第15卷第2期 火 灾 科 学 Vol. 15, No. 2
2005年4月 FIRE SAFETY SCIENCE Apr. 2005
文章编号:1004-5309(2005)02-0061-14

Two-layer Fire Zone Models with Symbolic Mathematics

Q. Kui, W. K. Chow

(Areas of Strength: Fire Safety Engineering Research Centre for Fire Engineering The Hong Kong Polytechnic University Hong Kong, China)

Abstract: The equations on the two－layer fire zone model ASET for a chamber with air supplied through leakage are solved by the symbolic mathematics programme MATLAB and MAPLE V. Two key equations are considered. A total number of 12 simulations were carried out on two design fires in six compartments with floor area varying from 10 to 100 m^2. The results are compared with those simulated by ASET itself and another software FIREWIND.

Keywords: Fire model, two－layer zone model, symbolic mathematics

0 Introduction

The use of symbolic mathematics[e.g. 1,2] is now very popular in science and engineering. How this technique can be applied in fire modeling is further studied in this paper. The software developed for handling matrices MATLAB[4] and MAPLE V[5] are selected. It is a high－performance language for technical computing, integrating computation, visualization, and programming in an easy－to－use environment. Typical uses include mathematical analysis, computational process, modeling, simulation, data analysis and visualization with scientific graphics. A good graphical processor is available for presenting the results.

Two－layer zone models are developed[e.g. 7－11] for simulating building fires by taking into account an upper hot smoke layer, a lower cool layer and a plume. There should be 11 variables on the properties of the two layers and the compartment pressure with 11 equations for solving them[e.g. 3,4,9]. But since there are seven physical constraints, the maximum number of ordinary differential equations required to be solved is only four. With intelligent use of assumptions, the number of differential equations to be solved can be less than four.

Available Safe Egress Time (ASET) is one of the earliest zone models[7] for calculating the temperature and interface height of the hot smoke layer in a single room with doors and windows closed. Available Safe Egress Time － BASIC (ASET－B) is a compact and easy－to－run program which solves the same equations as ASET but in BASIC[11]. Input data like the geometry of the room such as the floor area and ceiling height; heat loss fraction; the height and heat release rate of the fire; and the maximum time for the simulation are required for predicting the transient smoke layer temperature and interface height.

In this paper, the equations for ASET[6] are put into MATLAB[3] and MAPLE V [4] to simulate a typical room fire. The results are compared with those by FIREWIND version 3.4[12], which is a user－friendly tool for fire engineers to carry out fire hazard assessment.

收稿日期:2004-10-27;修改日期:2005-2-4

1 The Model ASET

ASET is a two-layer zone model for a closed chamber[7]. There are no openings assigned in the room with air supplied through leakage. In this way, the equations for vent flow need not be solved.

The following assumptions are made[13,14]:

- Room pressure is independent of height when the conservation of mass and energy equations are applied to both zones.
- The specific heat capacities of the gases in the room are assumed to be constant and estimated from the initial room temperature.
- The model may not be reliable for enclosures with a large length to width aspect ratio; or height to minimum horizontal dimension ratio[e.g. 15].
- The enclosure is assumed to be divided into two layers with heat and mass transfer through the plume. There might be problems when the upper layer temperature increases to give strong enough radiative heat flux to cause flashover.

2 Key Equations

A fire of heat release rate $Q(t)$ in a room of height H and cross-sectional area as shown in Figure 1 is considered. A smoke layer is formed with interface height Z_i above the floor. Equations for smoke layer interface height and smoke layer temperature are derived as shown in the literature and not repeated in here. The set of ordinary differential equations for such a two-layer zone model ASET for Z_i higher than the height of the fire are summarized as:

$$dZ_N/dt_N = -C_1 q - C_2 q^{1/3} Z_N^{5/3} \tag{1}$$

$$d\phi/dt_N = \phi[C_1 q - (\phi - 1) C_2 q^{1/3} Z_N^{5/3}]/(Z_{N,0} - Z_N) \tag{2}$$

where C_1 and C_2 are constants and the following dimensionless quantities are defined as:

$$C_1 = \frac{(1-L_C)Q(t)t_c}{\rho_{CL} C_p T_a A l_c} \tag{3}$$

$$C_2 = \frac{0.21 t_c}{A}\left[\frac{(1-L_C)Q(t)g l_c^2}{\rho_{CL} C_p T_a}\right]^{1/3} \tag{4}$$

$$Z_N = \frac{Z_i}{l_c} \quad l_c \sim 1\ \text{m} \tag{5}$$

$$t_N = \frac{t}{t_c} \quad t_c \sim 1\ \text{s} \tag{6}$$

$$T_a \sim 300\ \text{K} \tag{7}$$

$$q = Q(t) \quad Q_0 \sim 0.1\ \text{kW} \tag{8}$$

The equations are put into MATLAB and MAPLE V for predicting the smoke layer temperature and interface height.

3 Numerical Analysis

In this paper, equation (1) is solved by the Runge-Kutta (RK) method. There are many versions of the RK method but the choice of the time step is important. Rewriting the equation as:

$$\frac{dZ_N}{dt_N} = f(Z_N, t_N) \tag{9}$$

with

$$Z_N(t_0) = Z_0 \tag{10}$$

Advancing from the n^{th} time step to the $(n+1)^{th}$ time step:

$$t_{n+1} = t_n + h \tag{11}$$

$$Z_{n+1} = Z_n + \left(\frac{h}{6}\right)(p_n + 2\times q_n + 2\times r_n + s_n) \tag{12}$$

$$p_n = f(t_n, Z_n) \tag{13}$$

$$q_n = f\left(t_n + \frac{h}{2}, Z_n + \frac{h}{2}\times p_n\right) \tag{14}$$

$$r_n = f\left(t_n + \frac{h}{2}, Z_n + \frac{h}{2}\times q_n\right) \tag{15}$$

$$s_n = f(t_n + h, Z_n + h\times r_n) \tag{16}$$

In solving an ordinary differential equation (ODE), the rate of convergence, the accuracy (or even validity) of the predicted results, and the completeness of the response should be considered. Particularly, convergence must be judged by some global criteria.

"ODE45" in MATLAB for solving non-stiff differential equations is used in this paper to solve the two ODEs[16]. This is an explicit Runge-Kutta (4,5) pair of Dormand and Prince method with a "free" interpolate of order four[17,18] with local extrapolation.

The expression in MATLAB is:

```
[t, Z] = ode45('asetsub', [1:1:te], [3.5; 300], [], Tg, Lc, Lr, A, pcl, Cp, Qt, Q0)
```

tout = t

yout = Z

o3, o4, o5, o6 = null

odefile = asetsub, asetsub2

tspan = 1:1:620

y0 = 3.5;300

options = null

varargin = Tg, Lc, Lr, A, pcl, Cp, Q0, Q, k

function [tout, yout, o3, o4, o5, o6] = ode45(odefile, tspan, y0, options, varargin)

Non-stiff differential equations would be solved by ODE45 by the medium order method.

[T, Y] = ODE45('F', TSPAN, Y0) with

TSPAN = [T0 TFINAL] integrates the system of differential equations $y' = F(t, y)$ from time T0 to TFINAL with initial conditions Y0.

'F' is a string containing the name of an ODE file. The function F(T, Y) must return a column vector. Each row in solution array Y corresponds to a time returned in column vector T. To obtain solutions at specific times T0, T1, ..., TFINAL (all increasing or all decreasing), use TSPAN = [T0 T1 ... TFINAL].

As an example, the commands

options=odeset('RelTol', 1e-4, 'AbsTol', [1e-4 1e-4 1e-5]);

ode45('asetsub', [1:1:te], [3.5; 300], options);

solve the system $y' = \text{rigidode}(t, y)$ with relative error tolerance 10^{-4} and absolute tolerances of 10^{-4} for the first two components and 10^{-5} for the third. When called with no output arguments, as in this example, ODE45 calls the default output function ODEPLOT to plot the solution as it is computed.

4 Stiff ODE

A stiff ODE means an ODE in which the solution function exhibits rapid and extreme changes in the dependent variable with small variations in the independent variable[17,19]. As a result, a plot of the solution function over long time frames (since time is taken as the independent variable in this study) will look quite different from a plot over short time frames.

A set of differential equations is called "stiff"[17,19] when the maximum eigenvalue in its jacobian matrix is several orders of magnitude larger than the minimum eigenvalue. In such case, the step size of integration is determined by the largest eigenvalue, while the final time of integration usually depends on the smallest eigenvalue. There are easier, but less precise, definitions of stiff differential equations:

- A set of differential equations is "stiff" when an excessively small step is needed to obtain correct integration.
- A set of differential equations is "stiff" when it contains at least two "time constants" (where "time" is supposed to be the joint independent variable) that differ by several orders of magnitude.

Integrating such equations using traditional explicit (Runge Kutta) methods may take very a long computing time; implicit methods should be used to reduce the computation time. However, implicit methods are not very efficient in solving normal and non-stiff equations.

A non-stiff ODE means an ODE in which the solution function exhibits slow and smooth changes in the dependent variable with small variations. As a result, plotting the solution function over long time frames (time is the independent variable) will look quite the same as plotting over shorter time frames.

5 Numerical Experiments

A fire is placed at the centre of six rooms of different areas but of the same height of 3.5 m. Two heat release rates Q(t) are considered:

- Fire F1: Steady burning with a constant heat release rate of 0.8 MW
- Fire F2: NFPA slow t^2-fire[20], with a cutoff value of 0.8 MW

To cope with the rapid changes in the fire environment, smaller time steps are required for correct integration to ensure convergence and accuracy of the predicted results. Stiff ODE is selected to solve the above equations.

The fraction of heat lost by conduction and radiation L_C is taken to be 0.35, the specific heat capacity of air C_p is 1004 $Jkg^{-1}K^{-1}$ and the cool layer temperature T_{CL} is 300 K.

The floor areas of the six rooms are:

- Room 1: 10 m^2
- Room 2: 20 m^2
- Room 3: 30 m^2
- Room 4: 40 m^2
- Room 5: 50 m^2
- Room 6: 100 m^2

Typical examples of a MAPLE V program listing and a MATLAB program listing are shown in Appendices A and B.

The results predicted are compared with those by FIREWIND HotLayer[12] and ASET[7,13] itself as shown in Figures 2 to 7. There are differences in the results predicted by the three models.

Floor area of the compartment is a key factor. It is expected that the results of using MATLAB, i. e. curve A in all figures, should be similar to those simulated by ASET itself as given by curves C. However, there are many cases which the two do not agree. The results of FIREWIND (i. e. curves B) are used for comparison which agree with the results of MATLAB in some cases.

6 Conclusion

The following conclusion can be drawn from this study:

- Symbolic mathematics is now a powerful tool for doing mathematics with a computer. It can be applied to simulate a building fire with a two-layer model. A relatively simple model like ASET can be put into the computer easily. Graphics outputs can be achieved easily.
- It is easier to change the equations and parameters concerned for describing the physics concerned in comparison with traditional computer programming.
- The software is under active development, both on the numerical schemes and graphical presentation.

From the simulations, floor area should be watched carefully in using two-layer zone models. Symbolic mathematics should be put into the teaching curriculum of engineering degree programmes.

References

[1] S.M. Christensen. Resources for computer algebra [J]. Computers in Physics, 1996, 8(3): 308～316.

[2] M.H. Holmes, Y. Au, J.W. Stayman. Symbolic and numerical computing for chemical kinetic reaction schemes [J]. Computers in Physics, 1995, 9(6): 629～634.

[3] W.K. Chow, L. Meng. Analysis of key equations in a two－layer zone model and application with symbolic mathematics in fire safety engineering [J]. Journal of Fire Sciences, 2004, 22(2): 97～124.

[4] L. Meng, W.K. Chow. Mathematical analysis and possible application of the two－layer zone model FIRM [J]. International Journal on Engineering Performance－Based Fire Codes, 2002, 4(1): 13～22.

[5] MATLAB Guide and Tutor [Z]. Version 5, The Mathworks Inc.

[6] MAPLE V. Version 5.1 User's Manual [Z]. Ontario, Canada: Waterloo Maple Inc., 1998.

[7] L.Y. Cooper. A Mathematical model for estimating available safe egress time in fire [J]. Fire and Material, 1982, 6(3&4): 135～144.

[8] D.M. Birk. An Introduction to Mathematical Fire Modeling [M]. Technomic Publishing Co., Inc., 1991.

[9] R.G. Rehm, G.P. Forney. A Note on the Pressure Equations Used in Zone Fire Modelling [R]. NISTIR 4906, Building and Fire Research Laboratory, National Institute of Standard and Technology, Gaithersburg, Maryland, USA, 1992.

[10] M. Law. Fire and Smoke Models-Their Use on the Design of Some Large Buildings [J]. ASHRAE Transactions, 1990, 96(1): 963～971.

[11] D.M. Birk. An Introduction to Mathematical Fire Modeling [M]. Lancaster, Pennsylvania, U.S.A.: Technomic Publishing Co., Inc., 1991.

[12] FIREWIND version 3.3 User's Manual [Z]. Fire Modelling & Computing, 66 Westbrook Avenue, Wahroonga, N.S.W. 2074, Australia, 1997.

[13] L.Y. Cooper. Heat transfer from plume to an unconfined ceiling [J]. Journal of Heat Transfer, 1982, 104

(3) : 446～451.

[14] D. M. Brani, W. Z. Black. Two－zone model for single－room fire [J]. Fire Safety Journal, 1992, 19(2&3) : 189～216.

[15] J. Quintiere. Fundamental of Enclosure Fire Zone Models [M]. SFPE Fire Protection Engineering Handbook, SFPE－NFPA, 7th Edition, 1995.

[16] MATLAB Programming [Z]. The Mathworks Inc.

[17] J. R. Dormand. Numerical Method for Differential Equations [M]. New York, USA : CRC Press, Inc. , 1996.

[18] J. R. Dormand, P. J. Prince. A Family of embedded Runge-Kutta formulae [J]. J. Comp. and Appl. Math. , 1980, 6 : 19～26.

[19] Computational Ordinary Differential Equations: Based on the Proceedings of a Conference on Computational Ordinary Differential Equations [M]. New York, USA : Oxford Clarendon Press, Oxford University Press, 1992.

[20] NFPA 92B Guide for Smoke Management Systems in Malls, Atria and Large Areas [S]. Quincy, Mass. , USA : National Fire Protection Association, 1991.

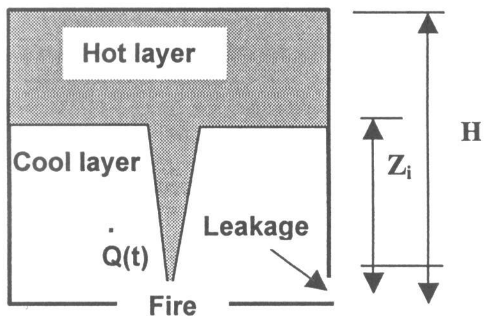


Figure 1: ASET zone model

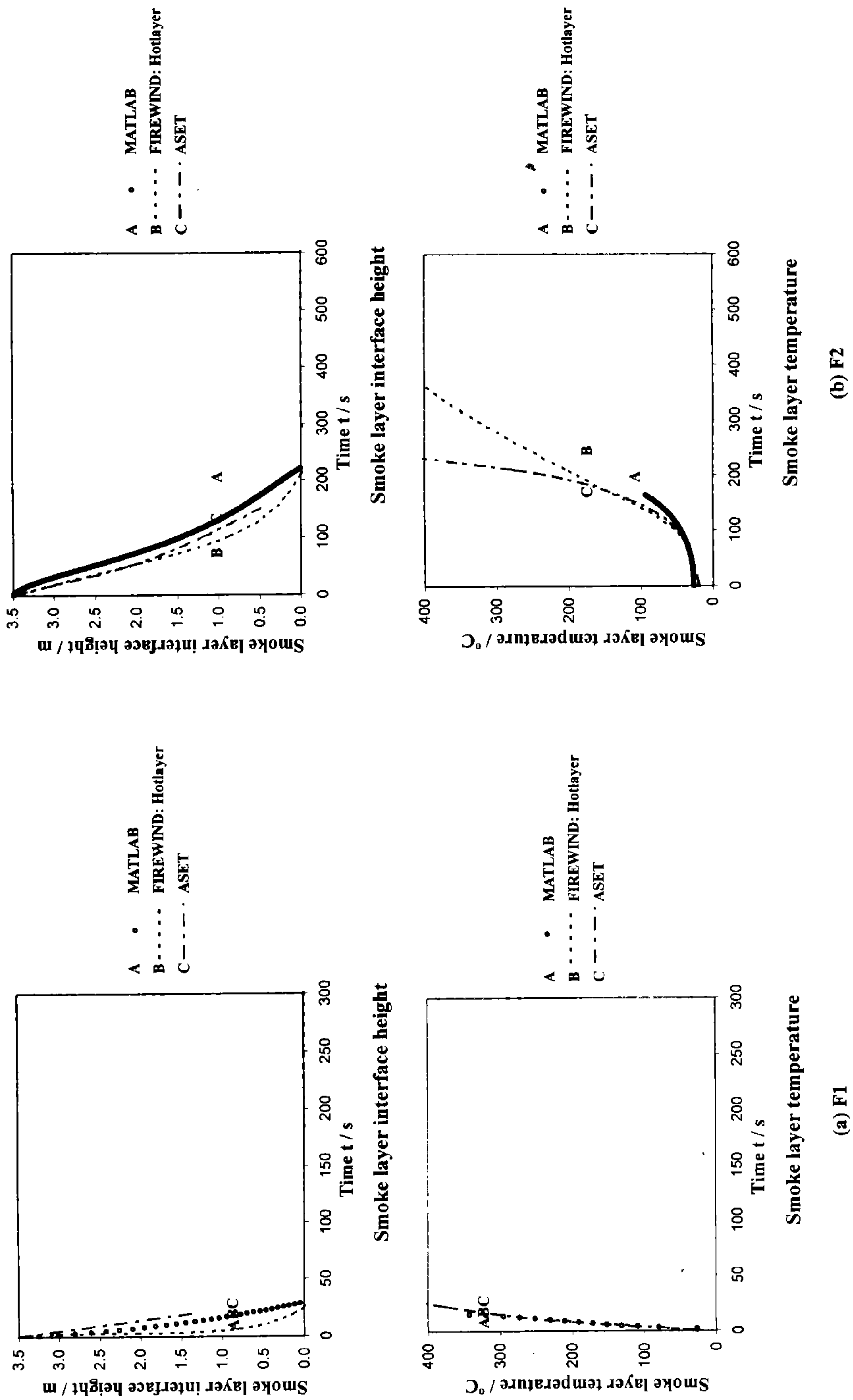


Figure 2: Results for room 1

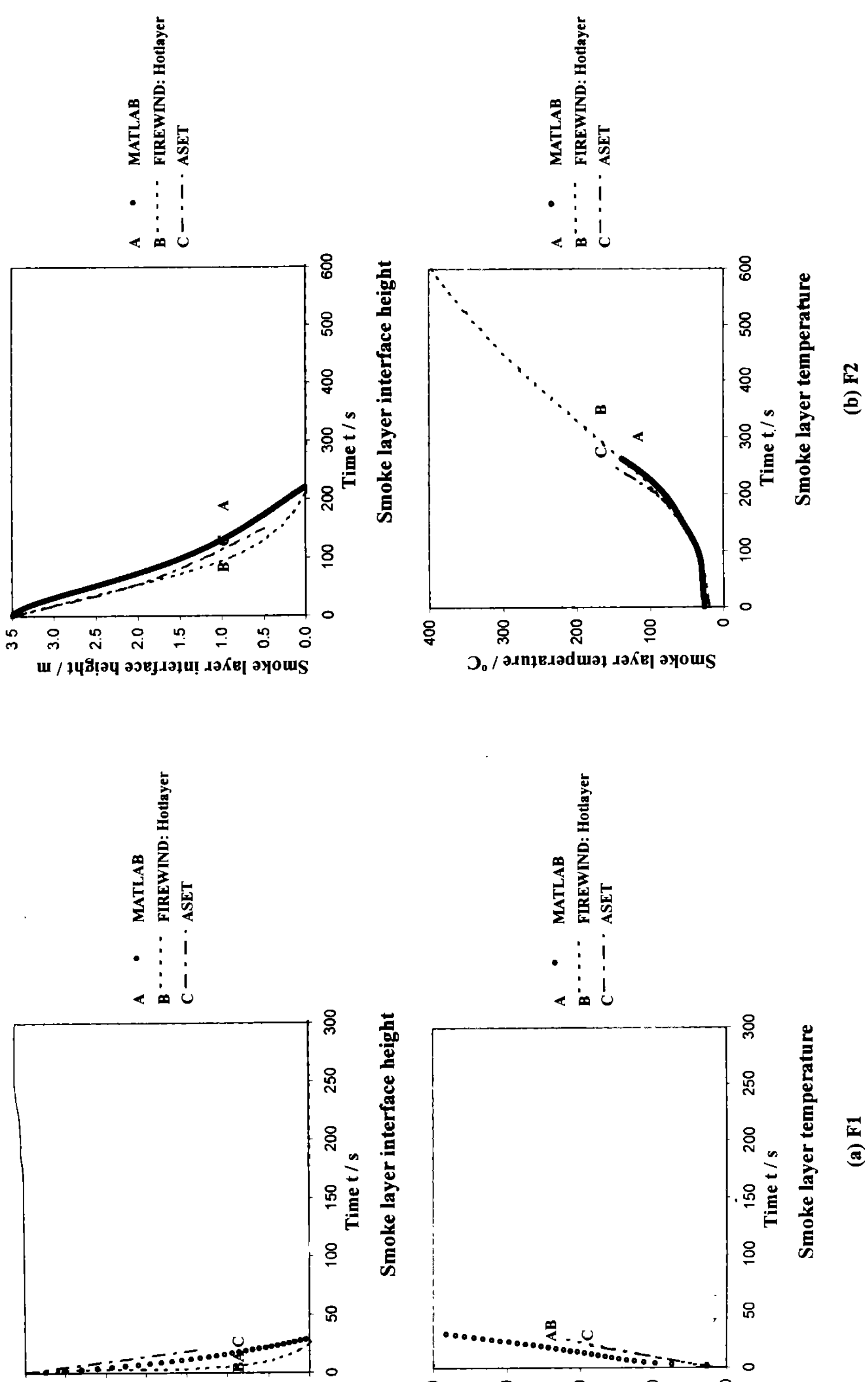
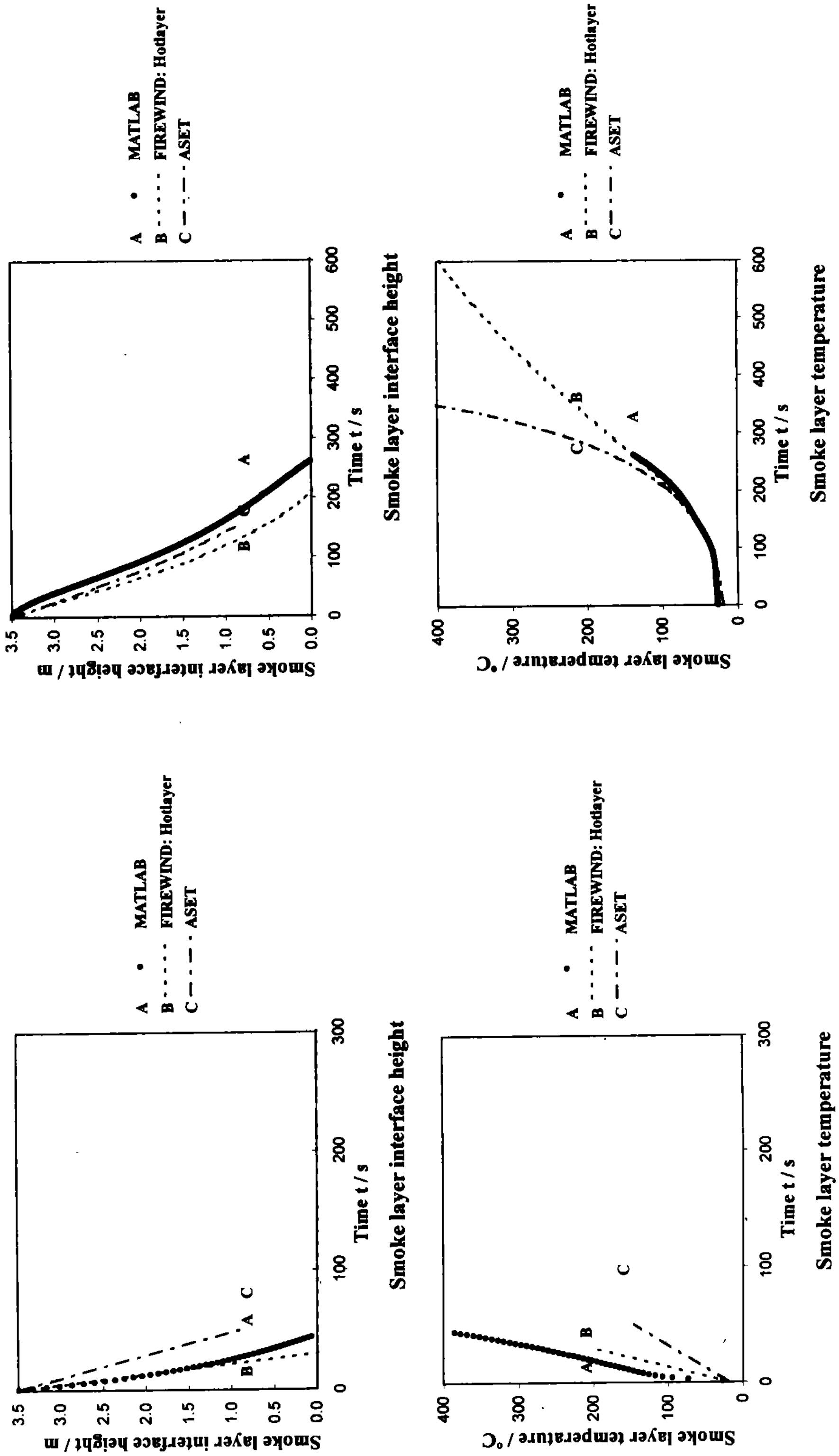


Figure 3: Results for room 2



(a) F1 (b) F2

Figure 4: Results for room 3

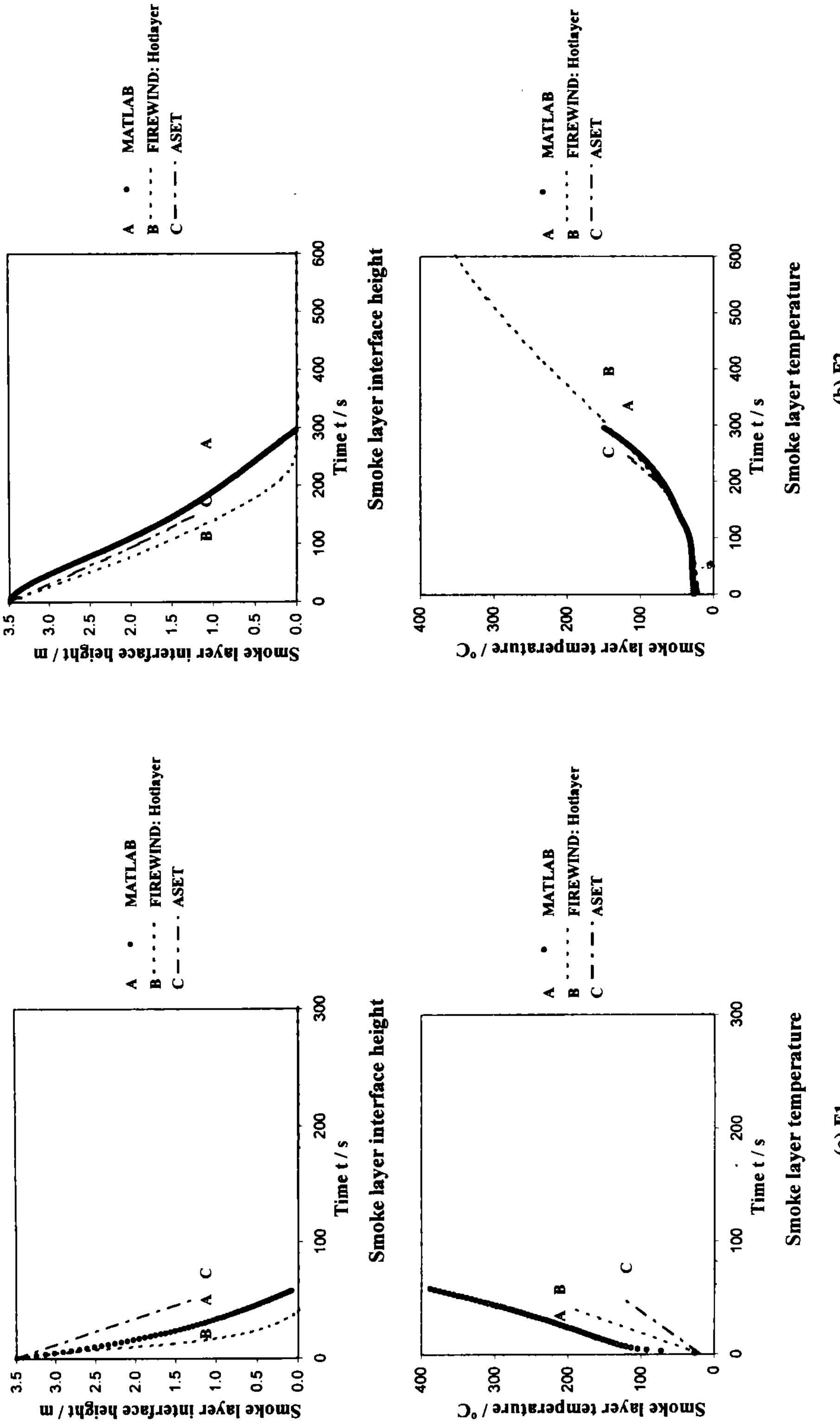


Figure 5: Results for room 4

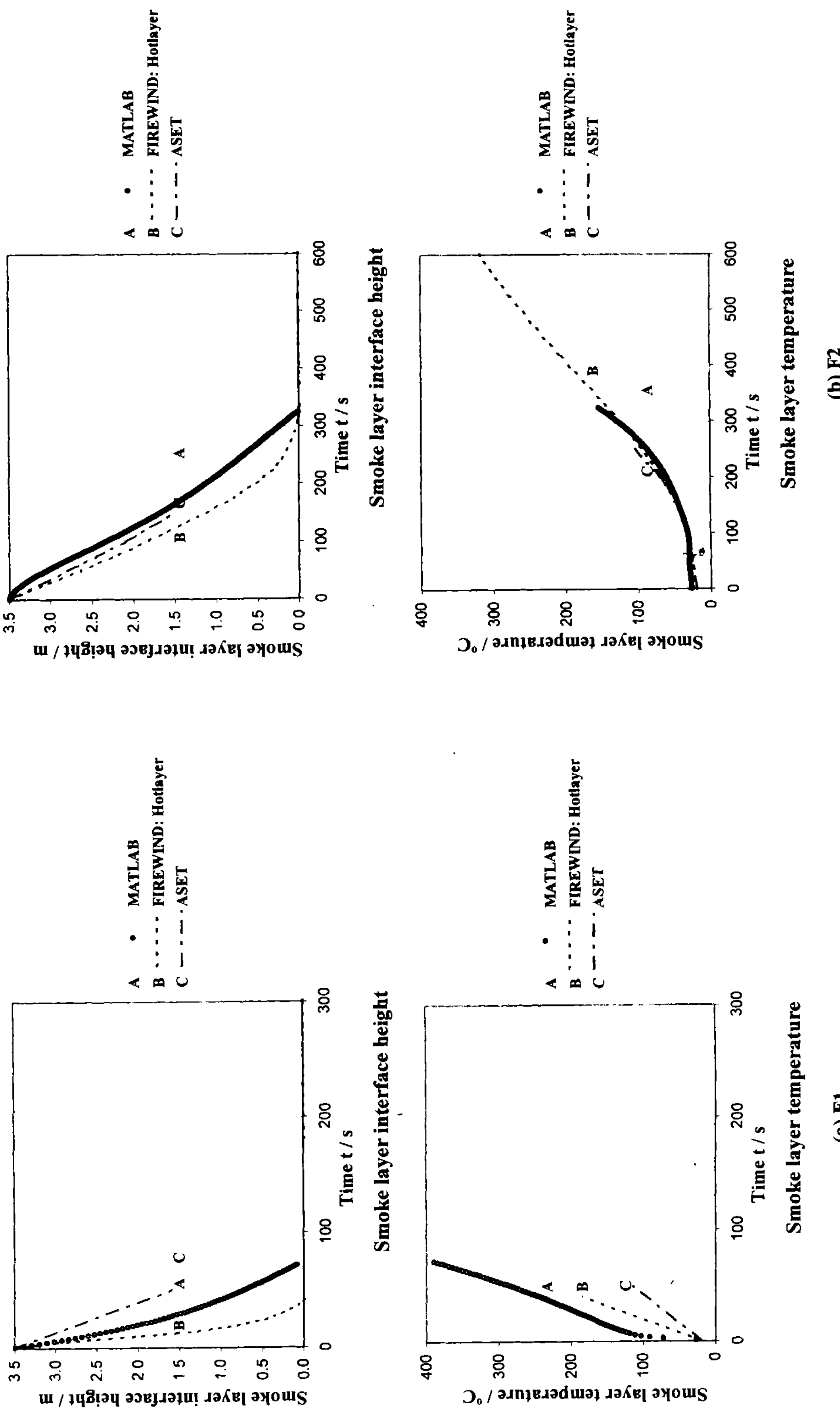
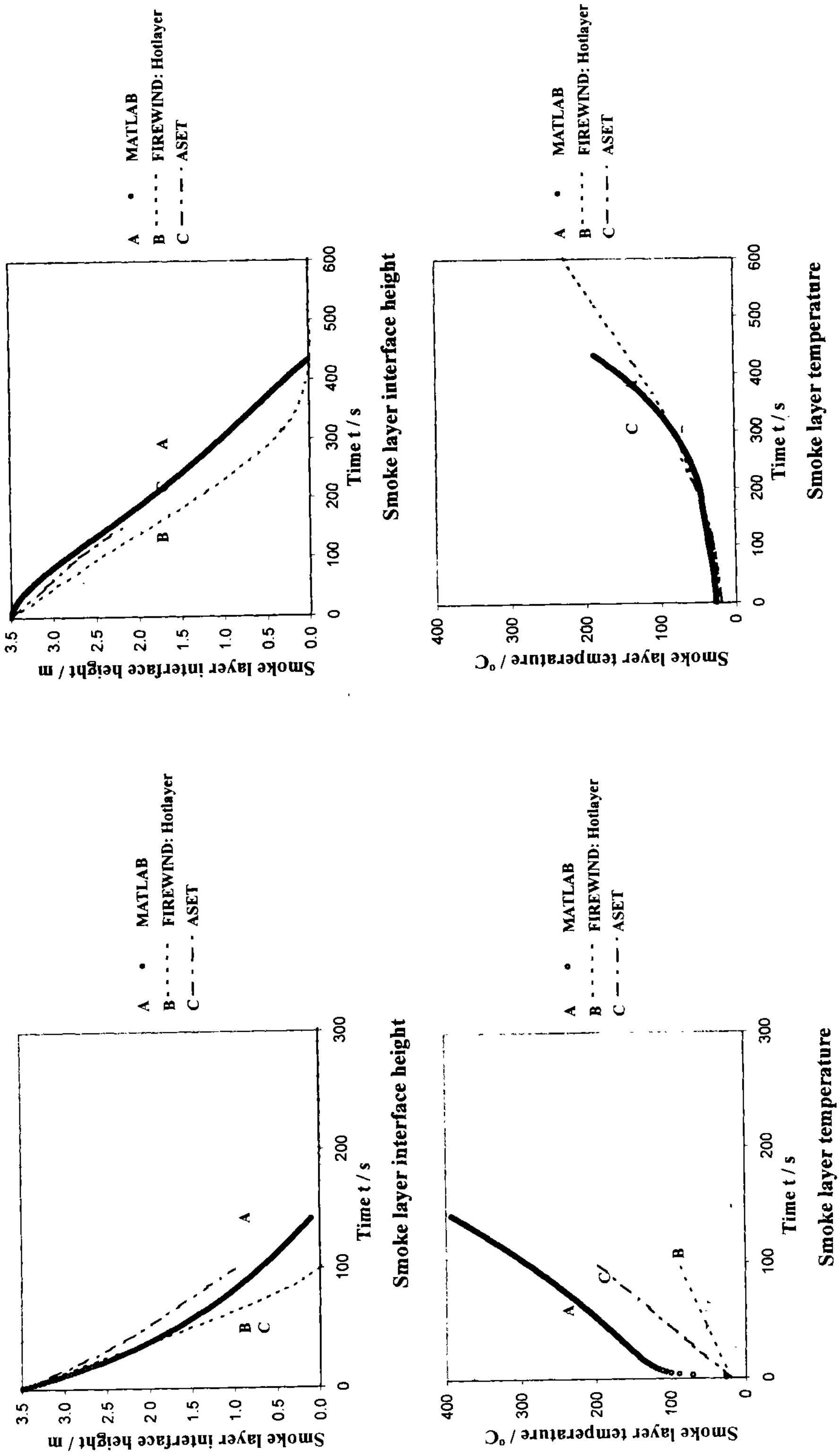


Figure 6: Results for room 5



(a) F1

(b) F2

Figure 7: Results for room 6

Appendix A: MAPLE V program listing

A1. Main program for constant fire:

```
> A:=10;
> H:=3.5;
> Q:=800000;
> Lc:=0.35;
> Cp:=1004;
> k:=600;
> Tg:=300;
> g:=9.8;
> pL:=1.293;
>
> TMAX1:=sqrt(Q/1000/1000)*k;
> TMAX:=trunc(TMAX1);
>
> C1:=(1-Lc)/(Cp*Tg*pL*A);
> C2:=0.21*((1-Lc)*g/(pL*Cp*Tg))^(1/3)/A;
>
> with(share);
> with(ODE);
> S0:=[0.0,H];
> eq0:=(t,Z)->-C2*(Q)^(1.0/3.0)*Z^(5.0/3.0)-C1*Q;
> Z1:=rungekuttahf(eq0,S0,0.1,1);
> Zt:=Z1[1];
> RS:=[Zt[1],Zt[2],Tg];
> dppts3:=array(0..909);
> dppts3[1]:=RS;
> for n from 1 by 1 while RS[2]>=0.001 do
>   for i from 1 by 1 while i<=10 do
>      eq12:=(t,Z,T)->-C2*(Q)^(1.0/3.0)*Z^(5.0/3.0)-C1*Q;
>      eq22:=(t,Z,T)->T/(H-Z)*(C1*(Q)+C2*
> ((Q)^(1.0/3.0))*(1.0-T/300)*Z^(5.0/3.0));
>      rkpts2:=rungekuttahf([eq12,eq22],RS,0.1,1);
>      RS:=rkpts2[1];
>   od;
>   dppts3[n]:=RS;
> od;
> plot(makelist(dppts3));
```

A2. Main program for t^2 fire:

```
> A:=10;
> H:=3.5;
> Q:=800000;
> Lc:=0.35;
> Cp:=1004;
> k:=600;
> Tg:=300;
> g:=9.8;
> pL:=1.293;
>
> TMAX1:=sqrt(Q/1000/1000)*k;
> TMAX:=trunc(TMAX1);
>
> C1:=(1-Lc)/(Cp*Tg*pL*A);
> C2:=0.21*((1-Lc)*g/(pL*Cp*Tg))^(1/3)/A;
>
> with(share);
> with(ODE);
> S0:=[0.0,H];
>eq0:=(t,Z)->-C2*(1000*(t/k)^2.0*1000.0)^(1.0/3.0)*Z^(5.0/3.0)-C1*1000*(t/k)^2.0*1000.0;
> Z1:=rungekuttahf(eq0,S0,0.1,1);
> Zt:=Z1[1];
> RS:=[Zt[1],Zt[2],Tg];
> dppts3:=array(0..909);
> dppts3[1]:=RS;
> for m from 1 by 1 to TMAX while RS[2]>=0.001 do
>   for i from 1 by 1 while i<=10 do
>      eq11:=(t,Z,T)->-C2*((1000*(t/k)^2.0*1000.0)^(1.0/3.0))*(Z^(5.0/
>      3.0))-C1*1000*(t/k)^2.0*1000.0;
>      eq21:=(t,Z,T)->T/(H-Z)*(C1*1000*(t/k)^2.0*1000.0+C1*
> ((1000*(t/k)^2.0*1000.0)^(1.0/3.0))*(1.0-T/300)*(Z^(5.0/3.0)));
>      rkpts1:=rungekuttahf([eq11,eq21],RS,0.1,1);
```

```
>     RS:=rkpts1[1];
>   od;
>   dppts3[m]:=RS;
> od;
> for n from 1 by 1 while RS[2]>=0.001 do
>   for i from 1 by 1 while i<=10 do
>     eq12:=(t,Z,T)->-C2*(Q)^(1.0/3.0)
*Z^(5.0/3.0)-C1*Q;
>     eq22:=(t,Z,T)->T/(H-Z)*(C1*(Q)
+C2*((Q)^(1.0/3.0))*(1.0-T/300)*Z^
(5.0/3.0));
>     rkpts2:=rungekuttahf([eq12,eq22],RS,0.1,1);
>     RS:=rkpts2[1];
>   od;
>   dppts3[m+n-1]:=RS;
> od;
> plot(makelist(dppts3));
```

Appendix B: MATLAB program listing

B1. Main program for constant fire:

COMMAND-File

```
clear; clf;
% initial conditions
H = 3.5;
A = 10;
Tg = 300;
Q = 800;
g = 9.8;
Lc = 0.35;
Cp = 1004;
pcl = 1.293;
p = 1.013;
k = 600;

% C1 and C2
C1=(1-Lc)/(pcl*Cp*Tg*A);
C2=0.21/A*((1-Lc)*g/(pcl*Cp*Tg))^(1/3);
% calculate first step for avoiding divided by zero
when Z(1) = H
Z0(1)=H;
[t,Z] = ode15s('asetsub1',[1:1:800],[Z0],[],
Tg,H,Q,k,C1,C2);
zrelt = [t,Z];
% time step of Z > 0, te
for i = 1:1:800
  if zrelt(i,2)> 0.1          ta = i;
  end
end
te = fix(ta);
% initial Z
Ze = zrelt(2,2);
% calculate other steps
Z0(1)=Ze;
Z0(2)=300;
[t,Z] = ode15s('asetsub',[1:1:te],[Z0],[],Tg,
H,Q,k,C1,C2);
  result = [t,Z];
% plot figures and ending
plot(t,Z(:,1),'k-');
figure
plot(t,Z(:,2),'k-');
save aset.dat Z-ascii
```

M-File

```
function dZ = asetsub1(t,Z,options,Tg,H,Q,k,
C1,C2)
% dZ/dt is dZ(1)
dZ = zeros(1,1);
dZ(1) = -C1*(1000*Q)-C2*(1000*Q)^
(1/3)*Z(1)^(5/3);
function dZ = asetsub(t,Z,options,Tg,H,Q,k,
C1,C2)
% dZ/dt is dZ(1)
% dT/dt is dZ(2)
dZ = zeros(2,1);
dZ(1) = -C1*(1000*Q)-C2*(1000*Q)^
(1/3)*Z(1)^(5/3);
dZ(2) = Z(2)/(H-Z(1))*(C1*(1000*Q)+C2*
(1000*Q)^(1/3)*Z(1)^(5/3)*(1-Z(2)/Tg));
```

B2. Main program for t^2 fire:

COMMAND-File

```
clear; clf;
% initial conditions
H = 3.5;
A = 100;
```

```
Tg = 300;
Q = 800;
g = 9.8;
Lc = 0.35;
Cp = 1004;
pcl = 1.293;
p = 1.013;
k = 600;
% C1 and C2
C1=(1-Lc)/(pcl*Cp*Tg*A);
C2=0.21/A*((1-Lc)*g/(pcl*Cp*Tg))^(1/3);
% calculate first step for avoiding divided by zero when Z(1) = H
Z0(1)=H;
[t,Z] = ode15s('aset2sub1',[1:1:800],[Z0],[],Tg,H,Q,k,C1,C2);
zrelt = [t,Z];
% time step of Z > 0, te
for i = 1:1:800
  if zrelt(i,2)>0
     ta = i;
  end
end
te = fix(ta);
% initial Z
Ze = zrelt(2,2);
% calculate other steps
Z0(1)=Ze;
Z0(2)=300;
   [t,Z] = ode15s('aset2sub',[1:1:te],[Z0],[],Tg,H,Q,k,C1,C2);
  result = [t,Z];
% plot figures and ending
plot(t,Z(:,1),'k-');
figure
plot(t,Z(:,2),'k-');
save aset2.dat Z-ascii
```

M-File

```
function dZ = aset2sub1(t,Z,options,Tg,H,Q,k,C1,C2)
% dZ/dt is dZ(1)
dZ = zeros(1,1);
dZ(1) = -C1*(1000*(t/k)^2*1000)-C2*(1000*1000*(t/k)^2)^(1/3)*Z(1)^(5/3);
function dZ = aset2sub(t,Z,options,Tg,H,Q,k,C1,C2)
% dZ/dt is dZ(1)
% dT/dt is dZ(2)
dZ = zeros(2,1);
dZ(1) = -C1*(1000*(t/k)^2*1000)-C2*(1000*1000*(t/k)^2)^(1/3)*Z(1)^(5/3);
dZ(2)=Z(2)/(H-Z(1))*(C1*(1000*1000*(t/k)^2)+C2*(1000*1000*(t/k)^2)^(1/3)*Z(1)^(5/3)*(1-Z(2)/Tg));
```

符号数学运作双层区域火灾模型

隗 乔,周允基

(香港理工大学 学术强项领域 消防工程研究中心)

摘要:本文运用符号数学 MATLAB 及 MAPLE V 计算双层区域模型 ASET 的方程式,考虑两条基本方程,报告由六所面积 10 至 100 平方米的房间,在 2 个设计火灾下共十二个模式例题的环境及结果也和 ASET 及 FIREWIND 程序比较。

关键词:火灾模型,双层区域模型,符号数学

中图分类号:X924.2 **文献标识码**:A

作者简介：翁立坚（1961—），公安部沈阳消防科研所副研究员，主要从事火灾探测报警技术研究及产品开发和工程应用工作，E-mail：wlj283@vip.sina.com。

作者简介：隗乔，香港理工大学博士毕业，现在北京从事建筑工作。

责任编辑：陈　军
英文编辑：刘乃安