Thermal Modeling of Fused Deposition Modeling Process Using Discrete Elements

Chelsea Charles Menezes
*Clemson University*, chelseamenezes@gmail.com

Follow this and additional works at: [https://tigerprints.clemson.edu/all_theses](https://tigerprints.clemson.edu/all_theses)

**Recommended Citation**
[https://tigerprints.clemson.edu/all_theses/3590](https://tigerprints.clemson.edu/all_theses/3590)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.
ABSTRACT

Over the years various models have been proposed to predict the material behavior during fused deposition modeling (FDM) process. FDM produced components show anisotropic material properties as compared to the virgin materials produced via injection molding. The thermal phenomena subjected to the filament during the FDM process leads to temperature gradient between filaments within a layer and across layers. This affects the adhesion and bond formation between the filaments and leads to directional differences in bonding strengths throughout the component. In this study, discrete elements have been used to study the thermal cooling behavior of FDM deposited filaments. This approach is a discontinuous methodology which follows the idea of dividing the filaments into discrete elements with simplified geometry for calculating the thermal interactions between the elements. The model uses the lumped capacitance model along with a set of heat transfer boundary conditions that considered the contact between the element and the surroundings, between the element and the build plate, and between elements. The model was applied to different testcases of various sizes and different printing conditions. The behavior of the model was found to be consistent with previous alternative models and followed the observed behaviors in FDM printing.
DEDICATION

I dedicate this work to my parents Charles and Jennifer, who have always encouraged and believed in me. Thank you for being my pillars of support; I owe it all to you.
ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Cameron Turner, for his insights and guidance throughout this endeavor, without whom this would not have been possible. I would also like to thank Dr. Garett Pataky and Dr. Richard Miller for being on my committee and for their input. Lastly, I would like to thank my family and friends who helped me along the way.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Background Of Additive Manufacturing</td>
<td>1</td>
</tr>
<tr>
<td>Fused deposition modelling</td>
<td>2</td>
</tr>
<tr>
<td>Objective and scope</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER 2: LITERATURE REVIEW</td>
<td>7</td>
</tr>
<tr>
<td>Simulation of thermal behavior of FDM process</td>
<td>7</td>
</tr>
<tr>
<td>Methods of Monitoring the FDM process</td>
<td>18</td>
</tr>
<tr>
<td>CHAPTER 3: METHODOLOGY AND PRELIMINARY TESTS</td>
<td>22</td>
</tr>
<tr>
<td>Introduction</td>
<td>22</td>
</tr>
<tr>
<td>Model implementation</td>
<td>22</td>
</tr>
<tr>
<td>Next steps</td>
<td>39</td>
</tr>
<tr>
<td>CHAPTER 4: ADDITIONAL TESTS , BOND TIME AND PARAMETRIC STUDY OF PRINTING CONDITIONS</td>
<td>40</td>
</tr>
<tr>
<td>Table</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>Table 1. Process parameters and material properties</td>
<td>32</td>
</tr>
<tr>
<td>Table 2. Test cases</td>
<td>33</td>
</tr>
<tr>
<td>Table 3: Geometry of testcases</td>
<td>41</td>
</tr>
<tr>
<td>Table 4: Different combinations of printing parameters</td>
<td>42</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 1 : Representation of FDM process [6]</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2 Impacts-of-in-process-parameters-on-part [7]</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3. Assumed deposition sequence</td>
<td>27</td>
</tr>
<tr>
<td>Figure 4 : Flow chart of model</td>
<td>30</td>
</tr>
<tr>
<td>Figure 5: Temperature history of the test cases</td>
<td>34</td>
</tr>
<tr>
<td>Figure 6: Temperature history of the test cases - continued</td>
<td>35</td>
</tr>
<tr>
<td>Figure 7: Edge Cooling Effects enlarged from Figure 4b</td>
<td>36</td>
</tr>
<tr>
<td>Figure 8: Individual Element Behaviors</td>
<td>37</td>
</tr>
<tr>
<td>Figure 9 : Temperature history of testcase 4</td>
<td>43</td>
</tr>
<tr>
<td>Figure 10 : Temperature history of testcase 4...continued</td>
<td>44</td>
</tr>
<tr>
<td>Figure 11: Temperature history of testcase 6</td>
<td>44</td>
</tr>
<tr>
<td>Figure 12: Temperature history of testcase 7</td>
<td>45</td>
</tr>
<tr>
<td>Figure 15: Temperature history of testcase 8</td>
<td>47</td>
</tr>
<tr>
<td>Figure 16 : Thermal behavior of core and corner elements of testcase 2</td>
<td>48</td>
</tr>
<tr>
<td>Figure 17 : Thermal behavior of core and corner elements from 13 to 53 seconds</td>
<td>48</td>
</tr>
<tr>
<td>Figure 18 : Average Bonding time for Testcase 1</td>
<td>50</td>
</tr>
<tr>
<td>Figure 19 : Average Bonding time for Testcase 3</td>
<td>50</td>
</tr>
<tr>
<td>Figure 20: Average Bonding time for Testcase 4</td>
<td>51</td>
</tr>
</tbody>
</table>
Figure 21: Average Bonding time for Testcase 5 ............................................................. 51
Figure 22: Average Bonding time for Testcase 6 ............................................................. 52
Figure 23: Average Bonding time for Testcase 7 ............................................................. 52
Figure 24: Average Bonding time for Testcase 2 ............................................................. 53
Figure 25: Average Bonding time for Testcase 8 ............................................................. 53
Figure 26: Rate of cooling for different speeds after 30 seconds .................................... 55
Figure 27: Rate of cooling for different speeds after 1 minute ........................................ 56
Figure 28: Rate of cooling for different speeds after 3 minutes .................................... 56
Figure 29: Rate of cooling for different layer heights after 30 seconds ........................... 57
Figure 30: Rate of cooling for different layer heights after 1 minute ............................... 57
Figure 31: Rate of cooling for different layer heights after 3 minutes ........................... 58
Figure 32: Rate of cooling for different build plate temperature after 30 seconds .......... 59
Figure 33: Rate of cooling for different build plate temperature after 1 minute ............ 59
Figure 34: Rate of cooling for different build plate temperature after 3 minutes .......... 60
Figure 35: Rate of cooling for different chamber temperatures after 30 seconds .......... 61
Figure 36: Rate of cooling for different chamber temperatures after 1 minute .......... 62
Figure 37: Rate of cooling for different chamber temperatures after 3 minutes .......... 62
CHAPTER 1: INTRODUCTION

1.1 Background Of Additive Manufacturing

Traditional manufacturing technologies, also known as Subtractive Manufacturing (SM) processes (e.g. milling, turning, and drilling) produce 3D objects by successively removing material from a solid block of material using cutting tools. SM techniques have drawbacks like material waste, inability to create complex geometries and therefore have limitations in the optimal design of components for lean production due to process constraints.

Additive Manufacturing (AM) is the process of creating parts by depositing material layer-by-layer. A generic process for fabricating a part by AM starts with generating a 3D Computer Aided Design (CAD) model. This model is converted into an STL (Standard Triangle Language) file, which transforms the CAD geometry into a triangulated mesh format. Next, slicing software slices the model into horizontal layers. This software also determines an optimized toolpath for the extruder to generate the part boundary and infill pattern; and generates computer numeric control (CNC) commands to operate the printer, enabling the machine to print the final part. While, traditional subtractive manufacturing imposes design constraints upon the geometry and materials of the part; these constraints can be relaxed or even eliminated through AM processes. However, AM cannot compete with the current cost models of SM techniques for high production volumes [1].

There are various AM techniques like Stereolithography (SLA), Selective Laser Sintering (SLS), Fused Deposition Modelling (FDM), Inkjet Modeling (IJM), Direct Metal Deposition (DMD). These techniques differ in the way they build layers and the types of materials that can be built [1]. Since the inception of SLA in 1984 [2], AM has grown from being exclusively used for prototype parts, to being used for manufacturing parts with highly complex geometries to obtain
performance levels unobtainable with conventional techniques, to being a manufacturing technol-
gy for individual consumers using consumer level 3D printers. Thus, as AM techniques are more
available than they have been before, resulting in an extremely commercial and competitive sector
[3].

1.2 Fused deposition modelling

One polymer-based AM technique is Fused Deposition Modelling (FDM) [2] which in-
volves the extrusion of the raw and support material through a heated nozzle. During pre-pro-
cessing, build-preparation software takes a 3D CAD file and converts to a .STL file. This format
tessellates the part into a set of triangles. Now, the software slices the model vertically into thin
sections. The software then generates G-code instructions which is the path along the sliced cross-
sections that will be traced by the nozzle. The thermoplastic is heated to a semi-molten state and is
deposited on a bed. Since the newly extruded material is in a semi-molten state, it fuses with adja-
cent material previously deposited due to internal thermal energy[4] and polymer diffusion [5]. The
head then moves in the X–Y plane and deposits material according to the G-code instructions. The
platform holding the part then moves vertically in the Z axis to begin depositing a new layer on top
of the previous one. Some FDM machine models possess a second nozzle that extrudes support
material and builds support for any structure that has an overhang angle of less than 45 from hori-
zontal as a default. If the angle is less than 45, more than one-half of one bead is overhanging the
contour below it, and therefore is likely to fall. Figure 1 is the representation of the FDM process
[6]. Parts made using FDM is being seen in applications involving aerospace, automotive and
medical devices [2].
The main parameters that define the FDM process are described below,

i. Build orientation: Build orientation is defined as the way a part is oriented on the build platform with respect to X-, Y-, and Z-axes.

ii. Extrusion temperature: The temperature at which the filament of a material is heated during the FDM process.
iii. Layer thickness: This is the height of the deposited layers along the Z-axis. It is usually less than the diameter of the extruder nozzle and depends on the diameter of the nozzle.

iv. Print speed: This is the distance traveled by the extruder along the XY plane per unit time while extruding.

v. Raster/Filament width: Raster/filament width is defined as the width of filament extruded.

vi. Raster/Filament orientation: This is the direction of the deposited filament with respect to the X-axis of the build platform of the FDM machine.

vii. Air gap: The gap between two adjacent filaments in a layer.

viii. Infill density: Infill density is the percentage of volume of the printed component that is occupied with the filament material. The strength and mass of FDM build parts should increase with increase in infill density as it closer to being a complete solid.

ix. Infill pattern: Different infill patterns are used in parts to produce a strong and durable internal structure.

1.3 Objective and scope

The effects of different process parameters on part characteristics is illustrated in the Figure 2 [7]. The strength of the FDM part is primarily dependent on the build orientation, layer thickness raster width and raster orientation [7, 8]. The decrease in quality of FDM manufactured parts is from in-process parameters and from environmental factors including humidity and surrounding temperature, as well as due to the properties of the raw materials such as color and density. It is also dependent on the accuracy and precision of software that convert the CAD model to .STL file and the .STL file to g-code instructions. As a result, FDM manufactured parts exhibit non-homogeneous, anisotropic and nonlinear behavior [8]. This leads to the products have quality issues including dimensional errors, delamination of layers, porosity and poor or indeterminate material properties [9]. Dimensional accuracy, surface roughness, strength and other mechanical properties
could be improved by optimizing the process parameters[7]. Optimizing process parameters will require understanding and building of models that mathematically and physically represent the process.

Due to the nature of the polymer materials used in FDM, the bonds between filaments can only be formed when the filament contacts are above glass transition temperature $T_g$. Because of how the filaments are deposited layer by layer in FDM process, filaments are not subjected to temperatures over $T_g$ at the same time. The bond strength increases over, again in FDM process the time during which the bond is over $T_g$ is not the same for all filaments. This is the main underlying reason we see anisotropic properties in FDM produced is because the bond strength formed is not equal throughout the part. Therefore, there exists a need to simulate the thermal aspect of the FDM process during and after deposition [3] to be able to predict the material properties of finished product. Discrete element method (DEM) is an approximation method based of the interactions between discrete rigid objects with simplified geometries. This discontinuous methodology has be

Figure 2 Impacts-of-in-process-parameters-on-part [7]
used to model other AM technologies like Selective Laser Sintering [10]. The focus of this study is to attempt to develop a strategy that can be used to model the thermal behavior of the FDM products and to that end the design statement for this work is: “A DEM model can simulate the thermal behavior of an FDM part during deposition and cooling.”

Chapter 2 describes briefly previous approaches that have been used to simulate the temperature history of the FDM produced parts as well experimental setups used to obtain the temperature values. Chapter 3 describes the mathematical model, assumptions, boundary conditions of the discrete element model used and is applied to 3 testcases to observe the behavior of the elements in the model. In chapter 4, the model is applied to more testcase the bonding time is estimated along with a study of the printing conditions on the rate of cooling and the results are discussed. In chapter 5, we conclude by making remarks on the validity of the DEM simulation and list the avenues for future development.
CHAPTER 2: LITERATURE REVIEW

As stated earlier there is need to model the thermal and mechanical behavior of FDM produced and over the years several approaches have been proposed. This section below mainly focuses on the thermal modeling of the FDM and how the results of these models can be used to make calculations about the bond strength and stresses induced in the deposited filament. With models being proposed there arises a need to validate these models against experimental values. A few experimental setups have been designed specifically for the FDM process to find the temperature of the filaments during and after deposition. Some of which have been explained later in this chapter.

2.1 Simulation of thermal behavior of FDM process

2.1.1 1D and 2D based models

An analytical framework was proposed in [4] which was a thermal model that provided a fundamental understanding of the behavior of the material within a layer by simulating a for a single layer consisting of ten rows where each row modeled as a 1D array of blocks. This model predicted temperature and bonding in 2D and considered the thermal interactions between the deposited rows and accounted for their subsequent cooling. Keeping in mind the simplified geometry studied, the model predicted variation in bonding between rows within a given layer due to predicted lower temperatures near the edges of parts. They also predicted less bonding near the edges of parts and large increases in bonding as the convective heat transfer coefficient value increased. The study was not validated against experimental results.
This work was followed up in [11] with a 2D thermal model with emphasis on investigating the thermal design of liquefier entrance, analysis of the meltpool (the molten material in near the extruder) location, degree of cooling in the nozzle and the impact of nozzle design on output pressure. The model attempted to establish the relationship between the meltpool and the feed speed, filament size, and material diffusivity. They also predicted greater dependence of temperature profiles on the Peclet number than the Biot number, highlighting the importance of advection in the system. Again, this work was not validated against experimental data. Both these models took into consideration the cooling of a filament due to convection with the environment and disregarded thermal contacts with adjacent filaments.

Another scope of thermal interactions was explored in [12] where they assumed deposited rows to be of semi-infinite length compared to the cross-section. They used lumped capacitance analysis that assumed the temperature distribution within the cross-section is uniform. The cooling process of a single filament is thus simplified into a one-dimensional transient heat transfer model where a single deposited row is modeled as a one-dimensional block whose cross-sectional shape of the deposited filament is an ellipse. The focus of model on the thermal behavior of a single row allowed them to explore its influence on the entire part for different process parameters (extrusion temperature, envelope temperature, extruder tip size, row dimensions, fiber gap, and deposition pattern). They concluded that extrusion temperature and ambient temperature were the most important parameters that control the thermal profile for the simplified geometry.

The bonding of polymers like ABS involves a process of interpenetration of the molecular chains across an interface which is a thermally activated and only occurs at temperatures above the polymer's glass transition temperature. As molecular interpenetration increases, the interface gradually disappears as the neck formation occurs, and mechanical strength develops. This process is undertaken in five steps, (i) surface rearrangement, (ii) surface approach, (iii) wetting, (iv)
diffusion, and (v) randomization and the healing model for bond formation for isothermal processes as described in [13, 14]. Intimate contact between polymer surface involves surface rearrangement and surface approach which is achieved when interface is free from bumps and texture. Instantaneous intimate contact occurs if the bumps are small enough and intermolecular forces are sufficient, then wetting and diffusion is dominant [15]. Strength is not developed across an interface until one surface begins to wet the other. Wetting can be assumed to occur instantaneously if the interface temperature is initially much higher than the glass transition temperature $T_g$. Once wetting has occurred, strength continues to grow as the polymer chains diffuse across the wetted interface and can go on long as the interface temperature is above $T_g$, but it occurs more rapidly at higher temperatures.

Instantaneous wetting and diffusion were included in model of an FDM process in [5, 16, 17] while wetting over time models were developed in [18-21]. In [17], thermal histories at the row-to-row interface were obtained from a 2D model and was used to develop model predictions for the fracture strength of the resulting bond. This work was indirectly validated by predicting interface toughness of extruded specimen using calculated temperature profiles. The 2D analysis neglected heat transfer along the length of rows and assumed that previous layers cooled to ambient temperature before the deposition of a new layer began. It was also assumed the fibers had a rectangular cross-section, thus increasing the heat transfer between fibers and neglected the effects of conduction to the build plate. The fracture strength model developed used inputs from isothermal heating data, so it did not depend entirely on material properties. Modifications need to be made to the heat transfer model to generate a predictive model.

In [18], a 1D lumped capacity analysis examined the temperature along the length of a filament. Since the mass of the build plate is much higher than that of the filament, the conduction heat transfer with the build plate was modeled as convective heat transfer. The convective heat transfer
coefficient, h, accounted for the effects of both heat convection with the air and conduction with the build plate. As compared to [17], where the cross-section was taken as rectangular and diffusion is modelled, this model used an ellipsoidal cross-section and only modelled wetting, leading to a temperature gap between the model prediction and empirical data. The results demonstrated that extrusion temperature has a more significant impact on the neck growth of the bonding zone than the envelope temperature.

The bond strength between layers (z-axis strength) based on instantaneous wetting and diffusion across layers and was predicted from a 1-D transient heat analysis in [5]. The heat analysis considered the top two layers of an FDM part and heat transfer from the upper and lower surfaces to the environment with uniform temperature across the cross-section was considered. The viscosity was measured experimentally at low shear rates to predict the diffusion coefficient as a function of temperature. Through rheological testing the diffusion coefficient was calculated and then integrated across the transient temperature results which was used to calculate the bond strength where the effect of the layer height on the bond strength was also considered. The simulated bond strengths matched the measured bond strengths with a coefficient of determination of 0.795.

An analytical solution for the transient heat transfer during deposition based of [4, 17, 18] was proposed in [22] for a simple deposition sequence. Both radial and axial heat conduction was considered as negligible and contacts between filaments and environment were assumed to be convective in nature.

1D approaches that simulate along the length of a deposited filament models like [4, 17, 18] neglect heat transfer normal to the layer while 2D simulations normal to the row axis neglect heat transfer along row lengths like [17], but they all show temperature variation. This indicates limitations in such approaches and simulating heat transfer in FDM should ideally be done in three dimensions and account both types of thermal contact.
2.1.2 3D based models

In recent years, 3D models proposed for FDM have examined the thermal history of the deposited filament or the thermal and mechanical behavior of the deposited filaments or the thermal behavior and bond formation between the filament surfaces. Some of these models are described below.

A 3D thermal model of FDM was proposed [23] based on the finite element method where time discretization for the 3D space is based on a Chernoff strategy that converges to the temperature solution as long as the time step and meshing size is small enough. The model uses the ANSYS Parametric Design Language (APDL) strategy also used in [24]. The ABS material is extruded the filament is in “death” state with its temperature set to zero at the start of extrusion and then as the material is deposited in the FDM process, the “birth” is simulated with the element having a uniform initial temperature and initial boundary conditions followed by removal of the heat reservoir input of current finished cycle, and then addition of new heat reservoir at current position until the last filament element set is solved. The heat transfer model was solved based on the assumptions that uniform temperature distribution across the cross-sectional area of the filament and a semi-infinite filament length. The model uses the natural convection heat and radiation heat with the surrounding medium and since the mass of the build plate is much higher than ABS filament extruded, the conduction heat transfer with the plate was modeled as convection. The model includes the influence of temperature on thermal conductivity of the deposited material and the latent heat from phase transformation of polymer materials like acrylonitrile butadiene styrene (ABS) is included as the enthalpy varied with temperature in the model which was proposed in [24] that indicated the latter extruding steps have a larger heat impact than the former. The results of [23]
showed that the temperature of the single filament varied non-linearly with the extruding time and natural cooling of the ABS material was also non-linear.

A model for heat transfer examining the effect of convection and radiation (with the environment, entrapped air), and conduction (between the filaments and between the printed part and support along with mechanical deformation of a filament subjected to its own weight, as well as to the weight of a vertical stack of similar filaments on top was proposed in [25] using ABAQUS. The thermal model estimated the differences in temperature of nodes at the center and opposing edges of a given cross-section to assess the existence of a ‘thermally thin’ filament (i.e. with uniform temperature at each cross-section) and if the temperature evolution of the two elements was identical for the two boundary conditions, the longitudinal heat flux was taken as negligible. They results showed that convection and conduction have the highest impact on the thermal profiles however, the convection and radiation in the air pockets between ellipsoidal filaments have a negligible effect and temperatures along small filament length increments (i.e. axial conduction) was treated as uniform. The mechanical deformation calculations were performed for elastic and viscoelastic response described by a Prony series at isothermal loadings at 95°C, 110°C and 135°C. The results showed that at the temperatures studied, the mechanical deformation of the filaments is negligible and therefore should not affect the dimensional accuracy of the part and heat transfer because the variation in surface area was small. This model was not validated against experimental data.

In another study, the thermal and stress analysis of the FDM process based on FEM using voxelization in ANSYS was presented in [26]. Here, the conversion of a three-dimensional model described by surface or volume information to voxels which fit the shape of the model is called voxelization. The model initially voxelizes the surface of the model, followed by the voxelization
of the model interior. The model adopted in [18] makes use of orthogonal parallel filling, which is when the layers are extruded in horizontal and vertical parallel filament lines alternatively. After the voxels were sorted according to the printing path and then converted to FE model which involves complete the transformation of the coordinate information of the voxel that is hexahedral in nature to the element and node information of the FE model. The FE element type SOLID70 was considered in this paper which has three directions of thermal conductivity, and each node has a temperature degree of freedom, enabling uniform heat flow. The element was set to a to a cube with a side length of 0.5 mm equal to the diameter of the extruded filament. The thermal analysis model using in the ANSYS Parametric Design Language (APDL) strategy is based on [23]. The results show that the lowest temperature was at the bottom of the component because the elements on the bottom are printed first as compared to the top elements that are printed last. The temperature gradient at a low printing speed was larger than that at a high printing speed because orthogonal parallel filling was used. The results also showed that the maximum temperature gradients under the different molding chamber temperatures and under different nozzle temperatures were consistent. The trends showed that increasing the chamber temperature and reducing the nozzle temperature can reduce the temperature gradient seen in the molding process. Using the sequential coupling method, the thermal analysis element SOLID70, was converted to a structural analysis element SOLID185. After the constraints were set, the calculated results of the temperature field were applied as loads to the FE model according to the procedure of the sequential coupling method and then solved and to study the residual stress and deformation distribution in the part after cooling, with a cooling period of 20 seconds set in the simulation. The results showed that the maximum deformation was seen at the corner of the top surface of the component and minimum deformation was seen in the central part of the bottom surface of the component. The results also showed that the bottom surface of the part warped. This is consistent with observations of actual printed parts
under the same printing conditions indicating that the simulated results were accurate. The simulation showed that increasing the printing speed and the molding chamber temperature and reducing the nozzle temperature helps reduce the internal stress and the warping deformation of the part. This was verified by experiments which produced the same correlation where the specimens were built in an experimental platform which integrated a printing mechanism inside an incubator whose model chamber temperature could be adjusted.

The stress and thermal coupling analysis used in [26] was also applied to different scanning patterns namely honeycomb, grid, wiggle, rectilinear in [27]. The simulated thermal distribution showed that the smallest temperature gradient is found for honeycomb infill pattern and also predicted uniform stress distribution and the smallest deformation for a honeycomb infill pattern which can provide guiding significance for actual processing.

Another model was proposed that material flow need not be directly simulated. Instead an increase in z-direction at any one point could be simulated by a uniform increase in the material deposited in the z-direction across the entire build plane [28]. The geometry and printing conditions in [20] matched those seen in [29], so let’s compare the simulated results against [29]. All modeling was done using finite element analysis (FEA) in COMSOL Multiphysics. Assuming that radiation is negligible, only when the nozzle is in contact with the printed sample the nozzle movement and material deposition is simulated using COMSOL’s “Deformed Geometry” node. This deforms the mesh as boundaries move and as new material is added to the system. COMSOL adds energy and maintains the system’s temperature as the boundaries move due to the addition of material. However, this involves calculating the average temperature of the part for every time step, making the model complex. To simplify the model it was assumed that the “Deformed Geometry” node does not add heat to ensure newly added material is at the same temperature as the surrounding material. This assumption is valid for parts with constant cross-sections discussed in the results and that a
full part may not need to be simulated and the simulation could be run until layer thermal profiles converge and non-simulated layers could be treated as identical to the final simulated layers. Conduction from the build plate to the part surface and between the nozzle tip and the part assumes no thermal resistance. The results of the simulation follow the trends of the experimental data, but the model predicts more rapid cooling to a lower steady-state temperature than observed in experimental results. This discrepancy is likely due to overestimation of the thermal transfer and the material properties of ABS. Also, cooling was shown to be most rapid for print speeds between 10 and 30 mm/sec, which could be because at low print speeds, the hot nozzle remains near the recently deposited point for a longer time and slows cooling while at high print speeds, the nozzle moves through each layer faster and begins depositing a new rows adjacent to old rows more rapidly, effectively raising the steady-state temperature and slowing the cooling rate. As larger layers of geometry were stimulated, the corner points cooled the fastest, followed by side points and finally the interior points. A line geometry with increasing layer count led to an increase in the minimum temperature and a decrease in heat loss due to the build plate. This points out complications that could arise when scaling these models as there is a dependence on geometry, and it also shows a need to simulate parts closer to build scales.

Roy et al [22] also used an FEM based birth-death model to predict the temperature history of the FDM process using Abaqus with a custom developed code used to discretize the volume of the printed component using the G-code to find the deposition order. Elements are activated or “birth” occurs from the information given by code leads to the location of the corresponding free surfaces being tracked in the user-defined ABAQUS-specific subroutine that adds these new elements to the mesh and updates the boundary conditions. This setup was shown to be computationally expensive, and the computational time increases exponentially with increase in number of layers. In this study,
the predicted results were compared with in-process data from an experimental setup where a test specimen was built in a Hyrel Hydra FFF (by Hyrel3D) machine using ABS material. This model assumes that the nozzle is a moving heat source, with the heat input originating from the extruded material deposited at the extruder temperature. The material extruded is homogeneous and isotropic and the material properties such as the specific heat capacity, density, and conductivity are temperature independent. The latent heat generated due to the change in the material from a liquid to solid-state is not considered. The shrinkage in the material due to cooling is neglected and the effect of warping and distortion on the shape of the deposited rows is not accounted for in the simulation. The ambient temperature and base plate temperature are considered constant during the process. The transient thermal model comprises of conductive heat transfer from the build plate at a constant temperature to the extruded filament, conductive heat transfer between the deposited filament within a layer and across layer assuming perfect contact between them. Convective and radiative heat transfer between the free surfaces of part and the build chamber kept at ambient temperature was considered in the model. The heat transfer due to forced convection because of the fan that blows air over the part, although not active during the entire fabrication process was also considered in the heat model. Heat transfer related phenomena like latent heat generation due to material solidification also was included in the model. As mentioned previously, the latent heat and convection effects within the melt pool was ignored. The test specimen used was a two-tier stepped-pyramid with a total of ten layers and each tier accounted for with five layers. This geometry lead to an effect on the thermal history where the same cyclical, repeating pattern in the temperature trends was not observed. Instead the temperature distribution for the second tier remained at a higher temperature for a longer time than the first tier. This could be due to the smaller area of the second tier and having shorter time to cool between deposition of individual rows and layers. Within each layer a cyclical pattern is observed, which corresponds to the deposition of an individual row within
each layer. The predicted temperature trends matched the experimental data and the 6% mean absolute percentage error, and 6 C root mean squared error of the experimental observations was attributed to no control over fan near the extruder, extruder and build plate temperature not being constant, vibrations caused by the equipment used in the setup and the distortion in the part affecting the temperature history.

An analytical solution for heat transfer in FDM by sectioning the part into elements that are added at discrete time points, assuming each element has a constant temperature across that element and conductive heat transfer between filaments and between filament and support, convective heat transfer between filament and environment was considered in [30]. The heat transfer coefficients were determined using experimental data and an adhesion criterion to predict the degree of healing between adjacent filaments was also adopted based on the predicted temperature values. The predicted filament surface temperatures matched with the experimental data and the results of the peel-like tests also matched with the adhesion predictions.

A computational framework for the simulating a two-stage thermal deposition and sintering model to predict the bond formation process was proposed in [31]. The thermal part calculated the temperatures of the filament segments taking into account the contacts between them and with the support based on [30] which then was used to in the sintering model where the surfaces disappear and neck formation occurs leading to updated contacts between filaments and support in the thermal part of the model. The surface tension due to neck formation was used to calculate the angle of intersection of the filaments which was used in predicting the void density and then the mechanical properties of the printed part. This was based on the treating ABS as an amorphous material and thus was modeled as a Newtonian viscous flow where surface tension is the main driving force. The cross-section of a filament was taken as half-discs and a rectangle compared to the circular
cross-section used in [30]. The model predictions were able to reproduce the trends observed in experimental results.

2.2 Methods of Monitoring the FDM process

2.2.1 Embedding sensors

In an experimental study [16], the specimens were made using ABS material on a Stratasys FDM 2000 and the print temperature settings recommended by the manufacturer were used. Thermocouples were placed on the lower portion of the extruder tip and in the foam of the base plate of the FDM machine in the center of the cross-sections of the specimens where the filament was deposited. The experimental temperature values were compared to those from the models mentioned in [32, 33] and were found to be inadequate as they underestimate the heat conduction within the parts, or neglect variations in the convective conditions within the parts and during the fabrication process. The quality of the bond between the filaments was measured based on neck growth and was calculated using the failure of the bonds under three-point bending tests. These experiments showed that the surrounding temperatures and variations in the convective conditions within the building chamber have strong effects on the necking and the overall quality of the bond strength. Under this experimental study’s printing condition, the neck formation was found to have a significant effect on bond formation, but only for the duration when the filament’s temperature was above the critical sintering temperature. This experiment proposes that creep deformation and molecular diffusion need to be considered in predicting the bond strength development if the extruded filament temperature is above the glass transition temperature, $T_g$, yet below the critical sintering temperature, during the fabrication process.
One of other type of sensors that can be embedded to monitor temperature distribution and strains during the FDM is using Fiber Bragg grating (FBG) sensors. In [34], FBG sensors and thermocouples placed within different layer locations inside different specimens made in a Stratasys (Dimension Elite) FDM 3D printer by using the commercial ABS P430 as the model material and the P400SR as the support material. The experiment investigated in-situ and in real-time the generated residual strains and temperature profiles development during the fabrication process and also calculated the post-fabrication residual strains of the samples. The results show that the magnitude of the residual strains that are developed when the first layers are deposited and solidified do not change considerably with subsequent layer deposition. Also, a correlation was found that residual strains are developed during the fabrication process when the in-situ temperature values of the deposited material remain below or close to its glass transition temperature. It also demonstrated that the component's position on the building platform influences the magnitude of the developed residual strains and the generated temperature variations.

Embedding thermocouples or contact based sensors inside the build plate is not a viable strategy in FDM because of the poor thermal conductivity of the polymer materials in FDM, lead to temperature fluctuations at locations that are not in proximity to the sensor not being measured. Therefore, using infrared sensor or IR cameras is a promising technique that could help track the temperature profile during printing.

2.2.2 IR thermography

In one experiment [35], an extended-range infrared (IR) camera was placed outside the chamber and behind a window to view the entire build volume and measure the temperature uniformity of the chamber and individual parts as they are being printed. Another longwave IR camera is mounted on the extruder head providing a high resolution image and temperature measurements of the thermoplastic as it is extruded and begins to cool on contact with the layer of material below.
The samples whose infrared imaging was developed were made in a Fortus 900mc Production Printer. The results from the camera placed outside the chamber showed that typically the temperature of the surroundings on the right side of the chamber was higher and that the chamber is cooler as you move from top to bottom. During printing, a temperature gradient within the sample was observed from top to bottom and from left to right. The horizontal temperature gradient was not expected and may be the cause of the small geometric distortions in the final part.

In other experiment [29], infrared (IR) thermography was used to measure temperature profiles during deposition of an ABS sample with following dimension: 160 mm long (x-axis), 2.4 mm tall (z-axis, 8 layers at 0.3 mm), and 0.4 mm wide (y-axis, extruder diameter). The sample was printed at the front edge of the build plate and 10 mm (y-axis) by 20 mm (x-axis) “feet” were added to the start and end of the first layer to prevent the sample from detaching from the build plate during the printing process. The build plate was set at 110 °C and the first layer was printed using the default MakerBot Desktop output. The thermal images were captured form the side because the camera was placed at angle at a considerable distance from the extruder. Here, a correction to the IR intensity is applied to remove reflected IR intensity generated by the hot extrusion head and then a separate set of offline measurements provide the conversion from IR intensity to temperature. The temperature of the layer at the decreases very quickly and a small amount of heat is transferred to the layer under it (first sublayer) while the second sublayer never reaches glass transition temperature (T_g). The temperature at the interface between the two layers were extremely difficult to measure so the interface (weld) temperature was estimated using the average of the two adjacent layers. The estimated weld temperatures between sublayer 1 and sublayer 2 does not rise above T_g, i.e., the weld was not annealed during printing and since the weld temperatures drop quickly, little time is available for weld formation (less than 2 seconds). The FEM based model in [22] was
experimentally evaluated. The experimental setup involved three infrared K-type thermocouple sensors (Exergen-150046) placed near the extruder head such that it can optically measure the temperature of the filament as it is being deposited at a distance from the extruder head [36]. The light-weight sensors travel along without interfering the kinetics of the machine or by imposing large inertial mass on the machine. Since the sensors are placed at an angle that causes a fan beam-type field of view which leads to more area than one row width being viewed, and a portion of the extruder is also scanned. To equate the temperature data acquired by the sensors with the trends predicted by the thermal model, the temperature distribution from the simulation is averaged as follows: \( T = w_1 T_p + w_2 T_{ext} \), where weights \( w_1 = 0.784 \), corresponding to the area fraction of the part and base, and \( w_2 = 0.216 \) is the area fraction blocked by the extruder as measured from the computer-aided design (CAD) model to be 21.6%. The extruder temperature \( T_{ext} \) is taken as constant, while the part temperature \( T_p \) is predicted by the model.
3.1 Introduction

This chapter contains a paper submitted and reviewed for the ASME IMECE® 2021 International Mechanical Engineering Congress & Exposition Virtual Conference. The paper was written by Chelsea Menezes and Dr. Cameron Turner.

The paper discusses the assumptions made in order to implement DEM model, the mathematical model used and how the changing boundary conditions are handled in the model. Preliminary tests are then carried out to check the validity of the model.

3.2 Model implementation

IMECE2021-71947 IMPLEMENTING A DISCRETE ELEMENT METHOD FOR FUSED DEPOSITION MODELING ADDITIVE MANUFACTURING THERMAL MODELING

3.2.1 Abstract

One of the challenges of additively manufactured parts is that additive manufacturing processes can lead to anisotropic material properties. For this research, we focus on fused deposition modeling (FDM) which is a common consumer grade process that also is often used for early prototyping due to its low investment and processing cost. In FDM, a semi-molten filament is extruded. Within the filament (the intra-filament bonds), the material properties are essentially that of a bulk material that has been injection molded. However, because of the differences in temperature when adjacent filaments are deposited, the material properties between filaments within a layer (the
intralayer bonds) are generally less than that of the intra-filament bonds leading to an anisotropic behavior within a layer. Similarly, the temperature difference between layers leads to yet another different material bonding strength (interlayer) that is also less than that of the intralayer or intra-filament bonds.

Our hypothesis is that these anisotropic property differences can be predicted using Discrete Element Models (DEM) to model the process of printing the part filament by filament and layer by layer and the subsequent cooling process. A DEM approach discretizes the filament into discrete elements that are treated as a lumped parameter elements connected to adjacent elements through a set of heat transfer boundary conditions. Elements with external part surfaces are therefore connected to the external environment, or in the case of elements in the base layer of the part, are connected to the print bed which is often heated to encourage bonding between filaments.

The DEM model is validated by comparing the predictions of the model against observed behaviors in FDM printing. For instance, the exposed surfaces of an FDM print will cool faster than elements in the core of the print, or elements that are in contact with the heated printing bed.

This paper describes the process of developing a thermal DEM model in MatLAB, including the assumptions underlying the element level heat transfer model. In addition, discussion of the model results is included to demonstrate the validity of the model as well as the comparisons made to available simulation and experimental data which allows us to validate the underlying behavior of the model. As a result of this research, there are several avenues available for future work including the estimation of bond strength between fibers and layers, the incorporation of viscosity effects, mechanical loading, and the possibilities for process optimization based on intelligent filament path planning, reheating technologies and adaptively controlling the build plate and environmental temperatures.
3.2.2 Introduction

Additive Manufacturing (AM) is the process of creating parts by depositing material layer-by-layer. There are various AM techniques like Stereolithography (SLA), Selective Laser Sintering (SLS), Fused Deposition Modelling (FDM), Inkjet Modeling (IJM), Direct Metal Deposition (DMD). These techniques differ in the way they build layers and the types of materials that can be used [1]. Since its inception in 1984 with SLA [2], AM has grown from being used for manufacturing parts with highly complex geometries used by companies during design stages of a product life cycle to the availability of consumer-level 3D printers. Thus, AM techniques are more available than before, resulting in a highly commercial and competitive sector [3]. One of these polymer-based AM techniques is Fused Deposition Modelling (FDM) [2] which involves the extrusion of the raw and support material through a heated nozzle as seen in Figure 1. The quality and material properties of FDM parts are affected by various in-process parameters such as layer thickness, build orientation, print speed, extrusion temperature, infill density, infill pattern [7]. The decrease in quality of FDM manufactured parts can also be a result from properties of the raw materials like color [37, 38] and environmental factors like humidity and temperature [38]. It is also dependent on the accuracy and precision of slicer software that converts the CAD model to an .STL file and the .STL file to g-code instructions which is the tool path followed by the nozzle. As a result, FDM manufactured parts often exhibit non-homogeneous, anisotropic and nonlinear behavior [8]. This leads to the products which may have quality issues including dimensional errors, layer delamination, undesirable porosity and poor or indeterminate material properties [9]. Therefore, there exists a need to simulate the process during and after deposition [3] to be able to predict the material properties of the finished product.
3.2.3 Background

FDM involves a moving heat source since the heated filament is extruded the nozzle which follows the tool path developed by a slicer algorithm. The bonds formed in the FDM process can be attributed to the thermal energy of semi-molten filament [39] and the resulting polymer diffusion [5]. Convection with the environment and conduction with support structures also affect the heat transfer in the FDM process [25]. The resulting cooling profile of an element is related to the bond strengths formed between elements. Therefore, calculating the temperature history is needed to understand the mechanical properties of the printed parts.

Finite difference methods [40, 41] and finite elemental analysis [24, 28, 42] have been used to determine the temperature history of the printed parts. Analytical solutions [22, 25, 30] have been developed where the transient heat model includes activating or deactivating all relevant local boundary conditions depending on part geometry, operating conditions and deposition strategy. The discrete element method (DEM) is used for modelling phenomena in which large numbers of discrete particles are in contact with each other and each particle with a single-node element that
has a rigid shape [43]. DEM has been used in modelling powder-based additive manufacturing techniques [44].

3.2.4 Methodology

In DEM approach, a discrete element (often) of spherical shape is used to represent the extruded filament. The use of a spherical shape is arbitrary but offers numerical advantages when determining whether adjacent elements are in fact in contact, although spherical elements only experience point contacts. In our formulation, the diameter of this element is equal to the layer height of the printed part. One of the assumptions taken in this model is that the filament width is equal to the layer height. The sequence of deposition of elements is shown in Figure 3. The layer height values range from (0.2 mm to 0.4mm) on most FDM printers, since leads to a Biot number, Bi < 0.1, and hence, each element can be considered as a lumped capacity model with a uniform temperature distribution throughout the element.
There are 3 types of heat transfer modes have to be taken into account to determine the temperature history. They are the conductive heat transfer between elements, conductive heat transfer between the build plate and elements and convective heat transfer between elements and the environment. The print is assumed to have 100% infill, and there are no voids between filaments.

Based on these assumptions, we can define some of the mathematical properties of the elements. The surface area of the element is that of a sphere with a radius, $R$.

$$A = 4\pi R^2$$ (1)

Similarly, the element has a volume of a sphere of radius, $R$. 

Figure 3. Assumed deposition sequence
\[ V = \frac{4}{3} \pi R^3 \]  

(2)

Since the thermal energy stored in an element is a function of the volume, density, heat capacity and the temperature of the element we can write

\[ U_{\text{deposition}} = \rho V C_p (T_{\text{deposition}} - T_{\text{amb}}) \]  

(3)

Here, we make another assumption, that the resulting element has “good” contact with any adjacent elements such that the area in contact is given by

\[ A_{\text{contact}} = \frac{A}{6} \]  

(4)

While a true rigid sphere would have only point contact with its neighbor, this contact approximation is that of a cube in contact with other cubes. Since in reality our sphere is not rigid, but is in fact “mushy”, this assumption is probably closer to the reality, although it is not also imperfect.

In the example given in Figure 3, the left, right, back, top contact areas of element 2 are elements 1, 3, 11, 24 respectively. The front contact area of element 2 is exposed to the atmosphere and its bottom contact area is in contact with the build plate. During printing, the type of heat transfer that a contact area is subjected to changes as new elements are deposited. For instance, in the above example, at the time step when the print is complete, the left, right, back, top contact areas are experiencing conductive heat transfer due to neighboring elements, the front and bottom contact areas are subjected to convective and conductive heat transfer due to the atmosphere and the build plate, respectively. However, at time step when element 2 has just been deposited, the front, right, back, top contacts are exposed to convective heat transfer due to the atmosphere,
whereas left and bottom contacts are facing conductive heat transfer due to element 1 and build plate respectively. The flowchart presented in Figure 4, shows how these changing boundary conditions are handled in the model.

Based on Fourier’s Law, we take the heat flux caused by conductive heat transfer to be

$$\dot{Q}_c = \frac{(T_A - T_B)}{R_{th}}$$

(5)

where the thermal resistance $R_{th}$ is

$$R_{th} = \frac{L}{kA_{contact}}$$

(6)

where $L$ is the characteristic length, and $k$ is the material conductivity. For this model, the characteristic length is given by

$$L = \frac{V}{A_{contact}} = \frac{R}{3}$$

(7)
We can apply similar techniques to model convection and even radiation effects on each of the six faces of the element. Initially, we will assume that only free convection is present, and
that radiative heat transfer is negligible. Thus, we can determine that the internal energy, \( U_i \), of the element at a time \( t_i \), is given by:

\[
U_i = U_{i-1} - \dot{Q}_{\text{conduction}} \Delta t - \dot{Q}_{\text{convection}} \Delta t
\]

and \( \Delta t = t_i - t_{i-1} \)

where, the heat transfer due to convection is given by:

\[
\dot{Q}_{\text{convection}} = h A_{\text{contact}} (T_A - T_{\text{amb}})
\]

and so, the temperature \( T_i \), of the element is given by rearranging Equation (3) as

\[
T_i = \frac{U_i}{\rho C_p V} + T_{\text{ambient}}
\]

where we assume that the deposition time, \( t_i \), of the element is at \( i = 0 \). At deposition, the temperature of the element is that of the extruder, which we refer to as the deposition temperature.

As a result, there are 12 heat transfer cases that need to be considered. Six of the cases involve the build plate and various numbers of faces exposed to the atmosphere or to other elements. The remaining 6 cases involve situations where some faces are in contact with other elements and some faces are in contact with the ambient atmosphere.

The model was initially formulated in MatLAB to evaluate our results against several test cases to determine if the initial assumptions appear to be valid. This formulation would also allow us to use the model to compare to previous experimental data sets generated by [9].
3.2.5 Preliminary Tests

The data files for the various test cases is defined using information in Table 2 and the process parameters and material properties needed for the modelling is given in Table 1.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of extruder (m/s)</td>
<td>0.05</td>
</tr>
<tr>
<td>Layer height (m)</td>
<td>0.0004</td>
</tr>
<tr>
<td>Extruder temperature (K)</td>
<td>493</td>
</tr>
<tr>
<td>Atmospheric temperature (K)</td>
<td>298</td>
</tr>
<tr>
<td>Specific heat capacity of PLA (J/kg·K)</td>
<td>1800</td>
</tr>
<tr>
<td>Density of PLA filament (kg/m3)</td>
<td>1240</td>
</tr>
<tr>
<td>Built plate temperature (K)</td>
<td>333</td>
</tr>
<tr>
<td>Rate of cooling of build plate (K/s)</td>
<td>0.08334</td>
</tr>
<tr>
<td>Thermal conductivity of PLA (W/m·K)</td>
<td>0.13</td>
</tr>
<tr>
<td>Thermal conductivity of Build plate Glass (W/m·K)</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 1. Process parameters and material properties.

The temperature histories for the various test cases are seen to be similar throughout the simulation as shown in Figures 5 and 6. In Figure 4b, the temperature value of the element of that just is deposited is much higher than the first element deposited which mimics reality. The multiple drops in temperature in each layer is seen at the first and last elements of each filament row. These “edge elements” have a higher cooling rate due to their exposure to the ambient environment at a lower temperature and therefore is cooling faster. The elements adjacent to the “edge elements” are
may also be affected by them and cool down as result but cool at a slower rate than the edge elements. This effect is propagated throughout the filament leading to the elements in the middle being at the highest temperature.

<table>
<thead>
<tr>
<th>Set</th>
<th>Layers</th>
<th>Rows</th>
<th>Column</th>
<th>Total</th>
<th>Ele-</th>
<th>Part Size (LxWxH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>25</td>
<td>10</td>
<td>1500</td>
<td></td>
<td>(4x10x2.4)</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>16</td>
<td>16</td>
<td>1536</td>
<td></td>
<td>(6.4x6.4x2.4)</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>25</td>
<td>2500</td>
<td></td>
<td>(10x4x4)</td>
</tr>
</tbody>
</table>

Table 2. Test cases.
In Fig 5c, we see a reversal of the trend mentioned observed previously where the coldest elements were the oldest elements. At this point, the oldest elements are seeing their cooling profiles slowed down by the heated build plate, and the newest elements are continuing to cool due to the exposure to the ambient environment. As the build plate is allowed to cool, the temperature difference between the layers reduces until it is negligible as can be seen in Figures 6b and 6c.
3.2.6 Conclusions & Future Work

These early results provide evidence to support the validation of the performance of the model. In reviewing the datasets, we observe several trends noted in both the literature and through experimental observation of actual FDM prints.

The first trend that can be observed is that the corner elements of the print tend to cool the fastest, followed by the edges and finally by the internal elements. Simply put, the print cools from
the inside out, and in cases where the build plate is heated, from the top to the bottom. This is entirely consistent with the behavior of most FDM prints.

Second, we observe rapid cooling of elements with exposed edges. In Figure 7, we have highlighted the behavior of two different layers printed with different row lengths. The odd layers are printed with long rows of 25 elements, while the even rows are only 10 elements long. The corresponding peaks observed are correlated to the ends of the rows that have the longest durations between passes of the extruder. When the extruder has to print 50 elements before returning to an element, we observe an acceleration in the cooling of those edge elements, as compared to elements that are interior. And this effect is most significant at the corner elements.

![Figure 7: Edge Cooling Effects enlarged from Figure 4b.](image-url)
We can also see the behaviors of the elements more easily when we examine the behavior of selected elements within a layer. Figure 8 shows the cooling profiles of elements from layer 6 of dataset 1.

Figure 8: Individual Element Behaviors.

In this layer, elements are laid down top to bottom and left to right. From the point of deposition, the general behavior of the element is to cool towards the build plate temperature. We
see consistent behavior in each row (column since the elements are laid down vertically in an even layer). The top row of the print even shows a slight disruption due to the neighboring element being deposited shortly after the deposition of the initial element. Similarly consistent behavior is observed in other layers and datasets.

In addition, as noted in the discussion of Figures 5 and 6, we observe accelerated cooling of elements located on the edges of the print, and we see the thermal profile of the print “flip” from the coolest elements being the first elements deposited, to the coolest elements being the last elements deposited due to the effects of reheating from the build plate. These effects are commonly observed in FDM prints. Therefore we observe at least a qualitative evaluation of the DEM model.

Considerable work remains to be completed with respect to the model. Computationally, there are numerous efficiencies that need to be implemented in order to improve performance. The next step in model validation will be to validate the observed cooling rates. This is a much more complex problem than might be initially believed due to the difficulty of obtaining precise measurements of the individual elements. While some experimental data exists, that data is also dependent upon the exact build parameters including the path described by the g-code. To that end, an essential improvement in the model is an interface that allows the path to be determined from g-code, as opposed to be a fixed parameter as is the current case. This improvement is already underway with a current student. Once complete, the cooling rate can be validated to provide a quantitative validation of the model.

Conceptually, since the initial goal is to use the thermal behavior of the extruded material to predict the effective bond strength between within filaments, between filaments within a layer, and between filaments between layers. The data produced with this model will allow for an estimation of the times for bond formation for elements in each of these cases. This time period for bond formation should be a factor in the bond strength between the elements. However, other
factors such as the surface contact conditions and the evolution of those conditions may also be
important. The results of this simulation model should significantly inform the investigation into
these factors.

In addition, DEM models allow for the subsequent incorporation of models to account for
mechanical effects, such as slump, mechanical loads due to cooling and contraction, and ultimately
an analysis of the part using as printed properties in actual design use cases.

Ultimately, simulation of the build process of additively manufactured processes will fa-
cilitate in-situ defect detection, optimal print configurations for design use cases of the mechanical
parts and improved computational support for computer-aided design for additive manufacturing.
By modeling the multiphysics associated with the FDM process, critical insights can be identified
and their significance characterized allowing for advances in the development and use of the tech-
nology with greater effectiveness and reliability. Finally, the DEM models described here can also
be extended to account for the effects present in other types of additively manufactured processes
using different printing processes.

Acknowledgements

The authors would like to acknowledge the support of the Department of Mechanical En-
gineering at Clemson University for their support of this work. All statements within are those of
the authors and may or may not represent the views of these institutions.

3.3 Next steps

The preliminary tests show that the model exhibits behaviors that are seen in FDM pro-
duced parts. However, additional testcases were run to confirm that the exhibited behaviors are
seen in other conditions. Using the temperature history, time available for bond formation is esti-
mated and a parametric study of printing conditions is conducted whose results are discussed in the
next chapter.
CHAPTER 4: ADDITIONAL TESTS, BOND TIME AND PARAMETRIC STUDY OF PRINTING CONDITIONS

4.1 Flow of MatLAB code

The information about the order of deposition of elements for each testcase is created using function creating_data_file.m that takes the number of rows, columns and layers of each testcase corresponding to the width along the y-axis, length along x-axis and height along z-axis, respectively. This data file is created under the assumption that the infill percentage is 100%. Also, shells that are usually deposited around the perimeter is not printed. The neighboring contacts of each element is calculated using above mentioned data file and function searching_functn_3D_file.m. The temperature values of all elements during printing and after printing when the build plate is cooled to atmospheric temperature at a rate of 5K/min. is calculated using the time_history3D_ver2.m file. The time that an element and its contact remains above glass transition $T_g$ is calculated is also calculated using bond_time_above_Tg.m function. When two contacts have different different bond times, the one with the lesser time is selected as the bonding time available between those elements. All these MatLAB codes are presented in Appendix A.

4.2 Printing Conditions and Geometry

The model was applied to testcases of various sizes to determine the temperature history trends and predict the possible bond strength trends. The material considered in the testcases was PLA whose material properties and process parameters are based on the printing conditions given in Table 1 in Section 3.
The geometry of the various testcases used and time it takes to print the testcases is given in Table 3.

<table>
<thead>
<tr>
<th>Testcase</th>
<th>Layers</th>
<th>Rows</th>
<th>Columns</th>
<th>Total No. of Elements</th>
<th>Part Size (LxWxH) (mm)</th>
<th>Time it takes to print 1 layer (seconds)</th>
<th>Time it takes to print testcase (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>25</td>
<td>10</td>
<td>1500</td>
<td>(4 x 10 x 2.4)</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>16</td>
<td>16</td>
<td>1536</td>
<td>(6.4 x 6.4 x 2.4)</td>
<td>2.048</td>
<td>12.288</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>25</td>
<td>2500</td>
<td>(10 x 4 x 4)</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>1440</td>
<td>(4 x 4.8 x 4.8)</td>
<td>0.96</td>
<td>11.52</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>35</td>
<td>16</td>
<td>3360</td>
<td>(6.4 x 14 x 2.4)</td>
<td>4.48</td>
<td>26.88</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>4</td>
<td>20</td>
<td>1600</td>
<td>(8 x 1.6 x 8)</td>
<td>0.64</td>
<td>12.8</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>10</td>
<td>20</td>
<td>4000</td>
<td>(8 x 4 x 8)</td>
<td>1.6</td>
<td>32</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>216</td>
<td>(2.4 x 2.4 x 2.4)</td>
<td>0.288</td>
<td>1.728</td>
</tr>
</tbody>
</table>

Table 3: Geometry of testcases

A parametric study to understand the effect of printing conditions on the rate of cooling for testcase 2 was also completed as shown in Table 4. The printing conditions that were varied included the speed of extruder, the layer height, the extruder temperature, the build plate temperature and the atmospheric temperature. The results are presented and discussed in Section 4.3.
4.3 Results and Discussion

4.3.1 Temperature Profile

The temperature profiles of testcases 1, 2, and 3 were previously discussed in Chapter 3. For testcases 4, 6, and 7, shown in Figures 9-12 demonstrate similar trends to testcases 1, 2, and 3. Here, we see in the odd layers, the elements at the end of each filament deposited lengthwise are at a lower temperature than the middle elements. This is because the length is much greater than the width in these testcases. As the extruder turns around the corner and prints the next filament, the first element of the previous filament is now at a much lower temperature. This causes the newly deposited element next to it to drop to a lower temperature as well. Hence, we see some of the end elements cooling at a higher rate than others. This difference however decreases over time.

During even layer deposition, since the width is smaller, each filament is deposited quickly. This results in the last element of a newly deposited filament to come in contact with the first element of the previous filament. The element in the previous filament was not able to cool quickly,
leading to a smaller temperature drop for the end elements.

Once deposition is complete, the final layer elements are at a higher temperature and the elements at the bottom are the coolest. This is because the bottom layer elements have had the longest time to cool since they were deposited first. This has also been observed in [28]. The final layer element, however, quickly reach temperatures similar to the elements at the bottom. This is due to more surfaces being in contact with the atmosphere and causing the elements at the top to have a higher cooling rate. Such a trend has also been observed in [45]. In Figure 10, we see a reversal where the elements in the final layer are at a lower temperature than the elements in the first layer. The build plate that is at a higher temperature is now affecting the bottom layer elements to a greater extent causing it to be at higher temperature and have a lower cooling rate. This same trend was seen for all testcases.

Figure 9: Temperature history of testcase 4
Figure 10: Temperature history of testcase 4...continued

Figure 11: Temperature history of testcase 6
Practically, the filaments in the middle of a layer take the longest time to cool. However, when we look at Figure 13, testcase 5 does not show this, instead the middle elements of the filaments deposited first are at the highest temperature, followed by the middle elements of the final filaments and then by the elements at the center of the layer. This could be due to the elements not cooling fast enough because here the width is larger than the length as supposed to the previous case where the odd layer more significant dimension. As time passes, the elements along the edges cool faster as it subjected to cooling from the atmosphere overcoming the thermal accumulation and then follows the cooling trends exhibited by testcases 4, 6, and 7 as seen in Figure 14 where the elements in the final layer are at lower temperature than elements in the first layer.
In Figure 15 we see that although length, width and height is equal the same cooling trends are not seen in each layer. This could be because testcase 8 has the least number of elements and the therefore the time it takes to deposit elements is not enough to see the real thermal effects.

Figure 13: Temperature history of testcase 5

Figure 14: Temperature history of testcase 5...continued
The temperature pattern of elements along a filament repeating within a layer seen in testcases 1-7 was also observed in the experimental results in [36]. The experimental values were calculated using IR sensors, an optical camera, and thermocouples. The same repeating temperature pattern was observed in all layers compared to the trends seen in this study that show similarity in alternate layers only. This could be because in [36], the filaments in every layer were deposited from left to right only as supposed to the orthogonal parallel filling along x-axis in odd layers and along y-axis in even layers in this study.

In Figure 16 and 17, we look at the element at the core (middle layer middle row middle element) and corner (end) elements of testcase 2. The trends show that the core element cools the slowest followed by the corner element at the top and then the corner elements at the bottom because the heated plate has an effect on the elements at the bottom for longer. As previously stated, end elements of a filament cool faster than middle elements, therefore we have demonstrated that corner elements cool the fastest, followed by the elements at the edges and then the elements in the
core of the testcase. This trend is seen in practice and also in previously proposed models in [24, 28, 45].
4.3.2 Bonding time

Anisotropic properties observed in FDM produced parts is because the bonds along the x-axis, y-axis and z-axis do not have the same strength. Bond strength increases over time in polymers and bond formation only occurs when the polymer is above glass transition \( T_g \). In this subsection, we look at each contact that can form a bond with an element and the average time per layer the bond is above \( T_g \). In Figures 18-23, we observe that bonds along x-axis and y-axis of the each testcase alternates with x-axis bond being above \( T_g \) for longer time in odd layers and y-axis bonds being above \( T_g \) for longer in the even layers. This is because of the difference between the length and width dimensions of testcases 1, 3, 4, 5, 6, and 7. This affects the cooling within a layer because in odd layers the filaments are deposited along x-axis and in even layer the filaments are deposited along y-axis. This alternating pattern is seen more prominently in Figures 22 and 23 here the number of layers is higher and the time it takes to reach below \( T_g \) increases.
Figure 16: Average Bonding time for Testcase 1

Figure 17: Average Bonding time for Testcase 3
Figure 18: Average Bonding time for Testcase 4

Figure 19: Average Bonding time for Testcase 5
Figures 24 and 25 show the results of testcases 2 and 8, where the x-axis and y-axis bonds have the same average time value above $T_g$. This could be due to the width and length dimensions being the same and causing the average above $T_g$ within a layer to be the same.
In Figures 19, 20, 22, and 23, it can be observed that the bond time available is highest for all elements in the middle layers for testcases 3, 4, 6, and 7 which follows the notion that the core of a testcase cools at slower rate followed by the bonds in the layer closest to the build plate. The bonds in the topmost layer cool the fastest due to being in contact with the atmosphere. However,
as the number of layers decrease, testcases 1, 2, 5, and 8, the effect of the heated build plate on the cooling of the elements in first layers increase causing them to above the Tg for the longest followed by middle layers and them the final layer.

Practically the strength of bonds within a layer is usually higher than bonds across layers. However, we see x-axis and y-axis bonds having lower times above Tg than z-axis bonds. This could be attributed to the assumption used in the model that the time it takes to deposit every element is the same. So if we go from last element in the previous layer to the first element in the next layer, it will take the same time to go from one element to the next within a layer. This results in bond creation at a earlier time than practically possible because it takes times for the extruder to move to a new position for the next layer. This assumption could be overcome by setting up the model to take the contact information and time at which the contact is first made from G-code.

The top contact of an element in one layer becomes the bottom contact of an element in the layer directly above it. This is observed in all the testcases, i.e the bottom bond of next layer has the same time available for bond formation as the top bond of the previous layer.

4.3.3 Parametric study of printing conditions

The rate cooling for the different printing parameters is plotted for elements at the start, middle and end of first, middle and last filament of each layer of testcase 2. The cooling rate for each parameter is calculated at 30 seconds after all elements have been deposited, 1 minute after all elements have been deposited and 3 minutes after all elements have been deposited.
4.3.3.1 Speed of extruder

Changing the speed of the extruder shows that higher the speed, higher is the rate of cooling for the elements in the earlier layers. In contrast, the elements in the upper layers are not affected by it drastically as seen in Figure 25. The time it takes to extrude the entire test case increases with a decrease in extruder speeds. This leads to the lower layers being at a higher temperature for longer and thus resulting in a lower rate of cooling as compared to the elements in the last filament of the final layer. The rate of cooling the elements is almost the same because these elements are subjected to cooling from the atmosphere.

![Rate of cooling for different speeds](image-url)

Figure 26: Rate of cooling for different speeds after 30 seconds
Comparing Figures 26, 27 and 28, it can be seen that, over time the difference between the rate of cooling decreases as well.
4.3.3.2 Layer height

The rate of cooling of the elements increases with a decrease in layer height seen in Figure 29. As the layer height decreases, the dimensions of the testcase decreases resulting in time it takes to print the testcase to lower as well. This facilitates the elements in the first layers of smaller layer heights to cool faster than elements of higher layer height.

Figure 29: Rate of cooling for different layer heights after 30 seconds

Figure 30: Rate of cooling for different layer heights after 1 minute
Comparing Figures 29, 30, and 31, we can observe that again the difference between rates of cooling decreases as time increase.

![Graph showing the rate of cooling for different layer heights after 3 minutes.](image)

**Figure 23**: Rate of cooling for different layer heights after 3 minutes

4.3.3.3 Build Plate temperature

As the build plate temperature increases, the rate of cooling decreases as seen in Figure 32. The effect of the build temperature is transmitted equally to all the elements in this testcase. The build plate temperature on all the layers is due to the limited number of elements and small dimensions of the testcase. Here, again we see there is a decrease between the rates of cooling as time passes when comparing Figures 32, 33 and 34.
Figure 24: Rate of cooling for different build plate temperature after 30 seconds

Figure 25: Rate of cooling for different build plate temperature after 1 minute
4.3.3.4 Atmospheric or chamber temperature

The change in temperature inside the chamber does not significantly effect on the rate of cooling in the beginning, as seen in Figure 35. However, as time passes, this parameter becomes more prominent with a higher chamber temperature leading to a lower cooling rate seen in Figures 36 and 37. The difference does not seem to be significant and could potentially increase with an increase in temperature. In this case, the build plate temperature for PLA was taken as 333K so increasing the chamber temperature would involving increasing the plate temperature as well.
From the observations made using the temperature profile, bond times and parametric study of printing conditions, we can say that the cooling of the element depends on the location of the element within the specimen, the geometry of the overall specimen, the time of deposition for each element and the nature of the boundary condition and the change of its values over time along with the printing conditions.

Figure 27: Rate of cooling for different chamber temperatures after 30 seconds
Figure 28: Rate of cooling for different chamber temperatures after 1 minute

Figure 29: Rate of cooling for different chamber temperatures after 3 minutes
5.1 Conclusions

The thermal model for FDM process based on using discrete elements was studied. Discrete element methodology is based on interactions between discontinuous elements. Since the FDM process involves separate filaments that are deposited next to each to form bonds to create the final component, the nature of the process is discontinuous, leading to the anisotropic properties observed in FDM produced parts. Hence, discrete elements have been considered in this study to represent FDM process.

In this model, a spherical shape is used for simplicity to determine element contact and the resulting heat transfer coefficient and boundary conditions. The sphere's diameter is taken as the layer height and temperature distribution across the cross-section of each element is considered constant, support structures are neglected, and infill percentage was taken as 100% assuming no voids are present in the structure. The contact area is approximated as a cube since true spheres have only point contact with neighboring spheres while the FDM filament has an area contact with its neighbors. Each element has six contact areas that are treated as constant and are subjected to convective heat transfer from the atmosphere, conductive heat transfer from the other elements, and the build plate.

This model was applied to testcases with cuboidal geometry of varying length, height and width. The temperature evolution of elements with varying printing conditions revealed the conclusions below.

a) The temperature profile plots show that during deposition and for a short time
after deposition the element in the final layer cools slower than elements in the first layer. However, due to the heated build plate being in contact with the elements in the first layer and the atmosphere in contact with the elements in the final layer, the cooling rates flip, with the element from the top to the bottom cooling fastest. The cooling rate of elements along the edges is higher than elements in the interior. Interior elements cool the slowest and the elements at the corner cooling the fastest among all edge elements. FDM prints cool outside to inside.

b) The average time elements within a layer stay above $T_g$ plots show that the middle layers elements stay above $T_g$ for the longest time followed by the elements near the build plate and then by elements in the final layer. However, testcases where the number of elements per layer was relatively less than the other testcases the heated build plate affected the elements near it for a longer time, causing those elements to be above $T_g$ longer.

c) The elements along the filament within a layer stay above the $T_g$ for a longer time than elements across filaments within the same layer. This matches the behavior seen in most FDM parts. However, contacts above and below the element were observed to remain above $T_g$ for a longer time than contacts with a layer. This does not match the behavior observed in most FDM parts where the bond is strongest within a layer followed by across the layers. This may be because the model assumes that the element of the next layer is printed instantly after the last element of the previous layer which is not in agreement with standard FDM printing practices.

d) The rate of element cooling after the entire testcase is printed with respect to a fixed time showed that lowering the speed of extrusion leads to a decrease in the cooling rate and decreasing the layer height leads to increase in the cooling rate. Increasing the
build plate leads to decrease in the rate of cooling because the reheating caused by the build temperature will affect the elements for longer time. Increasing the build chamber / atmospheric temperature by a small increment did not change the rate of cooling significantly. This is consistent with observations made when printing FDM components.

5.2 Future Work

a) Accounting for unequal contact areas

In the model, the element is considered to be a sphere with filament width and height being equal and the area of contact is assumed to be constant. In practice, however, when the filament is deposited the width and height is not the same and the element shape should be changed to represent that. The new elemental cross-section could be a rectangle with semi-circles along the width. The assumption that the contact areas for the boundary conditions in each direction are equal would have to be changed to match the new cross-sectional area and the heat transfer co-efficients would also have to be reformulated. As more layers are added, the layers at the bottom slump due to the increasing weight above it, causing the contact areas to change over time.

b) Better representation of the toolpath

The model uses simple data files representing the sequence of deposition of elements for a cuboidal geometry with small dimensions and one type of printing pattern. The model should be applied to real-life specimens and should create data files that will match the printing process of the specimen. In practice, when an object that is to be printed is sliced, it generates G-code that are instructions to the 3D printer about the location of the extruder and its speed, direction and time of deposition of the filament. All of this is necessary when the data file representing the sequence is created and during the thermal calculations when
the time step between elements within a layer and across layers has to be decided. Also, the speed of extrusion, which is taken as constant as throughout the model should also be changed to values obtained from the G-code. Therefore, it is necessary to incorporate the G-code instructions into the model to better represent the FDM produced parts.

c) Improving the code

The code can handle only a small number of elements less elements and uses the Palmetto Cluster at Clemson for computing the temperature values. Rewriting the code to more efficient is necessary because as the number of elements in a testcase increases the time it takes to calculates for each time step will also increase and the total computational time will increase exponentially. The current model calculates the values for the next time step using the previous time step values, so instead of calculating the temperature values of all elements at the same time steps sequentially, parallel programming could potentially be used to decrease the computational time. Each instruction would calculate the temperatures of all elements within each layer simultaneously. The total number of instructions running parallelly at any point of time will be equal to the number of layers for each time step.

d) Verification of model against experiments

The experimental setup of Infrared sensors or cameras used in [29, 35, 36] could be utilized to find the temperature values of the specimen during and after printing. Once the performance of the code has been improved where it take on specimens with large number of elements and G-code has been incorporated to replicate the printing process, the temperature values obtained from the model should be verified against the experimental values and the accuracy of the model should be explored.

e) Adding support structures, void density and radiation
The model assumes that the infill density is 100% and the model is applied to testcases that did not require support structures. In practice, one of FDM features is we can change the void density to save material when we are producing parts that will not be subjected to high stresses and therefore do not require high part strength. Incorporating the G-code could allow us to make better assumptions about the void density and support structures. These parameters could be added to the model to see how they affect the cooling of the elements. Once, void density has been added to the model, the assumption that radiation does not have an affect on the heat transfer can be verified.

f) Incorporating sintering and bond formation model

In FDM, the polymer bond formation results in an overlap between filaments next to each other. The longer time an element is above $T_g$ the stronger the bond becomes and more volume of the filaments next to each other gets overlapped. This leads to change in the contact area for the boundary conditions over time, leading to change in temperature values and cooling of the element that again affects the time above $T_g$. Therefore, there is a need to couple the thermal and sintering model of bond formation as they inform each other and calculate the bond strength. A similar approach has been used in [31] and could be incorporated to this model.

5.3 Summary

The tasks completed in this study is summarized below,

1. Data files were created for testcases of various dimensions which included the sequence of deposition of elements.
2. The neighboring contacts of each element was identified and the nature and value of these boundary conditions was tracked over time.
3. The temperature evolution of elements were calculated and certain trends were observed.

4. The average time each element bond that stays above $T_g$ within a layer was calculated to make observations about possible bond formation time.

5. A parametric study where the various printing conditions were changed and its effect on the cooling rate was observed.

These early results provide evidence to support the validation of the performance of the model. In reviewing the datasets, we observe several trends noted in both the literature and through experimental observation of actual FDM prints.

Recalling the design statement, “A DEM model can simulate the thermal behavior of an FDM part during deposition and cooling”.

From the research, we conclude that DEM can be used to simulate the thermal behavior of an FDM part during deposition and cooling. The study has been qualitatively validated because the cooling trends observed by the model follows the trends seen in FDM produced parts. However, the accuracy of the model has not been verified against experimental values. This is because, at its current state, the model uses simple geometry and printing pattern, idealized time of deposition; as a result the model values may not be able to match with the values obtained to experiments. G-code instructions have to included when creating the data files as it gives the exact build parameters and then the model values should be verified against experimental values for its accuracy. Therefore, further quantitative research has to be done to show that simulations of thermal modelling of FDM process using discrete elements.
REFERENCES

References


APPENDICES

Appendix A- MatLAB code

A1. creating_data_file.m

```matlab
function [data] = creating_data_file(rows,columns,layers)
% Summary of this function goes here
% creating a data file(excel/.mat) with user defined size constraints
% rows x columns x layers : info needed from user
% odd layers will have elements deposited left to right and turn
% around at end from right to left
% even layers will have elements deposited from top to bottom and turn
% around at end from bottom to top
% therefore, data file will give the sequence of deposition of elements

    data= zeros(rows,columns,layers);
    i=1;
    for no_layer=1:1:layers
        odd_layer_row=1; even_layer_col=1;
        if rem(no_layer,2)~=0 % odd layers
            for j=1:1:rows      %row
                if rem(odd_layer_row,2)~=0  %odd rows
                    for k=1:1:columns    %colm
                        data(j,k,no_layer) =i;
                        i=i+1;
                    end
                    odd_layer_row= odd_layer_row+1;
                end
            else % even rows
                for k=columns:-1:1    %colm
                    data(j,k,no_layer) =i;
                    i=i+1;
                end
                odd_layer_row= odd_layer_row+1;
            end
        end
        else % even layers rem(no_layer,2)==0
            for k=1:1:columns      %columns
                if rem(even_layer_col,2)~=0 %odd columns
                    for j=1:1:rows    %colm
                        data(j,k,no_layer) =i;
                        i=i+1;
                    end
                    even_layer_col= even_layer_col+1;
                end
            else % even columns
                for j=rows:-1:1 %colm
                    data(j,k,no_layer) =i;
                    i=i+1;
                end
                even_layer_col= even_layer_col+1;
            end
        end
    end

end
```

A2. searching_functn_3D_file.m

```matlab
function [mol] = searching_functn_3D_file(dataIR,row,colm,layer)
%this function takes in your dataIR set and the size of the dataIR set through
%input variables dataIR,row,colm respectively
%it gives out a variable (mol) which is a matrix where rows are no. of ele-
%ments in present in dataIR set
% and columns is 7
% first column- current element
% second column- left contact of current element
% third column- right contact of current element
% fourth column- front contact of current element
% fifth column- back contact of current element
% sixth column- top contact of current element
% seventh column- bottom contact of current element
% eighth column- tells which layer the element is

mol=ones((row*colm*layer),8);
col=colm;
c_=1;
counter=1;
counter_layer=1;
no_of_ele_in_layer= row*colm;
layer_no=0;
% the ones in the (mol) matrix are replaced with the element number values
% of the contact
% the function reads from left to right for the elements in the first row of
% the dataIR set
% and at the end of the row it reads the element below it and then proceeds to read
% from right to left,
% visually ->>>>>
% <<<<<<<
% >>>>>>

ele=1;
for i=1:1:(layer)
    for k=1:1:(row)
        for j=1:1:(colm)
            mol(ele,1)=dataIR(k,j,i);
            if j-1 <=0
                mol(ele,2)= 0 ;%left contact
            elseif dataIR(k,j-1,i)==0
```
mol(ele,2) = 0 ; % left contact
else
  mol(ele,2) = dataIR(k, j-1, i) ; % left contact
end

if j+1 > colm
  mol(ele,3) = 0 ; % right contact
elsif dataIR(k, j+1, i) == 0
  mol(ele,3) = 0 ; % right contact
else
  mol(ele,3) = dataIR(k, j+1, i) ; % right contact
end

if k-1 <= 0
  mol(ele,4) = 0 ; % front contact
elsif dataIR(k-1, j, i) == 0
  mol(ele,4) = 0 ; % front contact
else
  mol(ele,4) = dataIR(k-1, j, i) ; % front contact
end

if k+1 > row
  mol(ele,5) = 0 ; % back contact
elsif dataIR(k+1, j, i) == 0
  mol(ele,5) = 0 ; % back contact
else
  mol(ele,5) = dataIR(k+1, j, i) ; % back contact
end

if i-1 <= 0
  mol(ele,7) = -1 ; % bottom contact
elsif dataIR(k, j, i-1) == 0
  mol(ele,7) = 0 ; % bottom contact
else
  mol(ele,7) = dataIR(k, j, i-1) ; % bottom contact
end

if i+1 > layer
  mol(ele,6) = 0 ; % top contact
elsif dataIR(k, j, i+1) == 0
  mol(ele,6) = 0 ; % top contact
else
  mol(ele,6) = dataIR(k, j, i+1) ; % top contact
end

mol(ele,8) = i;

ele = ele + 1;
end
end

dataIR
function [ele_time_above_Tg] = bond_time_above_Tg(display,ij,ftg,time_pos,mol,Tg)
%UNTITLED3 Summary of this function goes here
% Detailed explanation goes here

ele_time_above_Tg= zeros((ij-1),20);
i=1;
for g=2:ij
    ele_time_above_Tg(i,1)=i;
    for j=1:ftg %each time step
        if display(j,g)>=Tg
            ele_time_above_Tg(i,2)=display(j,1);% for each element the time it takes to reach Tg from 0 seconds,
            % (2nd column)
        end
    end
    ele_time_above_Tg(i,3)=ele_time_above_Tg(i,2)-(time_pos*(i-1));% for each element the absolute time it takes to reach Tg (from its respective time of creation),
    % (3rd column)
i=i+1;
end

for g=1:(ij-1) % time it takes for each bond to reach Tg (from 0 seconds), will be the element in the bond that takes the longest time
    % (5th-10th column)
    ele_time_above_Tg(g,4)= 5555;
    k=2;
    for j=5:10
        if mol(g,k)==0 || mol(g,k)==-1
            ele_time_above_Tg(g,j)=0;
        elseif ele_time_above_Tg(g,3) < ele_time_above_Tg(mol(g,k),3)
            ele_time_above_Tg(g,j)= ele_time_above_Tg(g,3)- (time_pos*(mol(g,k)));% ele_time_above_Tg(g,j)= ele_time_above_Tg(g,3);
        else
            ele_time_above_Tg(g,j)=ele_time_above_Tg(mol(g,k),3)-(time_pos*3);% ele_time_above_Tg(g,j)=ele_time_above_Tg(mol(g,k),3);
        end
        k=k+1;
    end
end
function [display, disp_post, Uo, m, i] = time_history3D_ver2(time, mol, time_step, time_for_cooling, Uo, i, m, disp_post, disp_postns, display, layer, each_row, layer_height, velocity, ti_cond, t_plate, t_atomsphere)

% IMPORTANT: first column in display matrix is time step, therefore have to add one in col position in disp(row, col)
% pla properties
v=velocity;%m/s velocity of extruder/ max print speed 50 mm/sec
k=0.13;% thermal conductivity W/m·K
k_plate=0.8;% thermal conductivity of build plate tempered glass W/m·K
keq=2*k*k_plate/(k+k_plate);
r=layer_height/2;% radius of sphere
time_pos=(2*r)/(v); % time for extruding one sphere along the length
A=4*pi*(r^2); % area of sphere which is divided into 6 faces of cube
lc=r/3; % critical length
h=10; % forced convection heat transfer coefficient for low speed air over a surface W/K m^2
bno=(h*lc)/k;% biot number
% can be used as lumped parameter limited by heat going out
% time t d density cp specific heat
d=1.24*10^3;%kg/m3 1.24 g/cc density of filament;
cp=1800;% J/kg·K specific heat capacity of pla
al=k/(d*cp); % thermal diffusivity
fno=k/(d*cp*(lc^2));
% ti_cond=493; %220C extruder temperature
% t_plate=333; %60C Bed temperature
% t_atomsphere=298; %25C

volume= (4*pi*r.^3) /3;

% first element cooling temperatures
if time==0
  t_i=493;
  u0=(ti_COND-t_ATOMSPHERE)*d*cp*volume;
  ui= u0 - (((5*h*A*(t_i-t_ATOMSPHERE))/6)+(2*pi*keq*r*(t_i-t_plate))
     *(time-((time-time_step)))) ;
  change_u= ui-u0;
  % change_u~ cp (change in T) and change of u is negative here
  t_i=t_i +(( change_u)/(cp*d*volume));
  display(disp_post,1)=time;
  display(disp_post,2)=t_i;

end

A4. time_history3D_ver2.m
u0=\( (t_i - t_{\text{atmosphere}}) \times d \times cp \times \text{volume} \);
Uo(disp_post,1)=time;
Uo(disp_post,2)=u0;

disp_post=disp_post+1;
end

if time>0 && time< time_pos
a=1;
t_i= \text{display}(\text{disp_post-1,a+1});
u0= Uo(disp_post-1,a+1);
u_i= u0 - (((5*h*A*(t_i-t_{\text{atmosphere}})/6)+(2*pi*keq*r*(t_i-t_{\text{plate}}))) \times (time-time_{\text{step}})))
change_u= u_i-u0;
% change_u~ cp (change in T) and change of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
display(disp_post,1)=time;
display(disp_post,a+1)=t_i;
Ui(disp_post,1)=time;
Ui(disp_post,a+1)=ui;
u0=(t_i-t_{\text{atmosphere}})*d*cp*volume;
Uo(disp_post,1)=time;
Uo(disp_post,a+1)=u0;

disp_post=disp_post+1;
end

% time

%temperature values from 2nd element onwards
if time>=time_pos
if i<(display_postns)
    display_postns;
    time
    if (time-i*time_pos) >= 3.3333e-05
        iz=1;
        m=m+1;
        i=i+1;
        if i=(\text{rem}(iz,(\text{row*colm})))== i
    %
    %
    %layer_check=layer_check+1 ;
    %
    %
    end
end

if (i+2)<display_postns
t_inext=t_i_{\text{cond}};
u0=(t_inext-t_{\text{atmosphere}})*d*cp*volume;
display(disp_post,i+2)=t_inext;
Uo(disp_post,1)=time;
Uo(disp_post,1+i2)=u0;
end
for n=0:1:m
if (i-n)<=i
    z=i-n;
    if mol(z,2)<=i
        left=mol(z,2);
        if display(disp_post-1,left+1)==0
            t_left=ti_cond;
        else
            t_left=display(disp_post-1,left+1);
        end
    else
        left=0;
    end
    if mol(z,3)<=i
        right=mol(z,3);
        if display(disp_post-1,right+1)==0
            t_right=ti_cond;
        else
            t_right=display(disp_post-1,right+1);
        end
    else
        right=0;
    end
    if mol(z,4)<=i
        front=mol(z,4);
        if display(disp_post-1,front+1)==0
            t_front=ti_cond;
        else
            t_front=display(disp_post-1,front+1);
        end
    else
        front=0;
    end
    if mol(z,5)<=i
        back=mol(z,5);
        if display(disp_post-1,back+1)==0
            t_back=ti_cond;
        else
            t_back=display(disp_post-1,back+1);
        end
    else
        back=0;
    end
    if mol(z,6)<=i
        top=mol(z,6);
        if display(disp_post-1,top+1)==0
            t_top=ti_cond;
        else
            t_top=display(disp_post-1,top+1);
        end
    else
        top=0;
    end
    if mol(z,7)==-1
        bottom=mol(z,7);
    elseif mol(z,7)<=i
        bottom=mol(z,7);
if display(disp_post-1,bottom+1)==0
t_bottomi=ti_cond;
else
t_bottomi=display(disp_post-1,bottom+1);
end

else
bottom=0;
end

if display(disp_post-1,z+1)==0
t_i=ti_cond;
u0=(t_i-t_atomsphere)*d*cp*volume;
else
t_i= display(disp_post-1,z+1);
u0= Uo(disp_post-1,z+1);

if rem(mol(z,8),2)==0
% time
% z
% even layer & odd layer, first element, first column, top to bottom, no. of contacts=1
if left==0 && front==0 && back==0 && right==0 && bottom ~=-1 && top==0
% z29=z
time
% even layer & odd layer, first element, first strand, top to bottom, no. of contacts=2
elseif left==0 && front==0 && back~=0 && right==0 && bottom ~=-1 && top==0
% z30=z
\[
\begin{align*}
ui &= u0 - ((4*h*A*(t_i-t_atomsphere)/(2*r)) + ((k*A*(t_i-t_backi))/(2*r)) +\ldots \\
&* (time-(time-time_step))); \\
\text{change}_u &= ui-u0; \\
\text{change}_u \sim \text{cp} \text{ (change in T) and change of } u \text{ is negative here} \\
t_i &= t_i + ((\text{change}_u)/(\text{cp}*d*volume)); \\
u0 &= (t_i-t_atomsphere)*d*cp*volume; \\
\end{align*}
\]
change_u = ui-u0;
% change_u ~ cp (change in T) and change
of u is negative here
% even layer, middle element, top to bottom, no. of contacts=3
elseif left==0 && z-front==1 && right==0 &&
bottom ~=1 && top==0 && z-back==1
% % % %
ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_backi))/(2*r)) +
((k*A*(t_i-t_bottomi))/(2*r))) *(time-(time-time_step)));%change_u = ui-u0;
% change_u ~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% % % % % % % % % % % % % % % %
% even layer, middle element, top to bottom, no. of contacts=4
elseif left==0 && z-front==1 && right~=0 &&
bottom ~=1 && top==0 && z-back==1
% % % %
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_backi))/(2*r)) +
((k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_topi))/(2*r)) +(k*A*(t_i-t_bottomi))/(2*r)) *(time-(time-time_step)));%change_u = ui-u0;
% change_u ~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% % % % % % % % % % % % % % % %
% even layer, middle element, first strand, top to bottom, no. of contacts=5
elseif left==0 && z-front==1 && right~=0 &&
bottom ~=1 && top~=0 && z-back==1
% % % %
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_backi))/(2*r)) +
((k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_topi))/(2*r)) +(k*A*(t_i-t_topi))/(2*r)) *(time-(time-time_step)));%change_u = ui-u0;
% change_u ~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% % % % % % % % % % % % % % % %
% even layer, middle element, first strand, top to bottom, no. of contacts=6
elseif left==0 && z-front==1 && right~=0 &&
bottom ~=1 && top~=0 && z-back==1
% % % %
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_backi))/(2*r)) +
((k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_topi))/(2*r)) +(k*A*(t_i-t_topi))/(2*r)) *(time-(time-time_step)));%change_u = ui-u0;
of u is negative here

% change_u~ cp (change in T) and change of u is negative here

t_i = t_i + ((change_u)/(cp*d*volume));
u0 = (t_i - t_atomsphere)*d*cp*volume;

elseif left == 0 && z-front == 1 && right ~= 0 && bottom ~= -1 && top == 0 && back == 0

z38 = z
%
ui = u0 - (((3*h*A*(t_i-t_atomsphere)/(6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_righti))/(2*r)) +... + ((k*A*(t_i-t_bottomi))/(2*r))) * (time-(time-time_step)));
change_u = ui-u0;

% change_u~ cp (change in T) and change of u is negative here

t_i = t_i + ((change_u)/(cp*d*volume));
u0 = (t_i - t_atomsphere)*d*cp*volume;

% even layer, last element, first strand, top to bottom, no. of contacts=3

elseif left == 0 && z-front == 1 && right ~= 0 && bottom ~= -1 && top == 0 && back == 0

z39 = z
%
ui = u0 - (((2*h*A*(t_i-t_atomsphere)/(6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_righti))/(2*r)) +... + ((k*A*(t_i-t_topi))/(2*r)) + ((k*A*(t_i-t_bottomi))/(2*r))) * (time-(time-time_step)));
change_u = ui-u0;

% change_u~ cp (change in T) and change of u is negative here

t_i = t_i + ((change_u)/(cp*d*volume));
u0 = (t_i - t_atomsphere)*d*cp*volume;

% even layer, last element, first strand, top to bottom, no. of contacts=4

elseif left == 0 && z-front == 1 && right ~= 0 && bottom ~= -1 && top == 0 && back == 0

z40 = z
%
ui = u0 - (((4*h*A*(t_i-t_atomsphere)/(6) + ((k*A*(t_i-t_lefti))/(2*r)) +... + ((k*A*(t_i-t_topi))/(2*r)) + ((k*A*(t_i-t_bottomi))/(2*r))) * (time-(time-time_step)));
change_u = ui-u0;
% change_u= cp (change in T) and change of u is negative here

\[ t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right); \]
\[ u_0 = (t_i - t_{\text{atmosphere}}) \times d \times \text{cp} \times \text{volume}; \]

% even layer, first element, middle

strand, bottom to top, no. of contacts=3

\[
\text{elseif left} \neq 0 \ \&\& \ \text{right} = 0 \ \&\& \ z-\text{front} = -1 \ \&\& \ \text{back} = 0 \ \&\& \ \text{bottom} \neq -1 \ \&\& \ \text{top} = 0
\]

\[
\text{time}
\]
\[
z_{41} = z
\]

\[
\text{ui} = u_0 - \left( \frac{3hA(t_i - t_{\text{atmosphere}})}{6} + \frac{(kA(t_i - t_{\text{lefti}}))}{2r} + \frac{(kA(t_i - t_{\text{fronti}}))}{2r} + \frac{(kA(t_i - t_{\text{bottomi}}))}{2r} \right) \times (\text{time} - (\text{time} - \text{time step}));
\]
\[
\text{change}_u = \text{ui} - u_0;
\]
% change_u= cp (change in T) and change of u is negative here

\[ t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right); \]
\[ u_0 = (t_i - t_{\text{atmosphere}}) \times d \times \text{cp} \times \text{volume}; \]

% even layer, first element, middle

strand, bottom to top, no. of contacts=4

\[
\text{elseif left} \neq 0 \ \&\& \ \text{right} \neq 0 \ \&\& \ z-\text{front} = -1 \ \&\& \ \text{back} = 0 \ \&\& \ \text{bottom} \neq -1 \ \&\& \ \text{top} = 0
\]

\[
\text{time}
\]
\[
z_{42} = z
\]

\[
\text{ui} = u_0 - \left( \frac{2hA(t_i - t_{\text{atmosphere}})}{6} + \frac{(kA(t_i - t_{\text{lefti}}))}{2r} + \frac{(kA(t_i - t_{\text{fronti}}))}{2r} + \frac{(kA(t_i - t_{\text{righti}}))}{2r} + \frac{(kA(t_i - t_{\text{bottomi}}))}{2r} \right) \times (\text{time} - (\text{time} - \text{time step}));
\]
\[
\text{change}_u = \text{ui} - u_0;
\]
% change_u= cp (change in T) and change of u is negative here

\[ t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right); \]
\[ u_0 = (t_i - t_{\text{atmosphere}}) \times d \times \text{cp} \times \text{volume}; \]

% even layer, first element, bottom to top, no. of contacts=5

\[
\text{elseif left} \neq 0 \ \&\& \ \text{right} = 0 \ \&\& \ z-\text{front} = -1 \ \&\& \ \text{back} = 0 \ \&\& \ \text{bottom} \neq -1 \ \&\& \ \text{top} = 0
\]

\[
\text{time}
\]
\[
z_{43} = z
\]

\[
\text{ui} = u_0 - \left( \frac{1hA(t_i - t_{\text{atmosphere}})}{6} + \frac{(kA(t_i - t_{\text{lefti}}))}{2r} + \frac{(kA(t_i - t_{\text{fronti}}))}{2r} + \frac{(kA(t_i - t_{\text{righti}}))}{2r} + \frac{(kA(t_i - t_{\text{topi}}))}{2r} + \frac{(kA(t_i - t_{\text{bottomi}}))}{2r} \right) \times (\text{time} - (\text{time} - \text{time step}));
\]
\[
\text{change}_u = \text{ui} - u_0;\]
of $u$ is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{cp \cdot d \cdot \text{volume}} \right);
\]
\[
u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot cp \cdot \text{volume};
\]
% even layer, middle element & last element, middle strand, bottom to top, no. of contacts=3

\[
e = 0 \& \& z\text{-back}=1 \& \& \text{front}=0 \& \& \text{left}=0
\]
% even layer, middle strand, bottom to top, no. of contacts=4

\[
e = 0 \& \& z\text{-back}=1 \& \& \text{front}=-1 \& \& z\text{-bottom}=0 \& \& \text{left}=0
\]
% even layer, middle strand, bottom to top, no. of contacts=5

\[
e = 0 \& \& \text{z-bottom}=0 \& \& \text{left}=0 \& \& \text{right}=0
\]
elseif z-back==1 && z-front==-1 && bottom~=-1 && top~=-1 && left~=-1 && right=-1
%                                      z47=z
time

ui= u0 - (((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_topi))/(2*r))+
((k*A*(t_i-t_backi))/(2*r))+((k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_bottomi))/(2*r)) *(time-(time-time_step));
change_u= ui-u0;
% change_u~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
ui= (t_i-t_atomsphere)*d*cp*volume;

elseif z-back==1 && front==0 && bottom~=-1 && top~=-1 && left~=-1 && right~=-1
%                                      z48=z
time

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_lefti))/(2*r))+
((k*A*(t_i-t_topi))/(2*r))+((k*A*(t_i-t_backi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))+
((k*A*(t_i-t_bottomi))/(2*r)) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
ui= (t_i-t_atomsphere)*d*cp*volume;

elseif z-back==1 && front==0 && bottom~=-1 && top~=-1 && left~=-1 && right~=0
%                                      z49=z
time

ui= u0 - (((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_topi))/(2*r))+
((k*A*(t_i-t_backi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))+
((k*A*(t_i-t_bottomi))/(2*r)) *(time-(time-time_step)));
change_u= ui-u0;
of u is negative here

% change_u = cp (change in T) and change of u is negative here

\[ t_i = t_i +\left(\frac{\text{change}_u}{\text{cp} \cdot d \cdot \text{volume}}\right); \]
\[ u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}; \]

\% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\% even layer, first element, top to bottom, no. of contacts=3

\%      \% even layer, first element, middle

elseif left~=0 && front==0 && z-back==-1 &&
right==0 && bottom ~==-1 && top==0

\% z50=z

ui= u0 - (\left(3 \cdot h \cdot A \cdot (t_i-t_{\text{atmosphere}}) / 6\right) + ((k \cdot A \cdot (t_i-t_{\text{backi}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{lefti}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{bottomi}})) / (2 \cdot r)) \right) \cdot (t_i - t_{\text{time-time_step}}));

\% % change_u = ui-u0;
\% change_u = cp (change in T) and change
\% even layer, first element, top to bottom, no. of contacts=4

elseif left~=0 && front==0 && z-back==-1 &&
right~=0 && bottom ~==-1 && top==0

\% z51=z

ui= u0 - (\left(2 \cdot h \cdot A \cdot (t_i-t_{\text{atmosphere}}) / 6\right) + ((k \cdot A \cdot (t_i-t_{\text{backi}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{lefti}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{righti}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{bottomi}})) / (2 \cdot r)) \right) \cdot (t_i - t_{\text{time-time_step}}));

\% % change_u = ui-u0;
\% change_u = cp (change in T) and change
\% even layer, first element, top to bottom, no. of contacts=5

elseif left~=0 && front==0 && z-back==-1 &&
right~=0 && bottom ~==-1 && top~=0

\% z52=z

ui= u0 - (\left(\frac{h \cdot A \cdot (t_i-t_{\text{atmosphere}})}{6}\right) + ((k \cdot A \cdot (t_i-t_{\text{backi}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{lefti}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{righti}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{topi}})) / (2 \cdot r)) + ((k \cdot A \cdot (t_i-t_{\text{bottomi}})) / (2 \cdot r)) \right) \cdot (t_i - t_{\text{time-time_step}}));

\% % change_u = ui-u0;
\% change_u = cp (change in T) and change
\% change_u = cp (change in T) and change of u is negative here
elseif left~=0 && z-front==1 && right==0 && bottom ~=-1 && top==0 && back==0

%          % even layer, middle element, top to bottom, no. of contacts=3

z53=z

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
%
%          % even layer, middle element, top to bottom, no. of contacts=4

elseif left~=0 && z-front==1 && right==0 && bottom ~=-1 && top==0 && z-back==-1

%              z54=z

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_backi))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
%
%          % even layer, middle element, top to bottom, no. of contacts=5

elseif left~=0 && z-front==1 && right~=0 && bottom ~=-1 && top==0 && z-back==-1

%              z55=z

ui= u0 - (((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r)))* (time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;

%          % even layer, middle element, top to bottom, no. of contacts=6

else left==0 && z-front==1 && right==0 && back==0

%  z56=z

ui= u0 - (((0*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% even layer, middle element, top to bottom, no. of contacts=6
elseif left~=0 && z-front==1 && right~=0 &&
bottom ~==-1 && top==0 && z-back==-1
%
% time
%
ui= u0 - (((0*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+
((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_topi))/(2*r))+
((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
ui0=(t_i-t_atomsphere)*d*cp*volume;
%
% even layer, end element, middle strand, top to bottom, no. of contacts=4
elseif left~=0 && z-front==1 && right~=0 &&
bottom ~==-1 && top==0 && back==0
%
% time
%
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+
((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
ui0=(t_i-t_atomsphere)*d*cp*volume;
%
% even layer, end element, top to bottom, middle row, no. of contacts=5
elseif left~=0 && z-front==1 && right~=0 &&
bottom ~==-1 && top==0 && back==0
%
% time
%
ui= u0 - (((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_lefti))/(2*r))+
((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+((k*A*(t_i-t_topi))/(2*r))+
((k*A*(t_i-t_bottomi))/(2*r)))) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and change
of u is negative here
t_i=t_i +(( change_u)/(cp*d*volume));
% even layer, first element, end strand, top to bottom, no. of contacts=4

\textbf{else if} $\text{left} = 0 \&\& \text{front} = 0 \&\& \text{z-back} = -1 \&\& \text{right} = 0 \&\& \text{bottom} = -1 \&\& \text{top} = 0

\textbf{z150} = \text{z} \text{time}

\ni = \i_0 - \frac{\text{hA} (\text{t_i-t_atomsphere})}{6} + \frac{\text{kA} (\text{t_i-t_lefti})}{2\text{r}} + \frac{\text{kA} (\text{t_i-t_topi})}{2\text{r}} + \frac{\text{kA} (\text{t_i-t_bottomi})}{2\text{r}} \times (\text{time} - (\text{time-time_step}))

\text{change_u} = \text{i} - \text{i}_0

\text{of u is negative here}

\text{t_i} = \text{t_i} + \frac{\text{change_u}}{\text{cp} \times \text{d} \times \text{volume}}

\i_0 = \text{(t_i-t_atomsphere) \times d \times cp \times volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% even layer, middle element, end strand, top to bottom, no. of contacts=5

\textbf{else if} $\text{left} = 0 \&\& \text{z-front} = 1 \&\& \text{right} = 0 \&\& \text{bottom} = -1 \&\& \text{top} = 0 \&\& \text{z-back} = -1

\textbf{z156} = \text{z} \text{time}

\ni = \i_0 - \frac{\text{kA} (\text{t_i-t_atomsphere})}{6} + \frac{\text{kA} (\text{t_i-t_lefti})}{2\text{r}} + \frac{\text{kA} (\text{t_i-t_fronti})}{2\text{r}} + \frac{\text{kA} (\text{t_i-t_topi})}{2\text{r}} + \frac{\text{kA} (\text{t_i-t_bottomi})}{2\text{r}} \times (\text{time} - (\text{time-time_step}))

\text{change_u} = \text{i} - \text{i}_0

\text{of u is negative here}

\text{t_i} = \text{t_i} + \frac{\text{change_u}}{\text{cp} \times \text{d} \times \text{volume}}

\i_0 = \text{(t_i-t_atomsphere) \times d \times cp \times volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% even layer, end element, top to bottom, end strand, no. of contacts=4

\textbf{else if} $\text{left} = 0 \&\& \text{z-front} = 1 \&\& \text{right} = 0 \&\& \text{bottom} = -1 \&\& \text{top} = 0 \&\& \text{back} = 0

\textbf{z588} = \text{z} \text{time}

\ni = \i_0 - \frac{\text{kA} (\text{t_i-t_atomsphere})}{6} + \frac{\text{kA} (\text{t_i-t_lefti})}{2\text{r}} \times (\text{time} - (\text{time-time_step}))

\text{change_u} = \text{i} - \text{i}_0

\text{of u is negative here}

\text{t_i} = \text{t_i} + \frac{\text{change_u}}{\text{cp} \times \text{d} \times \text{volume}}

\i_0 = \text{(t_i-t_atomsphere) \times d \times cp \times volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% even layer, end element, top to bottom, no. of contacts=4
of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right);
\]
\[
u_0 = (t_i - t_{\text{atomsphere}}) \times d \times \text{cp} \times \text{volume};
\]

\[
%\ \text{else}
%\ \text{if rem(mol(z,8),2)==0}
%\ \text{even layer \& odd layer, first element, first column,}
\]
\[
top \text{ to bottom, no. of contacts=1}
\]
\[
\text{if left==0 \&\& front==0 \&\& back==0 \&\& right==0 \&\& bottom ~=-1 \&\& top==0}
\]
\[
%\ z29=z
\]
\[
ui = u_0 - \left( \frac{(5 \times h \times A \times (t_i - t_{\text{atomsphere}})}{6} + \frac{((\text{cp} \times (t_i - t_{bottomi}))}{(2 \times r)}) \times (time-(time-time_step))) \right);
\]
\[
\text{change}_u = ui - u_0;
\]
\[
% \ \text{change}_u \sim \text{cp} \ (\text{change in T}) \ \text{and change of u is negative here}
\]
\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right);
\]
\[
u_0 = (t_i - t_{\text{atomsphere}}) \times d \times \text{cp} \times \text{volume};
\]

% \ \text{else}
% \ \text{if rem(mol(z,8),2)==0}
% \ \text{even layer \& odd layer, first element, first column,}
\]
\[
top \text{ to bottom, no. of contacts=1}
\]
\[
\text{if left==0 \&\& front==0 \&\& back==0 \&\& right==0 \&\& bottom ~=-1 \&\& top==0}
\]
\[
%\ z29=z
\]
\[
ui = u_0 - \left( \frac{(4 \times h \times A \times (t_i - t_{\text{atomsphere}})}{6} + \frac{(2 \times \pi \times \text{keq} \times r \times (t_i - t_{\text{plate}})}{(2 \times r)}) \times (time-(time-time_step))) \right);
\]
\[
\text{change}_u = ui - u_0;
\]
\[
% \ \text{change}_u \sim \text{cp} \ (\text{change in T}) \ \text{and change of u is negative here}
\]
\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right);
\]
\[
u_0 = (t_i - t_{\text{atomsphere}}) \times d \times \text{cp} \times \text{volume};
\]

% \ \text{else}
% \ \text{molecule 1 in first strand, no. of contacts=1 (odd layer in contact with plate)}
\]
\[
\text{elseif left==0 \&\& front==0 \&\& back==0 \&\& right~0 \&\& bottom~-1 \&\& top==0}
\]
\[
%\ z1=z
\]
\[
ui = u_0 - \left( \frac{(4 \times h \times A \times (t_i - t_{\text{atomsphere}})}{6} + \frac{(2 \times \pi \times \text{keq} \times r \times (t_i - t_{\text{plate}}))}{(2 \times r)}) \times (time-(time-time_step))) \right);
\]
\[
\text{change}_u = ui - u_0;
\]
\[
% \ \text{change}_u \sim \text{cp} \ (\text{change in T}) \ \text{and change of u is negative here}
\]
\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \times d \times \text{volume}} \right);
\]
\[
u_0 = (t_i - t_{\text{atomsphere}}) \times d \times \text{cp} \times \text{volume};
\]

% \ \text{else}
% \ \text{molecule 1 in first strand, no. of contacts=1 in first strand (left)}
\]
elseif right==0 && front==0 && back==0 &&
%                                        z-left==1 && bottom==-1 && top==0
ui= u0 - (((4*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+
*(time-(time-time_step)));
% change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here
u0=(t_i-t_atomosphere)*d*cp*volume;

elseif  front==0 && back==0 && z-left==1
%                                          z3=z
%                                          time
ui= u0 - (((3*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))...+
((k*A*(t_i-t_lefti))/(2*r)))
*(time-(time-time_step)));
% change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here
u0=(t_i-t_atomosphere)*d*cp*volume;

elseif  front==0 && right==0 && z-left==1
%                                          z4=z
%                                          time
ui= u0 - (((3*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_backi))/(2*r))...+
((k*A*(t_i-t_lefti))/(2*r)))
*(time-(time-time_step)));
% change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here
u0=(t_i-t_atomosphere)*d*cp*volume;
t_backi=display(disp_post-1,back+1);
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))...+
((k*A*(t_i-t_backi))/(2*r))
+(k*A*(t_i-t_lefti))/(2*r)) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));

% change of u is negative here
u0=(t_i-t_atomsphere)*d*cp*volume;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%no.of contacts=2 , first strand, first ele-
ment (left&right)
elseif front==0 && z-left==1 && left==0 && bottom==-1 && top==0
z6=z
time

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))...+
((k*A*(t_i-t_fronti))/(2*r))
+(k*A*(t_i-t_backi))/(2*r)) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% start of strand, no.of contacts=2, right
to left
elseif right==0 && left==0 && back==0 &&
z-front==1 && bottom==-1 && top==0
z7=z
time

ui= u0 - (((4*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))...+
((k*A*(t_i-t_fronti))/(2*r)))
*(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
% start of strand, no.of contacts=2, right
to left
elseif right==0 && back==0 &&
z-front==1 && bottom==-1 && top==0
z8=z
time

% change of u is negative here
%  
%  
%  
%  
% ui = u0 - (((3*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))
% +((k*A*(t_i-t_fronti))/(2*r))
% +((k*A*(t_i-t_lefti))/(2*r))
% *(time-(time-time_step)));

calculate u = ui - u0; 

% change u = cp (change in T) and change of u is negative here 

t_i = t_i + ((change_u)/(cp*d*volume));

% start of strand, no.of contacts=3, right to left 

elseif right==0 && z-front==1 && bottom==0 && top==0 && back~=0 && left~=0

%  
%  
% t_backi = display(disp_post-1,back+1);

%  
%  
% ui = u0 - (((2*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))
% +((k*A*(t_i-t_fronti))/(2*r))
% +((k*A*(t_i-t_lefti))/(2*r))
% *(time-(time-time_step)));

calculate u = ui - u0; 

% change u = cp (change in T) and change of u is negative here 

t_i = t_i + ((change_u)/(cp*d*volume));

% end of strand, right to left , no.of contacts=2

elseif left==0 && back==0 && z-right==1 && bottom==-1 && top==0 && front~=0

%  
%  
% t_1 = display(disp_post-1,back+1);

%  
%  
% ui = u0 - (((3*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))
% +((k*A*(t_i-t_fronti))/(2*r))
% +((k*A*(t_i-t_righti))/(2*r))
% *(time-(time-time_step)));

calculate u = ui - u0; 

% change u = cp (change in T) and change of u is negative here 

t_i = t_i + ((change_u)/(cp*d*volume));

% middle of strand, right to left , no.of contacts=3

elseif back==0 && z-right==1 && bottom==-1 && top==0 && front~=0 && left~=0
\% z11=z
\% time

ui = u0 - (((2*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))... 
+((k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_lefti))/(2*r)))*(time-(time-time_step)));
change_u = ui-u0;
\% change_u~ cp (change in T) and
change of u is negative here
%
\% t_i=t_i +(( change_u)/(cp*d*volume));
%%
\% middle of strand, right to left , no.of
\% contacts=4
elseif z-right==1 && left~=0 && bottom==-1 && top==0 && back~=0 && front~=0
%
\% t_backi=display(disp_post-1,back+1);

ui = u0 - (((h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))... 
+((k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_lefti))/(2*r)) +((k*A*(t_i-t_backi))/(2*r))) *
(time-(time-time_step)));
change_u = ui-u0;
\% change_u~ cp (change in T) and
change of u is negative here
%
\% t_i=t_i +(( change_u)/(cp*d*volume));
%%
\% (last element), right to left, no.of
\% contacts=3
elseif z-right==1 && left==0 && bottom==-1 && top==0 && back~=0 && front~=0
%
\% t_backi=display(disp_post-1,back+1);

ui = u0 - (((2*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))... 
+((k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_backi))/(2*r)))*(time-(time-time_step)));
change_u = ui-u0;
change of $u$ is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}})*d*cp*volume;
\]

contacts=2

\[
\text{elseif left==0 && bottom==0 && top==0 && right==0}
\]

\[
ui = u_0 - \left( \frac{3*h*A*(t_i-t_{\text{atmosphere}})}{6} + \frac{2*\pi*keq*r*(t_i-t_{\text{plate}})}{2*r} \right)
\]

\[
+ \left( \frac{k*A*(t_i-t_{\text{fronti}})}{2*r} \right) *(time-(time-time_step))
\]

\[
\text{change}_u = ui - u_0;
\]

\[
% change_u~ cp (change in T) and change of u is negative here
%\]

\[
t_i = t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}})*d*cp*volume;
\]

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % (first element), left to right, no. of contacts=3
% %
% elseif left==0 && bottom==0 && top==0 && right==0
% %
% % u_i= u_0 - ((3*h*A*(t_i-t_at-
%omsphere)/6)+(2*pi*keq*r*(t_i-t_plate))+(k*A*(t_i-t_righti))/(2*r))...
%+((k*A*(t_i-t_fronti)))/(2*r))
% *(time-(time-time_step)));
% change_u= ui-u0;
% % change_u~ cp (change in T) and change of u is negative here
%\]

\[
t_i = t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}})*d*cp*volume;
\]

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% % (first element), left to right, no. of contacts=2
% %
% elseif right==0 && bottom==0 && top==0 && front~=0
% %
% % t_backi=display(disp_post-
%1,back+1);
% ui= u_0 - (((2*h*A*(t_i-t_at-
%omsphere)/6)+(2*pi*keq*r*(t_i-t_plate))+(k*A*(t_i-t_righti))/(2*r))...
%+((k*A*(t_i-t_fronti)))/(2*r))
%+((k*A*(t_i-t_backi)))/(2*r)) ) *(time-(time-time_step)));
% change_u= ui-u0;
% % change_u~ cp (change in T) and change of u is negative here
%\]

\[
t_i = t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}})*d*cp*volume;
\]

% % % middle of strand, left to right, no. of contacts=2
% %
% elseif right==0 && back==0 && z-left==1 &&
% bottom==-1 && top==0 && front==0
% %
% % z16=z
% % time
%
ui = u0 - (((3*h*A*(t_i-t_atmSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 2
elseif back==0 && top==0 && right==0 && bottom==0

et1 = display(disp_post, 1 + (z-t_lefti)/(2*r));

ui = u0 - (((h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)))
+ ((k*A*(t_i-t_lefti))/(2*r)) * (time-(time-time_step)));
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 3
elseif back==0 && z-left==1 && bottom==-1 && top==0 && front~=0 && right~=0

et1 = display(disp_post, back+1);

ui = u0 - (((3*h*A*(t_i-t_atmSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 4
elseif z-left==1 && right~=0 && bottom==-1 && top==0 && back~=0 && front~=0

ui = u0 - (((h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_righti))/(2*r))
+ ((k*A*(t_i-t_fronti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 5
elseif back==0 && z-left==1 && bottom==-1 && top==0 && front~=0 && right~=0

ui = u0 - (((3*h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r))
+ ((k*A*(t_i-t_righti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 6
elseif z-left==0 && right~0 && bottom==1 && top==0 && back==0 && front~0

ui = u0 - (((h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_righti))/(2*r))
+ ((k*A*(t_i-t_fronti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 7
elseif back==0 && z-left==0 && bottom==1 && top==0 && front~0 && right~0

ui = u0 - (((3*h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r))
+ ((k*A*(t_i-t_righti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 8
elseif z-left==0 && right~0 && bottom==1 && top==0 && back~0 && front~0

ui = u0 - (((h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_righti))/(2*r))
+ ((k*A*(t_i-t_fronti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 9
elseif back==0 && z-left==0 && bottom==1 && top==0 && front==0 && right==0

ui = u0 - (((3*h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r))
+ ((k*A*(t_i-t_righti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end

% middle of strand, left to right, no. of contacts = 10
elseif z-left==0 && right==0 && bottom==1 && top==0 && back==0 && front==0

ui = u0 - (((h*A*(t_i-t_atomSphere)/6) + ((k*A*(t_i-t_righti))/(2*r))
+ ((k*A*(t_i-t_fronti))/(2*r)) * (time-(time-time_step)))
change_u = ui-u0;
% change_u ~ cp (change in T) and change of u is negative here

if (t_i-t_plate) < 0

t_i=t_i +(( change_u)/(cp*d*volume));

else

t_i=t_i +(( change_u)/(cp*d*volume));

end
elseif right==0 && z-left==1 && bottom==1 && top==0 && back~=0 && front~=0
%                                          z19=z
%                                          time

ui= u0 - (((2*h*A*(t_i-t_atmoosphere))/6)+(2*pi*keq*r*(t_i-t_plate))...
+((k*A*(t_i-t_lefti))/(2*r))+...
*((time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

%                              t_i=t_i +(( change_u)/(cp*d*volume));
%                              u0=(t_i-t_atmosphere)*d*cp*volume;

elseif  front==0 && right==0 && z-left==1 && bottom==-1 && top~=0 && back~=0 && left~=0
%                                          z20=z
%                                          time

ui= u0 - (((2*h*A*(t_i-t_atmoosphere))/6)+(2*pi*keq*r*(t_i-t_plate))...
+((k*A*(t_i-t_topi))/(2*r))
+((k*A*(t_i-t_righti))/(2*r))
+((k*A*(t_i-t_backi))/(2*r))
+((k*A*(t_i-t_lefti))/(2*r)) *
(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

%                              t_i=t_i +(( change_u)/(cp*d*volume));
%                              u0=(t_i-t_atmosphere)*d*cp*volume;

elseif front==0 && z-left==1 && left~=0 && bottom==-1 && top~=0 && back~=0 && right~=0
%                                          z21=z
%                                          time

ui= u0 - (((h*A*(t_i-t_atmoosphere))/6)+(2*pi*keq*r*(t_i-t_plate))
+((k*A*(t_i-t_righti))/(2*r))
+((k*A*(t_i-t_backi))/(2*r))
+((k*A*(t_i-t_lefti))/(2*r))
+((k*A*(t_i-t_topi))/(2*r))
*
(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

%                              t_i=t_i +(( change_u)/(cp*d*volume));
%                              u0=(t_i-t_atmosphere)*d*cp*volume;
%no.of contacts=3 ,first strand,first element (left&right)

elseif front==0 && z-left==1 && left==0 && bottom==-1 && top~=0 && back~=0 && right~=0

\%                                              z22=z 
\%                                               time
\%
ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))
  +((k*A*(t_i-t_backi))/(2*r))+(k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

                    t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% start of strand, no.of contacts=3, right to left

elseif right==0 && z-front==1 && bottom==-1 && back~=0 && top~=0 && left~=0

\%                                          z23=z 
\%                                           time
\%
ui= u0 - (((h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_fronti))/(2*r))
  +((k*A*(t_i-t_backi))/(2*r))+(k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

                    t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% middle of strand, right to left ,no.of contacts=5

elseif z-right==1 && left~=0 && bottom==-1 && back~=0 && top~=0 && front~=0

\%                                          z24=z 
\%                                           time
\%
ui= u0 - (((k*A*(t_i-t_topi))/(2*r))+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))
  +((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_lefti))/(2*r)) * (time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

                    t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
contacts=4
% (last element), right to left, no. of
% contacts=4

elseif z-right==1 && left==0 && bottom==0
% %
% %
% z25=z
% time

% ui= u0 - ((h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))
% +((k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));
% change_u= ui-u0;
% % change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomosphere)*d*cp*volume;

% % start of strand, no. of contacts=3, right
to left, last strand

elseif right==0 && z-front==1 && bottom==0 && left==0
% %
% %
% z123=z
% time

% ui= u0 - ((2*h*A*(t_i-t_atomosphere)/6)+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))
% +((k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));
% change_u= ui-u0;
% % change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomosphere)*d*cp*volume;

% % middle of strand, right to left, no. of
% contacts=5 last strand

elseif z-right==1 && left==0 && bottom==0
% %
% %
% z124=z
% time

% ui= u0 - (((k*A*(t_i-t_topi))/(2*r))+(2*pi*keq*r*(t_i-t_plate))+((k*A*(t_i-t_righti))/(2*r))
% +((k*A*(t_i-t_lefti))/(2*r)) +((k*A*(t_i-t_atomosphere)/6))*(time-(time-time_step)));
% change_u= ui-u0;
% % change_u~ cp (change in T) and
t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

contacts=4, last strand
elseif z-right==1 && left==0 && bottom==0 && top==0
  z125=z

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate)) +((k*A*(t_i-t_righti))/(2*r))...+
  ((k*A*(t_i-t_topi))/(2*r)) ) *(time-(time-time_step)));
change_u= ui-u0;

change of u is negative here
  t_i=t_i +(( change_u)/(cp*d*volume));

contacts=4
else left==0 && z-front==1 && bottom==0 && left==1 && right==-1 && top==0 && back==0
  z26=z

ui= u0 - (((h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate)) +((k*A*(t_i-t_righti))/(2*r))...+
  ((k*A*(t_i-t_topi))/(2*r)) +((k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_backi))/(2*r)) ) *(time-(time-
  time_step)));
change_u= ui-u0;

change of u is negative here
  t_i=t_i +(( change_u)/(cp*d*volume));

contacts=5
else z-left==1 && right==0 && bottom==0
  z27=z

ui= u0 - (((((k*A*(t_i-t_topi))/(2*r))+(2*pi*keq*r*(t_i-t_plate)) +((k*A*(t_i-t_righti))/(2*r)))...+
  ((k*A*(t_i-t_lefti))/(2*r))+(k*A*(t_i-t_backi))/(2*r)) ) *(time-(time-time_step)));
change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}
\]

% (last element), left to right, no. of contacts=4

elseif right==0 && z-left==1 && bottom==-1 && top~0 && front~0 && back~0

\[
ui = u_0 - \left( \frac{h \cdot A \cdot (t_i - t_{\text{atmosphere}})}{6} + \frac{2\pi \cdot k \cdot e_{\text{q}} \cdot r \cdot (t_i - t_{\text{plate}})}{6} + \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

change of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}
\]

elseif left==0 && front==0 && back==0 && z-right==-1 && bottom~=-1 && top==0

\[
ui = u_0 - \left( \frac{4 \cdot h \cdot A \cdot (t_i - t_{\text{atmosphere}})}{6} + \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

change of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}
\]

% odd layer (not in contact with plate) first element, first strand, left to right, no. of contacts=2 (bottom, right)

elseif left==0 && front==0 && back==0 && z-right==-1 && bottom~=-1 && top==0

\[
ui = u_0 - \left( \frac{4 \cdot h \cdot A \cdot (t_i - t_{\text{atmosphere}})}{6} + \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

change of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{\text{cp}\cdot d\cdot \text{volume}} \right)
\]

\[
u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}
\]
\begin{verbatim}
elseif front==0 && z-right==-1 &&
left==0 && bottom==-1 && top==0 && back==0

\% z60=z\% time

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+...
\quad ((k*A*(t_i-t_backi))/(2*r))) *(time-(time-time_step)));

% change_u= ui-u0;
% change_u~ cp (change in T) and
% change of u is negative here

change_u= t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;
\end{verbatim}
elements (left&right)

elseif  front==0 && back==0 && z-left==1 && bottom==0 && top==0 && right~=0

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r)) + ((k*A*(t_i-t_bottomi))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%no.of contacts=4 ,first strand,middle
elements (left&right)

elseif  front==0 && z-left==1 && bottom==0 && top~=-1 && right~=0 && back~=0

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r)) + ((k*A*(t_i-t_bottomi))/(2*r)) + ((k*A*(t_i-t_backi))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r))) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%no.of contacts=5 ,first strand,middle
elements (left&right)

elseif  front==0 && z-left==1 && bottom==0 && top~=-1 && right~=0 && back~=0

ui= u0 - (((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_bottomi))/(2*r)) + ((k*A*(t_i-t_topi))/(2*r)) + ((k*A*(t_i-t_backi))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) *(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));

u0=(t_i-t_atomsphere)*d*cp*volume;
elseif front==0 && z-left==1 && bottom~=-1 && top==0 && right==0 && back~=0

z65=z
time

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_backi))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+((k*A*(t_i-t_lefti))/(2*r)))*(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;

elseif front==0 && z-left==1 && bottom~=-1 && top==0 && right==0 && back~=0

z66=z
time

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_topi))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+((k*A*(t_i-t_backi))/(2*r))+((k*A*(t_i-t_lefti))/(2*r)))*(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;

elseif right==0 && left==0 && back==0 && z-front==1 && bottom~=-1 && top==0

z67=z
time

ui= u0 - (((4*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_bottomi))/(2*r))+((k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_lefti))/(2*r))*(time-(time-time_step)));

change_u= ui-u0;
% change_u~ cp (change in T) and
change of u is negative here
\[ t_i = t_i + \left( \frac{\text{change}_u}{\text{cp} \cdot d \cdot \text{volume}} \right) \]

\[ u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot \text{cp} \cdot \text{volume}; \]

\%

% start of strand, no. of contacts = 3, right to left

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%

\%
t_i = t_i + \((\text{change}_u)/(cp \cdot d \cdot \text{volume})\);

u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot cp \cdot \text{volume};

% odd layer (not in contact with plate) middle elements, right to left,
% number of contacts=3 (bottom, right, front)
else if front~0 && back~0 && z-right~1 && bottom~=-1 && top==0 && left==0
%                                          z71=z
% time

ui = u_0 - (((3 \cdot h \cdot A \cdot (t_i - t_{\text{atmosphere}})/(2 \cdot r))... +((k \cdot A \cdot (t_i - t_{\text{fronti}}))/(2 \cdot r))
+((k \cdot A \cdot (t_i - t_{\text{bottomi}}))/(2 \cdot r)) \cdot \text{time} - (\text{time-time}_\text{step})));

change_u = ui - u_0;
% change_u ~ cp (change in T) and change of u is negative here
t_i = t_i + \((\text{change}_u)/(cp \cdot d \cdot \text{volume})\);

u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot cp \cdot \text{volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% middle of strand, right to left, no.of contacts=4
elseif front~0 && z-right~1 && left~0 && bottom~=-1 && top==0 && back~0
%                                          z72=z
% time

ui = u_0 - (((2 \cdot h \cdot A \cdot (t_i - t_{\text{atmosphere}})/(2 \cdot r))... +((k \cdot A \cdot (t_i - t_{\text{fronti}}))/(2 \cdot r))
+((k \cdot A \cdot (t_i - t_{\text{lefti}}))/(2 \cdot r)) +... +((k \cdot A \cdot (t_i - t_{\text{bottomi}}))/(2 \cdot r)) \cdot \text{time} - (\text{time-time}_\text{step})));

change_u = ui - u_0;
% change_u ~ cp (change in T) and change of u is negative here
t_i = t_i + \((\text{change}_u)/(cp \cdot d \cdot \text{volume})\);

u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot cp \cdot \text{volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% middle of strand, right to left, no.of contacts=5
elseif front~0 && z-right~1 && left~0 && bottom~=-1 && top==0 && back~0
%                                          z73=z
% time

ui = u_0 - (((2 \cdot h \cdot A \cdot (t_i - t_{\text{atmosphere}})/(2 \cdot r))... +((k \cdot A \cdot (t_i - t_{\text{fronti}}))/(2 \cdot r))
+((k \cdot A \cdot (t_i - t_{\text{lefti}}))/(2 \cdot r)) +... +((k \cdot A \cdot (t_i - t_{\text{bottomi}}))/(2 \cdot r)) \cdot \text{time} - (\text{time-time}_\text{step})));

change_u = ui - u_0;
% change_u ~ cp (change in T) and change of u is negative here
t_i = t_i + \((\text{change}_u)/(cp \cdot d \cdot \text{volume})\);

u_0 = (t_i - t_{\text{atmosphere}}) \cdot d \cdot cp \cdot \text{volume};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ui = u0 - (((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_topi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))
+((k*A*(t_i-t_lefti))/(2*r))+(k*A*(t_i-t_fronti))/(2*r))
+((k*A*(t_i-t_backi))/(2*r)))
* (time-(time-time_step)));

change_u = ui-u0;
% change_u ~ cp (change in T) and
% change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% middle of strand, right to left , no. of contacts=6

elseif front~=0 && z-right==1 && left~=0 && bottom~=-1 && top~=0 && back~=0

% z74=z;
% time;

ui = u0 - (((k*A*(t_i-t_topi))/(2*r))+(k*A*(t_i-t_bottomi))/(2*r))+(k*A*(t_i-t_righti))/(2*r)
+((k*A*(t_i-t_lefti))/(2*r))+(k*A*(t_i-t_fronti))/(2*r)
+((k*A*(t_i-t_backi))/(2*r))
* (time-(time-time_step)));

change_u = ui-u0;
% change_u ~ cp (change in T) and
% change of u is negative here

% t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% end of strand, right to left, no. of contacts=4

elseif front~=0 && left==0 && z-right==1

% z75=z;
% time;

ui = u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_topi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))
+((k*A*(t_i-t_lefti))/(2*r))+(k*A*(t_i-t_fronti))/(2*r))
+((k*A*(t_i-t_backi))/(2*r))
* (time-(time-time_step)));

change_u = ui-u0;
% change_u ~ cp (change in T) and
% change of u is negative here

% t_i=t_i +(( change_u)/(cp*d*volume));
u0=(t_i-t_atomsphere)*d*cp*volume;
contacts=5

elseif front~=0 && z-right==1 && left==0

% (last element), right to left, no. of contacts=5

& bottom~=-1 && top~=0 && back~=0

z76 = z
time

ui= u0 - (((h*A*(t_i-t_atomsphere)/6)+(2*pi*keq*r*(t_i-t_plate))...+
(k*A*(t_i-t_righti))/(2*r))
+((k*A*(t_i-t_backi))/(2*r)+((k*A*(t_i-t_topi))/(2*r)) ) * (time-(time-time_step)));

% change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));

else...

element, no.of contacts=3 (bottom, right, start)

bottom~=-1 && top==0 && left==0

z77 = z
time

ui= u0 - (((3*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))...+
(k*A*(t_i-t_bottomi))/(2*r)) * (time-(time-time_step)));

% change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

t_i=t_i +(( change_u)/(cp*d*volume));

else

element, no.of contacts=4 , middle strands, first

left==0 && bottom~=-1 && top==0 && back~=0

z78 = z
time

ui= u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))...+
((k*A*(t_i-t_backi))/(2*r)) ) * (time-(time-time_step)));

% change_u= ui-u0;
% change_u~ cp (change in T) and change of u is negative here

\[
t_i = t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right), \\
u_0 = (t_i - t_{\text{atmosphere}}) * d * cp * volume; \\
\] % odd layer no. of contacts=5 , middle
strands, strand, first element (left to right) 
\textbf{elseif} 
\begin{align*}
\text{front} &= 0 \\
\text{z-right} &= -1 \\
\text{left} &= 0 \\
\text{bottom} &= -1 \\
\text{top} &= 0 \\
\text{back} &= 0 \\
\end{align*}
\% %
\begin{align*}
t_i &= t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right), \\
u_0 &= (t_i - t_{\text{atmosphere}}) * d * cp * volume; \\
\] % odd layer no. of contacts=3 , middle
strands, strand, middle element (left to right) 
\textbf{elseif} 
\begin{align*}
\text{z-left} &= 1 \\
\text{left} &= 0 \\
\text{bottom} &= -1 \\
\text{top} &= 0 \\
\text{back} &= 0 \\
\text{right} &= 0 \\
\text{front} &= 0 \\
\end{align*}
\% %
\begin{align*}
t_i &= t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right), \\
u_0 &= (t_i - t_{\text{atmosphere}}) * d * cp * volume; \\
\] % odd layer no. of contacts=4 , middle
strands, strand, middle element (left to right) 
\textbf{elseif} 
\begin{align*}
\text{z-left} &= 1 \\
\text{left} &= 0 \\
\text{bottom} &= -1 \\
\text{top} &= 0 \\
\text{back} &= 0 \\
\text{right} &= 0 \\
\text{front} &= 0 \\
\end{align*}
\% %
\begin{align*}
t_i &= t_i + \left( \frac{\text{change}_u}{(cp*d*volume)} \right), \\
u_0 &= (t_i - t_{\text{atmosphere}}) * d * cp * volume; \\
\]
\begin{align*}
\text{change_u} &= \text{ui} - u_0; \\
\text{change_u} &= \text{cp} \times (\text{change in T}) \text{ and} \\
\text{change of u is negative here} \\
\text{t_i} &= \text{t_i} + \left(\text{change_u} / (\text{cp} \times d \times \text{volume})\right); \\
\text{u}_0 &= (\text{t_i} - \text{t_atmosphere}) \times d \times \text{cp} \times \text{volume}; \\
\text{else if} \quad z-left = 1 \land \text{left} = 0 \land \text{bottom} = -1 \land \text{top} = 0 \land \text{back} = 0 \land \text{right} = 0 \land \text{front} = 0 \\
\text{ui} &= u_0 - \frac{(t_i - t_atmosphere)}{6} + \frac{(k \times A \times (t_i - t_lefti)/(2r)) \times (t_i - t_righti)/(2r)}{2r} \times \text{time} - \text{time}\_\text{step}); \\
\text{change_u} &= \text{ui} - u_0; \\
\text{change_u} &= \text{cp} \times (\text{change in T}) \text{ and} \\
\text{change of u is negative here} \\
\text{t_i} &= \text{t_i} + \left(\text{change_u} / (\text{cp} \times d \times \text{volume})\right); \\
\text{u}_0 &= (\text{t_i} - \text{t_atmosphere}) \times d \times \text{cp} \times \text{volume}; \\
\text{else if} \quad z-left = 1 \land \text{left} = 0 \land \text{bottom} = -1 \land \text{top} = 0 \land \text{right} = 0 \land \text{front} = 0 \\
\text{ui} &= u_0 - \frac{(0 \times h \times A \times (t_i - t_atmosphere)}{6} + \frac{(k \times A \times (t_i - t_lefti)/(2r)) \times (t_i - t_righti)/(2r)}{2r} \times \text{time} - \text{time}\_\text{step}); \\
\text{change_u} &= \text{ui} - u_0; \\
\text{change_u} &= \text{cp} \times (\text{change in T}) \text{ and} \\
\text{change of u is negative here} \\
\text{t_i} &= \text{t_i} + \left(\text{change_u} / (\text{cp} \times d \times \text{volume})\right); \\
\text{u}_0 &= (\text{t_i} - \text{t_atmosphere}) \times d \times \text{cp} \times \text{volume}; \\
\text{else if} \quad \text{front} = 0 \land \text{z-left} = 1 \land \text{bottom} = 0 \\
\text{ui} &= u_0 - \frac{(1 \times h \times A \times (t_i - t_atmosphere)}{6} + \frac{(k \times A \times (t_i - t_lefti)/(2r)) \times (t_i - t_righti)/(2r)}{2r} \times \text{time} - \text{time}\_\text{step}); \\
\text{change_u} &= \text{ui} - u_0; \\
\text{change_u} &= \text{cp} \times (\text{change in T}) \text{ and} \\
\text{change of u is negative here} \\
\text{t_i} &= \text{t_i} + \left(\text{change_u} / (\text{cp} \times d \times \text{volume})\right); \\
\text{u}_0 &= (\text{t_i} - \text{t_atmosphere}) \times d \times \text{cp} \times \text{volume}; \\
\text{else if} \quad \text{front} = 0 \land \text{z-left} = 1 \land \text{bottom} = 0 \\
\text{ui} &= u_0 - \frac{(0 \times h \times A \times (t_i - t_atmosphere)}{6} + \frac{(k \times A \times (t_i - t_lefti)/(2r)) \times (t_i - t_righti)/(2r)}{2r} \times \text{time} - \text{time}\_\text{step}); \\
\text{change_u} &= \text{ui} - u_0; \\
\text{change_u} &= \text{cp} \times (\text{change in T}) \text{ and} \\
\text{change of u is negative here} \\
\text{t_i} &= \text{t_i} + \left(\text{change_u} / (\text{cp} \times d \times \text{volume})\right); \\
\text{u}_0 &= (\text{t_i} - \text{t_atmosphere}) \times d \times \text{cp} \times \text{volume}; \\
\text{else if} \quad \text{front} = 0 
\end{align*}
% z80 = z
% time

ui = u0 - (((2*h*A*(t_i-t_atomosphere)/6) + ((k*A*(t_i-t_backi))/(2*r)) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) + (k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));

change_u = ui - u0;
% change_u ~ cp (change in T) and
change of u is negative here
t_i = t_i + ((change_u)/(cp*d*volume));

u0 = (t_i-t_atomosphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% no.of contacts = 5, middle strand, end elements (left & right)
% z81 = z;  time;

else if front ~= 0 && z-left == 1 && bottom ~= -1 && top ~= 0 && right == 0 && back ~= 0

ui = u0 - (((1*h*A*(t_i-t_atomosphere)/6) + ((k*A*(t_i-t_topi))/(2*r)) + ((k*A*(t_i-t_backi))/(2*r)) + ((k*A*(t_i-t_fronti))/(2*r)) + ((k*A*(t_i-t_lefti))/(2*r)) + (k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));

change_u = ui - u0;
% change_u ~ cp (change in T) and
change of u is negative here
t_i = t_i + ((change_u)/(cp*d*volume));

u0 = (t_i-t_atomosphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% odd layer, no.of contacts = 4, end strands
% z82 = z;  time;

else if front ~= 0 && z-back == 0 && z-top == 0 && z-right == -1 && z-left == 0 && bottom ~= -1

ui = u0 - (((2*h*A*(t_i-t_atomosphere)/6) + ((k*A*(t_i-t_righti))/(2*r)) + ((k*A*(t_i-t_topi))/(2*r)) + ((k*A*(t_i-t_righti))/(2*r)) + (k*A*(t_i-t_topi))/(2*r)) * (time-(time-time_step)));

change_u = ui - u0;
% change_u ~ cp (change in T) and
change of u is negative here
t_i = t_i + ((change_u)/(cp*d*volume));

u0 = (t_i-t_atomosphere)*d*cp*volume;
% odd layer, no. of contacts = 5 , end

strand, middle element (left to right)

elseif front ~= 0 && back == 0 && z-left == 1 && z-right == -1 && top ~= 0

ui = u0 - ((((1*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_topi))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_lefti))/(2*r))) *(time-(time-time_step)));

t_i = t_i +(( change_u)/(cp*d*volume));

change of u is negative here

% change_u = cp (change in T) and change of u is negative here

u0 = (t_i-t_atomsphere)*d*cp*volume;

% odd layer, no. of contacts = 4 , end

strand, end element (left to right)

elseif front ~= 0 && back == 0 && z-left == 1 && bottom ~= -1 && top == 0 && right == 0

ui = u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_topi))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+(k*A*(t_i-t_righti))/(2*r))+(k*A*(t_i-t_fronti))/(2*r)) +((k*A*(t_i-t_lefti))/(2*r))) *(time-(time-time_step)));

t_i = t_i +(( change_u)/(cp*d*volume));

change of u is negative here

% change_u = cp (change in T) and change of u is negative here

u0 = (t_i-t_atomsphere)*d*cp*volume;

% odd layer, no. of contacts = 4 , end

, end element (right to left)

elseif front ~= 0 && back == 0 && z-right == 1 && left == 0 && bottom ~= -1 && top == 0

ui = u0 - (((2*h*A*(t_i-t_atomsphere)/6)+((k*A*(t_i-t_righti))/(2*r))+((k*A*(t_i-t_bottomi))/(2*r))+(k*A*(t_i-t_fronti))/(2*r))+(k*A*(t_i-t_lefti))/(2*r)) +((k*A*(t_i-t_topi))/(2*r))) *(time-(time-time_step)));

t_i = t_i +(( change_u)/(cp*d*volume));

change of u is negative here

% change_u = cp (change in T) and change of u is negative here
t_i = t_i + ((change_u)/(cp*d*volume));

u0 = (t_i - t_atomsphere)*d*cp*volume;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% odd layer, no. of contacts=5 , end

strand, middle element (right to left)
    elseif back==0 && front~=0 && z-right==1 && z-left==-1 && bottom~=-1 && top~=0
    z86 = z
time

ui = u0 - (((1*h*A*(t_i - t_atomsphere)/6) + ((k*A*(t_i - t_topi))/(2*r)) + ((k*A*(t_i - t_bottomi))/(2*r)) + ((k*A*(t_i - t_righti))/(2*r)) + ((k*A*(t_i - t_fronti))/(2*r)) + ((k*A*(t_i - t_lefti))/(2*r)))*(time-(time-time_step)));

change_u = ui - u0;
% change_u = cp (change in T) and change of u is negative here

end

end

end

% strand, first element (right to left)
    elseif front~=0 && back==0 && z-left==-1 && bottom~=-1 && top~=0 && right==0
    z87 = z
time

ui = u0 - (((2*h*A*(t_i - t_atomsphere)/6) + ((k*A*(t_i - t_topi))/(2*r)) + ((k*A*(t_i - t_bottomi))/(2*r)) + ((k*A*(t_i - t_righti))/(2*r)) + ((k*A*(t_i - t_fronti))/(2*r)) + ((k*A*(t_i - t_lefti))/(2*r)))*(time-(time-time_step)));

change_u = ui - u0;
% change_u = cp (change in T) and change of u is negative here

t_i = t_i + ((change_u)/(cp*d*volume));

end

end

end

% time
% t_i;
end
display(disp_post,1)=time;

display(disp_post,z+1)=t_i;

Uo(disp_post,1)=time;
Uo(disp_post,z+1)=u0;

%            t_inext=ti_cond;
%             u0=(t_inext-t_atomsphere)*d*cp*volume;
%            display(disp_post,z+2)=t_inext;
%
%             Uo(disp_post,1)=time;
%            Uo(disp_post,z+2)=u0;

    end
    disp_post=disp_post+1;

    end
end