

## X22B10T0 problem

**Slab with jump in heat flux at one boundary, zero heat flux at other boundary and initially at zero temperature<sup>1</sup>**

### Contents

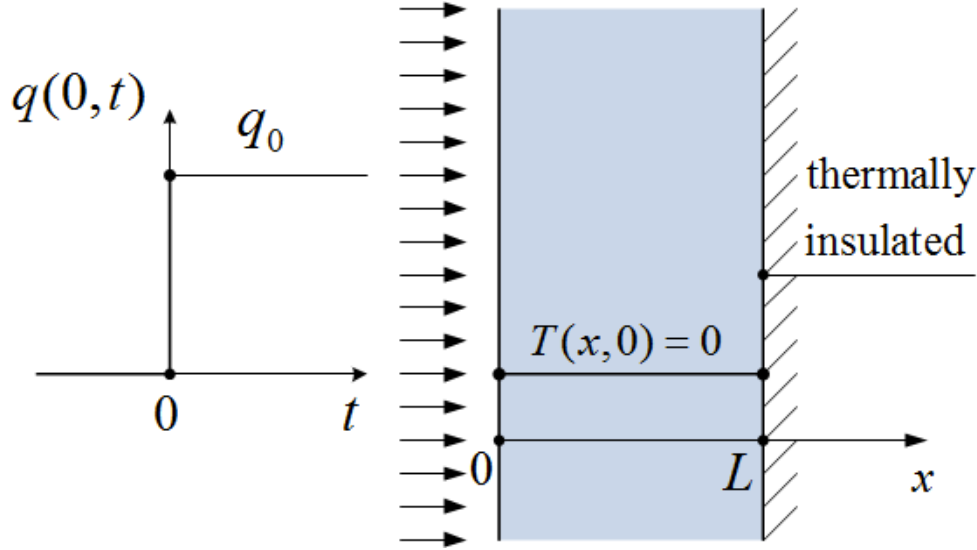
1. Problem description .....	2
2. Dimensional governing equations .....	2
3. Dimensionless variables .....	3
4. Dimensionless governing equations .....	3
5. Dimensionless temperature and heat flux solutions .....	3
5.1. Simplified solutions with errors less than 0.0025 .....	6
6. Plots and tables of dimensionless temperature and heat flux .....	6
Appendix. Matlab function .....	12

---

<sup>1</sup> **Copyright © by Filippo de Monte and James V. Beck.** Cite article as follows: *X22B10T0 Slab with jump in heat flux at one boundary, zero heat flux at other boundary and initially at zero temperature*, Exact Analytical Conduction Toolbox, exact.unl.edu, February 1, 2013, pp. 1-14.

## 1. Problem description

Slab of thickness  $L$ , initially at zero temperature and with temperature-independent properties, subject to a step change in heat flux at  $x = 0$  with the boundary at  $x = L$  thermally insulated, as depicted in Fig. 1.



**Fig. 1** – Schematic of the 1D transient X22B10T0 problem.

## 2. Dimensional governing equations

The mathematical formulation of the problem is

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (0 < x < L; t > 0) \quad (1a)$$

$$-k \left( \frac{\partial T}{\partial x} \right)_{x=0} = q_0 \quad (t > 0) \quad (1b)$$

$$\left( \frac{\partial T}{\partial x} \right)_{x=L} = 0 \quad (t > 0) \quad (1c)$$

$$T(x, 0) = 0 \quad (0 < x < L) \quad (1d)$$

### 3. Dimensionless variables

We have a total of four dimensionless groups

$$\tilde{T} = \frac{T}{q_0 L / k}, \quad \tilde{q} = \frac{q}{q_0} = -\frac{\partial \tilde{T}}{\partial \tilde{x}}, \quad \tilde{x} = \frac{x}{L}, \quad \tilde{t} = \frac{\alpha t}{L^2} = Fo, \quad (2)$$

where  $q_0 L / k$  is the temperature at  $x = 0$  of the steady-state X21B10 problem,  $\tilde{x} \in [0, 1]$  and  $Fo$  is the well-known Fourier number ( $\geq 0$ ).

### 4. Dimensionless governing equations

The mathematical formulation in dimensionless form is

$$\frac{\partial^2 \tilde{T}}{\partial \tilde{x}^2} = \frac{\partial \tilde{T}}{\partial \tilde{t}} \quad (0 < \tilde{x} < 1; \tilde{t} > 0) \quad (3a)$$

$$-\left(\frac{\partial \tilde{T}}{\partial \tilde{x}}\right)_{\tilde{x}=0} = 1 \quad (\tilde{t} > 0) \quad (3b)$$

$$\left(\frac{\partial \tilde{T}}{\partial \tilde{x}}\right)_{\tilde{x}=1} = 0 \quad (\tilde{t} > 0) \quad (3c)$$

$$\tilde{T}(\tilde{x}, 0) = 0 \quad (0 < \tilde{x} < 1) \quad (3d)$$

### 5. Dimensionless temperature and heat flux solutions

The solution to the current X22B10T0 problem is a well-established *exact* analytical solution available in the heat conduction literature [1, 2]. The solution is unique but two different forms of it exist.

- The *short-time* solution comes from the application of Laplace transform to the governing equations. It is valid at any time but it is computationally convenient at short times. It is given in Ref. [1, p. 112, Eq. (4)] for the X22B01T0 case. To obtain the solution of the current X22B10T0 problem, the space variable  $x$  has simply to be replaced by  $L - x$ .
- The *large-time* solution comes from the application of separation-of-variables (SOV) method to the defining equations. It is given in Ref. [2, p. 207, Eq. (6.94)] and is valid at any time though it is computationally convenient at large times.

The *computational* analytical solution is given below for different accuracies  $10^{-A}$  with  $A = 2, 3, \dots, 15$ . Note that  $A = 2$  (accuracy of 1 %) is for visual comparison and is acceptable in many engineering applications, while  $A = 15$  (machine accuracy) is for verification purposes of large numerical codes [3].

1. For  $0 \leq \tilde{t} \leq \tilde{t}_d^{(1)}$ , where  $\tilde{t}_d^{(1)} = (2 - \tilde{x})^2 / (10A)$  [4], we have (errors less than  $10^{-A}$ )

$$\tilde{T}(\tilde{x}, \tilde{t}) \approx 2\sqrt{\tilde{t}} \operatorname{ierfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) \quad (0 \leq \tilde{x} \leq 1) \quad (4a)$$

$$\tilde{q}(\tilde{x}, \tilde{t}) \approx \operatorname{erfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) \quad (0 \leq \tilde{x} \leq 1) \quad (4b)$$

where  $\operatorname{ierfc}(z)$  is the complementary error function integral defined by Eq. (4c) of the X20B1T0 problem. The computational solution has only one short-time term.

2. For  $\tilde{t}_d^{(1)} < \tilde{t} \leq \tilde{t}_d^{(2)}$ , where  $\tilde{t}_d^{(2)} = (2 + \tilde{x})^2 / (10A)$ , we have (errors less than  $10^{-A}$ )

$$\tilde{T}(\tilde{x}, \tilde{t}) \approx 2\sqrt{\tilde{t}} \operatorname{ierfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) + 2\sqrt{\tilde{t}} \operatorname{ierfc}\left(\frac{2 - \tilde{x}}{2\sqrt{\tilde{t}}}\right) \quad (0 \leq \tilde{x} \leq 1) \quad (5a)$$

$$\tilde{q}(\tilde{x}, \tilde{t}) \approx \operatorname{erfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) - \operatorname{erfc}\left(\frac{2 - \tilde{x}}{2\sqrt{\tilde{t}}}\right) \quad (0 \leq \tilde{x} \leq 1) \quad (5b)$$

The computational solution has two short-time terms.

3. For  $\tilde{t} > \tilde{t}_d^{(2)}$ , we have (with errors less than  $10^{-A}$ )

$$\tilde{T}(\tilde{x}, \tilde{t}) \approx \left( \tilde{t} + \frac{\tilde{x}^2}{2} - \tilde{x} + \frac{1}{3} \right) - 2 \sum_{m=1}^{m_{\max}} \frac{\cos(\beta_m \tilde{x})}{\beta_m^2} \exp(-\beta_m^2 \tilde{t}) \quad (0 \leq \tilde{x} \leq 1) \quad (6a)$$

$$\tilde{q}(\tilde{x}, \tilde{t}) \approx (1 - \tilde{x}) - 2 \sum_{m=1}^{m_{\max}} \frac{\sin(\beta_m \tilde{x})}{\beta_m} \exp(-\beta_m^2 \tilde{t}) \quad (0 \leq \tilde{x} \leq 1) \quad (6b)$$

where  $\beta_m = m\pi$  is the  $m$ -th eigenvalue ( $m = 1, 2, 3, \dots$ ) and  $\cos(\beta_m \tilde{x})$  is the corresponding  $m$ -th eigenfunction.

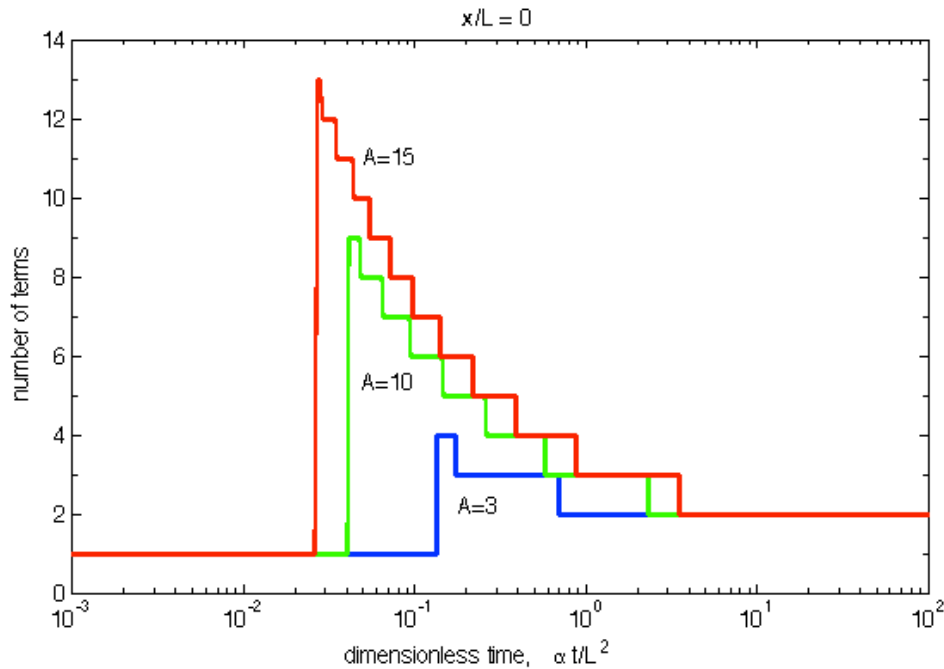
The computational solution has  $m_{\max} + 1$  terms, in detail  $m_{\max}$  large-time terms and one quasi-steady term (which is actually steady-state for the heat flux), where  $m_{\max}$  is given by

$$m_{\max} = \text{ceil} \left[ \frac{1}{\pi} \left( \frac{A \ln 10}{\tilde{t}} \right)^{1/2} \right] \quad (A = 2, 3, \dots, 15; \tilde{t} > \tilde{t}_d^{(2)}) \quad (7)$$

The function “ceil( $z$ )” is a Matlab function that rounds the number  $z$  to the nearest integer greater than or equal to  $z$ . Also, the error in neglecting the tail of the summation associated to Eqs. (6a) and (6b) is always less than  $10^{-A}$  for  $\tilde{t} > \tilde{t}_d^{(2)}$ .

The time  $\tilde{t}_d^{(2)}$  may be considered as the partition time, that is, the time matching the short-time terms and the large-time and quasi-steady (or steady-state) ones.

A plot of the number of terms appearing in the computational analytical solutions listed before is shown in Fig. 2 for the heated surface  $\tilde{x} = 0$ , which is the location requiring the major number of terms. (Note that, in the current case, this location is of interest only for the temperature as the heat flux is prescribed there.)



**Fig. 2** – Number of terms in the computational analytical temperature solution versus time for three different accuracies.

## 5.1. Simplified solutions with errors less than 0.0025

For  $0 \leq \tilde{x} \leq 1$ , we have:

$$\tilde{T}(\tilde{x}, \tilde{t}) \approx \begin{cases} 2\sqrt{\tilde{t}} \operatorname{ierfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) + 2\sqrt{\tilde{t}} \operatorname{ierfc}\left(\frac{2-\tilde{x}}{2\sqrt{\tilde{t}}}\right) & \text{for } 0 \leq \tilde{t} \leq 0.15 \\ \left(\tilde{t} + \frac{\tilde{x}^2}{2} - \tilde{x} + \frac{1}{3}\right) - 2 \frac{\cos(\pi\tilde{x})}{\pi^2} \exp(-\pi^2\tilde{t}) & \\ - \frac{\cos(2\pi\tilde{x})}{2\pi^2} \exp(-4\pi^2\tilde{t}) & \text{for } \tilde{t} > 0.15 \end{cases} \quad (8a)$$

$$\tilde{q}(\tilde{x}, \tilde{t}) \approx \begin{cases} \operatorname{erfc}\left(\frac{\tilde{x}}{2\sqrt{\tilde{t}}}\right) - \operatorname{erfc}\left(\frac{2-\tilde{x}}{2\sqrt{\tilde{t}}}\right) & \text{for } 0 \leq \tilde{t} \leq 0.15 \\ (1-\tilde{x}) - 2 \frac{\sin(\pi\tilde{x})}{\pi} \exp(-\pi^2\tilde{t}) & \\ - \frac{\sin(2\pi\tilde{x})}{\pi} \exp(-4\pi^2\tilde{t}) & \text{for } \tilde{t} > 0.15 \end{cases} \quad (8b)$$

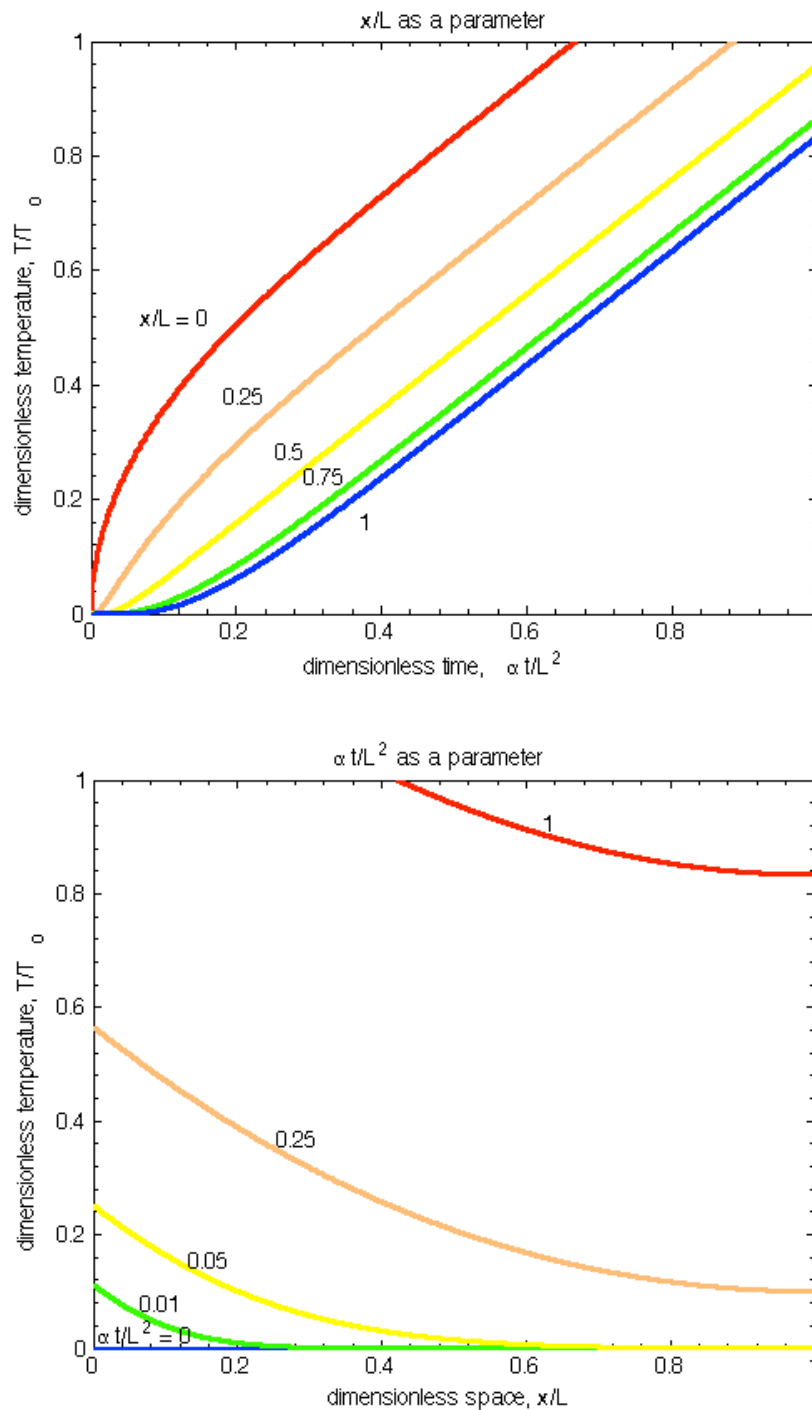
## 6. Plots and tables of dimensionless temperature and heat flux

By using the computational analytical solutions defined in Section 5 and implemented in Matlab ambient, as shown in Appendix, 2D plots, 3D plots and tables of temperature and heat flux can be derived. In detail,

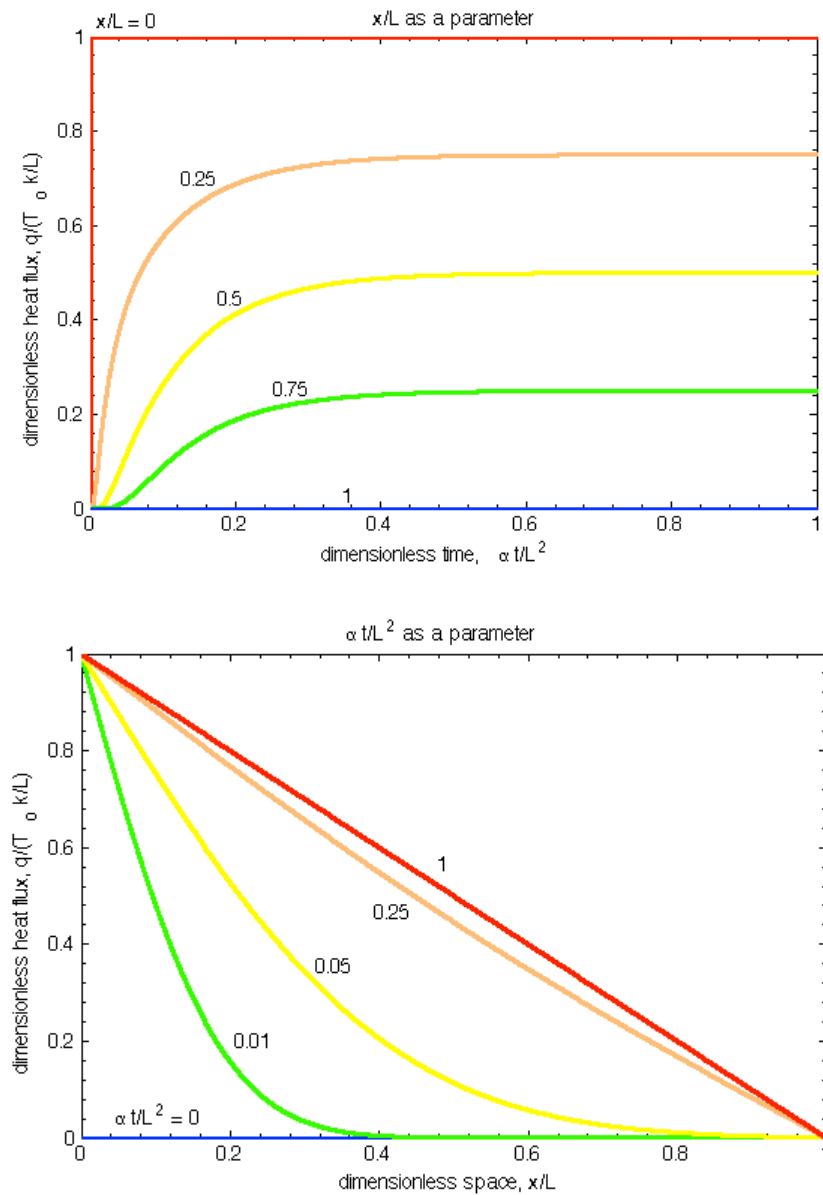
✚ 2D plots of temperature and heat flux are shown in Figs. 3 and 4.

✚ 3D plots of temperature and heat flux are given in Figs. 5 and 6.

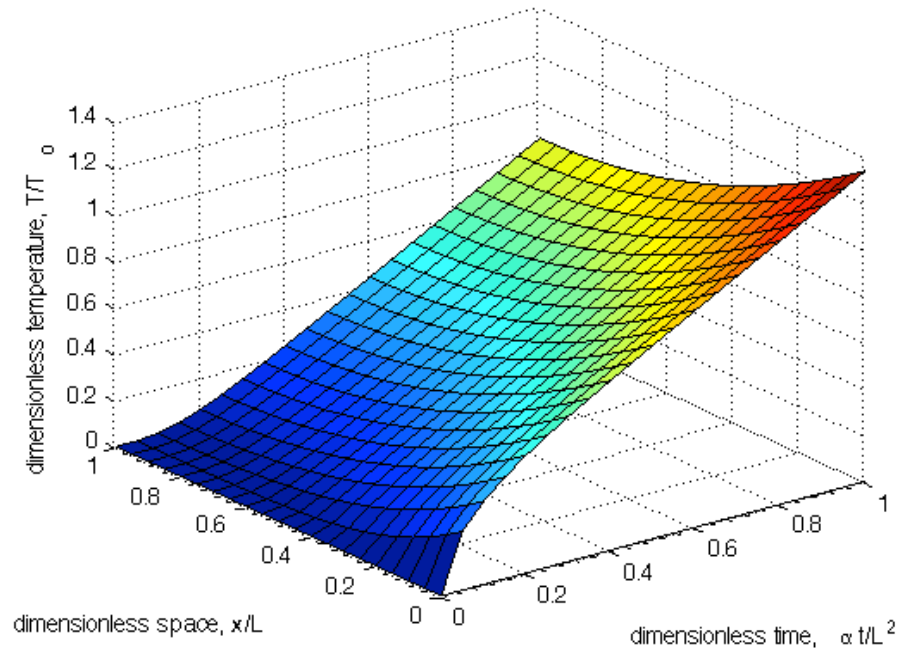
✚ Tables 1 and 2 provide numerical values of temperature and heat flux.



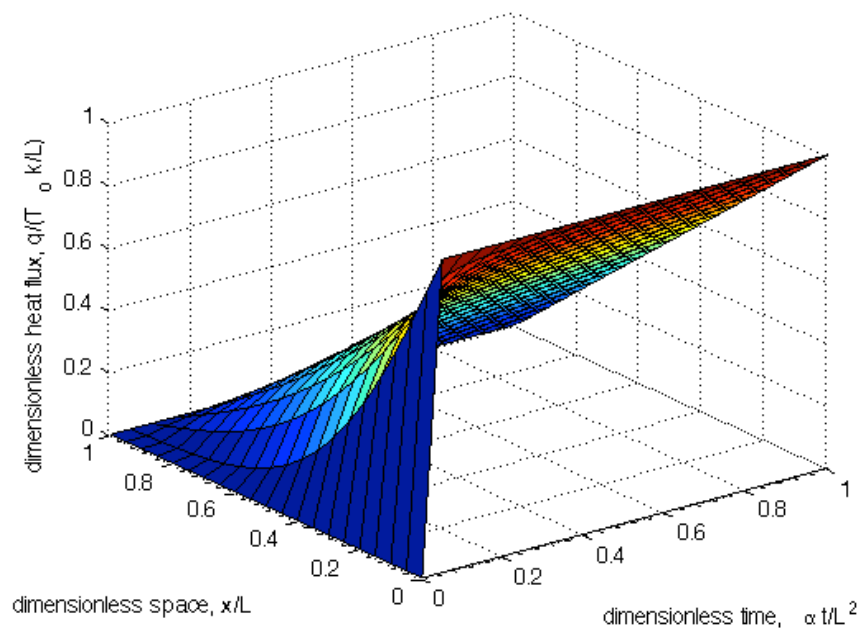
**Fig. 3** – Temperature graphs.



**Fig. 4** – Heat flux graphs.



**Fig. 5** – Temperature surface plot.



**Fig. 6** – Heat flux surface plot.

**Table 1** – Numerical values of temperature.

$\tilde{t}$	$\tilde{x} = 0$	$\tilde{x} = 0.25$	$\tilde{x} = 0.50$	$\tilde{x} = 0.75$	$\tilde{x} = 1.0$
0.000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.001	0.03568248	0.00000000	0.00000000	0.00000000	0.00000000
0.005	0.07978846	0.00040083	0.00000001	0.00000000	0.00000000
0.010	0.11283792	0.00437714	0.00001435	0.00000000	0.00000000
0.020	0.15957691	0.02023475	0.00080165	0.00000841	0.00000004
0.030	0.19544100	0.03923836	0.00372182	0.00015023	0.00000484
0.040	0.22567583	0.05851013	0.00875429	0.00070229	0.00005741
0.050	0.25231325	0.07729750	0.01536594	0.00187860	0.00026934
0.060	0.27639532	0.09540469	0.02307429	0.00376392	0.00078555
0.070	0.29854108	0.11280721	0.03152836	0.00635986	0.00173473
0.080	0.31915391	0.12953703	0.04048632	0.00962972	0.00320662
0.090	0.33851415	0.14564372	0.04978409	0.01352296	0.00525097
0.100	0.35682625	0.16118032	0.05931089	0.01798635	0.00788529
0.110	0.37424506	0.17619814	0.06899204	0.02296853	0.01110419
0.120	0.39089169	0.19074503	0.07877719	0.02842164	0.01488727
0.130	0.40686340	0.20486485	0.08863241	0.03430182	0.01920511
0.140	0.42224011	0.21859759	0.09853486	0.04056907	0.02402351
0.150	0.43708878	0.23197957	0.10846913	0.04718709	0.02930629
0.160	0.45146649	0.24504372	0.11842483	0.05412294	0.03501718
0.170	0.46542247	0.25781991	0.12839499	0.06134676	0.04112089
0.180	0.47899972	0.27033519	0.13837488	0.06883148	0.04758385
0.190	0.49223611	0.28261407	0.14836133	0.07655259	0.05437457
0.200	0.50516519	0.29467879	0.15835220	0.08448788	0.06146375
0.250	0.56614563	0.35243165	0.20833595	0.12673502	0.10051579
0.300	0.62284151	0.40716475	0.25833370	0.17200191	0.14382443
0.350	0.67692827	0.46005430	0.30833338	0.21911236	0.18973830
0.400	0.72942308	0.51181837	0.35833334	0.26734830	0.23724357
0.450	0.78094613	0.56289533	0.40833333	0.31627134	0.28572053
0.500	0.83187595	0.61355281	0.45833333	0.36561386	0.33479071
0.550	0.88244361	0.66395420	0.50833333	0.41521247	0.38422306
0.600	0.93279016	0.71419925	0.55833333	0.46496742	0.43387651
0.650	0.98300172	0.76434885	0.60833333	0.51481782	0.48366494
0.700	1.03313089	0.81444018	0.65833333	0.56472648	0.53353578
0.750	1.08320974	0.86449594	0.70833333	0.61467073	0.58345693
0.800	1.13325788	0.91452998	0.75833333	0.66463669	0.63340879
0.850	1.18328727	0.96455076	0.80833333	0.71461591	0.68337940
0.900	1.23330521	1.01456345	0.85833333	0.76460322	0.73336146
0.950	1.28331616	1.06457119	0.90833333	0.81459547	0.78335050
1.000	1.33332285	1.11457592	0.95833333	0.86459074	0.83334381
$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

**Table 2** – Numerical values of heat flux.

$\tilde{t}$	$\tilde{x} = 0$	$\tilde{x} = 0.25$	$\tilde{x} = 0.50$	$\tilde{x} = 0.75$	$\tilde{x} = 1.0$
0.000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.001	1.00000000	0.00000002	0.00000000	0.00000000	0.00000000
0.005	1.00000000	0.01241933	0.00000057	0.00000000	0.00000000
0.010	1.00000000	0.07709987	0.00040695	0.00000011	0.00000000
0.020	1.00000000	0.21129955	0.01241933	0.00017683	0.00000000
0.030	1.00000000	0.30743417	0.04122683	0.00219931	0.00000000
0.040	1.00000000	0.37675912	0.07709976	0.00800005	0.00000000
0.050	1.00000000	0.42919527	0.11384420	0.01762884	0.00000000
0.060	1.00000000	0.47048598	0.14889977	0.03007478	0.00000000
0.070	1.00000000	0.50403296	0.18138821	0.04418540	0.00000000
0.080	1.00000000	0.53195893	0.21112271	0.05901467	0.00000000
0.090	1.00000000	0.55565280	0.23818588	0.07388364	0.00000000
0.100	1.00000000	0.57605950	0.26275627	0.08834391	0.00000000
0.110	1.00000000	0.59384324	0.28503833	0.10212079	0.00000000
0.120	1.00000000	0.60948439	0.30523485	0.11506201	0.00000000
0.130	1.00000000	0.62333960	0.32353680	0.12709794	0.00000000
0.140	1.00000000	0.63568044	0.34012014	0.13821291	0.00000000
0.150	1.00000000	0.64671872	0.35514554	0.14842516	0.00000000
0.160	1.00000000	0.65662353	0.36875907	0.15777337	0.00000000
0.170	1.00000000	0.66553277	0.38109328	0.16630756	0.00000000
0.180	1.00000000	0.67356114	0.39226833	0.17408321	0.00000000
0.190	1.00000000	0.68080563	0.40239315	0.18115742	0.00000000
0.200	1.00000000	0.68734950	0.41156643	0.18758654	0.00000000
0.250	1.00000000	0.71180789	0.44601148	0.21184081	0.00000000
0.300	1.00000000	0.72669155	0.46704011	0.22669613	0.00000000
0.350	1.00000000	0.73577132	0.47987805	0.23577195	0.00000000
0.400	1.00000000	0.74131357	0.48771559	0.24131366	0.00000000
0.450	1.00000000	0.74469697	0.49250039	0.24469698	0.00000000
0.500	1.00000000	0.74676251	0.49542150	0.24676252	0.00000000
0.550	1.00000000	0.74802352	0.49720484	0.24802352	0.00000000
0.600	1.00000000	0.74879336	0.49829356	0.24879336	0.00000000
0.650	1.00000000	0.74926335	0.49895822	0.24926335	0.00000000
0.700	1.00000000	0.74955028	0.49936400	0.24955028	0.00000000
0.750	1.00000000	0.74972545	0.49961172	0.24972545	0.00000000
0.800	1.00000000	0.74983238	0.49976296	0.24983238	0.00000000
0.850	1.00000000	0.74989767	0.49985529	0.24989767	0.00000000
0.900	1.00000000	0.74993753	0.49991165	0.24993753	0.00000000
0.950	1.00000000	0.74996186	0.49994606	0.24996186	0.00000000
1.000	1.00000000	0.74997672	0.49996707	0.24997672	0.00000000
$\infty$	1.00000000	0.75000000	0.50000000	0.25000000	0.00000000

## Appendix. Matlab function

### **fdX22B10T0**

Heat conduction function for the X22B10T0 case.

### **Syntax**

```
[Td, qd] = fdX22B10T0(xd, td, A)
```

### **Description**

`fdX22B10T0(xd, td, A)` returns the dimensionless temperature  $Td$  and heat flux  $qd$  solutions at a given dimensionless location  $xd$  from the heated surface, between 0 and 1, and at a given dimensionless time  $td$ , with an accuracy of  $10^{-4}$  ( $A = 2, 3, \dots, 15$ ), for the X22B10T0 problem.

If  $xd$  and  $td$  are not single values but arrays ( $\text{length}(xd) = n$  and  $\text{length}(td) = m$ ) defining the dimensionless locations and times of interest, respectively, the above function returns the dimensionless temperature  $Td$  and heat flux  $qd$  as double subscripted arrays, where  $\text{size}(Td) = \text{size}(qd) = [m, n]$ .

### **Examples**

#### Example 1

```
>> [Td, qd]=fdX22B10T0(.25,.1,10)
```

```
Td =
```

```
0.161180315836933
```

```
qd =
```

```
0.576059497948475
```

## Example 2

```
>> A=15
```

```
A =
```

```
15
```

```
>> xd=[0.1 0.5 0.7]'
```

```
xd =
```

```
0.10000000000000000
```

```
0.50000000000000000
```

```
0.70000000000000000
```

```
>> td=[0.01 0.2]'
```

```
td =
```

```
0.01000000000000000
```

```
0.20000000000000000
```

```
>> [Td, qd]=fdX22B10T0(xd,td,A)
```

```
Td =
```

```
0.039928245674849    0.000014352414313    0.000000019773817
```

```
0.411546515326888    0.158352196668220    0.094884894165447
```

```
qd =
```

```
0.479500122186953    0.000406952017445    0.000000743098372
```

```
0.872602854083542    0.411566430126192    0.228568455486985
```



## Matlab function: fdX22B10T0.m

```
% fdX22B10T0 function
% Revision History
% 1 28 2013 James V. Beck, Matthew P. Lempke and Filippo de Monte
% INPUTS:
% xd: dimensionless location starting at xd=0 and ending at xd=1
% td: dimensionless time starting at td=0
% A: desired accuracy (1E-A = 10^-A); A = 2, 3, ..., 15
% OUPUTS:
% Td: dimensionless temperature calculated at (xd,td) to desired accuracy A
% qd: dimensionless heat flux calculated at (xd,td) to desired accuracy A
% Calling Sequence:
% ierfc(z) for computing the complementray error function integral
function [Td,qd]=fdX22B10T0(xd,td,A)
lengthx=length(xd);
lengtht=length(td);
Td=zeros(lengtht,lengthx); % Preallocating Arrays for speed
qd=zeros(lengtht,lengthx); % Preallocating Arrays for speed
for it=1:lengtht % Begin time loop
    td_it=td(it); % Set current time
    for ix=1:lengthx % Begin space loop
        xd_ix=xd(ix); % Set current space
        td_dev1=(1/(10*A))*(2-xd_ix)^2; % First deviation time
        td_dev2=(1/(10*A))*(2+xd_ix)^2; % Second deviation time
        if td_it == 0 % For time t=0 condition
            Td(it,ix)=0; % Set inital temperature
            qd(it,ix)=0; % Set inital heat flux
        elseif td_it <= td_dev1
            % Solution for first small times:
            Td(it,ix)=sqrt(4*td_it)*ierfc(xd_ix/sqrt(4*td_it));
            qd(it,ix)=erfc(xd_ix/sqrt(4*td_it));
        elseif td_it > td_dev1 && td_it <= td_dev2
            % Solution for second small times:
            Td(it,ix)=sqrt(4*td_it)*ierfc(xd_ix/sqrt(4*td_it))+...
                sqrt(4*td_it)*ierfc((2-xd_ix)/sqrt(4*td_it));
            qd(it,ix)=erfc(xd_ix/sqrt(4*td_it))-...
                erfc((2-xd_ix)/sqrt(4*td_it));
        else
            m_max=ceil(sqrt(A*log(10)/(td_it*pi^2))); %Set max.No.terms
            % Start X22B10T0 case for large times:
            Td(it,ix)=td_it+1/3-xd_ix+xd_ix^2/2; % quasi-steady T solution
            qd(it,ix)=(1-xd_ix); % steady-state heat flux solution
            for m=1:m_max % Continue X22B10T0 case for large times
                % Series solutions:
                betam=m*pi; % Define eigenvalues
                Td(it,ix)=Td(it,ix)-2*exp(-betam^2*td_it)*...
                    cos(betam*xd_ix)/betam^2;
                qd(it,ix)=qd(it,ix)-2*exp(-betam^2*td_it)*...
                    sin(betam*xd_ix)/betam;
            end % for m
        end % if td_it
        if xd_ix == 1 % For location x=1 condition
            qd(it,ix)=0; % Set x=1 boundary heat flux
        end % if xd_ix
    end % for ix
end % for it

function [ierfc]=ierfc(z)
ierfc =(1/sqrt(pi))*exp(-z^2)-z*erfc(z);
```