APPLIED MATHEMATICS FOR HEAT TRANSFER

Răzvan Tudor Tănasie, University of Craiova, ttodi21@yahoo.com

Abstract: Practical applications concerning heat transfer (e.g. heat propagation along wires) require representations based on superior degree algebraic equation or equations with partial derivatives. This paper deals with certain aspects concerning symbolic or numeric methods used to solve this type of equations. The practical engineering applications that have generated the mathematical problems do not represent the interest point of this paper. It focuses on the mathematical representation and on the way this representation can lead to results. Few aspects of programming implementation are included.

Keywords: applied mathematics, heat transfer, algebraic equation, equations with partial derivatives

1. MATHEMATICAL ISSUES

Ignoring the practical issues of the engineering applications concerning the heat transfer, we will focus on two main mathematical problems that involve finding certain solutions using either symbolic or numerical calculation.

For each and every issue the description of the initial conditions and of the requirements to be fulfilled, are presented clearly enough as to support a good understanding of the problems.

1.1 Solving a Heat Differential Equation

The heat differential equation (Lienhard 2001) is:

$$\frac{d^2T}{dx^2} - k_1 \cdot T + k_2 = 0 \qquad (1)$$

where T is a function of x and the initial conditions are $T(0) = T_1$ and $T(l) = T_2$, l is the length of the conductor heated from both ends.

There are three cases (Betounes 1998; Teodorescu 1980):

1. $k_1 > 0$
2. $k_1 = 0$
3. $k_1 < 0$

- The case $k_1 > 0$

Denoting

$$Y(x) = T(x) - \frac{k_2}{k_1} \Rightarrow \frac{d^2Y}{dx^2} - k_1 \cdot Y = 0 \qquad (2)$$

Knowing that $k_1 > 0$ then $\exists v \in R$, such that $v^2 = k_1$. The equation (2) then becomes:

$$\frac{d^2Y}{dx^2} - v^2 \cdot Y = 0 \Rightarrow y_{1,2} = \pm\frac{\sqrt{4v^2}}{2} = \pm v \quad (3)$$

Then the solution for *Y(x)* is:

$$Y(x) = c_1 \cdot e^{x \cdot v} + c_2 \cdot e^{-x \cdot v} \qquad (4)$$

Taking into consideration the initial conditions (Tveito 1998) it results that:

$$Y(0) = T_1 - \frac{k_2}{k_1} = c_1 + c_2 \qquad (5)$$

and

$$Y(l) = T_2 - \frac{k_2}{k_1} = c_1 \cdot e^{l.v} + c_2 \cdot e^{-l \cdot v} \qquad (6)$$

So

$$c_1 = -c_2 + T_1 - \frac{k_2}{k_1} \qquad (7)$$

$$\left(T_1 - \frac{k_2}{k_1}\right) \cdot e^{l \cdot v} + c_2(e^{-l \cdot v} - e^{l \cdot v}) = T_2 - \frac{k_2}{k_1} \qquad (8)$$

• The results included in this paper are a part of the work done by the author in a SOCRATES ERASMUS grant to the Universitat Bundeswehr in Munich, under the co-ordination of Prof. H. D. Liess.

$$c_2 = \frac{\frac{k_2}{k_1} - T_2 + \left(T_1 - \frac{k_2}{k_1}\right)\cdot e^{l\cdot v}}{e^{l\cdot v} - e^{-l\cdot v}} \quad (9)$$

$$c_1 = -\frac{\frac{k_2}{k_1} - T_2 + \left(T_1 - \frac{k_2}{k_1}\right)\cdot e^{-l\cdot v}}{e^{l\cdot v} - e^{-l\cdot v}} \quad (10)$$

Concluding:

$$T(x) = -\frac{\frac{k_2}{k_1} - T_2 + \left(T_1 - \frac{k_2}{k_1}\right)\cdot e^{-l\cdot\sqrt{k_1}}}{e^{l\cdot\sqrt{k_1}} - e^{-l\cdot\sqrt{k_1}}}\cdot e^{x\cdot\sqrt{k_1}} +$$

$$+\frac{\frac{k_2}{k_1} - T_2 + \left(T_1 - \frac{k_2}{k_1}\right)\cdot e^{l\cdot\sqrt{k_1}}}{e^{l\cdot\sqrt{k_1}} - e^{-l\cdot\sqrt{k_1}}}\cdot e^{-x\cdot\sqrt{k_1}} + \frac{k_2}{k_1} \quad (11)$$

- The case $k_1 = 0$

The equation (1) then becomes:

$$\frac{d^2T}{dx^2} = -k_2 \quad (12)$$

then

$$T(x) = -\frac{k_2}{2}\cdot x^2 + c_1\cdot x + c_2 \quad (13)$$

Without taking into consideration the initial conditions, $T(0) = T_1 = c_2$ and $T(l) = T_2 = -\frac{k_2}{2}\cdot l^2 + c_1\cdot l + T_1$. Then

$$c_1 = \frac{T_2 - T_1 + \frac{k_2}{2}\cdot l^2}{l} \quad (14)$$

Now the solution for *T(x)* is:

$$T(x) = -\frac{k_2}{2}\cdot x^2 + \frac{T_2 - T_1 + \frac{k_2}{2}\cdot l^2}{l}\cdot x + T_1 \quad (15)$$

- The case $k_1 < 0$

Denoting $k_3 = -k_1$ the equation (1) becomes:

$$\frac{d^2T}{dx^2} + k_3\cdot T + k_2 = 0 \quad (16)$$

Denoting $Y(x) = T(x) + \frac{k_2}{k_3}$ the equation (16) becomes:

$$\frac{d^2Y}{dx^2} + k_3\cdot Y = 0 \quad (17)$$

Knowing that $k_3 > 0$ then $\exists v \in R$, such that $v^2 = k_3$. The equation (17) then becomes:

$$\frac{d^2Y}{dx^2} + v^2\cdot Y = 0 \quad (18)$$

Then the solution for *Y(x)* is

$$Y(x) = c_1\cdot\cos(v\cdot x) + c_2\cdot\sin(v\cdot x) \quad (19)$$

Taking into consideration the initial conditions it results that :

$$Y(0) = T_1 + \frac{k_2}{k_3} = c_1 \quad (20)$$

and

$$Y(l) = T_2 + \frac{k_2}{k_3} = T_1 + \frac{k_2}{k_3}\cdot\cos(v\cdot l) + c_2\cdot\sin(v\cdot l) \quad (21)$$

So,

$$c_2 = \frac{T_2 + \frac{k_2}{k_3} - \left(T_1 + \frac{k_2}{k_3}\right)\cdot\cos(v\cdot l)}{\sin(v\cdot l)} \quad (22)$$

Then the solution for *T(x)* is

$$T(x) = \frac{T_2 + \frac{k_2}{k_3} - \left(T_1 + \frac{k_2}{k_3}\right)\cdot\cos\left(\sqrt{-k_1}\cdot l\right)}{\sin\left(\sqrt{-k_1}\cdot l\right)}\cdot$$

$$\cdot\sin\left(\sqrt{-k_1}\cdot x\right) - \frac{k_2}{k_3} + \left(T_1 + \frac{k_2}{k_3}\right)\cdot\cos\left(\sqrt{-k_1}\cdot x\right) \quad (23)$$

1.2 Computing Variables from Auto - Dependent Relations

- Computing *T*

In equation (1), it is possible for k_1 and k_2 to be T – dependent. This dependency can be a complex one so it would be easier to solve the equation without isolating *T*. It is known that the temperature value rises (depending on *x*) from the ends towards the point where it reaches the maximum, point where it tends to stabilize. A mathematical algorithm (Press 1997) will be elaborated in order to determine *T*.

T will be assigned the minimal value of T_0 and T_l. Then it is computed with the corresponding equation (see the cases from 1.1). If the difference between the computed temperature and the used one is less than ε (ε is selected around a very small value - e.g.: 10^{-6}) then it means that the stability zone has been reached and the needed temperature has been obtained. Otherwise, the temperature is again computed using the newly obtained temperature. The algorithm is repeated until the stability condition is reached or a number of iterations is exceeded (the number of iterations is selected by the user). If this number is exceeded, it means that the stability will never be reached because some values are too large.

- Computing a function of *T*

$$\text{Let } k_1 = \prod_{i=1}^{n} j_i \tag{24}$$

An other problem that may be encountered is the computing of one of j_i where j_i is a temperature dependent variable. As in the above paragraph, in some cases, it may be easier to compute j_i with an algorithm without isolating it. In order to compute j_i the value of *T* is needed. In most cases, the *T* dependency is a monotone or a quasi - monotone one. Only the algorithm for j_i will be explained, as being an increasing function of *T*. The decreasing case would be treated similarly.

For the beginning j_i will be assigned 0. It is considered a ratio –*x* (the value of this ratio is selected by the user). j_i is increased by this ratio *x*. With this newly obtained value, *T* is computed. We compare this new *T* with the real value of *T*.

If the modulus difference is less than ε, then j_i is appreciatively the value needed. Otherwise, if the new *T* is smaller than the real *T*, j_i is then again increased by *x*.

Else, if the new *T* is greater than the real *T*, j_i is decreased by *x*. *x* is divided by *r* (selected by the user - e.g.: *r* = 10) and j_i it is increased by the new *x* and *T* is again computed.

Otherwise, the temperature is again computed with the newly obtained temperature. The algorithm is repeated until the real temperature is almost obtained (the difference is less than ε) or a certain number of iterations is exceeded (the number of iterations is selected by the user). If this number is exceeded, it means that the real temperature can not be obtained with the values that the other parameters have.

2. ALGORITHMS, PROGRAMS AND RESULTS

2.1 Implementing the Solution for a Heat Differential Equation

The algorithm for the proposed solution is:

compute k_1 and k_2

if $k_2 > 0$
then $T \leftarrow T^+$
else if $k_1 = 0$
then $T \leftarrow T^0$
else $T \leftarrow T^-$

where

$$T^+ = -\frac{\dfrac{k_2}{k_1} - T_2 + \left(T_1 - \dfrac{k_2}{k_1}\right)\cdot e^{-l\cdot\sqrt{k_1}}}{e^{l\cdot\sqrt{k_1}} - e^{-l\cdot\sqrt{k_1}}}\cdot e^{x\cdot\sqrt{k_1}} +$$

$$+\frac{\dfrac{k_2}{k_1} - T_2 + (T_1 - \dfrac{k_2}{k_1})\cdot e^{l\cdot\sqrt{k_1}}}{e^{l\cdot\sqrt{k_1}} - e^{-l\cdot\sqrt{k_1}}}\cdot e^{-x*\sqrt{k_1}} + \frac{k_2}{k_1} \tag{24}$$

$$T^0 = -\frac{k_2}{2}\cdot x^2 + \frac{T_2 - T_1 + \dfrac{k_2}{2}\cdot l^2}{l}\cdot x + T_1 \tag{25}$$

$$T^- = \frac{T_2 + \dfrac{k_2}{k_3} - \left(T_1 + \dfrac{k_2}{k_3}\right)\cdot\cos(\sqrt{-k_1}\cdot l)}{\sin(\sqrt{-k_1}\cdot l)}\cdot$$

$$\cdot\sin\left(\sqrt{-k_1}\cdot x\right) - \frac{k_2}{k_3} + \left(T_1 + \frac{k_2}{k_3}\right)\cdot\cos\left(\sqrt{-k_1}\cdot x\right) \tag{26}$$

The implementation of this algorithm in C++ (Schilling 2000) is shortly presented bellow. The computing of k_1 and k_2 will not be presented because this part uses the particular equations for the two variables and thus it is of no interest to us.

```
-----------------------------------------------------------------
long intIter=0;
        // stores the number of iterations


double   m_Tm, m_Te, m_Tm, m_l, m_d, m_I;
```

```
// environment,

// ends, middle temperatures, length, diameter, current
int k;            // is k1 from equation (1)
CString strFinalValue;   //it will be used to format the
// values to be  shown

double temp (int k1)
{
switch (k1):
{
        case -1: return T1 =…
// the value T⁻ computed at 2.1
        case  0: return  T2 =…
// the value T⁰ computed at 2.1
        case  1: return  T3 =…
// the value T⁺ computed at 2.1
}
}
-----------------------------------------------------------------

T=temp(k);
// temp(int) is the function that actually computes the
// temperature; k is an integer variable which shows if
// k1 is positive, negative or zero

-----------------------------------------------------------------
if((m_Tm<0)||(intIter>suprem))
        m_dTf="Can not be computed";
//m_dTf is a CString
// variable associated to the combobox on the form
else
{
        strFinalValue.Format("%.2f",m_Tm);
        m_dTf=strFinalValue;
}
UpdateData(false);
```

2.2 Programming Algorithms for Computing Variables from Auto - Dependent Relations

➢ An algorithm to implement the proposed solution would be:

```
T←min(T1,T2)
Repeat
        T0←T
        Compute T
        it←it+1
Until T-T0 < ε or it >100000
```

where *T1* and *T2* are the variables for the temperatures at the ends, ε is a very small number (10^{-8}) and *it* is a variable that counts the iterations.

➢ An algorithm to implement the proposed solution for the increasing dependency would be:

```
j←0
x←0

Repeat
        j←j+x
        Compute T
        it←it+1
        if Tr > T+ε
                j←j-x
                x←x/10
Until T-Tr<ε or it>1000000
```

T_0	T_1	T_2	*I*	*L*	*d*	d_i	d_i /*d*	*iter*
20	24	42	50	4.5	***2.18***	***1.09***	0.5	***40***
21	25	45	150	6	***3.87***	***0.77***	0.2	***27***
60	70	80	100	8	***3.42***	***2.73***	0.8	***48***
20	24	42	***49.8***	4.5	2.18	1.09	0.5	***29***
21	25	45	***149.7***	6	3.87	0.77	0.2	***25***
60	70	80	***100.8***	8	3.42	2.73	0.8	***18***
20	24	***42.3***	50	4.5	2.18	1.09	0.5	***4***
21	25	***45.5***	150	6	3.87	0.77	0.2	***4***
60	70	***82.1***	100	8	3.42	2.73	0.8	***5***

Table 1

where j is the variable j_i from the paragraph 1.2 (it is T dependent), x is the ratio, and Tr is the real temperature.

2.3 Results

Here, the case of a electrical conductor heated from both ends with the same temperature as the result of current passing through (see figure 1) is taking into account.

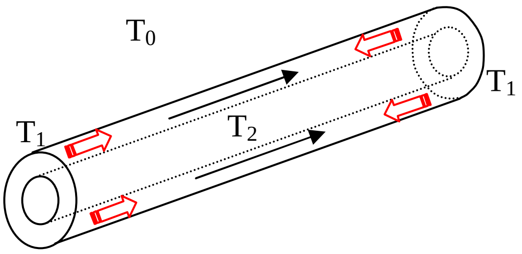

Figure 1 A conductor heated by the passing current

In the following table, table1, are listed computational examples. The computed variables are printed in bold & italic characters while the given ones are printed in normal characters.
where:

- T_0 – environment temperature
- T_1 – temperature at the ends of the conductor
- T_2 – temperature in the middle of the conductor
- I – current
- l – length
- d – diameter
- d_i – interior diameter
- *iter* – number of iterations to compute the wanted value

It follows two graphic examples concerning the computation. In figure 2, the Series 1, 2 and 3 plot the function $iter\left(\frac{I \cdot l}{d \cdot T}\right)$ when d, I respectively T_2 are computed.

In figure 3 on the X axis is the current multiplied by length over diameter, while on the Y axis is the diameter, dT is the temperature difference between the ends of the conductor and its middle and δ, which takes values in the range 0..0.9, represents the ratio between the interior and full diameter.

3. CONCLUSIONS AND FURTHER DEVELOPMENT

Problems concerning heat come up in certain engineering applications. The process of solving these problems exceeds the technical area and is to be found in mathematical problems that can be considered individually (Lienhard 2001).

There have been presented two such problems. Both problems were given theoretical solutions and these solutions were implemented using computing programs.

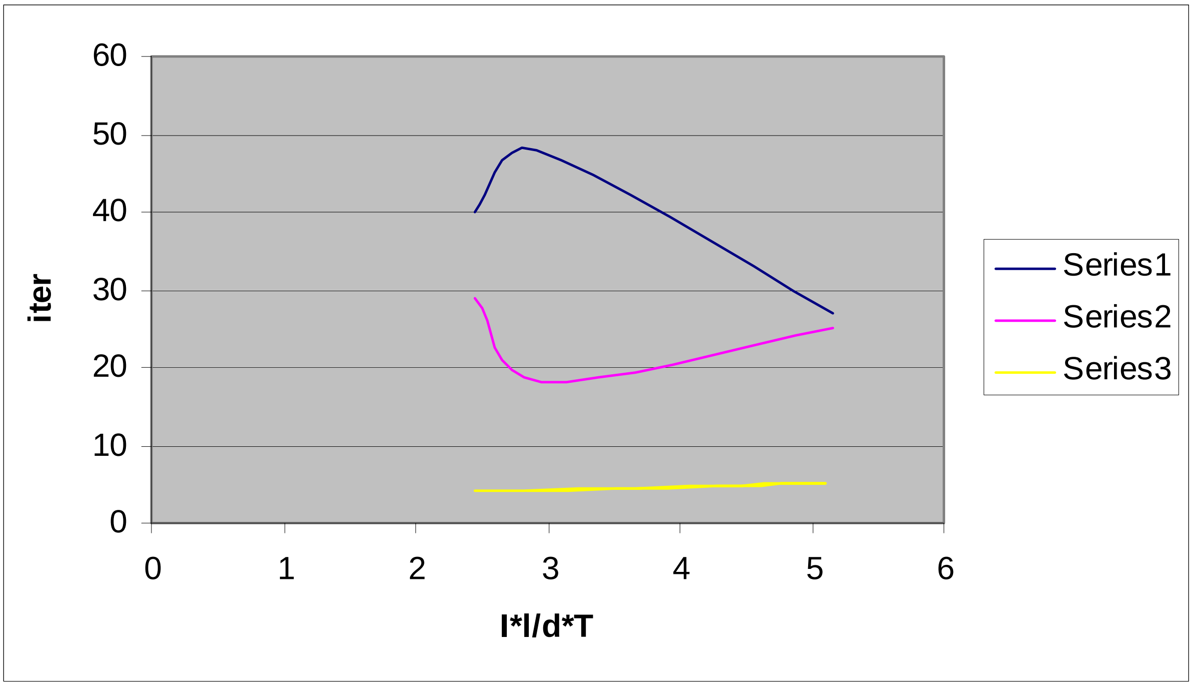

Figure 2 The variation of the number of iterations

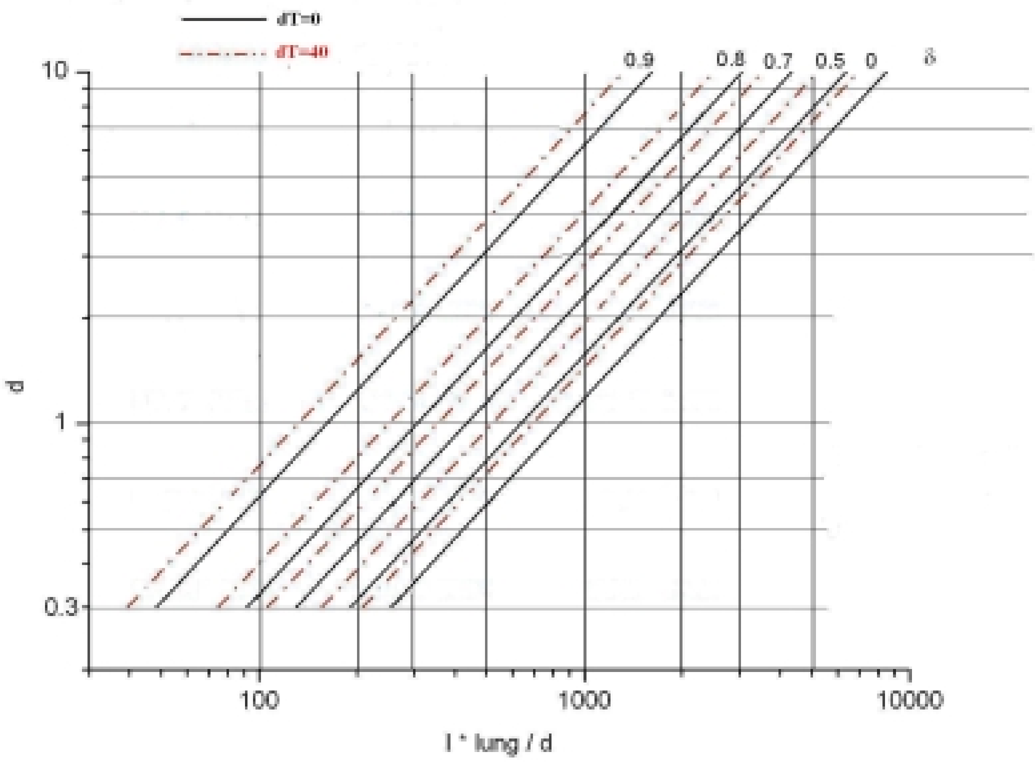

Figure 3 The values of d for ratios $\delta = d_i/d$

The solutions found for the given problems have a high degree of generality, which allows them to be used in different applications where heat problems come up, but only if the initial specified conditions are similar.

Further developments can be achieved taking into consideration the time dependency in the heat equation. When considering time, the equations will be different and also the programs will have slight differences compared to the ones presented by this paper.

REFERENCES

Betounes, D., *Partial Differential Equation for Computational Science,* New York, Springer Verlag,1998.

Lienhard IV, Lienhard V, J.H., *A Heat Transfer Text Book, Third Edition*, Phlogiston Press, Cambridge Massachuetts, 2001.

Press, W.H., Teukolski, A.S., Vetterling, T.W., Flannery., B. P., *Numerical Recipes in C,* Cambridge University Press,1997.

Schilling, R.J., Harris, S.L., *Applied Numerical Methods for Engineerig Using Matlab and C*, Brooks/Cole, 2000.

Teodorescu, N., Olariu, V., *Ecuatii diferentiale si cu derivate partiale, Editura Tehnica*, Bucuresti, 1980

Tveito, A., Winther, R., *Introduction to Partial Diferential Equation*; *A Computation Approach*, New York, Springer, 1998.