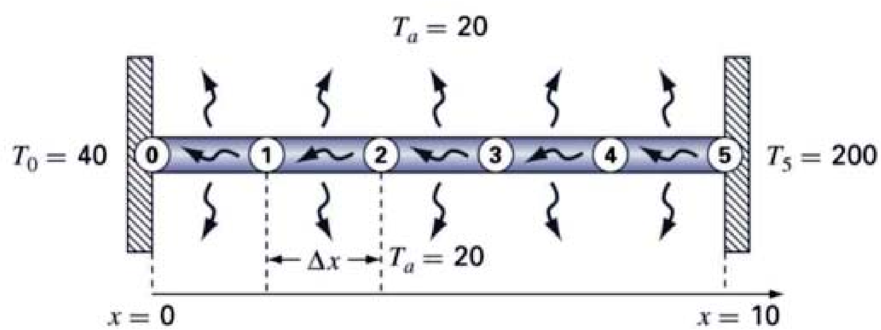
Introduction and Analytical Solution

Task 1 – Thin Rod with Unitary Convective and Heat Conduction Coefficients and No Sources

The task is to solve the 1-dimensional static (elliptic) heat equation with boundary conditions:

$$\begin{cases} \rho C \frac{\partial u}{\partial t} - \nabla \cdot (k \nabla u) = Q + h(u_{\text{ext}} - u) & \\ u(0) = 40 & \text{Diriclet' boundary condition 1} \\ u(10) = 200 & \text{Diriclet' boundary condition 2} \end{cases}$$

for the following geometry with $u_{\text{ext}} = T_a$ and $k = h = 1$, $u_{\text{ext}} = 20$ and $Q = 0$ (no sources and unitary coefficients):



Figur 1: Model problem, a noninsulated uniform rod positioned between two walls of constant but different temperature. The domain is discretized into 6 nodes.

Figure 1: Computational Domain.

We can discretize the 10-length thin rod into N elements ($N + 1$ points) $\{x_i\}_{i=0}^{N}$, with uniform step size $h = \frac{1}{N}$, denote the nodes $x_0, \ldots, x_n$. We use the Finite Element method approach to get the the approximate solution

$$\tilde{u}(x) = \sum_{j=0}^{N} c_j \phi_j(x), \tag{1}$$

where $\{\phi_i\}_{i=0}^{N}$ are the basis functions defined as $\phi_i(x_j) = 1$ if $i = j$ and 0 otherwise. We require that the differential equations and boundary conditions are satisfied at all the discretized points.

Galerkin's Method Solution

We use the Galerkin's method. We define a function space $V = \{v| < r(\tilde{u}), v >= 0 = \text{span}\{\phi_i\}_{i=0}^{N}$ so that the projection of residual to any of its elements gives 0 for all the $N+1$ points - this way we get the best possible solution. uhe basis functions $\{\phi_i\}_{i=0, i\neq j}^{N}$ are linearly independent, so it is guaranteed that we will get a unique solution.

Since we know that $u(0) = u_{\text{ext}}$ and the only nozero term in (1) in that region is $c_0\phi_0$, we get that $c_0 = 40$ and similarily $c_n = 200$. Since we have already used 2 restrictions for the terms for the first and last basis functions, we cannot demand anything else for these terms, since now our functional subspace becomes $(N-1)$-dimensional, that is $V_h = \{v = \phi_i| < r(\tilde{u}), \phi_i >= 0$ for every $i = 1, \ldots, N-1$.

Forming the residual $r = -u'' + u - u_{\text{ext}}$, we get

$$< r(\tilde{u}), v >= -\int_0^{10} u''v\mathrm{d}x + \int_0^{10} uv\mathrm{d}x - u_{\text{ext}} \int_0^{10} v\mathrm{d}x = 0$$

Integrating by parts $\int_0^{10} u''v\mathrm{d}x = vu'\Big|_0^{10} - \int_0^{10} u'v'\mathrm{d}x = -\int_0^{10} u'v'\mathrm{d}x$ since we always require the defined function space-valued functions to vanish on the boundaries.
So we get the weak formulation:

$$\int_0^{10} u'v'\mathrm{d}x + \int_0^{10} uv\mathrm{d}x = u_{\text{ext}} \int_0^{10} v\mathrm{d}x$$

$$\sum_{j=0}^{N} c_j \left(\int_0^{10} \phi_j' \phi_i' \mathrm{d}x + \int_0^{10} \phi_j \phi_i \mathrm{d}x \right) = u_{\text{ext}} \int_0^{10} \phi_i \mathrm{d}x \qquad \text{, for every } i = 1, \ldots, N-1$$

In the last we assumed that the sums exists and are finite. Since we know the boundary conditions, we separate these terms, getting:

$$\sum_{j=1}^{N-1} c_j \left(\int_0^{10} \phi_j' \phi_i' \mathrm{d}x + \int_0^{10} \phi_j \phi_i \mathrm{d}x \right) =$$
$$u_{\text{ext}} \int_0^{10} \phi_i \mathrm{d}x - u(0) \left(\int_0^{10} \phi_0 \phi_i \mathrm{d}x + \int_0^{10} \phi_0' \phi_i' \mathrm{d}x \right) - u(N) \left(\int_0^{10} \phi_n \phi_i \mathrm{d}x + \int_0^{10} \phi_n' \phi_i' \mathrm{d}x \right) \quad \text{, for every } i = 1, \ldots, N-1 \tag{2}$$

Now calculating the integrals gives:

$e = \int_0^{10} \phi_i' \phi_j' \mathrm{d}x=$

$$\begin{cases} \frac{-1}{h}, & |i-j| = 1 \\ \frac{2}{h}, & i = j \\ 0 & \text{otherwise} \end{cases}$$

and $w = \int_0^{10} \phi_i \phi_j \mathrm{d}x=$

$$\begin{cases} \frac{h}{6} = \frac{10}{6N}, & |i-j| = 1 \\ \frac{2h}{3} = \frac{20}{3N}, & i = j \\ 0 & \text{otherwise} \end{cases}$$

We see that the last terms in (2) are only nonzero for $i = 1$ and $i = N-1$ since these are the only overlaps between neighbouring basis functions.
Also $\int_0^{10} \phi_i \mathrm{d}x = \int_{x_i-1}^{x_i} \frac{1}{h} \mathrm{d}x + \int_{x_i}^{x_i+1} (1 - \frac{1}{h}) \mathrm{d}x = h$, since we have a uniform grid. So we get the matrix form of the equation (2) as

$$\begin{bmatrix} \frac{2}{h} + \frac{2}{3N} & \frac{-1}{h} + \frac{1}{6N} & 0 & \cdots & 0 & 0 \\ \frac{-1}{h} + \frac{1}{6N} & \frac{2}{h} + \frac{2}{3N} & \frac{-1}{h} + \frac{1}{6N} & \cdots & 0 & 0 \\ 0 & \frac{-1}{h} + \frac{1}{6N} & \frac{2}{h} + \frac{2}{3N} & \frac{-1}{h} + \frac{1}{6N} & 0 & 0 \\ \vdots & 0 & \cdots & \frac{-1}{h} + \frac{1}{6N} & \ddots & \frac{-1}{h} + \frac{1}{6N} \\ 0 & \cdots & 0 & \cdots & \frac{-1}{h} + \frac{1}{6N} & \frac{2}{h} + \frac{2}{3N} \end{bmatrix}$$

If we divide the rod into 6 points like in the assignment figure, then equation (2) becomes

$$\begin{bmatrix} \frac{2}{h}+\frac{2h}{3} & \frac{-1}{h}+\frac{h}{6} & 0 & 0 \\ \frac{-1}{h}+\frac{h}{6} & \frac{2}{h}+\frac{2h}{3} & \frac{-1}{h}+\frac{h}{6} & 0 \\ 0 & \frac{-1}{h}+\frac{h}{6} & \frac{2}{h}+\frac{2h}{3} & \frac{-1}{h}+\frac{h}{6} \\ 0 & 0 & \frac{-1}{h}+\frac{h}{6} & \frac{2}{h}+\frac{2h}{3} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} u_{\text{ext}}h - u(0)(\frac{h}{6}-\frac{1}{h}) \\ u_{\text{ext}}h \\ u_{\text{ext}}h \\ u_{\text{ext}}h - u(N)(\frac{h}{6}-\frac{1}{h}) \end{bmatrix} = \begin{bmatrix} 20h - 40(\frac{h}{6}-\frac{1}{h}) \\ 20h \\ 20h \\ 20h - 200(\frac{h}{6}-\frac{1}{h}) \end{bmatrix}$$

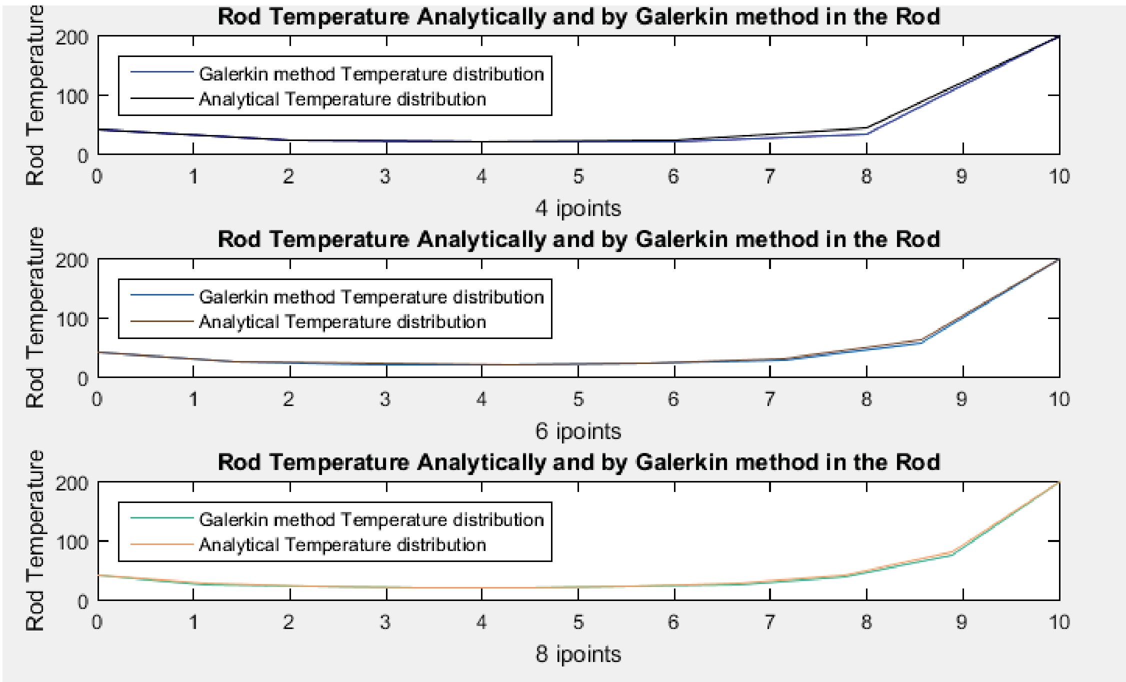


Figure 2: Error Rate Comparison for even number of points. The accuracy of Galerkin Method for the rod temperature problem is pretty good, I am satisfied with it. One could also study the norm, but I tend to like the graphical representations more.

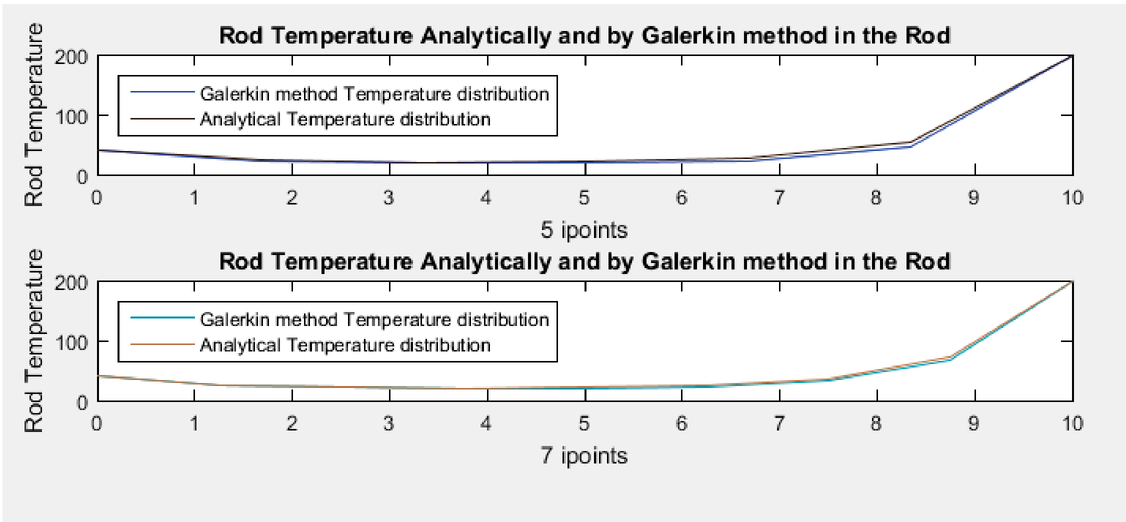


Figure 3: Error Rate Comparison for odd number of points.

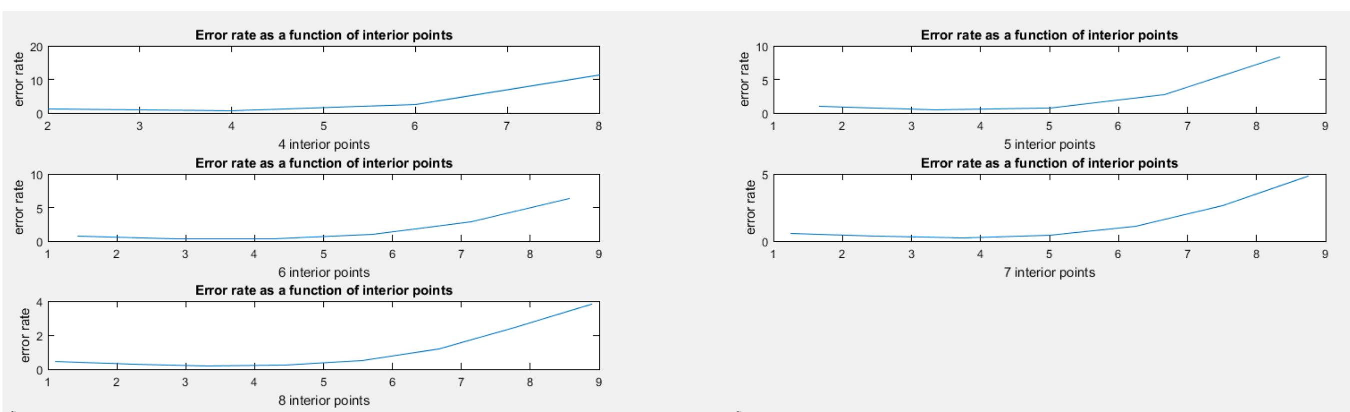


Figure 4: Galerkin Method Error Rate as a function of interior points.

Applied Heat Equation – 2D- Thin Plate Temperature Distribution for Thin Plate and My Custom Geometry

I develop the representation that is mostly useful for x-y symmetric geometry, but the Laplacian in polar coordinates should be used for the circular region. However, as the analytical part is not the focus of this course, I will just show how one could see why the heat equation in general is parabolic.

If we take $T = u$, $y = t$, then $x \to (x, z)$, we can write a general parabolic PDE of the following form:

$$\sigma \frac{\partial T}{\partial y} = k\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2}\right] + \bar{F}(x, z, T, \frac{\partial T}{\partial x}, \frac{\partial T}{\partial z}) \tag{3}$$

or

$$-k\left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2}\right] + 2 \times b \times 0 + c \times 0 = \bar{F}(x, z, T, \frac{\partial T}{\partial x}, \frac{\partial T}{\partial z}, \frac{\partial T}{\partial y}) \tag{4}$$

Compare to the general form of a 2nd order PDE

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} = F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) \tag{5}$$

We see that there is no explicit dependence on time, but only on its first derivative. the solution is smoothed over space. By comparing to (5)we see that $c = b = 0$. If

$$\begin{cases} \rho C \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q + h(T_{\text{ext}} - T) & \\ 3\vec{n} \cdot \nabla T + 2T = 2T_1 = 400 & \text{on} \quad \Gamma_1 \\ \vec{n} \cdot \nabla T = 0 & \text{on} \quad \Gamma_2 \\ 3\vec{n} \cdot \nabla T + 2T = 2T_3 = 800 & \text{on} \quad \Gamma_3 \end{cases}$$

is the stationary heat equation problem, the the parabolic problem in MATLAB PDEToolbox form when $\rho = C = 1$ The form can be obtained from parabolic PDE

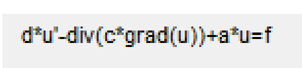


Figure 5: Parabolic Equation in MATLAB PDEToolbox Form.

form (4) by taking $k = 3$-term to LHS and taking $\frac{\partial T}{\partial x} = \frac{\partial T}{\partial y} = 0$

$$\frac{\partial T}{\partial t} - \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial z}\right) \cdot 3\left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial z}\right) = 2(T_{\text{ext}} - T)$$

for a retangular region or

$$\frac{\partial T}{\partial t} - \nabla \cdot 3\nabla T = 2(T_{\text{ext}} - T)$$

for a general region. It is convenient that one can write the general form into MATLAB's PDEToolbox without having to specify different forms of Laplacian for each region.

Task 1 – Stationary Solution

The problem is to solve the heat equation

$$\begin{cases} \rho C \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q + h(T_{\text{ext}} - T) & \\ 3\vec{n} \cdot \nabla T + 2T = 2T_1 = 400 & \text{on} \quad \Gamma_1 \\ \vec{n} \cdot \nabla T = 0 & \text{on} \quad \Gamma_2 \\ 3\vec{n} \cdot \nabla T + 2T = 2T_3 = 800 & \text{on} \quad \Gamma_3 \end{cases}$$

`stationary` form in 2D, with no sources, i.e. $Q = 0$ (we use $k = 3$ and $h = 2$.) We know that $b = 0$ and $b^2 - ac < 0$, so we have an **elliptic PDE**.
The equation claims that the temperature time dynamics is caused by sources of flux vector $-k\nabla T$, by sources or by convectional heat transfer.

Thus, the equation takes form $-3\nabla \cdot (\nabla T) = 2(T_{\text{ext}} - T)$, so we get

$$-3\triangle T + 2T = 580 \tag{6}$$

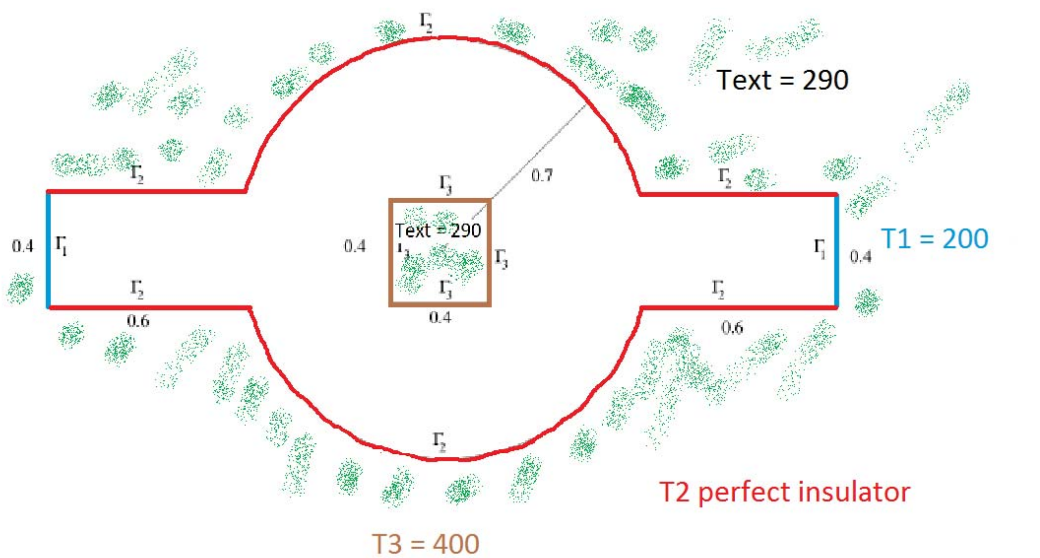


Figure 6: Thin plate with a whole in the Center.

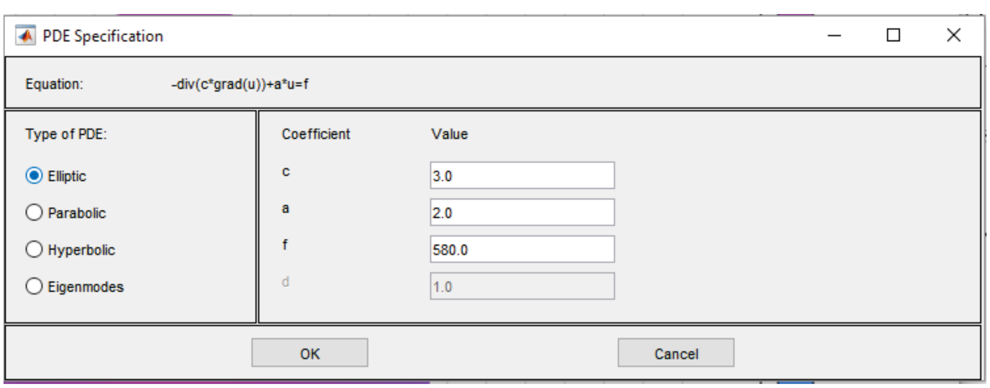


Figure 7: The Equation Parameters in MATLAB PDEToolbox with a generic template seems to give the same result as the heat equation specifier.

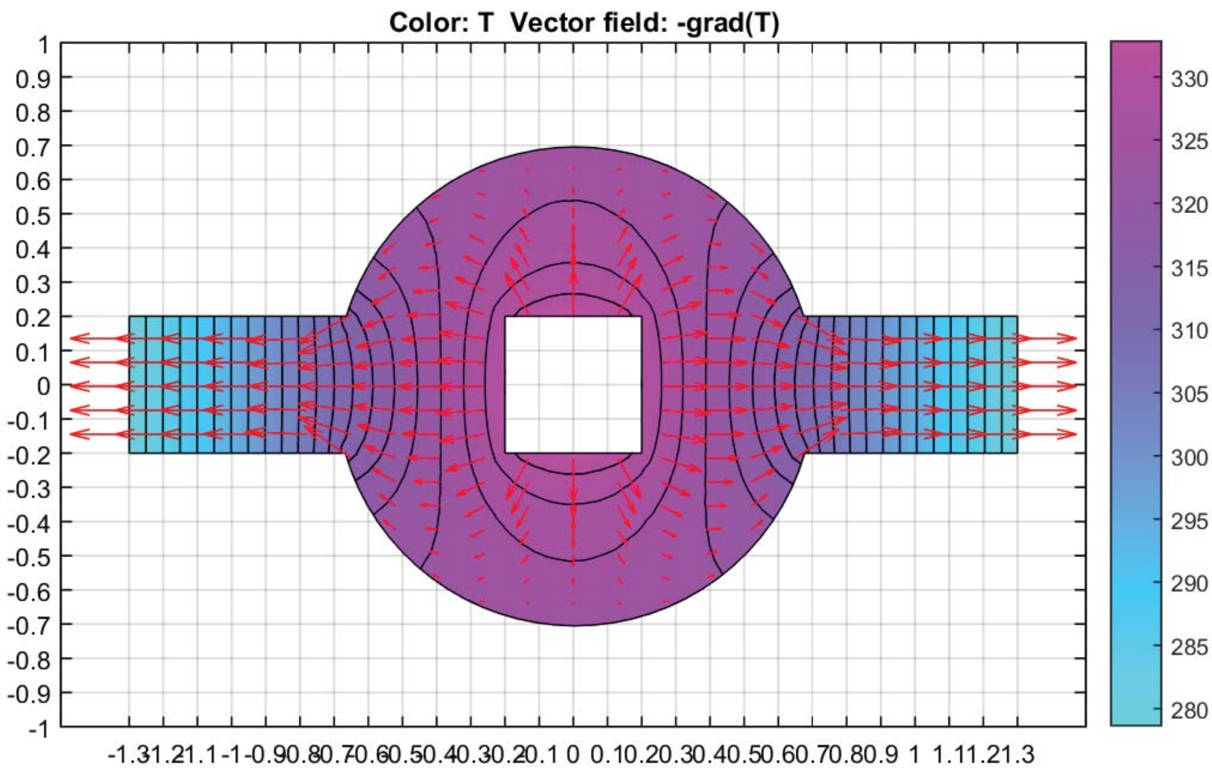


Figure 8: A generic solution for the time independent Elliptical form of the Heat Equation. This is what the solution for a generic Poisson Equation + linear term would look like, for example for the electrostatic potential as well. Isocontour lines are outlined in black .

Task2 – Time Dependent Solution

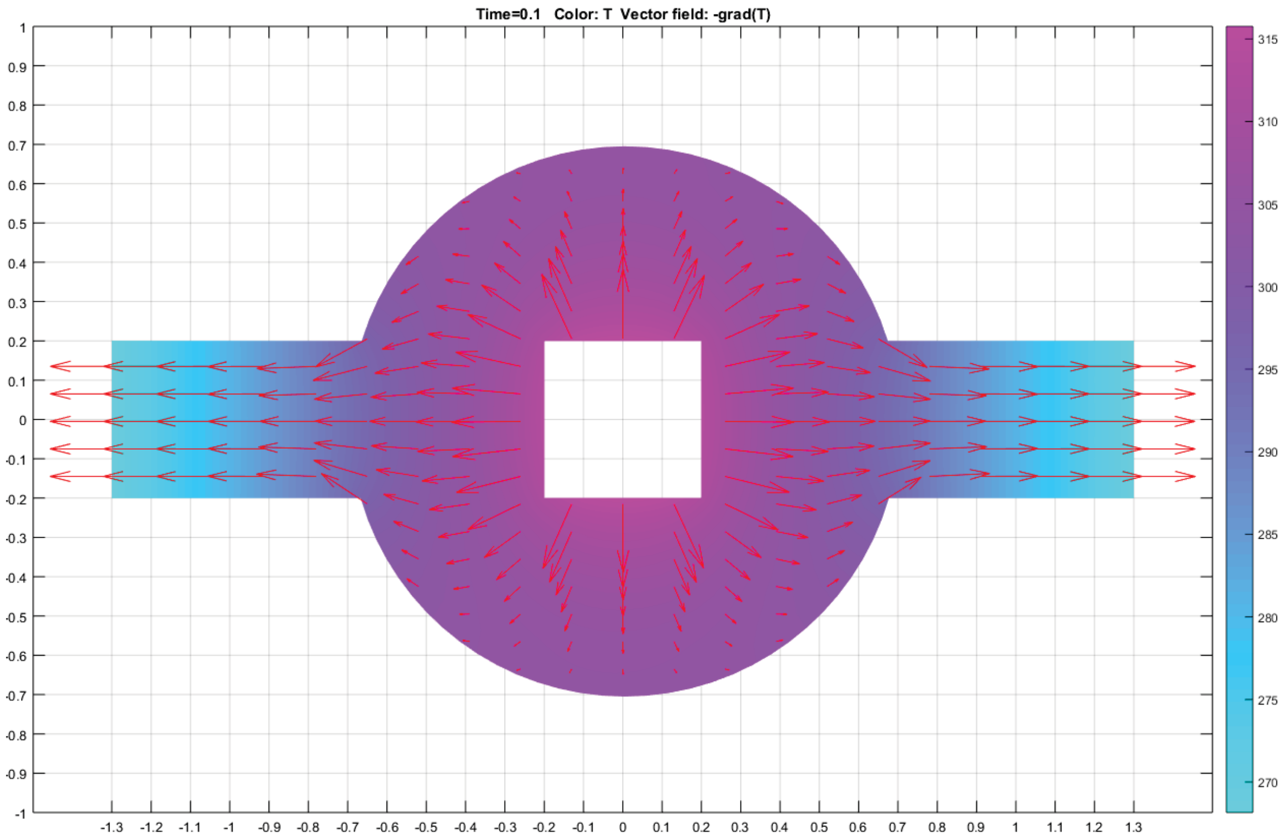


Figure 9: PDEToolbox Heat Equation time dependent solution, $t = 0.1$ sec.

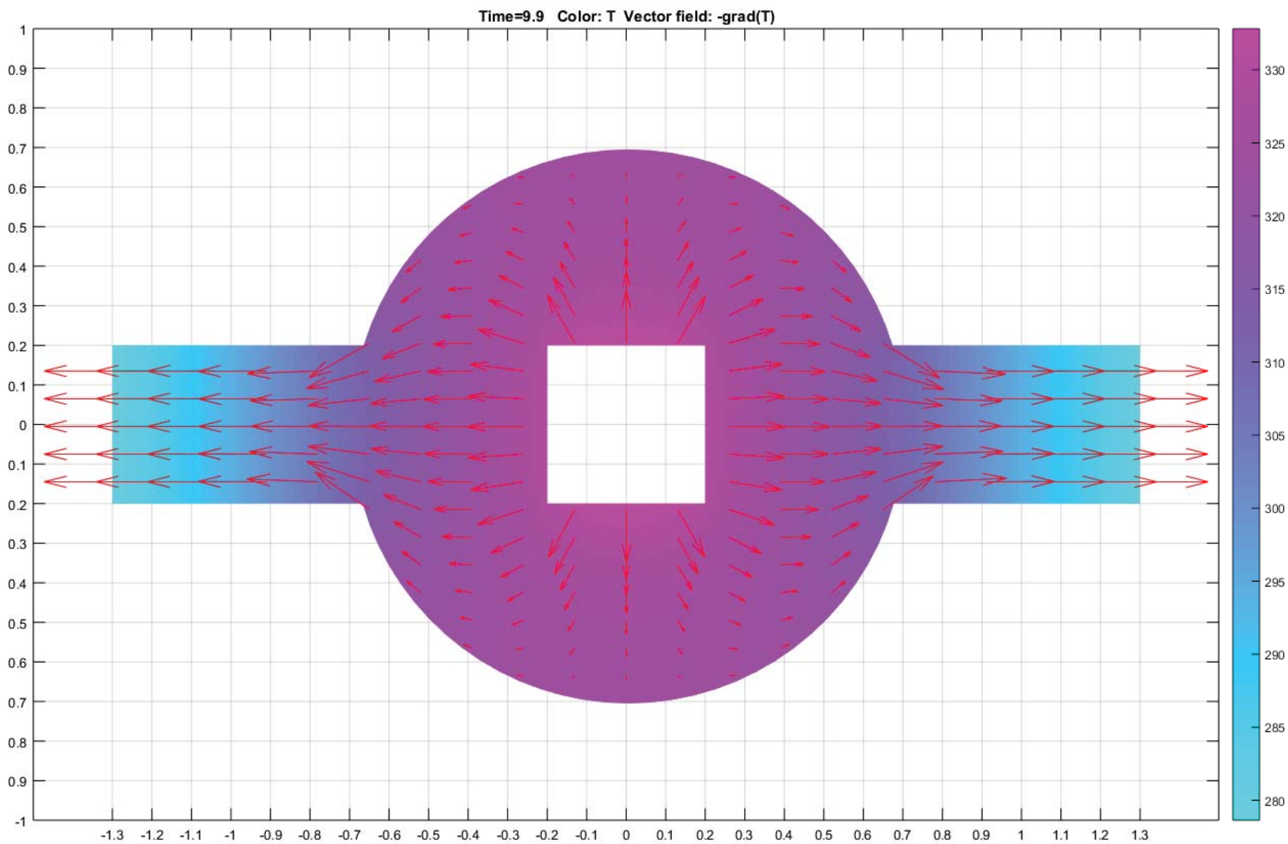


Figure 10: The solution seems to become stable at approximately 1.0 sec since I can no longer observe any changes in the colorbar from that time onwards. The solution plotted here corresponds to step size 0.01.

I have plotted the gradient of temperature vector field (it's opposite direction corresponds to the *flux vector (by Fourier' law)*) and the temperature distribution. We see indeed the evident parabolic equation property – the solution becomes smoothed out over the defined geometry with quite small range variance, from about 278 to 335 Kelvins. The solution seems physically viable since the initial condition for the inner boundary Γ_3 specified a high temperature (400 Kelvins) and at Γ_1 the temperature was initially 200 Kelvins. The smoothing means that due to heat transfer by convection from the environment and heat conduction in our domain, the hotter areas get colder and colder areas get hotter. We observe that the gradient vector field is always pointing perpendicular to the isotemperature lines and they are directed from the center, where it is hotter, to the sides where it is colder.

Task 3 – Heat Equation with Custom 2D Geometry

Aerospace industry has a hollow titanium disk that has a rectangular cut inside. The external temperature for some reason is very hot, 800 Kelvins. The density of titanium is $4.5g/cm^3$. For titanium, the specified heat capacity is very large, 520 $J/(kg \times K)$, it means that it takes a lot of energy to heat it. Heat conduction coefficient which determines how the material responds to changes in temperature within the body, is 19, so titanium has a heat conduction coefficient which is much larger than in our sample problem, which means that it should be more sensitive to heat transfer within the body itself. The heat transfer coefficient (convection coefficient) of ammonia heater is taken to be 500 and it is supposed that the titanium thin disk is inside the ammonium heater – the external temperature is taken to be 800K, whereas the titanium initial temperature is taken to be 20 Kelvins, which is pretty cold, and a heat leak is happening somewhere in the body, so $Q = -10$. We assume that the outer boundary is thermally isolated from the heating environment, so that $n \cdot \nabla T = 0$ on the outer boundary of the disk. On the inner boundary, we have a $\vec{n} \cdot (k\nabla T) = h(T_0 - T)$, where $k = 19$ (heat conduction coefficient) and the inner boundary is kept at a low temperature of 20K, so $T_0 = 20$.
In general, initially the titanium disk is very cold, and it even has a heat leak $Q = -10$ inside, but the external temperature is very hot – 800 Kelvins, but the heat flow can only happen through the inner boundary since the outer boundary is a perfect insulator. On the inner boundary, we have to use the titanium heat conduction coefficient $k = 19$ and ammonium heat transfer coefficient $h = 500$. We wish to model the parabolic heat equation, so we are interested how the solution is smoothed out across our defined geometry.

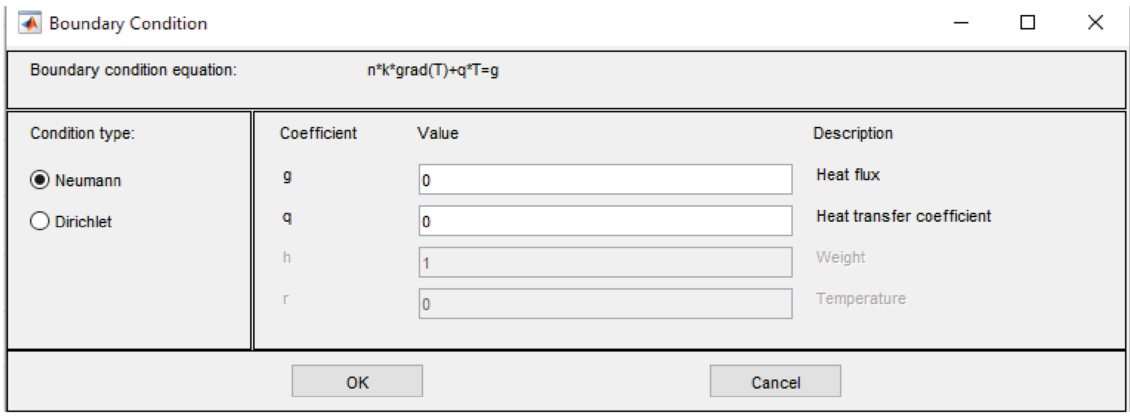


Figure 11: Outer Boundary conditions – no heat exchange.

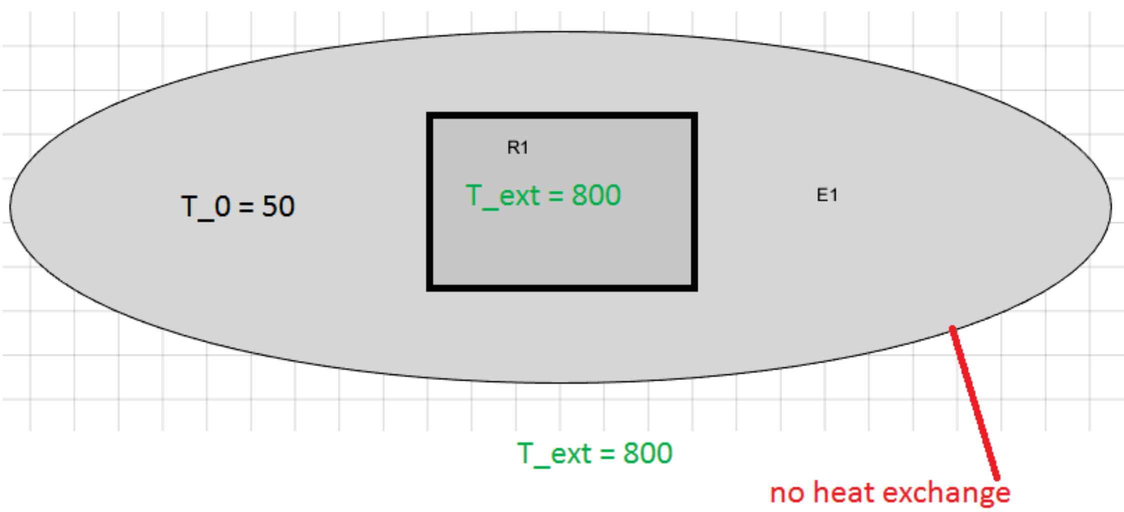


Figure 12: computational Domain.

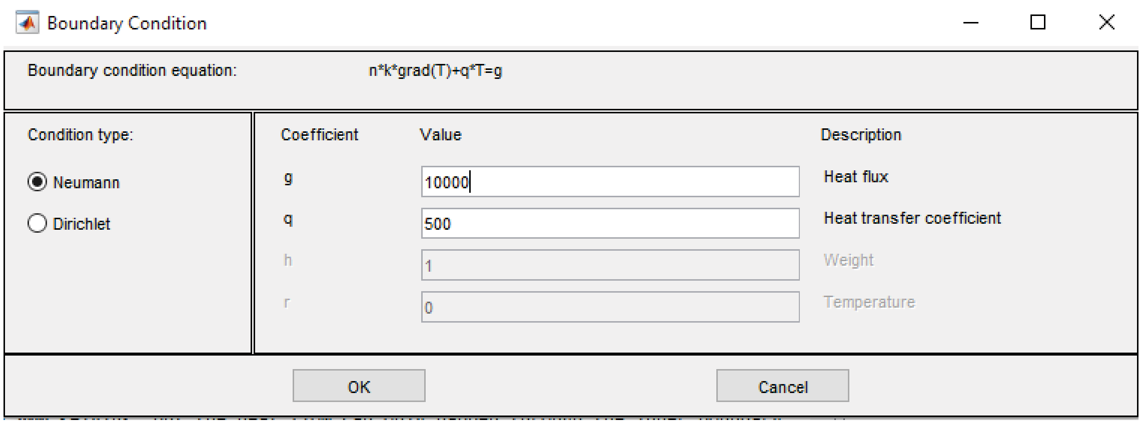


Figure 13: The Inner boundary temperature is kept the same as the initial condition for the plate, then $h \times T_0 = 10000$, since the heat transfer coefficient $h = 500$.

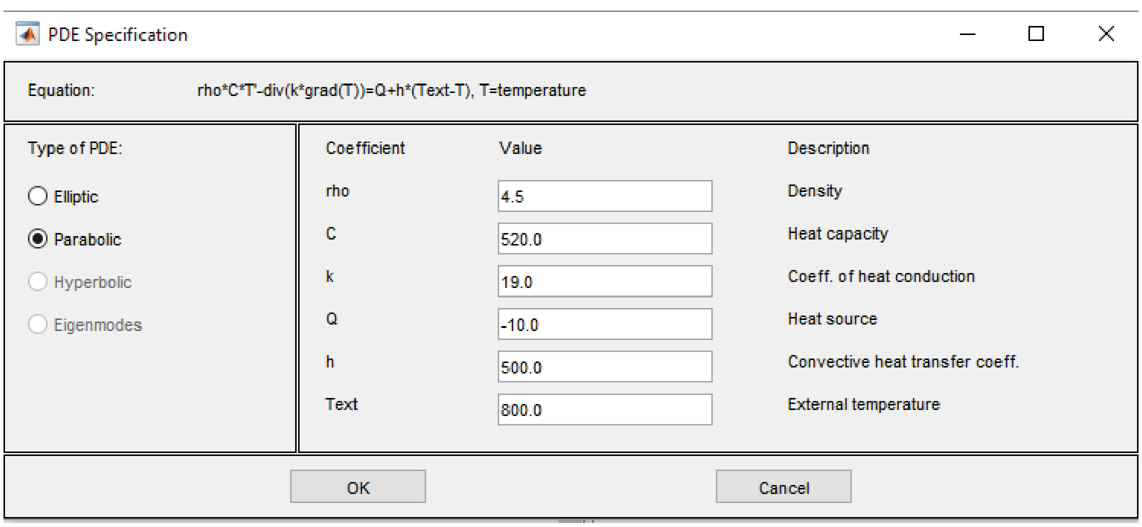


Figure 14: The Parabolic Heat Equation parameters for the PDEToolbox Solver.

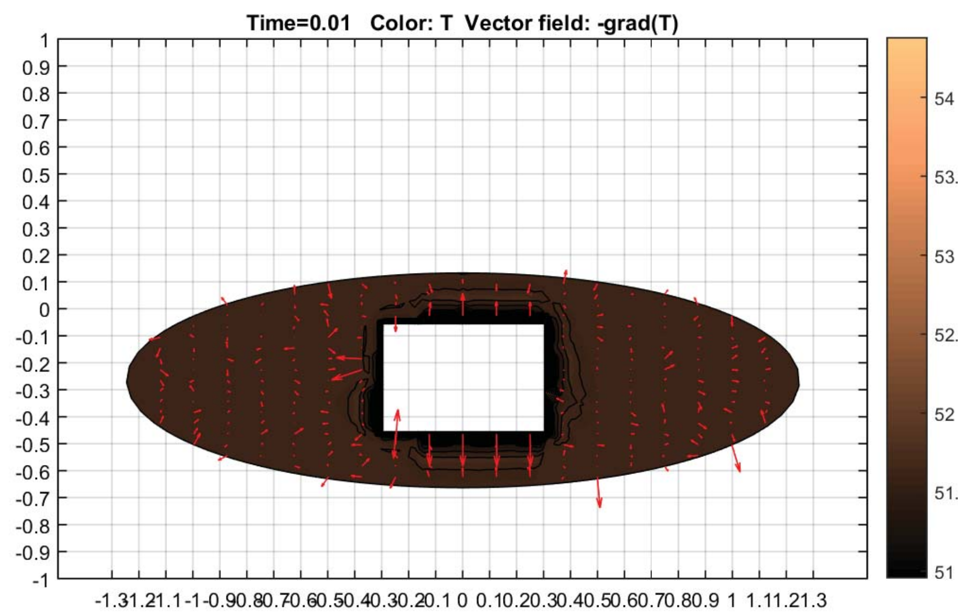


Figure 15: Solution at $t = 0.01$. We see that a lot of heat has already been conducted to the disk and the temperature decreases towards inner region only.

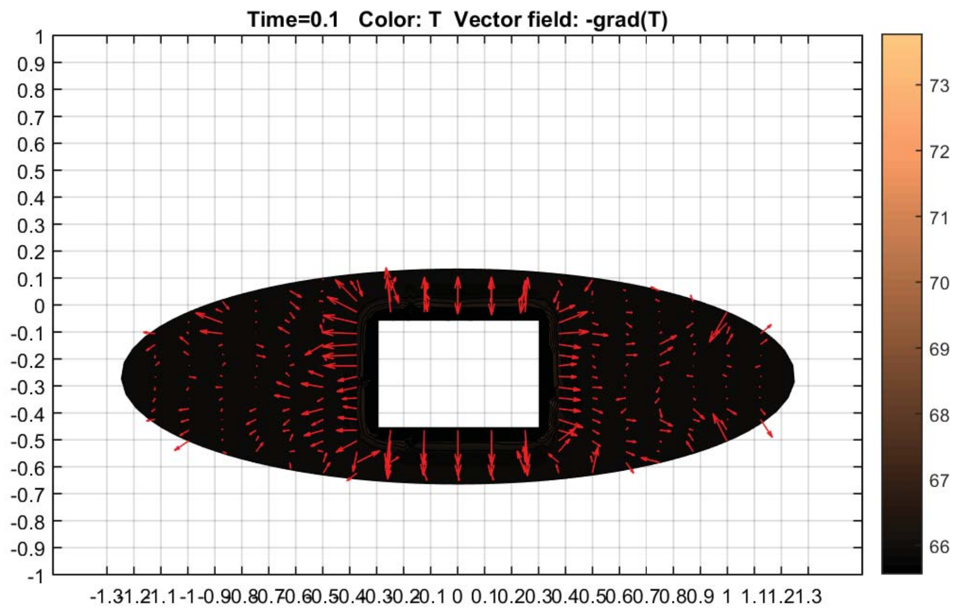


Figure 16: Solution at $t = 0.1$. More heat has entered our geometry. The gradient is still largest in the vertical direcetion since this end is closer to the external heat source environment. There is still a lot of chaotic heat conduction in the periphery.

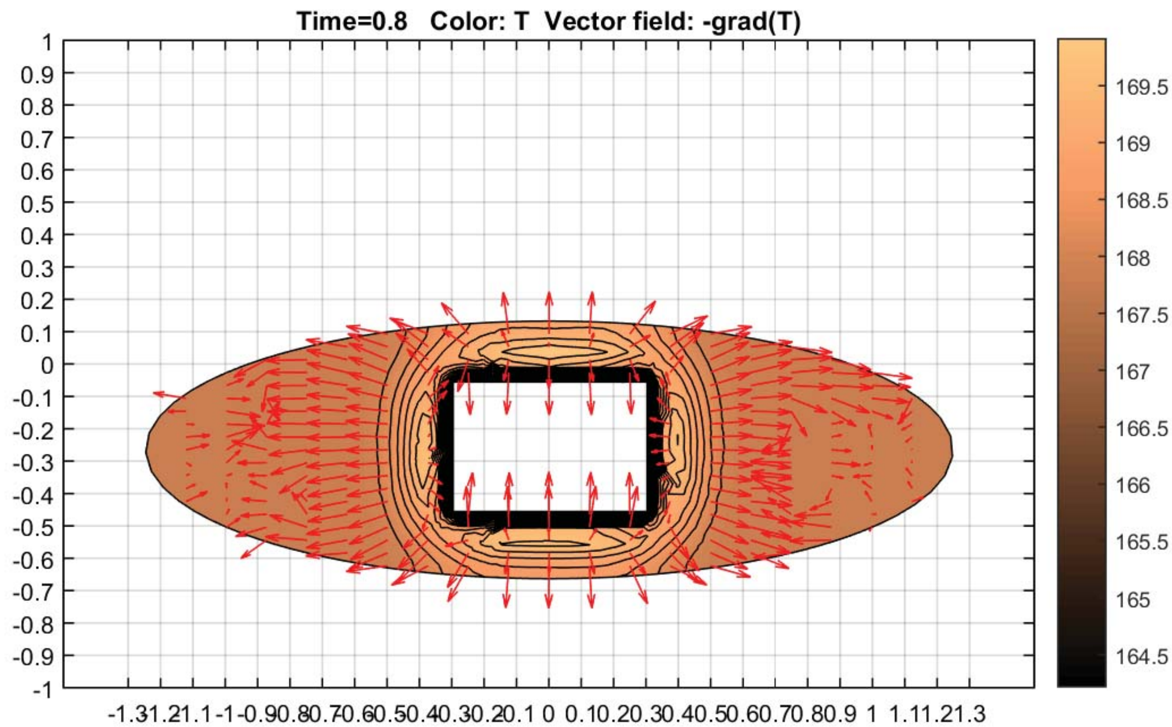


Figure 17: At $t = 0.8$ sec we see that a lot more heat has been transferred from the interior high temperature area to our region. The heat amount is so high in fact that the inner region temperature has dropped below 170 degrees from initial 800, and since the heat conduction is not very high as well, the high temperature front remains near the inner boundary. There are many inconsistencies in the peripheral regions, depicted by chaotic gradient directions.

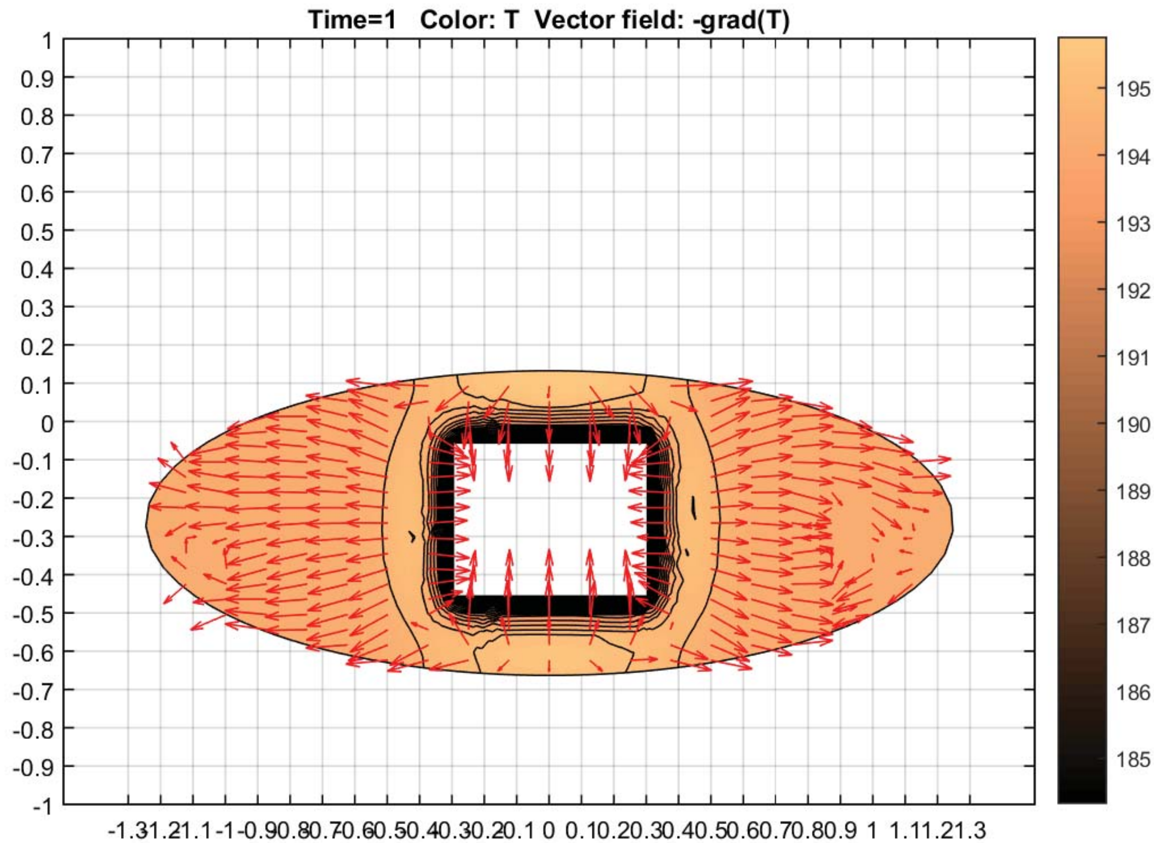


Figure 18: Just 0.2 seconds later, a lot has changed. The gradient vector field in the periphery is starting to get more organized. It is because the heat conduction takes time, and stable fronts have formed near the inner region that provide more structured support.

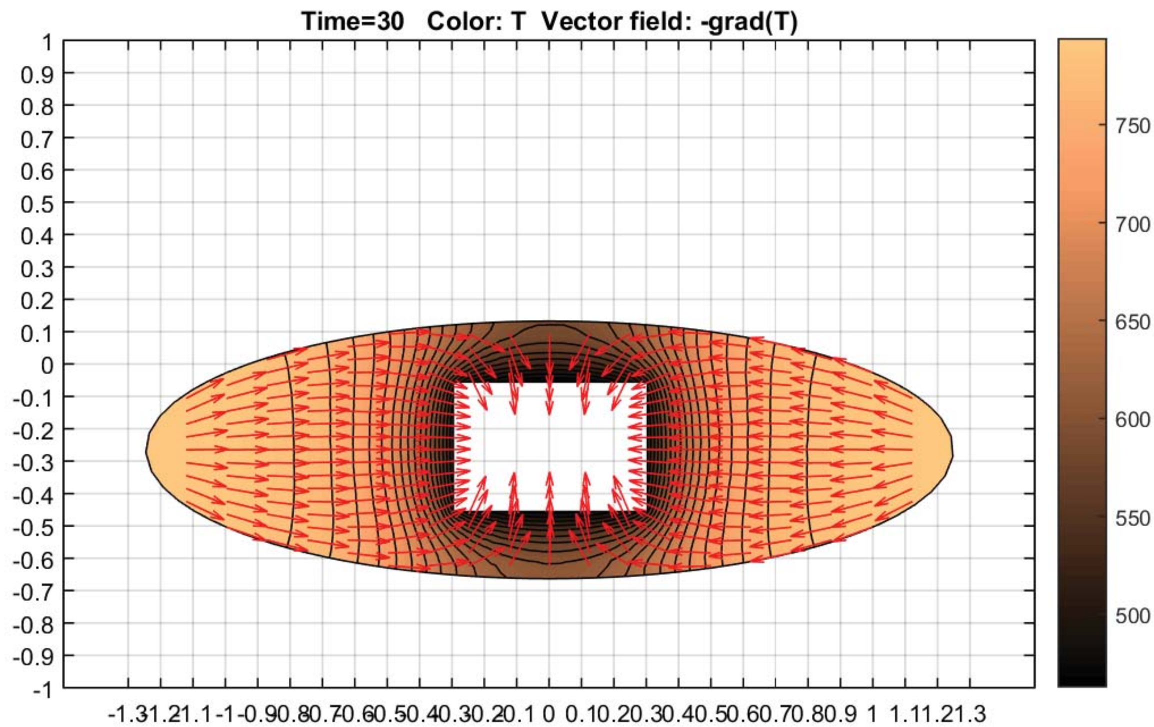


Figure 19: The stable time-independent state seems to be reached at around 25 seconds. The boundary region remains the coldest in our domain. The further we get from the cold interoior square, the higher the temperature – the heat flux (the gradient) is the highest just vertical from the cold interior square, since the stable state is reached one the environment is drained of all heat energy and the vertical distance in our domain is really close to the external environment .

Appendix

Task 1 Matlab Code

```
clf; close all; clear all;
syms  T(x)
DT = diff(T);
T(x) = dsolve(diff(T, x, 2) == T - 20, T(10) ==200 , T(0) == 40);
uvec = []; xmat = [];
nvec = 5:10; % n -how many elements
hvec = 10./nvec; % decreasing step size vector
errormat = zeros(nvec(end));
ncols=length(nvec)-2;
% first start with n = 5, so we have 4x4 matrix
colorMat = winter(6);
colorMat1 = copper(6);
for i = 1:length(hvec)-1
A = zeros(ncols,ncols); % build matrix
b = zeros(1,ncols);
v = 2/hvec(i) +  2*hvec(i)/(3);
vvec = ones(ncols,1).*v;
vdiag = diag(vvec);

r = -1/hvec(i) + hvec(i)/(6);
rvec = ones(ncols-1,1).*r;
rdiag =  diag(rvec,1) +  diag(rvec, -1);

A = rdiag + vdiag;

b(1) = 20*hvec(i)  - 40*( (hvec(i)/6) - 1/hvec(i));
b(end) = 20*hvec(i) - 200*(hvec(i)/6 - 1/hvec(i));
b(2:end-1) = 20*hvec(i);
b = b';

u = A\b; clear x; % one temperature solution for each step size

x = linspace(hvec(i),10-hvec(i),length(u)); % form x-space by taking as
    many equally spaced points on x axis as there are temperature points


error = eval(T) - u'; % vector size grows grows as N grows
subplot(length(hvec), 2, i);
 plot(  x, error); title('Error rate as a function of interior points')
  xlabel([num2str(nvec(i)-1') ' interior points']);
 ylabel('error rate');
 subplot(length(hvec), 2, length(hvec)+i);
 x = [0, x, 10];
 u = [40; u; 200];
 plot(x,u, 'color', colorMat(i,:));
 % legend('finite element method', 'analytical')
 hold on;
 plot(x,eval(T), 'color',colorMat1(i,:));
   xlabel([num2str(nvec(i)-1') ' ipoints']);
    ylabel('Rod Temperature');
```

```
 legend('Galerkin method Temperature distribution', 'Analytical Temperature
     distribution', 'Location', 'northwest')
title('Rod Temperature Analytically and by Galerkin method in the Rod');
ncols = ncols + 1;  % grow dimension
end

% hold on;
%
```