

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Using MATLAB to Compute Heat Transfer in Free Form Extrusion

Sidonie Costa, Fernando Duarte and José A. Covas
*University of Minho
 Portugal*

1. Introduction

Rapid Prototyping (RP) is a group of techniques used to quickly fabricate a scale model of a part or assembly using three-dimensional computer aided design (CAD) data (Marsan, Dutta, 2000). A large number of RP technologies have been developed to manufacture polymer, metal, or ceramic parts, without any mould, namely Stereolithography (SL), Laminated Object Manufacturing (LOM), Selected Laser Sintering (SLS), Ink-jet Printing (3DP) and Fused Deposition Modeling (FDM).

In Fused Deposition Modelling (developed by Stratasys Inc in U.S.A.), a plastic or wax filament is fed through a nozzle and deposited onto the support (Pérez, 2002; Ahn, 2002; Ziemian & Crown, 2001) as a series of 2D slices of a 3D part. The nozzle moves in the X-Y plane to create one slice of the part. Then, the support moves vertically (Z direction) so that the nozzle deposits a new layer on top of the previous one. Since the filament is extruded as a melt, the newly deposited material fuses with the last deposited material.

Free Form Extrusion (FFE) is a variant of FDM (Figure 1), where the material is melted and deposited by an extruder & die (Agarwala, Jamalabad, Langrana, Safari, Whalen & Danthord, 1996; Bellini, Shor & Gucer, 2005). FFE enables the use of a wide variety of polymer systems (e.g., filled compounds, polymer blends, composites, nanocomposites, foams), thus yielding parts with superior performance. Moreover, the adoption of co-extrusion or sequential extrusion techniques confers the possibility to combine different materials for specific properties, such as soft/hard zones or transparent/opaque effects.

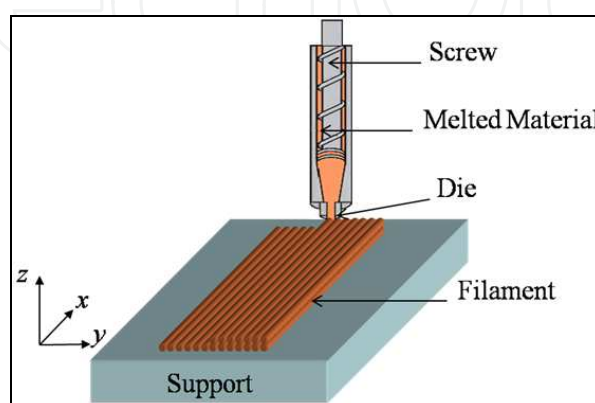


Fig. 1. Free Form Extrusion (FFE).

Due to their characteristics - layer by layer construction using melted materials - FDM and FFE may originate parts with two defects: i) excessive filament deformation upon cooling can jeopardize the final dimensional accuracy, ii) poor bonding between adjacent filaments reduces the mechanical resistance. Deformation and bonding are mainly controlled by the heat transfer, i. e., adequate bonding requires that the filaments remain sufficiently hot during enough time to ensure adhesion and, simultaneously, to cool down fast enough to avoid excessive deformation due to gravity (and weight of the filaments above them). Therefore, it is important to know the evolution in time of the filaments temperature and to understand how it is affected by the major process variables. Rodriguez (Rodriguez, 1999) studied the cooling of five elliptical filaments stacked vertically using via finite element methods and later found a 2D analytical solution for rectangular cross-sections (Thomas & Rodriguez, 2000). Yardimci and Gucer (Yardimci & S.I. Gucer, 1996) developed a more general 2D heat transfer analysis, also using finite element methods. Li and co-workers (Li, Sun, Bellehumeur & Gu, 2003; Sun, Rizvi, Bellehumeur & Gu, 2004) developed an analytical 1D transient heat transfer model for a single filament, using the Lumped Capacity method. Although good agreement with experimental results was reported, the model cannot be used for a sequence of filaments, as thermal contacts are ignored.

The present work expands the above efforts, by proposing a transient heat transfer analysis of filament deposition that includes the physical contacts between any filament and its neighbours or supporting table. The analytical analysis for one filament is first discussed, yielding an expression for the evolution of temperature with deposition time. Then, a MatLab code is developed to compute the temperature evolution for the various filaments required to build one part. The usefulness of the results is illustrated with two case studies.

2. Heat transfer modelling

During the construction of a part by FDM or FFE, all the filaments are subjected to the same heat transfer mechanism but with different boundary conditions, depending on the part geometry and deposition sequence (Figure 2).

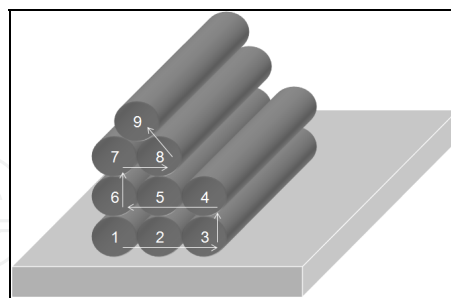


Fig. 2. Example of a sequence of filaments deposition.

Consider that N is the total number of deposited filaments and that $T_r(x,t)$ is the temperature at length x of the r -th filament ($r \in \{1, \dots, N\}$) at instant t . The energy balance for an element dx of the r -th filament writes as:

$$\begin{aligned} & \left\{ \begin{array}{l} \text{Energy in at one face} - \text{Heat loss by convection with surroundings} \\ - \text{Heat loss by conduction with adjacent filaments or with support} \end{array} \right. = \\ & = \text{Change in internal energy} + \text{Energy out at opposite face} \end{aligned}$$

This can be expressed as a differential equation. After some assumptions and simplifications (Costa, Duarte & Covas, 2008):

$$\frac{\partial T_r}{\partial t} = -\frac{P}{\rho CA} \left(h_{conv} \left(1 - \sum_{i=1}^n a_{r_i} \lambda_i \right) (T_r - T_E) + \sum_{i=1}^n h_i a_{r_i} \lambda_i (T_r - T_{r_i}) \right) \quad (1)$$

where P is filament perimeter, ρ is density, C is heat capacity, A is area of the filament cross-section, h_{conv} is heat transfer coefficient, n is number of contacts with adjacent filaments or with the support, λ_i is fraction of P that in contact with an adjacent filament, T_E is environment temperature, h_i is thermal contact conductance for contact i ($i \in \{1, \dots, n\}$) and T_{r_i} is temperature of the adjacent filament or support at contact i ($r_i \in \{1, \dots, N+1\}$, $r_i \neq r$, T_1, \dots, T_N are temperatures of filaments, T_{N+1} is support temperature). In this expression, variables a_{r_i} are defined as (see Figure 3):

$$a_{r_i} = \begin{cases} 1 & \text{if the } r\text{-th filament has the } i\text{-th contact} \\ 0 & \text{otherwise} \end{cases}, \forall i \in \{1, \dots, n\}, \forall r \in \{1, \dots, N\} \quad (2)$$

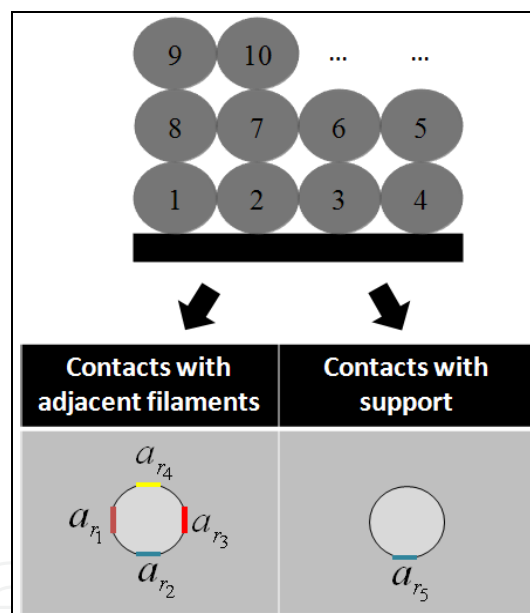


Fig. 3. Contact areas of a filament ($n = 5$).

Considering that:

$$\begin{cases} b(a_{r_1}, \dots, a_{r_n}) = h_{conv} \left(1 - \sum_{i=1}^n a_{r_i} \lambda_i \right) + \sum_{i=1}^n a_{r_i} h_i \lambda_i \\ Q(a_{r_1}, \dots, a_{r_n}) = \frac{h_{conv} \left(1 - \sum_{i=1}^n a_{r_i} \lambda_i \right) T_E + \sum_{i=1}^n a_{r_i} h_i \lambda_i T_{r_i}}{b(a_{r_1}, \dots, a_{r_n})} \end{cases} \quad (3)$$

equation (1) can be re-written as:

$$\frac{\rho VC}{PL b(a_{r_1}, \dots, a_{r_n})} \frac{\partial T_r(t)}{\partial t} + T_r(t) = Q(a_{r_1}, \dots, a_{r_n}) \quad (4)$$

Since the coefficients are constants, the characteristic polynomial method can be used to yield the solution:

$$T_r(t) = \left(T_{r0} - Q(a_{r_1}, \dots, a_{r_n}) \right) e^{\frac{-PL b(a_{r_1}, \dots, a_{r_n})}{\rho VC} (t - t_r)} + Q(a_{r_1}, \dots, a_{r_n}) \quad (5)$$

In this expression, t_r is the instant at which the r -th filament starts to cool down or contact with another filament and $T_{r0} = T_r(t_r)$ is the temperature of the filament at instant t_r . Taking k as thermal conductivity, the Biot number can be defined (Bejan, 1993):

$$Bi = \frac{A b(a_{r_1}, \dots, a_{r_n})}{P k} \quad (6)$$

When Bi is lower than 0.1, the filaments are thermally thin, i.e., thermal gradients throughout the cross section can be neglected. In this case, Eq. (5) becomes:

$$Bi \leq 0.1 \Rightarrow T_r(t) = \left(T_{r0} - Q(a_{r_1}, \dots, a_{r_n}) \right) e^{\frac{-PL b(a_{r_1}, \dots, a_{r_n})}{\rho VC} (t - t_r)} + Q(a_{r_1}, \dots, a_{r_n}) \quad (7)$$

3. Computer modelling

Equations (5) and (7) quantify the temperature of a single filament fragment along the deposition time. In practice, consecutive filament fragments are deposited during the manufacture of a part. Thus, it is convenient to generalize the computations to obtain the temperature evolution of each filament fragment at any point x of the part, for different deposition techniques and 3D configuration structures.

3.1 Generalizing the heat transfer computations

Up-dating the thermal conditions: The boundary conditions must be up-dated as the deposition develops. The code activates the physical contacts and redefines the boundary conditions for a specific filament position, time and deposition sequence. For all the filaments, three variables need to be up-dated:

- time t_r (TCV-1): instant at which the r -th filament starts cooling down, or enters in contact with another filament;
- temperature T_{r0} (TCV-2): temperature at t_r ;
- vector a_{ri} (TCV-3): in Eq. (3), sets in the contacts for the r -th filament ($i \in \{1, \dots, n\}$, where n is the number of contacts).

Simultaneous computation of the filaments temperature: During deposition, some filaments are reheated when new contacts with hotter filaments arise; simultaneously, the latter cool down due to the same contacts. This implies the simultaneous computation of the filaments temperature via an iterative procedure. The convergence error was set at $\varepsilon = 10^{-3}$, as a good compromise between accuracy and the computation time.

Deposition sequence: The deposition sequence defines the thermal conditions TCV-1, TCV-2 and TCV-3. Three possibilities were taken in: unidirectional and aligned filaments, unidirectional and skewed filaments, perpendicular filaments (see Figure 4). In all cases, the filaments are deposited continuously under constant speed (no interruptions occur between successive layers).

Some parts with some geometrical features may require the use of a support material, to be removed after manufacture. This possibility is considered in the algorithm for unidirectional and aligned filaments.

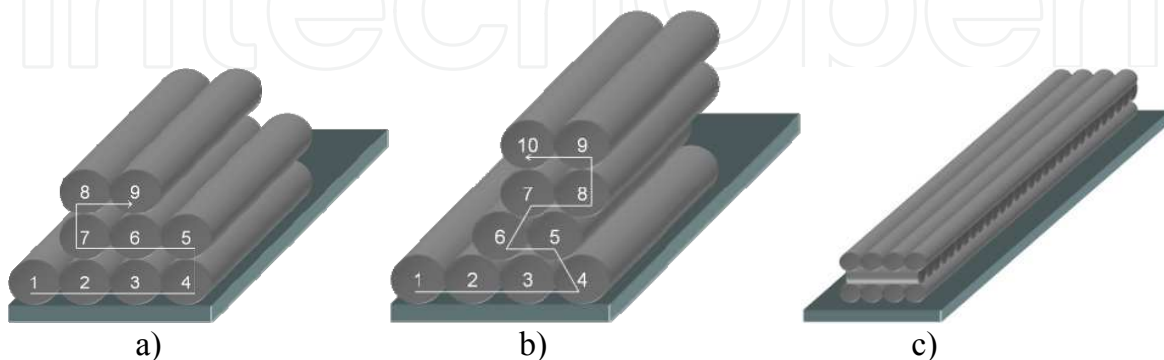


Fig. 4. Deposition sequences: a) unidirectional and aligned filaments; b) unidirectional and skewed filaments; c) perpendicular filaments.

3.2 Computing the temperatures

The computational flowchart is presented in Figure 5 and a MatLab code was generated. In order to visualize the results using another software (Excel, Tecplot...), a document in *txt* format is generated after the computations, that includes all the temperature results along deposition time.

4. MatLab code for one filament layer

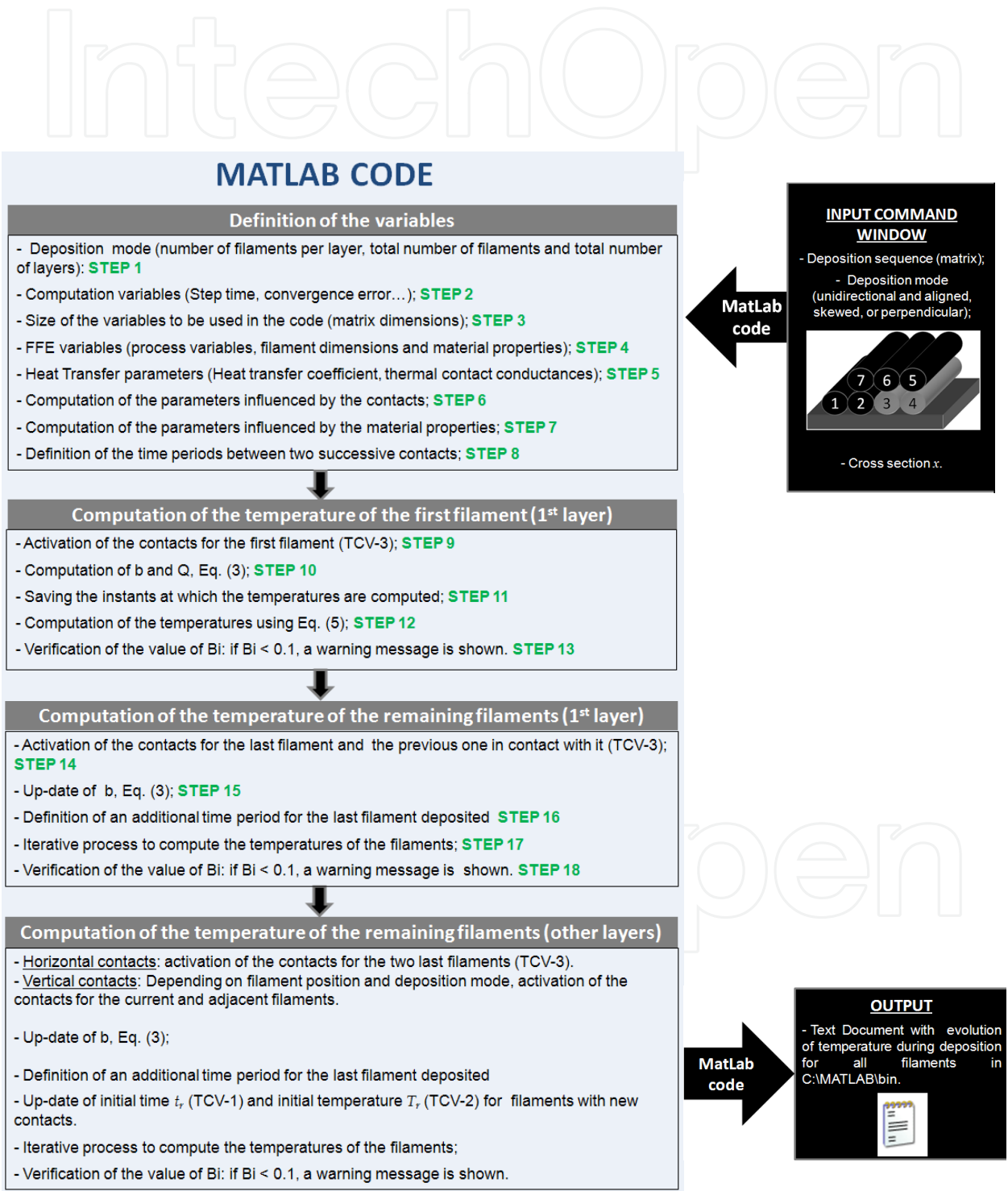
In order to illustrate how the MatLab code “*FFE.m*” was implemented, the segment dealing with the temperature along the deposition time for the first layer of filaments, using one or two distinct materials, is presented here. The code has the same logic and structure for the remaining layers.

4.1 Input variables

Two arguments need to be introduced in this MatLab function:

- A matrix representing the deposition sequence, containing m rows and n columns, for the number of layers and maximum number of filaments in a layer, respectively. Each cell is attributed a value of 0, 1, or 2 for the absence of a filament, the presence of a filament of material A or of a filament of material B, respectively. An example is given in Figure 6.
- The vertical cross section of the part (along the filament length) where the user wishes to know the temperature evolution with time.

The code includes one initial section where all the variables are defined (Figure 7), namely environment and extrusion temperatures, material properties, process conditions, etc. The dimensions of all matrixes used are also defined.



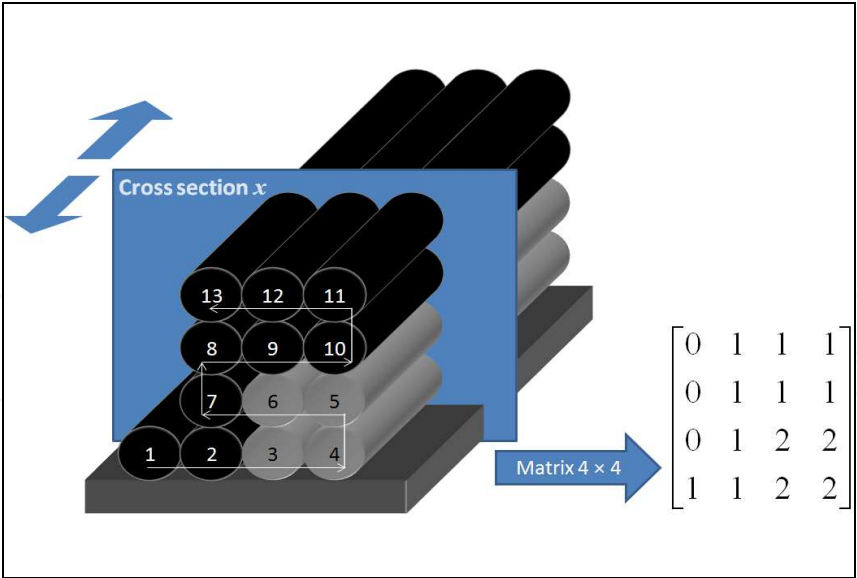


Fig. 6. Example of deposition sequence and corresponding input matrix.

```
function FFE(matrix,x)

% _____ STEP 1 _____
%Definition of the vector that contains the number of total filaments in each layer
matrix_lin = size(matrix,1);
matrix_col = size(matrix,2);
vector = zeros(matrix_lin,2);
contar = 0;
for i = matrix_lin:-1:1
    contar = contar + 1;
    for j = 1:matrix_col
        if matrix(i,j) ~= 0
            vector(contar,1) = vector(contar,1) + 1;
        end
    end
end
%Number of layers
m = length(vector(:,1));
%Number of filaments
n = 0;
for j = 1:m
    if m == 1
        n = vector(1,1);
    else
        if vector(j,2) <= 1
            n = n + vector(j,1);
        end
    end
end

% _____ STEP 2 _____
%Computation variables
passo = 0.05;           %Step time
temp_mais = 15;         %Additional time computation after construction of the part
erro = 0.001;           %Convergence error
```



```

% _____ STEP 3 _____
%Definition of the size of the variables
h = zeros(1,5); lambda = zeros(1,5); a = zeros(n,5); T = zeros (n,5);
vec_b = zeros(n,5); vec_Q = zeros(n,5); b = zeros(1,n); Q = zeros(1,n);
T_begin = zeros(1,n); dif = zeros(1,n); Biot = zeros(1,n); save_T = zeros(1,n);
old_T = zeros(1,n); save_lim = zeros(1,n); viz = zeros(11,n);

% _____ STEP 4 _____

%Process Variables
T_L = 270;           %Extrusion temperature (°C)
T_E = 70;           %Temperature of the envelope (°C)
v = 0.02;           %Velocity of the extrusion head (m/sec)
for lin = 1:n       %Temperature of support (°C)
    T(lin,1) = T_E;
end

%Filament dimensions
w = 0.0003;         %Layer Thickness (meters)
L = 0.02;           %Length of the filament (meters)
area = pi * (w/2)^2; %Area of the cross section of filament (meters^2)
per = pi * w;        %Perimeter of the cross section of filament (meters)
vol = area*L;        %Volume of the filament
A_p = per*L;         %Superficial area of the filament

% Material Properties

%Thermal conductivity (W/m.K)
conductivity(1) = 0.1768; % material A
conductivity(2) = 0.5;    % material B
%Density (kg/m^3)
ro(1) = 1050;            % material A
ro(2) = 1500;            % material B
%Specific heat (J/kg.K)
C(1) = 2019.7;           % material A
C(2) = 2500.7;           % material B

% _____ STEP 5 _____
% Heat transfer coefficient (lost of heat by natural convection)
h_conv = 45;

%Thermal contact conductances between
h(1,1) = 200;           % filament and left adjacent filament
h(1,2) = 200;           % filament and down adjacent filament
h(1,3) = 200;           % filament and right adjacent filament
h(1,4) = 200;           % filament and top adjacent filament
h(1,5) = 10;            % filament and support

%Fraction of perimeter contact between
lambda(1,1) = 0.2;      % filament and left adjacent filament
lambda(1,2) = 0.25;     % filament and down adjacent filament
lambda(1,3) = 0.2;      % filament and right adjacent filament
lambda(1,4) = 0.25;     % filament and top adjacent filament
lambda(1,5) = 0.25;     % filament and support

```

Fig. 7. Definition of the variables.

The parameters used in Equation (5) and those necessary to compute variables b and Q (in Eq. (3)) must also be defined. Finally, the time increment between two consecutive contacts is calculated taking into consideration the type of deposition sequence (Figure 8).

```

% _____ STEP 6 _____
%Definition of the parameters influenced by the contacts
for col = 1:5
    for lin = 1:n
        vec_b(lin,col) = h(1,col)*lambda(1,col);
        vec_Q(lin,col) = vec_b(lin,col)*T(lin,col);
    end
end

% _____ STEP 7 _____
%Definition of the parameters influenced by the material properties
contar = 0;
number_filament = 0;
for i = matrix_lin:-1:1
    contar = contar + 1;
    if isodd(contar) == 1
        for j = 1:matrix_col
            if matrix(i,j) ~= 0
                number_filament = number_filament + 1;
                escalar(number_filament) = -per/(ro(matrix(i,j))*area*C(matrix(i,j)));
                esc(number_filament) = h_conv/(ro(matrix(i,j))*L*C(matrix(i,j)));
                kt(number_filament) = conductivity(matrix(i,j));
            end
        end
    else
        for j = matrix_col:-1:1
            if matrix(i,j) ~= 0
                number_filament = number_filament + 1;
                escalar(number_filament) = -per/(ro(matrix(i,j))*area*C(matrix(i,j)));
                esc(number_filament) = h_conv/(ro(matrix(i,j))*L*C(matrix(i,j)));
                kt(number_filament) = conductivity(matrix(i,j));
            end
        end
    end
end
end

% _____ STEP 8 _____
%Definition of the periods of time between two successive contacts
for i = 1:(n+2)
    if isodd(i) == 1
        limite(i,1) = (i*L-x)/v;
        limite(i,2) = (i*L+x)/v;
    else
        limite(i,1) = limite(i-1,2);
        limite(i,2) = ((i+1)*L-x)/v;
    end
end
for road = 1:n
    linha = 0;
    for i = 0:passo:limite(n,2)
        linha = linha + 1;
        temp(linha,road) = T_L;
    end
end
end

```

Fig. 8. Definition of the parameters to be used for the computation of temperatures and time between two consecutive contacts.

4.2 Computation of the temperatures for the first filament of the first layer

Computation of the temperatures starts with the activation of the contact between the first filament and the support. Parameters b and Q (equation (3)) are calculated (Figure 9).

The temperatures are computed at each time increment; confirmation of the value of Biot number (Eq. (6)) is also made: if greater than 0.1, the code devolves a warning message (Figure 10).

```
for layer = 1:m
    if layer == 1
        for num = 1:vector(layer,1)
            if num == 1
                %----- STEP 9 -----
                a(num,5) = 1; %Activation of the contact with support
                %----- STEP 10 -----
                %Definition of the variables b and Q defined in equation Eq. 7
                b(num) = h_conv*(1-lambda*a(num,:)) + vec_b(num,:)*a(num,:);
                Q(num) = (h_conv*(1-lambda*a(num,:))*T_E +
                vec_Q(num,:)*a(num,:))/b(num);
```

Fig. 9. Activation of the contacts and computation of b and Q for the first filament.

```
%----- STEP 11 -----
p = 0;
for t = 0:passo:limite(num,1)
    p = p+1; abcissa(p) = t;
end

%----- STEP 12 -----
%Computation of the temperatures of the first filament
for t = (limite(num,1)+passo):passo:limite_final
    p = p+1; abcissa(p) = t;
    temp(p,num)=(T_L-Q(num))*exp(escalar(num)*b(num)*(t-limite(num,1)))
    +Q(num);
end
%Saving the last temperature of the period time of cooling down
T_begin(num) = temp(p,num);

%----- STEP 13 -----
%Verification of the value of Biot Number
Biot(num) = (vol/A_p)*(b(num)/kt(num));
if Biot(num)>=0.1
    'WARNING! We cannot use a Lumped System'
End
```

Fig. 10. Computation of the temperatures for the first filament and verification of the value of the Biot number.

4.3 Computation of the temperatures for the remaining filaments of the first layer

Before proceeding to the remaining filaments of the first layer, the lateral and support contacts for each filament being deposited must be defined, as well as for the previous one. Consequently, the variable b in expression Eq. (3) is up-dated (Figure 11).

At this point, only the lateral and support contacts must be defined, since only the first layer is being computed. For the remaining layers, other contacts (such as the vertical ones) must be considered. Once each filament is deposited, the code checks whether the part has been completed. If so, it remains in the same conditions during a pre-defined time, in order to reach the equilibrium temperature (Figure 12).

```

% _____ STEP 14 _____
else
    %Activation of the contacts
    a(num-1,3) = 1; a(num,1) = 1; a(num,5) = 1;

% _____ STEP 15 _____
    %Up-dating of the variable b
    for j = 1:num
        b(j) = h_conv*(1-lambda*a(j,:))' + vec_b(j,:)*a(j,:);
    end

```

Fig. 11. Activation of the contacts for the current and previous filaments and up-dating of variable b .

```

% _____ STEP 16 _____
if m == 1
    if num == vector(layer,1)
        limite_final = limite(num,2) + temp_mais;
    else
        limite_final = limite(num,2);
    end
else
    limite_final = limite(num,2);
end

```

Fig. 12. Definition of an additional time for the last filament.

Finally, the temperatures of the remaining filaments are computed. At each time increment, the temperatures of the adjacent filaments are saved and parameter Q (Eq. (3)) is up-dated. The value of the Biot number is checked before the deposition of a new filament (Figure 13).

```

for t = (limite(num,1)+passo):passo:limite_final
    p = p+1; abscissa(p) = t;
    last = p-1;
    for j = 1:num
        save_T(j) = temp(last,j);
    end

% _____ STEP 17 _____
    %Iterative process
    for q = 1:100000
        %Saving contacts and temperatures of adjacent filaments
        for j = 1:num
            if j == 1
                T(j,3) = save_T(j+1);
                viz(3,j) = j+1;
            end
            if j > 1 & j < num
                T(j,1) = save_T(j-1);
                viz(1,j) = j-1;
                T(j,3) = save_T(j+1);
                viz(3,j) = j+1;
            end
            if j == num
                T(j,1) = save_T(j-1);
                viz(1,j) = j-1;
            end
            for k = 1:5
                if T(j,k) ~= 0 & k ~= 5

```

```

        vec_Q(j,k) = vec_b(j,k)*T(j,k);
    end
end

%Up-dating of the variable Q
Q(j) = (h_conv*(1-lambda*a(j,:))'*T_E +
vec_Q(j,:)*a(j,:))'/b(j);
old_T(j) = save_T(j);
end
%Computation of the temperatures
if num == 2
    save_T(1) = (T_begin(1)-Q(1))*exp(escalar(1)*b(1)*
(t-limite(1,1)))+Q(1);
    save_T(2) = (T_L-Q(2))*exp(escalar(2)*b(2)*(t-
limite(1,1)))+Q(2);
    save_lim(1,1) = limite(num,1);
    save_lim(1,2) = limite(num,1);
else
    for j=1:num-2
        save_T(j) = (T_begin(j)-Q(1))*exp(escalar(j)*b(j)*
(t-save_limite(1,j)))+Q(j);
    end
    save_T(num-1) = (T_begin(num-1)-Q(num-1))*
exp(escalar(num-1)*b(num-1)*(t-limite(num,1)))+Q(num-
1);

    save_T(num) = (T_L-Q(num))*
exp(escalar(num)*b(num)*(t-limite(num,1)))+ Q(num);
    save_lim(1,num-1) = limite(num,1);
    save_lim(1,num) = limite(num,1);
end
for j = 1:num
    dif(j) = abs(save_T(j)-old_T(j));
end
try = 1;
stop = 0;
for j = 1:num
    if dif(try) < erro
        try = try+1;
    end
    if try == num+1;
        stop = 1;
    end
end
if stop == 1
    for j = 1:num
        temp(p,j) = save_T(j);
    end
    break;
end
end
end
T_begin(num) = temp(p,num);
%End of iterative process

% _____ STEP 18 _____
%Verification of the Biot Number
for j=1:num
    Biot(j) = (vol/A_p)*(b(j)/kt(j));
    if Biot(j)>=0.1
        'WARNING! We can not use a Lumped System'
        j
        Biot(j)
    end
end
end
end
end

```

```
end
end
```

Fig. 13. Computation of the temperature of the filaments of the first layer and verification of the Biot number.

5. Results

In order to demonstrate the usefulness of the code developed, two case studies will be discussed. The first deals with a part constructed with two distinct materials, while the second illustrates the role of the deposition sequence.

5.1 Case study 1

Consider the small part with the geometry presented in Figure 14, to be manufactured under the processing conditions summarized in Table 1.

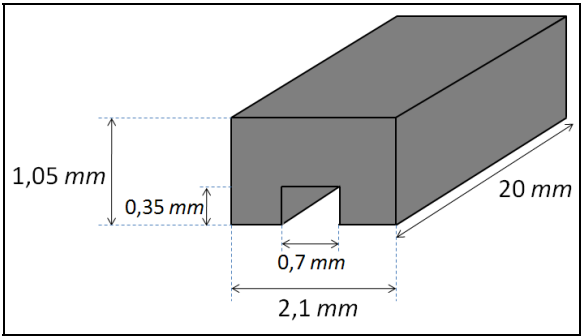


Fig. 14. Geometry of the part.

Property	Value
Extrusion temperature (°C)	270
Environment temperature (°C)	70
Extrusion velocity (m/s)	0.025
Filament length (m)	0.02
Cross section x (m)	0.01
Geometric form of cross section	circle
Cross section diameter (m)	0.00035
Contact ratio	88%
Heat transfer coefficient (convection) (W/m ² °C)	60
Thermal contact conductance with filaments (W/m ² °C)	180
Thermal contact conductance with support (W/m ² °C)	10
Thermal conductivity (W/m°C) materials A / B	0.1768 / 0.5
Specific heat (J/kg°C) materials A / B	2019.7 / 2500.7
Density materials A / B	1.05 / 1.5

Table 1. Processing conditions

The production of this part requires the use of a support material. Figure 15 shows the deposition sequence and corresponding material matrix, while Figure 16 presents the evolution of temperature of every filament with deposition time. As expected, once a new filament is deposited, the temperature of the preceding adjacent filaments increases and their rate of cooling decreases.

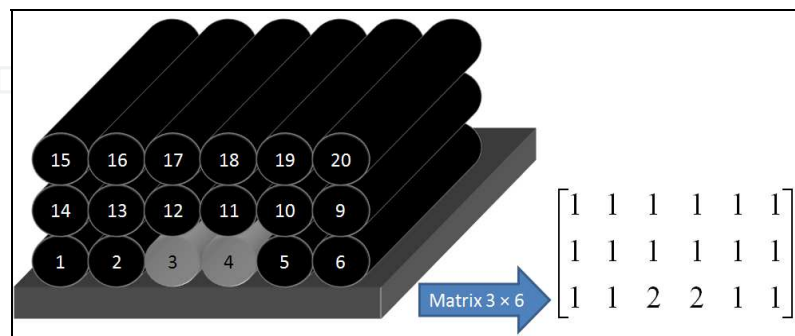


Fig. 15. Filaments deposition sequence and corresponding material matrix.

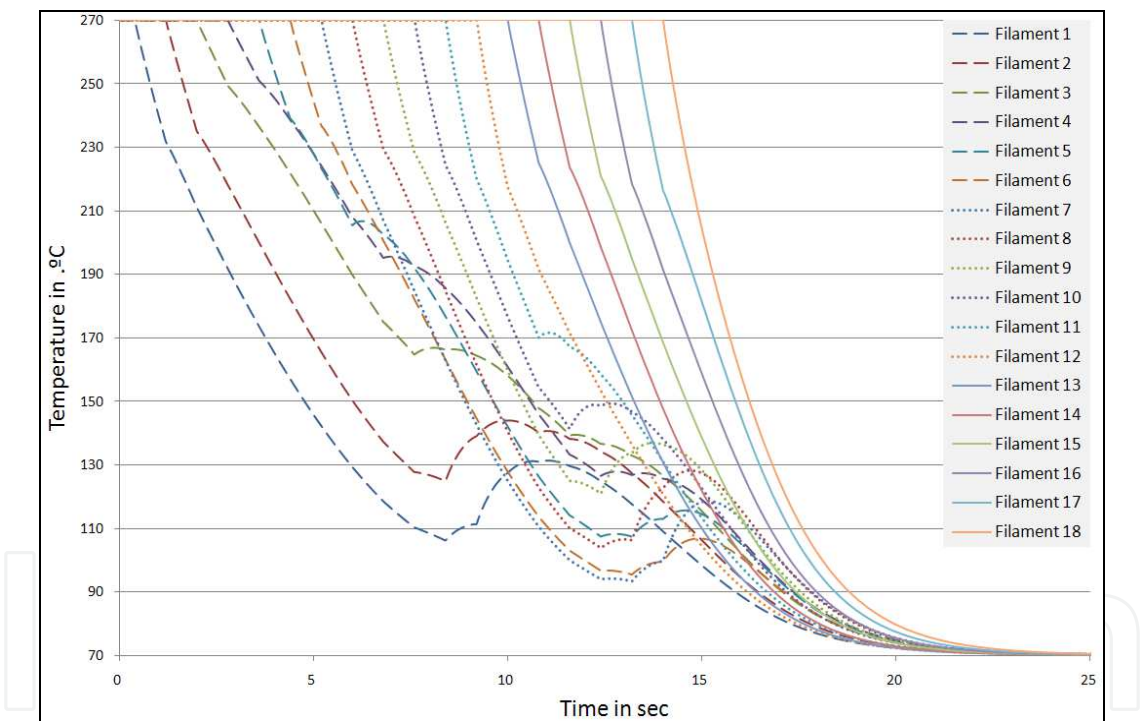


Fig. 16. Temperature evolution with time (at $x = 0.01\text{ m}$), for the deposition sequence illustrated in Figure 15.

5.2 Case study 2

Consider now the parallelepipedic part depicted in Figure 17, to be built using unidirectional and aligned and perpendicular sequences, respectively, under the processing conditions summarized in Table 2.

Figures 18 and 19 depict the deposition sequence and corresponding temperatures (this required an additional part of the code together with the use of the Tecplot software). At each time increment, a 1 mm or a 0.35 mm filament portion was deposited, for

unidirectional and aligned and perpendicular filaments, respectively. This lower value is related with the lower contact area arising from this deposition mode. Consequently, the total computation time was circa 7 minutes for unidirectional and aligned deposition and more than two and a half hours for perpendicular filaments for a conventional portable PC. As the manufacture is completed ($t = 14.4$ sec), the average part temperature is approximately 120 °C or 90 °C depending on the deposition mode. This information is relevant for practical purposes, such as evaluating the quality of the adhesion between adjacent filaments, or the extent of deformation.

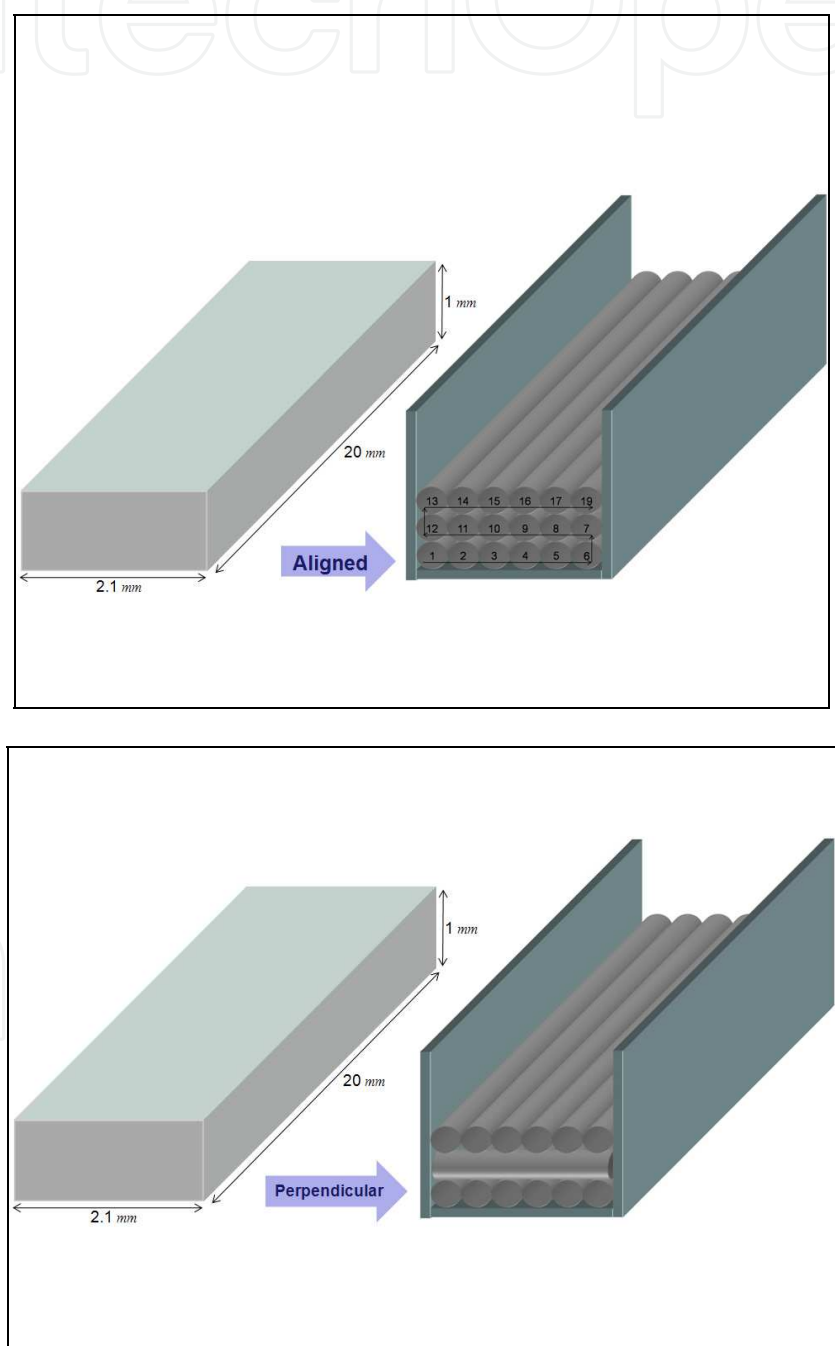
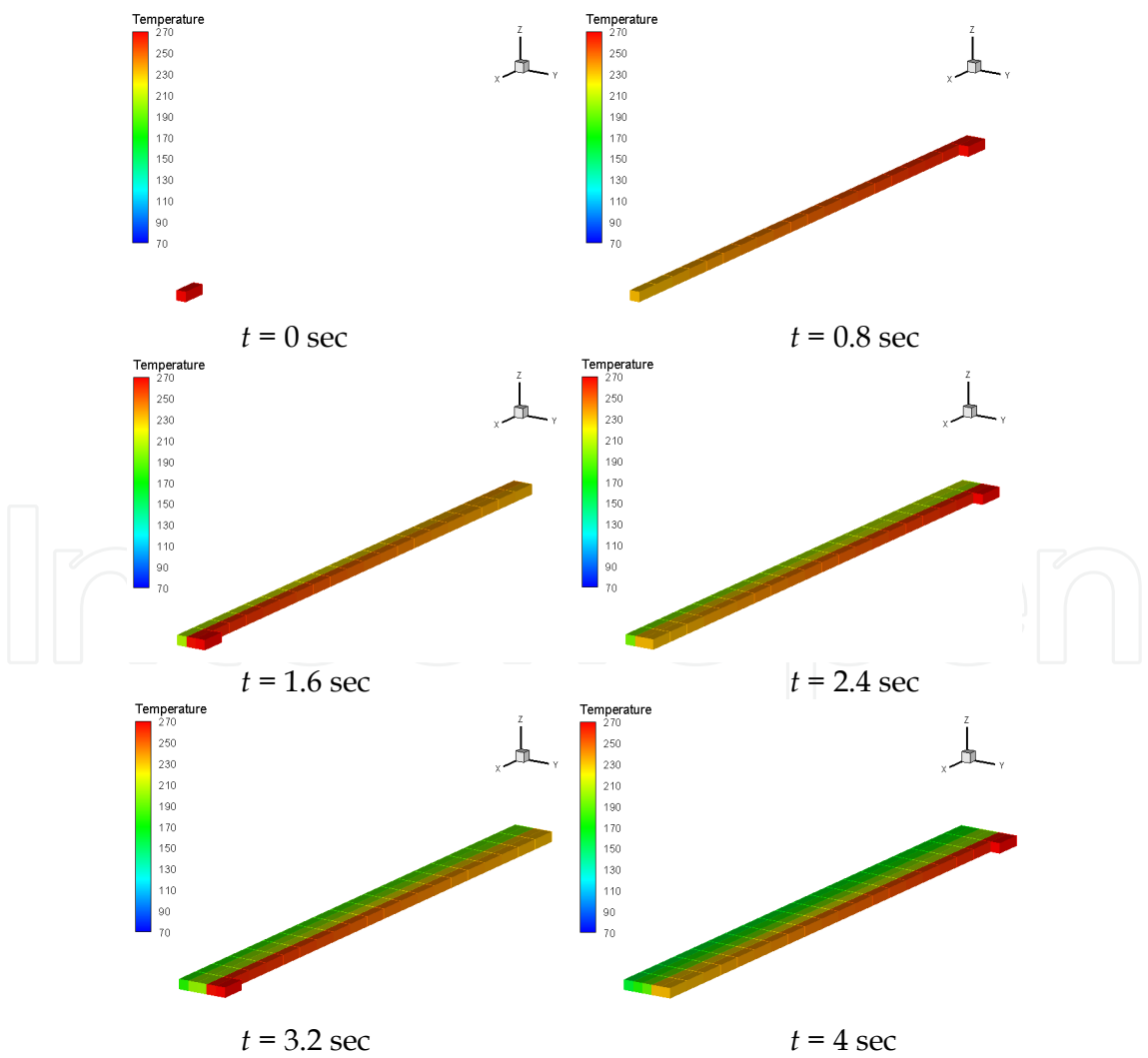
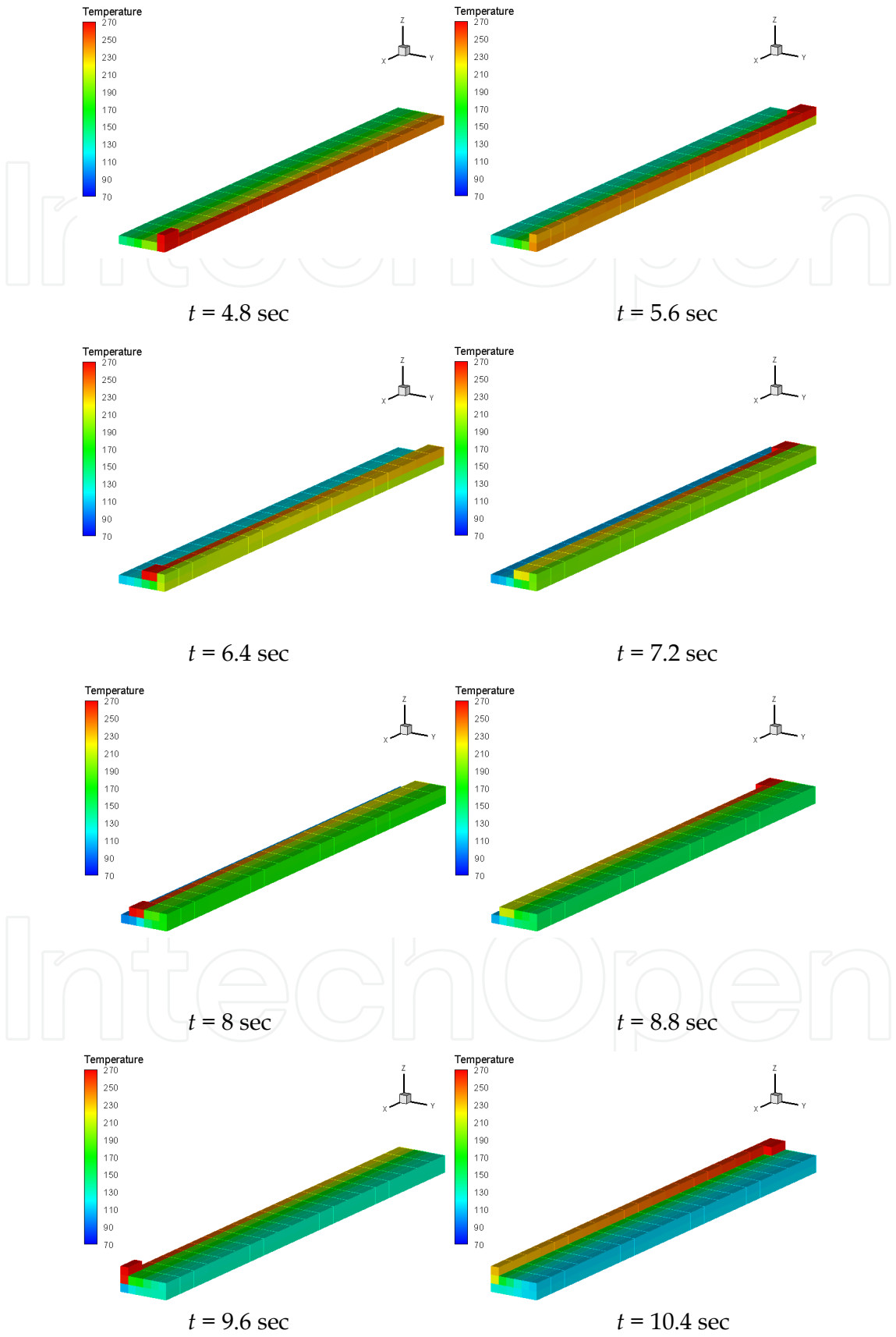


Fig. 17. Geometry of the part and corresponding deposition sequence: top: unidirectional and aligned; bottom: perpendicular.

Property	Value
Extrusion temperature (°C)	270
Environment temperature (°C)	70
Extrusion velocity (m/s)	0.025
Filament length (m)	0.02
Geometric form of cross section	circle
Cross section diameter (m)	0.00035
Contact ratio	88%
Heat transfer coefficient (convection) (W/m ² °C)	70
Thermal contact conductance with filaments (W/m ² °C)	200
Thermal contact conductance with support (W/m ² °C)	15
Thermal conductivity (W/m°C)	0.1768
Specific heat (J/kg°C)	2019.7
Density	1.05

Table 2. Processing conditions





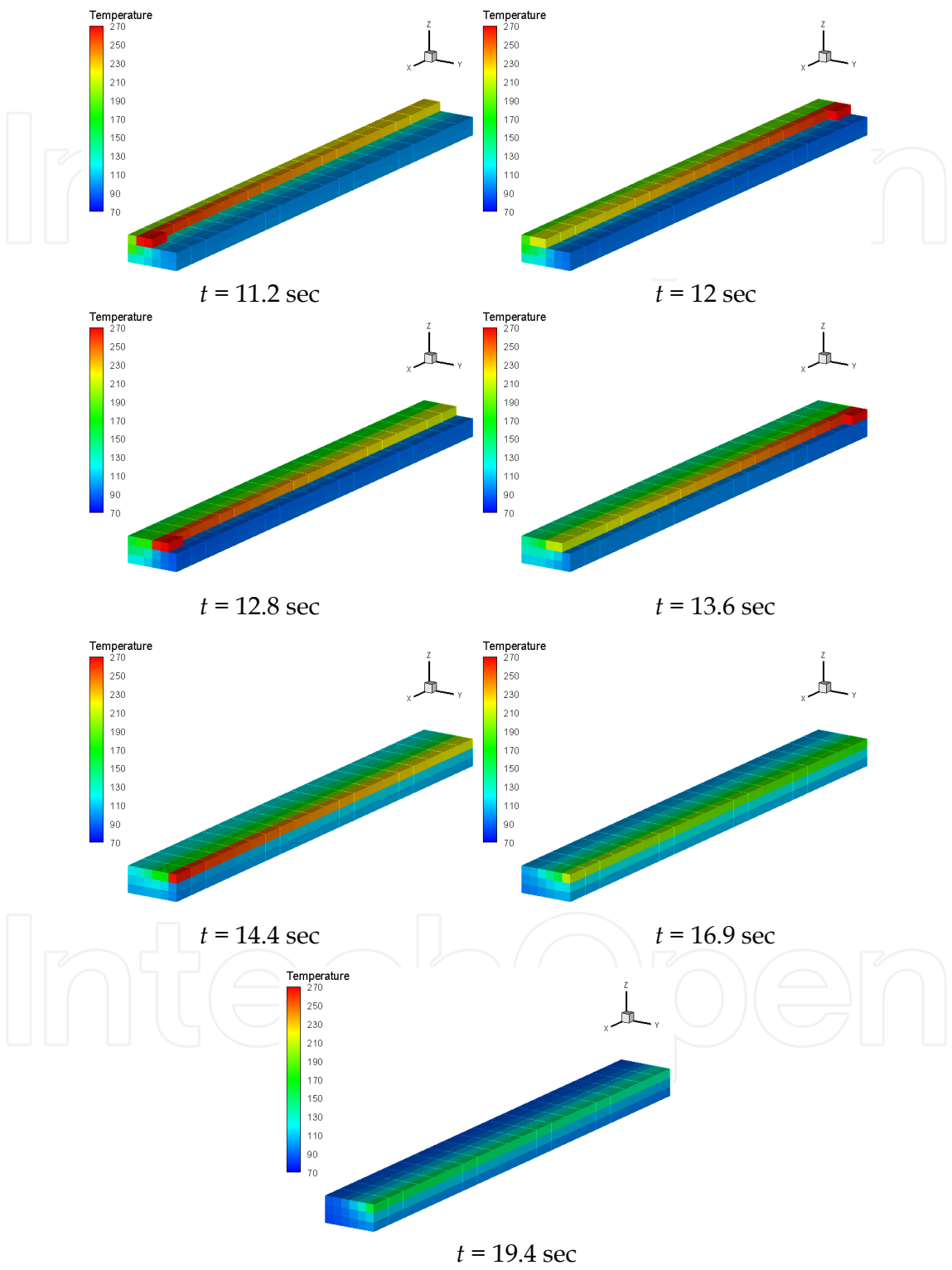
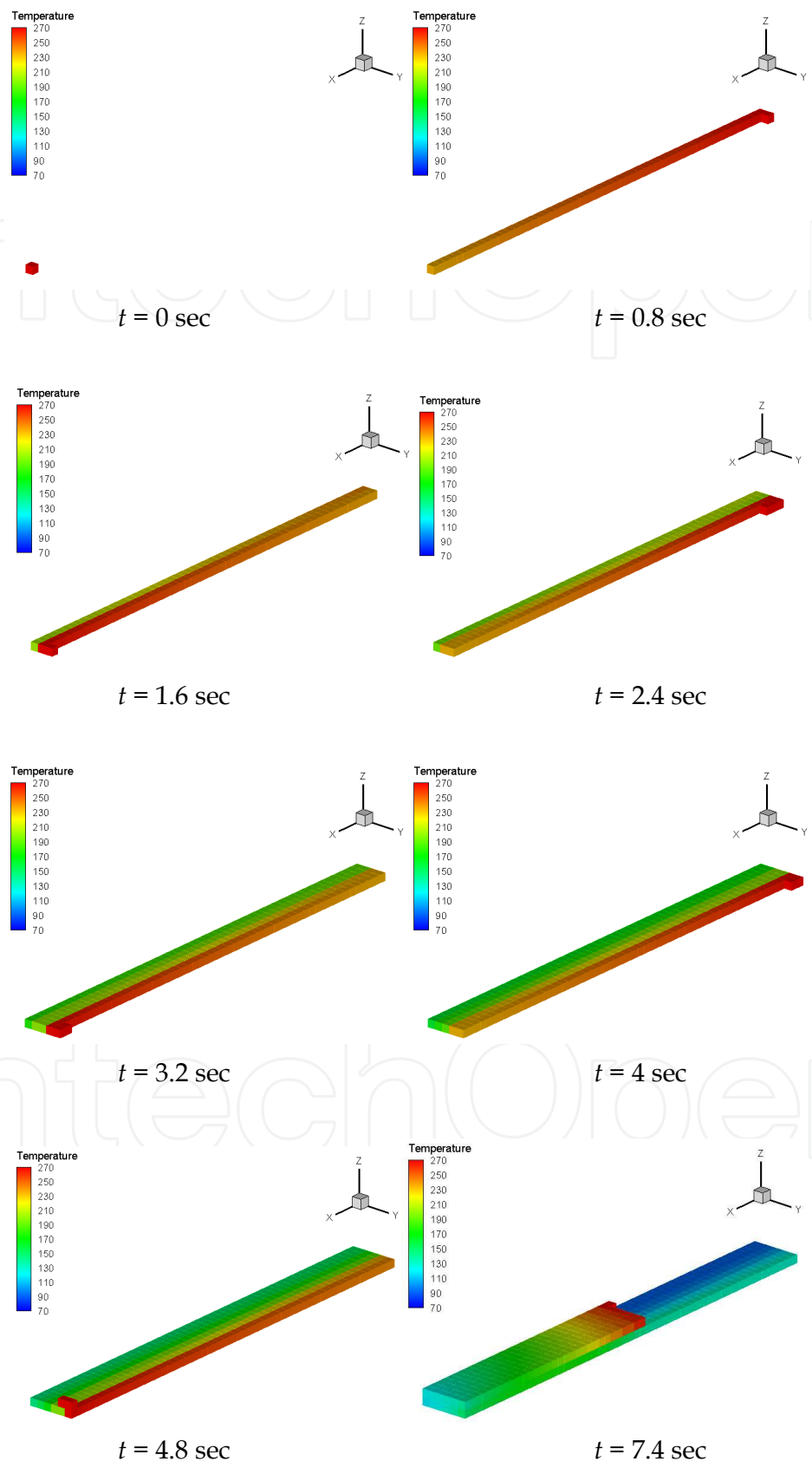
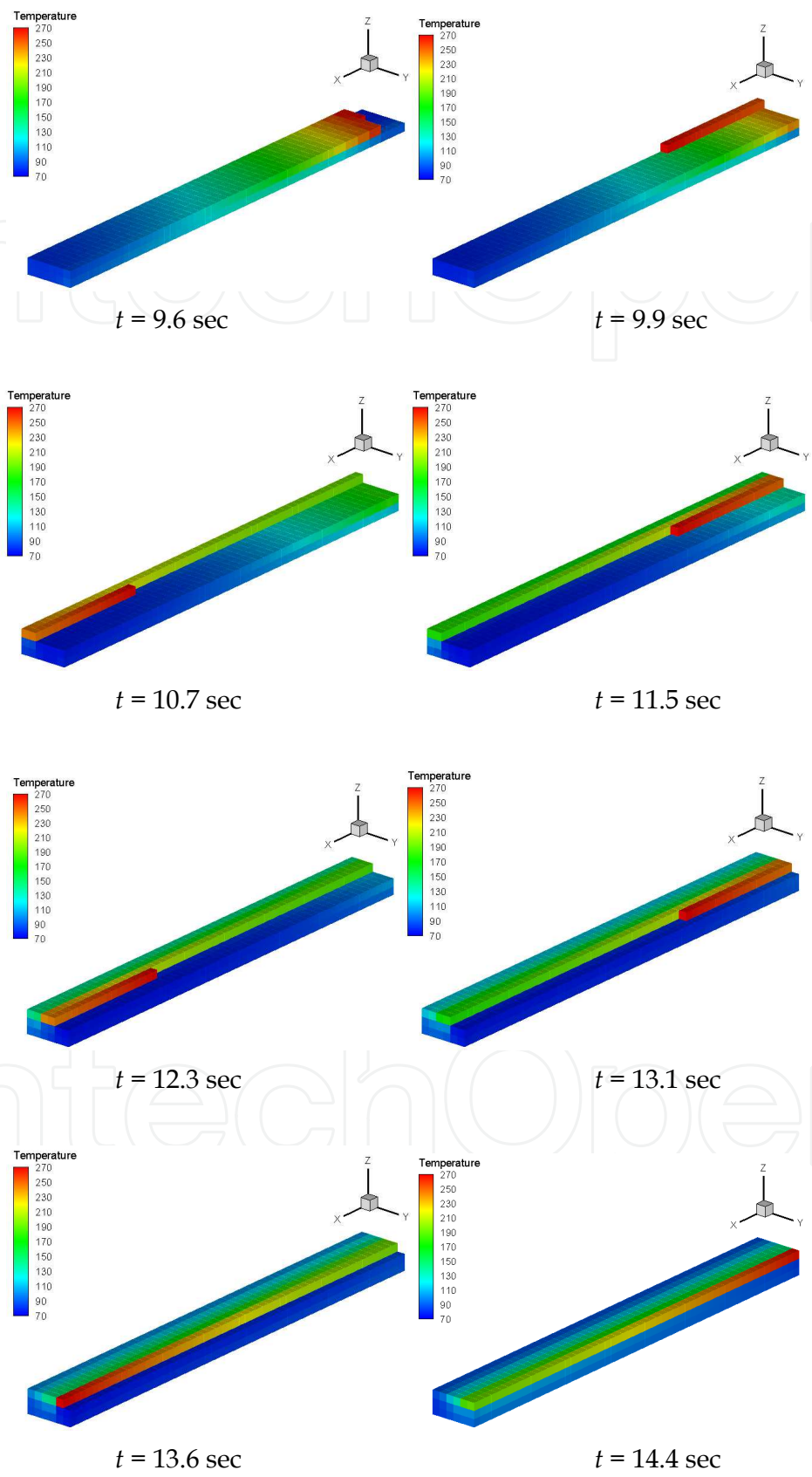


Fig. 18. Deposition sequence of the part of Figure 17 (unidirectional and aligned filaments).





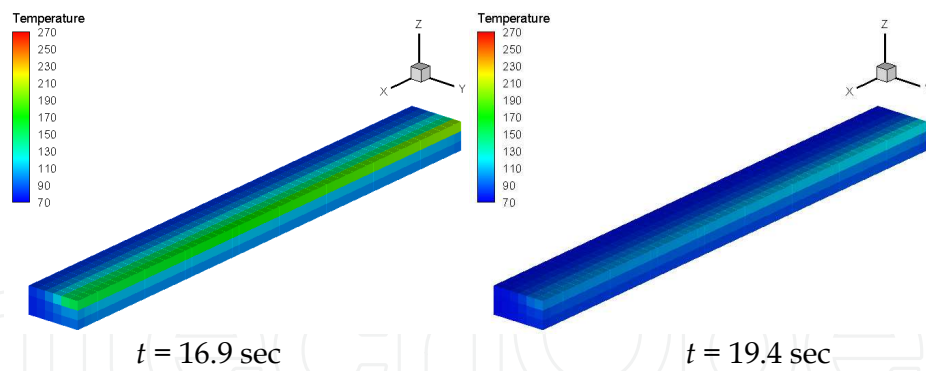


Fig. 19. Deposition sequence of the part of Figure 17 (perpendicular filaments).

6. Conclusion

In Free Form Extrusion, FFE, a molten filament is deposited sequentially to produce a 3D part without a mould. This layer by layer construction technique may create problems of adhesion between adjacent filaments, or create dimensional accuracy problems due to excessive deformation of the filaments, if the processing conditions are not adequately set. This chapter presented a MatLab code for modelling the heat transfer in FFE, aiming at determining the temperature evolution of each filament during the deposition stage. Two case studies illustrated the use of the programme.

7. References

- Rodriguez, J. F. (1999). *Modelling the mechanical behaviour of fused deposition acrylonitrile-butadiene-styrene polymer components*, Ph.D. Dissertation, Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, USA
- Thomas, J. P. & Rodríguez, J. F. (2000). *Modeling the fracture strength between fused deposition extruded roads*, Solid Freeform Fabrication Symposium Proceeding, Austin.
- Yardimci, M. A. & Guceri, S. I. (1996). *Conceptual framework for the thermal process modelling of fused deposition*, Rapid Prototyping Journal, 2, 26-31.
- Li, L.; Sun, Q.; Bellehumeur, C. & Gu, P. (2003). *Modeling of bond formation in FDM process*, Trans. NAMRI/SME, 8, 613-620.
- Sun, Q.; Rizvi, G.C.; Bellehumeur, C. & Gu, T. P. (2004). *Experimental study and modeling of bond formation between ABS filaments in the FDM process*, Proc. SPE ANTEC'2004.
- Costa, S.; Duarte, F. & Covas, J. A. (2008). *Towards modelling of Free Form Extrusion: analytical solution of transient heat transfer*, Esaform 2008, Lyon, France.
- Bejan, A. (1993). *Heat Transfer*, John Wiley & Sons, Inc., New York.
- Marsan, A.; Dutta, D. (2000). *A review of process planning techniques in layered manufacturing*, Rapid Prototyping Journal, Vol.6, No.1, pp. 18-35, ISSN 1355-2546.
- Pérez, C. J. L. (2002). *Analysis of the surface roughness and dimensional accuracy capability of fused deposition modelling processes*, International Journal of Production Research, Vol.40, Issue 12, pp. 2865 – 2881, ISSN 1366-588X.
- Ahn, S. H. (2002). *Anisotropic material properties of fused deposition modeling ABS*, Rapid Prototyping Journal, Vol.8, No.4, pp. 248-257, ISSN 1355-2546.
- Ziemian, C. W. & Crown, P. M. (2001). *Computer aided decision support for fused deposition modeling*, Rapid Prototyping Journal, Vol.7, No.3, pp. 138-147, ISSN 1355-2546.

- Agarwala, M. K.; Jamalabad, V. R.; Langrana, N. A.; Safari, A.; Whalen, P. J. & Danthord, S. C. (1996). *Structural quality of parts processed by fused deposition*, Rapid Prototyping Journal, Vol.2, No.4, pp. 4-19, ISSN 1355-2546.
- Bellini, A.; Shor, L. & Gucer, S. (2005). *New developments in fused deposition modeling of ceramics*, Rapid Prototyping Journal, Vol.11, No.4, pp. 214-220, ISSN 1355-25.

IntechOpen

IntechOpen



MATLAB - A Ubiquitous Tool for the Practical Engineer

Edited by Prof. Clara Ionescu

ISBN 978-953-307-907-3

Hard cover, 564 pages

Publisher InTech

Published online 13, October, 2011

Published in print edition October, 2011

A well-known statement says that the PID controller is the “bread and butter” of the control engineer. This is indeed true, from a scientific standpoint. However, nowadays, in the era of computer science, when the paper and pencil have been replaced by the keyboard and the display of computers, one may equally say that MATLAB is the “bread” in the above statement. MATLAB has become a de facto tool for the modern system engineer. This book is written for both engineering students, as well as for practicing engineers. The wide range of applications in which MATLAB is the working framework, shows that it is a powerful, comprehensive and easy-to-use environment for performing technical computations. The book includes various excellent applications in which MATLAB is employed: from pure algebraic computations to data acquisition in real-life experiments, from control strategies to image processing algorithms, from graphical user interface design for educational purposes to Simulink embedded systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sidonie Costa, Fernando Duarte and José A. Covas (2011). Using MATLAB to Compute Heat Transfer in Free Form Extrusion, MATLAB - A Ubiquitous Tool for the Practical Engineer, Prof. Clara Ionescu (Ed.), ISBN: 978-953-307-907-3, InTech, Available from: <http://www.intechopen.com/books/matlab-a-ubiquitous-tool-for-the-practical-engineer/using-matlab-to-compute-heat-transfer-in-free-form-extrusion>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen