1 სითბოგადაცემის განტოლება. 1D

სხეულში მოცულობის ერთეულში სითბოს რაოდენობის ცვლილება აღიწერება ფორმულით:

$$\delta q = C \cdot \rho \cdot \Delta T \tag{1}$$

დროის ერთეულში ცვლილება შესაბამისად იქნება

$$\frac{\partial q}{\partial t} = C \cdot \rho \cdot \frac{\partial T}{\partial t} \tag{2}$$

განვიხილოთ ერთ განზომილებიანი ამოცანა. ანუ სითბოს გადაცემა ხდება მხოლოდ 1 მიმართულებით. ვთქვათ გვაქვს რაღაც სიგრძის ერთგვაროვანი განიკვეთის (S) მქონე ცილინდრული ფორმის ღერო მიმართული x ღერძის გასწვრივ. მაშინ ზემოთ მოყვანილი ფორმულები სითბოს ცვლილებისათვის ღეროს a და b წერტილებს (რომელთა შორის მანძილია Δx) შორის მოთავსებულ მონაკვეთში იქნება:

$$\frac{\partial Q}{\partial t} = \int\limits_a^b C \cdot \rho \cdot \frac{\partial T}{\partial t} S dx \tag{3}$$

მეორე მხრივ ცნობილია, რომ სითბოს ცვლილება დროის ერთეულში პროპორციულია ტემპერატურის ცვლილებისა , ღეროს განიკვეთის ფართობის (რაც უფრო მეტია ფართობი მით მეტი სითბო გადის/შედის ღეროში) და უკუპროპორციულია სიგრძის (რაც უფრო დიდია მანძილი მით მეტი დრო სჭირდება სითბოს მიაღწიოს ერთი ადგილიდან მეორემდე). ანუ

$$\Delta Q = -k \frac{T(x + \Delta x, t) - T(x, t)}{\Delta x} S \tag{4}$$

სადაც k ეწოდება სითბოგამტარობის კოეფიციენტი. ზოგად შემთხვევაში

$$\Delta Q = -\int\limits_S k \cdot \mathrm{grad} T \cdot ds \cdot \vec{n} \tag{5}$$

სადაც $\vec{n}$ S ზედაპირის ნორმალია საიდანაც გაედინება სითბო. სიდიდეს $\vec{F} = -k \cdot \mathrm{grad} T$ ეწოდება სითბოს ნაკადი. ნიშანი "$-$" ნიშნავს, რომ როცა გარემოს ტემპერატურა ნაკლებია სხულის ტემპერატურაზე სითბო ტოვებს სხეულს. გავიხსენოთ დივერგენციის თეორემა:

$$\iiint_T (\nabla \cdot \boldsymbol{F}) \, \mathrm{d}V = \iint_S \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{S}$$

მაშინ (5) მარჯვენა მხარე ზედაპირული ინტეგრალით შეიცვლება მოცულობითი ინტეგრალით. შესაბამისად დროის ერტეულში სითბოს ცვლილება იქნება

$$\frac{\partial Q}{\partial t} = \int\limits_a^b \frac{d}{dx} \left(k \cdot \frac{\partial T}{\partial x} S \right) dx \tag{6}$$

თუ შევადარებთ (3) განტოლებას $\dfrac{\partial Q}{\partial t}$ თვის მუდმივი k შემთხვევაში გვექნება:

$$\frac{\partial T}{\partial t} = \frac{k}{C\rho} \frac{\partial^2 T}{\partial x^2} \tag{7}$$

Click to start

ხოლო ზოგად შემთხვევაში

$$\frac{\partial T}{\partial t} = \frac{1}{C\rho}\frac{\partial}{\partial x}\left(k \cdot \frac{\partial T}{\partial x}\right) \tag{8}$$

სითბოს ნაკადების ენაზე

$$\frac{\partial T}{\partial t} = -\frac{1}{C\rho}\mathrm{div} \cdot \vec{F} \tag{9}$$

2 სასრული სხვაობების სქემა

ჩვენი ამოცანაა (9) განტოლების ამოხსნა სხვადასხვა საწყისი და სასაზღვრო პირობებით ერთი განზომილების შემთხვევაში. გვექნება:

$$\frac{\partial T}{\partial t} = -\frac{1}{C\rho}\frac{\partial F}{\partial x} \tag{10ა}$$

$$F = -k \cdot \frac{\partial T}{\partial x} \tag{10ბ}$$

დავწეროთ (10) სასრულ სხვაობებში

$$\left(\frac{\partial T}{\partial t}\right)_i^n = -\left(\frac{1}{C\rho}\frac{\partial F}{\partial x}\right)_i^n \tag{11ა}$$

$$F_i^n = -\left(k \cdot \frac{\partial T}{\partial x}\right)_i^n \tag{11ბ}$$

სადაც n და i აღნიშნავენ დროით და სივრცით ინდექსებს. თუ გავშლით (11) შემდეგი სხვაობების მეთოდით , ხოლო ნაკადის განტოლებას სივრცის ინდექსის მიმართ წინა სხვაობების მეთოდით გვექნება:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = -\frac{1}{C_i\rho_i}\frac{F_{i+1}^n - F_i^n}{\Delta x} \tag{12ა}$$

$$F_i^n = -k_i \cdot \frac{T_i^n - T_{i-1}^n}{\Delta x} \tag{12ბ}$$

საიდანაც

$$T_i^{n+1} = T_i^n - \frac{\Delta t/\Delta x}{C_i\rho_i}(F_{i+1}^n - F_i^n) \tag{13ა}$$

$$F_i^n = -\frac{k_i}{\Delta x} \cdot (T_i^n - T_{i-1}^n) \tag{13ბ}$$

რატომ ავიღეთ ასეთი სქემა?
თუ ჩავთვლით რომ k მუდმივია, უფრო სწორედ სხეული ერთგავროვანია და ჩავსვამთ ნაკადის განტოლებას ტემპერატურის განტოლებაში მაშინ მივიღებთ რომ

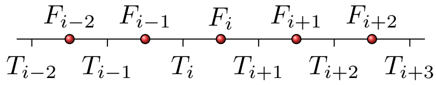
$$T_i^{n+1} = T_i^n - \frac{\Delta t/\Delta x}{C\rho} \cdot (-\frac{k}{\Delta x})((T_{i+1}^n - T_i^n) - (T_i^n - T_{i-1}^n)) \tag{14}$$

გამარტივებით და ცვლადების ერთად დალაგებით გვექნება

$$T_i^{n+1} = T_i^n + \frac{k\Delta t/\Delta x^2}{C\rho} \cdot (T_{i+1}^n - 2T_i^n + T^n \tag{15}$$

რაც ემთხვევა სითბოგამტარობის განტოლების გაშლას დროით არეში ცენტრალური სხვაობების მეთოდით ნაკადების შემოტანის გარეშე ერთგვაროვიანი სხეულისათვის

დავალება 1 *მიიღეთ* (??) *განტოლება სითბური განტოლების გაშლით დროით არეში ცენტრალური სხვაობების მეთოდით ნაკადების შემოტანის გარეშე.*



სურ. 1: სასრული სხვაობების სქემა

ასევე შესაძლებელია გასნხვავებული სქემა თუ გავშლით (11) წინა სხვაობების მეთოდით , ხოლო ნაკადის განტოლებას სივრცის ინდექსის მიმართ შემდეგი სხვაობების მეთოდით გვექნება:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = -\frac{1}{C_i \rho_i} \frac{F_i^n - F_{i-1}^n}{\Delta x} \tag{16ა}$$

$$F_i^n = -k_i \cdot \frac{T_{i+1}^n - T_i^n}{\Delta x} \tag{16ბ}$$

საიდანაც

$$T_i^{n+1} = T_i^n - \frac{\Delta t / \Delta x}{C_i \rho_i} (F_i^n - F_{i-1}^n) \tag{17ა}$$

$$F_i^n = -\frac{k_i}{\Delta x} \cdot (T_{i+1}^n - T_i^n) \tag{17ბ}$$

ჩვენ სწორედ მეორე მიდგომას გამოვიყენებთ. ეს სქემა გამოსახულია სურ.1 ეს სქემა შეიძლება წარმოვიდგინოთ შემდეგი სახით. დასათვლელი არე იწყება და მთავრდება ტემპერტურით. მათ შორის კი შეგვიძლია წარმოვიდგინოთ ტემპერატურის ნაკადი. მოცემული i ტემპერატურისათვის მეზობელი ნაკადებია i და $i-1$. შესაბამისად მოცემული i ნაკადისათვის მეზობელი ტემპერატურებია i და $i+1$. ცხადია ეს სქემაც მოგვცემს ცენტრალური სხვაობებით ტემპერატურის გაშლას ერთგვაროვანი ობიექტის შემთხვევაში.
განვიხილოთ მაგალითი. შეჰქმენით მატლაბის .m ფაილი და შეიყვანეთ ქვემოთ მოყვანილი კოდი:

Matlab კოდი 1: temper1.m

```
L=0.1; % ღეროს სიგრძე. ვიყენებთ მეტრულ სისტემას.
n=15000;%დროითი ბიჯის რაოდენობა
dx=0.005;%სივრცის დაყოფის ბიჯი
N=L/dx+1;%წერტილების რაოდენობა ღეროზე სადაც ვითვლით ტემპერატურას
T=ones(N,1);%ტემპერატურების მასივი (N სიგრძის ერთიანების ვექტორი)
TT=zeros(n,N-2);%ორგანზომილებიანი მასივი(შევსებული ნულებით)
%ტემპერატურებისთვის დროის ყოველ მომენტში
% არ შეიცავს პირველ და ბოლო წერტილებს(N-2)
x=0:dx:L; %L სიგრძის ღეროს დაყოფა dx ბიჯით იძლევა N წერტილს
T0=100; %ტემპერატურა ღეროს დასაწყისში
TL=20;%ტემპერატურა ღეროს ბოლოში.
T=T.*20;%ტემპერატურის ვექტორის ყოველ მნიშვნელობას ენიჭება 20;
T(1)=T0; %ღეროს დასაწყისს ენიჭება T0;
T(N)=TL;%ღეროს ბოლოს ენიჭება TL
Told=T;%იქნება მასივი Told სადაც შეინახება ტემპერატურების
%ძველ დროით ბიჯზე
k=90;%სითბოგამტარობის კოეფიციენტი
dt=690;%დროითი ბიჯი
C=4200;%სითბოტევადობა
ro=1000;%სიმკვრივე
alpha=k*dt/C/ro;
F1=zeros(N,1);%1 განზომილებიანი მასივები N ელემენტით
%სადაც შევინახავთ ნაკადებს
F2=F1;     %აუცილებელი არაა ამ მასივების შემოტანა თუ არ
%ვაპირებთ ნაკადის ვიზუალიზაციას
```

```
    j=0; %მიმდინარე დროითი ბიჯი
    while (j<n) %სანამ მიმიდანრე დროითი ბიჯი ნაკლებია დროითი ბიჯის
    %რაოდენობაზე
    j=j+1;%ვზრდით დროით ბიჯს
    for i=2:N-1 %გავდივართ ღეროს 1 და ბოლო წერტილების გარდა და
    %ვითვლით ნაკადებს და ტემპერატურებს
    F1(i)=-k*(Told(i+1)-Told(i));
    F2(i)=-k*(Told(i)-Told(i-1));
    T(i)=Told(i)-dt/C/ro*(F1(i)-F2(i));
    end;
    TT(j,1:N-2)=T(2:N-1);% ვინახავთ მიმდინარე ტემპერატურების
    %ვექტორს მოცემული დროითი ბიჯისთვის
    Told=T;% ვანახლებთ ძველი ტემპერატურების მასივს.ეს დაგვჭირდება
    %ახალ დროით ბიჯზე;
    if(mod(j,20)==0)% ყოველ მეოცე დროით ბიჯზე ვხატავთ ტემპერატურებს
    %და პირველ ნაკადს
    subplot(2,1,1);
    %surface(TT,'EdgeColor','none');
    plot(x,T);
    subplot(2,1,2); plot(x,F1);
    drawnow
    end
    end;
```

3 1D სითბოგამტარობის კოდის გაუმჯობესება

მიუხედავად იმისა რომ ზემოთმოყვანილი Matlab კოდი "მუშაობს" მას აქვს რამდენიმე ნაკლი. მთავარი ნაკლი ისაა, რომ შემოსაზღვრული ვართ დროითი ბიჯის რაოდენობით. ნაკლოვანება გამოიხატება იმაში, რომ ჩვენ წინასწარ არ ვიცით რამდენი ბიჯი, ანუ იტერაციის რაოდენობა, დაგვჭირდება იმისათვის, რომ პროცესმა მიაღწიოს სტაციონარულ მდგომარეობას. ეს ნიშნავს, რომ კოდი

```
while (j<n) %სანამ მიმიდანრე დროითი ბიჯი ნაკლებია დროითი ბიჯის
%რაოდენობაზე
```

უნდა შეიცვალოს და უნდა დაემატოს იტერაციის შეწყვეტის კრიტერიუმი, ანუ პირობა რომლის შესრულებისას შეგვიძლია ვთქვათ, რომ დათვლის პროცესი დამთავრებულია ცვადია, ასეთი კრიტერიუმია სტაციონალური მდგომარეობის მიღწევა, რაც ნიშნავს , რომ ტემპერატურა აღარ იცვლება. ანუ $while(j < n)$ ნაცვლად გვექნება უსასრულო ციკლი $while(1)$ და ტემპერატურა არ იცვლება, ანუ $abs(T(i) - Told(i)) < \varepsilon$ სადაც ε მცირე რიცხვია. ქვემოთ მოყვანილ კოდში ვამოწმებთ თუ რამდენი პროცენტით შეიცვალა ტემპერატურა. თუ ყველგან დასათვლელ არეში ტემპერატურის ცვლილება (პროცენტულად) ნაკლებია რაღაც წინასწარ მოცემულ რიცხვზე მაშინ ვწყვეტთ $while(1)$ ციკლს. აღსანიშნავია ერთი რამ. მიუხედავად იმისა , რომ მასივის ზომა, სადაც ვინახავთ ტემპერატურებს დახატვისათვის TT არის 100, Matlab თავად უცვლის ზომას საჭიროების შემთხვეაში. დათვლის დამთავრების შემდეგ გადადით Workspace სანიშნეზე და ნახავთ ცვლადს TT. გააააქტიურეთ ეს ცვლადი. "მაუსის" მარჯვენა ღილაკზე დაჭერით გამოჩნდება მენიუ სადაც შეგიძლიათ აირჩიოთ მონაცემების 3D ვიზუალიზაციის სხვადასხვა სახე. ნაწილი ვიზუალიზაციის შესაძლებლობისა მოყვანილია ქვემოთ კოდში. კომენტარი გაუკეთეთ დახატვის აქტიურ ბრძანებას და მოხსენით კომენტარის ნიშანი თქვენთვის სასურველ ბრძანებას. ქვემოთ მოყვანილ პროგრამულ კოდში კომენტარები ერთვის მხოლოდ იმ ნაწილს, რაც გამსხვავდება ძველი ვარიანტიდან

Matlab კოდი 2: temper2.m

```

clear all;
```

```
L=0.1;
n=1:100;
dx=0.001;
N=L/dx+1;
T=ones(N,1);
deltaT=zeros(N,1);% ტემპერატურების ცვლილების მასივი
TT=ones(size(n),N);
x=0:dx:L;
T0=100;
TL=20;
T=T.*20;
T(1)=T0;
T(N)=TL;
tol=0.01;%ცდომილება პროცენტებში. დათვლის დასრულების კრიტერიუმი
Told=T;
k=490;
dt=100;
C=200;
ro=1000;
alpha=k*dt/C/ro;
F1=zeros(N,1);
F2=F1;
graphcount=1;%გრაფიკების მთვლელი მასივისათვის
step=1;%მიმდინარე დროითი ბიჯი
while (1) %იწყება უსასრულო ციკლი
step=step+1;%ვზრდით დროით ბიჯს
for i=2:N-1%გავდივართ ტემპერატურების მასივს
F1(i)=-k*(Told(i+1)-Told(i));
F2(i)=-k*(Told(i)-Told(i-1));
T(i)=Told(i)+dt/C/ro*(-F1(i)+F2(i));
end; T(51)=T0;%51-ე წერტილში გვაქვს მუდმივი ტემპერატურა
%შეგიძლიათ დააკომენტიროთ ეს ტოლობა
for i=2:N-1
deltaT(i)= (T(i)-Told(i)); %ითვლება ტემპერატურის ცვლილება
end;
if(mod(step,20)==0)%ყოველ მეოცე ბიჯზე ვხატავთ
subplot(3,1,1);
graphcount=graphcount+1; %ვზრდით სურათის მთვლელს
TT(graphcount,1:N)=T(1:N);%ვინახავთ ტემპერატურების მასივს დასახატად
contourf(TT,'EdgeColor','none')%ვხატავთ.შეგიძლიათ დააკომენტოთ ეს კოდი
%და მოხსნათ კომენატრის ნიშანი
%ქვემოთ მოყვანილი რომელიმე ბრძანებიდან
%contour(TT);
%surfc(TT,'EdgeColor','none');
%surface(TT,'EdgeColor','none');
%surf(TT,'EdgeColor','none');
%waterfall(TT);
%contour3 (TT);
%plot(x(1:N),T)
colorbar;
subplot(3,1,3);
plot(x(2:N-1),F2(2:N-1));
drawnow
end;
done=1;%ამ ცვლადით კონტროლდება დამთავრდა თუ არა დათვლა
for i=1:N %ვამოწმებთ შეიძლება თუ არა დათვლის დამთავრება. თუ არა done=0;
if(abs(deltaT(i)/Told(i))*100>tol)done=0;
break;%დათვლა არაა დამტავრებული.ტემპერატურა ჯერ კიდევ იცვლება.
 %თუ წერტილში ტემპერატურის ცვლილება მეტია დასაშვებზე
 %ანუ ისეთი წერტილი 1 მაინც არის არაა საჭირო სხვების შემოწმება
 %და done=0; და ვწყვეტთ for ციკლს
 %თუ ყველა წერტილში ცვლილება ნაკლებია დასაშვებზე მაშნ შეიძლება
```

```
 %დათვლის დამთავრება ანუ done=1;
end;
end;
subplot(3,1,2);
plot(x(1:N),deltaT(1:N))
if(done)break;%დათვლა დასრულებულია ანუ done=1; და წყდება
%უსასრულო ციკლი
end ;
Told=T;% ვანახლებთ ძველი ტემპერატურების მასივს
end;
```

პირველ გრაფიკზე ვცდილობთ დავხატოთ ტემპერატურები. სცადეთ სხვადასხვა ფუნქციები. ნახავთ რომ ყველაზე ადეკვატური არის ფუნქცია *waterfall*, ვინაიდან TT ცვლადში შენახულია ტემპერატურები ყოველ მეოცე ბიჯზე. *waterfall* ფუნქცია მოსახერხებელია ისეთი ამოცანისთვის სადაც გვაინტერესებს ერთ გრაფიკზე იყოს მონაცემები დროის სხვადასხვა მომენტისათვის მტელ საძიებელ არეზე.

ასევე არაა აუცილებელი შევინახოთ მარჯვენა და მარცხენა ნაკადები ორ სხვადასხვა მასივში

ქვემოთ მოყვანილი კოდში გამოყენებულია ერთი მასივი ოღონდ უნდა აღვნიშნოთ, რომ ამ შემთხვევაში საჭიროა ტემპერატურების მასივის "ორჯერ გავლა": ერთი როცა ვითვლით ნაკადს მეორე როცა ვითვლით ტემპერატურას.

თუ არ გვჭირდება ნაკადების ვიზუალიზაცია და (ან) არ ვიყენებთ ნაკადებს თვლის დამტავრების კრიტერიუმად (ამაზე ქვემოთ) მაშინ საერთოდ არაა საჭირო ნაკადისთვის მასივის შემოტანა არამედ გადავთლით დროებით ცვლადში ტემპერატურის დათვლის ციკლში.

ქვემოთ მოყვანილია კოდის ახალი ვარიანტი

Matlab კოდი 3: temper3.m

```
clc

clear all;

close all;

L=0.1;

n=1:100;

dx=0.001;

N=L/dx+1;

T=ones(N,1);

deltaT=zeros(N,1);

x=0:dx:L;

T0=100;

TL=20;

T=T.*20;

T(1)=T0;

T(N)=TL;

tol=0.001;

Told=T;
```

```
k=490;
dt=100;
C=200;
ro=1000;
F=zeros(N,1);
Ftot=zeros(N,1);
graphcount=1;
TT=zeros(20,N);
step=1;
while (1)
step=step+1;
    for i=1:N-1
    F(i)=-k*(Told(i+1)-Told(i));
    end;
    for i=2:N-1
     Ftot(i)=F(i)-F(i-1);
    end;
    for i=2:N-1
        T(i)=Told(i)-dt/C/ro*Ftot(i);
    end;
    for i=2:N-1
    deltaT(i)= (T(i)-Told(i));
    end;
if(mod(step,20)==0)
temper=subplot(4,1,1);
TT(graphcount,1:N)=T(1:N);
waterfall(TT);
graphcount=graphcount+1;
%plot(x(1:N),T)
xlabel('x');
ylabel('T');
```

```
title('Temperature');
drawnow;
flux=subplot(4,1,2);
plot(x(2:N-1),F(2:N-1));
xlabel('x');
ylabel('F');
title('Flux')
totflux=subplot(4,1,3);
plot(x,Ftot)
title('Total Flux')
xlabel('x');
ylabel('Ftot');
end;
done=1;
    for i=1:N
        if(abs(deltaT(i)/Told(i))*100>tol)done=0;
        break;
    end;
end;
subplot(4,1,4);
dT=plot(x(2:N-1),deltaT(2:N-1));
title('Temperature Change');
xlabel('x');
ylabel('dT');
if(done)break;
end ;
Told=T;
end;
```

წინა კოდისგან განსხვავებით ამ შემთხვევაში ვხატავთ სრულ ნაკადსაც. როცა გაუშვებთ დათვალზე ნახავთ, რომ სტაციონალურ მდგომარეობაში სრული ნაკადი ნულია და ასეც უნდა იყოს– რა სითბოც შედის ღეროს მონაკვეთში სითბოს იგივე რაოდენობა გამოდის მეორე ბოლოდან ანუ ღეროს ამ ნაწილში ტემპერატურა აღარ იმატებს ადგილი არ აქვს სითბოს დაგროვებას.

ნაკადები ასევე საშუალებას გვაძლევს გამოვიყენოთ "სრული ნაკადი ნულის ტოლია" პირობა როგორც დათვლის დამთავრების კრიტერიუმი, ნაცვლად "მოცემულ წერტილში ტემპერატურა არ

იცვლება" კრიტერიუმისა

დავალება 2 *შეცვალეთ პროგრამა ისე რომ, გამოვიყენოთ "სრული ნაკადი ნულის ტოლია" პირობა როგორც დათვლის დამთავრების კრიტერიუმი.*

დანარჩენი ფუნქციები რაც დაკომენტირებული იყო კოდის ძველ ვერსიებში უფრო გამოყენებადია 2 განზომილებიანი ამოცანებისათვის. ამ შემთხვევაში სურათზე აისახება მდგომარეობა დროის ერთი მოცემული მომენტისათვის. მაგრამ ამ შემთხვევაში მოსახერხებელია თითოეული სურათი შევინახოთ ფაილის სახით და შემდეგ გავაკეთოთ ანიმაცია.

მატლაბის ახალ ვერსიებს (2010 და ახალი) ასევე შეუძლია ჩაწეროს თითოეული სურათი პირდაპირ avi ფორმატში. ასეთი მიდგომა თავიდან აგვაცილებს ყოველი სურათის მონაცემების შენახვას მეხსიერებაში და შემდეგ MakeMovie ფუნქციის გამოძახებას, რაც გარჩეული გვქონდა ადრე 1 განზომილებიანი სინუსოიდალური ტალღის მოდელირებისას.

4 ანიმაციის შენახვა სურათის და ფილმის სახით

შემდეგი ნაბიჯია ანიმაციის სურათების შენახვა ან ცალკეულ სურათებად ან ფილმ ფაილად.

მაგრამ აქ არის ერთი პრობლემა თუ გვსურს "ქვეფიგურის"(*subplot*) ცალკე შენახვა ფაილში. მატლაბ *getfigure* ვერ გამოთვლის ქვეფიგურის კოორდინატებს სწორად მანამ სანამ ყველა ქვეფიგურა არაა დახატული სურათზე.

ამის თავიდან ასაცილებლად შეიძლება მოვიქცეთ შემდეგნაირად:

ა) დავხატოთ ფიგურა ნულოვან ბიჯზე (ანუ მანამ სანამ შევალთ უსასრულო ციკლში) და შემდეგ გადავიღოთ სურათი საჭირო ბიჯზე ან

ბ) პირველი დახატვის დროს არ გადავიღოთ სურათი. ქვემოთ კოდში სწორედ ამ მიდგომას ვიყენებთ, მხოლოდ იმიტომ, რომ ა) მიდგომის შემთხვევაში ან კოდის დუბლირება უნდა მოვახდინოთ ან ინიციალიზაციის მსგავსი ფუნქცია უნდა შევქმნათ და დავახატინოთ სურატი ნულოვან ბიჯზე

დავალება 3 *დაწერეთ ფუნქცია როემელიც გააკეთებს გრაფიკის ინიციალიზაციას.*

ქვემოთ მოყვანილ კოდში დამატებულია ფილმ ფაილების შექმნა როგორც ქვეფიგურისთვის ასევე სრული ფიგურისთვის. ყურადღება მაიქციეთ, რომ ქვეფიგურის შემთხვევაში წარწერები ღერძებზე და სურათის სახელი არ ინახება. მატლაბში ისინი ცალკე, ქვეფიგურისგან განსხვავებული ობიექტებია. ცხადია ყოველთვის შეიძლება არა ქვეფიგურებში ხატვა არამედ თითოეული გრაფიკისთვის სხვადასხვა ფიგურის შექმნა და იქედან სრული სურათის აღება.

Matlab კოდი 4: temper4.m

```
clc
clear all;
close all;
L=0.1;
n=1:100;
dx=0.001;
N=L/dx+1;
T=ones(N,1);
deltaT=zeros(N,1);
x=0:dx:L;
T0=100;
TL=20;
T=T.*20;
T(1)=T0;
T(N)=TL;
tol=0.03;
```

```
Told=T;
k=490;
dt=100;
C=200;
ro=1000;
F=zeros(N,1);
Ftot=zeros(N,1);
step=1;
vidObj = VideoWriter('heat.avi');% video object. inaxeba sruli surati
vidObjT = VideoWriter('Temperature.avi');% video object. inaxeba mxolod temperatura
open(vidObj);%vido objectebis inicializacia..gaxsna chasacerad
open(vidObjT);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig_pos = [100 10 1024 768]; % figuris zoma da pozicia
fig_col = [1 1 1];  % fonis feri
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fh= figure('color', fig_col, 'name', 'Heat Flow', ...
    'Position', fig_pos);% figuris sheqmna
startcapture=0;% es cvladi aucileeblia...matlabi idzleva shecdomas roca
               % vickebt individualuri subplotis gadagebas videostvs
               % yvela subploti unda iyos daxazuli rom scored gamotvalos
               %gadasagebi aris koordinatebi
               % amitom pirvel daxatvisas arafers ar vigebt
               % ubralod startcapture cvlads vanichebt 1
               %startcapture=1
while (1)

step=step+1;
    for i=1:N-1
    F(i)=-k*(Told(i+1)-Told(i));% ori masivis nacvlad gvaqvs 1 masivi
    end;
    for i=2:N-1
     Ftot(i)=F(i)-F(i-1);%sruli nakadi. stacionarul mdgomareobashi=0
    end;

    for i=2:N-1
        T(i)=Told(i)-dt/C/ro*Ftot(i);
    end;
   % T(51)=T0;


    for i=2:N-1
    deltaT(i)= (T(i)-Told(i));
    end;
if(mod(step,20)==0)
temper=subplot(4,1,1); %temperaturis subploti
plot(x(1:N),T)
xlabel('x');
ylabel('T');
title('Temperature');
if(startcapture)
    currFramet = getframe(temper);% freimis ageba.
                                  %vugebt surats
writeVideo(vidObjT,currFramet);%vcert video failshi
end
flux=subplot(4,1,2);%nakadis daxatva
plot(x(2:N-1),F(2:N-1));
xlabel('x');
ylabel('F');
title('Flux')
```

```
%aqac analogiurad shegvedzlo nakadis suratis avi faishi chacera

totflux=subplot(4,1,3);%sruli nakadi
plot(x,Ftot)
title('Total Flux')
xlabel('x');
ylabel('Ftot');

dT=subplot(4,1,4);%temperaturis cvlileba
plot(x(2:N-1),deltaT(2:N-1));
title('Temperature Change');
xlabel('x');
ylabel('dT');

startcapture=1; % yvela subploti gvaqvs
                % sheidzleba suratebis gadageba
%aq ukve surats vugebt srul fanjaras anu figure-s

currFramedTot = getframe(fh);
writeVideo(vidObj,currFramedTot);% vumatebt video failshi

end;
done=1;
    for i=1:N
        if(abs(deltaT(i)/Told(i))*100>tol)done=0;
        break;
    end;
end;
if(done)break;
end ;
Told=T;
end;
%vxuravt video failebs
close(vidObj);
close(vidObjT);
```