Առաջին էջ

Երկրորդ էջ

**ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ**

էջ

ՆԵՐԱԾՈՒԹՅՈՒՆ ....................................................................................................................

Գլուխ 1. ԳԾԱՅԻՆ ՀԱՆՐԱՀԱՇՎԻ ԹՎԱՅԻՆ ՄԵԹՈԴՆԵՐԸ.

ՏԵՍԱԿԱՆ ՀԻՄՆԱԴՐՈՒՅԹՆԵՐ ................................................................................

* 1. Սեփական արժեքների լրիվ հիմախնդրի լուծման իտերացիոն մեթոդները ...................
  2. QR-ալգորիթմը ........................................................................................................................
  3. LR-ալգորիթմը .......................................................................................................................
  4. Յակոբիի պտտման ալգորիթմը ............................................................................................

Գլուխ 2. ՄԵԹՈԴՆԵՐԻ ՄԵՔԵՆԱՅԱԿԱՆ ԻՐԱԿԱՆԱՑՈՒՄԸ

1. QR – ալգորիթմը և մեքենայական ծրագիրը

2.1.1. Մեքենայական ծրագրի ալգորիթմը........................................................................

2.1.2. Մեքենայական ծրագրի տեքստը ............................................................................

1. LR – ալգորիթմը և մեքենայական ծրագիրը

2.2.1. Մեքենայական ծրագրի ալգորիթմը..........................................................................

2.2.2. Մեքենայական ծրագրի տեքստը................................................................................

1. Յակոբբի եղանակը և մեքենայական ծրագիրը

2.3.1. Մեքենայական ծրագրի ալգորիթմը.......................

2.3.2. Մեքենայական ծրագրի տեքստը.....................................................................

2.4 Մեքենայական իրականացման ծրագրային միջավայրը.....................................................

Գլուխ 3. ՓՈՐՁՆԱԿԱՆ ՀԵՏԱԶՈՏՈՒԹՅՈՒՆՆԵՐ

3.1 QR – ալգորիթմով ստացված արդյունքները .......................................................................

3.2 LR – ալգորիթմով ստացված արդյունքները .........................................................................

3.3 Յակոբիի մեթոդով ստացված արդյունքները ......................................................................

Գլուխ 4. ԱՇԽԱՏԱՆՔԻ ՏԵԽՆԻԿԱ-ՏՆՏԵՍԱԿԱՆ ՀԻՄՆԱՎՈՐՈՒՄ ......................................

4.1 Ծրագրայինփաթեթի մշակման ինքնարժեքի հաշվարկը ...................................

* 1. Գնովի նյութերի և համալրող առարկաների ծախսի հաշվարկը ....................
  2. Էլեկտրաէներգիայի ծախսի հաշվարկը ......................................................................
  3. Հիմնական աշխատավարձի հաշվարկը ...................................................................
  4. Հիմնական միջոցների ամորտիզացիա ......................................................................
  5. Տարածքի վարձակալության ծախսի հաշվարկ .........................................................
  6. Լրիվ ինքնարժքի կալկուլացիան .............................................................................
  7. Շահույթի և միավորի գնի հաշվարկը ....................................................................

Գլուխ 5.ԿԵՆՍԱԳՈՐԾՈՒՆԵՈՒԹՅԱՆ ԱՆՎՏԱՆԳՈՒԹՅՈՒՆ

5.1 Անվտանգության կաննոների պահպանումը .............................................................

Գլուխ 6.ԿԵՆՍԱԳՈՐԾՈՒՆԵՈՒԹՅԱՆ ԱՆՎՏԱՆԳՈՒԹՅՈՒՆ

6.1 Անվտանգության կանոնների պահպանումը ............................................................

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ ........................................................................................................................

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ ................................................................................................................

# **Ներածություն**

Ստացիոնար մատրիցների սեփական արժեքների և նրանց համապատասխանող սեփական վեկտորների որոշման հիմնախնդիրը հաճախ է հանդիպում գիտական և գործնական խնդիրներում, մասնավորապես համակարգերի կայունության հետազոտման, տարբեր համակարգերում անցողիկ պրոցեսների ուսումնասիրման, մեխանիկական, էլեկտրական և էլեկտրոնային համակարգերում սեփական տատանումների հաճախականության որոշման և այլ նպատկներով: Առաջին հայացքից պարզ թվացող այս խնդիրը հանդիսանում է բավականին բարդ հիմախնդիր, քանի որ դրա լուծումը կապված է հաշվողական տիպի մի շարք բարդությունների հետ: Ներկայումս դժվար է պատկերացնել հետազոտությունների որևէ բնագավառ, որտեղ քիչ թե շատ գիտականորեն հիմնավորված լուծումներ ստանալու նպատակով չօգտագործվեն հաշվողական մեթոդներ և հաշվիչ տեխնիկայի միջոցներ: Այս կամ այն խնդրի ճշգրիտ լուծումը գտնել հաճախ չի հաջողվում, քանի որ չի հաջողվում այն ներկայացնել հայտնի տարրական առնչություններով: Այնուամենայնիվ, անհրաժեշտ է ունենալ այս կամ այն դասի խնդիրների թեկուզև մոտավոր լուծման որոշակի հնարավորություններ: Ուստի, առանձնապես ժամանակակից ԷՀՄ-ների առկայության պայմաննեում, հատուկ նշանակություն են ձեռք բերում թվային մեթոդները, որոնք, սովորաբար, հանգեցնում են թվերի որոշակի թվաբանական և տրամաբանական գործողությունների կատարելուն: Թվային մեթոդների ուսոմնասիրման գրավիչ առանձնահատկություններից մեկը այն հարցի պարզաբանումն է, թե տարբեր մեթոդներ որքանով են բավարարում իրար հակասող այնպիսի պահանջների, ինչպիսիք են հուսալիությունը, ճշտությունը, կատարման արագագործությունը, հիշողության ներկայացվող պահանջները, մեքենայական ծրագրի համառոտությունը: Ճշտության տեսանկյունից առավել նպատակահարմար է ընտրել այնպիսի մեթոդներ, որոնք թույլ կտան ստանալ տվյալ խնդրի լուծումը: Հաշվողական մեթոդներին ներկայացվում են մի շարք պահանջներ, որոնց թվում ամենակարևորը հաշվողական գործողությունների կատարման տրված ճշտության ապահովումն է: Սովորաբար նախապատվություն է տրվում այն մեթոդներին, որոնք իրականացվում են ավելի քիչ թվով գործողու­թյուններով, տրամաբանորեն ավելի պարզ են, պահանջում են ԷՀՄ-ի ավելի փոքր հիշողություն [1]:

Սեփական թվերի կամ սեփական վեկտորների հաշվարկման արդյունավետ մեթոդի ընտրությունը տրված ճարտարագիտական խնդրի համար կախված է մի շարք գործոններից, ինչպիսիք են հավասարման տեսակը, փնտրվող սեփական թվերի քանակը և բնույթը: Սեփական թվերի խնդիրների լուծման ալգորիթմները բաժանվում են երկու խմբի: Իտերացիոն մեթոդները շատ լավ հարմարեցված են սեփական թվերի մինիմալ և մաքսիմալ արժեքները որոշելու համար: Ձևափոխությունների վրա հիմնված մեթոդներն ավելի բարդ են, սակայն թույլ են տալիս որոշել մատրիցների բոլոր սեփական թվերը և սեփական վեկտորները:

Ավարտական աշխատանքում ուսումնասիրվել են մատրիցների բոլոր սեփական արժեքների որոշման QR-ալգորիթմը, LR-ալգորիթմը, ինչպես նաև Յակոբիի եղանակը, մշակվել են բոլոր եղանակների մեքենայական ծրագրերը, լուծվել են մի շարք մոդելային օրինակներ, ինչի հիման վրա կատարվել է եղանակների համեմատական վերլուծություն:

**Գլուխ 1. ԳԾԱՅԻՆ ՀԱՆՐԱՀԱՇՎԻ ԹՎԱՅԻՆ ՄԵԹՈԴՆԵՐԸ.**

**ՏԵՍԱԿԱՆ ՀԻՄՆԱԴՐՈՒՅԹՆԵՐ**

Այս մեթոդների թվին են սովորաբար դասվում են գծային հանրահաշվական հավասարումների համակարգերի լուծման, հակադարձ մատրիցների որոշման, մատրիցների որոշիչների հաշվման, մատրիցների սեփական բազմանդամի, սեփական թվերի և սեփական վեկտորների որոշման եղանակները: Նշված խնդիրների լուծման գոյություն ունեցող բազմաթիվ եղանակների մեջ, սակայն առանձնահատուկ տեղ են զբաղեցնում գծային հանրահաշվական հավասարումների համակարգերի լուծման համար նախատեսված եղանակները, որոնք, սովորաբար, բաժանվում երկու խմբի՝

* ճշգրիտ կամ ուղղակի եղանակներ, որոնք թույլ են տալիս խնդիրների ճշգրիտ լուծումներ ստանալ վերջավոր թվով թվաբանական գործողություներ կատարելուց հետո;
* մոտավոր եղանակներ, որոնց կիրառման արդյունքում խնդիրների լուծումներն ստացվում են որպես ինչ-որ անվերջ թվային հաջորդականությունների սահմաններ:

Մինչ դրանց քննարկումը, կանգ առնենք այս խնդիրներին առնչվող մի շարք հարցերի վրա:

Շատ գիտատեխնիկական խնդիրներ հանգում են

(1.1)

-ն ինչ-որ թվային պարամետր է) գծային համասեռ հանրահաշվական հավասարումերի համակարգի ոչ փծուն X լուծման և λ-ի այն արժեքների որոշմանը, որոնց դեպքում այդ լուծումը գոյություն ունի: Այդ դեպքում λ-ն կոչվում է *A* մատրիցի սեփական կամ բնութագրիչ թիվ, իսկ X ոչ զրոյական լուծումը՝ λ սեփական թվի համապատասխան սեփական վեկտոր: Հայտնի է, որ  = 0 հավասարման ոչ փծուն լուծման գոյության համար անհրաժեշտ է և բավարար, որպեսզի տեղի ունենա հետևյալ պայմանը՝

(1.2)

արտահայտությունը կոչվում է *A* մատրիցի բնութագրիչ հավասարում, որի արմատները հենց *A* մատրիցի սեփական թվերն են: Պարզ է, որ դրանց թիվը հավասար է *A* մատրիցի կարգին, այսինքն n հատ է: Ուստի և յուրաքանչյուր , արժեքը դիտարկվող հավասարման մեջ տեղադրելու և այն *X* անհայտ վեկտորի նկատմամբ լուծելու դեպ քում կստացվի համապատասխան սեփական վեկտորը, ընդ որում, եթե իրական թիվ է, ապա իրական է նաև համապատասխան սեփական վեկտորը: Նկատենք, որ սեփական վեկտորը համապատասխան համասեռ հավասարումների համակարգից որոշվում է ինչ-որ հաստատունի ճշտությամբ:

Այսպիսով, *A* մատրիցի սեփական թվերի և սեփական վեկտորների որոշման համար պետք է իրագործել հետևյալ աշխատանքային փուլերը՝

1. կառուցել *A* մատրիցի սեփական բազմանդամը,
2. լուծել ստացված հավասարումը և ստանալ *A* մատրիցի սեփական թվերի արժեքները,
3. որոշել համապատասխան սեփական վեկտորները:

Այս փուլերի յուրաքանչյուրի իրականացում իրականացումը, ինքնին, բավականաչափ բարդ հաշվողական խնդիր է: Այստեղ ևս տարբերում են ճշգրիտ և մոտավոր եղանակներ՝

- «ճշգրիտ եղանակներ» անվանումը կապված է հանգամանքի հետ, որ եթե *A* մատրիցի տարրերը տրվում են ճշգրիտ ձևով, և սովորական կոտորակների հետ հաշվողական գործողությունները կատարվում են նույնպես ճշգրիտ, ապա բնութագրիչ բազմանդամ գործակիցները, ինչպես նաև սեփական թվերն ու սեփական վեկտորները ևս հաշվարկվում են ճշգրիտ ձևով,

- մոտավոր եղանակների օգտագործման դեպքում *A* մատրիցի սեփական թվերն ստացվում են անմիջականորեն, առանց բնութագրիչ բազմանդամին ուղղակիորեն դիմելու, որի դեպքում միժամանակ հաշվարկվում են նաև համապատասխան սեփական վեկտորները: Մոտավոր եղանակները կրում են իտերացիոն բնույթ և, սովորաբար, օգտագործում են բազմաթիվ անգամ մատրիցը վեկտորով բազմապատկելու գործուղությունը: Այս ընթացակարգերը բերում են որոշակի վեկտորների հաջորդականությունների, որոնց սահմանային արժեքները հենց սեփական վեկտորներն են, և որոշակի թվային հաջորդականությունների, որոնց սահմանային արժեքներն էլ սեփական թվերն են: Մոտավոր եղանակները սովորաբար թույլ են տալիս որոշելու մատրիցի որոշ սեփական թվերը և նրանց համապատասխան սեփական վեկտորները: Ուստի այդ դեպքում ասում են, որ գործ ունեն, այսպես կոչված «սեփական արժեքների մասնավոր հիմնախնդրի» հետ: Ճշգրիտ մեթոդները հնարավորություն են տալիս լուծելու նաև «սեփական արժեքների լրիվ հիմնախնդիրը», այսինքն որոշել ինչպես բոլոր սեփական թվերը, այնպես էլ սեփական վեկտորները: Այս խնդիրը երբեմն հնարավորություն է լինում լուծել նաև հատուկ իտերացիոն եղանակներով, որոնք առավել աշխատատար են, սակայն, ի տարբերություն ճշգրիտ մեթոդների, չեն օգտագործվում մատրիցի բնութագրիչ հավասարման պարտադիր կառուցման ընթացակարգերը: Այս հանգամանքը հատկապես կարևոր է այն տեսակետից, որ բնութագրիչ հավասարման գործակիցների որոշման ընթացքում թույլ տրված չնչին սխալները երբեմն կարող են զգալի ազդեցություն ունենալ սեփական թվերի, ինչպես նաև համապատասխան սեփական վեկտորների որոշման ճշտության վրա: Բացի այդ, մոտավոր եղանակները ճշգրիտ եղանակների նկատմամբ անհամեմատ պարզ են, ընդգրկում են միանման գործողություններ, ինչն առանձնապես կարևոր է ԷՀՄ-ների օգտագործման տեսանկյունից:

* 1. **Սեփական արժեքների լրիվ հիմնախնդրի լուծման**

**իտերացիոն մեթոդները**

Որպես կանոն*, n > 10* կարգերի դեպքում ԷՀՄ-ի օգտագործումը սովորաբար չի հանգեցնում բավարար արդյունքների, ուստի անհրաժեշտություն է առաջանում կիրառել իտերացիոն մեթոդներ:

Սիմետրիկ (կամ էրմիտային) մատրիցների սեփական արժեքների լրիվ հիմնախնդրի լուծման առաջին իտերացիոն մեթոդն առաջարկվել է Կ. Յակոբիի կողմից 1846 թ., ավելի շուտ, քան առաջ եկան ճշգրիտ մեթոդները: Սակայն մեծ աշխատատարության պատճառով Յակոբիի մեթոդը սովորաբար չէր կիրառվում, մինչև որ չստեղծվեցին ԷՀՄ-ները: ԷՀՄ-ների օգտագործումը բերեց աշխատատարության հասկացության վերանայման՝ առաջին պլան մղելով կայունության և հաշվարկների հուսալիության պահանջները: Յակոբիի մեթոդի վերածնունդը և լայն կիրառությունը սկսվեց 1960 թվականից:

Յակոբիի մեթոդի էությունը կայանում է հետևյալում. *A* սկզբնական մատրիցը օրթոգոնալ ձևափոխությունների միջոցով բերում է համարյա եռանկյունաձև տեսքի, որտեղ ոչ անկյունագծային տարրերը, ինչ-որ ճշտությամբ, հավասարվում են զրոյի: Այդ դեպքում սկզբմական մատրիցի սեփական թվերը նույն ճշտությամբ համընկնում են ստացված մատրիցի անկյունագծային տարրերի հետ, այսինքն՝ :

Սակայն Յակոբիի մեթոդը չի կարող համարվել սեփական թվերի հիմնախնդրի լուծման հանրանշանակ մեթոդը, քանի որ այստեղ դիտարկվում են միայն սիմետրիկ մատրիցները:

1960 թ.-ին Վ. Ն. Կուբլանովսկայայի և, նրանցից անկախ, 1961 թ.-ին Դժ. Ֆրենսիսի կողմից առաջարկվեց ոչ սիմետրկ մատրիցների սեփական արժեքների լրիվ հիմնախնդրի լուծման թվային կայուն մեթոդ, որը կոչվում է *QR*-ալգորիթմ: *QR*-ալգորիթմը, ինչպես Յակոբիի մեթոդը, հիմնված է օրթոգոնալ ձևափոխություների կիրառման վրա: Օրթոգոնալ ձևափոխություննեիր առավելությունն այն է, որ դրանք չեն մեծացնում հաշվարկների սխալների չափը:

**1.2 QR-ալգորիթմը**

*QR*-ալգորիթմը նախատեսված է *A* ավտոնոմ մատրիցը *Q* և *R* մատրիցների արտադրյալի տեսքով ներկայացման համար, այսինքն *A = QR* համարժեք տեսքով ներկայացնելու համար, որտեղ *Q-*ն n չափանի օրթոգոնալ մատրից է, իսկ *R*-ը n չափանի վերին եռանկյունաձև մատրից է: Այս ալգորիթմը սովորաբար իրականացվում է Հաուսհոլդերի կամ Գիվենսի ձևափոխման հիման վրա: Հաուսհոլդերին ձևափոխման սկզբունքը կայանում է հետևյալում.

որտեղ *W*-ն իրական տարրեր պարունակող չափանի վեկտոր-սյուն է, որի դեպքում տեղի ունի = 2 պայմանը: *I*-ն չափանի միավոր մատրից է, *P*-ն ևս n չափանի մատրից է, որն անվանում են Հաուսհոլդերի ձևափոխության *n* չափանի սիմետրիկ օրթոգոնալ մատրից:

*P* մատրիցի կառուցման համար իրականացվում է գործողությունների հետևյալ հաջորդականությունը.

**Քայլ 1:**

Որոշվում են հետևյալ տարրերը.

(1.3)

**Քայլ 2:**

Հաշվվում են տարրերը.

(1.4)

**Քայլ 3:**

Առանձնացվում են հետևյալ տարրերը.

(1.5)

**Քայլ 4:**

Որոշվում են հետևյալ վեկտոր-սյունները.

(1.6)

**Քայլ 5:**

Առանձնացվում են վեկտոր-սյուները.

(1.7)

**Քայլ 6:**

Կազմվում են մատրիցները ըստ (1.8) առնչության.

(1.8)

Հաուսհոլդերի մատրիցի կառուցումից հետո որոշվում են Q և R մատրիցները ըստ համապատասխան արտահայտությունների.

**Քայլ 7:**

(1.9)

**Քայլ 8:**

(1.10)

**Քայլ 9:**

(1.11)

A մատրիցի սեփական արժեքները ստանալու համար որոշվում է մատրցների հետևյալ հաջորդականությունը.

**Քայլ 10:**

(1.12)

ընդ որում, եթե մատրիցի բոլոր անկյունագծային մինորները չվերացված են, ապա այդ հաջորդականությունը (m ) սահմաններում հանգում է վերին եռանկյունաձև մատրիցի, որի անկյունագծային էլեմենտները հանդիսանում են A մատրիցի սեփական թվերը՝ : Եթե, բացի սեփական թվերից անհրաժեշտ է ունենալ նաև համապատասխան սեփական վեկտորները՝ ապա հաջորդականության կառուցման համար անհրաժեշտ է հիշել նաև հետևյալ օրթոգոնալ մատրիցները՝

**Քայլ 11:**

(1.13)

որոնք կարելի է որոշել հետևյալ ռեկուրենտ սխեմայի օգնությամբ.

**Քայլ 12:**

(1.14)

**1.3 LR-ալգորիթմը**

LR-ալգորիթմը թույլ է տալիս գտնել A մատրիցի բոլոր սեփական թվերը:

**ԹԵՈՐԵՄ:** Եթե մատրիցի բոլոր գլխավոր մինորները 0-ից տարբեր են, ապա A մատրիցը հնարավոր է ներկայացնել A = LR տեսքով, որտեղ , որի գլխավոր անկյունագծում մեկերն են, իսկ :

**ԱՊԱՑՈՒՅՑ:**  մատրիցի համար բոլոր գլխավոր անկյունագծային մինորները տարբերվում են զրոյից: մատրցի համար իրագործելի է LU-ձևափոխությունը՝

, որտեղ իսկ , որի գլխավոր անկյունագծում մեկերն են: Հետևաբար որտեղ , որի գլխավոր անկյունագծում մեկերն են, իսկ R:

Թեորեմն ապացուցված է:

**LR-ձևափոխության կառուցման ալգորիթմը կամայական մատրիցի համար:**

Անհրաժեշտ է գտնել L = () ներքին եռանկյունաձև մատրիցը, որի գլխավոր անկյունագծում մեկեր են և R = () վերին եռանկյունաձև մատրիցն այնպես, որ A = LR, այսինքն՝

(1.15)

Քանի որ = 0, երբ երբ , ապա (1,15)-ը -ի n(n+1)/2 անհայտների նկատմամբ, երբ և ի (n+1)/2 անհայտների նկատմամբ հավասարումներով համակարգ է, երբ և -ի անհայտների նկատմամբ,

երբ , ընդամենը անհայտ: Կստանանք (1,15) համակարգի լուծման բանաձևերը, որոնք կազմում են LR-ձևափոխության հաշվման ալգորիթմը:

Քանի որ , երբ = 0, երբ (1.15) հավասարման գումարը կլինի՝

կամ

Առանձնացնենք առաջին գումարից , դեպքը, իսկ երկրորդից՝ դեպքը և հաշվի առնենք, որ բոլոր համար

Վերախմբավորենք այս հավասարումները՝

Այս բանաձևերով հաշվարկը կատարվում է հետևյալ կերպ.

Սկզբից (1.16)-ի առաջին բանաձևից հաշվարկվում են R մատրիցի առաջին տողի անհայտ տարրերը՝ Ապա (1.16)-ի երկրերդ բանաձևից հաշվարկվում են L մատրիցի առաջին սյան տարրերը՝ (հիշեցնենք տարրը հայտնի է, այն հավասար է 1-ի): Հաջորդ հաշվարկներում օգտագործվում են միայն երրորդ և չորրորդ բանաձևերը: (1.16)-ի երրորդ բանաձևից հաշվարկվում են R մատրիցի երկրորդ տողի տարրերը՝ (հիշեցնենք քանի որ R–ը վերին եռանկյունաձև մատրից է):

(1.16)-ի չորրորդ բանաձևից հաշվարկվում են L մատրիցի երկրորդ սյան անհայտ տարրերը՝ (հիշեցնենք = 0 , քանի որ L-ը ներքին եռանկյունաձև մատրից է, = 1, քանի որ L-ի գլխավոր անկյունագծի տարրերը մեկեր են):

Ապա (1.16)-ի երրորդ բանաձևից պարզվում են R մատրիցի երրորդ տողի անհայտ տարրերը՝ և այսպես շարունակ, իսկ (1.16)-ի չորրորդ բանաձևից պարզվում են L մատրիցի երրորդ սյան անհայտ տարրերը՝ և այսպես շարունակ:

**1.4 Յակոբի (պտտման) ալգորիթմը**

Շարադրենք մի իտերացիոն մեթոդ ևս, որը հնարավորություն է տալիս սիմետրիկ, ատրիցների համար լուծել սեփական արժեքների լրիվ հիմնախնդիրը: Այս մեթոդը կոչվում է պտտման կամ Յակոբիի մեթոդը:

Պտտման մեթոդը հիմնված է հետևյալ թեորեմի վրա:

**ԹԵՈՐԵՄ:** Կամայական *A* սիմետրիկ մատրիցի համար գոյություն ունի այնպիսի C օրթոգոնալ մատրից, որ

D = (1.17)

մատրիցն անկյունագծային է:

Քանի որ , ապա (1.17)-ից բխում է, որ *A* և *D*-ն նման մատրիցներ են, ուստի նրանց սեփական արժեքները համընկնում են: D մատրիցի սեփական արժեքներն անկյունագծային տարրերն են (հիշենք, որ սիմետրիկ մատրիցի սեփական արժեքներն իրական թվեր են):

Ենթադրենք, որ սեփական արժեքը հայտնի է: Այդ դեպքում կարելի է

(1.18)

հավասարությունից գտնել մատրիցի համապատասխան սեփական վեկտորը:

Այժմ -ով նշանակենք A մատրիցի սեփական արժեքին համապատասխանող վեկտորը: Այդ դեպքում՝

(1.19)

(1.17) հավասարությունը աջից բազմապատկելով -ով և հաշվի առնելով (1.18) պայմանը կստանանք՝

կամ էլ

(1.20)

Վերջին հավասարությունը ձախից բազմապատկելով C մատրիցով, կստանանք՝

կամ էլ

(1.21)

(1.19) և (1.21) հավասարություններից բխում է, որ՝

քանի որ վեկտորը C մատրիցի i-րդ սյուն է, ապա A մատրիցի սեփական վեկտորները C մատրիցի սյուներն են:

Այսպիսով, A սիմետրիկ մատրիցի սեփական արժեքների լրիվ խնդրի լուծումը համարժեք է C օրթոգոնալ մատրիցի գտնելուն:

Պտտման իտերացիոն մեթոդը հնարավորություն է տալիս գտնելու C մատրիցի տարրերի մոտավոր արժեքները՝ ցանկացած ճշտությամբ:

Այժմ նկարագրենք պտտման մեթոդի ալգորիթմը:

Քայլ 1: Ընտրում ենք A մատրիցի բացարձակ արժեքով ամենամեծ ոչ անկյունագծային տարրերը: A մատրիցի սիմետրիկության շնորհիվ կարող ենք ասել, որ գոյություն ունի ամենաքիչը երկու հատ այդպիսի տարր: Ենթադրենք այն -ն է: A մատրիցի սիմետրիկության շնորհիվ կարելի է ենթադրել, որ : Եթե այդպիսի տարրեր շատ կան, ապա կարելի է վերցնել նրանցից ցանկացածը:

Կառուցենք՝

 (1.22)

i0 j0

պտտման մատրիցը: Ընտրենք անկյունն այնպես, որ

մատրիցի տարրը հավասար լինի զրոյի, այսինքն՝

(1.23)

Դիցուք, Այդ դեպքում A և B մատրիցները կտարբերվեն միայն և -րդ սյուներով, որոնց տարրերը կորոշվեն հետևյալ առընչություններով՝

; (1.24)

Այնուհետև, մատրիցը կտարբերվի B մատրիցից միայն և -րդ տողերով, որոնց տարրերը կորոշվեն հետևյալ առընչություններով՝

*,*

(1.25)

Այսպիսով, և մատրիցները կտարբերվեն միայն և -րդ տողերով և սյուներով, հետևաբար՝

(1.26)

(1.24) և (1.25) հավասարություններից՝ , , տարրերի համար կստանանք հետևյալ առնչությունները՝

ա) (1.25) բանաձևի առաջին հավասարումից հաշվենք -րդ տողի տարրը՝

Վերջին հավասարության մեջ տեղադրելով -ի և -ի արժեքները, (1.24) հավասարությունից կստանանք՝

,

կամ, հաշվի առնելով, որ (հիշենք, որ մատրիցը սիմետրիկ է), կունենանք՝

; (1.27)

բ) նույն ձևով՝

կամ էլ՝

(1.28)

գ) այնուհետև՝

կամ էլ՝

(1.29)

մատրիցի տարրերը կարելի է հաշվել՝ ելնելով (1.24), (1.25), (1.26), (1.27), (1.28), (1.29) առնչություններից:

Կստանանք՝

(1.30)

որտեղ

, :

անկյունն ընտրելու համար (1.27) հավասարության նկատմամբ կիրառենք (1.23) պայմանը, այսինքն ընդունենք, որ

, (1.31)

որտեղից, ենթադրելով, որ , կստանանք՝

, ,

(1.32)

(1.31) հավասարումից բխում է, որ պայմանի դեպքում առկա է հետևյալ առնչությունից՝

,

որտեղից, հաշվի առնելով, որ , կստանանք՝

կամ էլ

(1.33)

Նշնեք, որ պտտման մատրիցը կառուցելիս կարիք չկա գտնել անկյունը: Ունենալով-ի արժեքը կարելի է հաշվել -ի և -ի արժեքները, իսկ այնուհետև 1.30 բանաձևով՝ մատրիցի տարրերը: Հաշվի առնելով 1.32 և 1.33 առընչութունները, -ի և -ի արժեքները կարելի է հաշվել հետևյալ բանաձևերով՝

եթե (1.34)

,

եթե

որտեղ՝

, :

Սրանով ալգորիթմի առաջին քայլն ավարտվում է: Այնուհետև գործընթացը կրկնվում է:

Ենթադրենք, որ մատրիցն արդեն կառուցված է: Այդ դեպքում ալգորթմի -րդ քայլը հետևյալն է. մատրիցում, ըստ վերը շարադրված սկզբունքի, ընտրվում է () տարրը, փոխարինելով ն համապատասխանաբար ով, կառուցվում է (1.22) տեսքի

պտտման մատրիցը, այնուհետև անկյունն ընտրվում է

մատրիցի տարրի զրո լինելու պայմանից, այսինքն՝

(1.35)

(1.34) առնչությունից բխում է, որ իտերացիաների քայլում և -ի արժեքները կորոշվեն հետևյալ բանաձևերով՝

եթե (1.36)

,

եթե (1.37)

որտեղ՝

, :

Այդ դեպքում մատրիցի տարրերը կհաշվվեն հետևյալ կերպ.

(1.38)

(1.39)

(1.40)

բանաձևերով:

Այսպիսով կառուցեցինք մատրիցների հաջորդականությունը, որտեղ՝

Ընդ որում՝

մեծությունները և մատրիցի սյուները համապատասխամաբար մատրիցի սեփական արժեքների և սեփական վեկտորների մոտավոր արժեքներն են:

Իտերացիոն գործընթացի ավարտի պայմանը ունի հետևյալ տեսքը՝

*:* (1.41)

**Գլուխ 2. ՄԵԹՈԴՆԵՐԻ ՄԵՔԵՆԱՅԱԿԱՆ ԻՐԱԿԱՆԱՑՈՒՄԸ**

**2.1. QR-ալգորիթմը և մեքենայական ծրագիրը**

**2.1.1. Ծրագրի ալգորիթմը**

Սկիզբ

A մատրիցի չափայնության, տարրերի, ճշտության ներմուծում

1

2

տարրերի որոշում

մատրիցների տարրերի

հաշվարկ

3

մատրիցների

ստացում

մատրիցների

հաջորդականության կազմում

մատրիցի հակադարձի

հաշվարկ

վերջ

մատրիցի ստացում

Ծրագրի բլոկ-սխեմայի բացատրությունը

Նկ.2.1

մատրիցի անկյունագծային տարրերի արտածում

5

9

8

7

6

4

մատրիցի գլխավոր

անկյունագծից ներքև

գտնվող տարրերը

Ծրագրի բլոկ-սխեման ներկայացված է նկ. 2.1-ում, որտեղ`

Բլոկ 1 – ներմուծվում են A մատրիցի տարրերը, չափայնությունը և խնդրի լուծման ճշտությունը:

Բլոկ 2 – կատարվում են ալգորիթմով նախատեսված քայլերը և հաշվվում են տարրերը:

Բլոկ 3 - և մատրիցների հաշվարկ:

Բլոկ 4 - մատրիցների հաշվարկ:

Բլոկ 5 - և մատրիցների ստացում:

Բլոկ 6 - հակադարձ մատրիցների հաշվարկ:

Բլոկ 7 - մատրիցների հաջորդականությունների հաշվարկ:

Բլոկ 8 – պայմանի բլոկ է, որտեղ ստուգվում է հերթական մատրիցի գլխավոր

անկյունագծից ներքև տարրերի 0 լինելու պայմանը որոշակի ներմուծված

ճշտությամբ: Եթե այն ճշմարիտ է, անցում է կատարվում բլոկ 9-ին, հակառակ

դեպքում՝ բլոկ 2-ին:

Բլոկ 9 – արտածվում են վերջին իտերացիայում ստացվող մատրիցի անկյունագծային

տարրերը, որոնք հենց մատրիցի սեփական թվերն են:

**2.1.2 Մեքենայական ծրագրի տեքստը**

let linebreak ="<br />";

//հավասարաչափ մատրցի հայտարարում

function newMatrix(matrixSize){

let k, matrixIs = new Array(matrixSize);

for( k = 0; k < matrixSize; k++){

matrixIs[k] = new Array(matrixSize);

}

return matrixIs;

};

//անհավասարաչափ մատրցի հայտարարում

function newMatrixUnequal(matrixSize1,matrixSize2){

let k, matrixIs = new Array(matrixSize1);

for( k = 0; k < matrixSize1; k++){

matrixIs[k] = new Array(matrixSize2);

}

return matrixIs;

};

//Մատրիցի տարերերի ներմուծում

function promptMatrix(matrixSize){

let matrix = newMatrix(matrixSize), matrix2 = newMatrix(matrixSize);

for(let i =0; i < matrixSize; i++){

for(let j = 0; j < matrixSize; j++){

matrix[i][j] = prompt('Matrix['+(i+1)+']['+(j+1)+'] =');

matrix2[i][j] = parseFloat(matrix[i][j]);

document.write('Matrix2['+(i+1)+']['+(j+1)+'] = '+matrix2[i][j] + ' ');

}

document.write(linebreak);

}

document.write(linebreak);

return matrix2;

};

//մատրիցի հետք

function hetq(matrix){

let hetq = 0;

for(let i = 0; i < matrix.length; i++){

for(let j = 0; j < matrix.length; j++){

if(i === j){

hetq += matrix[i][j];

}

}

}

return hetq;

};

//Մատրիցների բազմապատկում

function mul(matrix,matrix2){

let len = matrix.length, mat = newMatrix(len);

for(let i = 0; i < len; i++){

for(let j=0; j < len; j++){

mat[i][j] = 0;

for(let k=0; k < len; k++){

mat[i][j] = mat[i][j]+ matrix[i][k]\*matrix2[k][j];

}

}

}

return mat;

};

//մատրիցների հանում

function sub(matrix,matrix2){

let len = matrix.length;

let mat = newMatrix(len);

for(let i = 0; i<len; i++){

for(let j=0; j<len; j++){

mat[i][j] = matrix[i][j]-matrix2[i][j];

}

}

return mat;

}

//հակադարձ մատրիցի հաշվարկը Ֆադեևի եղանակով

function fadeevInverse() {

let matrixSize = prompt("matrix size",);

matrixSize = parseInt(matrixSize);

document.write("A"+ linebreak);

let A = promptMatrix(matrixSize), p = [], p1 = hetq(A);

p.push(p1);

let A1 = A, B1 = newMatrix(matrixSize);

for(let k = 1; k < matrixSize; k++ ) {

let E = newMatrix(matrixSize);

for(let i =0; i < matrixSize; i++) {

for(let j = 0; j < matrixSize; j++) {

if (i ===j ) {

E[i][j] = p[k-1];

} else {

E[i][j] = 0;

}

}

}

let B = newMatrix(matrixSize);

B = sub(A1,E);

let C = mul(A,B);

let p2 = hetq(C)/(k+1);

p.push(p2);

A1 = C;

if(k === (matrixSize-1)) {

B1 = B;

}

}

let reverseA = newMatrix(matrixSize);

for(let i=0; i < matrixSize; i++) {

for(let j=0; j < matrixSize; j++) {

reverseA[i][j] = B1[i][j]/p[p.length-1];

}

}

return reverseA;

}

//AM տարրի որոշում

function AM(a, matrixSize){

let s = [], v = [] , myu = [];

for(let i = 0; i < matrixSize-1; i++) {

let sum = 0;

for(let j = i; j < matrixSize; j++) {

sum += Math.pow(a[j][i], 2);

}

sum = Math.pow(sum,1/2);

if ( a[i][i] > 0 ){

sum \*= (-1);

} else if ( a[i][i] < 0 ) {

sum \*= 1;

} else {

sum = 0;

}

s.push(sum);

let vSum = 0;

vSum = Math.pow(s[i], 2) - s[i]\*a[i][i];

v.push(1/vSum);

myu.push(Math.sqrt(v[i]));

}

let ua = [];

for(let i = 0; i < matrixSize-1; i++) {

var ui = [];

for(let j = 0; j < matrixSize; j++) {

if( i === j ) {

ui[i] = a[i][i] - s[i];

ua.push(ui[i]);

} else if ( i < j ) {

ui[i] = a[j][i];

ua.push(ui[i]);

} else {

ui[i] = 0;

ua.push(ui[i]);

}

}

}

let u = [];

while(ua.length) {

u.push(ua.splice(0,matrixSize));

}

let W = newMatrixUnequal(matrixSize-1,matrixSize);

for(let i = 0; i < matrixSize-1; i++) {

for(let j = 0; j < matrixSize; j++) {

W[i][j] = myu[i]\*u[i][j];

}

}

let wTrans = newMatrixUnequal(matrixSize,matrixSize-1);

for(let i = 0; i < matrixSize; i++) {

for(let j = 0; j < matrixSize-1; j++) {

wTrans[i][j] = W[j][i];

}

}

var pn = [];

for(let i = 0; i < matrixSize-1; i++) {

var pi = newMatrix(matrixSize, matrixSize);

for(var j = 0; j < matrixSize; j++) {

for(var k = 0; k < matrixSize; k++) {

pi[j][k] = W[i][j] \* wTrans[k][i];

}

}

pn.push(pi);

}

var p = pn;

for(let i = 0; i < matrixSize-1; i++) {

for(let j = 0; j < matrixSize; j++) {

for(let k = 0; k < matrixSize; k++) {

if( j === k ) {

p[i][j][k] = 1 - pn[i][j][k];

} else {

p[i][j][k] = 0 - pn[i][j][k];

}

}

}

}

let p0 = p[matrixSize-2];

let R = newMatrix(matrixSize,matrixSize);

for(let i = matrixSize-3; i >= 0; i--) {

p0 = mul(p0,p[i]);

}

R = mul(p0,a);

let Q =fadeev(p0,matrixSize);

let Am = mul(R,Q);

return Am;

}

// կանգռի պայման

function condition(a,matrixSize,tau){

let condition = [] , t = true;

for(let i = 1; i < matrixSize; i++) {

for(let j = 0; j < i; j++) {

if( Math.abs(a[i][j]) < tau) {

condition.push(true);

} else {

condition.push(false);

}

}

}

return condition;

}

//QR ի հաշվարկ

function QR(){

let matrixSize = prompt("matrix size",);

matrixSize = parseInt(matrixSize);

document.write("A"+ linebreak);

let A = promptMatrix(matrixSize);

let Am1 = AM(A,matrixSize);

let bool = condition(Am1,matrixSize);

let iteraci = 1;

let t = true;

while(t){

for(let i = 0; i < bool.length; i++){

if(bool[i] == false){

iteraci++;

Am1 = AM(Am1,matrixSize);

bool = condition(Am1,matrixSize,tau);

t = true;

break;

} else {

t=false;

}

}

}

print(Am1,'Am['+iteraci+"]");

let l = lambda(Am1,matrixSize);

document.write("Մատրիցի սեփական թվերը"+linebreak);

arrayPrint(l, 'lambda');

document.write("Իտերացիաների քանակը՝ K = " +iteraci);

}

let qrCount = true;

function qr() {

displayAndBtn("qr", "sub", "add", "mul", "inverse", "leverey", "fadev", "lr", "yakob","qrClick", "subClick", "addClick", "mulClick", "inverseClick", "levereyClick", "fadevClick", "lrClick", "yakobClick");

location.href = "#qr";

if(qrCount) {

let div1 = document.createElement('div');

div1.classList = "input-fluid col s2";

let select = document.createElement('select');

select.id ="select\_id";

let option = [];

for(let i = 2; i <= 100; i++){

option[i] = document.createElement('option');

option[i].setAttribute("value", i);

option[i].innerHTML = i;

}

div1.classList = "input-fluid col s3 offset-s4";

let input11 = document.createElement('input');

input11.setAttribute('type','number');

input11.setAttribute('min','0');

let div12 = document.createElement('div');

div12.appendChild(input11);

div12.classList = "input-fluid col s3";

let creatMatrix = document.createElement('input');

creatMatrix.setAttribute("value","Set Matrix");

creatMatrix.setAttribute("type", "submit");

creatMatrix.classList = "waves-effect waves-light btn";

creatMatrix.style.marginTop = "20px";

creatMatrix.addEventListener('click',getValue);

function getValue(){

let tr1 = [];

let size = select.value;

let tauNumber = parseFloat(input11.value);

let table1 = document.createElement('table');

let td1 = newMatrix(size);

let input1 = newMatrix(size);

let div2 = newMatrix(size);

let label1 = newMatrix(size);

let A = newMatrix(size);

let A1 = newMatrix(size);

let done = document.createElement('input');

let div3 = document.createElement('div');

let lammbda = [];

let text = [];

let header2 = document.createElement("h6");

let header1 = document.createElement("h6");

let table2 = document.createElement('table');

let tr2 = [];

done.setAttribute('value', 'calculate');

done.setAttribute('type','button');

done.classList = 'btn';

for(let i = 0; i < size; i++){

tr1[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td1[i][j] = document.createElement('td');

div2[i][j] = document.createElement('div');

if(size < 5 ){

div2[i][j].classList = "input-field col s5";

} else {

div2[i][j].classList = "input-field";

}

var inputValue = "A["+(i+1)+"]["+(j+1)+"]";

input1[i][j] = document.createElement('input');

label1[i][j] = document.createElement('label');

label1[i][j].id = inputValue;

label1[i][j].innerHTML = inputValue;

input1[i][j].setAttribute('type', 'number');

input1[i][j].setAttribute("id", inputValue);

input1[i][j].setAttribute("name", inputValue);

}

}

done.addEventListener('click', done1);

function done1() {

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

A[i][j] = parseFloat(input1[i][j].value);

}

}

A1 = AM(A,size);

let bool = condition(A1,size,tauNumber);

let iteraci = 0;

let t = true;

while(t){

for(let i = 0; i < bool.length; i++){

if(bool[i] == false){

iteraci++;

A1 = AM(A1,size);

bool = condition(A1,size,tauNumber);

t = true;

break;

} else {

t=false;

}

}

}

table2.classList = "bordered";

let td2 = newMatrix(size);

for(let i = 0; i < size; i++){

tr2[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td2[i][j] = document.createElement('td');

let a = "A["+(i+1)+"]["+(j+1)+"]";

td2[i][j].classList = "text-deep-purple";

td2[i][j].innerHTML = a + " = " + A1[i][j].toFixed(3);

tr2[i].appendChild(td2[i][j]);

}

table2.appendChild(tr2[i]);

}

document.getElementById('qr').appendChild(table2);

let l = lambda(A1,size);

header1.innerHTML = "The matrix's own numbers are :" + linebreak;

for(let i = 0; i < size; i++){

lammbda[i] = document.createElement('img');

lammbda[i].setAttribute("src", "https://cdn.pixabay.com/photo/2013/03/30/00/09/lambda-97846\_960\_720.png");

lammbda[i].classList = "letters";

lammbda[i].style.marginTop = '0px';

text[i] = document.createElement('p');

text[i].innerHTML = "["+(i+1)+"] = " + l[i]+linebreak;

div3.appendChild(lammbda[i]);

div3.appendChild(text[i]);

}

header2.innerHTML = "The number of Ithers: K = " + iteraci + linebreak;

document.getElementById('qrpart1').appendChild(header2);

document.getElementById('qrpart1').appendChild(header1);

document.getElementById('qrpart1').appendChild(div3);

}

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

div2[i][j].appendChild(input1[i][j]);

div2[i][j].appendChild(label1[i][j]);

td1[i][j].appendChild(div2[i][j]);

tr1[i].appendChild(td1[i][j]);

}

table1.appendChild(tr1[i]);

}

document.getElementById("qrpart1").appendChild(table1);

let clearElement = document.createElement('input');

clearElement.setAttribute("value","Clear matrix");

clearElement.setAttribute("type", "submit");

clearElement.classList = "waves-effect waves-light btn";

clearElement.addEventListener('click', clearValue);

document.getElementById("qrpart1").appendChild(done);

function clearValue() {

for(let i = 0; i < size; i++){

table1.removeChild(tr1[i]);

div3.removeChild(lammbda[i]);

div3.removeChild(text[i]);

table2.removeChild(tr2[i]);

}

document.getElementById('qrpart1').removeChild(header2);

document.getElementById('qrpart1').removeChild(header1);

document.getElementById('qrpart1').removeChild(div3);

document.getElementById("qrpart1").removeChild(done);

document.getElementById("qrpart1").removeChild(table1);

document.getElementById("qrpart1").removeChild(clearElement);

}

document.getElementById("qrpart1").appendChild(clearElement);

}

for(let i = 2; i <= 100; i++){

select.appendChild(option[i]);

}

div1.appendChild(select);

document.getElementById("qrdimension2").appendChild(div12);

document.getElementById("qrdimension1").appendChild(div1);

document.getElementById("qrdimension3").appendChild(creatMatrix);

qrCount = false;

}

}

**2.2. LR-ալգորիթմը և մեքենայական ծրագիրը**

**2.2.1. Ծրագրի ալգորիթմը**

Ծրագրի բլոկ-սխեմայի բացատրությունը

5

6

4

մատրիցի անկյունագծային

տարրերը վերագրում ենք λ

սեփական թվերին

Նկ.2.2

վերջ

λ սեփական թվերի արտածում

մատրիցի գլխավոր

անկյունագծից ներքև

գտնվող տարրերը

3

2

1

մատրիցների հաշվարկ

իտերացիոն մատրիցի հաշվարկ

A մատրիցի չափայնության և տարրերի, ճշտության ներմուծում

Սկիզբ

Ծրագրի բլոկ-սխեման ներկայացված է նկ. 2.2-ում, որտեղ`

Բլոկ 1 – ներմուծվում են A մատրիցի n չափայնությունը և տարրերը:

Բլոկ 2 – կատարվում է ալգորիթմով նախատեսված հաշվարկը և որոշվում են են R և L

մատրիցները:

Բլոկ 3 – կատարվում է իտերացիոն մատրիցի հաշվարկ:

Բլոկ 4 – պայման բլոկ է, որով ստուգվում է կանգառի պայմանը. եթե այն կատարվում

է, ապա անցնում ենք 5-րդ բլոկին, եթե ոչ՝ 2-րդ բլոկին:

Բլոկ 5 – հաշվվում են սեփական թվերը:

Բլոկ 6 – սեփական թվերի արտածում:

**2.2.2. Մեքենայական ծրագրի տեքստը**

//Մատրիցների բազմապատկում

function mul(matrix,matrix2){

let len = matrix.length, mat = newMatrix(len);

for(let i = 0; i < len; i++){

for(let j=0; j < len; j++){

mat[i][j] = 0;

for(let k=0; k < len; k++){

mat[i][j] = mat[i][j]+ matrix[i][k]\*matrix2[k][j];

}

}

}

return mat;

};

// կանգռի պայման

function condition(a,matrixSize,tau){

let condition = [] , t = true;

for(let i = 1; i < matrixSize; i++) {

for(let j = 0; j < i; j++) {

if( Math.abs(a[i][j]) < tau) {

condition.push(true);

} else {

condition.push(false);

}

}

}

return condition;

}

//AM տարրի որոշում

function am(A,matrixSize){

let L = newMatrix(matrixSize);

let R = newMatrix(matrixSize);

for(let i = 0; i < matrixSize; i++){

for(let j = 0; j < matrixSize; j++){

R[i][j] = 0;

L[i][j] = 0;

}

}

for(let i = 0; i< matrixSize; i++) {

R[0][i] = A[0][i];

L[i][0] = A[i][0]/R[0][0];

}

for(let i = 1; i <matrixSize; i++) {

if (i > 1){

for(let k = i-1; k < matrixSize; k++){

let sum2 = 0;

for(let j = 0; j < k-1; j++){

sum2 += L[i][j]\*R[j][k];

}

L[i][k] = (A[i][k]-sum2)/R[k][k];

}

}

for(let k = i; k < matrixSize; k++ ){

let sum1 = 0;

for(let j = 0; j < i; j++){

sum1 += L[i][j]\*R[j][k];

}

R[i][k] = A[i][k] - sum1;

}

}

for(let i = 0; i < matrixSize; i++){

for(let j = 0; j < matrixSize; j++){

if (i === j) {

L[i][j] = 1

}

}

}

let Aml = mul(R,L);

return Aml;

}

function LR(){

let matrixSize = prompt("matrix size",);

matrixSize = parseInt(matrixSize);

document.write("A"+ linebreak);

let A = promptMatrix(matrixSize);

let Am1 = am(A,matrixSize);

let bool = condition(Am1,matrixSize);

let iteraci = 1;

let t = true;

while(t){

for(let i = 0; i < bool.length; i++){

if(bool[i] == false){

iteraci++;

Am1 = AM(Am1,matrixSize);

bool = condition(Am1,matrixSize,tau);

t = true;

break;

} else {

t=false;

}

}

}

document.write("Մատրիցի սեփական թվերը"+linebreak);

let l = lambda(Am1,matrixSize);

arrayPrint(l, 'lambda');

document.write("Իտերացիաների քանակը՝ K = " +iteraci);

}

let lrCount = true;

function lr() {

location.href = "#lr";

displayAndBtn("lr", "sub", "add", "mul", "inverse", "leverey", "fadev", "qr", "yakob","lrClick", "subClick", "addClick", "mulClick", "inverseClick", "levereyClick", "fadevClick", "qrClick", "yakobClick");

if(lrCount) {

let div1 = document.createElement('div');

div1.classList = "input-fluid col s2";

let select = document.createElement('select');

select.id ="select\_id";

let option = [];

for(let i = 2; i <= 100; i++){

option[i] = document.createElement('option');

option[i].setAttribute("value", i);

option[i].innerHTML = i;

}

div1.classList = "input-fluid col s3 offset-s4";

let input11 = document.createElement('input');

input11.setAttribute('type','number');

input11.setAttribute('min','0');

let div12 = document.createElement('div');

div12.appendChild(input11);

div12.classList = "input-fluid col s3";

let creatMatrix = document.createElement('input');

creatMatrix.setAttribute("value","Set Matrix");

creatMatrix.setAttribute("type", "submit");

creatMatrix.classList = "waves-effect waves-light btn";

creatMatrix.style.marginTop = "20px";

creatMatrix.addEventListener('click',getValue);

function getValue(){

let tr1 = [];

let size = select.value;

let tauNumber = parseFloat(input11.value);

let table1 = document.createElement('table');

let td1 = newMatrix(size);

let input1 = newMatrix(size);

let div2 = newMatrix(size);

let label1 = newMatrix(size);

let A = newMatrix(size);

let A1 = newMatrix(size);

let done = document.createElement('input');

let div3 = document.createElement('div');

let lammbda = [];

let text = [];

let header2 = document.createElement("h6");

let header1 = document.createElement("h6");

let table2 = document.createElement('table');

let tr2 = [];

done.setAttribute('value', 'calculate');

done.setAttribute('type','button');

done.classList = 'btn';

for(let i = 0; i < size; i++){

tr1[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td1[i][j] = document.createElement('td');

div2[i][j] = document.createElement('div');

if(size < 5 ){

div2[i][j].classList = "input-field col s5";

} else {

div2[i][j].classList = "input-field";

}

var inputValue = "A["+(i+1)+"]["+(j+1)+"]";

input1[i][j] = document.createElement('input');

label1[i][j] = document.createElement('label');

label1[i][j].id = inputValue;

label1[i][j].innerHTML = inputValue;

input1[i][j].setAttribute('type', 'number');

input1[i][j].setAttribute("id", inputValue);

input1[i][j].setAttribute("name", inputValue);

}

}

done.addEventListener('click', done1);

function done1() {

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

A[i][j] = parseFloat(input1[i][j].value);

}

}

A1 = am(A,size);

let bool = condition(A1,size,tauNumber);

let iteraci = 0;

let t = true;

while(t){

for(let i = 0; i < bool.length; i++){

if(bool[i] == false){

iteraci++;

A1 = am(A1,size);

bool = condition(A1,size,tauNumber);

t = true;

break;

} else {

t=false;

}

}

}

table2.classList = "bordered";

let td2 = newMatrix(size);

for(let i = 0; i < size; i++){

tr2[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td2[i][j] = document.createElement('td');

let a = "A["+(i+1)+"]["+(j+1)+"]";

td2[i][j].classList = "text-deep-purple";

td2[i][j].innerHTML = a + " = " + A1[i][j].toFixed(3);

tr2[i].appendChild(td2[i][j]);

}

table2.appendChild(tr2[i]);

}

document.getElementById('lr').appendChild(table2);

let l = lambda(A1,size);

header1.innerHTML = "The matrix's own numbers are :" + linebreak;

for(let i = 0; i < size; i++){

lammbda[i] = document.createElement('img');

lammbda[i].setAttribute("src", "https://cdn.pixabay.com/photo/2013/03/30/00/09/lambda-97846\_960\_720.png");

lammbda[i].classList = "letters";

lammbda[i].style.marginTop = '0px';

text[i] = document.createElement('p');

text[i].innerHTML = "["+(i+1)+"] = " + l[i]+linebreak;

div3.appendChild(lammbda[i]);

div3.appendChild(text[i]);

}

header2.innerHTML = "The number of Ithers: K = " + iteraci + linebreak;

document.getElementById('lrpart1').appendChild(header2);

document.getElementById('lrpart1').appendChild(header1);

document.getElementById('lrpart1').appendChild(div3);

}

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

div2[i][j].appendChild(input1[i][j]);

div2[i][j].appendChild(label1[i][j]);

td1[i][j].appendChild(div2[i][j]);

tr1[i].appendChild(td1[i][j]);

}

table1.appendChild(tr1[i]);

}

document.getElementById("lrpart1").appendChild(table1);

let clearElement = document.createElement('input');

clearElement.setAttribute("value","Clear matrix");

clearElement.setAttribute("type", "submit");

clearElement.classList = "waves-effect waves-light btn";

clearElement.addEventListener('click', clearValue);

document.getElementById("lrpart1").appendChild(done);

function clearValue() {

for(let i = 0; i < size; i++){

table1.removeChild(tr1[i]);

div3.removeChild(lammbda[i]);

div3.removeChild(text[i]);

table2.removeChild(tr2[i]);

}

document.getElementById('lrpart1').removeChild(header2);

document.getElementById('lrpart1').removeChild(header1);

document.getElementById('lrpart1').removeChild(div3);

document.getElementById("lrpart1").removeChild(done);

document.getElementById("lrpart1").removeChild(table1);

document.getElementById("lrpart1").removeChild(clearElement);

}

document.getElementById("lrpart1").appendChild(clearElement);

}

for(let i = 2; i <= 100; i++){

select.appendChild(option[i]);

}

div1.appendChild(select);

document.getElementById("lrdimension2").appendChild(div12);

document.getElementById("lrdimension1").appendChild(div1);

document.getElementById("lrdimension3").appendChild(creatMatrix);

lrCount = false;

}

}

**2.3. Յակոբիի եղանակը և մեքենայական ծրագիրը**

**2.3.1. Ծրագրի ալգորիթմը**

1

Ծրագրի բլոկ-սխեմայի բացատրությունը

Նկ.2.3

5

4

3

2

վերջ

մատրիցի անկյունագծային տարրերի արտածում

մատրիցի գլխավոր

անկյունագծից ներքև

գտնվող տարրերը

տարրերի որոշում

մատրիցի տարրերի

որոշում

A մատրիցի չափայնության, տարրերի, ճշտության ներմուծում

Սկիզբ

Ծրագրի բլոկ-սխեման ներկայացված է նկ. 2.3-ում, որտեղ`

Բլոկ 1 – ներմուծվում են A մատրիցի n չափայնությունը և տարրերը:

Բլոկ 2 – կատարվում են ալգորիթմով նախատեսված քայլերը և հաշվվում են

տարրերը:

Բլոկ 3 – կատարվում է իտերացիոն մատրիցի հաշվարկ:

Բլոկ 4 – պայման բլոկ է, որով ստուգվում է հերթական մատրիցի գլխավոր

անկյունագծից ներքև և վերև տարրերի 0 լինելու պայմանը որոշակի ներմուծված ճշտությամբ: Եթե այն կատարվում է, ապա անցնում ենք 5-րդ բլոկից, եթե ոչ՝ 2-րդ բլոկին:

Բլոկ 5 – արտածվում են վերջին իտերացիային տարրերը, որոնք հենց մատրիցի

սեփական թվերն են:

**2.3.2. Մեքենայական ծրագրի տեքստը**

// Յակոբի տարրի որոշում

function yakobAm(A,matrixSize) {

let max = Math.abs(A[1][0]);

let i0 = 1, j0 = 0;

for(let i = 2 ; i < matrixSize; i++){

for(let j = 0 ; j < i; j++){

if(Math.abs(A[i][j])>max){

max = Math.abs(A[i][j]);

i0 = i;

j0 = j;

}

}

}

let fi = 0.5\*Math.atan((2\*max)/(A[i0][i0]-A[j0][j0]));

let c = Math.cos(0.5\*Math.atan((2\*A[i0][j0])/(A[i0][i0]-A[j0][j0])));

let s = Math.sin(0.5\*Math.atan((2\*A[i0][j0])/(A[i0][i0]-A[j0][j0])));

let Am = newMatrix(matrixSize);

for(let i = 0; i < matrixSize; i++){

Am[i0][i] = A[i][i0]\*c + A[i][j0]\*s;

Am[j0][i] = -A[i][i0]\*s + A[i][j0]\*c;

Am[i][i0] = Am[i0][i];

Am[i][j0] = Am[j0][i];

}

for(let i = 0; i < matrixSize; i++){

for(let j = 0; j < matrixSize; j++){

if(i === j){

Am[i][j] = A[i][j];

}

}

}

Am[i0][i0]=A[i0][i0]\*Math.pow(c,2)+2\*A[i0][j0]\*s\*c+A[j0][j0]\*Math.pow(s,2);

Am[j0][j0] = A[i0][i0]\*Math.pow(s,2)-2\*A[i0][j0]\*s\*c+A[j0][j0]\*Math.pow(c,2);

Am[i0][j0] = 0;

Am[j0][i0] = 0;

return Am;

}

function yakob(){

let matrixSize = prompt("matrix size",);

matrixSize = parseInt(matrixSize);

document.write("A"+ linebreak);

let A = promptMatrix(matrixSize);

let t = true;

while(t) {

let AM = yakobAm(A, matrixSize);

for(let i = 0 ; i < matrixSize; i++) {

for(let j = 0 ; j < matrixSize; j++){

if(i < j || i > j) {

AM[i][j] < 0;

t = false;

} else {

t = true;

}

}

}

}

}

let yakobCount = true;

function yakob() {

displayAndBtn("yakob", "sub", "add", "mul", "inverse", "leverey", "fadev", "lr", "qr","yakobClick", "subClick", "addClick", "mulClick", "inverseClick", "levereyClick", "fadevClick", "lrClick", "qrClick");

location.href = '#jakop';

if(yakobCount) {

let div1 = document.createElement('div');

div1.classList = "input-fluid col s2";

let select = document.createElement('select');

select.id ="select\_id";

let option = [];

for(let i = 2; i <= 100; i++){

option[i] = document.createElement('option');

option[i].setAttribute("value", i);

option[i].innerHTML = i;

}

div1.classList = "input-fluid col s4 offset-s4";

let input11 = document.createElement('input');

input11.setAttribute('type','number');

input11.style.backgroundColor = 'white';

let div12 = document.createElement('div');

div12.appendChild(input11);

div12.classList = "input-fluid col s4";

let creatMatrix = document.createElement('input');

creatMatrix.setAttribute("value","Set Matrix");

creatMatrix.setAttribute("type", "submit");

creatMatrix.classList = "waves-effect waves-light btn";

creatMatrix.addEventListener('click',getValue);

function getValue(){

let tr1 = [];

let size = select.value;

let tauNumber = parseFloat(input11.value);

let table1 = document.createElement('table');

let td1 = newMatrix(size);

let input1 = newMatrix(size);

let div2 = newMatrix(size);

let label1 = newMatrix(size);

let A = newMatrix(size);

let A1 = newMatrix(size);

let done = document.createElement('input');

let div3 = document.createElement('div');

let lammbda = [];

let text = [];

let header2 = document.createElement("h6");

let header1 = document.createElement("h6");

let table2 = document.createElement('table');

let tr2 = [];

done.setAttribute('value', 'calculate');

done.setAttribute('type','button');

done.classList = 'btn';

for(let i = 0; i < size; i++){

tr1[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td1[i][j] = document.createElement('td');

div2[i][j] = document.createElement('div');

if(size < 5 ){

div2[i][j].classList = "input-field col s5";

} else {

div2[i][j].classList = "input-field";

}

var inputValue = "A["+(i+1)+"]["+(j+1)+"]";

input1[i][j] = document.createElement('input');

label1[i][j] = document.createElement('label');

label1[i][j].id = inputValue;

label1[i][j].innerHTML = inputValue;

input1[i][j].setAttribute('type', 'number');

input1[i][j].setAttribute("id", inputValue);

input1[i][j].setAttribute("name", inputValue);

}

}

done.addEventListener('click', done1);

function done1() {

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

A[i][j] = parseFloat(input1[i][j].value);

}

}

A1 = yakobAm(A,size);

let iteraci = 1;

let t = true;

let aa = newMatrix(size);

while(t) {

let AM = yakobAm(A, size);

for(let i = 0 ; i < size; i++) {

for(let j = 0 ; j < size; j++){

if(i < j || i > j) {

AM[i][j] < 0;

iteraci++;

t = true;

} else {

aa[i][j] = AM[i][j];

t = false;

}

}

}

}

let td2 = newMatrix(size);

for(let i = 0; i < size; i++){

tr2[i] = document.createElement('tr');

for(let j = 0; j < size; j++){

td2[i][j] = document.createElement('td');

let a = "A["+(i+1)+"]["+(j+1)+"]";

td2[i][j].classList = "text-deep-purple";

td2[i][j].innerHTML = a + " = " + aa[i][j].toFixed(3);

tr2[i].appendChild(td2[i][j]);

}

table2.appendChild(tr2[i]);

}

document.getElementById('ypart2').appendChild(table2);

let l = lambda(A1,size);

header1.innerHTML = "The matrix's own numbers are :" + linebreak;

for(let i = 0; i < size; i++){

lammbda[i] = document.createElement('img');

lammbda[i].setAttribute("src", "https://cdn.pixabay.com/photo/2013/03/30/00/09/lambda-97846\_960\_720.png");

lammbda[i].classList = "letters";

lammbda[i].style.marginTop = '0px';

text[i] = document.createElement('p');

text[i].innerHTML = "["+(i+1)+"] = " + l[i]+linebreak;

div3.appendChild(lammbda[i]);

div3.appendChild(text[i]);

}

header2.innerHTML = "The number of Ithers: K = " + iteraci + linebreak;

document.getElementById('ypart11').appendChild(header2);

document.getElementById('ypart11').appendChild(header1);

document.getElementById('ypart11').appendChild(div3);

}

for(let i = 0; i < size; i++){

for(let j = 0; j < size; j++){

div2[i][j].appendChild(input1[i][j]);

div2[i][j].appendChild(label1[i][j]);

td1[i][j].appendChild(div2[i][j]);

tr1[i].appendChild(td1[i][j]);

}

table1.appendChild(tr1[i]);

}

document.getElementById("ypart11").appendChild(table1);

let clearElement = document.createElement('input');

clearElement.setAttribute("value","Clear matrix");

clearElement.setAttribute("type", "submit");

clearElement.classList = "waves-effect waves-light btn";

clearElement.addEventListener('click', clearValue);

document.getElementById("ypart11").appendChild(done);

function clearValue() {

for(let i = 0; i < size; i++){

table1.removeChild(tr1[i]);

div3.removeChild(lammbda[i]);

div3.removeChild(text[i]);

table2.removeChild(tr2[i]);

}

document.getElementById('ypart1').removeChild(header2);

document.getElementById('ypart1').removeChild(header1);

document.getElementById('ypart1').removeChild(div3);

document.getElementById("ypart1").removeChild(done);

document.getElementById("ypart1 1").removeChild(table1);

document.getElementById("ypart1").removeChild(clearElement);

}

document.getElementById("ypart11").appendChild(clearElement);

}

for(let i = 2; i <= 100; i++){

select.appendChild(option[i]);

}

div1.appendChild(select);

document.getElementById("ydimension2").appendChild(div12);

document.getElementById("ydimension1").appendChild(div1);

document.getElementById("ydimension3").appendChild(creatMatrix);

yakobCount = false;

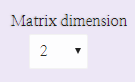
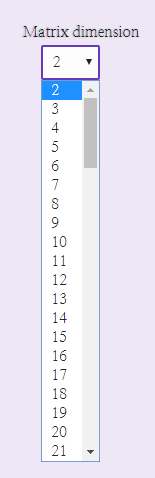
}

}

**2.4. Մեքենայական իրականացման ծրագրային միջավայրը**

Կիրառական ծրագրերի փաթեթի աշխատանքի կազմակերպումն իրականացվում է երկխոսային գործիքների միջոցով: Էկրանին բացվող էջում լրացնելով անհրաժեշտ տվյալները՝ օգտագործողը ընտում է մատրիցի չափը և ճշտությունը:

Մատրիցի չափը՝

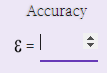
 

Նկ1:

Նկ2:

Նկար 1-ում ստեղնը սեղմելուց հետո բացվում է նկար 2-ը բացված ցուցակը, որտեղից էլ ընտում ենք մատրիցի չափաը:

Ճշտություն՝

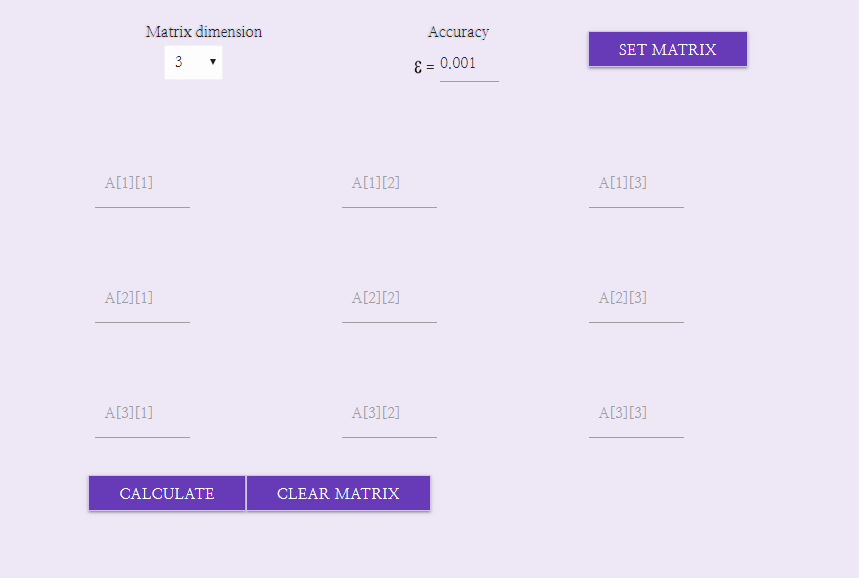


Ներմուծում ենք համապատասխան ճշտություն:

սետ.PNG ստեղնը սեղմելուց հետո անցնում է հաջորդ քայլին:

Ստեղծվում է մատրիցի չափողականության համապատասխան մատրից:

QR-ալգորիթմի, LR-ալգորիթմի,Յակոբիի-ալգորիթմի դեպքում բացվում հետևյալ պատուհանը:



Ներմուծում են մատրից և ցալ.PNG ստեղնը սեղմելուց հետո կատարվում են ալգորիթմով գրված ծրագրի գործողությունները, եթե խնդիրը լուծում չի ունենում, ապա 100 իտերացիա հետո ծրագիրը կանգ է առնում և բացվում է հետևյալ պատուհանը՝



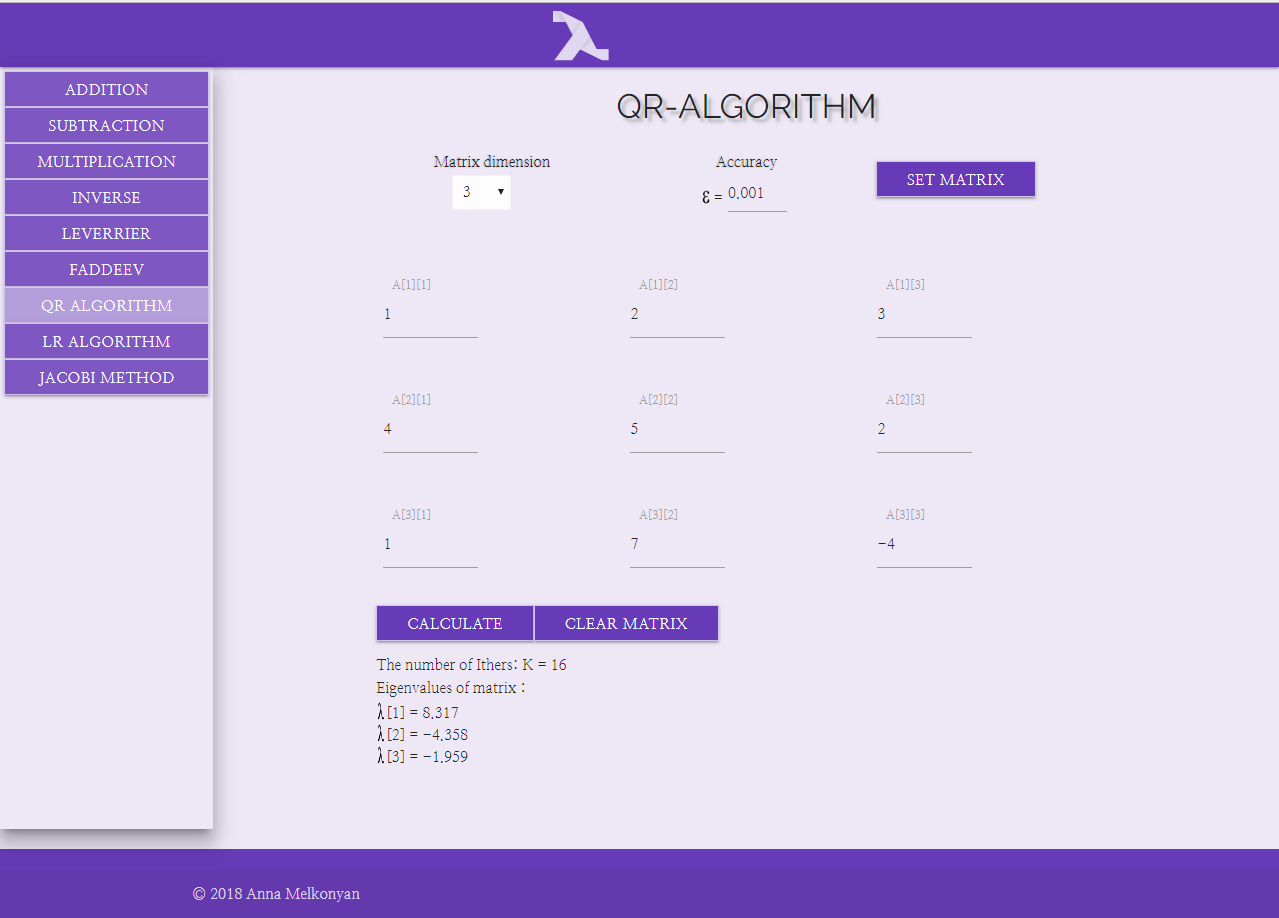
 պատուհանը ջնջում է և հնարավորություն է տալիս ներմուծել նոր մատրից:

Յակոբիի-ալգորիթմի դեպքում մատրիցը պետք է լինի սիմետրիկ : Սիմետրիկ չլինելու դեպքում ծրագիրը ընդհատվում է և բացվում է հետևյան պատուհանը՝

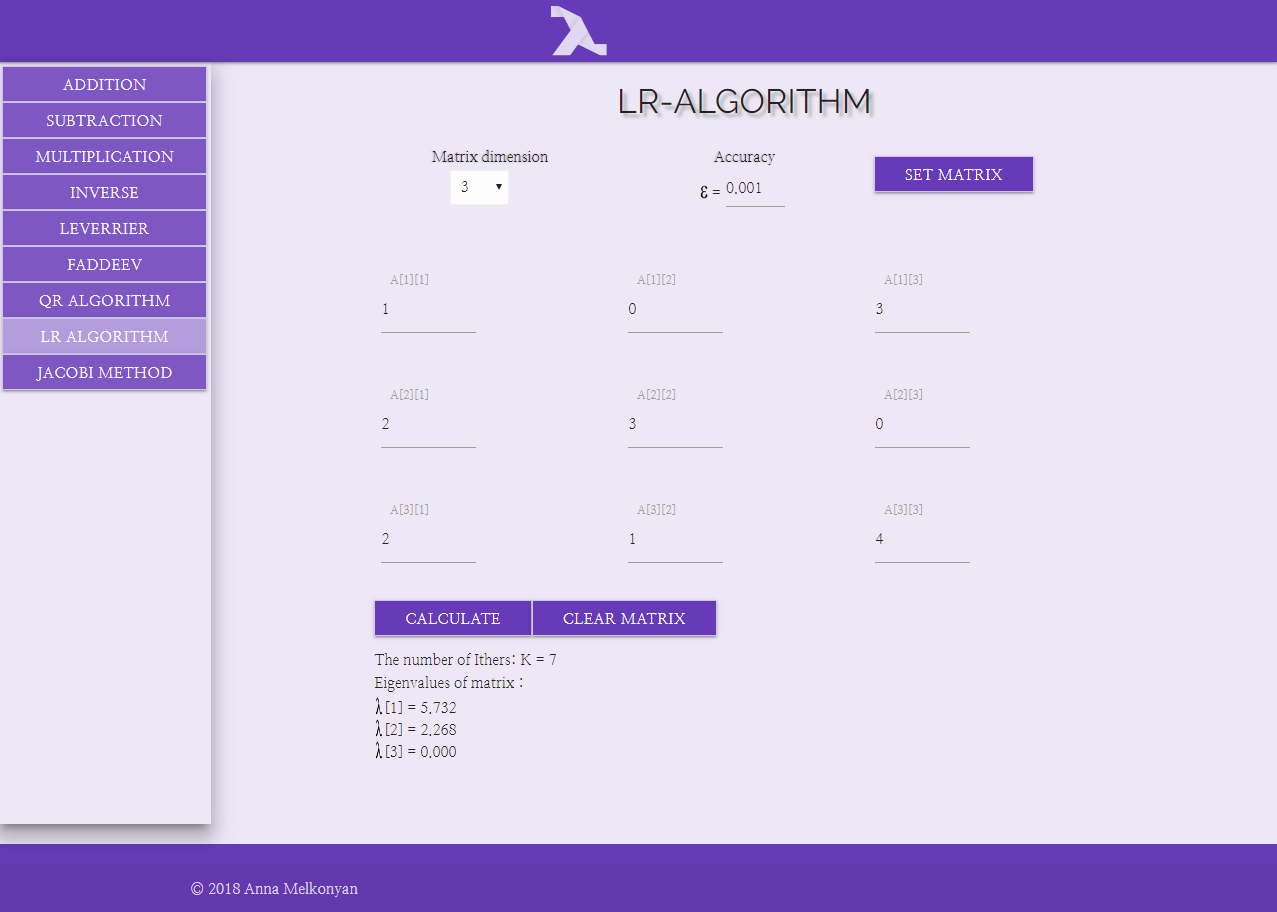


Վերջնական արդյունքում ունենք իտերացիրաների քանակը և սեփական թվերը:

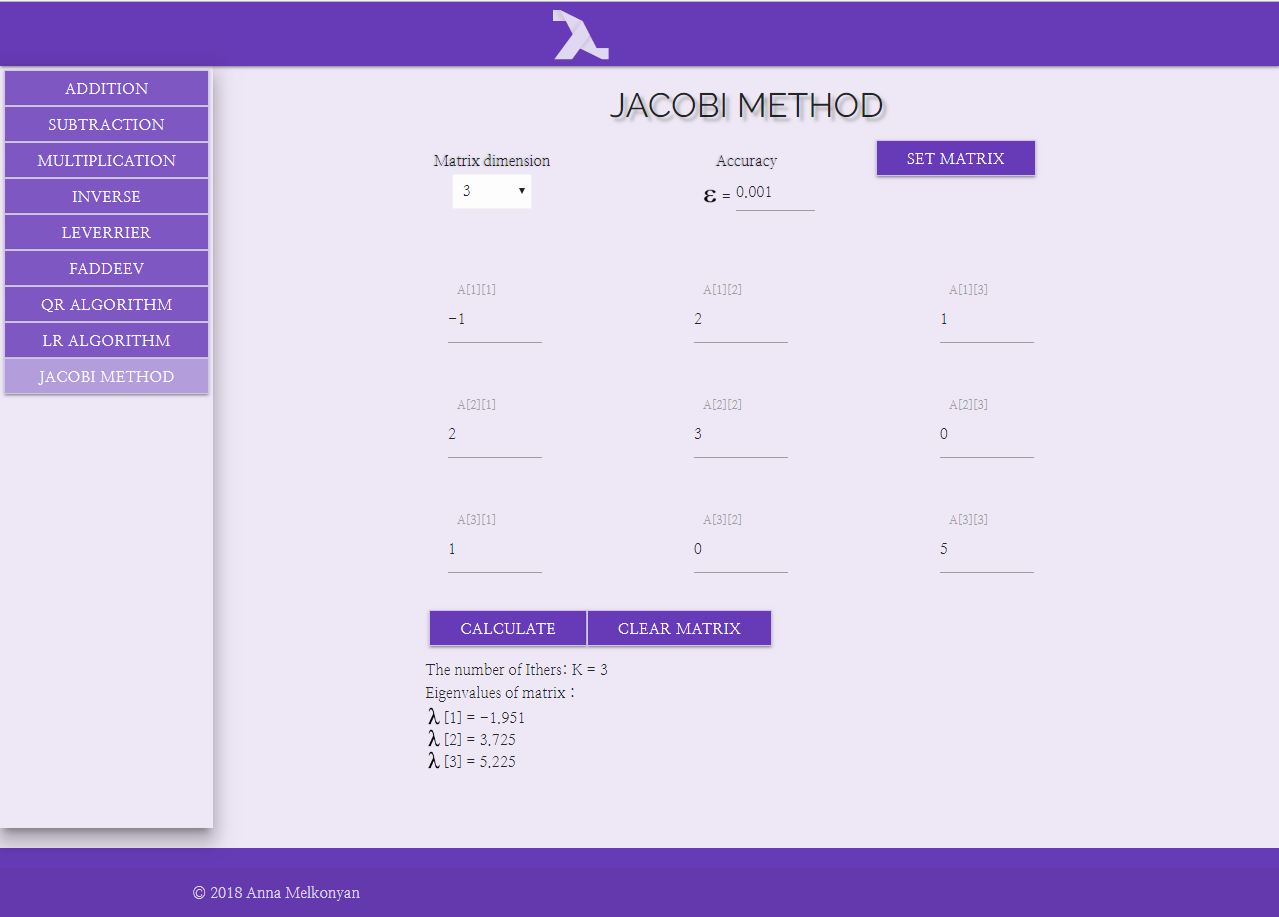
1. **QR – ալգորիթմ**



1. **LR-ալգորիթմ**

****

1. **Յակոբիի-ալգորիթմի**



**Գլուխ 3. ՓՈՐՁՆԱԿԱՆ ՀԵՏԱԶՈՏՈՒԹՅՈՒՆՆԵՐ**

**3.1. QR-ալգորիթմով ստացված արդյունքները**

Օրինակ 1.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

QR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 2.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

QL ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 3.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

QR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 4.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

QR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

**3.2. LR-ալգորիթմով ստացված արդյունքները**

Օրինակ 1.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

LR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 2.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

LR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 3.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

LR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 4.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

LR ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

**3.3 Յակոբիի ալգորիթմով ստացված արդյունքները**

Օրինակ 1.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

Յակոբիի ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 2.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

Յակոբիի ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 3.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

Յակոբիի ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

Օրինակ 4.

Սկզբնական տվյալներ.

ա) ճշտությունը՝

բ) մատրիցը, որի սեփական թվերը պետք է որոշվեն՝

Յակոբիի ալգորիթմի համաձայն խնդրի լուծման ընթացքում ստացված A մատրիցները՝

Մատրիցի սեփական թվերը՝

Իտերացիաների քանակը՝ :

ՏՆՏԵՍԱԳԻՏՈՒԹՅԱՆ

**Գլուխ 4. ԱՇԽԱՏԱՆՔԻ ՏԵԽՆԻԿԱ-ՏՆՏԵՍԱԿԱՆ ՀԻՄՆԱՎՈՐՈՒՄ.**

**4.1 Ծրագրային փաթեթի մշակման ինքնարժեքի հաշվարկը**

Ժամանակակից տնտեսական պայմաններում ցանկացած գործունեություն առնչվում է տնտեսական հարցերի հետ: Նոր տեխնիկայի, տեխնոլոգիական պրոցեսների, տեխնիկա-տնտեսական հիմնավորման համար, մասնավորապես մեր աշխատանքի պայմաններում կարևորագույն տնտեսական հիմնահարցերից է ինքնարժեքի որոշումը:

Արտադրանքի կամ ծառայությունների ինքնարժեքը` դա արտադրանքի (ծառայությունների) արտադրության և իրացման վրա կատարված բոլոր ծախսերի գումարն է դրամական արտահայտությամբ:

Ինքնարժեքի մեջ իրենց արտահայտությունն են գտնում սպառված շրջանառու ֆոնդերը, կենդանի աշխատանքի մի մասը, որը աշխատողներին վճարում է աշխատավարձի ձևով:

Ինքնարժեքի մեջ մտնող ծախսերը դասակարգվում են ըստ տնտեսական տարերի և ըստ կալկուլյացիոն հոդվածների:

Ներկայումս կիրառվում է ծախսերի ըստ կալկուլյացիոն հիմնական հոդվածների հետևյալ դասակարգումը`

1. համալրող առարկաներ,

2. էլեկտրաէներգիայի ծախսեր,

3. աշխատողների հիմնական աշխատավարձ,

4․ տարածքի վարձակալության համար ծախս:

**Լրիվ ինքնարժեք (1-4 կետերի գումարը)։**

Փաթեթը, որի ինքնարժեքն ու գինը ենթակա է որոշման, իրենից ներկայացնում է՝ պարզ թվերի որոշման հավանականային ալգորիթմների մշակման ծրագրային փաթեթ:

Հաշվարկի համար ելքային տվյալներ են հանդիսանում`

- փաթեթի մեջ մտնող համալրող առարկաների քանակն ու անվանացանկը;

- ժամանակի ամփոփ նորմերը, աշխատանքի կարգն ու աշխատավարձի ձևերը,

- ժամավճարային և գործարքային պարգևատրման չափերը (23%),

- տարածքի վարձակալություն

**4.2 Գնովի նյութերի և համալրող առարկաների ծախսի հաշվարկը**

Գնված բաղադրիչների և նյութերի արժեքը որոշվում է հետևյալ բանաձևով`

Cпк = ∑Mпkj\*Pпkj,

որտեղ` Mпkj j-րդ տեսակի գնված բաղադրիչների քանակն է, հատ, Pпkj j-րդ գնված բաղադրիչի գինը, դրամ: Հաշվարկի արդյունքները բերված են աղյուսակ 1-ում:

*Աղյուսակ 3.1.*

|  |  |  |  |
| --- | --- | --- | --- |
| Բաղադրիչի անվանումն ու տեսակը | Մեկ սարքին ընկնող քանակ.հատ | Միավորի գինը.դրամ | Մեկ սարքին ընկնող արժեք.դրամ |
| Գրիչ | 1 | 100 | 100 |
| Թուղթ | 90 | 10 | 900 |
| Կրիչ | 1 | 3600 | 3600 |
| Ընդհամենը |  |  | 4600 |

Ընդամենը` 4600 դրամ:

* 1. **Էլեկտրաէներգիայի ծախսի հաշվարկը**

Համակարգիչները և այլ սարքավորումները աշխատեցնելու համար էլեկտաէներգիայի տարեկան ծախսը որոշվում է հետևյալ բանաձևով`

**¾ = W\*U¾**

որտեղ` W-ն էլեկտրաէներգիայի տարեկան ծախսն է, UԷ-ն 1 կվ/ժ էլեկտրաէներգիայի արժեքն է (≈40 դրամ)։

Էլեկտրաէներգիայի ամսեկան ծախսը մոտավոր կկազմի W≈ 200կՎտ/ժ,

Է = 200 \* 40 = 8000 դրամ/ամսական: Ծրագրային փաթեթը իրականացնելու համար անհրաժեշտ է մոտ 4 ամիս: 4 ամսվա համար էլեկտրաէներգիայի ծախսը կկազմի 4\*8 000 = 32 000 դրամ

* 1. **Հիմնական աշխատավարձի հաշվարկը**

Հիմնական աշխատավարձի մեջ մտնում են`

● գործարքային դրույքաչափերով աշխատավարձ,

● ժամավճարային աշխատավարձ,

● պարգևավճար:

Գործարքային աշխատավարձն ըստ տարիֆային համակարգի որոշվում է հետևյալ բանաձևով`

Ահիմ.= Ժդ.\*Աարտ.,

որտեղ` Ժդ.-ն ժամային դրույքաչափն է, Աարտ.-ն արտադրանքի աշխատատարությունը, ժամային նորմը: Հաշվարկման արդյունքները բերված են աղյուսակ 2-ում:

*Աղյուսակ 3.2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Գործառույթի հաջորդական. | Մասնագետ | Վճարման ձև | Ժամանա-կային նորմ | Ժամային դրույք | Տարիֆային ֆոնդ |
| Նախապատ-րաստում | Նախագծող | պայմանագրային-պարգևա-վճարային | 120 | 2 000 | 240 000 |
| Մշակում | Ծրագրավորող | պայմանագրային -պարգևա-վճարային | 160 | 5 000 | 800 000 |
| Կարգավորում | Ծրագրավորող | պայմանագրային -պարգևա-վճարային | 240 | 5 000 | 1 200 000 |
| Տեստավորում | QA (որակի երաշխավորող) | պայմանագրային -պարգևա-վճարային | 120 | 3 000 | 360 000 |
| Ընդամենը |  |  | | | 2 600 000 |

Պարգևատրման չափը որոշվում է հետևյալ բանաձևով`

Պ= Ահիմ \* Պդ/100%,

որտեղ` Պդ -պարգևատրման դրույքաչափ, 23%

Պ = 2600000 \* 0.23 = 598 000 դրամ:

* 1. **Հիմնական միջոցների ամորտիզացիա**

Հիմնական արտադրական միջոցների տարեկան ամորտիզացիան (ԱՏ) հաշվարկվում է հետևյալ բանաձևով`

**ԱՏ=ՀԱ/Ն,**

որտեղ ՀԱ-ն հիմնական միջոցների սկզբնական արժեքն է, Ն-ն հիմնական միջոցների օգտակար գործունեության ժամկետն է:

*Աղյուսակ 3.*

|  |  |  |  |
| --- | --- | --- | --- |
| Հիմնական միջոցի անվանումն ու տեսակը | Հիմնական միջոցի սկզբնական արժեքը | Ամորտիզացիոն հատկացումներ | |
| ՀՄ օգտակար գործողության ժամկետ, տարի | Ամորտիզացիոն ծախս |
| Համակարգիչ  (Lenovo 80XR00L9RU) | 236 000 | 5 | 52 000 |
| Ընդամենը |  |  | 52 000 |

*Աղյուսակ 4*

|  |  |  |
| --- | --- | --- |
| Ծախսի անվանումը | Դրույքաչափը | Տարեկան ծախսը |
| Սարքավորումների ամորտիզացիա | - | 52 000 |
| Ընդամենը |  | 52 000 |

**4.6 Տարածքի վարձակալության ծախսի հաշվրկը**

Գործունեության արդյունավետության հիմնական պայմաններից մեկը տարածքի ճիշտ ընտրությունն է: Վարձակալվող տարածքը կգտնվի Երևանի կենտրոնական մասում:Այս տարածքում անհրաժեշտ տարածքի վարձակալության ամսեկան ծախսը կկազմի 200 000 դրամ: 3 ամսվա համար տարածքի վարձակալության ծախսը կկազմի՝

200 000 \* 3 = 600 000 դրամ ։

**4.7 Լրիվ ինքնարժքի կալկուլացիան**

Լրիվ Ինքնարժեքի կալկուլացիան բերված է աղյուսակ 5-ում:

*Աղյուսակ 5.*

|  |  |  |
| --- | --- | --- |
|  | Ծախսերի հոդվածի անվանումը | գումարը, դրամ |
| 1. | Համալրող առարկաներ | 4600 |
| 2. | Էլեկտրաէներգիա | 32 000 |
| 3. | Ընդհանուր հիմնական աշխատավարձ | 2 600 000 |
| 4. | Տարածքի վարձակալություն | 600 000 |
|  | Լրիվ ինքնարժեք | 2 858 000 |

**4.8 Շահույթի և միավորի գնի հաշվարկը**

Ծրագրային փաթեթի գնի հաշվարկը իրականացվում է արտադրական ծախսերին ուղված մեթոդով: Ձեռնարկության գինն իր մեջ ընդգրկում է ձեռնարկության շահույթն ու լրիվ ինքնարժեքը: Ձեռնարկության շահույթի հաշվարկն իրականացվում է հետևյալ բանաձևով`

Շ = Ի լրիվ\*%Շ/100,

որտեղ` Իլրիվ – ծրագրային փաթեթի լրիվ ինքնարժեքն է, %П – ձեռնարկության շահույթի դրույքաչափը, 21%

Շ = 2 858 000 \* 0.21 = 600 180 դրամ.

Ձեռնարկության գնի հաշվարկը իրականացվում է հետևյալ կերպ`

Գձեռ = Իլրիվ + Շ;

Գձեռ = 2 858 000 + 600 180 = 3 458 180 դրամ:

Ծրագրային փաթեթի բացթողնման գինը որոշվում է հետևյալ բանաձևով`

Գբաց. = Գձեռ + ԱԱՀ,

որտեղ` Գբաց.– ծրագրային փաթեթի բացթողնման գինն է, ԱԱՀ-ն ավելացված արժեքի հարկը (20%):

Գբաց. = 3 458 180 + 3 458 180 \* 0.2 = 4 149 816 դրամ:

Այսպիսով, կատարված հաշվարկների արդյունքում ստացանք, որ ծրագրային փաթեթի միավորի լրիվ ինքնարժեքը կկազմի 2 858 000 դրամ, իսկ գինը` 4 149 816 դրամ:

ԲՆԱՊԱՀՊԱՆՈՒԹՅՈՒՆ

**Գլուխ 5. ԷԼԵԿՏՐԱՄԱԳՆԻՍԱԿԱՆ ԱԼԻՔՆԵՐԻ ԱԶԴԵՑՈՒԹՅՈՒՆԸ ՇՐՋԱԿԱ ՄԻՋԱՎԱՅՐԻ ՎՐԱ**

**Համակարգիչը որպես փոփոխական էլեկտրամագնիսական դաշտի աղբյուր**

Անհատական համակարգչի կազմի հիմնական մասերն են համակարգային բլոկը (պրոցեսոր), տարբեր ինֆորմացիոն մուտքի-ելքի սարքեր՝ ստեղնաշար, տպիչ, սքաներ և այլն: Յուրաքանչյուր անհատական համակարգիչ ներառում է տեղեկատվության տեսողական արտացոլման միջոցը, որը կոչվում է տարբեր անուներով` մոնիտոր, դիսպլեյ: Որպես կանոն, նրա հիմքում ընկած է էլեկտրոնաճառագայթման փողակը: Անհատական համակարգիչը հաճախ սարքավորում են ցանցային զտիչներով՝ անխափան սնուցման աղբյուրով և այլ օժանդակ էլեկտրասարքավորումներով: Բոլոր այս տարրերը ձևավորում են բարդ էլեկտրամագնիսական իրադրություն աշխատանքի վայրում:

Աղյուսակ 2

|  |  |
| --- | --- |
| **Աղբյուրը** | **Հաճախության սահմանը** |
| Մոնիտոր | 50Հց |
| Ստատիկ կերպարանափոխիչ լարում սննուցման բլոկում | 20-100ԿՀց |
| Թակալված լայնիչ և սինխրոնիզացիայի բլոկ | 15, 110ԿՀց |
| Մոնիտորի անոդային լարման արագացնող | 0Հց |
| Համակարգչային բլոկ (պրոցեսոր) | 50Հց-1000ՄՀց |
| Մուտքի-ելքի ինֆորմացիայի սարք | 0Հց, 50Հց |
| Անխափան սնուցման աղբյուր | 50Հց, 20-100ԿՀց |

Անհատական համակարգիչով ստեղծված Էլեկտրամագնիսական դաշտն ունի բարդ լուսապատկերային կառուցվածք՝ 0 Հց-ից մինչև 1000 ՄՀց հաճախության սահմաններում:

Էլեկտրամագնիսական դաշտն ունի էլեկտրական (E) և մագնիսական (H) միակցում: Ընդ որում, նրանց փոխադարձ կապը բավականին բարդ է: Այդ պատճառով E-ի և H-ի գնահատումը տեղի է ունենում առանձին-առանձին:

Մոնիտորի հետ աշխատանքի ժամանակ էկրանի կինեսկոպին հավաքվում է էլեկտրաստատիկ դաշտ ստեղծող էլեկտրաստատիկ լիցք: Տարբեր հետազոտություններում, տարբեր պայմաններում էլեկտրաստատիկ դաշտի չափման նշանակությունը տատանվել է 8-ից մինչև 75կՎ/մ: Այդ դեպքում մոնիտորի հետ աշխատող մարդիկ ձեռք են բերում էլեկտրաստատիկ պոտենցիալ: Օգտագործողների էլեկտրաստատիկ պոտենցիալների ցրումը տատանվում է 3-ից մինչև 15կՎ/մ սահմանում:

Նկատելի ներդրում ընդհանուր էլեկտրաստատիկ դաշտը մտցնում է ստեղնաշարի և մկնիկի վերևի մասի շփման էլեկտրականացումից: Փորձերը ցույց են տալիս, որ նույնիսկ ստեղնաշարի հետ աշխատելուց հետո, էլեկտրաստատիկ դաշտը շատ արագ աճում է 2-ից մինչև 12կՎ/մ: Առանձին աշխատանքային տեղերում գրանցվել է 20կՎ/մ-ից մեծ ստատիկ էլեկտրական լարման դաշտ:

**Էլեկտրամագնիսական դաշտի և այլ անբարենպաստ գործոնների ազդեցությունը համակարգիչ օգտագործողի առողջության վրա**

Օգտագործողի առողջության վրա էլեկտրամագնիսական դաշտի հնարավոր անբարենպաստ ազդեցության թեմայով առաջին համալիր հետազոտություն անցկացվել է 1984 թվականին Կանադայում: Աշխատանքի իրականացման համար պատճառ է հանդիսացել հոսպիտալներից մեկի հաշվապահության աշխատակիցների կողմից բազմաթիվ բողոքները: Աշխատանքի վերջում հաստատվել է միանշանակ կապ հիվանդության առաջնային արտաքին ազդեցության գործոններից մեկի համակարգչի մոնիտորով գեներացնող էլեկտրամագնիսական դաշտով: Տվյալների ամփոփումից հետո պարզվել է, որ օրվա մեջ 2-ից 6 ժամ մոնիտորի հետ աշխատողների մոտ կենտրոնական նյարդային համակարգի ֆունկցիոնալ շեղումները 4,6 անգամ ավելի հաճախ են լինում, քան վերահսկված խմբերում, սրտանոթային համակարգի հիվանդությունները՝ 2 անգամ ավելի, վերին շնչառական ուղիների հիվանդությունները՝ 1,9 անգամ ավելի: 1996 թվականին էլեկտրամագնիսական անվտանգության կենտրոնում կատարված համակարգիչ օգտագործողի ֆունկցիոնալ վիճակի հետազոտությունները ցույց տվեցին, որ նույնիսկ կարճաժամկետ աշխատանքի (45 րոպե) արդյունքում մոնիտորի էլեկտրամագնիսական ճառագայթումից օգտագործողի օրգանիզմում տեղի է ունենում հորմոնային վիճակի և ուղեղի կենսահոսանքների շոշափելի փոփոխություններ:

Մոնիտորի Էկրանի ազդեցությունից աչքերը փառակալում են, այսինքն՝ աչքերի առաջ շղարշ շերտ է պատում, աչքերը հոգնում են, լարված են դառնում: Առաջանում են գլխացավեր, քունը խանգարվում է, փոփոխվում է օրգանիզմի հոգեբանաֆիզիկական վիճակը:

Որպես պաշտպանության հիմնական միջոց առաջարկում են պաշտպանիչ զտիչներ մոնիտորի էկրանի համար: Դրանք օգտագործում են սպառողի վրա վնասակար գործոնների ազդեցության դեմ պայքարի համար, լավացնում են էկրանի պարամետրերը, փոքրացնում են մոնիտերի դեպի սպառողն ուղղված ճառագայթումը:

Դիսփլեյից օգտվողների մոտ հաճախ առաջանում է սթրես: Հայտնաբերված է , որ սթրեսի աղբյուր կարող է հանդիսանալ՝ գործունեության ձևը, համակարգչի բնութագրիչ յուրահատկությունները, օգտագործվող ծրագրային ապահովումը, աշխատանքի կազմակերպումը, սոցիալական տեսակետերը:

***Օդաիոնային կազմի ազդեցությունը -*** Մարդու օրգանիզմի օդաիոնային ընկալման գոտին շնչառական ուղիները և մաշկն է: Օդաիոնների անբավարար պարունակությունը թոքերում անհատական համակարգիչների տարածքում բերում է արտահայտիչ բացասական էֆեկտի` ինքնազգացումի վատացման, բարձր հոգանծության, հաճախակի գլխացավերի, ճնշման բարձրացման:

***Տեսողության վրա ազդեցությունը-***  Տեսողության հոգնածությունը համակարգիչ օգտագործողի առողջությանը վերագրում է ախտանիշների մի ամբողջ հավաքածու. աչքերը հոգնում են, ցավում են, առաջանում է գլխացավ, խախտվում է քունը, փոփոխվում է օրգանիզմի հոգեֆիզիկական վիճակը:

Բոլոր այն արտադրական հաստատություններում, որտեղ աշխատանքը մոնիտորն

երի հետ հանդիսանում է հիմնական գործոն, պետք է ապահովվեն միկրոկլիմայի օպտի

մալ պարամետրեր: Օդի խոնավության բարձրացման համար անհրաժեշտ է կիրառել օդի խոնավացուցի ներ, որոնք ամեն օր լիցքավորում են թորած կամ եռացրած ջրով: Օրգա

նիզմում աերոիոնների նկատմամբ մեծ զգայնությամբ պարունակությամն դեպքում առա

ջացնում են բացասական անբավարարվածություն, օդի ո թարմության զգացողություն:

Բացասական ազդեցություն են ունենում նաև դրական աերոիոնների՝ հանգեցնում են ին

քնազգացողության վատացման, անքնության, աշխատանակության անկման: Համակա

րգ հետ աշխատանքը բացասական է ազդում նաև տեսողության վրա և մեծացնում է կա

տարախտի հավանակություն: Դիսփլեյից օգտվողների մոտ առաջանում է նաև մկանայի

ն թուլություն, ողնաշարի կառուցվածքի փոփոխություն: Երկարատև նույն աշխատանք

ային դիրքում մնալը առաջացնում է ոտքերի, ուսերի, վզի և ձեռքերի մկաննբերի կարճա

ցում, որի հետևանքով խախտվում է նյութերի փոխանակությունը և կուտակվում է կաթն

աթթու: Դիսփլեյից օգտվողները հաճախ նաև գտնվում են սթրեսային վիճակում:

Աղմուկը արտադրությունում լայն տարածված վտանգավոր գործոններից է, որի

ազդեծությունը ի սահմանափակվում միայն լսողական օրգաններով, նյարդային

համակարգի միջոցով, այն ազդում է նաև ներքին օրգանների վրա: Բարձր աղմուկի

պայմաններում աշխատող մարդիկ բողոքում են գլխացավերից, գերհոնածությունից,

անքնությունից, որն էլ դառնում է աշխատանքի արտադրողականության անկման

պատճառ: Աղմուկի դեմ պայքարի արդյունավետ միջոցներից է ձայնակլանի

հատկություններով օժտված նյութերի օգտագործումը: Փակ տարածությամ մեջ

ձայնային ալիքները ընկնում ոն պատերին, առաստաղին կամ հատակին որոնք ո

միայն կլանում, այլև անդրադարձնում են ձայնային էներգիան: Աղմուկից

պաշտպանելու համար գրյություն ունեն անհատական միջոցներ՝ հակաաղմուկային

ականջակալներ, խցաններ և այլն: Բոլոր այն սրահներում, որտեղ աշխատում են

վերլուծություն իրականացնող կամ պատասխանատու աշխատանք կատարող անձիք,

աղմուկի մակարդակը պետք է գերազանցի 60 դբ սահմանը, իսկ օպերատորի

սենյակում՝ 65դբ սահմանը: Հաշվողական մեքենաների <աղմկոտ սարքերով>

աքնալոգթվային փոխակերփի ներ և տպի ներ կահավորված սեմյակում աղմուկի

մակարդակը պետք է գրսզանցի 75 դբ սահմանը: Աղմկոտ սարքավորումնները, օրինակ

տպի ները, որոնց աղմուկի մակարդակը գերազանցում է թույլատրելի ափանիշը, պետք

է գտնվեն սենյակի դուրս: Իջեցել աղմուկի մակարդակը այդ սենյակում կարելի է

63...8000Հց հաճախությունների միջակայքոմ ձայնակլանման առավելագույն

գործակիցներով ձայնակլանի նյութերի կիրառմամբ: Որպես լրացուցի ձայնակլանի

մակերևությներ ծառայում են կոշտ գործվածքից միատոն վարդագույրները, որոնք

համապատասխանում են պաերի գույնին և կախված են դասերով:

ԿԵՆՍԱԱՆՎՏԱՆԳՈՒԹՅՈՒՆ

**ԳԼՈՒԽ 6.ԿԵՆՍԱԳՈՐԾՈՒՆԵՈՒԹՅԱՆ ԱՆՎՏԱՆԳՈՒԹՅՈՒՆ**

**ԱՆՎՏԱՆԳՈՒԹՅԱՆ ԿԱՆՈՆՆԵՐԻ ՊԱՀՊԱՆՈՒՄԸ**

Արտադրության մեջ կենսագործունեության անվտանգությունը բազում օրենքների և նորմերի հավաքածու է, որը ստեղծված է մարդու կյանքի անվտանգության և առողջության պահպանման համար: Աշխատանքի ընդունվելուց անմիջապես հետո աշխատողը պարտավոր է սովորել տեխնիկական անվտանգության կանոնները: Ձեռնարկությունների ղեկավարները և նրանց ենթակա ստորաբաժանումները հրանգավորում են անձնակազմին և խստորեն հետևում են անվտանգությանը: Որպես պարտադիր պայման կազմվում է փաստաթուղթ, որտեղ աշխատողները ստորագրությամբ հավաստում են, որ ծանոթացել են ձեռնարկությունում հաստատված կանոններին և նորմերին:

Արտադրության անվտանգ միկրոկլիման ապահովում է օպտիմալ ջերմաստիճանը, օդի խոնավությունն ու արագությունը: Որոշ արտադրություններ նաև վերահսկում են մթնոլորտային ճնշումը, ձայնի մակարդակը, լուսավորությունը, օդափոխությունը, օդի աղտոտվածությունը: Ըստ անվտանգության կանոնների՝ յուրաքանչյուր աշխատող պետք է ունենա իր հագուստը, գլխարկը, ձեռնոցները և այլ իրեր, որոնք կնպաստեն նրա անվտանգությանը:

Անվտանգության կանոնների խիստ պահպանումը աշխատողին զերծ է պահում վտանգներից և ռիսկերից, որոնք կարող են առաջանալ աշխատանքի ընթացքում: Յուրաքանչյուր աշխատողի համար շատ կարևոր է իմանալ, որ իր աշխատավայրը համապատասխանում է բոլոր ստանդարտներին: Աշխատատեղում դժբախտ պատահարներից խուսափելու համար մենք պարտավոր ենք իմանալ բոլոր վնասակար գործոնները և դրանցից պաշտպանվելու միջոցները: Այն տարածքում, որտեղ մենք իրականացնում ենք մեր աշխատանքային գործունեությունը, առկա են հետևյալ վնասակար գործոնները՝

* Էլեկտրական հոսանք
* Տարածքի բնական լուսավորվածության անբավարարվածություն
* Տարածքի արհեստական լուսավորվածության անբավարարվածություն
* Աղմուկի բարձր մակարդակ
* Հրդեհներ
* Ճառագայթումներ

Ոսումնասիրենք դրանք ավելի մանրամասն և գտնենք դրանցից պաշտպանվելու միջոցները:

**Էլեկտրական հոսանք**

Ներկայումս էլեկտրական հոսանքը լայն կիրառություն է գտել ժողովրդական տեսչության բոլոր բնագավառներում: Դրա համար էլ էլեկտրաանվտանգության հարցերին պետք է մեծ ուշադրություն դարձնել: Էլեկտրաանվտանգությունը կազմակերպչական տեխնիկական միջոցառումների համակարգ է, որը ապահովում է մարդկանց պաշտպանությունը էլեկտրական հոսանքի վնասակար և վտանգավոր գործոններից: Էլեկտրական հոսանքը առավել վտանգավոր է նրանով, որ արտաքուստ ոչնչով չի արտահայտվում: Հաղորդչին չհպտված մարդը չի զգում հոսանքի առկայությունը և հետևաբար չի կարող կանխատեսել սպառնացող վտանգը: Պատահարների երկարամյա ուսումնասիրությունները ցույց են տվել, որ հոսանքահարումների գերակշռող մասը տեղի են ունենում մինչև 1000Վ լարումով աշխատող սարքավորումները սպասարկող անձանց մոտ, որը բացատրվում է նրանով, որ այդ սարքերը երբեմն տեղակայվում են այնպիսի շենքերում, որտեղ մուտք ունեն կամ որտեղ աշխատում են համապատասխան մասնագիտական որակավորում չունեցող անձինք:

*Էլեկտրական հոսանքից պաշտպանվելու միջոցները.*

* Պաշտպանական հողանցում
* Զրոյացում
* Պաշտպանօչ անջատիչների օգտագործում
* Ցածր լարումների օգտագործում
* Կրկնակի մեկուսացման իրագործում
* Բաց հոսանքատար մասերի մոտ պատնեշների տեղադրում
* Օգտագործվում են ազդանաշանային սարքեր, շրջափակիչ սարքեր, հեռահար կառավարման սարքեր և ապահհովիչ սարքեր
* Աշխատանքային տեղերում փակցնում են նախազգուշական պլակատներ
* Օգտագործում են անհատական պաշտպանական միջոցներ՝ ռեզինե կոշիկներ, ձեռնոցներ, մեկուսիչ գորգեր, մեկուսիչ բռնակներով գործիքներ:

**Տարածքի բնական լուսավորվածության անբավարավածություն**

Կենսագործունեության անվտանգության ապահովման համար կարևոր խնդիրներից է աշխատանքային տեղերում սահմանված նորմերին համապատասխանող լուսավորվածության ապահովումը: Անբավարար լուսավորվածության պայմաններում աշխատողների մոտ գերլարվում են տեսողական օրգանները, խախտվում է նյարդային համակարգը, ինչպես նաև աշխատողները կարող են դիպչել սարքավորումների վտանգավոր գոտիներին և վատացնել թողարկվող արտադրանքի որակը:

Բնական լույսը աշխատանքային շինություններում կարող է ներթափանցել.

* Պատերից (պատուհանների միջոցով)
* Վերևից (տանիքներում արված լուսային երթիկների միջոցով)
* Վերևից ու կողքից միաժամանակ

Քանի որ բնական լույսն օրվա մեջ կախված տարբեր պատճառներից փոփոխվում է, հետևաբար աշխատանքային շինությունում բնական լուսավորվածության աստիճանը գնահատելու համար կիրառվում է բնական լուսավորվածության գործակցի հասկացությունը: Բնական լուսավորվածության գործակցի նորմերը սահմանելիս` հաշվի է առնվում կատարվող աշխատանքի բնույթը: Տեսողական աշխատանքի կարգը կախված է այն դետալի կամ օբյեկտի չափսերից, որի վրա կատարվում է տվյալ տեսողական աշխատանքը: Այսինքն, եթե դետալի կամ օբյեկտի չափերը փոքր են, ապա նրա վրա կատարվող տեսողական աշխատանքի կարգը կլինի բարձր: Այդ կարգի տեսողական աշխատանքի համար բնական լուսավորվածության գործակցի սահմանված նորման պետք է լինի մեծ: Դետալի կամ օբյեկտի մեծ չափսերի համար տեսողական աշխատանքի կարգը կլինի ցածր, և այդ ցածր աշխատանքի բնական լուսավորվածության գործակցի սահմանված նորման կլինի փոքր:

Բնական լուսավորվածությունը կարելի է մեծացնել` ավելացնելով լուսամուտների քանակը, տանիքում արված երթիկների քանակը և այլն:

**Տարածքի արհեստական լուսավորվածության անբավարարվածություն**

Աշխատանքային շինություններում արհեստական լուսավորվածությունը պետք է համապատասխանի հետևյալ պահանջներին:

* Լուսավորվածությունը պետք է բավարարի սահմանված նորմերին և իր սպեկտրով նմանվի բնական լույսին
* Աշխատանքային մակերևույթի վրա պայծառությունը պետք է տեղաբաշխված լինի հավասարաչափ
* Աշխատանքային մակերևույթի վրա չպետք է լինեն ցայտուն ստվերներ
* Տեսողական դաշտում չպետք է լինեն ուղղակի և անդրադարձնող փայլեր
* Լուսավորվածությունն ամբողջ աշխատաժամանակի ընթացքում պետք է լինի հաստատուն և չբաբախող
* Լույսի հոսքի ուղղվածությունը պետք է լինի օպտիմալ
* Լույսի անհրաժեշտ սարքավորումները պետք է լինեն անվտանգ և ընտրվեն տնտեսապես նպաստավոր միջոցներով
* Լուսավորման տեղակայանքը պետք է լինի շահագործման համար հարմար, պարզ և գեղագիտական պահանջները բավարարող:

**Աղմուկի բարձր մակարդակ**

Աղմուկի դեմ պայքարի հարցերը ներկայումս նշանակություն ունեն բոլոր բնագավառներում: Աղմուկը կենսագործունեության միջավայրում պատճառում է մեծ վնասներ: Այն բացասաբար ազդելով մարդու օրգանիզմի վրա` իջեցնում է աշխատանքի արտադրողականությունը և պատճառ է դառնում սխալ գործերի, ինչն էլ նպաստում է դժբախտ դեպքերի առաջացմանը:

*Աղմուկից պաշտպանվելու միջոցներն են.*

* Աղմուկի մակարդակի նվազեցում առաջացման աղբյուրում
* Աղմուկի մակարդակի նվազեցում աշխատանքային շինությունում ակուսիկ մշակմամբ
* Աղմուկի մակարդակի նվազեցումը նրա տարածման ճանապարհին
* Աղմուկի մակարդակի նվազեցում ձեռնարկությունների և արտադրամասերի ռացիոնալ հատակագծմամբ
* Անհատ պաշտպանության միջոցներ՝ ներդրակներ, ականջակալներ և սաղավարտներ:

**Հրդեհների առաջացումը և մարումը**

Հրդեհները կարող են առաջանալ հետևյալ պատճառներով.

* Սահմանված տեխնիկական ռեժիմի խախտումից
* Վառարանների ոչ ճիշտ շահագործումից
* Էլեկտրական ցանցերի և սարքավորումների գերբեռնվածությունից
* Դյուրավառ նյութերի ոչ ճիշտ շահագործումից
* Շանթարգելքների բացակայությունից
* Օդափոխիչ սարքերի անսարքությունից
* Կրակի հետ անզգույշ վարվելուց

Կառուցվածքների հրակայության սահմանները մեծացնելու համար դրանց մակերևույթները պատվում են զանազան ծածկույթներով, որոնք պետք է օժտված լինեն ցածր ջերմահաղորդականությամբ և ունենան փոքր զանգված: Առանձնապես բարձր արդյունք են տալիս գիպսի և ցեմենտի սվաղները: Փայտե կառուցվածքները հաճախ ներծծվում են զանազան քիմիական նյութերով:

**Ճառագայթումներ**

Ներկայումս արդյունաբերության մեջ և տարբեր բնագավառներում լայն կիրառություն են գտել ռադիոակտիվ նյութերն ու իզոտոպերը, որոնք առանձնապես օգտագործվում են գիտահետազոտական աշխարհում, չափիչ, կարգավորիչ սարքերում, ավտոմատ համակարգերում և այլն: Դրանց դրական հատկություններին զուգընթաց ռադիոակտիվ նյութերն ու իզոտոպերն ունեն նաև մարդկանց համար չափազանց վտանգավոր հատկություններ, հետևաբար նրանց օգտագործման համար պահանջվում են յուրահատուկ անվտանգ պայմաններ:

Ռադիոակտիվ ճառագայթներից պաշտպանվելու առավել տարածված միջոց է էկրանավորումը: Այն հնարավորություն է տալիս աշխատատեղերում ճառագայթման ինտենսիվությունը հնարավոր չափով թուլացնելու և հասցնելու մինչև անվնաս մակարդակների: Ճառագայթներից պաշտպանվելու համար բավական է միայն մի քանի սանտիմետր օդային շերտի, չնչին հաստությամբ ապակու, նրբաթիթեղի կամ այլ խիտ նյութերի առկայությունը: Մասնիկներից էկրանի նյութ է հանդիսանում ալյումինը, պողպատը և թուջը: Նեյտրոնային ճառագայթներից պաշտպանվելու համար օգտագործում են ջուր, պարաֆին, գրաֆիտ և բերիլիում: Աշխատողները պետք է կրեն անհատական պաշտպանության միջոցներ:

**ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ**

Աշխատանքում ուսումնասիրվել են ավտոնոմ մատրիցների բոլոր սեփական թվերի որոշման մի շարք եղանակներ, որոնցից են QR-ալգորիթմը, LR- ալգորիթմը և սիմետրիկ մատրիցների սեփական արժեքների փնտրման Յակոբիի պտտման եղանակը: Ուսումնասիրելով այս եղանակները հանգում ենք հետևյալ եզրակացություններին.

1. QR-ալգորիթմը թույլ է տալիս լուծել ավտոնոմ մատրիցների սեփական արժեքների լրիվ հիմնախնդիրները: Այս մեթոդի մեքենայական իրականացման տրամաբանությունը բավականին պարզ է: Մյուս կողմից, ընդհանուր առմամբ, QR-ալգորիթմի միջոցով խնդիրը լուծելիս անհրաժեշտ է իրակաացնել մեծ թվով իտերացիաներ, ինչն իր հերթին պահանջում է մեծ մեքեայական ժամանակ:
2. LR-ալգորիթմի մեքենայական իրականացումը նույնպես բավականին պարզ է, սակայն այն կատարում է շատ ավելի իտերացիաներ քան QR-ալգորիթմը, ինչն էլ ավելի շատ մեքենայական է պահանջում:
3. Յակոբիի պտտման մեթոդը հիմնված է օրթոգոմալ ձևափոխությունների կիրառման վրա, որոնք չեն մեծացնում հաշվակների սխալների չափը: Մյուս կողմից, իտերացիայի յուրաքանչյուր քայլում բացարձակ արժեքով ամենամեծ ոչ անկյունագծային տարրի ընտրության վրա սովորաբար ծախսվում է մեծ մեքենայական ժամանակ: Սակայն ի տաբերություն մյուս երկու մեթոդների սիմետրիկ մատրիցների դեպքում Յակոբիի մեթոդով կատարվում են մինիմալ քանակությամբ իտերացիաներ:

Այս եզրակացությունից ելնելով կարելի է ասել, որ այս երեք եղանակներից ոչ սիմետրիկ մատրիցների սեփական թվերի որոշման համար լավագույնը հանդիսանում է QR-ալգորիթմը, իսկ սիմետրիկ մատրիցների համար՝ Յակոբիի պտտման եղանակը:

Օգտագործվող գրականության ցանկ

1. Սիմոնյան Ս.Հ. Հաշվողական մեթոդների կիրառման տեսություն, Երևան, 2001. – 215 էջ:
2. Анализ и оптимальный синтез на ЭВМ систем управления / Под ред. А. А. Воронова и И. А. Орурка. М., «Наука», 1984. – 343с.
3. Афанасьев В.Н., Кольмановский В. Б., Носов В. Р. Математическая теория конструирования систем управления. М., Вышая школа, 1984. – 565с.
4. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы, М., «Наука», 1987.-598с.
5. Березин И. С., Жидков Н. П., Методы вычислений. М., Физматгиз, 1959, ч. 1;

1962, ч2:

1. Богачев К. Ю. Методы решения линейных систев и нахождения собственных значий – МоскваИзд.-во МГУ, Москва 1998.-98с.
2. Воеводин В. В. Вычислительные основы линейной алгебры. М., «Наука», 1980-303 с.
3. Волков Е. А. Численные методы. М., «Наука», 1975 – 254с
4. Икрамов Х. Д. Численные решение матричных уравнений. М., «Наука», 1984 -190с
5. Кормен Т., Лейзерсон Ч., Р. Алгоритмы. Построение и анализ МЦНМО. М., 2000.-955с
6. Нособ В. Р. Алгоритмы решения задач и линейной алгебры. М., Изд. МИЭМ, 1983
7. Самарский А. А. Введение в численные методы. М., «Наука», 1982. – 271 с
8. Уилкинсон Дж. Алгебраическая проблема собственных значений. М., «Наука» 1970. – 564с.
9. Фаддеев Д. К., Фаддеев В. Н. Вычислительные методы линейной алгебры. М., «Наука», 1978: