Conductive cooling of a finite domain

```
clear, close all, clc
set_demo_defaults()
```

Dimensional Problem

To determine the cooling timescale of a finite domain of length, L, with initial temperature, T_0, and boundary temperature, T_b, we need to solve the following Initial and Boundary Value Problem (IBVP)

PDE: $\rho c_p \frac{\partial T}{\partial t} - \nabla \cdot [\kappa \nabla T] = 0$ on $x \in [0, L]$ and $t \geq 0$.

BC's: $T(x=0,t) = T_b$ and $\nabla T \cdot \hat{\mathbf{n}}|_{x=L} = 0$.

IC: $T(x, t=0) = T_0$ on $x \in [0, L]$

Here we assume that c_p and κ are constant so that the problem is linear. Note that we assume that $T_0 > T_b$ so that $\Delta T = T_0 - T_b > 0$.

This problem has six paramters: L, c_p, ρ, κ, T_0 and T_b

To determine how many independent paramters this problem has we non-dimensionalize the variables, T, x and t by choosing the following characteristic scales.

$T' = \frac{T - T_b}{\Delta T}$, $x' = \frac{x}{L}$ and $t' = \frac{t}{t_c}$.

Here we have chosen external scales imposed by the domain size and boundary for T and x. There is no external scale for t and here we just use a generic characteristic time, t_c, that will be determined by the equation itself.

Dimensionless Problem

Substituting the generic scales we obtain the following problem

PDE: $\frac{\partial T'}{\partial t'} = \Pi \nabla'^2 T'$ on $x' \in [0, 1]$ and $t' \geq 0$.

BC's: $T'(0, t') = 0$ and $\nabla' T' \cdot \hat{\mathbf{n}}|_{x'=1} = 0$.

IC: $T'(x', 0) = 1$ on $x \in [0, 1]$

where the dimensionless group $\Pi = \frac{\kappa t_c}{\rho c_p L^2} = \frac{t_c}{t_D}$ is the ratio of t_c and the internal diffusive timescale $t_D = \frac{\rho c_p L^2}{\kappa} = \frac{L^2}{D}$, where $D = \frac{\kappa}{\rho c_p}$ is the thermal diffusivity. To simplify the problem we choose the internal diffusive timescale as characteristic scale $t_c = t_D$. So that the final dimensionless problem is parameter free and here written in one dimension as

PDE: $\frac{\partial T'}{\partial t'} = \frac{\partial^2 T'}{\partial x'^2}$ on $x' \in [0,1]$ and $t' \geq 0$.

BC's: $T'(0,t') = 0$ and $\left.\frac{\partial T'}{\partial x'}\right|_{x'=1} = 0$.

IC: $T'(x',0) = 1$ on $x \in [0,1]$

Analytic solution

The analytic solution is obtained by separation of variables, $T'(x',t') = g(t')h(x')$ and the dimensionless solution is given by

$$T'(x',t') = \frac{4}{\pi}\sum_{n=1}^{\infty}\frac{1}{2n-1}\sin\left(\left(n-\frac{1}{2}\right)\pi x'\right)\exp\left(-(n-\frac{1}{2})^2\pi^2 t'\right)$$

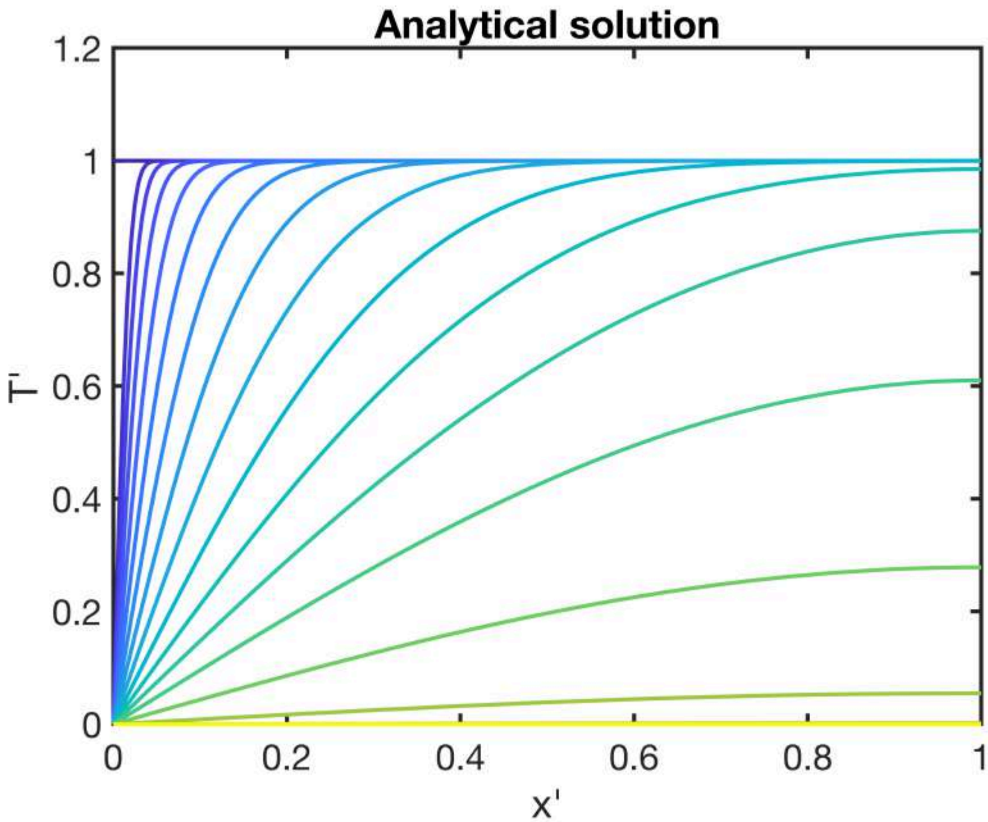
The corresponding dimensional solution is obtained by substituting the definitions of the dimensionles variables and given by

$$T(x,t) = T_b + \frac{4\Delta T}{\pi}\sum_{n=1}^{\infty}\frac{1}{2n-1}\sin\left(\frac{\left(n-\frac{1}{2}\right)\pi x}{L}\right)\exp\left(-\frac{\left(n-\frac{1}{2}\right)^2\pi^2 D}{L^2}t\right)$$

For simplicity we work with the dimensionless solution below.

```
Nt = 20; Nx = 1e3; Nn = 1e2;

tvec = logspace(-4,2,Nt);
figure
map = colormap(parula(Nt+1));
plot([0 1],[1 1],'color',map(1,:)), hold on
for i = 1:Nt
    [xd,Td] = analytic_soln_finite(tvec(i),Nx,Nn);
    plot(xd,Td,'color',map(i+1,:))
end
title 'Analytical solution'
xlabel 'x''', ylabel 'T'''
```



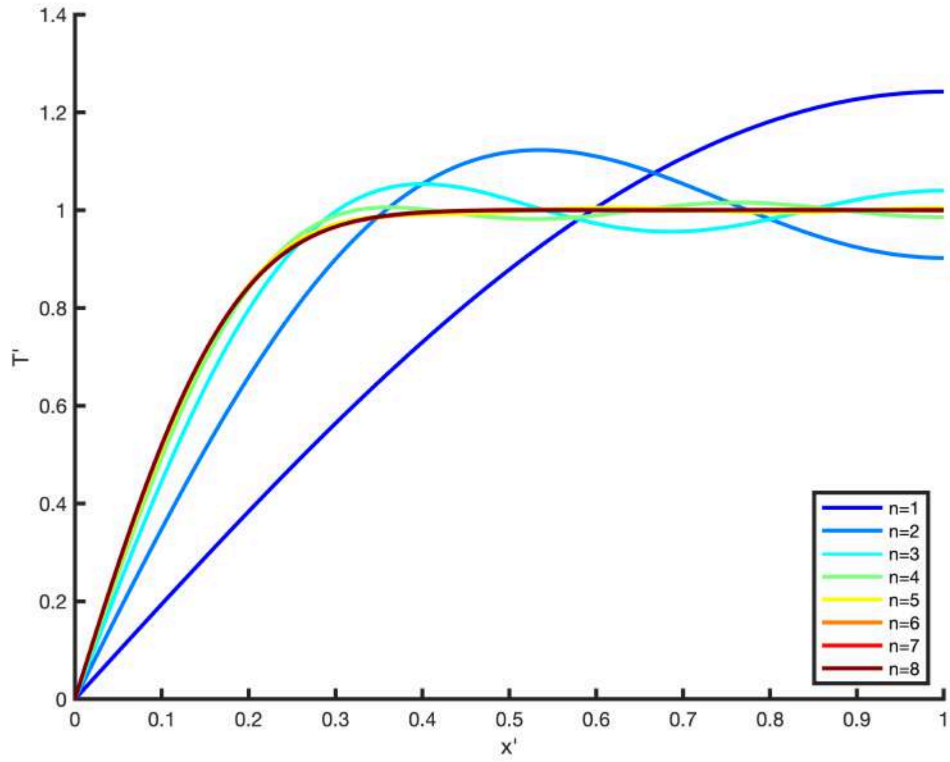

```
drawnow
```

Convergence of the series

The analytic solution is a sum of sin functions of increasing wave number that are superposed to generate the solution

```
Nx = 1e3; Nn = 8;

figure
map = colormap(jet(Nn));
for i = 1:Nn
    [xd,Td] = analytic_soln_finite(.01,Nx,i); hold on
    plot(xd,Td,'color',map(i,:))
end
legend('n=1','n=2','n=3','n=4','n=5','n=6','n=7','n=8','location','southeast')
xlabel 'x''', ylabel 'T'''
```




```
drawnow
```

Temperature decay

The contribution of each eigenfunction/mode decays proportional to its eigenvalue. The higher order modes decay quickly and at late time the solution is well approximated by the first mode.

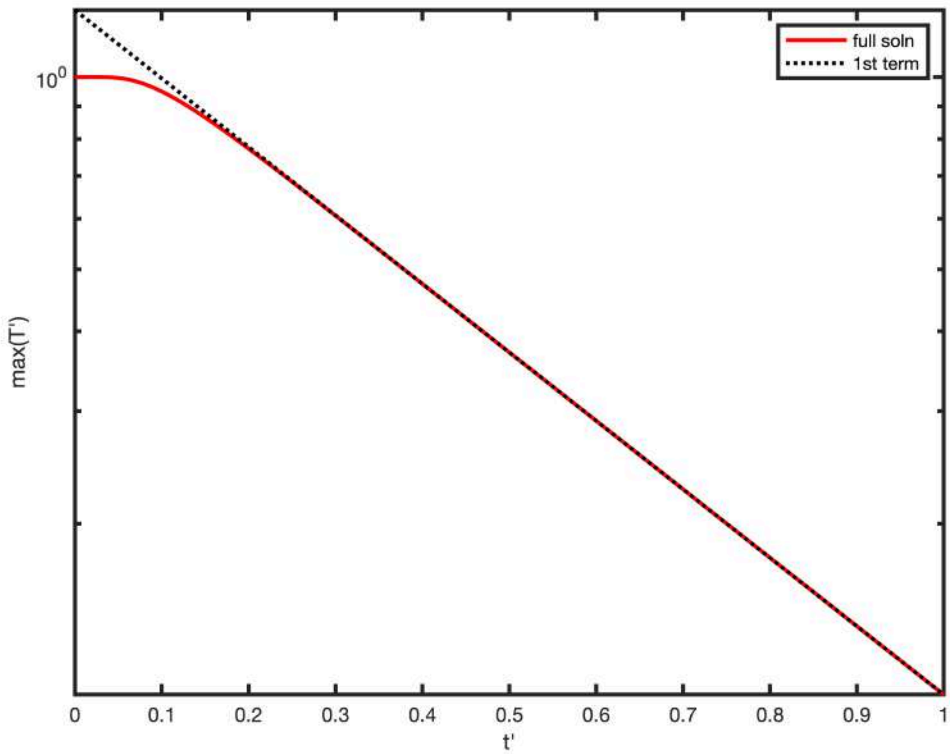
$$T'(x', t') \approx \frac{4}{\pi} \sin\left(\frac{\pi x'}{2}\right) \exp\left(-\frac{\pi^2 t'}{4}\right)$$

The figure show the decay of the maximum temperature from the full solution and the approximation

```
Nt = 200; Nx = 2;
tvec = linspace(0,1,Nt);
Td_max = zeros(Nt,1); Td_max_approx = Td_max;
figure
for i = 1:Nt
    [xd,Td] = analytic_soln_finite(tvec(i),Nx,1e3);
    Td_max(i) = Td(2);
    [xd,Td] = analytic_soln_finite(tvec(i),Nx,1);
    Td_max_approx(i) = Td(2);
end
semilogy(tvec,Td_max,'r-'), hold on
semilogy(tvec,Td_max_approx,'k:')

ylabel 'max(T'')', xlabel 't'''
```

```
legend('full soln','1st term')
```




```
drawnow
```

Heat flow (to be continued)

The heat flow at the boundary is given by Fourier's law in its dimensionless form

$$q = -\kappa \frac{\partial T}{\partial x}$$

or in dimensionless form

$$q' = -\frac{\partial T'}{\partial x'}$$

if we introduce the caracteristic scale, $q_c = K\frac{\Delta T}{L}$ for the heat flow. Differentiating the analytic solution we have

```

```

Numerical solution

Below is the code to solve the dimensionless problem numerically

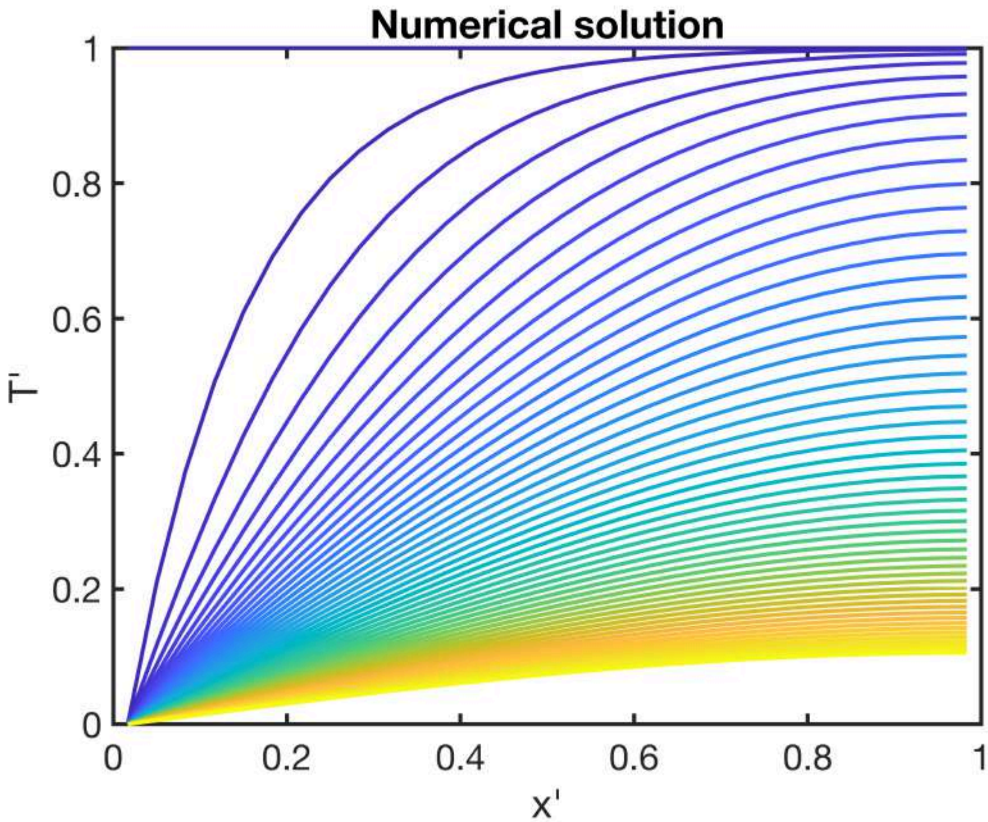
```
tmax = 1;
Nt = 50;
theta = 0;

% Grid and operators
Grid.xmin = 0; Grid.xmax = 1; Grid.Nx = 30;
Grid = build_grid(Grid);
[D,G,~,I,M] = build_ops(Grid);
L = -D*G; fs = zeros(Grid.Nx,1);
IM = @(theta,dt) I + (1-theta)*dt*L;
EX = @(theta,dt) I - theta*dt*L;

% Boundary conditions
BC.dof_dir   = Grid.dof_xmin;
BC.dof_f_dir = Grid.dof_f_xmin;
BC.g         = 0;
BC.dof_neu   = [];
BC.dof_f_neu = [];
BC.qb        = [];
[B,N,fn] = build_bnd(BC,Grid,I);

% Initial condition
u = ones(Grid.Nx,1);
figure
map = colormap(parula(Nt+1));
plot(Grid.xc,u,'color',map(1,:)), hold on

% Timestepping
dt_max = Grid.dx^2/2;
dt = tmax/Nt;
time = zeros(Nt+1,1);
for n = 1:Nt
    time(n+1) = time(n) + dt;
    u = solve_lbvp(IM(theta,dt),dt*fs+EX(theta,dt)*u,B,BC.g,N);
    plot(Grid.xc,u,'-','color',map(n+1,:)), drawnow
end
title 'Numerical solution'
xlabel 'x''', ylabel 'T'''
```




```
function [xd,Td] = analytic_soln_finite(td,Nx,Nn)
% Input:
% td = dimensionless time
% Nx = number of gridpoints
% Nn = numer of terms in the Fourier series

% Output:
% xd = dimensionless distance
% Td = dimensionless temperature
[xd] = linspace(0,1,Nx)';
Td = zeros(Nx,1);
for n = 1:Nn
    Td = Td + 1/(2*n-1)*sin((n-.5)*pi*xd)*exp(-(n-.5).^2*pi^2*td);
end
Td = 4/pi*Td;
end
```