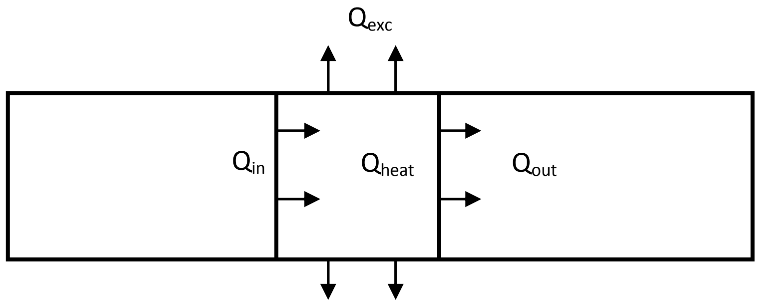
Korobka Roman

Transient analysis of temperature distribution in a disk

1. Consider the heat balance at thin circular layer in a disk with thickness of Δr from time t to $t+\Delta t$



$Q_{in} = Q_{out} + Q_{heat} + Q_{exc}$

$Q_{in} = -Sk\ \partial u(t, r)/\partial t\ \Delta t$ – heat, that comes from left side

$Q_{out} = -Sk\ \partial u(t, r+\Delta r)/\partial t\ \Delta t$ - heat, that goes to right side

$Q_{heat} = CS\Delta r\Delta u$ – heat that goes for heating of layer

$Q_{exc} = 4\pi r\Delta rW(u-\Theta)$ – heat, that goes to air through exchange

where:

- $u(t, r)$ – temperature distribution function as function of time and radius
- $S = 2\pi rh$ – cross sectional area of layer, h – disc thickness
- k – material heat conductivity
- C – material heat capacity
- Δu – temperature change in a layer
- W – heat transfer coefficient with ambient air
- Θ – air temperature

After conversion we get such **heat equation**

$$a\, u_{rr} = u_t + b\,(u - \Theta) \qquad \textbf{(1)}$$

where

- $a = k/C$ - parameter, that characterize thermal conductivity
- $u_{rr} = (1/r\ \partial u/\partial r + \partial^2 u/dr^2)$ – second derivative of u in cylindric coordinates
- $u_t = \partial u/\partial t$ – first time derivative of u
- $b = 2W/Ch$ - parameter, that characterize heat exchange to air

We have such **boundary and initial conditions**:

- at $r = r_{int}$ $T=400\ K = const$,
- at $r = r_{ext}$ heat transfer should be neglected, mean that $\partial u/\partial r|_{r\_ext} = 0$
- at $t = 0$ $u(r, 0) = \Theta$, disc was at equilibrium with air

2. To solve this equation we use a method of finite differences. Radius and time should take discrete values with step Δr and Δt respectively, and represent as a vectors $r(j)$ and $t(n)$. Temperature $u\ (t, r)$ will represent as a matrix $u(n, j)$. Let's write derivatives in term of finite differences:

$\partial u/\partial r = [\ u(n, j+1) - u(n, j)\ ] / \Delta r$

$\partial^2 u/dr^2 = [\ u(n, j+1) - 2\ u(n, j) + u(n, j-1)\ ] / \Delta r^2$

$\partial u/\partial t = [\ u(n+1, j) - u(n, j)\ ] / \Delta t$

After replacing equation (1) should rewrite in such form

$$\boldsymbol{u(n+1, j) = A\ u(n, j) + B\ u(n, j+1) + C\ u(n, j-1) + D\ \Theta} \qquad \textbf{(2)}$$

where

$A = 1 - a\Delta t/r\Delta r - 2a\ a\Delta t/\Delta r^2\ -b\Delta t$

$B = a\Delta t/r\Delta r + a\Delta t/\Delta r^2$

$C = a\Delta t/\Delta r^2$

$D = b\Delta t$

3. For modeling we need to choose value of Δr and Δt. For $\Delta r = 10^{-3}\ m$ coefficients should equal to such values:

Δt, sec	10^{-3}	10^{-4}	10^{-5}
A	0.76	0.98	~1
B	0.12	0.012	$1.2 \cdot 10^{-3}$
C	0.12	0.012	$1.2 \cdot 10^{-3}$
D	$\sim 10^{-4}$	$\sim 10^{-5}$	$\sim 10^{-6}$

4. Scripts and functions, developed in MATLAB:

initParameters – initiate parameters of system

heatDisk (a, b, t, r, init, int_temp, air_temp, transfer) – returns temperature distribution as a matrix *u*

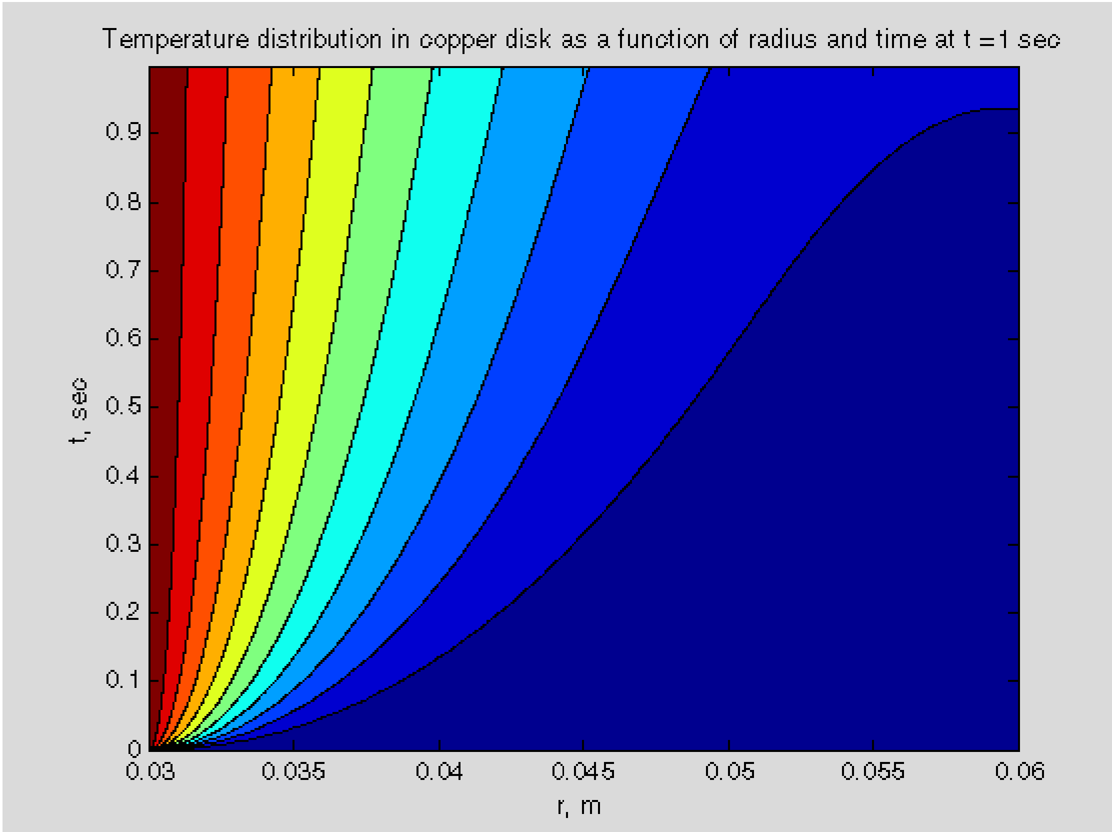
showDistribution(u_vals , r_vals , t_vals , t_moment) - returns temperature distribution at time moment *t_moment*

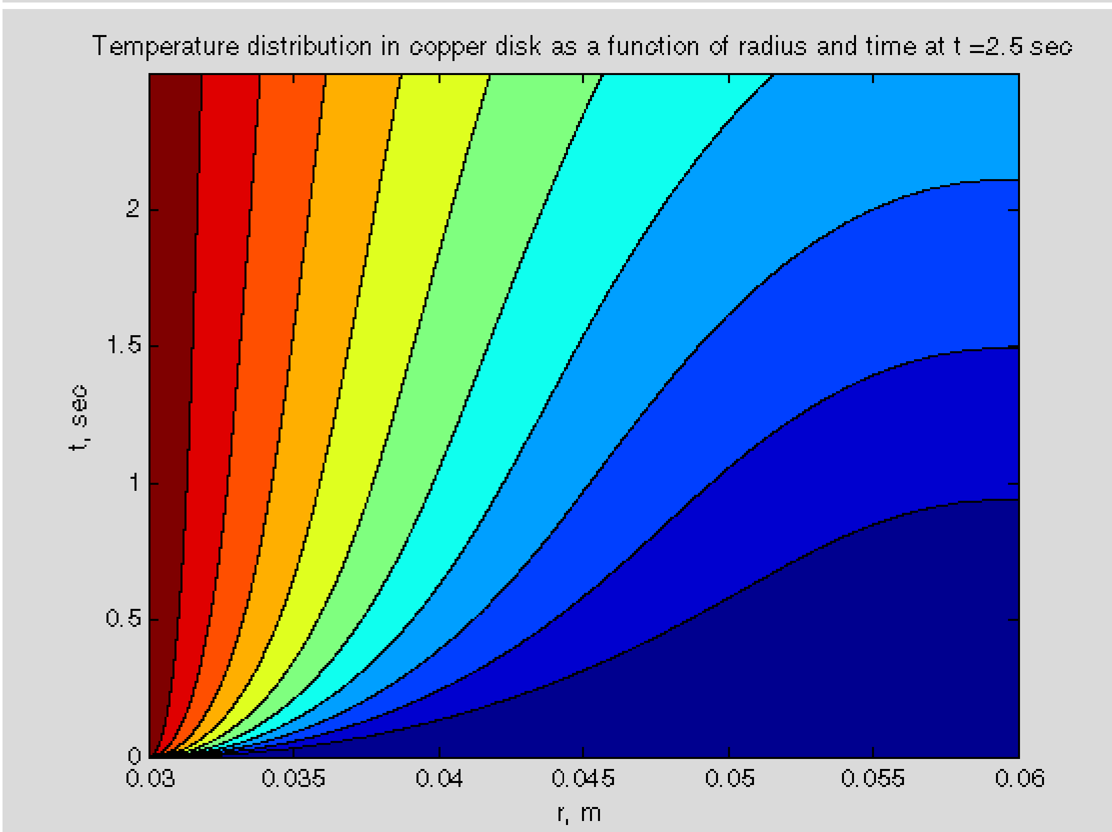
animateCylindricDistribution(u_vals , r_vals , t_vals , temp_limits, steps , contour) – returns contour or 3D animation of temperature distribution

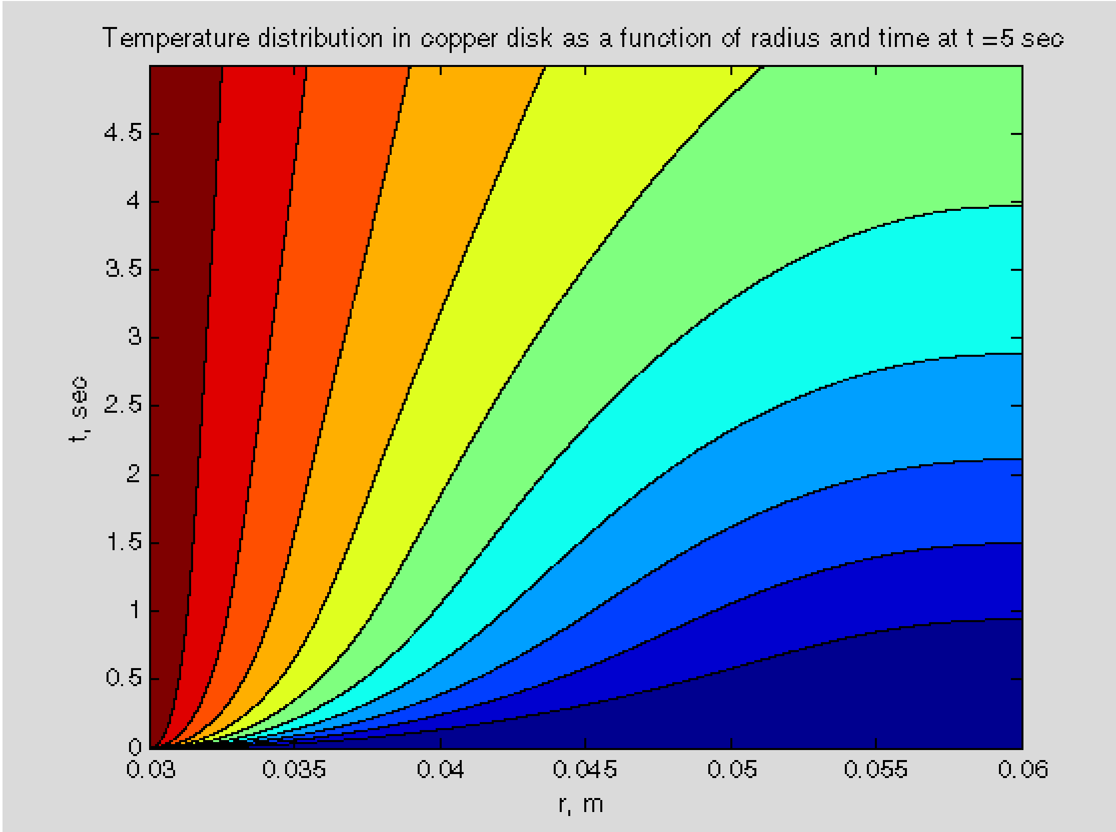
animateDistribution(u_vals , r_vals , t_vals , steps , new_figure) – returns contour 2D animation of temperature distribution

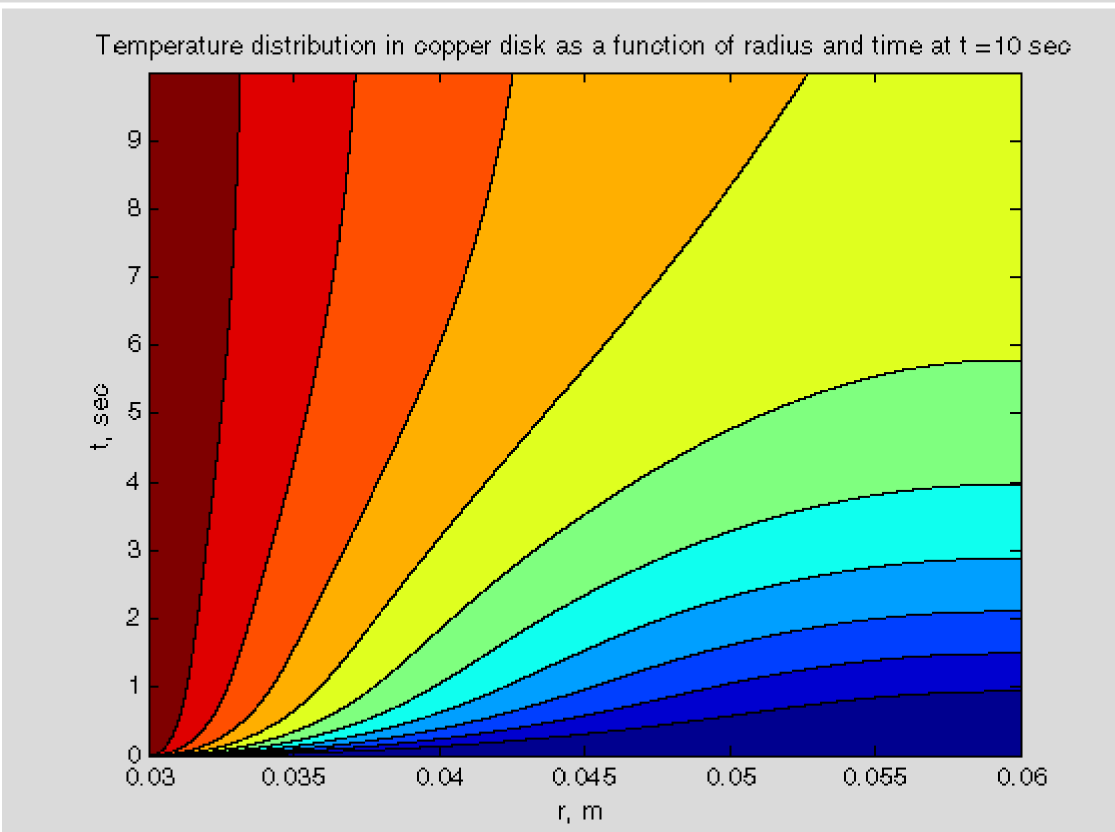
evalDistribution – evaluate all developed functions and script *initParameters*

5. Here are presented obtained distributions at different moments of time. Color changed every 10 K.









6. For obtain all graphics and animations evaluate script 'evalDistribution'.