ივანე ჯავახიშვილის სახელობის თბილისის სახელმწიფო
უნივერსიტეტი

სამაგისტრო ნაშრომი

ავტორი: სოლომონ კურტანიძე

ერთი დარეზერვებული სისტემა და მისი რიცხვითი ამოხსნა

სამაგისტრო პროგრამა: გამოყენებითი მათემატიკა

ნაშრომი წარმოდგენილია მეცნიერებათა მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი:

თემურ ჯანგველაძე, ფიზიკა-მათემატიკის მეცნიერებათა დოქტორი, პროფესორი

თბილისი, 2017

ანოტაცია

განხილულია დარეზერვებული სისტემა m რაოდენობის ძირითადი და n რაოდენობის სარეზერვო ელემენტებით, სადაც მიმდინარეობს აღდგენისა და ჩანაცვლების პროცესები. აგებულია ამ სისტემის რიცხვითი ამოხსნის ალგორითმი. კონკრეტულად, შესასწავლი პირველი რიგის კერძოწარმოებულებიანი დიფერენციალური განტოლებათა სისტემისათვის განხილულია სასაზღვრო ამოცანა, სადაც ფიგურირებს ინტეგრალური სასაზღვრო პირობებიც.

რიცხვითი ექსპერიმენტები ჩატარებულია იმ შემთხვევებში, როცა $m = 3$ და $n = 2$. გაანალიზებულია რიცხვითი ექსპერიმენტის შედეგები.

Annotation

Considered “Redundant System” with m-number main and n-number stand-by components, where repairman and replacement operations are performed. Built the algorithm of the numerical solve of this system. Concrete, considered boundary conditions that also contain integral boundary conditions for the system of first order partial differential equations that are being studied.

Carried out numerical experiments when $m = 3$ and $n = 2$. Analyzed the results of this numerical experiments.

ს ა რ ჩ ე ვ ი

შესავალი 5

თავი 1. ერთი დარეზერვებული სისტემის დიფერენციალური და ინტეგრო-დიფერენციალური მოდელი 13

1.1. ამოცანის დასმა 13

1.2. მათემატიკური მოდელი. 14

თავი 2. სასრულ-სხვაობიანი სქემა და მისი ალგორითმიზაცია 20

თავი 3. რიცხვითი ექსპერიმენტები და მათი შედეგები 23

3.1. ექსპერიმენტი N1. 23

3.2. ექსპერიმენტი N2. 25

3.3. ექსპერიმენტი N3. 28

3.4. დასკვნა. 31

გამოყენებული ლიტერატურა 32

დანართი. 34

შესავალი

გასული საუკუნის 50-იან წლების დასაწყისში ა.შ.შ.-ში ჩამოყალიბდა მეცნიერებისა და ტექნიკის ახალი მიმართულება - საიმედოობის თეორია და ის ძალიან ინტენსიურად ვითარდებოდა (იხ. მაგალითად, [2] და მასში მოცემული ლიტერატურული მითითებები). ამის მთავარი მიზეზი იყო ის, რომ ტექნიკის საიმედოობის პრობლემები მოდიოდა ა.შ.შ.-ს სამხედრო სამრეწველო კომპლექსიდან. ამ თეორიის ერთ-ერთი არსებითად მნიშვნელოვანი ნაწილი არის საიმედოობის მათემატიკური თეორია. მისი დანიშნულებაა ტექნიკური სისტემების საიმედოობის ანალიზის, შეფასებისა და ოპტიმიზაციის მათემატიკური მოდელების შექმნა, გამოკვლევა და გამოყენების მეთოდების შემუშავება.

იმავე 50-იანი წლების შუა პერიოდში დაიწყო ასევე ინტენსიური მუშაობა ყოფილ საბჭოთა კავშირში, სადაც სამხედრო სამრეწველო კომპლექსის როლი განსაკუთრებით მნიშვნელოვანი იყო. სწორედ საიმედოობის პრობლემა გამოცხადდა ნომერ პირველ პრობლემად.

60-იანი წლების შუა ხანებში უკვე მიღებული იყო საიმედოობის თეორიის მნიშვნელოვანი შედეგები, განსაკუთრებით საიმედოობის მათემატიკურ თეორიაში.

ამ დარგის ჩამოყალიბებაში საყოველთაოდ აღიარებულია ისეთი მეცნიერები როგორებიცაა ჯ. ფონ ნეიმანი, კ. შეონი, ვ. სიფოროვი, ნ. ბრუევიჩი, ბ. გნედენკო (იხ. მაგალითად, [4] და მასში მოცემული ლიტერატურული მითითებები), ი. უშაკოვი და სხვები. მათ გვერდით ღირსეული ადგილი უჭირავს ცნობილ ქართველ მეცნიერს - შ. ზებიაშვილს.

დიდია ქართველ მეცნიერთა წვლილი საიმედოობის თეორიის შემდგომ განვითარებაში. ი. კაკუბავას, ც. მირცხულავას, ი. მიქაძის, ო. ნამიჩეიშვილის, გ. ცირამუას, რ. ხუროძის (იხ. მაგალითად, [6] - [10] და მასში მოცემული ლიტერატურული მითითებები) შრომები აღიარებულია წამყვან სამეცნიერო წრეებში. მათ საფუძველი ჩაუყარეს და დაამუშავეს საიმედოობის თეორიისა და პრაქტიკის მრავალი ასპექტი.

რთული სისტემების ხარისხისა და საიმედოობის პრობლემის მოგვარების მიმართულებით გასული საუკუნის 70-იან წლებში ჩაისახა ახალი მიდგომა, რომელმაც აღიარება მოიპოვა გასული საუკუნის 80-90-იან წლებში. ამ მიდგომის თეორიული დაფუძნების სათავეებთან იდგნენ აკადემიკოსები ვ. კაფაროვი და ვ. ტრაპეზნიკოვი, ხოლო შემდეგში მის განვითარებაში დიდი წვლილი შეიტანეს ცნობილმა მეცნიერებმა: გ. გრუნმა,

გ. დრუჟინიანმა, ბ. ვოლოკმა, ი. ზარენინმა, ი. კოფანოვმა, ბ. კრედენცერმა, ვ. მეშალკინმა, ვ. ნოიმანმა, ი. რიაბინინმა, ი. უშაკოვმა და სხვებმა.

პრობლემის არსი იმაში მდგომარეობს, რომ რთული ტექნიკური ობიექტის ხარისხისა და საიმედოობის უზრუნველყოფის ღონისძიებათა ერთობლიობა განხილულ იქნას, როგორც ობიექტის სტრუქტურულ შეშფოთებათა (მტყუნებებისა და სხვა სახის შეშფოთებებისა) კომპენსაციის ორგანიზაციის, ანუ სტრუქტურული მართვის პრობლემა. მართვის ქვეშ აქ იგულისხმება როგორც ავტომატური, ასევე ავტომატიზებული მართვის სახეობები, მათ შორის ორგანიზაციულ-ადმინისტრაციული მართვა, ხარისხისა და საიმედოობის კონტროლი და მენეჯმენტი.

გასული საუკუნის 90-იანი წლებიდან აღნიშნული მიდგომის მრავალი ასპექტი წარმატებით ვითარდება საქართველოშიც. ამ მიმართულებით მნიშვნელოვანი შედეგები აქვთ მიღებული ქართველ მკვლევარებს ი. კაკუბავას, ი. მიქაძეს, რ. ხუროძეს და მათ მიმდევრებს. წარმოდგენილი ნაშრომი, გარკვეულწილად ხსენებულ მეცნიერეთა შედეგებითაა ინიცირებული.

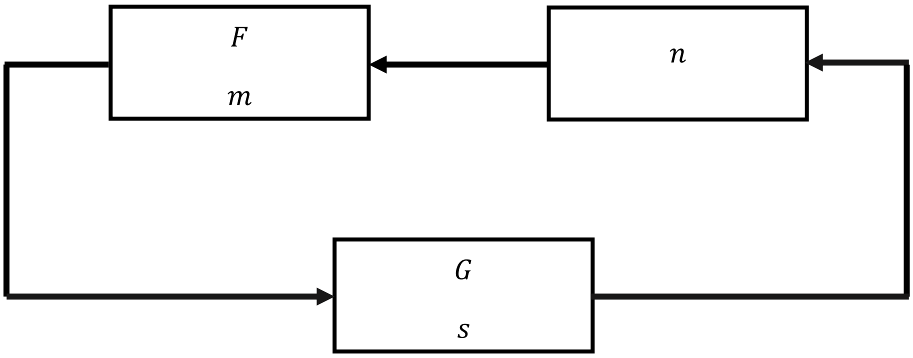
კლასიკური საიმედოობის თეორია ძირითადად ორიენტირებული იყო მოწყობილობათა საიმედოობის უზრუნველყოფაზე. ესენია საიმედოობის თვალსაზრისით ე.წ. ბინარული ტიპის ტექნიკური სისტემები. ეს სისტემები საიმედოობის თვალსაზრისით შეიძლება იმყოფებოდეს მხოლოდ ორ მდგომარეობაში: ქმედუნარიანი ან მტყუნებული. საიმედოობის უზრუნველყოფაში გამოყენებულ მეთოდებს შორის ერთ–ერთი უმთავრესი იყო დარეზერვება. ამასთან, უმრავლეს შემთხვევებში მტყუნებული ელემენტი ექვემდებარება რემონტს, რომლის შემდეგ გარემონტებული ელემენტი იდენტურია ახალი ელემენტის.

სწორედ აღდგენადი დარეზერვებული ტექნიკური სისტემების მათემატიკური მოდელირება შეადგენს კლასიკური საიმედოობის მათემატიკური თეორიის ძირითად მიმართულებას.

სარეზერვო ელემენტები უშუალოდ იყო მიმაგრებული ძირითად ელემენტებზე, უფრო ზუსტად, ძირითადი ელემენტის მუშაობის დროს, სარეზერვო ელემენტი სამუშაო მზადყოფნაში იმყოფებოდა და ძირითადი ელემენტის გაფუჭებისთანავე იწყებდა ფუნქციონირებას. ამის გამო, ჩანაცვლების პროცესს დრო არ სჭირდებოდა. ამის მაგალითად შეგვიძლია მოვიყვანოთ გემები, რომლებსაც ორი რადარი უყენიათ და

ერთდროულად ტრიალებენ, მაგრამ ამათგან ერთი ძირითადია, ხოლო მეორე სარეზერვო. პირველის დაზიანების მომენტშივე ფუნქციონირებას იწყებს სარეზერვო რადარი.

კლასიკურ თეორიაში ეს ამოცანები სქემატურად ასე გამოისახებოდა.



დიაგრამა N1. ბარლოუს დიაგრამა (Richard E. Barlow (იხ. მაგალითად, [3]))

მოცემულ სქემაში განიხილავდნენ ისეთ სისტემებს, სადაც F და G განაწილებული იყო ექსპონენტურად. ხოლო ელემენტების რაოდენობა m და n სასრული იყო. განიხილებოდა მათი სხვადასხვა ვარიაციები.

თემის აქტუალობა

სამეცნიერო-ტექნიკური პროგრესის მიღწევები განაპირობებენ საზოგადოების განვითარების დონეს. შესაბამისად, მეცნიერული მიღწევებით იქმნება რთული ტექნიკური და ტექნოლოგიური სისტემები, რომლებიც უზრუნველყოფენ ადამიანის როგორც ფიზიკური, ისე გონებრივი შრომის ნაყოფიერების მრავალჯერად გაზრდას.

რაც უფრო რთულია ამოცანები, რომელთა შესასრულებლადაც იქმნება ტექნიკური სისტემები, მით უფრო რთულია ამ სისტემების და მათი შემადგენელი კომპონენტების (ელემენტების) სტრუქტურა და რაოდენობა. რადგან არც ერთი ფიზიკური ობიექტი არ შეიძლება იყოს სრულყოფილად დამზადებული, ანუ მას შეიძლება გააჩნდეს უმცირესი დეფექტები მაინც. შესაბამისად ყოველთვის არის საშიშროება, რომ ეს დეფექტები ელემენტის პარამეტრების გადახრას და გარკვეული შინაგანი და გარეგანი შეშფოთებების პირობებში მის (დაზიანებას) მტყუნებას გამოიწვევს. (დაზიანების) მტყუნების საშიშროების ალბათობა დროთა განმავლობაში იზრდება, რადგან ყველა ფიზიკური ობიექტი ექვემდებარება ცვეთას და დაბერების ბუნებრივ პროცესებს. კომპონენტთა

რაოდენობის მატებასთან ერთად, იზრდება სისტემაში მტყუნების საშიშროების ალბათობაც. ამ თვალსაზრისით, აუცილებელია სულ უფრო მეტად საიმედო ელემენტების წარმოება. მაშასადამე, რთული ტექნიკური სისტემების, მათ შორის კომპიუტერული სისტემების და ქსელების საიმედოობის ამაღლების პირდაპირი გზაა შემადგენელი კომპონენტების, მასში შემავალი მოწყობილობებისა და დამაკავშირებელი საშუალებების საიმედოობის ამაღლება. მაგრამ, იმის გამო, რომ ნებისმიერ ტექნოლოგიურ შესაძლებლობას გააჩნია ზღვარი, მტყუნებათა და შეფერხებათა შემთხვევითი მოვლენების აბსოლუტური აცილება შეუძლებელია. ამ დროს იყენებენ საიმედოობის ამაღლების სხვადასხვა მეთოდებს სიჭარბის გამოყენებით, რომლებიც სისტემაში ბუნებრივად არსებობენ ან ხელოვნურადაა დამატებული.

სიჭარბის სახეობებია: აპარატურული, დროითი და ინფორმაციული.

- აპარატურული სიჭარბე სისტემაში შეიტანება ელემენტების, მოწყობილობების ან სისტემის დარეზერვებით.
- დროითი რეზერვი ნიშნავს, რომ ამოცანის გადაწყვეტის დრო ნაკლებია იმ ოპერატიულ დროზე, როდესაც საჭიროა ამ ამოცანის შედეგის გამოყენება. დროის არსებული რეზერვის გამოყენება შეიძლება ამოცანის ან მისი დამახინჯებული ნაწილის (ეტაპის) ხელახალი გათვლისთვის.
- ინფორმაციული სიჭარბის გამოყენება შეიძლება იმ შემთხვევაში, თუ მონაცემებში არსებობს (ან ხელოვნურად შეტანილია) ისეთი ინფორმაცია, რომელიც უშუალოდ გათვლისათვის არ არის აუცილებელი, მაგრამ მუშავდება მონაცემთან ერთად და საშუალებას იძლევა გაკონტროლდეს ჩატარებული ოპერაციების შედეგები.

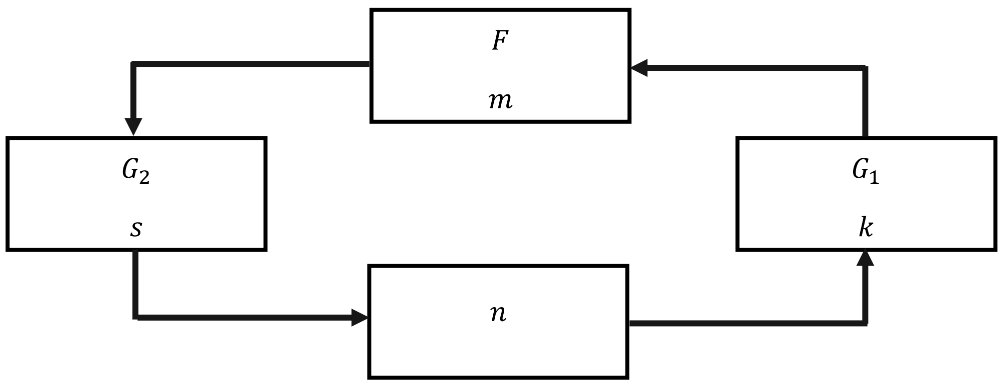
ცნობილია, რომ ანალიზური მოდელები რთული ტექნიკური სისტემებისთვის ძნელი გამოსაყენებელია, რადგანაც მკვეთრად იზრდება პარამეტრების რაოდენობა და უკიდურესად რთულდება მათემატიკური მოდელი, მაგრამ ადეკვატური მოდელების შემთხვევაში ისინი მკაფიო წარმოდგენას იძლევიან ობიექტის ძირითადი მახასიათებლების რაოდენობრივი და ხარისხობრივი ურთიერთდამოკიდებულებების შესახებ, მათ შორის საიმედოობის მახასიათებლების შესახებ. ამდენად, ანალიზური მოდელების გამოყენება მიზანშეწონილია ტექნიკური სისტემების პროექტირების საწყის სტადიაში, როდესაც განისაზღვრება ობიექტის სტრატეგიული პარამეტრები და თვისებები, ხდება ვარიანტების ურთიერთშედარება და შეჯერება.

მოყვანილია განზოგადებული გამოსახულებები იმის ალბათობისა, რომ მოცემული მოცულობის სამუშაო შესრულებული იქნება დაგეგმილ დროში მისი საიმედოობის გათვალისწინებით.

უკანასკნელ ათწლეულში რთული სისტემების დაპროექტების ორგანიზაციისა და მართვის სფეროში ხდება არსებითი ცვლილებები, რაც გავლენას ახდენს ამ სფეროში ჩართულ ყველა დაინტერესებულ მხარეზე: დამპროექტებელი ორგანიზაცია, სისტემის მფლობელი, კონტროლისა და მენეჯმენტის სამსახურები, მომხმარებელი. სწორედ ეს ცვლილებები მატებენ სიმწვავეს ხარისხისა და საიმედოობის უზრუნველყოფის პრობლემას. ამავე დროს, ისინი განაპირობებენ პრობლემის მოგვარების ახალი მიდგომების, მეთოდებისა და საშუალებების ძიებას, დაუფლებასა და გამოყენებას.

ჩვენ გამოვყოფთ რთულ სისტემებს შორის ტელე-კომუნიკაციურ სისტემებს და ქსელებს, აგრეთვე მათ ბაზაზე აგებულ მართვის ავტომატიზებულ სისტემებს, მათ შორის ტექნოლოგიური პროცესების მართვისა და ორგანიზაციულ-ადმინისტრაციული მართვის სისტემებს.

განვიხილავთ აპარატურული სიჭარბის მქონე ტერიტორიულად განაწილებულ სისტემას, სადაც სარეზერვო ელემენტები თან არ ახლავს ძირითად ელემენტებს. ამიტომ ჩვენს ამოცანაში პრობლემას წარმოადგენს როგორც რემონტის, ასევე ჩანაცვლების ოპერაციაც.



დიაგრამა N2.

ამ შემთხვევაში ბარლოუს დიაგრამას (იხ. დიაგრამა N1) ემატება ახალი ბლოკი, რომელიც ასახავს ჩანაცვლებას, როგორც ტექნიკური მომსახურების ოპერაციის გათვალისწინებას მათემატიკურ მოდელებში. ჩვენ განვიხილავთ ექსპონენტურად

განაწილებულ მტყუნების და აღდგენის ოპერაციას, ხოლო ჩანაცვლების ოპერაციის განაწილება ზოგადია.

მიზანი

სამაგისტრო ნაშრომის ძირითად მიზანს წარმოადგენს ტერიტორიულად განაწილებული სისტემების მათემატიკური მოდელების რიცხვითი ალგორითმის აგება და რეალიზება.

თეორიული და პრაქტიკული ღირებულება

ნაშრომის თეორიული ღირებულება იმაში მდგომარეობს, რომ რთული ტექნიკური ობიექტების ხარისხისა და საიმედოობის ამოცანის აღნიშნული ინტერპრეტაცია იძლევა საშუალებას, ასეთი ამოცანების გადასაწყვეტად ფართოდ იქნას გამოყენებული მართვის თეორიაში (ავტომატური, ავტომატიზებული, მათ შორის ორგანიზაციულ-ადმინისტრაციული მართვის სისტემების) არსებული მეთოდები.

გარდა ამისა, ნაშრომში გამოკვლეულია ტექნიკურ სისტემებში სტრუქტურული სიჭარბის შემოტანისა და ოპტიმიზაციის მოდელები ფართო კლასის ობიექტების დაპროექტების საჭიროებისათვის.

ნაშრომში გადმოცემული თეორიული შედეგები უშუალოდ იძლევა რეალური ტენიკური სისტემების ფუნქციობის ხარისხისა და საიმედოობის რაოდენობრივ დახასიათებას, რაც განსაკუთრებით მნიშვნელოვანია მათი დაპროექტების ეტაპზე რაციონალურ საპროექტო გადაწყვეტილებათა მისაღებად.

სამაგისტრო ნაშრომის პრაქტიკული ღირებულება მდგომარეობს მასში აგებული სასრულ-სხვაობიანი სქემებისა და ალგორითმების გამოყენების შესაძლებლობაში. ისინი დაყვანილი არიან პრაქტიკულად მისაღები გამოთვლების დონეზე. მიღებული შედეგები განკუთვნილია მტყუნება-აღდგენადი ტექნიკურ სისტემათა დასაპროექტებლად და საიმედოობის ასამაღლებლად ექსპლუატაციის პროცესში. მთლიანობაში, სამაგისტრო ნაშრომის შედეგები შეადგენს ტექნიკური სისტემების საიმედოობის ანალიზისა და შეფასების გამოსაყენებელი პროგრამების დამუშავებისათვის მეთოდურ იარაღს.

კვლევის მეთოდები

ნაშრომში გამოყენებულია სისტემური ანალიზისა და ოპერაციათა კვლევის ზოგადი მეთოდები, შემთხვევით პროცესთა გამოყენებითი თეორიის მეთოდები (საიმედოობის მათემატიკური თეორია, რიგების თეორია). არსებითი როლი ეთმობა კერძოწარმოებულებიანი და ინტეგრო-დიფერენციალური განტოლებათა სისტემებისა და არალოკალური ამოცანების სასრულ-სხვაობიანი მეთოდით აპროქსიმაციასა და რიცხვითი ამოხსნის საკითხებს. აგრეთვე გამოყენებულია წრფივ ალგებრულ განტოლებათა ამოხსნის კომპიუტერული და რიცხვითი ოპტიმიზაციის მეთოდები. ნაშრომში ასევე მნიშვნელოვანია პროგრამირებისა და ინფორმაციული ტექნოლოგიების საკითხების როლი.

ნაშრომის შინაარსი

დისერტაცია შედგება შესავალის, სამი თავისა და დანართისგან.

პირველ თავში ჩამოყალიბებულია ამოცანის შინაარსი დეტალურად. აღწერილია დარეზერვებული სისტემა, შემოტანილია ცვლადები და გაკეთებულია ამოცანის სქემური დიაგრამა.

ამის შემდეგ, განხილულია სხვადასხვა შემთხვევითი პროცესები და მათ საფუძველზე შედგენილია მათემატიკური მოდელი. მიღებულია სამი სისტემა და ჩატარებულია სისტემებში შემავალი განტოლებების გამოყვანა. ორი მათგანი არის პირველი რიგის ინტეგრო-დიფერენციალურ განტოლებათა სისტემა, ხოლო მესამე - პირველი რიგის კერძო-წარმოებულიანი დიფერენციალურ განტოლებათა სისტემა. მესამე სისტემისთვის ჩაწერილია ინტეგრალური სასაზღვრო პირობების სისტემა და მათთვისაც ჩატარებულია სისტემაში შემავალ განტოლებათა გამოყვანა.

მეორე თავი ეძღვნება სასრულ-სხვაობიანი მეთოდით მიახლოებითი ანალოგის აგებას და ალგორითმიზაციას.

ჩვენი მიზანია მე-3 სისტემის ამოხსნის რიცხვითი ალგორითმის შედგენა. ჩაწერილია მესამე სისტემისა და მისი სასაზღვრო პირობებისგან შემდგარი სისტემის სასრულ-სხვაობიანი ანალოგი. გამოსახულია უცნობები ორივე სისტემისთვის, მათ შორის სასაზღვრო პირობების სისტემაში შემავალი განტოლებები დაჯგუფებულია მათი ტიპების

მიხედვით. მიღებულია 7 განტოლება, რომელთაგან 5 არის მე-3 სისტემიდან, ხოლო დანარჩენი ორი - სასაზღვრო პირობების სისტემიდან.

დეტალურადაა აღწერილი ალგორითმი, თუ როგორი თანმიმდევრობით უნდა იქნას გამოყენებული მიღებული განტოლებები რიცხვითი შედეგების მისაღებად.

მესამე თავში მოცემულია რიცხვითი ექსპერიმენტების შედეგები და ანალიზი. მოცემულია რიცხვითი ექსპერიმენტების შედეგები ცხრილებისა და გრაფიკების სახით, შედარებულია ზუსტ მნიშვნელობები და რიცხვითი ექსპერიმენტით მიღებული შედეგები. კერძოდ, ორ ტესტურ მაგალითზე ნაჩვენებია აგებული ალგორითმის ეფექტურობა. ამ ყველაფრის საფუძველზე გაკეთებულია დასკვნა.

დანართში მოყვანილია მეორე თავში მიღებული მათემატიკური ალგორითმის შესაბამისი კოდი. პროგრამა დაწერილია ვიზუალ სტუდიოში C# ენაზე.

თავი 1.

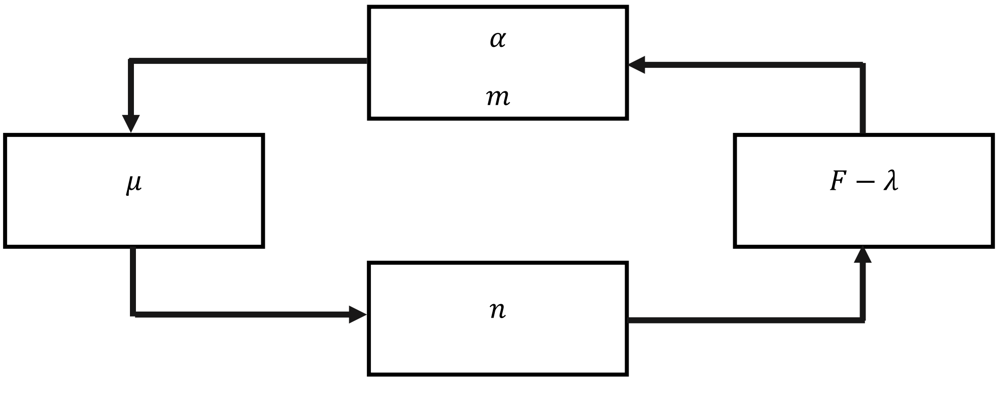
ერთი დარეზერვებული სისტემის დიფერენციალური და ინტეგრო-დიფერენციალური მოდელი

1.1. ამოცანის დასმა

დარეზერვებული სისტემა შედგება m რაოდენობის ძირითადი და n რაოდენობის სარეზერვო ელემენტისგან. ძირითადი ელემენტის მტყუნება ხდება α ინტენსივობით. სარეზერვო ელემენტი არ მტყუნდება, $0 \leq \alpha$. მტყუნებული ძირითადი ელემენტი იცვლება ქმედუნარიანი სარეზერვო ელემენტით ამ უკანასკნელით ჩანაცვლების შემდეგ. თუ სისტემაში არ არის ქმედუნარიანი სარეზერვო ელემენტი, მაშინ ჩანაცვლება ხდება პირველივე აღდგენის დასრულების შემდეგ.

გვაქვს ერთი ჩანაცვლების ორგანო. ჩანაცვლების დროის განაწილების ფუნქცია ზოგადია - $F(t)$. თუ ჩანაცვლების ორგანო დაკავებულია, ელემენტი დგება ჩანაცვლების რიგში. ჩანაცვლების დროს ელემენტთა მტყუნება არ ხდება. მტყუნებული ელემენტი გადაეცემა აღსადგენად. გვაქვს ერთი აღმდგენი ორგანო. აღდგენა ხდება პრინციპით: პირველად მტყუნებული აღდგება პირველი. თუ აღმდგენი ორგანო დაკავებულია, მტყუნებული ელემენტი დგება აღდგენის რიგში.

აღდგენის დროის განაწილების ფუნქცია მაჩვენებლიანია μ პარამეტრით. აღდგენილი ელემენტი აღიდგენს ყველა თავის პირვანდელ თვისებას და დგება რეზერვში, ან ჩანაცვლების რიგში, თუ ძირითად ელემენტთა სიმრავლის შესავსებად საჭიროა მისი დამატება. ამგვარად, სისტემა წარმოადგენს მასობრივი მომსახურების მრავალმხრიან სისტემას - ჩანაცვლებისა და აღდგენათა მომსახურების სისტემას, რომელშიც შემოდის ორი ტიპის განაცხადები.



დიაგრამა N3.

1.2. მათემატიკური მოდელი.

განვიხილოთ შემთხვევითი პროცესები:

- $k(t)$ - ელემენტთა რაოდენობა, რომლებიც აკლია ძირითად ელემენტთა ჯგუფს t მომენტში;
- $l(t)$ - მტყუნებულ ელემენტთა რაოდენობა t მომენტში;
- $\theta(t)$ - ჩანაცვლების პროცესის დაწყებიდან გასული დრო t მომენტისთვის;

შემდეგში $\lambda(u)$ აღნიშნავს ჩანაცვლების ინტენსივობას

$$\lambda(u) = \frac{f(u)}{1 - F(u)},$$

სადაც

$$f(u) = F'(u).$$

შემოვიტანოთ აღნიშვნები:

- $P_l(t) = P\{k(t) = 0,\ l(t) = l\}, \qquad l = \overline{0, n};$
- $R_k(t) = P\{k(t) = k,\ l(t) = n + k\}, \qquad k = \overline{0, m};$
- $q_{k,l}(t, u) = \lim\limits_{h \to 0} \left(\frac{1}{h} P\{k(t) = k,\ l(t) = l - 1,\ u < \theta(t) < u + h\}\right),$

$$k = \overline{1, m}\,; \quad l = \overline{1, n + k};$$

სისტემა 1. $P_l(t)$ სიდიდეები აკმაყოფილებენ შემდეგ ინტეგრო-დიფერენციალურ განტოლებათა სისტემას:

1. $\frac{dP_0(t)}{dt} = -m\alpha P_0(t) + \mu P_1(t) + \int_0^t q_{1,1}(t, u)\, \lambda(u) du,$
2. $\frac{dP_l(t)}{dt} = -(m\alpha + \mu) P_l(t) + \mu P_{l+1}(t) + \int_0^t q_{1,l+1}(t, u)\, \lambda(u) du, \qquad 1 \le l < n;$
3. $\frac{dP_n(t)}{dt} = -(m\alpha + \mu) P_n(t) + \int_0^t q_{1,n+1}(t, u)\, \lambda(u) du.$

განვიხილოთ დროის უსასრულოდ მცირე ინტერვალი $(t, t + h)$ და დავაკვირდეთ მდგომარეობის ცვლილებას.

ჩავატაროთ სისტემის პირველი განტოლების გამოყვანა. განვიხილოთ შემთხვევითი პროცესი $\{i(t+h)=0, j(t+h)=0\}$. ეს ნიშნავს, რომ დროის $t+h$ მომენტში ძირითად ელემენტთა სისტემას არ აკლია ელემენტი და ასევე, არც ერთი ელემენტი არაა დაზიანებული. მოვიყვანოთ ის შესაძლო შემთხვევითი პროცესები t მომენტში, რომელთა ალბათობებიც $O(h)$ რიგისაა:

- ვთქვათ დროის t მომენტში იყო შემთხვევითი პროცესი ალბათობით $P\{i(t)=0, j(t)=0\}$. ეს გულისხმობს, რომ დროის h ინტერვალში არ დაზიანებულა არც ერთი ელემენტი. ამ შემთხვევითი პროცესის ალბათობაა
$$(1-m\alpha h)P_0(t).$$
- ვთქვათ დროის t მომენტში ძირითად ელემენტთა სისტემას არ აკლდა ელემენტი და სისტემაში მტყუნებული იყო ერთი ელემენტი. დროის h ინტერვალში ეს ერთი ელემენტი აღდგა. ამ შემთხვევითი პროცესის ალბათობაა
$$\mu h P_1(t).$$
- ვთქვათ დროის t მომენტში ძირითად ელემენტთა სისტემას აკლდა ერთი ელემენტი, არც ერთი ელემენტი არ იყო დაზიანებული. მიმდინარეობდა ჩანაცვლების პროცესი და ამ პროცესის დაწყებიდან გასული დრო იყო დროის $(u-h, u)$ ინტერვალში. u-ს ყველა მნიშვნელობის გათვალისწინებით, ალბათობა ამ შემთხვევითი პროცესისა არის
$$h\int_0^t q_{1,1}(t,u)\lambda(u)\,du.$$
- სხვა ყველა შემთხვევითი პროცესის ალბათობის რიგი $O(h)$-ზე მაღალია.

ამ ალბათობების შეკრებით მივიღებთ:

$$P_0(t+h)=(1-m\alpha h)P_0(t)+\mu h P_1(t)+h\int_0^t q_{1,1}(t,u)\lambda(u)\,du+O(h),$$

რაც ექვივალენტურია შემდეგის:

$$\frac{P_0(t+h)-P_0(t)}{h}=-m\alpha P_0(t)+\mu P_1(t)+\int_0^t q_{1,1}(t,u)\lambda(u)\,du+\frac{O(h)}{h}.$$

როდესაც $h\to 0$, მივიღებთ საძიებელ განტოლებას.

სისტემა 2. $R_i(t)$ სიდიდეები აკმაყოფილებენ შემდეგ ინტეგრო-დიფერენციალურ განტოლებათა სისტემას:

1. $\frac{dR_1(t)}{dt} = -\big((m-1)\alpha + \mu\big)R_1(t) + m\alpha P_n(t) + \int_0^t q_{2,n+2}(t,u)\,\lambda(u)du,$
2. $\frac{dR_k(t)}{dt} = -\big((m-i)\alpha + \mu\big)R_k(t) + \big(m-(i-1)\big)\alpha R_{k-1}(t) + \int_0^t q_{k+1,n+k+1}(t,u)\,\lambda(u)du,$

$1 < k < m;$

3. $\frac{dR_m(t)}{dt} = -\mu R_m(t) + \alpha R_{m-1}(t).$

ამ სისტემის განტოლებების გამოყვანა პირველი სისტემის გამოყვანის ანალოგიური მსჯელობით ხდება.

სისტემა 3. $q_{k,l}(t,u)$ ფუნქციები აკმაყოფილებენ შემდეგ კერძოწარმოებულიან დიფერენციალურ განტოლებათა სისტემას:

1. $\frac{\partial q_{1,l}(t,u)}{\partial t} + \frac{\partial q_{1,l}(t,u)}{\partial u} = -\big((m-1)\alpha + \mu + \lambda(u)\big)q_{1,l}(t,u) + \mu q_{1,l+1}(t,u), \qquad 1 < l \le n;$
2. $\frac{\partial q_{1,n+1}(t,u)}{\partial t} + \frac{\partial q_{1,n+1}(t,u)}{\partial u} = -\big((m-1)\alpha + \mu + \lambda(u)\big)q_{1,n+1}(t,u),$
3. $\frac{\partial q_{k,1}(t,u)}{\partial t} + \frac{\partial q_{k,1}(t,u)}{\partial u} = -\big((m-i)\alpha + \lambda(u)\big)q_{k,1}(t,u) + \mu q_{k,2}(t,u), \qquad 1 \le k \le m;$
4. $\frac{\partial q_{k,l}(t,u)}{\partial t} + \frac{\partial q_{k,l}(t,u)}{\partial u} = -\big((m-i)\alpha + \mu + \lambda(u)\big)q_{k,l}(t,u) +$

$$+\big(m-(i-1)\big)\alpha q_{k-1,l-1}(t,u) + \mu q_{k,l+1}(t,u),$$

$$1 < k \le m; \qquad 1 < l < n+k;$$

5. $\frac{\partial q_{k,n+k}(t,u)}{\partial t} + \frac{\partial q_{k,n+k}(t,u)}{\partial u} = -\big((m-i)\alpha + \mu + \lambda(u)\big)q_{k,n+k}(t,u)$

$$+\big(m-(i-1)\big)\alpha q_{k-1,n+k-1}(t,u), \qquad 1 < k \le m;$$

მაგალითისთვის ჩავატაროთ სისტემის ბოლო მე-5 განტოლების გამოყვანა. განვიხილოთ ალბათობა

$$P\{k(t+h) = k, l(t+h) = n+k-1, u < \theta(t+h) < u+h\}.$$

ეს ალბათობა აღწერს სიტუაციას $t+h$ მომენტში, როცა ძირითად ელემენტთა სისტემას აკლია k რაოდენობის ელემენტი, მტყუნებულია $n+k-1$ ცალი ელემენტი. სისტემაში არის მხოლოდ ერთი ქმედუნარიანი თავისუფალი ელემენტი. ეს ელემენტი ჩანაცვლების

პროცესშია და ჩანაცვლების პროცესის დაწყებიდან გასული დრო არის $(u, u+h)$ ინტერვალში. t მომენტში შესაძლოა ყოფილიყო ქვემოთმოყვანილი ხდომილობები:

- ვთქვათ დროის t მომენტში იყო ხდომილობა ალბათობით
$$P\{k(t)=k, l(t)=n+k-1, u-h<\theta(t)<u\}.$$
ეს ნიშნავს, რომ ძირითად ელემენტთა სისტემას აკლია $k-1$ ცალი ელემენტი, დაზიანებულ ელემენტთა რიცხვია $n+k-2$, მიმდინარეობს ჩანაცვლების პროცესი და ამ პროცესის დაწყებიდან გასული დრო არის $(u-h, u)$ ინტერვალში; h დროის განმავლობაში არც ერთი ელემენტი არ გამტყუნდა და ჩანაცვლების პროცესიც არ დასრულებულა. ამ ხდომილობის ალბათობა არის
$$\big(1-\big((m-k)\alpha+\mu+\lambda(u-h)\big)h\big)q_{k,n+k}(t, u-h)h;$$
- ვთქვათ დროის t მომენტში შესაძლოა ყოფილიყო ხდომილება:
$$P\{i(t)=k-1, j(t)=n+k-2, u-h<\theta(t)<u\},$$
ანუ, ძირითად ელემენტთა სისტემას აკლდა $k-1$ ცალი ელემენტი, დაზიანებულ ელემენტთა რიცხვია $n+k-2$, მიმდინარეობს ჩანაცვლების პროცესი და ამ პროცესის დაწყებიდან გასული დრო არის $(u-h, u)$ ინტერვალში; h დროის განმავლობაში ერთი ელემენტი გამტყუნდა და ჩანაცვლების პროცესი არ დასრულებულა. ამ ხდომილების ალბათობაა
$$\big(m-(k-1)\big)\alpha h q_{k-1,n+k-1}(t, u-h);$$
- ყველა დანარჩენი ხდომილების ალბათობის რიგი $O(h)$ რიგისაა.

ამის შემდეგ, ამ ალბათობების შეკრებით მივიღებთ:

$$q_{k,n+k}(t+h, u) =$$
$$= \big(1-\big((m-k)\alpha+\mu+\lambda(u-h)\big)h\big)q_{k,n+k}(t, u-h)$$
$$+\big(m-(k-1)\big)\alpha h q_{k-1,n+k-1}(t, u-h)+O(h);$$

რაც ექვივალენტურია შემდეგის:

$$\frac{q_{k,n+k}(t+h,u)-q_{k,n+k}(t,u)}{h}+\frac{q_{k,n+k}(t+h,u)-q_{k,n+k}(t,u)}{u}=$$
$$-\big((m-k)\alpha+\mu+\lambda(u-h)\big)q_{k,n+k}(t, u-h)+$$
$$+\big(m-(k-1)\big)\alpha q_{k-1,n+k-1}(t, u-h)+\frac{O(h)}{h}.$$

როდესაც $h\to 0$, მივიღებთ გამოსაყვან განტოლებას.

სისტემის ყველა დანარჩენი განტოლება გამოიყვანება იდენტური მსჯელობით.

$q_{k,l}(t,u)$ ფუნქციებისათვის ბუნებრივია განვიხილოთ შემდეგი **სასაზღვრო პირობები:**

1. $q_{k,l}(t,0)=\int_0^t q_{k+1,l}(t,u)\,\lambda(u)du,$ $\qquad k=1;\ l=1;$
2. $q_{k,l}(t,0)=\int_0^t q_{k+1,l}(t,u)\,\lambda(u)du+m\alpha P_{l-2}(t),$ $\qquad k=1;\ 1<l<n+k;$
3. $q_{k,l}(t,0)=\int_0^t q_{k+1,l}(t,u)\,\lambda(u)du+m\alpha P_{l-2}(t)+\mu R_k(t),$ $\qquad k=1;\ l=n+k;$
4. $q_{k,l}(t,0)=\int_0^t q_{k+1,l}(t,u)\,\lambda(u)du,$ $\qquad 1<k<m;\quad 1\le l<n+k;$
5. $q_{k,l}(t,0)=\int_0^t q_{k+1,l}(t,u)\,\lambda(u)du+\mu R_k(t),$ $\qquad 1<k<m;\quad l=n+k;$
6. $q_{k,l}(t,0)=0,$ $\qquad k=m; 1<l<n+k;$
7. $q_{k,l}(t,0)=\mu R_k(t),$ $\qquad k=m; l=n+k;$

მაგალითისათვის გამოვიყვანოთ სისტემის მე-3 განტოლება. განვიხილოთ სიტუაცია $t+h$ მომენტში, როცა ძირითად ელემენტთა სისტემას აკლია 1 ელემენტი; დაზიანებულ ელემენტთა რიცხვი არის $n+k-1$; ერთადერთი თავისუფალი ქმედუნარიანი სარეზერვო ელემენტი ჩანაცვლების პროცესშია და ჩანაცვლების პროცესის დაწყებიდან გასული დრო არის $(0,h)$ დროის ინტერვალში; t მომენტში შესაძლოა ყოფილიყო შემდეგი ხდომილობები:

- ვთქვათ დროის t მომენტში ძირითად ელემენტთა სისტემას აკლდა 2 ელემენტი. ქმედუნარიანი სარეზერვო ელემენტების რიცხვი იყო 2. აქედან ერთ-ერთი ჩანაცვლების პროცესში იყო და $\theta(t)\in(u,u+h)$. ჩანაცვლების პროცესი დასრულდა h დროში. ამ ხდომილობის ალბათობა u-ს ყველა მნიშვნელობის გათვალისწინებით არის

$$h\int_0^t q_{k+1,l}(t,u)\lambda(u)\,du,\quad \text{როცა } k=1; l=n+k.$$

- ვთქვათ დროის t მომენტში ძირითად ელემენტთა სისტემას არაფერი აკლდა; დაზიანებული იყო $l-2$ რაოდენობა ელემენტები. h დროში გამტყუნდა ერთი ელემენტი და დაიწყო ჩანაცვლების პროცესი. ამ ხდომილობის ალბათობაა

$$m\alpha hP_{l-2}(t),\quad \text{როცა } k=1; l=n+k.$$

- ვთქვათ დროის t მომენტში ძირითად ელემენტთა სისტემას აკლდა 1 ელემენტი და ყველა ელემენტი, ძირითად ელემენტთა სისტემაში მდგომი ელემენტების გარდა,

იყო დაზიანებული. h დროის განმავლობაში აღდგა ერთი ელემენტი და დაიწყო მისი ჩანაცვლება. ამ ხდომილობის ალბათობაა

$$\mu h R_k(t), \quad \text{როცა } k = 1; l = n + k.$$

- სხვა ყველა ხდომილობის ალბათობა $O(h)$ რიგისაა.

ამ ყველაფრის შემდეგ, ალბათობების შეკრებით, მთელი გამოსახულების h-ზე შეფარდებით და ბოლოს, როცა $h \to 0$, მივიღებთ მე-3 განტოლებას.

სასაზღვრო განტოლებათა სისტემის დანარჩენი განტოლებებიც ანალოგიური მსჯელობით გამოიყვანება.

თავი 2.

სასრულ-სხვაობიანი სქემა და მისი ალგორითმიზაცია

ჩვენი განხილვის ძირითადი საგანია მე-3 სისტემა.

სასრულ-სხვაობიანი სქემების აგების ცნობილი მიდგომის გამოყენებით (იხ. მაგალითად, [16]) სიბრტყეზე შემოვიტანოთ ბადე და მოვახდინოთ მოცემული მე-3 სისტემის აპროქსიმაცია სასრულ-სხვაობიანი სქემებით, რომელთა აპროქსიმაციის ცდომილება ამოსახსნელი სისტემის საკმაოდ გლუვ ამონახსნზე ბადის ბიჯების მიმართ არის პირველი რიგის:

1. $$\frac{{q_{k,l}}_i^{j+1} - {q_{k,l}}_i^{j}}{\tau} + \frac{{q_{k,l}}_{i+1}^{j+1} - {q_{k,l}}_i^{j+1}}{h} = -r{q_{k,l}}_i^{j} + \mu {q_{k,l+1}}_i^{j} + {f_{k,l}}_i^{j},$$

სადაც $k = 1; 1 < l \leq n; r = \big((m-k)\alpha + \mu + \lambda(u)\big)$.

2. $$\frac{{q_{k,l}}_i^{j+1} - {q_{k,l}}_i^{j}}{\tau} + \frac{{q_{k,l}}_{i+1}^{j+1} - {q_{k,l}}_i^{j+1}}{h} = -r{q_{k,l}}_i^{j} + {f_{k,l}}_i^{j},$$

სადაც $k = 1; l = n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big)$.

3. $$\frac{{q_{k,l}}_i^{j+1} - {q_{k,l}}_i^{j}}{\tau} + \frac{{q_{k,l}}_{i+1}^{j+1} - {q_{k,l}}_i^{j+1}}{h} = -r{q_{k,l}}_i^{j} + \mu {q_{k,l+1}}_i^{j} + {f_{k,l}}_i^{j},$$

სადაც $1 \leq k \leq m; l = 1; r = \big((m-k)\alpha + \lambda(u)\big)$.

4. $$\frac{{q_{k,l}}_i^{j+1} - {q_{k,l}}_i^{j}}{\tau} + \frac{{q_{k,l}}_{i+1}^{j+1} - {q_{k,l}}_i^{j+1}}{h}$$
$$= -r{q_{k,l}}_i^{j} + \big(m-(k-1)\big)\alpha {q_{k-1,l-1}}_i^{j} + \mu {q_{k,l+1}}_i^{j} + {f_{k,l}}_i^{j},$$

სადაც $1 < k \leq m; 1 < l < n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big)$.

5. $$\frac{{q_{k,l}}_i^{j+1} - {q_{k,l}}_i^{j}}{\tau} + \frac{{q_{k,l}}_{i+1}^{j+1} - {q_{k,l}}_i^{j+1}}{h} = -r{q_{k,l}}_i^{j} + \big(m-(k-1)\big)\alpha {q_{k-1,l-1}}_i^{j} + {f_{k,l}}_i^{j},$$

სადაც $1 < k \leq m; l = n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big)$.

ამ სისტემის ყველა განტოლებაში უცნობითა გამოსახვით მიიღება შემდეგი დამოკიდებულებები:

1. $q_{k,l}{}_{i+1}^{j+1} = \left(\frac{h}{\tau} - hr\right) q_{k,l}{}_i^j + \left(1 - \frac{h}{\tau}\right) q_{k,l}{}_i^{j+1} + h\mu q_{k,l+1}{}_i^j + hf_{k,l}{}_i^j,$

სადაც $k = 1; 1 < l \leq n; r = \big((m-k)\alpha + \mu + \lambda(u)\big).$

2. $q_{k,l}{}_{i+1}^{j+1} = \left(\frac{h}{\tau} - hr\right) q_{k,l}{}_i^j + \left(1 - \frac{h}{\tau}\right) q_{k,l}{}_i^{j+1} + hf_{k,l}{}_i^j,$

სადაც $k = 1; l = n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big).$

3. $q_{k,l}{}_{i+1}^{j+1} = \left(\frac{h}{\tau} - hr\right) q_{k,l}{}_i^j + \left(1 - \frac{h}{\tau}\right) q_{k,l}{}_i^{j+1} + h\mu q_{k,l+1}{}_i^j + hf_{k,l}{}_i^j,$

სადაც $1 \leq k \leq m; l = 1; r = \big((m-k)\alpha + \lambda(u)\big).$

4. $q_{k,l}{}_{i+1}^{j+1} = \left(\frac{h}{\tau} - hr\right) q_{k,l}{}_i^j + \left(1 - \frac{h}{\tau}\right) q_{k,l}{}_i^{j+1} + h\big(m - (k-1)\big)\alpha q_{k-1,l-1}{}_i^j + h\mu q_{k,l+1}{}_i^j + hf_{k,l}{}_i^j,$

სადაც $1 < k \leq m; 1 < l < n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big).$

5. $q_{k,l}{}_{i+1}^{j+1} = \left(\frac{h}{\tau} - hr\right) q_{k,l}{}_i^j + \left(1 - \frac{h}{\tau}\right) q_{k,l}{}_i^{j+1} + h\big(m - (k-1)\big)\alpha q_{k-1,l-1}{}_i^j + hf_{k,l}{}_i^j,$

სადაც $1 < k \leq m; l = n + k; r = \big((m-k)\alpha + \mu + \lambda(u)\big).$

სასაზღვრო პირობები მათი დაჯგუფების შედეგად იღებენ ასეთ სახეს:

$$1.\ \ q_{k,l}(i,0) = \sum_{c=0}^{i-1} q_{k+1,l}(i,c) + b_{k,l}(i,0);$$

როცა $k < m$.

$$2.\ q_{k,l}(i,0) = b_{k,l}(i,0);$$

როცა $k = m$.

ჩატარებული მსჯელობის შედეგად, ჩვენ ვიღებთ 7 განტოლებას, რომელთაგან ბოლო ორი გვაძლევს საძიებელი ფუნქციების ურთიერთდამოკიდებულებას სასაზღვრო წერტილებში, ხოლო პირველი 5 - ყველა დანარჩენ წერტილში.

t და u ცვლადების შესაბამის ღერძებზე დანაყოფთა რაოდენობაა $p + p$. ანუ საძიებელ წერტილთა რაოდენობაა $p{\times}p$.

პირველად სრულდება ის განტოლება, რომლის დროსაც $k = m$ და $l = n + k$. როცა $i = 0$, $j = 0$. ამის შემდეგ, l იცვლება $(n + k)$-დან 1-მდე ყოველი k-თვის, როცა k იცვლება m-დან 1-მდე, ყოველი j-თვის, როცა j იცვლება 0-დან p-მდე, ყოველი i-თვის, როცა i იცვლება 0-დან p-მდე.

შედეგად, ჩვენ ვიღებთ ერთმანეთში ჩალაგებულ დამოუკიდებელ 4 ციკლს.

აღვნიშნოთ, რომ ინტეგრო-დიფერენციალური და არალოკალური ამოცანების გამოკვლევას და რიცხვით ამოხსნას ეძღვნება მრავალი ნაშრომი (იხ. მაგალითად, [2], [5] და მათში მოყვანილი ლიტერატურული მითითებები).

თავი 3.

რიცხვითი ექსპერიმენტები და მათი შედეგები

მეორე თავში აგებულ სასრულ-სხვაობიან სქემაზე დაყრდნობით ჩატარებულია მრავალი რიცხვითი ექსპერიმენტი. ჩვენ აქ მოვიყვანთ ორ მათგანს.

3.1. ექსპერიმენტი N1.

ტესტურ მაგალითად აღებულია შემდეგი ფუნქციები:

$$q_{k,l} = \frac{\sin(35t + 22u)}{k + l},$$

ხოლო შესაბამისი პარამეტრებია:

$$m = 3; \quad n = 2;$$

$$\alpha = 1; \quad \mu = 1; \quad \lambda = 1;$$

$$t = 100; \quad u = 0.001;$$

$$p = 2000.$$

ქვემოთ მოყვანილია $q_{3,5}$ ფუნქციის ზუსტი მნიშვნელობების ცხრილი:

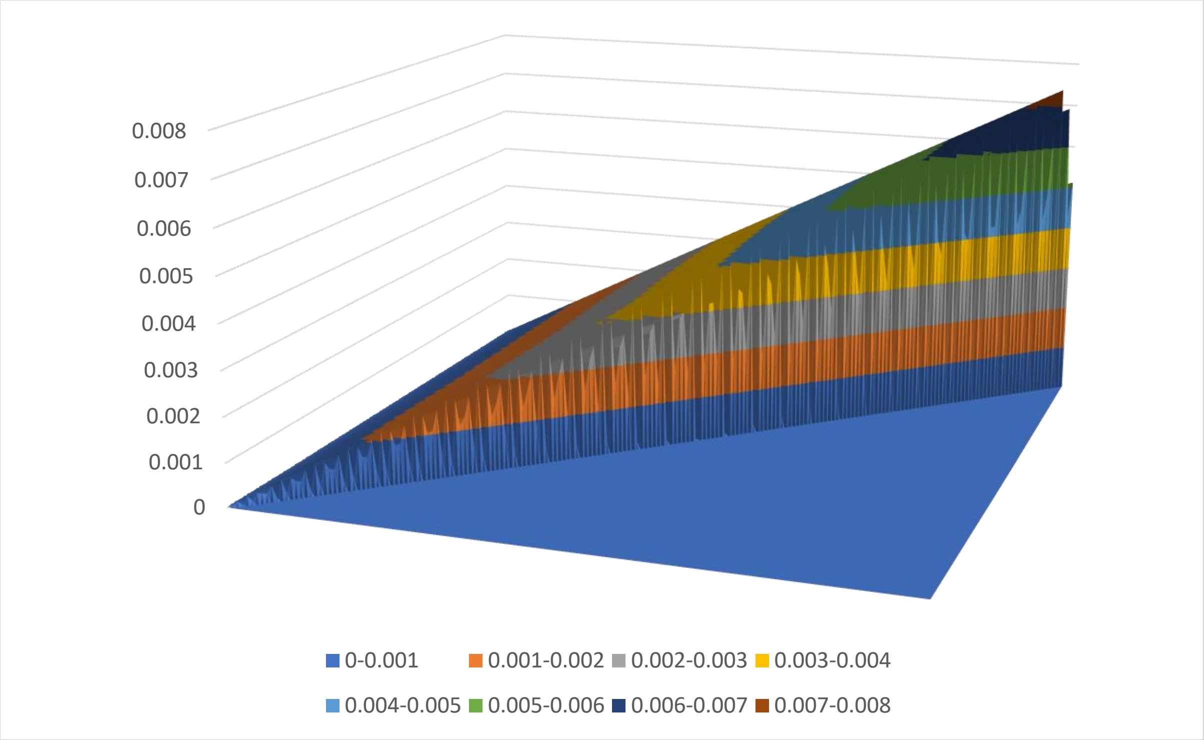
	200	400	600	800	1000	1200	1400	1600	1800	2000
200	-0.11994	0	0	0	0	0	0	0	0	0
300	-0.04359	0	0	0	0	0	0	0	0	0
400	0.067765	0.067534	0	0	0	0	0	0	0	0
500	0.124706	0.124686	0	0	0	0	0	0	0	0
600	0.081503	0.081712	0.081919	0	0	0	0	0	0	0
700	-0.02715	-0.02688	-0.02661	0	0	0	0	0	0	0
800	-0.114	-0.11389	-0.11377	-0.11366	0	0	0	0	0	0
900	-0.10931	-0.10944	-0.10957	-0.1097	0	0	0	0	0	0
1000	-0.01684	-0.01711	-0.01738	-0.01765	-0.01792	0	0	0	0	0
1100	0.089154	0.088961	0.088767	0.088574	0.088379	0	0	0	0	0
1200	0.12355	0.123591	0.123632	0.123672	0.123712	0.123751	0	0	0	0
1300	0.058731	0.058974	0.059216	0.059458	0.0597	0.059941	0	0	0	0
1400	-0.05325	-0.053	-0.05275	-0.0525	-0.05225	-0.052	-0.05175	0	0	0
1500	-0.12247	-0.12241	-0.12236	-0.1223	-0.12225	-0.12219	-0.12213	0	0	0
1600	-0.09334	-0.09353	-0.09371	-0.09389	-0.09407	-0.09425	-0.09443	-0.09461	0	0
1700	0.010742	0.010468	0.010194	0.00992	0.009646	0.009372	0.009097	0.008823	0	0
1800	0.1062	0.106055	0.105909	0.105763	0.105616	0.105469	0.105321	0.105172	0.105024	0
1900	0.116376	0.116476	0.116576	0.116675	0.116773	0.116871	0.116968	0.117065	0.117161	0
2000	0.033098	0.033364	0.033629	0.033893	0.034158	0.034422	0.034687	0.034951	0.035215	0.035478

ახლა კი მოვიყვანოთ ჩვენს მიერ აგებული ალგორითმით დათვლილი $q_{3,5}(i,j)$ ფუნქციის მიახლოებითი მნიშვნელობების ცხრილი.

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	-0.12015	0	0	0	0	0	0	0	0	0
300	-0.04313	0	0	0	0	0	0	0	0	0
400	0.06853	0.069062	0	0	0	0	0	0	0	0
500	0.125153	0.12558	0	0	0	0	0	0	0	0
600	0.081275	0.081255	0.081236	0	0	0	0	0	0	0
700	-0.02787	-0.02832	-0.02877	0	0	0	0	0	0	0
800	-0.11463	-0.11516	-0.11567	-0.11619	0	0	0	0	0	0
900	-0.10934	-0.10952	-0.10969	-0.10986	0	0	0	0	0	0
1000	-0.01625	-0.01593	-0.01562	-0.01531	-0.015	0	0	0	0	0
1100	0.089896	0.090443	0.090989	0.091532	0.092073	0	0	0	0	0
1200	0.123851	0.124193	0.124534	0.124874	0.125213	0.12555	0	0	0	0
1300	0.058348	0.058209	0.058071	0.057934	0.057797	0.057662	0	0	0	0
1400	-0.05401	-0.05452	-0.05503	-0.05553	-0.05603	-0.05653	-0.05703	0	0	0
1500	-0.123	-0.12346	-0.12393	-0.1244	-0.12486	-0.12532	-0.12578	0	0	0
1600	-0.09321	-0.09327	-0.09332	-0.09337	-0.09343	-0.09348	-0.09354	-0.09359	0	0
1700	0.011423	0.011828	0.012231	0.012632	0.013032	0.01343	0.013826	0.01422	0	0
1800	0.106886	0.107425	0.107963	0.108498	0.109031	0.109562	0.110091	0.110618	0.111143	0
1900	0.116515	0.116754	0.116992	0.11723	0.117467	0.117703	0.117939	0.118174	0.118408	0
2000	0.03258	0.032328	0.032077	0.031827	0.031579	0.031332	0.031086	0.030841	0.030597	0.030355

სხვაობა ტესტური მაგალითის ზუსტ მნიშვნელობებსა და რიცხვითი ექსპერიმენტით მიღებულ შედეგებს შორის მოცემულია შემდეგ ცხრილში:

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	0.000205	0	0	0	0	0	0	0	0	0
300	0.000467	0	0	0	0	0	0	0	0	0
400	0.000765	0.001528	0	0	0	0	0	0	0	0
500	0.000447	0.000893	0	0	0	0	0	0	0	0
600	0.000229	0.000456	0.000684	0	0	0	0	0	0	0
700	0.00072	0.001439	0.002156	0	0	0	0	0	0	0
800	0.000635	0.001269	0.001901	0.002532	0	0	0	0	0	0
900	3.84E-05	7.7E-05	0.000116	0.000155	0	0	0	0	0	0
1000	0.000589	0.001176	0.001762	0.002346	0.002928	0	0	0	0	0
1100	0.000742	0.001483	0.002221	0.002958	0.003694	0	0	0	0	0
1200	0.000301	0.000602	0.000902	0.001202	0.001501	0.001799	0	0	0	0
1300	0.000383	0.000764	0.001145	0.001524	0.001902	0.002279	0	0	0	0
1400	0.00076	0.001517	0.002273	0.003027	0.003779	0.00453	0.005279	0	0	0
1500	0.000525	0.001049	0.001572	0.002094	0.002615	0.003134	0.003653	0	0	0
1600	0.00013	0.000259	0.000387	0.000515	0.000643	0.00077	0.000896	0.001022	0	0
1700	0.000681	0.001359	0.002037	0.002712	0.003386	0.004058	0.004728	0.005397	0	0
1800	0.000686	0.00137	0.002053	0.002735	0.003415	0.004093	0.00477	0.005445	0.006119	0
1900	0.000139	0.000278	0.000416	0.000555	0.000694	0.000832	0.00097	0.001109	0.001247	0
2000	0.000519	0.001036	0.001552	0.002066	0.002579	0.003091	0.003601	0.00411	0.004617	0.005123



მაქსიმალური სხვაობა საძიებელი ფუნქციის მიახლოებით და ზუსტ მნიშვნელობებს შორის (i, j) წერტილებში შემდეგი რიცხვის ტოლია

0.00749821187702269.

3.2. ექსპერიმენტი N2.

ტესტურ მაგალითად აღებულია შემდეგი ფუნქციები:

$$q_{k,l} = \frac{\sin(35t + 22u)}{k + l},$$

ხოლო შესაბამისი პარამეტრებია:

$$m = 3; \quad n = 2;$$

$$\alpha = 1; \quad \mu = 1; \quad \lambda = 1;$$

$$t = 1; \quad u = 1;$$

$$p = 3000.$$

ქვემოთ მოყვანილია $q_{2,2}$ ფუნქციის ზუსტი მნიშვნელობების ცხრილი:

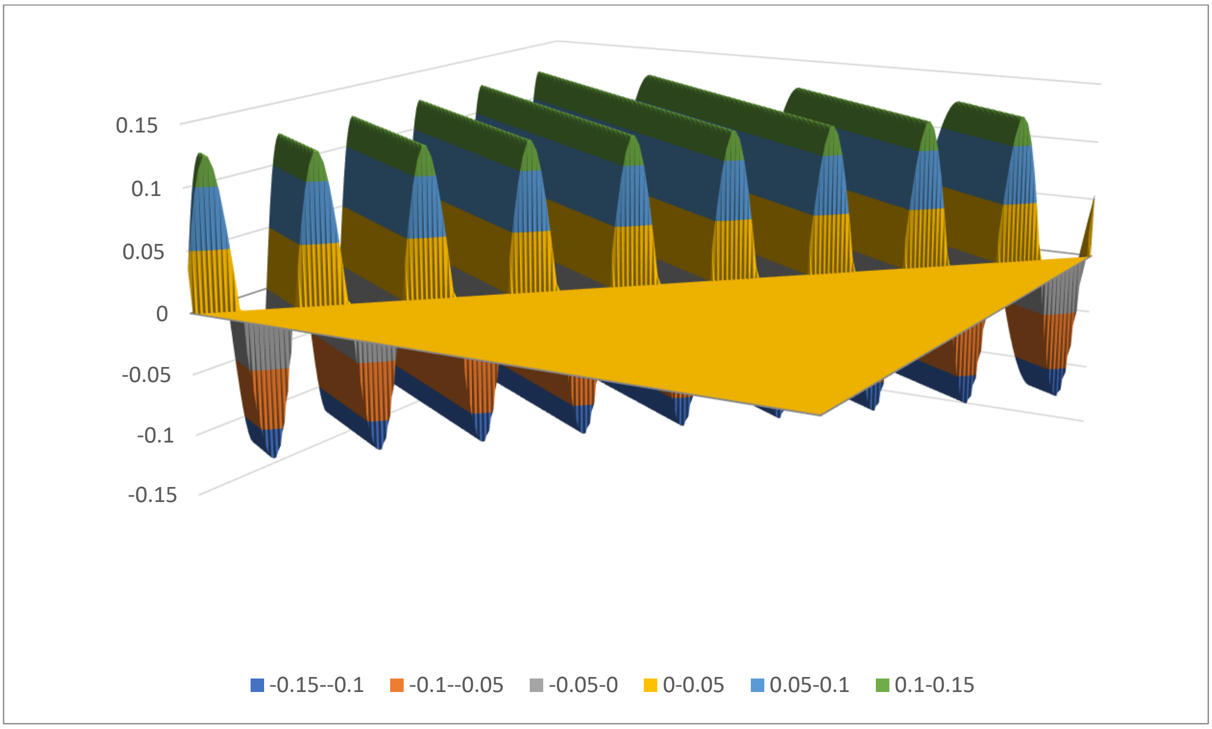
	200	400	600	800	1000	1200	1400	1600	1800	2000
200	-0.06884	0	0	0	0	0	0	0	0	0
300	0.114938	0	0	0	0	0	0	0	0	0
400	0.027861	-0.11492	0	0	0	0	0	0	0	0
500	-0.12487	0.068882	0	0	0	0	0	0	0	0
600	0.016654	0.09036	-0.12301	0	0	0	0	0	0	0
700	0.118933	-0.10109	5.55E-05	0	0	0	0	0	0	0
800	-0.05905	-0.05432	0.122988	-0.09044	0	0	0	0	0	0
900	-0.09788	0.12046	-0.0439	-0.06879	0	0	0	0	0	0
1000	0.093947	0.011378	-0.10734	0.11496	-0.02797	0	0	0	0	0
1100	0.06439	-0.12452	0.082165	0.027807	-0.11489	0	0	0	0	0
1200	-0.1169	0.033011	0.078047	-0.12487	0.068928	0.043744	0	0	0	0
1300	-0.02272	0.112748	-0.10999	0.016709	0.090322	-0.12302	0	0	0	0
1400	0.124999	-0.0732	-0.03884	0.118916	-0.10113	0.000111	0.100997	0	0	0
1500	-0.02185	-0.08665	0.123833	-0.0591	-0.05427	0.122978	-0.09048	0	0	0
1600	-0.11721	0.104095	-0.00531	-0.09785	0.120474	-0.04395	-0.06874	0.124863	0	0
1700	0.06363	0.049542	-0.12194	0.093983	0.011323	-0.10731	0.114981	-0.02802	0	0
1800	0.094528	-0.12176	0.04878	0.064343	-0.12451	0.082207	0.027753	-0.11487	0.107452	0
1900	-0.09733	-0.00614	0.104552	-0.11692	0.033065	0.078004	-0.12488	0.068975	0.043692	0
2000	-0.05983	0.123944	-0.08605	-0.02266	0.112724	-0.11001	0.016764	0.090283	-0.12303	0.054521

ახლა კი მოვიყვანოთ ჩვენს მიერ აგებული ალგორითმით დათვლილი $q_{2,2}(i,j)$ ფუნქციის მიახლოებითი მნიშვნელობების ცხრილი.

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	-0.06872	0	0	0	0	0	0	0	0	0
300	0.11383	0	0	0	0	0	0	0	0	0
400	0.028996	-0.11435	0	0	0	0	0	0	0	0
500	-0.12491	0.066937	0	0	0	0	0	0	0	0
600	0.016108	0.091967	-0.12184	0	0	0	0	0	0	0
700	0.118819	-0.10101	-0.00229	0	0	0	0	0	0	0
800	-0.05837	-0.05497	0.124538	-0.08874	0	0	0	0	0	0
900	-0.09785	0.119999	-0.04375	-0.07104	0	0	0	0	0	0
1000	0.092844	0.012363	-0.10768	0.116166	-0.02601	0	0	0	0	0
1100	0.065372	-0.12413	0.081364	0.027799	-0.11665	0	0	0	0	0
1200	-0.11693	0.03119	0.078898	-0.12467	0.069776	0.045578	0	0	0	0
1300	-0.02282	0.114078	-0.10914	0.015782	0.089879	-0.12407	0	0	0	0
1400	0.124168	-0.0732	-0.04088	0.119291	-0.10036	0.000803	0.102342	0	0	0
1500	-0.02061	-0.08648	0.125071	-0.05792	-0.055	0.121909	-0.09085	0	0	0
1600	-0.11753	0.102489	-0.00541	-0.09964	0.120222	-0.04282	-0.0679	0.125489	0	0
1700	0.063036	0.051389	-0.12123	0.094953	0.012543	-0.10753	0.113268	-0.02797	0	0
1800	0.094764	-0.12204	0.046671	0.063881	-0.12572	0.081408	0.028897	-0.11361	0.107344	0
1900	-0.09678	-0.00697	0.106429	-0.11559	0.033846	0.078905	-0.12441	0.066804	0.043809	0
2000	-0.06005	0.124011	-0.08618	-0.02486	0.111681	-0.11055	0.015696	0.091073	-0.12122	0.053881

სხვაობა რეალურ და ექსპერიმენტით მიღებულ შედეგებს შორის მოცემულია შემდეგ ცხრილში:

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	0.000117	0	0	0	0	0	0	0	0	0
300	0.001108	0	0	0	0	0	0	0	0	0
400	0.001135	0.000565	0	0	0	0	0	0	0	0
500	4.32E-05	0.001945	0	0	0	0	0	0	0	0
600	0.000546	0.001607	0.001166	0	0	0	0	0	0	0
700	0.000114	8.38E-05	0.002342	0	0	0	0	0	0	0
800	0.000687	0.000647	0.00155	0.001697	0	0	0	0	0	0
900	2.82E-05	0.00046	0.000151	0.002252	0	0	0	0	0	0
1000	0.001103	0.000985	0.000345	0.001207	0.001956	0	0	0	0	0
1100	0.000982	0.000384	0.000801	8.55E-06	0.001758	0	0	0	0	0
1200	2.49E-05	0.001821	0.000851	0.000202	0.000848	0.001834	0	0	0	0
1300	0.000102	0.001331	0.000846	0.000927	0.000442	0.001052	0	0	0	0
1400	0.000831	5.34E-06	0.002043	0.000375	0.000767	0.000692	0.001345	0	0	0
1500	0.001236	0.000171	0.001237	0.001183	0.000726	0.001069	0.000375	0	0	0
1600	0.000322	0.001606	0.000105	0.00179	0.000253	0.001129	0.000841	0.000627	0	0
1700	0.000594	0.001847	0.000714	0.000969	0.00122	0.000217	0.001714	5.59E-05	0	0
1800	0.000237	0.000282	0.002109	0.000461	0.001213	0.000799	0.001144	0.001263	0.000108	0
1900	0.000552	0.000837	0.001877	0.001333	0.000782	0.000901	0.000463	0.002171	0.000117	0
2000	0.000217	6.76E-05	0.000125	0.002202	0.001042	0.00054	0.001068	0.000789	0.001808	0.00064



მაქსიმალური სხვაობა საძიებელი ფუნქციის მიახლოებით და ზუსტ მნიშვნელობებს შორის (i, j) წერტილებში შემდეგი რიცხვის ტოლია

$$0.00541433345187875.$$

3.3. ექსპერიმენტი N3.

ტესტურ მაგალითად აღებულია შემდეგი ფუნქციები:

$$q_{k,l} = \frac{\sin(t+u)}{k+l},$$

ხოლო შესაბამისი პარამეტრებია:

$$m = 3; \quad n = 2;$$

$$\alpha = 1; \quad \mu = 1; \quad \lambda = 1;$$

$$t = 2; \quad u = 2;$$

$$p = 3000.$$

ქვემოთ მოყვანილია $q_{3,1}$ ფუნქციის ზუსტი მნიშვნელობების ცხრილი:

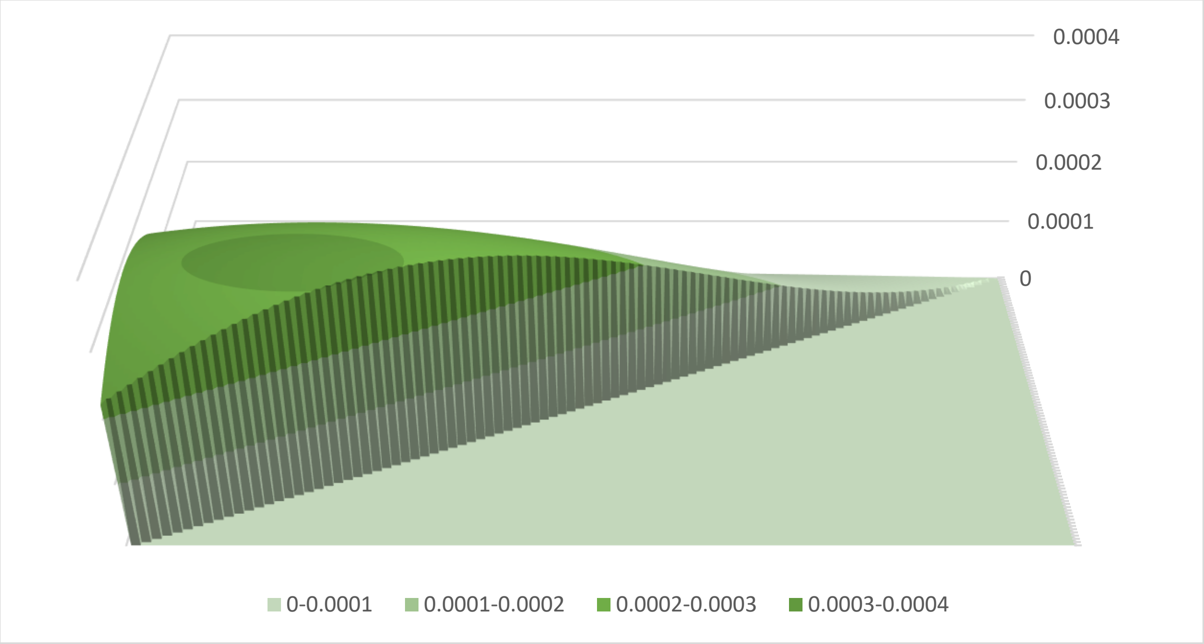
	200	400	600	800	1000	1200	1400	1600	1800	2000
200	0.097355	0	0	0	0	0	0	0	0	0
300	0.119856	0	0	0	0	0	0	0	0	0
400	0.141161	0.179339	0	0	0	0	0	0	0	0
500	0.161054	0.195832	0	0	0	0	0	0	0	0
600	0.179339	0.210368	0.23301	0	0	0	0	0	0	0
700	0.195832	0.222802	0.24089	0	0	0	0	0	0	0
800	0.210368	0.23301	0.246362	0.249893	0	0	0	0	0	0
900	0.222802	0.24089	0.249374	0.247916	0	0	0	0	0	0
1000	0.23301	0.246362	0.249893	0.243462	0.227324	0	0	0	0	0
1100	0.24089	0.249374	0.247916	0.236575	0.215802	0	0	0	0	0
1200	0.246362	0.249893	0.243462	0.227324	0.202124	0.168866	0	0	0	0
1300	0.249374	0.247916	0.236575	0.215802	0.186426	0.149618	0	0	0	0
1400	0.249893	0.243462	0.227324	0.202124	0.168866	0.128875	0.083747	0	0	0
1500	0.247916	0.236575	0.215802	0.186426	0.149618	0.106845	0.059812	0	0	0
1600	0.243462	0.227324	0.202124	0.168866	0.128875	0.083747	0.03528	-0.01459	0	0
1700	0.236575	0.215802	0.186426	0.149618	0.106845	0.059812	0.010395	-0.03944	0	0
1800	0.227324	0.202124	0.168866	0.128875	0.083747	0.03528	-0.01459	-0.06389	-0.11063	0
1900	0.215802	0.186426	0.149618	0.106845	0.059812	0.010395	-0.03944	-0.0877	-0.13246	0
2000	0.202124	0.168866	0.128875	0.083747	0.03528	-0.01459	-0.06389	-0.11063	-0.15296	-0.1892

ახლა კი მოვიყვანოთ ჩვენს მიერ აგებული ალგორითმით დათვლილი $q_{3,1}(i,j)$ ფუნქციის მიახლოებითი მნიშვნელობების ცხრილი.

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	0.097368	0	0	0	0	0	0	0	0	0
300	0.119876	0	0	0	0	0	0	0	0	0
400	0.141187	0.179388	0	0	0	0	0	0	0	0
500	0.161086	0.195893	0	0	0	0	0	0	0	0
600	0.179377	0.210441	0.233113	0	0	0	0	0	0	0
700	0.195875	0.222886	0.241009	0	0	0	0	0	0	0
800	0.210416	0.233104	0.246496	0.250058	0	0	0	0	0	0
900	0.222855	0.240993	0.249521	0.248098	0	0	0	0	0	0
1000	0.233067	0.246474	0.250052	0.243659	0.227549	0	0	0	0	0
1100	0.24095	0.249492	0.248086	0.236786	0.216042	0	0	0	0	0
1200	0.246425	0.250017	0.24364	0.227546	0.202378	0.169138	0	0	0	0
1300	0.249439	0.248045	0.23676	0.216033	0.18669	0.149902	0	0	0	0
1400	0.24996	0.243594	0.227514	0.202362	0.169138	0.129168	0.084046	0	0	0
1500	0.247984	0.236709	0.215995	0.186668	0.149896	0.107144	0.060117	0	0	0
1600	0.24353	0.227458	0.202318	0.16911	0.129156	0.084049	0.035588	-0.01429	0	0
1700	0.236642	0.215936	0.18662	0.149861	0.107125	0.060114	0.010703	-0.03914	0	0
1800	0.22739	0.202255	0.169057	0.129116	0.084024	0.035579	-0.01429	-0.06359	-0.11036	0
1900	0.215867	0.186554	0.149804	0.10708	0.060084	0.010688	-0.03914	-0.08741	-0.13219	0
2000	0.202186	0.168989	0.129055	0.083974	0.035543	-0.01431	-0.0636	-0.11035	-0.15271	-0.18898

სხვაობა რეალურ და ექსპერიმენტით მიღებულ შედეგებს შორის მოცემულია შემდეგ ცხრილში:

	200	400	600	800	1000	1200	1400	1600	1800	2000
200	1.3E-05	0	0	0	0	0	0	0	0	0
300	1.96E-05	0	0	0	0	0	0	0	0	0
400	2.59E-05	4.94E-05	0	0	0	0	0	0	0	0
500	3.21E-05	6.16E-05	0	0	0	0	0	0	0	0
600	3.79E-05	7.32E-05	0.000103	0	0	0	0	0	0	0
700	4.33E-05	8.41E-05	0.000119	0	0	0	0	0	0	0
800	4.83E-05	9.41E-05	0.000134	0.000165	0	0	0	0	0	0
900	5.28E-05	0.000103	0.000147	0.000182	0	0	0	0	0	0
1000	5.67E-05	0.000111	0.000159	0.000197	0.000224	0	0	0	0	0
1100	6.02E-05	0.000118	0.000169	0.000211	0.00024	0	0	0	0	0
1200	6.3E-05	0.000124	0.000178	0.000222	0.000253	0.000272	0	0	0	0
1300	6.51E-05	0.000128	0.000185	0.000231	0.000264	0.000284	0	0	0	0
1400	6.67E-05	0.000132	0.00019	0.000238	0.000272	0.000293	0.000299	0	0	0
1500	6.75E-05	0.000134	0.000193	0.000242	0.000278	0.000299	0.000305	0	0	0
1600	6.77E-05	0.000134	0.000194	0.000244	0.00028	0.000302	0.000308	0.0003	0	0
1700	6.72E-05	0.000133	0.000193	0.000243	0.00028	0.000302	0.000308	0.000299	0	0
1800	6.61E-05	0.000131	0.000191	0.00024	0.000277	0.000299	0.000304	0.000295	0.000273	0
1900	6.43E-05	0.000128	0.000186	0.000235	0.000271	0.000293	0.000298	0.000289	0.000266	0
2000	6.18E-05	0.000123	0.00018	0.000227	0.000263	0.000284	0.000289	0.000279	0.000256	0.000222



მაქსიმალური სხვაობა საძიებელი ფუნქციის მიახლოებით და ზუსტ მნიშვნელობებს შორის (i,j) წერტილებში შემდეგი რიცხვის ტოლია

$$0.000308095269856791.$$

3.4. დასკვნა.

მრავალი ჩატარებული რიცხვითი ექსპერიმენტის შედეგებზე დაკვირვება გვაჩვენებს, რომ როცა $p \to \infty$, მაქსიმალური სხვაობა მიახლოებით და ზუსტ მნიშვნელობებს შორის მიისწრაფის 0-სკენ, როცა ბადის ბიჯებს ვამცირებთ. ალგორითმი, რომლითაც სამაგისტრო ნაშრომში მოცემული რიცხვითი ექსპერიმენტებია ჩატარებული, გადმოცემულია სამეცნიერო ნაშრომში [14]. შესაბამისი მოხსენება გაკეთებული იქნა ი. ვეკუას სახელობის გამოყენებითი მათემატიკის ინსტიტუტის სემინარის გაფართოებული სხდომების კერძოწარმოებულებიანი დიფერენციალური განტოლებების სექციაზე 2017 წლის 19 აპრილს.

სამაგისტრო ნაშრომში მოცემული ალგორითმი და მისი გამოყენებით მიღებადი რიცხვითი ექსპერიმენტების შედეგები ვფიქრობთ შეიძლება გათვალისწინებული და გამოყენებული იქნას კვლევით და საპროექტო ორგანიზაციებში რეალური პრაქტიკული ამოცანების გადაწყვეტისას. რათა დაპროექტების საწყის სტადიაში შეირჩეს ტექნიკური სისტემის საიმედოობისა და მასზე გადასაწყვეტი ამოცანების ისეთი მახასიათებლები, რომლებიც უზრუნველყოფენ შედეგების მაქსიმალურ საიმედოობას.

გამოყენებული ლიტერატურა

1. ჭყოიძე გ. ალბათური მოდელირება სტრუქტურული მართვისა და მენეჯმენტის ამოცანებში. საქართველოს ტექნიკური უნივერსიტეტი. თბილისი. 2012.
2. ჯანგველაძე თ. ჩვეულებრივი დიფერენციალური განტოლებების მიახლოებითი ამოხსნის მეთოდები. თბილისის უნივერსიტეტის გამომცემლობა. თბილისი 2005.
3. Barlow R.E., Proschan F. Mathematical Theory of Reliability. Wiley, New York. 1965.
4. Gnedenko B.V., A course in the theory of probability. Nauka, Moscow. 1965.
5. Jangveladze T., Kiguradze Z., Neta B. Numerical Solutions of Three Classes of Nonlinear Parabolic Integro-Differential Equations. Elsevier, Academic Press. 2016.
6. Kakubava R. Multi-line Markov Closed Queuing System for Two Maintenance Operations. RTA Journal. No. 1, (2010), 15-22.
7. Kakubava R. Reliability Model for Standby System with Replacement Delays of Failed Elements. Automatic Control and Computer Sciences. Vol.47, No. 2, © Allerton Press, Inc., (2013), 94-98.
8. Kakubava R., Khurodze R., Probabilistic Analysis of the Downtime of a Duplicated System with Recovery and Switching. Automation and Remote Control. Vol.61, No.9, Part 1. (Translation from Avtomatikai Telemekhanika, No. 9.) 2000.
9. Kakubava R., Svanidze N., The Semi Markovian Model for Economic Analysis of Standby. International Journal of Engineering, Science and Innovative Technology, Vol. 2, No. 6, (2013), 240-246.
10. Kakubava R., Sztrik J., Queuing Models with Two Types of Service: Applications for Dependability Planning of Complex System. In Proc. VII International Conference. MMR 2011 - Mathematical Methods in Reliability. Theory. Methods. Applications (Beijing, China), June 20-24, 2011.
11. Korolyuk V., Stochastic Models of Systems. Dordrecht: Kluwer Academic Publisher. 1999.
12. Korolyuk V., Limnios N., Stochastic Systems in Merging Phase Space. Singapore: World Scientific, Imperial College Press. 2005.
13. Kovalenko I., Pegg P., Kuznetsov N., Mathematical Theory of Reliability of Time Dependent Systems with Practical Applications. John Wiley & Sons. 1997.

14. Kurtanidze S., On One Redundant System and Its Numeral Solution., Rep. Enlarged Sessions of the Seminar of I. Vekua Institute of Applied Mathematics, 2017, V.31 (accepted).
15. Limnios N., Oprisan G., Semi-Markov Processes and Reliability, Boston: Birkhauser. 2001.
16. Samarskii A.A., The Theory of Difference Schemes, Nauka, Moscow, 1977 (in Russian).
17. Ushakov I., Reliability Theory: History & Current State in Bibiliographies. RTA Journal. No. 1, (2012), 8-35.
18. Ushakov I., Reliability: Past, Present, Future. RTA Journal. No. 1, (2006), 10-27.

დანართი.

ალგორითმი დაწერილია დღესდღეობით არსებულ ერთ-ერთ ყველაზე მძლავრ კომპიუტერულ პროგრამაში - C#.

დაწერილია კონსოლ პროგრამა, რომელსაც ჩაშენებული აქვს ფუნქცია, რომლის მეშვეობითაც ის შედეგებს ექსელის ფაილის სახით აბრუნებს.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Excel = Microsoft.Office.Interop.Excel;

namespace FD_1
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Enter M value");
      int m = int.Parse(Console.ReadLine());
      Console.WriteLine("Enter N Value");
      int n = int.Parse(Console.ReadLine());

      Console.WriteLine("Which Function Write To Excel?");
```

```
Console.WriteLine("k : ");
int ex1 = int.Parse(Console.ReadLine());
Console.WriteLine("l : ");
int ex2 = int.Parse(Console.ReadLine());
Console.WriteLine("Enter Answer Type");
Console.WriteLine("For Real Results Type - 1");
Console.WriteLine("For Calculated Results Type - 2");
Console.WriteLine("For Difference Type - 3");
int type = int.Parse(Console.ReadLine());

double alpha = 1;
double miu = 1;
double lambda = 1;
int p = 3000;
double t = 1;
double u = 1;
double tau = t / p;
double h = u / p;

double[,,,] realQs = RealQs(tau, h, m, n, p);
double[,,,] f = new double[m + 1, m + n + 1, p + 1, p + 1];

for (int k = 1; k < m + 1; k++)
{
    for (int l = k + n; l > 0; l--)
```

```
    {
        f = Fs(tau, h, k, l, m, n, p, alpha, miu, lambda, f);
    }
}
Console.WriteLine("Real Functions - Done!");


double[,,] bFs = new double[m + 1, n + m + 1, p + 1];
for (int k = m; k >= 1; k--)
{
    for (int l = n + k; l >= 1; l--)
    {
        bFs = BoundaryCoefs(tau, h, k, l, m, n, p, bFs, realQs);
    }
}
Console.WriteLine("Boundary Coeffitients - Done!");


double[,,,] qsij = new double[m + 1, n + m + 1, p + 1, p + 1];
for (int i = 0; i <= p; i++)
{
    for (int j = 0; j <= i; j++)
    {
        for (int k = m; k >= 1; k--)
        {
            for (int l = n + k; l >= 1; l--)
            {
```

```
                if (j == 0)
                {
                    qsij = Boundary(tau, h, k, l, m, n, i, p, alpha, miu, lambda, qsij, bFs);
                }
                else
                {
                    qsij = Qsij(tau, h, k, l, m, n, i, j, p, alpha, miu, lambda, qsij, f);
                }
            }
        }
    }
}
Console.WriteLine("Qsij - Done!");

double[,,,] dif = Dif(tau, h, m, n, p, realQs, qsij);
Console.WriteLine("Difference - Done!");



var excelapp = new Excel.Application();
excelapp.Visible = true;
excelapp.Workbooks.Add();
Excel._Worksheet workSheet = (Excel.Worksheet)excelapp.ActiveSheet;
double max = 0;
int counter = 1;
```

```
Console.WriteLine("Starting Excel Printing ...");
int co1 = 0;
int co2 = 0;
for (int i = 0; i <= p; i += p / 200)
{
   co1++;
   for (int j = 0; j <= i; j += p / 200)
   {
      co2++;
      if (max <= dif[ex1, ex2, i, j])
      {
         max = dif[ex1, ex2, i, j];
      }

      workSheet.Cells[co1, 1] = (decimal)i / p;
      workSheet.Cells[1, co2] = (decimal)j / p;
      if (type == 1)
      {
         workSheet.Cells[co1, co2] = realQs[ex1, ex2, i, j];
      }
      if (type == 2)
      {
         workSheet.Cells[co1, co2] = qsij[ex1, ex2, i, j];
      }
      if (type == 3)
```

```
            {
                workSheet.Cells[co1, co2] = dif[ex1, ex2, i, j];
            }
        }
        co2 = 0;
    }
    Console.WriteLine(max);
    Console.WriteLine();
}




static double Function(int i, double tau, int j, double h, int k, int l)
{
    double result = Math.Sin(35 * i * tau + 22 * j * h) / (k + l);
    return result;
}
static double Grad(int i, double tau, int j, double h, int k, int l)
{
    double result = 57 * Math.Cos(35 * i * tau + 22 * j * h) / (k + l);
    return result;
}
static double Integer(int i, double tau, int j, double h, int k, int l, double tk)
```

```
        {
            double result = (Math.Cos(35 * tk * tau) - Math.Cos(57 * tk * tau)) / (22 * (k + l));
            return result;
        }
        static double[,,,] Fs(double tau, double h, int k, int l, int m, int n, int p, double alpha, double miu,
double lambda, double[,,,] F)
        {
            var result = F;

            /// (1) ///
            if (k == 1 && l < n + 1 && l > 1)
            {
                double R = ((m - 1) * alpha + miu + lambda);
                for (int j = 0; j <= p; j++)
                {
                    for (int i = j; i <= p; i++)
                    {
                        result[k, l, i, j] = Grad(i, tau, j, h, k, l) + R * Function(i, tau, j, h, k, l) - miu * Function(i,
tau, j, h, k, l + 1);
                    }
                }
            }

            /// (2) ///
            if (k == 1 && l == n + 1)
```

```
{
    double R = ((m - 1) * alpha + miu + lambda);
    for (int j = 0; j <= p; j++)
    {
        for (int i = j; i <= p; i++)
        {
            result[k, l, i, j] = Grad(i, tau, j, h, k, l) + R * Function(i, tau, j, h, k, l);
            double check = result[k, l, i, j];
        }
    }
}


/// (3) ///
if (k >= 1 && k <= m && l == 1)
{
    double R = ((m - k) * alpha + lambda);
    for (int j = 0; j <= p; j++)
    {
        for (int i = j; i <= p; i++)
        {
            result[k, l, i, j] = Grad(i, tau, j, h, k, l) + R * Function(i, tau, j, h, k, l) - miu * Function(i,
tau, j, h, k, l + 1);
        }
    }
}
```

```
/// (4) ///
if (k > 1 && k <= m && l > 1 && l < n + k)
{
    double R = ((m - k) * alpha + miu + lambda);
    for (int j = 0; j <= p; j++)
    {
        for (int i = j; i <= p; i++)
        {
            result[k, l, i, j] = Grad(i, tau, j, h, k, l) + R * Function(i, tau, j, h, k, l) -
                (m - (k - 1)) * alpha * Function(i, tau, j, h, k - 1, l - 1) - miu * Function(i, tau, j, h, k, l +
1);
        }
    }
}

/// (5) ///
if (k > 1 && k <= m && l == n + k)
{
    double R = ((m - k) * alpha + miu + lambda);
    for (int j = 0; j <= p; j++)
    {
        for (int i = j; i <= p; i++)
        {
            result[k, l, i, j] = Grad(i, tau, j, h, k, l) + R * Function(i, tau, j, h, k, l) -
```

```
                (m - (k - 1)) * alpha * Function(i, tau, j, h, k - 1, l - 1);
            }
        }
    }
    return result;
}
static double[,,] BoundaryCoefs(double tau, double h, int k, int l, int m, int n, int p, double[,,]
coefs, double[,,,] realQs)
{
    var result = coefs;
    for (int i = 0; i <= p; i++)
    {
        if (k < m)
        {
            double c = realQs[k, l, i, 0];
            double d = Integer(i, tau, 0, 0, k + 1, l, i);
            result[k, l, i] = c - d;
        }
        if (k == m)
        {
            double a = realQs[k, l, i, 0];
            result[k, l, i] = a;
        }
    }
```

```
    return result;
}
static double[,,,] RealQs(double tau, double h, int m, int n, int p)
{
    double[,,,] result = new double[m + 1, m + n + 1, p + 1, p + 1];
    for (int k = m; k >= 1; k--)
    {
        for (int l = n + k; l >= 1; l--)
        {
            for (int j = 0; j <= p; j++)
            {
                for (int i = j; i <= p; i++)
                {
                    double a = Function(i, tau, j, h, k, l);
                    result[k, l, i, j] = a;
                }
            }
        }
    }
    return result;
}
static double[,,,] Qsij(double tau, double h, int k, int l, int m, int n, int i, int j, int p, double alpha,
double miu, double lambda, double[,,,] qsij, double[,,,] f)
{
    var result = qsij;
```

```
/// (1) ///
if (k == 1 && l < n + 1 && l > 1)
{
    double R = ((m - k) * alpha + miu + lambda);
    double a1 = result[k, l, i - 1, j - 1];
    double a2 = result[k, l, i, j - 1];
    double a3 = result[k, l + 1, i - 1, j - 1];
    double f1 = f[k, l, i - 1, j - 1];
    result[k, l, i, j] = (h / tau - R * h) * a1 + (1 - h / tau) * a2 + h * miu * a3 + h * f1;
}

/// (2) ///
if (k == 1 && l == n + k)
{
    double R = ((m - k) * alpha + miu + lambda);
    double a1 = result[k, l, i - 1, j - 1];
    double a2 = result[k, l, i, j - 1];
    double f1 = f[k, l, i - 1, j - 1];
    result[k, l, i, j] = (h / tau - R * h) * a1 + (1 - h / tau) * a2 + h * f1;
}

/// (3) ///
if (k >= 1 && k <= m && l == 1)
{
```

```
        double R = ((m - k) * alpha + lambda);

        double a1 = result[k, l, i - 1, j - 1];

        double a2 = result[k, l, i, j - 1];

        double a3 = result[k, l + 1, i - 1, j - 1];

        double f1 = f[k, l, i - 1, j - 1];

        result[k, l, i, j] = (h / tau - R * h) * a1 + (1 - h / tau) * a2 + h * miu * a3 + h * f1;

    }


    /// (4) ///

    if (k > 1 && k <= m && l > 1 && l < n + k)

    {

        double R = ((m - k) * alpha + miu + lambda);

        double a1 = result[k, l, i - 1, j - 1];

        double a2 = result[k, l, i, j - 1];

        double a3 = result[k - 1, l - 1, i - 1, j - 1];

        double a4 = result[k, l + 1, i - 1, j - 1];

        double f1 = f[k, l, i - 1, j - 1];

        result[k, l, i, j] = (h / tau - R * h) * a1 + (1 - h / tau) * a2 + h * (m - (k - 1)) * alpha * a3 + h *
miu * a4 + h * f1;

    }


    /// (5) ///

    if (k > 1 && k <= m && l == n + k)

    {

        double R = ((m - k) * alpha + miu + lambda);
```

```
            double a1 = result[k, l, i - 1, j - 1];

            double a2 = result[k, l, i, j - 1];

            double a3 = result[k - 1, l - 1, i - 1, j - 1];

            double f1 = f[k, l, i - 1, j - 1];

            result[k, l, i, j] = (h / tau - R * h) * a1 + (1 - h / tau) * a2 + h * (m - (k - 1)) * alpha * a3 + h *
f1;

        }

        return result;

    }

    static double[,,,] Boundary(double tau, double h, int k, int l, int m, int n, int i, int p, double alpha,
double miu, double lambda, double[,,,] qsij, double[,,] bFS)

    {

        var result = qsij;


        if (k < m && i == 0)

        {

            double a = bFS[k, l, i];

            qsij[k, l, i, 0] = a;

        }

        if (k < m && i != 0)

        {

            double sum = 0;

            for (int j = 0; j < i; j++)

            {

                sum += tau * qsij[k + 1, l, i, j];
```

```
        }
        double a = sum + bFS[k, l, i];
        qsij[k, l, i, 0] = a;
    }

    if (k == m)
    {
        double a = bFS[k, l, i];
        qsij[k, l, i, 0] = a;
    }

    return result;
}
static double[,,,] Dif(double tau, double h, int m, int n, int p, double[,,,] realQs, double[,,,] qsij)
{
    double[,,,] result = new double[m + 1, m + n + 1, p + 1, p + 1];
    for (int k = 1; k < m + 1; k++)
    {
        for (int l = k + n; l > 0; l--)
        {
            for (int j = 0; j <= p; j++)
            {
                for (int i = j; i <= p; i++)
                {
                    result[k, l, i, j] = Math.Abs(realQs[k, l, i, j] - qsij[k, l, i, j]);
```

```
                    }
                }
            }
        }
        return result;
    }
}
}
```