HHSIM - a computer program for simulation of unsteady-state sorption measurements

Wadsö, Lars

1994

Citation for published version (APA):
Wadsö, L. (1994). *HHSIM - a computer program for simulation of unsteady-state sorption measurements*. (Report TVBM (Intern 7000-rapport); Vol. 7079). Division of Building Materials, LTH, Lund University.

Total number of authors:
1

LUNDS TEKNISKA HÖGSKOLA
Byggnadsmaterial

UNIVERSITY OF LUND
LUND INSTITUTE OF TECHNOLOGY
Division of Building Materials

HHSIM - a computer program for simulation of unsteady-state sorption measurements

Lars Wadsö

Rapport TVBM-7079

Lund, juni 1994

HHSIM - a computer program for simulation of unsteady-state sorption measurements

Lars Wadsö, Building Materials, Lund University, Box 118, S-221 00 Lund, Sweden

Introduction

In the science of wood-water relations, one common experiment is to observe the weight change of a wood sample after a step change in relative humidity. The result of such a measurements is often used to calculate the diffusivity of the wood material. It is, however, impossible to make such sorption measurements in which the rate of sorption is not also governed by other processes than diffusion.

A sorption measurement is made in a measurement chamber. In this chamber one tries to produce a step change in relative humidity. This can be by flowing humidified air through the chamber or by having saturated salt solutions in the chamber. The samples in the chamber are weighed regularly to obtain the weight changes following the change in relative humidity.

The computer program HHSIM described in this paper is a simple tool for studying sorption processes governed by:

1 diffusion in the sample

2 surface mass transfer

3 non-perfect steps in relative humidity

4 heat of sorption

Some processes *not* included in the present model are:

1 diffusion of heat in the sample (the sample is assumed to have one temperature, which differs somewhat from the chamber temperature due to the heat of sorption. The assumption os uniform sample temperature is valid for normal wood samples as the heat diffusivity is much higher than the moisture diffusivity)

2 temperature differences in the chamber (the air and the chamber walls are assumed to have the same constant temperature)

3 relative humidity differences in the chamber (complete mixing is assumed)

4 time dependent sorption behavior ("non-Fickian" or "anomalous" sorption)

The problem

Mass equations

Figure 1 shows the mass balance for water vapor in the measuring chamber. The flow of moisture into and out of the chamber is (the nomenclature is given on the next page):

$$F_{\mathrm{in}} = f \cdot v_{\mathrm{f}} \tag{1}$$

$$F_{\mathrm{out}} = f \cdot v_{\mathrm{c}} \tag{2}$$

The flow of moisture to the sample from the chamber air is (cf. next section):

$$F_{\mathrm{sorp}} = k_v (v_{\mathrm{c}} - v_{\mathrm{s}}) A \tag{3}$$

The mass balance is then:

$$\frac{\mathrm{d}v_{\mathrm{c}}}{\mathrm{d}t} = \frac{F_{\mathrm{in}} - F_{\mathrm{out}} - F_{\mathrm{sorp}}}{V_{\mathrm{c}}} \tag{4}$$

The mass flow from the air to the sample surface is governed by Eq. 3 above. The vapor content of the surface is given by:

$$v_{\mathrm{s}} = v_{\mathrm{sat}}(T_{\mathrm{s}}) \phi(m_{\mathrm{s}}, T_{\mathrm{s}}) \tag{5}$$

Here, $\phi(m, T)$ is the sorption isotherm.

Nomenclature

A	total exposed sample area	$\mathrm{m^2}$
c	moisture concentration in wood	$\mathrm{kg/m^3}$
c_p	heat capacity of wood	$\mathrm{J/(kg\ K)}$
D_c	diffusivity in wood with c as potential	$\mathrm{m^2/s}$
f	flow of air through chamber	$\mathrm{m^3/s}$
F	moisture flux	kg/s
h	surface heat transfer coefficient	$\mathrm{J/(m^2\ s\ K)}$
H	differential heat (entalphy) of sorption	$\mathrm{J/kg_{water}}$
k_v	surface mass transfer coefficient	m/s
ℓ	half sample thickness	m
m	moisture content	kg/kg
M	dry mass of sample	kg
n	number of computational cells	
Q	heat flux	J/s
T	temperature	K
v	vapor content of air	$\mathrm{kg/m^3}$
V_c	chamber volume	$\mathrm{m^3}$
Δx	cell size for the calculation of diffusion	m
Δt	time step in simulation	s
ϕ	relative humidity	Pa/Pa
ρ	wood dry density	$\mathrm{kg/m^3}$

Indices

i	initial (before step)
f	final (equilibrium after infinite time)
c	measurement chamber
s	sample surface
conv	convection
rad	radiation
sorp	sorption
sat	saturation

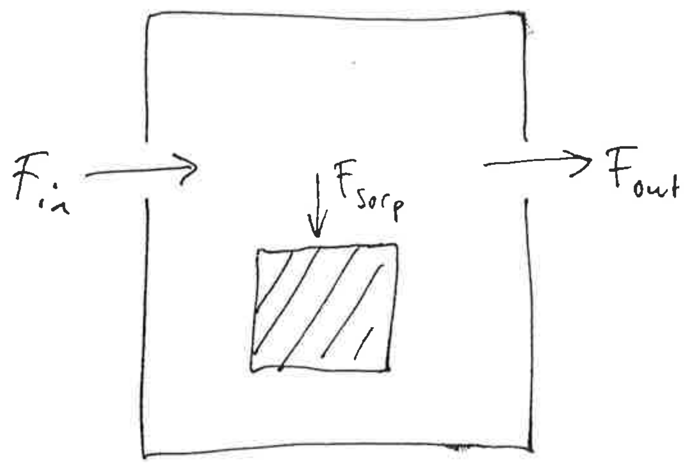


Figure 1. The mass balance of the measuring chamber

The mass flow in the sample is governed by Fick's law:

$$F = D_c \frac{\mathrm{d}c}{\mathrm{d}t} A \tag{6}$$

Heat equations

Figure 2 shows the heat balance of the sample. The temperature change of the sample is the result of the heat of sorption:

$$Q_{\mathrm{sorp}} = F_{\mathrm{sorp}} H \tag{7}$$

Temperature differences in the system are evened out by convection and radiation:

$$Q_{\mathrm{conv}} = h_{\mathrm{conv}}(T_{\mathrm{s}} - T\{INc)A \tag{8}$$

$$Q_{\mathrm{rad}} = h_{\mathrm{rad}}(T_{\mathrm{s}} - T_{\mathrm{c}})A \tag{9}$$

The heat balance for the sample is then:

$$\frac{\mathrm{d}T_{\mathrm{s}}}{\mathrm{d}t} = \frac{Q_{\mathrm{sorp}} - Q_{\mathrm{conv}} - Q_{\mathrm{rad}}}{c_p M} \tag{10}$$

The simulation method

The program uses the method of explicit forward finite differences, i.e. the actual physical equations are calculated at each time step with the derivatives replaced with finite differences.

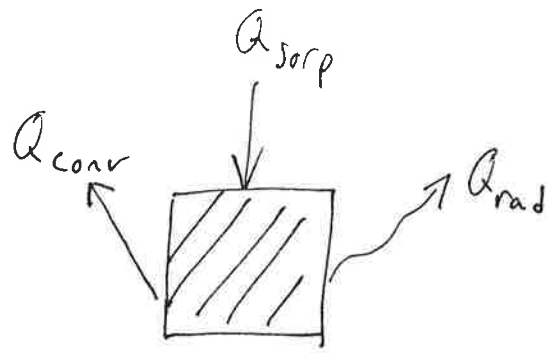


Figure 2. The heat balance of the sample

The mass balance for the chamber is then:

$$\Delta v_{\mathrm{c}} = \frac{F_{\mathrm{in}} - F_{\mathrm{out}} - F_{\mathrm{sorp}}}{V_{\mathrm{c}}} \Delta t \tag{11}$$

In the forward finite difference method, Fick's law of diffusion is:

$$\Delta c_1 = (\frac{F_{\mathrm{sorp}}}{A} - D_c \frac{c_1 - c_2}{\Delta x}) \frac{\Delta t}{\Delta x} \tag{12}$$

$$\Delta c_i = D_c (\frac{c_{i-1} - c_i}{\Delta x} - \frac{c_i - c_{i+1}}{\Delta x}) \frac{\Delta t}{\Delta x} \tag{13}$$

$$\Delta c_n = D_c \frac{c_{n-1} - c_n}{\Delta x} \frac{\Delta t}{\Delta x} \tag{14}$$

Here, indices 1, n and i denote the surface cell, the center cell, and any other cell.

The heat balance is:

$$\Delta T_{\mathrm{s}} = \frac{Q_{\mathrm{sorp}} - Q_{\mathrm{conv}} - Q_{\mathrm{rad}}}{c_p M} \Delta t \tag{15}$$

The computer code

The computer code (given in Appendix A) is written in MATLAB 4.0. The structure of the program is:

- The main program hhsim.m is a MATLAB-function which is called by the following command:

  ```
  [t,Ts,vc,c]=hhsim(rhi,rhf,Tc,tend);
  ```

  The input parameters are:

`rhi`	ϕ_i	initial RH
`rhf`	ϕ_f	final RH
`Tc`	T_c	chamber temperature (°C)
`tend`		max simulation time

If only two input arguments are given T_c=20°C. If only two or three input arguments are given, there is no max simulation time. In such a case the simulation proceeds until:

$$\frac{c_n - c_\mathrm{i}}{c_\mathrm{f} - c_\mathrm{i}} > 0.999 \tag{16}$$

Here, c_n is the value of c is the center cell of the simulation. If a value of `tend` is given, the simulation still stops if 99,9% of the sorption process is completed.

The output parameters are:

`t`	t	time
`Ts`	$T_\mathrm{s} - T_\mathrm{c}$	sample temperature change
`vc`	v_c	chamber vapor content
`c`	c	sample concentration profiles

The number and times of the outputs are determined by the factors `tutstart` and `tutinc` in hhsim.m.

- Input data for the measurement chamber and the sample is given in a data-file hhdat.m.

- A number of function are called on to give values of:
  - diffusivity (m2dcl.m or m2dctr.m)
  - heat capacity (m2cp.m)
  - sorption enthalpy (m2hs.m)
  - the sorption isotherm (m2fi.m and fi2m.m)

All programs are listed in Appendix A.

All parameters are taken as constant during each simulation. Some of them (f, k_v, h_conv, h_rad, ℓ, A, M, ρ and V_c) are constant, but for some parameters (D_c, c_p and H) a mean value is calculated. It is possible to move this calculation into the iterative loop, but this would slow down the execution of the program, without improving the result in most cases (one exception may be simulations in very large ϕ-intervals).

The sources from which the data for the calculation of the sorption isotherm, D_c, c_p and H were taken, are given in respective function-program (Appendix A).

Choosing the time-step

One of the main problems in running a program simulating many different processes simultaneously is to find efficient time-steps (Δt) for the program. The time-step should be the largest possible which still gives the correct result. In the present program, an optimal time-step is calculated every tenth iteration.

We have found the following empirical equation to give a good time-step for the mass balance of the chamber (a deeper analysis of the problem could certainly give a better expression than this):

$$\Delta t = \frac{1}{\sqrt{\alpha(v_\mathrm{f} - v_\mathrm{c}) - \beta(v_\mathrm{c} - v_\mathrm{s})}} \tag{17}$$

Here, α and β are given by:

$$\alpha = \frac{f}{qV_\mathrm{c}(v_\mathrm{f} - v_\mathrm{i})} \tag{18}$$

$$\beta = \frac{Ak_v}{qV_\mathrm{c}(v_\mathrm{f} - v_\mathrm{i})} \tag{19}$$

Here, q is a factor which normally is given the value 0,0005. The above equations gives a limitation on the change in v_c during a time-step.

A largest possible time step of the diffusion process in the sample is given by:

$$\Delta t = (\Delta x)^2 D_c \tag{20}$$

The smallest of the above time-steps have to be used. Usually, the Δt calculated by Eq. 20

is higher than the Δt calculated by Eq. 19 during the first part of the process. Typically Δt starts at 0,2 s and then increases to 2 s (given by Eq. 20).

Options

There are some options that are built in to the program to make it more versatile:

- Instead of giving f and V_c it is possible to give ϕ_c as a function of time (see hhdat.m).
- It is possible to chose between diffusion in longitudinal (m2dcl) or transversal (m2dctr) directions (see hhdat.m).
- It is possible to chose between a number of different sorption isotherms (see hhdat.m and the isotherm functions).

Verification of HHSIM

A number of measures have been taken to assure that the program is functioning as it should. These include a sensitivity analysis, and running the program in extreme cases so that it is possible to study only one process a time. We check our program with the following four tests after each major change in the program.

The chamber

If k_v is made extremely small, the sample absorbs only negligible amounts of water until v_c (the vapor content of the chamber) has changed from v_i to v_f. The differential equation will then be:

$$\frac{\mathrm{d}v_c}{\mathrm{d}t} = \frac{f}{V_c}(v_f - v_c) \qquad v_c(0) = v_i \qquad (21)$$

The equation for v_c after a step in the vapor content of the incoming air is then:

$$v_c = (v_i - v_f)e^{-ft/V_c} + v_f \qquad (22)$$

As the result of a run with k_v=1·10^{-12} gave the above result with more than four figures correct, this part of the program functions well.

The sample

If ℓ is made very high, the diffusion in the sample will govern the sorption. This was tested with ℓ=0,1 m and D_c=600·10^{-12}m^2/s. The result was very close to the solution to the diffusion problem given on p. 48 in Crank (1975):

$$E(t) = 1 - \frac{8}{\pi^2}\sum_{n=0}^{\infty}\left[\frac{1}{(2n+1)^2}\cdot\right.$$
$$\left.\cdot\exp\left(\frac{-(2n+1)^2\pi^2 D_c t}{4\ell^2}\right)\right] \qquad (23)$$

Surface mass transport

It is possible to make the sorption process be completely governed by the surface mass transport coefficient k_v by increasing f and D_c and letting H=0. The appropriate differential equation is:

$$\frac{\mathrm{d}v_s}{\mathrm{d}t} = \frac{k_v A}{M\xi}(v_f - v_s) \qquad v_s(0) = v_i \qquad (24)$$

Here, ξ is the slope of the sorption isotherm divided by v_{sat}:

$$\xi = \frac{\mathrm{d}m}{\mathrm{d}v} = \frac{1}{v_{sat}}\frac{\mathrm{d}m}{\mathrm{d}\phi} \qquad (25)$$

The solution to this problem is:

$$v_s = (v_i - v_f)e^{-\frac{k_v A}{M\xi}t} + v_f \qquad (26)$$

We have obtained the result from Eq. 26 by running one of our standard cases, only increasing f and D_c by a factor of 100 and 10, respectively, and making H=0 (k_v was 0,005 m/s).

Testing the heat effect

We have tested the heat effect by checking that we get a similar change in sample temperature as was obtained in a measurement.

We used the measurements called case 1 and 2 in Wadsö (1994).

The temperature change is a function only of the rate of sorption, the differential heat of sorption (H), the sample mass (M) and heat capacity (c_p), and the surface heat transfer coefficients for convection (h_{conv}) and radiation (h_{rad}). The change in sample temperature should be similar between the measurement and the simulation if the rate of sorption is the same as in a measurement and H, M, c_p, h_{conv} and h_{rad} are correctly given.

Sample results

Appendix B contains results from simulations with HHSIM. These simulations were made to investigate the anomalous sorption behavior found in sorption measurements on wood by Håkansson (1994). The present program is a valuable tool as it makes it possible to prove that the sorption is anomalous (i.e. that the measured sorption curves cannot be explained by the processes which the program simulates).

References

Crank, J. (1975)
The mathematics of diffusion, Oxford University Press, Oxford

Håkansson, H. (1994)
Experimental studies of transient sorption on the cell walls of wood (in Swedish) Dept. of Building Sci. Lund University, Lund, Sweden (in press)

Wadsö, L. (1994)
An error analysis of the sorption method. Part 2. *Holzforschung* 48:2 133–138

940613

```
function [tut,Tsut,vcut,cut]=hhsim(rhi,rhf,Tin,tend);
%rhi=0.5;rhf=0.60;Tin=20;tend=2000000;nargin=4;
%  [t,Ts,vc,c]=hhsim(rhi,rhf,Tin,tend);
% A SIMULATION PROGRAM FOR REALISTIC SORPTION EXPERIMENTS
% Lars Wadsö 940613
% input:  rhi=initial rh [Pa/Pa]
%         rhf=final rh [Pa/Pa]
%         T [deg Celsius]
%         tend=max time of simulation
% output: t=time [s]
%         Ts=sample T - chamber T [K]
%         vc=chamber vapor content [kg/m3]
%         c=sample concentration matrix [kg/m3]
% simulation of sorption governed by
%    1. internal diffusion
%    2. surface mass transfer coeff.
%    3. heat of sorption
%    4. limited rate of added/removed moisture

if nargin<3; Tin=20; end    %default: 20 C
if nargin<4; tend=1e15; end %default: run simulation ...
                      %until (c(t)-ci)/(cf-ci)=0.999
tid=clock;
clear tut Tsut vcut cut dtut
T0=273.15;

hhdat  %in module hhdat values are given to the following parameters:
%      --------------------------------------------------------
%      ell   half sample thickness (dry sample) [m]
%                      (sorption from both open surfaces)
%      A     total exposed area  (dry sample) [m2]
%      rho   density [kg/m3]
%      vccrit=1 : f and Vc given
%               f     flow (with vf) into/out of chamber [m3/s]
%               Vc    chamber volume [m3]
%      vccrit=2 : [tin rhin] given
%          (vector with time and rh (linear interpolation between points))
%      kv    surface mass transfer coefficient with vapor content as pot. [m/s]
%      hconv  convection heat transfer coefficient [W/(m2K)]
%      hrad   radiation heat transfer coefficient [W/(m2K)]
%      flowdir=1 : longitudinal diffusion
%      flowdir=2 : transversal diffusion
%      n     no of computational cells
%      --------------------------------------------------------

%-----SAMPLE DATA------
 V=ell*A;  %sample volume [m3]
 M=rho*V;  %sample mass [kg]

%------STEP DATA------
Tc=T0+Tin;          %chamber temperature [K]
 vsat=t2vsat(Tc);  %chamber saturation vapor content [kg/m3]
 vi=vsat*rhi;      %initial vapor content in chamber and sample [kg/m3]
 vf=vsat*rhf;      %final vapor content in chamber and sample [kg/m3]
 ci=rho*fi2m(rhi,isotherm,Tc); %initial concentration in sample [kg/m3]
 cf=rho*fi2m(rhf,isotherm,Tc); %final concentration in sample [kg/m3]
 cmean=(ci+cf)/2;   %mean concentration in sample [kg/m3]
 mmean=cmean/rho;   %mean moisture content in sample [kg/kg]

%------CONC. DEP. PARAM.------
 if flowdir==1
```

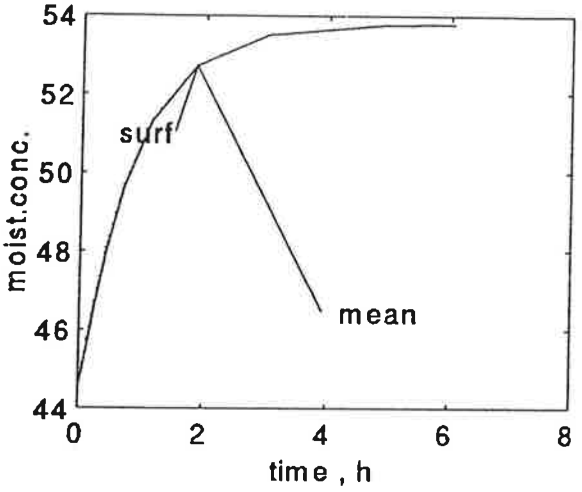
```
      Dc=m2dcl(mmean,Tc-T0);  %longitudinal diffusivity with
                        %concentration as potential [m2/s]
    elseif flowdir==2
      Dc=m2dctr(mmean,Tc-T0); %transversal diffusivity with
                        %concentration as potential [m2/s]
    end
    cp=m2cp(mmean,Tc);  %heat capacity [J/(kgK)]
    Hs=m2hs(mmean);      %differential heat of sorption [J/kg(water)]

  %------RUN PARAM. (TIME)------    (is seconds!)
  tstart=0;   %time at start of simulation [s]
          %max end time (tend) is given as input [s]
  %cend=ci+0.999*(cf-ci);  %end concentration of center cell [kg/m3]
  tutstart=60;   %first output [s]
  tutinc=1.6;    %time increment t(new)=t(old)+tutstart^(no_of_outputs)

  %------RUN PARAM. (OTHER)------
   dx=ell/n;  %cell size [m]

  %------START VALUES------
   vc=vi;     %chamber start vapor content [kg/m3]
   c=ci*ones([1,n]);  %sample start concentration [kg/m3]
   Ts=Tc;        %sample start temperature [K]
   vs=vi;     %sample surface start vapor content [kg/m3]

  %------INITIATION!!------
  slut=0;  %slut==1 --> stop!
  t=tstart;
  K1=kv*A;  %K1, K2, K4, K5 and K6 are used in the simulation
  K2=Dc/dx;
  K4=Dc/dx/dx;
  K5=(hconv+hrad)*A;
  K6=cp*M;
  tnextut=tutstart;
  i=0; %no. of iterations
  ii=0; %mod(i,10), for change of time-step
  j=1; %no. of outputs
  tut=zeros([1,100]);  %creation of arrays
  Tsut=tut;
  vcut=tut;
  cut=zeros([n,100]);
  dtut=tut;
  tut(1)=0;   %giving correct start values to output arrays
  Tsut(1)=Ts;
  vcut(1)=vc;
  cut(:,1)=c(:);

  if vccrit==1
    q=0.0005/5;              %q is an empirical factor in the calculation of dt
    alpha=f/q/Vc/(vf-vi);    %alpha is used to calculate dt
    beta=A*kv/q/Vc/(vf-vi);  %beta is used to calculate dt
    dt=1/sqrt(alpha*(vf-vc)-beta*(vc-vs));   %calculation of dt for dtut
  elseif vccrit==2
    vcin=vsat*rhin;
    dvdc=(vf-vi)/(cf-ci);
    dxkv=Dc/(kv*dvdc);       %equivalent thickness of wood corresponding to kv
    dxused=min([dxkv dx]);  %dx used to calculate dt
    dt=0.5*dxused^2/Dc;
    k=1; %counter for vc determination
  end
  dtut(1)=dt;    %start value to output array dtut
```

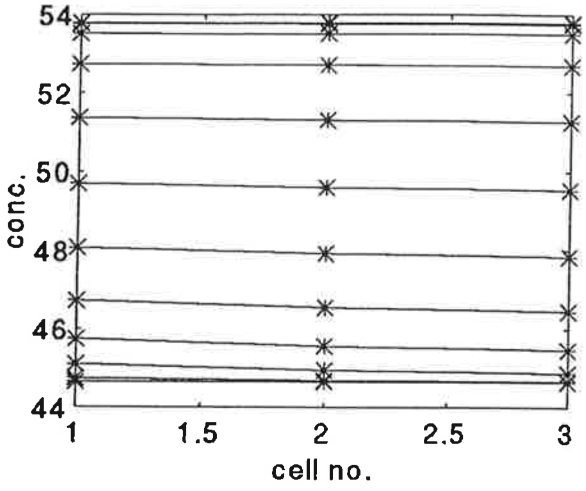
```
    %-------OUTPUT OF INPUT---------
    figure(3)
    clf
    text(0.1,0.95,['ell=',num2str(ell),' m; A=',num2str(A),' m2'])
    text(0.1,0.91,['rho=',num2str(rho),' kg/m3; M=',num2str(M),' kg; n=',int2str(n)])
    text(0.1,0.87,['Tc=',num2str(Tc-T0),' C; rhi=',num2str(rhi),' rhf=',num2str(rhf)])
    text(0.1,0.83,['vi=',num2str(vi),' kg/m3; vf=',num2str(vf)])
    text(0.1,0.79,['ci=',num2str(ci),' kg/m3; cf=',num2str(cf)])
    text(0.1,0.75,['Dc=',num2str(Dc),'m2/s; cp=',num2str(cp),' J/(kgK); Hs=' ...
            ,num2str(Hs),' J/kg'])
    text(0.1,0.71,['f=',num2str(f),' m3/s; Vc=',num2str(Vc),' m3'])
    text(0.1,0.67,['kv=',num2str(kv),' m/s; hconv=',num2str(hconv),' W/(m2K); hrad=' ...
            ,num2str(hrad)])
    drawnow
    %axis([0 1 0 1])

   %++++++++  MAIN LOOP  +++++++ Lines marked %# are useful for finding
   %                     the cause of instabilities in the calculations.
   %                     The expression for dt is empirical and may need
   %                     changes... .
   while slut==0
    %------TIME------
    i=i+1;
    ii=ii+1;
    %------MOISTURE-----
    Fsorp=kv*(vc-vs)*A;   %surface mass transfer from chamber to sample
    if ii==10
     ii=0;                %# if Fsorp==[];disp('help (Fsorp=[])');pause;end
     if vccrit==1
       dt=min([1/sqrt(alpha*(vf-vc)-beta*(vc-vs)) ...  %calculation of time step
          0.5*dx^2/Dc]);
                     %# if imag(dt)~=0; disp('help (imag(dt)~=0)');pause; end
     elseif vccrit==2
      dt=0.5*dxused^2/Dc;

     end
     %------END??-----
     if (t>=tend)|((c(n)-ci)/(cf-ci)>0.999)  %test for end of simulation
      slut=1;
      tnextut=t;
     end
     %------OUTPUT------
     if t>=tnextut  %test for output
      disp(['time=',int2str(t),'  E=',num2str((mean(c)-ci)/(cf-ci)),'  tend=',int2str(tend),'
dt=',num2str(dt)])
      j=j+1;
      tnextut=tnextut+tutstart*tutinc^j;
      tut(j)=t;
      Tsut(j)=Ts;
      vcut(j)=vc;
      cut(:,j)=c(:);
      dtut(j)=dt;
    end
end
t=t+dt;
%------CHAMBER VAPOR CONTENT-------
if vccrit==1
 vc=vc+(f*(vf-vc)-Fsorp)/Vc*dt;   %mass balance for chamber
elseif vccrit==2   %CHECK BEFORE USE!!!!
 if t>tin(k)
```

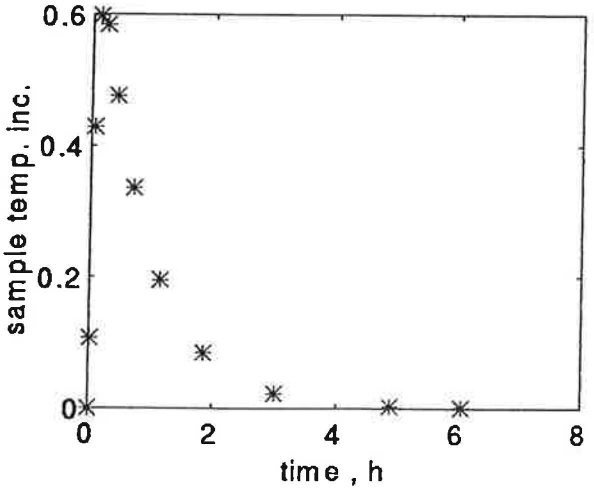
```
        k=k+1;
        slope=(vcin(k)-vcin(k-1))/(tin(k)-tin(k-1));
      end
      vc=vcin(k-1)+slope*(t-tin(k-1)); %linearly interpolated vc from given data
    end
                        %# if vc==[];disp('help (vc=[])');pause;end
    cc=c(1)+(Fsorp/A-K2*(c(1)-c(2)))/dx*dt; %save new surface cell concentration
                        %# if cc==[];disp('help (cc=[])');pause;end
                        %# if cc<0;disp('help (cc<0');pause;end
    cc2=c(n-1);  %save concentration of cell next to last (center) cell
    c(2:n-1)=c(2:n-1)+K4*(c(1:n-2)+c(3:n)-2*c(2:n-1))*dt; %Fick's law
    c(1)=cc;  %surface cell
    c(n)=c(n)+K4*(cc2-c(n))*dt; %center cell
    %------HEAT------
    Ts=Ts+(K5*(Tc-Ts)+Fsorp*Hs)/K6*dt;   %heat balance for sample
    %------NEW VS-------
    vs=t2vsat(Ts)*m2fi(c(1)/rho,isotherm,Ts);  %calculation of new sample temperature
                        %# if vc<vi; disp(['help (vc<vi)']); pause; end
  end
  %+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  ++++
  %------------CLEANING UP-------------
  tid=clock-tid;
  if tid(6)<0; tid(5)=tid(5)-1; tid(6)=tid(6)+60; end
  if tid(5)<0; tid(4)=tid(4)-1; tid(5)=tid(5)+60; end
  if tid(4)<0; tid(3)=tid(3)-1; tid(4)=tid(4)+24; end
  timestr='';
  if tid(3)>0; timestr=[int2str(tid(3)),' d  ']; end
  if tid(4)>0; timestr=[timestr,int2str(tid(4)),' h  '];end
  if tid(5)>0; timestr=[timestr,int2str(tid(5)),' m  '];end
  timestr=[timestr,int2str(tid(6)),' s'];end
  text(0.1,0.63,['iter=',int2str(i),'; outputs=',int2str(j)])
  text(0.1,0.59,['total time= ',timestr])
  figure(3)
  drawnow
  if j<100              %deleting parts of arrays not used
   dtut(j+1:100)=[];
   tut(j+1:100)=[];
   Tsut(j+1:100)=[];
   vcut(j+1:100)=[];
   cut(:,j+1:100)=[];
  end

  figure(4)
  clf
  L=length(tut);      %plot surface and mean concentration
  subplot(221)
  plot(tut/3600,cut(1,:),':')
  hold on
  plot(tut/3600,mean(cut),'-')
  text(tut(1)/3600+0.1*(tut(L)-tut(1))/3600, ...
      cut(1,1)+0.7*(cut(1,L)-cut(1,1)),'surf')
  text(tut(1)/3600+0.7*(tut(L)-tut(1))/3600, ...
      cut(1,1)+0.2*(cut(1,L)-cut(1,1)),'mean')
  Li=ceil(0.7*L);
  plot([tut(1)/3600+0.25*(tut(L)-tut(1))/3600 tut(Li)/3600], ...
      [cut(1,1)+0.7*(cut(1,L)-cut(1,1)) cut(1,Li)])
  plot([tut(1)/3600+0.65*(tut(L)-tut(1))/3600 tut(Li)/3600], ...
      [cut(1,1)+0.2*(cut(1,L)-cut(1,1)) mean(cut(:,Li))])
  hold off
  xlabel('time , h')
```

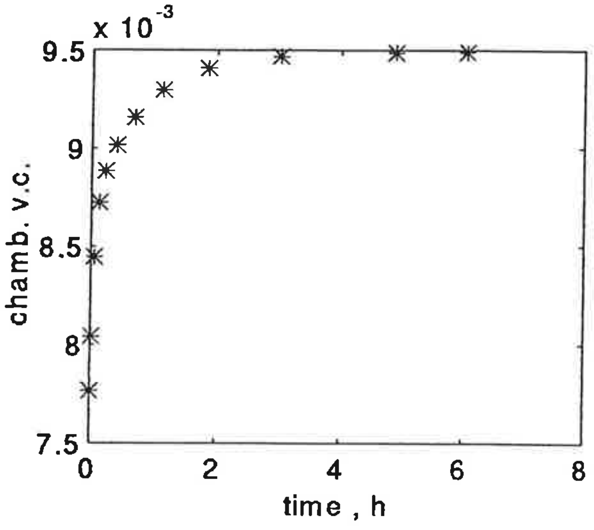
```
ylabel('moist.conc.')

subplot(222)    %plot concentration profiles
plot(cut);hold on;plot(cut,'*');hold off
xlabel('cell no.')
ylabel('conc.')

subplot(223)    %plot sample temperature
plot(tut/3600,Tsut-ones(size(Tsut))*(Tin+T0),'*')
xlabel('time , h')
ylabel('sample temp. inc.')

subplot(224)    %plot chamber vapor content
plot(tut/3600,vcut,'*')
xlabel('time , h')
ylabel('chamb. v.c.')
figure(4)
```

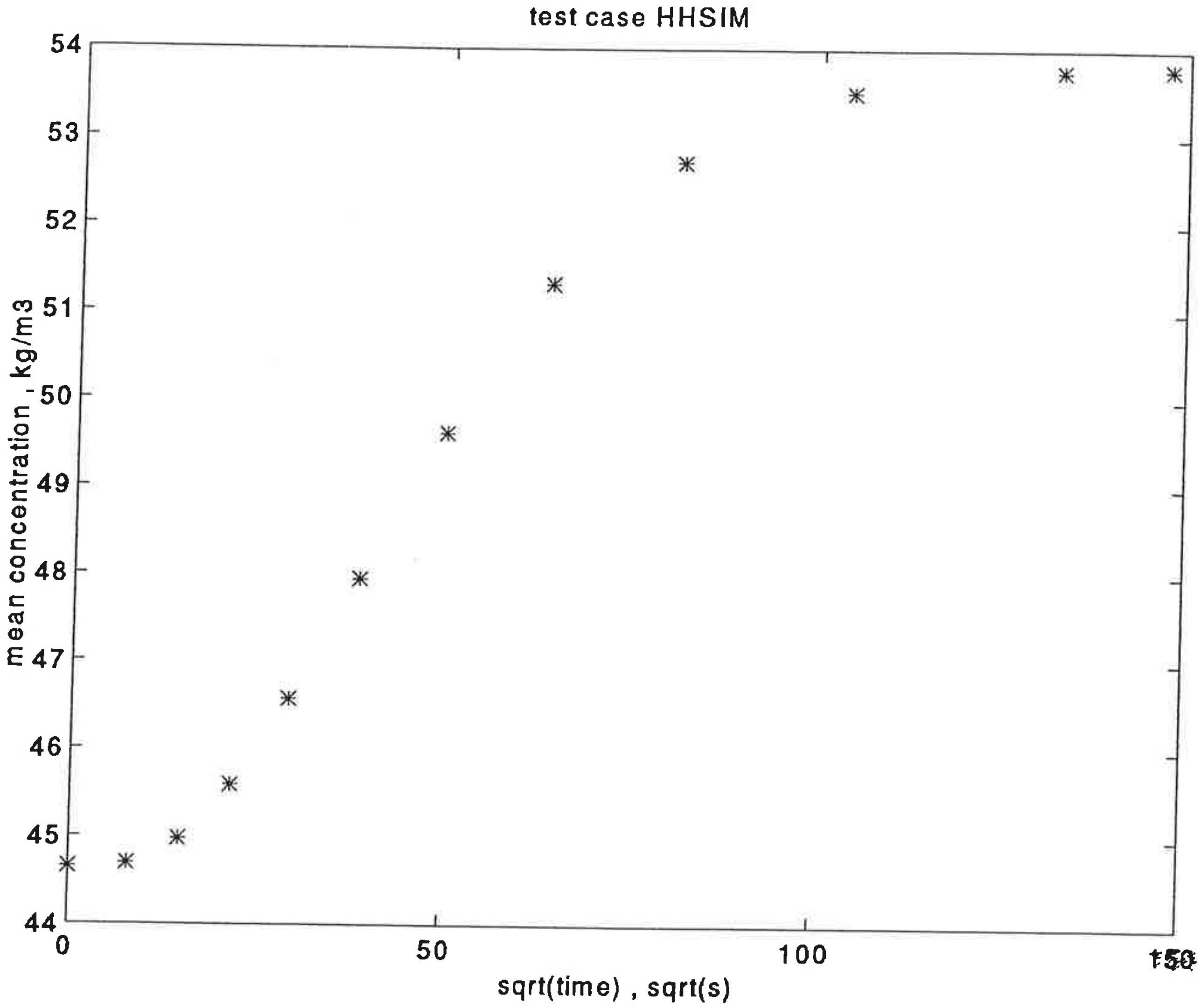
```
ell=0.00085 m; A=0.01782 m2
rho=530 kg/m3; M=0.008028 kg; n=3
Tc=20 C; rhi=0.45 rhf=0.55
vi=0.007765 kg/m3; vf=0.009491
ci=44.65 kg/m3; cf=53.8
Dc=2.188e-009m2/s; cp=1600 J/(kgK); Hs=2.8e+006 J/kg
f=0.00007 m3/s; Vc=0.022 m3
kv=0.005 m/s; hconv=6 W/(m2K); hrad=4
iter=8445; outputs=12
total time= 1 m  16 s
```










test case HHSIM
mean concentration , kg/m3
54
53
52
51
50
49
48
47
46
45
44
0
50
100
150
sqrt(time) , sqrt(s)

SNABBERRAKA

Lab W "HHSIM ..."

TVBM-7079

Eq 20 : $\Delta t = \frac{(\Delta x)^2}{2 \cdot D_c}$