



IEGULDĪJUMS TAVĀ NĀKOTNĒ

ESF projekts

**”Atomāro un nepārtrauktās vides tehnoloģisko fizikālo procesu modelēšana, matemātisko metožu pilnveide un kvalitatīvā izpēte”,
Nr. 2009/0223/1DP/1.1.1.2.0/09/APIA/VIAA/008**

Tematiskā aktivitāte

4.2. Tehnoloģisko procesu matemātiskā modelēšana

Tematiskā apakšaktivitate

Nr. 4.2.1 „Siltuma un vielas apmaiņas procesu elektromagnētiskā vadība kausējumos, ievērojot procesu nelineāro mijiedarbību un starojuma siltuma apmaiņu”

Programmas

**nestacionāru siltuma un ūdens apmaiņas procesu modelēšanai daudzslāņu konstrukcijās, ievērojot to ietekmi uz materiālu raksturlielumiem
dokumentācija**

Rīga, 2012.g.

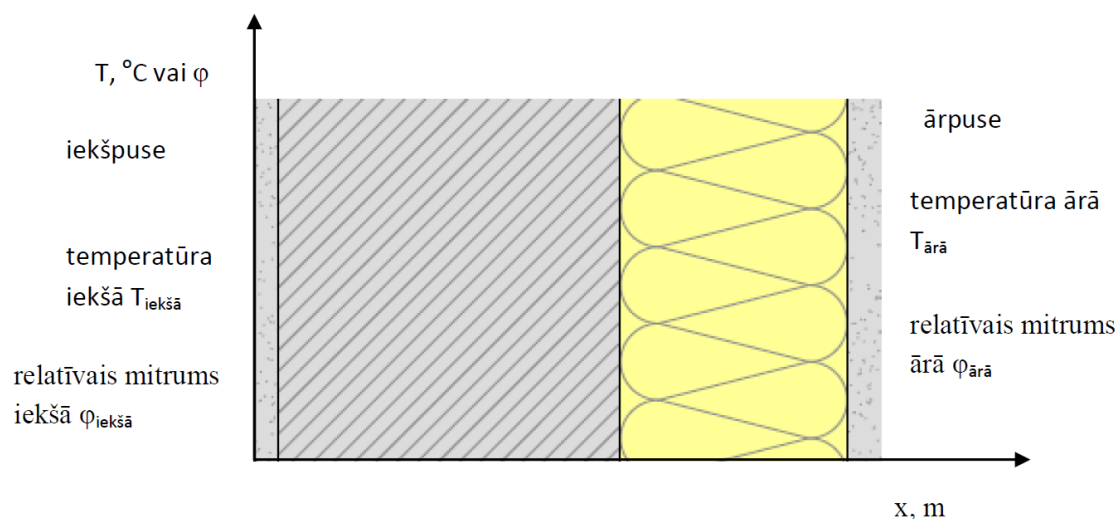
Satura rādītājs

1. Ievads.....	3
2. Izmantotie vienādojumi.....	3
3. Programmas stuktūras apraksts.....	4
4. Īss apraksts, kā strādāt ar programmu	5
5. Aprēķina piemērs	6
6. Programmas koda teksts	8
7. Apzīmējumu saraksts.....	15
8. Izmantotā literatūra	16

1. Ievads

Programmas mērķis ir aprēķināt temperatūras un relatīvā mitruma sadalījumu būvkonstrukcijā, novērtēt kondensāta rašanās iespējamību, novērtēt būvkonstrukcijas gaisa caurlaidības U vērtību un tās izmaiņu mitruma ietekmē

2. Izmantotie vienādojumi



Attēls 1: daudzslāņaina būvkonstrukcija

Saistītā diferenciālvienādojumu sistēma relatīvā mitruma un temperatūras sadalījuma konstrukcijā aprēķināšanai ir šāda (skatīt [1], [2]):

$$\frac{\partial}{\partial x} \left(\lambda \frac{\partial T}{\partial x} \right) + h_v \frac{\partial}{\partial x} \left(\delta \frac{\partial(\varphi P_{sat})}{\partial x} \right) = (\rho c + w c_w) \frac{\partial T}{\partial t}$$

$$\frac{\partial}{\partial x} \left(D_\varphi \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial x} \left(\delta \frac{\partial(\varphi P_{sat})}{\partial x} \right) = \frac{dw}{d\varphi} \frac{\partial \varphi}{\partial t}$$

Robežnosacījumi uz iekšējās un ārējās sienas virsmām ir sekojoši:

$$\alpha_{iekšā} (T_{iekšā} - T_{virsmā}) = -\lambda \frac{\partial T}{\partial x}$$

$$\alpha_{ārā} (T_{ārā} - T_{virsmā}) = \lambda \frac{\partial T}{\partial x}$$

$$\beta_{iekšā} (\varphi_{iekšā} P_{sat,iekšā} - \varphi_{virsmā} P_{sat,virsmā}) = -\delta \frac{\partial(\varphi P_{sat})}{\partial x}$$

$$\beta_{ārā} (\varphi_{ārā} P_{sat,ārā} - \varphi_{virsmā} P_{sat,virsmā}) = \delta \frac{\partial(\varphi P_{sat})}{\partial x}$$

kur $\beta_1 = 2.5 * 10^{-8}$, $\beta_2 = 7.5 * 10^{-8}$, $\alpha_1 = 25$, $\alpha_2 = 8$.

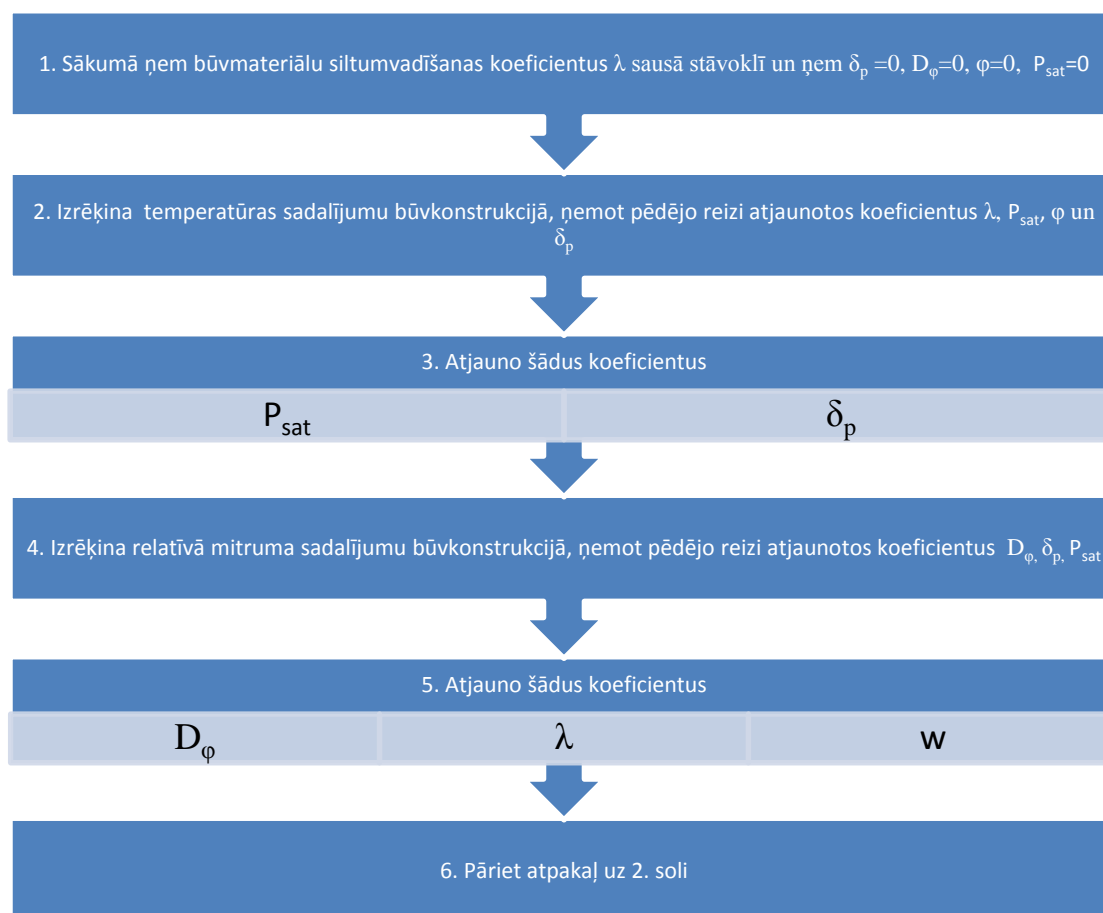
P_{sat} aprēķina šādi:

$$P_{sat} = 610.5 \exp\left(\frac{17.08T}{234.8+T}\right), \text{ ja } T \geq 0$$

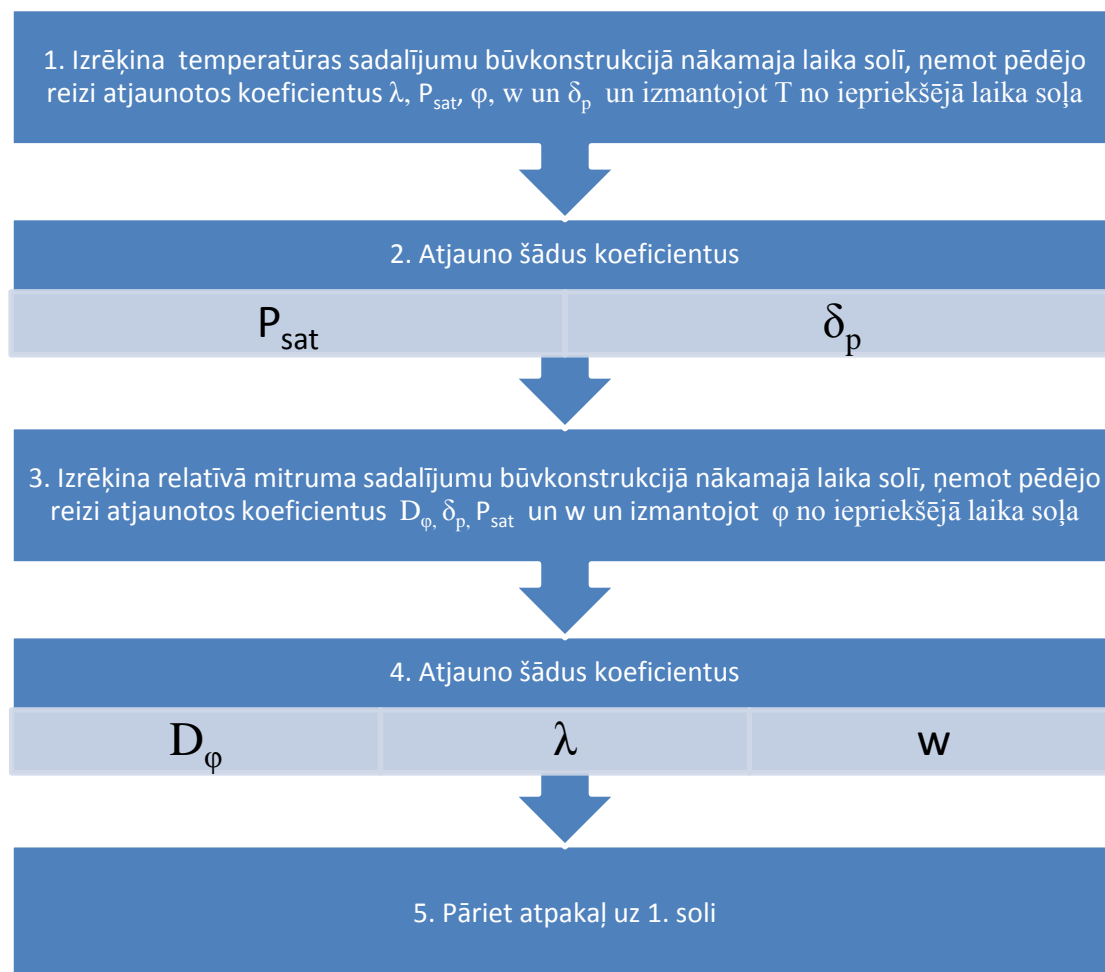
$$P_{sat} = 610.5 \exp\left(\frac{22.44T}{272.44+T}\right), \text{ ja } T < 0$$

3. Programmas stuktūras apraksts

Programma sastāv no 2 daļām. Vienādojumi tiek risināti skaitliski, izmantojot galīgo diferencu shēmu. Vispirms tiek izrēķināti temperatūras un relatīvā mitruma sadalījuma sākuma nosacījumi pēc šādas shēmas



Pēc tam pēc šādas shēmas rēķina nestacionāro gadījumu, kad temperatūra ārā un iekšā mainās laikā



Skaitliskais aprēķins ir implementēts datorprogrammu paketē MATLAB.

4. Īss apraksts, kā strādāt ar programmu

MATLAB kods izstrādātajai programmai šāda vienādojuma risināšanai sniegts

4. Nodaļā. Lai izrēķinātu mitruma un temperatūras sadalījumu būvkonstrukcijā, jārikojas šādi:

- Jāuzdod sekojoši lielumi:
 - **k** ir vektors, kurā ir deklarētie siltumvadīšanas koeficienti, piemēram, ievadīt $k=[0.8 \ 0.04 \ 1]$ nozīmē, ka 1. būvkonstrukcijas slānī silt. vad. koeficients $\lambda = 0.8$, 2. slānī $\lambda = 0.04$, 3. slānī $\lambda = 1$.
 - **d** ir vektors, kurā ir ierakstīti slāņa biezumi (metros).
 - **Cpd** ir vektors, kurā ievadīti īpatnējās siltumietilpības koeficienti katram slānim.
 - **rho** ir vektors, kurā ievadīti blīvumi katram slānim.
 - **mu** ir vektors, kurā ievadīti ūdens tvaika pretestības faktori katram slānim

- **Nsl** ir vektors, kurā ir ievadīti režģa punktu skaits katrā būvkonstrukcijas slānī.
 - **dien** ir dienu skaits, kurā mēs apskatām nestacionāro temperatūras un relatīvo mitruma sadalījumu
 - **dien1** ir laika režģa punktu skaits vienā dienā.
 - **Tii** ir funkcija no laika, kas parāda, kā mainās temperatūra iekšpusē (telpā)
 - **Too** ir funkcija no laika, kas parāda, kā mainās temperatūra ārā, piemēram, $T_{oo} = @t(10 \cdot \cos(2 \cdot \pi \cdot (t - 86400) / 86400))$ nozīmē, ka temperatūra ārā mainās periodiski dienanakts ciklā robežās no +10 līdz -10 grādiem sinusoīdas veidā
 - **rhin** ir relatīvais mitrums iekšā, piemēram, $rh_{in} = 0.5$ nozīmēs, ka relatīvais mitrums iekšā būs 50%.
 - **rhout** ir relatīvais mitrums ārā.
- Tad MATLAB logā jāieraksta:
 - `[r,T]=heat_and_moisture_transfer(k,d,Cpd,rho,mu,Aw,b,wf,Nsl,dien,dien1,Tii,Too,rhin,rhout)`; Pēc aprēķina pabeigšanas programma izdod $Nt \times N$ matricas r un T , kur $r(j,i)$ ir relatīvais mitrums j -tajā laika solī un i -tajā būvkonstrukcijas pozīcijā, bet $T(j,i)$ ir temperatūra j -tajā laika solī un i -tajā būvkonstrukcijas pozīcijā. Ja mēs vēlamies redzēt, kā mainās relatīvais mitruma vai temperatūras sadalījums būvkonstrukcijā laikā, tad MATLAB lodziņā ir vienkārši jāuzraksta *filmina_relativais_mitrums* vai *filmina_temperatura*

5. Aprēķina piemērs

Apskatām šādu būvkonstrukciju (dati ņemti no [3], [4]):

1. Slānis. Kaļķu smilšu apmetums ar slāņa biezumu 2cm, $\lambda = 0.7$, $c = 840$, $\rho = 1600$, $\mu = 15$.
2. Slānis. Akmens vate – 10 cm., $\lambda = 0.036$, $c = 1030$, $\rho = 40$, $\mu = 1$.
3. Slānis. Aeroc 300 putubetons – 30 cm., $\lambda = 0.1$, $c = 850$, $\rho = 425$, $\mu = 4$.
4. Slānis. Kaļķu smilšu apmetums - 2cm (īpašības skat. iepriekš).

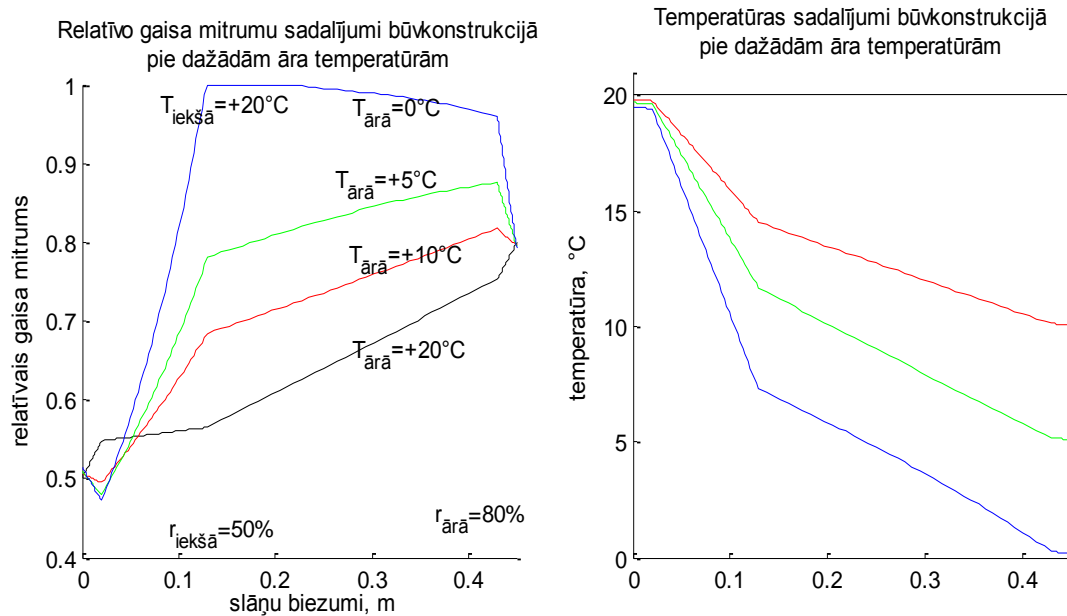
Šai konstrukcijai, izmantojot deklarētās λ vērtības, siltuma caurlaidības U vērtība ir $0.16 \frac{W}{m^2K}$, taču ņemot vērā mitrumu ietekmi uz λ vērtību, stacionārā gadījumā pie āra temperatūras +5 C, iekšā +20C, relatīvā mitruma ārā 80%, iekšā 50%, U vērtība būs $0.20 \frac{W}{m^2K}$.

Efektīvie siltumvadīšanas koeficienti ir šādi:

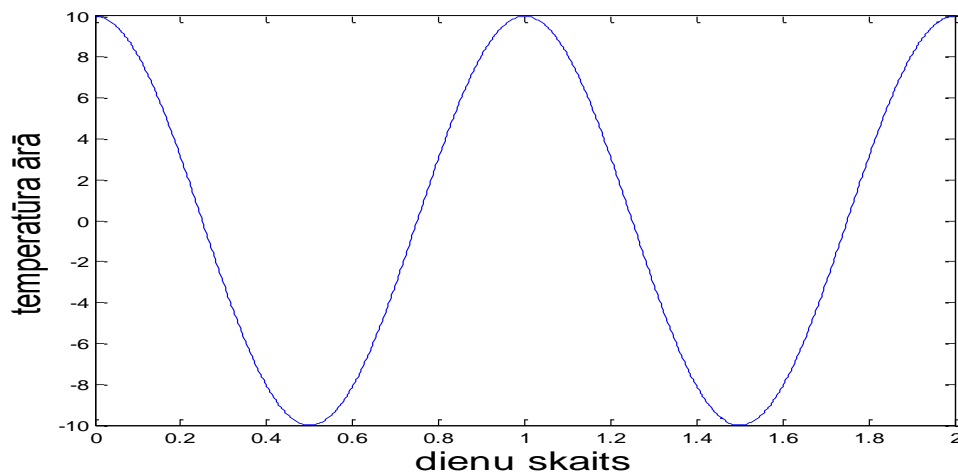
$$\lambda = \lambda_{deklarētais} (1 + 10w/\rho) - \text{kaļķu smilšu apmetumam.}$$

$$\lambda = \lambda_{deklarētais} (1 + 4w/\rho) - \text{Aeroc 300 putubetonam.}$$

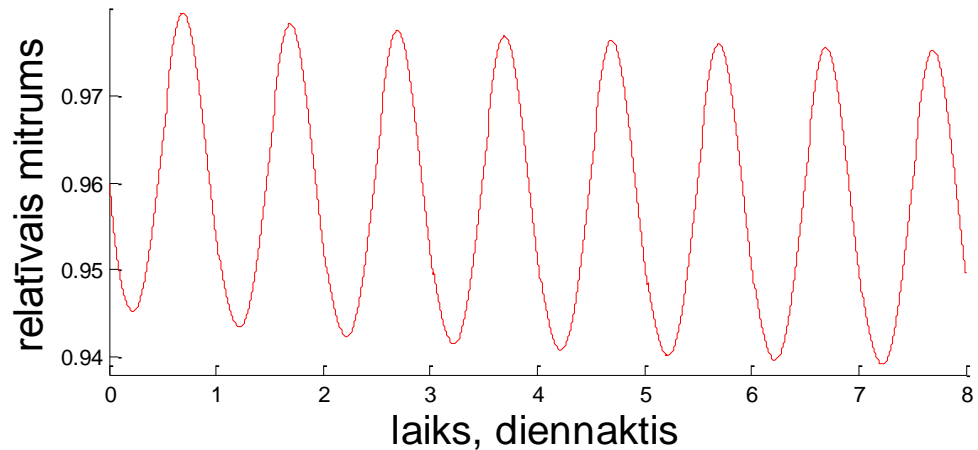
Šinī piemērā ir pieņemts, ka temperatūra iekšā ir visu laiku konstanta $+20^{\circ}\text{C}$ un relatīvais mitrums ārā ir 80%, bet relatīvais mitrums iekšā ir 50%. Galarezultātā ar izstrādāto programmu aprēķinot stāvokli stacionāros gadījumos iegūstam sekojošajā attēlā parādītos grafikus.



Ja apskata nestacionāro gadījumu, tad paņem, ka temperatūra ārā mainās kā harmoniska funkcija (skat. nākamo attēlu).



Nākamais grafiks parāda relatīvā mitruma izmaiņu starpslānī starp aeroc putubetonu un apmetumu ārpusē.



6. Programmas koda teksts

```
function
[r,T]=heat_and_moisture_transfer(k,d,Cpd,rho,mu,Aw,b,wf,Nsl,dien,dien
1,Tii,Too,rhin,rhout);
R=zeros(1,sl+2); hfg=2260000; Cpw=4187;
R(1)=1/8; R(sl+2)=1/25;
for i=2:(sl+1)
    R(i)=d(i-1)/k(i-1);
end;
U=1/sum(R);
hco=1/R(sl+2); hro=0; hri=1/R(1); hci=0; a1=hri+hci; a2=hco;
asol=0; Hsol=0; Hin=0; along=0; Heqp=0;
ysl=zeros(1,sl+1); ysl(1)=0;
for i=2:(sl+1)
    ysl(i)=ysl(i-1)+d(i-1);
end;
t0=0; tb=round(dien*86400); Nt=round(dien1*dien);
t=linspace(0,tb,Nt); dt=(tb-t0)/(Nt-1);
N=sum(Nsl); Nsl1=zeros(1,sl); Nsl1(1)=Nsl(1);
for i=2:sl
    Nsl1(i)=Nsl1(i-1)+Nsl(i);
end;
hh(1)=(ysl(2)-ysl(1))/(Nsl(1)-1);
for i=2:sl
    hh(i)=(ysl(i+1)-ysl(i))/Nsl(i);
end;
h=zeros(1,N-1); h(1:(Nsl(1)-1))=hh(1)*ones(1,Nsl(1)-1);
for f=2:sl
    h((Nsl1(f-1)):(Nsl1(f)-1))=hh(f)*ones(1,Nsl(f));
end;
y(1)=ysl(1);
for i=2:N
    y(i)=y(i-1)+h(i-1);
end;
c=24*3600;
T=zeros(1,N); r=zeros(1,N); perm=zeros(1,N-1);
kk=zeros(1,N-1);
for i=2:(Nsl(1)-1)
    CT(i)=k(1)/h(i);
    AT(i)=k(1)/h(i-1);
```



```

        BT(i)=-(AT(i)+CT(i));
        DT(i)=0;
    end;
    for f=2:s1
        for i=(Ns11(f-1)+1):(Ns11(f)-1)
            CT(i)=k(f)/hh(f);
            AT(i)=k(f)/hh(f);
            BT(i)=-(AT(i)+CT(i));
            DT(i)=0;
        end;
    end;
    for f=1:(s1-1)
        AT(Ns11(f))=k(f)/hh(f); CT(Ns11(f))=k(f+1)/hh(f+1);
        BT(Ns11(f))=-(CT(Ns11(f))+AT(Ns11(f))); DT(Ns11(f))=0;
    end;
    AT(N)=CT(N-1);
    DT(N)=-(hco*To+hro*Tsky+asol*Hsol);
    BT(N)=-(hco+hro+k(s1)/hh(s1));
    CT(N)=0;
    AT(1)=0;
    CT(1)=k(1)/h(1);
    BT(1)=-(CT(1)+hri+hci);
    DT(1)=-asol*Hin-along*Heqp-(hri+hci)*Ti;
    A=diag(BT)+diag(CT(1:(N-1)),1)+diag(AT(2:N),-1);
    T(1:N)=thomas(CT,BT,AT,DT);
    for i=2:(Ns11(1)-2)
        perm(i)=caurl((T(1,i)+T(1,i+1))/2,1);
    end;
    perm(1)=caurl((T(1,1)+T(1,2))/2,1);
    perm(Ns11(1)-1)=caurl((T(1,Ns11(1))+T(1,Ns11(1)-1))/2,1);
    for f=2:s1
        perm(Ns11(f-1))=caurl((T(1,Ns11(f-1))+T(1,Ns11(f-1)+1))/2,f);
        perm(Ns11(f)-1)=caurl((T(1,Ns11(f))+T(1,Ns11(f)-1))/2,f);
        for i=(Ns11(f-1)+1):(Ns11(f)-2)
            perm(i)=caurl((T(1,i)+T(1,i+1))/2,f);
        end;
    end;
    end;
    Ps=zeros(1,N); w=zeros(1,N); Drel=zeros(1,N-1); Pv=zeros(1,N);
    for i=1:N
        Ps(i)=Pvair(T(i));
    end;
    humidity_iekسا=0.5; humidity_ارا=0.8;
    Pv_iekسا=Pvair(Ti)*humidity_iekسا; Pv_ارا=Pvair(To)*humidity_ارا;
    Me1=25*10^(-9); Me2=75*10^(-9);
    Meluzl=Me1/perm(1); Me2uzl=Me2/perm(N-1);
    for i=2:(Ns1(1)-1)
        CR(i)=Ps(i+1)*perm(i)/hh(1)+Drel(i)/hh(1);
        AR(i)=Ps(i-1)*perm(i-1)/hh(1)+Drel(i-1)/hh(1);
        BR(i)=-Ps(i)*(perm(i)/hh(1)+perm(i-1)/hh(1))-...
            Drel(i-1)/hh(1)-Drel(i)/hh(1);
        DR(i)=0;
    end;
    for f=2:s1
        for i=(Ns11(f-1)+1):(Ns11(f)-1)
            CR(i)=Ps(i+1)*perm(i)/hh(f)+Drel(i)/hh(f);
            AR(i)=Ps(i-1)*perm(i-1)/hh(f)+Drel(i-1)/hh(f);
            BR(i)=-Ps(i)*(perm(i)/hh(f)+perm(i-1)/hh(f))-...
                Drel(i-1)/hh(f)-Drel(i)/hh(f);
            DR(i)=0;
        end;
    end;
end;
end;

```

```

    for f=2:s1
        AR(Ns11(f-1))=Ps(Ns11(f-1)-1)*perm(Ns11(f-1)-1)/hh(f-1)+Drel(Ns11(f-1)-1)/hh(f-1);
        CR(Ns11(f-1))=perm(Ns11(f-1))/hh(f)*Ps(Ns11(f-1)+1)+Drel(Ns11(f-1))/hh(f);
        BR(Ns11(f-1))=- (Ps(Ns11(f-1))*perm(Ns11(f-1)-1)/hh(f-1)+...
            Ps(Ns11(f-1))*perm(Ns11(f-1))/hh(f))-...
            (Drel(Ns11(f-1)-1)/hh(f-1)+Drel(Ns11(f-1))/hh(f));
        DR(Ns11(f-1))=0;
    end;
    AR(N)=-perm(N-1)*Ps(N-1)/hh(s1);
    DR(N)=Me2*Pv_ara(1);
    BR(N)=Me2*Ps(N)+perm(N-1)*Ps(N)/hh(s1);
    CR(N)=0;
    AR(1)=0;
    CR(1)=perm(1)*Ps(2)/hh(1);
    BR(1)=- (perm(1)*Ps(1)/hh(1)+Me1*Ps(1));
    DR(1)=-Me1*Pv_ieksa(1);
    r(1,1:N)=thomas(CR,BR,AR,DR);
    for i=1:N
        if r(1,i)>1
            r(1,i)=1;
        end;
    end;
    for i=1:N
        Pv(i)=r(1,i)*Pvair(T(1,i));
    end;
    for i=2:(Ns11(1)-1)
        w(i)=rho(1)*mc(r(1,i),1);
    end;
    for f=2:s1
        for i=(Ns11(f-1)+1):(Ns11(f)-1)
            w(i)=rho(f)*mc(r(1,i),f);
        end;
    end;
    kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,1);
    end;
    kk(1)=lambda_slanis(w(2)-(w(3)-w(2))/2,T(2)-(T(3)-T(2))/2,1);
    kk(Ns11(1)-1)=lambda_slanis(w(Ns11(1)-1)+(w(Ns11(1)-1)-w(Ns11(1)-2))/2,...
        T(Ns11(1)-1)+(T(Ns11(1)-1)-T(Ns11(1)-2))/2,1);
    for f=2:s1
        kk(Ns11(f-1))=lambda_slanis(w(Ns11(f-1)+1)-(w(Ns11(f-1)+2)-w(Ns11(f-1)+1))/2,...
            T(Ns11(f-1)+1)-(T(Ns11(f-1)+2)-T(Ns11(f-1)+1))/2,f);
        kk(Ns11(f)-1)=lambda_slanis(w(Ns11(f)-1)+(w(Ns11(f)-1)-w(Ns11(f)-2))/2,...
            T(Ns11(f)-1)+(T(Ns11(f)-1)-T(Ns11(f)-2))/2,f);
        for i=(Ns11(f-1)+1):(Ns11(f)-2)
            kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,f);
        end;
    end;
    end;
    for i=2:(Ns11(1)-2)
        Drel(i)=diffus((r(1,i)+r(1,i+1))/2,1);
    end;
    Drel(1)=diffus((r(1,1)+r(1,2))/2,1);
    Drel(Ns11(1)-1)=diffus((r(1,Ns11(1))+r(1,Ns11(1)-1))/2,1);
    for f=2:s1
        Drel(Ns11(f-1))=diffus((r(1,Ns11(f-1))+r(1,Ns11(f-1)+1))/2,f);
        Drel(Ns11(f)-1)=diffus((r(1,Ns11(f))+r(1,Ns11(f)-1))/2,f);
    end;

```

```

    for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
        Drel(i)=diffus((r(1,i)+r(1,i+1))/2,f);
    end;
end;
for p=1:50
    for i=2:(Nsl(1)-1)
        CT(i)=kk(i)/hh(1);
        AT(i)=kk(i-1)/hh(1);
        BT(i)=- (AT(i)+CT(i));
        DT(i)=-hfg*(perm(i-1)*(Pv(i-1)-Pv(i))+perm(i)*(Pv(i+1)-
Pv(i)))/hh(1);
    end;
    for f=2:s1
        for i=(Nsl1(f-1)+1):(Nsl1(f)-1)
            CT(i)=kk(i)/hh(f);
            AT(i)=kk(i-1)/hh(f);
            BT(i)=- (AT(i)+CT(i));
            DT(i)=-hfg*(perm(i-1)*(Pv(i-1)-Pv(i))+perm(i)*(Pv(i+1)-
Pv(i)))/hh(f);
        end;
    end;
    for f=1:(s1-1)
        AT(Nsl1(f))=kk(Nsl1(f)-1)/hh(f);
CT(Nsl1(f))=kk(Nsl1(f))/hh(f+1);
        BT(Nsl1(f))=- (CT(Nsl1(f))+AT(Nsl1(f)));
        DT(Nsl1(f))=hfg*(perm(Nsl1(f)-1)*(Pv(Nsl1(f))-Pv(Nsl1(f)-
1))/hh(f)-...
            perm(Nsl1(f))*(Pv(Nsl1(f)+1)-Pv(Nsl1(f)))/hh(f+1));
    end;
    AT(N)=CT(N-1);
    DT(N)=- (hco*To+hro*Tsky+asol*Hsol);
    BT(N)=- (hco+hro+kk(Nsl1(s1)-1)/hh(s1));
    CT(N)=0;
    AT(1)=0;
    CT(1)=kk(1)/hh(1);
    BT(1)=- (CT(1)+hri+hci);
    DT(1)=-asol*Hin-along*Heqp- (hri+hci)*Ti;
    T(1,1:N)=thomas(CT,BT,AT,DT);
    for i=2:(Nsl1(1)-2)
        perm(i)=caurl((T(1,i)+T(1,i+1))/2,1);
    end;
    perm(1)=caurl((T(1,1)+T(1,2))/2,1);
    perm(Nsl1(1)-1)=caurl((T(1,Nsl1(1))+T(1,Nsl1(1)-1))/2,1);
    for f=2:s1
        perm(Nsl1(f-1))=caurl((T(1,Nsl1(f-1))+T(1,Nsl1(f-1)+1))/2,f);
        perm(Nsl1(f)-1)=caurl((T(1,Nsl1(f))+T(1,Nsl1(f)-1))/2,f);
        for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
            perm(i)=caurl((T(1,i)+T(1,i+1))/2,f);
        end;
    end;
    for i=1:N
        Ps(i)=Pvair(T(1,i));
    end;
    Me1=25*10^(-9); Me2=75*10^(-9);
    Meluz1=Me1/perm(1); Me2uz1=Me2/perm(N-1);
    for i=2:(Nsl(1)-1)
        CR(i)=Ps(i+1)*perm(1,i)/h(i)+Drel(1,i)/hh(1);
        AR(i)=Ps(i-1)*perm(1,i-1)/h(i-1)+Drel(1,i-1)/hh(1);
        BR(i)=-Ps(i)*(perm(1,i)/h(i)+perm(i-1)/h(i-1))-...
            Drel(1,i-1)/hh(1)-Drel(1,i)/hh(1);
        DR(i)=0;
    end;

```

```

end;
for f=2:s1
    for i=(Ns11(f-1)+1):(Ns11(f)-1)
        CR(i)=Ps(i+1)*perm(1,i)/hh(f)+Drel(1,i)/hh(f);
        AR(i)=Ps(i-1)*perm(1,i-1)/hh(f)+Drel(1,i-1)/hh(f);
        BR(i)=-Ps(i)*(perm(1,i)/h(i)+perm(i-1)/h(i-1))-...
            Drel(1,i-1)/hh(f)-Drel(1,i)/hh(f);
        DR(i)=0;
    end;
end;
for f=2:s1
    AR(Ns11(f-1))=Ps(Ns11(f-1)-1)*perm(Ns11(f-1)-1)/hh(f-
1)+Drel(Ns11(f-1)-1)/hh(f-1);
    CR(Ns11(f-1))=perm(Ns11(f-1))/hh(f)*Ps(Ns11(f-
1)+1)+Drel(Ns11(f-1))/hh(f);
    BR(Ns11(f-1))=- (Ps(Ns11(f-1))*perm(Ns11(f-1)-1)/hh(f-1)+...
        Ps(Ns11(f-1))*perm(Ns11(f-1))/hh(f))-...
        (Drel(Ns11(f-1)-1)/hh(f-1)+Drel(Ns11(f-1))/hh(f));
    DR(Ns11(f-1))=0;
end;
AR(N)=-perm(N-1)*Ps(N-1)/hh(s1);
DR(N)=Me2*Pv_ara(1);
BR(N)=Me2*Ps(N)+perm(N-1)*Ps(N)/hh(s1);
CR(N)=0;
AR(1)=0;
CR(1)=perm(1)*Ps(2)/hh(1);
BR(1)=- (perm(1)*Ps(1)/hh(1)+Me1*Ps(1));
DR(1)=-Me1*Pv_ieksa(1);
r(1,1:N)=thomas(CR,BR,AR,DR);
rkr=r(Ns11(s1-1));
Pvkr=r(Ns11(s1-1))*Ps(Ns11(s1-1));
for i=1:N
    if r(1,i)>1
        r(1,i)=1;
    end;
end;
end;
for i=1:N
    Pv(i)=r(1,i)*Pvair(T(1,i));
    taupunkts(i)=taupunkt(r(1,i),T(1,i));
end;
for i=2:(Ns11(1)-1)
    w(i)=rho(1)*mc(r(1,i),1);
end;
for f=2:s1
    for i=(Ns11(f-1)+1):(Ns11(f)-1)
        w(i)=rho(f)*mc(r(1,i),f);
    end;
end;
for i=2:(Ns11(1)-2)
    kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,1);
end;
kk(1)=lambda_slanis(w(2)-(w(3)-w(2))/2,T(2)-(T(3)-T(2))/2,1);
kk(Ns11(1)-1)=lambda_slanis(w(Ns11(1)-1)+(w(Ns11(1)-1)-w(Ns11(1)-
2))/2,...
    T(Ns11(1)-1)+(T(Ns11(1)-1)-T(Ns11(1)-2))/2,1);
for f=2:s1
    kk(Ns11(f-1))=lambda_slanis(w(Ns11(f-1)+1)-(w(Ns11(f-1)+2)-
w(Ns11(f-1)+1))/2,...
        T(Ns11(f-1)+1)-(T(Ns11(f-1)+2)-T(Ns11(f-1)+1))/2,f);
    kk(Ns11(f)-1)=lambda_slanis(w(Ns11(f)-1)+(w(Ns11(f)-1)-w(Ns11(f)-
2))/2,...

```

```

        T(Nsl1(f)-1)+(T(Nsl1(f)-1)-T(Nsl1(f)-2))/2,f);
    for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
        kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,f);
    end;
end;
for i=2:(Nsl1(1)-2)
    Drel(i)=diffus((r(1,i)+r(1,i+1))/2,1);
end;
Drel(1)=diffus((r(1,1)+r(1,2))/2,1);
Drel(Nsl1(1)-1)=diffus((r(1,Nsl1(1))+r(1,Nsl1(1)-1))/2,1);
for f=2:sl
    Drel(Nsl1(f-1))=diffus((r(1,Nsl1(f-1))+r(1,Nsl1(f-1)+1))/2,f);
    Drel(Nsl1(f)-1)=diffus((r(1,Nsl1(f))+r(1,Nsl1(f)-1))/2,f);
    for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
        Drel(i)=diffus((r(1,i)+r(1,i+1))/2,f);
    end;
end;
end;
end;

return
cond=0; SdT=sum(mu.*d); Sdc=sum(d(1:(sl-1)).*mu(1:(sl-1)));
Pvkr=r(Nsl1(sl-1))*Ps(Nsl1(sl-1));
cond=86400*2*10^(-10)*((Pv_iekasa-Pvkr)/(SdT-Sdc)-(Pvkr-Pv_ara)/Sdc);
figure(2); hold on;
subplot(1,2,1); plot(y,T,':k');
rkr=zeros(Nt,1); rkr(1)=r(Nsl1(sl-1));
AT=zeros(1,N); BT=zeros(1,N); CT=zeros(1,N); DT=zeros(1,N);
AR=zeros(1,N); BR=zeros(1,N); CR=zeros(1,N); DR=zeros(1,N);

for j=2:Nt
    j
    To=aratem(t(j));
    for i=2:(Nsl(1)-1)
        DHDv=w(i)*Cpw+rho(1)*Cpd(1);
        CT(i)=kk(i)/hh(1);
        AT(i)=kk(i-1)/hh(1);
        BT(i)=- (AT(i)+CT(i)+DHDv*hh(1)/dt);
        DT(i)=-DHDv*hh(1)/dt*T(1,i)-hfg*(perm(i-1)*...
            (Pv(i-1)-Pv(i))+perm(i)*(Pv(i+1)-Pv(i)))/hh(1);
    end;
    for f=2:sl
        for i=(Nsl1(f-1)+1):(Nsl1(f)-1)
            DHDv=w(i)*Cpw+rho(f)*Cpd(f);
            CT(i)=kk(i)/hh(f);
            AT(i)=kk(i-1)/hh(f);
            BT(i)=- (AT(i)+CT(i)+DHDv*hh(f)/dt);
            DT(i)=-DHDv*hh(f)/dt*T(1,i)-hfg*(perm(i-1)*...
                (Pv(i-1)-Pv(i))+perm(i)*(Pv(i+1)-Pv(i)))/hh(f);
        end;
    end;
    for f=1:(sl-1)
        AT(Nsl1(f))=kk(Nsl1(f)-1)/hh(f);
        CT(Nsl1(f))=kk(Nsl1(f))/hh(f+1);
        BT(Nsl1(f))=- (CT(Nsl1(f))+AT(Nsl1(f)));
        DT(Nsl1(f))=hfg*(perm(Nsl1(f)-1)*(Pv(Nsl1(f))-Pv(Nsl1(f)-
1))/hh(f)-...
            perm(Nsl1(f))*(Pv(Nsl1(f)+1)-Pv(Nsl1(f)))/hh(f+1));
    end;
    AT(N)=CT(N-1);
    DT(N)=- (hco*To+hro*Tsky+asol*Hsol);
    BT(N)=- (hco+hro+kk(N-1)/hh(sl));

```

```

CT(N)=0;
AT(1)=0;
CT(1)=kk(1)/hh(1);
BT(1)=-(CT(1)+hri+hci);
DT(1)=-asol*Hin-along*Heqp-(hri+hci)*Ti;
T(1,1:N)=thomas(CT,BT,AT,DT);
for i=2:(Nsl1(1)-2)
    perm(i)=caurl((T(1,i)+T(1,i+1))/2,1);
end;
perm(1)=caurl((T(1,1)+T(1,2))/2,1);
perm(Nsl1(1)-1)=caurl((T(1,Nsl1(1))+T(1,Nsl1(1)-1))/2,1);
for f=2:s1
    perm(Nsl1(f-1))=caurl((T(1,Nsl1(f-1))+T(1,Nsl1(f-1)+1))/2,f);
    perm(Nsl1(f)-1)=caurl((T(1,Nsl1(f))+T(1,Nsl1(f)-1))/2,f);
    for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
        perm(i)=caurl((T(1,i)+T(1,i+1))/2,f);
    end;
end;
for i=1:N
    Ps(i)=Pvair(T(i));
end;
Pv_iekسا=Pvair(Ti)*humidity_iekسا; Pv_ara=Pvair(To)*humidity_ara;
%Tagad meklejam relativno mitrumu
for i=2:(Nsl(1)-1)
    DwDr=rho(1)*mc_atv(r(1,i),1);
    CR(i)=(Ps(i+1)*perm(i)/hh(1)+Drel(i)/hh(1));
    AR(i)=(Ps(i-1)*perm(i-1)/hh(1)+Drel(i-1)/hh(1));
    BR(i)=(-Ps(i)*(perm(i)/hh(1)+perm(i-1)/hh(1))-...
        Drel(i-1)/hh(1)-Drel(i)/hh(1)-hh(1)*DwDr/dt);
    DR(i)=-DwDr*hh(1)/dt*r(1,i);
end;
for f=2:s1
    for i=(Nsl1(f-1)+1):(Nsl1(f)-1)
        DwDr=rho(f)*mc_atv(r(1,i),f);
        CR(i)=(Ps(i+1)*perm(i)/hh(f)+Drel(i)/hh(f));
        AR(i)=(Ps(i-1)*perm(i-1)/hh(f)+Drel(i-1)/hh(f));
        BR(i)=(-Ps(i)*(perm(i)/hh(f)+perm(i-1)/hh(f))-...
            Drel(i-1)/hh(f)-Drel(i)/hh(f)-hh(f)*DwDr/dt);
        DR(i)=-DwDr*hh(f)/dt*r(1,i);
    end;
    AR(Nsl1(f-1))=(Ps(Nsl1(f-1)-1)*perm(Nsl1(f-1)-1)/hh(f-1))+...
        Drel(Nsl1(f-1)-1)/hh(f-1);
    CR(Nsl1(f-1))=(perm(Nsl1(f-1))/hh(f)*Ps(Nsl1(f-1)+1))+...
        Drel(Nsl1(f-1))/hh(f);
    BR(Nsl1(f-1))=(-Ps(Nsl1(f-1))*perm(Nsl1(f-1)-1)/hh(f-1)+...
        Ps(Nsl1(f-1))*perm(Nsl1(f-1))/hh(f))-...
        (Drel(Nsl1(f-1)-1)/hh(f-1)+Drel(Nsl1(f-1))/hh(f));
    DR(Nsl1(f-1))=0;
end;
AR(N)=(-perm(N-1)*Ps(N-1)/hh(s1));
DR(N)=(Me2*Pv_ara);
BR(N)=(Me2*Ps(N)+perm(N-1)*Ps(N)/hh(s1));
CR(N)=0;
AR(1)=0;
CR(1)=(perm(1)*Ps(2)/hh(1));
BR(1)=(-perm(1)*Ps(1)/hh(1)+Me1*Ps(1));
DR(1)=(-Me1*Pv_iekسا);
r=thomas(CR,BR,AR,DR);
Pvkr=r(Nsl1(s1-1))*Ps(Nsl1(s1-1));
for i=1:N
    if r(i)>1

```

```

        r(i)=1;
    end;
end;
Pv=r.*Ps;
w(2:(Nsl1(1)-1))=rho(1)*mc(r(2:(Nsl1(1)-1)),1);
for f=2:sl
    w((Nsl1(f-1)+1):(Nsl1(f)-1))=rho(f)*mc(r((Nsl1(f-1)+1):(Nsl1(f)-1)),f);
end;
for i=2:(Nsl1(1)-2)
    kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,1);
end;
kk(1)=lambda_slanis(w(2)-(w(3)-w(2))/2,T(2)-(T(3)-T(2))/2,1);
kk(Nsl1(1)-1)=lambda_slanis(w(Nsl1(1)-1)+(w(Nsl1(1)-1)-w(Nsl1(1)-2))/2,...
    T(Nsl1(1)-1)+(T(Nsl1(1)-1)-T(Nsl1(1)-2))/2,1);
for f=2:sl
    kk(Nsl1(f-1))=lambda_slanis(w(Nsl1(f-1)+1)-(w(Nsl1(f-1)+2)-w(Nsl1(f-1)+1))/2,...
    T(Nsl1(f-1)+1)-(T(Nsl1(f-1)+2)-T(Nsl1(f-1)+1))/2,f);
    kk(Nsl1(f)-1)=lambda_slanis(w(Nsl1(f)-1)+(w(Nsl1(f)-1)-w(Nsl1(f)-2))/2,...
    T(Nsl1(f)-1)+(T(Nsl1(f)-1)-T(Nsl1(f)-2))/2,f);
    for i=(Nsl1(f-1)+1):(Nsl1(f)-2)
        kk(i)=lambda_slanis((w(i+1)+w(i))/2,(T(i+1)+T(i))/2,f);
    end;
end;

    rkr(j)=r(1,Nsl1(sl-1));
end;
function p=caurl(T,f);
global mu
    p=1/mu(f)*2*10^(-7)*(T+273.15)^(0.81)/101325;
function d=diffus(r,f);
global Aw wf rho
w=mc(r,f);
d=rho(f)*mc_atv(r,f)*3.8*(Aw(f)/(wf(f)*rho(f)))^2*1000^(w/(wf(f)*rho(f))-1);

```

7. Apzīmējumu saraksts

λ – būvmateriāla siltumvadīšanas koeficients reālos apstākļos;

λ_d – siltumvadīšanas koeficients sausam būvmateriālam;

T – temperatūra ;

x – garums;

φ – relatīvais gaisa mitrums;

δ – būvmateriāla ūdens tvaiku caurlaidība ;

w – ūdens daudzums tilpuma vienībā (kg/m^3);

D_φ – šķidrums vadītspējas koeficients;

P_{sat} - tvaiku piesātinājuma spiediens;

t – laiks;

h_v – iztvaikošanas siltuma koeficients;

μ – materiālu ūdens tvaiku pretestības faktors ;

ρ – būvmateriāla blīvums;

c – būvmateriāla īpatnējā siltumietilpība;

c_w – īpatnējā siltumietilpība ūdenim;

8. Izmantotā literatūra

1. Zhipeng Zhong, James E. Braun (2008) Combined heat and moisture transport modelling for residential buildings.
2. Hartwig M. Kunzel (1995) Simultaneous Heat and Moisture transport in Building Components.
3. Peter Haupl (2008) Bauphysik. Klima Warme Feuchte Schall. Grundlagen, Anwendungen, Beispiele. Ernst & Sohn.
4. Noteikumi par Latvijas būvnormatīvu LBN 002-01 "Ēku norobežojošo konstrukciju siltumtehnika"