UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

Electric motor heat dissipation during high-speed drone flight

Heat Transfer A414

Numerical Methods Project

Bianca Kendall
21682410

9 June 2021

saam vorentoe · masiye phambili · forward together

Department of Mechanical and Mechatronic Engineering
Departement Meganiese en Megatroniese Ingenieurswese
Privaat Sak X1, Private Bag X1, Matieland, 7602
Tel: +27 21 808 4204 | www.eng.sun.ac.za

Engineering
Ezobunjineli
Ingenieurswese

Plagiarism declaration

I have read and understand the Stellenbosch University Policy on Plagiarism and the definitions of plagiarism and self-plagiarism contained in the Policy [Plagiarism: The use of the ideas or material of others without acknowledgement, or the re-use of one's own previously evaluated or published material without acknowledgement or indication thereof (self-plagiarism or text-recycling)].

I also understand that direct translations are plagiarism, unless accompanied by an appropriate acknowledgement of the source. I also know that verbatim copy that has not been explicitly indicated as such, is plagiarism.

I know that plagiarism is a punishable offence and may be referred to the University's Central Disciplinary Committee (CDC) who has the authority to expel me for such an offence.

I know that plagiarism is harmful for the academic environment and that it has a negative impact on any profession.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully (acknowledged); further, all verbatim copies have been expressly indicated as such (e.g. through quotation marks) and the sources are cited fully.

I declare that, except where a source has been cited, the work contained in this assignment is my own work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

I declare that have not allowed, and will not allow, anyone to use my work (in paper, graphics, electronic, verbal or any other format) with the intention of passing it off as his/her own work.

I know that a mark of zero may be awarded to assignments with plagiarism and also that no opportunity be given to submit an improved assignment.

Signature: ....

Name: Bianca Kendall .... Student no: 21682410 ....

Date: 9 June 2021 ....

Table of contents

Page

List of figures

List of tables

List of symbols

n_x	Total number of nodes in axial direction
n_r	Total number of nodes in radial direction
j	Radial direction node count
m	Axial direction node count
i	Time step count

1 Problem Definition and Analysis

1.1 Assumptions and Justifications

I. The motor may be treated as one solid body of a single material.
II. The thermophysical properties of the motor may be assumed to remain constant.
III. The exposed surfaces of the motor may be assumed to be smooth.
IV. Just before operation, the motor temperature is uniform at the prevailing ambient temperature (14.973°C).
V. The effects of mountings or obstructions on air flow over the motor are negligible.
VI. Radiation heat transfer may be neglected.
VII. Compressibility effects of air for flight at high speeds may be neglected.
VIII. The motor operates continuously at its rated power output (1000W) for the complete flight duration and heat generation is uniform throughout the motor (for a conservative solution).
IX. The motor is thermally symmetrical. The motor can be split along its centreline and the resulting symmetry plane of the motor can be approximated as an insulated boundary.
X. The maximum temperature of the motor will occur along its centreline, at the midpoint.
XI. All power losses are converted to heat. There are no other losses besides the specified power losses from the motor.
XII. All heat loss is due to convection.
XIII. The motor is symmetrical and thus conduction occurs in the axial (x) and radial (r) directions only (two-dimensional).
XIV. The motor is a solid incompressible body and thus heat transfer through the motor is pure conduction.
XV. Heat transfer at the cylindrical surface of the motor is by forced convection and the Nusselt number used is for crossflow over a cylinder.
XVI. Heat transfer at the side surfaces of the motor is by forced convection and the Nusselt number used is for laminar or turbulent flow over a flat plate.
XVII. The air is assumed to be dry so that the relevant fluid constants can be determined.
XVIII. The thermophysical properties of the air change with time over a 30-minute period.

1.2 Problem definition

- The temperature distribution through a motor is to be determined during a 30-minute flight, so that the maximum temperature during operation can be found. The maximum temperature is to be compared with a critical temperature that the motor may not exceed (150°C).

- The motor is approximated as a thermally symmetric cylinder, and thus cylindrical coordinates are used (axial and radial directions). Furthermore, the problem is analysed as a transient two-dimensional heat conduction problem for a short cylinder.

- The problem is a transient conduction one, where the symmetry plane is treated as an insulated boundary, and the outer radial and axial boundary has a transient forced convection boundary condition.

1.3 Relevant Variables and Parameters

Table 1: Electric motor properties and dimensions

Parameter	Value	Unit	Parameter	Value	Unit	Parameter	Value
L	100	mm	ρ	8800	kg/m^3	η	0.9
r_m	25	mm	c	420	J/kgK		
k	0.5	W/mK	P	1000	W		

Ambient air parameters (temperature, pressure, and velocity) as a function of time (from 0 to 1800 s):

$$T_a(t) = -7.0263E^{-17}t^6 + 3.7942E^{-13}t^5 - 6.5001E^{-1}\ t^4 + 2.9117E^{-7}t^3 + 2.4549E^{-4}t^2 - 2.4975E^{-1}t + 1.4973E^1\ [°C]$$

$$p_a(t) = 7.6189E^{-14}t^6 - 4.1143E^{-1}\ t^5 + 9.9884E^{-7}t^4 - 1.3741E^{-3}t^3 + 1.078t^2 - 4.3423E^2t + 1.0065E^5\ [Pa]$$

$$v_a(t) = 3.5132E^{16}t^6 - 1.8971E^{-12}t^5 + 3.2501E^{-9}t^4 - 1.4559E^{-6}t^3 - 1.2275E^{-3}t^2 + 1.2487t + 1.3409E^{-1}\ \left[\frac{m}{s}\right]$$

Fluid properties for determination of the convection heat transfer coefficient:

Film temperature: $T_{film} = \frac{T_a+T_s}{2} + 273.15\ [K]$

Air density (Kroger, 2004): $\rho_a(T_{film}) = \frac{p_a}{(287.08)T_{film}}\ [\frac{kg}{m^3}]$

Thermal conductivity, specific heat, dynamic viscosity (Kroger, 2004):

$$k_a(T_{film}) = -4.937787E^{-4} + 1.018087E^{-4}T_{film} - 4.627937E^{-8}{T_{film}}^2 + 1.250603E^{-11}{T_{film}}^3\ [W/mK]$$

$$C_{pa}(T_{film}) = 1.045356E^3 - 3.161783E^{-1}T_{film} + 7.083814E^{-4}{T_{film}}^2 - 2.705209E^{-7}{T_{film}}^3\ [J/kgK]$$

$$\mu_a(T_{film}) = 2.287973E^{-6} + 6.259793E^{-8}T_{film} - 3.131956E^{-11}{T_{film}}^2 + 8.15038E^{-15}{T_{film}}^3\ [kg/sm]$$

Reynold's number and Prandtl number (Cengel and Ghajar, 2015):

$$Re = \frac{p_a v_a D}{\mu_a} \qquad Pr = \frac{\mu_a C_{pa}}{k_a}$$

Average Nusselt number for crossflow over a cylinder, laminar and turbulent flow over a flat plat (Cengel and Ghajar, 2015):

$$Nu_{cyl} = \frac{hD}{k} = 0.3 + \frac{0.62Re^{\frac{1}{2}}Pr^{\frac{1}{3}}}{[1 + (0.4/Pr)^{\frac{2}{3}}]^{\frac{1}{4}}}[1 + (\frac{Re}{282{,}000})^{\frac{5}{8}}]^{\frac{4}{5}}$$

$$Nu_{lam,plate} = \frac{hD}{k} = 0.664(Re^{0.5})(Pr^{\frac{1}{3}}) \qquad Nu_{turb,plate} = \frac{hD}{k} = 0.037(Re^{0.8})(Pr^{\frac{1}{3}})$$

2 Solution Formulation

2.1 Finite Difference Equations

The 2-D explicit finite difference method is used to analyse the motor's temperature during the drone's 30-minute flight. Cylindrical volume elements are used for the heat transfer analysis of each element.

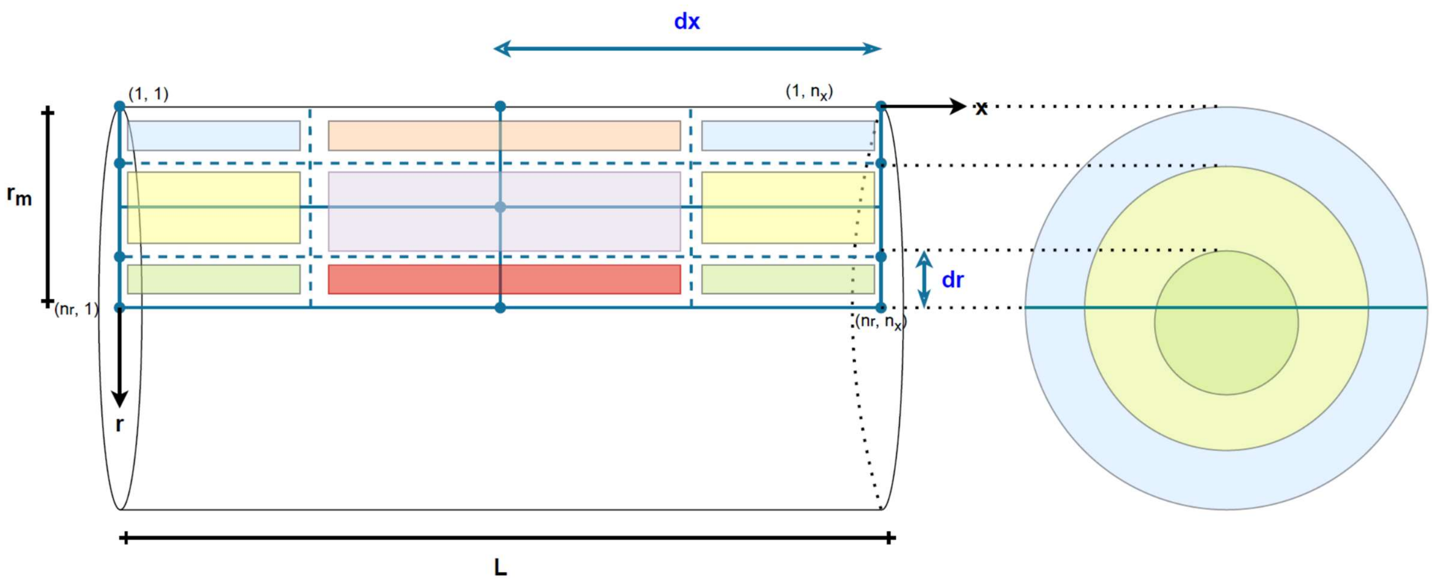


Figure 1: Motor discretized into different types of elements for numerical analysis

The electric motor is discretised as demonstrated in Figure 1. The cylinder is split along its symmetrical axis and the boundary is replaced with an insulated boundary, after isolation. The control volumes of elements on the axial and radial boundaries have their outer surfaces experiencing forced convection heat transfer due to the moving ambient air, while the interior surfaces experience transient conduction. The number of nodes is specified to create a nodal mesh which represents half of the cylinder. The finite difference equation for each type of element of the system is determined. There is convection over the cylinder (with convection heat transfer coefficient determined using the Nusselt number for crossflow over a cylinder) and convection at the sides (with convection heat transfer coefficient determined using the Nusselt number for flow over a flat plate). The temperature at the current time step is used to calculate the temperature at the next time step. Stability criteria are used to determine the time step used in the nodal analysis. The corner boundary nodes which are exposed to convection on two sides, are found to be the restrictive nodes in terms of the stability requirement. Therefore, the time step is determined using:

$\tau = \frac{\alpha \Delta t}{\Delta l^2} \leq \frac{1}{4(1+Bi)}$ where, $Bi = \frac{h\Delta l}{k}$ and, $\Delta l = \min(\Delta x, \Delta r)$

The volumes and areas for each element are determined according to whether they are along the centreline or not. Those on the centreline have short cylindrical volume elements (yellow in Figure 1) and those not on the centreline have hoop volume elements (blue in Figure 1).

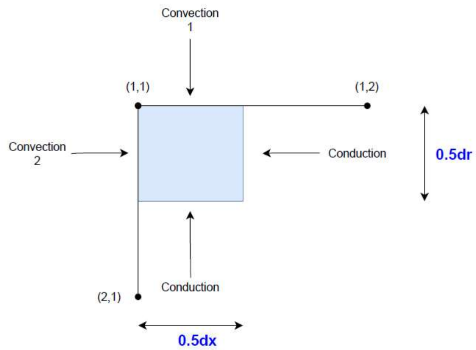
The generalised energy equation used to derive the finite difference equations for each type of element is:

$$\sum_{sides} \dot{Q}^i + \dot{e}_{gen} V_{element} = \rho V_{element} C_p \frac{(T_m^{i+1} - T_m^i)}{\Delta t}$$

$$where, \dot{Q}_{convection,i} = hA_s(T_\infty - T_i) \text{ } and \text{ } \dot{Q}_{conduction,i} = \frac{kA_{exposed}(T_\infty - T_i)}{length}$$

2.1.1 Blue element (corner)

This element will occur at coordinates $(1,1) and \ (1, n_x)$.

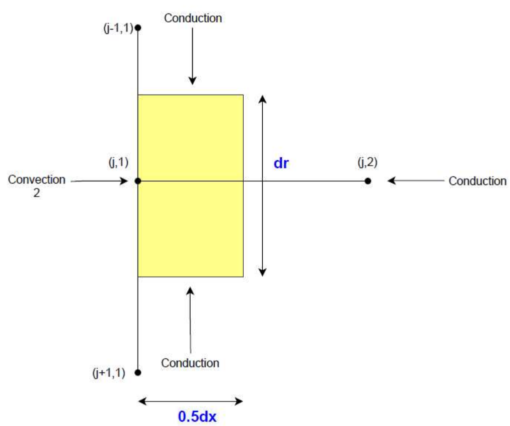


$A_{c,sides}$	$\pi({r_m}^2 - \left(r_m - \frac{dr}{2}\right)^2)$
$A_{c,top}$	$(2\pi r_m)\frac{dx}{2}$
$A_{c,bottom}$	$\left(2\pi(r_m - \frac{dr}{2})\right)\frac{dx}{2}$
V_c	$A_{c,sides}\frac{dx}{2}$

$$T_{1,1}^{i+1} = T_{1,1}^i + \left(\dot{e}_{gen}V_c + h_1(A_{c,top})(T_\infty^i - T_{1,1}^i) + h_2(A_{c,sides})(T_\infty^i - T_{1,1}^i) + k(A_{c,bottom})\frac{(T_{2,1}^i - T_{1,1}^i)}{\frac{1}{2}dr} + k(A_{c,sides})\frac{(T_{1,2}^i - T_{1,1}^i)}{\frac{1}{2}dx}\right)\left(\frac{dt}{\rho V_c C_p}\right)$$

2.1.2 Yellow element (side, non-centreline)

This element will occur over the range of $j \in (2, n_r - 1)$ and for $m = 1$ and $m = n_x$.

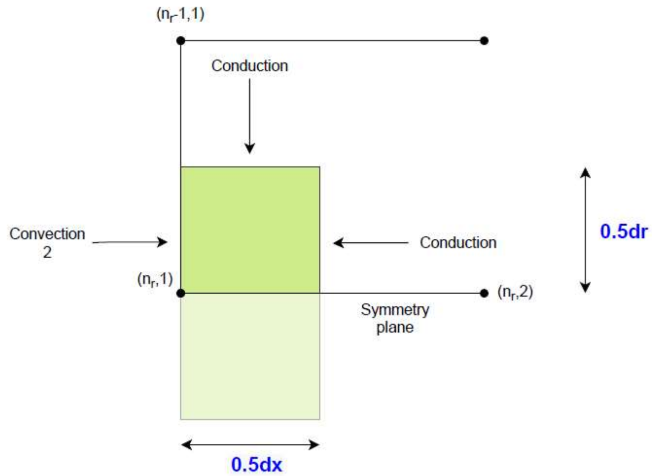


$A_{s,sides}$	$\pi[\left((n_r - j)dr + \frac{dr}{2}\right)^2 - \left((n_r - j)dr - \frac{dr}{2}\right)^2]$
$A_{s,top}$	$\pi(2(n_r - j)dr + \frac{dr}{2})\frac{dx}{2}$
$A_{s,bottom}$	$\pi(2(n_r - j)dr - \frac{dr}{2})\frac{dx}{2}$
V_s	$A_{s,sides}(\frac{dx}{2})$

$$T_{j,1}^{i+1} = T_{j,1}^i + \left(\dot{e}_{gen}V_s + h_2(A_{s,sides})(T_\infty^i - T_{j,1}^i) + k(A_{s,top})\frac{(T_{j+1,1}^i - T_{j,1}^i)}{dr} + k(A_{s,bottom})\frac{(T_{j-1,1}^i - T_{j,1}^i)}{dr} + k(A_{s,sides})\frac{(T_{j,2}^i - T_{j,1}^i)}{\frac{1}{2}dx}\right)\left(\frac{\Delta t}{\rho V_s C_p}\right)$$

2.1.3 Green element (side, centreline)

This element will occur at coordinates $(n_r, 1)$ and (n_r, n_x).

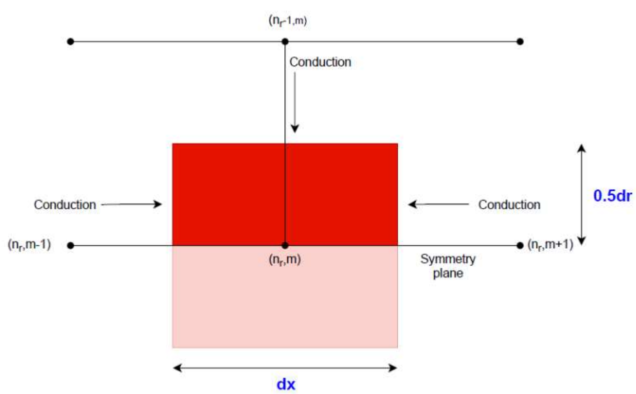


$A_{sc,sides}$	$\pi\left(\frac{dr}{2}\right)^2$
$A_{sc,top}$	$2\pi(\frac{dr}{2})\frac{dx}{2}$
V_{sc}	$A_{sc,sides}(\frac{dx}{2})$

$$T_{n_r,1}^{i+1} = T_{n_r,1}^{i} + \left(\dot{e}_{gen}V_{sc} + h_2\left(A_{cs,sides}\right)\left(T_\infty^i - T_{n_r,1}^i\right) + k\left(A_{sc,top}\right)\frac{\left(T_{n_r-1,1}^i - T_{n_r,1}^i\right)}{\frac{1}{2}dx} + k\left(A_{sc,sides}\right)\frac{\left(T_{n_r,2}^i - T_{n_r,1}^i\right)}{dr} \right)\left(\frac{\Delta t}{\rho V_{sc} C_p}\right)$$

2.1.4 Red element (interior, centreline)

This element will occur over the range of $m \in (2, n_r - 1)$ and for $j = n_r$.

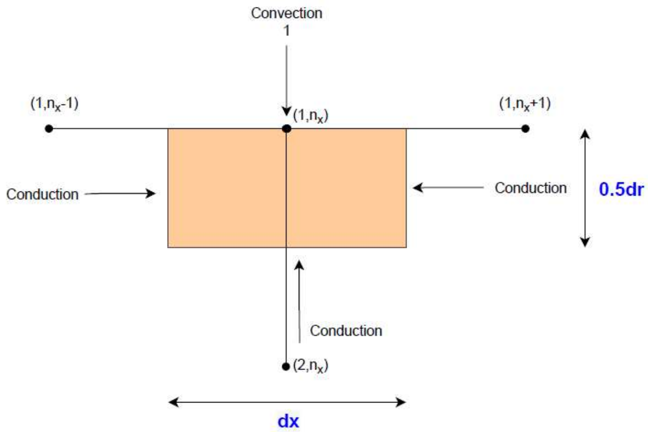


$A_{ic,sides}$	$\pi\left(\frac{dr}{2}\right)^2$
$A_{ic,top}$	$2\pi(\frac{dr}{2})dx$
V_{ic}	$A_{ic,sides}(dx)$

$$T_{n_r,m}^{i+1} = T_{n_r,m}^{i} + \left(\dot{e}_{gen}V_{ic} + k\left(A_{ic,sides}\right)\frac{\left(T_{n_r,m-1}^i - T_{n_r,m}^i\right)}{dx} + k\left(A_{ic,top}\right)\frac{\left(T_{n_r-1,m}^i - T_{n_r,m}^i\right)}{dr} + k\left(A_{ic,sides}\right)\frac{\left(T_{n_r+1,m}^i - T_{n_r,m}^i\right)}{dr} \right)\left(\frac{\Delta t}{\rho V_{ic} C_p}\right)$$

2.1.5 Orange element (interior, top)

This element will occur over the range of $m \in (2, n_r - 1)$ and for $j = 1$.

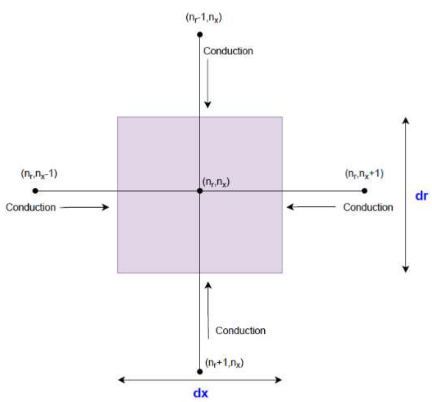


$A_{t,sides}$	$\pi[(r_m{}^2 - \left(r_m - \frac{dr}{2}\right)^2]$
$A_{t,top}$	$(2\pi r_m)dx$
$A_{t,bootom}$	$(2\pi(r_m - \frac{dr}{2}))dx$
V_t	$A_{t,sides}(dx)$

$$T_{1,m}^{i+1} = T_{1,m}^{i} + \left(\dot{e}_{gen} V_t + h_1(A_{t,top})(T_\infty^i - T_{1,m}^i) + k(A_{t,sides})\,\frac{(T_{1,m-1}^i - T_{1,m}^i)}{dx} + k(A_{t,bottom})\frac{(T_{2,m}^i - T_{1,m}^i)}{\frac{dr}{2}} + k(A_{t,sides})\,\frac{(T_{1,m+1}^i - T_{1,m}^i)}{dx} \right)\left(\frac{\Delta t}{\rho V_5 C_p}\right)$$

2.1.6 Purple element (interior)

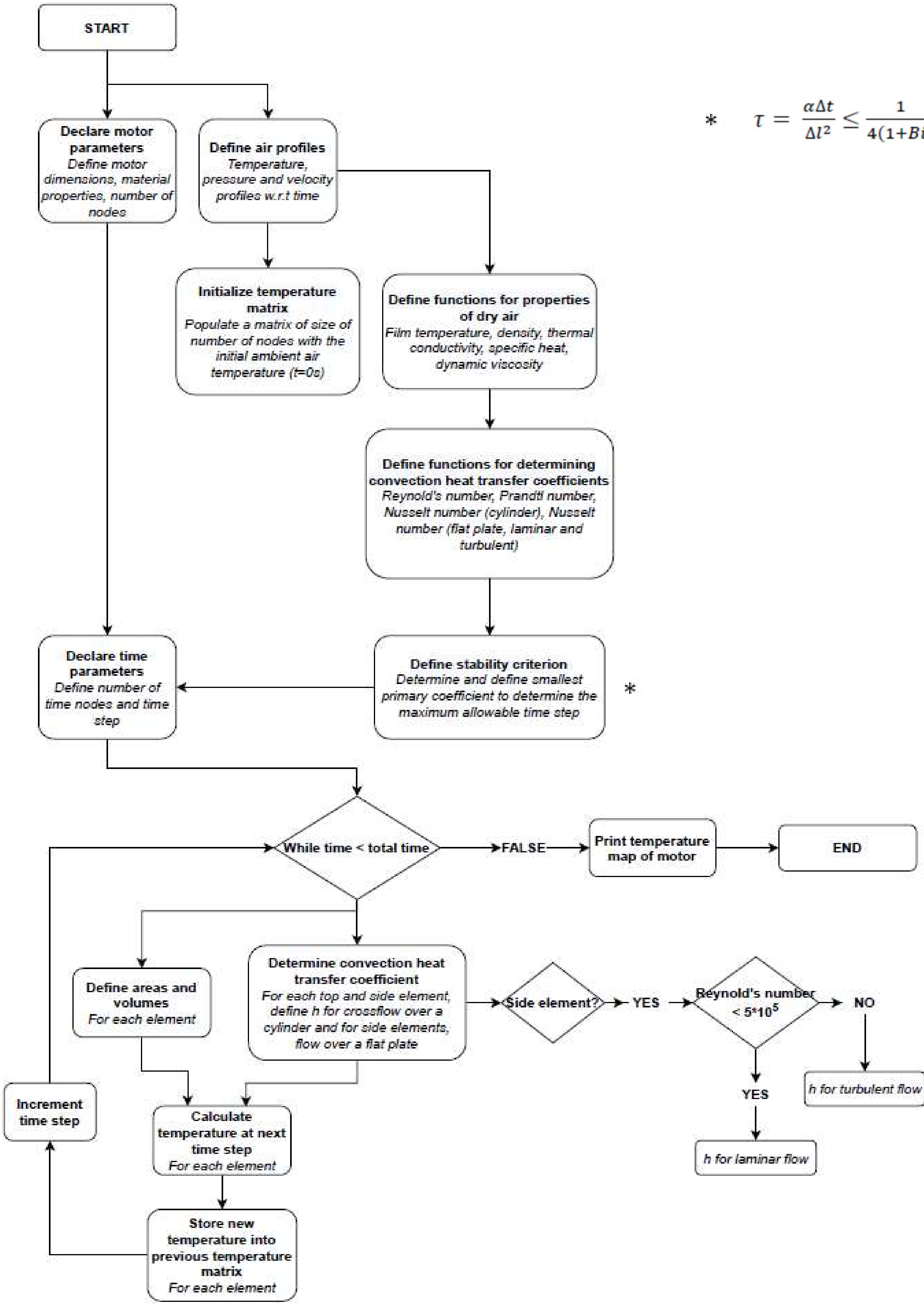
This element will occur over the range of $m \in (2, n_r - 1)$ and for $j \in (2, n_x - 1)$.



$A_{i,sides}$	$\pi[\left((n_r - j)dr + \frac{dr}{2}\right)^2 - \left((n_r - j)dr - \frac{dr}{2}\right)^2]$
$A_{i,top}$	$\pi(2(n_r - j)dr + \frac{dr}{2})dx$
$A_{i,bottom}$	$\pi(2(n_r - j)dr - \frac{dr}{2})dx$
V_i	$A_{i,sides}(dx)$

$$T_{j,m}^{i+1} = T_{j,m}^{i} + \left(\dot{e}_{gen} V_i + k(A_{i,top})\,\frac{(T_{j-1,m}^i - T_{j,x}^i)}{dr} + k(A_{i,sides})\,\frac{(T_{j,m-1}^i - T_{j,x}^i)}{dx} + k(A_{i,bottom})\frac{(T_{j+1,m}^i - T_{j,m}^i)}{dr} + k(A_{i,sides})\frac{(T_{j,m+1}^i - T_{j,m}^i)}{dr} \right)\left(\frac{\Delta t}{\rho V_i C_p}\right)$$

2.2 Solution Algorithm



$$* \quad \tau = \frac{\alpha \Delta t}{\Delta l^2} \leq \frac{1}{4(1+Bi)}.$$

2.3 Source Code

Refer to Appendix A for the full source code that is implemented in MATLAB, which follows the procedure outlined in Section 2.2, using the finite difference equations derived in Section 2.1, and with the assumptions, variables, and parameters defined in Section 1.

3 Numerical Model Validation

3.1 Analytical Model and Simplified Validation Case

To validate the model, an analytical solution is found with which to compare the numerical solution. The motor under investigation is approximated as a short cylinder, which can be physically formed by the intersection of a long cylinder and a plane wall, as shown in Figure 2, thus a superposition method of one-term approximate solutions for a long cylinder and a plane wall is to be used. To facilitate a comparison to the numerical model, the following assumptions are made: Heat conduction is transient and two-dimensional (varying in the x- and r-directions). The thermal properties (including convection heat transfer coefficient) are constant. The Fourier number is greater than 0.2, so that the one-term approximate solutions can be used. Heat generation is equal to zero. Furthermore, the following conditions are set: The motor temperature at the cylinder core at a time of 900s will be found. The convection heat transfer coefficient is set at $250\ W/m^2K$ and all other properties are kept as those stated in Table 1. The initial motor temperature is approximated as the ambient air at a time of 0 s, and is found to be 14.973°C. The ambient air temperature at 900 s is -38.4607°C.

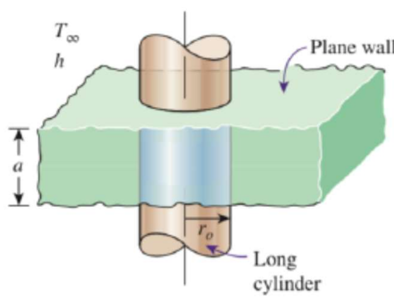


Figure 2: Analytical model physical formation

The solution is then found as follows:

Centre of plane wall:

$$\tau = \frac{\alpha \Delta t}{L^2} = \frac{(1.3528 \times 10^{-7})(900)}{(0.05)^2} = 0.0487$$

$$Bi = \frac{hL}{k} = \frac{(250)(0.05)}{(0.5)} = 25$$

At this Fourier and Biot number for a plane wall, from Table 4-2 (Cengel & Ghajar, 2015),

$$\theta_{wall}(0,t) = \frac{T(0,t)-T_\infty}{T_i-T_\infty} = A_1 e^{-\lambda_1{}^2\tau} = (1.2708)e^{-(1.50815)^2(0.0487)} = 1.13755$$

Centre of cylinder:

$$\tau = \frac{\alpha \Delta t}{{r_o}^2} = \frac{(1.3528 \times 10^{-7})(900)}{(0.025)^2} = 0.1948$$

$$Bi = \frac{hr_o}{k} = \frac{(250)(0.025)}{(0.5)} = 12.5$$

Similarly for a cylinder, from Table 4-2 (Cengel & Ghajar, 2015),

$$\theta_{cyl}(0,t) = \frac{T(0,t) - T_\infty}{T_i - T_\infty} = A_1 e^{-\lambda_1{}^2\tau} = (1.574476)e^{-(2.20988)^2(0.1948)} = 0.60841$$

Then, the two-dimensional solution for the short cylinder can be expressed as:

$$\left(\frac{T(0,0,t) - T_\infty}{T_i - T_\infty}\right)_{short\ cylinder} = \left(\frac{T(0,t) - T_\infty}{T_i - T_\infty}\right)_{plane\ wall} \times \left(\frac{T(0,t) - T_\infty}{T_i - T_\infty}\right)_{infinite\ cylinder}$$
$$= \theta_{wall}(0,t) \times \theta_{cyl}(0,t) = (1.13755)(0.60841) = 0.692097$$

$$T(0,0,900) = (0.692097)\big(14.973 - (-38.4607)\big) + (-38.4607) = -1.479℃$$

3.2 Numerical Model Evaluation

The numerical model shown in the source code (Appendix A) is run using the same parameters defined in Section 3.1 above. The parameters and properties are much the same, but for comparison to the analytical model, heat generation is set to zero, the convection heat transfer coefficient is set to $250\ W/m^2K$ and the total time is set to 900s. The number of nodes in the x and r directions is set to 10. Then, the heatmap shown in Figure 3 is produced:

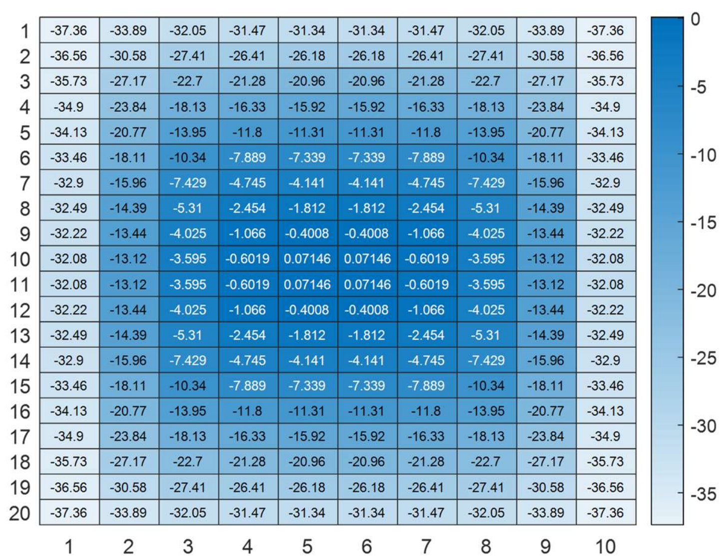


Figure 3: Heatmap of motor for numerical model validation

At the core nodes (nodes 5 and 6 in the x direction, and 10 and 11 in the r direction), the core temperature is $T(0,0,900) = 0.07146℃$. This is deemed sufficiently close to the core temperature determined using the analytical model in Section 3.1 of $-1.479℃$, and thus the numerical model is considered sufficiently accurate to be used in the analysis of the problem to produce reliable results. The slight discrepancy between the two values is expected since the one-term approximate method is less accurate when $\tau < 0.2$.

4 Solution Results, Analysis and Conclusions

4.1 Solution Results

The numerical model outlined in Appendix A was validated in Section 3, and the conclusion was that the numerical model is a reasonably accurate tool for the approximation of the temperature in the motor for this problem. The finite difference equations that were derived in Section 2, along with the determined initial and boundary conditions, is used to determine the temperature distribution of the electric motor during the drone's flight time of 30-minutes. The time step is determined using the stability criterion for the corner node (limiting element), and for a node size of 20 x 20 nodes, this is found to be 0.3 s. It is then seen that, as expected, the maximum temperature occurs at the central node of the motor, i.e., at the midpoint of the centreline (the core). A vector is populated with the temperatures at this location, at each time step for the 30-minute flight and plotted in Figure 4. From this plot, it is subsequently found that the maximum core temperature occurs at the end of the 30-minute flight, again as expected, since the ambient air temperature is at a maximum at t = 0 s and t = 1800 s, and the heat generation from the motor will be at a maximum the longer it is in operation. A contour plot depicting the heat generation in the motor at the end of the 30-minute flight (when the core temperature is at its maximum during operation) is shown in Figure 5.

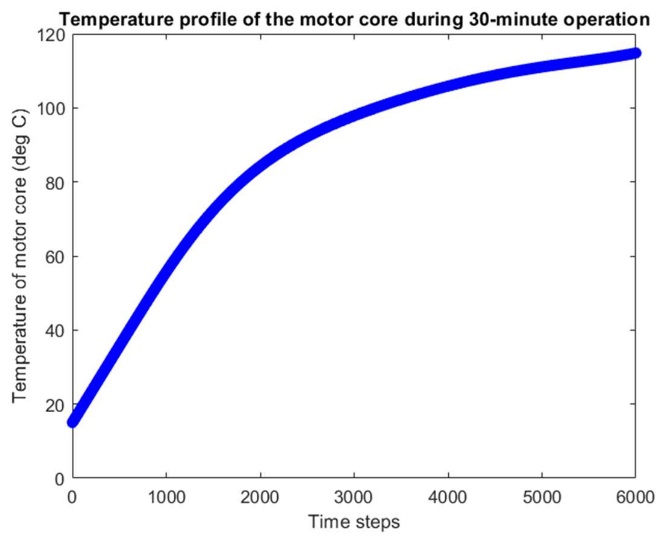


Figure 4: Temperature profile of motor core from 0 - 30 minutes (0 - 6000 time steps for a time step size of 0.3 s)

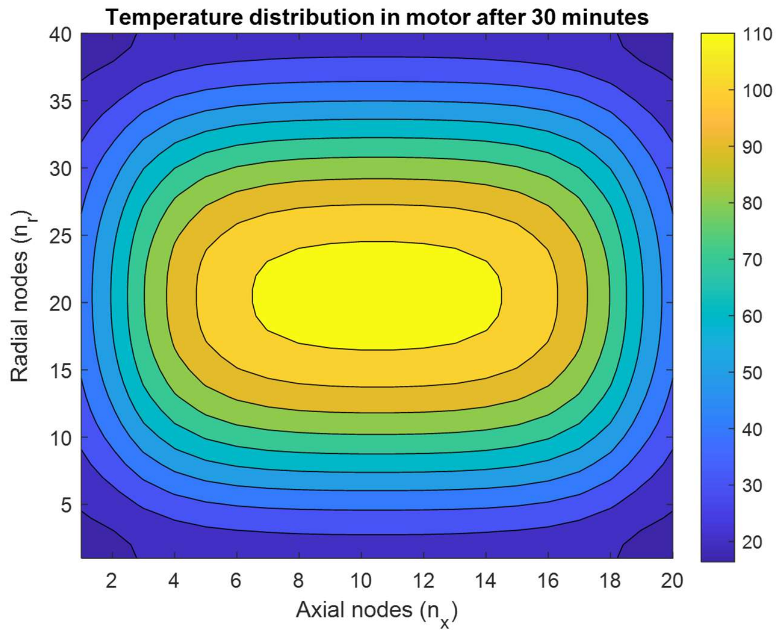


Figure 5: Temperature distribution in motor at the end of the flight (30 minutes)

4.2 Analysis

From the contour plot shown in Figure 5, it is clear to see that the temperature of the motor decreases outwardly from the centreline. The maximum temperature that the motor core experiences is at the end of the 30-minute flight and is approximately $114.8149°\mathrm{C}$. This is sufficiently less than the internal motor temperature limit of $150°\mathrm{C}$, which the motor may not exceed at any point during operation to avoid motor burn-out. It is therefore concluded that the current configuration of the motor on the engine is suitable without the need for a cooling jacket since the ambient air is successful in keeping the motor below its upper operational temperature limit.

The temperature distribution clearly shows that the internal motor temperature is dependent on the ambient air conditions. Thus, it is noted that should the ambient air conditions change, there is a risk that it may not be sufficient in cooling the motor. Furthermore, should the duration of the flight increase, the heat generation in the motor may increase too much for the ambient air alone to cool it.

4.3 Conclusions

In this report, a numerical model was derived to analyse the heat dissipation and subsequent internal motor temperature of the motor in its current configuration in a drone engine, for an operating time of 30 minutes. Finite difference equations for the discretised motor were determined using the explicit method, subsequent area and volume elements were defined and boundary conditions applied. The numerical model was first validated to determine whether it could be used to produce reliable results for the problem at hand. This was done by first simplifying it, performing the simplified analysis, and comparing the results to those of an analytical analysis performed using the same simplifications. The results of the validation indicated that the numerical model can accurately define the heat dissipation in the motor during the high-speed drone flight. Thereafter, the numerical model was used to determine the temperature profile of the motor at the location where it experiences the highest temperature (the core). It was found that the maximum temperature the motor experiences is at its core at the end of the 30-minute flight time. This temperature was below the operational temperature limit of 150°C, meaning the motor in its current configuration is safe to use without a cooling jacket, as the ambient air is sufficient in cooling the motor. There is a risk that the ambient air alone may not be sufficient to cool the motor should the ambient conditions change or should the flight time increase. It is thus recommended that the current configuration be re-evaluated to maximise the airflow over the motor during operation, increasing the amount of convection heat transfer occurring to cool the motor.

Appendix A Source code

A.1 Numerical model MATLAB code

```
%Motor properties and dimensions
L = 0.1; %m
r_m = 0.025; %m
k = 0.5; %W/mK
rho = 8800; %kg/m^3
c = 420; %J/kgK
Power = 1000; %W
efficiency = 0.9;


Volume = pi*(r_m^2)*L; %m^3
egen = (Power-efficiency*Power)/Volume; %W/m^3
alpha = k/(rho*c); %thermal diffusivity


%Air properties
%Ambient air temperature profile with respect to time in seconds from 0 to 1800
T_a = @(t) (-(7.0263*10^-17)*t.^6 + (3.7942*10^-13)*t.^5 - (6.5001*10^-10)*t.^4 + ...
    (2.9117*10^-7)*t.^3 + (2.4549*10^-4)*t.^2 - ...
    (2.4975*10^-1)*t + (1.4973*10)); %degrees Celsius


%Ambient air pressure profile with respect to time in seconds from 0 to 1800
P_a = @(t) ((7.6189*10^-14)*t.^6 - (4.1143*10^-10)*t.^5 + (9.9884*10^-7)*t.^4 - ...
    (1.3741*10^-3)*t.^3 + (1.078)*t.^2 - ...
    (4.3423*10^2)*t + (1.0065*10^5)); %Pa


%Ambient air velocity profile with respect to time in seconds from 0 to 1800
V_a = @(t) ((3.5132*10^-16)*t.^6 - (1.8971*10^-12)*t.^5 + (3.2501*10^-9)*t.^4 - ...
    (1.4559*10^-6)*t.^3 - (1.2275*10^-3)*t.^2 + ...
    (1.2487)*t + (1.3409*10^-1)); %m/s


%Nodes
n_x = 20; %number of nodes in x direction
n_r = 20; %number of nodes in y direction
x = linspace(0, L, n_x); %x direction nodes
r = linspace(0, r_m, n_r); %radial direction nodes
dx = L/(n_x-1); %grid size in x direction
dr = r_m/(n_r-1); %grid size in radial direction


%Initializing with initial conditions (at time = 0 s)
T = T_a(0)*ones(n_x,n_r); %initializing T matrix of size n_x x n_r
T_prev = T;
reflect_initial = flipud(T_prev); %reflecting over the symmetrical plane
T_i = [reflect_initial(2:r,:); T_prev]; %initial temperature of motor
%heatmap(T_i);
%colorbar;


%Temperature dependent fluid properties
R = 287.08;
%Film temperature
```

```
T_film = @(t,r,x) ((T_a(t) + T_prev(n_r,n_x))./2 + 273);
%Density
rho_air = @(t,r,x) (P_a(t)./(R.*T_film(t,r,x)));
%Thermal conductivity
k_air = @(t,r,x) (-4.937787e-4 + 1.018087e-4.*T_film(t,r,x)...
    - 4.627937e-8.*(T_film(t,r,x)).^2 + 1.250603e-11.*(T_film(t,r,x)).^3);
%Specific heat
c_p_air = @(t,r,x) (1.045356e3 - 3.161783e-1.*T_film(t,r,x)...
    + 7.083814e-4.*T_film(t,r,x).^2 - 2.705209e-7.*T_film(t,r,x).^3);
%Dynamic viscosity
mu = @(t,r,x) (2.287973e-6 + 6.259793e-8.*T_film(t,r,x)...
    - 3.131956e-11.*(T_film(t,r,x)).^2 + 8.15038e-15.*(T_film(t,r,x)).^3);


%Transient convection heat transfer coefficient
%Reynolds number (for a cylinder)
Re = @(t,r,x) (rho_air(t,r,x).*V_a(t)*(2*r_m)./mu(t,r,x));
%Prandtl number
Pr = @(t,r,x) (mu(t,r,x).*c_p_air(t,r,x)./k_air(t,r,x));
%Nusselt number (for cross-flow over a cylinder)
Nu_cyl = @(t,r,x) (0.3 +
((0.62.*(Re(t,r,x).^0.5).*Pr(t,r,x).^(1/3))./(1+(0.4./Pr(t,r,x)).^(2/3)^0.25)).*(1+(Re(t,r
,x)./282000).^(5/8)).^(4.5));
%Nusselt number (for laminar flow over a flat plate)
Nu_plate_lam = @(t,r,x) (0.664*(Re(t,r,x).^0.5).*Pr(t,r,x)^(1/3));
%Nusselt number (for turbulent flow over a flat plate)
Nu_plate_turb = @(t,r,x) (0.037*(Re(t,r,x).^0.8).*Pr(t,r,x)^(1/3));
%Heat transfer coefficients
h_cyl = @(t,r,x) (Nu_cyl(t,r,x).*k_air(t,r,x))./(2*r_m);
h_lam = @(t,r,x) (Nu_plate_lam(t,r,x).*k_air(t,r,x))./(2*r_m);
h_turb = @(t,r,x) (Nu_plate_turb(t,r,x).*k_air(t,r,x))./(2*r_m);


%Time
time_total = 1800; %s
Re(900,ceil(n_r/2),1);
h_cyl_max = h_cyl(900,1,ceil(n_x/2));
h_side_max_ = h_turb(900,ceil(n_r/2),1);
    Ac_sides = 0.5*((pi*r_m^2)-(pi*(r_m-0.5*dr)^2));
    Ac_top = (2*pi*r_m*0.5*dx);
    Ac_bottom = (2*pi*(r_m-0.5*dr)*0.5*dx);
    Vc = Ac_sides*0.5*dx;
primary_coefficient = (-(h_cyl_max*Ac_top)-(h_side_max_*Ac_sides)...
    -(k*Ac_bottom)/(0.5*dr)-(k*Ac_sides)/(0.5*dx))*(1/(rho*Vc*c))
```

```
primary_coefficient = -2.9391
```

```
dt_max = -1/primary_coefficient %s
```

```
dt_max = 0.3402
```

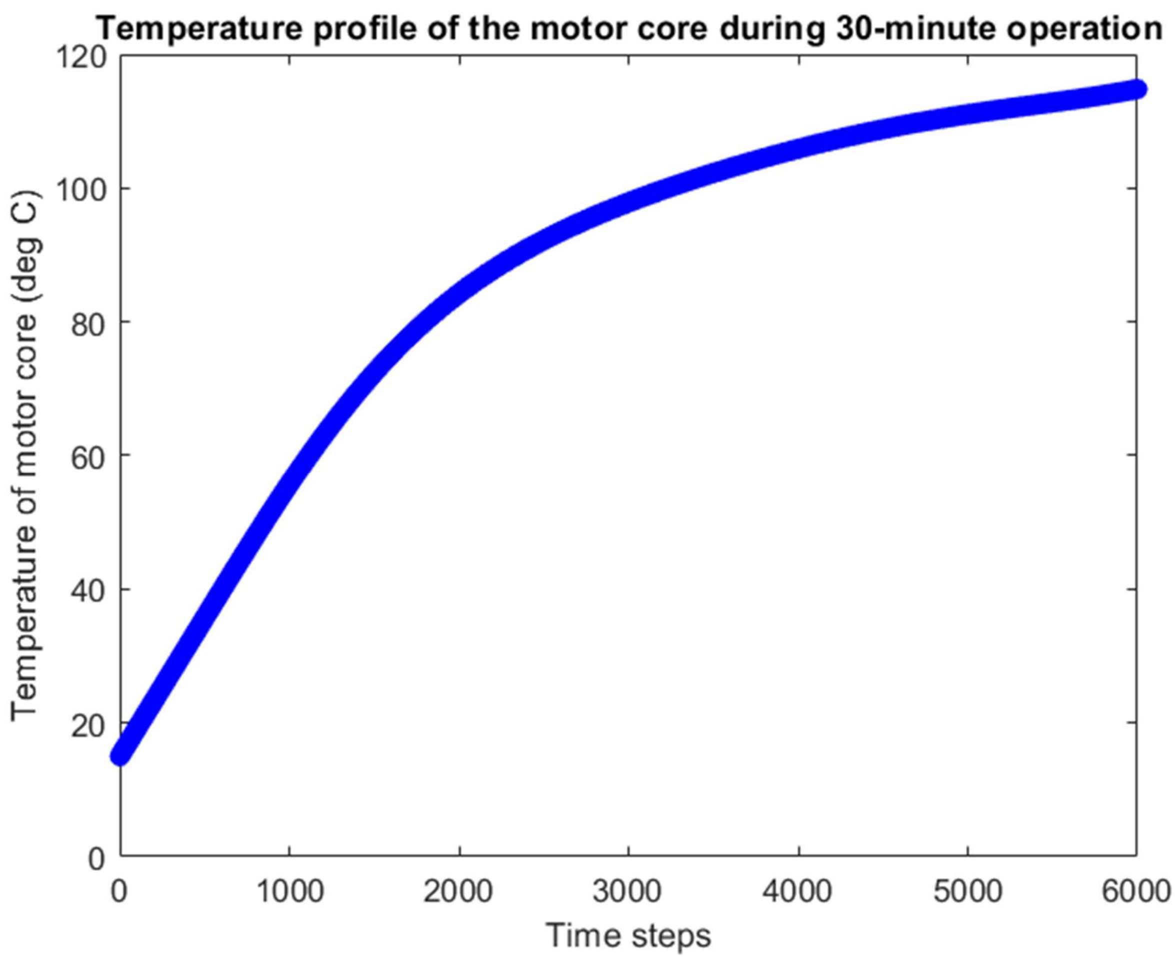
```
dt = 0.3; %s
n_time = time_total/dt; %number of time steps
max_temp_vector = zeros(n_time,1);
```

```
%Explicit method for transient 2D heat transfer
%Time loop
for i = 1:n_time
    T = zeros(n_r, n_x);
    t = i*dt;
    %Corner nodes
```

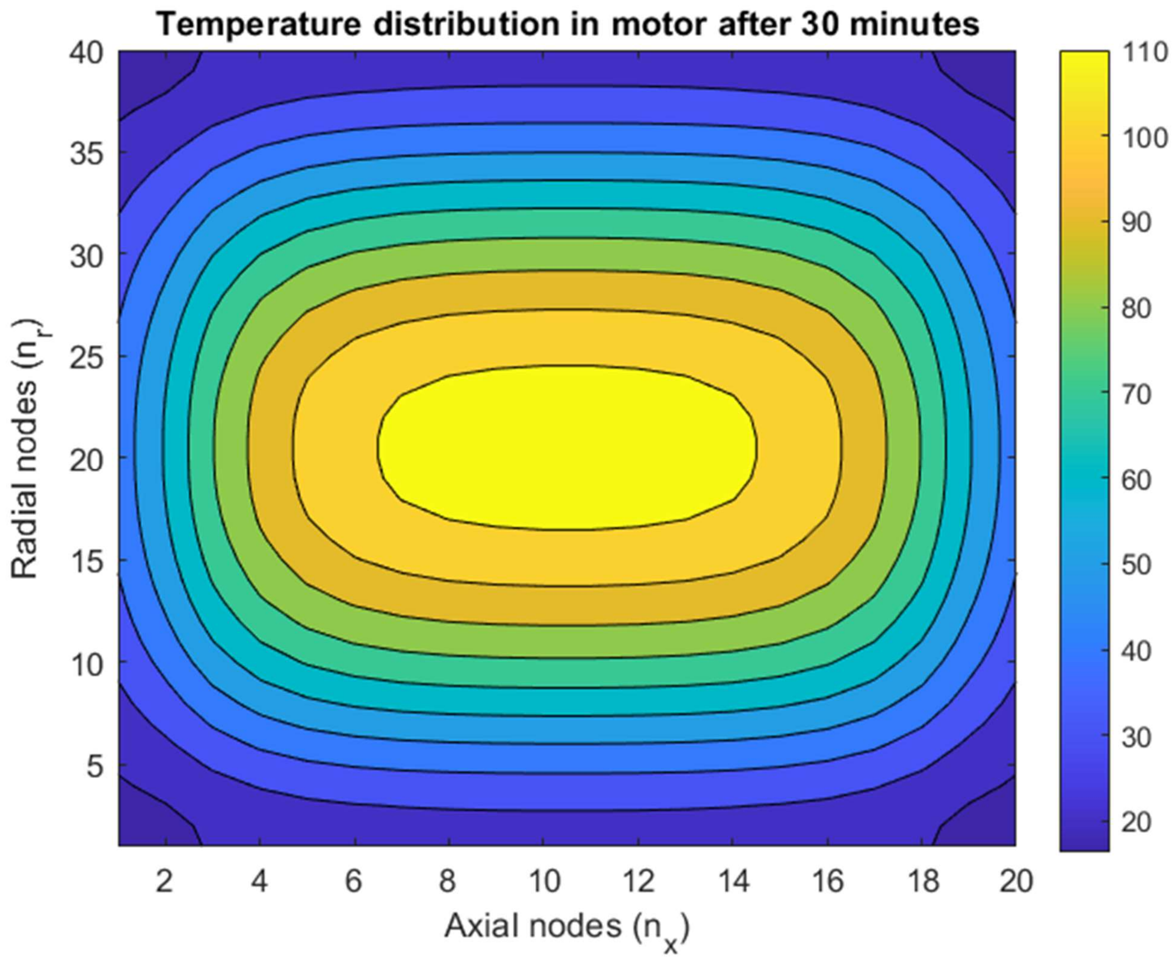
```
    %Element areas and volume (1,1) and (1, n_x)
    Ac_sides = ((pi*r_m^2)-(pi*(r_m-0.5*dr)^2));
    Ac_top = (2*pi*r_m*0.5*dx);
    Ac_bottom = (2*pi*(r_m-0.5*dr)*0.5*dx);
    Vc = Ac_sides*0.5*dx;
    if (Re(t,1,1) < 5*10^5)
        h_2 = h_lam(t,1,1);
    else
        h_2 = h_turb(t,1,1);
    end
    T_film(t,1,1);
    h_1 = h_cyl(t,1,1);
    %Left corner
    T(1,1) = T_prev(1,1) + (egen.*Vc + h_1.*Ac_top.*(T_a(t)-T_prev(1,1))...
        + h_2.*Ac_sides.*(T_a(t)-T_prev(1,1)) + k.*Ac_bottom.*(T_prev(2,1)-
T_prev(1,1))./(0.5*dr)...
        + k.*Ac_sides.*(T_prev(1,2)-T_prev(1,1))./(0.5*dx)).*(dt/(rho*Vc*c));
    %Right corner
    T(1,n_x) = T_prev(1,n_x) + (egen.*Vc + h_1.*Ac_top.*(T_a(t)-T_prev(1,n_x))...
        + h_2.*Ac_sides.*(T_a(t)-T_prev(1,n_x)) + k.*Ac_bottom.*(T_prev(2,n_x)-
T_prev(1,n_x))./(0.5*dr)...
        + k.*Ac_sides.*(T_prev(1,n_x-1)-T_prev(1,n_x))./(0.5*dx)).*(dt/(rho*Vc*c));
    %Side central nodes
    %Element areas and volume (n_r,1) and (n_r, n_x)
    Asc_sides = (pi*(0.5*dr)^2);
    Asc_top = (2*pi*0.5*dr*0.5*dx);
    Vsc = Asc_sides*0.5*dx;
    if (Re(t,n_r,1) < 5*10^5)
        h_2 = h_lam(t,n_r,1);
    else
        h_2 = h_turb(t,n_r,1);
    end
    T_film(t,n_r,1);
    h_1 = h_cyl(t,n_r,1);
    %Left side
    T(n_r,1) = T_prev(n_r,1) + (egen.*Vsc + h_2.*Asc_sides.*(T_a(t)-T_prev(n_r,1))...
        + k.*Asc_sides.*(T_prev(n_r,2)-T_prev(n_r,1))./(0.5*dx)...
        + k.*Asc_top.*(T_prev(n_r-1,1)-T_prev(n_r,1))./(dr)).*(dt/(rho*Vsc*c));
    %Right side
    T(n_r,n_x) = T_prev(n_r,n_x) + (egen.*Vsc + h_2.*Asc_sides.*(T_a(t)-
T_prev(n_r,n_x))...
        + k.*Asc_sides.*(T_prev(n_r,n_x-1)-T_prev(n_r,n_x))./(0.5*dx)...
        + k.*Asc_top.*(T_prev(n_r-1,n_x)-T_prev(n_r,n_x))./(dr)).*(dt/(rho*Vsc*c));
    %Nodes loop
    for j = 1:n_r %rows
        if (j>1 && j<n_r)
            %Side non-central nodes
            %Element areas and volume (2,1) -> (n_r-1,1) and (2, n_x) -> (n_r-1, n_x)
            As_sides = ((pi*((n_r-j).*dr+0.5*dr)^2)-(pi*((n_r-j).*dr-0.5*dr)^2));
            As_top = (2*pi*((n_r-j).*dr+0.5*dr)*0.5*dx);
            As_bottom = (2*pi*((n_r-j).*dr-0.5*dr)*0.5*dx);
            Vs = As_sides*0.5*dx;
            if (Re(t,j,1) < 5*10^5)
            h_2 = h_lam(t,j,1);
            else
            h_2 = h_turb(t,j,1);
            end
            T_film(t,j,1);
```

```
            h_1 = h_cyl(t,j,1);
            %Left side
            T(j,1) = T_prev(j,1) + (egen.*Vs + k.*As_top.*(T_prev(j-1,1)-
T_prev(j,1))./(0.5*dr)...
            + h_2.*As_sides.*(T_a(t)-T_prev(j,1)) + k.*As_bottom.*(T_prev(j+1,1)-
T_prev(j,1))./(0.5*dr)...
            + k.*As_sides.*(T_prev(j,2)-T_prev(j,1))./(0.5*dx)).*(dt/(rho*Vs*c));
            %Right side
            T(j,n_x) = T_prev(j,n_x) + (egen.*Vs + k.*As_top.*(T_prev(j-1,n_x)-
T_prev(j,n_x))./(0.5*dr)...
            + h_2.*As_sides.*(T_a(t)-T_prev(j,n_x)) + k.*As_bottom.*(T_prev(j+1,n_x)-
T_prev(j,n_x))./(0.5*dr)...
            + k.*As_sides.*(T_prev(j,n_x-1)-T_prev(j,n_x))./(0.5*dx)).*(dt/(rho*Vs*c));
        end
        for m = 1:n_x %columns
            if (m>1 && m<n_x)
                %Interior central nodes
                %Element areas and volume (1,2) -> (1,n_x-1)
                Aic_sides = (pi*(0.5*dr)^2);
                Aic_top = (2*pi*0.5*dr*dx);
                Vic = Aic_sides*dx;
                %Interior
                T(n_r,m) = T_prev(n_r,m) + (egen.*Vic + k.*Aic_top.*(T_prev(n_r-1,m)-
T_prev(n_r,m))./(dr)...
                + k.*Aic_sides.*(T_prev(n_r,m-1)-T_prev(n_r,m))./(dx)...
                + k.*Aic_sides.*(T_prev(n_r,m+1)-T_prev(n_r,m))./(dx)).*(dt/(rho*Vic*c));

                %Top nodes
                %Element areas and volume (1,2) -> (1,n_x-1)
                At_sides = ((pi*(r_m)^2)-(pi*(r_m-0.5*dr)^2));
                At_top = (2*pi*r_m*dx);
                At_bottom = (2*pi*(r_m-0.5*dr)*dx);
                Vt = At_sides*dx;
                if (Re(t,1,m) < 5*10^5)
                    h_2 = h_lam(t,1,m);
                else
                    h_2 = h_turb(t,1,m);
                end
                T_film(t,1,m);
                h_1 = h_cyl(t,1,m);
                %Top
                T(1,m) = T_prev(1,m) + (egen.*Vt + h_1.*At_top.*(T_a(t)-T_prev(1,m))...
                + k.*At_sides.*(T_prev(1,m-1)-T_prev(1,m))./(dx)...
                + k.*At_bottom.*(T_prev(2,m)-T_prev(1,m))./(0.5*dr)...
                + k.*At_sides.*(T_prev(1,m+1)-T_prev(1,m))./(dx)).*(dt/(rho*Vt*c));
            end

            if (m>1 && m<n_x && j>1 && j<n_r)
                %Interior non-central nodes
                %Element areas and volume (2,2) -> (2,n_x-1) and (2,2) -> (n_r-1,2)
                Ai_sides = ((pi*((n_r-j).*dr+0.5*dr)^2)-(pi*((n_r-j).*dr-0.5*dr)^2));
                Ai_top = (2*pi*((n_r-j).*dr+0.5*dr)*dx);
                Ai_bottom = (2*pi*((n_r-j).*dr-0.5*dr)*dx);
                Vi = Ai_sides*dx;
                %Interior
                T(j,m) = T_prev(j,m) + (egen.*Vi + k.*Ai_top.*(T_prev(j-1,m)-
T_prev(j,m))./(dr)...
```

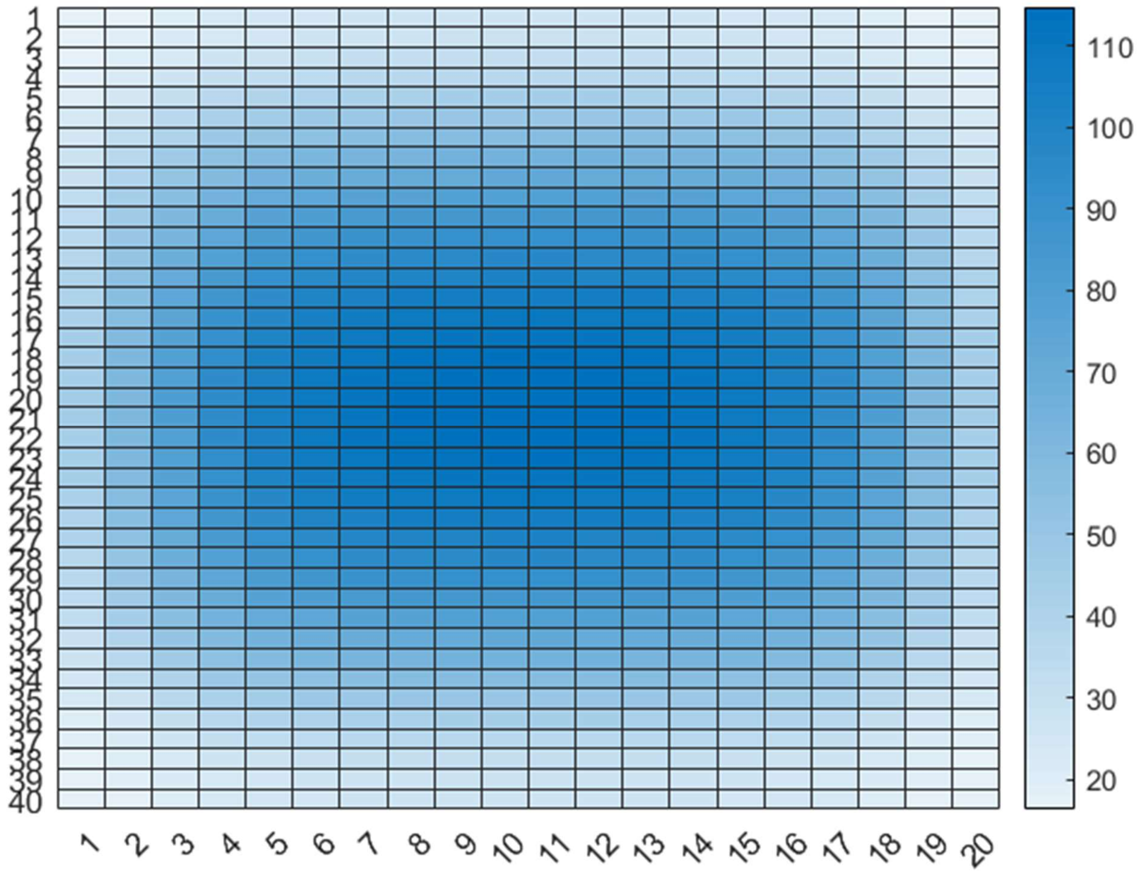
```
                + k.*Ai_sides.*(T_prev(j,m-1)-T_prev(j,m))./(dx) +
k.*Ai_bottom.*(T_prev(j+1,m)-T_prev(j,m))./(dr)...
                + k.*Ai_sides.*(T_prev(j,m+1)-T_prev(j,m))./(dx)).*(dt/(rho*Vi*c));
            end
        end
    end
    T_prev = T;
    T_whole4 = [T;flipud(T)];
    max_temp_vector(i) = T(20,10);
    plot(i,max_temp_vector(i,1), 'bo');
    xlabel("Time steps");
    ylabel("Temperature of motor core (deg C)");
    title("Temperature profile of the motor core during 30-minute operation");
    hold on;
end
hold off
```




```
contourf(1:n_x, 1:2*n_r, T_whole4);
colorbar;
hold on
xlabel("Axial nodes (n_x)");
ylabel("Radial nodes (n_r)");
title("Temperature distribution in motor after 30 minutes");
hold off;
```




```
heatmap(T_whole4);
colorbar;
```




```
T_max = T(20,10)
T_max = 114.8149
```

A.2 Validation of numerical model MATLAB code

To validate the numerical model, certain motor material properties are chosen so as to compare the results to the analytical model. I.e., the same conditions are chosen for both the analytical model and the numerical model for this validation.

The centre of the motor after a time of 15 minutes being exposed to ambient air conditions (during the flight) is to be determined using both methods and compared. For the purpose of the validation, the following assumptions are made:

- Heat generation is set to zero.
- The convection heat transfer coefficient is constant throughout.
- Fourier number > 0.2 so that the one-term approximate solution can be used.
- The initial temperature of the motor is equal to the ambient air temperature at time = 0 s.

Properties of air for validation model

```
%Ambient air temperature profile with respect to time in seconds from 0 to 1800
T_a = @(t) (-(7.0263*10^-17)*t.^6 + (3.7942*10^-13)*t.^5 - (6.5001*10^-10)*t.^4 + ...
    (2.9117*10^-7)*t.^3 + (2.4549*10^-4)*t.^2 - ...
    (2.4975*10^-1)*t + (1.4973*10)); %degrees Celsius
h = 250; %W/m^2K
T_a_900 = T_a(900) %deg C
```

```
T_a_900 = -38.4607
```

Properties of motor for validation model

```
k = 0.5; %W/mK
rho = 8800; %kg/m^3
Cp = 420; %J/kgK
%Intial motor temperature equals ambient air temperature at t = 0 s
T_i = T_a(0); %deg C
r_m = 0.025; %m
L = 0.1; %m


alpha = k/(rho*c); %thermal diffusivity
Volume = pi*(r_m^2)*L; %m^3
egen = 0; %W/m^3


%Air properties
%Ambient air temperature profile with respect to time in seconds from 0 to 1800
T_a = @(t) (-(7.0263*10^-17)*t.^6 + (3.7942*10^-13)*t.^5 - (6.5001*10^-10)*t.^4 + ...
    (2.9117*10^-7)*t.^3 + (2.4549*10^-4)*t.^2 - ...
    (2.4975*10^-1)*t + (1.4973*10)); %degrees Celsius


%Ambient air pressure profile with respect to time in seconds from 0 to 1800
P_a = @(t) ((7.6189*10^-14)*t.^6 - (4.1143*10^-10)*t.^5 + (9.9884*10^-7)*t.^4 - ...
    (1.3741*10^-3)*t.^3 + (1.078)*t.^2 - ...
    (4.3423*10^2)*t + (1.0065*10^5)); %Pa


%Ambient air velocity profile with respect to time in seconds from 0 to 1800
V_a = @(t) ((3.5132*10^-16)*t.^6 - (1.8971*10^-12)*t.^5 + (3.2501*10^-9)*t.^4 - ...
    (1.4559*10^-6)*t.^3 - (1.2275*10^-3)*t.^2 + ...
```

```
    (1.2487)*t + (1.3409*10^-1)); %m/s


%Nodes
n_x = 10; %number of nodes in x direction
n_r = 10; %number of nodes in y direction
x = linspace(0, L, n_x); %x direction nodes
r = linspace(0, r_m, n_r); %radial direction nodes
dx = L/(n_x-1); %grid size in x direction
dr = r_m/(n_r-1); %grid size in radial direction


%Initializing with initial conditions (at time = 0 s)
T = T_a(0)*ones(n_x,n_r); %initializing T matrix of size n_x x n_r
T_prev = T;
reflect_initial = flipud(T_prev); %reflecting over the symmetrical plane
T_i = [reflect_initial(2:r,:); T_prev]; %initial temperature of motor
%heatmap(T_i);
%colorbar;


%Temperature dependent fluid properties
R = 287.08;
%Film temperature
T_film = @(t,r,x) ((T_a(t) + T_prev(n_r,n_x))./2 + 273);
%Density
rho_air = @(t,r,x) (P_a(t)./(R.*T_film(t,r,x)));
%Thermal conductivity
k_air = @(t,r,x) (-4.937787e-4 + 1.018087e-4.*T_film(t,r,x)...
    - 4.627937e-8.*(T_film(t,r,x)).^2 + 1.250603e-11.*(T_film(t,r,x)).^3);
%Specific heat
c_p_air = @(t,r,x) (1.045356e3 - 3.161783e-1.*T_film(t,r,x)...
    + 7.083814e-4.*T_film(t,r,x).^2 - 2.705209e-7.*T_film(t,r,x).^3);
%Dynamic viscosity
mu = @(t,r,x) (2.287973e-6 + 6.259793e-8.*T_film(t,r,x)...
    - 3.131956e-11.*(T_film(t,r,x)).^2 + 8.15038e-15.*(T_film(t,r,x)).^3);


%Time
time_total = 1800; %s
Ac_sides = 0.5*((pi*r_m^2)-(pi*(r_m-0.5*dr)^2));
Ac_top = (2*pi*r_m*0.5*dx);
Ac_bottom = (2*pi*(r_m-0.5*dr)*0.5*dx);
Vc = Ac_sides*0.5*dx;
primary_coefficient = (-(h*Ac_top)-(h*Ac_sides)...
    -(k*Ac_bottom)/(0.5*dr)-(k*Ac_sides)/(0.5*dx))*(1/(rho*Vc*c))
```

```
primary_coefficient = -0.2530
```

```
dt_max = -1/primary_coefficient %s
```

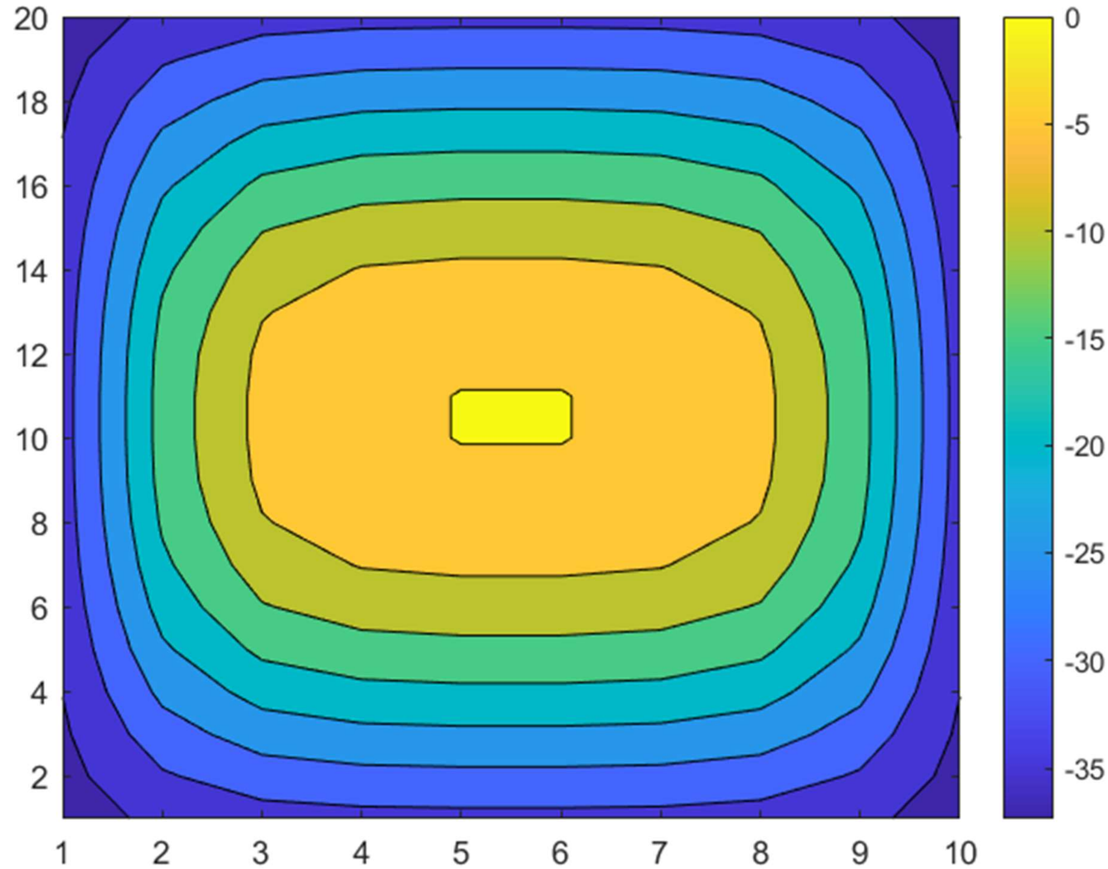
```
dt_max = 3.9526
```

```
dt = 0.5; %s
n_time = time_total/dt; %number of time steps
```

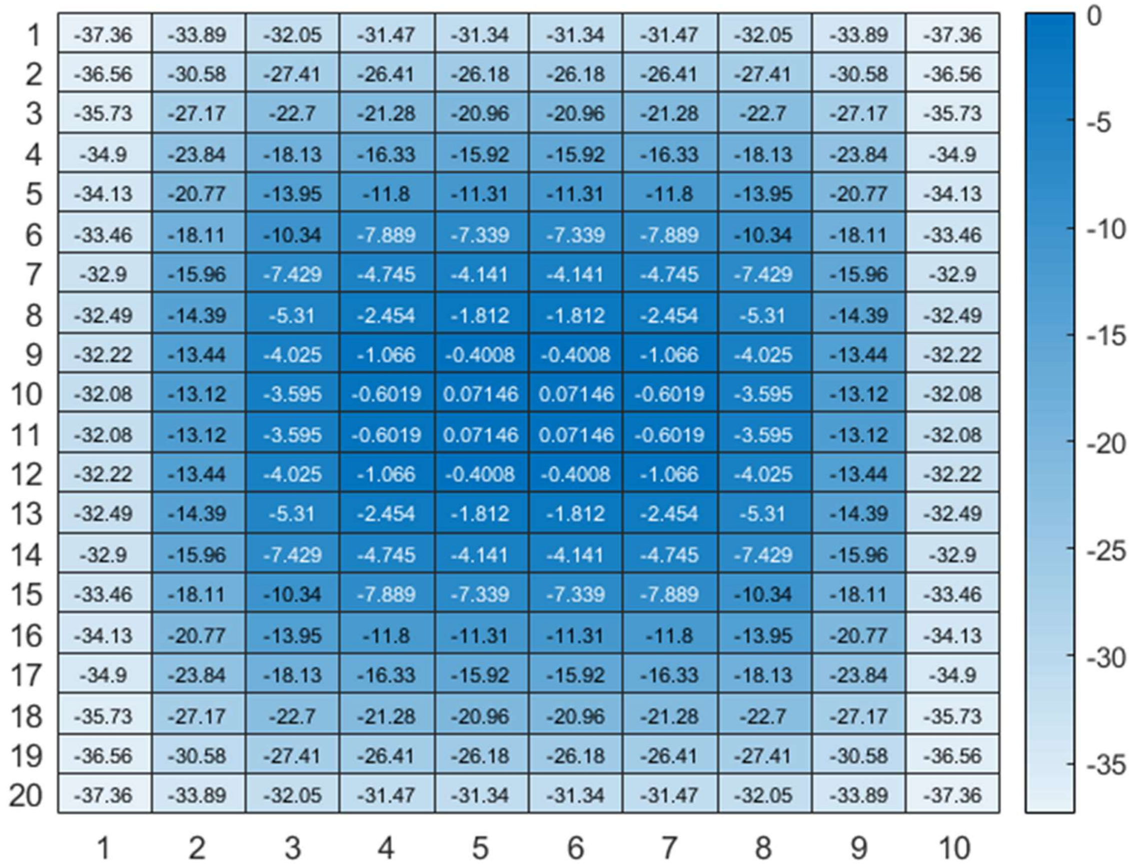
```
%Explicit method for transient 2D heat transfer
%Time loop
for i = 1:(n_time/2)
    T = zeros(n_r, n_x);
    t = i*dt;
    %Corner nodes
```

```
    %Element areas and volume (1,1) and (1, n_x)
    Ac_sides = ((pi*r_m^2)-(pi*(r_m-0.5*dr)^2));
    Ac_top = (2*pi*r_m*0.5*dx);
    Ac_bottom = (2*pi*(r_m-0.5*dr)*0.5*dx);
    Vc = Ac_sides*0.5*dx;
    %Left corner
    T(1,1) = T_prev(1,1) + (egen.*Vc + h.*Ac_top.*(T_a(t)-T_prev(1,1))...
        + h.*Ac_sides.*(T_a(t)-T_prev(1,1)) + k.*Ac_bottom.*(T_prev(2,1)-
T_prev(1,1))./(0.5*dr)...
        + k.*Ac_sides.*(T_prev(1,2)-T_prev(1,1))./(0.5*dx)).*(dt/(rho*Vc*c));
    %Right corner
    T(1,n_x) = T_prev(1,n_x) + (egen.*Vc + h.*Ac_top.*(T_a(t)-T_prev(1,n_x))...
        + h.*Ac_sides.*(T_a(t)-T_prev(1,n_x)) + k.*Ac_bottom.*(T_prev(2,n_x)-
T_prev(1,n_x))./(0.5*dr)...
        + k.*Ac_sides.*(T_prev(1,n_x-1)-T_prev(1,n_x))./(0.5*dx)).*(dt/(rho*Vc*c));
    %Side central nodes
    %Element areas and volume (n_r,1) and (n_r, n_x)
    Asc_sides = (pi*(0.5*dr)^2);
    Asc_top = (2*pi*0.5*dr*0.5*dx);
    Vsc = Asc_sides*0.5*dx;
    %Left side
    T(n_r,1) = T_prev(n_r,1) + (egen.*Vsc + h.*Asc_sides.*(T_a(t)-T_prev(n_r,1))...
        + k.*Asc_sides.*(T_prev(n_r,2)-T_prev(n_r,1))./(0.5*dx)...
        + k.*Asc_top.*(T_prev(n_r-1,1)-T_prev(n_r,1))./(dr)).*(dt/(rho*Vsc*c));
    %Right side
    T(n_r,n_x) = T_prev(n_r,n_x) + (egen.*Vsc + h.*Asc_sides.*(T_a(t)-T_prev(n_r,n_x))...
        + k.*Asc_sides.*(T_prev(n_r,n_x-1)-T_prev(n_r,n_x))./(0.5*dx)...
        + k.*Asc_top.*(T_prev(n_r-1,n_x)-T_prev(n_r,n_x))./(dr)).*(dt/(rho*Vsc*c));
    %Nodes loop
    for j = 1:n_r %rows
        if (j>1 && j<n_r)
            %Side non-central nodes
            %Element areas and volume (2,1) -> (n_r-1,1) and (2, n_x) -> (n_r-1, n_x)
            As_sides = ((pi*((n_r-j).*dr+0.5*dr)^2)-(pi*((n_r-j).*dr-0.5*dr)^2));
            As_top = (2*pi*((n_r-j).*dr+0.5*dr)*0.5*dx);
            As_bottom = (2*pi*((n_r-j).*dr-0.5*dr)*0.5*dx);
            Vs = As_sides*0.5*dx;
            %Left side
            T(j,1) = T_prev(j,1) + (egen.*Vs + k.*As_top.*(T_prev(j-1,1)-
T_prev(j,1))./(0.5*dr)...
            + h.*As_sides.*(T_a(t)-T_prev(j,1)) + k.*As_bottom.*(T_prev(j+1,1)-
T_prev(j,1))./(0.5*dr)...
            + k.*As_sides.*(T_prev(j,2)-T_prev(j,1))./(0.5*dx)).*(dt/(rho*Vs*c));
            %Right side
            T(j,n_x) = T_prev(j,n_x) + (egen.*Vs + k.*As_top.*(T_prev(j-1,n_x)-
T_prev(j,n_x))./(0.5*dr)...
            + h.*As_sides.*(T_a(t)-T_prev(j,n_x)) + k.*As_bottom.*(T_prev(j+1,n_x)-
T_prev(j,n_x))./(0.5*dr)...
            + k.*As_sides.*(T_prev(j,n_x-1)-T_prev(j,n_x))./(0.5*dx)).*(dt/(rho*Vs*c));
        end
        for m = 1:n_x %columns
            if (m>1 && m<n_x)
                %Interior central nodes
                %Element areas and volume (1,2) -> (1,n_x-1)
                Aic_sides = (pi*(0.5*dr)^2);
                Aic_top = (2*pi*0.5*dr*dx);
                Vic = Aic_sides*dx;
                %Interior
```

```
                T(n_r,m) = T_prev(n_r,m) + (egen.*Vic + k.*Aic_top.*(T_prev(n_r-1,m)-
T_prev(n_r,m))./(dr)...
                + k.*Aic_sides.*(T_prev(n_r,m-1)-T_prev(n_r,m))./(dx)...
                + k.*Aic_sides.*(T_prev(n_r,m+1)-T_prev(n_r,m))./(dx)).*(dt/(rho*Vic*c));

                %Top nodes
                %Element areas and volume (1,2) -> (1,n_x-1)
                At_sides = ((pi*(r_m)^2)-(pi*(r_m-0.5*dr)^2));
                At_top = (2*pi*r_m*dx);
                At_bottom = (2*pi*(r_m-0.5*dr)*dx);
                Vt = At_sides*dx;
                %Top
                T(1,m) = T_prev(1,m) + (egen.*Vt + h.*At_top.*(T_a(t)-T_prev(1,m))...
                + k.*At_sides.*(T_prev(1,m-1)-T_prev(1,m))./(dx)...
                + k.*At_bottom.*(T_prev(2,m)-T_prev(1,m))./(0.5*dr)...
                + k.*At_sides.*(T_prev(1,m+1)-T_prev(1,m))./(dx)).*(dt/(rho*Vt*c));
            end

            if (m>1 && m<n_x && j>1 && j<n_r)
                %Interior non-central nodes
                %Element areas and volume (2,2) -> (2,n_x-1) and (2,2) -> (n_r-1,2)
                Ai_sides = ((pi*((n_r-j).*dr+0.5*dr)^2)-(pi*((n_r-j).*dr-0.5*dr)^2));
                Ai_top = (2*pi*((n_r-j).*dr+0.5*dr)*dx);
                Ai_bottom = (2*pi*((n_r-j).*dr-0.5*dr)*dx);
                Vi = Ai_sides*dx;
                %Interior
                T(j,m) = T_prev(j,m) + (egen.*Vi + k.*Ai_top.*(T_prev(j-1,m)-
T_prev(j,m))./(dr)...
                + k.*Ai_sides.*(T_prev(j,m-1)-T_prev(j,m))./(dx) +
k.*Ai_bottom.*(T_prev(j+1,m)-T_prev(j,m))./(dr)...
                + k.*Ai_sides.*(T_prev(j,m+1)-T_prev(j,m))./(dx)).*(dt/(rho*Vi*c));
            end
        end
    end
    T_prev = T;
    reflected = flipud(T);
    T_whole = [reflected(2:n_r,:); T];
    T_whole2 = flipud(T_prev);
    T_whole3 = [T_whole2;T_prev];
    T_whole4 = [T;flipud(T)];
end
contourf(1:n_x, 1:2*n_r, T_whole4);
colorbar;
```




```
heatmap(T_whole4);
colorbar;
```


1 -37.36 -33.89 -32.05 -31.47 -31.34 -31.34 -31.47 -32.05 -33.89 -37.36
2 -36.56 -30.58 -27.41 -26.41 -26.18 -26.18 -26.41 -27.41 -30.58 -36.56
3 -35.73 -27.17 -22.7 -21.28 -20.96 -20.96 -21.28 -22.7 -27.17 -35.73
4 -34.9 -23.84 -18.13 -16.33 -15.92 -15.92 -16.33 -18.13 -23.84 -34.9
5 -34.13 -20.77 -13.95 -11.8 -11.31 -11.31 -11.8 -13.95 -20.77 -34.13
6 -33.46 -18.11 -10.34 -7.889 -7.339 -7.339 -7.889 -10.34 -18.11 -33.46
7 -32.9 -15.96 -7.429 -4.745 -4.141 -4.141 -4.745 -7.429 -15.96 -32.9
8 -32.49 -14.39 -5.31 -2.454 -1.812 -1.812 -2.454 -5.31 -14.39 -32.49
9 -32.22 -13.44 -4.025 -1.066 -0.4008 -0.4008 -1.066 -4.025 -13.44 -32.22
10 -32.08 -13.12 -3.595 -0.6019 0.07146 0.07146 -0.6019 -3.595 -13.12 -32.08
11 -32.08 -13.12 -3.595 -0.6019 0.07146 0.07146 -0.6019 -3.595 -13.12 -32.08
12 -32.22 -13.44 -4.025 -1.066 -0.4008 -0.4008 -1.066 -4.025 -13.44 -32.22
13 -32.49 -14.39 -5.31 -2.454 -1.812 -1.812 -2.454 -5.31 -14.39 -32.49
14 -32.9 -15.96 -7.429 -4.745 -4.141 -4.141 -4.745 -7.429 -15.96 -32.9
15 -33.46 -18.11 -10.34 -7.889 -7.339 -7.339 -7.889 -10.34 -18.11 -33.46
16 -34.13 -20.77 -13.95 -11.8 -11.31 -11.31 -11.8 -13.95 -20.77 -34.13
17 -34.9 -23.84 -18.13 -16.33 -15.92 -15.92 -16.33 -18.13 -23.84 -34.9
18 -35.73 -27.17 -22.7 -21.28 -20.96 -20.96 -21.28 -22.7 -27.17 -35.73
19 -36.56 -30.58 -27.41 -26.41 -26.18 -26.18 -26.41 -27.41 -30.58 -36.56
20 -37.36 -33.89 -32.05 -31.47 -31.34 -31.34 -31.47 -32.05 -33.89 -37.36
1 2 3 4 5 6 7 8 9 10
0
-5
-10
-15
-20
-25
-30
-35