Numerical Methods in Spontaneous Combustion with the Help of MATLAB

LUO Quan-bing\textsuperscript{a,b}, FU You-hua\textsuperscript{c}, LIANG Dong\textsuperscript{a,b,*}

\textsuperscript{a}School of Engineering, Sun Yat-sen University, Guangzhou 510006, China
\textsuperscript{b}Guangdong Provincial Key Laboratory of Fire Science and Technology, Guangzhou 510006, China
\textsuperscript{c}Department of Industrial Engineering, Narvik University College, Narvik 8505, Norway

Abstract

Spontaneous combustion is a complicated process and its control function is a partial differential equation (PDE) of heat conduction. With the help of initial conditions and boundary conditions, we can get unique solution. The correspondence conditions are called conditions for unique solution. However, analytical solution is only possible for a few conditions. Hereby, with the help of computer, we introduce numerical methods to get numerical solutions. We will introduce both the analytical method and the numerical methods for a certain kind of spontaneous combustion. We will mainly focus on the numerical methods, since it has better competence when solving complex problems. And also to insure the accuracy of the numerical methods, we will contrast it with analytical solution. MATLAB is used as the tool of programme to solve the problem easily.

1. Introduction

Spontaneous combustion is a combustion process. And according to some previous research, it will obey to the fundamental principles of heat transfer \cite{1}. However, the heat source of spontaneous combustion is so complex that we can’t get the analytical solution of it. Before the computer era, many scientists like Semenov and Frank-Ramenetskil think out many ways to simplify the problem. They develop the theories which could determine the probability of the occurrence of spontaneous combustion. However, the work of their theories are neither easy to be understand, nor does the results is satisfying. Thus, we introduce the numerical method which can be fulfilled easily with the help of computers. Programming languages, such as C, FORTRAN, MATLAB etc, are all available for this purpose. MATLAB is used in this article since it could implement the algorithm much easier compared with other languages.

To demonstrate the rationality of this new method, we start with introducing the analytical solution of an easy partial differential equation, and then use analytical solution to prove the rationality of this numerical method. After that, we’ll make a comparison between analytical solution, numerical method and finite difference analysis when solving the same problem.

2. Mathematical Model

The mathematical model of spontaneous combustion is actually in three-dimension. However, to simplify the problem,
we will focus on one-dimensional problem \([2]\) in this article. The simplified model is shown in Fig 1.

![Fig.1. The mathematical model](image)

Control function:
From the theory of the heat transfer, we get the control function.

\[
\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} + \frac{Q}{\rho c}, (0 \leq x \leq l)
\]

\(Q\) is the heat source, \(W / m^3\).
\(\lambda\) is thermal conductivity, \(W / (m \cdot K)\)
\(a = \lambda / \rho c\) is thermal diffusivity.
\(l\) is a half of the whole size.

Initial conditions:
Suppose that there was uniform initial temperature in the material, which means

\[
T\bigg|_{t=0} = T_0, \quad 0 \leq x \leq l.
\]

Boundary conditions are as follows:

\[
\begin{align*}
\frac{\partial T}{\partial x} \bigg|_{x=0} &= 0, \\
-\lambda \frac{\partial T}{\partial x} \bigg|_{x=l} &= h(T \bigg|_{x=l} - T_\infty),
\end{align*}
\]

\(h\) is convective heat transfer coefficient, \(W / (m^2 \cdot K)\).
\(T_\infty\) is the temperature of the environment.

**N.B.** the aim of this article is to introduce the numerical method, for this reason, we need to first simplify the question to get the analytical solution.

To have a better understanding, we provide an example which has the following parameters:

\(\lambda = 0.1165 W / (m \cdot K)\), \(\rho = 942.7kg / m^3\), \(c = 1072.7J / (kg \cdot K)\), \(h = 10W / (m^2 \cdot ^\circ C)\), \(l = 5(m)\),

\(T_0 = 27(\circ C) = 300(K)\), \(T_\infty = 27(\circ C) = 300(K)\), \(Q = 5J / (m^3 \cdot s)\)

3. Analytical Solution

As an easy example, we can use the method of separation of variables to get the analytical solution \([3]\).
First, we bring in excess temperature $\theta = T - T_\infty$. We get
\[
\frac{\partial \theta}{\partial t} = a \frac{\partial^2 \theta}{\partial x^2} + \frac{Q}{\rho c}, \quad (0 \leq x \leq l)
\]
(4)
\[
\theta|_{t=0} = T_0 - T_\infty, \quad (0 \leq x \leq l)
\]
(5)
\[
\left. \frac{\partial \theta}{\partial x} \right|_{x=0} = \left. \left(h\theta + \lambda \frac{\partial \theta}{\partial x} \right) \right|_{x=l} = 0
\]
(5)

Then we suppose $\theta = u - \frac{Q}{2\lambda} x^2 + \frac{Ql^2}{2\lambda} + \frac{Ql}{h}$, we get
\[
\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2}, \quad (0 < x < l, \ t > 0)
\]
(6)
\[
\left. u \right|_{t=0} = \varphi(x), \quad (0 \leq x \leq l)
\]
(7)
\[
\left. \frac{\partial u}{\partial x} \right|_{x=0} = 0, \quad \left. \left(hu + \lambda \frac{\partial u}{\partial x} \right) \right|_{x=l} = 0
\]
(7)
\[
\varphi(x) = T_0 - T_\infty + \frac{Q}{2\lambda} x^2 - \frac{Ql^2}{2\lambda} - \frac{Ql}{h}
\]
(8)

Next, we will use the method of separation of variables to get the solution.
\[
u(x,t) = \sum_{n=1}^{\infty} C_n e^{\beta_n c} \cos \beta_n x,
\]
(9)
\[
C_n = \left( \frac{T_0 - T_\infty}{\beta_n} - \frac{Q}{\lambda \beta_n^3} - \frac{Ql}{h \beta_n} \right) \sin \beta_n l + \frac{Ql}{\lambda \beta_n} \cos \beta_n l
\]
\[
L_n = \frac{1}{4 \beta_n^2 \sin 2 \beta_n l + \frac{l}{2}}
\]
(10)
\[
(\beta_1, \beta_2, \ldots, \beta_n, \ldots \text{ is the positive solution of function } \frac{h}{\lambda \beta} = \tan \beta l \text{ in sequence})
\]

And we know that
\[
\theta = u - \frac{Q}{2\lambda} x^2 + \frac{Ql^2}{2\lambda} + \frac{Ql}{h}
\]
(11)
\[
\theta = T - T_\infty
\]
(12)

So we get the solution of our problem, which is
\[
T(x,t) = u - \frac{Q}{2\lambda} x^2 + \frac{Ql^2}{2\lambda} + \frac{Ql}{h} + T_\infty
\]
(13)
\[
= \sum_{n=1}^{\infty} C_n e^{\beta_n c} \cos \beta_n x - \frac{Q}{2\lambda} x^2 + \frac{Ql^2}{2\lambda} + \frac{Ql}{h} + T_\infty
\]
(14)
\[
C_n = \left( \frac{T_0 - T_\infty}{\beta_n} - \frac{Q}{\lambda \beta_n^3} - \frac{Ql}{h \beta_n} \right) \sin \beta_n l + \frac{Ql}{\lambda \beta_n} \cos \beta_n l
\]
\[
L_n = \frac{1}{4 \beta_n^2 \sin 2 \beta_n l + \frac{l}{2}}
\]
(15)
\((\beta_1, \beta_2, \ldots, \beta_n, \ldots)\) is the positive solution of function \(\frac{h}{\lambda \beta} = \tan \beta l\) in sequence.

Last but not least, we will get the solution of the example with the help of MATLAB. For the reason that there are infinite eigenvalue, we can’t get the exact solution in the end. Instead, we can get a reasonable solution by the use of a large number (e.g. 100) of eigenvalues in the front. The result is showed below:

Fig.2. The distribution of temperature

Fig.3. The temperature’s change of the centre of the broad

4. The Function Of MATLAB (pdepe)

With the help of MATLAB, we can solve many mathematical problems. We now use the function provided by MATLAB to solve partial differential function problem.

The pdepe function provided by MATLAB can only be used to solve one-dimensional partial differential function \(^4\). Here is a brief introduction.

The syntax of pdepe is

\[
sol = \text{pdepe} (m, \text{@pdefun}, \text{@pdeic}, \text{@pdebc}, x, t)
\]

And here is how we implement the pdepe function to our problem.

\(m\): A parameter corresponding to the symmetry of the problem. \(m\) can be slab = 0, cylindrical = 1, or spherical = 2.

\(@\text{pdefun}\): A handle to a function that defines the components of the PDE.

\[
(c \left(x, t, \frac{\partial u}{\partial x}\right) \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} \left[x^m f \left(x, t, u, \frac{\partial u}{\partial x}\right)\right] + s \left(x, t, u, \frac{\partial u}{\partial x}\right)
\]

So the syntax of it is that

\([c, f, s] = \text{pdefun} (x, t, u, du)\)

For our problem

\[
\frac{\partial T}{\partial t} = a \frac{\partial^2 T}{\partial x^2} + \frac{Q}{\rho c} \quad (0 \leq x \leq l)
\]

It can be written like that

\[
\frac{1}{a} \frac{d}{dt} \frac{\partial T}{\partial x} = \frac{\partial^2 T}{\partial x^2} + \frac{Q}{\Lambda} \quad (0 \leq x \leq l)
\]

\(@\text{pdeic}\): A handle to a function that defines the initial conditions.

\[
u \left(x, t_0\right) = u_0
\]
The syntax is 
\[ u_0 = \text{pdeic}(x) \]

As we have mentioned earlier, the initial condition is 
\[ T_{|x=0} = T_0, \quad 0 \leq x \leq l. \]  \hspace{1cm} (20)

@pdebc: A handle to a function that defines the boundary conditions. 
\[ p(x,t,u) + q(x,t,u) f(x,t,u, \frac{\partial u}{\partial x}) = 0 \]  \hspace{1cm} (21)

The syntax is: 
\[ [pa, qa, pb, qb] = \text{pdebc}(xa, ua, xb, ub, t) \]

In our example, the boundary condition is 
\[
\begin{align*}
\frac{\partial T}{\partial x} \bigg|_{x=0} &= 0, \\
-x \frac{\partial T}{\partial x} \bigg|_{x=l} &= h(T_{|x=l} - T_w),
\end{align*}
\]  \hspace{1cm} (22)

That is 
\[
\begin{align*}
0 + 1 \times \frac{\partial T}{\partial x} \bigg|_{x=0} &= 0, \\
h(T_{|x=l} - T_w) + \lambda \times \frac{\partial T}{\partial x} \bigg|_{x=l} &= 0,
\end{align*}
\]  \hspace{1cm} (23)

Then we will get the solution of this example with the help of MATLAB.

![Distribution of temperature](image1.png) ![Temperature change of the center of the bread](image2.png)

**Fig.4.** The distribution of temperature  
**Fig.5.** The temperature's change of the centre of the bread

---

5. Finite Difference Method

Before we start using finite difference method, let's disperse the function: 
For the control function, we’ll use implicit scheme for stability \[5\].

\[
\frac{T_{i}^{n+1} - T_{i}^{n}}{\Delta t} = a \frac{T_{i+1}^{n+1} - 2T_{i}^{n+1} + T_{i-1}^{n+1}}{(\Delta x)^2} + \frac{Q(T_{i}^{n})}{\rho c}
\]  \hspace{1cm} (24)

That is
\[
- \frac{a \Delta t}{(\Delta x)^2} T_{i+1}^{n+1} + \left( 1 + 2 \frac{a \Delta t}{(\Delta x)^2} \right) T_i^{n+1} - \frac{a \Delta t}{(\Delta x)^2} T_{i-1}^{n+1} = T_i^n + \frac{\Delta t}{\rho c} Q(T_i^n)
\]  
(25)

For the initial conditions and boundary conditions, we get

Initial conditions: \( T_i^0 = T_0 \)

Boundary conditions:

\[
\begin{align*}
\frac{T_2^n - T_1^n}{\Delta x} &= 0, \\
-\lambda \frac{T_{N+1}^n - T_N^n}{\Delta x} &= h(T_{N+1}^n - T_\infty),
\end{align*}
\]  
(26)

That is

\[
\begin{align*}
T_1^n - T_2^n &= 0 \\
T_N^n - \left( 1 + \frac{h \Delta x}{\lambda} \right) T_{N+1}^n &= -\frac{h \Delta x}{\lambda} T_\infty
\end{align*}
\]  
(27)

When the equations above being expressed in MATLAB, it is a matrix equation as follows:

\[
\begin{pmatrix}
1 & -1 \\
-\frac{a \Delta t}{(\Delta x)^2} & 1 + 2 \frac{a \Delta t}{(\Delta x)^2} & -\frac{a \Delta t}{(\Delta x)^2} \\
\vdots & \ddots & \ddots \\
-\frac{a \Delta t}{(\Delta x)^2} & 1 + 2 \frac{a \Delta t}{(\Delta x)^2} & -\frac{a \Delta t}{(\Delta x)^2} \\
1 & -\left( 1 + \frac{h \Delta x}{\lambda} \right)
\end{pmatrix}
\begin{pmatrix}
T_1^{n+1} \\
T_2^{n+1} \\
\vdots \\
T_N^{n+1}
\end{pmatrix}
= \begin{pmatrix}
0 \\
\frac{\Delta t}{\rho c} Q(T_2^n) \\
\vdots \\
\frac{\Delta t}{\rho c} Q(T_N^n) \\
-\frac{h \Delta x}{\lambda} T_\infty
\end{pmatrix}
\]  
(28)

\[
\begin{pmatrix}
0 \\
1 \\
\vdots \\
1 \\
0
\end{pmatrix}
\begin{pmatrix}
T_1^n \\
T_2^n \\
\vdots \\
T_N^n \\
T_{N+1}^n
\end{pmatrix}
= \begin{pmatrix}
\frac{\Delta t}{\rho c} Q(T_2^n) \\
\vdots \\
\frac{\Delta t}{\rho c} Q(T_N^n) \\
-\frac{h \Delta x}{\lambda} T_\infty
\end{pmatrix}
\]  
(29)

Suppose that \( A = \begin{pmatrix}
1 & -1 \\
-\frac{a \Delta t}{(\Delta x)^2} & 1 + 2 \frac{a \Delta t}{(\Delta x)^2} & -\frac{a \Delta t}{(\Delta x)^2} \\
\vdots & \ddots & \ddots \\
-\frac{a \Delta t}{(\Delta x)^2} & 1 + 2 \frac{a \Delta t}{(\Delta x)^2} & -\frac{a \Delta t}{(\Delta x)^2} \\
1 & -\left( 1 + \frac{h \Delta x}{\lambda} \right)
\end{pmatrix} \)
\begin{equation}
B = \begin{pmatrix}
0 \\
1 \\
\vdots \\
1 \\
0
\end{pmatrix}
\end{equation}

\begin{equation}
T^n = \begin{pmatrix}
T_1^n \\
T_2^n \\
\vdots \\
T_N^n \\
T_{N+1}^n
\end{pmatrix}, \quad T^{n+1} = \begin{pmatrix}
T_1^{n+1} \\
T_2^{n+1} \\
\vdots \\
T_N^{n+1} \\
T_{N+1}^{n+1}
\end{pmatrix}, \quad b^n = \begin{pmatrix}
0 \\
\frac{\Delta t}{\rho c} Q(T_2^n) \\
\vdots \\
\frac{\Delta t}{\rho c} Q(T_N^n) \\
-\frac{h\Delta x}{\lambda} T^n
\end{pmatrix}
\end{equation}

Then we get

\begin{equation}
AT^{n+1} = BT^n + b^n
\end{equation}

And the initial condition is

\begin{equation}
T^1 = \begin{pmatrix}
T_0 \\
T_0 \\
\vdots \\
T_0 \\
T_0
\end{pmatrix}
\end{equation}

Then we will get the solution of this example:

![Fig.6. The distribution of temperature](image1)

![Fig.7. The temperature's change of the centre of the broad](image2)

**6. Summary**

With the help of MATLAB, we get the solution of the example. Let’s contrast our solution first. What we concern most is the temperature of the centre of our problem. It was the decisive factor whether our spontaneous combustion problem can happen in the end.
As it is shown in the graphs, there is a tiny difference between analytical solution and the other two methods which is because of the accumulated error from computer where eigenvalues of the analytical solution is not absolutely accurate. However, the second method is based on internal MATLAB pdepe function which has a better general error control when compared with analytical solution. In the same manner, finite difference method also has a higher precision.

Advantages and limitations of the three methods to the problem of spontaneous combustion:

① The analytical solution of problem is a good way. We can get the theoretically exact solution. However, not all of the spontaneous combustion problems can be solved in this way which is because the heat source of spontaneous combustion is a non-linear function. But the exact solution can be used to test the rationality of our numerical method.

② With the help of computer, we can get the numerical solution of more complex problem. MATLAB provide us the pdepe function to solve partial differential equation. It can be used to solve more complex problem easier and quicker. However, this method can only be used when solving one-dimensional problems.

③ The finite difference method is the mainstream method for heat transfer problems. It also has the competence when the model is in three-dimension. Thus, this method is recommended to solve the spontaneous combustion problems.

Acknowledgements

This work was supported by funds of Guangdong Provincial Scientific and Technological Project (No. 2011B090400518) and Guangdong Provincial Key Laboratory of Fire Science and Technology (No. 2010A060801010).

References
[1] HUANG Yong, 2009. Combustion And Combustor, Beihang University Press, Beijing, China. pp.146-152.
[2] YANG Shi-ming, TAO Wen-quan, 2006. Heat Transfer (4th Edition), Higher Education Press, Beijing, China. pp.123-133.
[3] LIANG Kun-miao, 2010. Mathematical Physical Equation (4th Edition), Higher Education Press, Beijing, China, pp. 107-220.
[4] XUE Ding-yv, CHEN Yang-quan, 2008. Advanced Applied Mathematical Problem Solution With MATLAB (Second Edition), Tsinghua University Press, Beijing, China, pp. 255-257.
[5] ZHANG Wen-sheng, 2006. Finite Difference Methods For Partial Differential Equations In Science Computation, Higher Education Press, Beijing, China, pp. 235-245.