

# სივრცითი ელემენტის შემოტანა

უჯრედის საშუალებით სიმსივნეს ზრდაზე იმუნური პასუხის აღმწერ  
ჩვეულებრივ დიფერენციალურ განტოლებებიან მოდელში

4 ივლისი 2014

თბილისი

ავტორები:

გიორგი გაჩეჩლაძე, გიორგი გაბუნია,  
ქეთი გომიაშვილი, ქეთი კახიძე

ხელმძღვანელები:

პროფ. რამაზ ბოჭორიშვილი,  
პროფ. თინათინ დავითაშვილი

საკვანძო სიტყვები: მათემატიკური მოდელირება, ჩვეულებრივი დიფერენციალური  
განტოლებები, კერძოწარმოებულის დიფერენციალურ განტოლებები, დიფუზია,  
იმუნური პასუხი, სიმსივნე, პარალელური გამოთვლები, რიცხვითი მეთოდები



## სარჩევი

|                                         |    |
|-----------------------------------------|----|
| ანოტაცია.....                           | 3  |
| მოდელის ჩამოყალიბება.....               | 4  |
| ჩვენი მოდიფიცირებული მოდელი.....        | 6  |
| რეაქციულ-დიფუზური სისტემა .....         | 6  |
| გლობალური და ლოკალური კომპონენტები..... | 7  |
| დიფუზიის კოეფიციენტი.....               | 9  |
| სხვაობიანი რიცხვითი სქემა.....          | 9  |
| პარალელური გამოთვლები .....             | 13 |
| სიმულაციების შედეგები.....              | 17 |
| გამოყენებული ლიტერატურა.....            | 21 |

# ანოტაცია

პროექტის მიზანია შევისწავლოთ მათემატიკური მოდელები კერძოწარმოებულის დიფერენციალური განტოლებებით, მოვახდინოთ მათი მოდიფიკაცია და კომპიუტერზე იმპლემენტაცია რიცხვით მეთოდების საშუალებით. კვლევის სფეროდ ავირჩიეთ იმუნოლოგია, კერძოდ კი სიმსივნეს მოქმედება იმუნურ სისტემაზე. დღესდღეობით მრავლადაა ჩვეულებრივ დიფერენციალურ განტოლებებიანი მოდელები იმუნოლოგიაში, თუმცა ძნელად თუ მოიპოვება ეფექტური მოდელი რომელიც კერძო წარმოებულ განტოლებებს იყენებს, რაც ნიშნავს რომ სფერო ჯერ კიდევ ახალგაზრდაა და ნამდვილად მნიშვნელოვანი საგანია კვლევისთვის.

ჩვენი მიზანი იყო ჩვეულებრივ დიფ. განტოლებებიანი მოდელის კერძო წარმოებულის დიფ. განტოლებებიანად გადარდაქმნა, რათა აღგვეწერა პროცესი უფრო ზუსტად და ეფექტურად. ჩვენ განვიხილეთ რიგი სამეცნიერო ნაშრომებისა აღნიშნულ სფეროში და საბოლოოდ ავარჩიეთ დოქ. ფილისის მათემატიკური მოდელი რომელიც აღწერს იმუნურ პასუხს სიმსივნეს ზრდაზე უჯრედის შუამავლობით. [1] ერთ-ერთი უპირატესობა არჩეული მოდელისა იყო მისი ვალიდურობა, მოდელის სანდოობა დადასტურებული ექსპერიმენტების საფუძველზე. ჩვენი დაშვებით ფილისის მოდელი სამართლიანია და ვიწყებთ მოდიფიკაცია მის საფუძველზე.

ერთ-ერთი სუსტი მხარე აღნიშნული მოდელისა იყო ის რომ აღწერდა მთლიან პროცესს ერთ კვანძში, ორგანიზმის ერთ წერტილად წარმოდგენით, რომელიც საკმაოდ უხეში წარმოდგენაა მოცემული სიმულაციისა. ჩვენ ჩავთვალეთ რომ უჯრედების მოძრაობა შესაძლებელია აღიწეროს დიფუზიის განტოლებებით; ამის შემდეგ ჩვენ შემოვიღეთ სივრცითი ელემენტი რითიც გავშალეთ ფილისის 1-განზომილებიანი მოდელი მრავალ განზომილებაში. შემოვიღეთ შესასწავლი არე და მისი საზღვრები. ჩავატარეთ სიღრმისეული ანალიზი ყველა ფუნქციისა და დავყავით გლობალურ და ლოკალურად კომპონენტებად იმის მიხედვით თუ სად ხდება უჯრედებს შორის ურთიერთქმედება. მაგალითად CD8+ უჯრედების T-სიმსივნეს მიერ განადგურება ლოკალური, უშუალო ურთიერთქმედებაა, ხოლო იმუნური უჯრედების წარმოქმნა ძვლის ტვინში გლობალურ კომპონენტად გავიტანეთ სასაზღვრო პირობებში. საზღვარზე დაგროვებული უჯრედები შესასწავლ არეზე ასევე დიფუზიის საშუალებით შემოედინებიან. ასეთი მიდგომით ჩვენ გავაფართოვეთ ფილისის მოდელი და შევეცადეთ აღგვეწერა მისი მოქმედება უფრო ზუსტად კერძოწარმოებულის დიფერენციალური განტოლებებისა და დიფუზიის საშუალებით.

მოგვიანებით შევქმენით რიცხვითი სქემა და მოვახდინეთ მისი იმპლემენტაცია C++ პროგრამირების ენაზე. სიმულაციის სიდიდის გამო ასევე მოვახდინეთ პროგრამული კოდის გაპარალელება რამდენიმე განსხვავებული მეთოდით.

# მოდელის ჩამოყალიბება

წარმოგიდგენთ ახალ მათემატიკურ მოდელს, რომელიც დაფუძნებულია დოქ. ფილისის მოდელზე და შეისწავლის იმუნურ პასუხს სიმსივნეს ზრდაზე NK და CD8<sup>+</sup> T უჯრედების მონაწილეობით. ჩვენ დავუშვით, რომ ფილისის მოდელი სწორია და საკმაო სიზუსტით ასახავს პროცესს. იმისათვის, რომ ჩამოვყალიბოთ ჩვენი მოდელის განტოლებები, ვიყენებთ იმ დაშვებებს, რომლებსაც ემყარება ფილისის მოდელი.

- იმუნური რეაქციის არარსებობისას სიმსივნე იზრდება ლოჯისტიკურად.
- NK და CD8<sup>+</sup> T-უჯრედები ანადგურებენ სიმსივნურ უჯრედებს
- ორგანიზმში სიმსივნის არსებობაზე NK და CD8<sup>+</sup> T-უჯრედები რეაგირებენ გამრავლებით
- აქტიური NK უჯრედები ყოველთვის არიან ორგანიზმში
- სიმსივნისთვის სპეციფიკური CD8<sup>+</sup> T-უჯრედების წარმოქმნა ხდება ორგანიზმში სიმსივნის აღმოჩენისას.
- NK და CD8<sup>+</sup> T-უჯრედები სიმსივნესთან ურთიერთქმედების შემდეგ განიცდიან დეაქტივაციას.

ფილისის მოდელი შედგება სამი განტოლებისგან, რომლებიც აღწერენ სიმსივნურ-იმუნური რეაქციის მთავარი მოქმედი პოპულაციების რაოდენობის ცვლილებებს. დაშვებების და ექსპერიმენტალურად მიღებული შედეგების საფუძველზე მიღებული ფილისის მოდელს შემდეგი სახე აქვს:

$$\frac{dT}{dt} = aT(1 - bT) - cNT - D$$

$$\frac{dN}{dt} = \sigma - fN + \frac{gT^2}{h + T^2} N - pNT$$

$$\frac{dL}{dt} = -mL + \frac{jD^2}{k + D^2} L - qLT + rNT$$

$$D = d \frac{(L/T)^\lambda}{s + (L/T)^\lambda} T$$

სადაც  $T(t)$  აღნიშნავს სიმსივნის უჯრედების პოპულაციის  $t$  დროში,  $N(t)$  - NK უჯრედების ეფექტურობა  $t$  დროში, ხოლო  $L(t)$  - სიმსივნისთვის სპეციფიკურსიმსივნური უჯრედების ( $CD8^+ T$ ) პოპულაცია  $t$  დროში. მოდელში არსებული პარამეტრები მიღებულია ექსპერიმენტებზე დაყრდნობით.

პირველი განტოლება აღწერს სიმსივნის რაოდენობის ზრდას იმუნური რეაქციის პირობებში. განტოლების მარჯვენა მხარის პირველი წევრი  $aT(1-bT)$  აღწერს მის ნატურალურ ზრდას, მეორე წევრი გამოხატავს NK უჯრედების ზემოქმედებას სიმსივნურ უჯრედებზე, ვინაიდან ნატურალური მკვლელები უარყოფით ზაგავლენას ახდენენ სიმსივნის უჯრედებზე, ამიტომ ეს წევრი მინუს ნიშნითაა, ასევეა ამ განტოლების მარჯვენა მხარის ბოლო წევრი  $D$ , რომელიც გამოხატავს  $CD8^+ T$ -უჯრედების მომედებას სიმსივნეზე.

მეორე განტოლებაში განხილულია NK უჯრედების რაოდენობის ცვლილება, რომელიც დამოკიდებულია მათ ნატურალურ ზრდაზე ( $\sigma - fN$ ), ორგანიზმში სიმსივნის არსებობით გამოწვეულ NK უჯრედების გამრავლებაზე ( $\frac{gT^2}{h+T^2}N$ ) და სიმსივნურ უჯრედებთან ურთიერქმედების გამო განადგურებულ NK უჯრედების რაოდენობაზე ( $-pNT$ ).

მესამე განტოლებაში აღწერილია სიმსივნისთვის სპეციფიკური  $CD8^+ T$ -უჯრედების რაოდენობის ცვლილება. იგი დამოკიდებულია ამ უჯრედების ნატურალურ განადგურებაზე ( $-mL$ ), ორგანიზმში სიმსივნის არსებობის პირობებში წარმოქმნილი  $CD8^+ T$ -უჯრედების რაოდენობაზე ( $\frac{jD^2}{k+D^2}L$ ), სიმსივნესთან ურთიერთქმედების დროს განადგურებული უჯრედების რაოდენობაზე ( $-qLT$ ) და NK უჯრედების მიერ  $CD8^+ T$ -უჯრედების სტიმულაციაზე ( $-rNT$ ).

$a$ \_ სიმსივნის ზრდის სიხშირე

$b$ \_  $1/b$  სიმსივნის კვდომის  
მახასიათებელი

$c$ \_ NK უჯრედების მიერ სიმსივნის  
განადგურების მაჩვენებელი

$d$ \_  $CD8^+ T$  უჯრედების მიერ სიმსივნის  
განადგურების მაჩვენებელი

$\sigma$ \_ NK უჯრედების წარმოქმნის  
მუდმივი მაჩვენებელი

$\lambda$ \_  $CD8^+ T$  უჯრედების მიერ სიმსივნის  
განადგურების მაჩვენებლის ხარისხი

$f$ \_ NK უჯრედების სიკვდილიანობის  
სიხშირე

$g$ \_ სიმსივნის მიერ რეკრუტირებული NK უჯრედების  
მაჩვენებელი

$h$ \_ NK უჯრედების რეკრუტირების მრუდის დახრის  
კოეფიციენტი

$j$ \_  $CD8^+ T$  უჯრედების რეკრუტირების მაჩვენებელი

$k$ \_  $CD8^+ T$  უჯრედების რეკრუტირების მრუდის დახრის  
კოეფიციენტი

$m$ \_  $CD8^+ T$  უჯრედების სიკვდილიანობის მაჩვენებელი

$q$ \_ სიმსივნის უჯრედების მიერ  $CD8^+ T$  უჯრედების  
დეაქტივაციის მაჩვენებელი

$p$ \_ სიმსივნის უჯრედების მიერ NK უჯრედების  
დეაქტივაციის მაჩვენებელი

$s$ \_  $CD8^+ T$  უჯრედების ბრძოლისუნარიანობის  
კოეფიციენტი სიმსივნესთან

$r$ \_  $CD8^+ T$  უჯრედების სტიმულაციის სისწრაფის  
მაჩვენებელი

# ჩვენი მოდიფიცირებული მოდელი

მოდელში სივრცის დამატების მიზნით ვიყენებთ შემდეგ დაშვებებს:

- უჯრედების მოძრაობა ორგანიზმში ხდება დიფუზიით
- ფილისის მოდელი სწორად აღწერს სიმსივნურ-იმუნურ ურთიერთქმედებას.
- სიმსივნურ-იმუნურ ურთიერთქმედებაში შემავალი პროცესების განცალკევება გლობალურ და ლოკალურ პროცესებად მათი მდებარეობის მიხედვით.

როგორც ანოტაციაში აღვნიშნეთ, იმ მოდელებში, რომლებიც აგებულია ჩვეულებრივი დიფერენციალური განტოლებებით, სხეული მიჩნეულია ერთ წერტილად და ყველა პროცესი განხილულია ამ კონკრეტულ წერტილში. იმისათვის, რომ მოდელი უფრო რეალურად ასახავდეს მიმდინარე მოვლენებს, პროცესი განხილული უნდა იყოს სივრცეში.

მიჩნეულია, რომ უჯრედის მოძრაობა შესაძლოა საკმაო სიზუსტით აღიწეროს დიფუზიის დახმარებით. ჩვენ შემოგვაქვს დიფუზიის ელემენტი რომ აღწეროთ სიმსივნესა და იმუნური უჯრედების მოძრაობა. პარამეტრების შერჩევაზე ოდნავ მოგვიანებით განვაგრძობთ ახლა კი წარმოგიდგენთ რეაქციულ-დიფუზიურ სისტემას რომელზეც დავაფუძვნედ ჩვენი მოდელი:

## რეაქციულ-დიფუზიური სისტემა

სისტემა წარმოადგენს მათემატიკურ მოდელს, რომელიც აღწერს ერთი ან რამდენიმე ნივთიერების სივრცეში გავრცელების ცვლილებადობის კონცენტრაციას, რომელიც ორი პროცესის ზემოქმედების ქვეშ მიმდინარეობს: ლოკალური ქიმიური რეაქცია და დიფუზია.

რეაქციულ-დიფუზიურ სისტემას აქვს ნახევრად წრფივი პარაბოლური კერძო წარმოებულიანი დიფერენციალური განტოლების სახე.

$$\partial_t q = \underline{D} \nabla^2 q + R(q),$$

სადაც  $q(x, t)$  არის ცვლადი, რომელიც წარმოადგენს უჯრედების პოპულაცია და  $x \in C \subset \mathbb{R}^n$ ,  $C$  ჩვენ მიერ აღებული არეა,  $D$  დიფუზიის კოეფიციენტი,  $R$  არის გლუვი ფუნქცია  $R(q): \mathbb{R} \rightarrow \mathbb{R}$ , რომელიც ისეთ პროცესს აღწერს (მაგ. უჯრედების წარმოქმნა, სიკვდილი... )რაც ჩვენ მიერ აღებულ სივრცეში არ დიფუზირდება.

განვიხილოთ რეაქციულ-დიფუზური სისტემებზე აგებული მოდელები „Reaction-diffusion model for the growth of avascular tumor“<sup>[2]</sup> და „A Reaction-Diffusion Model of Cancer Invasion“<sup>[3]</sup>, მათზე დაყრდნობით შევქმენით ჩვენი მოდელი.

## გლობალური და ლოკალური კომპონენტები

ფორმალურად შემოგვაქვს არე, რომელშიც მოთავსებული იქნება სიმსივნური და იმუნური პოპულაცია და მათი ურთიერთქმედება განვიხილოთ ამ არესთან მიმართებაში, კერძოდ, დავყოთ მიმდინარე პროცესები ლოკალურ და გლობალურ პროცესებად იმის მიხედვით, თუ სად მიმდინარეობენ ისინი. სიმსივნის ნატურალური ზრდა მიმდინარეობს ჩვენს მიერ დაფიქსირებულ არეში, ამიტომ ის ლოკალური პროცესია, ასევეა ნატურალური მკვლეელი (NK) უჯრედებისა და  $CD8^+$  T-უჯრედების (L) მიერ სიმსივნის განადგურებაც. ჩვენი დაშვების თანახმად უჯრედების მოძრაობა ხდება დიფუზიით. ამ ყველაფერზე დაყრდნობით მოდელის პირველი განტოლება მიიღებს შემდეგ სახეს:

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( \xi_T(T) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( \xi_T(T) \frac{\partial T}{\partial y} \right) + aT(1-bT) - cNT - d \frac{(L/T)^2}{s + (L/T)^2} T$$

ხოლო არის საზღვარზე სიმსივნის უჯრედების რაოდენობა ნულის ტოლია.

$$T|_{\partial\Omega} = 0$$

რაც შეეხება ნატურალურ მკვლელ უჯრედებს, მათი ნატურალური ზრდა და სიმსივნის სტიმულირება მის გამრავლებაზე ხდება მთელ ორგანიზმში, ამიტომ უნდა აღიწეროს გლობალურ კომპონენტად. უჯრედების კვდომა ბუნებრივი მიზეზების გამო ხდება როგორც ჩვენს შესასწავლ არეზე ისევე მთელს ორგანიზმშიც ამიტომ კომპონენტი  $[-fN]$  წარმოადგენს როგორც გლობალურ ისევე ლოკალურ კომპონენტს. ხოლო NK უჯრედების კვდომა სიმსივნესთან ბრძოლის გამო აშკარა ლოკალური ურთიერთქმედებაა და კომპონენტი რჩება ლოკალურად. ასე და ამგვარად ლოკალური კომპონენტები განიხილება უშუალოდ შესასწავლ არეზე ხოლო გლობალურები სასაზღვრო პირობებად ჯდება. დიფუზია NK უჯრედების შემთხვევაში გამოიყენება როგორც არეში სამოძრაოდ ასევე მეტად მნიშვნელოვანი საზღვრებიდან დიფუზირებისთვის. ამ ყველაფრის გათვალისწინებთ განტოლება ნატურალური მკვლეელი უჯრედებისთვის მიიღებს შემდეგ სახეს:

$$\frac{\partial N}{\partial t} = \frac{\partial}{\partial x} \left( \xi_N(N) \frac{\partial N}{\partial x} \right) + \frac{\partial}{\partial y} \left( \xi_N(N) \frac{\partial N}{\partial y} \right) - pNT - fN$$

ხოლო საზღვრისთვის მივიღებთ შემდეგ განტოლებას:

$$\frac{\partial N}{\partial t} = \frac{1}{4} (\sigma - fN + \frac{gT^2}{h + T^2} N)$$

შემდეგი ნაბიჯია სიმსივნისთვის სპეციფიკური CD8<sup>+</sup> T-უჯრედების პოპულაციის განტოლების აგება. CD8<sup>+</sup> T-უჯრედების კვდომა მიმდინარეობს როგორც არეში, ასევე მის გარეთ, ორგანიზმში სიმსივნის აღმოჩენისას CD8<sup>+</sup> T-უჯრედების წარმოქმნა ხდება

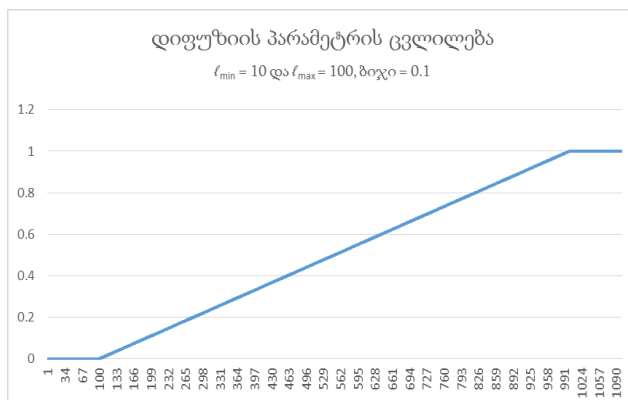
გლობალურად, ხოლო რაც შეეხება სიმსივნესთან ურთიერთქმედებისას CD8<sup>+</sup> T-უჯრედების განადგურებას და მათ სტიმულაციას NK უჯრედების მიერ, ისინი ლოკალურიპროცესებია რადგან მიმდინარეობს დაფიქსირებულ არეში. საბოლოოდ განტოლებები ასეთ სახეს მიიღებენ:

$$\frac{\partial L}{\partial t} = \frac{\partial}{\partial x} \left( \xi_L(L) \frac{\partial L}{\partial x} \right) + \frac{\partial}{\partial y} \left( \xi_L(L) \frac{\partial L}{\partial y} \right) - qLT + rNT - mL$$

ხოლო საზღვარზე: 
$$\frac{\partial L}{\partial t} = \frac{1}{4} (-mL + \frac{jD^2}{k + D^2} L) + rNT$$

# დიფუზიის კოეფიციენტი

დიფუზიის კოეფიციენტი განვიხილოთ სამივე პოპულაციისთვის ასეთი პრინციპით: უჯრედების მცირე რაოდენობისას დიფუზია არ ხდება, რადგან მათთვის საკმარია ის ადგილი, რომლებზეც განლაგებულები არიან, ხოლო მათი რაოდენობის გაზრდისას იზრდება დიფუზიის ხარისხიც. ამიტომ დიფუზიის კოეფიციენტი შეგვიძლია ჩავწეროთ შემდეგი სახით:



$$\xi_{\ell}(\ell) = \begin{cases} 0, & \ell \leq \ell_{\min} \\ \frac{\ell - \ell_{\min}}{\ell_{\max} - \ell_{\min}}, & \ell_{\min} \leq \ell \leq \ell_{\max} \\ 1, & \ell \geq \ell_{\max} \end{cases}$$

სადაც  $\ell = T, N, L$

დიფუზიის კოეფიციენტის შერჩევაზე ვეყრდნობით ფრიდმანის[4] სტატიას სიმსივნის მოდელის იერარქიაზე და მათ მათემატიკურ გამოწვევებზე.

## სხვაობიანი რიცხვითი სქემა

განტოლებების ამოსახსნელად და რიცხვითი სქემის შესადგენად ვიყენებთ ეილერის ცხად სქემას. შესასწავლი არე წარმოადგენს 1-ის ტოლი სიგრძის კვადრატულ სიბრტყეს სადაც  $0 \leq x \leq 1$  და  $0 \leq y \leq 1$ . სიბრტყის მაშტაბირება შესაძლებელია  $\ell_{\min}$  და  $\ell_{\max}$  პარამეტრების ცვლილებით იმ დაშვებით რომ 1 სმ<sup>3</sup>-ში ეტევა  $10^9$  სიმსივნის უჯრედი. პროცესს ვაკვირდებით T დროის მანძილზე  $\tau$  დროითი ბიჯით ხოლო ბადე დაყოფილია  $h$ -ის ტოლ ინტერვალებად. შესაბამისად რიცხვითი სქემა სიმსივნის, NK და CD8<sup>+</sup> T-უჯრედების ზრდაზე დასაკვირვებლად იღებს შემდეგ სახეს:

### სიმსივნეს უჯრედების პოპულაცია

$$T_{i,j}^{k+1} = T_{i,j}^k + \frac{\tau}{h^2} \left( \xi_T(T) \Big|_{i+1,j} (T_{i+1,j}^k - T_{i,j}^k) - \xi_T(T) \Big|_{i,j} (T_{i,j}^k - T_{i-1,j}^k) \right) +$$

$$+ \frac{\tau}{h^2} \left( \xi_T(T) \Big|_{i,j+1} (T_{i,j+1}^k - T_{i,j}^k) - \xi_T(T) \Big|_{i,j} (T_{i,j}^k - T_{i,j-1}^k) \right) + \pi T_{i,j}^k (1 - b T_{i,j}^k) -$$

$$- \pi N_{i,j}^k T_{i,j}^k - \pi d \frac{(L_{i,j}^k / T_{i,j}^k)}{s + (L_{i,j}^k / T_{i,j}^k)} T_{i,j}^k$$

საზღვარზე:  $T_{i,j}^k \Big|_{\partial\Omega} = 0$

### NK უჯრედების პოპულაცია:

$$N_{i,j}^{k+1} = N_{i,j}^k + \frac{\tau}{h^2} \left( \xi_N(N) \Big|_{i+1,j} (N_{i+1,j}^k - N_{i,j}^k) - \xi_N(N) \Big|_{i,j} (N_{i,j}^k - N_{i-1,j}^k) \right) +$$

$$\frac{\tau}{h^2} \left( \xi_N(N) \Big|_{i,j+1} (N_{i,j+1}^k - N_{i,j}^k) - \xi_N(N) \Big|_{i,j} (N_{i,j}^k - N_{i,j-1}^k) \right) + \pi N_{i,j}^k T_{i,j}^k - f N_{i,j}^k$$

საზღვარზე:  $N_{i,j}^{k+1} = N_{i,j}^k + \frac{\tau}{4n} \left( \sigma - f \sum N_{i,j}^k + \frac{g(\sum N_{i,j}^k / n^2)^2}{h + (\sum N_{i,j}^k / n^2)^2} (\sum N_{i,j}^k / n^2) \right)$

### CD8<sup>+</sup> T-უჯრედების პოპულაცია

$$L_{i,j}^{k+1} = L_{i,j}^k + \frac{\tau}{h^2} \left( \xi_L(L) \Big|_{i+1,j} (L_{i+1,j}^k - L_{i,j}^k) - \xi_L(L) \Big|_{i,j} (L_{i,j}^k - L_{i-1,j}^k) \right) +$$

$$+ \frac{\tau}{h^2} \left( \xi_L(L) \Big|_{i,j+1} (L_{i,j+1}^k - L_{i,j}^k) - \xi_L(L) \Big|_{i,j} (L_{i,j}^k - L_{i,j-1}^k) \right) + \pi L_{i,j}^k T_{i,j}^k + \pi N_{i,j}^k T_{i,j}^k - m L_{i,j}^k$$

ხოლო საზღვარზე:

$$L_{i,j}^{k+1} = L_{i,j}^k + \frac{\tau}{h^2} \left( -m L_{i,j}^k + \frac{j(D^k)^2}{h + (D^k)^2} L_{i,j}^k + r (\sum N_{i,j}^k / n^2) (\sum T_{i,j}^k / n^2) \right)$$

$$D = d \frac{(\sum L_{i,j}^k / \sum T_{i,j}^k)^\lambda}{s + (\sum L_{i,j}^k / \sum T_{i,j}^k)^\lambda} \frac{1}{n^2} \sum T_{i,j}^k$$

# კომპიუტერული იმპლემენტაცია

კომპიუტერული იმპლემენტაცია დაფუძნებულია ზევით აღწერილ რიცხვით სქემაზე. მოდელის მოდიფიცირებამ, რა თქმა უნდა, შედეგად პარამეტრების მკვეთრი ცვლილება მოგვცა. ზუსტი პარამეტრების დადგენის ერთადერთი გზა ბიოლოგიური ექსპერიმენტებია. სწორედ ამიტომ საწყის მიახლოებად ფილისის ODE მოდელის ვალიდური პარამეტრები ავიღეთ და დავაკვირდით მოდელის მუშაობას ამ პარამეტრების მნიშვნელობებიდან გადახრისას. ძირითადი კანონ-ზომიერებები, მაგალითად ის, რომ სიმსივნე ყველაზე მეტად მგრძნობიარეა CD8+ უჯრედების მიმართ, კვლავ ძალაშია რაც გვაძლევს საფუძველს იმისა რომ მოდელი სწორია. ქვემოთ განვიხილავთ პროგრამის მუშაობას დეტალურად ისევე როგორც მის იმპლემენტაციას სხვადასხვა პარალელური გამოთვლების სქემაზე.

## მოდელის იმპლემენტაცია

მოდელის იმპლემენტაციისას ვიყენებთ დინამიურ მასივებს, დამთვლელ ფუნქციებსა და დამხმარე ფუნქციებს. გვაქვს ექვსი მასივი, 2-2 კიბოს, NK და T უჯრედებისათვის. თითოეულ წერტილში რიცხვი განსაზღვრავს, თუ დაახლოებით რამდენი უჯრედია მის მიდამოში. ორ-ორი მასივი საჭიროა, რადგან ყოველ დროით შრეზე წინა შრის მონაცემებიდან გამომდინარე ვითვლით მნიშვნელობებს.

```
int main() {
    //ინიციალიზაცია
    //true = გააგრძელე წინა სიმულაცია
    Initialise(false);
    //პროგრამული პარამეტრების გამოტანა
    printParams(cout);
    //მთავარი გამოთვლის ფუნქცია
    Calculate();

    system("pause");
    return 0;
}
```

თავდაპირველად ხდება ინიციალიზაცია, რომლის დროსაც ტექსტური ფაილიდან შეგვყავს საწყისი და საბოლოო წერტილები და კვანძითი წერტილების რაოდენობა (მასივის განზომილებები), შემდგომ კი სიმულაციის დღეების რაოდენობა. ამის შემდეგ ვითვლით რიცხვით ღერძზე ინტერვალებს და მასზე დამოკიდებულ დროით ბიჯს- $\tau$ -ს და  $\tau$ -ზე დამოკიდებულ დროითი ბიჯების რაოდენობას. ამის შემდეგ შეგვყავს განტოლების პარამეტრები,

რომლებიც ცვალებადია და სხვადასხვა მნიშვნელობებზე სხვადასხვა შედეგებს იძლევა. თითოეულ მაგანს წინასწარ ვამრავლებთ  $\tau$ -ზე, რადგან ეს პარამეტრები 1 დღეზეა

გათვლილი, ჩვენ კი ყოველ  $\tau$  დროით ბიჯზე ვითვლით განტოლებას და ყოველ იტერაციაზე გადამრავლებისას ცდომილების დაგროვების შანსი უფრო დიდია.

ამის შემდეგ ვქმნით მასივებს და ვანაწილებთ მნიშვნელობებს: NK უჯრედების რაოდენობის ნახევარს თანაბრად ვანაწილებთ საზღვრებზე, მეორე ნახევარს - შიდა უჯრედებში. კიბოს უჯრედებისთვის ვიღებთ რაიმე, საზღვრისაგან საკმაოდ მოშორებულ ადგილს, სადაც მათ შეძლებისდაგვარად მკვრივად ვანაწილებთ. T-უჯრედების რაოდენობა კი თავდაპირველად 0-ის ტოლია.

პარამეტრების შეყვანისა და საწყისისი მონაცემების ინიციალიზაციის შემდეგ ვიძახებთ პროგრამის მთავარ ფუნქციას - ფუნქციას Calculate(). ფუნქციის ძირითადი ნაწილი for ციკლშია მოთავსებული, რომელიც 0-დან ბოლო დროით შრემდე მუშაობს. ყოველი იტერაციისას ხდება ორი შემოწმება: თუკი რაიმე წინასწარ შერჩეულ  $\epsilon$ -ზე მეტმა კიბოს უჯრედმა მიაღწია საზღვარს, ითვლება, რომ პაციენტი მოკვდა და სიმულაცია მთავრდება. ხოლო თუ კიბოს უჯრედების ჯამური რაოდენობა ნაკლებია  $\epsilon$ -ზე, ითვლება, რომ პაციენტი განიკურნა და ამ შემთხვევაშიც მთავრდება სიმულაცია.

```
bool Calculate(){
    printAll(masout);
    gout<<"Tumors (T)\tCD8+ (L)\tNK (N)\n";
    printSum(gout);
    for( int t = 0; t < timeLayers; t++){
        if (checkTumorHasGrown()){
            endCalculation(false);
            return true;
        }
        if (checkTumorIsCured()){
            endCalculation(true);
            return true;
        }
        for (int i = 0; i < Points; i++)
            for (int j = 0; j < Points; j++)
            {
                Told[i][j] = Tnew[i][j];
                Lold[i][j] = Lnew[i][j];
                Nold[i][j] = Nnew[i][j];
            }
        for (int i = 0; i < Points; i++)
        {
            Lnew[i][0] = L_glob(i, 0);
            Nnew[i][0] = N_glob(i, 0);

            Lnew[0][i] = L_glob(0, i);
            Nnew[0][i] = N_glob(0, i);

            Lnew[i][Points - 1] = L_glob(i, Points - 1);
            Nnew[i][Points - 1] = N_glob(i, Points - 1);

            Lnew[Points - 1][i] = L_glob(Points - 1, i);
            Nnew[Points - 1][i] = N_glob(Points - 1, i);
        }
        for (int i = 1; i < Points - 1; i++)
            for (int j = 1; j < Points - 1; j++)
            {
                Tnew[i][j] = T_loc(i, j);
                Lnew[i][j] = L_loc(i, j);
                Nnew[i][j] = N_loc(i, j);
            }
        if ( t%1000 == 0 )
        {
            printCondition(cout,t);
            printAll(masout);
        }
        printSum(gout);
    }
    saveData(save);
    return false;
}
```

ამის შემდეგ ხდება „წინა ბიჯის მასივებში“ „ახალი ბიჯის მასივების“ კოპირება. შემდეგ, NK და T უჯრედებისთვის ვითვლით მნიშვნელობებს საზღვრებზე შესაბამისი ფუნქციების გამოძახებით. ამასთან, რაკი უჯრედების რაოდენობა არ შიძლება იყოს უარყოფითი, ფუნქციებში ხდება შემოწმება: თუ წერტილში მნიშვნელობა უარყოფითია, ვაბრუნებთ 0-ს. კიბოს უჯრედებისთვის ითვლება, რომ საზღვარზე მათი რაოდენობა 0-ა.

საზღვრებზე მნიშვნელობების დათვლის შემდეგ ხდება სამივე მასივისთვის „შიგა წერტილებზე“ მნიშვნელობების დათვლა, აქაც შესაბამისი ფუნქციების გამოძახებით და იმავე შემოწმების ჩატარებით. მნიშვნელოვანია, რომ ნებისმიერ ფუნქციაში უჯრედების ურთიერთქმედება ხდება მხოლოდ მაშინ, თუკი მოქმედი უჯრედების რაოდენობა მეტია გარკვეულ  $\epsilon$ -ზე, რადგან რეალურად 0.00005 უჯრედი გავლენას ვერ მოახდენს სხვა უჯრედებზე, საჭიროა 1 უჯრედის არსებობა მაინც. უჯრედების მოძრაობა აღწერილია

```
double qsiDiffusion(double (*qsi)(int, int), double ** Mas, int i, int j)
{
    return (Tau/(Int*Int))*(
        0.5*((*qsi)(i+1,j) + (*qsi)(i,j)) * (Mas[i+1][j] - Mas[i][j])
        - 0.5*((*qsi)(i,j) + (*qsi)(i-1,j)) * (Mas[i][j] - Mas[i-1][j])
        + 0.5*((*qsi)(i,j+1) + (*qsi)(i,j)) * (Mas[i][j+1] - Mas[i][j])
        - 0.5*((*qsi)(i,j) + (*qsi)(i,j-1)) * (Mas[i][j] - Mas[i][j-1])
    );
}
```

ჩვეულებრივი დიფუზიისა და ცვლადპარამეტრიანი დიფუზიის მეშვეობით. ქვევით მოყვანილია პარამეტრის მნიშვნელობის დასათვლელი ფორმულა სიმსივნეს უჯრედების დიფუზიისთვის.

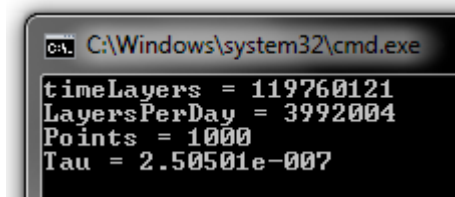
```
double qsi_t(int i, int j) {
    if ( Told[i][j] <= Tmin )
        return 0;
    if ( Told[i][j] >= Tmax )
        return 1;
    return (Told[i][j]-Tmin)/(Tmax - Tmin);
}
```

როდესაც პროგრამა დაასრულებს გამოთვლებს, იგი მთავრდება. ამასთან, თვლის დაწყებამდე, დასრულების მერე და გარკვეული ინტერვალებით თვლის პროცესშიც ხდება სამივე ტიპის უჯრედის

მასივების გამოტანა ტექსტურ ფაილში,

რომელსაც შემდგომ ვიყენებთ უჯრედების გავრცელების ვიზუალიზაციისათვის. ამასთან, ყოველ ბიჯზე იბეჭდება სამივე ტიპის უჯრედების ჯამური რაოდენობა, რათა შევძლოთ გრაფიკის აგება მაღალი მიახლოებით. (იხ. სრული პროგრამული კოდი ჩვენს Google Drive-ის მისამართზე)

## პარალელური გამოთვლები



```
C:\Windows\system32\cmd.exe
timeLayers = 119760121
LayersPerDay = 3992004
Points = 10000
Tau = 2.50501e-007
```

ჩვენი მოდელი ყოველ დროით ბიჯზე ითვლის სამ ტოლ კვადრატულ მატრიცას. იმ შემთხვევაში თუ სიზუსტეს (წერტილების რაოდენობას) ავირჩევთ 1000x1000ზე მაშინ ყოველი დღის სიმულაციისთვის დაგვჭირდება 3,992,004 დროითი შრე ხოლო იმას თუ გავითვალისწინებთ რომ

სიმსივნეზე დაკვირვება 1 თვეს მაინც გრძელდება ხოლმე მაშინ შრეების რაოდენობა 119,760,121-ს აღწევს. რისი სიმულაციაც ჩვენს ლეპტოპებზე დაახლოებით 203 წელიწადს გასტანდა. სწორედ ამიტომ ნათელი იყო რომ კომპიუტერული იმპლემენტაცია საჭირო იყო დღევანდელ ყველაზე ცნობილ გაპარალელების ტექნიკების გამოყენებითაც. ამისათვის ავირჩიეთ 3 განსხვავებული ტექნიკა.

# OpenMP აპარალებს პროცესს კომპიუტერის ბირთვებზე, რაც

შესაძლებელს ხდის მრავალბირთვიან პროცესორებზე ბირთვის რაოდენობაჯერ ნაკლებ დროში დაასრულოს პროცესი სწორად გაპარალებების შემთხვევაში. გაპარალებების ტექნიკის დახვეწილობამ და თანამედროვე კომპილერების სრული მხარდაჭერის წყალობით კოდის გაპარალება მეტად მარტივი აღმოჩნდა. (იხ. სრული პროგრამული კოდი ჩვენს Google Drive-ის მისამართზე)



```
#pragma omp parallel num_threads(4)
{
    for( int t = 0; t < timeLayers; t++){

        #pragma omp single
        {
            //შემოწმებები რომლებიც ერთ ბირთვზე უნდა შესრულდეს
        }

        #pragma omp for
        for (int i = 0; i < Points; i++)
        for (int j = 0; j < Points; j++)
        {
            //მასივების კოპირება
        }

        #pragma omp for
        for (int i = 0; i < Points; i++)
        {
            //ახალი შრეების საზღვრები
        }

        #pragma omp for
        for (int i = 1; i < Points - 1; i++)
        for (int j = 1; j < Points - 1; j++)
        {
            //ახალი შრეების შიდა არე
        }

        #pragma omp single
        {
            //სერვისები, ბეჭდვა
        }
    }
}
```

# Nvidia CUDA ახალგაზრდა ტექნოლოგიაა ვიდეო დაფაზე

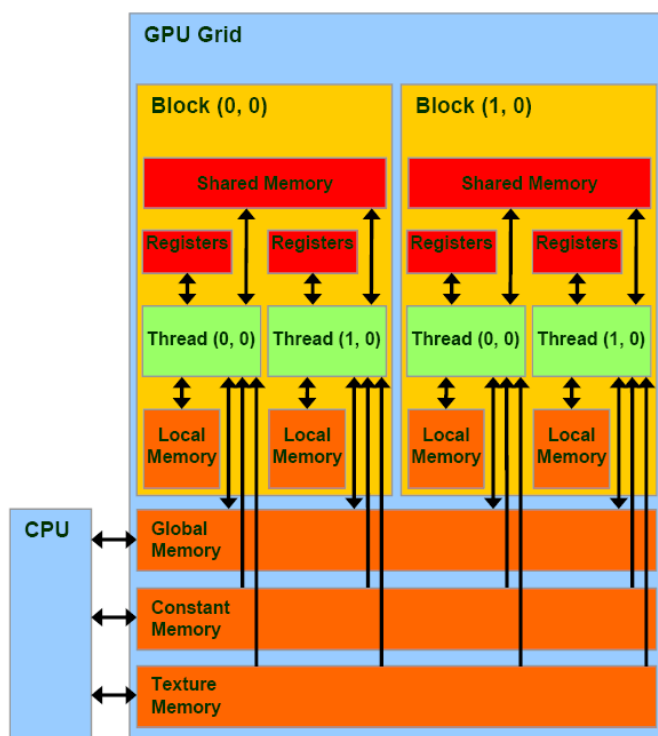
გაპარალელისთვის. მოგეხსენებათ თანამედროვე ვიდეო დაფების არქიტექტურა სწორედ რომ ოპტიმიზირებულია მატრიცების გადამრავლებაზე და სხვა მათემატიკურ ოპერაციებზე რათა რაც შეიძლება სწრაფად დაარენდეროს რთული გრაფიკული გამოსახულებები და გეომეტრიები. სწორედ ამიტომ ვიდეო დაფაზე გაპარალელება ერთ-ერთ საუკეთესოდ ითვლება პერსონალურ კომპიუტერებზე გამოთვლებისას.



CUDA გაპარალელების მოდელი ეყრდნობა მეხსიერების განცალკევებას. მისი წარმოდგენით არსებობს Device(ვიდეო დაფა) და Host(პროცესორი) დამოუკიდებელი მეხსიერებები. იმისათვის რომ შესაძლებელი იყოს ჩვენს მიერ შეტანილ მატრიცებზე მოქმედებების ჩატარება ჯერ საჭიროა მოხდეს დინამიური მეხსიერების განთავსება ვიდეო დაფაზე და შემდეგ კოპირება მეხსიერების სასურველი ფრაგმენტის ჰოსტიდან დივაიზზე. კუდა ფუნქციებს კერნელები ეწოდებათ რომლებიც განისაზღვრება `__global__` სუფიქსით და დასაბრუნებელი მნიშვნელობა აუცილებლად `void`-ია. გლობალური ფუნქციები გამოიძახება ჰოსტზე და მიეთითება პროცესების რაოდენობა სპეციალური ოპერატორით `<<< N, M >>>` სადაც N წარმოადგენს ბლოკების რაოდენობას ბადეზე, ხოლო

M პროცესების რაოდენობას თითო ბლოკზე. მოდელის ფუნქციები გამოისახა `__device__` კერნელების სახით

ხოლო პარამეტრები განისაზღვრა `__constant__` მეხსიერების ფრაგმენტში რომელიც ითვლება ყველაზე სწრაფი წვდომის მეხსიერებათ ვიდეო დაფაზე თუმა მისი მოცულობა მხოლოდ 64 კილობაიტია. პროგრამული მოდელი სრულებით განსხვავდებოდა ყველა დანარჩენისგან სწორედ ამიტომ CUDA-ზე გაპარალელება ყველაზე შრომატევადი აღმოჩნდა. საჭირო გახდა მოდელის ყველა ფუნქციის თავიდან დაგეგმვა. (იხ. სრული პროგრამული კოდი ჩვენს Google

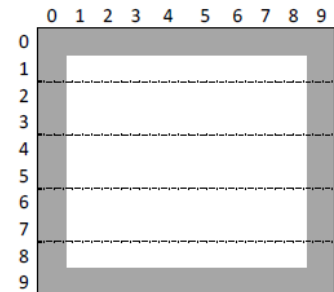


Drive-ის მისამართზე)



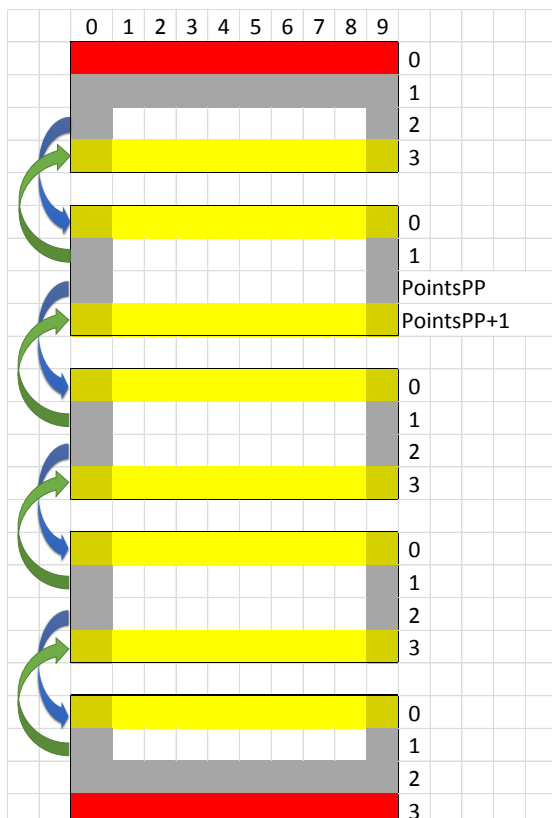
**MPI** ტექნოლოგია აკავშირებს ქსელში ჩართულ კომპიუტერებს და იყენებს მათ საერთო რესურსებს დროტევადი გამოთვლების სწრაფად ჩასატარებლად.

MPI გაპარალელების ტექნიკისას ორად-ორი სირთულე წარმოიქმნა. ეს იყო დიფუზია და ყოველი შრისთვის საჭირო ყველა სხვა მასივის ელემენტების ჯამის დათვლა. თავად გაპარალელების მოდელი იყენებს მასივების დანაწევრების პრინციპს. პარალელური გამოთვლისათვის ვანაწევრებთ მასივს ჩართული კომპიუტერების რაოდენობის ტოლ ინტერვალად, სწორედ ისე როგორც სურათზეა გამოსახული. 10x10-ზე მატრიცა დაყოფილია 5 ტოლ ქვემატრიცად ზომით



2x10 რომლებზეც პარალელურად შესრულდება გამოთვლები. თუმცა ამ შემთხვევაში დიფუზია პრობლემას ქმნის ვინაიდან მას სჭირდება მეზობელი წერტილების მნიშვნელობები თავისი დასათვლელად რაც პრობლემაა რადგან ყველა პროცესორმა იცის მხოლოდ თავისი ფრაგმენტის მნიშვნელობები. ამისათვის საჭირო გახდა ქვემატრიცებისთვის 2 სტრიქონის დამატება ზევით და ქვევით. ზედა სტრიქონი ზედა მეზობელ ქვემატრიცის ბოლო სტრიქონზე მოახდენდა გადაფარვას ხოლო ქვედა სტრიქონი ქვედა მეზობლის პირველ სტრიქონზე.

ემდეგ სურათზე ნათლად ჩანს პრობლემის გადაჭრის გზა. ყოველი შრის დათვლის წინ ყველა პროცესი მისი მეზობლებისგან წამოიღებს და გაუგზავნის საჭირო სტრიქონს და დაიწყებს გამოთვლას. სურათზე სწორედ ამ ისრებითაა აღნიშნული გაგზავნა მიღება.



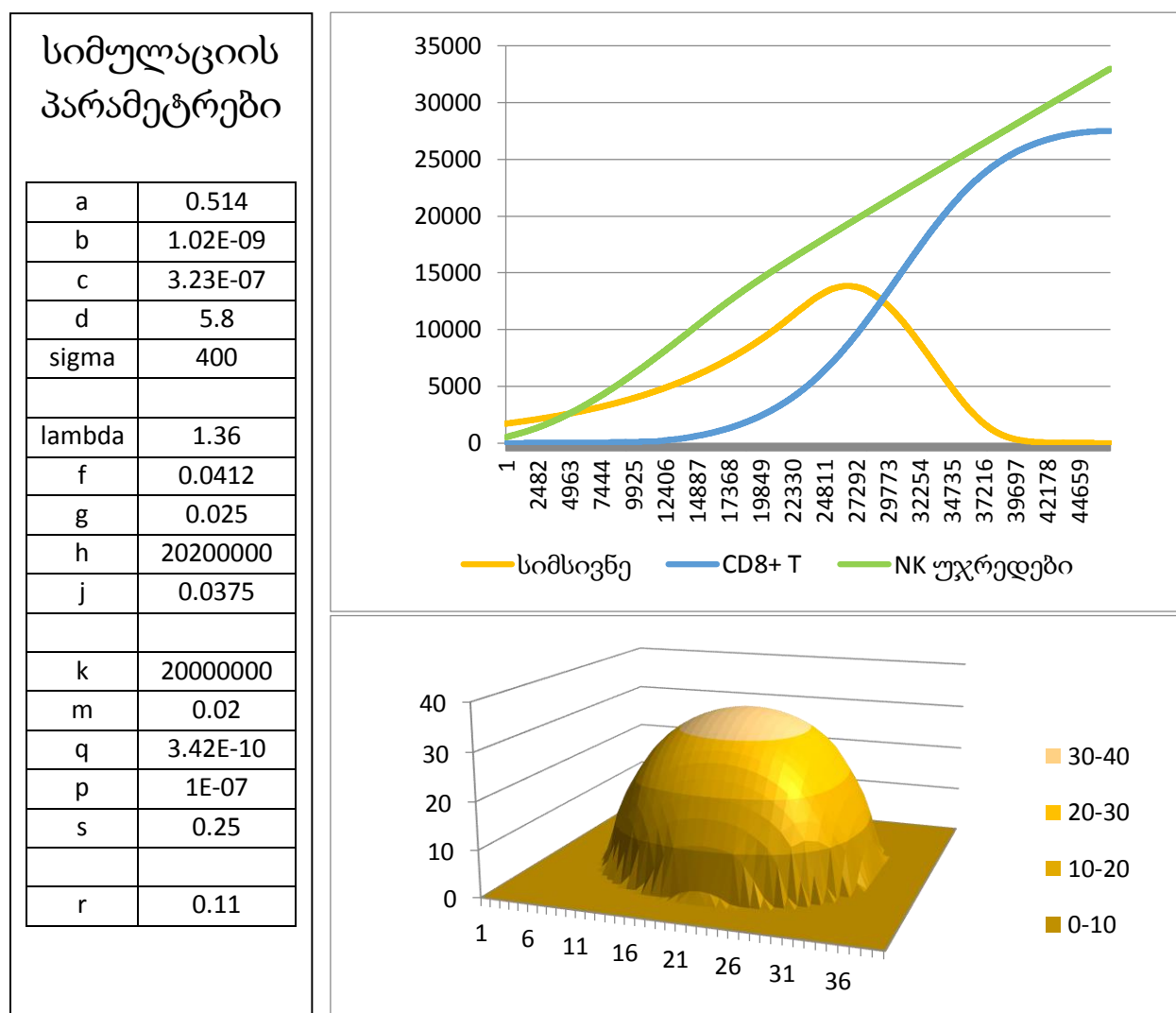
ყვითელი ფერით აღნიშნულია გაცვლითი სტრიქონები, წითლად აღნიშნულია გამოუყენებელი სტრიქონები რადგან პირველ პროცესს არ ჰყავს ზედა მეზობელი ხოლო ბოლოს - ქვედა. PointsPP – PointsPerProcess ამ შემთხვევაში 2-ის ტოლია რადგან ყოველი ქვემატრიცის სიგრძე 2ია.

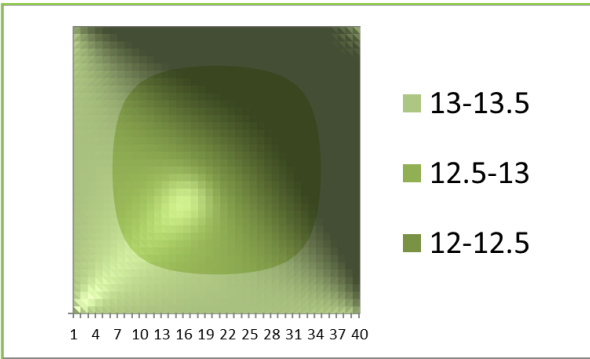
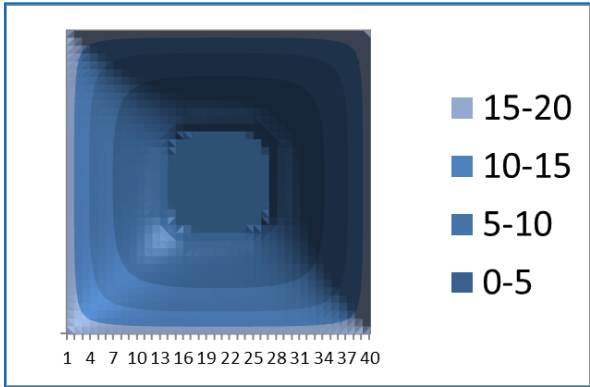
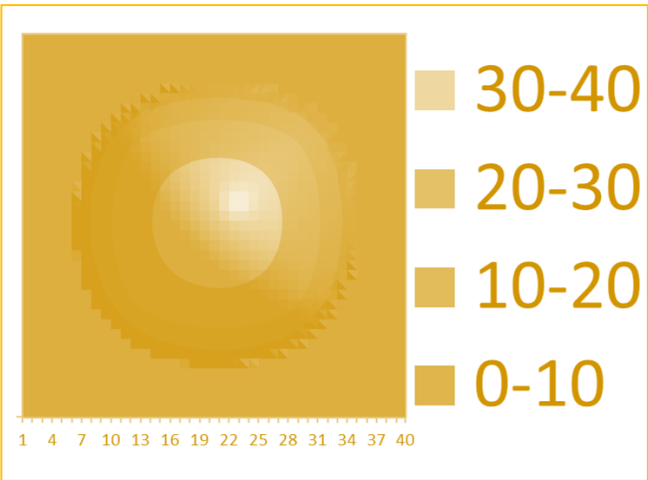
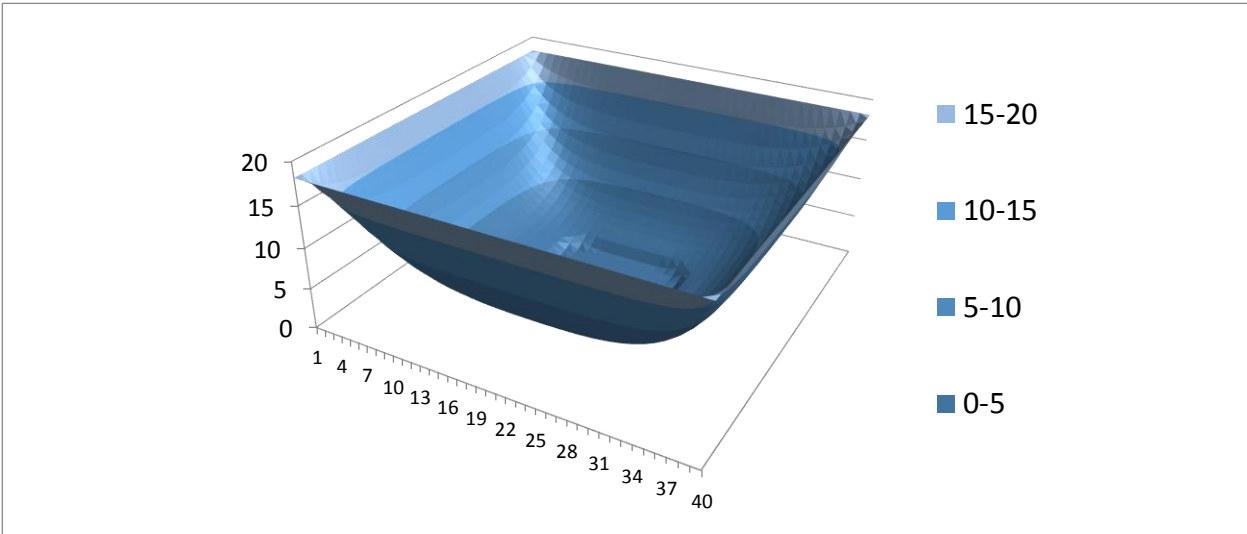
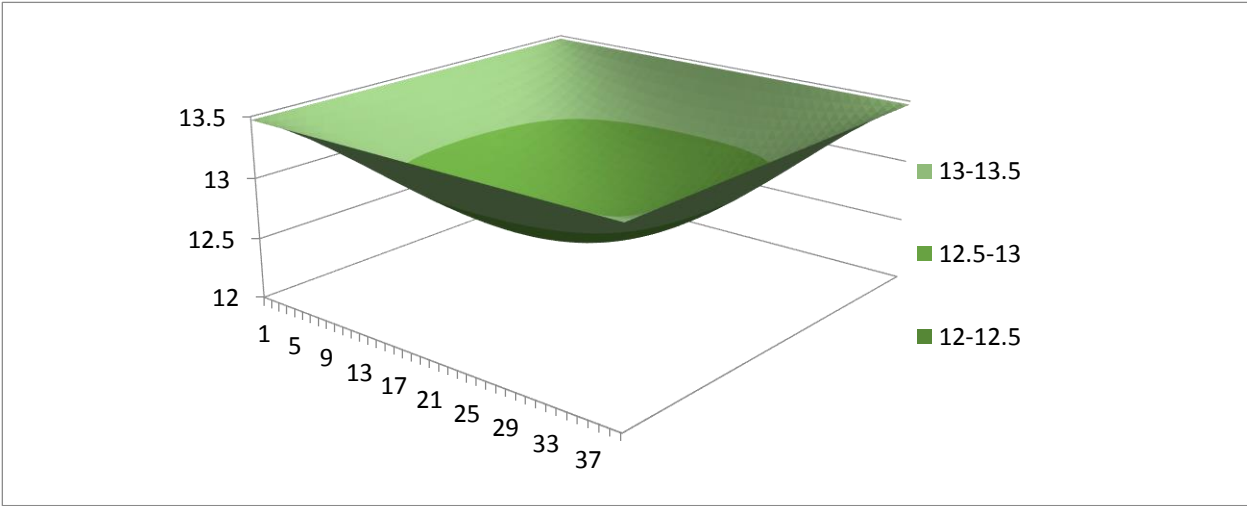
მეორე პრობლემა, ის რომ ყველა შრის დათვლის ფორმულებში ფიგურირებს მასივთა ჯამები გადაწყდა ყოველი შრის წინ ჯამების ურთიერთგადაგზავნით. შრის დაწყების წინ ყველა პროცესი ითვლის თავიანთი ფრაგმენტის ჯამს და უგზავნის საწყის root პროცესს რომელიც აჯამავს ქვეჯამებს და გადაუგზავნის უკან შვილ პროცესებს დათვლილ სრულ ჯამს. (იხ. სრული პროგრამული კოდი ჩვენს Google Drive-ის მისამართზე)

# სიმულაციების შედეგები

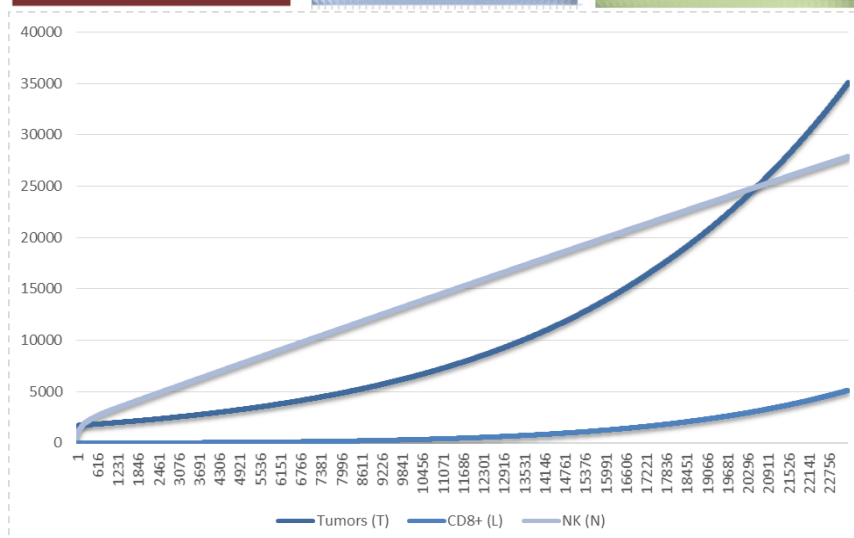
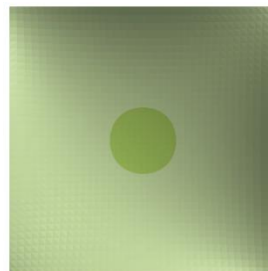
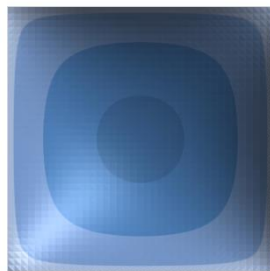
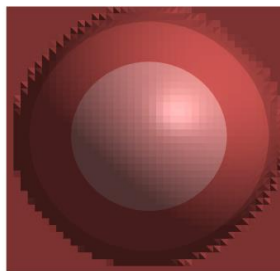
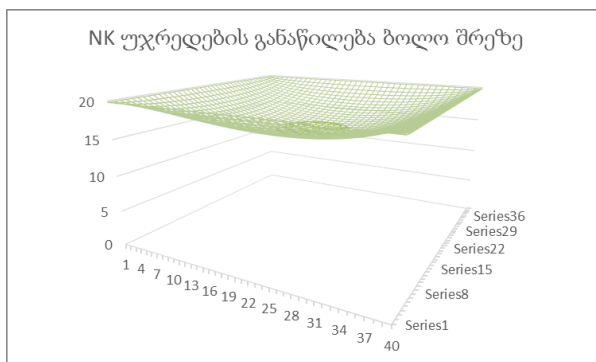
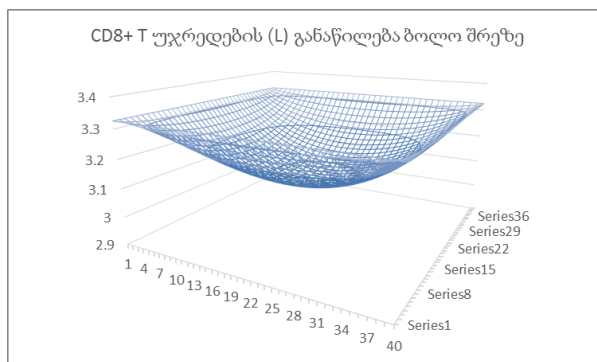
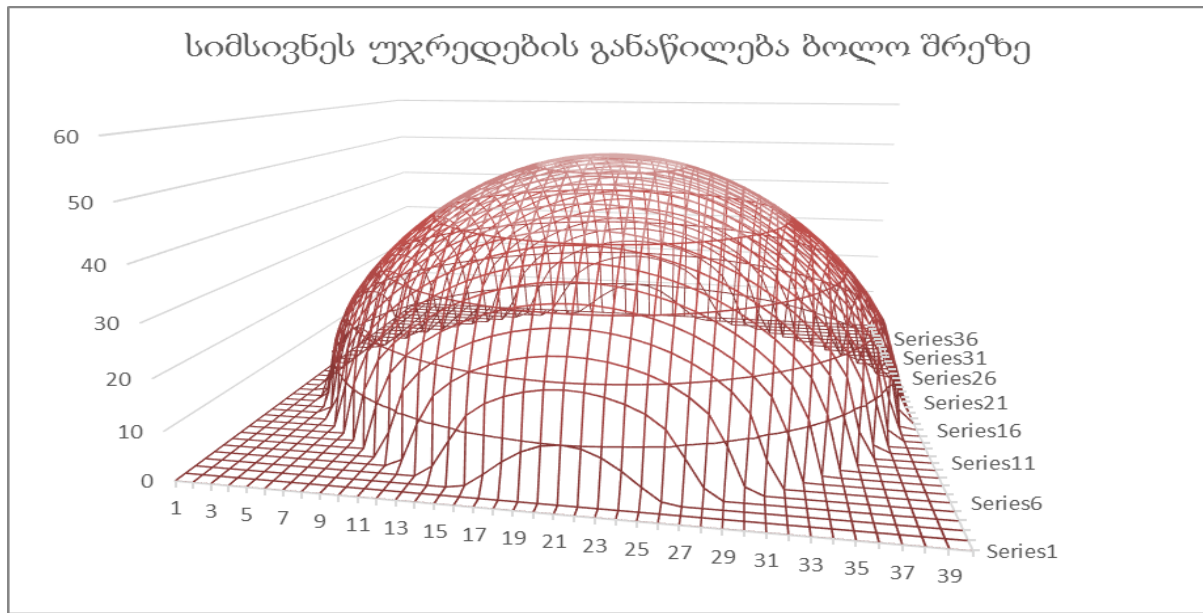
როგორც უკვე აღვნიშნეთ ჩატარებული სიმულაციები არ უნდა ჩაითვალოს რაღაცა დონის მიახლოებად ზუსტ პასუხებთან. საწყისი მიახლოება ფილისის მოდელის ადამიანის იმუნური სისტემის პარამეტრებია. ჩვენი მიზანი იყო მოდელის კომპონენტებს შორის კანონზომიერების ვიზუალიზაცია. პროგრამას გამოაქვს ორი განსხვავებული პასუხი ორ სხვადასხვა ნაკადში. პირველ ფაილში იწერება ჯამური მნიშვნელობები სამივე მასივისა ცალცალკე ყოველ შრეზე რომლითაც შეგვიძლია 2-გ-გრაფიკის აგება. ხოლო მეორე ფაილში გამოდის წინასწარ განსაზღვრულ ბიჯების ინტერვალებზე 3-განზომილებიანი ვითარების სურათი. შესაბამისად ჩვენ არამართო შეგვიძლია გრაფიკს დავაკვირდეთ როგორც ოდეს მოდელებშია არამედ გრაფიკის ყოველ წერტილში ვნახოთ რა ვითარება იყო ორგანიზმში (კერძოდ კი შესასწავლ არეზე). მეორე ფაილის პასუხი მატრიცაა ყოველ წერტილში უჯრედის რაოდენობის მნიშვნელობებით შევსებული, რაც საბოლოოდ 3-განზომილებიან გრაფიკს გვაძლევს. სიმულაციების სრული კასკადის ნახვა ასევე შეგვიძლია იხილოთ ჩვენს Google Drive მისამართზე [bit.ly/tsunmpde](http://bit.ly/tsunmpde).

გრაფიკი 1. სიმსივნის ზრდა შეაჩერა ექვეტურმა CD8+ T უჯრედების მოქმედებამ რომლის ფაქტორი განისაზღვრება  $r$  კოეფიციენტით. 3D ვიზუალიზაცია 26-ე დღეს როცა სიმსივნის მოცულობა ყველაზე დიდი იყო





გრაფიკი 2. 22-ე დღეზე სიმსივნეს უჯრედებმა მიაღწიეს საზღვრებს და სიმულაცია შეწყდა. L უჯრედების სრული აქტივაცია ვერ მოესწრო.



სიმულაციის ფილისის ნაშრომში გამოყენებული პარამეტრებით გაშვებისას მივიღეთ არადამაკმაყოფილებელი შედეგი: NK უჯრედები გაიზარდნენ ძალიან სწრაფად, კიბოს უჯრედების ზრდა იყო ნორმალური, ხოლო CD8+T-უჯრედების ზრდა იმდენად მცირე იყო, რომ არც კი შეიმჩნეოდა. ამის გამო საჭირო გახდა ამ ორი უჯრედის ბუნებრივი ზრდის პარამეტრების შეცვლა.

როდესაც NK-ს სიგმა პარამეტრი შევამცირეთ 13000-დან 800-მდე, მისი ზრდის სიჩქარე გახდა ნორმალური, ამიტომ გადავწყვიტეთ, რომ ასე დაგვეტოვებინა. რთული იყო CD8+T უჯრედების პარამეტრების შერჩევა: ბალანსი იყო საპოვნელი. მცირე პარამეტრზე იგი ფაქტობრივად არ წარმოიქმნებოდა, ხოლო დიდი პარამეტრის შემთხვევაში იგი იმდენად სწრაფად იზრდებოდა, რომ მყისვე ანადგურებდა კიბოს უჯრედებს.

საბოლოო ჯამში შევარჩიეთ  $r=0.11$  პარამეტრი, როგორც ოპტიმალური ჩვენი სიმულაციებისათვის. NK-ს სიკვდილიანობის  $f$  პარამეტრის გაზრდა იწვევს ამ ტიპის უჯრედების ზედმეტად სწრაფ განადგურებას. როდესაც ვამცირებ  $d$  პარამეტრს, ვიღებთ სიმულაციას, სადაც ყველა უჯრედის გრაფიკი არის ზრდადი და ისინი ფაქტობრივად არ ახდენენ გავლენას ერთმანეთზე (არ ანადგურებენ ერთმანეთს).  $a$ , კიბოს ბუნებრივი ზრდის, პარამეტრის დიდი ცვლილება იწვევს მის შეუჩერებლ, სწრაფ ზრდას პარამეტრის გაზრდისას და უსუსურობას - შემცირებისას. ხოლო სხვა პარამეტრების მცირე ცვლილებები არ ახდენენ რეალურ გავლენას, დიდი ცვლილებები კი აფუჭებენ სიმულაციებს.

ამგვარად, დავასკვნით, რომ დასაკვირვებლად საუკეთესო სიმულაციების მისაღებად საჭირო არის  $a$ , სიგმა და  $r$  პარამეტრებით მანიპულირება და მივიღეთ სხვადასხვა, ODE-მოდელთან მიახლოებული გრაფიკები ამ პარამეტრების გამოყენებით.

# გამოყენებული ლიტერატურა

- [1] – A Validated Mathematical Model of Cell-Mediated Immune Response to Tumor Growth  
Lisette G. de Pillis, Ami E. Radunskaya and Charles L. Wiseman (2005) [available here](#)
- [2] – A reaction-diffusion model for the growth of avascular tumor.  
S. C. Ferreira Junior<sup>1</sup>, M. L. Martins and M. J. Vilela (2008) [available here](#)
- [3] – A Reaction-Diffusion Model of Cancer Invasion  
Robert A. Gatenby and Edward T. Gawlinski (1996) [available here](#)
- [4] – A HIERARCHY OF CANCER MODELS AND THEIR MATHEMATICAL CHALLENGES.  
Avner Friedman (2004) [available here](#)

