

საქართველოს ტექნიკური უნივერსიტეტი

ბესიკ კველია

გეომეტრიული აღწერების დაპროგრამების მეთოდების კვლევა
ატლასის ექსპერიმენტის სიმულაციისათვის

წარმოდგენილია

დოქტორის აკადემიური ხარისხის მოსაპოვებლად

სადოქტორო პროგრამა „ინფორმატიკა“

შიფრი 0401

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი, 0175, საქართველო

2021 წელი

საავტორო უფლება © 2021 წელი, ბესიკ კეკელია

საქართველოს ტექნიკური უნივერსიტეტი

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ჩვენ, ქვემოთ ხელისმომწერი ვადასტურებთ, რომ გავაცანით ბესიკ კეკელიას მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: „გეომეტრიული აღწერების დაპროგრამების მეთოდების კვლევა ატლასის ექსპერიმენტის სიმულაციისათვის“ და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის -----
----- საუნივერსიტეტო სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

-----, ----- 2021წელი

ხელმძღვანელი: პროფესორი ალექსანდრე შარმაზანაშვილი

რეცენზენტი: -----

რეცენზენტი: -----

საქართველოს ტექნიკური უნივერსიტეტი

2021

ავტორი: ბესიკ კეკელია

დასახელება: „გეომეტრიული აღწერების დაპროგრამების მეთოდების კვლევა ატლასის ექსპერიმენტის სიმულაციისათვის“

სადოქტორო პროგრამა: ინფორმატიკა

ხარისხი: დოქტორი

სხდომა ჩატარდა: -----

ინდივიდუალური პიროვნებების ან ინსტიტუტების მიერ შემომოყვანილი დასახელების დისერტაციის გაცნობის მიზნით მოთხოვნის შემთხვევაში მისი არაკომერციული მიზნებით კოპირებისა და გავრცელების უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსიტეტს.

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა ან სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო უფლებებით დაცულ მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა იმ მცირე ზომის ციტატებისა, რომლებიც მოითხოვენ მხოლოდ სპეციფიურ მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს პასუხისმგებლობას.

რეზიუმე

ბირთვული ფიზიკის ექსპერიმენტები, დღეისათვის მსოფლიოში მრავალ ქვეყანაში მიმდინარეობს. მათ ამბიციური მიზნები გააჩნიათ. როგორცაა: სიმსივნური დაავადების წინააღმდეგ ბრძოლა, დიდი აფეთქების თეორიის ექსპერიმენტალური კვლევა და ა.შ. მათ შორის გამორჩეულია: LHC (The Large Hadron Collider)-ის ექსპერიმენტი და მისი შემადგენელი ATLAS-ის ექსპერიმენტი. LHC-ის ექსპერიმენტი მიმდინარეობს CERN-ში, ქენევა, შვეიცარია. მისი ძირითადი მიზნებია:

1. სტანდარტული მოდელის ექსპერიმენტალური დადასტურება და აქედან გამომდინარე დიდი მასის მქონე Z ბოზონის და ჰიგსის ბოზონის აღმოჩენა
2. დიდი აფეთქების თეორიის ექსპერიმენტალური კვლევა
3. სუპერ სიმეტრიის დადასტურება

ზემოთ ჩამოთვლილი საკითხების კვლევა LHC-ის ექსპერიმენტში პროტონების დაჯახების გზით მიმდინარეობს.

ATLAS-ის ექსპერიმენტში ერთერთი მთავარი მიმართულებაა ექსპერიმენტის სიმულაცია. სიმულაციის მიზანია ის, რომ რიგი ექსპერიმენტები ჩატარდეს არა რეალურ დეტექტორებზე, არამედ მათ მათემატიკურ მოდელებზე. სიმულაცია საშუალებას იძლევა გამოკვლეულ იქნეს დეტექტორში არ არსებული, თეორიული მოვლენები. ATLAS-ის ექსპერიმენტის სიმულაცია დაფუძნებულია Geant4 მოდელირების პაკეტზე.

წელიწადში დაახლოებით 5-7 მილიარდი ხელოვნური მოვლენის ფორმირება ხდება. და მათ დასამუშავებლად Geant4-ს სჭირდება 77 მლნ საათზე მეტი. სწორედ ამიტომ ძალიან მნიშვნელოვანი პარამეტრია სიმულაციის წარმადობა. წარმადობაზე მნიშვნელოვან გავლენას ახდენს სიმულაციის პაკეტებში გამოყენებული ATLAS-ის დეტექტორის გეომეტრიული აღწერები, რომლებიც დაპროგრამების სხვადასხვა მეთოდებით არიან შექმნილი. თუ რა გავლენას ახდენს გეომეტრიული აღწერების დაპროგრამების მეთოდები სიმულაციის პროცესის წარმადობაზე შესწავლილი არ არის და მისი გამოკვლევა წარმოადგენს აქტუალურ ამოცანას.

I თავში განხილულია დღეისათვის მსოფლიოში მიმდინარე ბირთვული ფიზიკის სხვადასხვა ექსპერიმენტები. მათი მიზნები და ამოცანები. ასევე განხილულია ATLAS-ის ექსპერიმენტის პროგრამული უზრუნველყოფა და ფიზიკური პროცესების სიმულაციის პროცესი. განხილულია მოდელირების პაკეტი Geant4 და აქ მიმდინარე სხვადასხვა პროექტები, მათ შორის ATLAS-ის ექსპერიმენტის სიმულაციის ინფრასტრუქტურა. I თავის ბოლოს ჩამოყალიბებულია საკვლევი ამოცანა და კვლევის მეთოდური გეგმა

II თავი შეიცავს ინფორმაციას კვლევისთვის საჭირო სატესტო მაგალითების ფორმირების შესახებ. სატესტო მაგალითების ფორმირება რამდენიმე ეტაპად განხორციელდა ესენია:

1. ATLAS-ის დეტექტორის ტიპური გეომეტრიული აღწერების შერჩევა
2. სატესტო მაგალითებისთვის დაპროგრამების მეთოდების შერჩევა
3. შერჩეული დაპროგრამების მეთოდების ანალიზი
 - 3.1. მსგავსი ტოპოლოგიების გამორიცხვა
 - 3.2. ირაციონალური მეთოდების გამორიცხვა
 - 3.3. თეორიული მეთოდების გამორიცხვა

1-ლი ეტაპის დასრულების შემდეგ განხორციელდა 66 სატესტო მაგალითის ფორმირება, რომლებიც 3 კლასში წახნაგოვანი, ცილინდრული და შერეულ კლასში განაწილდა. მე-2 ეტაპის შემდეგ 66 მაგალითისთვის შერჩეულ იქნა 2713 დაპროგრამების მეთოდი. მე-3 ეტაპის შემდეგ კი კვლევისთვის ჩამოყალიბდა 42 სატესტო მაგალითი თავისი 416 დაპროგრამების მეთოდით. ამავე თავში შესულია მე-2 თავის დასკვნები.

III თავში შესულია საკვლევი ინფრასტრუქტურის განსაზღვრის საკითხები. ეს თავის თავში მოიცავს შემდეგ საკითხებს: სიმულაციის შესასვლელი ინფორმაციის განსაზღვრა და სიმულაციის პროცესის წარმადობის შეფასების კრიტერიუმების და CPU Time-ის ხმაურის კოეფიციენტის განსაზღვრა. 1-ლი ამოცანის შესრულების შედეგად განისაზღვრა სიმულაციის შესასვლელი ინფორმაცია:

1. მონტეკარლოს გენერატორების მიერ შექმნილი ხელოვნური მოვლენები
2. დეტექტორის გეომეტრიული აღწერები
3. Condition Tag

განხორციელდა სიმულაციის წარმადობის შეფასების კრიტერიუმების განსაზღვრა და ეს კრიტერიუმებია ოპერატიული მეხსიერების დატვირთვა, რომელიც იზომება კბ-ებში და CPU Time, რომელის იზომება მილი წამებში.

სერვერებზე, რომლებზეც სიმულაცია მიმდინარეობს, პარალელურად სხვა ამოცანების შესრულებასაც ემსახურება, აქედან გამომდინარე განისაზღვრა CPU Time-ის ხმაურის კოეფიციენტის მიხედვით საუკეთესო სერვერი. განხორციელდა 3 სხვადასხვა სერვერის გამოკვლევა, გამოკვლეულ იქნა lxplus703.cern.ch, lxplus723.cern.ch და lxplus782.cern.ch სერვერები. ტესტირებისთვის კი შერჩეულ იქნა lxplus703.cern.ch სერვერი, რომელმაც CPU Time-ის ხმაურის მიხედვით საუკეთესო შედეგი აჩვენა.

IV თავში მოცემულია ინფორმაცია სიმულაციის სატესტო სესიების შესახებ. სესიები ჩატარდა რამდენიმე ეტაპად. განხორციელდა ცილინდრული ობიექტების, წახნაგოვანი ობიექტების და შერეული ობიექტების ტესტირება. შედეგად განისაზღვრა დაპროგრამების მეთოდების რეიტინგი წარმადობის თვალსაზრისით. ასევე განხორციელდა

ობიექტების სხვადასხვა რაოდენობის ტესტირება, რომლის მიზანიც არის სიმულაციის წარმადობაზე ობიექტების რაოდენობის გავლენის გამოკვლევა. კვლევების შედეგად დადასტურდა, რომ დაპროგრამების მეთოდებს გავლენა აქვს სიმულაციის წარმადობაზე, ასევე ობიექტების რაოდენობის ცვლილება გავლენას ახდენს წარმადობაზე. თავის ბოლოს არის მოცემული მე-4 თავის დასკვნები.

V თავში განხილულია კვლევით მიღებული შედეგების აპრობაციის საკითხები. აპრობაცია განხორციელდა ATLAS-ის დეტექტორის ერთერთი ქვე-დეტექტორის NSW (New Small Wheel)-ზე. განხორციელდა NSW-ის გეომეტრიული აღწერის სხვადასხვა მეთოდით გადაწერა. შეიქმნა მისი კოდის ოპტიმალური და არა ოპტიმალური ვარიანტები. მათი კვლევის შედეგად დადგინდა, რომ მიღებული მეთოდების ეფექტურობა ოპერატიული მეხსიერების დატვირთვის მიხედვით შეადგენს 62.1%-ს, ხოლო CPU Time-ის მიხედვით კი 70.3%-ს.

დისერტაცია ასევე შეიცავს კვლევის შედეგად მიღებულ ძირითად დასკვნებს:

1. გამოიყო ატლასის დეტექტორის 66 ტიპური გეომეტრიული აღწერა
2. დამუშავდა გეომეტრიული აღწერების დაპროგრამების მეთოდების კლასიფიკაცია, რომელშიც გაერთიანდა - ცილინდრული კლასისთვის 29, წახნაგოვანი კლასისთვის 135 და კომბინირებული კლასისთვის 252 ტიპური დაპროგრამების მეთოდი
3. გამოიყო ატლასის დეტექტორის გეომეტრიული აღწერების დაპროგრამების ტიპური კლასები და მეთოდები
4. წახნაგოვანი ობიექტების დაპროგრამებისათვის დამტკიცდა, რომ პოლიგონ მეთოდები განაპირობებენ უკეთეს წარმადობას მყარი ტანის პრიმიტივების მეთოდებთან შედარებით, რაც ობიექტების რაოდენობის ზრდასთან ერთად აღწევს - CPU-თვის 8%-ს, ხოლო RAM-თვის 20%-ს
5. ცილინდრული ობიექტების დაპროგრამებისათვის დადგინდა, რომ მეთოდების ცვლილება განაპირობებს წარმადობის უმნიშვნელო ცვლილებას, რაც ობიექტების რაოდენობის ზრდასთან ერთად აღწევს 4%-ს
6. კომბინირებული ობიექტების დაპროგრამებისათვის პოლიგონ მეთოდებს უკეთესი წარმადობა აქვთ ვიდრე მყარი ტანის პრიმიტივების მეთოდებს და ეს სხვაობა პირდაპირპროპორციულად იზრდება ობიექტების რაოდენობასთან ერთად
7. შეიქმნა სიმულაციის წარმადობის შესწავლის საკვლევო ინფრასტრუქტურა
8. გამოვლენილ იქნა Tube მეთოდის ეფექტურობა ცილინდრული ობიექტების დაპროგრამების ამოცანებისათვის

9. გამოვლენილ იქნა Arbitrary_Polygon მეთოდის ეფექტურობა წახნაგოვანი ობიექტების დაპროგრამების ამოცანებისათვის
10. გამოვლინდა უმნიშვნელო სხვაობა Cube და Pyramid მეთოდებს შორის როგორც წახნაგოვანი, ასევე კომბინირებული ობიექტების დაპროგრამების ამოცანებისათვის
11. განხორციელდა დაპროგრამების მეთოდების რანჟირება. შეიქმნა დაპროგრამების მეთოდების რეიტინგი წარმადობის თვალსაზრისით

Abstract

Nuclear physics experiments are currently conducted in many countries around the world. They have ambitious goals. Such as: the fight against cancer, the study of the Big Bang theory, and so on. Among them are the LHC (The Large Hadron Collider) experiment and the ATLAS experiment. The LHC experiment is taking place at CERN, Geneva, Switzerland. Its main goals are:

1. Experimental validation of a standard model and hence the discovery of a large mass Z boson and a Higgs boson
2. Study of the Big Bang theory
3. Confirmation of super symmetry

The above issues are being researched by proton collisions in the LHC experiment.

Simulation is one of the main cornerstones for the success of ATLAS Experiment. The aim of the simulation is to conduct a series of experiments not on real detectors but on their mathematical models. The simulation allows the non-existent, theoretical events to be investigated. The ATLAS experiment simulation is based on the Geant4 modeling package.

Approximately 5-7 billion artificial events are produced annually and Geant4 needs more than 77 million hours to process them. This is why simulation performance is a very important parameter. Performance is significantly influenced by the geometric descriptions of the ATLAS detector used in the simulation packages, which are created by different programming methods. The influence of geometric description programming methods on the performance of the simulation process has not been studied and its investigation is an important task.

Chapter I discusses various experiments in nuclear physics in the world today. Their goals and objectives. Also discussed are the ATLAS experiment software and the physical process simulation process. The Geant4 modeling package and various ongoing projects are discussed here, including the ATLAS experiment simulation infrastructure. At the end of the chapter, the research task and the methodological plan are formulated.

Chapter II contains information on the formation of test samples required for research. The formation of test samples was carried out in several stages, as follows:

1. Selection of the typical geometric descriptions of the ATLAS detector
2. Selection of programming methods for test examples
3. Analysis of selected programming methods
 - 3.1. Exclude similar topologies
 - 3.2. Exclude irrational methods
 - 3.3. Exclude theoretical methods

After the completion of the 1st stage, 66 test samples were produced, which were divided into 3 classes: hollow, cylindrical and mixed. After step 2, 2713 programming methods were selected for 66 examples. After the third stage, 42 test samples were produced for the study with its 416 programming methods. The conclusions of Chapter 2 are included in the same chapter.

Chapter III covers the issues of defining research infrastructure. This includes the following topics: Determining input information for the simulation, determining the criteria for evaluating the performance of the simulation process and the CPU Time noise ratio. As a result of the of the 1st task, the input information of the simulation was determined:

1. Simulated events created by Monte Carlo generators
2. Geometric descriptions of the detector
3. Condition Tag

The simulation performance assessment criteria were defined and these criteria are RAM memory measured in KBs and CPU Time measured in milliseconds.

The servers that run simulation also work other tasks in parallel, hence the best server in terms of CPU Time noise ratio has been determined. 3 different servers were analyzed: lxplus703.cern.ch, lxplus723.cern.ch and lxplus782.cern.ch. The lxplus703.cern.ch server was selected for testing, because it showed the best result in terms of CPU Time noise.

Chapter IV provides information on simulation test sessions. The sessions were conducted in several stages. Testing was performed on cylindrical, polygon and mixed objects. As a result, ranking of programming methods in terms of performance was determined. Different number of objects were also tested to investigate the impact of the number of objects on the simulation performance. Studies have shown that programming methods have an impact on simulation performance, and that changing the number of objects has an impact on performance. At the end of the chapter are the conclusions of Chapter 4.

Chapter V discusses research findings and its approbation. Approbation was performed on one of the ATLAS detector sub-detectors NSW (New Small Wheel). NSW geometry was described using different programming methods. Optimal and non-optimal versions of its code were created. Their study found that the efficiency of the adopted methods was 62.1% according to the RAM load and 70.3% according to the CPU Time.

The dissertation also contains the main conclusions of the research:

1. 66 typical geometric descriptions of the Atlas Detector were selected

2. Developed classification of programming methods for geometric descriptions, which combined - typical programming methods for the cylindrical class 29, 135 for the concave class and 252 for the combined class
3. Identified Typical classes and methods for programming geometric descriptions of the Atlas Detector were identified
4. For programming polygonal objects, it has been proven that polygon methods provide better performance compared to solid-body primitive methods, which increases with the number of objects - 8% for CPU and 20% for RAM
5. For the programming of cylindrical objects, it was found that the change of methods causes a slight change in performance, which reaches 4% with the increase in the number of objects
6. Polygon methods for programming combined objects have better performance than solid-body primitive methods, and this difference increases in direct proportion to the number of objects
7. A simulation performance study infrastructure has been created
8. The effectiveness of the Tube method for the programming tasks of cylindrical objects was revealed
9. The effectiveness of the Arbitrary Polygon method for the programming of polygonal objects was revealed
10. Slight differences were found between the Cube and Pyramid methods for both polygonal and combined object programming tasks.
11. Programming methods were ranked by performance

სარჩევი

შესავალი	20
თავი I: ATLAS-ის ექსპერიმენტის კომპიუტინგი	24
1.1 ბირთვული ფიზიკის პროექტები	24
1.2 ATLAS-ის ექსპერიმენტის პროგრამული უზრუნველყოფა	27
1.3 ფიზიკური პროცესების სიმულაცია	30
1.4 ATLAS-ის სიმულაციის ინფრასტრუქტურა GEANT4-ის გაზაზხე	38
1.5 საკვლევი ამოცანის ჩამოყალიბება	53
თავი II: სატესტო მაგალითების ფორმირება	55
2.1 დეტექტორის ტიპური გეომეტრიების გამოყოფა	55
2.2 დაპროგრამების მეთოდების შერჩევა	56
2.3 დაპროგრამების მეთოდების ანალიზი	101
2.3.1 მსგავსი ტოპოლოგიების გამორიცხვა	102
2.3.2 ირაციონალური მეთოდების გამორიცხვა	107
2.3.3 თეორიული მეთოდების გამორიცხვა	107
2.4 მე-2 თავის დასკვნები	109
თავი III: საკვლევი ინფრასტრუქტურის განსაზღვრა	111
3.1 სიმულაციის შესასვლელი ინფორმაციის განსაზღვრა	111
3.2 სიმულაციის პროცესის წარმადობის შეფასების კრიტერიუმების და CPU-ს ხმაურის კოეფიციენტის მიხედვით ტესტირების სერვერის განსაზღვრა	118
3.3 მე-3 თავის დასკვნები	121
თავი IV: სიმულაციის სატესტო სესიები	122
4.1 ცილინდრული ობიექტების ტესტირება	122
4.1.1 ტესტირება: მაგალითი № 23	122
4.1.2 ტესტირება: მაგალითი № 28	124
4.1.3 ტესტირება: მაგალითი № 11	126
4.2 წახნაბოვანი ობიექტების ტესტირება	128
4.2.1 ტესტირება: მაგალითი № 03	132
4.2.2 ტესტირება: მაგალითი № 27	135
4.2.3 შერეული ობიექტების ტესტირება	137
4.2.4 ობიექტების რაოდენობის სიმულაციის წარმადობაზე გავლენის გამოკვლევა	141
4.3 მე-4 თავის დასკვნები	148
თავი V: კვლევის შედეგების აპრობაცია	151
5.1 კვლევის შედეგების აპრობაციის ეტაპები	151
5.2 ATLAS-ის დეტექტორის გეომეტრიული აღწერების კომპლექსურობის განსაზღვრა	152
5.3 სატესტო მაგალითის შერჩევა აპრობაციისთვის	153

5.4 NSW-ის სიმულაციის სატესტო სესიებისთვის კოდეზის ფორმირება	155
5.5 სიმულაციის სატესტო სესიები NSW-თვის	158
5.6 აპრობაციის შედეგების ანალიზი	160
5.7 მე-5 თავის დასკვნები	162
დისერტაციის დასკვნები:	163
ბამოყენებული ლიტერატურა:	165

სურათების ნუსხა

სურ. 1 LHC-ის ამაჩქარებელი	24
სურ. 2 LHC-ის ამაჩქარებელის სქემატური გეოლოგიური გეგმა	25
სურ. 3 ATLAS-ის დეტექტორი	26
სურ. 4 Athena-ს ფრეიმვორკი	28
სურ. 5 BABAR დეტექტორი	31
სურ. 6 BABAR ექსპერიმენტის სიმულაციის სქემა	31
სურ. 7 ALICE-ის დეტექტორი	32
სურ. 8 VMC კონცეფციის სქემა	33
სურ. 9 ატლასის სრული სიმულაციის სქემა	34
სურ. 10 ატლასის სრული სიმულაციის სქემა	34
სურ. 11 Calorimeter-ში დაფიქსირებული ჰიტები ერთ-ერთი მოვლენისთვის	35
სურ. 13 რეკონსტრუქცია	36
სურ. 12 დიგიტიზაცია	36
სურ. 14 ინტეგრირებული სიმულაციის ფრეიმვორკი (ISF)	37
სურ. 15 ISF-ის მუშაობის პრინციპი	38
სურ. 16 გეომეტრიული აღწერების სიმულაციის მიწოდების ჯაჭვი	39
სურ. 17 ATLAS-ის დეტექტორის აქტიური კომპონენტების მონაცემთა ბაზა	40
სურ. 18 აქტიური კომპონენტების მონაცემთა ბაზის სტრუქტურა	40
სურ. 19 HVS-ის კომპონენტები	41
სურ. 20 Leaf Node-ის კომპონენტები	41
სურ. 21 Tile Calorimeter-ის GeoModel-ის კოდის ფრაგმენტი	42
სურ. 22 Tile Calorimeter-ის ერთ-ერთი კომპონენტის GeoModel-ის კოდის ფრაგმენტი	42
სურ. 24 Tile Calorimeter-ის ერთ-ერთი კომპონენტის GeoModel-ის კოდის ფრაგმენტი	43
სურ. 23 TileGlobals -ის მონაცემთა ბაზის ფრაგმენტი	43
სურ. 25 Box პრიმიტივი GeoModel-ში	44
სურ. 26 GeoCons პრიმიტივი GeoModel-ში	45
სურ. 27 GeoPcon პრიმიტივი GeoModel-ში	45
სურ. 28 GeoTrap პრიმიტივი GeoModel-ში	45
სურ. 29 GeoTube პრიმიტივი GeoModel-ში	46
სურ. 30 GeoTubes პრიმიტივი GeoModel-ში	46
სურ. 31 Logical Volume-ის სტრუქტურა GeoModel-ში	46
სურ. 32 Physical Volume -ის სტრუქტურა GeoModel-ში	47
სურ. 33 დაპროგრამების მეთოდი cube	47
სურ. 34 დაპროგრამების მეთოდი Tube	47
სურ. 36 დაპროგრამების მეთოდი Cylinder	48
სურ. 35 დაპროგრამების მეთოდი Pyramid	48
სურ. 37 დაპროგრამების მეთოდი Cylinder	49
სურ. 38 დაპროგრამების მეთოდი arbitrary polygon	50
სურ. 39 დაპროგრამების მეთოდი Symmetric polygon	50
სურ. 40 დაპროგრამების მეთოდი Double Symmetric polygon	50
სურ. 41 დაპროგრამების მეთოდი combined	51

სურ. 42 დაპროგრამების მეთოდი merged.....	51
სურ. 43 დაპროგრამების მეთოდი subtract.....	52
სურ. 44 დაპროგრამების მეთოდი Hybridized	52
სურ. 46 დანართი N1-ის #35 და #51 მაგალითები	55
სურ. 45 დანართი N1-ის მაგალითები	55
სურ. 47 დანართი N1-ის #42 და #53 მაგალითები	56
სურ. 48 #1 სატესტო მაგალითი.....	57
სურ. 49 #2 სატესტო მაგალითი.....	58
სურ. 50 #3 სატესტო მაგალითი.....	58
სურ. 51 #4 სატესტო მაგალითი.....	59
სურ. 52 #5 სატესტო მაგალითი.....	60
სურ. 53 #6 სატესტო მაგალითი.....	60
სურ. 54 #7 სატესტო მაგალითი.....	61
სურ. 55 #8 სატესტო მაგალითი.....	62
სურ. 56 #9 სატესტო მაგალითი.....	62
სურ. 57 #10 სატესტო მაგალითი.....	63
სურ. 58 #11 სატესტო მაგალითი.....	64
სურ. 59 #12 სატესტო მაგალითი.....	64
სურ. 60 #13 სატესტო მაგალითი.....	65
სურ. 61 #14 სატესტო მაგალითი.....	66
სურ. 62 #15 სატესტო მაგალითი.....	66
სურ. 63 #16 სატესტო მაგალითი.....	67
სურ. 64 #17 სატესტო მაგალითი.....	68
სურ. 65 #18 სატესტო მაგალითი.....	68
სურ. 66 #19 სატესტო მაგალითი.....	69
სურ. 67 #20 სატესტო მაგალითი.....	69
სურ. 68 #21 სატესტო მაგალითი.....	70
სურ. 69 #22 სატესტო მაგალითი.....	71
სურ. 70 #23 სატესტო მაგალითი.....	72
სურ. 71 #24 სატესტო მაგალითი.....	73
სურ. 72 #25 სატესტო მაგალითი.....	74
სურ. 73 #26 სატესტო მაგალითი.....	74
სურ. 74 #27 სატესტო მაგალითი.....	75
სურ. 75 #28 სატესტო მაგალითი.....	75
სურ. 76 #29 სატესტო მაგალითი.....	76
სურ. 77 #30 სატესტო მაგალითი.....	77
სურ. 78 #32 სატესტო მაგალითი.....	78
სურ. 79 #33 სატესტო მაგალითი.....	79
სურ. 80 #34 სატესტო მაგალითი.....	79
სურ. 81 #35 სატესტო მაგალითი.....	80
სურ. 82 #36 სატესტო მაგალითი.....	80
სურ. 83 #37 სატესტო მაგალითი.....	81
სურ. 84 #38 სატესტო მაგალითი.....	82
სურ. 85 #39 სატესტო მაგალითი.....	83

სურ. 86 #40 სატესტო მაგალითი.....	83
სურ. 87 #41 სატესტო მაგალითი.....	84
სურ. 88 #43 სატესტო მაგალითი.....	85
სურ. 89 #44 სატესტო მაგალითი.....	86
სურ. 90 #45 სატესტო მაგალითი.....	88
სურ. 91 #46 სატესტო მაგალითი.....	88
სურ. 92 #47 სატესტო მაგალითი.....	89
სურ. 93 #48 სატესტო მაგალითი.....	90
სურ. 94 #50 სატესტო მაგალითი.....	90
სურ. 95 #51 სატესტო მაგალითი.....	91
სურ. 96 #54 სატესტო მაგალითი.....	92
სურ. 97 #55 სატესტო მაგალითი.....	92
სურ. 98 #56 სატესტო მაგალითი.....	93
სურ. 99 #57 სატესტო მაგალითი.....	93
სურ. 100 #58 სატესტო მაგალითი.....	94
სურ. 101 #59 სატესტო მაგალითი.....	95
სურ. 102 #60 სატესტო მაგალითი.....	96
სურ. 103 #61 სატესტო მაგალითი.....	96
სურ. 104 #62 სატესტო მაგალითი.....	97
სურ. 105 #63 სატესტო მაგალითი.....	98
სურ. 106 #64 სატესტო მაგალითი.....	98
სურ. 107 #65 სატესტო მაგალითი.....	99
სურ. 108 #66 სატესტო მაგალითი.....	100
სურ. 109 პრიმიტივების სრული ნუსხის #26 და#27 მაგალითები.....	102
სურ. 110 პრიმიტივების სრული ნუსხის #29, #46 და #48 მაგალითები.....	103
სურ. 111 პრიმიტივების სრული ნუსხის #28, #34 და #53 მაგალითები.....	104
სურ. 112 პრიმიტივების სრული ნუსხის #30, და #49 მაგალითები.....	104
სურ. 113 პრიმიტივების სრული ნუსხის #38, და #54 მაგალითები.....	105
სურ. 114 პრიმიტივების სრული ნუსხის #51, და #52 მაგალითები.....	106
სურ. 115 ATLAS-ის ექსპერიმენტის ტრიგერული სისტემები.....	111
სურ. 116 ივენთის მახასიათებელი პარამეტრები	112
სურ. 117 სიმულაციის გამშვები პითონის სკრიპტი	113
სურ. 118 ფრაგმენტი ატლასის დეტექტორის გეომეტრიული აღწერების მონაცემთა ბაზიდან.....	114
სურ. 119 დეტექტორში ნაწილაკების გავრცელების სქემა.....	114
სურ. 120 ფრაგმენტი წარმატებული სიმულაციის log ფაილიდან	116
სურ. 121 ფრაგმენტი წარუმატებელი სიმულაციის log ფაილიდან	116
სურ. 122 ფრაგმენტი ხელოვნურად გამოწვეული წარუმატებელი სიმულაციის log ფაილიდან.....	117
სურ. 123 ჩვენი გეომეტრიული აღწერებით ჩატარებული სიმულაციის შედეგი ..	117
სურ. 124 default პარამეტრების ჩატარებული სიმულაციის შედეგი.....	118
სურ. 125 ფრაგმენტი default პარამეტრების ჩასანაცვლებელი პითონის სკრიპტიდან.....	118

სურ. 126 CPU Time და RAM-ის შესახებ ინფორმაცია სიმულაციის პროცესის log ფაილში.....	118
სურ. 127 სერვერ lxplus703.cern.ch - ის სიმულაციის შედეგები.....	120
სურ. 128 სერვერ lxplus723.cern.ch - ის სიმულაციის შედეგები.....	120
სურ. 129 სერვერ lxplus782.cern.ch - ის სიმულაციის შედეგები.....	120
სურ. 130 # 23 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	124
სურ. 131 # 28 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	126
სურ. 132 # 11 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	128
სურ. 133 cube და pyramid მეთოდების კვლევის შედეგები.....	130
სურ. 134 cube და pyramid მეთოდების კვლევის შედეგები. ოპერატიული მესხიერების დატვირთვა.....	132
სურ. 135 # 03 მაგალითი	133
სურ. 136 # 03 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	134
სურ. 137 # 27 მაგალითი	135
სურ. 138 # 27 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	136
სურ. 139 ფრაგმენტი geometry-ATLAS-R2-2016-01-00-01-ის GDML კოდიდან.....	141
სურ. 140 #23 მაგალითი	142
სურ. 141 Tube-vs-Chain-vs-Cylinder-ის ობიექტების რაოდენობაზე დამოკიდებული სიმულაციის შედეგები.....	143
სურ. 142 #11 მაგალითი ობიექტების რაოდენობაზე დამოკიდებული სიმულაციის შედეგები	145
სურ. 143 #39 მაგალითი	145
სურ. 144 #39 სიმულაციის შედეგები	146
სურ. 145 #19 მაგალითი	147
სურ. 146 #19 მაგალითის სიმულაციის შედეგები	148
სურ. 147 NSW-ში პრიმიტივების რაოდენობა. ფრაგმენტი CATIA-დან.....	153
სურ. 148 ATLAS-ექსპერიმენტის New Small Wheel.....	154
სურ. 149 NSW-ის ძირითადი ნაწილები.....	155
სურ. 150 ფრაგმენტი NSW-ის პროგრამული კოდიდან.....	156
სურ. 151 ოპტიმალური და არა ოპტიმალური CGS მეთოდების სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	159
სურ. 152 ოპტიმალური და არა ოპტიმალური Arbitrary vs Pyramid + Boolean მეთოდების სიმულაციის შედეგები წარმოდგენილი გრაფიკულად	160

ცხრ. 43 #42 მაგალითის დაპროგრამების მეთოდები.....	85
ცხრ. 44 #43 მაგალითის დაპროგრამების მეთოდები.....	86
ცხრ. 45 #44 მაგალითის დაპროგრამების მეთოდები.....	87
ცხრ. 46 #45 მაგალითის დაპროგრამების მეთოდები.....	88
ცხრ. 47 #46 მაგალითის დაპროგრამების მეთოდები.....	89
ცხრ. 48 #47 მაგალითის დაპროგრამების მეთოდები.....	89
ცხრ. 49 #48 მაგალითის დაპროგრამების მეთოდები.....	90
ცხრ. 50 #50 მაგალითის დაპროგრამების მეთოდები.....	91
ცხრ. 51 #51 მაგალითის დაპროგრამების მეთოდები.....	91
ცხრ. 52 #54 მაგალითის დაპროგრამების მეთოდები.....	92
ცხრ. 53 #55 მაგალითის დაპროგრამების მეთოდები.....	92
ცხრ. 54 #57 მაგალითის დაპროგრამების მეთოდები.....	94
ცხრ. 55 #58 მაგალითის დაპროგრამების მეთოდები.....	95
ცხრ. 56 #59 მაგალითის დაპროგრამების მეთოდები.....	95
ცხრ. 57 #60 მაგალითის დაპროგრამების მეთოდები.....	96
ცხრ. 58 #61 მაგალითის დაპროგრამების მეთოდები.....	97
ცხრ. 59 #62 მაგალითის დაპროგრამების მეთოდები.....	97
ცხრ. 60 #63 მაგალითის დაპროგრამების მეთოდები.....	98
ცხრ. 61 #64 მაგალითის დაპროგრამების მეთოდები.....	99
ცხრ. 62 #65 მაგალითის დაპროგრამების მეთოდები.....	99
ცხრ. 63 #66 მაგალითის დაპროგრამების მეთოდები.....	100
ცხრ. 64 #26, #27 მაგალითების დაპროგრამების მეთოდები.....	102
ცხრ. 65 #29, #46 და #48 მაგალითების დაპროგრამების მეთოდები.....	103
ცხრ. 66 #28, #34 და #54 მაგალითების დაპროგრამების მეთოდები.....	104
ცხრ. 67 #30 და #49 მაგალითების დაპროგრამების მეთოდები.....	105
ცხრ. 68 #38 და #54 მაგალითების დაპროგრამების მეთოდები.....	105
ცხრ. 69 #51 და #52 მაგალითების დაპროგრამების მეთოდები.....	106
ცხრ. 70 #05 და #05 მაგალითების დაპროგრამების მეთოდები.....	106
ცხრ. 71 სიმულაციის შედეგები CPU Time-ის ხმაურის დასადგენად.....	119
ცხრ. 72 #23 მაგალითის დაპროგრამების მეთოდები.....	123
ცხრ. 74 #23 მაგალითის სიმულაციის წარმადობის შედეგები.....	123
ცხრ. 73 #23 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი.....	123
ცხრ. 75 #28 მაგალითის დაპროგრამების მეთოდები.....	124
ცხრ. 76 #28 მაგალითის სიმულაციის წარმადობის.....	125
ცხრ. 77 #28 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი.....	125
ცხრ. 78 #11 მაგალითის დაპროგრამების მეთოდები.....	126
ცხრ. 79 #11 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი.....	127
ცხრ. 80 #08, #21, #26, #32 მაგალითების დაპროგრამების მეთოდები.....	127
ცხრ. 81 #11 მაგალითის სიმულაციის შედეგები.....	127
ცხრ. 82 #01, #02; #04; #05; #30; #24; #31; #39; #42 მაგალითების დაპროგრამების მეთოდები.....	129
ცხრ. 83 CPU Time-ის ხმაურის კოეფიციენტი cube და pyramid მეთოდებისთვის.	129
ცხრ. 84 სიმულაციიდან მიღებული შედეგები cube და pyramid მეთოდებისათვის.....	130

ცხრ. 85 სიმულაციიდან მიღებული შედეგები I-დან VII-მდე მაგალითებისათვის	132
ცხრ. 86 #03 მაგალითის დაპროგრამების მეთოდები.....	133
ცხრ. 87 #03 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი	133
ცხრ. 88 #03 მაგალითის სიმულაციის შედეგები.....	134
ცხრ. 89 #27 მაგალითის დაპროგრამების მეთოდები.....	135
ცხრ. 90 #27 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი	135
ცხრ. 91 #27 მაგალითის სიმულაციის შედეგები.....	136
ცხრ. 92 #19-ის დაპროგრამების მეთოდები	137
ცხრ. 93 #19 მაგალითის სიმულაციის შედეგები.....	140
ცხრ. 94 ხმაურის კოეფიციენტი #23 მაგალითისთვის	142
ცხრ. 95 #23 მაგალითი. სხვადასხვა რაოდენობის ობიექტების სიმულაციის შედეგები	143
ცხრ. 96 #11 მაგალითის დაპროგრამების მეთოდები.....	144
ცხრ. 97 ხმაურის კოეფიციენტი #11 მაგალითისთვის	144
ცხრ. 98 #11 მაგალითი. სხვადასხვა რაოდენობის ობიექტების სიმულაციის შედეგები	144
ცხრ. 99 #39 მაგალითის დაპროგრამების მეთოდები.....	145
ცხრ. 100 #39 მაგალითის CPU time-ის ხმაურის კოეფიციენტი.....	146
ცხრ. 101 #39 მაგალითის სიმულაციის შედეგები.....	146
ცხრ. 102 #19 CPU time-ის ხმაურის კოეფიციენტი.....	147
ცხრ. 103 #19 მაგალითის დაპროგრამების მეთოდები.....	147
ცხრ. 104 #19 მაგალითის სიმულაციის შედეგები.....	148
ცხრ. 105 მიღებული კვლევის შედეგების აპრობაციის ეტაპები გამოსახული სქემატურად	151
ცხრ. 106 ATLAS-ის დეტექტორის გეომეტრიული აღწერების ზოგადი პარამეტრები	152
ცხრ. 107 NSW -ში გამოყენებული arbitrary მეთოდის ანალიზის შედეგები	157
ცხრ. 108 ოპტიმალური და არა ოპტიმალური CGS მეთოდების სიმულაციის შედეგები	158
ცხრ. 109 ოპტიმალური და არა ოპტიმალური CGS მეთოდების CPU time-ის ხმაურის კოეფიციენტი.....	158
ცხრ. 110 ოპტიმალური და არა ოპტიმალური Arbitrary vs Pyramid + Boolean.....	159
ცხრ. 111 მეთოდის ეფექტურობა დეტექტორის სხვადასხვა სტრუქტურებისთვის	161

შესავალი

თემის აქტუალობა და მიზანი.

დღეისათვის მსოფლიოში რამდენიმე დიდი პროექტი ხორციელდება მაღალი ენერგიების ფიზიკაში. მათ შორის განსაკუთრებული ადგილი უჭირავს LHC (large Hadron Collider) -ის ექსპერიმენტს. ასეთ ამბიციური მიზნები გააჩნიათ. როგორცაა: სიმსივნური დაავადების წინააღმდეგ ბრძოლა, დიდი აფეთქების თეორიის ექსპერიმენტალური კვლევა და ა.შ. LHC-ის ექსპერიმენტი მიმდინარეობს CERN-ში, ჟენევა, შვეიცარია. LHC მდებარეობს მიწის ქვეშ 50 - 175 მ-ის სიღრმეზე. ის არის წრიული ფორმის ამაჩქარებელი, რომლის სიგრძეა 27 კმ. ამაჩქარებლის გვირაბი დედამიწის ზედაპირის მიმართ დახრილია 1.4° -ით. მისი ძირითადი მიზნებია: სტანდარტული მოდელის ექსპერიმენტალური დადასტურება. სტანდარტული მოდელის დადასტურებისთვის საჭიროა, დიდი მასის მქონე Z ბოზონის და ჰიგსის ბოზონის ექსპერიმენტალურად აღმოჩენა. ზემოთ ჩამოთვლილი საკითხების კვლევა LHC-ის ექსპერიმენტში პროტონების დაჯახების გზით მიმდინარეობს.

ATLAS-ის ექსპერიმენტში ერთერთი მთავარი მიმართულებაა ექსპერიმენტის სიმულაცია. სიმულაციის მიზანია ის, რომ რიგი ექსპერიმენტები ჩატარდეს არა რეალურ დეტექტორებზე, არამედ მათ მათემატიკურ მოდელებზე. სიმულაცია საშუალებას იძლევა გამოკვლეულ იქნეს დეტექტორში არ არსებული, თეორიული მოვლენები. ATLAS-ის ექსპერიმენტის სიმულაცია დაფუძნებულია Geant4 მოდელირების პაკეტზე.

წელიწადში დაახლოებით 5-7 მილიარდი ხელოვნური მოვლენის ფორმირება ხდება. და მათ დასამუშავებლად Geant4-ს სჭირდება 77 მლნ საათზე მეტი. სწორედ ამიტომ ძალიან მნიშვნელოვანი პარამეტრია სიმულაციის წარმადობა. წარმადობაზე მნიშვნელოვან გავლენას ახდენს სიმულაციის პაკეტებში გამოყენებული ATLAS-ის დეტექტორის გეომეტრიული აღწერები, რომლებიც დაპროგრამების სხვადასხვა

მეთოდებით არიან შექმნილი. თუ რა გავლენას ახდენს გეომეტრიული აღწერების დაპროგრამების მეთოდები სიმულაციის პროცესის წარმადობაზე შესწავლილი არ არის და მისი გამოკვლევა წარმოადგენს აქტუალურ ამოცანას.

ATLAS-ის ექსპერიმენტის სიმულაციას გეომეტრიული აღწერები სხვადასხვა წყაროდან მიეწოდება და მათი დაპროგრამების რამდენიმე მეთოდი არსებობს. ისინი 3 ძირითად ნაწილად შეიძლება დაიყოს

1. CGS (Constructive Geometry Solids) - მყარი ტანის პრიმიტივების მეთოდები
2. პოლიგონ მეთოდები
3. Boolean ოპერატორები

ამ მეთოდებით შესაძლებელია შეიქმნას ცილინდრული, წახნაგოვანი ან შერეული ზედაპირების მქონე ობიექტები. კვლევის ძირითად მიზანს წარმოადგენს გაირკვეს რა გავლენას ახდენს დაპროგრამების მეთოდები სიმულაციის წარმადობაზე, როგორც დამოუკიდებლად ასევე სხვადასხვა კომბინაციის შემთხვევაში.

კვლევის ობიექტი და მეთოდები

კვლევის ობიექტს წარმოადგენს ATLAS-ის დეტექტორის გეომეტრიული აღწერების დაპროგრამების მეთოდები. კვლევის ფარგლებში გამოყენებულია შემდეგი მეთოდები: გეომეტრიული კლასიფიკაციის მეთოდი, კოდის სტრუქტურების ფორმალიზაციის მეთოდი, დაჯგუფების მეთოდი, საკონტროლო მაგალითების ტესტირების მეთოდი, სიმულაციის საცდელი სესიების შედეგების ანალიზის მეთოდი.

მეცნიერული სიახლე

1. გამოიყო ატლასის დეტექტორის გეომეტრიული აღწერების დაპროგრამების ტიპური კლასები და მეთოდები

2. წახნაგოვანი ობიექტების დაპროგრამებისათვის დადასტურდა, რომ პოლიგონ მეთოდები განაპირობებენ უკეთეს წარმადობას (CPU 8%, RAM 20%) ვიდრე CGS მეთოდები

3. ცილინდრული ობიექტებისათვის დადგინდა, რომ დაპროგრამების მეთოდების ცვლილებით წარმადობა მნიშვნელოვნად არ იცვლება (4%)

4. დადგინდა, რომ კომბინირებული ობიექტებისათვის პოლიგონ დაპროგრამების მეთოდებს უკეთესი წარმადობა აქვთ ვიდრე CGS მეთოდებს და ეს სხვაობა ობიექტების რაოდენობის პირდაპირპროპორციულია.

პრაქტიკული მნიშვნელობა.

1. შეიქმნა ატლასის დეტექტორის ტიპური გეომეტრიული აღწერების ბიბლიოთეკა, რომლის საშუალებითაც შესაძლებელია ატლასის დეტექტორის გეომეტრიული მოდელის აგება

2. გეომეტრიული აღწერების ბიბლიოთეკისთვის დამუშავდა გეომეტრიული აღწერების დაპროგრამების მეთოდები - ცილინდრული კლასისთვის 29, წახნაგოვანი კლასისთვის 135 და კომბინირებული კლასისთვის 252

3. შეიქმნა სიმულაციის წარმადობის შესწავლის საკვლევი ინფრასტრუქტურა

4. გამოვლენილ იქნა Tube მეთოდის ეფექტურობა ცილინდრული ობიექტების დაპროგრამების ამოცანებისათვის

5. გამოვლენილ იქნა Arbitrary_Polygon მეთოდის ეფექტურობა წახნაგოვანი ობიექტების დაპროგრამების ამოცანებისათვის

6. გამოვლინდა უმნიშვნელო სხვაობა Cube და Pyramid მეთოდებს შორის როგორც წახნაგოვანი, ასევე კომბინირებული ობიექტების დაპროგრამების ამოცანებისათვის

7. განხორციელდა დაპროგრამების მეთოდების რანჟირება რისთვისაც განხორციელდა მეთოდების დაჯგუფება და განისაზღვრა დაპროგრამების მეთოდების რეიტინგი წარმადობის თვალსაზრისით.

სამუშაოს აპრობაცია.

მეთოდის აპრობაცია ჩატარდა ატლასის დეტექტორის ერთერთი ქვესისტემის „მცირე თვალი“ New Small Wheel (NSW) დაპროგრამების ამოცანაზე. მიღებული შედეგების განვრცობით მთლიანად ატლასის დეტექტორის ყველა ქვესისტემისთვის განისაზღვრა მოსალოდნელი ეფექტურობა - CPU-თვის 62% და RAM-თვის 70%.

კვლევის შედეგები განხილულ იქნა შემდეგ სამეცნიერო კონფერენციაზე:

1. "საინფორმაციო საზოგადოება და განათლების ინტენსიფიკაციის ტექნოლოგიები" - ISITE'21. თბილისი, საქართველო, 20-22 მაისი, 2021

დისერტაციის თემატიკაზე გამოქვეყნდა 3 სტატია:

1. Alexander Sharmazanashvili, Besik Kekelia. Analysis of Geometry Description for Optimized Simulation Performance. *GESJ: Computer Science and Telecommunications 2021, No.1(59)*. გვ. 31-40
2. ბ. კეკელია. სატესტო მაგალითების ფორმირება ატლასის ექსპერიმენტის სიმულაციის ამოცანებისათვის. *საქართველოს საინჟინრო სიახლენი, Georgian Engineering News, №1, 2021*. გვ. 41-44
3. ბ. კეკელია. გეომეტრიული აღწერების მეთოდების გავლენა ატლასის ექსპერიმენტის სიმულაციის წარმადობაზე. *საქართველოს საინჟინრო სიახლენი, Georgian Engineering News, №1, 2021*. 36-40

სადისერტაციო ნაშრომის სტრუქტურა.

სადისერტაციო ნაშრომი შეიცავს 5 თავს, ძირითად დასკვნებს, გამოყენებულ ლიტერატურას და 3 დანართს. დანართების გარეშე ნაშრომი არის 167 გვერდი.

თავი I: ATLAS-ის ექსპერიმენტის კომპიუტინგი

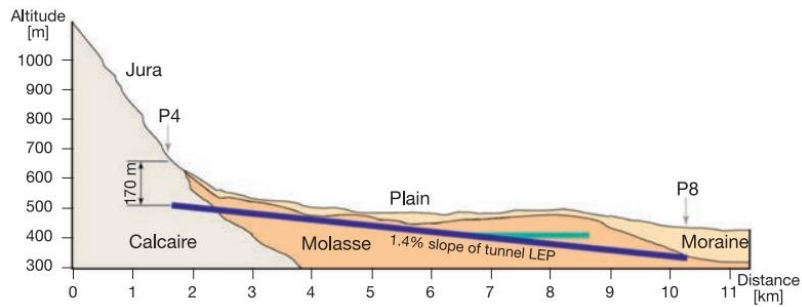
1.1 ბირთვული ფიზიკის პროექტები

ბირთვული ფიზიკის ექსპერიმენტები, დღეისათვის მსოფლიოში მრავალ ქვეყანაში მიმდინარეობს. ჯამში 30000-ზე მეტი ამაჩქარებელი არის აღრიცხული [1]. მათ ამბიციური მიზნები გააჩნიათ. როგორცაა: სიმსივნური დაავადების წინააღმდეგ ბრძოლა, დიდი აფეთქების თეორიის ექსპერიმენტალური კვლევა და ა.შ. მათ შორის რამდენიმე პროექტი არის აღსანიშნავი: LHC (The Large Hadron Collider), KEK (The High Energy Accelerator Research Organization), LINAC (Linear particle accelerator), COSY (Two-dimensional nuclear magnetic resonance spectroscopy), DESY (Deutsches Elektronen-Synchrotron) და ა.შ. აქედან კი თავისი მასშტაბებიდან და მიზნებიდან გამომდინარე გამორჩეულია LHC (The Large Hadron Collider).



სურ. 1 LHC-ის ამაჩქარებელი

LHC ნაწილაკების ფიზიკის ყველაზე დიდი ექსპერიმენტია მსოფლიოში [2]. ის მიმდინარეობს CERN (ბირთვული კვლევების ევროპული ორგანიზაცია, შვეიცარია, ჟენევა) - ში და მასში ჩართულია მსოფლიოს 36 ქვეყნის 608 უნივერსიტეტი. LHC-ის ამაჩქარებელი მდებარეობს შვეიცარიის და საფრანგეთის საზღვარზე, ის განთავსებულია წრიული ფორმის 26.7 კმ სიგრძის გვირაბში, მიწის ქვეშ 45-170მ-ის სიღრმეზე და ჰორიზონტალური მდგომარეობიდან 1.4°-ით დახრილია [3]. სურ.2. ამაჩქარებლის გვირაბში ტემპერატურა თითქმის აბსოლუტური 0-ის ტოლია და ის შეადგენს -271°C, ასეთი ტემპერატურის მისაღებად გამოიყენება 96 ტ ჰელიუმი. ეს



სურ. 2 LHC-ის ამაჩქარებელის სქემატური გეოლოგიური გეგმა

ტემპერატურა საჭიროა იმისათვის, რომ ასეთ პირობებში ყველაფერი კარგავს თავის თვისებას და ასევე წინაღობას, შედეგად შესაძლებელია სიგნალების და ელ. ენერგიის წინაღობის გარეშე გადაცემა. LHC-ის შექმნის ძირითადი მიზნებია:

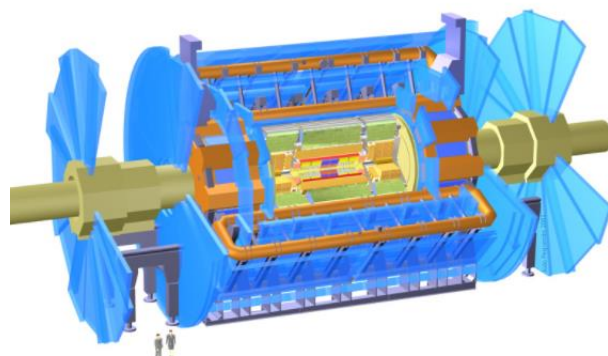
1. სტანდარტული მოდელის ექსპერიმენტალური დადასტურება და აქედან გამომდინარე დიდი მასის მქონე Z ბოზონის და ჰიგსის ბოზონის აღმოჩენა
2. დიდი აფეთქების თეორიის ექსპერიმენტალური კვლევა
3. სუპერსიმეტრიის დადასტურება

LHC-ის ამაჩქარებელზე ურთიერთსაწინააღმდეგო მიმართულებით ხორციელდება პროტონების აჩქარება საბოლოოდ ისინი აღწევენ 14 GeV ენერგიას და შემდეგ წინააღმდეგ განსაზღვრულ ადგილებში ერთმანეთს ეჯახებიან [4], შედეგად კი წარმოიქმნებიან ახალი ნაწილაკები, რომელთა სიცოცხლის ხანგრძლივობა შეადგენს წამის 40/1000000 ნაწილს. მათი დაფიქსირება, გადაადგილების ტრაექტორიის დადგენა და ასევე მათი ენერგიების გაზომვა დეტექტორებში ხორციელდება. ნაწილაკების დეტექტირებისთვის სულ 7 დეტექტორი გამოიყენება [5][6], სოლო აქედან 4 ძირითადი დეტექტორია ესენია:

1. ATLAS (A Toroidal LHC AparaturuS)
2. CMS (Compact Muon Solenoid)
3. ALICE (A Large Ion Collider Experiment)
4. LHCb (LHC Beauty)

ისინი LHC ექსპერიმენტში დამოუკიდებელი ექსპერიმენტების სახით არიან წარმოდგენილები. დეტექტორები განკუთვნილია, პროტონების შეჯახების შემდეგ წარმოქმნილი ახალი ნაწილაკების და ჰიგსის ბოზონის აღმოსაჩენად. ATLAS-ის დეტექტორში ნაწილაკების კვლევა ძლიერ მაგნიტურ ველში ხორციელდება. CMS - ის დანიშნულება ჰიგსის ბოზონის აღმოჩენა და შავი მატერიის საკითხების კვლევაა. ATLAS-ის ექსპერიმენტისაგან განსხვავებით CMS-ში ნაწილაკების კვლევა მყარ სხეულებში მიმდინარეობს. ის ყველაზე მძიმე დეტექტორია და მისი წონაა 12,5 ათასი ტონა. ALICE - ის მიზანი არის, დიდი აფეთქების საკითხების კვლევა, ხოლო LHCb - ის შექმნის მიზანია, მატერიის და ანტი მატერიის კვლევა დიდი აფეთქების პირობებში და ასევე ერთ-ერთი უმთავრესი მიზანი CP სიმეტრიის დარღვევის შესწავლა. ნაწილაკების დაჯახების შედეგად სხვადასხვა ნაწილაკები წარმოიქმნება მაგ. მიონი, ელექტრონი, ფოტონი და ა.შ. და დეტექტორმა ყველა მათგანის დაფიქსირება უნდა შეძლოს. ამიტომ თითოეული დეტექტორი რამდენიმე ქვე-დეტექტორისგან შედგება, ქვე-დეტექტორები კონკრეტული ნაწილაკის აღმოჩენაზე არიან პასუხისმგებლები.

ATLAS-ის დეტექტორი ერთ-ერთი ყველაზე დიდი დეტექტორია



სურ. 3 ATLAS-ის დეტექტორი

ზემოთ აღნიშნულ ექსპერიმენტში სურ. 3. ის არის სიგრძით 45 მ და სიმაღლით 25 მ. წონა კი შეადგენს 7000 ტონას. დეტექტორი 4 ძირითადი სტრუქტურისგან შედგება ესენია: მაგნიტური სისტემა, შიდა დეტექტორი, კალორიმეტრების სისტემა და მიონების სისტემა. მაგნიტური სისტემა ქმნის

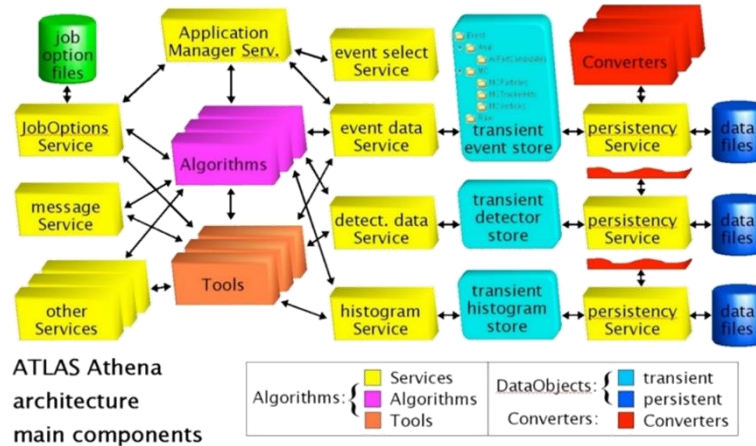
2 ტესლა მაგნიტურ ველს, რომელიც ძლიერ ურთიერთქმედებაში მყოფი ნაწილაკების გადახრისათვის გამოიყენება [7]. შიდა დეტექტორის დანიშნულებაა ელექტრულად დამუხტული ნაწილაკების აღმოჩენა, რომელთა სიცოცხლის ხანგძლივობა ძალიან მცირეა. კალორიმეტრების სისტემის დანიშნულებაა ნაწილაკების ენერგიების გაზომვა. მიონური სისტემა კი განკუთვნილია ნაწილაკის - მიონის აღმოსაჩენად. მიონებზე დეტექტორის სხვა სისტემები გავლენას ვერ ახდენს. მისი მთავარი მიზანი ჰიგსის ბოზონის აღმოჩენა იყო. 2012 წლის 4 ივლისს ATLAS და CMS ექსპერიმენტებმა ერთმანეთის დამოუკიდებლად პრაქტიკული და თეორიული კვლევების შედეგად აღმოაჩინეს ჰიგსის ბოზონი [8]. ATLAS-ის დეტექტორში ნაწილაკების დეტექტირების პროცესი სხვადასხვა IT ამოცანასთან არის კავშირში როგორცაა [7]: მონაცემების მიღება და დამუშავება და ფიზიკური პროცესების სიმულაცია.

1.2 ATLAS-ის ექსპერიმენტის პროგრამული უზრუნველყოფა

ATLAS-ის დეტექტორში პროტონების შეჯახების შედეგად მიიღება დიდი რაოდენობის ინფორმაცია - 15 პეტაბაიტი წამის განმავლობაში [9]. ტრიგერული სისტემების მიერ ხდება ამ ინფორმაციის გაფილტვრა ონლაინ რეჟიმში და შედეგად მიღებული ინფორმაცია (წამში 300 მბ) ინახება მონაცემთა ბაზებში. სწორედ ეს ინფორმაცია არის საინტერესო ფიზიკოსებისათვის, რომლის დამუშავებაც ხდება Athena-ში. Athena არის ATLAS-ის ექსპერიმენტის offline პროგრამული უზრუნველყოფა. ის დაფუძნებულია GAUDI-ის ფრეიმვორკზე [10]. GAUDI არის პროგრამული არქიტექტურა, რომელიც გამოიყენება მაღალი ენერგიების ფიზიკის ექსპერიმენტის მონაცემთა დამუშავების აპლიკაციების შესაქმნელად და განსავითარებლად. GAUDI-ის ფრეიმვორკი დაყრდნობილია CLHEP (short for A Class Library for High Energy Physics) C++ ბიბლიოთეკაზე. ის შექმნილია სპეციალურად მაღალი ენერგიების ფიზიკის პროგრამული პაკეტებისათვის. აღნიშნული ბიბლიოთეკა წარმოადგენს კლასებს მათემატიკური

გამოთვლების დაპროგრამებისათვის, ფსევდო რიცხვების გენერირებისთვის, წრფივი ალგორითმების და მაღალი ენერჯის ფიზიკის სიმულაციისა და ანალიზისათვის [11].

Athena-ს ფრეიმვორკი (სურ. 4) შედგება შემდეგი კომპონენტებისაგან:



სურ. 4 Athena-ს ფრეიმვორკი

Algorithm; Data object; Service; JobOptions; Tool; Auditors; Transient Store; Persistent Store; Converter. ეს კომპონენტები სამ ძირითად კლასში არიან გაერთიანებულები [8]:

1. ალგორითმები
2. მონაცემთა ობიექტები (Data object)
3. კონვერტორები

ალგორითმების კლასში გაერთიანებულები არიან სერვისები, ალგორითმები და დამხმარე აპლიკაციები. ალგორითმები - ეს არის C++ დაწერილი პროგრამული კოდები, სადაც წინასწარ არის განსაზღვრული დავალების შესრულების თანმიმდევრობა. ალგორითმები იღებენ შესასვლელ ინფორმაციას, ასრულებენ და აგენერირებენ გამომავალ ინფორმაციას, რომელსაც მონაცემთა ობიექტები ეწოდება. სერვისები - არის პროგრამების ერთობლიობა რომლებიც ასრულებენ სხვადასხვა სახის დავალებებს, როგორცაა: მესიჯის გამოტანა, ჰისტოგრამების გამოტანა და სხვ. მაგ. JobOptions სერვისი. ეს არის პითონის სკრიპტი, რომელიც მომხმარებელს საშუალებას აძლევს Athena-ს მიაწოდოს თავისი ინსტრუქცია.

კიდევ ერთი მაგალითი არის Auditors, ეს არის პროგრამული უზრუნველყოფა, რომელიც მონიტორინგს უკეთებს Athena-ს ფრეიმვორკს, ის იყოფა 3 ნაწილად:

1. NameAuditor
2. ChronoAuditor
3. MemoryAuditor

NameAuditor - ის მოვალეობაა ალგორითმების დიაგნოსტიკა, შედეგად ფიქსირდება იმ ალგორითმების სახელი, სადაც ხარვეზი იქნება აღმოჩენილი. ChronoAuditor ახორციელებს პროცესორის დატვირთვის მონიტორინგს, ხოლო MemoryAuditor კი მეხსიერების კონტროლისთვის გამოიყენება. ის იძლევა გაფრთხილებას აღმოჩენილი დარღვევების შესახებ, როგორცაა მეხსიერების გადავსება და სხვ. დამხმარე აპლიკაციები - ეს არის პროგრამები, ფუნქციები, რომლებზეც წვდომა აქვს ყველა ალგორითმს, მაგ. ძიების ფუნქცია, წაშლის ფუნქცია. და ა.შ.

რაც შეეხება მონაცემთა ობიექტების კლასს აქ გაერთიანებული არიან Transient Store და Persistent Store. Transient Store არის სივრცე ოპერატიულ მეხსიერებაში, სადაც ხდება მონაცემთა ობიექტების შენახვა, რომლებიც შემდეგ ალგორითმებს უნდა გადაეცეს შესასრულებლად, ხოლო Persistent Store არის ადგილი დისკზე, სადაც ხდება მონაცემთა ობიექტების შენახვა.

Convertor-ები არის პროგრამული პაკეტი, რომლებიც Persistent Store -იდან კითხულობენ და აკონვერტირებენ ინფორმაციას სხვადასხვა ფორმატებში, მათ შორის გრაფიკულ ფორმატებშიც.

Athena-ს ფრეიმვორკის უპირატესობას განპირობებულია სერვისების მრავალფეროვნებით, რომლებიც ხელმისაწვდომია მომხმარებლისთვის. ისინი ხელს უწყობენ კვლევის პროცესს და შედეგის პროდუქტიულობას.

1.3 ფიზიკური პროცესების სიმულაცია

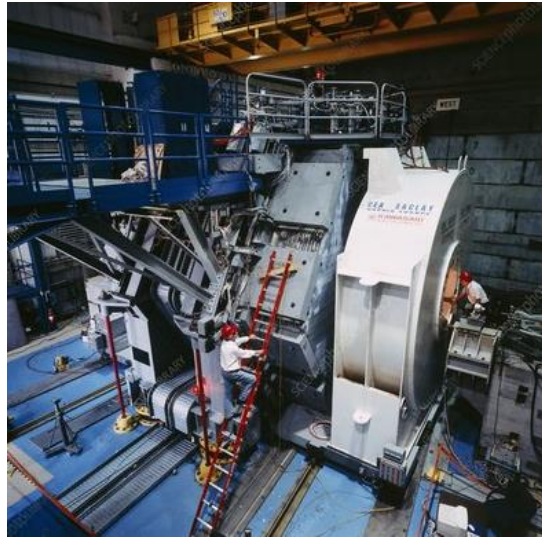
ექსპერიმენტის სიმულაცია მნიშვნელოვანი პროცესია, გამომდინარე იქიდან, რომ თუ რეალური ექსპერიმენტის და სიმულაციის შედეგები ერთმანეთს დაემთხვევა რიგი ექსპერიმენტები შესაძლოა ჩატარდეს არა რეალურ დეტექტორებზე, არამედ მათ მათემატიკურ მოდელებზე. სიმულაცია საშუალებას იძლევა გამოკვლევულ იქნეს დეტექტორში არ არსებული, თეორიული მოვლენები, იგივე ივენთი. ივენთი არის პროტონების დაჯახება და ამ პროცესის შედეგად მიღებული ინფორმაცია. ასევე ნებისმიერ მეცნიერს აქვს შესაძლებლობა თავისი თეორიის ექსპერიმენტალური კვლევა განახორციელოს მათემატიკურ მოდელებზე.

ATLAS-ის ექსპერიმენტის სიმულაციის პროცესი დაფუძნებულია Geant4 (Geometry and Tracking) მოდელირების პაკეტზე [12][13]. Geant4 არის პლატფორმა, რომლის საშუალებითაც ხორციელდება სხვადასხვა სიმკვრივის ობიექტებში ნაწილაკების გავლის პროცესის მოდელირება მონტე-კარლოს მეთოდების გამოყენებით. ეს არის ალგორითმების ფართო კლასი, რომლებიც რიცხვითი შედეგების მისაღებად იყენებენ შემთხვევითი შერჩევის პრინციპს [14]. პროტონების დაჯახების შედეგად წარმოქმნილი ნაწილაკები გადიან დეტექტორის გეომეტრიაში და იცვლიან მოძრაობის ტრაექტორიას, Geant4-ში კი სწორედ ამ პროცესის მოდელირებაა შესაძლებელი. ამისათვის მას სჭირდება დეტექტორის გეომეტრიული აღწერები და ივენთები. Geant4, როგორც მოდელირების პაკეტი სხვადასხვა სფეროში გამოიყენება, როგორცაა [15]: მაღალი ენერგიების ფიზიკის ექსპერიმენტები, კოსმოსი, მედიცინა და ა.შ.

Geant4, როგორც სიმულაციის ინსტრუმენტი გამოიყენება მაღალი ენერგიების შემდეგ პროექტებში: BABAR, ATLAS, CMS, ALICE, Fermilab, ILC, HARP და ა. შ.

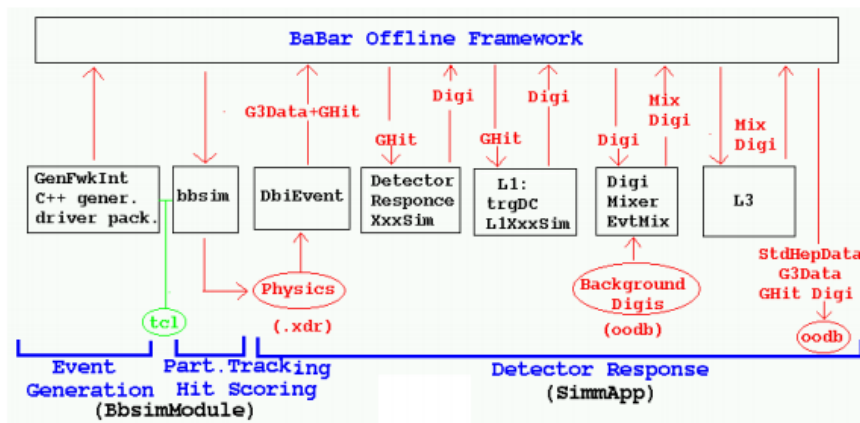
BABAR ექსპერიმენტი მიმდინარეობს SLAC ის ეროვნული ამაჩქარებლის ლაბორატორიაში და მონაწილეობს 500-ზე მეტი ფიზიკოსი და

ინჟინერი სურ.5. მისი ძირითადი მიზანია CP სიმეტრიის დარღვევის



სურ. 5 BABAR დეტექტორი

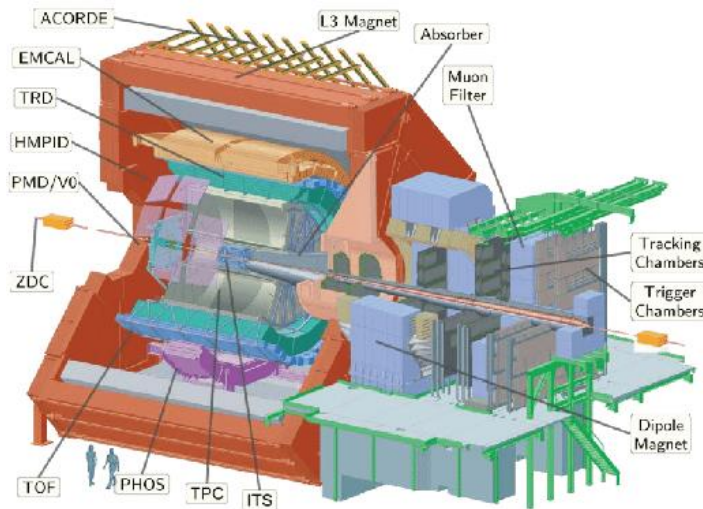
სისტემური შესწავლა და ასიმეტრია B მეზონების დაშლაში [16]. BABAR ექსპერიმენტში 2 სახის სიმულაცია გამოიყენება და დამყარებულია Geant4-ის მოდულზე BOGUS ის იმიგრება, როგორც BaBar Object-oriented Geant4-based Unified Simulation [17]. ექსპერიმენტში გამოიყენება, სწრაფი სიმულაცია რომელსაც უწოდებენ ASLUND და სრული სიმულაცია - BBSIM. ეს უკანასკნელი იყენებს დეტექტორის დეტალურ გეომეტრიულ აღწერებს და ივენტებს, რომლებიც წარმოიქმნებიან ივენტ გენერატორების მიერ. ისინი ინტეგრირებულნი არიან Geant4-ში. შემდეგ ხდება ე.წ hit-ების ფორმირება. hit არის დეტექტორის და ნაწილაკის ურთიერთქმედების შედეგად მიღებული სიგნალები. ამ ექსპერიმენტში მათ უწოდებენ GHit-ს. შემდგომ



სურ. 6 BABAR ექსპერიმენტის სიმულაციის სქემა

ეტაპზე კი ხდება დიჯიტიზაცია. ეს არის პროცესი როცა ხდება GHit-ების კონვერტაცია რეალური ექსპერიმენტიდან მიღებული ინფორმაციის ფორმატის მსგავს ფორმატში. BABAR ექსპერიმენტის სიმულაციის სქემა მოცემულია სურ. 6-ზე.

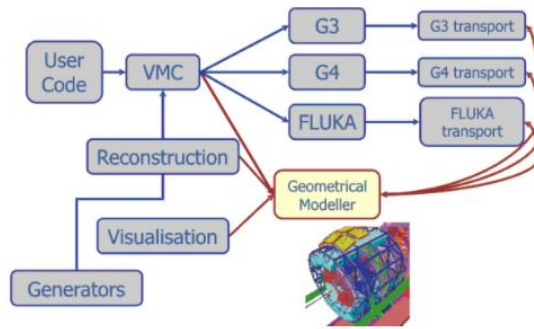
Geant4-ზე არის დაფუძნებული ასევე ALICE ექსპერიმენტის სიმულაცია. ALICE ექსპერიმენტი LHC-ზე მიმდინარე ექსპერიმენტებიდან ერთერთია სურ.7. BARBAR-ის ექსპერიმენტის მსგავსად ALICE-ის



სურ. 7 ALICE-ის დეტექტორი

ექსპერიმენტის ერთ-ერთ უმთავრეს მიზანს წარმოადგენს CP სიმეტრიის დარღვევის სისტემური შესწავლა. ALICE არის ზოგადი დანიშნულების მძიმე იონების დეტექტორი [18].

ALICE ექსპერიმენტის სიმულაცია მიმდინარეობს AliRoot-ში. ის არის პროგრამული უზრუნველყოფა სიმულაციისთვის. მისი ერთ-ერთი კომპონენტია VMC (The Virtual Monte Carlo)-ში სურ. 8. [19]. ის წარმოადგენს ინტერფეისს, მომხმარებლის კოდის შესაქმნელად. ის ასევე უზრუნველყოფს ინტერფეისს დეტექტორის გეომეტრიული აღწერების შესაქმნელად. VMC კონცეფციის უპირატესობას წარმოადგენს ის, რომ გეომეტრიული აღწერები არ არის ინტეგრირებული სხვა კომპონენტებთან და მისი გამოყენება Geant-ს და FLUKA-ს შეუძლიათ დამოუკიდებლად. რაც შეეხება კონკრეტულად სიმულაციის პროცესს, BARBAR ექსპერიმენტის სიმულაციის მსგავსად აქაც



სურ. 8 VMC კონცეფციის სქემა

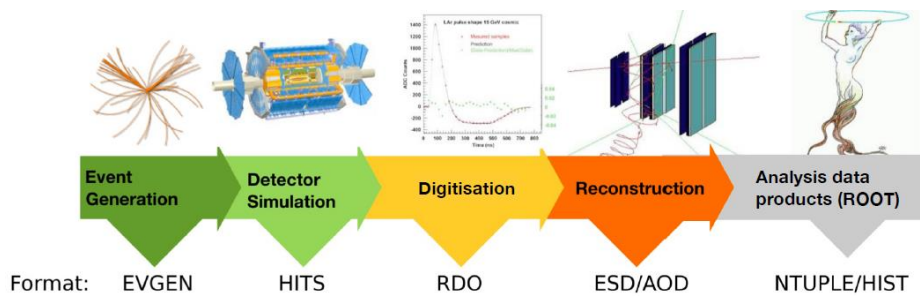
ხდება ივენთების წარმოქმნა გენერატორების მიერ, შემდეგ Hit-ების ფორმირება, შემდეგ დიჯიტიზაცია და ა.შ.

როგორც ზემოთ აღინიშნა Geant4 როგორც სიმულაციის პაკეტი გარდა ბირთვული ფიზიკის პროექტებისა გამოიყენება კოსმოსურ პროექტებში, ევროპული კოსმოსური სააგენტოს მიერ, სტანფორდის უნივერსიტეტის ერთ-ერთ პროექტში GLAST (The Gamma ray Large Area space Telescope) [20]. 10 წლის წინ Geant ორიენტირებული იყო ძირითადად მაღალი ენერგიების ფიზიკის ექსპერიმენტებზე. დღეს უკვე კარგად არის ადაპტირებული კოსმოსურ ექსპერიმენტებზეც. განსაკუთრებით, კოსმოსური მისიების საფრენი აპარატების და ზოგადი დანიშნულების ინსტრუმენტების სიმულაციისათვის.

Geant4 აქტიურად გამოიყენება ასევე მედიცინის მიმართულებით [15]. სამედიცინო სფეროში რამდენიმე კარგად ცნობილი სიმულაციის აპლიკაცია არსებობს, რომლებიც Geant4-ის ბაზაზე არია შექმნილები: G4DNA არის Geant4-ზე დაფუძნებული სიმულაციის აპლიკაცია რადიობიოლოგიის მიმართულებით. GAMOS (Geant4-based Architecture for Medicine-oriented Simulation) - ეს არის Geant4-ზე დაფუძნებული არქიტექტურა მედიცინაზე ორიენტირებული სიმულაციისთვის. GATE – არის პროგრამული უზრუნველყოფა რომელიც განკუთვნილია სამედიცინო გამოსახულებისა და სხივური თერაპიის სიმულაციისათვის. ის დაფუძნებულია Geant4-ზე. GHOST არის რესურსი, რომელიც გამოიყენება ონკოლოგიური დაავადებების სიმულაციისათვის. ისიც Geant4-ზეა დაფუძნებული.

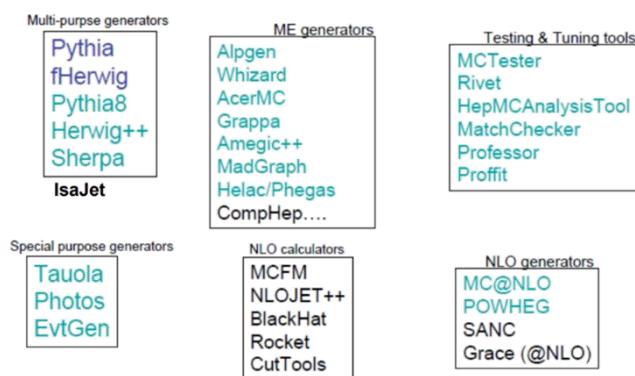
Geant4-ში მიმდინარეობს ATLAS-ის დეტექტორის სიმულაცია, გამოიყენება ორი სახის სრული და სწრაფი სიმულაცია.

სრული სიმულაცია ხორციელდება Geant4-ში და ის იყენებს დეტექტორის ზუსტ და დეტალურ გეომეტრიულ აღწერებს. სურ. 9-ზე წარმოდგენილია ატლასის სრული სიმულაციის სქემა და ის მოიცავს 5 ეტაპს ესენია: MC ივენთების გენერირება, დეტექტორის სიმულაცია, დიჯიტიზაცია, რეკონსტრუქცია და ფიზიკური ანალიზი.



სურ. 9 ატლასის სრული სიმულაციის სქემა

ატლასის სიმულაციის event გენერატორები სიმულაციის პირველ ეტაპზე მუშაობენ. მონტეკარლოს ალგორითმები - ივენთ გენერატორები სურ. 10 წარმოქმნიან ხელოვნურ ივენთებს. სულ ATLAS-ის სიმულაციაში 40-მდე ასეთი გენერატორი არსებობს [22]. წარმოქმნილი ივენთები ინახება EVGEN ფორმატში, რომელიც შემდეგ დეტექტორის სიმულაციაში

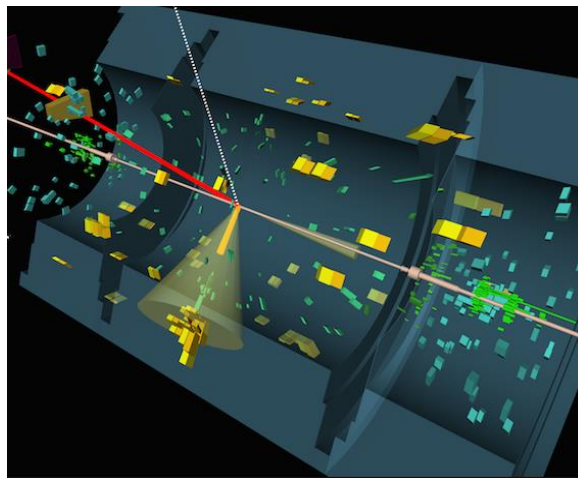


სურ. 10 ატლასის სრული სიმულაციის სქემა

გამოიყენება. მაგალითისათვის Pythia გენერატორის საშუალებით შესაძლებელია დადებითად და უარყოფითად დამუხტული ელექტრონების, ასევე პროტონის და ანტი პროტონის წარმოქმნა. fHerwing გენერატორის

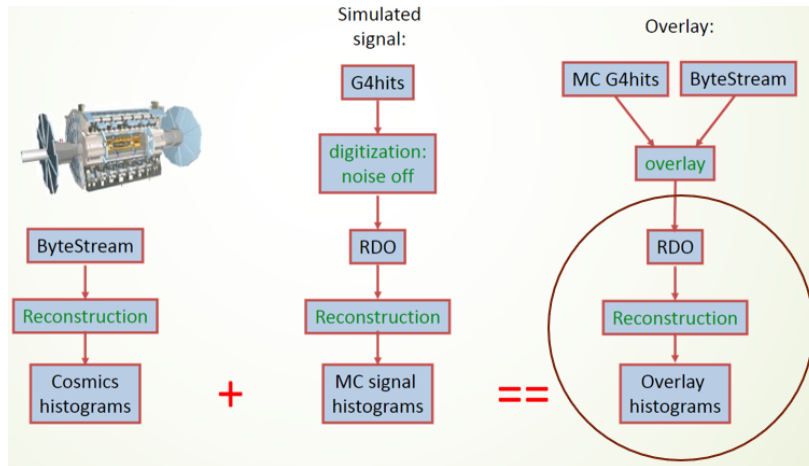
საშუალებით ადრონების გენერაცია ხორციელდება. ხოლო Herwing++ ის მეშვეობით W და Z ნაწილაკების წარმოქმნა.

შემდგომი ეტაპია დეტექტორის სიმულაცია. ATLAS-ის რეალური ექსპერიმენტში ნაწილაკის და დეტექტორის მგრძნობიარე სტრუქტურების ურთიერთქმედების შედეგად წარმოიქმნება ელექტრული იმპულსები [23][26]. ისინი გარდაიქმნება რიცხვით ByteStream ფორმატში. სიმულაციაში კი, დეტექტორის და ნაწილაკის ურთიერთქმედების შედეგად ელექტრული იმპულსის ნაცვლად იქმნება HITS-ები (სურ. 11). ამ დროს ხორციელდება



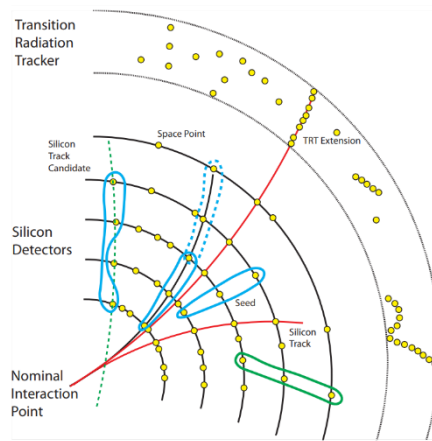
სურ. 11 Calorimeter-ში დაფიქსირებული ჰიტები ერთ-ერთი მოვლენისთვის ნაწილაკის საწყისი ენერგიის წარმოქმნა, გადახრის კუთხის და სხვა გეომეტრიული პარამეტრების შესახებ ინფორმაციები და იწერება HITS ფორმატში, ფაილის ზომა თითოეული მოვლენისთვის არის ასობით კილობაიტი.

HITS ფორმატი მხოლოდ სიმულაციაში არსებობს და აქვს ByteStream ფორმატისგან განსხვავებული ფორმატი, რადგან არ შეიცავს ინფორმაციას ფიზიკური პარამეტრების შესახებ, ამიტომ დღის წესრიგში დგება დიჯიტიზაციის ამოცანა (სურ. 12), ამ დროს ხდება HITS ფორმატის კონვერტაცია რიცხვით ფორმატში და ფიზიკური პარამეტრების ფორმირება. შედეგად იქმნება RDO (Row Data Objects) ფორმატის ინფორმაცია, რომელიც სრულად ByteStream - ის შესაბამისია.



სურ. 13 დიგიტიზაცია

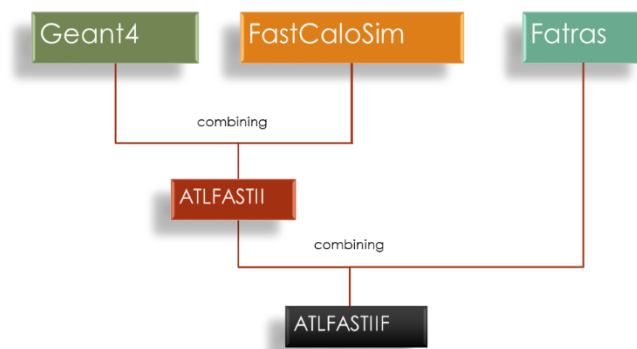
დიჯიტიზაციის შემდგომ აქტუალური ხდება რეკონსტრუქციის ამოცანა (სურ. 13). აქ მოცემულია ერთ-ერთი მოვლენის რეკონსტრუქციის



სურ. 12 რეკონსტრუქცია

შედეგები, კერძოდ თუ რა კვალი დატოვა ნაწილაკმა Silicon Detector-ში და TRT-ში გავლის შემდეგ. რეკონსტრუქციის ეტაპზე, შესაბამისი ალგორითმების მიერ, RDO ინფორმაციის დამუშავების შედეგად შესაძლებელი ხდება დადგინდეს ნაწილაკის ტრაექტორია, გაიზომოს მისი იმპულსი და ენერგია, ასევე შესაძლებელი ხდება ნაწილაკის იდენტიფიკაცია. რეკონსტრუქციის შედეგად ინფორმაცია იწერება ESD (Event Summary Data) ფორმატში. ეს ინფორმაცია ხანგძლივი შენახვისთვის არ გამოიყენება და ხდება ანალიზისთვის საჭირო AOD (Analysis Object Data) ფორმატის ფაილის ფორმირება. ეს ფაილი შეიცავს ყველა იმ ინფორმაციას რომელიც ფიზიკური ანალიზის ჩატარებისათვისაა საჭირო.

ATLAS-ის ექსპერიმენტის სიმულაციისთვის, გარდა სრული სიმულაციისა, ასევე გამოიყენება სწრაფი სიმულაცია. სრული სიმულაცია, სრულად Geant4-ში მიმდინარეობს. სწრაფ სიმულაციას კი თავის ძრავები აქვს ესენია [24]: FastCaloSim და Fatras. FastCaloSim გამოიყენება Calorimeter-ის სიმულაციისათვის, ხოლო Fatras კი Inner Detector-ის სიმულაციისათვის. FastCaloSim იყენებს პარამეტრიზებულ ინფორმაციას, რომელსაც იღებს სრული სიმულაციის შედეგებიდან, Fatras კი იყენებს ძალიან გამარტივებულ Inner Detector-ის გეომეტრიულ აღწერებს. შეიძლება სიმულაციის 3 ძრავის კომბინირება, Geant4 და FastCaloSim-ის კომბინაციას ეწოდება ATLFASTII. ATLFASTII ფრეიმვორკის გამოყენებისას სიმულაციის პროცესი საგრძნობლად ჩქარდება, Geant4-ს 10 მოვლენის დასამუშავებლად სჭირდება 560 წმ, ხოლო ATLFASTII-ის შემთხვევაში კი 100 მოვლენის დამუშავებას 25 წმ სჭირდება [25]. შესაძლებელია ასევე ATLFASTII და Fatras-ის კომბინაცია, მას ATLFASTIIF-ს უწოდებენ. ასეთი კომბინაციებით მიიღება ინტეგრირებული სიმულაციის ფრეიმვორკი (ISF) სურ.14. ATLFASTIIF ფრეიმვორკი არის ძალიან სწრაფი. 1000 მოვლენის დამუშავებას ანდომებს 0,75 წმ-ს.



სურ. 14 ინტეგრირებული სიმულაციის ფრეიმვორკი (ISF)

ინტეგრირებული სიმულაციის ფრეიმვორკი მუშაობს შემდეგნაირად: სრული სიმულაცია ყველა დეტექტორისთვის მიმდინარეობს Geant4-ში. ATLFASTII ფრეიმვორკის გამოყენების დროს Inner Detector-ის სიმულაცია მიმდინარეობს Geant4-ში, Calorimeter-ების შემთხვევაში muons-ის ნაწილაკის სიმულაცია Geant4-ში, ხოლო სხვა

დანარჩენის FastCaloSim-ში მიმდინარეობს. Muon Spectrometer-ის სიმულაცია, ხორციელდება Geant4-ში. ATLFASIIIF ფრეიმვორკის გამოყენების შემთხვევაში, Inner Detector-ის სიმულაცია მიმდინარეობს Fatras-ში, Calorimeter-ების - FastCaloSim-ში, გარდა muons-ის ნაწილაკისა. მისი სიმულაცია ხდება Fatras-ში. Muon Spectrometer-ის სიმულაცია კი ხდება

- **Full Geant4**
G4 –ში ყველა დეტექტორი
- **ATLFASII**
ID: Geant4
Calo: Geant4 muons-ისთვის, FastCaloSim დანარჩენისთვის
MS: Geant4
- **ATLFASIIIF**
ID: Fatras
Calo: Fatras muons -ისთვის, FastCaloSim დანარჩენისთვის
MS: Fatras

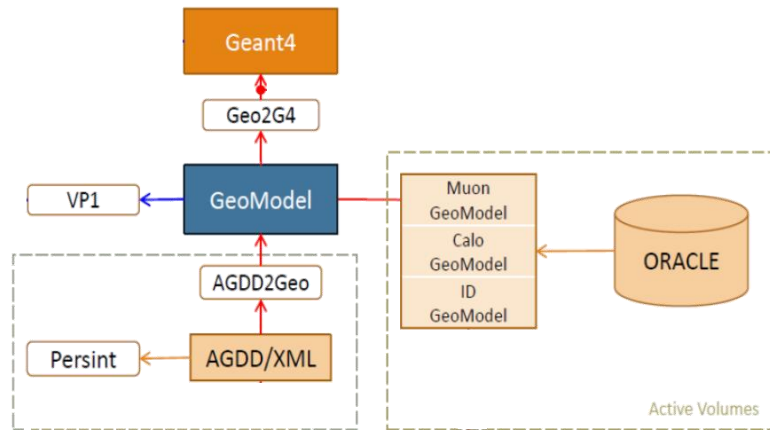
სურ. 15 ISF-ის მუშაობის პრინციპი

Fatras-ში. ეს პროცესი სქემატურად შემდეგნაირად გამოიყურება (სურ.15)

1.4 ATLAS-ის სიმულაციის ინფრასტრუქტურა GEANT4-ის ბაზაზე

Geant4 სიმულაციისთვის იყენებს ATLAS-ის დეტექტორის გეომეტრიულ აღწერებს. ისინი 2 ნაწილადაა დაყოფილი. 1-ლი, სტრუქტურები რომლებიც უშუალოდ მონაწილეობენ ნაწილაკის აღმოჩენაში, მათ აქტიური კომპონენტები ეწოდებათ და მე-2 დეტექტორის სხვა დანარჩენი დამხმარე სტრუქტურები, როგორცაა: სერვისები, პლატფორმები, და ა.შ. მათ პასიურ კომპონენტებს უწოდებენ. ეს სტრუქტურები აუცილებელია იმისათვის, რომ მუშაობა შეძლონ აქტიურმა კომპონენტებმა.

ATLAS-ის დეტექტორის პასიური კომპონენტები აღწერილია AGDD (ATLAS Generic Detector Description) ენაში და შენახულია XML ფორმატში [27] [28]. ის რამდენიმე ასეულ კომპონენტს შეიცავს, ხოლო რაც შეეხება აქტიურ კომპონენტებს ის დაყოფილია სამ ნაწილად: Muon Spectrometer, Calorimeter, Inner Detector და აღწერილია C++ ენის გამოყენებით GeoModel-ში სურ.16.



სურ. 16 გეომეტრიული აღწერების სიმულაციის მიწოდების ჯაჭვი

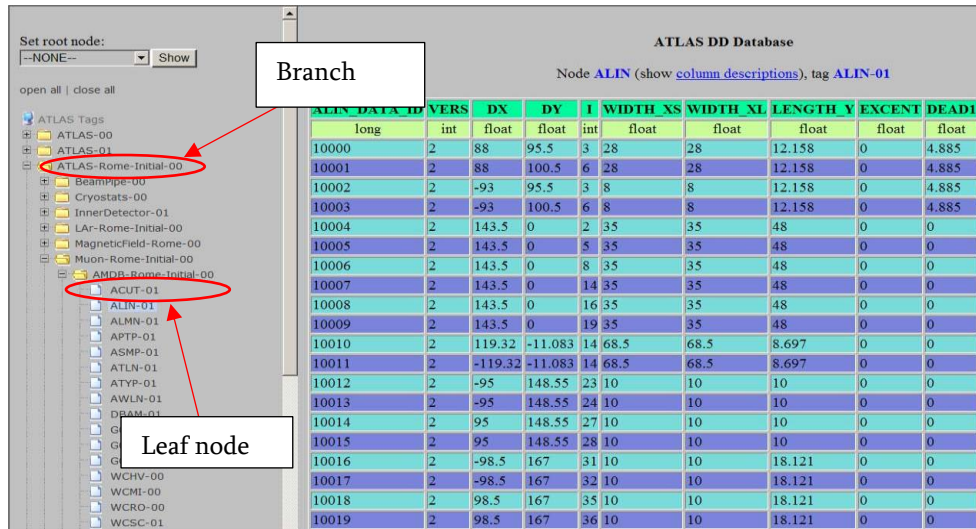
GeoModel არის ინსტრუმენტი და ასევე ბიბლიოთეკა, რომელიც შექმნილია სპეციალურად დეტექტორის გეომეტრიული აღწერების შესაქმნელად [31]. GeoModel-ში აღწერილია, მხოლოდ სემანტიკური მხარე, რაც შეეხება რიცხვით მნიშვნელობებს, ისინი განთავსებულია Oracle-ის მონაცემთა ბაზაში. ის არის რელაციური მონაცემთა ბაზა. რელაციური მოდელი გულისხმობს იმას, რომ ცხრილში განთავსდება ინფორმაცია, მხოლოდ ერთი ლოგიკური ობიექტის შესახებ და ეს ობიექტები ერთმანეთთან დაკავშირებულები იქნებიან ამ ცხრილშივე განთავსებული ინფორმაციის საშუალებით. დეტექტორის აქტიური სტრუქტურები შედგება დაახლოებით 1021500 კომპონენტისგან ცხრ.1 [13].

detector	number of volumes	comments
Pixel	6000	
SCT	40500	
TRT	300000	parameterized
LAr	142500	parameterized
Tile	80500	parameterized
Muon Chambers	451000	parameterized
Toroids	1000	

ცხრ. 1 GeoModel-ში აღწერილი კომპონენტების რაოდენობა

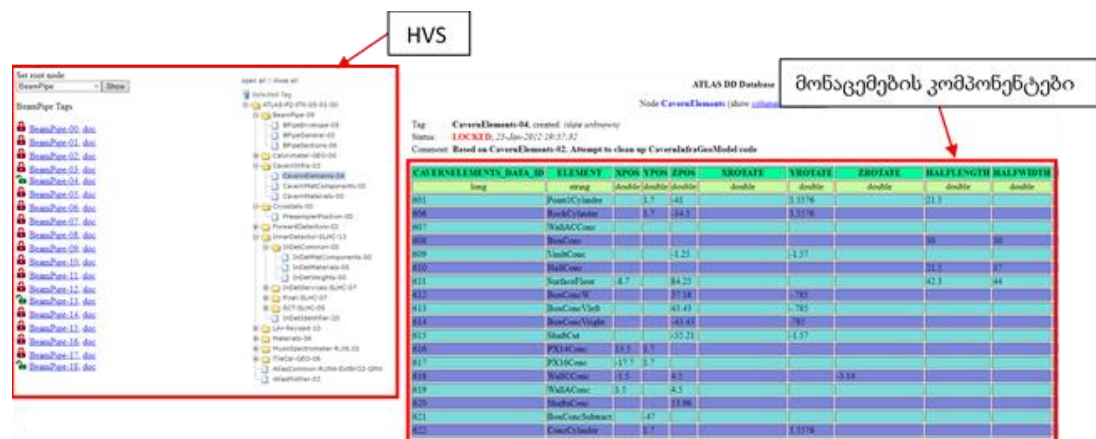
ATLAS-ის დეტექტორის აქტიური კომპონენტების მონაცემთა ბაზა 1 სქემისგან შედგება. ის შეიცავს 900 ცხრილს და 5 მილიონ ხაზს. ინფორმაციის მთლიანი მოცულობაა 0.1 გბ. ATLAS-ის დეტექტორის გეომეტრიის რიცხვითი პარამეტრები დაჯგუფებულია ATLAS-ის დეტექტორის

ქვესისტემებში და სტრუქტურებში [32]. სტრუქტურები კი ლოგიკურად დაჯგუფებულია დირექტორიებში. თითოეული დირექტორია შეიცავს რამდენიმე სტრუქტურას. ასე იქმნება გარკვეული იერარქია. იერარქიულ სისტემებში (HVS - Hierarchical Versioning System) დირექტორიებს ეწოდებათ „Branch node“, ხოლო სტრუქტურებს კი „Leaf nodes“. ორივე იწოდება როგორც



სურ. 17 ATLAS-ის დეტექტორის აქტიური კომპონენტების მონაცემთა ბაზა ტაგი. სურ. 17-ზე მარცხენა მხარეს წარმოდგენილია Branch და Leaf node-ები. ხოლო სურ. 17-ის მარჯვენა მხარეს წარმოდგენილია, მონიშნულ Leaf node-ში არსებული მონაცემები.

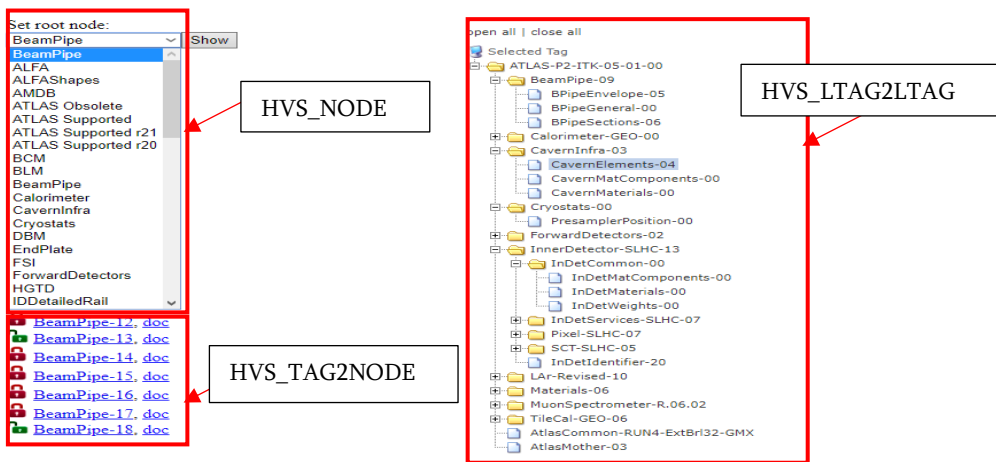
ATLAS-ის გეომეტრიის აქტიური სტრუქტურების მონაცემთა ბაზა 2 ნაწილად არის დაყოფილი სურ.18 [10], პირველი ეს არის იერარქიული



სურ.18 აქტიური კომპონენტების მონაცემთა ბაზის სტრუქტურა სისტემების კომპონენტები (HVS), რომლებიც ახორციელებენ დეტექტორის

რიცხვითი პარამეტრების ლოგიკურ კავშირებს HVS node-ებში და მეორე მონაცემების კომპონენტები, რომლებიც შეიცავენ დეტექტორის გეომეტრიული აღწერების რიცხვით პარამეტრებს. თავის მხრივ HVS კომპონენტები 3 ცხრილისგან შედგება:

1. HVS_NODE
2. HVS_TAG2NODE
3. HVS_LTAG2LTAG



სურ. 19 HVS-ის კომპონენტები

HVS_NODE (სურ.19) ცხრილი შეიცავს ინფორმაციას მონაცემთა ბაზაში არსებული ყველა node-ის შესახებ და ასევე დეტექტორის გეომეტრიული სტრუქტურების იერარქიული კავშირის შესახებ ინფორმაციას. HVS_TAG2NODE ცხრილი შეიცავს ინფორმაციას ტაგების შესახებ თითოეული HVS NODE-ისთვის და HVS_LTAG2LTAG ცხრილები შეიცავენ ინფორმაციას ტაგებს შორის იერარქიის შესახებ.

რაც შეეხება Leaf Node-ს ის 2 ცხრილისგან შედგება პირველი ეს არის მონაცემთა ცხრილი (სურ.20) სადაც თავმოყრილია ყველა რიცხვითი

Tag: CavernElements-04, created: (raze unstrony)
 Status: LOCKED, 25-Jan-2012 19:57:32
 Comment: Based on CavernElements-02. Attempt to clean up CavernInfraGeoModel code

მონაცემთა ცხრილი

CAVERNLEMENTS_DATA_ID	ELEMENT	XPOS	YPOS	ZPOS	XROTATE	YROTATE	ZROTATE	HALFLENGTH	HALFWIDTH
	long	string	double	double	double	double	double	double	double
01	Point1Cylinder		1.7	-41		1.5576		21.5	
06	RockCylinder		1.7	-34.5		1.5576			
07	WallACConc								
08	BoxConc							30	30
09	VaultConc			-1.25		-1.57			
10	HallConc							21.5	17
11	SurfaceFloor		-8.7	84.25				42.3	44
12	BoxConcW			37.18		-0.785			
13	BoxConcVleft			43.43		-0.785			

სურ. 20 Leaf Node-ის კომპონენტები

პარამეტრი და მეორე ცხრილი სადაც არის ინფორმაცია მონაცემთა ცხრილებსა და HVS-ის კომპონენტებს შორის კავშირების შესახებ.

```

#include "TileGeoModel/TileAtlasFactory.h"
#include "TileDetDescr/TileDetDescrManager.h"
#include "TileDetDescr/TileDetDescriptor.h"
#include "TileGeoModel/TileGeoSectionBuilder.h"
#include "TileGeoModel/TileGeoCutBuilder.h"
#include <stdexcept>
#include <iostream>
#include <iomanip>
#include <algorithm>
#include <cmath>
#include <string>

using namespace Genfun;
using namespace GeoXF;

// Constructor:
TileAtlasFactory::TileAtlasFactory(StoreGateSvc *pDetStore,
                                   TileDetDescrManager *manager,
                                   bool addPlates,
                                   MagStream *log,
                                   bool fullGeo)
: detectorStore(pDetStore)
, detectorManager(manager)
, m_log(log)
, m_addPlatesToCellVolume(addPlates)
, m_testBeamGeometry(false)
, m_verbose(log->level() < MSG::VERBOSE)
, m_fullGeo(fullGeo)

// Destructor:
TileAtlasFactory::~TileAtlasFactory() {}

// Creation of geometry:
void TileAtlasFactory::create(GeoPhysVol *world)

```

მონაცემთა ბაზასთან კავშირის ბიბლიოთეკა

სურ. 21 Tile Calorimeter-ის GeoModel-ის კოდის ფრაგმენტი

Oracle-ის მონაცემთა ბაზებში თავმოყრილი დეტექტორის გეომეტრიული აღწერების რიცხვითი პარამეტრების ინტერპრეტაცია ხორციელდება GeoModel-ის საშუალებით. GeoModel არის ინსტრუმენტი და ასევე ბიბლიოთეკა, რომელიც შექმნილია სპეციალურად დეტექტორის გეომეტრიული აღწერების შესაქმნელად. GeoModel-ში აღწერილია, მხოლოდ სემანტიკური მხარე (სურ 21). მონაცემთა ბაზასთან კავშირისთვის

```

// nominal position of end plate
zEndSection = ZLength[1]/2 - BFingerLength;

GeoTubs* GeneralMother = new GeoTubs(dbManager->GetEnvRin()*cm,
                                     dbManager->GetEnvRout()*cm,
                                     zEndSection,
                                     AnglMin, AnglMax);

```

სურ. 22 Tile Calorimeter-ის ერთ-ერთი კომპონენტის GeoModel-ის კოდის ფრაგმენტი გამოიყენება ფუნქცია, რომელიც აღწერილია TileDetDescrManager-ში მაგალითისთვის განვიხილოთ Tile Calorimeter-ის ერთ ერთი კომპონენტი TileGlobals. სურ. 22-ზე ნაჩვენებია მისი GeoModel-ის კოდი. მონიშნულია ის ადგილი სადაც რიცხვითი პარამეტრის მისაღებად მიმართვა ხდება მონაცემთა ბაზისათვის. ამ დროს აქტიურდება dbmanager ფუნქცია, რომელიც იძახებს GetEnvRin და GetEnvRout ფუნქციას (სურ. 22). სურ.23-ზე

მონიშნულია GetEnvRin და GetEnvRout ფუნქციების შესაბამისი RIN და ROUT რიცხვითი პარამეტრები მონაცემთა ბაზაში.

ATLAS-ის დეტექტორის გეომეტრიული აღწერების შესაქმნელად GeoModel იყენებს GeoModelKernel კლასს [33][34]. სადაც აღწერილია გეომეტრიის ყველა ის კლასი რომელიც საჭიროა პრიმიტივებია დასაპროგრამებლად. GeoModel-ში დეტექტორის გეომეტრიის მასალის მისანიჭებლად გამოიყენება 2 კლასი: GeoElement და GeoMaterial. GeoElement კლასში აღწერილია ქიმიური ელემენტები, სახელის, ქიმიური სიმბოლოს, სიმკვრივის, მისი ატომური ნომერის და წონის გათვალისწინებით თვალსაჩინოებისათვის ქვემოთ წამოდგენილია წყალბადის და ჟანგბადის კოდების GeoModel-ში:

```
GeoElement *hydrogen = new GeoElement("Hydrogen", "H", 1.0, 1.010);
```

```
GeoElement *oxygen = new GeoElement("Oxygen", "O", 8.0, 16.0);
```

რაც შეეხება GeoMaterial-ს, ის არის კლასი, სადაც თავმოყრილია ქიმიური ელემენტების სია, რომლებიც შედგენილი მატერიალების შესაქმნელად გამოიყენება მაგალითად განვიხილოთ წყალი ქვემოთ მოცემული მისი კოდი სურ. 24:

```
double TileDddbManager::GetEnvRin() const
{
    if (currentTileGlob) {
        return currentTileGlob->getDouble("RIN");
    }
    else {
        std::cerr << "\nTileDddbManager ERROR! Current Envelope not set, returning -999\n";
        return -999;
    }
}

double TileDddbManager::GetEnvRout() const
{
    if (currentTileGlob) {
        return currentTileGlob->getDouble("ROUT");
    }
    else {
        std::cerr << "\nTileDddbManager ERROR! Current Envelope not set, returning -999\n";
        return -999;
    }
}
```

სურ. 23 Tile Calorimeter-ის ერთ-ერთი კომპონენტის GeoModel-ის კოდის ფრაგმენტი

```
water->add(hydrogen,0.11);
```

TILEGLOBALS_DATA_ID	PART	NAM	SIDE	NMODULES	RIN	ROUT	ZLENGTH	DX	DY	DZ	DPHI	DPSI	DTHETA	ZSHIFT
0	1	LB	0	64	228.8	425	650.1	0	0	0	0	0	0	0
1	2	EBC	-1	64	228.8	425	296.75	0	0	0	0	0	0	4
2	3	EBA	1	64	228.8	425	296.75	0	0	0	0	0	0	4
3	4	ITCC	-1	64	146.5	425	30.9	0	0	0	0	0	0	4
4	5	ITCA	1	64	146.5	425	30.9	0	0	0	0	0	0	4

სურ. 24 TileGlobals -ის მონაცემთა ბაზის ფრაგმენტი

```
water->add(oxygen,0.89);
water->lock();
```

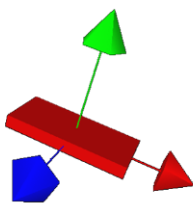
add მეთოდი უზრუნველყოფს ელემენტის დამატებას და მის წილს ამა თუ იმ მასალაში, კოდიდან ჩანს, რომ, რომ წყლის მისაღებად 0.11 ნაწილი წყალბადი და 0.89 ნაწილი ჟანგბადი გამოიყენება.

GeoBox	Box
GeoCons	Cone Section
GeoPara	Parallelepiped
GeoPcon	Polycone
GeoPgon	Polygon
GeoTrap	Trapezoid (complex)
GeoTrd	Trapezoid (simple)
GeoTube	Tube
GeoTubs	Tube Section

ცხრ. 2 GeoModel-ში არსებული პრიმიტივები

პრიმიტივების აღწერისათვის GeoModelKernel-ში გამოიყენება Shapes კლასი. აქ თავმოყრილია პრიმიტივების აღწერის ყველა მეთოდი. ცხრ.2-ში წარმოდგენილია ყველა ის პრიმიტივი რომლის აღწერაც GeoModel-ში არის შესაძლებელი.

GeoBox მეთოდი გამოიყენება პრიზმის შესაქმნელად (სურ.25) და მისი სემანტიკა გამოიყურება შემდეგნაირად:



სურ. 25 Box პრიმიტივი GeoModel-ში

GeoBox (double XHalfLength, double YHalfLength, double ZHalfLength)

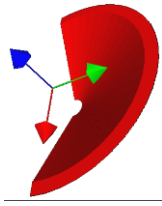
სადაც მას მიეწოდება განახევრებული რიცხვითი პარამეტრები სიგრძე, სიმაღლე და სიგანე

შესაბამისად X,Y და Z ღერძებზე. GeoModel-ში ნებისმიერი პრიმიტივის ასაგებად, გამოიყენება

განახევრებული ზომების რიცხვითი პარამეტრები.

შემდეგი პრიმიტივია GeoCons - კონუსი (სურ.26). GeoModel-ში მისი სემანტიკური მხარე შემდეგნაირია:

GeoCons (double RMin1, double RMin2,
double RMax1, double RMax2,
double DZ, double SPhi, double DPhi)



სურ. 26 GeoCons
პრიმიტივი GeoModel-ში
აპროქსიმაცია და ა.შ.

კონუსის მისაღებად, კონსტრუქტორს მიეწოდებათ შემდეგი პარამეტრები: RMin1, RMin2, RMax1, RMax2, DZ, SPhi, DPhi. სადაც RMin1, RMin2, RMax1, RMax2 არის შიდა და გარე რადიუსები, DZ არის სიმაღლე. SPhi, DPhi დამატებითი პარამეტრებია, როგორცაა

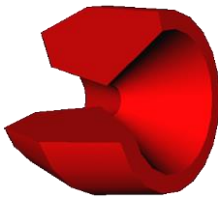
გეომეტრიის აღწერის შემდეგი მეთოდია GeoPara - პარალელებიპედი.

მისი კოდის სტრუქტურა შემდეგნაირია:

```
GeoPara (double XHalfLength, double YHalfLength, double ZHalfLength,
         double Alpha, double Theta, double Phi)
```

მას მიენიჭება შემდეგი რიცხვითი პარამეტრები: სიგრძე, სიგანე, სიმაღლე და დახრის კუთხეები.

კიდევ ერთი პრიმიტივი, რომელიც GeoModel-ში გამოიყენება არის



სურ. 27 GeoPcon
პრიმიტივი GeoModel-ში

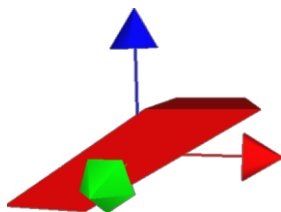
GeoPcon (სურ.27). GeoPcon-ის სემანტიკური მხარე GeoModel-ში შემდეგნაირად გამოიყურება:

```
GeoPcon (double SPhi, double DPhi, double getZPlane, double
         getRMinPlane, double getRMaxPlane, )
```

ობიექტის ასაგებად უნდა მიეთითოს კუთხეები (SPhi- DPhi), თუ რა არეალში შეიქმნას ობიექტი და ასევე მიეწოდება Z0-იდან კონკრეტულ

მნიშვნელობაზე შიდა და გარე რადიუსები.

შემდეგი პრიმიტივი GeoModel-ში არის GeoPgon. ის არის GeoPcon ის



სურ. 28 GeoTrap პრიმიტივი
GeoModel-ში

მსგავსი მეთოდი და კოდის სტრუქტურაც მსგავსი:

```
GeoPgon (double SPhi, double DPhi, unsigned int NSides)
```

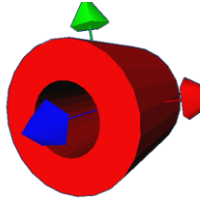
GeoModel-ში გამოიყენება ასევე GeoTrap - ტრაპეცია (სურ. 28) მისი GeoModel-ის კოდი

შემდეგნაირია:

```
GeoTrap (double ZHalfLength, double Theta, double Phi, double Dydzn, double
         Dxdyndzn, double Dxdypdzn, double Angleydzn, double Dydzp, double
         Dxdyndzp, double Dxdypdzp, double Angleydzp)
```

ტრაპეციის ფორმის ობიექტის ასაგებად შეიძლება მიწოდებულ იქნეს სხვადასხვა პარამეტრები როგორცაა: სიგრძე, კუთხეების სხვადასხვა ღერძების მიმართ და. ა.შ

შემდეგი არის GeoTube - ცილინდრი (სურ.29). მისი სემანტიკა შემდეგნაირია:

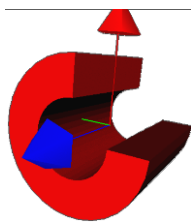


სურ. 29 GeoTube პრიმიტივი GeoModel-ში

GeoTube (double RMin, double RMax, double ZHalfLength)
 ობიექტის შესაქმნელად საჭიროა შიდა და გარე რადიუსების და სიმაღლის პარამეტრების მიწოდება და ბოლო არის GeoTubs - ამოჭრილი ცილინდრი (სურ. 30). მისი კოდი არის შემდეგნაირი:

GeoTubs (double RMin, double RMax, double ZHalfLength, double SPhi, double DPhi)

ობიექტის ასაგებად მეთოდ უნა მიეწოდოს შემდეგი პარამეტრები: კუთხეები



სურ. 30 GeoTubs პრიმიტივი GeoModel-ში

(SPhi- DPhi) თუ რა არეალში უნდა აიგოს გეომეტრია. ასევე შიდა და გარე (RMin -RMax) რადიუსები და სიმაღლე (ZHalfLength).

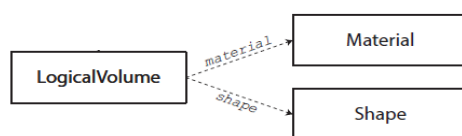
რთული გეომეტრიული

დასაპროგრამებლად საჭიროა Boolean ოპერაციები. ამისათვის გამოიყენება GeoShape

კლასში შემავალი რამდენიმე მეთოდი. ქვემოთ მოცემულია ობიექტში სხვა ობიექტის დამატების და ამოჭრის მაგალითი:

```
GeoShape *A, *B, *C;
const GeoShape & D = (*A). add((*B). subtract (*C));
```

აქ ხდება A ობიექტისათვის B ობიექტის დამატება და მიღებულიდან C ობიექტის ამოჭრა. GeoModel-ში ობიექტის საბოლოო პოზიციონირებისათვის გამოიყენება გადაადგილების ტრანზაქციები.



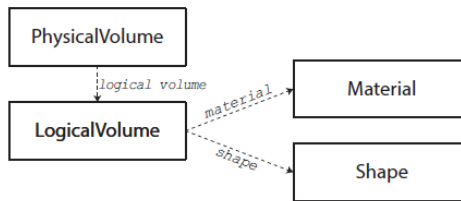
სურ. 31 Logical Volume-ის სტრუქტურა GeoModel-ში

მას შემდეგ რაც განხორციელდება ობიექტის შექმნა GeoModel-ში, ის უნდა განთავსდეს Logical Volume-ში (სურ.31). სადაც მას მიენიჭება სახელი და მასალა. მისი სემანტიკა შემდეგნაირად გამოიყურება:

```
GeoLogVol (const std::string & Name, const GeoShape * Shape, const GeoMaterial * Material)
```

საბოლოოდ Logical Volume უნდა განთავსდეს Physical Volume-ში (სურ.32). GeoModel-ში ის შემდეგნაირად გამოიყურება:

```
GeoPhysVol (const GeoLogVol * LogVol)
```

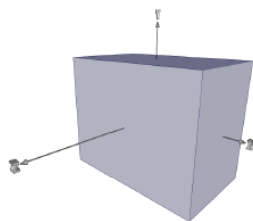


სურ. 32 Physical Volume -ის სტრუქტურა GeoModel-ში

გეომეტრიის ასაგებად სხვადასხვა მეთოდებია გამოყენებული, როგორცაა: cube, tube, pyramid, cylinder, chain, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized [29]. განვიხილოთ თითოეული მათგანი.

Cube (სურ. 33) - ამ მეთოდის სემანტიკა არის შემდეგნაირი:

```
<box name="Box" material="Aluminium" X_Y_Z="2500.; 2000.; 1500."/>
```



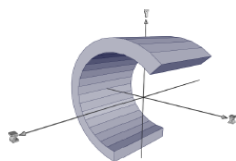
სურ. 33 დაპროგრამების მეთოდი cube

ამ მეთოდით ხდება სტანდარტული გეომეტრიული პრიმიტივის - პრიზმის დაპროგრამება, X,Y და Z არის პრიზმის სიგრძე, სიგანე და

სიმაღლე, კოორდინატა სისტემის

შესაბამისად. Material-ში ენიჭება მასალა, ხოლო box არის ობიექტის სახელი.

Tube (სურ. 34) - ამ მეთოდის სემანტიკა არის შემდეგნაირი:



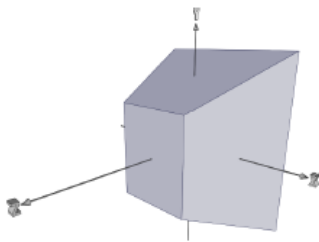
სურ. 34 დაპროგრამების მეთოდი Tube

```
<tubs name="Tube" material="Iron"
Rio_Z="1000.; 1250.; 1500." profile="45.;225."
nbPhi="20"/>
```


ამ მეთოდით ხდება სტანდარტული გეომეტრიული პრიმიტივის - ცილინდრის დაპროგრამება. „Tube“ არის ობიექტის სახელი. Material-ში ხდება მასალის მინიჭება. Rio_Z-ში პირველი პარამეტრი არის შიდა რადიუსი, მეორე გარე რადიუსი და მესამე კი ცილინდრის სიმაღლე. Profile-ში ხდება ისეთი დამატებითი პარამეტრების მინიჭება, როგორცაა ცილინდრის აგების დაწყების და დასრულების კუთხეები. მაგალითად profile="45.;225 ნიშნავს, რომ ცილინდრის აგება დაიწყო 45 გრადუსზე და დასრულდეს 225 გრადუსზე. nbPhi-ში კი მოცემულია აპროქსიმაციის პარამეტრები. nbPhi="20" ნიშნავს, რომ ცილინდრის ფუძის წრეწირი აიგება 20 წერტილზე.

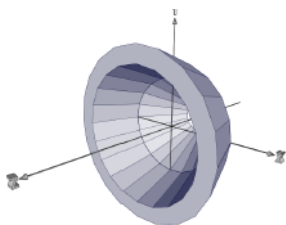
Pyramid (სურ. 35) - ამ მეთოდის სემანტიკა არის შემდეგნაირი:

```
<trd name="Pyramid" material="Aluminium" Xmp_Ymp_Z="2000.; 1000.; 2500.; 1250.; 1500.0" />
```



სურ. 36 დაპროგრამების მეთოდი Pyramid

ამ მეთოდით ხდება სტანდარტული გეომეტრიული პრიმიტივის - პირამიდის დაპროგრამება. "Pyramid" არის ობიექტის სახელი. Material-ში ხდება მასალის მინიჭება. Xmp_Ymp_Z-ში 1-ლი და მე-3 პარამეტრი არის პირამიდის ერთი ფუძის სიგრძე და სიგანე, ხოლო მე-2 და მე-4 პარამეტრი კი მე-2 ფუძის სიგრძე და სიგანე. მე-5 რიცხვითი პარამეტრი კი პირამიდის სიმაღლეა.



სურ. 35 დაპროგრამების მეთოდი Cylinder

Cylinder - ამ მეთოდით შეიძლება დაპროგრამდეს, როგორც სტანდარტული გეომეტრიული პრიმიტივები კონუსი და სფერო, ასევე არასტანდარტული ობიექტებიც. სურ. 36. ამ მეთოდის სემანტიკა შემდეგნაირია:

```
<pcon name="Cone" material="PolyBoron5percent" nbPhi="20">
```



```

<polyplane Rio_Z="1000.; 1250.; 750."/>
<polyplane Rio_Z=" 750.; 1000.; 0."/>
<polyplane Rio_Z=" 250.; 500.; -750."/>
</pcon>

```

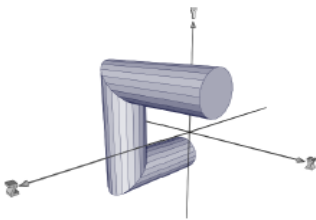
„Cone“ არის ობიექტის სახელი. Material კი მისი მასალაა. NbPhi კი არის აპროქსიმაციის პარამეტრი. polyplane Rio_Z -ში ხდება ობიექტის რიცხვითი პარამეტრების მინიჭება. 1-ლი პარამეტრი არის შიდა რადიუსი, მე-2 პარამეტრი კი გარე რადიუსი. მე-3 პარამეტრი ობიექტის სიმაღლეა საკოორდინატო ცენტრიდან.

Chain - მეთოდით აგებული ობიექტი არ არის სტანდარტული გეომეტრიული ობიექტი. მისი საშუალებით ხდება ცილინდრული ობიექტის შექმნა. სურ.37. მისი სემანტიკა შემდეგნაირია:

```

<snake name="Chain" material="Brass" radius="300." nbPhi="20">
<snake_point X_Y_Z="-750.; -750.; -750." />
<snake_point X_Y_Z="-750.; -750.; 500." />
<snake_point X_Y_Z="-750.; 750.; 500." />
<snake_point X_Y_Z=" 750.; 750.; 500." />
</snake>

```



სურ. 37 დაპროგრამების მეთოდი Cylinder

"Chain" არის ობიექტის სახელი. მატერიალში კი მასალაა მინიჭებული. Radius-ში მითითებულია ცილინდრული ობიექტის რადიუსი. NbPhi კი აპროქსიმაციის პარამეტრია. snake_point X_Y_Z-ში კი წერტილის X, Y და Z კოორდინატია ჩაწერილი.

ამ მაგალითში მოცემულია 4 წერტილის კოორდინატი რაც ნიშნავს იმას, რომ chain-ის აგება მოხდება 4 წერტილზე შექმნილ პოლიგონზე.

Arbitrary polygon (სურ. 38) არც ამ მეთოდით აგებული ობიექტი არის სტანდარტული გეომეტრიული ობიექტი.

ამ მეთოდის სემანტიკა არის ასეთი:

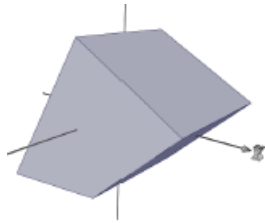
```

<gvxy name="Polygon" material="Copper" dZ="1500.">
<gvxy_point X_Y="-1250.;-1000."/>
<gvxy_point X_Y="0.;1000."/>
<gvxy_point X_Y="1250.;250."/>
<gvxy_point X_Y="250.;-750."/>

```

</gvxy>

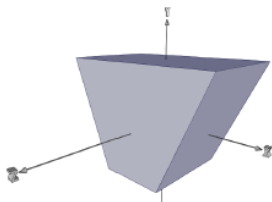
აღნიშნული მეთოდით ობიექტის შექმნა ხდება შემდეგნაირად, gvxy_point



სურ. 38 დაპროგრამების მეთოდი arbitrary polygon

X_Y-ში იწერება წერტილის x და y კოორდინატი სიბრტყეზე. შემდეგ ამ წერტილებზე უნდა შეიკრას პოლიგონი, ამისთვის საჭიროა მინიმუმ 3 წერტილი და ბოლოს ამ პოლიგონს ენიჭება „სისქე“ dZ -ში მითითებული რიცხვითი პარამეტრის შესაბამისად.

Symmetric polygon (სურ. 39) – აღნიშნული მეთოდით ხორციელდება



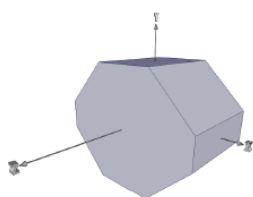
სურ. 39 დაპროგრამების მეთოდი Symmetric polygon

არასტანდარტული პრიმიტივების შექმნა. symmetric polygon-ის სემანტიკა გამოიყურება ასე:

```
<gvxysx name="SymmetricPolygon" material="Carbon"
dZ="1500.">
<gvxy_point X_Y="-1250.;1000."/>
<gvxy_point X_Y=" -500.;-750."/>
</gvxysx>
```

ობიექტის ფუძის აგება ხდება შემდეგნაირად: თავდაპირველად იგება 2 წერტილი. gvxy_point X_Y-ში იწერება თითოეული წერტილის X და Y კოორდინატი და შემდეგ ავტომატურად ხდება მათი სიმეტრიული წერტილების შექმნა. საბოლოოდ ამ წერტილებზე იგება პოლიგონი და შემდეგ arbitrary-ს მსგავსად მიენიჭება სისქე “dZ“-ში მითითებული რიცხვითი პარამეტრის შესაბამისად. ამ მეთოდით შესაძლებელია 4 კუთხა პრიზმის შექმნა.

Double symmetric polygon (სურ. 40) ამ მეთოდით ხდება მრავალკუთხა პრიზმის დაპროგრამება. მეთოდის სემანტიკა ასეთია:



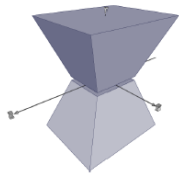
სურ. 40 დაპროგრამების მეთოდი Double Symmetric polygon

```
<gvxysxy name="DoubleSymmetricPolygon"
material="Lead" dZ="1500.">
<gvxy_point X_Y="-1250.; 250."/>
<gvxy_point X_Y=" -500.;1000."/>
</gvxysxy>
```

როგორც ვიცით კოორდინატთა სისტემა დაყოფილია 4 ნაწილად. პრიზმის ფუძის

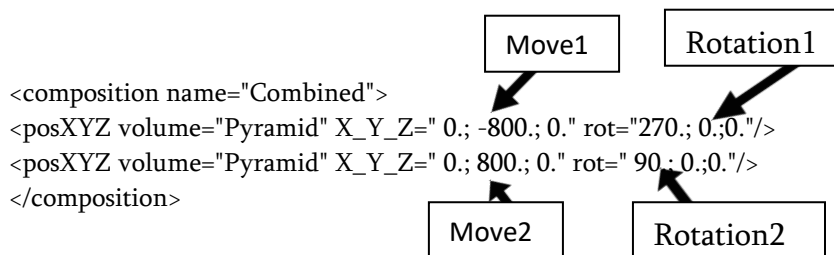
აგება ხდება შემდეგნაირად. წერტილის x და y კოორდინატების იწერება gvxy_point X_Y-ში. კოორდინატთა სისტემის მხოლოდ 1/4-ში და ნარჩენ მეოთხედებში ისინი სიმეტრიულად განთავსდებიან, შემდეგ იკვრება ამ წერტილებზე პოლიგონი და ბოლოს კი arbitrary-ის და symmetric-ის მსგავსად ხდება მოცულობის მინიჭება.

combined (სურ. 41) ეს მეთოდი გამოიყენება ობიექტების



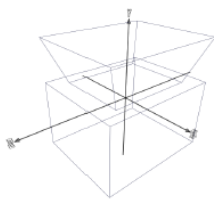
სურ. 41 დაპროგრამების მეთოდი combined

გასაერთიანებლად. ის საშუალებას გვაძლევს რამდენიმე პრიმიტივი გავაერთიანოთ ერთი სახელის ქვეშ, თუმცა ისინი სიმულაციაში რჩებიან დამოუკიდებელ ობიექტებად. მისი სემანტიკა ასე გამოიყურება:



როგორც აქ ჩანს ხდება 2 პირამიდის გაერთიანება და ასევე მათი გადაადგილებაც არის შესაძლებელი. ეს შესაძლებელია განხორციელდეს ფუნქციაში Move1, Rotation1, Move2, Rotation2 პარამეტრებით.

Merged (სურ. 42) ეს მეთოდი წინა მეთოდის მსგავსია, აქაც ხდება

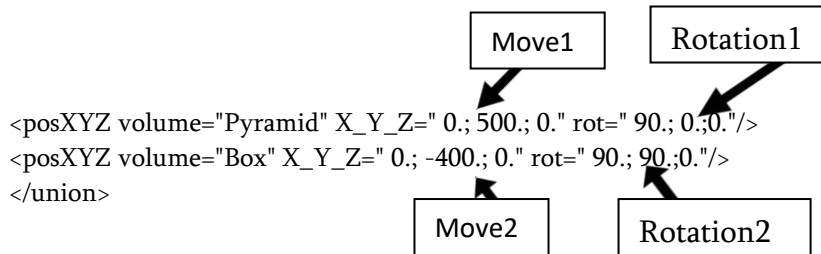


სურ. 42 დაპროგრამების მეთოდი merged

პრიმიტივების გაერთიანება, თუმცა მთავარი განსხვავება არის ის, რომ საბოლოოდ მიიღება 1 ობიექტი boolean პროცესორის საშუალებით. ამ დროს შესაძლებელია გამოყენებულ პრიმიტივებს ერთმანეთში გადაფარვები ქონდეთ, რადგან საბოლოოდ მაინც მათი 1

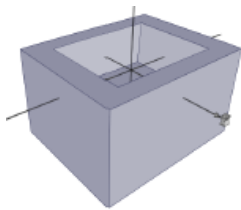
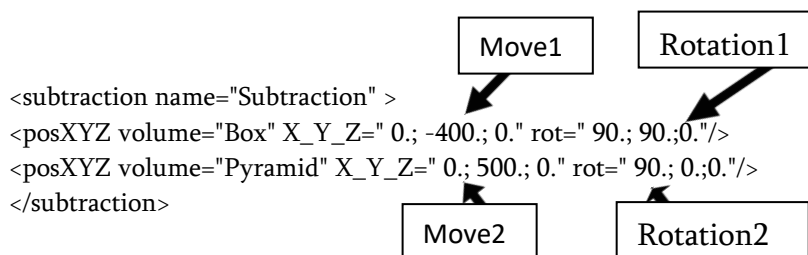
ობიექტად გაერთიანება ხდება. მეთოდის სინტასი შემდეგნაირია:

```
<union name="Merged" >
```



როგორც კოდიდან ჩანს აქ ხდება 2 დამოუკიდებელი პრიმიტივის Pyramid და Box-ის გაერთიანება. შესაძლებელია ასევე მათი გადაადგილება და შემოტრიალებაც Move1, Rotation1, Move2, Rotation2 პარამეტრებით.

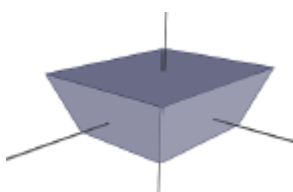
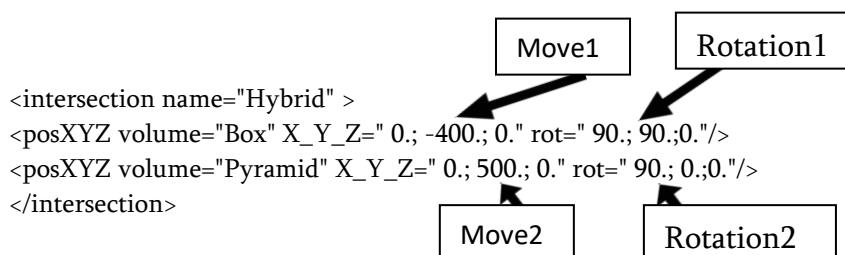
Subtracted (სურ. 43) ეს მეთოდიც დაფუძნებულია boolean პროცესორზე. მისი სემანტიკა არის ასეთი:



სურ. 43 დაპროგრამების მეთოდი subtract

ამ დროს ხდება ერთი ობიექტით მეორე ობიექტის ამოჭრა, ამ შეთხვევაში ხდება Box-ის ამოჭრა Pyramid-ით. აქ ასევე შესაძლებელია ობიექტების გადაადგილება და სასურველ პოზიციაზე დასმა. Move1, Rotation1, Move2, Rotation2 პარამეტრებით.

Hybridized (სურ. 44) ეს მეთოდიც დაფუძნებულია Boolean-ზე. მეთოდის სემანტიკა შემდეგნაირია:



სურ. 44 დაპროგრამების მეთოდი Hybridized

მეთოდი იყენებს 2 დამოუკიდებელ პრიმიტივს, რომლებიც გარკვეული ნაწილებით ერთმანეთს ფარავენ. ამ შემთხვევაში გამოყენებულია Box და Pyramid.

Hybridized ფუნქციით გამოყენების შედეგად რჩება მხოლოდ ის ნაწილი, რომელიც ამ 2 ობიექტს ერთმანეთთან შემხებლობაში ქონდათ. ასევე შესაძლებელია მათი გადაადგილება და შემოტრიალებაც Move1, Rotation1, Move2, Rotation2 პარამეტრებით.

ამრიგად, დეტექტორის გეომეტრიული აღწერები ზემოთ განხილული მეთოდებით არის დაპროგრამებული, XML-სა და GeoModel-ში. მათ გეომეტრიის დაპროგრამების მსგავსი სტურქურა და მეთოდები გააჩნიათ. საბოლოოდ მათი კონვერტაცია ხორციელდება GDML (Geometry Description Markup Language) ფორმატში. ის დაფუძნებულია xml-ის ენაზე. GDML -ში კონვერტირებული გეომეტრიული აღწერები წარმოადგენს Geant4-ში შესასვლელ ინფორმაციას [35].

1.5 საკვლევი ამოცანის ჩამოყალიბება

როგორც ზემოთ არის აღნიშნული სიმულაციის ჩასატარებლად აუცილებელია ხელოვნურად წარმოქმნილი მოვლენა და დეტექტორის გეომეტრიული აღწერები. წელიწადში დაახლოებით 5-7 მილიარდი ხელოვნური მოვლენის ფორმირება ხდება [24]. Geant4-ში 10 მოვლენის დამუშავებას დაახლოებით 560 წამი სჭირდება [36]. აქედან გამომდინარე კი 5 მილიარდი მოვლენის დასამუშავებლად Geant4-ს დასჭირდება 77 მლნ საათზე მეტი. სწორედ ამიტომ ძალიან მნიშვნელოვანი პარამეტრია სიმულაციის წარმატობა. წარმატობაზე მნიშვნელოვან გავლენას ახდენს სიმულაციის პაკეტებში გამოყენებული ATLAS-ის დეტექტორის გეომეტრიული აღწერები, რომლებიც დაპროგრამების სხვადასხვა მეთოდებით არიან შექმნილი. თუ რა გავლენას ახდენს გეომეტრიული აღწერების დაპროგრამების მეთოდები სიმულაციის პროცესის წარმატობაზე შესწავლილი არ არის და მისი გამოკვლევა წარმოადგენს აქტუალურ ამოცანას.

ამრიგად, სადისერტაციო ნაშრომის საკვლევ ამოცანას წარმოადგენს „გეომეტრიული აღწერების დაპროგრამების მეთოდების კვლევა ატლასის

ექსპერიმენტის სიმულაციისათვის“. აღნიშნული ამოცანის შესასრულებლად შემუშავდა კვლევის მეთოდური გეგმა:

1. წარმომადგენლობითი გეომეტრიული აღწერების შერჩევა
2. გეომეტრიული აღწერების დაპროგრამების მეთოდების შერჩევა და საკონტროლო მაგალითების ფორმირება
3. საკვლევი ინფრასტრუქტურის შერჩევა
4. სიმულაციის წარმადობის შემოწმების კრიტერიუმების განსაზღვრა
5. წარმომადგენლობითი Run-ების შერჩევა
6. საკონტროლო მაგალითების გაშვება სიმულაციაში და სატესტო შედეგების მიღება
7. სატესტო შედეგების განზოგადება და სისტემატიზაცია
8. დაპროგრამების მეთოდების აპრობაცია
9. დისერტაციის დასკვნები

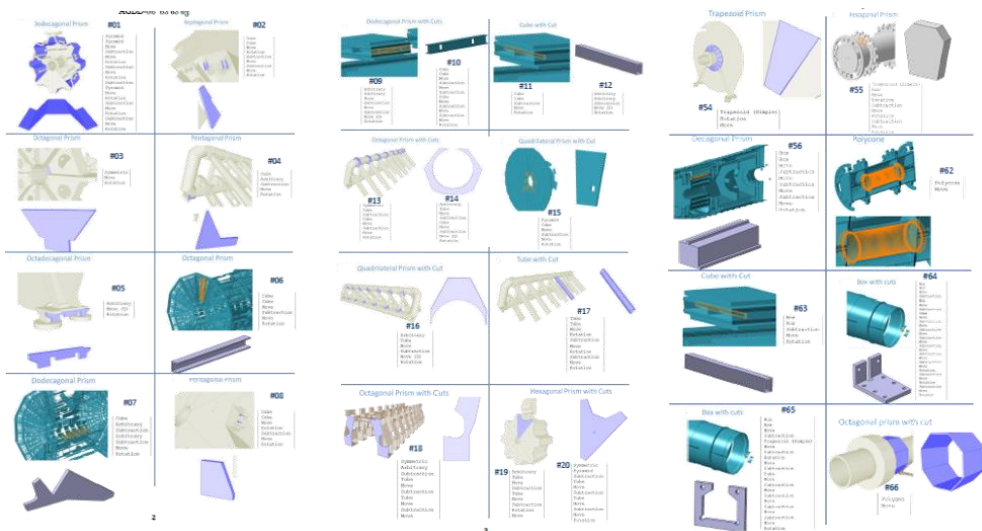
თავი II: სატესტო მაგალითების ფორმირება

გეომეტრიული აღწერების დაპროგრამების მეთოდების, სიმულაციის წარმადობაზე გავლენის გამოკვლევის მიზნით შერჩეულ იქნა სატესტო მაგალითები. აღნიშნული სამუშაო ჩატარდა რამდენიმე ეტაპად:

1. Atlas-ის დეტექტორის ტიპური გეომეტრიული აღწერების შერჩევა
2. სატესტო მაგალითებისთვის დაპროგრამების მეთოდების შერჩევა
3. სატესტო მაგალითების დაპროგრამების მეთოდების ანალიზი

2.1 დეტექტორის ტიპური გეომეტრიების გამოყოფა

Atlas-ის დეტექტორის ტიპური გეომეტრიული აღწერების შერჩევის მიზნით განხორციელდა ნიკო ცუცქერიძის სადოქტორო დისერტაციის არსებული მაგალითების შესწავლა [4], რომლებიც წამოადგენენ სიმულაციაში არსებული ატლასის დეტექტორის გეომეტრიის ტიპურ შემთხვევებს. აქ სულ თავმოყრილია 60 სატესტო მაგალითი სურ.45. აქედან #35 და #69 მაგალითები ერთმანეთის ანალოგიურები არიან სურ. 46 აქედან



სურ. 46 დანართი N1-ის მაგალითები

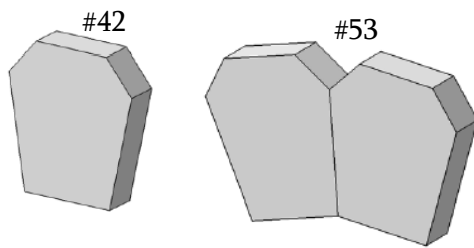


სურ. 45 დანართი N1-ის #35 და #51 მაგალითები

გამომდინარე #51 კვლევაში გათვალისწინებული არ იქნება. ანალოგიური

სტრუქტურა აქვთ #42 და #53 მაგალითებს სურ. 47. ამ მაგალითებში გეომეტრიებს ერთნაირი ფორმები აქვთ, განსხვავება მხოლოდ ობიექტების რაოდენობაშია, ამიტომ კვლევაში მონაწილეებს არ მიიღებს #53 მაგალითი.

2019 წელს განხორციელდა ATLAS-ის ექსპერიმენტის დეტექტორის ერთ-ერთი ქვე-სტრუქტურის Small Wheel-ის განახლება, შესაბამისად დღის წესრიგში დადგა მისი გეომეტრიული აღწერების სიმულაციის პაკეტებში განახლებაც. აღნიშნული სამუშაოები შესრულდა საქართველოს ტექნიკური უნივერსიტეტის ბირთვული ინჟინერიის ცენტრის მიერ. სიმულაციაში



სურ. 47 დანართი N1-ის #42 და #53 მაგალითები

დაემატა Small Wheel-ის ახალი გეომეტრიები. ამიტომ განხორციელდა დანართი 1-ში არსებული Small Wheel-ის ძველი გეომეტრიული აღწერების შედარება ახალთან. აღმოჩნდა რომ ახალ

ვერსიაში იყო ისეთი შემთხვევები რომლებიც ნ. ცუცქირიძის კვლევაში არ მოხვდა. ასეთი სულ 12 შემთხვევა აღმოჩნდა. შედეგად დანართი N1-დან ამოღებულ იქნა შემდეგი მაგალითები #5, # 12 და #27 მაგალითები, თუმცა მეორეს მხრივ დაემატა 12 ახალი მაგალითი. ამრიგად, ახალი გეომეტრიული აღწერების ანალიზის შედეგად შემდგომი ანალიზისთვის გამოყოფილ იქნა 66 სატესტო მაგალითი იხ. დანართი #1.

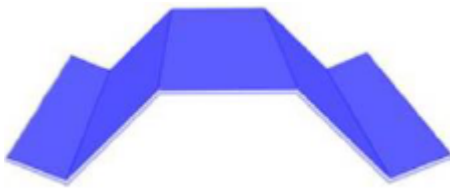
2.2 ღაპროგრამების მეთოდების შერჩევა

შემდგომ ეტაპზე თითოეული მაგალითისათვის შერჩეულ იქნა მათი დაპროგრამების შესაძლო სხვადასხვა მეთოდები. განვიხილოთ თითოეული მათგანი.

მაგალითი #1

მაგალითი #1 - ში წარმოდგენილია პრიმიტივი, რომელის ზედაპირიც არის წახნაგოვანი (სურ. 48). ამიტომ მის დასაპროგრამებლად შეიძლება

გამოყენებულ იქნეს ისეთი მეთოდები, რომელიც ასეთი ზედაპირების



სურ. 48 #1 სატესტო მაგალითი

მიღების საშუალებას მოგვცემს. ესენია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. აღნიშნული მეთოდების ანალიზის შედეგად შერჩეულ

იქნა ზემოთ მოცემული ობიექტის დაპროგრამების ყველა შესაძლო მეთოდი ცხრ.3. რაც შეეხება double symmetric polygon, combined, hybridized მეთოდებს, ისინი ამ პრიმიტივის ასაგებად არ გამოიყენება, მათი ობიექტების აგების თავისებურებების გამო. ცხრ. 3-ში ჩანს, რომ ზოგიერთ შემთხვევაში, III და IV ვარიანტებში, მეთოდები ერთმანეთისგან გამოყოფილია. რაც იმას ნიშნავს,

<p>I Pyramid Pyramid Subtraction Subtraction Pyramid Subtraction</p>	<p>II Arbitrary</p>	<p>III Cube/symmetric/pyramid Arbitrary Arbitrary Subtraction Subtraction</p>	<p>IV Cube/symmetric/pyramid Cube/symmetric/pyramid Cube/symmetric/pyramid Cube/symmetric/pyramid Cube/symmetric/pyramid merged</p>
-----------------------------------------------------------------------------------------------------	--------------------------------	----------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

ცხრ.3 #1 მაგალითის დაპროგრამების მეთოდები

რომ შეიძლება გამოყენებულ იქნეს ნებისმიერი მათგანი. დაპროგრამების მეთოდების ჩაწერის ეს ხერხი გამოყენებულ იქნება სხვა შემთხვევების განხილვის დროსაც. ცხრილიდან იკვეთება, რომ #1 ობიექტის ასაგებად გამოიყენება დაპროგრამების 4 განსხვავებული გზა. განვიხილოთ თითოეული მათგანი. პირველ შემთხვევაში იქმნება პირამიდა და შემდგომ ხდება ისევ პირამიდებით ამოჭრა subtract Boolean ოპერატორის საშუალებით. მე-2 შემთხვევაში გამოიყენება მხოლოდ arbitrary. მე-3 შემთხვევაში საწყისი ობიექტი შესაძლოა შეიქმნას სხვადასხვა მეთოდით, ხოლო Boolean-ისთვის საჭირო ობიექტების შექმნა კი arbitrary-ის საშუალებით ხორციელდება. მე-4 შემთხვევაში კი საჭიროა შეიქმნას 5 ობიექტი. ისინი შეიძლება დაპროგრამდეს სხვადასხვა მეთოდებით, რომლებიც ცხრ. 1-ში IV ვარიანტში არის აღწერილი. ბოლოს კი, merged Boolean ოპერატორის საშუალებით ხდება მათი გაერთიანება ერთ ობიექტად.

მაგალითი #2

მაგალითი #2 - ში წარმოდგენილი ობიექტი არის წახნაგოვანი სურ.49.



სურ. 49 #2 სატესტო მაგალითი

ამიტომ მის დასაპროგრამებლად განხილულ ისეთი მეთოდები, რომელიც წახნაგოვანი ზედაპირების აგების საშუალებას იძლევა. ესენია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined,

merged, subtracted, hybridized. მეთოდების ანალიზის შედეგად შეირჩა აღნიშნული ობიექტის დაპროგრამების შესაძლო ვარიანტები ცხრ.4.

I
Cube/symmetric/pyramid
Cube/symmetric/pyramid/arbitrary
Cube/symmetric/pyramid/arbitrary
Subtraction
Subtraction

II
Arbitrary

ცხრ.4 #2 მაგალითის დაპროგრამების

გამოიკვეთა დაპროგრამების 2 გზა. პირველ შემთხვევაში საწყის ობიექტად უნდა შეიქმნას პრიმიტივი და ეს შესაძლებელია 3 სხვადასხვა მეთოდით, ხოლო subtract Boolean ოპერატორისთვის

საჭირო ობიექტებს, მათი დაპროგრამება შესაძლებელია 4 სხვადასხვა მეთოდით. რაც შეეხება მე-2 გზას, ის პირველი შემთხვევისგან განსხვავებულია და ობიექტის აგება შესაძლებელია, მხოლოდ arbitrary-ის საშუალებით. როგორც ვხედავთ double symmetric polygon, combined, merged, hybridized მეთოდები არ იქნა გამოყენებული მათი ობიექტის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #3

მაგალითი #3 - ში წარმოდგენილი ობიექტი წარმოადგენს წახნაგოვან პრიმიტივს. სურ.50. მათ დასაპროგრამებლად განხილულ ისეთი მეთოდები, რომელიც წახნაგოვანი ზედაპირების აგების საშუალებას იძლევა. ესენია:



სურ. 50 #3 სატესტო მაგალითი

ესენია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

ანალიზის შემდგომ შეირჩა აღნიშნული ობიექტის დაპროგრამების ყველა შესაძლო გზა ცხრ.3. როგორც ცხრილიდან

I
Symmetric/
Arbitrary

II
Cube/Pyramid
Arbitrary
Subtraction

ცხრ. 5 #3 მაგალითის დაპროგრამების მეთოდები

ჩანს სულ დაპროგრამების 2 გზა არის გამოყენებული 1-ლი ეს არის გზა, სადაც გამოყენებულია ობიექტის აგების 2 მეთოდი Symmetric და Arbitrary, ხოლო მე-2 შემთხვევაში კი ობიექტი მიიღება

Boolean-ის შედეგად. საწყისი პრიმიტივის დაპროგრამება შესაძლებელია 2 სხვადასხვა მეთოდით. ხოლო Boolean-ისთვის საჭირო ობიექტის შექმნა ხდება Arbitrary-ის საშუალებით. რაც შეეხება, double symmetric polygon, combined, merged, hybridized მეთოდებს ისინი ამ შემთხვევაში არ იქნა გამოყენებული მათი ობიექტის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #4

მაგალითი #4 წინა შემთხვევების მსგავსად წარმოადგენს წახნაგოვან ობიექტს სურ. 51 და ამიტომ მისი დაპროგრამების შესაძლო მეთოდებიც



სურ. 51 #4 სატესტო მაგალითი

მათი ანალოგიურია. ესენია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted,

hybridized. მისი დაპროგრამების მეთოდების შესაძლო ვარიანტების შესწავლის შემდგომ შეირჩა რამდენიმე ვარიანტი ცხრ. 6 პირველ გზაში

I
Cube/symmetric/pyramid
Arbitrary/pyramid
Subtraction

II
Arbitrary

ცხრ. 6 #4 მაგალითის დაპროგრამების მეთოდები

ობიექტის შესაქმნელად გამოყენებულია დაპროგრამების 3 მეთოდი Cube, symmetric და pyramid, ხოლო Boolean

ოპერაციისთვის საჭირო პრიმიტივის შესაქმნელად კი 2 მეთოდი. ესენია Arbitrary და pyramid. მე-2 გზა აღნიშნული ობიექტის შესაქმნელად არის Arbitrary. ზემოთ ჩამოთვლილი სხვა მეთოდები #4 მაგალითში არსებული ობიექტის შესაქმნელად არ გამოგდება მათი ობიექტის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #5

მაგალითი #5-ში არსებული პრიმიტივიც არის წახნაგოვანი ზედაპირის სურ. 52, ამიტომ მის დასაპროგრამებლად მხოლოდ შემდეგი



სურ. 52 #5 სატესტო მაგალითი

მეთოდები შეიძლება იქნეს გამოყენებული: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged,

subtracted, hybridized. ობიექტის ფორმის და დაპროგრამების მეთოდების ანალიზის შედეგად გამოიკვეთა, რომ სურ. 52-ზე წარმოდგენილი

<p>I Cube/symmetric/pyramid Cube/symmetric/pyramid/ Arbitrary Subtraction</p>	<p>II Arbitrary</p>
------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------

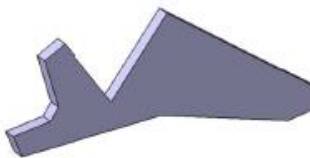
ცხრ. 7 #5 მაგალითის დაპროგრამების მეთოდები

პრიმიტივი შეიძლება დაპროგრამდეს შემდეგი გზებით ცხრ. #7 ამ შემთხვევაშიც დაპროგრამების 2 გზა გამოიკვეთა.

პირველი გზა არის შემდეგნაირი. უნდა შეიქმნას საწყისი პრიმიტივი, ამ შემთხვევაში ეს არის პრიზმა და ეს შესაძლებელია შემდეგი მეთოდებით: Cube/ symmetric/ pyramid. ხოლო Boolean-ისთვის საჭირო პრიმიტივი შეიძლება შეიქმნას შემდეგი მეთოდებით Cube/ symmetric/ pyramid/ Arbitrary. რაც შეეხება მე-2 გზას, აქ გამოიყენება მხოლოდ Arbitrary მეთოდი. თავიანთი სპეციფიკიდან გამომდინარე double symmetric polygon, combined, merged და hybridized მეთოდები ამ შემთხვევაში არ გამოიყენება.

მაგალითი #6

მე-6 მაგალითში არსებული პრიმიტივიც არის წახნაგოვანი ზედაპირის. სურ. 53. ამიტომ წინამორბედი პრიმიტივების მსგავსად მის შესაქმნელად



სურ. 53 #6 სატესტო მაგალითი

შეიძლება გამოყენებულ იქნეს შემდეგი დაპროგრამების მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. თუმცა მე-6 მაგალითის დასაპროგრამებლად მხოლოდ შემდეგი მეთოდები

შეიძლება იქნეს გამოყენებული ცხრ. 8. გამოიკვეთა ობიექტის შექმნის 2 გზა. პირველი ეს არის ობიექტის შექმნა Boolean ოპერაციის გამოყენებით,

I
Cube/Pyramid/
Symmetric
Arbitrary
Subtraction
Arbitrary
Subtraction

II
Arbitrary

ამისათვის საწყისი ობიექტის შექმნა უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით, ხოლო Boolean-ისთვის საჭირო

ცხრ. 8 #6 მაგალითის დაპროგრამების

პრიმიტივი უნდა შეიქმნას Arbitrary მეთოდით. მე-2 გზის გამოყენების

შემთხვევაში ობიექტი უნდა შეიქმნას მხოლოდ Arbitrary მეთოდის გამოყენებით. double symmetric polygon, combined, merged, hybridized მეთოდები ამ ობიექტის შესაქმნელად არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #7

მე-7 მაგალითში არსებული პრიმიტივის ზედაპირიც წახნაგოვანი ფორმისაა სურ.54. ამიტომ ამ შემთხვევაშიც განხილულ იქნა ის მეთოდების,



სურ. 54 #7 სატესტო მაგალითი

რომლების ასეთი ზედაპირების შექმნის შესაძლებლობას იძლევა, ესენია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon,

combined, merged, subtracted, hybridized. გამოიკვეთა, რომ ამ პრიმიტივის დასაპროგრამებლად შეიძლება მხოლოდ შემდეგი მეთოდების გამოყენება ცხრ. 9. შერჩეულ იქნა დაპროგრამების 2 გზა. 1-ლი ეს არის როცა ობიექტის შესაქმნელად გამოიყენება Boolean ოპერაციები. ამ შემთხვევაში საწყისი პრიმიტივის დაპროგრამება ხდება Cube, Pyramid ან Symmetric მეთოდით, ხოლო Subtraction Boolean-ისთვის საჭირო პრიმიტივის დაპროგრამება კი ხდება

I
Cube/Pyramid/Symmetric
Cube/Pyramid/Symmetric/Arbitrary
Subtraction

II
Arbitrary

Cube, Pyramid, Symmetric ან Arbitrary მეთოდით. მე-2

ცხრ. 9 #7 მაგალითის დაპროგრამების მეთოდები

შემთხვევაში კი გამოიყენება მხოლოდ Arbitrary მეთოდი.

დანარჩენი შესაძლო ვარიანტები გამოირიცხა, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #8

მაგალითი #8-ში არსებული პრიმიტივი არის წახნაგოვანი ზედაპირის მქონე (სურ. 55) და მისი დაპროგრამება შესაძლებელს უშემდეგი მეთოდებით:



სურ. 55 #8 სატესტო მაგალითი პრიმიტივისთვის შეირჩა დაპროგრამების 1 გზა, სადაც სხვადასხვა მეთოდების რამდენიმე კომბინაცია არის გაერთიანებული და გამოიყენება Subtraction Boolean ოპერაციები ცხრ. 10.

Cube/ Pyramid/ Symmetric/ Arbitrary
Cube/ Pyramid/ Symmetric/ Arbitrary
Cube/ Pyramid/ Symmetric/ Arbitrary
Subtraction
Subtraction
Subtraction

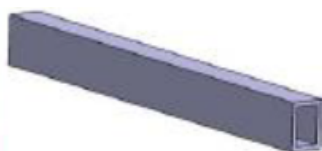
cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. ამ

საწყისი პრიმიტივის დაპროგრამება შეიძლება განხორციელდეს Cube, Pyramid, Symmetric ან Arbitrary მეთოდით. იგივე მეთოდებით არის Boolean-ისთვის საჭირო პრიმიტივების

ცხრ. 10 #8 მაგალითის დაპროგრამების მეთოდები დაპროგრამებაც შესაძლებელი. სხვა მეთოდები აღნიშნული ობიექტის შესაქმნელად არ გამოდგება, მათი ობიექტის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #9

მე-9 მაგალითის პრიმიტივი, ისევე როგორც ყველა წინამორბედი არის წახნაგოვანი ზედაპირის მქონე სურ. 56. წახნაგოვანი ზედაპირების



სურ. 56 #9 სატესტო მაგალითი

Cube/ Pyramid/ Symmetric/ Arbitrary
Cube/ Pyramid/ Symmetric/ Arbitrary
Subtraction

დასაპროგრამებლად კი შეიძლება გამოყენებულ იქნება შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined,

merged, subtracted, hybridized. მათი ანალიზი შედეგად გაირკვა, რომ ამ პრიმიტივის ასაგებად შემდეგი მეთოდები შეიძლება იქნეს გამოყენებული ცხრ. 11. როგორც ცხრ. #9-იდან ჩანს, პრიმიტივის შესაქმნელად

ცხრ. 11 #9 მაგალითის დაპროგრამების მეთოდები

გამოიყენება Subtraction Boolean ოპერაცია. სხვადასხვა მეთოდების რამდენიმე კომბინაცია შეიძლება იქნეს გამოყენებული. საწყისი პრიმიტივის შესაქმნელად საჭიროა Cube, Pyramid, Symmetric ან Arbitrary მეთოდები. იგივე მეთოდებია საჭიროა Boolean-ისთვის საჭირო პრიმიტივის დასაპროგრამებლად. დანარჩენი ზემოთ ჩამოთვლილი მეთოდები კი მათი სპეციფიკიდან გამომდინარე ამ შემთხვევაში არ გამოდგება.

მაგალითი #10

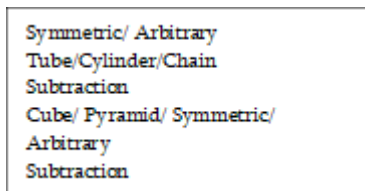
მაგალითი #10-ში არსებული პრიმიტივის ზედაპირი არის შერეული,



როგორც წახნაგოვანი, ასევე ცილინდრული სურ. 57. ამიტომ მის შესაქმნელად შესაძლებელია გამოყენებულ იქნას შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. პრიმიტივის ანალიზის

სურ. 57 #10 სატესტო მაგალითი

შედეგად დადგინდა მისი დაპროგრამების მეთოდები. ცხრ. 12 როგორც ცხრ. 10-დან ირკვევა აღნიშული პრიმიტივის შესაქმნელად საჭიროა Subtraction Boolean ოპერატორი. ამისათვის დაპროგრამების სხვადასხვა მეთოდების



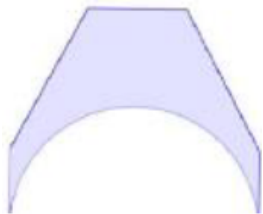
რამდენიმე კომბინაცია შეიძლება იქნეს გამოყენებული. საწყისი პრიმიტივის შესაქმნელად გამოყენებულ უნდა იქნას Symmetric ან Arbitrary მეთოდი. Boolean-ისთვის საჭიროა როგორც ცილინდრული

ცხრ. 12 #10 მაგალითის დაპროგრამების მეთოდები

ასევე წახნაგოვანი ზედაპირის მქონე პრიმიტივები. ცილინდრული პრიმიტივის შესაქმნელად გამოყენებულ უნდა იქნას Tube, Cylinder ან Chain დაპროგრამების მეთოდი, ხოლო წახნაგოვანი ზედაპირის პრიმიტივის შესაქმნელად კი Cube, Pyramid, Symmetric ან Arbitrary მეთოდები. დანარჩენი ზემოთ ჩამოთვლილი დაპროგრამების მეთოდები კი მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე ამ შემთხვევისთვის არ გამოდგება.

მაგალითი #11

მე-11 მაგალითის პრიმიტივის ზედაპირი არის შერეული სახის. მას გააჩნია, როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირები სურ. 58.



სურ. 58 #11 სატესტო მაგალითი

ამიტომ მის დასაპროგრამებლად შესაძლებელია გამოყენებულ იქნეს შემდეგი მეთოდები: ამიტომ მის შესაქმნელად შესაძლებელია გამოყენებულ იქნას შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

ამ ობიექტის შექმნის თავისებურებებიდან და მეთოდების შესაძლებლობებიდან გამომდინარე შერჩეულ იქნა დაპროგრამების შემდეგი მეთოდების რამდენიმე კომბინაცია ცხრ.13. აღნიშნული პრიმიტივის დასაპროგრამებლად საჭიროა Subtraction Boolean ოპერაციის გამოყენება.

Arbitrary/ Symmetric
Tube/ Cylinder/ Chain
Subtraction

ცხრ. 13 #11 მაგალითის დაპროგრამების მეთოდები

საწყისი გეომეტრიის დაპროგრამება ხდება Arbitrary ან Symmetric მეთოდით, ხოლო Boolean-ისთვის საჭირო პრიმიტივის შექმნა უნდა მოხდეს Tube, Cylinder ან Chain მეთოდით. სხვა დანარჩენი მეთოდი, რომელიც ზემოთ არის მოყვანილი, აღნიშნული ობიექტის დასაპროგრამებლად არ გამოდგება, მათი სპეციფიკიდან გამომდინარე.

მაგალითი #12

მაგალითი #12 წარმოადგენს ცილინდრული ტიპის პრიმიტივს სურ.

59. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: tube, cylinder, chain,



სურ. 59 #12 სატესტო მაგალითი

combined, merged, subtracted, hybridized. ობიექტის ანალიზის შედეგად დადგინდა მისი შექმნის რამდენიმე კომბინაცია ცხრ.14. პრიმიტივის შესაქმნელად საჭიროა Subtraction Boolean ოპერაციის გამოყენება. ვინაიდან აღნიშნული

Tube/ Cylinder/ Chain
 Tube/ Cylinder/ Chain
 Subtraction
 Subtraction

ცხრ. 14 #12 მაგალითის
 დაპროგრამების მეთოდები

ობიექტი ცილინდრულია, მის
 დასაპროგრამებლად შერჩეულია შესაბამისი
 მეთოდები. კერძოდ საწყისი გეომეტრიის
 ასაგებად გამოყენებულია Tube, Cylinder ან

Chain მეთოდი. იგივე მეთოდებია გამოყენებული Boolean-ისთვის საჭირო
 პრიმიტივის შესაქმნელად. რაც შეეხება merged და hybridized მეთოდებს მათი
 სპეციფიკიდან გამომდინარე მათი გამოყენება ამ შემთხვევაში არ ხდება.

მაგალითი #13

მე-13 მაგალითში არსებული გეომეტრიის ზედაპირები არის შერეული
 სახის, როგორც წახნაგოვანი, ასევე ცილინდრული სურ 60. ამიტომ მის
 შესაქმნელად შეიძლება გამოყენებულ იქნეს შემდეგი მეთოდები: cube,



სურ. 60 #13 სატესტო
 მაგალითი

pyramid, arbitrary polygon, symmetric polygon, double
 symmetric polygon, tube, cylinder, chain, combined,
 merged, subtracted, hybridized. აღნიშნული
 პრიმიტივის ანალიზის შემდეგ, მისი
 დაპროგრამების რამდენიმე კომბინაცია იქნა
 შემუშავებული ცხრ. 15. ამ ობიექტის

შესაქმნელად აუცილებელია Subtraction Boolean ოპერაციის გამოყენება.
 მისთვის საჭირო საწყისი ობიექტი შეიძლება დაპროგრამდეს 2 სხვადასხვა
 მეთოდით ესენია: Symmetric და Arbitrary. Boolean-ისთვის კი 3 სხვადასხვა

Symmetric/ Arbitrary
 Arbitrary
 Subtraction
 Tube/ Cylinder/ Chain
 Subtraction
 Tube/ Cylinder/ Chain
 Subtraction

ცხრ. 15 #13 მაგალითის
 დაპროგრამების მეთოდები

პრიმიტივია საჭირო. პირველ შემთხვევაში მისი
 შექმნა უნდა მოხდეს Arbitrary-ით, ხოლო მე-2
 და მე-3 პრიმიტივების კი Tube, Cylinder ან Chain
 მეთოდებით. დანარჩენი მეთოდები ამ
 შემთხვევაში არ გამოდგება მათი გეომეტრიის

აგების სპეციფიკიდან გამომდინარე.

მაგალითი #14



სურ. 61 #14 სატესტო მაგალითი

მე-14 მაგალითის პრიმიტივი წარმოადგენს წინა შემთხვევის მსგავსად შერეულ გეომეტრიას. ის შედგება, როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირისგან სურ. 61. შესაბამისად მისი დაპროგრამების სავარაუდო

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ამ ობიექტის ანალიზის შედეგად დადგინდა მისი დაპროგრამების სხვადასხვა

<p>I Arbitrary Tube/ Cylinder/Chain Subtraction Tube/ Cylinder/Chain Subtraction</p>	<p>II Symmetric Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction</p>
-----------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

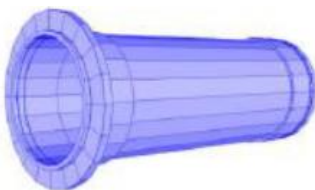
ცხრ. 16 #14 მაგალითის დაპროგრამების მეთოდები

კომბინაციები ცხრ. 16. როგორც ცხრ. 16-იდან ჩანს, ობიექტის შესაქმნელად 2 სხვადასხვა გზა არის შერჩეული. ორივე შემთხვევაში გამოყენებულია Subtraction Boolean ოპერაცია. პირველ შემთხვევაში საწყისი ობიექტის შექმნა ხდება Arbitrary--

ის საშუალებით, ხოლო დამხმარე 2 პრიმიტივის შექმნა ხდება Tube, Cylinder ან Chain მეთოდებით. მე-2 შემთხვევაში. პირველისგან განსხვავებული არის ის, რომ საწყისი ობიექტის შექმნა ხდება Symmetric მეთოდით და დამხმარე ობიექტების რაოდენობა კი არის 3. აქედან 2 პირველის მსგავსად Tube, Cylinder ან Chain მეთოდებით იქმნება, ხოლო მე-3 ობიექტი კი Pyramid მეთოდით. სხვა ზემოთ ჩამოთვლილი მეთოდების გამოყენება ამ შემთხვევაში არ ხდება მათი სპეციფიკიდან გამომდინარე.

მაგალითი #15

მაგალითი #15 არის ცილინდრული ობიექტი სურ. 62. ამიტომ მისი შესაძლო დაპროგრამების მეთოდები იქნება შემდეგი: tube, cylinder, chain,



სურ. 62 #15 სატესტო მაგალითი

combined, merged, subtracted, hybridized. პრიმიტივის ანალიზის შედეგად დადგინდა, რომ მის დასაპროგრამებლად გამოყენებულ უნდა იქნას შემდეგი მეთოდები ცხრ. 17. 2 სხვადასხვა გზა გამოიკვეთა, რომლითაც

I Cylinder	II Tube Tube Tube Tube
---------------	------------------------------------

ცხრ. 17 #15 მაგალითის დაპროგრამების მეთოდები

შესაძლებელია შეიქმნას #15 პრიმიტივი. პირველი გზა გულისხმობს მხოლოდ Cylinder მეთოდის გამოყენებას, ხოლო მე-2 გზა კი Tube მეთოდის გამოყენებას. სხვა მეთოდების გამოყენება ამ შემთხვევაში არ ხდება.

მაგალითი #16

მე-16 მაგალითის პრიმიტივი წარმოადგენს შერეულ ობიექტს. ის შედგება, როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირებისგან სურ.



სურ. 63 #16 სატესტო მაგალითი

63. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ანალიზის შემდგომ ჩამოყალიბდა დაპროგრამების მეთოდების სხვადასხვა კომბინაციები ცხრ. 18. შეირჩა 2 სხვადასხვა

გზა. ორივე შემთხვევაში ობიექტი იქმნება Subtraction Boolean ოპერაციის საშუალებით. პირველ შემთხვევაში საწყისი პრიმიტივის დაპროგრამება ხორციელდება Arbitrary მეთოდით, ხოლო მისი ამოჭრა ხდება პრიმიტივით,

I Arbitrary Tube/ Cylinder/ Chain Subtraction	II Pyramid/ Cube/ Symmetric Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Arbitrary Subtraction Subtraction
-----------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------

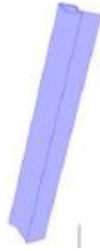
ცხრ. 18 #16 მაგალითის დაპროგრამების მეთოდები

რომელიც უნდა დაპროგრამდეს შემდეგი მეთოდებით: Tube, Cylinder ან Chain. რაც შეეხება მე-2 გზას, აქ საწყისი ობიექტის შექმნა ხდება Pyramid, Cube ან Symmetric მეთოდით და Subtraction Boolean ოპერაციისთვის საჭიროა 3

სხვადასხვა ობიექტი. აქედან 2-ის დაპროგრამება ხორციელდება Arbitrary მეთოდით, ხოლო მე-3 გეომეტრიის Tube, Cylinder ან Chain მეთოდით. სხვა მეთოდების ამ შემთხვევაში გამოყენება არ ხდება, რადგან მათ გააჩნიათ გეომეტრიის აგების სხვა სპეციფიკები.

მაგალითი #17

მაგალითი #17 არის შერეული ტიპის, რადგან მას გააჩნია როგორც წახნაგოვანი ასევე ცილინდრული ტიპის ზედაპირები სურ. 64. ამიტომ მის



სურ. 64 #17
სატესტო მაგალითი

შესაძლოა გამოყენებულ იქნეს დაპროგრამების შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

ამ ობიექტის ანალიზის შედეგად ჩამოყალიბდა დაპროგრამების მეთოდების სხვადასხვა კომბინაცია

ცხრ. 19. როგორ ცხრილიდან ჩანს #17 ობიექტის შესაქმნელად გამოიყენება

Cube/ Pyramid/ Symmetric/ Arbitrary
Cube/ Pyramid/ Symmetric/ Arbitrary
Subtraction
Tube/ Cylinder/ Chain
Subtraction
Subtraction

Subtraction Boolean ოპერაცია. მისი საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს 4 სხვადასხვა მეთოდიდან ერთერთით. ესენია: Cube, Pyramid, Symmetric ან

ცხრ. 19 #17 მაგალითის დაპროგრამების მეთოდები

Arbitrary მეთოდები. მის ამოსაჭრელად კი 2 სხვადასხვა პრიმიტივია საჭირო. პირველის დაპროგრამება ასევე Cube, Pyramid, Symmetric ან Arbitrary მეთოდებით არის შესაძლებელი, ხოლო მე-2 პრიმიტივის კი Tube, Cylinder ან Chain მეთოდებით.

მაგალითი #18

მე-18 მაგალითის პრიმიტივი წარმოადგენს ცილინდრულ ობიექტს სურ. 65 და მისი დაპროგრამების შესაძლო მეთოდებია: tube, cylinder, chain,



სურ. 65 #18 სატესტო მაგალითი

combined, merged, subtracted, hybridized. ამ ობიექტის დაპროგრამების მეთოდებად კი შერჩეულ იქნა შემდეგი კომბინაციები ცხრ. 20. აღნიშნული გეომეტრიის შესაქმნელად საჭიროა Subtraction

Boolean ოპერაციის გამოყენება. ამისთვის პირველ რიში უნდა შეიქმნას

Tube/ Cylinder
Cube/ Pyramid/ Symmetric/ Arbitrary
Subtraction

ცხრ. 20 #18 მაგალითის დაპროგრამების მეთოდები

საწყისი ობიექტი რომლის დაპროგრამებაც ხდება შემდეგი მეთოდებით: Tube ან Cylinder, ხოლო მის ამოსაჭრელად კი პრიმიტივი უნდა შეიქმნას შემდეგნაირად: Cube,

Pyramid, Symmetric ან Arbitrary. დარჩენილი მეთოდები კი ამ შემთხვევაში არ გამოიყენება, რადგან მათ გეომეტრიის აგების სხვა სპეციფიკა გააჩნიათ.

მაგალითი #19

ეს მაგალითი წარმოადგენს წახნაგოვანი ზედაპირის მქონე ობიექტს სურ. 66.

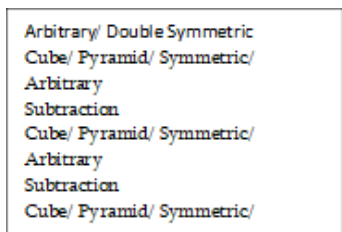


სურ. 66 #19 სატესტო მაგალითი

ამიტომ მის დასაპროგრამებლად შესაძლებელია შემდეგი მეთოდების გამოყენება: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

აღნიშნული გეომეტრიის ანალიზის შემდეგ შეირჩა მისი დაპროგრამების რამდენიმე

კომბინაცია ცხრ 21. სურ. 66 ზე წარმოდგენილი გეომეტრიის შესაქმნელად საჭიროა Subtraction Boolean ოპერაციის გამოყენება. მისთვის საწყისი ობიექტის დაპროგრამება უნდა მოხდეს შემდეგი მეთოდებით: Arbitrary ან



Double Symmetric, ხოლო მისი ამოჭრა უნდა განხორციელდეს 3 სხვადასხვა პრიმიტივით, რომლებიც ერთი და იგივე მეთოდებით უნდა შეიქმნას და ეს მეთოდებია: Cube, Pyramid,

ცხრ. 21 #19 მაგალითის დაპროგრამების

Symmetric ან Arbitrary. სხვა შესაძლო მეთოდები ამ შემთხვევისთვის არ გამოდგება, რადგან მათ

გეომეტრიის აგების განსხვავებული სპეციფიკა გააჩნიათ.

მაგალითი #20

მე-20 მაგალითში არსებული გეომეტრია არის შერეული ტიპის,



სურ. 67 #20 სატესტო მაგალითი

რადგანაც მისი ზედაპირი არის როგორც წახნაგოვანი ასევე ცილინდრული სურ. 67. ამიტომ მისი დაპროგრამებისათვის შესაძლებელია შემდეგი მეთოდების გამოყენება: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted,

hybridized. აღნიშნული ობიექტის გეომეტრიული ფორმის და მეთოდების შესაძლებლობის ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების სხვადასხვა კომბინაციები ცხრ. 22. ცხრილიდან ჩანს, რომ 2 გზა სხვადასხვა გზა არის გამოკვეთილი ობიექტის შესაქმნელად. ორივე გზის მიხედვით ობიექტი იქმნება Subtraction Boolean ოპერაციის საშუალებით. პირველ შემთხვევაში საწყისი პრიმიტივის დაპროგრამება ხდება Arbitrary მეთოდის

I
Arbitrary
Tube/ Cylinder/Chain
Subtraction

II
Double Symmetric
Tube/ Cylinder/ Chain
Subtraction

საშუალებით, ხოლო მისი ამოჭრა ხდება ობიექტით, რომელიც უნდა შეიქმნას Tube, Cylinder ან Chain მეთოდით. რაც შეეხება მეორე გზას აქ საწყისი

ცხრ. 22 #20 მაგალითის დაპროგრამების მეთოდები

ობიექტი იქმნება Double Symmetric მეთოდით, ხოლო მისი ამოჭრა 3 პრიმიტივით ხორციელდება. რომელიც უნდა შეიქმნას Tube, Cylinder ან Chain მეთოდით. სხვა დაპროგრამების მეთოდები, გარდა აქ განხილულისა ამ შემთხვევისთვის არ გამოდგება, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #21

21-ე მაგალითში წარმოდგენილია შერეული ტიპის გეომეტრია. მას გააჩნია როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირი სურ. 68.



სურ. 68 #21 სატესტო მაგალითი

ამიტომ მისი დაპროგრამებისათვის შესაბამისი მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. გეომეტრიის ანალიზის შემდეგ დადგინდა, რომ მისი შექმნა უნდა განხორციელდეს Subtraction Boolean ოპერაციის

საშუალებით ცხრ. 23. ცხრილიდან ჩანს, რომ საწყისი ობიექტის შექმნა უნდა მოხდეს Arbitrary ან Symmetric მეთოდის გამოყენებით, ხოლო მის ამოსაჭრელად 2 პრიმიტივი უნდა შეიქმნას. 1-ლი პრიმიტივი, რომლის

Arbitrary/ Symmetric
 Arbitrary/ Symmetric/ Double Symmetric
 Subtraction
 Subtraction
 Subtraction
 Subtraction
 Subtraction
 Subtraction
 Tube/ Cylinder/ Chain
 Subtraction
 Subtraction

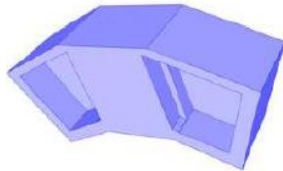
ცხრ. 23 #21 მაგალითის დაპროგრამების მეთოდები

დაპროგრამდეს Tube, Cylinder ან Chain მეთოდით და შემდეგ 2-ჯერ იქნება გამოყენებული. დანარჩენი მეთოდები, რომლებიც დაპროგრამების შესაძლო ვარიანტებად განიხილებოდა არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

საშუალებითაც საწყისი ობიექტი 5 ადგილზე ამოიჭრება, უნდა დაპროგრამდეს Arbitrary, Symmetric ან Double Symmetric მეთოდით. ხოლო მე-2 ობიექტი, რომელიც ასევე Boolean-ისთვის არის საჭირო უნდა

მაგალითი #22

#22 მაგალითი არის წინა შემთხვევის მსგავსად შერეული ტიპის. ანუ მისი ზედაპირი არის, როგორც წახნაგოვანი, ასევე ცილინდრული ფორმის სურ. 69. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid,



სურ. 69 #22 სატესტო მაგალითი

arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. აღნიშნული ობიექტის ასაგებად აუცილებელია გამოყენებულ იქნეს Subtraction Boolean ოპერაცია. კვლევის

შედეგად დადგინდა დაპროგრამების მეთოდების რამდენიმე კომბინაცია,

Arbitrary/ Symmetric
 Arbitrary
 Arbitrary
 Subtraction
 Subtraction
 Subtraction
 Subtraction
 Subtraction
 Arbitrary
 Subtraction
 Tube/ Cylinder/ Chain
 Subtraction

ცხრ. 24 #22 მაგალითის დაპროგრამების

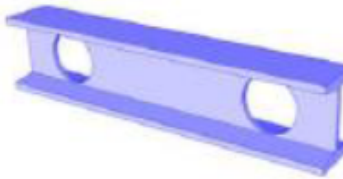
დანარჩენი მეთოდები, გარდა განხილული მეთოდებისა, რომლებიც ზემოთ

რომლებიც ამ ობიექტის გეომეტრიის აგებას უზრუნველყოფს ცხრ. 24. როგორც ცხრ. 24 -იდან იკვეთება საწყისი გეომეტრიის აგება უნდა მოხდეს Arbitrary ან Symmetric მეთოდით. ხოლო Boolean-ისთვის საჭიროა 4 პრიმიტივი, რომელთაგან 3 უნდა აიგოს Arbitrary მეთოდით, ხოლო 1- Tube, Cylinder ან Chain მეთოდით.

არის აღწერილი ამ შემთხვევისთვის არ გამოდგებიან, მათი სპეციფიკიდან გამომდინარე.

მაგალითი #23

მაგალითი #23 წარმოადგენს შერეული ტიპის, როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირების მქონე ობიექტს სურ. 70. ამიტომ წინა



სურ. 70 #23 სატესტო მაგალითი

შემთხვევის მსგავსად, აქაც მისი დაპროგრამების სავარაუდო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. მეთოდების

შესაძლებლობის და გეომეტრიის ფორმის ანალიზის შედეგად დამუშავდა ის მეთოდები ცხრ. 25, რომლებიც საშუალებას იძლევა მიღებულ იქნეს სურ. 70-ზე წარმოდგენილი ობიექტი, ამ შემთხვევაშიც საჭიროა Subtraction Boolean ოპერაცია გამოყენება. როგორც ცხრილიდან ჩანს, ობიექტის შექმნის 2 განსხვავებული გზა არსებობს. განსხვავება არის ის, რომ მე-2 შემთხვევაში კიდევ 2 დამატებითი Subtraction

I
Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction Subtraction

II
Cube/Pyramid Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction

ცხრ. 25 #23 მაგალითის დაპროგრამების მეთოდები

არის გასაკეთებელი. პირველი გზის მიხედვით საწყისი ობიექტი უნდა დაპროგრამდეს Arbitrary ან Symmetric მეთოდით, ხოლო მის ამოსაჭრელად პრიმიტივი, რომელსაც ცილინდრული ფორმა

აქვს უნდა შეიქმნას Tube, Cylinder ან Chain მეთოდით. მე-2 გზის მიხედვით, კი როგორც უკვე ითქვა 1 დამატებითი პრიმიტივია საჭირო Boolean-ის თვის, რომელიც უნდა შეიქმნას Cube, Pyramid, Symmetric ან Arbitrary მეთოდით. მე-2 პრიმიტივი კი პირველი გზის მსგავსად უნდა შეიქმნას Tube, Cylinder ან Chain მეთოდით. რაც შეეხება საწყის ობიექტს, ის უნდა დაპროგრამდეს Cube ან Pyramid მეთოდით. სხვა მეთოდები რომლების აქ განხილულ არ იქნა, ამ

შემთხვევაში არ გამოდგება, მათი გეომეტრიის აგების თავისებურებების გათვალისწინებით.

მაგალითი #24

24-ე მაგალითი წარმოადგენს ასევე წინა შემთხვევის მსგავსად შერეული, წახნაგოვანი და ცილინდრული ზედაპირების მქონე ობიექტს



სურ. 71 #24 სატესტო მაგალითი

სურ. 71. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. მეთოდების და გეომეტრიის ფორმის ანალიზის შედეგად მისი დაპროგრამების რამდენიმე კომბინაცია შეიქმნა ცხრ.

26. დაპროგრამების 2 გზა იქნა გამოყოფილი, ორივე შემთხვევაში გამოყენებულ უნდა იქნეს Subtraction Boolean ოპერაცია. ამ 2 გზას შორის განსხვავება არის ის, რომ ერთ შემთხვევაში ნაკლები რაოდენობის Subtraction

I
Arbitrary
Tube/ Cylinder/Chain
Subtraction
Subtraction

II
Cube/ Pyramid/ Symmetric
Cube/ Pyramid/ Symmetric/ Arbitrary
Subtraction
Tube/ Cylinder/ Chain
Subtraction
Subtraction

ცხრ. 26 #24 მაგალითის დაპროგრამების მეთოდები

Boolean ოპერაცია გამოიყენება. პირველ გზის მიხედვით საწყისი პრიმიტივი იქმნება Arbitrary მეთოდით და სწორედ ეს არის ზემოთ ხსენებული სხვაობის მიზეზი. რაც შეეხება Boolean-ისთვის საჭიროა 1

პრიმიტივი და ის უნდა შეიქმნას Tube, Cylinder ან Chain მეთოდით. მე-2 გზის გამოყენების დროს კი საწყისი ობიექტის ფორმირება ხორციელდება Cube, Pyramid ან Symmetric მეთოდით და ამ შემთხვევაში მის ამოსაჭრელად 2 სხვადასხვა პრიმიტივია საჭირო. 1-ლი პრიმიტივის დაპროგრამება უნდა მოხდეს Cube, Pyramid, Symmetric ან Arbitrary მეთოდით, ხოლო მე-2 პრიმიტივის კი Tube, Cylinder ან Chain მეთოდით. ყველა სხვა მეთოდი, რომელიც ზემოთ იყო ჩამოთვლილი და არ განხილულა ამ შემთხვევაში არ გამოიყენება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #25

25-ე მაგალითი წარმოადგენს წახნაგოვანი ტიპის ობიექტს სურ. 72. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary

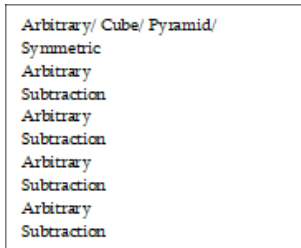


სურ. 72 #25 სატესტო მაგალითი

polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

მეთოდების და გეომეტრიის ფორმის ანალიზის შედეგად მისი დაპროგრამების რამდენიმე კომბინაცია შეიქმნა ცხრ. 27. აღნიშნული ობიექტის მისაღებად აუცილებელია

გამოყენებულ იქნეს Subtraction Boolean ოპერაცია. ამისთვის საწყისი



პრიმიტივი უნდა შეიქმნას Arbitrary, Cube, Pyramid ან Symmetric მეთოდით. რაც შეეხება

პრიმიტივებს მის ამოსაჭრელად, საჭიროა სულ 4 სხვადასხვა პრიმიტივი და მათი

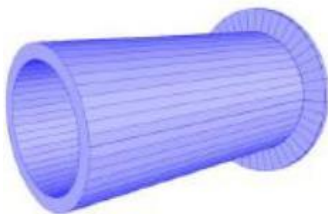
ცხრ. 27 #25 მაგალითის დაპროგრამების მეთოდები

დაპროგრამება უნდა განხორციელდეს Arbitrary მეთოდით. სხვა მეთოდები,

რომლებიც ზემოთ იყო აღწერილი და არ განხილულია, ამ შემთხვევაში არ გამოდგება, მათი გეომეტრიის აგების თავისებურებების გათვალისწინებით.

მაგალითი #26

26-ე მაგალითი წარმოადგენს ცილინდრული ზედაპირის მქონე ობიექტს სურ. 73. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია:



სურ. 73 #26 სატესტო მაგალითი

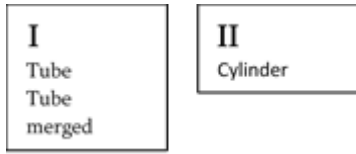
tube, cylinder, chain, combined, merged, subtracted, hybridized. აღნიშნული ობიექტის და მეთოდების

ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 28.

როგორც ცხრილიდან ჩანს 2 სხვადასხვა გზა არსებობს აღნიშნული ობიექტის შესაქმნელად. 1-

ლი ეს არის გზა როცა იქმნება ორი ცილინდრი Tube მეთოდით და შემდეგ

ხდება მათი გაერთიანება merged Boolean ოპერატორით. მე-2 გზა



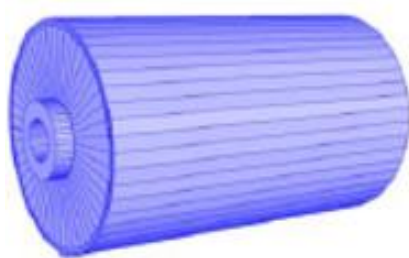
განსხვავებულია, რადგანაც Cylinder მეთოდის საშუალებით 1 ტრანზაქციით იქმნება აღნიშნული ობიექტი. დანარჩენი მეთოდები გარდა განხილულისა ამ შემთხვევისთვის არ გამოდგება, მათი გეომეტრიის აგების

ცხრ. 28 #26 მაგალითის დაპროგრამების

სპეციფიკიდან გამომდინარე.

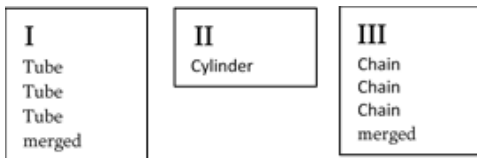
მაგალითი #27

მაგალითი #27-ში არსებული გეომეტრია არის ცილინდრული



სურ. 74 #27 სატესტო მაგალითი

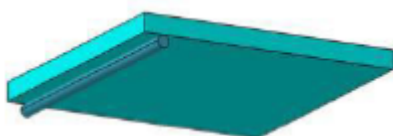
ზედაპირის მქონე სურ. 74. ამიტომ მისი დაპროგრამების შესაძლო მეთოდები წინა შემთხვევის მსგავსია: tube, cylinder, chain, combined, merged, subtracted, hybridized. აღნიშნული მეთოდებისა და ობიექტის ფორმის ანალიზის შემდეგ ჩამოყალიბდა მისი დაპროგრამების რამდენიმე მეთოდი ცხრ. 29. როგორც ცხრილიდან ჩანს ეს შემთხვევა არის თითქმის #26-ის ანალოგიური. განსხვავება მხოლოდ ის არის, რომ პირველ და მესამე გზაში 2 ის ნაცვლად 3 ცილინდრი უნდა შეიქმნას.



ცხრ. 29 #27 მაგალითის დაპროგრამების მეთოდები

მაგალითი #28

28-ე შემთხვევა არის შერეული ტიპის, რადგანაც ამ ობიექტს გააჩნია, როგორც



სურ. 75 #28 სატესტო მაგალითი

წახნაგოვანი, ასევე ცილინდრული ზედაპირი სურ 75. ამიტომ მისი დაპროგრამების შესაძლო ვარიანტებია შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double

symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ობიექტის ფორმის და მეთოდების შესაძლებლობების გამოკვლევის შედეგად მიღებულ იქნა დაპროგრამების მეთოდების რამდენიმე კომბინაცია ცხრ. 30. სურ. 75-ზე წარმოდგენილი ობიექტი შედგება ორი პრიმიტივისგან, რომლებიც საბოლოოდ combined Boolean ოპერატორის საშუალებით ერთ ობიექტად ერთიანდება. 1-ლი ობიექტის ასაგებად გამოყენებულ უნდა იქნეს Arbitrary, Cube, Pyramid ან Symmetric მეთოდი, ხოლო მე-2 პრიმიტივის

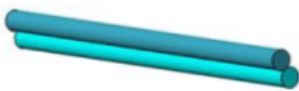
Arbitrary/ Cube/ Pyramid/
Symmetric
Tube/ Cylinder/ Chain
combined

ასაგებად გამოიყენება Tube, Cylinder ან Chain მეთოდი. რაც შეეხება double symmetric polygon, tube, cylinder, chain, subtracted, merged, hybridized მეთოდებს ისინი ამ შემთხვევაში არ

ცხრ. 30 #28 მაგალითის დაპროგრამების მეთოდები გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #29

29-ე მაგალითი წარმოადგენს ცილინდრული ტიპის ობიექტს სურ. 76.



სურ. 76 #29 სატესტო მაგალითი

ამიტომ მისი დაპროგრამების შესაძლო ვარიანტებია: tube, cylinder, chain, combined, merged, subtracted, hybridized. გეომეტრიის ფორმის და მეთოდების ანალიზის შედეგად განხორციელდა

მისი დაპროგრამების მეთოდების რამდენიმე კომბინაციის ფორმირება ცხრ.

31. როგორც ცხრილიდან ჩანს მაგალითის ობიექტი 2 პრიმიტივისგან შედგება, რომლებიც საბოლოოდ combined Boolean ოპერატორის

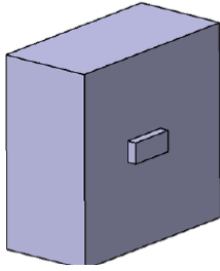
Tube/ Cylinder/ Chain
Tube/ Cylinder/ Chain
combined

საშუალებით ერთ ობიექტად ერთიანდება.

ცხრ. 31 #29 მაგალითის დაპროგრამების მეთოდები უნდა განხორციელდეს: Tube, Cylinder ან Chain. რაც შეეხება merged, subtracted, hybridized მეთოდებს მათი თავისებურებების გამო #29 მაგალითში არ გამოიყენებიან.

მაგალითი #30

30-ე მაგალითი წარმოადგენს წახნაგოვანი ზედაპირის მქონე ობიექტს სურ. 77. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube,



სურ. 77 #30 სატესტო მაგალითი

pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. პრიმიტივების ფორმისა და მეთოდების შესწავლის შემდეგ, მათი დაპროგრამების შემდეგი მეთოდები გამოიკვეთა ცხრ.32. #30 ობიექტი 2 პრიზმისგან შედგება და

შესაბამისად მათი დაპროგრამებისათვის შერჩეულია შემდეგი მეთოდები:

Arbitrary, Cube, Pyramid ან Symmetric. საბოლოოდ უნდა განხორციელდეს ამ

Arbitrary/ Cube/ Pyramid/ Symmetric
Arbitrary/ Cube/ Pyramid/ Symmetric
combined

ცხრ. 32 #30 მაგალითის დაპროგრამების მეთოდები

2 პრიმიტივის გაერთიანება combined მეთოდის საშუალებით. რაც შეეხება double symmetric polygon, merged,

subtracted, hybridized მეთოდებს, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე ამ მაგალითში არ გამოიყენებიან.

მაგალითი #31

31-ე მაგალითს აქვს როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირი. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. აღნიშნული მაგალითის ფორმებისა და მეთოდების ანალიზის შედეგად შეიქმნა მისი დაპროგრამების მეთოდების რამდენიმე კომბინაცია ცხრ. 33. აღნიშნული ობიექტი 2

Tube/ Cylinder
Arbitrary/ Cube/ Pyramid/
Symmetric
Subtraction
Tube/ Cylinder/ Chain
Subtraction
Subtraction
combined

ცხრ. 33 #31 მაგალითის დაპროგრამების მეთოდები

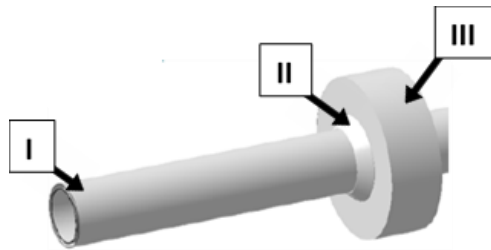
პრიმიტივისგან შედგება და როგორც ირკვევა ამ ობიექტის შესაქმნელად საჭიროა 2 Boolean-ის ოპერაცია: Subtraction და combined. 2 საწყისი პრიმიტივი უნდა აიგოს Tube ან Cylinder მეთოდით, ხოლო მის ამოსაჭრელად საჭიროა პრიმიტივი.

რომელიც უნდა შეიქმნას Arbitrary, Cube, Pyramid ან Symmetric მეთოდით,

საბოლოოდ კი ამ 2 პრიმიტივის გაერთინება ხდება combined ბოლვან ოპერაციის საშუალებით. რაც შეეხება დარჩენილ მეთოდებს: double symmetric polygon, tube, cylinder, chain, merged, hybridized. ისინი გეომეტრიის აგების სპეციფიკიდან გამომდინარე ამ შემთხვევისთვის არ გამოდგებიან.

მაგალითი #32

მაგალითი #32-ში არსებული არის ცილინდრული ზედაპირის მქონე ობიექტი სურ. 78. შესაბამისად მისი დაპროგრამების შესაძლო ვარიანტებია: tube, cylinder, chain, combined, merged, subtracted, hybridized. ობიექტის გეომეტრიული ფორმის ანალიზის შედეგად მისი დაპროგრამების



სურ. 78 #32 სატესტო მაგალითი

მეთოდების ფორმირება განხორციელდა ცხრ. 34. როგორც ცხრილიდან ჩანს აღნიშნული ობიექტი 3 სხვადასხვა პრიმიტივისგან შედგება. 1-ლი და მე-3 პრიმიტივი უნდა შეიქმნას შემდეგი მეთოდებით: Tube ან Cylinder. მე-2 პრიმიტივი კი

უნდა დაპროგრამდეს Cylinder მეთოდით. ბოლოს კი ორივე პრიმიტივი

```

Tube/ Cylinder
Cylinder
Tube/ Cylinder
combined
    
```

ცხრ. 34 #32 მაგალითის დაპროგრამების მეთოდები

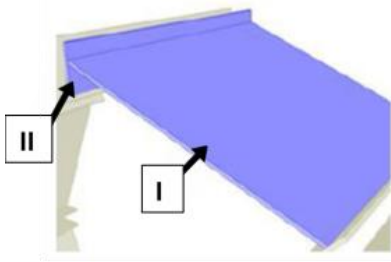
უნდა გაერთიანდეს combined Boolean ოპერატორის საშუალებით. რაც შეეხება ზემოთ ასევე აღწერილ merged, subtracted, hybridized, ისინი მათი სპეციფიკიდან გამომდინარე ამ მაგალითისთვის არ

გამოდგებიან.

მაგალითი #33

33-ე მაგალითი წარმოადგენს წახნაგოვანი ზედაპირის მქონე ობიექტს სურ. 79, რომელიც შედგება 2 სხვადასხვა პრიმიტივისგან. ამიტომ მისი დაპროგრამებისათვის შესაძლოა გამოყენებულ იქნას შემდეგი მეთოდები: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon,

combined, merged, subtracted, hybridized. ობიექტის გეომეტრიული ფორმების



სურ. 79 #33 სატესტო მაგალითი

და მეთოდების შესაძლებლობების ანალიზის შემდეგ ჩამოყალიბდა მისი დაპროგრამების მეთოდების რამდენიმე კომბინაცია ცხრ. 33. 2 სხვადასხვა გზა არის შერჩეული ობიექტის შესაქმნელად. 1 გზის მიხედვით პირველი პრიმიტივი უნდა

შეიქმნას Symmetric ან Arbitrary მეთოდით, ხოლო მე-2 პრიმიტივი კი Arbitrary, Cube, Pyramid ან Symmetric მეთოდით და ბოლოს ეს ორი პრიმიტივი combined Boolean ოპერატორის საშუალებით უნდა გაერთიანდნენ. მე-2 გზის მიხედვით კი 1-ლი პრიმიტივი უნდა შეიქმნას Cube ან Pyramid მეთოდით.

I
Symmetric/ Arbitrary
Arbitrary/ Cube/ Pyramid/
Symmetric
combined

II
Cube/ Pyramid
Arbitrary/ Cube/ Pyramid/
Symmetric
Subtraction
Arbitrary/ Cube/ Pyramid/
Symmetric
combined

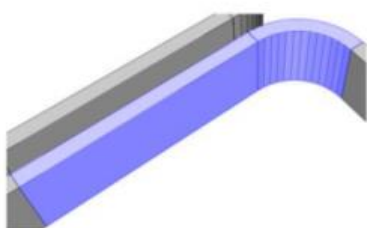
შემდეგ კი Subtraction-ის საშუალებით ჩამოიჭრება პრიმიტივით, რომელიც დაპროგრამება Arbitrary, Cube, Pyramid ან Symmetric მეთოდით. მე-2 პრიმიტივი

ცხრ. 35 #33 მაგალითის დაპროგრამების მეთოდები

კი შეიქმნება Arbitrary, Cube, Pyramid ან Symmetric მეთოდით. საბოლოოდ მათი გაერთიანება უნდა მოხდეს combined Boolean ოპერატორის საშუალებით. სხვა მეთოდები, რომლის განხილვაც არ განხორციელებულა, ამ მაგალითისთვის არ გამოდგებიან მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #34

34-ე მაგალითი არის როგორც ცილინდრული ასევე წახნაგოვანი



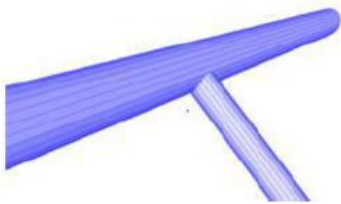
სურ. 80 #34 სატესტო მაგალითი

ზედაპირის მქონე ობიექტი სურ. 80. ის შედგება 2 სხვადასხვა პრიმიტივისგან. მათი დაპროგრამების შესაძლო ვარიანტებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

მეთოდების და პრიმიტივების გეომეტრიული ფორმების ანალიზის შემდეგ აღმოჩნდა რომ მისი დაპროგრამების მეთოდები #28 მაგალითის მეთოდების ანალოგიურია.

მაგალითი #35

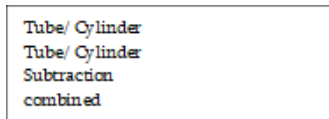
#35 მაგალითში არსებული პრიმიტივი არის ცილინდრული ზედაპირის მქონე სურ. 81. ის შედგება 2 სხვადასხვა ცილინდრისგან.



სურ. 81 #35 სატესტო მაგალითი

შესაბამისად ობიექტის დაპროგრამების შესაძლო მეთოდებია: tube, cylinder, chain, combined, merged, subtracted, hybridized. პრიმიტივების გეომეტრიული ფორმების და მეთოდების ანალიზის შედეგად, დაპროგრამების მეთოდების რამდენიმე

კომბინაცია ჩამოყალიბდა ცხრ. 36. როგორც ცხრილიდან ჩანს 2 პირველ ეტაპზე უნდა დაპროგრამდეს 2 სხვადასხვა ცილინდრი შემდეგი მეთოდებით



ცხრ. 36 #35 მაგალითის დაპროგრამების

Tube ან Cylinder. შემდეგ უნდა განხორციელდეს Subtraction Boolean ოპერაცია. ბოლოს კი პრიმიტივების გაერთიანება განხორციელდება combined Boolean პროცესორის საშუალებით.

სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს ცილინდრული ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #36

36-ე მაგალითი წარმოადგენს როგორც წახნაგოვანი, ასევე



სურ. 82 #36 სატესტო მაგალითი

ცილინდრული ტიპის ობიექტების კომბინაციას სურ. 82. ის შედგება 2 პრიმიტივისგან და მათი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged,

subtracted, hybridized. გეომეტრიული ფორმების და მეთოდების ანალიზის შემდეგ ჩამოყალიბდა სურ.82-ზე წარმოდგენილი ობიექტის შექმნის მეთოდების რამდენიმე კომბინაცია. პირველ რიგში უნდა შეიქმნას პრიმიტივი, რომელსაც შემდეგი დაპროგრამების მეთოდები გააჩნია: Tube ან

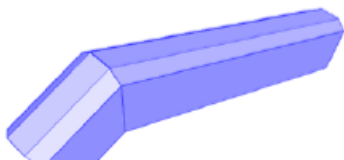
Tube/ Cylinder
Cube/ Arbitrary/ Pyramid/
Symmetric
subtraction
combined

Cylinder. ამის შემდეგ უნდა განხორციელდეს მე-2 პრიმიტივის შექმნა შემდეგი მეთოდების გამოყენებით: Cube, Arbitrary, Pyramid ან

ცხრ. 37 #36 მაგალითის დაპროგრამების მეთოდები Symmetric ცხრ.37. ამის შემდეგ უნდა შესრულდეს subtraction Boolean ოპერაცია და ბოლოს combined მეთოდის საშუალებით ყველა პრიმიტივი უნდა გაერთიანდეს. სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს წახნაგოვანი და ცილინდრული ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #37

37-ე ობიექტი არის წახნაგოვანი ზედაპირის მქონე სურ83. ის 2 სხვადასხვა პრიმიტივისგან შედგება. შესაბამისად მათი შექმნის სავარაუდო



სურ. 83 #37 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

მეთოდების და გეომეტრიული ფორმების ანალიზის შემდეგად დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია

ცხრ. 38. პირველ ეტაპზე უნდა განხორციელდეს 2 პრიმიტივის აგება შემდეგი მეთოდებით: Arbitrary ან Double Symmetric. შემდეგ უნდა განხორციელდეს

Arbitrary/ Double Symmetric
Arbitrary/ Double Symmetric
Cube/ Arbitrary/ Pyramid/
Symmetric
Subtraction
Subtraction
combined

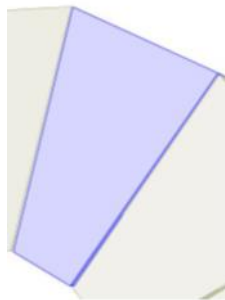
ამ პრიმიტივების ამოჭრა Subtraction-ით. ამისთვის საჭირო პრიმიტივის შექმნა შემდეგი მეთოდებით უნდა მოხდეს: Cube, Arbitrary, Pyramid ან Symmetric. საბოლოოდ კი ყველა პრიმიტივი გაერთიანდება combined

ცხრ. 38 #37 მაგალითის დაპროგრამების მეთოდები

Boolean მეთოდის გამოყენებით. სხვა მეთოდები, რომლის განხილვაც აქ არ განხორციელებულა, ამ მაგალითისთვის არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #38

38-ე მაგალითში წარმოდგენილია წახნაგოვანი ზედაპირის მქონე ობიექტი სურ. 84. შესაბამისად მათი შექმნის სავარაუდო მეთოდებია: cube,



სურ. 84 #38 სატესტო მაგალითი

pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მეთოდების და გეომეტრიული ფორმების ანალიზის შემდეგ დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია ცხრ. 39 როგორც ცხრილიდან ჩანს მოცემული პრიმიტივის აგება 2 გზით არის შესაძლებელი. პირველი გზით ეს

შესაძლებელია Arbitrary ან Symmetric მეთოდის გამოყენებით. მე-2 გზის გამოყენების შემთხვევაში კი აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. ამისათვის საწყისი ობიექტი უნდა შეიქმნას Cube ან Pyramid მეთოდით, ხოლო მისი ამოჭრა უნდა მოხდეს ასევე პრიმიტივით, რომლის დაპროგრამების მეთოდებია: Cube, Arbitrary, Pyramid ან Symmetric. სხვა

I
Arbitrary/ Symmetric

II
Cube/ Pyramid
Cube/ Arbitrary/ Pyramid/
Symmetric

დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს წახნაგოვანი ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ

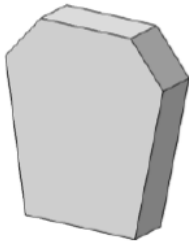
ცხრ. 39 #38 მაგალითის დაპროგრამების მეთოდები

გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #39

39-ე ობიექტი არის წინა შემთხვევის მსგავსად წახნაგოვანი ზედაპირების მქონე სურ. 85. მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მეთოდების და გეომეტრიული

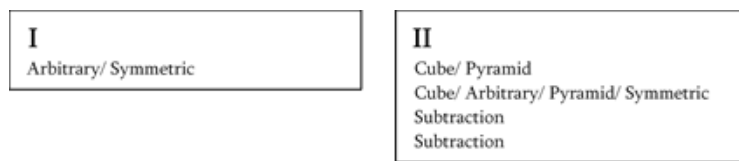
ფორმების ანალიზის შემდეგ დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია ცხრ. 40.



სურ. 85 #39
სატესტო მაგალითი

როგორც ცხრილიდან იკვეთება ეს შემთხვევა თითქმის #38 მაგალითის ანალოგიურია, ერთი განსხვავებით. დაპროგრამების მეთოდების მე-2 გზაში 1 დამატებითი Subtraction Boolean ოპერაცია არის გასაკეთებელი. რაც შეეხება ამისთვის საჭირო პრიმიტივს მისი აგების

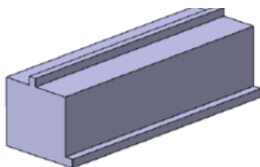
მეთოდები იგივეა რაც წინა შემთხვევაში.



ცხრ. 40 #39 მაგალითის დაპროგრამების მეთოდები

მაგალითი #40

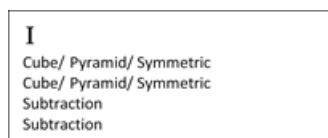
40-ე მაგალითი წარმოადგენს წახნაგოვანი ზედაპირის მქონე ობიექტს სურ. 86 და მისი დაპროგრამების შესაძლო ვარიანტები არის შემდეგი: cube,



სურ. 86 #40 სატესტო
მაგალითი

pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მოცემული ობიექტის გეომეტრიული ფორმების და მეთოდების ანალიზის შემდეგ განხორციელდა მისი დაპროგრამების

მეთოდების რამდენიმე კომბინაციის ფორმირება ცხრ. 41. ამ ობიექტის შექმნის 2 გზა არსებობს. პირველი გზის მიხედვით უნდა შეიქმნას საწყისი ობიექტი Cube, Pyramid ან Symmetric მეთოდის გამოყენებით. იგივე



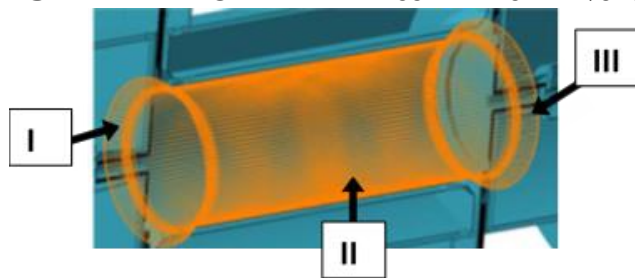
ცხრ. 41 #40 მაგალითის
დაპროგრამების მეთოდები

მეთოდებით უნდა შეიქმნას კიდევ 1 პრიმიტივი, რომლის გამოყენებითაც საწყისი პრიმიტივზე უნდა

განხორციელდეს Subtraction Boolean ოპერაცია. სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს წახნაგოვანი ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #41

41-ე მაგალითში არსებულ ობიექტს გააჩნია ცილინდრული ზედაპირი სურ. 87. ამიტომ მისი შექმნა შესაძლებელია განხორციელდეს შემდეგი



მეთოდებით: tube, cylinder, chain, combined, merged, subtracted, hybridized. განხორციელდა ანალიზი მოცემული ობიექტის ფორმების და იმ მეთოდების, რომლებს იძლევა

სურ. 87 #41 სატესტო მაგალითი

ცილინდრული ზედაპირების შექმნის შესაძლებლობას. შედეგად დადგინდა ის დაპროგრამების მეთოდები, რომლებს სურ. 87-ზე არსებული ობიექტის

I
Cylinder

II
Cylinder
Tube
Cylinder
merged

შექმნის შესაძლებლობას იძლევა ცხრ.42. ეს 2 გზით არის შესაძლებელი. 1-ლი გზის მიხედვით ობიექტის შექმნა შესაძლებელია Cylinder მეთოდის საშუალებით. მე-2 გზის მიხედვით კი მოცემული ობიექტი 3

ცხრ. 42 #41 მაგალითის დაპროგრამების მეთოდები

ნაწილად უნდა აიგოს. 1-ლი და მე-3 ნაწილი Cylinder მეთოდით და მე-2 კი Tube მეთოდის საშუალებით, შემდეგ კი ამ 3 პრიმიტივის გაერთიანება უნდა განხორციელდეს merged მეთოდის გამოყენებით. სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს ცილინდრული ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #42

42-ე მაგალითი წახნაგოვანი ზედაპირის მქონე და შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon,

Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Subtraction

symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

ცხრ. 43 #42 მაგალითის დაპროგრამების მეთოდები

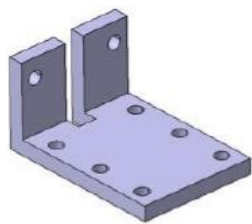
მეთოდების შესაძლებლობების და ობიექტის გეომეტრიული ფორმების ანალიზის

შედეგად დადგინდა გეომეტრიის დაპროგრამების რამდენიმე მეთოდი ცხრ.

43. როგორც ცხრილიდან ჩანს #53 სურათზე მოცემული ობიექტის შესაქმნელად აუცილებელია subtract Boolean ოპერაციის გამოყენება. საწყისი პრიმიტივის დაპროგრამება უნდა განხორციელდეს შემდეგი მეთოდებით: Cube, Arbitrary, Pyramid ან Symmetric. შემდეგ ეს ობიექტი უნდა ამოიჭრას იგივე მეთოდებით შექმნილი პრიმიტივით. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #43

43-ე მაგალითი არის შერეული, როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირების მქონე სურ. 88. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon,



სურ. 88 #43 სატესტო მაგალითი

symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

ობიექტის ზედაპირის და მეთოდების ანალიზის შემდეგ შედგა მისი შექმნის მეთოდების რამდენიმე კომბინაცია

ცხრ. 44. ამ ობიექტის შესაქმნელად 2 გზა შეიძლება იქნეს გამოყენებული. ორივე შემთხვევაში საჭიროა Subtraction Boolean ოპერატორის გამოყენება. განვიხილოთ 1-ლი გზა. ამ შემთხვევაში საწყისი ობიექტი უნდა აიგოს შემდეგი მეთოდებით: Cube, Pyramid ან Symmetric. Subtraction-ისთვის კი 3 სხვადასხვა პრიმიტივი არის საჭირო. პირველი და მე-2 უნდა დაპროგრამდეს Cube, Arbitrary, Pyramid ან Symmetric,

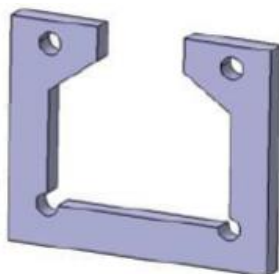
<p>I</p> <p>Cube/ / Pyramid/ Symmetric</p> <p>Cube/ Arbitrary/ Pyramid/ Symmetric</p> <p>Subtraction</p> <p>Cube/ Arbitrary/ Pyramid/ Symmetric</p> <p>Subtraction</p> <p>Tube/ Cylinder/Chain</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p>	<p>II</p> <p>Arbitrary</p> <p>Cube/ Arbitrary/ Pyramid/ Symmetric</p> <p>Subtraction</p> <p>Tube/ Cylinder/ Chain</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p> <p>Subtraction</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ცხრ. 44 #43 მაგალითის დაპროგრამების მეთოდები

ხოლო მე-3 კი Tube, Cylinder ან Chain მეთოდით. რაც შეეხება მე-2 გზას ამ შემთხვევაში განსხვავებულია საწყისი ობიექტის აგების მეთოდი და ამისთვის გამოყენებულია Arbitrary. ამიტომ შემდეგ 1-ით ნაკლების Subtraction არის გასაკეთებელი. ანუ საჭიროა 2 დამხმარე პრიმიტივი. 1-ლის დაპროგრამების მეთოდებია: Cube, Arbitrary, Pyramid ან Symmetric, ხოლო მე-2 პრიმიტივის კი Tube, Cylinder ან Chain. სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს წახნაგოვანი და ცილინდრული ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #44

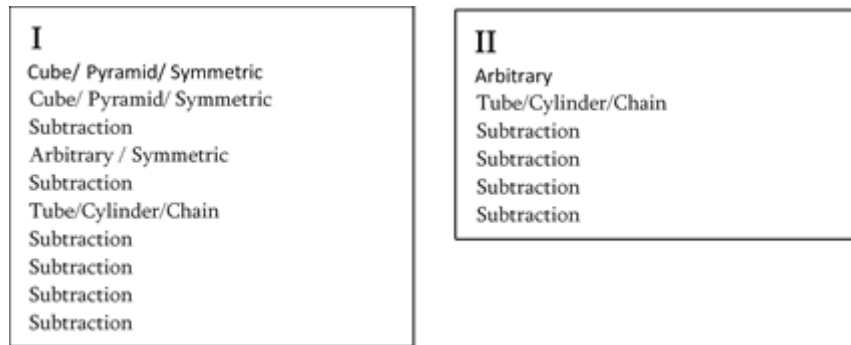
44-ე მაგალითი წინა შემთხვევის მსგავსად წარმოადგენს როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირების მქონე სურ. 89. შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon,



სურ. 89 #44 სატესტო მაგალითი

symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

განხორციელდა ანალიზი მოცემული ობიექტის ფორმების და იმ მეთოდების, რომლების იძლევა ცილინდრული ზედაპირების შექმნის შესაძლებლობას. შედეგად დადგინდა ის დაპროგრამების მეთოდები,



ცხრ. 45 #44 მაგალითის დაპროგრამების მეთოდები

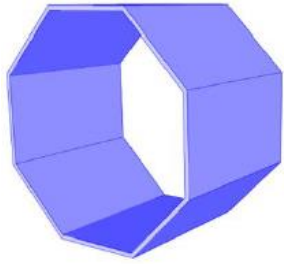
რომლებს სურ. 89-ზე არსებული ობიექტის შექმნის შესაძლებლობას იძლევა ცხრ.45. ეს 2 გზით არის შესაძლებელი და ორივე შემთხვევაში აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. 1-ლი გზის გამოყენების შემთხვევაში საწყისი პრიმიტივი უნა დაპროგრამდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისთვის კი საჭიროა 3 სხვადასხვა პრიმიტივი. მათ შესაქმნელად კი შემდეგი მეთოდები უნდა იქნეს გამოყენებული:

1. Cube, Pyramid, Symmetric ან Arbitrary
2. Arbitrary ან Symmetric
3. Tube, Cylinder ან Chain

რაც შეეხება მე-2 გზას აქ საწყისი ობიექტის შექმნა ხორციელდება Arbitrary მეთოდით და პირველი გზისგან განსხვავებით, მხოლოდ Subtraction არის გასაკეთებელი და მისთვის საჭირო პრიმიტივის დაპროგრამება უნდა მოხდეს Tube, Cylinder ან Chain მეთოდით. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #45

45-მაგალითი არის წახნაგოვანი ზედაპირის მქონე სურ. 90 და შესაბამისად მისი დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. ობიექტის ზედაპირის და მეთოდების ანალიზის



სურ. 90 #45 სატესტო მაგალითი

შემდეგ შედგა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 46. ობიექტის შექმნა 2 გზით არის შესაძლებელი და ორივე შემთხვევაში აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. 1-ლი გზის გამოყენების შემთხვევაში საწყისი პრიმიტივის დაპროგრამება უნდა განხორციელდეს შემდეგი

მეთოდის გამოყენებით: Cube, Pyramid ან Symmetric. დამხმარე პრიმიტივის

I
Cube/ Pyramid/ Symmetric
Arbitrary
Arbitrary
Subtraction
Subtraction

II
Arbitrary
Arbitrary
Subtraction

კი Arbitrary მეთოდით.

სულ 2 დამხმარე პრიმიტივი არის საჭირო.

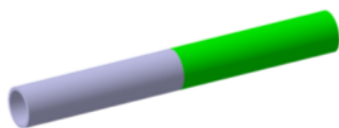
მე-2 გზაში კი საწყისი და დამხმარე ობიექტიც

ცხრ. 46 #45 მაგალითის დაპროგრამების მეთოდები

Arbitrary მეთოდით უნდა შეიქმნას და ამიტომ 1-ით ნაკლები Subtraction-ის გამოყენებაა საჭირო. სხვა დანარჩენი მეთოდი რომელიც ასევე უზრუნველყოფს წახნაგოვანი ზედაპირების მქონე ობიექტების შექმნას, ამ მაგალითისთვის არ გამოდგება მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #46

46-ე მაგალითში არსებული პრიმიტივი წარმოადგენს ცილინდრული ზედაპირის მქონე ობიექტს სურ. 91. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: tube, cylinder, chain, combined, merged, subtracted, hybridized. ამ ობიექტის და დაპროგრამების მეთოდების ანალიზის შედეგად დადგინდა მისი დაპროგრამების სხვადასხვა კომბინაციები ცხრ. 47. სურ. 91-ზე



სურ. 91 #46 სატესტო მაგალითი

წარმოდგენილი ობიექტი შედგება ორი პრიმიტივისგან, რომლებიც საბოლოოდ combined Boolean ოპერატორის საშუალებით ერთ ობიექტად ერთიანდება. ორივე ობიექტის

ასაგებად გამოყენებულ უნდა იქნეს Tube, Cylinder ან Chain მეთოდი.

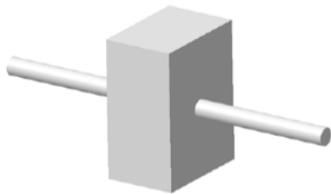
Tube/ Cylinder/ Chain
Tube/ Cylinder/ Chain
combined

ცხრ. 47 #46 მაგალითის დაპროგრამების მეთოდები

რომლებიც საბოლოოდ combined Boolean ოპერატორის საშუალებით ერთ ობიექტად ერთიანდება. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #47

47-ე მაგალითი წარმოადგენს როგორც წახნაგოვანი ასევე ცილინდრული ზედაპირების მქონე სურ. 92. შესაბამისად მისი



სურ. 92 #47 სატესტო მაგალითი

დაპროგრამების შესაძლო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ობიექტის ზედაპირის და მეთოდების

ანალიზის შემდეგ შედგა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 48. სურ. 92-ზე წარმოდგენილი ობიექტი შედგება ორი პრიმიტივისგან,

Cube/ Arbitrary/ Pyramid/ Symmetric
Tube/ Cylinder/ Chain
Subtraction
Tube/ Cylinder/ Chain
combined

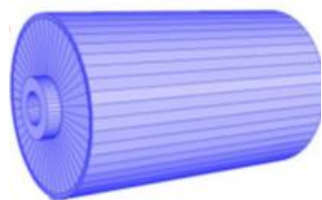
ცხრ. 48 #47 მაგალითის დაპროგრამების მეთოდები

რომლებიც საბოლოოდ combined Boolean ოპერატორის საშუალებით ერთ ობიექტად ერთიანდება. როგორც ცხრილიდან ჩანს 1-ლი ობიექტის შესაქმნელად აუცილებელია

Subtraction Boolean ოპერაციის გამოყენება. ამისათვის საწყისი ობიექტი უნდა დაპროგრამდეს Cube, Arbitrary, Pyramid ან Symmetric მეთოდით, ხოლო მისი ამოჭრა მოხდება პრიმიტივით, რომელიც შემდეგი მეთოდით შეიქმნება: Tube, Cylinder ან Chain. რაც შეეხება 47-ე მაგალითის მე-2 პრიმიტივს, ის უნდა დაპროგრამდეს Tube, Cylinder ან Chain მეთოდით. საბოლოოდ ეს 2 პრიმიტივი combined მეთოდის გამოყენებით ერთ ობიექტად უნდა გაერთიანდეს. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #48

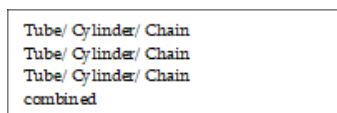
48-ე მაგალითი არის ცილინდრული ზედაპირს მქონე სურ. 93. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: tube, cylinder, chain, combined, merged, subtracted, hybridized. აღნიშნული ობიექტის და მისი დაპროგრამების



სურ. 93 #48 სატესტო მაგალითი

მეთოდების ანალიზის შედეგად დადგინდა მისი დაპროგრამების სხვადასხვა კომბინაციები ცხრ. 49. სურ. 93-ზე წარმოდგენილი ობიექტი შედგება სამი პრიმიტივისგან, რომლებიც საბოლოოდ combined Boolean ოპერატორის საშუალებით ერთ ობიექტად უნდა გაერთიანდეს. სამივე პრიმიტივის

დაპროგრამება უნდა მოხდეს Tube, Cylinder ან Chain მეთოდით. აღნიშნული



ცხრ. 49 #48 მაგალითის დაპროგრამების მეთოდები

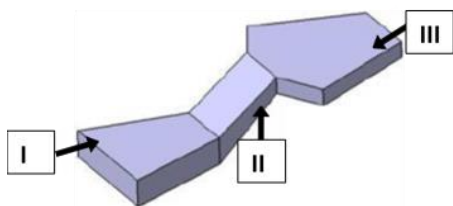
ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #49

49-ე მაგალითი არის ზუსტად 30-ე მაგალითის ანალოგიური. ამიტომ ამ შემთხვევას არ განვიხილავთ.

მაგალითი #50

50-ე მაგალითი არის წახნაგოვანი ზედაპირის მქონე სურ. 94. შესაბამისად მისი დაპროგრამების სავარაუდო მეთოდებია: cube, pyramid, arbitrary polygon,



სურ. 94 #50 სატესტო მაგალითი

symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მეთოდების შესაძლებლობების და ობიექტის გეომეტრიული ფორმების ანალიზის შედეგად დადგინდა გეომეტრიის დაპროგრამების რამდენიმე

მეთოდი ცხრ. 50. სურ. 94-ზე წარმოდგენილი ობიექტი შედგება სამი

Arbitrary/ Symmetric
Arbitrary
Arbitrary/ Symmetric
combined

პრიმიტივისგან, რომლებიც საბოლოოდ

combined Boolean ოპერატორის საშუალებით

ერთ ობიექტად უნდა გაერთიანდეს. 1-ლი

ცხრ. 50 #50 მაგალითის
დაპროგრამების მეთოდები

და მე-3 პრიმიტივები უნდა დაპროგრამდეს

Arbitrary ან Symmetric მეთოდით. ხოლო მე-2 Arbitrary მეთოდით.

საბოლოოდ ეს 3 პრიმიტივი combined მეთოდის გამოყენებით ერთ ობიექტად

უნდა გაერთიანდეს. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები,

რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი

გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #51

51-ე მაგალითი წარმოადგენს როგორც წახნაგოვანი ასევე

ცილინდრული ზედაპირების მქონე სურ. 95.

შესაბამისად მისი დაპროგრამების შესაძლო

მეთოდებია: cube, pyramid, arbitrary polygon,

symmetric polygon, double symmetric polygon, tube,

cylinder, chain, combined, merged, subtracted,

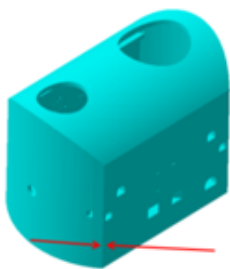
hybridized. მისი დაპროგრამების მეთოდები

შემდეგნაირია ცხრ. 51. საწყისი ობიექტები უნდა

სურ. 95 #51 სატესტო
მაგალითი

აიგოს tube/cylinder/chain და cube/pyramid, arbitrary polygon/symmetric polygon

მეთოდებით და საბოლოოდ მათი გაერთიანება უნდა განხორციელდეს merged



Tube/Cylinder/Chain
Cube/ Arbitrary/ Pyramid/ Symmetric
merged

ცხრ. 51 #51 მაგალითის დაპროგრამების მეთოდები

Boolean მეთოდით.

მაგალითი #52

52-ე მაგალითი არის 51-ე მაგალითის ანალოგიური. ამიტომ ამ შემთხვევას არ

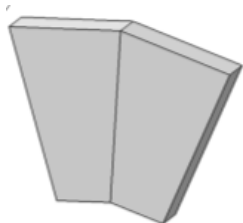
განვიხილავთ.

მაგალითი #53

53-ე მაგალითი არის ზუსტად 28-ე მაგალითის ანალოგიური. ამიტომ ამ შემთხვევას არ განვიხილავთ.

მაგალითი #54

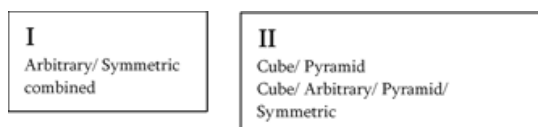
54-ე მაგალითი არის 38- მაგალითის მსგავსი სურ. 96. განსხვავება არის



სურ. 96 #54
სატესტო მაგალითი

ის, რომ 38-ე-ში 1 პრიმიტივია. ხოლო 54-ე-ში კი 2. რაც შეეხება პრიმიტივებს მათი ფორმები ერთნაირია და გააჩნიათ წახნაგოვანი ზედაპირი. მათი დაპროგრამების მეთოდებში განსხვავება არის ის, რომ 54-ე მაგალითში საწყისი პრიმიტივი 2-ჯერ უნდა იქნას გამოყენებული და combined მეთოდის

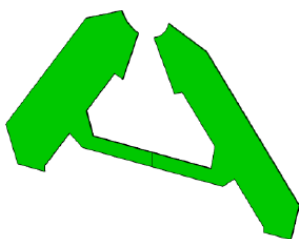
საშუალებით უნდა გაერთიანდნენ ცხრ.52.



ცხრ. 52 #54 მაგალითის დაპროგრამების მეთოდები

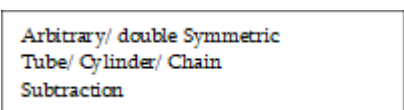
მაგალითი #55

55-მაგალითს გააჩნია როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირი სურ. 97. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია: cube,



სურ. 97 #55 სატესტო
მაგალითი

pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ობიექტის ზედაპირის და მეთოდების ანალიზის შემდეგ შედგა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 53. აღნიშნული ობიექტის



ცხრ. 53 #55 მაგალითის
დაპროგრამების მეთოდები

შექმნისთვის აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. საწყისი ობიექტის დასაპროგრამებლად გამოყენებულ უნდა იქნას Arbitrary ან double Symmetric

მეთოდი. ხოლო მის ამოსაჭრელად პრიმიტივი უნდა დაპროგრამდეს Tube, Cylinder ან Chain მეთოდით. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #56

56-ე მაგალითი არის წახნაგოვანი ზედაპირის მქონე სურ.98. ამიტომ მისი დაპროგრამების შესაბამისი მეთოდებია: cube, pyramid, arbitrary polygon,

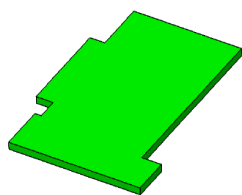


სურ. 98 #56 სატესტო მაგალითი

symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

აღნიშნული ობიექტის და მეთოდების ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების რამდენიმე კომბინაცია. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. 1-ლი გზის მიხედვით საჭიროა Subtraction-ის გაკეთება. ამისთვის საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისთვის საჭიროა 2 სხვადასხვა პრიმიტივის შექმნა, რომლებიც Cube, Arbitrary, Pyramid ან Symmetric მეთოდით უნდა დაპროგრამდეს. მე-2 გზის გამოყენების შემთხვევაში კი 56-ობიექტის ასაგებად მხოლოდ Arbitrary-ის გამოყენებაა საკმარისი. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #57



სურ. 99 #57 სატესტო მაგალითი

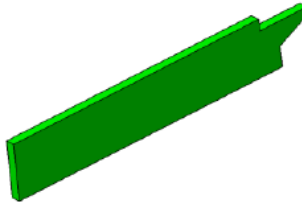
57-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 99. შესაბამისად მისი დაპროგრამების სავარაუდო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted,

I
 Cube/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Subtraction
 Subtraction
 Subtraction
 Subtraction

II
 Arbitrary

ცხრ. 54 #57 მაგალითის დაპროგრამების მეთოდები hybridized. აღნიშნული ობიექტის და მეთოდების ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 54. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. 1-ლი გზის მიხედვით საჭიროა Subtraction-ის გაკეთება. ამისთვის საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისთვის საჭიროა 4 სხვადასხვა პრიმიტივის შექმნა, რომლებიც Cube, Arbitrary, Pyramid ან Symmetric მეთოდით უნდა დაპროგრამდეს. მე-2 გზის გამოყენების შემთხვევაში კი 57-ობიექტის ასაგებად მხოლოდ Arbitrary-ის გამოყენებაა საკმარისი. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #58



58-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 100. შესაბამისად მისი დაპროგრამების სავარაუდო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. აღნიშნული ობიექტის და მეთოდების ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 55. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. 1-ლი გზის მიხედვით საჭიროა Subtraction-ის გაკეთება. ამისთვის საწყისი ობიექტის დაპროგრამება უნდა

სურ. 100 #58 სატესტო მაგალითი

I
 Cube/ Pyramid/ Symmetric
 Arbitrary
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Subtraction
 Subtraction

II
 Arbitrary

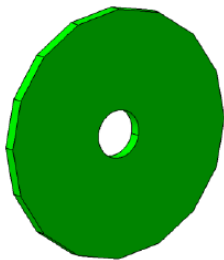
განხორციელდეს Cube,
 Pyramid ან Symmetric
 მეთოდით. Subtraction-
 ისთვის საჭიროა 2
 სხვადასხვა პრიმიტივის

ცხრ. 55 #58 მაგალითის
 დაპროგრამების მეთოდები

შექმნა, პირველი უნდა შეიქმნას Arbitrary მეთოდით. მე-2 კი Cube, Arbitrary, Pyramid ან Symmetric მეთოდით უნდა დაპროგრამდეს. მე-2 გზის გამოყენების შემთხვევაში კი 58-ობიექტის ასაგებად მხოლოდ Arbitrary-ის გამოყენებაა საკმარისი. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #59

59-მაგალითს გააჩნია როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირი სურ. 101. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია:



სურ. 101 #59 სატესტო
 მაგალითი

cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized. ობიექტის ზედაპირის და მეთოდების ანალიზის შემდეგ შედგა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 56. აღნიშნული ობიექტის

შექმნისთვის აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. საწყისი ობიექტის დასაპროგრამებლად გამოყენებულ უნდა იქნას Arbitrary ან double Symmetric მეთოდი. ხოლო მის ამოსაჭრელად პრიმიტივი უნდა დაპროგრამდეს Tube, Cylinder ან Chain

Arbitrary/ double symmetric
 Tube/Cylinder/Chain
 Subtraction

ცხრ. 56 #59 მაგალითის
 დაპროგრამების მეთოდები

მეთოდით. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების

სპეციფიკიდან გამომდინარე.

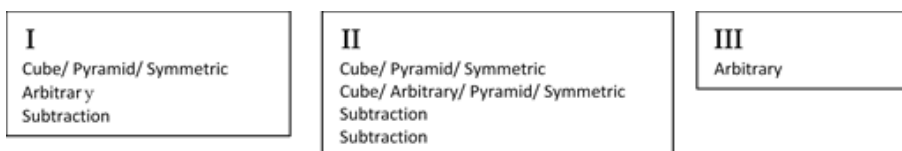
მაგალითი #60

60-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 102. შესაბამისად მისი დაპროგრამების სავარაუდო



სურ. 102 #60 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მეთოდების და გეომეტრიული ფორმების ანალიზის შემდეგად დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია ცხრ.57. 3 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. 1-ლი და მე-2 გზის მიხედვით საჭიროა Subtraction-ის გაკეთება. ამისთვის ორივე შემთხვევაში საწყისი ობიექტის



ცხრ. 57 #60 მაგალითის დაპროგრამების მეთოდები დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისთვის საჭიროა პრიმიტივის შექმნა, პირველი შემთხვევისთვის მისი დაპროგრამების მეთოდია Arbitrary, ხოლო მეორე შემთხვევისთვის Cube, Arbitrary, Pyramid ან Symmetric მეთოდი. მე-3 გზის გამოყენების შემთხვევაში კი 60-ობიექტის ასაგებად მხოლოდ Arbitrary-ის გამოყენებაა საკმარისი. აღიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #61

60-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 103. შესაბამისად



სურ. 103 #61 სატესტო მაგალითი

მისი დაპროგრამების სავარაუდო მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon,

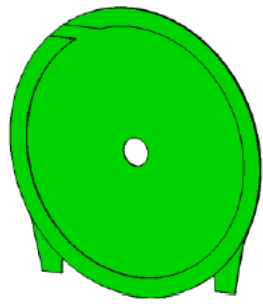
Symmetric/ Arbitrary

combined, merged, subtracted, hybridized.

ცხრ. 58 #61 მაგალითის დაპროგრამების მეთოდები მეთოდების და გეომეტრიის ფორმის ანალიზის შედეგად მისი დაპროგრამების რამდენიმე კომბინაცია შეიქმნა ცხრ. 58. როგორც ცხრილიდან ჩანს მოცემული ობიექტის დაპროგრამება შესაძლებელია Symmetric ან Arbitrary მეთოდით. სხვა მეთოდები რომლების აქ განხილულ არ იქნა, ამ შემთხვევაში არ გამოდგება, მათი გეომეტრიის აგების თავისებურებების გათვალისწინებით.

მაგალითი #62

62-მაგალითს გააჩნია როგორც წახნაგოვანი, ასევე ცილინდრული ზედაპირი სურ. 104. ამიტომ მისი დაპროგრამების შესაძლო მეთოდებია:



cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, tube, cylinder, chain, combined, merged, subtracted, hybridized.

სურ. 104 #62 სატესტო მაგალითი

მეთოდების და გეომეტრიული ფორმების ანალიზის შემდეგ დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია

ცხრ.59. აღნიშნული ობიექტის შექმნისთვის აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. თავდაპირველად უნდა შეიქმნას 2 პრიმიტივი, Tube, Cylinder ან Chain მეთოდით და შემდეგ ისინი უნდა ამოიჭრას ორი სხვადასხვა პრიმიტივით. 1-ლი Tube, Cylinder ან Chain მეთოდით. უნდა შეიქმნას, ხოლო

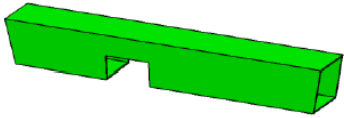
Tube/Cylinder/Chain
Tube/Cylinder/Chain
Subtraction
Tube/Cylinder/Chain
Cube/ Arbitrary/ Pyramid/ Symmetric
Subtraction
Arbitrary

მე-2 Cube, Arbitrary, Pyramid ან Symmetric მეთოდით. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

ცხრ. 59 #62 მაგალითის დაპროგრამების მეთოდები

მაგალითი #63

63-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 105. შესაბამისად მისი დაპროგრამების სავარაუდო



სურ. 105 #63 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

მეთოდების და გეომეტრიული ფორმების ანალიზის შემდეგ დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია ცხრ.60. აღნიშნული ობიექტის შექმნისთვის აუცილებელია Subtraction Boolean ოპერატორის გამოყენება. საწყისი ობიექტის დასაპროგრამებლად გამოყენებულ უნდა იქნას Cube, Arbitrary,

Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Cube/ Arbitrary/ Pyramid/ Symmetric
 Subtraction
 Subtraction
 Subtraction
 Subtraction

Pyramid ან Symmetric მეთოდი. ხოლო მის ამოსაჭრელად საჭიროა 3 პრიმიტივი და ისინი უნდა დაპროგრამდეს Cube, Arbitrary, Pyramid

ცხრ. 60 #63 მაგალითის დაპროგრამების მეთოდები

ან Symmetric მეთოდით. აღნიშნული ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც

ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #64

64-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 106. შესაბამისად მისი დაპროგრამების სავარაუდო



სურ. 106 #64 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. მეთოდების

და გეომეტრიული ფორმების ანალიზის შემდეგ დაგინდა მოცემული ობიექტის დაპროგრამების რამდენიმე კომბინაცია ცხრ. 61. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. ორივე შემთხვევაში საჭიროა Subtraction-ის გაკეთება. 1-ლი გზის საწყისი ობიექტის დაპროგრამება უნდა

I
Arbitrary
Cube/ Arbitrary/ Pyramid/ Symmetric
Subtraction

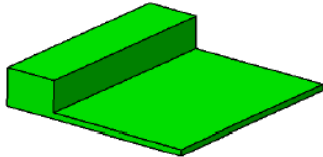
II
Cube/ Pyramid/ Symmetric
Arbitrary
Arbitrary
Subtraction
Subtraction

განხორციელდეს Arbitrary
მეთოდით. Subtraction-
ისტვის საჭიროა 1
პრიმიტივის შექმნა,

ცხრ. 61 #64 მაგალითის დაპროგრამების მეთოდები რომელიც უნდა შეიქმნას Cube, Arbitrary, Pyramid ან Symmetric მეთოდით. მე-2 გზის გამოყენების შემთხვევაში კი 71-ე ობიექტის ასაგებად საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისტვის საჭიროა 2 სხვადასხვა პრიმიტივის შექმნა, რომლებიც უნდა შეიქმნას Arbitrary მეთოდით. ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #65

65-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 107. შესაბამისად მისი დაპროგრამების სავარაუდო



სურ. 107 #65 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized.

აღნიშნული ობიექტის და მეთოდების ანალიზის შედეგად, დადგინდა მისი შექმნის მეთოდების რამდენიმე კომბინაცია ცხრ. 62. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. 1-ლი გზის მიხედვით საჭიროა Subtraction-ის გაკეთება. ამისთვის საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისტვის

I
Cube/ Pyramid/ Symmetric
Cube/ Arbitrary/ Pyramid/ Symmetric
Subtraction

II
Arbitrary

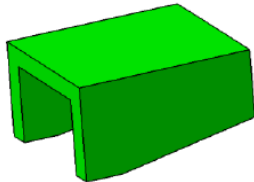
საჭიროა პრიმიტივის
შექმნა, მისი
დაპროგრამების მეთოდია
Cube, Arbitrary, Pyramid ან

ცხრ. 62 #65 მაგალითის დაპროგრამების მეთოდები Symmetric. მე-2 გზის გამოყენების შემთხვევაში კი 65-ობიექტის ასაგებად მხოლოდ Arbitrary-ის გამოყენებაა საკმარისი. აღნიშნული ობიექტის შექმნის

სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

მაგალითი #66

66-მაგალითი წინა შემთხვევის მსგავსად არის წახნაგოვანი ზედაპირის მქონე სურ. 108. შესაბამისად მისი დაპროგრამების სავარაუდო



სურ. 108 #66 სატესტო მაგალითი

მეთოდებია: cube, pyramid, arbitrary polygon, symmetric polygon, double symmetric polygon, combined, merged, subtracted, hybridized. აღნიშნული ობიექტის და მეთოდების ანალიზის შედეგად, დადგინდა მისი შექმნის

მეთოდების რამდენიმე კომბინაცია ცხრ. 63. 2 გზით არის შესაძლებელი მოცემული ობიექტის დაპროგრამება. ორივე შემთხვევაში საჭიროა Subtraction-ის გაკეთება. 1-ლი გზის საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Cube, Pyramid ან Symmetric მეთოდით. Subtraction-ისთვის

I
Cube/ Pyramid/ Symmetric
Cube/ Arbitrary/ Pyramid/ Symmetric
Cube/ Arbitrary/ Pyramid/ Symmetric
Subtraction
Subtraction

II
Arbitrary
Cube/ Arbitrary/ Pyramid/ Symmetric
Subtraction

ცხრ. 63 #66 მაგალითის დაპროგრამების მეთოდები

საჭიროა 2 პრიმიტივის შექმნა, რომლებიც უნდა შეიქმნან Cube, Arbitrary, Pyramid ან Symmetric მეთოდით. მე-2 გზის

გამოყენების შემთხვევაში კი 72-ე ობიექტის ასაგებად საწყისი ობიექტის დაპროგრამება უნდა განხორციელდეს Arbitrary მეთოდით. Subtraction-ისთვის საჭიროა 1 პრიმიტივის შექმნა, რომელიც უნდა შეიქმნას Cube, Arbitrary, Pyramid ან Symmetric მეთოდით. ობიექტის შექმნის სხვა შესაძლო მეთოდები, რომლებიც ზემოთ არის აღწერილი ამ შემთხვევაში არ გამოდგებიან, მათი გეომეტრიის აგების სპეციფიკიდან გამომდინარე.

ამრიგად, კვლევისთვის 66 პრიმიტივი და მათი 2713 დაპროგრამების მეთოდი შეირჩა. შემდგომ ეტაპზე კი განხორციელდა ამ მაგალითების დაჯგუფება კლასებში შემდეგი კრიტერიუმებით:

1. პრიმიტივები, რომელთაც გააჩნიათ წახნაგოვანი ზედაპირები
2. პრიმიტივები, რომელთაც ცილინდრული ზედაპირები გააჩნიათ
3. პრიმიტივები, რომელთაც შერეული (როგორც წახნაგოვანი, ასევე ცილინდრული) ზედაპირები გააჩნიათ

პირველ, წახნაგოვანი ზედაპირების კლასში, სულ 32 პრიმიტივის გაერთიანება განხორციელდა. ესენია: 1, 2, 3, 4, 5, 6, 7, 8, 9, 18, 19, 25, 30, 33, 37, 38, 39, 40, 42, 45, 49, 50, 54, 56, 57, 58, 60, 61, 63, 64, 65, 66. მათი დაპროგრამებისათვის კი სულ 1883 მეთოდია გამოყენებული.

მე-2, ცილინდრული ზედაპირების კლასში გაერთიანებულია 11 პრიმიტივი. ესენია: 12, 15, 26, 27, 29, 31, 32, 35, 41, 46, 48. მათი დაპროგრამებისათვის კი სულ 126 მეთოდია გამოყენებული.

რაც შეეხება მე-3 კლასს, აქ გაერთიანებულია შემდეგი 23 პრიმიტივი: 10, 11, 13, 14, 16, 17, 20, 21, 22, 23, 24, 28, 34, 36, 43, 44, 47, 51, 52, 53, 55, 59, 62. მათი დაპროგრამებისათვის კი სულ 704 მეთოდია გამოყენებული.

2.3 დაპროგრამების მეთოდების ანალიზი

შემდგომი ეტაპის ამოცანას წარმოადგენს მაგალითების განთესვა დაპროგრამების მეთოდების საშუალებით. წინასწარი შესწავლის შედეგად გამოიკვეთა, სატესტო მაგალითებისთვის შერჩეულ მეთოდებს შორის არის სხვადასხვაგვარი კავშირები, მაგალითად ზოგ მათგანს აქვს მსგავსი ტოპოლოგია, ზოგიერთი მათგანი არის თეორიული, ვინაიდან პრაქტიკაში მათი ამ ფორმით გამოყენება არ ხორციელდება. გამოიკვეთა ასევე მეთოდების ირაციონალობაც. გამომდინარე აქედან გეომეტრიული აღწერების დაპროგრამების მეთოდების ანალიზი წარმოადგენს აქტუალური ამოცანას. ამოცანის შესასრულებლად დამუშავდა 3 კრიტერიუმი:

1. მსგავსი ტოპოლოგიების გამორიცხვა
2. თეორიული მეთოდების გამორიცხვა
3. ირაციონალური მეთოდების გამორიცხვა

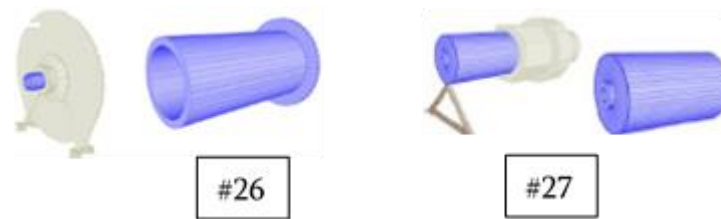
განვიხილოთ თითოეული მათგანი.

2.3.1 მსბავსი ტოპოლოგიების გამოცხვბა

განხორციელდა პრიმიტივების თითოეულ კლასში შემავალი ობიექტების ერთმანეთთან შედარება და შედეგად გამოიკვეთა ის პრიმიტივები, რომლებსაც დაპროგრამების მეთოდების მსგავსი სტრუქტურა გააჩნიათ. განვიხილოთ ასეთი შემთხვევები.

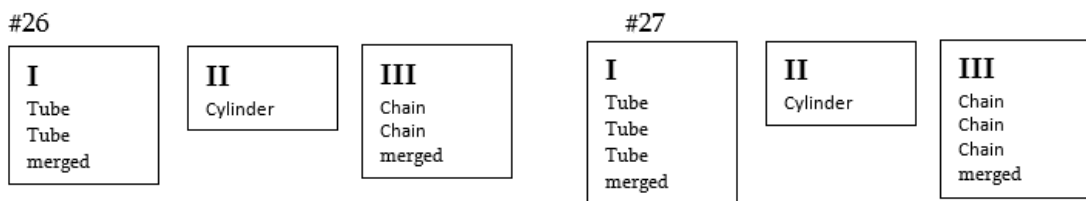
მაგალითი #26 და #27

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #26 და #27, მაგალითები სურ.109. ისინი არიან ცილინდრული ზედაპირების მქონე



სურ. 109 პრიმიტივების სრული ნუსხის #26 და #27 მაგალითები

პრიმიტივების კლასის წარმომადგენლები. შესაბამისად ორივე შემთხვევაში

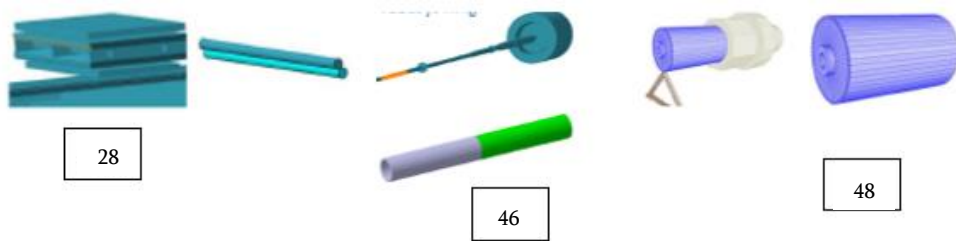


ცხრ. 64 #26, #27 მაგალითების დაპროგრამების მეთოდები მოცემულია ცილინდრული ობიექტები და ამიტომ მათი დაპროგრამებისათვის გამოყენებულია შემდეგი მეთოდები ცხრ. 64. 1. Tube მეთოდი. 2. Cylinder და 3. Chain. ამ ორი მაგალითის დაპროგრამების მეთოდების ანალიზის შედეგად გამოიკვეთა, რომ 26-ე მაგალითთან შედარებით, 27-ე მაგალითის პირველ და მესამე გზაშია განსხვავება. პირველ შემთხვევაში 26-ე მაგალითში Tube მეთოდი 2-ჯერ არის გამოყენებული, ხოლო 27-ში კი 3-ჯერ, იგივე შემთხვევაა მე-3 გზაშიც. განსხვავება არის ის,

რომ აქ გამოყენებულია Chain მეთოდი Tube მეთოდის ნაცვლად. ეს სხვაობა კვლევისთვის მნიშვნელოვანი არ არის, ვინაიდან შემდგომ სიმულაციის სატესტო სესიებში თითოეული მეთოდი გამოყენებული იქნება იმ რაოდენობით, რამდენიც მისი შესაბამისი ობიექტი არის წარმოდგენილი ატლასის ექსპერიმენტის სიმულაციაში. აქედან გამომდინარე #27 პრიმიტივი ამოღებულ უნდა იქნას პრიმიტივების სრული ნუსხიდან.

მაგალითი #29, #46 და #48

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #29, #46, #48



სურ. 110 პრიმიტივების სრული ნუსხის #29, #46 და #48 მაგალითები

მაგალითები სურ.110. აღნიშნული პრიმიტივები წარმოადგენენ ცილინდრული ზედაპირების კლასს. ამ შემთხვევაშიც წინას მსგავსად, მოცემულია ცილინდრული ობიექტები. შესაბამისად მათაც დაპროგრამების მეთოდებიც მსგავსია. განსხვავება Boolean-ის ოპერატორებშია და მათი გაყოფაც ამიტომ მოხდა. რაც შეეხება #29, #46 და #48 პრიმიტივებს მათი დაპროგრამების მეთოდები მოცემულია ცხრ. 65-ში. პრიმიტივების სრულ

#29	#46	#48
Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined

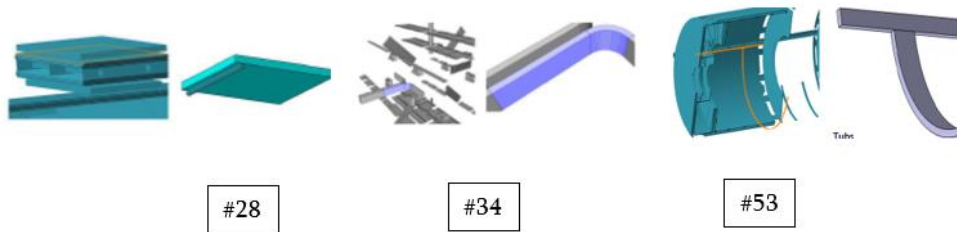
ცხრ. 65 #29, #46 და #48 მაგალითების დაპროგრამების მეთოდები

ნუსხაში მათი აგებისათვის გამოყენებულია შემდეგი მეთოდები ცხრ. 65. 1. Tube მეთოდი, 2. Cylinder და 3. Chain. დაპროგრამების მეთოდების ანალიზის შედეგად გამოიკვეთა, რომ 29-ე და 46-ე პრიმიტივებს ზუსტად ერთნაირი სტრუქტურა აქვთ, ხოლო 48-ე პრიმიტივს კი მათი მსგავსი სტრუქტურა

გააჩნია. აქედან გამომდინარე მათგან მხოლოდ 1 უნდა დარჩეს. პრიმიტივების ნუსხაში დარჩა #29, ხოლო #46 და #48 ამოღებულ იქნა.

მაგალითი #28, #34 და #53

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #28, #34 და #53 მაგალითები სურ.111. ისინი წარმოადგენენ შერეული ზედაპირების



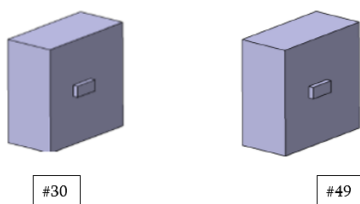
სურ. 111 პრიმიტივების სრული ნუსხის #28, #34 და #53 მაგალითები მქონე პრიმიტივების კლასს და არიან ერთმანეთის მსგავსი ობიექტები. პრიმიტივების ნუსხაში მათი დაპროგრამების მეთოდებად არჩეულია შემდეგი მეთოდები. ცხრ. 66. #28, #34 და #53 მაგალითებში 2 პრიმიტივის დაპროგრამება ხდება ესენია პრიზმის და ცილინდრის საშუალებით. როგორც ცხრ.66-იდან ჩანს თითოეულ მაგალითს დაპროგრამების 12 მეთოდი შეესაბამება რომლებიც ზუსტად ერთნაირია. ამიტომ პრიმიტივების

#28	#34	#53
Arbitrary/ Cube/ Pyramid/ Symmetric Tube/ Cylinder/ Chain combined	Arbitrary/ Cube/ Pyramid/ Symmetric Tube/ Cylinder/ Chain combined	Tube/ Cylinder/ Chain Arbitrary/ Cube/ Pyramid/ Symmetric combined

ცხრ. 66 #28, #34 და #54 მაგალითების დაპროგრამების მეთოდები ნუსხაში დარჩა #28 მაგალითი, #34 და #53 კი წაიშალა.

მაგალითი #30 და #49

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #30 და #49 მაგალითები სურ.112. ეს პრიმიტივები პირველ წახნაგოვანი ზედაპირების



სურ. 112 პრიმიტივების სრული ნუსხის #30, და #49 მაგალითები

კლასს წარმოადგენენ. პრიმიტივების ნუსხაში მათი დაპროგრამებისათვის გამოყენებული მეთოდები მოცემულია ცხრ.67-ში. როგორც ჩანს

#30

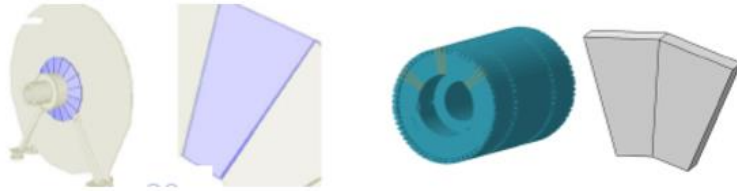
Arbitrary/ Cube/ Pyramid/ Symmetric
Arbitrary/ Cube/ Pyramid/ Symmetric
combined

#49

Arbitrary/ Cube/ Pyramid/ Symmetric
Arbitrary/ Cube/ Pyramid/ Symmetric
combined

ცხრ. 67 #30 და #49 მაგალითების დაპროგრამების მეთოდები თითოეული პრიმიტივის დასაპროგრამებლად 16 სხვადასხვა მეთოდი გამოყენებული. ცხრილიდან ირკვევა, რომ #30 და #49 დაპროგრამების მეთოდების სტრუქტურა ერთი და იგივეა. ამიტომ პრიმიტივების ნუსხაში დარჩა #30 მაგალითი, ხოლო #49 კი ამოღებულ იქნა.

მაგალითი #38 და #54



სურ. 113 პრიმიტივების სრული ნუსხის #38, და #54 მაგალითები

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #38 და #54 მაგალითები სურ.113. ეს პრიმიტივები პირველ წახნაგოვანი ზედაპირების კლასს წარმოადგენენ. ამ მაგალითებში მოცემული პრიმიტივებს

#38

I
Arbitrary/ Symmetric

II
Cube/ Pyramid
Cube/ Arbitrary/ Pyramid/
Symmetric

#54

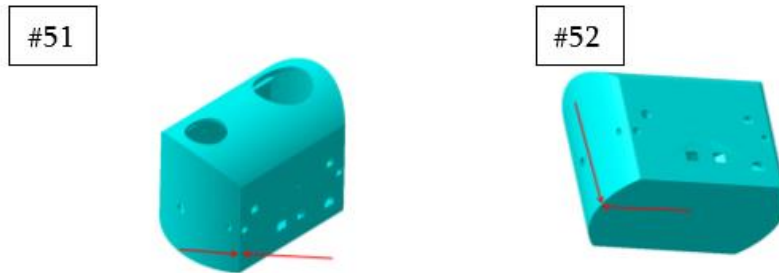
I
Arbitrary/ Symmetric
combined

II
Cube/ Pyramid
Cube/ Arbitrary/ Pyramid/
Symmetric

ცხრ. 68 #38 და #54 მაგალითების დაპროგრამების მეთოდები წარმოადგენს პრიზმა რომლის დაპროგრამებაც შესაძლებელია ორნაირად. ცხრ. 68. თუ შევადარებთ #38 და #54 პრიმიტივების დაპროგრამების პირველ გზას ვნახავთ რომ მათ ერთნაირი სტრუქტურა გააჩნიათ. ასევე ერთნაირი სტრუქტურა აქვთ მე-2 გზის გამოყენების შემთხვევაში. ამიტომ პრიმიტივების ნუსხაში დარჩა #38 მაგალითი, ხოლო #54 კი წაიშალა.

მაგალითი #51 და #52

განვიხილოთ პრიმიტივების სრულ ნუსხაში არსებული #51 და #52 მაგალითები სურ.114. ეს პრიმიტივები პირველ წახნაგოვანი ზედაპირების



სურ. 114 პრიმიტივების სრული ნუსხის #51, და #52 მაგალითები

კლასს წარმოადგენენ. ამ მაგალითებში მოცემული პრიმიტივებს წარმოადგენს პრიმიტივი რომლის დაპროგრამების მეთოდები მოცემულია ცხრ. 69. თუ შევადარებთ #51 და #52 პრიმიტივების დაპროგრამების

#51

Tube/Cylinder/Chain
Cube/ Arbitrary/
Pyramid/ Symmetric
merged

#52

Tube/Cylinder/Chain
Cube/ Arbitrary/
Pyramid/ Symmetric
merged

მეთოდებს აღმოვაჩენთ, რომ ისინი ერთნაირია. ასევე ერთნაირია მათი გეომეტრიული ფორმებიც. ამიტომ პრიმიტივების ნუსხაში დარჩა #51 მაგალითი, ხოლო #52 კი წაიშალა.

ცხრ. 69 #51 და #52 მაგალითების დაპროგრამების მეთოდები

მაგალითი #51 და #52

აღნიშნულ მაგალითებსაც ერთნაირი სტრუქტურა გააჩიათ ცხრ. 70.

#05

Cube/ pyramid
Cube/ pyramid
Subtraction

#07

Cube/ Pyramid
Cube/ Pyramid
Subtraction

ცხრ. 70 #05 და #07 მაგალითების დაპროგრამების მეთოდები

ამრიგად, 1 -ლი კრიტერიუმის მიხედვით პრიმიტივების სრული ნუსხის ანალიზის შედეგად განხორციელდა პრიმიტივების

გაფილტვრა. შედეგად სიიდან ამოღებულ იქნა 7 პრიმიტივი. ესენია: #07, #27, #34, #46, #48, #49, #52, #53, #54. შესაბამისად პრიმიტივების სრულ ნუსხაში დარჩა 58 პრიმიტივი. ამ მეთოდით განხორციელდა მთლიანი სიის გაცხრილვა და შედეგად 2365 მეთოდი შემცირდა 531-მდე. აქედან 207 არის წახნაგოვანი კლასის წარმომადგენელი, 31 ცილინდრული და 293 შერეული.

(იხ. დანართი 2). საბოლოოდ სიიდან ასევე ამოღებულ იქნა შემდეგი მაგალითები: #04, #06, #07, #09, #24, #34, #36, #40, #47, #48, #51, #52, #53, #56, #57, #58

2.3.2 ირაციონალური მეთოდების გამორიცხვა

1-ლი პრიმიტივის დაპროგრამების მეთოდებს შორის ერთ-ერთი გზა არის:

```
Cube/ symmetric/ pyramid  
Cube/ symmetric/ pyramid  
Cube/ symmetric/ pyramid  
Cube/ symmetric/ pyramid  
Cube/ symmetric/ pyramid  
merged
```

აქ ჩანს რომ პრიმიტივის დასაპროგრამებლად 5 დამოუკიდებელი პრიმიტივი უნდა შეიქმნას და შემდეგ Boolean-ის საშუალებით განხორციელდეს მათი გაერთიანება. ამ შემთხვევისთვის სულ 243 სხვადასხვა მეთოდი შეიძლება იქნას გამოყენებული, როცა იგივე პრიმიტივის შექმნა ასევე კიდევ ერთი გზით არის შესაძლებელი და მხოლოდ Arbitrary მეთოდია საკმარისი, აქედან გამომდინარე პირველი შემთხვევა აშკარად შეიცავს ირაციონალურ გზებს, ამიტომ ეს მეთოდი ამოღებულ იქნა. ასევე ამოღებულ იქნა მსგავსი შემთხვევები დანართი 1-დან.

2.3.3 თეორიული მეთოდების გამორიცხვა

ამ შემთხვევის განხილვისას მეთოდების გაფილტვრის კრიტერიუმად გამოყენებულ უნდა იქნეს დაპროგრამების მეთოდების ტოპოლოგია. განვიხილოთ მაგალითი. Cube-ის დაპროგრამების მეთოდი არის შემდეგი:

```
<box name="Box" material="Aluminium" X_Y_Z=""/>
```

როგორც ჩანს ამ მეთოდით Cube იგება 1 ტრანზაქციით „X_Y_Z=“ პრიმიტივის ასაგებად arbitrary მეთოდის გამოყენების შემთხვევაში კი მისი ტოპოლოგია შემდეგნაირია:

```
<gvxy name="Polygon" material="Copper" dZ="1500.">  
<gvxy_point X_Y="-1250.;-1000."/>
```

```

<gvxy_point X_Y="0.;1000."/>
<gvxy_point X_Y="1250.;250."/>
<gvxy_point X_Y="250.;-750."/>
</gvxy>

```

აქ ჩანს, რომ 2 ტრანზაქციაა საჭირო პრიმიტივის დასაპროგრამებლად. 1-ლი „gvxy_point X_Y=" რომლის საშუალებითაც იკვრება პოლიგონი და მე-2 „dZ="" რომლის საშუალებითაც ამ პოლიგონს მიენიჭება მოცულობა. 3 ტრანზაქციაა საჭირო Symmetric მეთოდის გამოყენებისას:

```

<gvxysx name="SymmetricPolygon" material="Carbon"
dZ="1500.">
<gvxy_point X_Y="-1250.;1000."/>
<gvxy_point X_Y=" -500.;-750."/>
</gvxysx>

```

1-ლი ტრანზაქცია არის ის, რომ „gvxy_point X_Y=""-ის საშუალებით მეთოდს უნდა მიეწოდება პოლიგონის კოორდინატები, საკოორდინატო სისტემის ერთ მხარეს, ხოლო მე-2 მხარის კოორდინატებს მეთოდი სარკისებურად ავტომატურ რეჟიმში აიღებს და ეს წარმოადგენს მე-2 ტრანზაქციას. რაც შეეხება მე-3 ტრანზაქციას ეს არის dZ="", რომლის საშუალებითაც პოლიგონს მიენიჭება მოცულობა.

ამრიგად, თეორიული მსჯელობის შედეგად შეგვიძლია დავასკვნათ, რომ განხილული 3 მაგალითიდან 1-ელ შემთხვევას დანარჩენ 2-ზე უკეთესი წარმადობა ექნება, გამომდინარე იქიდან, რომ ამ მეთოდის გამოყენებისას მხოლოდ 1 ტრანზაქცია სრულდება და მისი გამოყენების შემთხვევაში, ლოგიკურად ნაკლები დრო და კომპიუტერული რესურსი იქნება საჭირო. ზემოთ აღწერილი ანალიზის საფუძველზე შესაძლებელია სრული ნუსხაში არსებული მეთოდების გაცხრილვა. განვიხილოთ მაგალითი:

```

Cube/ symmetric/ pyramid
Cube/ symmetric/ pyramid/ arbitrary
Cube/ symmetric/ pyramid/ arbitrary
Subtraction
Subtraction

```

როგორც ჩანს, აქ გამოყენებულია დაპროგრამების შემდეგი მეთოდები Cube, symmetric, pyramid და arbitrary. სულ არის 48 სხვადასხვა კომბინაცია. Symmetric მეთოდი პრიმიტივის შესაქმნელად იყენებს 3 ტრანზაქციას,

arbitrary-2-ს, ხოლო Cube და pyramid-1-ს. აქედან გამომდინარე Symmetric და arbitrary შეგვიძლია აღარ განვიხილოთ. შედეგად მივიღებთ:

Cube/ pyramid
Cube/ pyramid
Cube/ pyramid
Subtraction
Subtraction

მეთოდებს. სადაც 48 კომბინაციის ნაცვლად, დარჩა 8 სხვადასხვა კომბინაცია.

ზემოთ მოცემული მსჯელობის თუ გავავრცელებთ სხვა მეთოდებზეც შეგვიძლია დავასკვნათ, რომ საუკეთესო მეთოდების რეიტინგი ასე გამოიყურება Cube ან pyramid->arbitrary, symmetric ან double symmetric. დანარჩენი მეთოდებისთვის საჭიროა სიმულაციის სატესტო სესიები.

ამრიგად, პრიმიტივების სრულ ნუსხაში დარჩა 42 პრიმიტივი და მათი დაპროგრამების 416 მეთოდი დარჩა (იხ. დანართი 3). აქედან ცილინდრულ კლასში დაჯგუფდა 7 მაგალითი 29 დაპროგრამების მეთოდით, წახნაგოვან კლასში 16 მაგალითი 135 დაპროგრამების მეთოდით და შერეულ კლასში 19 მაგალითი 252 დაპროგრამების მეთოდით.

2.4 მკ-2 თავის დასკვნები

სატესტო მაგალითების ფორმირება განხორციელდა 3 ეტაპად: 1. Atlas-ის დეტექტორის ტიპური გეომეტრიული აღწერების შერჩევა. 2. სატესტო მაგალითებისთვის დაპროგრამების მეთოდების შერჩევა. 3. სატესტო მაგალითების დაპროგრამების მეთოდების ანალიზი. შედეგად მიღებულ იქნა შემდეგი შედეგები:

1. გამოიყო ატლასის დეტექტორის ტიპური გეომეტრიული აღწერების 66 შემთხვევა, რომლებიც 3 კლასში განაწილდნენ. ესენია წახნაგოვანი, ცილინდრული და შერეული წახნაგოვან კლასში განთავსდა 32 მაგალითი, ცილინდრულ კლასში განთავსდა 11 მაგალითი და შერეულ კლასში განთავსდა 23 მაგალითი.

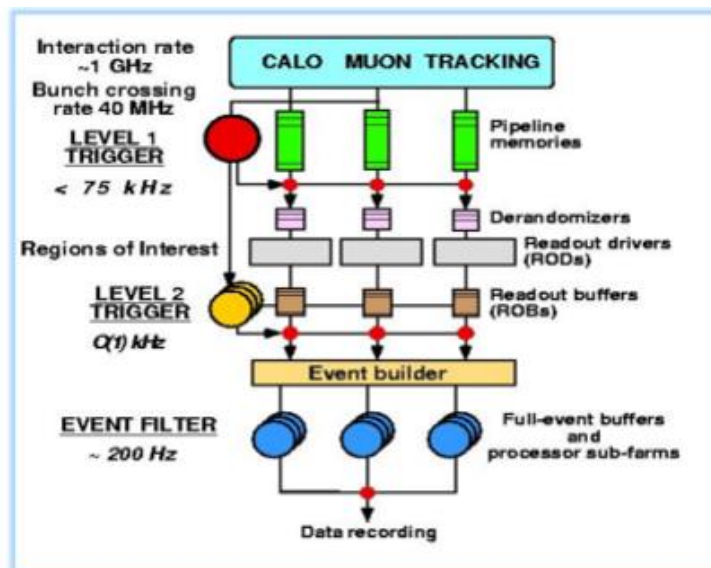
2. 66 მაგალითისთვის შეირჩა დაპროგრამების 2713 მეთოდი, აქედან წახნაგოვანი კლასის წარმომადგენლებისთვის 1883, ცილინდრული კლასის მაგალითებისათვის 126, ხოლო შერეული კლასის მაგალითებისათვის კი 704 დაპროგრამების მეთოდი.
3. განხორციელდა გეომეტრიული აღწერების დაპროგრამების მეთოდების ანალიზი 3 კრიტერიუმის მიხედვით, შედეგად გამოიყო 42 სატესტო მაგალითი თავისი 416 დაპროგრამების მეთოდით, აქედან ცილინდრულ კლასში დაჯგუფდა 7 მაგალითი 29 დაპროგრამების მეთოდით, წახნაგოვან კლასში 17 მაგალითი 135 დაპროგრამების მეთოდით და შერეულ კლასში 18 მაგალითი 252 დაპროგრამების მეთოდით.

თავი III: საკვლევი ინფრასტრუქტურის განსაზღვრა

3.1 სიმულაციის შესასვლელი ინფორმაციის განსაზღვრა

ATLAS-ის დეტექტორში ხორციელდება პროტონების დაჯახება. ისინი დაჯახების შედეგად იშლებიან და წარმოიქმნება მრავალი სხვადასხვა ნაწილაკი, რომლებიც სივრცეში სხვადასხვა მიმართულებით იფანტებიან. ამ პროცესს უწოდებენ ივენოს [40]. ერთი ივენი დამოუკიდებლად ვერ დარეგისტრირდება. იმისთვის, რომ ივენოები დარეგისტრირდეს, მათ უნდა გადალახონ რაოდენობის მინიმალური ზღვარი და მხოლოდ ამის შემდეგ ჩაიწერებიან Run-ებში. Run უწოდება ტრიგერული სისტემების მიერ გაფილტრულ ივენოებს. ტრიგერული სისტემები ახდენენ დეტექტორიდან ინფორმაციის მიღებას და დამუშავებას [37] [38]. ATLAS-ის ექსპერიმენტის ტრიგერული სისტემები შემდეგნაირად არიან ორგანიზებული სურ 115.

1. პირველი დონის ტრიგერი
2. მაღალი დონის ტრიგერი
3. მონაცემთა ნაკადის სისტემა (DFS- DataFlow System)



სურ. 115 ATLAS-ის ექსპერიმენტის ტრიგერული სისტემები

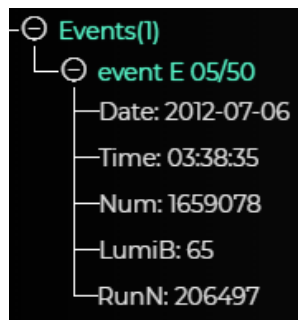
1-ლი დონის ტრიგერის საშუალებით მონაცემების გაფილტვრა ხორციელდება აპარატურულ დონეზე. როგორც სურ. 116-დან ჩანს ის 3 ნაწილისგან შედგება: 1. Calorimeter Trigger; 2. Muon Trigger და 3.

ცენტრალური ტრიგერული პროცესორი (CTP- Central Trigger Processor). 1-ლი დონის ტრიგერული სისტემები ინფორმაციას იღებენ Calorimeter, Muon დეტექტორებიდან და ცენტრალური გამშვები პროცესორი აერთიანებს სიგნალებს ლოგიკურად, ტრიგერის მენიუს შესაბამისად.

მაღალი დონის ტრიგერი კი ნაწილაკების გაფილტვრას ახდენს პროგრამულად. პროგრამა ახორციელებს ნაწილაკების ტრაექტორიის აღდგენას და შემდეგ საინტერესო ნაწილაკების გაფილტვრას კინემატიკური მახასიათებლების მიხედვით.

მონაცემთა ნაკადის სისტემა (DFS) კი ახორციელებს იმ მონაცემთა შენახვას რომლებმაც მაღალი დონის ტრიგერული სისტემების ფილტრაცია წარმატებით გაიარეს [39].

Run-ების იდენტიფიცირება შესაძლებელია Run Number-ის საშუალებით. სურ. 116-ზე მოცემულია მაგალითი Run #206497. რომელიც



სურ. 116 ივენთის მახასიათებელი პარამეტრები

შეიცავს 50 ივენთს, ხოლო სურათზე კი მოცემულია მე-5 ივენთი. Run არის ნაწილაკების დაჯახების პროცესი, რომელიც რამდენიმე საათს გრძელდება [40]. პროტონების აჩქარება LHC-ში ხორციელდება ე.წ. კონების სახით, რომლებსაც ბიმებს უწოდებენ [41]. ბიმს

შესაძლებელია დროის სხვადასხვა მონაკვეთში შესაძლოა ქონდეს სხვადასხვა ენერგია, ამიტომ დეტექტორის და ბიმის მდგომარეობა არის ცვალებადი დროის სხვადასხვა პერიოდში. შედეგად Run-ებს იყოფა Luminosity ბლოკებად [40][42], Luminosity ბლოკებში გაერთიანებულნი არიან ისეთი ივენთები, რომლებიც დეტექტორის და ბიმის მდგომარეობის შეცვლამდე დაფიქსირდა, შესაბამისად Luminosity ბლოკების ხანგრძლივობა რამდენიმე წუთს გრძელდება. მათი იდენტიფიცირება ხდება LBN (Luminosity Block Number) პარამეტრით. ამავე სურათიდან ჩანს რომ ივენთი

ხასიათდება კიდევ რამდენიმე პარამეტრით, ესენი: Date, Time და Num. Data არის პარამეტრი რომლის საშუალებითაც დგინდება ივენთის შექმნის თარიღი, ხოლო Time ის საშუალებით, ივენთის შექმნის დროის იდენტიფიცირებაა შესაძლებელი. რაც შეეხება Num პარამეტრს, აქ მოცემულია ივენთის ნომერი.

Atlas-ის ექსპერიმენტის სიმულაცია ხორციელდება ცერნში არსებულ lxplus-ის სერვერებზე. მისი გაშვება შესაძლებელია სპეციალური სერვისის ფაილით, რომელშიც პითონის სკრიპტია ჩაწერილი სურ. 117. როგორც

```
AtlasG4_tf.py \  
--inputEVNTFile /afs/cern.ch/user//share/OUT_EVNT_50.root \  
--geometryVersion 'default:ATLAS-R3S-2021-01-00-00_VALIDATION' \  
--conditionsTag 'default:OFLCOND-RUN12-SDR-25' \  
|--outputHITSfile OUT_HITS.root
```

სურ. 117 სიმულაციის გამშვები პითონის სკრიპტი

სურათიდან ჩანს სიმულაციის ჩასატარებლად საჭიროა ხელოვნურად, მონტეკარლოს გენერატორების მიერ შექმნილი მოვლენები (events), დეტექტორის გეომეტრიული აღწერები და Conditions Tag.

სიმულაციის პროცესისთვის მოვლენა წინასწარ არის შექმნილი სურ.117-ზე ჩანს, რომ მასზე მიმართვა ხდება --inputEVNTFile ბრძანებით. ეს ბრძანება მიმართავს სპეციალურ ფაილს, ამ შემთხვევაში მისი სახელია OUT_EVNT_50.root რომელიც განთავსებულია სერვერზე და მასში ჩაწერილია ივენთები. გეომეტრიულ აღწერებზე მიმართვა ხდება -- geometryVersion ბრძანებით. ეს ბრძანება იძახებს გეომეტრიულ აღწერებს მონაცემთა ბაზებიდან. ამ შემთხვევაში კი ის იყენებს გეომეტრიული აღწერების „ATLAS-R3S-2021-01-00-00_VALIDATION“ ვერსიას. ვინაიდან ის შემდგომში მაინც უნდა ჩანაცვლდეს ჩვენი გეომეტრიული აღწერებით არ აქვს მნიშვნელობა რომელ ვერსიას გამოვიყენებთ. სასურველი გეომეტრიის არჩევა შესაძლებელია ატლასის დეტექტორის გეომეტრიული აღწერების მონაცემთა ბაზიდან (სურ. 118), რომელიც აღნიშნულ მისამართზეა განთავსებული <https://atlas-geometry-db.web.cern.ch/atlas-geometry->

- [ATLAS-R2-2016-01-00-17.doc](#)
- [ATLAS-R2-2016-01-00-18.doc](#)
- [ATLAS-R2-2016-01-01-00.doc](#)
- [ATLAS-R3-2021-00-00-00.doc](#)
- [ATLAS-R3-2021-01-00-00.doc](#)
- [ATLAS-TEST-ASYM.doc](#)
- [ATLAS-TEST-ATLASCOMMON.doc](#)
- [ATLAS-TEST-CAVERN1.doc](#)
- [ATLAS-TEST-SFCAL.doc](#)
- [ATLAS-TEST-SOLD.doc](#)
- [ATLAS-TEST-SYMM.doc](#)

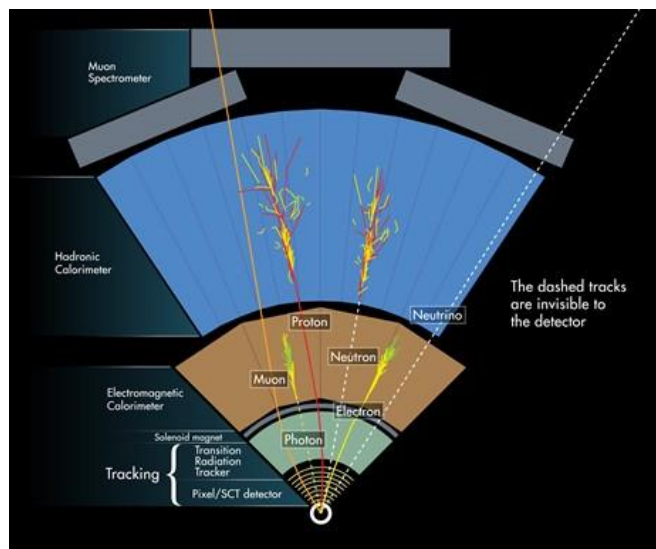
[db/tag_hierarchy_browser.php](#).

ასევე

მონაცემთა ბაზიდან ხდება „conditionsTag“-ის ამოღება. --conditionsTag ბრძანებას „ATLAS Condition Base“ მონაცემთა ბაზიდან მოაქვს ინფორმაცია, დროის კონკრეტულ მონაკვეთში დეტექტორის მდგომარეობის შესახებ [43][44].

სურ. 118 ფრაგმენტი ატლასის დეტექტორის გეომეტრიული აღწერების მონაცემთა ბაზიდან

როგორც ზემოთ იქნა განხილული სიმულაციის შესასვლელი ერთ-ერთი მთავარი პარამეტრია Run, დეტექტორის გეომეტრიასთან ერთად, ამიტომ მისი სწორად შერჩევა უმნიშვნელოვანესი საკითხია. აღნიშნული



სურ. 119 დეტექტორში ნაწილაკების გავრცელების სქემა

კვლევისთვის შესაძლებელია შერჩეულ იქნეს Run, რომელიც შეიძლება შეიცავდეს ელექტრონებს, პროტონებს, ნეიტრონებს (ნეიტრალური ადრონი), ნეიტრონებს, ფოტონებს და მიონებს [4]. სურ.119-ზე-იდან ჩანს, რომ ელექტრონები ნაწილაკების დაჯახების შედეგად წარმოიქმნებიან Z0-ში და შემდეგ ვრცელდებიან დეტექტორში, მათი გავრცელების ზონაა მაქსიმუმ ელექტრომაგნიტური კალორიმეტრი [45]. ამავე სურათზე წყვეტილი კონტურით აღნიშნულია ფოტონი, რომელიც ელექტრონის მსგავსად ელექტრომაგნიტურ კალორიმეტრში ვრცელდება და შემდეგ მიიღევა. ასევე მოცემულია პროტონები, რომლებიც ადრონულ კალორიმეტრამდე აღწევენ

და შემდეგ მიიღვიან. რაც შეეხება ნეიტრონს, ისიც პროტონის მსგავსად ადრონულ კალორიმეტრში მიიღვივ. სურათზე ასევე მოცემულია ნაწილაკი, მიონი. მიონი ერთადერთი ნაწილაკია, რომელიც დეტექტორის ყველა კომპონენტში ვრცელდება დაწყებული შიდა დეტექტორიდან, დამთავრებული მიონური სისტემებით. თუ ჩვენ კვლევისთვის ავიღებთ, ივენთს, რომელიც შეიცავს ელექტრონებს, როგორც ზემოთ არის განხილული, ის ელექტრომაგნიტურ კალორიმეტრში მიიღვივ და შესაბამისად არის დიდი შანსი იმის, რომ ის ყველა გეომეტრიულ ობიექტთან ურთიერთქმედებას ვერ მოახერხებს, იგივე შედეგი იქნება პროტონების შემთხვევაშიც, რადგანაც ის მხოლოდ ადრონულ კალორიმეტრამდე აღწევს. მხოლოდ ერთი ნაწილაკია ისეთი, რომელსაც შეუძლია მთელი დეტექტორის გამჭოლად გავლა და შესაბამისად, ყველა გეომეტრიულ ობიექტთან ურთიერთქმედების ალბათობაც გაცილებით დიდია. ეს ნაწილაკია მიონი. ამიტომ კვლევისათვის შერჩეულ იქნა Run (#222500), რომლის ივენთებიც შედგება მიონებისგან და მისი შესაბამისი ფაილი განთავსებულია lxplus-ზე. აღნიშნული ფაილის განთავსება შესაძლებელია სერვერზე ნებისმიერ ფოლდერში.

კვლევის ჩასატარებლად უნდა ვიყოთ დარწმუნებულები, რომ სიმულაციის პროცესი წარმატებით დასრულდა. ამიტომ სიმულაციის პროცესის გამოკვლევა აქტუალური ამოცანაა. სიმულაციის პროცესის გამოკვლევა უნდა განხორციელდეს, როგორც default პარამეტრებით, აგრეთვე ჩვენს მიერ მიწოდებული Run-ებით და გეომეტრიული აღწერებით. ეს მნიშვნელოვანია გამომდინარე იქიდან, რომ საბოლოოდ წარმატობის კვლევა უნდა განხორციელდეს ჩვენს მიერ შერჩეული გეომეტრიული აღწერებით.

სიმულაციის სესიის ჩატარების შემდეგ ხორციელდება Log ფაილის გენერაცია, რომელსაც ქვია log.AtlasG4Tf. სადაც მოცემულია ჩატარებული სიმულაციის პროცესის შესახებ სხვადასხვა ინფორმაცია. ამავე ფაილშია ინფორმაცია სიმულაციის პროცესის წარმატებით, ან წარუმატებლად დასრულების შესახებ. სურ. 120-ზე წარმოდგენილია default პარამეტრებით

```

7032 08:46:47 Py:PerfMonSvc      INFO --> [ntuple.pmon.dat] => 16.000 kb
7033 08:46:47 Py:PerfMonSvc      INFO --> [ntuple.pmon.stream] => 204.994 kb
7034 08:46:47 Py:PerfMonSvc      INFO --> [ntuple.pmon.pmonsd.txt] => 82.523
7035 08:46:47 Py:PerfMonSvc      INFO Writing out collected data... [ntuple.pmon
7036 08:46:47 ApplicationMgr    INFO Application Manager Finalized successfully
7037 08:46:47 ApplicationMgr    INFO Application Manager Terminated successfull
7038 08:46:47 Py:Athena          INFO leaving with code 0: "successful run"
7039 08:46:48 CoralApplication Info Delete the COOL CoralApplication...
7040 08:46:48 CoralApplication Info Delete the COOL database service
7041 08:46:48 RalDatabaseSvc Info Delete the RalDatabaseSvc...
7042 08:46:48 RalDatabaseSvc Info Purge the connection pool
7043 08:46:48 RalDatabaseSvc Info Reset the ICS pointer
7044 08:46:48 RalDatabaseSvc Info Delete the RalDatabaseSvc... DONE
7045 08:46:48 CoralApplication Info Delete the CORAL connection service
7046 08:46:48 CoralApplication Info Delete the COOL CoralApplication... DONE
7047 08:46:48 CORAL/Services/ConnectionService Info Deleting the ConnectionPool
7048

```

სურ. 120 ფრაგმენტი წარმატებული სიმულაციის log ფაილიდან

ჩატარებული წარმატებული სიმულაციის log ფაილის ფრაგმენტი. წარმატებული პროცესის log ფაილი მოიცავს 7048 სტრიქონს.

წარმატებული პროცესის გამოწვევა შესაძლებელია ხელოვნურად გეომეტრიის აქტიური კომპონენტების გათიშვით. სურ. 121-ზე 1295-ე

```

1278 08:44:53 ToolSvc.NewSmall... INFO -----
1279 08:44:53 ToolSvc.NewSmall... INFO Sections
1280 08:44:53 ToolSvc.NewSmall... INFO -----
1281 08:44:53 ToolSvc.NewSmall... INFO Volumes
1282 08:44:53 ToolSvc.NewSmall... INFO volume NewSmallWheel
1283 08:44:53 ToolSvc.NewSmall... INFO -----
1284 08:44:53 CORAL/Services/ConnectionService Info Re-ordering 3 replicas usi
1285 08:44:53 CORAL/Services/ConnectionService Info Sorting algorithm selected
1286 08:44:53 CORAL/Services/ConnectionService Warning Failed to connect to se
1287 08:44:53 CORAL/Services/ConnectionService Info Connection to service "geo
1288 08:44:53 CORAL/Services/ConnectionService Warning Failure while attemptin
1289 08:44:53 CORAL/RelationalPlugins/frontier Info The current compression lev
1290 08:44:53 CORAL/Services/ConnectionService Info New connection to service
1291 08:44:53 CORAL/Services/ConnectionService Info New user session with sess
1292 08:44:53 ToolSvc.NewSmall... INFO this is NSWAGDDTool::construct()!!!!
1293 08:44:53 ToolSvc.NewSmall... INFO Name = ToolSvc.NewSmallWheel
1294 08:44:53 ToolSvc.NewSmall... INFO trying to parse files
1295 08:44:53 ToolSvc.NewSmall... INFO Building NSW Readout Geometry
1296 08:44:53 AGDDtoGeoSvc      INFO initializing
1297 08:44:53 EvtIdModifierSvc  INFO Initializing EvtIdModifierSvc...
1298 08:44:53 ActiveStoreSvc    INFO Initializing ActiveStoreSvc - package
1299 08:44:53 AthenaEventLoopMgr INFO Initializing AthenaEventLoopMgr - pack
1300 08:44:53 AthMasterSeq      INFO Member list: AthSequencer/AthAlgSeq, A
1301 08:44:53 ClassIDSvc        INFO getRegistryEntries: read 1431 CLIDReg
1302 08:44:53 ClassIDSvc        INFO getRegistryEntries: read 853 CLIDRegi
1303 08:44:53 AthAlgSeq         INFO Member list: TimingAlg/SimTimerBegin,
1304 08:44:53 SimTimerBegin     INFO number of CPUs found: 10 of type Genui

```

სურ. 121 ფრაგმენტი წარმატებული სიმულაციის log ფაილიდან

სტრიქონიდან იწყება ჩვენს მიერ მიწოდებული NSW-ის გეომეტრიის აგება სიმულაციისათვის და ვინაიდან ის ხელოვნურად იყო გათიშული, პროცესი დასრულდა წარმატებლად სურ 122.

წარმატებული სიმულაციის შემთხვევაში ასევე ხორციელდება ფაილის გენერაცია, რომელსაც ქვია HITS.root. აღნიშნული ფაილი

```

1295 08:41:31 ToolSvc.NewSmal... INFO Building NSW Readout Geometry
1296 08:41:31 -----
1297 08:41:31 Core dump from CoreDumpSvc on lxplus609.cern.ch at Mon Nov 18 08:41:31 2019
1298 08:41:31 -----
1299 08:41:31 Caught signal 11(Segmentation fault). Details:
1300 08:41:31 errno = 0, code = 1 (address not mapped to object)
1301 08:41:31 pid = 24, uid = 0
1302 08:41:31 value = (420573184, 0x19117000)
1303 08:41:31 vmem = 1270.4 MB
1304 08:41:31 rss = 735.863 MB
1305 08:41:31 total-ram = 29378.2 MB
1306 08:41:31 free-ram = 237.031 MB
1307 08:41:31 buffer-ram= 223.707 MB
1308 08:41:31 total-swap= 10240 MB
1309 08:41:31 free-swap = 3614.73 MB
1310 08:41:31 addr = 0x18
1311 08:41:31 -----
1312 08:41:31 Event counter: 0
1313 08:41:31 Current algorithm: <NONE>
1314 08:41:31 Algorithm stack: <EMPTY>
1315 08:41:31 -----
1316 08:41:31 | AtlasBaseDir : /cvmfs/atlas-nightlies.cern.ch/repo/sw/21.3/2019-11-17T2149 |
1317 08:41:31 | AtlasVersion : 21.3.16 |
1318 08:41:31 | CMTCONFIG : x86_64-slc6-gcc62-opt |
1319 08:41:31 -----
1320 08:41:31 Note: to see line numbers in below stacktrace you might consider running following :
1321 08:41:31 atlasAddress2Line --file <logfile>
1322 08:41:31 -----
1323 08:41:31
1324 08:41:31 *** Break *** segmentation violation
1325 08:41:52
1326 08:41:52
1327 08:41:52
1328 08:41:52 -----
1329 08:41:52 There was a crash.

```

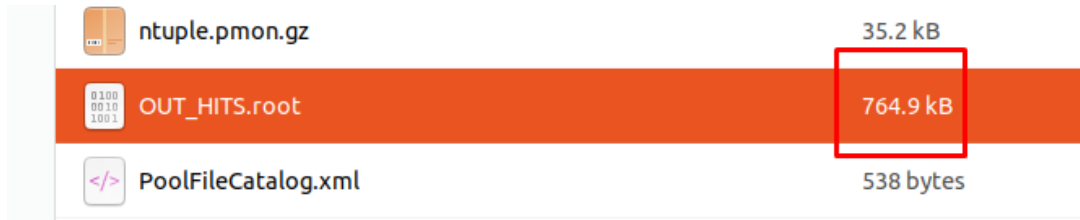
სურ. 122 ფრაგმენტი ხელოვნურად გამოწვეული წარუმატებელი სიმულაციის log ფაილიდან წარმოადგენს სიმულაციის შემდგომი ეტაპის შესასვლელ ინფორმაციას. წარუმატებელი სიმულაციის შემთხვევაში კი მისი შექმნა არ ხდება. აქედან გამომდინარე, თუ სიმულაციის ჩატარების შედეგად მივიღებთ log ფაილს შეცდომების გარეშე და ასევე, HITS.root ფაილს და შეგვიძლია დავასკვნათ, რომ სიმულაციის პროცესი წარმატებით დასრულდა.

როგორც ზემოთ არის აღწერილი კვლევის ჩასატარებლად აუცილებელია განხორციელდეს default გეომეტრიული აღწერების ჩანაცვლება ჩვენს მიერ შერჩეული გეომეტრიული აღწერებით, ამისათვის



სურ. 123 ჩვენი გეომეტრიული აღწერებით ჩატარებული სიმულაციის შედეგი უნდა შეიქმნას პითონის სკრიპტი სურ. 123, შემდგომ ის გამოძახებული იქნება სიმულაციის გამშვებ ფაილში, რომელიც default გეომეტრიულ აღწერებს ჩანაცვლებს ჩვენს მიერ xml-ში შექმნილ გეომეტრიულ აღწერებით. ამ სკრიპტის მოვალეობაა ის რომ წაიკითხოს ინფორმაცია xml ფაილიდან. რომელიც სურ. 123-ზე არის მონიშნული. სიმულაციის პროცესის გაშვებისას იტვირთება ის გეომეტრია, რომელიც თავდაპირველად იყო

ნაგულისხმევი და შემდგომ ხდება მისი ჩანაცვლება ჩვენი გეომეტრიული აღწერებით. Default პარამეტრებით სიმულაციის შედეგად მიღებული OUT_HITS.root ფაილის ზომა არის 764,9 კბ, სურ. 124. ხოლო ჩვენი



სურ. 124 default პარამეტრების ჩატარებული სიმულაციის შედეგი

გეომეტრიული აღწერებით ჩატარებული სიმულაციის შედეგად მიღებული იგივე ფაილის ზომა არის 498,2კბ, სურ.125 ანუ ჩვენმა გეომეტრიულმა

```

from GeoModelSvc.GeoModelSvcConf import GeoModelSvc
GeoModelSvc = GeoModelSvc()
GeoModelSvc.MuonVersionOverride="MuonSpectrometer-R.09.00.NSW"
from AthenaCommon import CfgMgr
from AthenaCommon.AppMgr import ToolSvc

from MuonAGDD.MuonAGDDConf import NSWAGDDTool
NSWAGDDTool = CfgMgr.NSWAGDDTool("NewSmallWheel", ReadAGDD=False, XMLFiles=["amdb.xml"])
ToolSvc+= NSWAGDDTool
    
```

სურ. 125 ფრაგმენტი default პარამეტრების ჩასანაცვლებელი პითონის სკრიპტიდან

აღწერებმა გარკვეული გავლენა იქონია შედეგზე.

3.2 სიმულაციის პროცესის წარმადობის შეფასების კრიტერიუმების და CPU-ს ხმაურის კოეფიციენტის მიხედვით ტესტირების სერვისის განსაზღვრა

სიმულაციის პროცესის წარმადობის შეფასება, 2 კრიტერიუმით განხორციელდება ესენია: დაკავებული მეხსიერება (RAM). დავალების

```

=== Units are milliseconds (cpu/wall) and kilobytes (vmem/malloc).
=== Full output inside: ntuple.pmon.gz
===== step ini =====
      n  cpu  vmem  malloc  component
[ini]  1  54990  275200  257461  G4AtlasSvc
[ini]  1   3780  130500  135688  GeoModelSvc
[ini]  1   3650  118768  126900  DetectorGeometrySvc
[ini]  1    390   45056   45591  PoolSvc
[ini]  1   2810   89672   73051  [collapsed_181_comps]
[ini]  1  65620  659196  638690  [total_for_185_comps]
    
```

სურ. 126 CPU Time და RAM-ის შესახებ ინფორმაცია სიმულაციის პროცესის log ფაილში

შესრულების პროცესორული დრო (CPU Time) [13]. ამ პარამეტრების შესახებ

ინფორმაცია მიღებულ იქნება სიმულაციის log ფაილიდან სურ 126. სურათზე ჩანს, რომ მონიშნულ პირველ სვეტში მოცემულია პროცესორის მიერ დავალების შესასრულებლად საჭირო დრო, ანუ CPU Time, რომელიც გაზომილია მილიწამებში, ხოლო მონიშნულ მეორე სვეტში malloc-ში იწერება სიმულაციის პროცესის მიერ დაკავებული ოპერატიული მეხსიერების RAM-ის შესახებ ინფორმაცია და ის გაზომილია კილობაიტებში. ჩვენი კვლევისთვის კი ყველაზე საინტერესო ინფორმაცია განთავსებულია ბოლო სტრიქონში, სადაც პროცესის ყველა ეტაპის დაჯამებული ინფორმაცია წარმოდგენილი CPU Time და RAM-ის შესახებ.

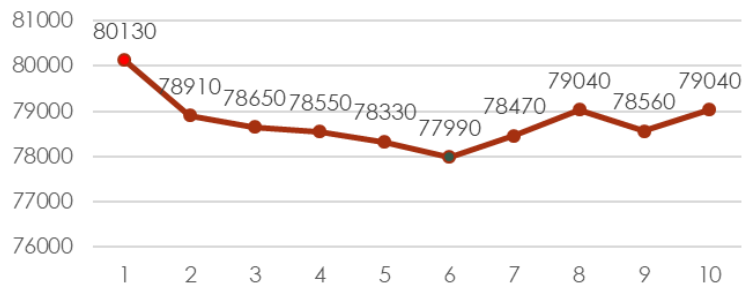
სიმულაციის საცდელი სესიების დროს, ერთი და იგივე პარამეტრების გამოყენებისას, ყოველ ჯერზე CPU Time-ის განსხვავებული შედეგები დაფიქსირდა, განსხვავებით RAM-ის შედეგებისგან. დაკავებული ოპერატიული მეხსიერება ასეთ შემთხვევაში არ იცვლება. CPU Time-ის განსხვავებული შედეგები განპირობებულია იმით, რომ სერვერები სადაც

serverlxplus703											სხვაობა max-min	ხმაური/%
CPU	80130	78910	78650	78550	78330	77990	78470	79040	78560	79040	2140	2.67
RAM	639526	639526	639526	639526	639526	639526	639526	639526	639526	639526		
serverlxplus723												ხმაური
CPU	66540	66620	66190	66400	66430	66770	66220	66890	66850	66750	550	0.82
RAM	639526	639526	639526	639526	639526	639526	639526	639526	639526	639526		
serverlxplus782												ხმაური
CPU	77500	76210	76340	79590	77760	78380	78510	76810	77070	76640	3380	4.24
RAM	639526	639526	639526	639526	639526	639526	639526	639526	639526	639526		

ცხრ. 71 სიმულაციის შედეგები CPU Time-ის ხმაურის დასადგენად

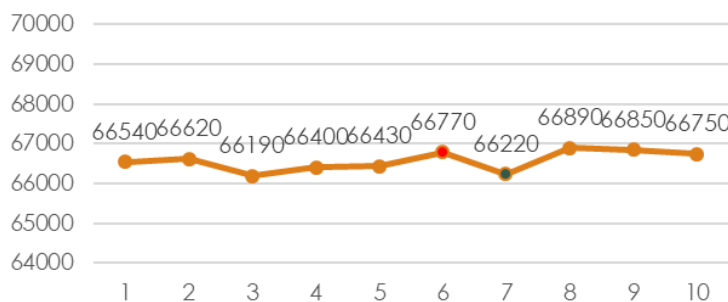
სიმულაციის სესიები მიმდინარეობს, პარალელურად სხვადასხვა ამოცანებით არიან დატვირთულები. აქედან გამომდინარე დღის წესრიგში დადგა ე.წ. ხმაურის კოეფიციენტის განსაზღვრა CPU Time-თვის. პირველ ეტაპზე განხორციელდა კვლევა ყველაზე სტაბილური სერვერის გამოსავლენად. ამისათვის შესწავლილ იქნა ცერნში მდებარე 3 lxplus სერვერი: lxplus703.cern.ch, lxplus723.cern.ch და lxplus782.cern.ch. lxplus703.cern.ch ცხრ. 71-ში წარმოდგენილია სამივე სერვერის სიმულაციის შედეგები. lxplus703.cern.ch სერვერის გამოსაკვლევად ჩატარდა სიმულაციის 10 საცდელი სესია. როგორც ჩანს CPU-თვის დაფიქსირებულ მაქსიმალურ და მინიმალურ შედეგებს შორის სხვაობა შეადგენს 2140 მილიწამს. ამრიგად

აღნიშნული სერვერის CPU-ს ხმაურის კოეფიციენტი შეადგენს დაახლოებით 2.67%-ს. შედეგები გრაფიკის სახით წარმოდგენილია სურ. 127-ზე. ასევე 10



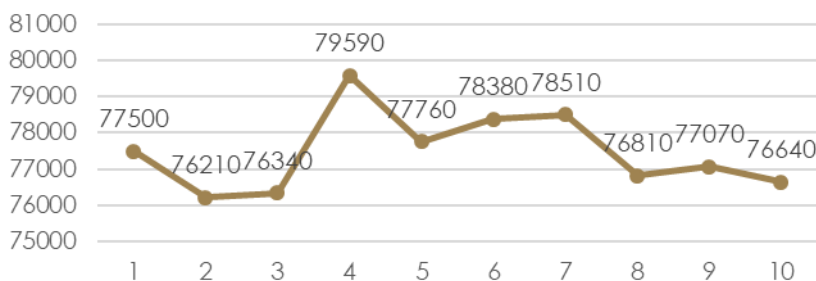
სურ. 127 სერვერ lxplus703.cern.ch - ის სიმულაციის შედეგები

სესია ჩატარდა სერვერ lxplus723.cern.ch-ზე. დაფიქსირებული შედეგების მიხედვით მაქსიმალურ და მინიმალურ მნიშვნელობებს შორის სხვაობა შეადგენს 550 მილიწამს, ანუ ამ სერვერის CPU-ს ხმაურის კოეფიციენტი არის 0.88%. შედეგები გრაფიკის სახით მოცემულია სურ. 128-ზე. რაც შეეხება



სურ. 128 სერვერ lxplus723.cern.ch - ის სიმულაციის შედეგები

lxplus782.cern.ch სერვერს, წინა შემთხვევების მსგავსად აქაც 10 სესია ჩატარდა და CPU-ის მაქსიმალურ და მინიმალურ შედეგებს შორის სხვაობა შეადგინა 3380 მილიწამი. შედეგად დადგინდა, რომ ამ სერვერის ხმაურის



სურ. 129 სერვერ lxplus782.cern.ch - ის სიმულაციის შედეგები

კოეფიციენტი შეადგენს 4.24%-ს. მიღებული შედეგები გრაფიკის სახით მოცემულია სურ. 129-ზე.

ამრიგად, ყველაზე მცირე ხმაურის CPU-ს ხმაურის კოეფიციენტი დაფიქსირდა lxplus723.cern.ch სერვერისთვის. ის წარმოადგენს ყველაზე სტაბილურ სერვერს ამ თვალსაზრისით და ამიტომ შემდგომი კვლევისთვის საჭირო სიმულაციის სესიები ჩამტერებულ იქნება აღნიშნულ სერვერზე.

3.3 მე-3 თავის დასკვნები

1. განსაზღვრულ იქნა სიმულაციის შესასვლელი ინფორმაცია. ესენია: დეტექტორის გეომეტრიული აღწერები (ATLAS-R3S-2021-01-00-00_VALIDATION), Run # 222500 და Condition Tag (OFLCOND-RUN12-SDR-25).
2. განხორციელდა სიმულაციის წარმატებული და წარუმატებელი პროცესის იდენტიფიცირება.
3. განისაზღვრა სერვერი სიმულაციის სატესტო სესიებისთვის და ეს სერვერია lxplus723.cern.ch.

თავი IV: სიმულაციის სატესტო სესიები

სიმულაციის სატესტო სესიების მიზანს წარმოადგენს გამოკვლევულ იქნას შემდეგი საკითხები:

1. რომელი დაპროგრამების მეთოდია უკეთესი სიმულაციის წარმადობისათვის.
2. როგორი გავლენა აქვთ ობიექტების რაოდენობას სიმულაციის პროცესის წარმადობაზე.
3. რა გავლენა აქვს CGS და პოლიგონ მეთოდების სხვადასხვა კომბინაციებს სიმულაციის პროცესის წარმადობაზე.

პირველ საკითხზე პასუხი ნაწილობრივ გაცემულია როდესაც განხორციელდა დაპროგრამების მეთოდების კლასიფიკაცია თეორიული ვარიანტების მიხედვით, თუმცა დარჩა ისეთი მეთოდები რომლებიც უნდა შემოწმდეს უშუალოდ სიმულაციაში ზემოთ ჩამოთვლილი საკითხების გამოსაკვლევად განხორციელდა სიმულაციის სატესტო სესიები.

4.1 ცილინდრული ობიექტების ტესტირება

პირველ ეტაპზე განხორციელდა ცილინდრული კლასის მაგალითების გამოკვლევა. დანართი #3-ში სულ ასეთი 7 მაგალითია: #08; #11; #21; #23; #26; #28; #32. ამ მაგალითებში ობიექტების დასაპროგრამებლად 3 მეთოდია გამოყენებული: Tube, Cylinder და Chain, უნდა გაირკვეს თუ რომელი მეთოდია უმჯობესი სიმულაციის წარმადობისათვის.

4.1.1 ტესტირება: მაგალითი № 23

მაგალითი #23-ზე ჩატარდა ცილინდრული ობიექტების დაპროგრამების მეთოდების წარმადობაზე გავლენის შესწავლა. კვლევის ფარგლებში გამოკვლევულ იქნა Tube, Cylinder და Chain მეთოდები. ცხრ. 72-

Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined

ცხრ. 72 #23 მაგალითის დაპროგრამების მეთოდები

ზე წარმოდგენილია #23 მაგალითის დაპროგრამების მეთოდები. პირველ ეტაპზე განხორციელდა შესაბამისი კოდების შექმნა:

```

I
<tubs name="Tube1" material="Iron" Rio_Z="0.; 500.; 10000." profile="0.;360." nbPhi="20"/>
<composition name="HFTruckRail" >
  <posXYZ volume="Tube1" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Tube1" X_Y_Z=" 0.; 1001.; 0.;" rot=" 0.; 0.;0." />
</composition>

II
<snake name="Chain" material="ShieldSteel" radius="500." nbPhi="20">
<snake_point X_Y_Z="0.; 0.; -5000." />
<snake_point X_Y_Z="0.; 0.; 5000." />
</snake>
<composition name="HFTruckRail" >
  <posXYZ volume="Chain" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Chain" X_Y_Z=" 0.; 1001.; 0.;" rot=" 0.; 0.;0." />
</composition>

III
<pcon name="Cone" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="0.; 500.; 5000."/>
<polyplane Rio_Z=" 0.; 500.; -5000."/>
</pcon>
<composition name="HFTruckRail" >
  <posXYZ volume="Cone" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Cone" X_Y_Z=" 0.; 1001.; 0.;" rot=" 0.; 0.;0." />
</composition>

```

შემდგომ ეტაპზე კი თითოეული მათგანი გაშვებულ იქნა სიმულაციაში. ტესტის მომენტში CPU Time-ის ხმაურის კოეფიციენტი შეადგენს: 0.4% ცხრ. 73 მიღებული შედეგები კი წარმოდგენილია ცხრ. 74-ში როგორც ცხრილიდან

	CPU Time/ms										ხმაური/%
lxplus723.cern.ch	67010	69990	67150	67270	67220	66970	67020	67110	66980	67070	0.4

ცხრ. 73 #23 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი

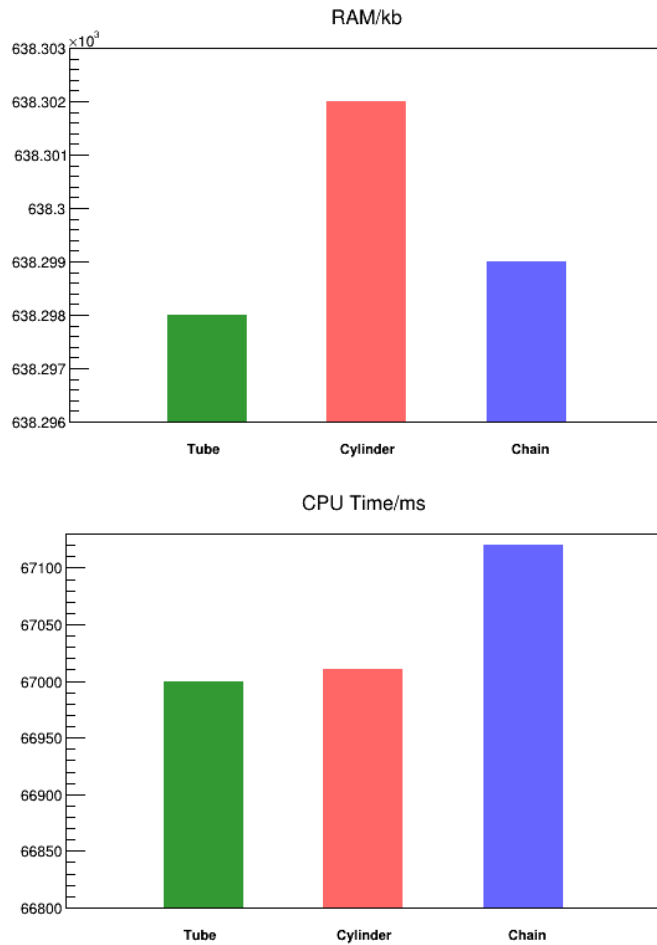
	Tube	Chain	Cylinder
CPU/ms	67000	67010	67120
RAM/kb	638298	638302	638299
სხვაობა CPU/ms		10	120
სხვაობა RAM/kb		4	1
სხვაობა CPU/%		0.0149	0.1788
სხვაობა RAM/%		0.0006	0.00016

ირკვევა წარმადობის თვალსაზრისით საუკეთესო შედეგს აფიქსირებს Tube მეთოდი, როგორც ოპერატიული მეხსიერების დატვირთვის ასევე CPU Time-ის მიხედვით. რაც

ცხრ. 73 #23 მაგალითის სიმულაციის წარმადობის შედეგები

შეეხება Cylinder და Chain მეთოდებს CPU Time-ში უპირატესობა Chain მეთოდის მხარესაა და ეს სხვაობა შეადგენს 0.18%, ხოლო ოპერატიული მეხსიერების დატვირთვის მიხედვით კი პირიქით უპირატესობა Cylinder

მეთოდის მხარესა და ეს სხვაობა შეადგენს 0.0006%-ს. სურ. 130-ზე იგივე შედეგები წარმოდგენილია გრაფიკული სახით.



სურ. 130 # 23 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად

4.1.2 ტესტირება: მაგალითი № 28

განვიხილოთ #28 მაგალითი. ამ შემთხვევაშიც მოცემულია ცილინდრული ობიექტი, რომლის დაპროგრამების მეთოდი მოცემულია ცხრ.75-ში:

```

Tube/ Cylinder
Tube/ Cylinder
Subtraction
combined
    
```

ცხრ. 75 #28 მაგალითის დაპროგრამების მეთოდები

თავდაპირველად განხორციელდა შესაბამისი კოდის ფორმირება: I კოდი შეესაბამება Tube მეთოდს და II კი Cylinder მეთოდს. თითოეული მათგანი

```

I
<tubs name="Tube1" material="Iron" Rio_Z="400.; 500.; 10000." profile="0.;360." nbPhi="20"/>
<tubs name="Tube2" material="Iron" Rio_Z="0.; 500.; 1000." profile="0.;360." nbPhi="20"/>
<tubs name="Tube3" material="Iron" Rio_Z="400.; 499.; 10000." profile="0.;360." nbPhi="20"/>
<subtraction name="05_subtro" >
<posXYZ volume="Tube1" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Tube2" X_Y_Z=" 500.; 0.; 0.; 0." rot=" 0.; 90.;0."/>
</subtraction>
<composition name="HFTruckRail" >
<posXYZ volume="05_subtro" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
<posXYZ volume="Tube3" X_Y_Z=" 5000.; 0.; 0.;" rot=" 0.; 90.; 0." />
</composition>

```

```

II
<pcon name="Cone" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="400.; 500.; 5000."/>
<polyplane Rio_Z=" 400.; 500.; -5000."/>
</pcon>
<pcon name="Cone1" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="0.; 500.; 500."/>
<polyplane Rio_Z=" 0.; 500.; -500."/>
</pcon>
<pcon name="Cone2" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="400.; 500.; 5000."/>
<polyplane Rio_Z=" 400.; 500.; -5000."/>
</pcon>
<subtraction name="05_subtro" >
<posXYZ volume="Cone" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Cone1" X_Y_Z=" 500.; 0.; 0.; 0." rot=" 0.; 90.;0."/>
</subtraction>
<composition name="HFTruckRail" >
<posXYZ volume="05_subtro" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
<posXYZ volume="Cone2" X_Y_Z=" 5000.; 0.; 0.;" rot=" 0.; 90.;0." />
</composition>

```

გაშვებულ იქნა სიმულაციაში, რა დროსაც ხმაურის კოეფიციენტი შეადგენდა

0.6%-ს ცხრ. 76. მიღებული შედეგები წარმოდგენილია ცხრ. 77-ში. როგორც

	CPU Time/ms										ხმაური/%
lxplus723.cern.ch	67570	67370	67480	67710	67510	67330	67490	67650	66630	67390	0.6

ცხრ. 76 #28 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი

	Tube	Cylinder
CPU/ms	67560	66140
RAM/kb	638292	638297
სხვაობა CPU/ms		80
სხვაობა RAM/kb		5
სხვაობა CPU/%		0.120956
სხვაობა RAM/%		0.000783

ცხრ. 77 #28 მაგალითის სიმულაციის წარმადობის

ცხრილიდან ირკვევა ოპერატიული მეხსიერების დატვირთვის მიხედვით 0.000783%-ით, ხოლო CPU-ს მიხედვით 0.12%-ით Tube მეთოდს აქვს უკეთესი შედეგი ვიდრე Cylinder მეთოდს. მეტი თვალსაჩინოებისთვის სურ. 131-ზე მოცემულია #28 მაგალითი კვლევის შედეგები გრაფიკულად.



სურ. 131 # 28 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად

4.1.3 ტესტირება: მაგალითი № 11

განვიხილოთ კიდევ ერთი მაგალითი #11. ეს მაგალითი არის ცილინდრული კლასის წარმომადგენელი. მისი დაპროგრამებისათვის შერჩეულია 2 მეთოდი ცხრ. 78: პირველ შემთხვევაში ობიექტის შექმნა ხდება

```

I      II
Cylinder  Tube
          Tube
          Tube
          Tube
  
```

ცხრ. 78 #11 მაგალითის დაპროგრამების მეთოდები

Cylinder მეთოდით, ხოლო მეორე მეთოდის მიხედვით 5 სხვადასხვა Tube-ით. როგორც წინა შემთხვევებში აქაც ამ ორი მეთოდისთვის შეიქმნა პროგრამული კოდი:

```

I
<pcon name="Cone" material="PolyBoron5percent" nbPhi="20">
<polyplane Rio_Z="400.; 500.; -1665."/>
<polyplane Rio_Z="400.; 500.; -1405."/>
<polyplane Rio_Z="400.; 760.; -1405."/>
<polyplane Rio_Z="400.; 760.; -1205."/>
<polyplane Rio_Z="400.; 500.; -1205."/>
<polyplane Rio_Z="400.; 500.; -970."/>
<polyplane Rio_Z="400.; 550.; -970."/>
<polyplane Rio_Z="400.; 550.; -780."/>
<polyplane Rio_Z="400.; 500.; -780."/>
<polyplane Rio_Z="400.; 500.; 1240."/>
<polyplane Rio_Z="400.; 500.; 1240."/>
<polyplane Rio_Z="400.; 550.; 1240."/>
<polyplane Rio_Z="400.; 550.; 1430."/>
<polyplane Rio_Z="400.; 500.; 1430."/>
<polyplane Rio_Z="400.; 500.; 1665."/>
</pcon>

II
<tubs name="Tube1" material="Iron" Rio_Z="400.; 500.; 3330." profile="0.;360." nbPhi="20"/>
<tubs name="Tube2" material="Iron" Rio_Z="500.; 550.; 190." profile="0.;360." nbPhi="20"/>
<tubs name="Tube3" material="Iron" Rio_Z="500.; 760.; 200." profile="0.;360." nbPhi="20"/>
<composition name="Tube" >
  <posXYZ volume="Tube1" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Tube2" X_Y_Z=" 0.; 0.; 1340.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Tube2" X_Y_Z=" 0.; 0.; -880.;" rot=" 0.; 0.;0." />
  <posXYZ volume="Tube3" X_Y_Z=" 0.; 0.; -1305.;" rot=" 0.; 0.;0." />
</composition>

```

შემდგომ ეტაპზე კი, თითოეული მათგანი გაშვებულ იქნა სიმულაციაში. ტესტირების დროს CPU Time-ის ხმაურის კოეფიციენტი შეადგენდა 0.8%-ს ცხრ. 79. სიმულაციის შედეგები კი წარმოდგენილია ცხრ. 80-ში. როგორც

	CPU Time/ms										ხმაური/%
lxplus723.cern.ch	63920	63880	63930	64190	62830	63680	63970	64010	62800	64010	0.8

ცხრ. 79 #11 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი

	Cylinder	Tube
CPU/ms	64910	63670
RAM/kb	638289	638286
სხვაობა CPU/ms	1240	
სხვაობა RAM/kb	3	
სხვაობა CPU/%	1.91033739	
სხვაობა RAM/%	0.000470007	

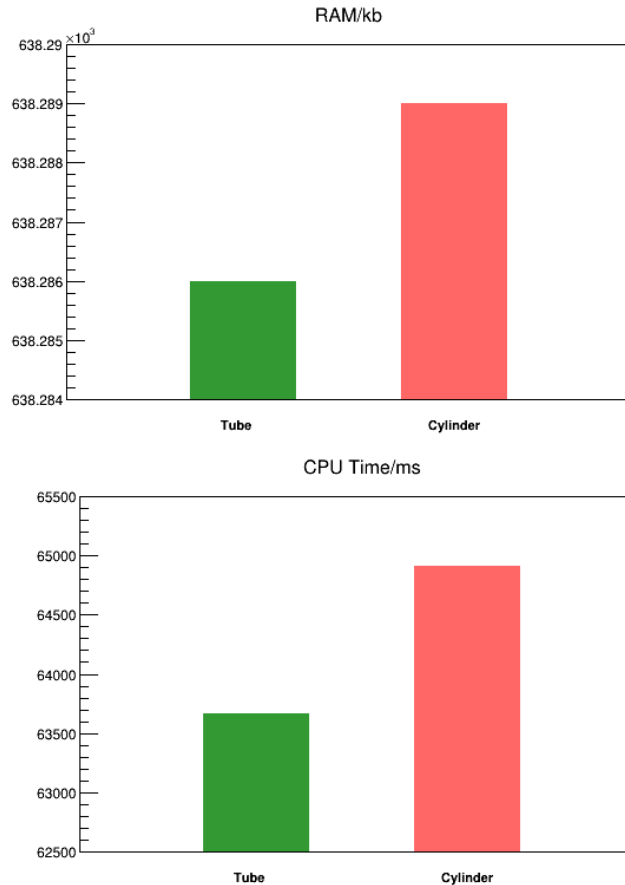
ცხრ. 80 #11 მაგალითის სიმულაციის შედეგები

მიღებულმა შედეგებმა აჩვენა CPU-ს მიხედვით წინა შემთხვევების მსგავსად უპირატესობა არის Tube მეთოდის მხარეს და ეს სხვაობა შეადგენს 1.91%-ს. ხოლო რაც შეეხება დაკავებულ ოპერატიულ მეხსიერებას, აქაც უპირატესობა Tube მეთოდის მხარესაა სხვაობა კი არის 0.00047%. სურ. 132-ზე #11 მაგალითის კვლევის შედეგებია წარმოდგენილი გრაფიკული სახით. რაც

მაგალითი #08	მაგალითი #21	მაგალითი #26	მაგალითი #32
Tube/ Cylinder	Tube	Tube/ Cylinder	Cylinder
Tube/ Cylinder	Tube	Cylinder	Tube
Subtraction	merged	Tube/ Cylinder	Cylinder
Subtraction		combined	merged

ცხრ. 81 #08, #21, #26, #32 მაგალითების დაპროგრამების მეთოდები

შეეხება მაგალითებს #08; #21; #26; #32. ცხრ. 81. ცხრილიდან ჩანს, რომ დაპროგრამების მეთოდების ანალიზის წინა ეტაპებიდან გამომდინარე #21 და #32 მაგალითში დარჩენილია მხოლოდ 1 დაპროგრამების მეთოდი, ხოლო



სურ. 132 # 11 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად რაც შეეხება # 08 და # 26 მაგალითებს აქ ობიექტის შესაქმნელად გამოყენებულია Tube და Cylinder მეთოდები. ამ მეთოდების კვლევა კი სხვა მაგალითებზე განხორციელდა. შესაბამისად ზემოთ ჩამოთვლილი მაგალითების სიმულაციის სატესტო სესიებში არ იქნა გამოყენებული.

4.2 წახნაგოვანი ობიექტების ტესტირება

წახნაგოვანი ზედაპირების მქონე ობიექტების დაპროგრამების მეთოდების კვლევაში მონაწილეობა მიიღებენ დანართი # 3-ში არსებული შემდეგი მაგალითები: #01, #02, #03, #04, #05, #15, #20, #24, #27, #29, #30, #31, #35, #37, #39, #40, #42. პირველ ეტაპზე განხორციელდა აღნიშნული

მაგალითების ანალიზი და გამოიკვეთა, რომ შემდეგ მაგალითებს: #01, #02; #04; #05; #30; #24; #31; #39; #42 აქვთ დაპროგრამების მსგავსი სტრუქტურები, კერძოდ ყველა მათგანის დაპროგრამება ხორციელდება Cube და Pyramid მეთოდების კომბინაციით Boolean მეთოდებთან. ამიტომ განხორციელდა ამ მეთოდების განზოგადება შემდეგი სახით ცხრ. 82. ცხრილში მოცემულია

I	II	III	IV	V	VI	VII	VIII
Cube	Cube	Cube	Cube	Pyramid	Pyramid	Pyramid	Pyramid
Cube	Cube	Cube	Cube	Pyramid	Pyramid	Pyramid	Pyramid
Subtraction	Cube	Cube	Cube	Subtraction	Pyramid	Pyramid	Pyramid
Subtraction	Subtraction	Cube	Cube	Subtraction	Subtraction	Pyramid	Pyramid
	Subtraction	Subtraction	Cube		Subtraction	Subtraction	Pyramid
	Subtraction	Subtraction	Subtraction		Subtraction	Subtraction	Subtraction
		Subtraction	Subtraction			Subtraction	Subtraction
		Subtraction	Subtraction			Subtraction	Subtraction
			Subtraction			Subtraction	Subtraction
			Subtraction				Subtraction

ცხრ. 82. #01, #02; #04; #05; #30; #24; #31; #39; #42 მაგალითების დაპროგრამების მეთოდები

წახნაგოვანი ობიექტების დაპროგრამების მეთოდები, რომლებიც სრულად ფარავენ ზემოთ მოცემულ 8 შემთხვევას. როგორც 82-ე ცხრილიდან ჩანს IV და VIII შემთხვევები ტავის თავში მოიცავენ დანარჩენ შემთხვევებს, ამიტომ საკმარისია სიმულაციაში გაშვებულ იქნას მხოლოდ ეს შემთხვევები. განხორციელდა მათი შესაბამისი კოდის ფორმირება:

```

IV
<box name="Box" material="Aluminium" X_Y_Z="185.; 120.; 40." />
<box name="Box1" material="Aluminium" X_Y_Z="30.; 22.; 42." />
<box name="Box2" material="Aluminium" X_Y_Z="30.; 22.; 42." />
<box name="Box3" material="Aluminium" X_Y_Z="30.; 22.; 42." />
<box name="Box4" material="Aluminium" X_Y_Z="30.; 22.; 42." />
<subtraction name="cube" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Box1" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Box2" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Box3" X_Y_Z=" 65.; 40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Box4" X_Y_Z=" 65.; -40.; 0." rot=" 0.; 0.; 0." />
</subtraction>

VIII
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="185.; 185.; 120.; 120.; 40." />
<trd name="Pyramid_sub2" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub3" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub4" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub5" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<subtraction name="cube" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Pyramid_sub2" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Pyramid_sub3" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Pyramid_sub4" X_Y_Z=" 65.; -40.; 0." rot=" 0.; 0.; 0." />
<posXYZ volume="Pyramid_sub5" X_Y_Z=" 65.; 40.; 0." rot=" 0.; 0.; 0." />
</subtraction>

```

შემდგომ ეტაპზე განხორციელდა მათი გაშვება სიმულაციაში, რა დროსაც ხმაურის კოეფიციენტი იყო 0.7% ცხრ. 83. სიმულაციის შედეგები კი

	CPU Time/ms								ხმაური/%		
lxplus723.cern.ch	74420	74490	74010	74500	74450	73990	74470	74480	74010	74440	0.7

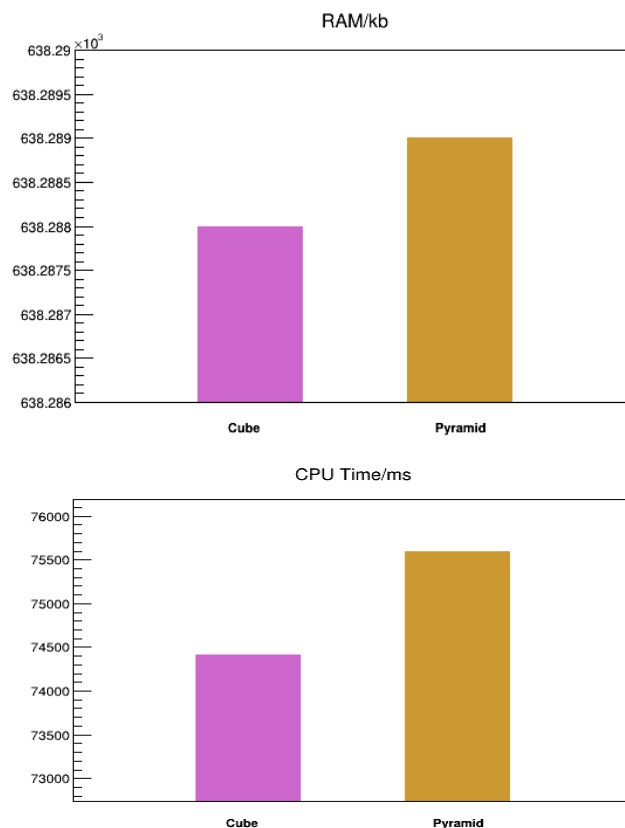
ცხრ. 83 CPU Time-ის ხმაურის კოეფიციენტი cube და pyramid მეთოდებისთვის შემდეგნაირია ცხრ. 84. CPU/ms და RAM/kb სტრუქტურებში მოცემულია

	I	IV
CPU/ms	74410	75600
RAM/kb	638588	638589
სხვაობა CPU/ms		1190
სხვაობა RAM/kb		1
სხვაობა CPU/%		1.57407
სხვაობა RAM/%		0.000157

ცხრ. 84. სიმულაციიდან მიღებული შედეგები cube და pyramid მეთოდებისათვის

სიმულაციის პროცესის განმავლობაში გამოყენებული პროცესორის და ოპერატიული მეხსიერების შესახებ ინფორმაცია. ცხრილიდან ირკვევა, რომ CPU/ms შედეგებს შორის სხვაობები მაქსიმუმ 1.574%. რაც შეეხება ოპერატიულ მეხსიერებას, უპირატესობა აქვს Cube მეთოდს Pyramid მეთოდთან მიმართებაში და ეს სხვაობა შეადგენს 000157%, შესაბამისად შეგვიძლია დავასკვნათ, რომ წარმადობის თვალსაზრისით, გეომეტრიული აღწერების შექმნის დროს უმჯობესია გამოყენებულ იქნეს Cube მეთოდი.

სურ. 133-ზე cube და pyramid მეთოდების კვლევის შედეგები წარმოდგენილი გრაფიკული სახით.



სურ. 133 cube და pyramid მეთოდების კვლევის შედეგები

შემდგომ ეტაპზე გამოკვლეულ იქნა Boolean ოპერატორების რაოდენობის გავლენა სიმულაციის წარმადობაზე. ამის გასარკვევად პირველ რიგში დამუშავდა ცხრ.82-ში მოცემული I, II, III, V, VI და VII შემთხვევების დაპროგრამების კოდები.

VI

```
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="185.; 185.; 120.; 120.; 40." />
<trd name="Pyramid_sub2" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub3" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<subtraction name="cube" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid_sub2" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Pyramid_sub3" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

VII

```
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="185.; 185.; 120.; 120.; 40." />
<trd name="Pyramid_sub2" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub3" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<trd name="Pyramid_sub4" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<subtraction name="cube" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid_sub2" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Pyramid_sub3" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Pyramid_sub4" X_Y_Z=" 65.; -40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

I

```
<box name="Box" material="Aluminium" X_Y_Z="185.; 120.; 40."/>
<box name="Box1" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<subtraction name="cube" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box1" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

II

```
<box name="Box" material="Aluminium" X_Y_Z="185.; 120.; 40."/>
<box name="Box1" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<box name="Box2" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<subtraction name="cube" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box1" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box2" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

III

```
<box name="Box" material="Aluminium" X_Y_Z="185.; 120.; 40."/>
<box name="Box1" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<box name="Box2" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<box name="Box3" material="Aluminium" X_Y_Z="30.; 22.; 42."/>
<subtraction name="cube" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box1" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box2" X_Y_Z=" -65.; -40.; 0." rot=" 0.; 0.; 0."/>
<posXYZ volume="Box3" X_Y_Z=" 65.; 40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

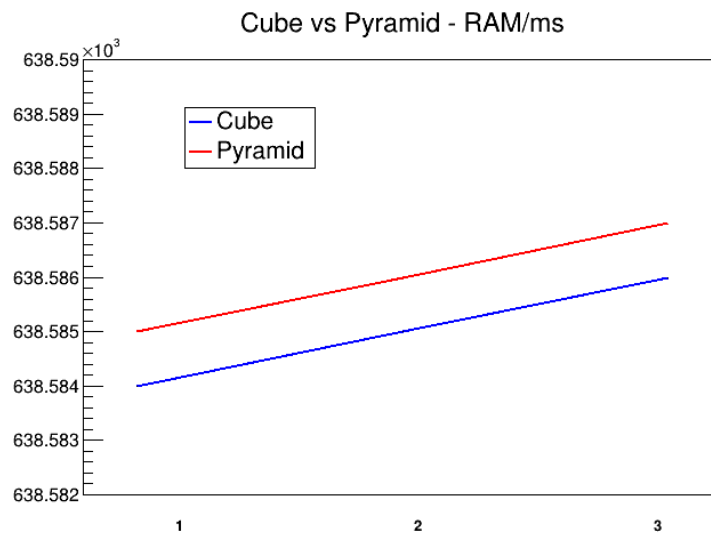
V

```
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="185.; 185.; 120.; 120.; 40." />
<trd name="Pyramid_sub2" material="ShieldSteel" Xmp_Ymp_Z="30.; 30.; 22.; 22.; 42." />
<subtraction name="cube" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid_sub2" X_Y_Z=" -65.; 40.; 0." rot=" 0.; 0.; 0."/>
</subtraction>
```

შემდგომ ეტაპზე განხორციელდა მათი სიმულაციაში გაშვება და შედეგები წარმოდგენილია ცხრ.85-ში. ობიექტების და Boolean ოპერატორების

	cube - I	cube- II	cube- III
CPU/ms	73450	75410	73560
RAM/kb	638584	638585	638586
	pyramid- V	pyramid- VI	pyramid- VII
CPU/ms	74750	77110	75600
RAM/kb	638585	638586	638587
სხვაობა CPU/ms	1300	1700	2040
სხვაობა RAM/kb	1	1	1
სხვაობა CPU/%	1.739	2.2	2.698
სხვაობა RAM/%	0.000157	0.000157	0.000157

ცხრ. 85. სიმულაციიდან მიღებული შედეგები I-დან VII-მდე მაგალითებისათვის



სურ. 134 cube და pyramid მეთოდების კვლევის შედეგები. ოპერატიული მეხსიერების დატვირთვა

რაოდენობის გაზრდასთან ერთად ოპერატიული მეხსიერების დატვირთვა იზრდება პირდაპირ პროპორციულად სურ. 134. აქედან გამომდინარე შეგვიძლია დავასკვნათ, რომ გეომეტრიული აღწერებში ერთნაირი მეთოდების და Boolean ოპერატორების რაოდენობა, სიმულაციის წარმადობაზე გავლენას არ ახდენს.

4.2.1 ტესტირება: მაბალოთი № 03

შემდგომ ეტაპზე განხორციელდა # 03 მაგალითის გამოკვლევა სურ. 135. ის წარმოადგენს წახნაგოვანი ზედაპირების კლასს. მისთვის შერჩეული



სურ. 135 # 03 მაგალითი

I	II	III
Symmetric	Arbitrary	Cube
		Arbitrary
		Subtraction

ცხრ.86 #03 მაგალითის დაპროგრამების მეთოდები

დაპროგრამების მეთოდები შემდეგნაირია ცხრ. 86: მეორე ეტაპზე მათი

```

I
<gvxysx name="SymmetricPolygon" material="Carbon" dZ="40.">
<gvxy_point X_Y="1100.;550."/>
<gvxy_point X_Y="638.1;-250."/>
<gvxy_point X_Y="325.-;250."/>
<gvxy_point X_Y="325.-;-550."/>
</gvxysx>

II
<gvxy name="arbitrary" material="ShieldSteel" dZ="40.">
<gvxy_point X_Y="1100.;550."/>
<gvxy_point X_Y="638.1;-250."/>
<gvxy_point X_Y="325.-;250."/>
<gvxy_point X_Y="325.-;-550."/>
<gvxy_point X_Y="-325.-;-550."/>
<gvxy_point X_Y="-325.-;-250."/>
<gvxy_point X_Y="-638.1;-250."/>
<gvxy_point X_Y="-1100.;550."/>
</gvxy>

III
<box name="Box" material="Aluminium" X_Y_Z="2200.; 1100.; 40."/>
<gvxy name="arbitrary" material="ShieldSteel" dZ="50.">
<gvxy_point X_Y="1100.;550."/>
<gvxy_point X_Y="638.1;-250."/>
<gvxy_point X_Y="325.-;250."/>
<gvxy_point X_Y="325.-;-550."/>
<gvxy_point X_Y="-325.-;-550."/>
<gvxy_point X_Y="-325.-;-250."/>
<gvxy_point X_Y="-638.1;-250."/>
<gvxy_point X_Y="-1100.;550."/>
<gvxy_point X_Y="-1150.;650."/>
<gvxy_point X_Y="-1150.-;-650."/>
<gvxy_point X_Y="1150;-650."/>
<gvxy_point X_Y="1150;650."/>
</gvxy>
<subtraction name="03_subtro" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="arbitrary" X_Y_Z=" 0.; 0.; 0.; 0." rot=" 0.; 0.;0."/>
</subtraction>

```

შესაბამისი კოდების ფომირება განხორციელდა:

თითოეული მათგანი გაშვებულ იქნა სიმულაციაში. ხმაურის კოეფიციენტი კი ამ ტესტის დროს შეადგენდა 0.6%-ს. ცხრ. 87 და მიღებული იქნა სიმულაციის შედეგები, რომელიც მოცემულია ცხრ. 88-ში. როგორც ცხრილიდან ჩანს CPU-ს მიხედვით Symmetric და Arbitrary მეთოდებს ერთნაირი შედეგები ავთ და მათ დაახლოებით 2.15-ით ჩამორჩება Cube და

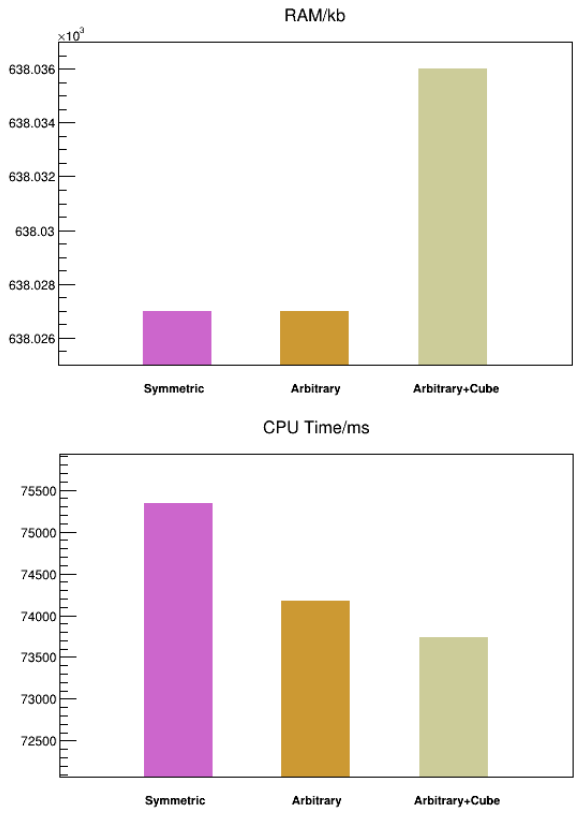
	CPU Time/ms										ხმაური/%
lxfplus723.cern.ch	75350	75300	75150	75480	75250	74990	75280	75370	75450	75410	0.6

ცხრ. 87 #03 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი

	Symmetric	Arbitrary	Cube + Arbitrary
CPU/ms	75340	74170	73740
RAM/kb	638027	638027	638036
სხვაობა CPU/ms		1170	1600
სხვაობა RAM/kb		0	9
სხვაობა CPU/%		1.577	2.1698
სხვაობა RAM/%		0	0.0014

ცხრ.88 #03 მაგალითის სიმულაციის შედეგები

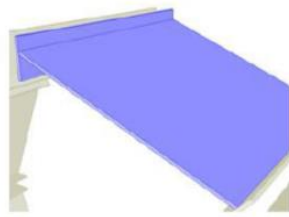
Arbitrary-ის კომბინაცია. რაც შეეხება ოპერატიული მეხსიერების დატვირთვას Symmetric და Arbitrary აჩვენებენ ერთნაირ შედეგებს, მე-3 მეთოდი კი როგორც მოსალოდნელი იყო დაახლოებით 0.0014%-ით უფრო ცუდ შედეგს აფიქსირებს ვიდრე პირველი 2. აქედან გამომდინარე შეგვიძლია დავასკვნათ, რომ გეომეტრიული აღწერების დაპროგრამების დროს Symmetric და Arbitrary მეთოდების გამოყენებისას შეგვიძლია თანაბარი პრიორიტეტი მივანიჭოთ. სურ. 136-ზე წარმოდგენილია კვლევის შესაბამისი



სურ. 136 # 03 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად გრაფიკული გამოსახულება.

4.2.2 ტესტირება: მაგალითი № 27

შემდეგი მაგალითი, რომლის გამოკვლევაც მოხდა არის #27 სურ. 137.



სურ. 137 # 27 მაგალითი

მას შეესაბამება დაპროგრამების 2 მეთოდი ცხრ. 89. განხორციელდა ამ

I	II
Cube	Arbitrary
Cube	Cube
Subtraction	combined
Cube	
Combined	

ცხრ.89 #27 მაგალითის დაპროგრამების მეთოდები

მეთოდების შესაბამისი კოდების გენერაცია:

```

I
<box name="Box" material="Aluminium" X_Y_Z="500.; 200.; 40."/>
<box name="Box1" material="Aluminium" X_Y_Z="50.; 300.; 100."/>
<subtraction name="05_subtrc" >
<posXYZ volume="Box" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 20.;0."/>
<posXYZ volume="Box1" X_Y_Z=" 220.; 0.; -80.;" rot=" 0.; 0.;0."/>
</subtraction>
<composition name="HFTruckRail" >
<posXYZ volume="05_subtrc" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.; 0." />
<posXYZ volume="Box1" X_Y_Z=" 220.; 0.; -80.;" rot=" 0.; 0.; 0." />
</composition>

II
<gvxy name="29_cut" material="ShieldSteel" dZ="200.">
<gvxy_point X_Y="-241.764; -66.711"/>
<gvxy_point X_Y="-228.083; -104.299"/>
<gvxy_point X_Y="195.; 49.691"/>
<gvxy_point X_Y="195.; 92.258"/>
</gvxy>
<box name="Box1" material="Aluminium" X_Y_Z="50.; 100.; 300."/>
<composition name="HFTruckRail" >
<posXYZ volume="29_cut" X_Y_Z=" 0.; 0.; 0.;" rot=" 0.; 0.; 0." />
<posXYZ volume="Box1" X_Y_Z=" 220.; 80.; 0.;" rot=" 0.; 0.; 0." />

```

I ვარიანტში მოცემული ობიექტის დაპროგრამება ხდება 2 Cube და Boolean-ის საშუალებით, ხოლო II ვარიანტში arbitrary და Cube მეთოდებით. დათვლილ იქნა CPU Time-ის ხმაურის კოეფიციენტი და ის შეადგენს 0.3%-ს ცხრ. 90. შემდგომ ეტაპზე ორივე ვარიანტი გაშვებულ იქნა სიმულაციაში.

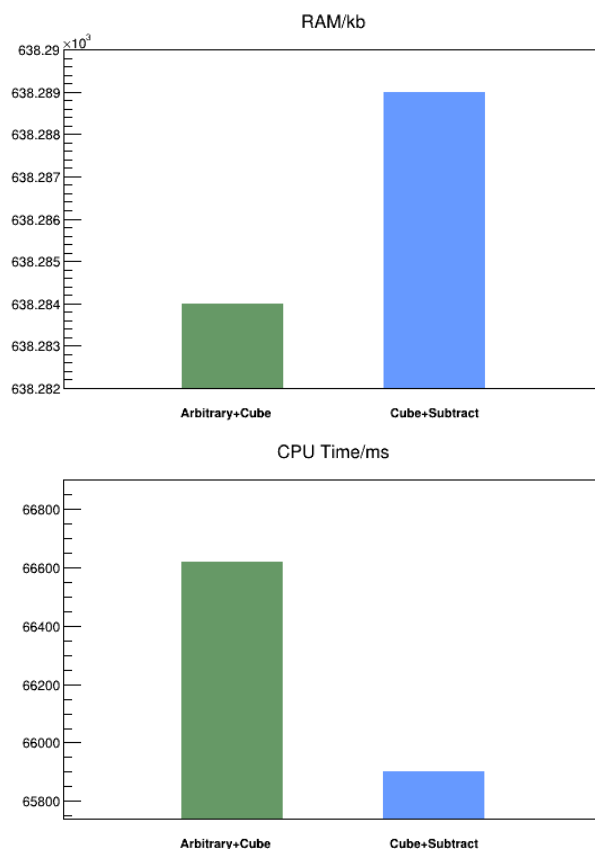
	CPU Time/ms										ხმაური/%
lxplus723.cern.ch	66610	66650	66460	66660	66510	66450	66590	66480	66490	66450	0.3

ცხრ. 90 #27 მაგალითის CPU Time-ის ხმაურის კოეფიციენტი

	Cube+subtract.	Arbitrary+cube
CPU/ms	66620	65900
RAM/kb	638284	638289
სხვაობა CPU/ms		80
სხვაობა RAM/kb		5
სხვაობა CPU/%		0.121396
სხვაობა RAM/%		0.000783

ცხრ.91 #27 მაგალითის სიმულაციის შედეგები

შედეგები მოცემულია ცხრ. 91-ში, როგორც ცხრილიდან იკვეთება CPU Time-ის თვალსაზრისით 0.12%-ით უკეთესი შედეგი აქვს CGS მეთოდებს. რაც შეეხება ოპერატიული მეხსიერების დატვირთვას უპირატესობა აქვს I



სურ. 138 # 27 მაგალითის სიმულაციის შედეგები წარმოდგენილი გრაფიკულად მეთოდს აქვს და ეს სხვაო შეადგენს 0.0007%-ს. სურ.138-ზე #27 მაგალითის კვლევის შედეგებია წარმოდგენილი გრაფიკული სახით.

რაც შეეხება #15, #20, #29, #35, #37, #40 მაგალითებს მათი კვლევაში გამოყენება არ განხორციელებულა, ვინაიდან მათ დასაპროგრამებლად გამოყენებული მეთოდების კვლევა სხვ მაგალითებზე განხორციელდა.

4.2.3 შერეული ობიექტების ტესტირება

გარდა ცილინდრული და წახნაგოვანი ობიექტებისა სიმულაციაში ასევე გამოყენებულია შერეული პრიმიტივები, რომლების შეიცავენ როგორც ცილინდრულ ასევე წახნაგოვან ზედაპირებს. კვლევისათვის სულ ასეთი 18 მაგალითია შერჩეული (დანართი #3), რომლებსაც 252 დაპროგრამების მეთოდი შეესაბამება. ეს მაგალითებია: #06, #07, #09, #10, #12, #13, #14, #16, #17, #18, #19, #20, #25, #33, #34, #36, #38, #41. კვლევის მიზანია გაირკვეს რა დამოკიდებულება აქვთ ცილინდრულ და წახნაგოვან დაპროგრამების მეთოდებს შერეული ზედაპირების მქონე ობიექტების დაპროგრამებისას.

განვიხილოთ #19 მაგალითი, რადგანაც მისი დაპროგრამების მეთოდები მოიცავს ისეთ შემთხვევებს, რომლის ანალიზის შედეგად შესაძლებელი

19-1	19-2
Arbitrary	Cube/pyramid
Tube /chain / cylinder	Cube /pyramid
Subtraction	Subtraction
Subtraction	Subtraction
	Tube /chain / cylinder
	Subtraction
	Subtraction

ცხრ.92 #19-ის დაპროგრამების მეთოდები

ხდება ძირითად კითხვებზე პასუხის გაცემა. Atlas-ის დეტექტორში #19 პრიმიტივი სულ 64 სხვადასხვა პოზიციაზეა განთავსებული, შესაბამისად კვლევაში გამოყენებულ იქნა 64 პრიმიტივის შედეგი. მისი დაპროგრამების მეთოდები შემდეგნაირად გამოიყურება ცხრ.92.

19-1-ში ობიექტის აგება ხდება arbitrary, tube, chain ან cylinder და Subtraction მეთოდებით, რომელთაგანაც arbitrary არის წახნაგოვანი კლასის, ხოლო tube chain და cylinder კი ცილინდრული კლასის წარმომადგენელი. რაც შეეხება

19-2 შემთხვევას, აქ დაპროგრამების ჩვენთვის საინტერესო 9 სხვადასხვა კომბინაციაა გაერთიანებული და წახნაგოვანი კლასიდან გამოყენებულია Cube ან pyramid, ხოლო ცილინდრული კლასიდან კი Tube, chain ან cylinder აქ ასევე გამოყენებულია Subtraction მეთოდიც. თავდაპირველად განხორციელდა შესაბამისი კოდების შექმნა. 19-1 ის შემთხვევაში შეიქმნა 3 ვარიანტი:

19-1-1

```
<gvxy name="testBox1" material="Aluminium" dZ="60000.">
  <gvxy_point X_Y="-5000.; -5000."/>
  <gvxy_point X_Y="-5000.; -4000."/>
  <gvxy_point X_Y="-500.; -4000."/>
  <gvxy_point X_Y="-500.; 4000."/>
  <gvxy_point X_Y="-5000.; 4000."/>
  <gvxy_point X_Y="-5000.; 5000."/>
  <gvxy_point X_Y="5000.; 5000."/>
  <gvxy_point X_Y="5000.; 4000."/>
  <gvxy_point X_Y="500.; 4000."/>
  <gvxy_point X_Y="500.; -4000."/>
  <gvxy_point X_Y="5000.; -4000."/>
  <gvxy_point X_Y="5000.; -5000."/>
</gvxy>
```

19-1-3

```
<gvxy name="testBox1" material="Aluminium" dZ="60000.">
  <gvxy_point X_Y="-5000.; -5000."/>
  <gvxy_point X_Y="-5000.; -4000."/>
  <gvxy_point X_Y="-500.; -4000."/>
  <gvxy_point X_Y="-500.; 4000."/>
  <gvxy_point X_Y="-5000.; 4000."/>
  <gvxy_point X_Y="-5000.; 5000."/>
  <gvxy_point X_Y="5000.; 5000."/>
  <gvxy_point X_Y="5000.; 4000."/>
  <gvxy_point X_Y="500.; 4000."/>
  <gvxy_point X_Y="500.; -4000."/>
  <gvxy_point X_Y="5000.; -4000."/>
  <gvxy_point X_Y="5000.; -5000."/>
</gvxy>
<pcon name="Cone" material="ShieldSteel" nbPhi="20">
  <polyplane Rio_Z="0.; 1900.; 2000."/>
  <polyplane Rio_Z=" 0.; 1900.; -2000."/>
</pcon>
<subtraction name="05_subtrc1" >
  <posXYZ volume="testBox1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
  <posXYZ volume="Cone" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
  <posXYZ volume="Cone" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>
```

19-1-2

```
<gvxy name="testBox1" material="Aluminium" dZ="60000.">
  <gvxy_point X_Y="-5000.; -5000."/>
  <gvxy_point X_Y="-5000.; -4000."/>
  <gvxy_point X_Y="-500.; -4000."/>
  <gvxy_point X_Y="-500.; 4000."/>
  <gvxy_point X_Y="-5000.; 4000."/>
  <gvxy_point X_Y="-5000.; 5000."/>
  <gvxy_point X_Y="5000.; 5000."/>
  <gvxy_point X_Y="5000.; 4000."/>
  <gvxy_point X_Y="500.; 4000."/>
  <gvxy_point X_Y="500.; -4000."/>
  <gvxy_point X_Y="5000.; -4000."/>
  <gvxy_point X_Y="5000.; -5000."/>
</gvxy>
<tubs name="cut_Tube1" material="ShieldSteel" Rio_Z="0.; 1900.; 4000." profile="0.;360." nbPhi="36"/>
<subtraction name="05_subtrc1" >
  <posXYZ volume="testBox1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
  <posXYZ volume="cut_Tube1" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
  <posXYZ volume="cut_Tube1" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>
```

რაც შეეხება 19-2 შემთხვევას აქ განხორციელდა დაპროგრამების კოდების შექმნა pyramid-ებისთვის და Cube-ებისთვის. 19-2-ში მოცემულია pyramid-ების კოდები, ხოლო 19-3-ის ქვეშ კი Cube-ის:

19-2-1

```
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="10000.; 10000.; 10000.; 10000.; 60000." />
<trd name="Pyramid2" material="ShieldSteel" Xmp_Ymp_Z="8000.; 8000.; 8000.; 8000.; 70000." />
<tubs name="cut_Tube" material="ShieldSteel" Rio_Z="0.; 1900.; 4000." profile="0.;360." nbPhi="36"/>
<subtraction name="05_subtrc" >
  <posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
  <posXYZ volume="Pyramid2" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
  <posXYZ volume="Pyramid2" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
  <posXYZ volume="cut_Tube" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
  <posXYZ volume="cut_Tube" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>
```

```

19-2-2
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="10000.; 10000.; 10000.; 10000.; 60000." />
<trd name="Pyramid2" material="ShieldSteel" Xmp_Ymp_Z="8000.; 8000.; 8000.; 8000.; 70000." />
<snake name="Chain" material="ShieldSteel" radius="1900." nbPhi="20">
<snake_point X_Y_Z="-20000.; 0.; 0." />
<snake_point X_Y_Z="20000.; 0.; 0." />
</snake>
<subtraction name="05_subtrc" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid2" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid2" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Chain" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 0.;0."/>
<posXYZ volume="Chain" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 0.;0."/>
</subtraction>
<trd name="Pyramid1" material="ShieldSteel" Xmp_Ymp_Z="10000.; 10000.; 10000.; 10000.; 60000." />
<trd name="Pyramid2" material="ShieldSteel" Xmp_Ymp_Z="8000.; 8000.; 8000.; 8000.; 70000." />
<pcon name="Cone" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="0.; 1900.; 2000."/>
<polyplane Rio_Z=" 0.; 1900.; -2000."/>
</pcon>
<subtraction name="05_subtrc" >
<posXYZ volume="Pyramid1" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid2" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Pyramid2" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Cone" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
19-3-1
<box name="testBox" material="ShieldSteel" X_Y_Z="10000.; 10000.; 60000."/>
<box name="testBox_sub" material="ShieldSteel" X_Y_Z="8000.; 8000.; 70000."/>
<tube name="cut_Tube" material="ShieldSteel" Rio_Z="0.; 1900.; 4000." profile="0.;360." nbPhi="36"/>
<subtraction name="05_subtrc" >
<posXYZ volume="testBox" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="cut_Tube" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
<posXYZ volume="cut_Tube" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>
19-3-2
<box name="testBox" material="ShieldSteel" X_Y_Z="10000.; 10000.; 60000."/>
<box name="testBox_sub" material="ShieldSteel" X_Y_Z="8000.; 8000.; 70000."/>
<snake name="Chain" material="ShieldSteel" radius="1900." nbPhi="20">
<snake_point X_Y_Z="0.; 0.; -2000." />
<snake_point X_Y_Z="0.; 0.; 2000." />
</snake>
<subtraction name="05_subtrc" >
<posXYZ volume="testBox" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Chain" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
<posXYZ volume="Chain" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>
19-3-3
<box name="testBox" material="ShieldSteel" X_Y_Z="10000.; 10000.; 60000."/>
<box name="testBox_sub" material="ShieldSteel" X_Y_Z="8000.; 8000.; 70000."/>
<pcon name="Cone" material="ShieldSteel" nbPhi="20">
<polyplane Rio_Z="0.; 1900.; 2000."/>
<polyplane Rio_Z=" 0.; 1900.; -2000."/>
</pcon>
<subtraction name="05_subtrc" >
<posXYZ volume="testBox" X_Y_Z=" 0.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" 4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="testBox_sub" X_Y_Z=" -4500.; 0.; 0." rot=" 0.; 0.;0."/>
<posXYZ volume="Cone" X_Y_Z=" 0.; 0.; 13500." rot=" 0.; 90.;0."/>
<posXYZ volume="Cone" X_Y_Z=" 0.; 0.; -13500." rot=" 0.; 90.;0."/>
</subtraction>

```

შემდეგ ეტაპზე კი განხორციელდა მათი სიმულაციაში ცხრ.93-ში მოცემულია სიმულაციიდან მიღებული შედეგები. როგორც წახნაგოვანი ზედაპირების მქონე მაგალითების კვლევის შედეგად ცნობილია cube მეთოდს pyramid მეთოდთან შედარებით აქვს 0.0001570% -ით უკეთესი შედეგი. ცხრ. 93-დან იკვეთება, რომ თუ cube და pyramid მეთოდებთან გამოყენებულია Tube მეთოდი, ისინი ერთნაირ შედეგებს აჩვენებენ. ასევე ერთნაირი შედეგები აქვთ cube და pyramid მეთოდებს Chain მეთოდთან

	2	3	4	5
1	#19-1	arb+tube	arb+chain	arb+cylinder
2	CPU/ms	77370	77760	77850
3	RAM/kb	638713	638717	638714
4	სხვაობა CPU/ms		90	480
5	სხვაობა RAM/kb		256	64
6	სხვაობა CPU/%		0.12	0.62
7	სხვაობა RAM/%		0.04	0.01
8	#19-2	pyramid+tube	pyramid+chain	pyramid+cylinder
9	CPU/ms	78630	78390	78150
10	RAM/kb	638702	638707	638704
11	სხვაობა CPU/ms		240	480
12	სხვაობა RAM/kb		320	128
13	სხვაობა CPU/%		0.31	0.61
14	სხვაობა RAM/%		0.05	0.02
15	#19-3	Cube+tube	Cube+chain	Cube+cylinder
16	CPU/ms	78750	78060	78440
17	RAM/kb	638702	638707	638703
18	სხვაობა CPU/ms	690		380
19	სხვაობა RAM/kb		320	64
20	სხვაობა CPU/%	0.88	0.00	
21	სხვაობა RAM/%		0.05	0.01

ცხრ.93 #19 მაგალითის სიმულაციის შედეგები

კომბინაციაში, თუმცა მაშინ როცა Tube არის გამოყენებული წარმადობას აქვს 0.05%-ით უკეთესი შედეგი Chain-თან მიმართებაში. 0.01%-ით უკეთესი შედეგი აქვს cube და Tube-ის კომბინაციას, cube და cylinder კომბინაციასთან მიმართებით და 0.02% ით უკეთესი შედეგი აქვს pyramid და Tube-ის კომბინაციას ვიდრე, pyramid და cylinder-ს.

რაც შეეხება arbitrary-ის კომბინაციას ცილინდრული ზედაპირების დაპროგრამების მეთოდებთან ცხრ. 93-დან ჩანს, რომ ყველაზე კარგი შედეგი აქვს tube მეთოდთან კომბინაციას, რომელსაც 0,04% ჩამორჩება chain და 0,01%-ით cylinder, ასევე განხორციელდა იქნა arbitrary-> tube და cube/pyramid->tube-ის შედეგების ანალიზი შედეგად გაირკვა, რომ სხვაობა cube/pyramid->tube-ის სასარგებლოდ შეადგენს 0.11%. arbitrary->chain და და cube/pyramid-> chain -ის ანალიზი შედეგები ასეთია: წარმადობის თვალსაზრისით cube/pyramid-> chain აქვს 0.1%-ით უკეთესი შედეგი და ასევე arbitrary->cylinder და Cube->cylinder-ის შედარების შედეგები ასეთია: 0.11%-ით უკეთესი შედეგი აქვს Cube->cylinder კომბინაციას. ამ შემთხვევაში

გამოყენებულ იქნა მხოლოდ Cube->cylinder-ის შედეგები, ვინაიდან კვლევით დადგინდა, რომ მას pyramid ->cylinder კომბინაციასთან შედარებით უკეთესი მაჩვენებლები ქონდა.

ამრიგად, განსაზღვრა დაპროგრამების მეთოდების რეიტინგი. ცილინდრული ზედაპირების დაპროგრამების მეთოდებს შორის ყველაზე კარგი შედეგები წარმადობის თვალსაზრისით აქვს Tube მეთოდს, რაც შეეხება Chain მეთოდს ის უკეთეს შედეგს აფიქსირებს CPU Time-ის მიხედვით Cylinder მეთოდთან, ხოლო უარესს შედეგს ოპერატიული მეხსიერების დატვირთვის მიხედვით. რაც შეეხება წახნაგოვანი ზედაპირების დაპროგრამების მეთოდებს, ყველაზე კარგი შედეგი აქვს CGS მეთოდებს Cube მეთოდს და შემდეგ Pyramid მეთოდს. შემდეგ მოდის პოლიგონ მეთოდების წარმომადგენლები Arbitrary, Symmetric და Double Symmetric მეთოდები, მათ ერთნაირი შედეგები აქვს წარმადობის თვალსაზრისით.

4.2.4 ობიექტების რაოდენობის სიმულაციის წარმადობაზე ბავლენის გამოკვლევა

გარდა ზემოთ აღნიშნული კვლევისა ასევე საინტერესოა ის თუ რა გავლენა აქვს სიმულაციის პროცესის წარმადობაზე ობიექტების რაოდენობას. ამის გამოსაკვლევად კი დღის წესრიგში დადგა შემდეგი ამოცანა, უნდა განხორციელდეს სიმულაციაში არსებული ცილინდრული და წახნაგოვანი ზედაპირების მქონე ობიექტების დათვლა. როგორც პირველ თავშია განხილული ატლასის სიმულაციას გეომეტრია მიეწოდება სხვადასხვა წყაროებიდან xml, GeoModel და ა.შ. საბოლოოდ მათი

```
1157 <second ref="G4Tubs0x426ae780"/>
1158 <position name="EMIShieldJoint0x426a2fa0_pos" unit="mm" x="0" y="0" z="-780.1035"/>
1159 </subtraction>
1160 <tube aunit="deg" deltaphi="360" lunit="mm" name="AttachmentPixelToSCT0x426ae870" rmax="270" rmin="255" startphi="0"
1161 <tube aunit="deg" deltaphi="360" lunit="mm" name="G4Tube0x40ae30e0" rmax="547.1" rmin="255" startphi="0" z="1610"/>
1162 <tube aunit="deg" deltaphi="360" lunit="mm" name="DiscSupport00x426aea50" rmax="567" rmin="267" startphi="0" z="9"/>
1163 <box lunit="mm" name="G4Box0x426a0540" x="0.375" y="10" z="5"/>
1164 <box lunit="mm" name="G4Box0x426a05b0" x="0.5" y="80.09" z="18.47"/>
1165 <trd lunit="mm" name="G4Trd0x426b6300" xl="0.5" x2="0.5" yl="80.09" y2="35" z="9.31"/>
1166 <union name="SCT_FwdHybrid00x40be4270">
1167 <first ref="G4Box0x426a05b0"/>
1168 <second ref="G4Trd0x426b6300"/>
```

სურ. 139 ფრაგმენტი geometry-ATLAS-R2-2016-01-00-01-ის GDML კოდიდან

გაერთიანება ხდება GDML-ში და ამ სახით მიეწოდება სიმულაციას. ამჟამად კი გეომეტრიის ოფიციალურ ვერსიას წარმოადგენს „geometry-ATLAS-R2-2016-01-00-01“. შესაბამისად განხორციელდა აღნიშნული ვერსიის ექსპორტი Geant4-დან GDML ფორმატში სურ. 139. სურათზე მოცემულია რამდენიმე პრიმიტივის დაპროგრამების სინტაქსი, კერძოდ აქ იქმნება 3 Tube პრიმიტივი, 2 Cube (box) და 1 pyramid (trd) პრიმიტივი. GDML კოდში შესაძლებელია ყველა მათგანის იდენტიფიცირება. შედეგად განხორციელდა ობიექტების რაოდენობის დათვლა და აღმოჩნდა, რომ ATLAS-ის ექსპერიმენტის სიმულაციაში ჯამში არის 7`060 ცილინდრული ობიექტი, აქედან 6`776 Tube, 169 Cylinder და 115 Chain მეთოდით არის შექმნილი. ასევე დათვლილ იქნა წახნაგოვანი ზედაპირების მქონე ობიექტებიც და მათი რაოდენობა შეადგენს 40`124-ს საიდანაც Cube არის 13`814 ცალი, ხოლო Pyramid კი 26`310 ცალი. აქედან გამომდინარე სიმულაციის პროცესის წარმადობაზე ობიექტების რაოდენობის წარმადობაზე გავლენის კვლევა განხორციელდება შემდეგ რაოდენობებზე: 1, 1000, 2000, 3000, 4000, 5000, 6000, 7000 ცილინდრული ობიექტებისათვის და 3000, 6000, 9000, 12000, 15000, 18000, 40000 წახნაგოვანი ობიექტებისათვის.

ცილინდრული ობიექტების კვლევა განხორციელდა #23 მაგალითზე სურ. 140, მას დაპროგრამების 3 მეთოდი შეესაბამება ცხრ. 94. ტესტის



სურ. 140 #23 მაგალითი

პერიოდში დაფიქსირებულ მაქსიმალურ და მინიმალურ მნიშვნელობებს შორის სხვაობა 550 მილი წამს შეადგენს და შესაბამისად ხმაურის კოეფიციენტი არის 0,82%-ს ცხრ. 94. შემდგომ ეტაპზე განხორციელდა

	CPU/ms										ხმაური%
lplus723.cern.ch	66540	66620	66190	66400	66430	66770	66220	66890	66850	66750	0.82

ცხრ.94 ხმაურის კოეფიციენტი #23 მაგალითისთვის

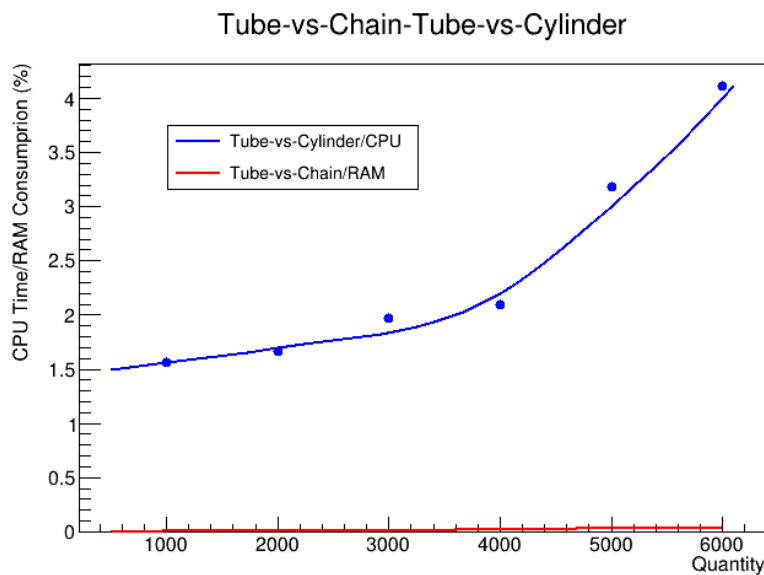
რაოდენობების მიხედვით შესაბამისი კოდების ფორმირება და შემდეგ

სიმულაციაში გაშვება. შედეგები მოცემულია ცხრ. 95-ში. როგორც ცხრილიდან ჩანს ნებისმიერი რაოდენობის ობიექტების გამოყენებისას როგორც ოპერატიული მეხსიერების ასევე CPU-ს დატვირთვის მიხედვით საუკეთესო შედეგი აქვს Tube მეთოდს. ოპერატიული მეხსიერების

Serie	Measurm.	Tube	Chain	Cylinder	Diff			%		
					Tube	Chain	Cylinder	Tube	Chain	Cylinder
1	CPU/ms	66810	67280	66510	300	770	0	0.45	1.15	0
	RAM/kb	638886	638889	638887	0	3	1	0	0.0005	0.0002
1000	CPU/ms	66740	66770	67800	0	30	1060	0	0.045	1.563
	RAM/kb	641014	641048	641016	0	34	2	0	0.005	0.0003
2000	CPU/ms	67030	69300	68170	0	2270	1140	0	3.2	1.67
	RAM/kb	643161	643222	643163	0	61	2	0	0.009	0.0003
3000	CPU/ms	67230	66570	67910	660	0	1340	0.98	0	1.97
	RAM/kb	645408	645498	645409	0	90	1	0	0.0014	0.0002
4000	CPU/ms	67780	68010	67790	380	0	1420	0.57	0	2.09
	RAM/kb	647445	647572	647447	0	127	2	0	0.02	0.003
5000	CPU/ms	66720	67000	68910	0	280	2190	0	0.42	3.18
	RAM/kb	649935	650109	649936	0	174	1	0	0.03	0.0002
6000	CPU/ms	66670	67490	69530	0	820	2860	0	1.22	4.11
	RAM/kb	651924	652126	651925	0	202	1	0	0.03	0.0002
7000	CPU/ms	66910	67300	69620	0	390	2710	0	0.57949	3.89256
	RAM/kb	653925	654161	653926	0	236	1	0	0.03608	0.00015

ცხრ.95 #23 მაგალითი. სხვადასხვა რაოდენობის ობიექტების სიმულაციის შედეგები

დატვირთვის მიხედვით ყველაზე ცუდი შედეგი აქვს Chain მეთოდს, ხოლო CPU-ს მიხედვით კი Cylinder მეთოდი აფიქსირებს ყველაზე უარეს შედეგს, სხვაობა 4%-ზე მეტია Tube მეთოდთან მიმართებაში. სურ. 141-ზე წარმოდგენილია # 23 მაგალითის შედეგები გრაფიკის სახით.



სურ. 141 Tube-vs-Chain-vs-Cylinder-ის ობიექტების რაოდენობაზე დამოკიდებული სიმულაციის შედეგები

ტესტირება ჩატარდა ასევე #11 მაგალითზე, რომელიც ცილინდრული კლასის წარმომადგენელია მას 2 დაპროგრამების მეთოდი შეესაბამება ცხრ. 96. პირველ შემთხვევაში გამოყენებულია 1 Cylinder მეთოდი, ხოლო მეორე შემთხვევაში იგივე შედეგი მიიღწევა 5 დამოუკიდებელი Tube მეთოდის გამოყენებით. განხორციელდა ტესტირებისთვის საჭირო 1, 1000, 2000, 3000,

#11-1	#11-2
001 Cylinder	001 Tube
	002 Tube
	003 Tube
	004 Tube
	005 Tube

ცხრ.96 #11 მაგალითის დაპროგრამების მეთოდები

4000, 5000, 6000, 7000 რაოდენობის ობიექტების შესაბამისი კოდების შექმნა.

შემდეგ ეტაპზე კი განისაზღვრა ხმაურის კოეფიციენტი ცხრ. 97. ტესტის

	CPU/ms									ხმაური %
lxplus723.cern.ch	67090	66610	66890	66880	66860	66690	66940	66790	67200	0.88

ცხრ.97 ხმაურის კოეფიციენტი #11 მაგალითისთვის

ცხრ. 97 ხმაურის კოეფიციენტი #11 მაგალითისთვის

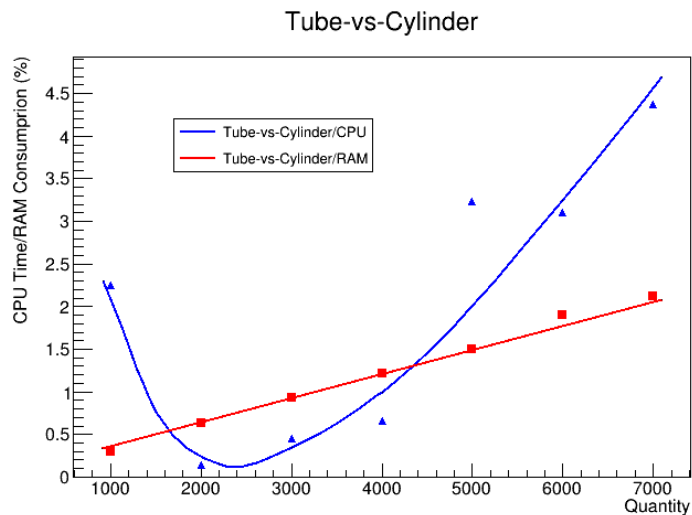
პერიოდში დაფიქსირებულ მაქსიმალურ და მინიმალურ მნიშვნელობებს შორის სხვაობა 590 მილი წამს შეადგენს და შესაბამისად ხმაურის

Serie	Measurm.	Tube	Cylinder	Diff		%	
				Tube	Cylinder	Tube	Cylinder
1	CPU/ms	66740	66380	360	0	0.54	0
	RAM/kb	638889	638892	0	3	0	0.0005
1000	CPU/ms	67870	66340	1530	0	2.25	0
	RAM/kb	642302	640355	1947	0	0.3	0
2000	CPU/ms	66970	66880	90	0	0.14	0
	RAM/kb	645859	641794	4065	2	0.63	0
3000	CPU/ms	66930	66630	300	0	0.45	0
	RAM/kb	649363	643350	6013	0	0.93	0
4000	CPU/ms	67710	67260	450	0	0.66	0
	RAM/kb	652618	644724	7894	0	1.21	0
5000	CPU/ms	66960	69200	0	2240	0	3.24
	RAM/kb	656327	646486	9841	0	1.5	0
6000	CPU/ms	68250	70440	0	2190	0	3.11
	RAM/kb	660085	647832	12253	0	1.9	0
7000	CPU/ms	67800	70900	0	3100	0	4.37
	RAM/kb	663342	649306	14036	0	2.12	0

ცხრ. 98 #11 მაგალითი. სხვადასხვა რაოდენობის ობიექტების სიმულაციის შედეგები

კოეფიციენტი არის 0,88%-ს. ცხრ. 98-ზე წარმოდგენილია სიმულაციის შედეგები. როგორც იკვეთება Cylinder მეთოდმა RAM-ის გამოყენების

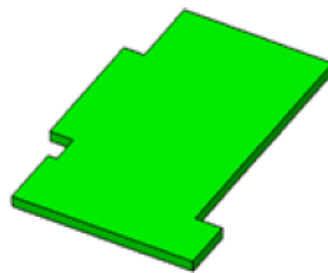
მიხედვით დააფიქსირა საუკეთესო შედეგი ვიდრე 5 ცალი Tube-მა, რაც შეეხება CPU Time-ს როცა მცირე რაოდენობებია გამოყენებული მაგალითად 2200-ზე ნაკლები, მაშინ უფრო მეტ CPU Time-ს მოითხოვს Tube ვიდრე Cylinder, რაც შეეხება 2200 ზე მეტი ობიექტის გამოყენებისას კი, უპირატესობა გადადის Tube მეთოდის სასარგებლოდ და სხვაობა შეადგენს



სურ. 142 #11 მაგალითი ობიექტების რაოდენობაზე დამოკიდებული სიმულაციის შედეგები

4,4%-ს. სურ. 142-ზე წარმოდგენილია # 11 მაგალითის სიმულაციის შედეგები გრაფიკის სახით.

შემდგომ ეტაპზე იგივე კვლევა ჩატარდა წახნაგოვან კლასის #39 მაგალითზე სურ. 143. აღნიშნულ მაგალითს შეესაბამება 2 დაპროგრამების



სურ. 143 #39 მაგალითი

#39-1	#39-2
001 Cube 002 Cube 003 Cube 004 Cube 005 Cube 006 Subtraction 007 Subtraction 008 Subtraction 009 Subtraction	001 Arbitrary

ცხრ.99 #39 მაგალითის დაპროგრამების მეთოდები

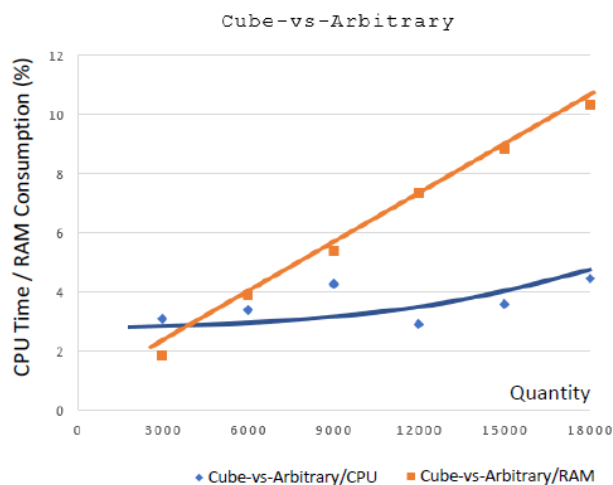
მეთოდი ცხრ. 99. პირველ მეთოდში გამოყენებულია CGS მეთოდების წარმომადგენელია Cube მეთოდები, ვინაიდან რთული ობიექტია დასაპროგრამებელი, ამიტომ ასევე აუცილებელია Boolean ოპერატორების გამოყენება. რაც შეეხება მე-2 მეთოდს აქ გამოყენებულია „პოლიგონი“ მეთოდების ერთერთი წარმომადგენელი Arbitrary. შემდგომ ეტაპზე განხორციელდა 3000, 6000, 9000, 12000, 15000, 18000 და 40000 ობიექტის

	CPU/ms										ხმაური %
lxplus723.cern.ch	67290	66930	67350	67200	68230	67920	67460	69150	69570	69230	3.92

ცხრ.100 #39 მაგალითის CPU time-ის ხმაურის კოეფიციენტი კოდების ფორმირება და მათი სიმულაციაში გაშვება. ტესტირების მომენტში CPU Time-ის ხმაურის კოეფიციენტი შეადგენს 2.92%-ს. ცხრ. 100. ცხრ. 101-

Serie	Measurm.	Cube	Pyramid	arbitrary	Diff				%			
					Cube vs Pyramid	Cube vs arbitrary	Cube vs Pyramid	Cube vs arbitrary				
3000	CPU/ms	69520	68910	67380	610	0	2140	0	0.88	0	3.08	0
	RAM/kb	655615	655616	643414	0	1	12201	0	0	0.00015	1.86	0
6000	CPU/ms	68960	69650	66540	0	690	2420	0	0	0.99	3.51	0
	RAM/kb	672590	672591	647846	0	1	24744	0	0	0.00015	3.68	0
9000	CPU/ms	69870	70350	66880	0	480	2990	0	0	0.68	4.28	0
	RAM/kb	689757	689758	652628	0	1	37129	0	0	0.00014	5.38	0
12000	CPU/ms	69820	70950	67790	0	1130	2030	0	0	1.59	2.91	0
	RAM/kb	709003	709004	656649	0	1	52354	0	0	0.00014	7.38	0
15000	CPU/ms	71030	71180	68470	0	150	2560	0	0	0.21	3.60	0
	RAM/kb	725126	725127	660945	0	1	64181	0	0	0.00014	8.85	0
18000	CPU/ms	72840	74730	69600	0	1890	3240	0	0	2.53	4.45	0
	RAM/kb	743084	743085	666264	0	1	76820	0	0	0.00013	10.34	0
40000	CPU/ms	77330	78340	71050	0	1010	6280	0	0	1.29	8.12	0
	RAM/kb	869102	869103	699145	0	1	169957	0	0	0.00012	19.56	0

ცხრ.101 #39 მაგალითის სიმულაციის შედეგები ზე კი წარმოდგენილია #39 მაგალითის სიმულაციის შედეგები. როგორც



სურ. 144 #39 სიმულაციის შედეგები

ცხრილიდან ჩანს, რომ პოლიგონ მეთოდი, ყველა შემთხვევაში უკეთეს შედეგს აფიქსირებს ვიდრე ამ შემთხვევაში გამოყენებული CGS მეთოდები. ობიექტების რაოდენობების ზრდასთან ერთად იზრდება სხვაობები როგორც RAM-ში ასევე CPU Time- ში. მაქსიმალური სხვაობა CPU Time- ებს შორის 8%-ს, ხოლო RAM-ებს შორის 20%-ს აღწევს. სურ. 144-ზე წარმოდგენილია #39 მაგალითის სიმულაციის შედეგები გრაფიკის სახით.

შემდგომი კვლევა ჩატარდა შერეული კლასის # 19 მაგალითზე სურ. 145. მის შესაქმნელად 2 დაპროგრამების მეთოდი იქნა შერჩეული ცხრ. 102.



სურ. 145 #19 მაგალითი

#19-1	#19-2
001 Arbitrary	001 Pyramid
002 Tube	002 Pyramid
003 Subtraction	003 Subtraction
004 Subtraction	004 Subtraction
	005 Tube
	006 Subtraction
	007 Subtraction

ცხრ.102 #19 მაგალითის დაპროგრამების მეთოდები

პირველ შემთხვევაში გამოყენებულია Arbitrary, Tube მეთოდთან და Boolean-ის ოპერატორებ თან ერთად, ხოლო მეორე შემთხვევაში კი Pyramid, Tube მეთოდთან და Boolean-ის ოპერატორებთან ერთად. ტესტირების მომენტში CPU Time-ის ხმაურის კოეფიციენტი შეადგენს 3.15%-ს ცხრ. 103. შესწავლილ

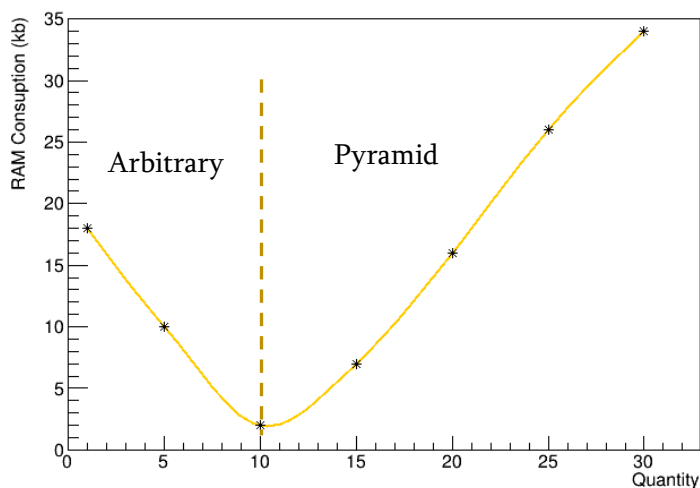
Ixpplus723.cern.ch	CPU/ms										ხმაური %
	66280	64930	67040	66330	65460	65130	66660	65970	65610	65070	
											3.15

ცხრ.103 #19 CPU time-ის ხმაურის კოეფიციენტი

იქნა ობიექტების შემდეგი რაოდენობები 1, 10, 15, 20, 25, 30. სიმულაციის შედეგები წარმოდგენილია ცხრ. 104-ში. მიღებული შედეგები შემდეგნაირია, 10 ობიექტამდე რაოდენობის ტესტირების შედეგად უპირატესობა აქვს Pyramid მეთოდს, ხოლო ობიექტების რაოდენობის გაზრდასთან ერთად უპირატესობა გადადის Arbitrary მეთოდის მხარეს. #19 მაგალითის სიმულაციის შედეგები გრაფიკის სახით წარმოდგენილია სურ. 146-ზე.

Serie	Measurm.	Diff				%	
		Arb+Tube	Pyr+Tube	Arb+Tube	Pyr+Tube	Arb+Tube	Pyr+Tube
1	CPU/ms	65190	65830	0	640	0	0.97
	RAM/kb	638911	638893	18	0	0.003	0
10	CPU/ms	64970	65490	0	520	0	0.79
	RAM/kb	638925	638913	12	0	0.002	0
15	CPU/ms	65050	65320	0	270	0	0.41
	RAM/kb	638945	638943	2	0	0.0003	0
20	CPU/ms	64840	64920	0	80	0	0.12
	RAM/kb	638966	638973	0	7	0	0.001
25	CPU/ms	65170	64880	290	0	0.44	0
	RAM/kb	638984	639000	0	16	0	0.003
30	CPU/ms	65220	65580	0	360	0	0.55
	RAM/kb	639002	639028	0	26	0	0.004

ცხრ.104 #19 მაგალითის სიმულაციის შედეგები
Arbitrary-vs-Pyramid



სურ. 146 #19 მაგალითის სიმულაციის შედეგები

ამრიგად, ობიექტების რაოდენობა გავლენას ახდენს სიმულაციის პროცესის წარმადობაზე.

4.3 მე-4 თავის დასკვნები

1. დაპროგრამების „CGS“ მეთოდებში ცილინდრული ობიექტებისათვის ყველაზე კარგი შედეგი აქვს Tube მეთოდს, რაც შეეხება Chain მეთოდს ის უმჯობესია ვიდრე cylinder მეთოდი CPU Time-ის მიხედვით და უარესი ვიდრე cylinder ოპერატიული მეხსიერების დატვირთვის მიხედვით.

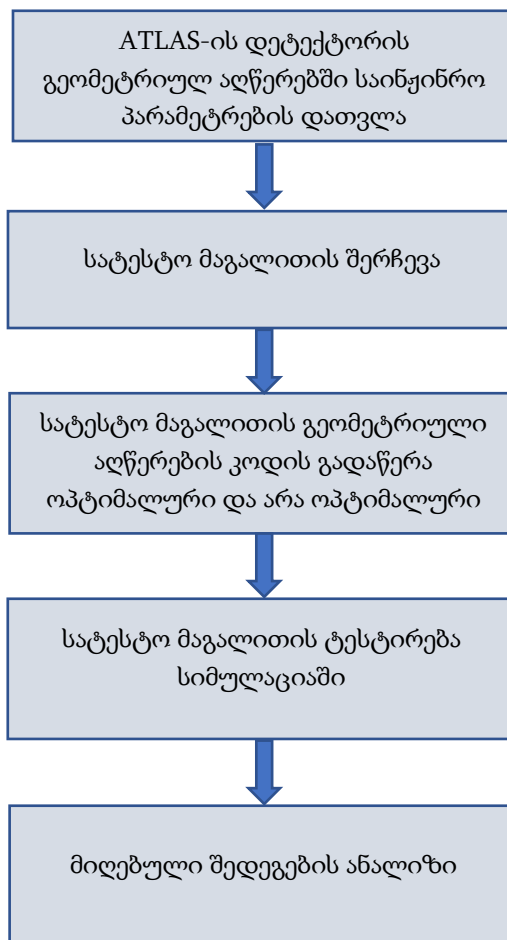
2. წახნაგოვანი ზედაპირის მქონე ობიექტების დაპროგრამების მეთოდებს Cube და Pyramid-ს შორის, Cube მეთოდს აქვს უკეთესი წარმადობა.
3. წახნაგოვანი ზედაპირის მქონე ობიექტების დაპროგრამების მეთოდებს Arbitrary და Symmetric წარმადობის თვალსაზრისით ერთნაირი შედეგი აქვთ.
4. შერეულ კლასში cube და pyramid მეთოდებს Tube მეთოდთან კომბინაციაში წარმადობის თვალსაზრისით ერთნაირი შედეგები აქვთ, მაშინ როცა დამოუკიდებლად Cube მეთოდი უკეთეს შედეგს აჩვენებს ვიდრე pyramid.
5. შერეულ კლასში თუ გამოყენებულია cube და pyramid მეთოდები chain მეთოდთან კომბინაციაში წარმადობის თვალსაზრისით უკეთეს შედეგს აჩვენებს Cube მეთოდი, ვიდრე pyramid. წახნაგოვანი კლასის კვლევის დროს დაფიქსირებულ შედეგებისგან განსხვავებით გაიზარდა სხვაობა Cube მეთოდის სასარგებლოდ.
6. შერეულ კლასში თუ გამოყენებულია cube და pyramid მეთოდები cylinder მეთოდთან კომბინაციაში წარმადობის თვალსაზრისით უკეთეს შედეგს აჩვენებს Cube მეთოდი, ვიდრე pyramid. ეს იგივეა, რაც წახნაგოვანი კლასის კვლევის დროს დაფიქსირებულ შედეგები.
7. დაპროგრამების „CGS“ მეთოდებში ცილინდრული ობიექტებისათვის ყველაზე კარგი შედეგი აქვს Tube მეთოდს რაც მკვეთრად გამოხატულია CPU Time-ში. იმვითავე შემთხვევებში კი Cylinder მეთოდს აქვს უპირატესობა, როცა ის წახნაგოვანი ზედაპირების დაპროგრამების „პოლიგონი“ მეთოდებთან არის კომბინაციაში, მაგალითად Arbitrary.
8. წახნაგოვან ობიექტების ასაგებად „პოლიგონი“ მეთოდების ერთერთი წარმომადგენლის Arbitrary-ის გამოყენება, რიგ შემთხვევებში 20%-ით უკეთეს შედეგს იძლევა, ვიდრე „CGS“ მეთოდებიდან cube/Pyramid-ის გამოყენება Boolean ფუნქციასთან კომბინაციაში.
9. განისაღზვრა „CGS“ და „პოლიგონების“ მეთოდებში შემავალი მეთოდების რეიტინგი, თუ რომელი მათგანი იძლევა საუკეთესო

წარმადობას სიმულაციას. ცილინდრული ზედაპირების დაპროგრამების მეთოდები: 1. Tube 2-3. Chain უკეთესია ვიდრე Cylinder პროცესორის დატვირთვის მიხედვით და უარესი ოპერატიული მახსიერების დატვირთვის მიხედვით. წახნაგოვანი ზედაპირების დაპროექტების მეთოდები 1. Cube, 2. Pyramid, 3-4-5. Arbitrary, Symmetric, Double Symmetric.

თავი V: კვლევის შედეგების აპრობაცია

5.1 კვლევის შედეგების აპრობაციის ეტაპები

დისერტაციის ფარგლებში განხორციელებული კვლევის შედეგად მიღებული შედეგების აპრობაციისათვის რამდენიმე ეტაპის განხორციელებაა საჭირო. ცხრ.105-ში ეს პროცესი წარმოდგენილია სქემატურად. პირველ ეტაპზე უნდა განხორციელდეს ATLAS-ის დეტექტორის გეომეტრიულ აღწერებში საინჟინრო პარამეტრების დათვლა. მეორე ეტაპზე კი სატესტო მაგალითის შერჩევა. მესამე ეტაპზე უნდა მოხდეს სატესტო მაგალითის გეომეტრიული აღწერების კოდის გადაწერა ოპტიმალური და არა ოპტიმალური გზებით. მეოთხე ეტაპის ამოცანას კი წარმოადგენს სატესტო მაგალითის ტესტირება სიმულაციაში და ბოლო მეხუთე ეტაპის ამოცანაა მიღებული შედეგების ანალიზი.



ცხრ.105 მიღებული კვლევის შედეგების აპრობაციის ეტაპები გამოსახული სქემატურად

5.2 ATLAS-ის დეტექტორის გეომეტრიული აღწერების კომპლექსურობის ბანსაზღვრა

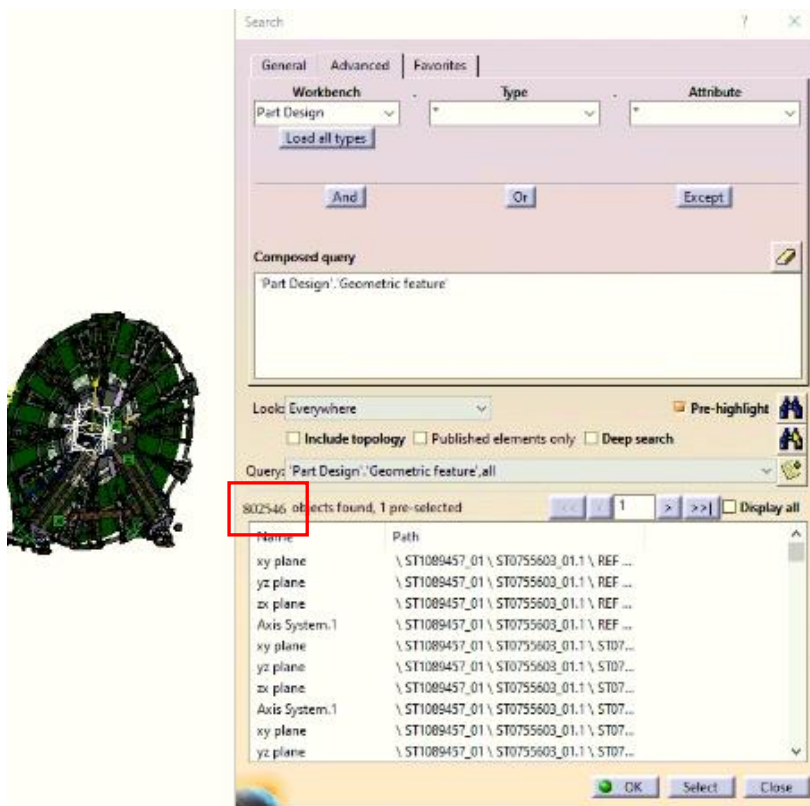
ამ ეტაპის ამოცანაა ATLAS-ის დეტექტორის გეომეტრიულ აღწერებში სხვადასხვა საინჟინრო პარამეტრების დათვლა, როგორცაა ანაწყოების, დეტალების და პრიმიტივების რაოდენობა. ამ სამუშაოს ჩატარების აუცილებლობა განპირობებულია იმით, რომ თუ გვეცოდინება რამდენი პრიმიტივისგან შეგდება დეტექტორის სხვადასხვა სტრუქტურები, შეგვეძლება განვსაზღვროთ მათი მასშტაბურობა რაც საშუალებას მოგვცემს სატესტო მაგალითი შერჩეულ იქნეს ოპტიმალურად. მიღებული შედეგების მიხედვით ასევე შესაძლებელი იქნება კვლევის შედეგების განზოგადება დეტექტორის გეომეტრიულ აღწერებზე და მეთოდების ეფექტურობის გამოთვლა. აღნიშნული სამუშაო ჩატარდა საინჟინრო პროგრამა CATIA-ში და მთლიან დეტექტორში ჯამში პრიმიტივების რაოდენობა 47`033`799-ს, ანაწყოების 5`529-ს, ხოლო დეტალების რაოდენობა კი 1`398`162-ს შეადგენს ცხრ. 106. აქედან მაგნიტური სისტემა შედგება 347 ანაწყოების,

ატლასის დეტექტორი .	ანაწყოები	დეტალი	პრიმიტივი
მაგნიტური სისტემა	347	16`345	398`924
შიდა დეტექტორი	413	337`484	9`024`069
კალორიმეტრები	568	119`111	23`490`492
მიონური სისტემა	3`266	915`756	13`380`286
მექანიკური სისტემა	935	9`466	740`028
ჯამი:	5`529	1`398`162	47`033`799

ცხრ.106 ATLAS-ის დეტექტორის გეომეტრიული აღწერების ზოგადი
პარამეტრები

16`345 დეტალის და 398`924 პრიმიტივისგან. შიდა დეტექტორი შეიცავს 413 ანაწყოებს 337`484 დეტალს და 9`024`069 პრიმიტივს. კალორიმეტრების სისტემა შედგება 568 ანაწყოების, 119`111 დეტალის და 23`490`492

პრიმიტივისგან. მიონური სისტემები კი შეიცავს 3`266 ანაწყობს 915`756 დეტალს და 13`380`286 პრიმიტივს და ბოლოს მექანიკური სტრუქტურები,

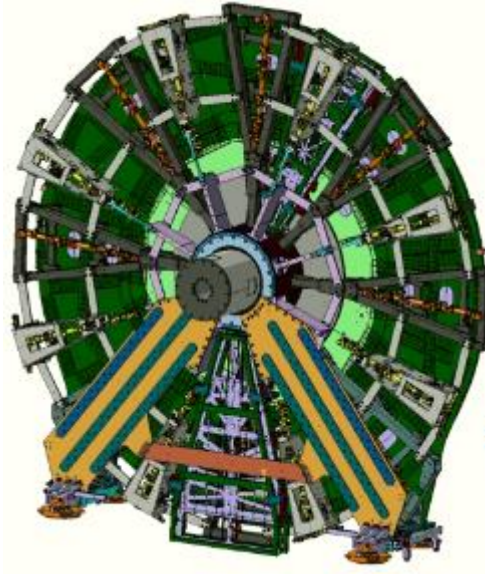


სურ. 147 NSW-ში პრიმიტივების რაოდენობა. ფრაგმენტი CATIA-დან ისინი შედგებიან 935 ანაწყობის, 9`466 დეტალის და 740`028 პრიმიტივისგან. მაგალითი თუ როგორ განხორციელდა გეომეტრიული აღწერების სხვადასხვა საინჟინრო პარამეტრების CATIA-ში დათვლა წარმოდგენილია სურ. 147-ზე სადაც ხდება NSW-ში პრიმიტივების რაოდენობის კალკულაცია.

5.3 სატესტო მაბალოთის შერჩევა აპრობაციისთვის

აღნიშნული ეტაპის ამოცანას წარმოადგენს სატესტო მაგალითის შერჩევა. ოპტიმალური მაგალითის შერჩევა არის აქტუალური ამოცანა გამომდინარე იქიდან, რომ მასზედ ჩატარებული კვლევის შედეგების აპრობაციამ უნდა მოგვცეს რეალურთან მიახლოებული შედეგი. აღნიშნულ მაგალითზე სიმულაციიდან მიღებული შედეგების მიხედვით უნდა განხორციელდეს დაპროგრამების მეთოდების ეფექტურობის შეფასება. როგორც ცხრ. 107-იდან ჩანს ATLAS-ის დეტექტორში ერთერთი ყველაზე

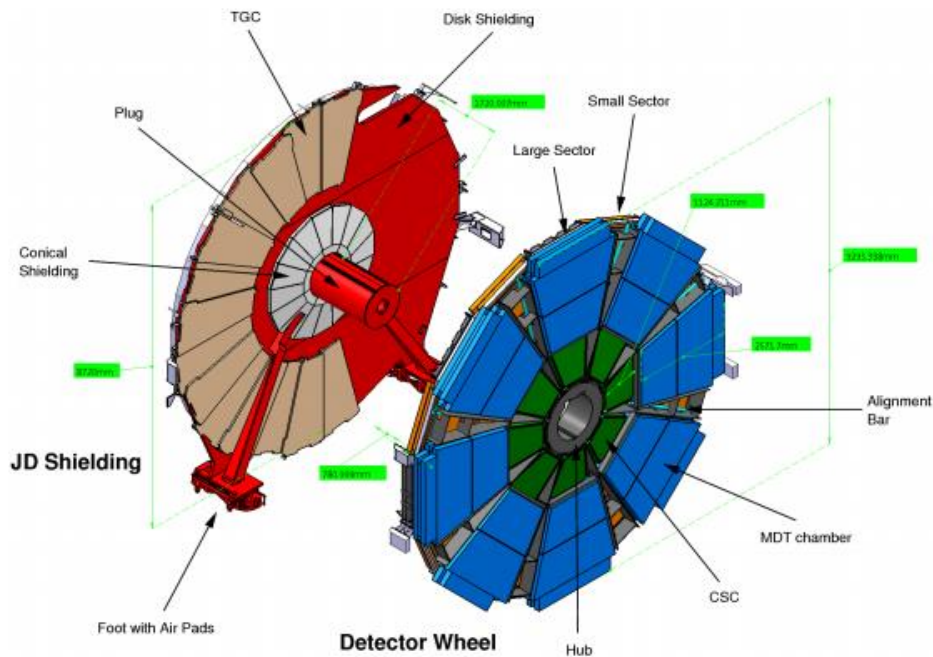
კომპლექსურია მიონური სისტემა. ამიტომ განხორციელდა მისი ერთერთი ქვე დეტექტორის NSW (New Small Wheel)-ის გამოკვლევა სურ. 147,



სურ. 148 ATLAS-ექსპერიმენტის New Small Wheel

აღმოჩნდა, რომ მასში პრიმიტივების რაოდენობა შეადგენს 802'546-ს ანაწყოების-623-ს და დეტალების კი - 4'082-ს. როგორც ჩანს NSW საკმაოდ კომპლექსურია და მოიცავს ყველა ტიპის გეომეტრიულ ობიექტებს, როგორც წახნაგოვანს, ასევე ცილინდრულს და შერეულს. ამიტომ აპრობაცია სწორედ NSW-ის მაგალითზე განხორციელდება სურ. 148. ის არის ATLAS-ის დეტექტორის მიონური სისტემის ერთერთი დეტექტორი. LHC-ის ექსპერიმენტის მიმდინარეობის გარკვეულ ეტაპზე დღის წესრიგში დადგა Luminosity-ის 5-ის ფაქტორით გაზრდის საკითხი [46][47]. Luminosity არის ექსპერიმენტის ერთერთი მნიშვნელოვანი პარამეტრი, რომლითაც განისაზღვრება პროტონების წარმოქმნის სიხშირე. ამ მიზნის უზრუნველსაყოფად 2018-2019 წლებში განხორციელდა NSW-ის დეტექტორის განახლება. ამიტომ საჭირო გახდა სიმულაციაში არსებული NSW-ის გეომეტრიული აღწერების განახლებაც. შესაბამისად საქართველოს ტექნიკური უნივერსიტეტის „ბირთვული ინჟინერიის ცენტრის“ მიერ 2020 წელს დამუშავდა NSW-ის გეომეტრიული აღწერების დამუშავება ATLAS-ის ექსპერიმენტის სიმულაციისათვის [48]. სწორედ ეს აღწერები იქნა გამოყენებული კვლევაში. NSW თავისი მონაცემებით ერთ-ერთი

გამორჩეული დეტექტორია. მისი დიამეტრი დაახლოებით 10მ-ია, ხოლო მასა 70ტ-ზე მეტი [49]. NSW რამდენიმე ძირითადი ნაწილისგან შედგება [47].



სურ. 149 NSW-ის ძირითადი ნაწილები

როგორცაა: Chambers, JD disk, Hub Services. სურ 149-ზე წარმოდგენილია დეტალურად NSW-ის შედგენილობა.

5.4 NSW-ის სიმულაციის სატესტო სესიებისთვის კოდების ფორმირება

როგორც წინა პარაგრაფშია აღწერილი კვლევის შედეგების აპრობაციისათვის შერჩეულ იქნა NSW. ამ ეტაპის ამოცანას წარმოადგენს მისი კოდის 2 ვარიანტის შექმნა სიმულაციისთვის, 1-ლი იქნება ოპტიმალური დაპროგრამების მეთოდებით აღწერილი, მე-2 კი არა ოპტიმალური დაპროგრამების მეთოდებით. ამ ამოცანის შესრულების შემდეგ საშუალება გვექნება თითოეული მათგანის ტესტირება განხორციელდეს სიმულაციაში და მიღებული შედეგებით, კი შეფასდეს მეთოდების ეფექტურობა. NSW-ის გეომეტრიული აღწერების კოდი საკმაოდ კომპლექსურია. ის შედგება 2074 პროგრამული სტრიქონისგან სურ.150.

```

2 <!--cube arbitrary-->
3 <!-- LS Spacer Frame -->
4 <gvxy name="LS_StructuralPlate_Main" material="Aluminium" dZ="5.9">
5 <gvxy_point X_Y="226.; 958."/>
6 <gvxy_point X_Y="914.15; 3281."/>
7 <gvxy_point X_Y="1007.3; 4613."/>
8 <gvxy_point X_Y="0.01; 4613."/>
9 <gvxy_point X_Y="0.01; 958."/>
10 </gvxy>

```



```

2064 <composition name="NSW_missing_parts" >
2065 <posXYZ volume="NSW_Missing_Parts_SideA" />
2066 <posXYZ volume="NSW_Missing_Parts_SideC" />
2067 </composition>
2068 <composition name="NSW" >
2069 <posXYZ volume="NSW_Spacer" />
2070 <posXYZ volume="NSW_Aluminum_Structure_and_HUB" />
2071 <posXYZ volume="NJD_Disk" />
2072 <posXYZ volume="A_Platee" />
2073 <posXYZ volume="NSW_missing_parts" />
2074 </composition>

```

სურ. 150 ფრაგმენტი NSW-ის პროგრამული კოდიდან

შესრულებული სამუშაოს ფარგლებში განხორციელდა NSW-ის გომეტრიული აღწერის გადაწერა 2 სხვადასხვა მეთოდით, პირველ შემთხვევაში მისი კოდი შეიქმნა კვლევების შედეგად დადასტურებული ოპტიმალური მეთოდებით, და მეორე პირიქით ყველაზე უარესი შედეგების მქონე მეთოდებით:

ოპტიმალური მეთოდები - წახნაგოვანი

```

<box name="SS_Central_reference_bar" material="Aluminium" X_Y_Z="1012.; 100.; 28." />
<box name="SS_Channel_section_L1100_Main" material="Aluminium" X_Y_Z="50.; 1100.; 30." />
<subtraction name="SS_Channel_section_L1100" >
  <posXYZ volume="SS_Channel_section_L1100_Main" />
  <posXYZ volume="LS_Channel_section_L2000_Cut" />
</subtraction>

```

არა ოპტიმალური მეთოდები - წახნაგოვანი

```

<trd name="SS_Central_reference_bar" material="Aluminium" Xmp_Ymp_Z="1012.; 1012.; 100.; 100.; 28." />
<trd name="SS_Channel_section_L1100_Main" material="Aluminium" Xmp_Ymp_Z="50.; 50.; 1100.; 1100.; 30." />
<subtraction name="SS_Channel_section_L1100" >
  <posXYZ volume="SS_Channel_section_L1100_Main" />
  <posXYZ volume="LS_Channel_section_L2000_Cut" />
</subtraction>

```

ოპტიმალური მეთოდები - ცილინდრული

```

<tubs name="HUB_Extension_outer_tube" material="ShieldSteel" Rio_Z="455.5; 530.; 998.7" />

```

არა ოპტიმალური მეთოდები - ცილინდრული

```

<snake name="test5" material="ShieldSteel" radius="530." nbPhi="20">
  <snake_point X_Y_Z="0.; 0.; -499.35" />
  <snake_point X_Y_Z="0.; 0.; 499.35" />
</snake>
<snake name="test6" material="ShieldSteel" radius="455.5" nbPhi="20">
  <snake_point X_Y_Z="0.; 0.; -500." />
  <snake_point X_Y_Z="0.; 0.; 500." />
</snake>
<subtraction name="HUB_Extension_outer_tube" >
  <posXYZ volume="test5" X_Y_Z="0.; 0.; 0." />
  <posXYZ volume="test6" X_Y_Z="0.; 0.; 0." />
</subtraction>

```

როგორც ზემოთ მოცემული კოდიდან ჩანს ოპტიმალურ შემთხვევაში CGS მეთოდებიდან წახნაგოვანი ობიექტებისთვის გამოყენებულ იქნა 2 ცალი Cube, და ასევე 1 ცალი Subtract Boolean ოპერატორი ობიექტის ამოსაჭრელად.

Cube მეთოდი იქნა არჩეული იმიტომ, რომ მან როგორც კვლევებით დადასტურდა წახნაგოვან CGS მეთოდებს შორის საუკეთესო წარმადობა დააფიქსირა. ხოლო არა ოპტიმალურ შემთხვევაში კი გამოყენებულია 2 ცალი Pyramid მეთოდი. ასევე აქ გამოყენებულია ასევე 1 ცალი Subtract Boolean ოპერატორი ობიექტის ამოსაჭრელად. განვიხილოთ CGS მეთოდები ცილინდრული ზედაპირების მქონე ობიექტების დასაპროგრამებლად, ამ მაგალითში ოპტიმალურ შემთხვევაში გამოყენებულ იქნა 1 ცალი Tube მეთოდი და იგივე ობიექტის არა ოპტიმალური გზით დასაპროგრამებლად გამოყენებულ იქნა 2 Chain მეთოდი და ასევე 1 ცალი Subtract Boolean ოპერატორი ობიექტის ამოსაჭრელად.

რაც შეეხება პოლიგონ მეთოდებს, ოპტიმალურ ვარიანტში გამოყენებულია 98 Arbitrary მეთოდი. არა ოპტიმალურ შემთხვევაში კი ისინი უნდა შეიცვალონ pyramid მეთოდით, რომელიც კომბინაციაში იქნება Boolean მეთოდებთან. NSW-ში გამოყენებულ 98 Arbitrary-დან 7 ცალი არის 3 წერტილიანი, 41 ცალი არის 4 წერტილიანი, 4 ცალი 5 წერტილიანი. 11 ცალი 6 წერტილიანი, 10 ცალი 7 წერტილიანი,, 15 ცალი 8 წერტილიანი, 3 ცალი 9 წერტილიანი, 1 ცალი 10 წერტილიანი, 1 ცალი 11 წერტილიანი, 4 ცალი 12 წერტილიანი და 1 ცალი 14 წერტილიანი ცხრ. 107. როგორც გამოიკვეთა

	3	4	5	6	7	8	9	10	11	12	14	ჯამი
წერტილების რაოდენობა arbitrary-ში	7	41	4	11	10	15	3	1	1	4	1	98
წერტილების რაოდენობა-ჯამი	21	164	20	66	70	120	27	10	11	48	14	571

ცხრ.107 NSW -ში გამოყენებული arbitrary მეთოდის ანალიზის შედეგები ყველაზე დიდი რაოდენობით NSW-ში გამოყენებულია 4 წერტილიანი Arbitrary, სულ 41 ცალი, ამიტომ კვლევისათვის გამოყენებულ უნდა იქნეს სწორედ 4 წერტილიანი 98 Arbitrary, ვინაიდან ჯამში სულ 98 ცალს შეადგენს Arbitrary-ების რაოდენობა. თუმცა რა თქმა უნდა ეს რეალურ შედეგს არ ასახავს, რადგანაც რეალურად მათი რაოდენობა არის 41 დანარჩენი კი განსხვავებული რაოდენობის წერტილს შეიცავს. ამიტომ უნდა გამოთლილ იქნა შესაძლო ცდომილება. 41 ცალი 4 წერტილიანი Arbitray ჯამში შეიცავს 392 წერტილს, ანუ საერთო რაოდენობის დაახლოებით 69%-ს. ჯამში

წერტილების რაოდენობა შეადგენს 571-ს. ცდომილება შედგენს 31%-ს. სხვაობის მინიმუმამდე შემცირების მიზნით საჭიროა კვლევისთვის გამოყენებული 98 ცალი 4 წერტილიანი Arbitray-ის რაოდენობის გაზრდა 31%-ით და ეს რიცხვი შეადგენს 45-ს. ამრიგად, კვლევისთვის გამოყენებულ უნდა იქნეს 143 ცალი 4 წერტილიანი Arbitrary.

ამრიგად, სიმულაციის სატესტო სესიებისთვის განხორციელდა NSW-ის დაპროგრამება, ოპტიმალური და არა ოპტიმალური გზებით.

5.5 სიმულაციის სატესტო სესიები NSW-ში

ამ ეტაპის მიზანია განხორციელდეს სიმულაციის სატესტო სესიების ჩატარება იმიტომ, რომ მიღებულ იქნეს ექსპერიმენტალურად დადასტურებული შედეგები, რათა შემდგომ განხორციელდეს კვლევის შედეგად მიღებული მეთოდების ეფექტურობის შეფასება.

მას შემდეგ რაც განხორციელდა NSW-ის კოდების ფორმირება, ისინი გაშვებულ იქნა სიმულაციის სატესტო სესიაში, რა დროსაც ხმაურის კოეფიციენტი შეადგენდა 2.1% ცხრ. 108. რაც შეეხება სიმულაციის შედეგებს,

	CPU Time/ms										ხმაური%
lxplus723.cern.ch	71150	69650	70650	69740	70340	69780	70570	69980	70120	71050	2.1

ცხრ.108 ოპტიმალური და არა ოპტიმალური CGS მეთოდების CPU time-ის ხმაურის კოეფიციენტი

CGS მეთოდებისათვის, ის წარმოდგენილია ცხრ. 109-ში. როგორც

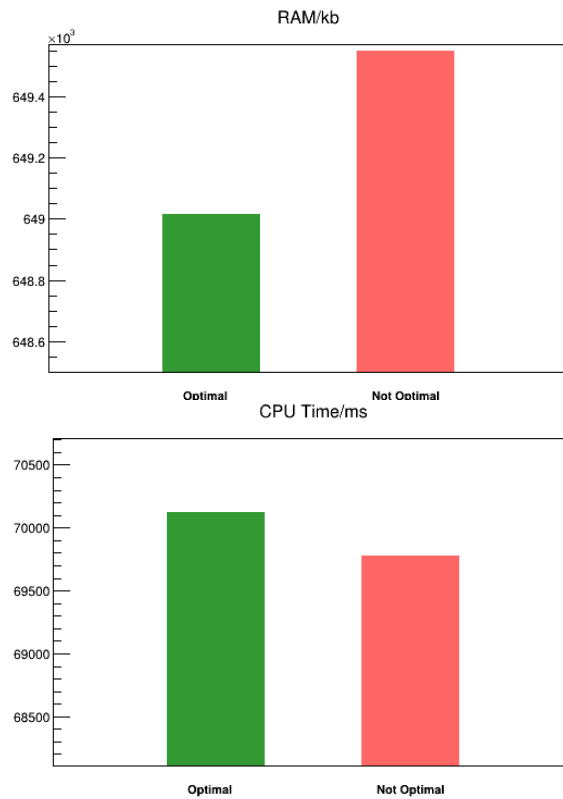
	განზომ.	ოპტიმალური	არა ოპტიმალური	სხვაობა		%	
				ოპტ./არა ოპტ.	ოპტ./არა ოპტ.		
NSW	CPU/ms	70120	69780	340	0	0.48	0
	RAM/kb	642014	649549	0	7535	0	1.16

ცხრ.109 ოპტიმალური და არა ოპტიმალური CGS მეთოდების სიმულაციის შედეგები

ცხრილიდან ჩანს CPU Time ის მიხედვით უპირატესობა არა ოპტიმალური ვარიანტის მხარესაა, თუმცა ეს შეიძლება იყოს ხმაური, რადგანაც კვლევის მომენტში მაურის კოეფიციენტი შეადგენდა 2.1%-ს. რაც შეეხება ოპერატიული მეხსიერების დატვირთვას ოპტიმალურ ვარიანტს აქვს 1.16%-

იანი უპირატესობა. სურ. 151-ზე წარმოდგენილია იგივე შედეგები გრაფიკულად. ზედა გრაფიკზე მოცემულია CPU Time-ის შედეგები, ხოლო ქვედა გრაფიკზე მოცემულია ინფორმაცია იგივე მაგალითის ოპერატიული მეხსიერების დატვირთვის შესახებ.

ასევე განხორციელდა პოლიგონ მეთოდების გამოკვლევა



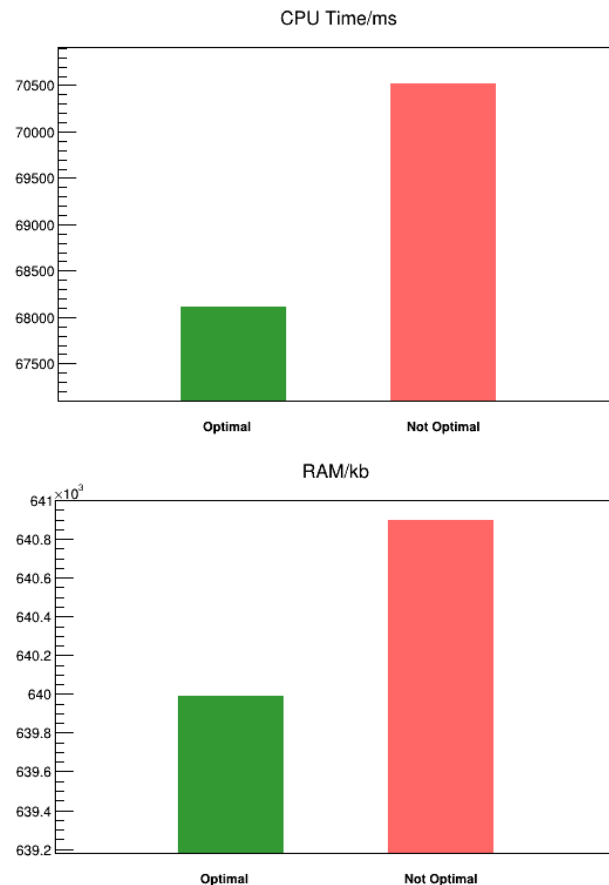
სურ. 151 ოპტიმალური და არა ოპტიმალური CGS მეთოდების სიმულაციის შედეგები წარმოდგენილი გრაფიკულად

სიმულაციაში. მიღებული შედეგები წარმოდგენილია ცხრ. 110-ში. როგორც ცხრილიდან ირკვევა CPU Time-ის და ასევე ოპერატიული მეხსიერების დატვირთვის მიხედვით უპირატესობა ოპტიმალური ვარიანტის მხარესაა.

	განზომ.	ოპტიმალური	არა ოპტიმალური	სხვაობა		%	
				ოპტ./არა ოპტ.	ოპტ./არა ოპტ.		
NSW	CPU/ms	68110	70520		2410		3.42
	RAM/kb	639991	640896		905		0.14

ცხრ.110 ოპტიმალური და არა ოპტიმალური Arbitrary vs Pyramid + Boolean სხვაობა CPU Time-ში შეადგენს 3.42%-ს, ხოლო ოპერატიული მეხსიერების დატვირთვის მიხედვით კი 0.14%-ს. შემდგომ ეტაპზე განხორციელდა გამოყენებული CGS და პოლიგონ მეთოდების კვლევის შედეგების

განზოგადება და დადგინდა, რომ NSW-თვის ჯამში ოპტიმალურ და არა ოპტიმალურ მეთოდებს შორის CPU Time-ის მიხედვით სხვაობა შეადგენს 1.06%, ხოლო ოპერატიული მეხსიერების დატვირთვის თვალსაზრისით კი სხვაობა არის 1.2%. 152-ზე წარმოდგენილია პოლიგონ მეთოდების სიმულაციის შედეგები გრაფიკულად. ზედა გრაფიკზე მოცემულია CPU Time-ის შედეგები, ხოლო ქვედა გრაფიკზე მოცემულია ინფორმაცია იგივე



სურ. 152 ოპტიმალური და არა ოპტიმალური Arbitrary vs Pyramid + Boolean მეთოდების სიმულაციის შედეგები წარმოდგენილი გრაფიკულად

მაგალითის ოპერატიული მეხსიერების დატვირთვის შესახებ.

5.6 აპრობაციის შედეგების ანალიზი

ამ ეტაპის ამოცანას წარმოადგენს შემდეგი, განხორციელდეს წინა პარაგრაფში მიღებული შედეგების განზოგადება ATLAS-ის დეტექტორის სრულ გეომეტრიულ აღწერებზე, რათა შეფასებულ იქნეს დისერტაციის კვლევის შედეგად მიღებული მეთოდების ეფექტურობა. პარაგრაფ #5.2-ში

განსაზღვრულ იქნა დეტექტორის სხვადასხვა სტრუქტურების გეომეტრიული აღწერების კომპლექსურობა. აღნიშნული შედეგები მოცემულია ცხრ. 107-ში, საიდანაც ირკვევა, რომ ATLAS-ის დეტექტორის მაგნიტური სისტემის პრიმიტივების რაოდენობა შეადგენს 398`924-ს. ხოლო ჩვენს მიერ გამოკვლეული NSW-ის პრიმიტივების რაოდენობაა 802`456. ასევე დადასტურებულია, რომ NSW-ისთვის მიღებული მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 1.06%-ს ხოლო RAM-ისთვის კი 1.2%. თუ ამ შედეგებს განვავრცობთ მაგნიტურ სისტემაზე პროპორციულად მივიღებთ, რომ ეფექტურობა CPU Time-ისთვის შეადგენს 0.53%-ს ხოლო RAM-ისთვის კი 0.6%. იგივე მსჯელობით, მეთოდის ეფექტურობა შიდა დეტექტორისთვის CPU Time მიხედვით შეადგენს 11.9%-ს ხოლო RAM-ისთვის 13.5%. რადგანაც მისი პრიმიტივების რაოდენობაა 9`024`069. რაც შეეხება კალორიმეტრების სისტემას, რადგანაც მისი პრიმიტივების რაოდენობაა 23`490`492, შესაბამისად მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 31%-ს ხოლო RAM-ისთვის კი 35.1%-ს. მიონური სისტემისთვის პრიმიტივების რაოდენობაა 13`380`286 და ამიტომ მეთოდის ეფექტურობა CPU Time-ისთვის იქნება 17.7% ხოლო RAM-ისთვის კი 20%. ბოლოს მექანიკური სტრუქტურები. მათი პრიმიტივების რაოდენობაა 740`028. შესაბამისად მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 1%-ს ხოლო RAM-ისთვის კი 1.1%-ს. აქედან გამომდინარე შეგვიძლია შევაფასოთ მეთოდის ეფექტურობა მთლიანად ATLAS-ის დეტექტორის გეომეტრიული აღწერებისათვის, რომელთა პრიმიტივების რაოდენობა

ATLAS-დეტექტორის სტრუქტურები	პრიმიტივების რაოდენობა/ ცალი	მეთოდის ეფექტურობა	
		CPU Time %	RAM %
მაგნიტური სისტემები	398`924	0.53	0.6
შიდა დეტექტორი	9`024`069	11.9	13.5
კალორიმეტრების სისტემა	23`490`492	31	35.1
მიონების სისტემა	13`380`286	17.7	20
მექანიკური სტრუქტურები	740`028	1	1.1
ჯამი	47`033`799	62.13	70.3

ცხრ.111 მეთოდის ეფექტურობა დეტექტორის სხვადასხვა სტრუქტურებისთვის

ჯამში შეადგენს 47` 033` 799-ს. აქედან გამომდინარე მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 62.1%-ს ხოლო RAM-ისთვის კი 70.3%-ს. ზემოთ განხილული შედეგები ცხრილის სახით მოცემულია ცხრ. 111-ში.

5.7 მე-5 ტაპის დასკვნები

1. შერჩეულ იქნა სატესტო მაგალითი პრობაციისთვის და ეს არის ATLAS-ის დეტექტორი ერთერთი ქვე დეტექტორი NSW
2. განხორციელდა NSW-ის ოპტიმალური და არა ოპტიმალური კოდების ფორმირება.
3. NSW-ზე ჩატარებული კვლევის მიხედვით მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 1.06%-ს ხოლო RAM-ისთვის კი 1.2%.
4. ATLAS-ის დეტექტორის სრული გეომეტრიული აღწერების გათვალისწინებით მეთოდის ეფექტურობა CPU Time-ისთვის შეადგენს 62,1%-ს ხოლო RAM-ისთვის კი 70,3%.

დისერტაციის დასკვნები:

1. გამოიყო ატლასის დეტექტორის 66 ტიპური გეომეტრიული აღწერა სატესტო სესიების ჩასატარებლად
2. დამუშავდა გეომეტრიული აღწერების დაპროგრამების მეთოდების კლასიფიკაცია, რომელშიც გაერთიანდა - ცილინდრული კლასისთვის 29, წახნაგოვანი კლასისთვის 135 და კომბინირებული კლასისთვის 252 ტიპური დაპროგრამების მეთოდი
3. გამოიყო ატლასის დეტექტორის გეომეტრიული აღწერების დაპროგრამების ტიპური კლასები და მეთოდები სატესტო სესიების ჩასატარებლად
4. სატესტო სესიების ჩასატარებლად შეიქმნა სიმულაციის წარმადობის შესწავლის საკვლევი ინფრასტრუქტურა
5. წახნაგოვანი ობიექტების დაპროგრამებისათვის დადგინდა, რომ პოლიგონ მეთოდები განაპირობებენ უკეთეს წარმადობას მყარი ტანის პრიმიტივების მეთოდებთან შედარებით. აღნიშნული სხვაობა იზრდება ობიექტების რაოდენობასთან ერთად და აღწევს CPU-თვის 8%-ს, ხოლო RAM-თვის 20%-ს
6. ცილინდრული ობიექტებისათვის დადგინდა, რომ დაპროგრამების მეთოდების ცვლილება განაპირობებს წარმადობის უმნიშვნელო ცვლილებას, რაც ობიექტების რაოდენობის ზრდასთან ერთად აღწევს 4%-ს
7. კომბინირებული ობიექტებისათვის დაპროგრამების პოლიგონ მეთოდებს უკეთესი წარმადობა აქვთ ვიდრე მყარი ტანის პრიმიტივების მეთოდებს და ეს სხვაობა პირდაპირპროპორციულად იზრდება ობიექტების რაოდენობასთან ერთად
8. გამოვლენილ იქნა Tube მეთოდის ეფექტურობა ცილინდრული ობიექტების დაპროგრამების ამოცანებისათვის
9. გამოვლენილ იქნა Arbitrary_Polygon მეთოდის ეფექტურობა წახნაგოვანი ობიექტების დაპროგრამების ამოცანებისათვის

10. გამოვლინდა უმნიშვნელო სხვაობა Cube და Pyramid მეთოდებს შორის როგორც წახნაგოვანი, ასევე კომბინირებული ობიექტების დაპროგრამების ამოცანებისათვის
11. განხორციელდა დაპროგრამების მეთოდების რანჟირება. შეიქმნა დაპროგრამების მეთოდების რეიტინგი წარმადობის თვალსაზრისით



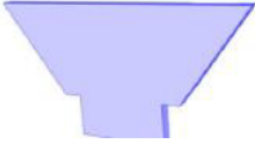

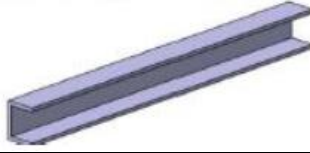
ბამოყენებული ლიტერატურა:

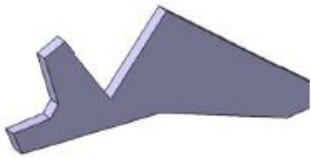


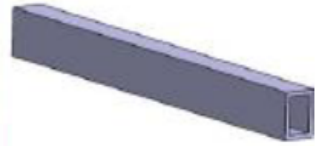

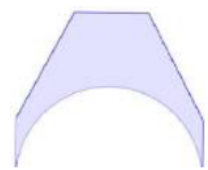


1. Witman Sarah. Ten things you might not know about particle accelerators. *Symmetry Magazine*. Fermi National Accelerator Laboratory. 2014.,,
2. <https://home.cern/science/accelerators/large-hadron-collider>,
უკანასკნელად იქნა გადამოწმებული - 12.04.2021
3. Evans L. The Large Hadron Collider. Switzerland: EPFL Press, 2009, 1 edition, 305 p.
4. ნიკო ცუცქირიძე. ATLAS-ის დეტექტორის გეომეტრიული აღწერის დამუშავება სიმულაციისა და რეკონსტრუქციის პროგრამული პაკეტებისათვის. *საქართველოს ტექნიკური უნივერსიტეტი*. 2016
5. ბესიკ კველია. ატლასის დეტექტორის ტოროიდული მაგნიტის შედარებითი ანალიზი სიმულაციის ამოცანებისათვის. *საქართველოს ტექნიკური უნივერსიტეტი*. 2018
6. არჩილ სურმავა. ATLAS დეტექტორის ერთიანი გეომეტრიული მოდელის ანალიზის და სინთეზის მეთოდების დამუშავება. *საქართველოს ტექნიკური უნივერსიტეტი*. 2017
7. ATLAS Collaboration. ATLAS Detector and Physics Performance, Volume I. CERN/LHCC 99-14. 1999
8. P. Brucman de Renstrom. Status of Beyond Standard Model Higgs searches at the LHC. ATL-PHYS-PROC-2015-206. 2015
9. ATLAS. Athena - The ATLAS Common Framework. 2001
10. ATLAS Computing Group. Technical Design Report. 2005
11. Leif Lonnblad. CLHEP—a project for designing a C++ class library for high energy physics. *Computer Physics Communications*. 84 (1–3): 307–316. 1994
12. S. Agostinelli, J. Allison, et al. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research A* 506 250–303. 2003
13. D. Costanzo, A. Dell’Acqua, A. Di Simone, M. Gallas, A. Nairz, et al. ATLAS detector simulation: status and outlook. 2005
14. P. Kroese, T. Brereton et al. Why the Monte Carlo Method is so important today. *WIREs Comput Stat*. 6 (6): 386–39. 2014
15. <https://geant4.web.cern.ch/applications>,
უკანასკნელად იქნა გადამოწმებული - 16.04.2021
16. B. Aubert^a, A. Bazan^a, A. Boucham^a et al. The BABAR detector. *Nuclear Instruments and Methods in Physics Research A* 479 (2002) 1–116. 2002, pp. 1-10
17. Nikolaos Savvas. The fast Simulation of the BarBar Calorimeter and a Measurement of the B-> K⁰s X Branching Ratio. *The University of Manchester*. 2001
18. K. Aamodt,¹ A. Abrahantes Quintana,² R. Achenbach,³. The ALICE experiment at the CERN LHC. *INSTITUTE OF PHYSICS PUBLISHING AND SISSA*. 2008


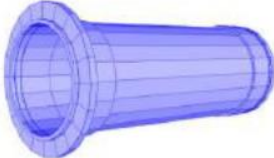

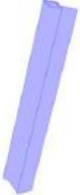



19. F. Carminati, A. Morsch. Simulation in ALICE. *proceeding of Computing in High Energy and Nuclear Physics, La Jolla, California*. 2003
20. Makoto Asai. Geant4 Applications in spase. *SLAC-PUB-12966*. 2008
21. Matt Dobbs, Jørgen Beck Hansen, et al. HepMC 2 a C++ Event Record for Monte Carlo Generators. *Computer Physics Communications*. 2000
22. C.Ay¹, A. Buckley², et al. Monte Carlo Generators in ATLAS software. *Journal of Physics: Conference Series 219 (2010) 032001*. 2010
23. Aad, G.; et al. The ATLAS simulation infrastructure. *European Physical Journal C DOI:10.1140/epjc/s10052-010-1429-9*. 2010
24. Flavia de Almeida Dias “The New ATLAS Fast Calorimeter Simulation”. *ICHEP2016*. 2016
25. Robert Harrington. ATLAS fast simulation and digitisation/reconstruction. *2nd LPCC Detector Simulation Workshop*. 2014
26. J. Apostolakis, A. Buckley, A. Dotti, Z. Marshall. Final Report of the ATLAS Detector Simulation. *CERN-LCGAPP-2010-01*. 2010
27. A. Sharmazanashvili, N. Tsutskiridze. Simulation loop between cad systems Geant4, and GeoModel: Implementation and results. *Physics of Particles and Nuclei Letters volume 13, pages724–727(2016)*. 2016
28. J. Boudreau, V. Tsulaia. The GeoModel toolkit for detector description. *Computing in High Energy and Nucl. Phys.* 2004
29. <https://gdml.web.cern.ch/GDML/> - უკანასკნელად იქნა გადამოწმებული - 27.04.2021
30. Laurent Chevalier, Andrea Dell’Acqua, Jochen Meyer. An XML generic detector description system and geometry editor for the ATLAS detector at the LHC. 2012
31. Riccardo Maria Bianchi et al. A new experiment-independent mechanism to persistify and serve the detector geometry of ATLAS. *J. Phys.: Conf. Ser. 898 072015*. 2017
32. Education, Communications and Outreach Group. CERN-Brochure-2017-002-Eng. 2017
33. Joe Boudreau. Geometry Kernel Classes. 2018
34. <https://geomodel.web.cern.ch/home/components/kernel/reference/#shapes> - უკანასკნელად იქნა გადამოწმებული - 5.05.2021
35. W. Pokorski, R. Chytracsek et al. Geometry description markup language and its application-specific bindings. CHEP06. 2006
36. Elmar Ritsch. ATLAS Simulation Framework. Fast Detector Simulation in High Energy Physics, DESY Zeuthen. 2013
37. არჩილ სურმავა, ბესიკ კველია, ნიკო ცუცქირიძე. ტოროიდული მაგნიტის გეომეტრიული ანალიზი ATLAS-ის მოდელირების ამოცანებისათვის. *AUTOMATED CONTROL SYSTEMS - No 2(22)*. 2016.
38. <https://www.fsp103-atlas.de/e17619/e17626/> - უკანასკნელად იქნა გადამოწმებული - 5.06.2021


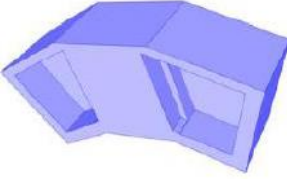



39. G. Lehmann, J. Bogaerts, M. Ciobotaru. et al. The DataFlow System of the ATLAS Trigger and DAQ. *CLEP, La Jolla, California*. 2003
40. E J Gallas, D Malon, R J Hawkings, S Albrand⁴ and E Torrence⁵. An integrated overview of metadata in ATLAS. ATL-GEN-PROC-2013-004. 2013
41. O'Luanaigh, Cian. "Proton beams are back in the LHC". *CERN*. 2015
42. V. Cindro. et al. The ATLAS Beam Conditions Monitor. *INSTITUTE OF PHYSICS PUBLISHING AND SISSA*. 2008
43. Monica Verducci on behalf of the ATLAS Collaboration. ATLAS Conditions Database Experience with the LCG COOL Conditions Database Project. *Journal of Physics: Conference Series 119 (2008) 042031*. 2008
44. A. Sharmazanashvili et al. A Tool for Conditions Tag Management in ATLAS. *ATL-SOFT-PROC-2013-028*. 2013
45. <http://collider.physics.ox.ac.uk/detecting.html> - უკანასკნელად იქნა გადამოწმებული - 7.06.2021
46. Bernd Stelzer. The New Small Wheel Upgrade Project of the ATLAS Experiment. *Nuclear and Particle Physics Proceedings 273–275 (2016) 1160–1165*. 2016
47. ATLAS Collaboration. New Small Wheel Technical Design Report. 2013
48. [https://www.cadcamge.ch/htmls/proflife/meeting/NSW-2017-vs-2019-Report-Part II.pdf](https://www.cadcamge.ch/htmls/proflife/meeting/NSW-2017-vs-2019-Report-Part-II.pdf) - უკანასკნელად იქნა გადამოწმებული - 7.06.2021
49. Benoit Lefebvre. Muon Spectrometer Phase-I Upgrade for the ATLAS Experiment: the New Small Wheel Project. *CIPANP2018-Lefebvre*. 2018

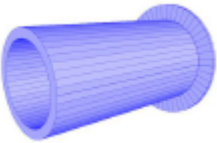
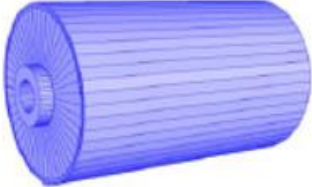
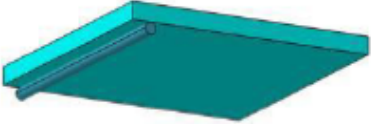
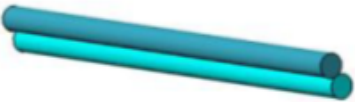
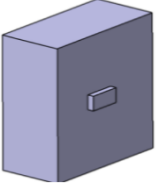

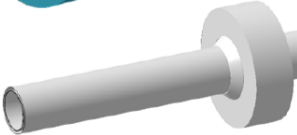
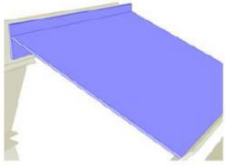
დანართი 1

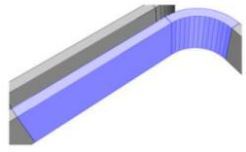
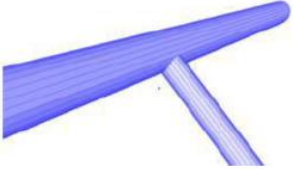

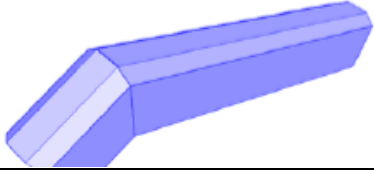
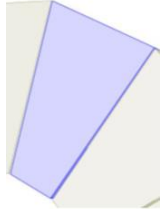
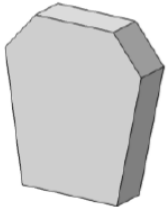
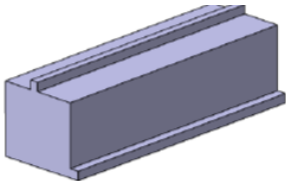
##	დასახელება		მეთოდები
#01		01	Pyramid Pyramid Subtraction Subtraction Pyramid Subtraction Subtraction
		02	Arbitrary
			Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid merged
		03-05	Cube/ symmetric/pyramid Arbitrary Arbitrary Subtraction Subtraction
#02		06-53	Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid/ arbitrary Cube/ symmetric/ pyramid/ arbitrary Subtraction Subtraction
		54	Arbitrary
#03		55	Symmetric
		56-57	Cube/ Pyramid Arbitrary Subtraction
		58	Arbitrary
#04		59-64	Cube/ symmetric/ pyramid Arbitrary/ pyramid Subtraction
		65	Arbitrary
#05		66-77	Cube/ symmetric/ pyramid Cube/ symmetric/ pyramid/ Arbitrary Subtraction
		78	Arbitrary
#06		79-90	Cube/ Pyramid/ Symmetric Cube/ Pyramid/ Symmetric/ Arbitrary

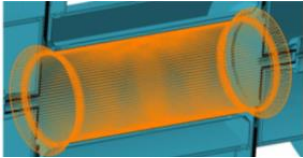
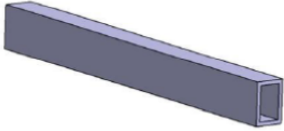
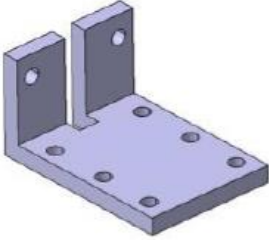
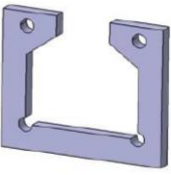
			Subtraction
		91	Arbitrary
#07		92-103	Cube/ Pyramid/ Symmetric Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction
		104	Arbitrary
#08		105-168	Cube/ Pyramid/ Symmetric/ Arbitrary Cube/ Pyramid/ Symmetric/ Arbitrary Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Subtraction Subtraction
#09		169-184	Cube/ Pyramid/ Symmetric/ Arbitrary Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction
#10		185-196	Symmetric/ Arbitrary Tube/ Cylinder/ Chain Subtraction Cube/ Pyramid Subtraction
#11		197-202	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction
#12		203-211	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Subtraction Subtraction
#13		212-229	Symmetric/ Arbitrary Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction

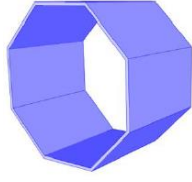
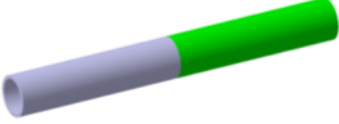
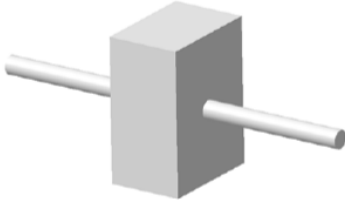
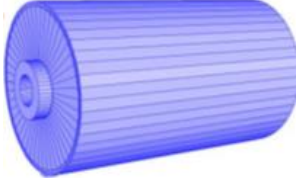
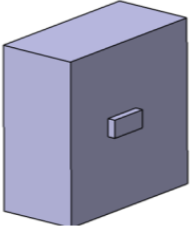
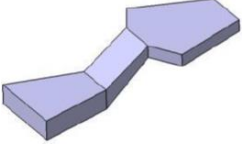
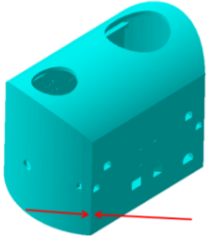
#14		230-238	Arbitrary Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
		239-247	Symmetric Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
#15		248	Cylinder
		249	Tube Tube Tube Tube
#16		250-252	Arbitrary Tube/ Cylinder/ Chain Subtraction
		253-261	Pyramid/ Cube/ Symmetric Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Arbitrary Subtraction Subtraction
#17		262-309	Cube/ Pyramid/ Symmetric/ Arbitrary Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#18		310-317	Tube/ Cylinder Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction
#19		318-445	Arbitrary/ Double Symmetric Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Subtraction
#20		446-448	Arbitrary Tube/ Cylinder/ Chain Subtraction
		449-454	Double Symmetric Tube/ Cylinder/ Chain Subtraction

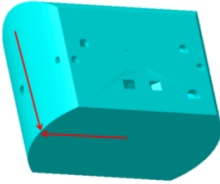

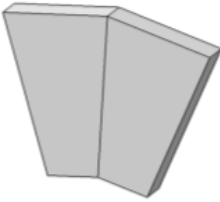
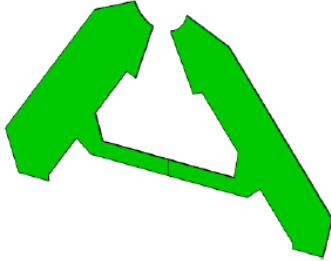

#21		455-478	Arbitrary/ Symmetric/ Arbitrary/ Symmetric/ Double Symmetric Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#22		479-484	Arbitrary/ Symmetric Arbitrary Arbitrary Subtraction Subtraction Subtraction Subtraction Subtraction Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction
#23		485-490	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction Subtraction
		491-514	Cube/ Pyramid Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#24		515-517	Arbitrary Tube/ Cylinder/ Chain Subtraction Subtraction
		518-553	Cube/ Pyramid/ Symmetric Cube/ Pyramid/ Symmetric/ Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#25		554-557	Arbitrary/ Cube/ Pyramid/ Symmetric Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction
#26		558	Tube Tube merged
		559	Cylinder

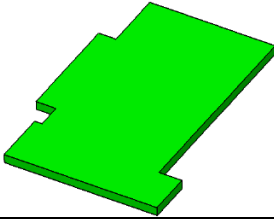
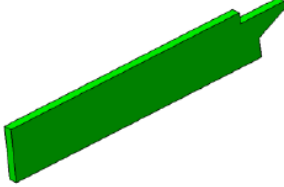
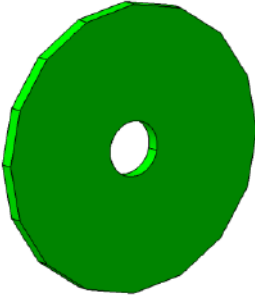
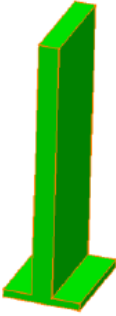

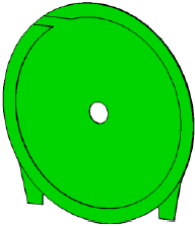
		560	Chain Chain merged
#27		561	Tube Tube Tube merged
		562	Cylinder
#28		563-574	Arbitrary/ Cube/ Pyramid/ Symmetric Tube/ Cylinder/ Chain combined
#29		575-583	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined
#30		584-599	Arbitrary/ Cube/ Pyramid/ Symmetric Arbitrary/ Cube/ Pyramid/ Symmetric combined
#31		600-623	Tube/ Cylinder Arbitrary/ Cube/ Pyramid/ Symmetric Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction combined
#32		624-627	Tube/ Cylinder Cylinder Tube/ Cylinder combined
#33		628-635	Symmetric/ Arbitrary Arbitrary/ Cube/ Pyramid/ Symmetric combined
		636-667	Cube/ Pyramid Arbitrary/ Cube/ Pyramid/ Symmetric Subtraction Arbitrary/ Cube/ Pyramid/ Symmetric combined

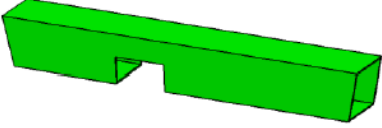
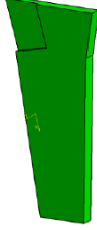
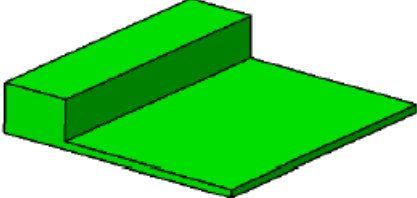
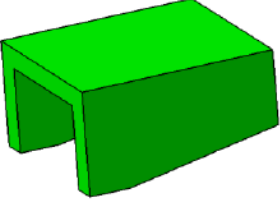
#34		668-675	Arbitrary/ Cube/ Pyramid/ Symmetric Tube/ Cylinder/ Chain combined
#35		676-679	Tube/ Cylinder Tube/ Cylinder Subtraction combined
#36		680-687	Tube/ Cylinder Cube/ Arbitrary/ Pyramid/ Symmetric subtraction combined
#37		688-703	Arbitrary/ Double Symmetric Arbitrary/ Double Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction combined
#38		704-705	Arbitrary/ Symmetric
		706-713	Cube/ Pyramid Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction
#39		714	Arbitrary/ Symmetric
		715-722	Cube/ Pyramid Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction
#40		723-731	Cube/ Pyramid/ Symmetric Cube/ Pyramid/ Symmetric Subtraction Subtraction Subtraction
		732	Arbitrary
#41		733	Cylinder

		734	Cylinder Tube Cylinder merged
#42		735-750	Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction
#43		751-894	Cube/ / Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
		895-906	Arbitrary Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
#44		907-958	Cube/ Pyramid/ Symmetric Cube/ Pyramid/ Symmetric Subtraction Arbitrary / Symmetric Subtraction Tube/Cylinder/Chain Subtraction Subtraction Subtraction Subtraction
		959-961	Arbitrary Tube/Cylinder/Chain Subtraction Subtraction Subtraction
#45		962-964	Cube/ Pyramid/ Symmetric Arbitrary Arbitrary Subtraction Subtraction

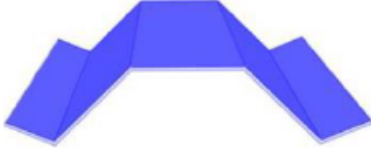

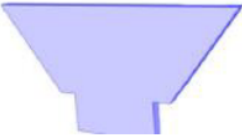


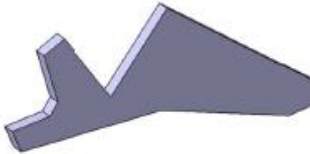

		965	Arbitrary Arbitrary Subtraction
#46		966-974	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined
#47		975-1010	Cube/ Arbitrary/ Pyramid/ Symmetric Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Subtraction
#48		1011-1037	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined
#49		1038-1053	Arbitrary/ Cube/ Pyramid/ Symmetric Arbitrary/ Cube/ Pyramid/ Symmetric combined
#50		1054-1057	Arbitrary/ Symmetric Arbitrary Arbitrary/ Symmetric combined
#51		1059-1069	Tube/Cylinder/Chain Cube/ Arbitrary/ Pyramid/ Symmetric merged


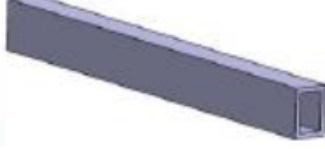





#52		1070-1081	Tube/Cylinder/Chain Cube/ Arbitrary/ Pyramid/ Symmetric merged
#53		1082-1093	Tube/ Cylinder/ Chain Arbitrary/ Cube/ Pyramid/ Symmetric combined
#54		1094	arbitrary/ Symmetric combined
		1095-1102	Cube/ Pyramid Cube/ Arbitrary/ Pyramid/ Symmetric combined
#55		1103-1108	Arbitrary/ double Symmetric Tube/ Cylinder/ Chain Subtraction
#56		1109-1156	Cube/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction
		1157	Arbitrary
#57		1158-1925	Cube/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction Subtraction Subtraction

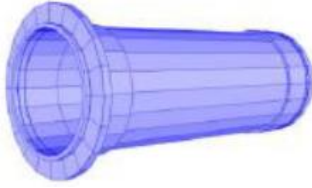





		1926	Arbitrary
#58		1927-1938	Cube/ Pyramid/ Symmetric Arbitrary Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction
		1939	Arbitrary
#59		1940-1945	Arbitrary/ double symmetric Tube/Cylinder/Chain Subtraction
#60		1946-1948	Cube/ Pyramid/ Symmetric Arbitrary Subtraction
		1949-1960	Cube/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction
		1961	Arbitrary
#61		1962	Symmetric/ Arbitrary
#62		1963-2070	Tube/Cylinder/Chain Tube/Cylinder/Chain Subtraction Tube/Cylinder/Chain Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Arbitrary

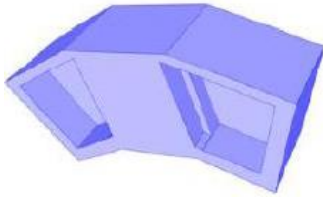
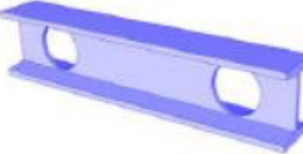


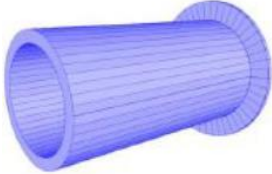
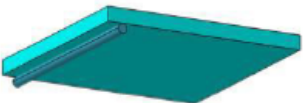
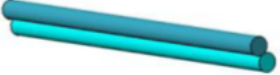
#63		2071-2526	Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction Subtraction Subtraction
#64		2527-2530	Arbitrary Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction
		2531-2533	Cube/ Pyramid/ Symmetric Arbitrary Arbitrary Subtraction Subtraction
#65		2534-2545	Cube/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction
		2546	Arbitrary
#66		2547-2709	Cube/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction Subtraction
		2710-2710	Arbitrary Cube/ Arbitrary/ Pyramid/ Symmetric Subtraction

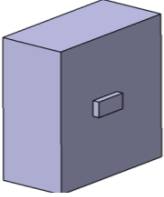

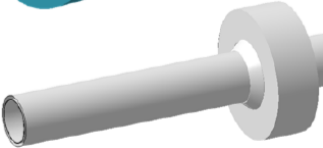
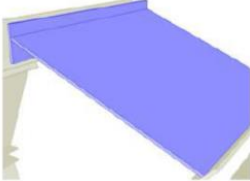
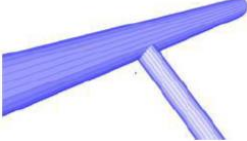

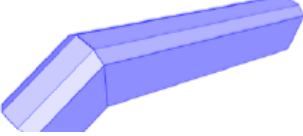


დანართი 2

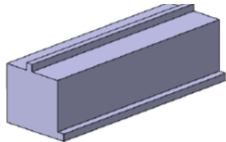
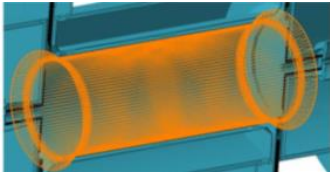
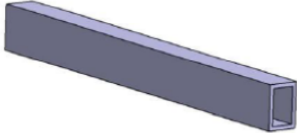
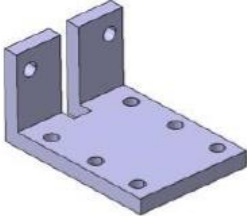
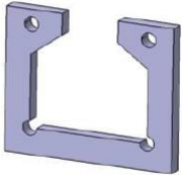
#	სურათი	მეტოდეები	მეთოდები
#01		01	Pyramid Pyramid Subtraction Subtraction Pyramid Subtraction
02		Arbitrary	
03-04		Cube/ pyramid Arbitrary Arbitrary Subtraction Subtraction	
#02		05-12	Cube/ pyramid Cube/ pyramid Cube/ pyramid Subtraction Subtraction
13		Arbitrary	
#03		14	Symmetric
15-16		Cube/ Pyramid Arbitrary Subtraction	
17		Arbitrary	
#04		18-19	Cube/ pyramid Arbitrary Subtraction
20		Arbitrary	
#05		21-24	Cube/ pyramid Cube/ pyramid Subtraction
25		Arbitrary	
#06		26-27	Cube/ Pyramid Arbitrary Subtraction Arbitrary Subtraction
28		Arbitrary	
#07		29-32	Cube/ Pyramid Cube/ Pyramid Subtraction
33		Arbitrary	

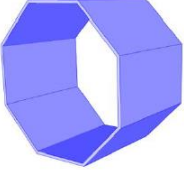
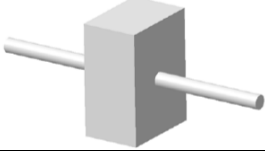
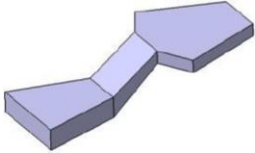
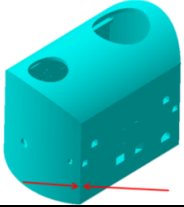
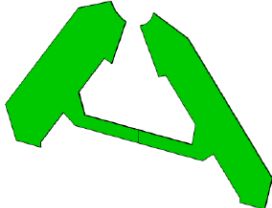

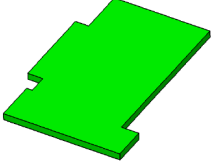
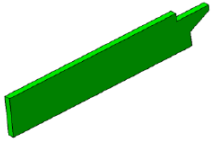
#08		34-41	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction
#09		42-45	Cube/ Pyramid Cube/ Pyramid Subtraction
#10		46-57	Symmetric/ Arbitrary Tube/ Cylinder/ Chain Subtraction Cube/ Pyramid Subtraction
#11		58-63	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction
#12		64-67	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Subtraction Subtraction
#13		68-85	Symmetric/ Arbitrary Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
#14		86-94	Arbitrary Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
		95-103	Symmetric Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
#15		104	Cylinder/Chain

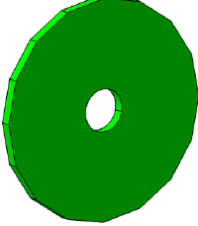
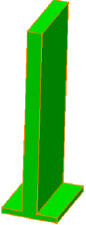

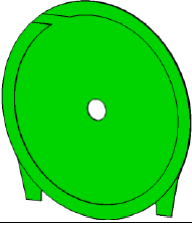
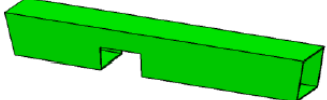
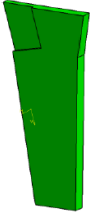
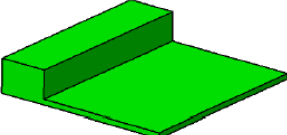
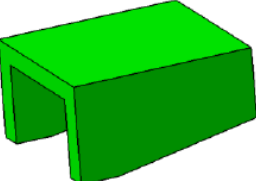
		105	Tube Tube Tube Tube Tube
#16		106-108	Arbitrary Tube/ Cylinder/ Chain Subtraction
		109-114	Pyramid/ Cube Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Arbitrary Subtraction Subtraction
#17		115-118	Cube/ Pyramid Cube/ Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#18		119-122	Tube/ Cylinder Cube/ Pyramid Subtraction
#19		123-138	Arbitrary/ Double Symmetric Cube/ Pyramid Subtraction Cube/ Pyramid Subtraction Cube/ Pyramid Subtraction Subtraction
#20		139-141	Arbitrary Tube/ Cylinder/ Chain Subtraction
		142-144	Double Symmetric Tube/ Cylinder/ Chain Subtraction
#21		145-162	Arbitrary/ Symmetric Arbitrary/ Symmetric/ Double Symmetric Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction

#22		163-168	Arbitrary/ Symmetric Arbitrary Arbitrary Subtraction Subtraction Subtraction Subtraction Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction
#23		169-174	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction Subtraction
		175-186	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#24		187-189	Arbitrary Tube/ Cylinder/ Chain Subtraction Subtraction
		190-201	Cube/ Pyramid Cube/ Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#25		202-204	Arbitrary/ Cube/ Pyramid Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction
#26		205	Tube Tube merged
		206	Cylinder/ Chain
#27		207-212	Cube/ Pyramid Tube/ Cylinder/ Chain combined
#28		213-221	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined

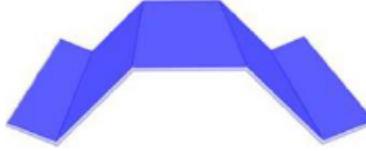







#29		222-225	Cube/ Pyramid Cube/ Pyramid combined
#30		226-233	Tube/ Cylinder Cube/ Pyramid Subtraction Tube/ Cylinder Subtraction Subtraction combined
#31		234-237	Tube/ Cylinder Cylinder Tube/ Cylinder combined
#32		238-241	Symmetric/ Arbitrary Cube/ Pyramid combined
		242-249	Cube/ Pyramid Cube/ Pyramid Subtraction Cube/ Pyramid combined
#33		250-253	Tube/ Cylinder Tube/ Cylinder Tube/ Cylinder Subtraction combined
#34		254-257	Tube/ Cylinder Cube/ Pyramid combined
#35		258-265	Arbitrary/ Double Symmetric Arbitrary/ Double Symmetric Cube/ Pyramid Subtraction Subtraction combined
#36		266-267	Arbitrary/ Symmetric
		268-271	Cube/ Pyramid Cube/ Pyramid Subtraction
#37		272-273	Arbitrary/ Symmetric
		274-277	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction



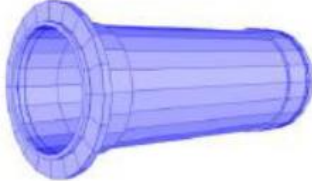




#38		278-281	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction
		282	Arbitrary
#39		283	Cylinder
		284	Cylinder Tube Cylinder merged
#40		285-288	Cube/ Pyramid Cube/ Pyramid Subtraction
#41		289-312	Cube/ Pyramid Cube/ Pyramid Subtraction Cube/ Pyramid/ Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
		312-318	Arbitrary Cube/ Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
#42		319-342	Cube/ Pyramid Cube/ Pyramid Subtraction Arbitrary/ Symmetric Subtraction Tube/Cylinder/Chain Subtraction Subtraction Subtraction Subtraction
		343-345	Arbitrary Tube/Cylinder/Chain Subtraction Subtraction Subtraction
#43		346-347	Cube/ Pyramid Arbitrary Arbitrary Subtraction Subtraction



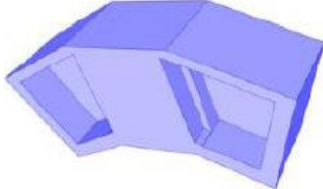
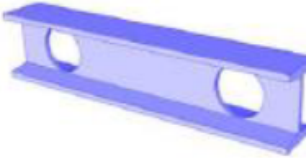

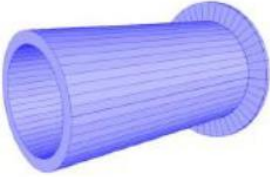
		348	Arbitrary Arbitrary Subtraction
#44		349-366	Cube/ Pyramid Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Subtraction
#45		367-370	Arbitrary/ Symmetric Arbitrary Arbitrary/ Symmetric combined
#46		371-374	Tube/ Cylinder Cube/ Pyramid merged
#47		380	Arbitrary/ double Symmetric Tube/ Cylinder/ Chain Subtraction
#48		381-388	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction
		389	Arbitrary
#49		390-421	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction Subtraction
		422	Arbitrary
#50		423-426	Cube/ Pyramid Arbitrary Cube/ Pyramid Subtraction Subtraction
		427	Arbitrary

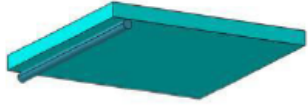
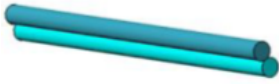
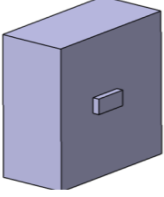

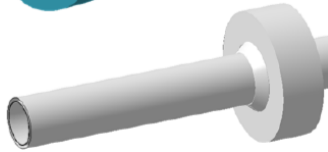
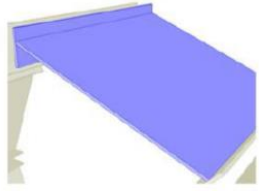
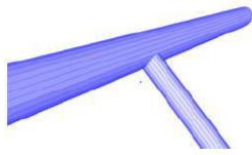
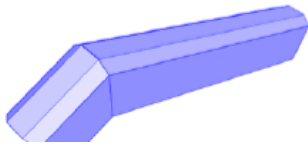
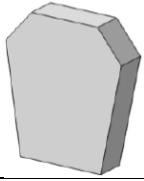
#51		428-433	Arbitrary/ double symmetric Tube/Cylinder/Chain Subtraction
#52		434-435	Cube/ Pyramid Arbitrary Subtraction
		436-439	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction
		440	Arbitrary
#53		441-442	Symmetric/ Arbitrary
#54		443-496	Tube/Cylinder/Chain Tube/Cylinder/Chain Subtraction Tube/Cylinder/Chain Cube/ Pyramid Subtraction Arbitrary
#55		497-512	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction Subtraction
#56		513-514	Arbitrary Cube/ Pyramid Subtraction
		515-516	Cube/ Pyramid Arbitrary Arbitrary Subtraction Subtraction
#57		517-520	Cube/ Pyramid Cube/ Pyramid Subtraction
		521	Arbitrary
#58		522-529	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction
		530-531	Arbitrary Cube/ Pyramid Subtraction

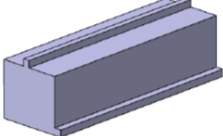
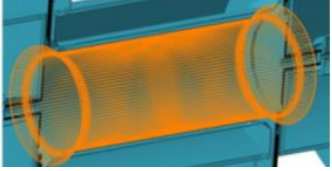
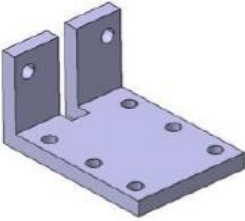
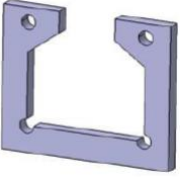
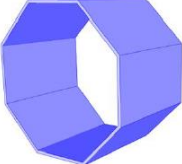
დანართი 3

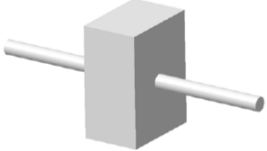
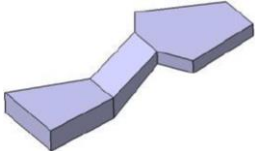
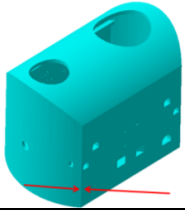
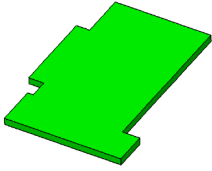
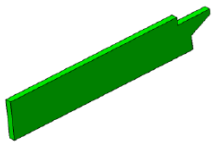
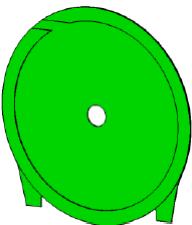
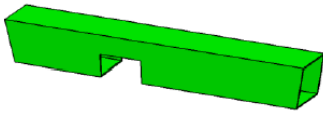
#	სურათი	მეტოდეები	მეთოდები
#01		01	Pyramid Pyramid Subtraction Subtraction Pyramid Subtraction
		02	Arbitrary
		03-04	Cube/ pyramid Arbitrary Arbitrary Subtraction Subtraction
#02		05-12	Cube/ pyramid Cube/ pyramid Cube/ pyramid Subtraction Subtraction
#03		13	Symmetric/ Arbitrary
		14-15	Cube/ Pyramid Arbitrary Subtraction
#04		16-19	Cube/ pyramid Cube/ pyramid Subtraction
#05		20-27	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction Subtraction
#6		28-39	Symmetric/ Arbitrary Tube/ Cylinder/ Chain Subtraction Subtraction Cube/ Pyramid Subtraction
#7		40-45	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction
#8		46-49	Tube/ Cylinder Tube/ Cylinder Subtraction Subtraction

#9		50-67	Symmetric/ Arbitrary Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
#10		68-76	Arbitrary Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
		77-85	Symmetric Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Tube/ Cylinder/ Chain Subtraction
#11		86	Cylinder
		87	Tube Tube Tube Tube Tube
#12		88-93	Pyramid/ Cube Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction Arbitrary Subtraction Subtraction
#13		94-105	Cube/ Pyramid Cube/ Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#14		106-109	Tube/ Cylinder Cube/ Pyramid Subtraction
#15		110-125	Arbitrary/ Double Symmetric Arbitrary/ Double Symmetric Subtraction Cube/ Pyramid Subtraction Cube/ Pyramid Subtraction Subtraction

#16		126-128	Double Symmetric Tube/ Cylinder/ Chain Subtraction
#17		129-146	Arbitrary/ Symmetric Arbitrary/ Symmetric/ Double Symmetric Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#18		147-152	Arbitrary/ Symmetric Arbitrary Arbitrary Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Arbitrary Subtraction Tube/ Cylinder/ Chain Subtraction
#19		153-158	Arbitrary/ Symmetric Tube/ Cylinder/ Chain Subtraction Subtraction
		159-170	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction
#20		171-173	Arbitrary/ Cube/ Pyramid Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction Arbitrary Subtraction
#21		174	Tube Tube merged

#22		175-180	Cube/ Pyramid Tube/ Cylinder/ Chain combined
#23		181-189	Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain combined
#24		190-193	Cube/ Pyramid Cube/ Pyramid combined
#25		194-201	Tube/ Cylinder Cube/ Pyramid Subtraction Tube/ Cylinder Subtraction Subtraction combined
#26		202-205	Tube/ Cylinder Cylinder Tube/ Cylinder combined
#27		206-209	Symmetric/ Arbitrary Cube/ Pyramid combined
		210-217	Cube/ Pyramid Cube/ Pyramid Subtraction Cube/ Pyramid combined
#28		218-221	Tube/ Cylinder Tube/ Cylinder Subtraction combined
#29		222-229	Arbitrary/ Double Symmetric Arbitrary/ Double Symmetric Cube/ Pyramid Subtraction Subtraction combined
#30			
		230-233	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction

#31		224-237	Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction
#32		238	Cylinder Tube Cylinder merged
#33		239-250	Cube/ Pyramid Cube/ Pyramid Subtraction Cube/ Pyramid/ Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
		251-256	Arbitrary Cube/ Pyramid Subtraction Tube/ Cylinder/ Chain Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction Subtraction
#34		257-280	Cube/ Pyramid Cube/ Pyramid Subtraction Arbitrary/ Symmetric Subtraction Tube/Cylinder/Chain Subtraction Subtraction Subtraction Subtraction
		281-283	Arbitrary Tube/Cylinder/Chain Subtraction Subtraction Subtraction Subtraction
#35		284	Arbitrary Arbitrary Subtraction

#36		285-302	Cube/ Pyramid Tube/ Cylinder/ Chain Tube/ Cylinder/ Chain Subtraction
#37		303-306	Arbitrary/ Symmetric Arbitrary Arbitrary/ Symmetric combined
#38		307-310	Tube/ Cylinder Cube/ Pyramid merged
#39		311-342	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction Subtraction
#40		343-346	Cube/ Pyramid Arbitrary Cube/ Pyramid Subtraction Subtraction
#41		347-400	Tube/Cylinder/Chain Tube/Cylinder/Chain Subtraction Tube/Cylinder/Chain Cube/ Pyramid Subtraction Arbitrary
#42		401-416	Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Cube/ Pyramid Subtraction Subtraction Subtraction Subtraction