujtesës së shpërndarë të shkëmbimit të mesazheve.

Ndër veprimet matematikore më të rëndësishme që mund të kryen në paralel lidhur me metodat e nënhapësirës Krylov janë: përditësimi i vektorëve (SAXPY), prodhimi i brendshëm apo skalar, veprime vektor me vektor, prodhimi matricë-vektor, llogaritja e normave matricore dhe llogaritja e vlerave vetjake dhe vektorëve vetjakë përkatës për matrica tridiagonale apo trekëndore. Të trajtojmë shkurtimisht disa prej tyre.

Veprimet e përditësimit të vlerave të vektorit të njohura si veprimet SAXPY quhen veprimet e trajtës $y(i) = y(i) + ax(i)$, për $i = 1, 2, \dots, n$, ku koeficienti a është një skalar dhe x, y janë dy vektorë me përmasë n .

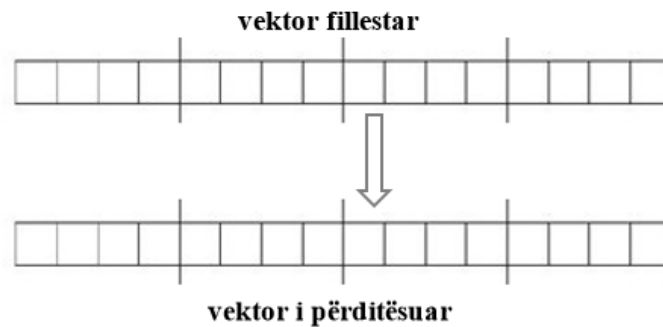


Figura 5.5 Skema paralele e përditësimit të vektorëve.

Prodhimi i brendshëm i vektorëve është një veprim i cili përdoret shpesh në njehsimet matricore, njëllor si mbledhja, zbritja apo shumëzimi i vektorit me një numër në hapësirën Euklidiane. Prodhim i brendshëm i vektorëve x dhe y mund të shkruhet si prodhim i vektorit të transpozuar të x me vektorin y , pra $x^T y$. Nisur nga një vlerë fillestare zero e shumës, përditësimi kryhet sipas $s = s + x(i) * y(i)$.

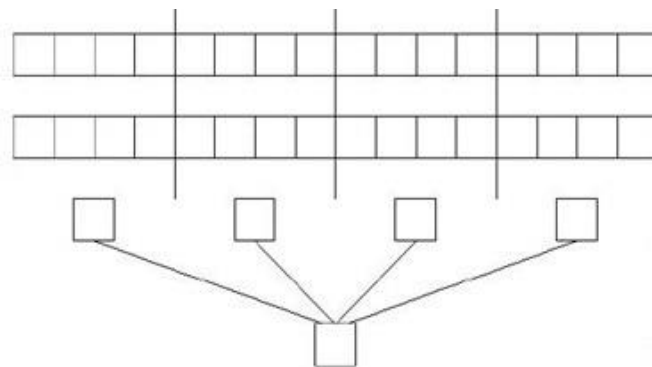


Figura 5.6 Skema paralele e prodhimit skalar dhe normës matricore $\| \cdot \|_2$.

Veprimet vektor me vektor përfshijnë veprime si mbledhja, zbritja, shumëzimi (qoftë dhe me një skalar) i elementeve të një bashkësie vektorësh.

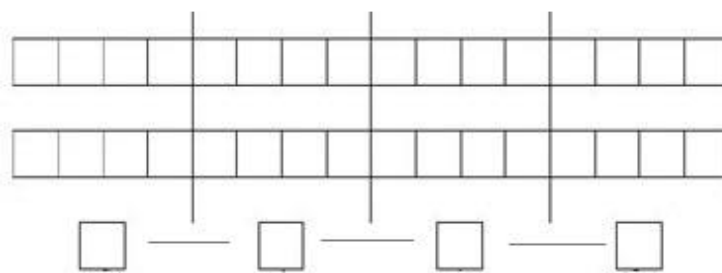


Figura 5.7 Skema paralele e veprimeve vektor me vektor.

Prodhimet matricë-vektor të njohura më termin *Matvec* mund të kryen në paralel në tre mënyra të ndryshme duke ju referuar skemave të prodhimit të brendshëm dhe të përditësimit të vektorëve (SAXPY) si të përmendura më lart por gjithashtu edhe skemës së prodhimit nëpërmjet diagonaleve. Kjo e fundit është një nga formatet më të përshtatshme për rastin e matricave të rralla duke qenë se numri i diagonaleve (kryesore, mbi apo nën diagonalen kryesore) është i vogël.

Duke përdorur skemën e prodhimit të brendshëm për të llogaritur prodhimin Ax , në mënyrë të pavarur elementi i i -të i rezultatit fitohet si një prodhim skalar i rreshtit të i -të i matricës A me vektorin x , kjo për $i = 1, 2, \dots, n$. Zakonisht secilit procesor i caktohen një numër rreshtash në mënyrë të vazhduar dhe po ashtu edhe elementet e secilit vektor. Fillimisht ngarkohet në kujtesë pjesa e matricës që nevojitet. Kur matrica ruhet në kujtesë sipas rreshtave formati njihet si CSR (compressed sparse row). Për të kryer prodhimet matricë-vektorë, është shumë i nevojshëm komunikimi midis procesorëve në mënyrë që të llogariten elementet e vektorit x të cilët nuk ruhen në të njëjtin procesor.

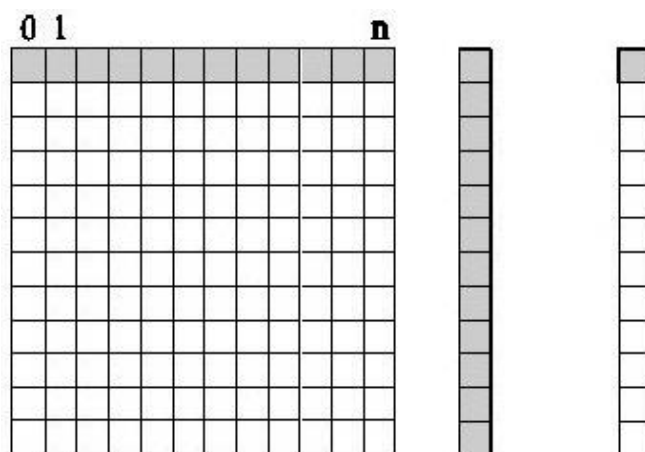


Figura 5.8 Prodhimi matricë-vektor paralel duke përdorur prodhimin e brendshëm.

Skema SAXPY llogarit rezultatit e këtij prodhimi si një kombinim linear të shtyllave të matricës A . Zakonisht llogaritet si shuma e prodhimit të pavarur të element të i -të të vektorit x me shtyllën e i -të të matricën A , kjo për $i = 1, 2, \dots, n$. Kur matrica ruhet në kujtesë sipas shtyllave formati njihet si CSC (compressed sparse column) dhe ndarja e punës midis procesorëve bëhet në mënyrë të ngjashme me rastin e formatit CSR. Formatit CSR mund të përdoret gjithashtu edhe për të llogaritur prodhimin e matricës së transpozuar të A me një vektor x , kur matrica A është ruajtur në kujtesë në formatin CSR.

Prodhimi matricë-vektor mund të kryhet gjithashtu edhe duke ruajtur matricën në formatin diagonal kur matrica A është e rrallë dhe jep një performancë të lartë llogaritëse me këto lloj matricash. Për këtë format të ruajtjes së matricës ekzistojnë algoritme të ndryshëm, ku mund të përmendim formatin DIA bazuar në skemën e prodhimit të brendshëm, formatin Ellpack i cili përdoret në rastet kur numri maksimal i elementeve të ndryshme nga zero në rreshtat e matricës është i vogël apo formatin

diagonal Jagged që është një përgjithësim i format Ellpack që heq supozimin e rreshtave me gjatësi të fiksuar.

Për të ndërtuar këtë të fundit fillohet nga formati SCR duke renditur rreshtat e matricës sipas numrit të elementeve të ndryshëm nga zero në rendin zbritës. Për të ndërtuar diagonalen e parë me indeks j merret elementi i parë nga secili rresht i CSR. Për të ndërtuar diagonalen e dytë merret elementi i dytë i secilit rresht në CSR e kështu me radhë deri në elementin e fundit dhe diagonalet e këtij formati sa vijnë e zvogëlohen.

Në rastin e zgjidhjes së një sistemi ekuacionesh lineare me matricë të rrallë në një mjedis me kujtesë jo të përbashkët, është më se normale që një pjesë e ndryshoreve të sistemit t'i caktohen të njëjtit procesor në një mënyrë të paracaktuar. Ky caktim mund të kryhet automatikisht duke përfaqësuar elementet e matricës nëpërmjet një grafi dhe duke zbatuar më pas ndarjen e tij si në Figurën 5.9 ose kryhet në varësi të informacionit që kemi në dispozicion mbi problemin, ku matrica e marrë në shqyrtim përftohet nga diskretizimi i një EDDP në një fushë përcaktimi.

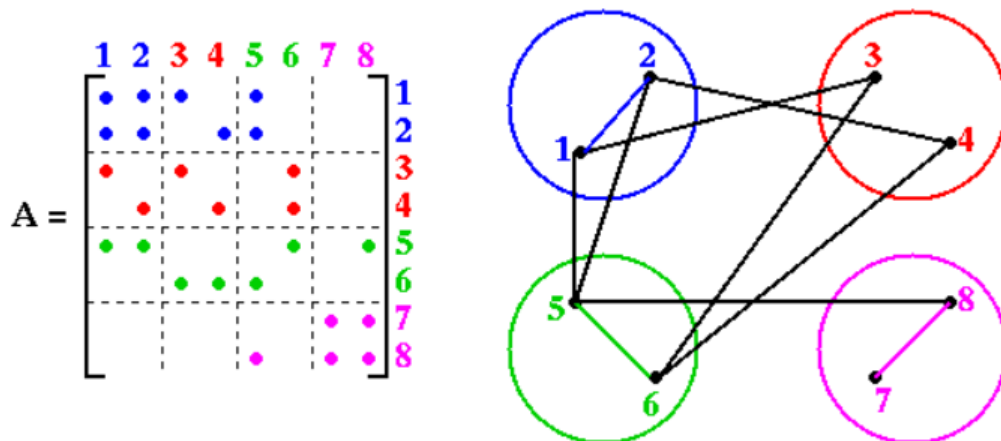


Figura 5.9 Caktimi i ndryshoreve në procesorët përkatës sipas një grafi ndarës.

Nëse rreshti i i matricës i caktohet procesorit p edhe ndryshorja i i caktohet gjithashtu procesorit p pra matrica shpërndahet sipas rreshtave nëpër procesorë në përputhje me shpërndarjen e ndryshoreve. Grafi në këtë rast është i paorientuar pra matrica është simetrike. Në mënyrë që procesi i paralelizimit të kryet me sukses është e nevojshme që paraprakisht të përcaktohet lista e procesorëve me të cilët do të nis komunikimi. Të përcaktohen kulmet lokale të ndarjeve që lidhen me kulmet e jashtme si dhe të bëhet një përfaqësim lokal i matricës së shpërndarë në secilin procesor.

Matrica globale A , rreshtat e të cilës janë të caktuar tek një procesor, duhet të shumëzohet me ndonjë vektor global v . Elementet e këtij vektori caktohen një pjesë në procesorët lokalë dhe një pjesë në procesorët e jashtëm. Ndryshoret e jashtme u korrespondojnë pikave të ndërfaqes që u takojnë nënfushave fqinje. Kur kryhet prodhimi matricë-vektor, procesorët fqinjë duhet të shkëmbejnë vlerat e nyjeve të ndërfaqeve fqinje.

PËRFUNDIME

Analizimi i sjelljes dhe vetive të metodave Krylov është kryer nëpërmjet analizës empirike bazuar në veti të ndryshme të sistemit si denduria e numrit të elementeve të ndryshme nga zero, simetria/jo-simetria e sistemit, mirë/keq kushtëzimi i sistemit apo luhatja e përmasave të tij. Janë shqyrtuar katër kategori të ndryshme problemesh dhe janë aplikuar dy llojet e parakushtëzimit pa plotësim IC(0) dhe ILU(0) duke alternuar disa teknika të ndryshme parakushtëzimi. Parakushtëzimi për këto sisteme aplikohet pasi disa nga metodat e shqyrtuara dështojnë ose janë specifike vetëm për një grup të caktuar problemesh, në variantin e tyre standard apo dhe të rifilluar. Duke parakushtëzuar sistemet, zbatimi i metodave të nënhapësirës Krylov kërkon vetëm disa iteracione për të marrë një përafrim mjaftueshëm të mirë të zgjidhjes. Ky grup metodash ka avantazh pasi ato kanë mjaft liri në përcaktimin dhe ndërtimin e parakushtëzuesve pa qenë nevoja që të njihet secili element i matricës A apo modifikimi i pjesëve të saj. Parakushtëzuesit jo të plotë si ILU dhe IC janë një nga mënyrat më të thjeshta të përcaktimit të një parakushtëzuesi duke kryer një faktorizim jo të plotë të matricës origjinale A .

Në rastin e parë, ku matrica A e rendit 494 është simetrike dhe pozitivisht e përcaktuar, metoda Krylov më kompakte është BiCGStab, performanca e të cilës përmirësohet akoma më shumë mbas parakushtëzimit ILU nga e majta (edhe pse në shumicën e eksperimenteve numerike parakushtëzimi nga e djathta është më efikas). BiCGStab ka konvergencë mirë krahasuar me metodat e tjera të përdorura mbi sistemet simetrike si p.sh CG apo BiCG. Duke shtuar dhe një teknikë parakushtëzimi të mirë BiCGStab vështirë se dështon falë iteracionit të Bi-Lanczos që përdor.

Në rastin e dytë ku matrica e rastit A nuk është simetrike dhe as pozitivisht e përcaktuar, metoda më e dobishme rezultoi GMRES pa rifillim dhe pa parakushtëzim [5]. Teknikat e marra në konsideratë nuk përmirësuan konvergencën e metodave duke na lënë të kuptohet që mund të kërkohen skema të tjera parakushtëzimi sic është ILU(k) me nivel plotësimi k . Në të dy rastet e para rritja e parametrin të rifillimit uli dukshëm shpejtësinë e konvergencës.

Në eksperimentin e tretë, ku matrica A është jo-simetrike me madhësi 4,253, të gjitha metodat përveç MINRES japin gabim përafrimi nën saktësinë (tolerancën) e dhënë. BiCGStab jep gabimin dhe mbetjen relative më të vogël pa parakushtëzim. Në përgjithësi, GMRES është metoda që jep përmirësim të rezultateve nga versioni standard, në versionin me rifillim dhe me parakushtëzim të dyanshëm ILU. BiCGStab ka gabimin mesatar më të vogël dhe mbetjen relative mesatare më të vogël.

Në problemin e fundit, ku diskretizohet ekuacioni eliptik dypërmasor i difuzion-konveksionit me kushte kufitare të tipit Dirichlet nëpërmjet Metodës së Elementeve të Fundme (MEF), për nivele të larta rafinimi të elementeve diskrete, fitohen sisteme të përmasave shumë të mëdha. Konkretisht, për nivel rafinimi 7 fitohet një sistem

ekuacionesh lineare i rrallë me përmasa $33,025 \times 33,025$. Përafrimi i zgjidhjes së EDDP kur zgjidhja analitike është mjaft komplekse apo e pamundur për t'u derivuar, bëhet shumë praktik me anë të metodave të nënhapësirës Krylov. Ashtu siç është tipike e këtyre modeleve të diskretizuara që lindin nga probleme reale të ekuacioneve diferenciale me derivate të pjesshme, matrica e sistemit është e keqkushtëzuar. Zbatimi i teknikave të parakushtëzimit përmirëson ndjeshëm rezultatin. Teknika ILU e dyanshme është parakushtëzimi më i mirë dhe metoda QMR jep gabime më të vogla përafrimi krahasuar me metodat e tjera të parakushtëzuara.

Problemet reale të botës moderne prodhojnë një sasi shumë të madhe të dhënash të marra nga eksperimente të punës kërkimore në fusha të ndryshme si studimi i rrjetave komplekse reale biologjike, sociale, biomjekësore, përpunimi i sinjaleve apo imazheve apo gjenetika. Këto eksperimente prodhojnë të dhëna të mëdha të pa përpunuara të cilat janë shumë të vështira për t'u manipuluar, për të nxjerrë informacion të vlefshëm prej tyre apo për tu ruajtur në kujtesë. Zakonisht për këto lloje të dhënash numri i ndryshoreve p është më i madh se numri i vrojtimeve n . Për këtë arsye lind nevoja e:

- A. marrjes në konsideratë vetëm e disa ndryshoreve më të rëndësishme nga gjithë bashkësia.
- B. transformimit të bashkësisë së të dhënave në një nënbashkësi më të vogël e cila ruan informacionin e rëndësishëm të të dhënave fillestare.

Qasjet e zakonshme, në të dyja rastet, janë shumë të kushtueshme dhe jo praktike. Nënhapësirat Krylov janë një mjet i shkëlqyer për analizimin e rrjetave komplekse dhe vlerësimin e nënhapësirës të rëndësishme në probleme të reduktimit të përmasave duke u bazuar në vetitë e tyre të:

- ortogonalitetit,
- minimizimit të mbetjeve,
- konvergencës së shpejtë
- dhe menaxhimit të kujtesës.

Në këtë disertacion nënhapësirat Krylov janë zbatuar në rrjeta komplekse dhe baza të dhënash reale, që për mbajnë dhjetëra mijëra elemente dhe ndryshore, për të vlerësuar:

- matës të qendërsisë (Qendërsia eksponenciale, Estrada Index, Shpërndarja probabilitare)
- komunikueshmërisë, lidhshmërinë e elementeve
- analizën Laplasiane (Lidhshmëria algjebrike, vektori Feidler)
- k -vlerat vetjake më të mëdha dhe vektorët vetjakë përkatës të matricave përfaqësuese
- sfondit në imazhe/video me nivele të ndryshme zhurme.

Bazuar në variante të iteracionit Arnoldi apo Lanczos, përpunimi i të dhënave të rralla dhe të mëdha, sidomos në rastin kur numri i ndryshoreve tejkalon numrin e vrojtimit ($p > n$), menaxhohet më lehtë se në rastin e metodave klasike si:

- Analiza e komponentëve kryesorë (PCA)
- SPCA
- Përafrimi Padé
- Metoda Scaling & Squaring

Një tjetër avantazh i metodave të nënhapësirës Krylov është se janë teknika lehtësisht të paralelizueshme duke qenë se bazohen kryesisht në veprimet e prodhimit matricë-vektor, përditësimin e vektorëve, prodhimin e brendshëm (apo skalar), veprimet vektor me vektor apo llogaritjen e vlerave vetjake dhe vektorëve vetjakë përkatës për matrica tridiagonale apo trekëndore.

Bibliografia

- [1] G. M. Del Corso, O. Menchi and F. Romani, Krylov subspace methods for solving linear systems, Computer Science, 2015.
- [2] A.Zeqiri and A. Habili, INDUCED DIMENSION REDUCTION (IDR) METHOD”, Proceedings of “Statistics, Probability and Numerical Analysis” International Conference, Botuar në Buletinin e Shkencave të Natyrës, Nr. Special; ISSN 2305-882X, Tiranë, 5 Dhjetor 2015.
- [3] H. A. van der Vorst, Iterative Krylov Methods for Large Linear Systems. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2003.
- [4] Y. Saad, Numerical methods for large eigenvalue problems, Second Edition, Society for Industrial and Applied Mathematics 2011.
- [5] A. Zeqiri, L. Hoda dhe F. Hoxha, ZBATIME NUMERIKE PËR METODËN GMRES TË NËNHAPËSIRËS KRYLOV, Buletini i Shkencave të Natyrës, Nr. 29; ISSN 2305-882X, Shkurt 2021.
- [6] Y. Saad, Iterative Methods for Sparse Linear Systems, Second Edition. Society for Industrial and Applied Mathematics, 2003.
- [7] Zhong-Zhi Bai, Motivations and realizations of Krylov subspace methods for large sparse linear systems. Journal of Computational and Applied Mathematics, 2015.
- [8] P. N. Brown, A theoretical comparison of the Arnoldi and GMRES algorithms. SIAM Journal on Scientific and Statistical Computing, 12, 1991.
- [9] Martin H. Gutknecht, A Brief Introduction to Krylov Space Methods for Solving Linear Systems, Frontiers of Computational Science, 2007.
- [10] D. Chin-Lung Fong and M. Saunders, CG Versus MINRES: An Empirical Comparison. SQU Journal for Science, 2012.
- [11] M. H. Gutknecht, A completed theory of the unsymmetric Lanczos process and related algorithms, Part I. SIAM J. Matrix Anal. Applic, 1992.
- [12] R. W. Freund and N. M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numerische Mathematik, 1991.

- [13] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 1992.
- [14] M. Benzi, Preconditioning Techniques for Large Linear Systems: A Survey. *Journal of Computational Physics*, 182, 2002.
- [15] K. Fan, Note on M -matrices, *Quarterly Journal of Mathematics*, Oxford series (2), 1960.
- [16] C. Lin and R. Saigal, An Incomplete Cholesky Factorization for Dense Symmetric Positive Definite Matrices, *BIT Numerical Mathematics*, 2000.
- [17] M. A. Ajiz and A. Jennings, A robust incomplete Choleski-conjugate gradient algorithm, *Int. J. Numer. Methods Eng.* 20, 1984.
- [18] A. Zeqiri dhe A. Malko, Krylov Methods for Large Sparse Systems: A Comprehensive Overview, *European Scientific Journal (ESJ) NATURAL/LIFE/MEDICAL SCIENCES*, Vol. 17, No. 17; ISSN (Print): 1857-7881 (Print); ISSN (Online): 1857-7431, Maj 2021.
- [19] A. Ghai, C. Lu and X. Jiao, A Comparison of Preconditioned Krylov Subspace Methods for Large-Scale Nonsymmetric Linear Systems. Cornell University, 2018.
- [20] J. Li and Yi-T. Chen, *Computational Partial Differential Equations Using MATLAB*, CRC Press, 2008.
- [21] A. Zeqiri, E. Kalluçi, E. Noka dhe F. Hoxha, KRYLOV METHODS FOR COMPLEX NETWORK ANALYSIS, *Proceedings of GSRD International Conference*, Budapest, Hungary, 11-12 Shkurt 2021.
- [22] Michele Benzi and C. Klymko, “Total communicability as a centrality measure”, *Journal of Complex Networks* 1, pp. 124-149, 2013.
- [23] Hao Wang, *The Krylov Subspace Methods for the Computation of Matrix Exponentials, Theses and Dissertations—Mathematics*, University of Kentucky, 2015.
- [24] M. Ilic, I. W. Turner, D. P. Simpson, A restarted Lanczos approximation to functions of a symmetric matrix, *IMA Journal of Numerical Analysis*, October 2009.

- [25] M. Arioli, B. Codenotti, and C. Fassino, The Padé method for computing the matrix exponential, *Lin. Alg. Applic.* 240, 111–130, 1996.
- [26] A. Zeqiri, M. Muca dhe A. Malko, PCA, SPCA & Krylov-based PCA for Image and Video Processing, *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 19, No. 5, USA, Maj 2021.
- [27] A. Zeqir, E. Kalluçi dhe Xh. Merkaj, Genetic and Genomic Analysis Using Krylov Subspace Methods, *IAI International Virtual Academic Conference Proceedings*, Republic of North Macedonia, 17 Mars 2021.
- [28] I.T. Jolliffe, *Principal Component Analysis*, Second Edition, Springer, USA, 2002.
- [29] M. Johnstone and A.Y. Lu, On Consistency and Sparsity for Principal Components Analysis in High Dimensions, *Journal of the American Statistical Association*, 2009.
- [30] X. Liu, Z. Wen, and Y. Zhang, Limited Memory Block Krylov Subspace Optimization for Computing Dominant Singular Value Decomposition,” *SIAM Journal on Scientific Computing*, 2013.
- [31] Sh. Ubaru, A. Seghouane and Y. Saad, Find the dimension that counts: Fast dimension estimation and Krylov PCA, *arXiv*, Cornell University, 2018.
- [32] H. Zou, T. Hastie and R. Tibshirani, Sparse Principal Component Analysis, *Journal of Computational and Graphical Statistics*, 2006.
- [33] Stewart G. W., A Krylov-Schur Algorithm for Large Eigenproblems, *SIAM Journal on Matrix Analysis and Applications*, Vol. 23, No. 3, 2002.
- [34] B. Wilkinson and M. Allen, *Parallel programming techniques and applications using networked workstations and parallel computers*, 2nd Edition, Pearson Prentice Hall, Pearson Education, Inc. 2005.
- [35] Pacheco P.S., *An Introduction to Parallel Programming*, ELSEVIER 2011.
- [36] F. Hoxha, *Njehsim Paralel*, Shblu, Tiranë, 2007.

Shtojca A

Metodat e nënhapësirës Krylov për sistemet lineare - Kode në Matlab:

BiCGStab

```
function [x,error,fl,iter]=BiCGStab(A,b,tol,MaxIt,K1,K2,x0)
%Biconjugate Gradient Stabilized Method
%Inputet
%   A -- matrica e sistemit
%   b -- vectori i termave te lire
%   tol -- toleranca ne gabim
%   MaxIt -- Numri maksimal i iteracioneve
%   x0 -- perafrimi fillestar
%   K1 -- Matrica 1 e parakushtezimit
%   K2 -- Matrica 2 e parakushtezimit
%Outputet
%   x -- zgjidhja e perafert e sistemit
%   error -- gabimi metodes
%   fl -- 1 metoda konvergjon 0 metoda nuk konvergjon
%   iter -- numri i iteracioneve te konvergjences

error=0; % vleresimi fillestar
fl=false;
x=x0;
r=b-A*x;
r1=r;
sigma=1;
if(isempty(K1))
    K1=eye(length(A)); % kontrolli i parakushtezimit
end
if(isempty(K2))
    K2=eye(length(A));
end
for iter=1:MaxIt
    rho=r1'*r;
    if (rho==0)
        break
    end
    if (iter==1)
        p=((K1*r)'*K2)';
    else
        beta = ( rho/rho1 )*(alpha/sigma);
        p = r + beta*(p-sigma*z);
    end
    z=A*p;
    alpha=rho/(r1'*z);
    k=r-alpha*z;
    if (norm(k)<tol) % kontrolli i pare i konvegjences
        x=x+alpha*p;
        res=norm(k)/norm(b);
        break
    end
end
```

```

s=A*k;
sigma=(s'*k)/(s'*s);
x=x+alpha*p+sigma*k; % perafrimi perfundimtar
r=k-sigma*s;
error = norm(r)/norm(b);
if ( error <= tol ) % kontrolli i fundit i konvegjences
    fl=true;
    break
end
if ( sigma == 0 )
    break
end
rho1=rho;
end
end

```

BiCG

```

function [x,error,fl,iter] = BiCG(x0,MaxIt,tol,A,b,K,restart)
%Biconjugate Gradients Method
%Input
% A -- matrica e sistemit
% b -- vectori i termave te lire
% tol -- toleranca ne gabim
% MaxIt -- Numri maksimal i iteracioneve
% x0 -- perafrimi fillestar
% K -- Matrica e parakushtezimit
% restart -- parametri i ristartimit
%Outputet
% x -- zgjidhja e perafert e sistemit
% error -- gabimi metodes
% fl -- 1 metoda konvergjon 0 metoda nuk konvergjon
% iter -- numri i iteracioneve te konvegjences

for i=1:restart % vleresimi fillestar
iter=0;
fl=false;
b_norm=norm(b);
if (b_norm==0)
    b_norm=1;
end
x=x0;
r=b-A*x;
rt=r;
error=norm(r)/b_norm;
if(error<tol)
    return
end
if isempty(K)
    K=eye(length(A)); % kontrolli i parakushtezimit
end

```



```

for iter=1:MaxIt
    z=K*r;
    zt=K'*rt;
    rho=z'*rt;
    if (rho==0)
        break
    end
    if (iter==1)
        p=z;
        pt=zt;
    else
        beta = ( rho/rho1 );
        p = z + beta*p;
        pt = zt + beta*pt;
    end
    k=A*p;
    kt=A'*pt;
    alpha=rho/(pt'*k);
    x=x+alpha*p;
    r=r-alpha*k;
    rt=rt-alpha*kt;
    error=norm(r)/b_norm;
    if(error<tol)
        fl=true;
    break
    end
    rho1=rho;
end
end
end

```

FOM

```

function [X,R,H,Q]=FOM(A,b,x0)
%Full Orthogonalization Method
%Kryen faktorizimin A=QHQ'
%Inputet
%    A -- matrica e sistemit
%    b -- vectori i termave te lire
%    x0 -- perafrimi fillestar
%Outputet
%    Matrica ortogonale Q
%    X -- zgjidhja e perafert e sistemit
%    R -- matrica e mbetjeve
%    Matrica Hessenberg e siperme H

n=length(A); X=x0;
r=b-A*x0; R=r; r0norm=norm(r);
Q(:,1)=r/r0norm;
for k=1:n
    v =A*Q(:,k);

```

```

for j=1:k
    H(j,k)=Q(:,j)'\*v; v=v-H(j,k)*Q(:,j);
end
e0=zeros(k,1); e0(1)=r0norm;           % zgjidhja e sistemit
y=H\e0; x= x0+Q*y;
X=[X x];
R=[R b-A*x];
if k<n
    H(k+1,k)=norm(v); Q(:,k+1)=v/H(k+1,k);
end
end
end
end

```

QRM

```

function [x,error,fl,iter]=QMR(A,b,tol,MaxIt,K1,K2,x0)
%Quasi Minimal Residual Method
%Inputet
%   A -- matrica e sistemit
%   b -- vectori i termave te lire
%   tol -- toleranca ne gabim
%   MaxIt -- Numri maksimal i iteracioneve
%   x0 -- perafrimi fillestar
%   K1 -- Matrica 1 e parakushtezimit
%   K2 -- Matrica 2 e parakushtezimit
%Outputet
%   x -- zgjidhja e perafert e sistemit
%   error -- gabimi metodes
%   fl -- 1 metoda konvergjon 0 metoda nuk konvergjon
%   iter -- numri i iteracioneve te konvergjences

iter=0;           % vleresimi fillestar
fl = false;
if(isempty(K1))
    K1=eye(length(A)); % kontrolli i parakushtezimit
end
if(isempty(K2))
    K2=eye(length(A));
end
b_norm = norm( b );
if ( b_norm == 0 )
    b_norm = 1;
end
x=x0;
r = b - A*x;
error = norm( r ) / b_norm;
if ( error < tol )
    return;
end
v0 = r;
y = K1 \ v0;

```

```

rho = norm( y );
w0 = r;
z = K2' \ w0;
xi = norm( z );
gamma = 1;
eta = -1;
theta = 0;

for iter = 1:MaxIt

    if ( rho == 0 || xi == 0 )
        break;
    end
    v = v0 / rho;
    y = y / rho;
    w = w0 / xi;
    z = z / xi;
    delta = z'*y;
    if ( delta == 0)
        break;
    end
    y0 = K2 \ y;
    z0 = K1'\ z;
    if ( iter > 1 )
        p = y0 - ( xi*delta / ep )*p;
        q = z0 - ( rho*delta / ep )*q;
    else
        p = y0;
        q = z0;
    end
    end

    p0 = A*p;
    ep = q'*p0;
    if ( ep == 0)
        break;
    end

    beta = ep / delta;
    if ( beta == 0)
        break;
    end
    v0 = p0 - beta*v;
    y = K1 \ v0;
    rho1 = rho;
    rho = norm( y );
    w0 = ( A'*q ) - ( beta*w );
    z = K2' \ w0;
    xi = norm( z );
    gamma1 = gamma;
    theta1 = theta;
    theta = rho / ( gamma1*beta );
    gamma = 1/sqrt( 1+ (theta^2));
    if ( gamma == 0)

```

```

        break;
    end
    eta = -eta*rho1*(gamma^2) / ( beta*(gamma1^2) );

    if ( iter > 1 )
        d = eta*p + (( thetal*gamma )^2)*d;
        s = eta*p0 + (( thetal*gamma )^2)*s;
    else
        d = eta*p;
        s = eta*p0;
    end

    x = x + d;                                % perditesimi perafrimeve

    r = r - s;                                % perditesimi mbetjeve
    error = norm( r ) / b_norm;               % kontrolli i konvergjences
    if ( error <= tol )
        fl=true;
        break;
    end
end
end
end

```

GMRES

```

function
[gmres]=modified_GMRES(x0,MaxIt,tol,A,b,method,K1,K2,restart)
% Generalized Minimal Residual Method
%Inputet
%   A -- matrica e sistemit
%   b -- vectori i termave te lire
%   tol -- toleranca ne gabim
%   MaxIt -- Numri maksimal i iteracioneve
%   method -- method = 1 per GMRES ortogonalizimin
%             e modifikuar te Gram-Schmidt
%             dhe method = 2 GMRES per sistemet komplekse.
%   restart -- parametri i ristartimit
%   x0 -- perafrimi fillestar
%   K1 -- Matrica 1 e parakushtezimit
%   K2 -- Matrica 2 e parakushtezimit
%Outputet
% gmres---grupi outputeve:
%   x -- zgjidhja e perafert e sistemit
%   error -- gabimi metodes
%   fl -- 1 metoda konvergjon 0 metoda nuk konvergjon
%   iter -- numri i iteracioneve te konvergjences

    iter = 0;                                % vleresimi fillestar
    fl = false;
    if(isempty(K1))
        K1=eye(length(A));                   % kontrolli i parakushtezimit
    end
end

```

```

end
if(isempty(K2))
    K2=eye(length(A));
end

b_norm = norm( b );
if ( b_norm == 0)
    b_norm = 1;
end
x=x0;
r =b-A*x;
error = norm( r ) / b_norm ;
if ( error < tol )
    return;
end

[~,n] = size(A);
m = restart;
V(1:n,1:m+1) = zeros(n,m+1);
H(1:m+1,1:m) = zeros(m+1,m);
cs(1:m) = zeros(m,1);
sn(1:m) = zeros(m,1);
e1 = zeros(n,1);
e1(1) = 1.0;

for iter = 1:MaxIt
    r = b-A*x;
    V(:,1) = r / norm( r );
    s = norm( r )*e1;
    for i = 1:m
        M=(K2^(-1))*(K1^(-1));
        z=M*V(:,i);
        w = A*z;           % ort. Gram-Schmidt
        for k = 1:i
            H(k,i)= w'*V(:,k);
            w = w - H(k,i)*V(:,k);
        end
        H(i+1,i) = norm( w );
        V(:,i+1) = w / H(i+1,i);
        for k = 1:i-1
            lambda = cs(k)*H(k,i) + sn(k)*H(k+1,i);
            % zbatimi i rrotullimit Givens
            H(k+1,i) = -sn(k)*H(k,i) + cs(k)*H(k+1,i);
            H(k,i) = lambda;
        end
    end
    if (method==2)
        delta=sqrt(H(i,i)^2+H(i+1,i)^2);
        if(abs(H(i,i))<abs(H(i+1,i)))
            mu=H(i,i)/H(i+1,i);
            tao=mu/abs(mu);
            cs(i)=abs(H(i,i))/delta;
            sn(i)=abs(H(i+1,i))*tao/delta;
        else

```

```

        mu=H(i+1,i)/H(i,i);
        tao=mu/abs(mu);
        cs(i)=abs(H(i,i))/delta;
        sn(i)=abs(H(i+1,i))*tao/delta;
    end
else
    delta=sqrt(H(i,i)^2+H(i+1,i)^2);
    cs(i)=H(i,i)/delta;
    sn(i)=H(i+1,i)/delta;
end
    lambda = cs(i)*s(i); % perafrimi normes se mbetjeve
    s(i+1) = -sn(i)*s(i);
    s(i) = lambda;
    H(i,i) = cs(i)*H(i,i) + sn(i)*H(i+1,i);
    H(i+1,i) = 0;
    error = abs(s(i+1)) / b_norm;
    if ( error <= tol )
        y = H(1:i,1:i) \ s(1:i);
        x = x + V(:,1:i)*y;
        break;
    end
end

    if ( error <= tol )
        break;
    end
    y = H(1:m,1:m) \ s(1:m);
    x = x + V(:,1:m)*y; % perditesimi i perafprimeve
    r = b-A*x; % perditesimi mbetjeve

    s(i+1) = norm(r);
    error = s(i+1) / b_norm; % kontrolli i konvergjences
    if ( error <= tol )
        break;
    end
end

    if ( error > tol )
        fl = true;
    end
    gmres={x,error,fl,iter};
end

```

MINRES

```

function [x,error,fl,iter]=MINRES(A,b,tol,MaxIt,K1,K2,x0)
% Minimum Residual Method
%Inputet
% A -- matrica e sistemit
% b -- vectori i termave te lire
% tol -- toleranca ne gabim
% MaxIt -- Numri maksimal i iteracioneve

```

```

%      x0 -- perafrimi fillestar
%      K1 -- Matrica 1 e parakushtezimit
%      K2 -- Matrica 2 e parakushtezimit
%Outputet
%      x -- zgjidhja e perafert e sistemit
%      error -- gabimi metodes
%      fl -- 1 metoda konvergjon 0 metoda nuk konvergjon
%      iter -- numri i iteracioneve te konvergjences

iter=0;                % vleresimi fillestar
fl = false;
if(isempty(K1))
    K1=eye(length(A)); % kontrolli i parakushtezimit
end
if(isempty(K2))
    K2=eye(length(A));
end
K=K1*K2;
b_norm = norm( b );
if ( b_norm == 0 )
    b_norm = 1;
end
x=x0;
r = b - A*x;
error = norm( r ) / b_norm;
if ( error < tol )
    return;
end
[~,n] = size(A);
b_1=M\b;
l_1=A*b_1;
u_1=sqrt(b_1'*l_1);
l_1=l_1/u_1;
b_1=b_1/u_1;
V = zeros(n,n);
W = zeros(n,n);
beta = zeros(n,1);
lambda = zeros(n,1);
sigma = zeros(n,1);
V(:,1)=b-A*x;
beta(1)=norm(V(:,1));
new=beta(1);
lambda(1)=1;
sigma(1)=0;
sigma0=0;v0=0;w0=0;lambda0=0;w_1=0;
for iter = 1:MaxIt
    r=b-A*x;
    for i=1:n-1
        if (i>1)
            sigma0=sigma(i-1);
            w0=K1/W(i-1);
            v0=K2/V(i-1);
            lambda0=lambda(i-1);

```

```

        end
        if (i>2)
            w_1=W(i-2);
        end
        V(:,i)=(1/beta(i))*V(:,i);
        alpha=V(:,i) '*A*V(:,i);
        V(:,i+1)=A*V(:,i)-alpha*V(:,i)-beta(i)*v0;
        beta(i+1)=norm(V(:,i+1));
        delta=lambda(i)*alpha-lambda0*alpha*beta(i);
        ro1=sqrt(delta^2+beta(i+1)^2);
        ro2=sigma(i)*alpha+lamba0*lambda(i)*beta(i);
        ro3=sigma0*beta(i);
        lambda(i+1)=delta/ro1;
        sigma(i+1)=beta(i+1)/ro1;
        W(:,i)=(V(:,i)-ro3*w_1-ro2*w0)/ro1;
        x=x+lambda(i+1)*new*W(:,i); % perditesimali i perafriemeve
        r1=abs(sigma(i+1))*r;
        error=norm(r1)/b_norm;
        if ( error <= tol ) % kontrolli i konvergences
            break;
        end
        new=-sigma(i+1)*new;
    end
    if ( error <= tol )
        fl = true;
        break;
    end
end
end
end

```


Shtojca B

Algoritmi i vlerësimit të eksponencialit të matricës simetrike së fqinjësisë A bazuar në iteracionin standard Lanczos:

Hapi 1. Zgjidhet vektori fillestar $v_1 = r_0 / \|r_0\|_2$, vendos $\beta_1 = 0, v_0 = 0$

Hapi 2. Për $j = 1, 2, \dots, m$ kryej

Hapi 3. $w_j = Av_j - \beta_j v_{j-1}$

Hapi 4. $\alpha_j = t_{jj} = (w_j, v_j)$

Hapi 5. $w_j = w_j - \alpha_j v_j$

Hapi 6. $\beta_{j+1} = t_{j,j-1} = \|w_j\|_2$

Hapi 7. Në qoftë së $\beta_{j+1} = 0$ atëherë Ndal

Hapi 8. $v_{j+1} = w_j / \beta_{j+1}$

Hapi 9. Fund

Hapi 10. Vlerëso $\exp(A) \approx V_m \exp(T_j) V_m^T$

Algoritmi i rifilluar i Lanczos për përafrimin e $f(A)b$:

Inputet: Funkzioni i matricës f , matrica simetrike A , vektori b , nënhapsira Krylov me përmasë m dhe vlera maksimale e parametrin të rifillimit $Itmax$.

Hapi 1. Përcakto $\beta_{m+1}^{(0)} = \|b\|_2, v_{m+1}^{(0)} = b / \beta_{m+1}^{(0)}, \rho_1 = \beta_{m+1}^{(0)}, f_k = 0$
dhe parametrin e rifillimit $k = 0$.

Hapi 2. Derisa $k < Itmax$ dhe nuk konvergjon

Hapi 3. Vendos $k = k + 1$

Hapi 4. Zgjero zbrëthimin standard të Lanczos: $AV_m^{(k)} = V_m^{(k)} T_m^{(k)} + \beta_m^{(k)} v_{m+1}^{(k)} e_m^T$ për $\mathcal{K}_m(A, v_{m+1}^{(k)})$

Hapi 5. Kryej zbrëthimin e vlerave vetjake: $T_m^{(k)} Y_m^{(k)} = Y_m^{(k)} D_m^{(k)}$
ku $D_m^{(k)} = \text{diag}(\mu_1^{(k)}, \mu_2^{(k)}, \dots, \mu_m^{(k)})$

Hapi 6. Vlerëso funksionin e rifillimit $G_k = g_k(D)$.

Hapi 7. Përditëso funksionin e matricës: $f_k = f_k + \rho_k V_m^{(k)} Y_m^{(k)} G_k [Y_m^{(k)}]^T e_1$

Hapi 8. Vendos $\rho_{k+1} = \rho_k \beta_m$

Hapi 9. Fund

Outputi: Përafrimi i f_k .

Matrica $Y_m^{(k)}$ ka vektorët vetjakë të normalizuar të matricës tridiagonale $T_m^{(k)}$ si shtylla të saj dhe $\mu_i^{(k)}$ për $i = 1, 2, \dots, m$ janë vlerat vetjake të $T_m^{(k)}$.

Funksioni i rifillimit g_k vlerësohet duke përdorur funksionin e matricës f , matricën diagonale D , parametrin $\alpha_i^{(j)} = [e_m^{(j)}]^T Y_m^{(j)} e_i^{(j)}$ dhe $\gamma_i^{(j)} = [e_1^{(j)}]^T Y_m^{(j)} e_i^{(j)}$ për $j < k$ dhe $i = 1, 2, \dots, m$ dhe vlerat e mëparshme $g_j(\mu_i^{(j)})$.

Së pari, funksioni i rifillimit nisët nga $f(D)$ dhe nuk ndryshon në rast se parametri i rifillimit është 1. Më pas nëse $k > 1$ dhe për $j = 1, 2, \dots, k - 1$, gD fillimisht vendoset zero dhe për $i = 1, 2, \dots, m$ vlera e përditësuar është:

$$gD = gD + \alpha_i^{(j)} \gamma_i^{(j)} \left(\text{old_}gD - g_j \left(\mu_i^{(j)} \right) \right) / \left(D - \mu_i^{(j)} \right).$$

Në përfundim të ciklit *Për* në lidhje me ndryshoren e kontrollit i , vendoset $\text{old_}gD = gD$ dhe në përfundim të ciklit *Për* në lidhje me ndryshoren e kontrollit j , jepet vlera $g_k(D) = gD$.

Matës të tjerë të qendërsisë së kulmeve:

- Qendërsia e fuqisë për kulmin i , e dhënë nga barazimi:

$$C_d(i) = d(i) = [A \cdot \mathbf{1}]_i,$$

ku $\mathbf{1}$ është vektor më të gjithë komponentët njësha.

- Qendërsia e afërsisë së kulmit i me pjesën tjetër të rrjetës është:

$$CC(i) = \frac{n-1}{\sum_{j \neq i} d(i,j)},$$

ku $d(i,j)$ është largësia midis kulmeve i dhe j , si për shembull, gjatësia e shtegut më të shkurtër midis kulmeve i dhe j .

- Qendërsia e ndërmjetësisë për kulmin i , jepet nga barazimi:

$$BC(i) = \sum_{j \neq i \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}},$$

ku σ_{jk} është numri total i shtigjeve më të shkurtër midis kulmeve j dhe k dhe $\sigma_{jk}(i)$ është numri i këtyre shtigjeve më të shkurtër që kalojnë nga kulmi i .

- Qendërsia Page Rank e kulmit i jepet në mënyrë rekursive:

$$PR(i) = \sum_{\{j | (j,i) \in E\}} \frac{PR(j)}{d(j)}.$$

Algoritmi LMSVD (Limited Memory Singular Value Decompozition) i disponueshëm si një direktori e Matlab dhe mund të shkarkohet në <https://www.caam.rice.edu>. Ky algoritëm përshpejton procedurën e zbërthimit SVD gjatë analizës së komponentëve kryesorë:

Inputet: Matrica $X_{n \times p}$ dhe k numri fillestar i komponentëve. Jep vlerat fillestare $W = W^{(0)} \in R^{p \times k}$, $Y = Y^{(0)} = XW^{(0)}$ dhe $m = i = 0$;

Hapi 1. Derisa “nuk konvergjon” kryej

/*Optimizimi i Nënhapësirës në Bllok*/

- Hapi 2. Llogarit \mathbf{P}_W dhe kryej stabilizimin e matricës sipas Teknikës 1;
- Hapi 3. Llogarit \mathbf{P}_Y me të njëjtën mënyrë fshirjeje të shtyllave si për \mathbf{P}_W ;
- Hapi 4. Llogarit zbërthimin e vlerave vetjake të $\mathbf{P}_W^T \mathbf{P}_W = U_W \Lambda_W U_W^T$.
- Hapi 5. Kryej Teknikën 2 të stabilizimit për të tkurrur Λ_W dhe U_W ;
- Hapi 6. Llogarit $\mathbf{R} = R_m^{(i)} := [Y^{(i)}, \mathbf{P}_Y U_W \Lambda_W^{-\frac{1}{2}}]$
dhe zbërthimin e vlerave vetjake të $\mathbf{R}^T \mathbf{R}$;
- Hapi 7. Vlerëso \hat{V} e cila përbëhet nga k vektorët vetjakë kryesor të $\mathbf{R}^T \mathbf{R}$;
- Hapi 8. Llogaritni iteracionin e ndërmjetëm $\hat{W}^{(i)} = \mathbf{Q} \hat{V}$
dhe $\hat{Y}^{(i)} = \mathbf{R} \hat{V}$ (e cila është e barabartë me $X \hat{W}^{(i+1)}$)
/*Iteracioni i kryer njëkohësisht*/
- Hapi 9. Llogarit $W^{(i+1)} \in orth(X^T \hat{Y}^{(i)})$ dhe $Y^{(i+1)} = X W^{(i+1)}$;
- Hapi 10. Rrit ndryshoren i , përditëso m , \mathbf{W} dhe \mathbf{Y} , dhe vazhdo.
- Outputi: U , Λ , dhe V .

Theksojmë se me shkronja të mëdha dhe **Bold** janë shënuar matricat bllok.

Në hapin 2 matrica e vlerësuar nga:

$$\mathbf{P}_W = P_W^{(i)} := (I - W^{(i)}(W^{(i)})^T)[W^{(i-1)}, W^{(i-2)}, \dots, W^{(i-m)}]$$

me përmasa $p \times (k \cdot s)$ stabilizohet duke përdorur Teknikën 1 e cila fshin shtyllat e \mathbf{P}_W të cilat kanë normë Euklidiane poshtë një pragu $\epsilon_1 > 0$.

Matrica e vlerësuar si:

$$\mathbf{P}_Y = P_Y^{(i)} := [Y^{(i-1)}, \dots, Y^{(i-m)}] - Y^{(i)}(W^{(i)})^T[W^{(i-1)}, W^{(i-2)}, \dots, W^{(i-m)}]$$

në hapin 3 i nënshtrohet të njëjtit proces fshirje të shtyllave që janë fshirë në \mathbf{P}_W . Mbas zbërthimit në hapin 4, zbatohet një tjetër teknikë stabilizimi (Teknika 2) mbi Λ_W dhe U_W , e cila fshin vlerat vetjake dhe shtyllat përkatëse që janë më pak se $\epsilon_2 > 0$. Në hapin 8 $\mathbf{Q} = Q_m^{(i)} \in orth(W_m^{(i)})$ është një bazë ortonormale e nënhapësirës $\mathcal{S}^{(i)} := span\{W^{(i)}, W^{(i-1)}, \dots, W^{(i-m)}\}$ me një strukturë të nënhapësirës Krylov ne Bllok, e tillë që matrica $W \in \mathcal{S}^{(i)}$ shprehet si $W = \mathbf{Q}V$ dhe ndërtohet si $\mathbf{Q} = Q_s^{(i)} := [W^{(i)}, \mathbf{P}_W U_W \Lambda_W^{-1/2}]$.

Algoritmi Krylov PCA i vlerësimit të përmasës së nënhapësirës kryesore (dominante) dhe komponentëve kryesorë:

Hyrje: Matrica e transpozuar e të dhënave $X_{p \times n}$, zhurma σ , parametri C_n dhe toleranca ϵ

Hapi 1. Vendos $IC = zeros(p, 1)$, $Q = []$, $k = 1$ dhe $m = \log(p) / \sqrt{\epsilon}$

Hapi 2. Llogarit normën $\phi = \frac{1}{n^2} \|X\|_F^4 - \frac{2\sigma}{n} \|X\|_F^2 + p\sigma^2$

Hapi 3. Për $k = 1, 2, \dots, p$ kryej

Hapi 4. Gjenero një vektor të rastit v_k me $\|v_k\|_2 = 1$

Hapi 5. $K = \frac{1}{n} [Xv_k, (XX^T)Xv_k, \dots, (XX^T)^{m-1}Xv_k]$

Hapi 6. $Q = \text{orth}([Q, K]), Q = Q(:, 1:k)$

Hapi 7. $T = \frac{1}{n} Q^T XX^T Q$

Hapi 8. Zbato zbërthimin e vlerave vetjake: $[V, \theta] = \text{eig}(T)$

Hapi 9. $IC(k) = n(\phi - \sum_{i=1}^k (\theta_i - \sigma)^2) - C_n \frac{(p-k)(p-k-1)}{2}$

Hapi 10. Në qoftë se $(k > 1)$ dhe $IC(k) > IC(k - 1)$ atëherë

Hapi 11. Ndalo;

Hapi 12. Fund

Hapi 13. Fund

Outputi: Përmasa e vlerësuar e nënhapësirës kryesore $q = k - 1$ dhe QV , e cila përmban vektorët vetjakë të përfruar si shtylla.