Layerwise Additive Manufacturing Predictions and Simulations

(LAMPS version 2.0)

User's Manual

-Operation-

Contact:

Jinquan Cheng

3315 S Waterfront Dr.

Chandler, AZ, 85248

Phone: 480-725-8068

E-mail: Jinquan.cheng@cs3dm.com

hitcjq@hotmail.com

http://www.cs3dm.com

Contents

1. Fundamentals and Features of the Software

1.1. Background and Significance of the LAMPS Software

It is well-known that the general additive manufacturing is a procedure of line-by-line and layer-by-layer directly depositing the host materials by a heat source, such as laser and electron beam etc., to fabricate the final desired part from a digital design file. Thus, the procedure is a problem of a complex heat transfer with phase change [1]. However, the procedure involves more than 50 factors having a certain effect on the temperature, temperature gradient distribution and cooling rate in the AM part from the environment, the deposition material properties, the powder granulometry and deposition, the powder bed properties, the heat source properties and qualities and the processing parameters as shown in Table 1.1 [2]. So many effect factors significantly induced the much difficulty to control the quality of as-deposited part. For examples, as revealed by the experiments, the oxygen percentage in the deposition chamber has a significant effect on the surface tension of the melt pool and results in different microstructure solidifications [3]; For the powder bed thermophysical properties, it is evident that the loose powder's heat conductivity is much lower than bulky materials, which results in the localized temperature gradient change [4]; For the deposition location effect, it is clearly shown from the x-ray experimental results that there are over 70% defects occurred nearby the free-surface[5].

Even though there are lots of experiments to study the individual factor effect respectively, it is still difficult to outline a feasible processing parameter set to get a high quality as-deposited part since so many factors involving the processes make the extremely complexity of heat transfer problem in AM, as indicated in Table 1.1. Thus, lots of numerical tools / software have been developed to analyze the AM procedure. However, since the thickness of the depositing layer could be several micrometers to hundred micrometers, one Additive Manufacturing part generally needs thousands of such a layer to be deposited. Thus, it becomes very difficult to use element-based numerical tools to emulate the whole additive manufacturing procedure in the line-by-line and layer-by-layer due to the limitation of the element-size and element-number in the current available simulation tools. Therefore, how to envision the effect of those processing parameters on the microstructure evolution and further predict/determine the whole part quality becomes an impossible mission. But, it is more and more demanded in the additive manufacturing industries. Under such a kind of struggling ambient, the layer-wise additive manufacturing predictions and simulations (L.A.M.P.S.) software was born to provide a way enables to predict and simulate the whole additive manufacturing procedure from the line-by-line and layer-by-layer to the final part based on an analytical tool and block technique. In the version 1 of the LAMPS software, it can predict and analyze the effect of the key processing parameters, including scan speed, input energy, deposition location and deposit pattern etc., deposition material properties, baseplate material properties and geometries etc. on the distribution of the temperature and temperature gradient, cooling rate in the whole part without the simulation size limitation during the depositing.

Table 1.1: List of the effect factors on the part quality in Additive Manufacturing

Sources	Effect Factors
Environments	Inert gas, Molecular mass, Viscosity, Thermal conductivity, Heat exchange coefficient of convection, Thermal capacity, Pressure, Oxygen level %O2, Ambient temperature, Surface free energy.
Powder composition: material	Bulk density, Thermal conductivity, Thermal capacity, Latent heat of fusion, Melting temperature, Boiling temperature, Dynamic viscosity, Thermal expansion coefficient, Surface free energy, Vapour pressure, Reaction energy, Absorptivity, Diffusion coefficient, Solubility, Melt enthalpy, Pollution
Powder granulometry and deposition	Morphology, Surface roughness, Particle size distribution, Depositioning system
Powder bed properties	Density, Conductivity, Absorptivity, Emissivity, Diffusivity,
Laser based heat source	Mode, Wave length, Intensity profile, Average power, Peak power, Beam quality, Frequency, Pulse width, *Spot size dx & dy,* Polarity
Processing parameters	*Scan Strategy, Scan spacing, Scan speed, Energy Density, Layer thickness, 3D environment*

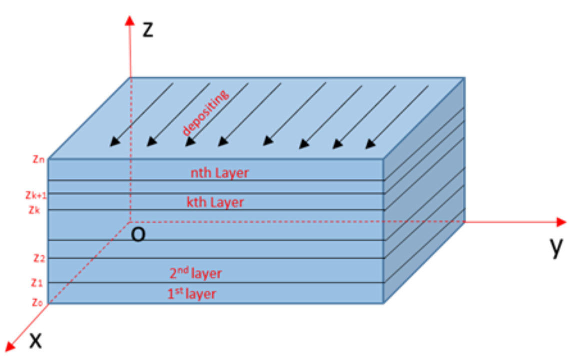


Figure 1-1: Schematic view of layerwise additive manufacturing model

As is well known, the irregular temperature gradient and temperature cooling rate always exists during the additive manufacturing processing which can results from the irregular part geometry/shape, input energy fluctuation, dissimilar deposition, scan speed, or scan strategy etc intrinsic factors, or the layer thickness, powder particle radius distribution etc. extrinsic factors. The intrinsic and extrinsic factors induced irregular temperature field will always result in the distributed localized deformation in the final deposited part, which is the stem of the part distortion and residual stress[7-11]. Thus, the distortions and residual stresses always exist for the as-deposited part in a high level or low level value, which does not matter which type of materials for deposition, such as polymer or metal. Thus, how to emulate the additive manufacturing procedure and optimize the processing parameters to achieve the minimum distortion and residual stress for the as-deposited part becomes the top concern for designing the deposit path and the processing parameters. In Version 2 of LAMPS, we provide the thermo-mechanical analyses for the whole part. Here, we start the heat transfer problem in additive manufacturing to obtain the full-scale temperature field throughout the whole deposition procedure and further simulate the mechanical fields for the deposition in terms of the resulted temperature field. Using our patents for quick full-scale field simulation techniques, it is convenient to realize the quick design and analysis of the thermo-mechanical problem in Version 2 of LAMPS. The following sections present the fundamental of the thermo-mechanical model for the software and its features for additive manufacturing.

1.2. Fundamental of the Thermo-Mechanical Model for Additive Manufacturing

This section is showing the fundamental equations to describe the layerwise thermo-mechanical model for simulating and predicting the effect of processing parameters on the distortion and residual stresses of as-deposited part due to the irregular temperature field.

1.2.1. Heat Transfer Governing Equations with the Relevant Boundary Conditions

1.2.1.1. Heat Transfer Problem

In terms of the additive manufacturing fabrication characteristics, a layerwise additive manufacturing heat conducting/transfer model as shown in Figure 1.1, was developed to understand the effect of such many process parameters on the temperature and temperature gradient distribution and cooling rate. The heat transfer governing equation for such a laminated plate can be described by

$$\frac{\partial}{\partial x}\left(k\frac{\partial T(x,y,z,t)}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T(x,y,z,t)}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T(x,y,z,t)}{\partial z}\right)+q_{int}=\rho C\frac{\partial T(x,y,z,t)}{\partial t} \quad (1\text{-}1a)$$

or

$$\nabla\cdot(k\nabla T)=f \text{ on V} \quad (1\text{-}1b)$$

with the relevant boundary conditions are listed as follows:

$$T=T_1 \text{ on } S_1 \quad (1\text{-}2a)$$

$$k\tilde{n}\cdot\nabla T=q_s \text{ on } S_2 \quad (1\text{-}2b)$$

$$-k\tilde{n}\cdot\nabla T=h(T-T_\infty)+r(T^4-T_\infty^4) \text{ on } S_3 \quad (1\text{-}2c)$$

where $T(x,y,z,t)$ is the temperature of the plate. K, ρ and C is the thermal conductivity, density and specific heat capacity of the plate, which are dependent on the location. q_{int} is referring to the internal heat source and q_s is the surface applied heat flux. In version 1.0, the radiation effect is neglected.

1.2.1.2. Application to Additive Manufacturing Procedure

It is well-known that for one additive manufacturing part, its final temperature should be a steady-state temperature which is denoted by a function of T_d (x, y, z). Thus, it should satisfy the steady-state heat transfer equation as

$$\frac{\partial}{\partial x}\left(k\frac{\partial T(x,y,z,t)}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T(x,y,z,t)}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T(x,y,z,t)}{\partial z}\right)+q_{int}=0 \quad (1\text{-}3)$$

And

$$\frac{\partial T_d(x,y,z,t)}{\partial t}=0 \quad (1\text{-}4)$$

Based on Eq. (1-2) and consider Eq. (1-3) one can obtain the dynamics in temperature difference $\theta(x,y,z,t)=T(x,y,z,t)-T_d(x,y,z)$ as follows,

$$\frac{\partial}{\partial x}\left(k\frac{\partial \theta(x,y,z,t)}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial \theta(x,y,z,t)}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial \theta(x,y,z,t)}{\partial z}\right)+q_{int}=\rho C\frac{\partial \theta(x,y,z,t)}{\partial t} \quad (1\text{-}5)$$

In terms of the additive manufacturing line-by-line and layer-by-layer deposition characteristics, a laminated plate as shown in Figure 1-1 can be used to describe and analyze the deposition. Here the governing equation becomes

$$\frac{\partial}{\partial x}\left(k_{xi}\frac{\partial \theta(x,y,z,t)}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_{yi}\frac{\partial \theta(x,y,z,t)}{\partial y}\right)+\frac{\partial}{\partial z}\left(k_{zi}\frac{\partial \theta(x,y,z,t)}{\partial z}\right)=\rho_i C_i\frac{\partial \theta(x,y,z,t)}{\partial t} \quad (1\text{-}6)$$

where the subscript " *i*" denotes the *i*th layer and the subscripts "x, y, z" denote the values for x, y and z-axes respectively. And the additional continuity conditions will be applied to the interface between two adjacent layers, *i.e.* $Z_i= b_i$ and $Z_{i+1}=0$ as follows:

$$\theta_i=\theta_{i+1} \quad (1\text{-}7a)$$

$$k_i^z\frac{\partial \theta_i}{\partial z_i}=k_{i+1}^z\frac{\partial \theta_{i+1}}{\partial z_{i+1}} \quad (1\text{-}7b)$$

It is seen that the developed layerwise additive manufacturing model can handle the different materials deposit in the whole part based on Equation (1-6) and (1-7). Using the governing Equation (1-6) and continuity conditions Equation (1-7), the LAM model can be analytically solved based on the series expansion. On the other hand, a block technique(CS3DM patent) can deal with the connection between the baseplate and the deposited section with the different geometries in terms of the basis solution.

1.2.2. Layerwise Plate model for Thermo-mechanical Problem of Additive Manufacturing

1.2.2.1. Governing Equations

As shown in Figure 1-1, the additive manufacturing procedure can be modeled by using laminated plate problem for its layer by layer deposition. Thus, the equations of the motion for each lamina can be described by

$$\frac{\partial \sigma_x}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}=\rho\frac{\partial^2 u}{\partial t^2} \quad (1\text{-}8a)$$

$$\frac{\partial \tau_{xy}}{\partial x}+\frac{\partial \sigma_{y}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}=\rho\frac{\partial^2 v}{\partial t^2} \quad (1\text{-}8b)$$

$$\frac{\partial \tau_{xz}}{\partial x}+\frac{\partial \tau_{yz}}{\partial y}+\frac{\partial \sigma_{z}}{\partial z}=\rho\frac{\partial^2 w}{\partial t^2} \quad (1\text{-}8c)$$

where u, v and w is the displacements along the x-, y- and z- axes. ρ is the mass density. These variables are referred to a lamina.

1.2.2.2. Strain-Displacement Relationship for a Lamina

Generally, the total strain can be obtained from the total displacement change along different directions as follows

$$\varepsilon_x=\frac{\partial u}{\partial x}, \quad (1\text{-}9a)$$

$$\varepsilon_y=\frac{\partial v}{\partial y}, \quad (1\text{-}9b)$$

$$\varepsilon_z=\frac{\partial w}{\partial z}, \quad (1\text{-}9c)$$

$$\gamma_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\ , \quad (1\text{-}9d)$$

$$\gamma_{yz}=\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}, \quad (1\text{-}9e)$$

$$\gamma_{xz}=\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x} \quad (1\text{-}9f)$$

where the total strain include the thermal expansion deformation and inherent shrinkage due to the solidification etc..

1.2.2.3. General Stress-strain Relationship of a Lamina

In general, considering the temperature change effect, the elastic stress-strain relationship for a monoclinic lamina can be described by following equation with the temperature change-induced thermal strain:

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \tau_{yz}\\ \tau_{xz}\\ \tau_{xy}\end{Bmatrix}_k=\begin{bmatrix}C_{11}&C_{12}&C_{13}&0&0&C_{16}\\ C_{12}&C_{22}&C_{23}&0&0&C_{26}\\ C_{13}&C_{23}&C_{33}&0&0&C_{36}\\ 0&0&0&C_{44}&C_{45}&0\\ 0&0&0&C_{45}&C_{55}&0\\ C_{16}&C_{26}&C_{36}&0&0&C_{66}\end{bmatrix}_k\begin{Bmatrix}\varepsilon_x^e\\ \varepsilon_y^e\\ \varepsilon_z^e\\ \gamma_{yz}^e\\ \gamma_{xz}^e\\ \gamma_{xy}^e\end{Bmatrix}_k \quad (1\text{-}10a)$$

$$\begin{Bmatrix}\varepsilon_x^e\\ \varepsilon_y^e\\ \varepsilon_z^e\\ \gamma_{yz}^e\\ \gamma_{xz}^e\\ \gamma_{xy}^e\end{Bmatrix}_k=\begin{Bmatrix}\varepsilon_x-\alpha_x\Delta T\\ \varepsilon_y-\alpha_y\Delta T\\ \varepsilon_z-\alpha_z\Delta T\\ \gamma_{yz}\\ \gamma_{xz}\\ \gamma_{xy}\end{Bmatrix}_k \quad (1\text{-}10b)$$

where the superscript "e" denotes the elastic deformation, and the subscript "k" denotes the kth layer. $\alpha_i (i = x, y, z)$ is the thermal expansion coefficients of the lamina, which can be different from each direction for one anisotropic materials.

However, for the additive manufacturing, the temperature at the deposition point always exceeds the melting point temperature. After cooling, the shrinkage will always happen to the deposition segment. Thus, the stress-strain relationship for additive manufacturing must consider the fact of inherent shrinkage after deposition and cooling down to room temperature.

The following formulations present the stress-strain relationship for the different types of material properties as

For an orthotropic material:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}_k = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_x^e \\ \varepsilon_y^e \\ \varepsilon_z^e \\ \gamma_{yz}^e \\ \gamma_{xz}^e \\ \gamma_{xy}^e \end{Bmatrix}_k \tag{1-11}$$

For a transversely isotropic material:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}_k = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{12} & C_{23} & C_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{55} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_x^e \\ \varepsilon_y^e \\ \varepsilon_z^e \\ \gamma_{yz}^e \\ \gamma_{xz}^e \\ \gamma_{xy}^e \end{Bmatrix}_k \tag{1-12}$$

where the C_{22} is not an independent constant for a transversely isotropic material and can be obtained by

$$C_{22} = C_{23} + 2C_{44}$$

For a cubic material

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}_k = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_x^e \\ \varepsilon_y^e \\ \varepsilon_z^e \\ \gamma_{yz}^e \\ \gamma_{xz}^e \\ \gamma_{xy}^e \end{Bmatrix}_k \tag{1-13}$$

where there are only three independent constants for such a type material.

For an isotropic material

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{Bmatrix}_k = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix}_k \begin{Bmatrix} \varepsilon_x^e \\ \varepsilon_y^e \\ \varepsilon_z^e \\ \gamma_{yz}^e \\ \gamma_{xz}^e \\ \gamma_{xy}^e \end{Bmatrix}_k \tag{1-14}$$

where C_{11} is not independent constant for the isotropic material and presented by the following the relationship as

$$C_{11} = C_{12} + 2C_{44} \tag{1-15}$$

Thus, there are only two independent constants for the isotropic materials.

1.2.2.4. Inherent Shrinkage

Due to the melting and solidification during the additive manufacturing, the shrinkage always happens to the deposition segment.[reference] For the inherent shrinkage, a four-linear-line inherent shrinkage (plastic) deformation model is introduced to describe it based on the temperature change and two inherent yield temperatures T_{y1} and T_{y2} as shown in Figure 1-2: (1) while temperature change from T_0 does not exceed the first yielding temperature T_{y1}, there is not shrinkage occurrence, *i.e.* Segment ❶. (2) If the temperature exceeds the first yielding temperature T_{y1} and reaches the maximum temperature T_{max}, the shrinkage strain is linearly related to the difference between the temperature and the first yielding temperature, *i.e.* Segment ❷. (3) While temperature decreases from the T_{max} to the second yielding T_{y2}, the shrinkage strain will be kept same to the shrinkage at T_{max}, *i.e.* Segment ❸. (4). If the temperature reduces from the T_{y2} to the T_0, there is a permanent shrinkage strain existing as ε_p^*, *i.e.* Segment ❹. Thus, considering the effect of inherent shrinkage in additive manufacturing, the elastic strain in Equation (1-10b) has to be rewritten as following format:

$$\begin{Bmatrix} \varepsilon_x^e \\ \varepsilon_y^e \\ \varepsilon_z^e \\ \gamma_{yz}^e \\ \gamma_{xz}^e \\ \gamma_{xy}^e \end{Bmatrix}_k = \begin{Bmatrix} \varepsilon_x - \alpha_x \Delta T - \varepsilon_{px}^* \\ \varepsilon_y - \alpha_y \Delta T - \varepsilon_{py}^* \\ \varepsilon_z - \alpha_z \Delta T - \varepsilon_{pz}^* \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{Bmatrix}_k \tag{1-10b}$$

where the $\varepsilon_{pi}^*(i = x, y, z)$ denotes the permanent shrinkages(plastic deformation) along the x-,y- and z-direction in the local coordination system.

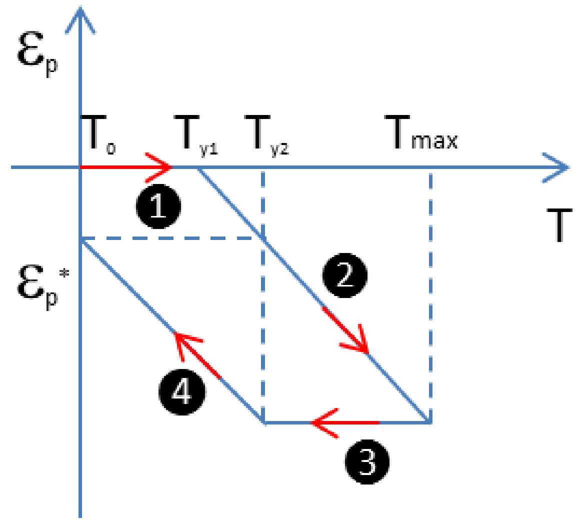


Figure 1-2: A four-linear line model for the inherent shrinkage due to solidification.

1.2.2.5. Interfacial Continuous Condition

As indicated above, the additive manufacturing processing is a procedure of layer-by-layer deposition in the baseplate. Thus, for a perfect bonded interface, the relevant interfacial condition between the adjacent two layers can be presented by the following format:

$$u_k = u_{k+1} \quad v_k = v_{k+1}, \; w_k = w_{k+1} \tag{1-16a}$$

$$\sigma_{zz(k)} = \sigma_{zz(k+1)} \quad \sigma_{xz(k)} = \sigma_{xz(k+1)}, \; \sigma_{yz(k)} = \sigma_{yz(k+1)} \tag{1-16b}$$

If the interface bonding condition is imperfect, the interfacial continuous condition can be rewritten as

$$\sigma_{zz(k)} = \sigma_{zz(k+1)} = k_{zz}(w_k - w_{k+1}) \tag{1-17a}$$

$$\sigma_{xz(k)} = \sigma_{xz(k+1)} = k_{xz}(u_k - u_{k+1}) \;, \tag{1-17b}$$

$$\sigma_{yz(k)} = \sigma_{yz(k+1)} = k_{yz}(v_k - v_{k+1}) \tag{1-17c}$$

where the k_{zz}, k_{xz} and k_{yz} is the coefficient of the spring stiffness for the imperfect interfacial bonding. For a perfect bonded interface, their values will be infinite. For a void, their values are equal to zero.

1.2.2.6. Boundary Condition for the Part

In a same manner to laminated plate, there are three general boundaries can be used to describe the prescribed conditions as shown in Figure 1-3. n and s is the normal and tangent direction at one edge point. The prescribed conditions can be either the given stress or given displacement, which can be summarized as follows:

1. Simply supported edge:

$$u_s = 0; w = 0; \sigma_{nn} = 0 \tag{1-18a}$$

2. Clamped edge:

$$u_n = 0 \; ; u_s = 0; w = 0; \tag{1-18b}$$

3. Free edge:

$$\sigma_{nn} = 0; \;\; \sigma_{ns} = 0; \sigma_{nz} = 0; \tag{1-18c}$$

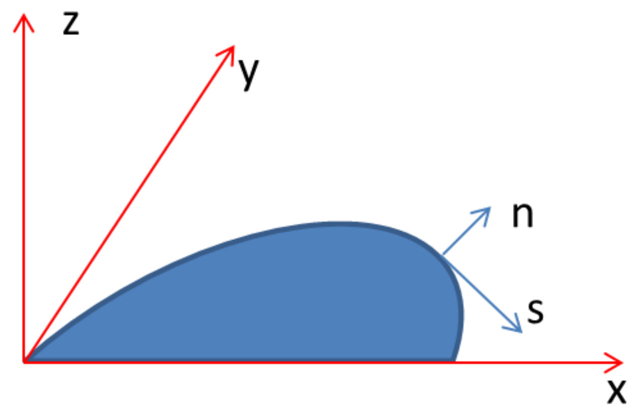


Figure 1-3: Local coordination system along the boundary of a laminated plate.

1.2.2.7. Solving Procedure

Obviously, the developed layerwise additive manufacturing elastic model can model the different materials deposition in the different sections or different layer in the whole part based on Equation (1-8). Using the governing Equation (1-8) and continuity conditions in Eqs (1-16) or (1-17) and prescribed boundary condition in Eqs.(1-18), the Layerwise Additive Manufacturing model can be analytically solved based on the state-space method. On the other hand, a block technique (CS3DM patent) can deal with the connection between the baseplate and the deposited section with the different geometries in terms of the basis solution.

1.3. Features of the LAMPS Software

In terms of the above software fundamental, the LAMPS software is developed and programmed by the object-oriented programming language C++. The LAMPS software's features can be specified as

1. Based on analytical block technique to achieve the fast computation and design of processing parameters.
2. Be capable of simulating the full-scale additive manufacturing part without the layer number limitation and predefine element.
3. Enable to simulate the multi-materials depositing, such as composite materials and handle the multi-heat sources problem.

4. Be able to predict the processing parameters on the distortion and residual stresses in the as-deposited part.

In version 2.0 of LAMPS software, it can carry out the effect of the processing parameters and part geometry on both the steady-state and unsteady-state temperature distribution in the full-scale part with linear properties as well as the temperature gradient and cooling rate, as shown in Table 1.2, and the distortion and residual stress as shown in Table 1.3, through the whole additive manufacturing procedure.

Table 1.2. The simulation capability of LAMPS on temperature fields in Version 2.0

Processing parameters	steady		transient		
	temperature	Temperature gradient	temperature	Temperature gradient	Cooling rate
Scan speed			√	√	√
Input energy	√	√	√	√	√
Beam radius	√	√	√	√	√
Absorption efficiency	√	√	√	√	√
Deposit materials	√	√	√	√	√
Part geometry	√	√	√	√	√
Deposition location	√	√	√	√	√
Deposit strategy /path			√	√	√

Table 1.3. The simulation capability of LAMPS on mechanical fields in Version 2.0

Processing parameters	steady			transient		
	displacement	strain	stress	displacement	strain	stress
Scan speed				√	√	√
Input energy	√	√	√	√	√	√
Beam radius	√	√	√	√	√	√
Absorption efficiency	√	√	√	√	√	√
Deposit materials	√	√	√	√	√	√
Part geometry	√	√	√	√	√	√
Deposition location	√	√	√	√	√	√
Deposit strategy /path				√	√	√

2. Installation

The current version can be installed in Windows system. The Installation of LAMPS software is very easy like other application software. The following steps are shown how to install the LAMPS.

Step 1: Double click the " LAMPS_installer_WManual" file as shown in Figure 2.1.

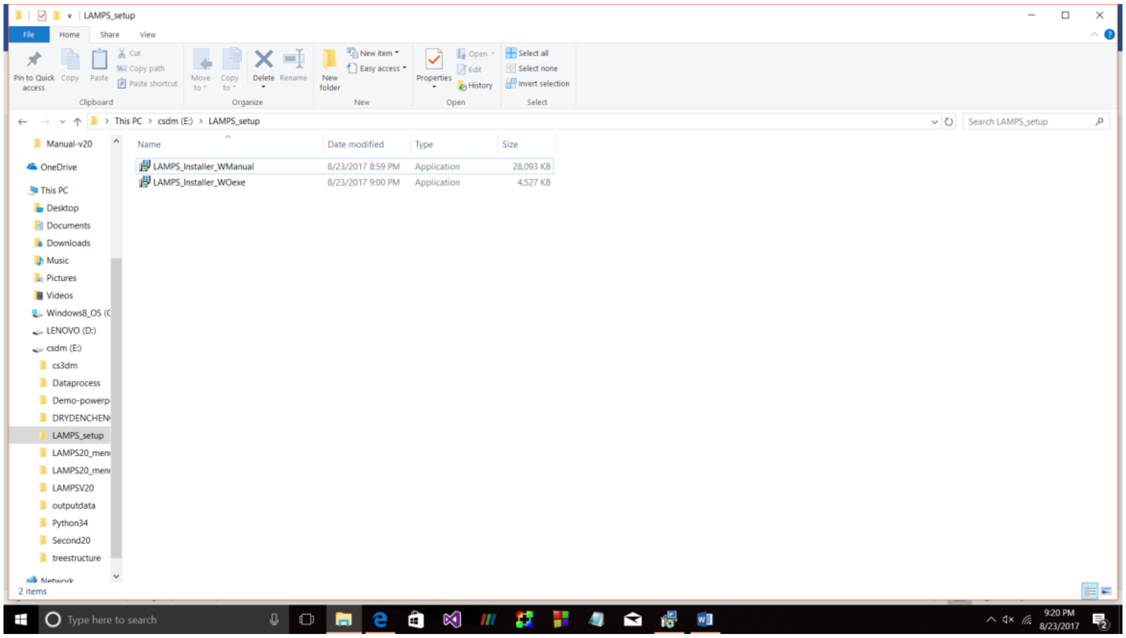


Figure 2-1: File view of the LAMPS_installer.exe

Step 2: After clicking one of the file, Figure 2-2 will show up for the next installation.

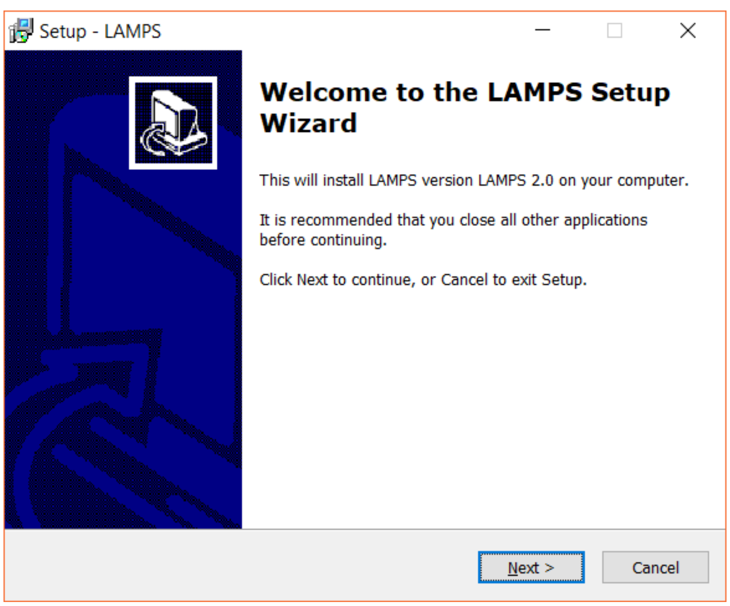


Figure 2-2: Welcome interface of LAMPS setup

Step 3: After clicking the "Next" button, the user can see the agreement of the LAMPS software as shown in Figure 2-3. After selecting "I accept the agreement ", click "Next" button for next step.

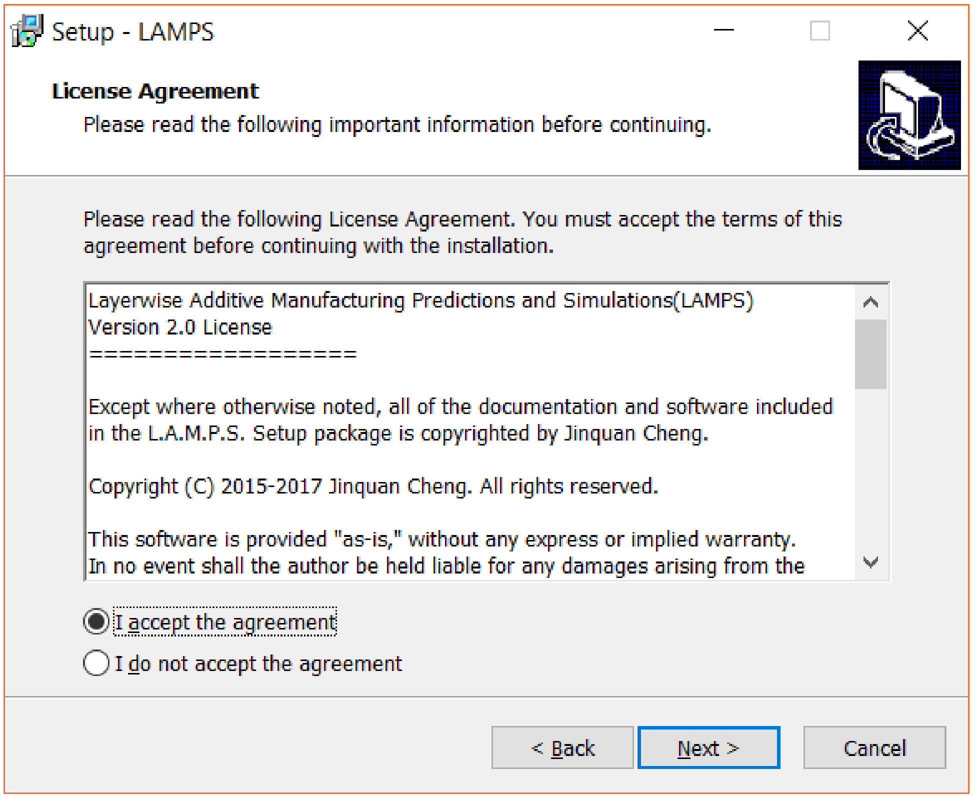


Figure 2-3: Password entering window

Step 4: After clicking the "next" button, a password requirement will show up like Figure 2-4. The Password can be queried from CS3DM company. After entering the password in the field, click "next " button for next step.

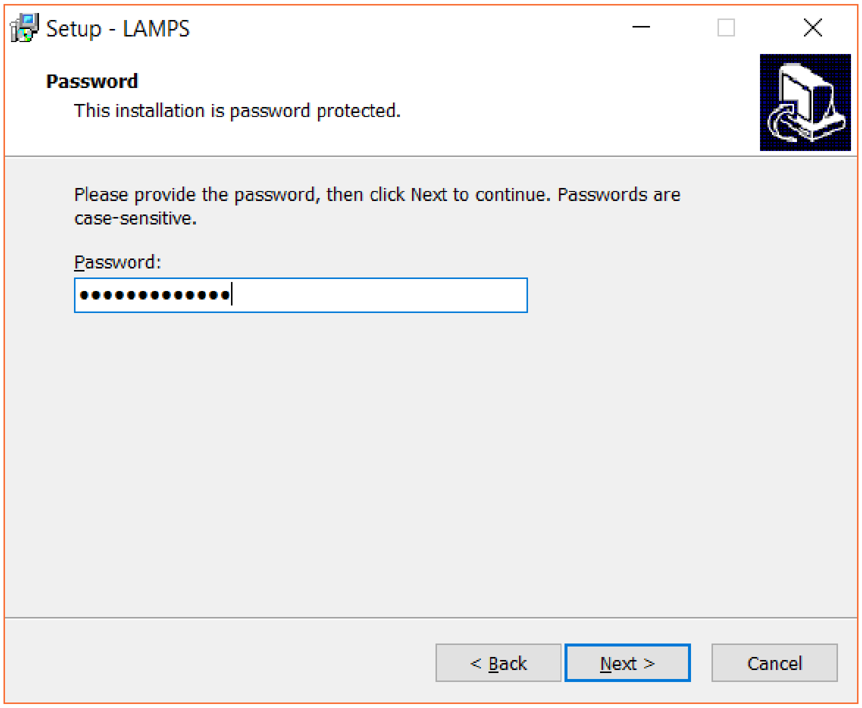


Figure 2-4: Installation confirmation

Step 5: After clicking the "Next" button in Figure 2-4, the install window will show up like Figure 2-5. After choosing the installation folder, Click the "Next" button, the LAMPS software will be installed in the destination location.

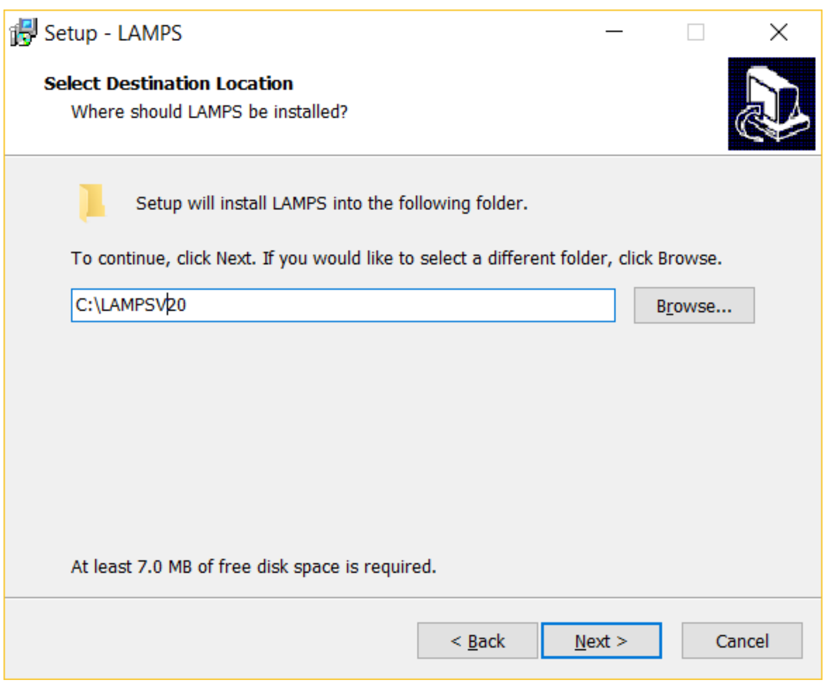


Figure 2-5: Installation interface for LAMPS software

Step 6: After click “Next” in Figure 2-5, the selection start menu folder window is shown up to determine the shortcut location. Generally, users do not need to revise it.

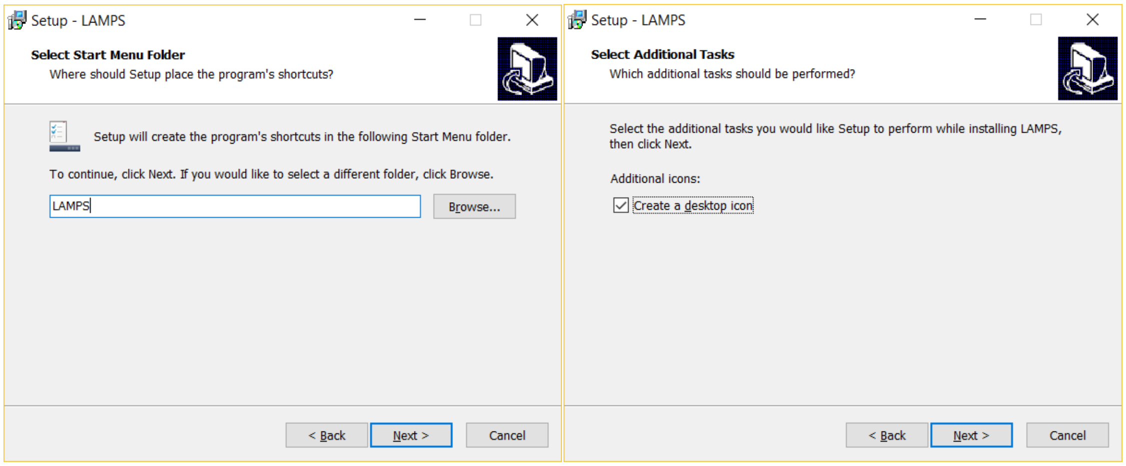


Figure 2-6: LAMPS shortcut creation

Step 7: After clicking the “Next” button in Figure 2-6, the ready to install window will display in Figure 2-7

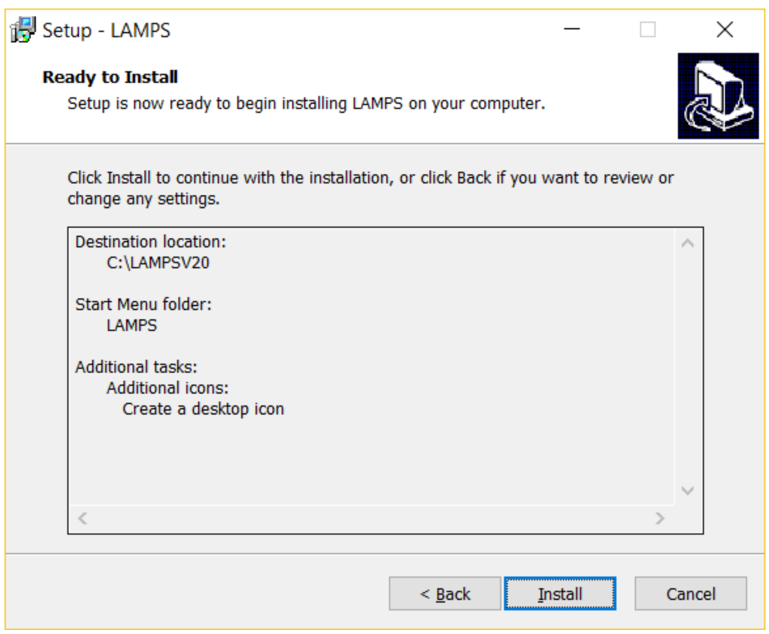


Figure 2-7: Ready to Install.

Step 8: After clicking the “Install” button, it will start to install the software and take some minutes to finish the installation. After finishing the installation, Figure 2-8 will pop up for confirmation. After clicking the “Finish” button, the user can run the LAMPS software directly from start menu or the installed folder.

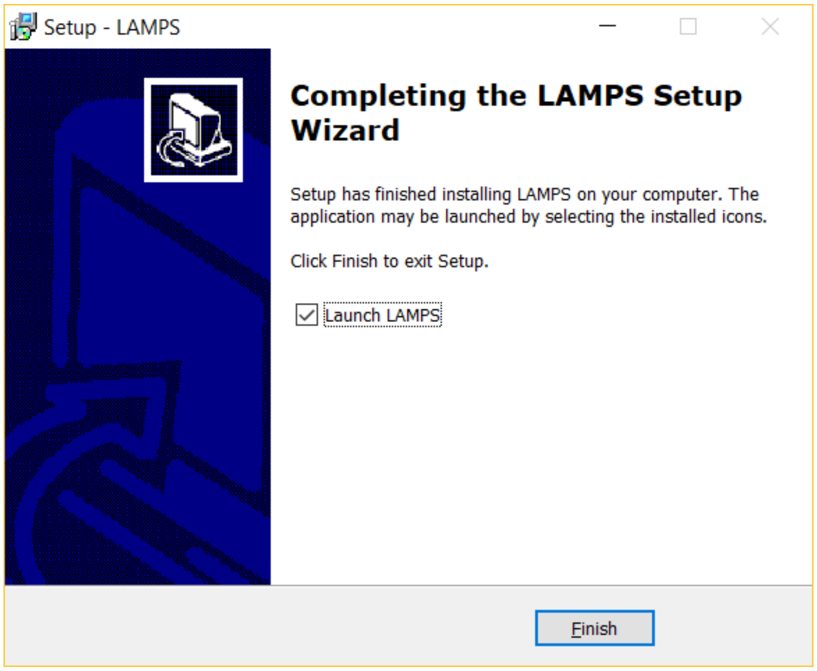


Figure 2-8: Installation finish

3. Running Program and I/O Data and Files: Overview

3.1. Running methods

After installed, the user can find the LAMPS shortcut in the start menu /program and double click the shortcut to run the LAMPS and enter the LAMPS GUI. Furthermore, there are two other ways to run the program via: (1) Command line and (2) GUI. The command line based running is using Windows command line to enter "LAMPS" or just double click "LAMPS" file under the installed directory with necessary completed six input files and an active license file. These seven required files are: (1) casetype.txt, (2) base_data, (3) materials_data, (4) layers_data, (5) sections_data, (6) depositpath_data and (7) license.txt. The file names for Files (2) - (6) will be given by the user while creating the necessary input files. The file name for File (1) is fixed as casetype.txt and the license file name as license.txt. The running interface is shown in Figure 3.1. How to build or create the seven input files is briefly summarized in the next sections and further details are explained in the next 7 chapters.

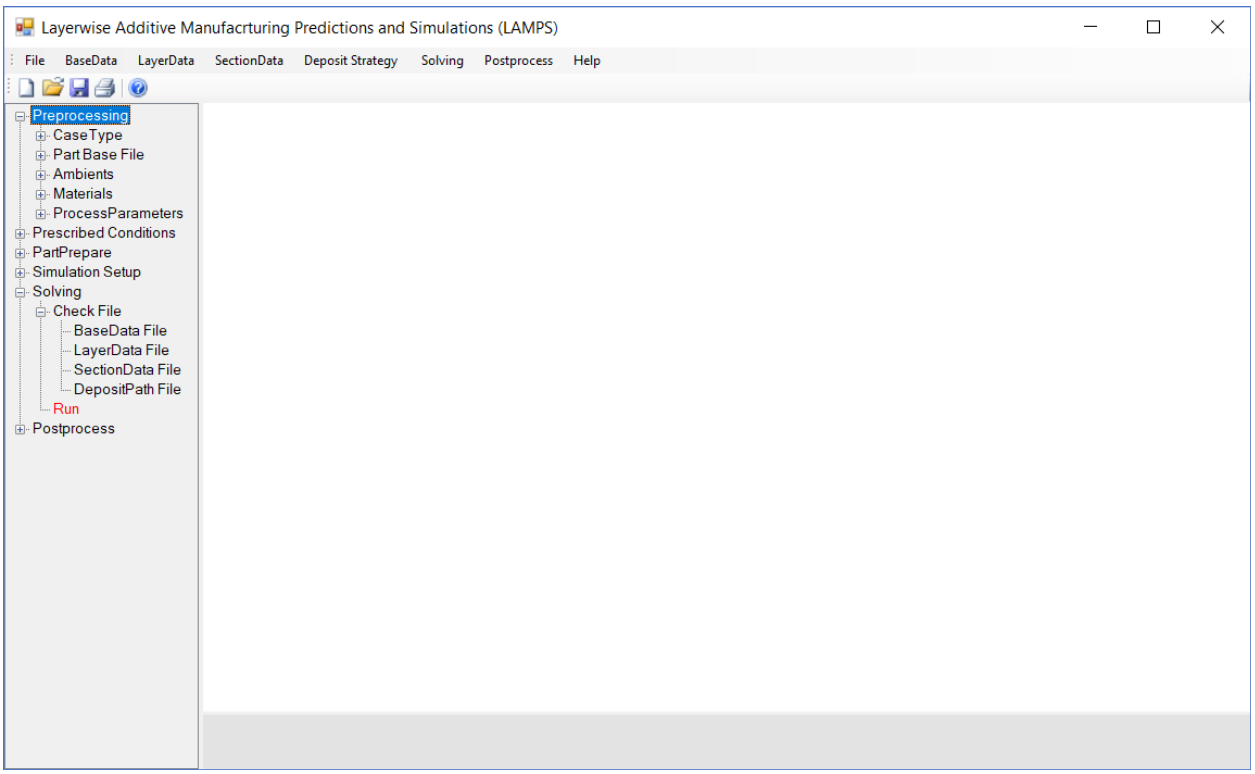


Figure 3.1: Interface after clicking "LAMPS" in the folder with completed three input files and an active license

The GUI based running is to double click "LAMPS_GUI" and then the LAMPS Graphics User Interface is displaying like Figure 3.1. The LAMPS GUI includes four zones: (1) main menubar, (2) side menubar and (3) editor zone. The user can build and complete the four input files by following the main menubar items or side menubar, which details are introduced in Chapter 5.

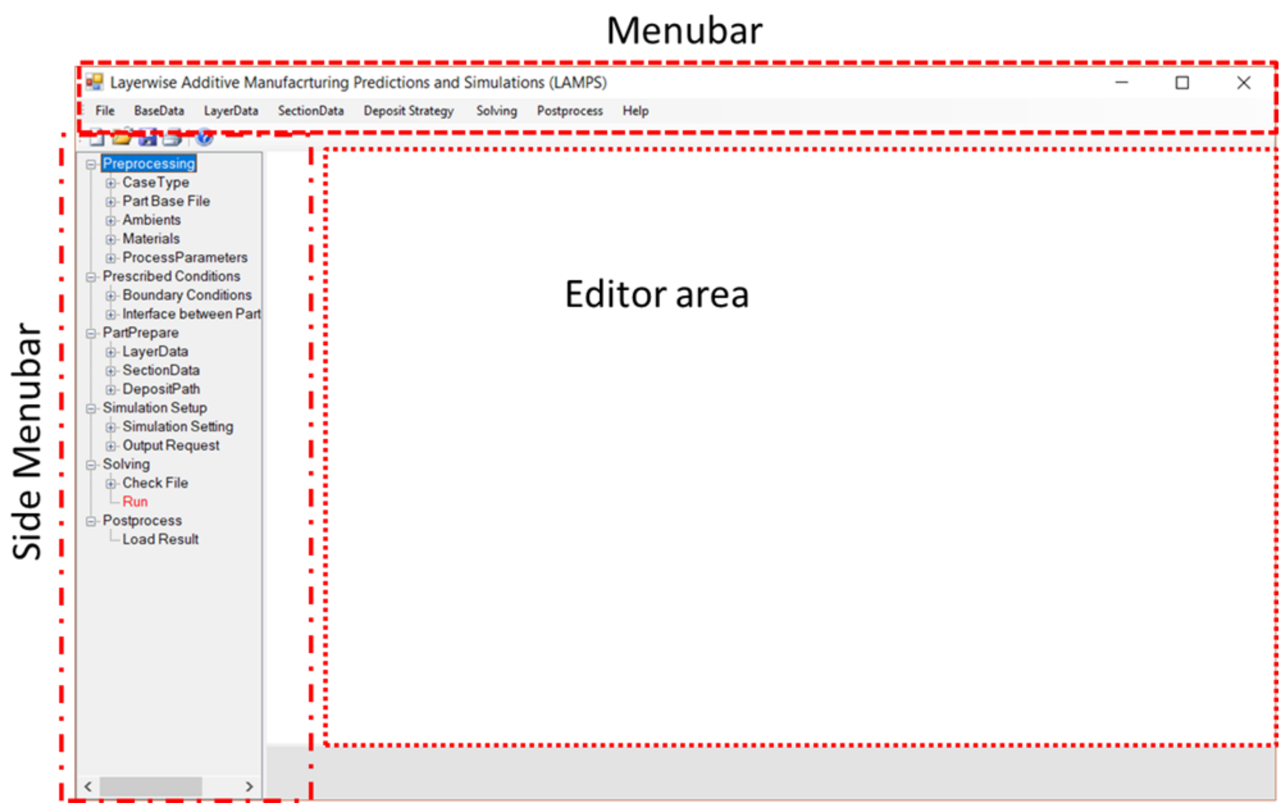


Figure 3.2: GUI for LAMPS program

3.2. Briefing of the input files required to run the simulation.

As above mentioned, there are seven input files required to finally execute the program with an active LAMPS license as follows

1. The casetype.txt is the main file to input the modeling case and the real file name of base_data. This is the first step to create the input data for further necessary input as shown in Chapter 6. Only after this step finished, the next data input can be executed.

2. The base_data file is the main file to input the deposition environment, the file name of the part and deposition materials, the file name of sections' data input, the file name of layers' data input, the file's name of the deposition path, the processing parameters for the additive manufacturing, such as the scan speed, the powder feed rate, the powder-bed thickness and densification, the cladding nozzle parameters, the heat source tool (laser or electron beam). The Chapter 7 will explain the details of the input parameters and their respective input formats in the base_data input file. After the respective files' names are given in the base_data file, the following data input for each individual file can be created.

3. The materials_data file is the file to list the materials' properties of all the materials in the AM part, which includes their thermo-mechanical properties, inherent properties. The real

file's name is given in the real simulation base_data file. The Chapter 8 will explain the details to input the data for material.

4. The layers_data file is focusing on giving the detail information of each depositing layer, including the layer number, layer geometry and depositing material name and ambient condition etc. The real file name of the layerdata.txt is given in the real base_data file. The Chapter 9 will introduce the input parameters and their individual input formats in the layerdata.txt respectively.

5. The sections_data file is to set up the block model for the depositions inside one layer, currently there are two block models to describe the deposition part shape: (a) Quadratic 8-node model (b) Cubic 12-node model, which is presented in Chapter 10 for more details. Its real file

6. The depositpath_data file is setting up the deposition path in the additive manufacturing for each layer. The format is mainly based on the GCODE format to create the simulation path in current version but it can be extended to the other 3D printing data format. The Chapter 11 will present the relevant G-Code and input format.

3.3. Output data files

In the current version of LAMPS (i.e. Version 2.0), there are two output files for temperature related fields and mechanical related fields respectively in an unstructured grid data format of VTK which extension is "*.vtu". In the file of temperature related fields, it includes the distributed temperature, temperature gradient and cooling rate in the unsteady-state/transient simulation case but only the temperature distribution and temperature gradient in the steady-state simulation case. And, the file for the mechanical related fields consists of the displacement, strain and stress. Furthermore, the output files locate in the folder of *executedfolder/outputdata*. In order to distinguish the file, the file name of three output files consists of three parts of characters: head_characters, middle_characters and tail_chatracters in a fix format as follows

Filename: Head_characters-middle_characters-tail_characters

where the head_characters are set by the sub-keyword : *type* under the keyword: simulation , the tail_characters are given by the sub-keyword: *outputfile* under the keyword: simulation, and the middle_characters are based on which type of the data is output as Table 3.1.

Table 3.1. Output data v.s. the middle_characters in the output file name

Output	Temperature relevant fields	Mechanical relevant fields
Middle_characters	Temp	Mech

For example, while the characters of the sub-keyword: *type* under the simulation keyword are "transient" and the sub-keyword: *outputfile* under the simulation keyword is set as "corner", the output file name for the temperature output data are given by "*transient-Temp-corner.vtu*" and the output file name for the mechanical output data are given by "*transient-Mech-corner.vtu*".

3.4. Log files

There are two log files in the folder of *executedfolder/outputdata* to output the real-time simulation progress: (1) *logfile.txt* is mainly to collect the screen output and monitoring the simulation progress; (2) *error.txt* is used to collect the error output.

4. Methodology to Create the Input Files by the File Editor

In order to conveniently manage the input data, two level data management approach is developed to inquire the requirement of the input data, i.e. *keyword* and *sub-keywords*. The upper level keyword will represent the overall property of an object and the lower level sub-keywords give the detail individual items/properties for the object. And an *end-keyword*, i.e. *~end, is used to terminate the current object and the comment line starts with ** using to explain the choices or meanings of the sub-keyword in the following line, like the following sample in Figure 4.1:

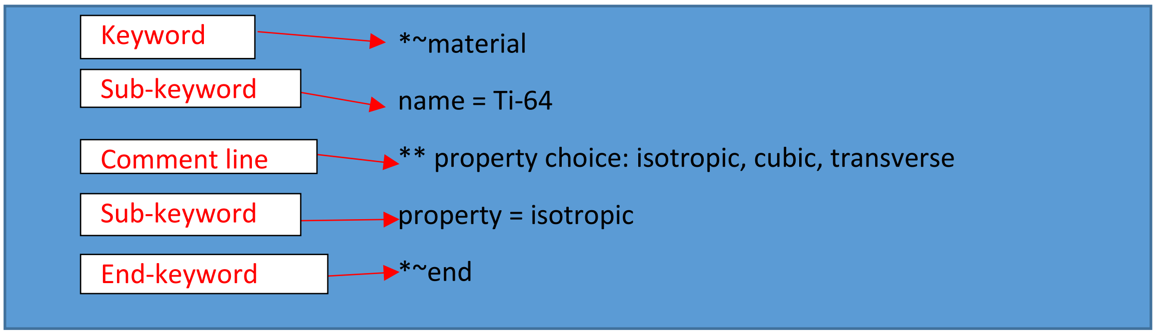


Figure 4.1: A sample for a multilevel data input format with one keyword, some sub-keywords, comment line and one end-keyword

The details of how to set up the above three input files are discussed in the Chapter 7 for the base_data input file, the Chapter 8 for the materials_data input file, the Chapter 9 for the layers_data input file, the Chapter 10 for the sections_data and the Chapter11 for the depositpath_data input file.

5. GUI Based Program Running

5.1. Five-steps to create the simulation data and then run LAMPS

Figure 3.2 displays the graphic user interface. The user can follow the main menubar to create the items one-by-one and set up the necessary input files for finally running LAMPS: *BaseData* → *LayerData* → *SectionData*→ *DepositStrategy* → *Solving*, as shown in Figure 5-1.

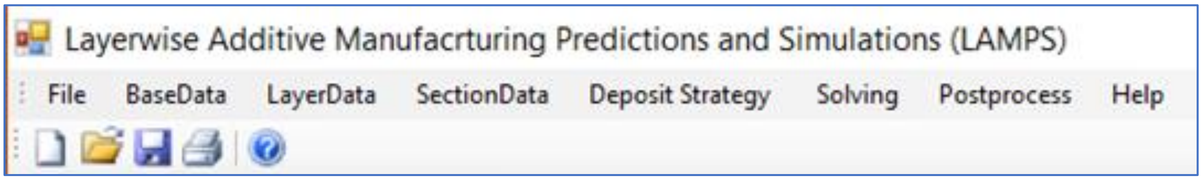


Figure 5.1: Main menubar of LAMPS software and the five steps to build and run a simulation case

Or, one can follow the side menubar to construct the relevant six input data files step by step as shown in Figure 5.2.

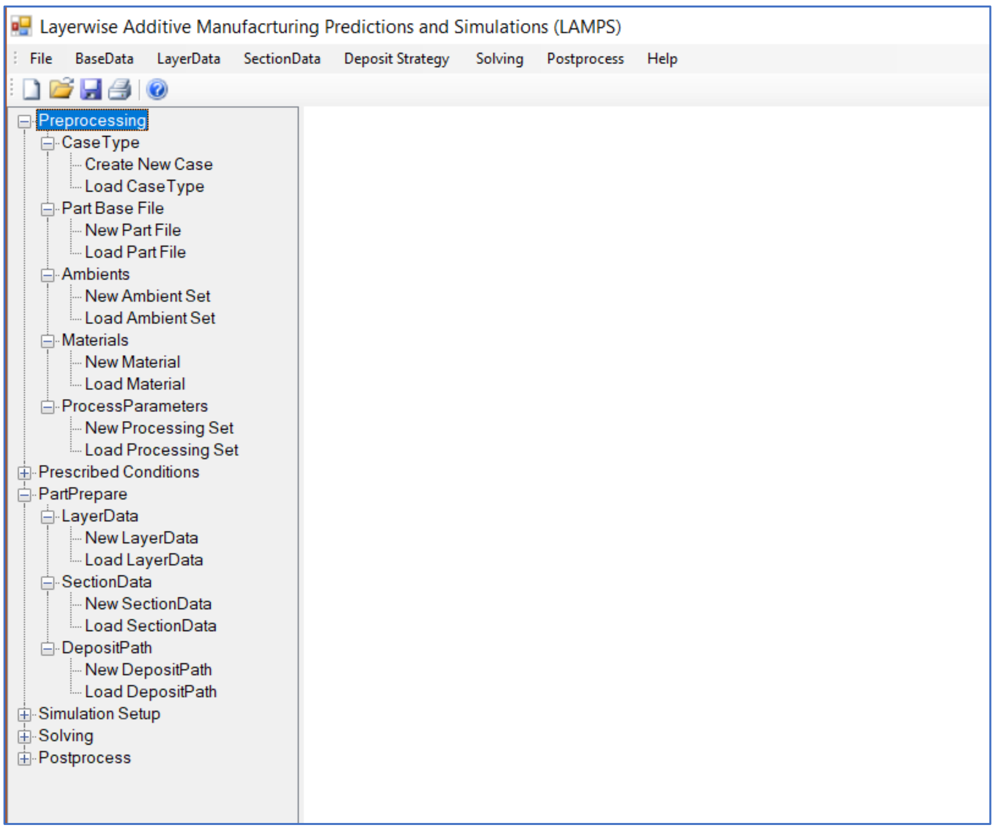


Figure 5-2: Side menubar to create the required seven input data files

5.2. Step 1: CaseType.txt creation

First step is to create the casetype.txt following the submenu “CaseType” of Menu “BaseData”. After clicking the “Create new CaseType” item, Figure 5.3 shows the GUI to get the input data to build up the simulation. There are two choices in the simulation model combox as: (1) additive manufacturing model and (2) reactive multilayer foils model. The additive manufacturing model is only for additive manufacturing simulation and the reactive multilayer foils model is for reactive multilayer foils case.

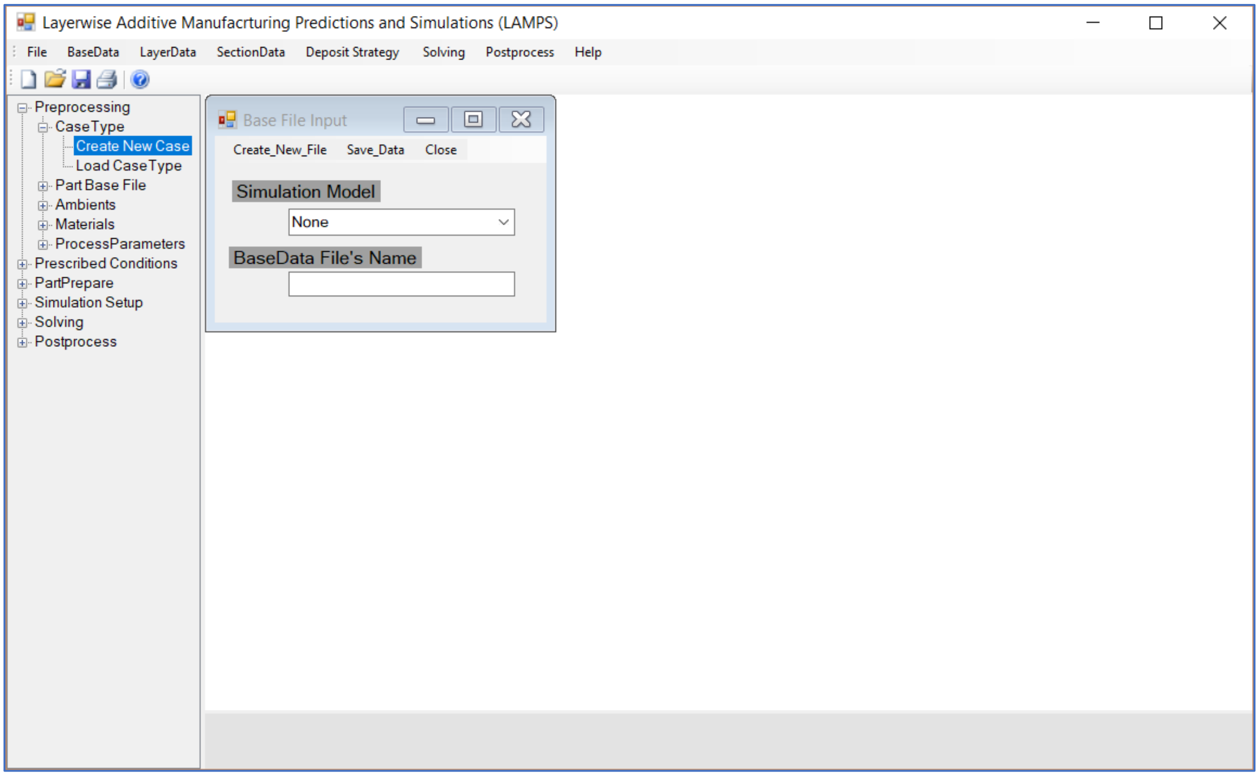


Figure 5-3: GUI for casetype.txt input data and creation after clicking the "Create new Case" item.

5.3. Step 2: Base_data creation

Second step is to create the basedata.txt following the submenus of Menu "BaseData" step-by-step. Here the file name of the Basedata is given in the Step 1 Figure 5-3 displays the GUI after clicking the "BaseData" item in the main-menubar. There are seven sub-items need further actions: (1) "Part BaseFile"; (2) "Ambient"; (3) "Materials"; (4) "Processing"; (5) "Boundary", (6) "Simulation" and (7) "Output" which are separated by a line as shown in Figure 5-3. The other GUI option to create the basedata.txt is from the side-menubar step-by-step as shown in Figure 5-2. Therefore, the basedata.txt file can be created from an empty file step-by-step from GUI or copied from an existed basedata.txt file. The following guide is to show how to create a basedata.txt step-by-step from GUI.

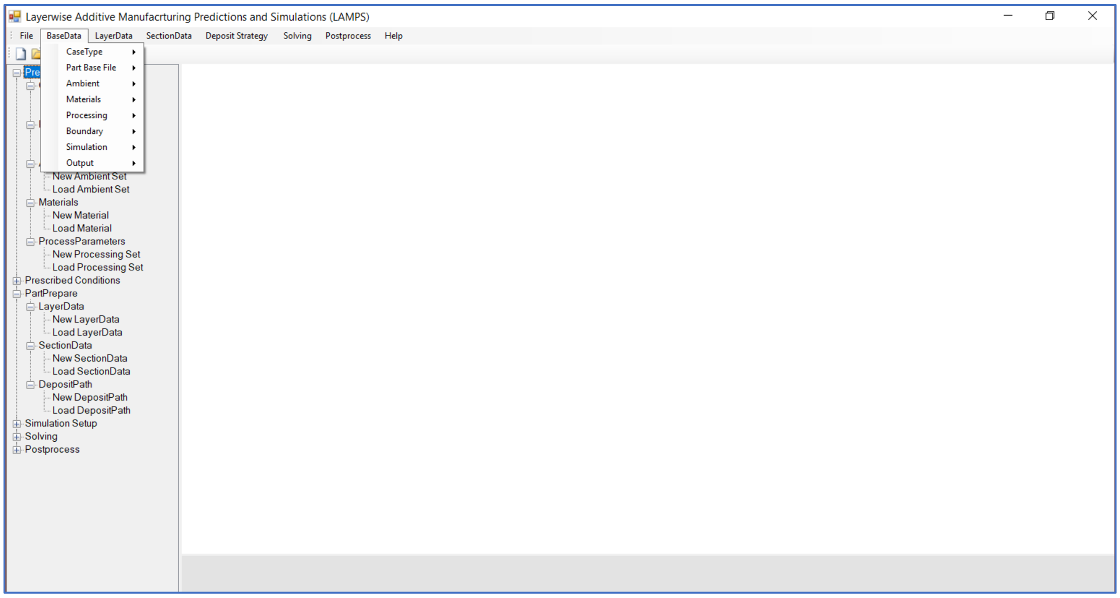


Figure 5-3: GUI after clicking "BaseData" item in the main menubar

Step 2-1: Click "Part Base File" item and submenu "New Part File", a data input dialog form of Base File Input will pop up like Figure 5.4. Then, one can input the files' names for the three input data files: materials_data, layer_data and section_data respectively. The input file names for the three data files will be used to store the required data for simulation about each material's data, each layer's data and each section's data. After clicking the "Save_To_BaseData", the input data will be saved to the base_data file which name given in CaseType.txt input. After setting up all the files' name, the details input will be handled by the following steps.

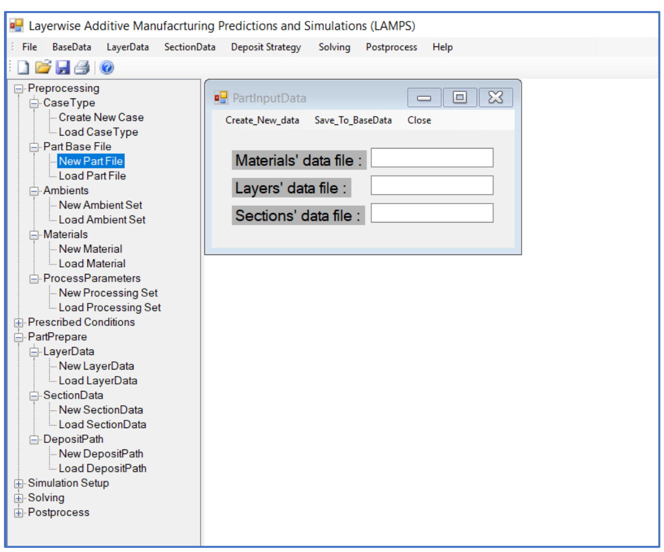


Figure 5-4: GUI input for files' name for the materials_data, layers_data and sections_data, and save to base_data file which name is given in the CaseType.txt.

Step 2-2: Click "Ambient" item and submenu "New Ambient" in the main menubar or click "Preprocessing" item, select submenu "Ambient" and its sub-item "New Ambient Set" in the side menubar, a data input dialog form of *Ambient Keyword Input* will pop up like Figure 5.5. Then, one can input the required data. The detailed explanation for each input item is described in Chapter 6.1 about the *ambient* keyword. After finishing the data setting, click the "Save_to_BaseData" button of the menustrip in the top of the form to add the created data set to the basedata.txt file. If more ambient condition sets wanted, click the "Create_New_Data" button of the menustrip in the top of the form to clear the old input and re-enter the new data to create new data set.

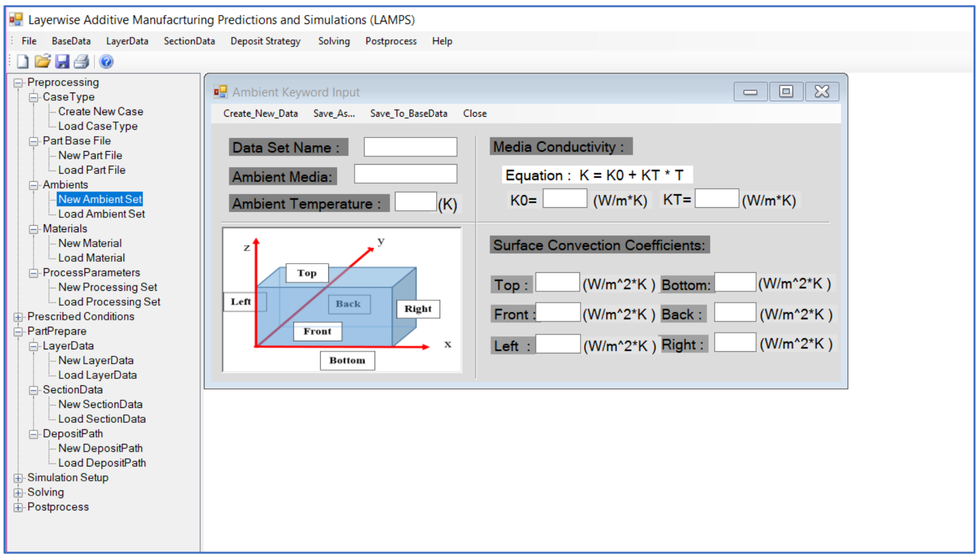


Figure 5-5: GUI input for ambient keyword and save to base_data file which name is given in the CaseType.txt input

Step 2-3: Click "BaseData" item and select the "Materials" sub-item and its submenu "New Material" in the main menubar or click "Materials" item and then select "New Material" under the "Preprocessing" menu in the side menubar. The *Material Keyword Input Form* will show up like Figure 5-6 for setting and creating the fourth keywords in the material_data file: (1) *~material keyword, (2) *~elastic keyword, (3) *~inherent keyword and (4) *~thermophysical keyword. The file name of the materials_data file is given by the input in the "Base Part File" step.

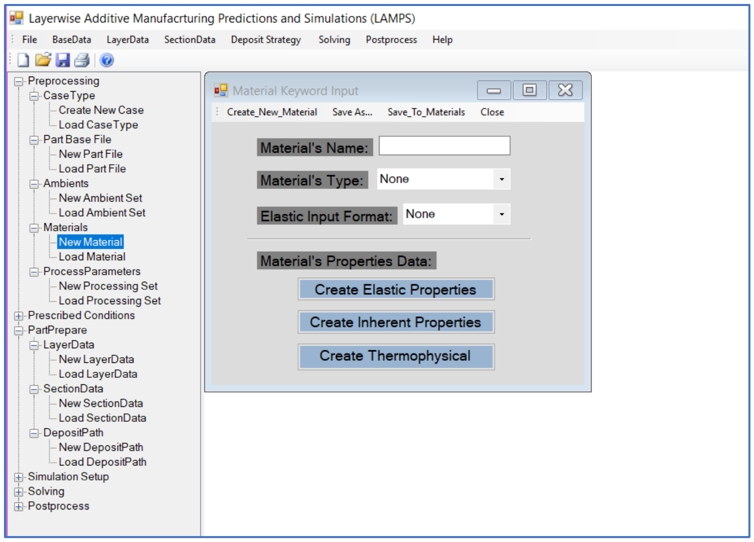
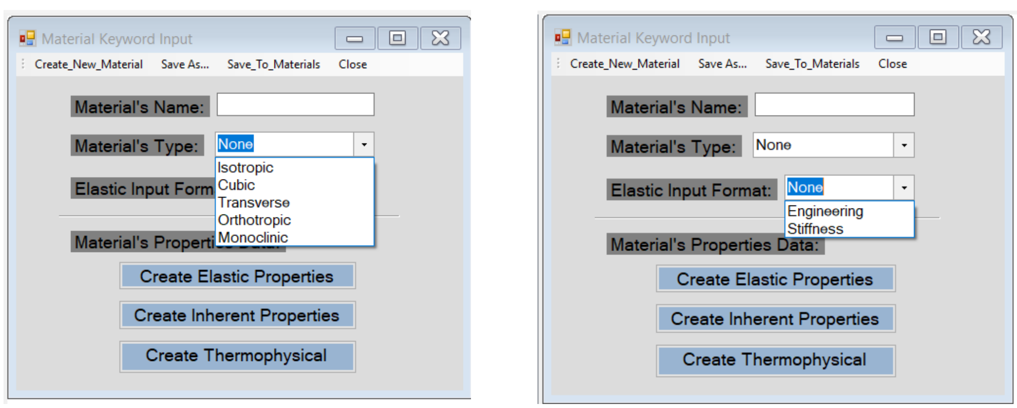


Figure 5-6: GUI input for creating a material through three keywords

Step 2-4: Based on the material properties, one needs enter the name of the material and select the material type and data input format by dropdown selection as shown in Figure 5-7(a). There are five available options for the material property: *isotropic*, *cubic*, *transversal isotropic* (i.e. *transverse*), *orthotropic* and *monoclinic* respectively, i.e. current software can model and analyze the above listed types of the materials. Furthermore, current software can automatically calculate the model for the two input formats of the elastic constants: *Engineering* and *Stiffness*, as indicated in Figure 5-7.



(a) material type (b) data input format

Figure 5-7: Selection of the material type and its input form

Step 2-5: After entering the material's name and selecting the respective material property type and input format, one must click the following three buttons respectively: (1) "Create Elastic Properties" button to input the elastic constants, (2) "Create Inherent Properties" button to input the inherent shrinkage and yielding temperatures as described in Chapter

1.2.2.3. and (3) "Create Thermophysical Properties" button to enter the thermos-physical property data required for simulation, as shown in the bottom section of Figure 5-8.

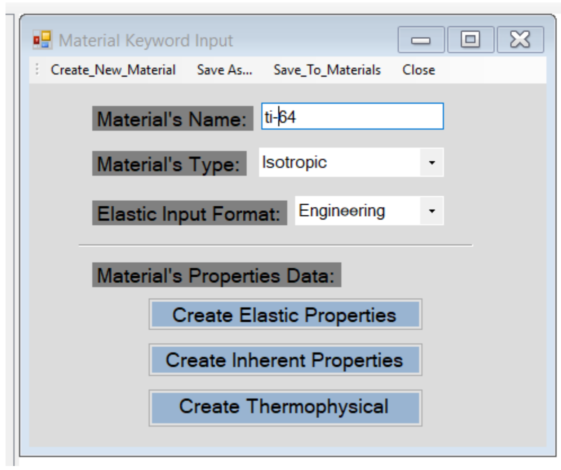


Figure 5-8: Options of material's type and elastic input formation

Step 2-6: Click "Create Elastic Properties" button, Figure 5-9 displays the elastic properties input form for an isotropic material entered in the *Material Keyword Input form,* named as Ti-64 with isotropic property and Engineering input format as shown in Figure 5-9. Here the first three text inputs about the material's name, material property and input format in the left top of the form inherit from the *Material Keyword Input form*. For the different property types of the materials, the different elastic input forms will pop up on the basis of the input data in the *Material Keyword Input form*. Figure 5-10 displays the stiffness constant input form for an isotropic material; Figure 5-11 presents the data input for a cubic material; Figure 5-12 illustrates the data input for a transversal isotropic material; Figure 5-13 shows the data input for an orthotropic material; Figure 5-14 provides the data input form for a monoclinic material. After finishing the data input in the *material property input form*, click "Save_To_Materials" to attach the elastic input data to the material named and required in the *Material Keyword Input form* and the input form will be closed automatically.

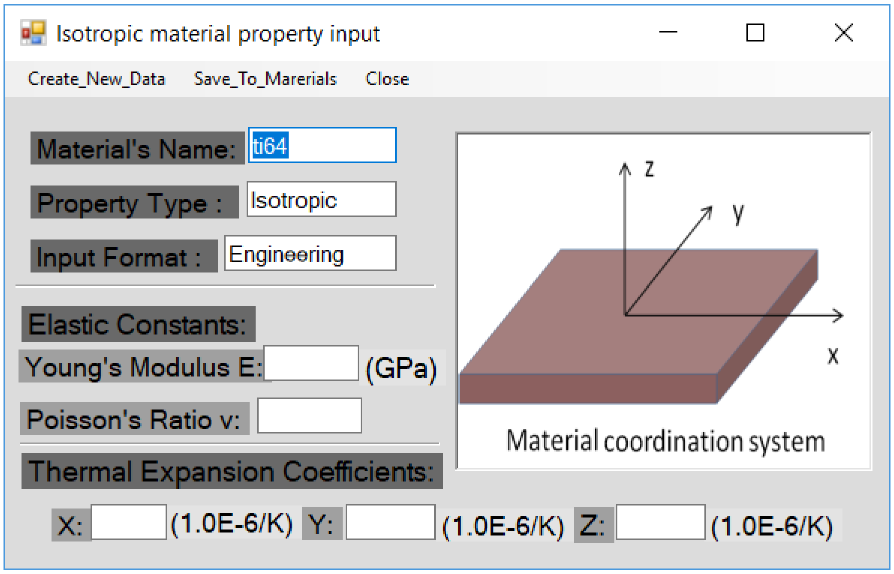


Figure 5-9: Engineering constants input for an isotropic material

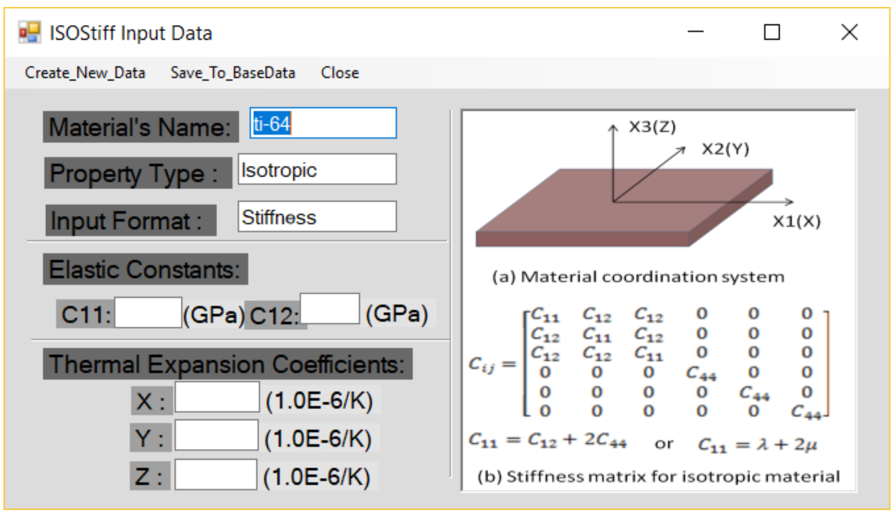
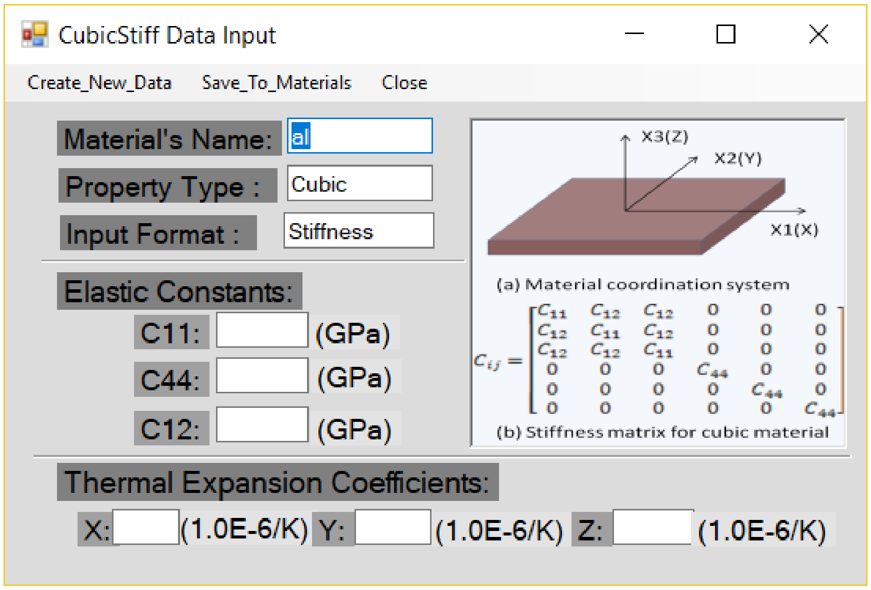
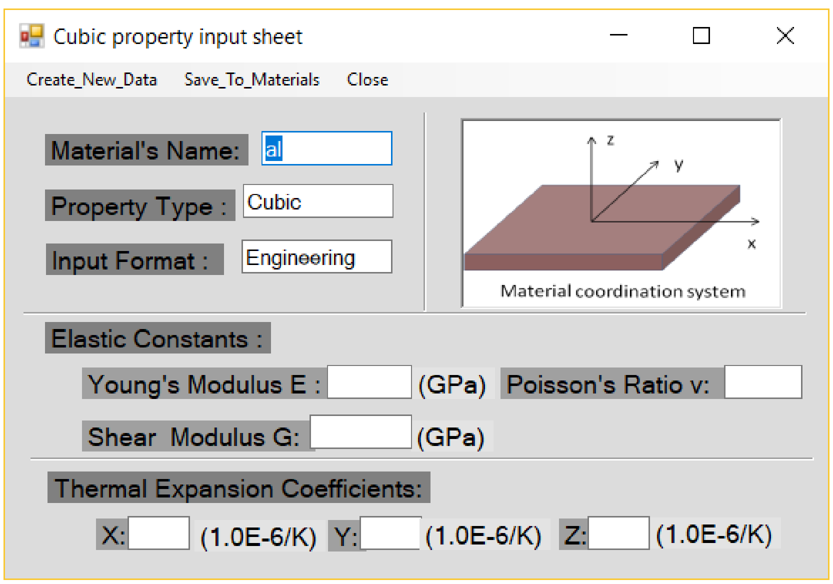


Figure 5- 10: Stiffness coefficients input for an isotropic material

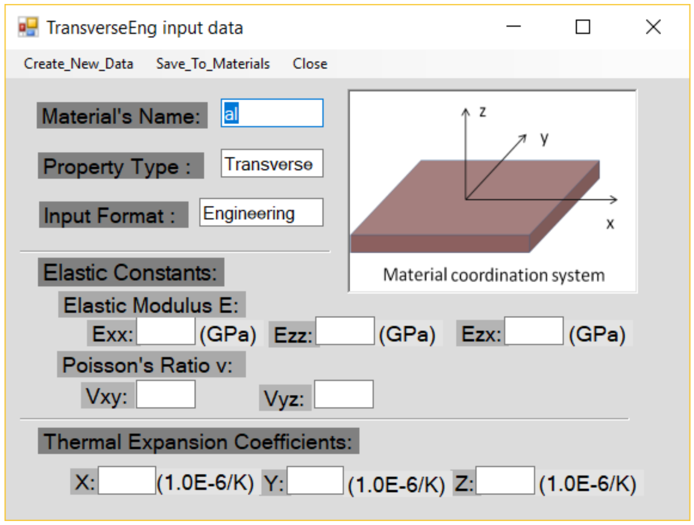


(a) Stiffness input format for a cubic material

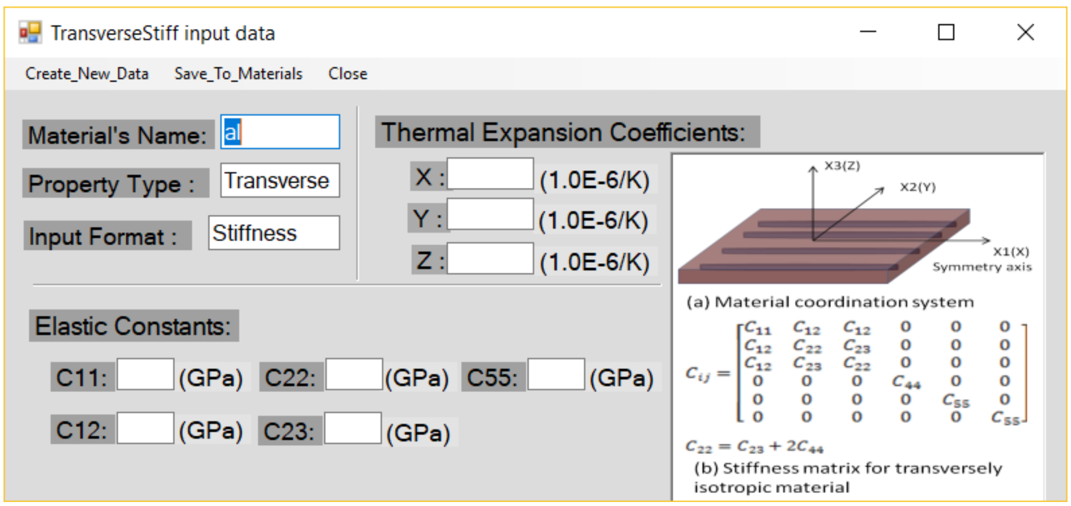


(b) Engineering input format for a cubic material

Figure 5-11: Data input of elastic constants and thermal expansion coefficients for a cubic material

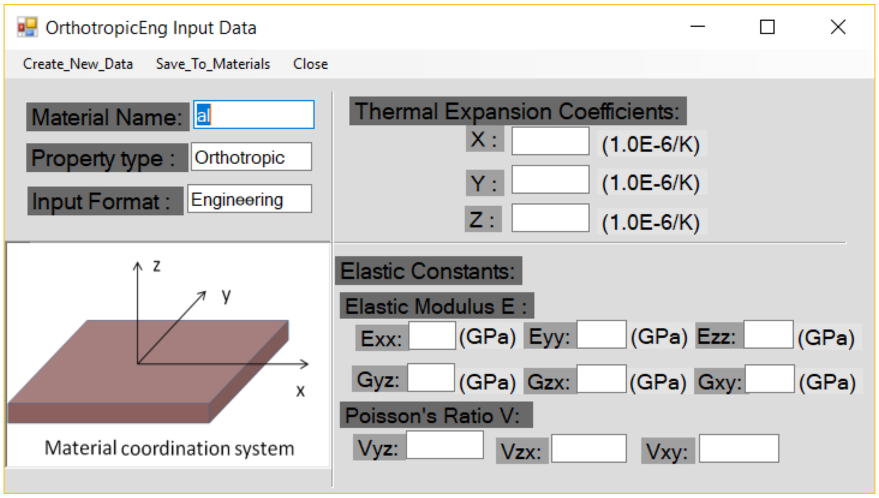


(a) Engineering input format for a transversal isotropic material

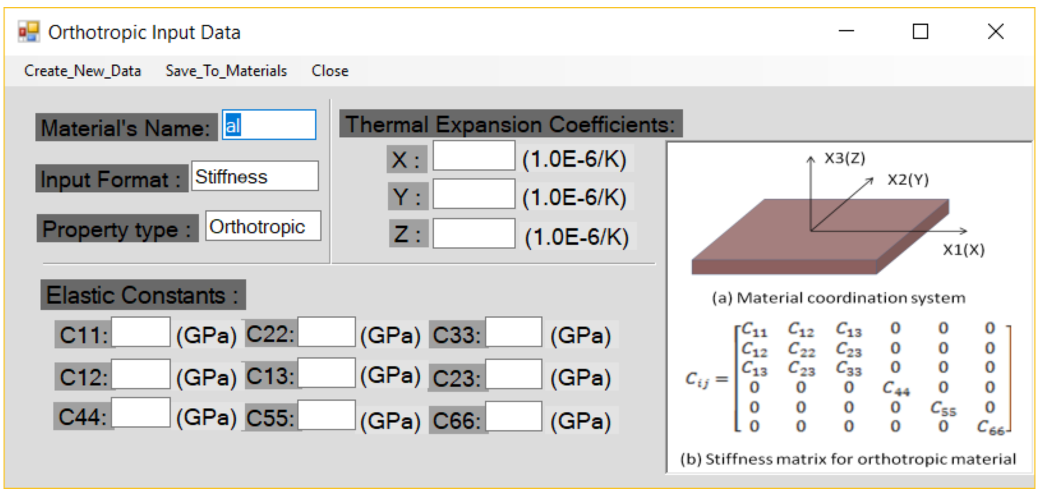


(b) Stiffness input format for a transversal isotropic material

Figure 5-12: Data input of elastic constants and thermal expansion coefficients for a transversal isotropic material

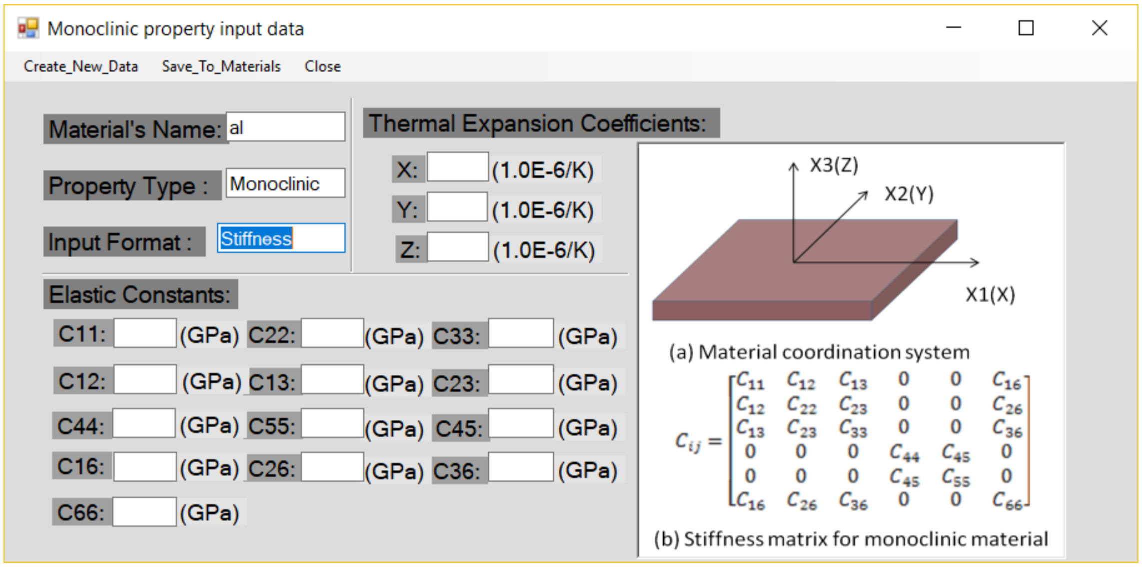


(a) Engineering input format for an orthotropic material



(b) Stiffness input format for an orthotropic material

Figure 5-13: Data input of elastic constants and thermal expansion coefficients for an orthotropic material



$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & C_{16} \\ C_{12} & C_{22} & C_{23} & 0 & 0 & C_{26} \\ C_{13} & C_{23} & C_{33} & 0 & 0 & C_{36} \\ 0 & 0 & 0 & C_{44} & C_{45} & 0 \\ 0 & 0 & 0 & C_{45} & C_{55} & 0 \\ C_{16} & C_{26} & C_{36} & 0 & 0 & C_{66} \end{bmatrix}$$

Figure 5-14: Data input of elastic constants and thermal expansion coefficients for a monoclinic material

Step 2-7: Click "Create Inherent Properties" button in the *Material Keyword Input form*, the *Inherent Keyword Input form* will prompt for the inherent properties data input, as shown in Figure 5-15. Here the material's name inherit from the *Material Keyword Input form*. In this form, one need provide the shrinkage ratio and the two yielding temperatures for the four-linear-line model as described in Chapter 1. 2.2.3. After completing the data input, click "Save_To_Materials" in the top menustrip of the form to return the inherent properties data to the material required in the *Material Keyword Input form*.

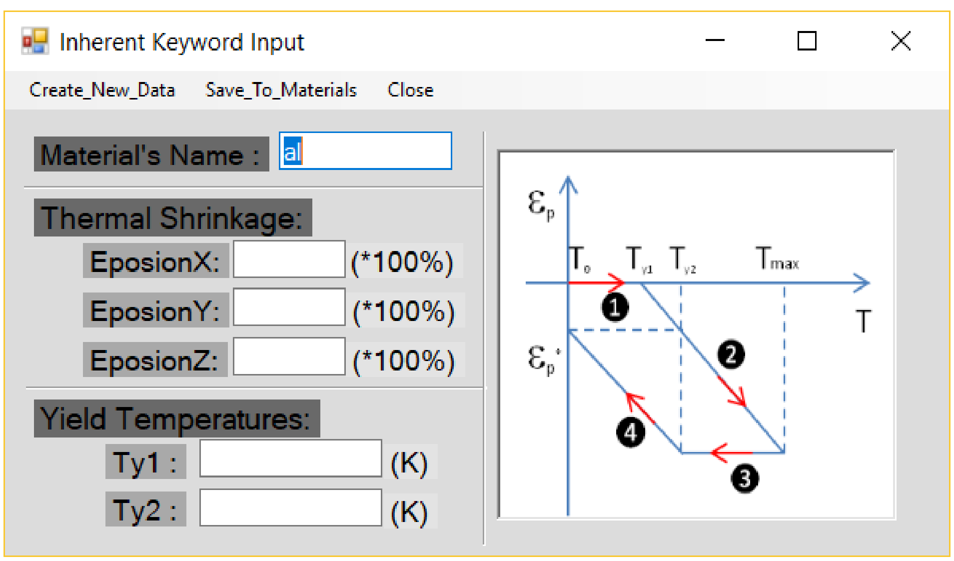


Figure 5-15: Inherent properties data input form for the four-linear-line model for shrinkage

Step 2-8: Click "Create Thermophysical Properties" button in the *Material Keyword Input form* to display the *Thermal Input Data Sheet* in Figure 5-16. It is noted that current version software only works for constant thermophysical property case. Once the data input finished, click "Save_To_Materials" in the top menustrip of the form to return the thermophysical properties data to the material required in the *Material Keyword Input form.*

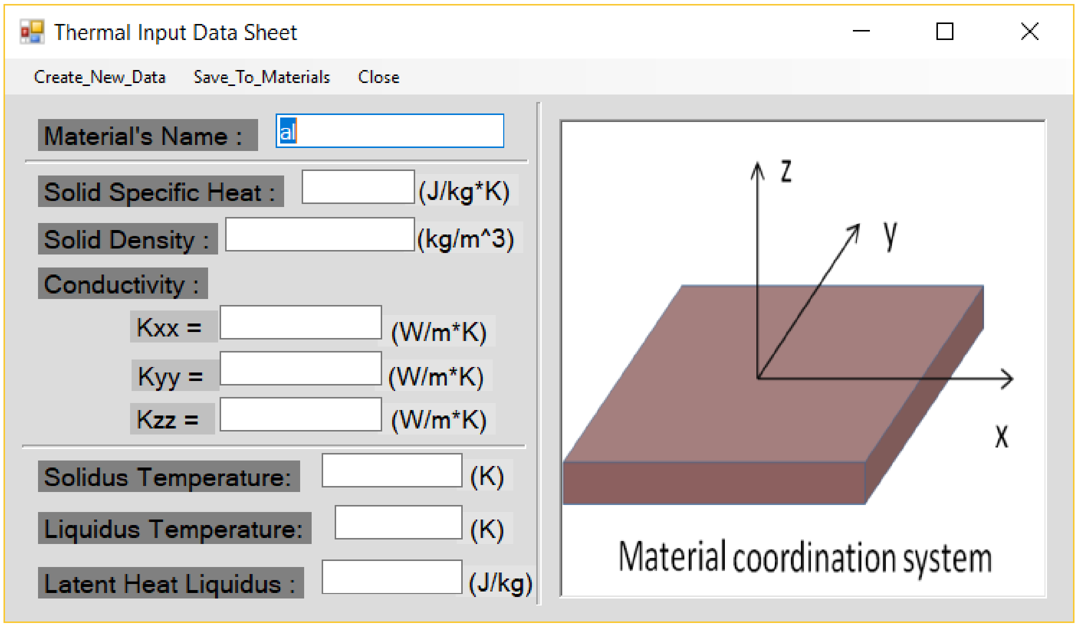


Figure 5-16: GUI data input for constant thermophsyical properties acquircy

Step 2-9: After finishing the data input for the elastic properties, inherent properties and thermophysical properties, user must click "Save_To_Materials" in the top menustrip of the *Material Keyword Input form* to save the created material to the Materials_Data file as shown in Figure 5-6. Some warning or confirmation messages will show up to indicate whether the input data is successfully saved to the BaseData.txt file or not. Furthermore, the user can save the input data as an alone file for future application by clicking "Save-As…"in the top of the form.

Step 2-10: Click the "Processing" sub-menu and "New Processing" sub-item under the "BaseData" Item in the main menubar as shown in Figure 5-3 or Click the "New Processing Set" item of the "ProcessParameters" sub-menu under the "Preprocessing" item in the side menubar, the GUI for process parameter input will prompt in Figure 5-17. Enter the characters in the text box following the "Processing Name" label to define the processing's name. Select one of the two processing types from the dropdown menu: (1) *powderbed* and (2) *cladding*. If choosing the powderbed, the *Powderbed Keyword Input Form* will show up like Figure 5-18. There are two variables required for powderbed processing: (1) penetration depth and (2) powder densification of the powder bed. After finishing the data for powderbed keyword, click the "Save_To_BaseData" to return the data to the main form of process parameters and the form will close automatically. If choosing the cladding processing, the *Cladding Keyword Input From* will show up like Figure 5-19. The are eight parameters required for modeling the cladding type additive manufacturing. For more details of *the processing keyword input*, please reference to the Chapter 7.

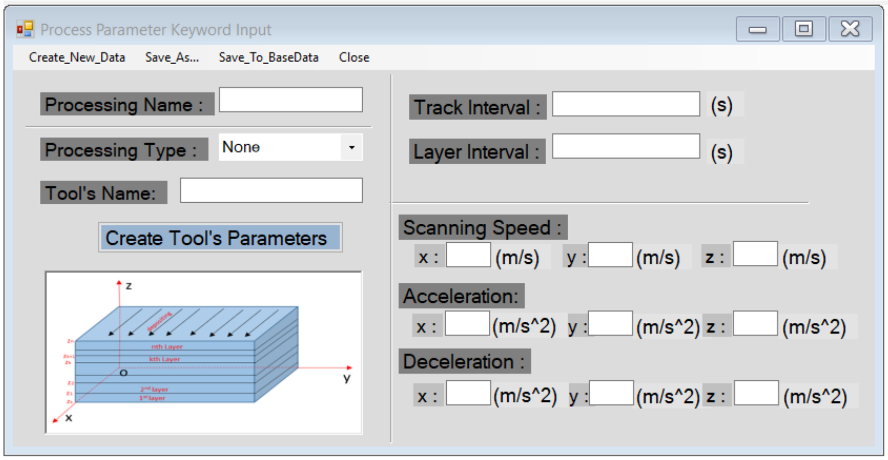


Figure 5-17: Processing parameter set up for the model

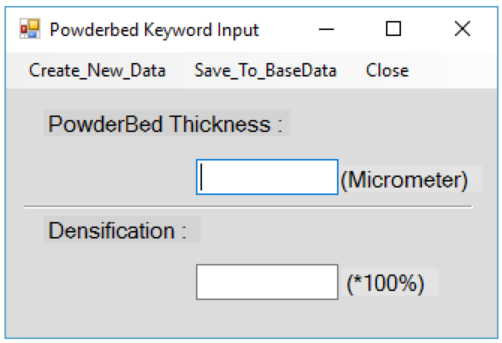


Figure 5-18: powder bed parameter input

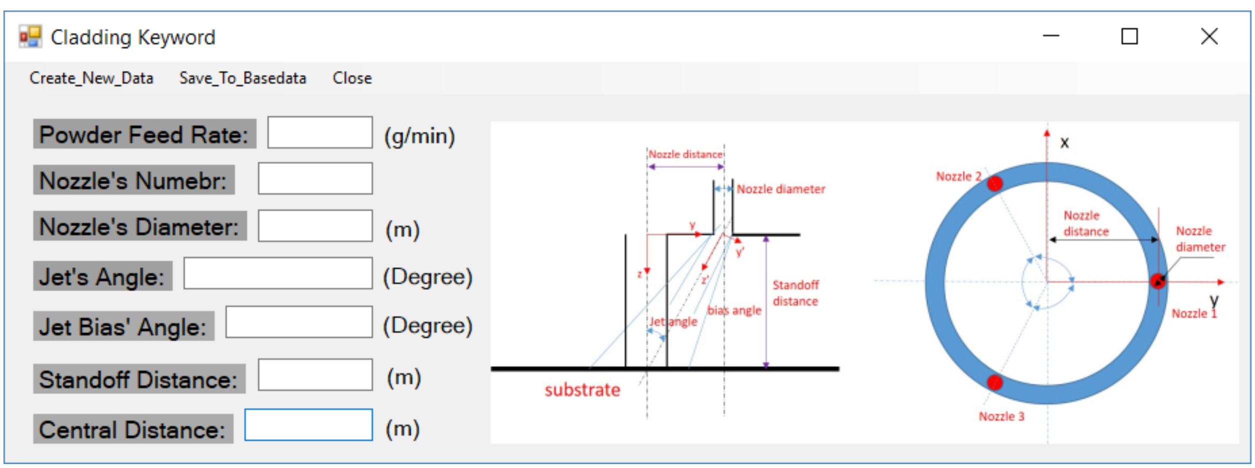


Figure 5-19: Cladding parameter input

Step 2-11: To input the tool parameters for the process named in the text box of "Processing Name, enter the tool's name following the label "Tool's name" and then click "Create Tool's Parameters" button to display the *Tool Keyword Input Form* like Figure 5-20. For the tool, there are two types of the tool source to model: (1) laser and (2) electron beam. And two types of the beam shape can be chosen: (1) square and (2) Gaussian. In current version, only square shape is modeled. In terms of the working principal of the laser and electron beam, the penetration is set as zero for laser, but for electron beam, the penetration depth must be given. While the study is only for steady-status, the initial position (x,y,z) is the heat source location. For the additive manufacturing, the tool movement (deposition strategy) is one of the most important processing parameters. The item of "Tool path file" is to give the file name of the deposit path. After fully filling in the data, click "Save_To_BaseData" in the top menustrip of the form to return the tool information to the required "Processing set". After the processing parameter data finished with the relevant data of the relevant powderbed and tool, click "Save_To_BaseData" in the top of the *Process Parameters Keyword Input form* to save the created processing parameters to the file of Base_Data which named in the casetype.txt.

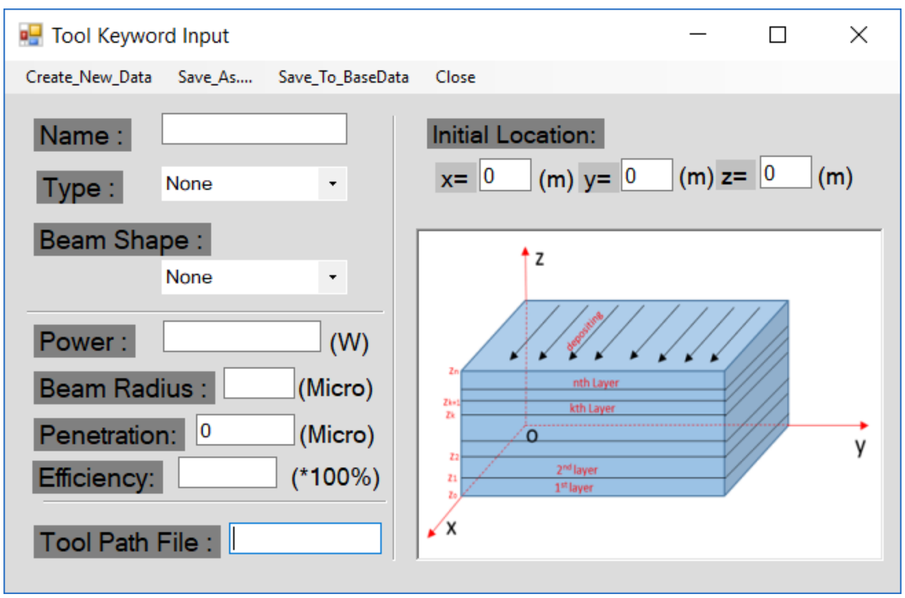


Figure 5-20: Tool keyword input form for entering the tool parameters

Step 2-12. Click "Boundary" item under the "BaseData" in the main menubar and then select "New Boundary" submenu to input the prescribed boundary conditions. Or, click "Prescribed Conditions" and then select the "New Boundary Set" item of the "Boundary Conditions" sub-menu in the side menubar. After selection, the *Boundary Keyword Input form* will show up to input the boundary conditions, i.e. Figure 5-21. There are three types of the boundary conditions considered in the software: (1) *simply-supported condition*, (2) *clamped condition* and (3) *free condition*. User can use the dropdown menu to select the boundary condition for one boundary edge on a part. For the top and bottom surface, there are two types of the loading enable to apply: (1) given forces or (stresses) -$\sigma_{xz}, \sigma_{yz}, \sigma_{zz}$, and (2) given displacements u, v, w , as shown in Figure 5-22. After the applied fields selected, user must enter the relevant prescribed conditions in the following boxes respectively with respective to the component.

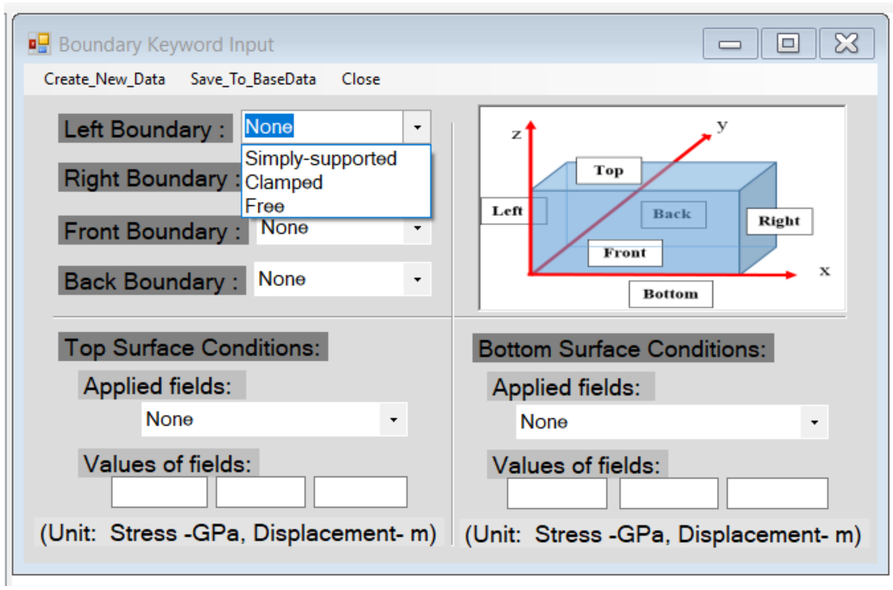


Figure 5-21: Boundary condition input for the deposition part

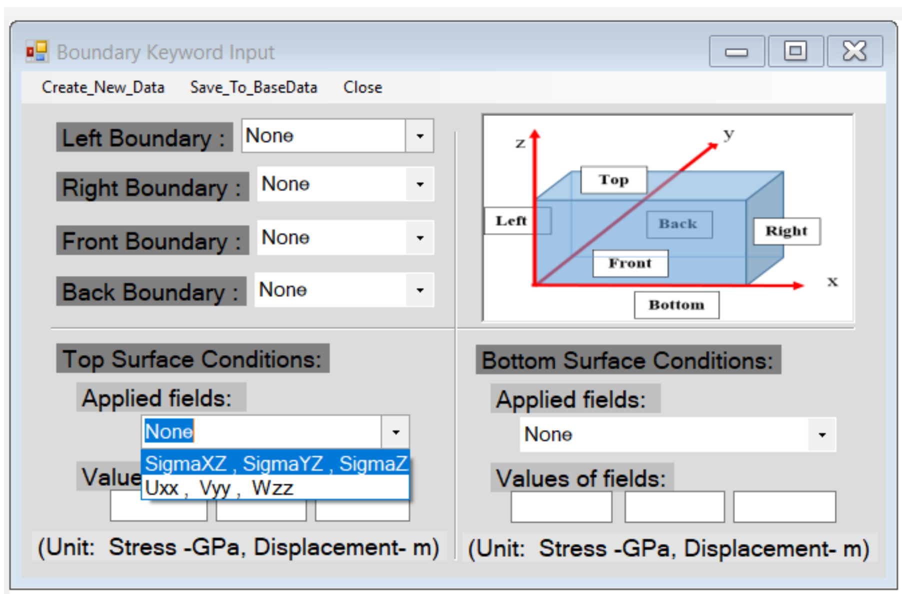


Figure 5-22: Prescribed top and bottom surface condition input for the boundary condition input

Step 2-13: Click the "Simulation" item and select "New Simulation" sub–menu beyond the "BaseData" menu in the main menubar, or, click the "Simulation Setup" and then select "New Simulation Setting" item under the "Simulation Setting" submenu in the side menubar, Figure 5-23 will prompt to allow user to input the necessary data for the *~*simulation keyword*. The detailed meanings of each input item are explained in Chapter 6.11. After finishing the data setting, click "Save_To_BaseData" in the top of the form to add the input data to the basedata.txt file. Only a set of the *Simulation keyword* data is allowed in the file of basedata.txt.

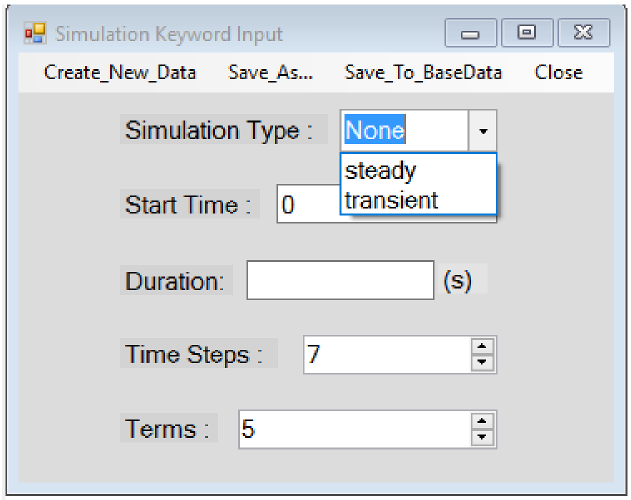


Figure 5-23: GUI for Simulation keyword input

Step 2-14: Click the "Output" sub-menu under the "Basedata" menu and select "New Request", Figure 5-24 will prompt to allow user to input the necessary data for *~*outputdata keyword*. The detailed meanings of each input item are explained in Chapter 7.8. After finishing the data setting, click the "Save_To_Basedata" in the top of the form to add the input data to the file of basedata.txt. Only a set of the *Outputdata keyword* data is allowed in the file of base_data.

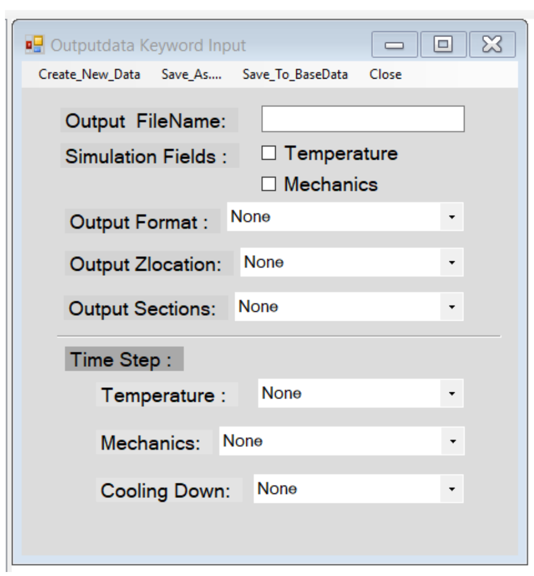


Figure 5-24: GUI input for the Outputdata keyword

If familiar with each item's mean in the basedata.txt and how to create the file, one can create the file of basedata.txt by any text editor. Then, he/she can directly copy the created basedata.txt file to the *inputdata* folder or click "Solving" item and its sub-menu "Check file" and then select "BaseData File" in the main menubar or the side menubar to show the file content in the editor area. After checking the data, click the "Save_To_DataSet" to save the displaying data to the file of basedata.txt in the running directory.

5.3. Step 3: Layer_data file GUI creation

The file of layer_data is to store the deposition information of each layer, include the layer material, slice thickness etc. The file name of the layer_data is given in the "Part Base File" as described in the Step 2. The file of layer_data can be created step-by-step by the following GUI:

Step 3-1: Click the "LayerData" item and select "New LayerData" sub-menu in the main menubar or click "PartPrepare" item and "LayerData" sub-item and then select "New LayerData", the *LayerData Keyword Input form* will show up in Figure 5-25. There are two types of layer property for one layer: (1) *base* and (2) *deposit*. The base layer is referred to the base plate for deposition, and the deposition indicates the layer is an additive layer. This can help user to do with the deposition of the different material in the different layer. And the layer material can be selected from the dropdown list of the materials' name created in previous materials' data input. Generally, the base plate is always a rectangle shape plate, thus, the model requires the input of the layer' length and width. And the layer height is the total deposition height in a same deposition shape/profile. For ambient set number, it will be always set as 0. For the *globalposition term*, it will be always set as (0, 0, 0). The *slicethickness* term is not required for base type layer, but for the deposition layer, one must enter the *slicethickness* data for the model. If not, the software will show a warning message for the input requirement. After finishing the data enter, click "Add_To_LayerData" in the top of the form to save the layer data to the file of LayerData.txt in the *inputdata* folder. If there is more than one layer to deposition, repeat the above step to add a new layer to the file of layer_data which name is given in the "Part Base File".

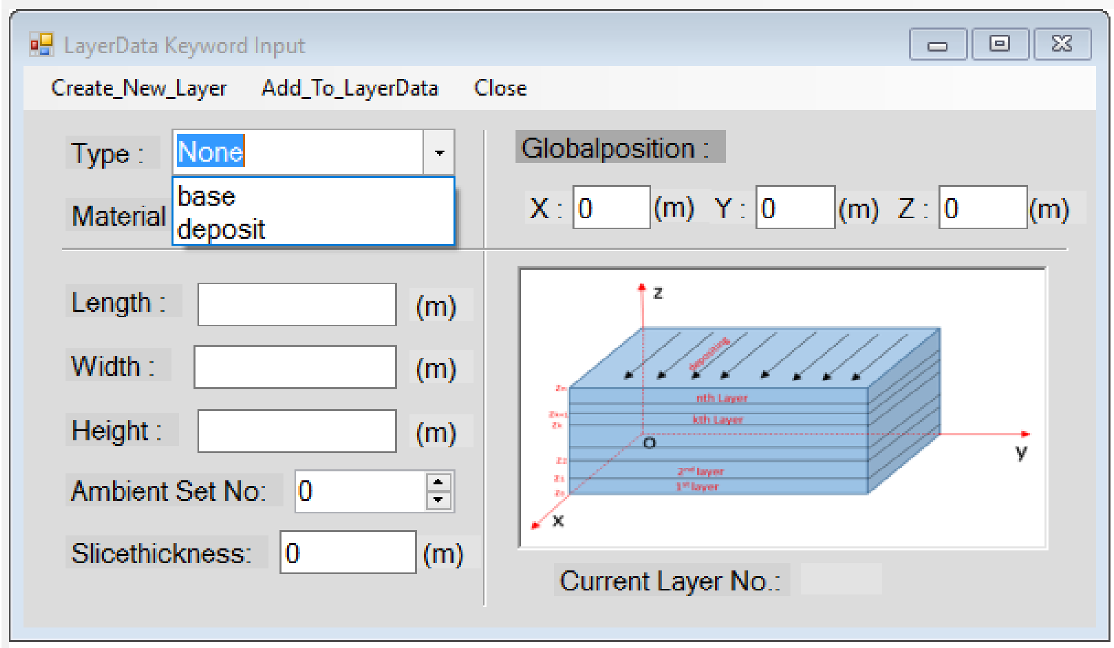


Figure 5-25: GUI for creating layerdata.txt input file.

Step 3-2: Click the "Create_New_Layer" in the top of the *LayerData Keyword Input form* to create the different layer and repeat the *Step 3-1* to fill in the required data and then click "Add_To_LayerData" to add to the file of layer_data which name is given in "Part Base File" till all the layers to be deposited finish. It is noticed that the length and width is same for each layer.

As mentioned above, if familiar with the meanings of each item in the file of layer_data and how to create the file, the user can directly create the file of layerdata.txt via any text editor. Then he/she can copy the created layerdata.txt file to the *inputdata* folder under the installed folder. In order to avoid overwriting the data, some warning message boxes will show up to warm for choosing the next step operation.

5.4. Step 4: Sections_Data file creation

The file of sections_data is used to create the block model for simulation, which name is given in the "Part Base File" The file of sections_data can be created step-by-step by GUI as follows:

Step 4-1: Click "SectionData" item and choose "New SectionData" sub-menu in the main menubar, the *SectionData Keyword Input form* will show up as Figure 5-26. Or, user can reach the same input form from the "New SectionData" sub-menu in "SectionData" item under the "PartPrepare" menu in the side menubar. Firstly, the user must know how many sections will be created with the total number of points. Furthermore, for creating a section, one type of section model must be selected from (1) quadratic section - 8 node model, or (2) cubic section - 12 node model as illustrated in the left bottom of the form. After entering the above data, click the following "Create SectionData" button and the *SectionPoint form* will prompt as shown Figure 5-26.

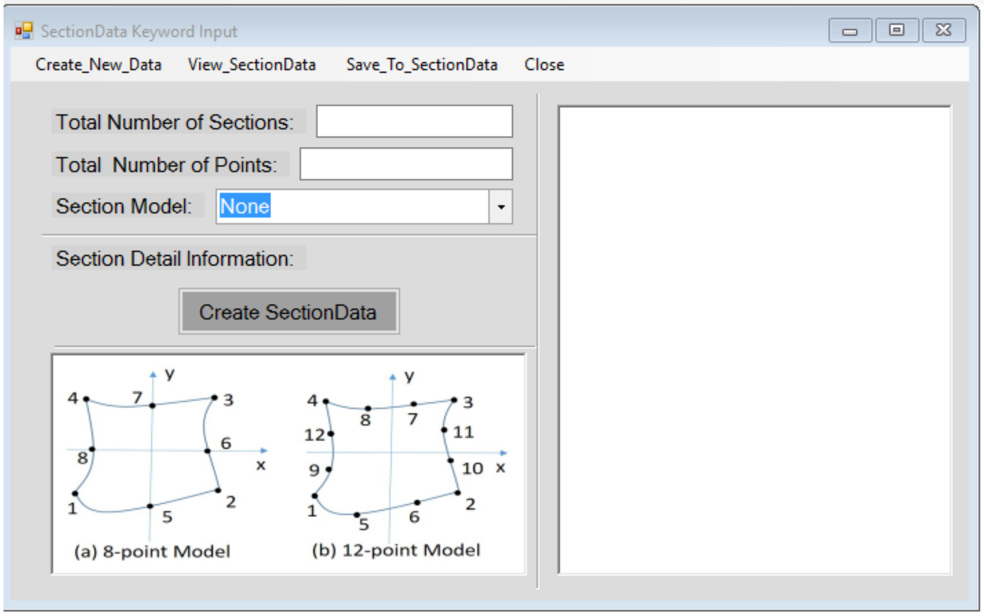


Figure 5-26: SectionData creation GUI for the basic data input

Step 4-2: In Figure 5-27, user can enter the point information for all the points one-by-one in the text box following the Point(x,y) label in the format: Xvlaue ␣ Yvalue or Xvlaue , Yvalue. Or, user can directly load the text file which contains all of the points information listed line-by-line, see Chapter 10. The loaded points are listed in the left bottom of the form and plotted in the right-side drawing panel of the form, as shown in Figure 5-28. After all the points uploaded, one can further create the sections using the input point information by clicking "Create_Sections" in the top menustrip of the form.

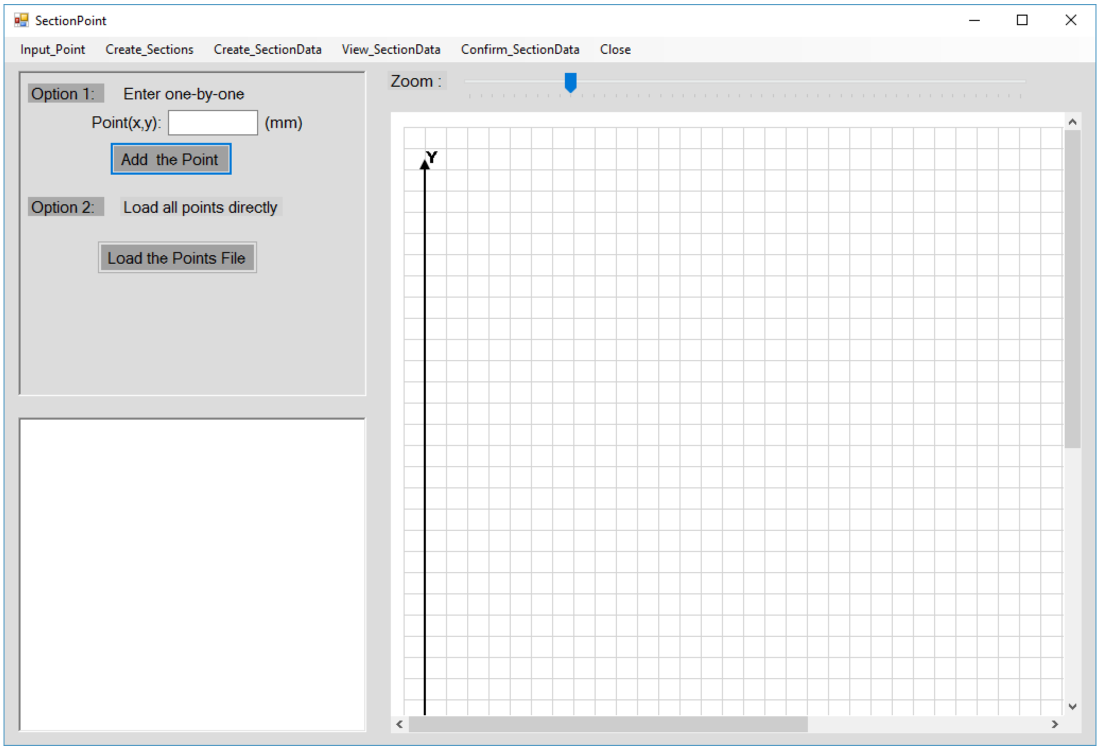


Figure 5-27: SectionPoint Input form and drawing for creating the section information

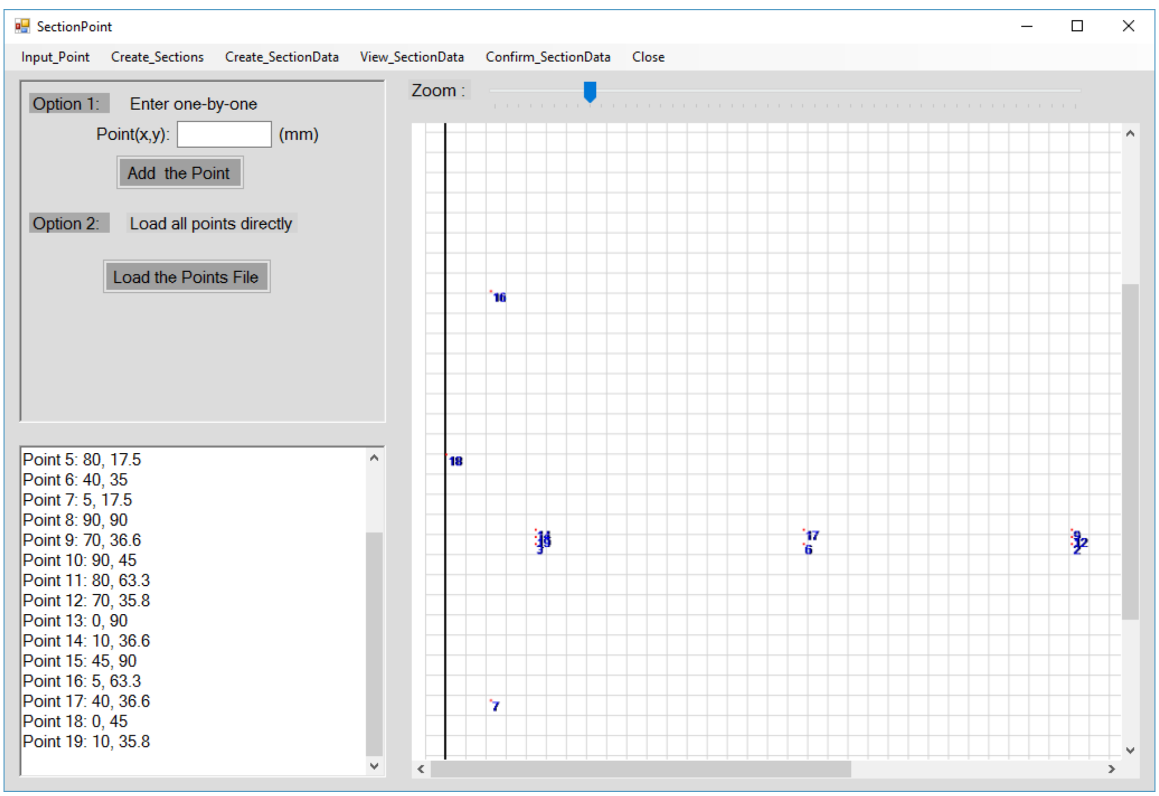


Figure 5-28: Importing and displaying the total points in the drawing panel

Step 4-3: Click "Create_Sections" item in the top menustrip of the form and the form will look like Figure 5-29. The first text box requires the points' set for one section and point number order is based on the selected section model, i.e. quadratic 8-node model and cubic 12-node model. Here the sample is about a quadratic 8-node model. There are two types of section: (1) *base* section where does not have any deposition, and (2) *deposit* section where has deposition. Thus, if the section will be deposited, user must choose the deposit for its section type. Furthermore, any sections must be either boundary section or internal section. A boundary section indicates it must have at least one edge is boundary edge. And an internal section does not have one boundary edge. Thus, for a boundary section, the user must choose the boundary conditions in the dropdown menu for all the boundary edges in the section and other internal edge will choose "*Continuous*" condition. For an internal section, all the section edges can choose "*Continuous*" condition in the dropdown menu. After finishing the above input, click "Create a Section" button to create a section, the righthand side panel will show the created section. Figure 5-30 displays the final results of 5 sections model with one deposition section in the center.

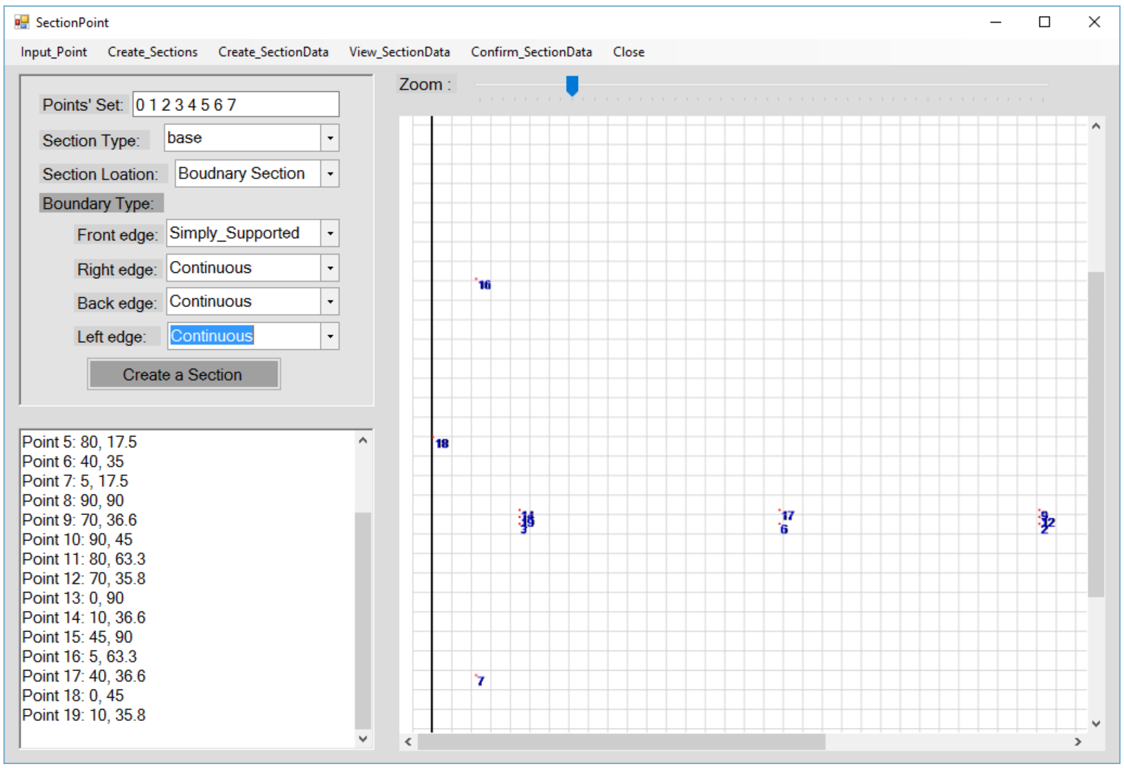


Figure 5-29: Sections creation based on the point information.

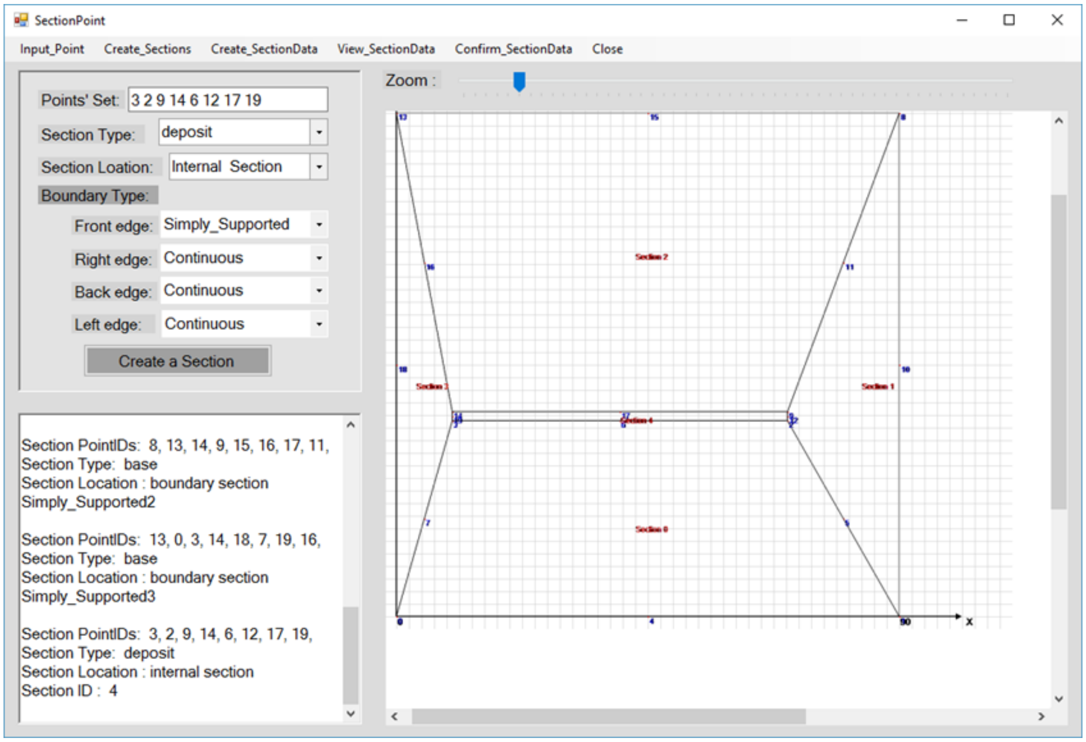


Figure 5-30: Section creation result for a 5-sections model with one central deposition section.

Step 4-4: After finishing the creation of all the sections, click "Create_SectionData" in the top menstrip of the form to calculate and create the final data for the sectionData.txt file. After clicking "Create_SectionData" item in the top menustrip, user can further view the created sectiondata.txt in the left bottom of the form by clicking "View_SectionData". If confirming the created sectiondata information correct, user must click "Confirm_SectionData" to return the created data to the *SectionData Keyword Input form.*

Step 4-5: After *the SectionPoint form* closed, click "View_SectionData" in the top of the *SectionData Keyword Input form* to view and check the returned sectionData in the left side textbox, as shown in Figure 5-31. If the data confirmed, the user can click "Save_To_SectionData" to save the data to the file of the sectiondata.txt. If the file sectiondata.txt existing in current working directory, a warning message box will show up and let user make a decision for the next operation, as shown in Figure 5-32.

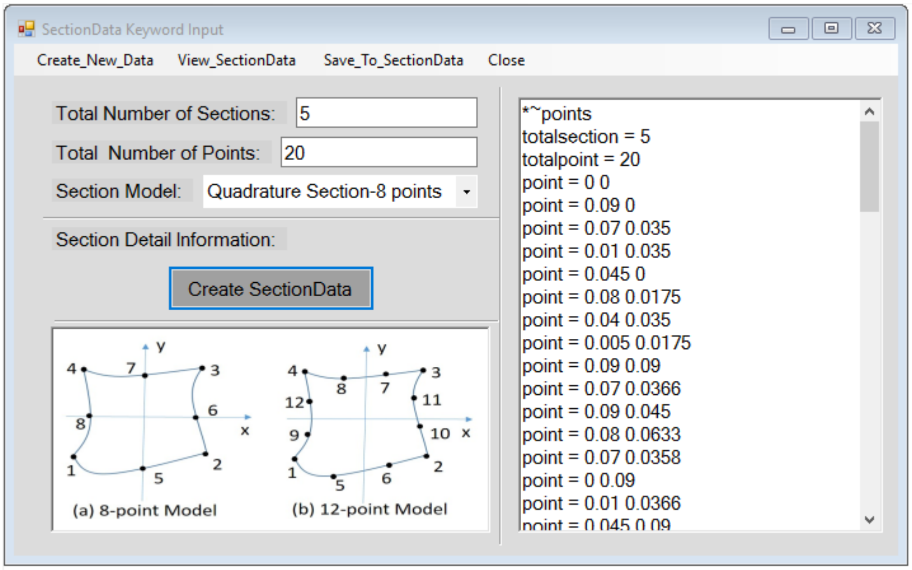


Figure 5-31: SectionData information checking and saving

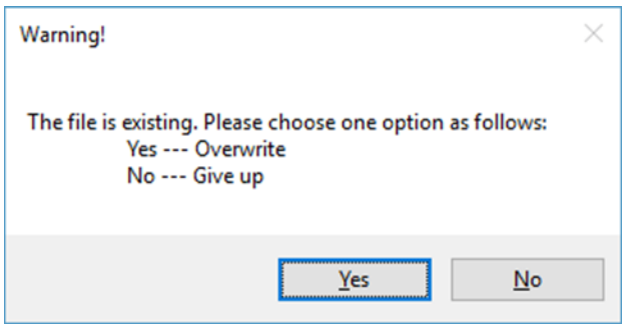


Figure 5-32: Warning if the file of sectiondata.txt existing in current working directory while saving the sectiondata information to the file

5.5. Step 5: Depositpath_data creation

The forth step is to create the depositpath_data which is the deposit strategy applied to the real deposition. The details of the depositpath_data contents are explained in Chapter 11. After the sectiondata was created, one can use the depositpath_data GUI to create and design the deposit strategy. In version 2.0, it can create and design the deposit pattern for polygon section and save to the depositpath.txt for the model simulation. In the current version, the LAMPS software only recognizes the G-Code. But in the future, more general 3D printing codes can be imported and recognized. About creating the file of depoistpath.txt, please refer to Chapter 11 for more information. The following sections are to show how to create the simply deposition pattern through the depositpath_data GUI.

Step 5-1: Click “Deposit Strategy” item and choose “New Deposit Path” sub-menu in the main menubar, Figure 5-33 for the Depositpath form will prompt. Or the user can get the same form from the side menubar by choosing the “New DepositPath” sub-menu in the

"DepositPath" item under the "PartPrepare" menu. User can design the hatch distance, rotation angle with respect to the global coordination, and raster pattern, i.e. the same direction or the opposite direction, and the deposition start side, i.e. from left-hand side or from right-hand side. In current version, only polygon section can be calculated for the deposition pattern. Figures 5-33 and 5-34 display the samples of deposition pattern. The user can design the wanted deposition patterns and further integrate and construct the deposit path for the whole part in the next step.

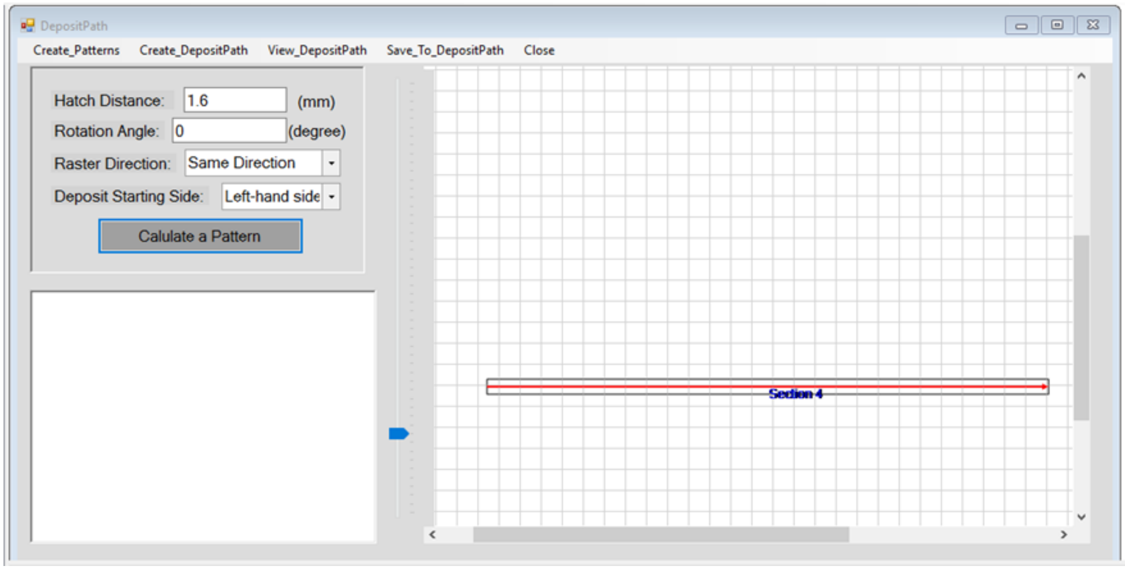


Figure 5-33: Deposit path design in the deposition section

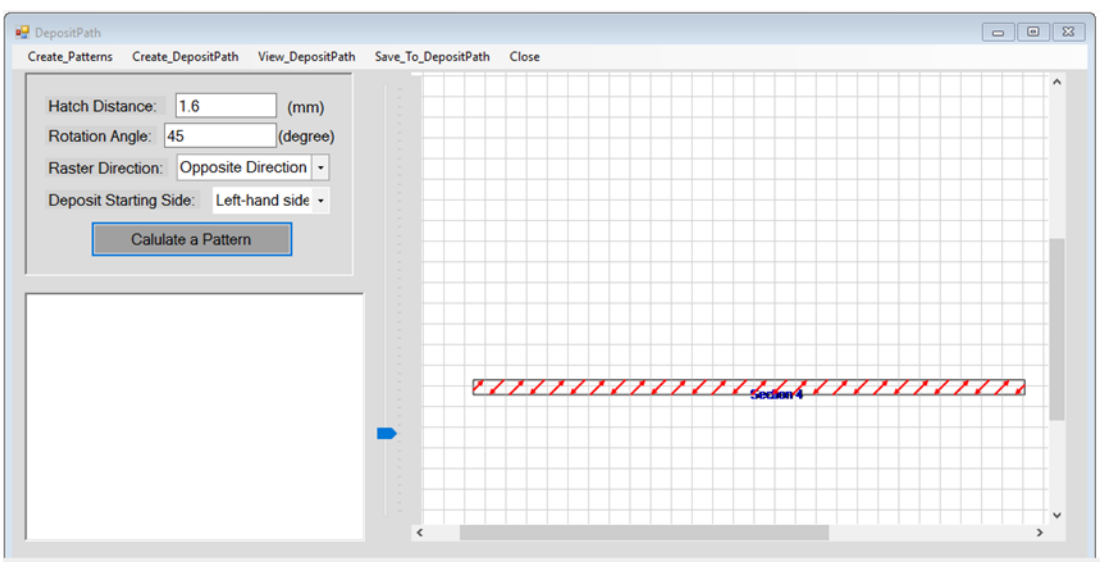


Figure 5-34: Opposite direction deposition pattern

Step 5-2: Click "Create_DepoistPath" in the top menustrip of the form, Figure 5-35 will prompt for creating the deposit path. In the repeat pattern set input, user can enter the pattern No. which details were created in the previous step to create the deposit path. Click "View Parameters" to view the previous input data in the *ProcessParameter Keyword Input form*. Further, click "Create DepositPath" button to construct the final deposit path for the whole part. In order to view the created deposit path, click "View_DepositPath" in the top menustrip of the form to view the deposit path in the left bottom text box of the form, as shown in Figure 5-36. If confirming the deposit path is desired, user can save the data to the file of depositpath.txt by clicking "Save_To_DepositPath" in the top menustrip of the form. If the file of depositpath.txt exists in current working directory, an overwritten warning message box will prompt for warning, as shown in Figure 5-37.

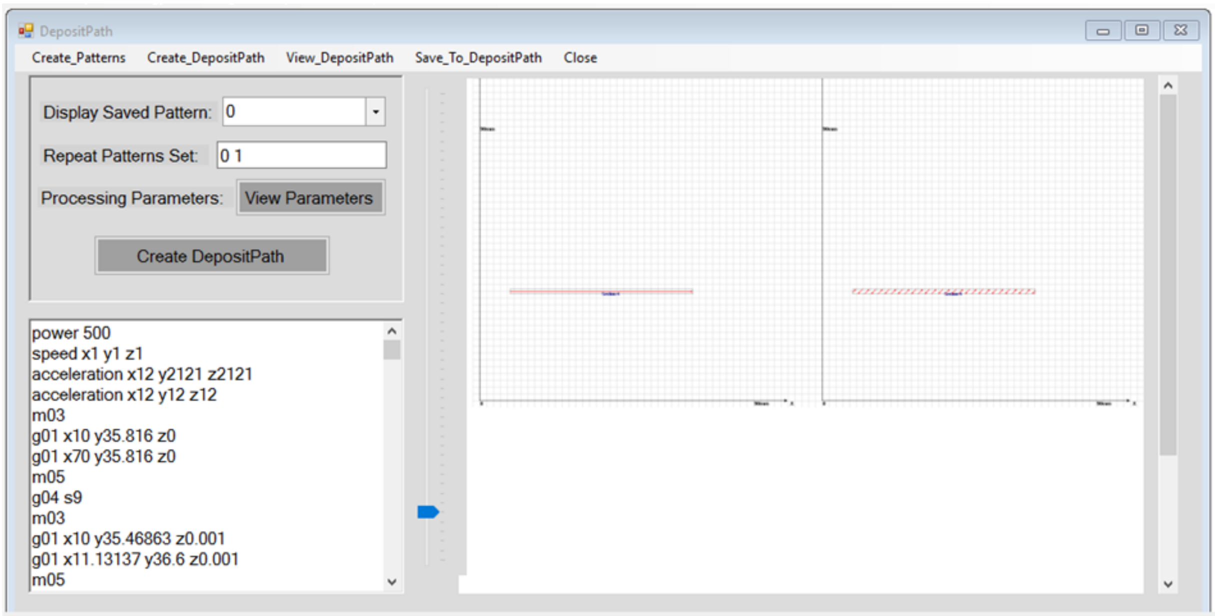


Figure 5-35: Sample of the constructed deposit patterns and deposit path based on the given deposit pattern set.

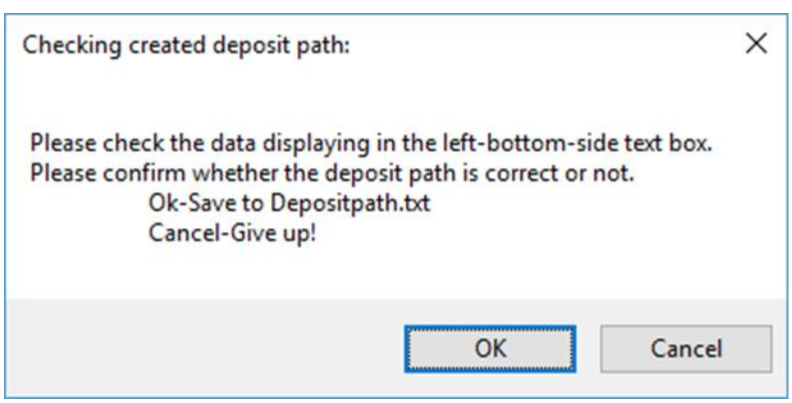


Figure 5-36: Warning message box for overwriting the file of depositpath.txt

5.6. Step 6: Confirmation of the input data for the four input files

The fifth step is to view and confirm the input data for the four input files. Thus, there are four items under the “Check File” sub-menu in the main menubar, include “BaseData File”, “LayerData File”, “SectionData File” and “DepositPath File” as shown in Figure 5-37. Or, user can check the input data files from the side menubar as shown in Figure 5-38. Figure 5-39 displays a sample of the file of base_data; Figure 5-40 shows the data of a file of layers_data; Figure 5-41 presents a sample data of a file of sections_data; Figure 5-42 illustrates the contents of a file of depositpath_data.

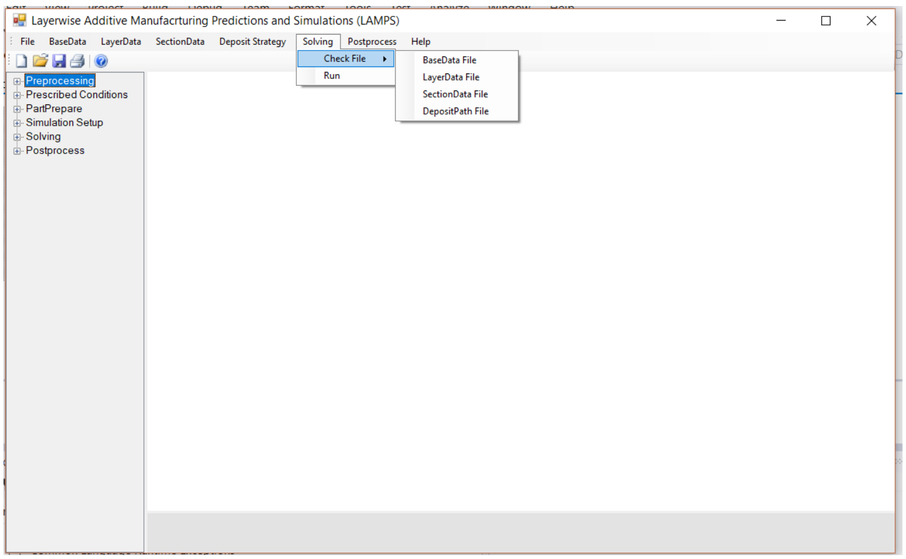


Figure 5-37: Enter the check file from the main menubar

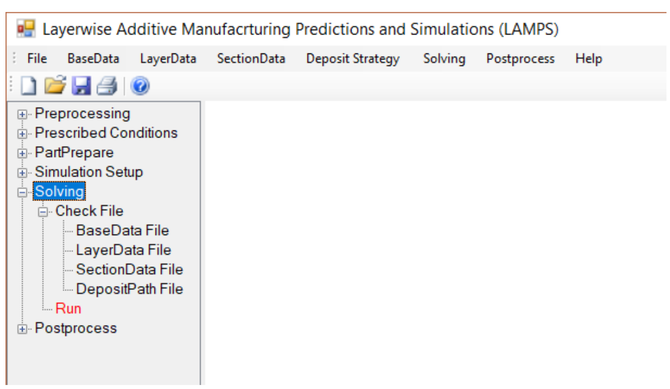


Figure 5-38: Access the check file from the side menubar.

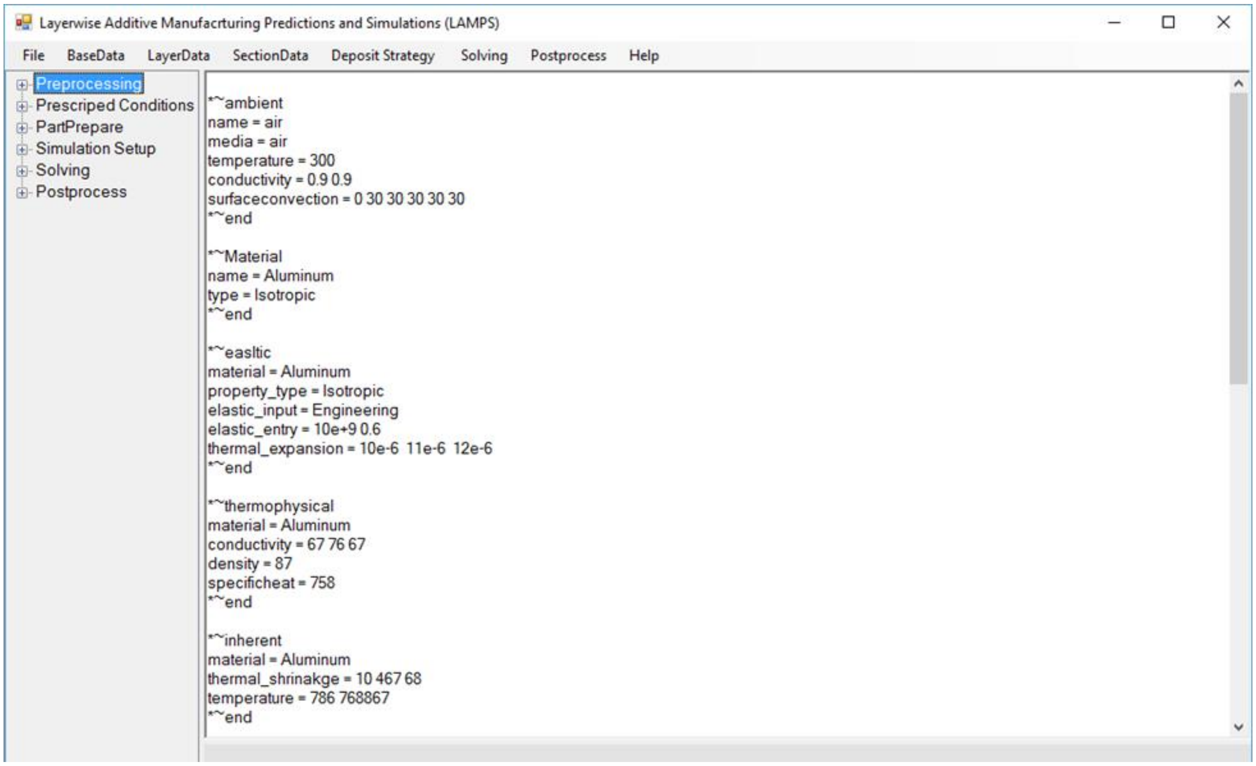


Figure 5-39: Showing, editing and saving the created Basedata.txt in the Editor zone

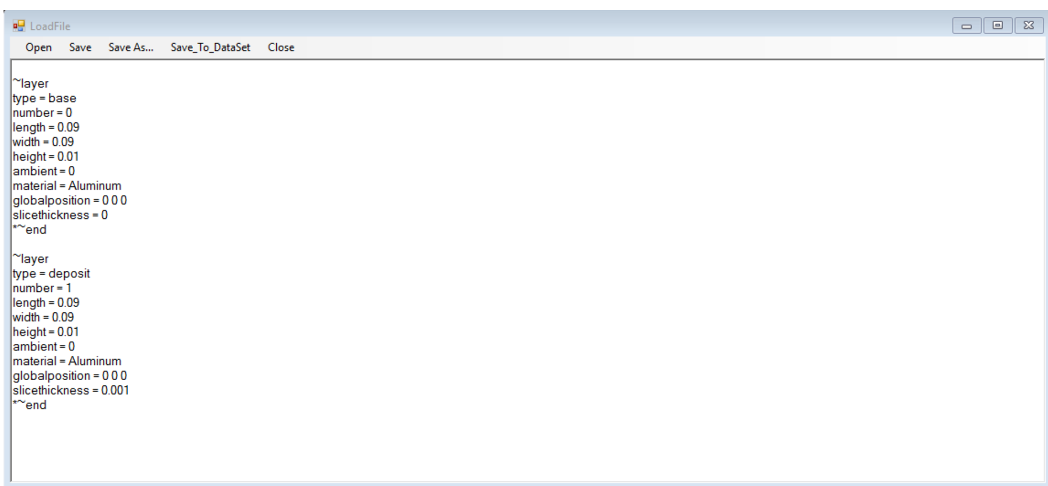


Figure 5-40: Showing, editing and saving the created layerdata.txt in the Editor zone

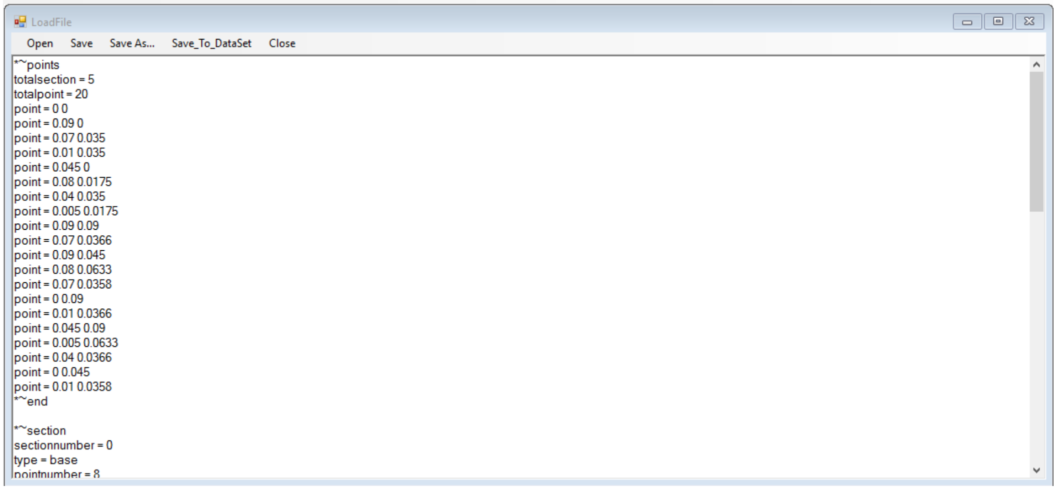


Figure 5-41: Showing, editing and saving the created sectiondata.txt in the Editor zone

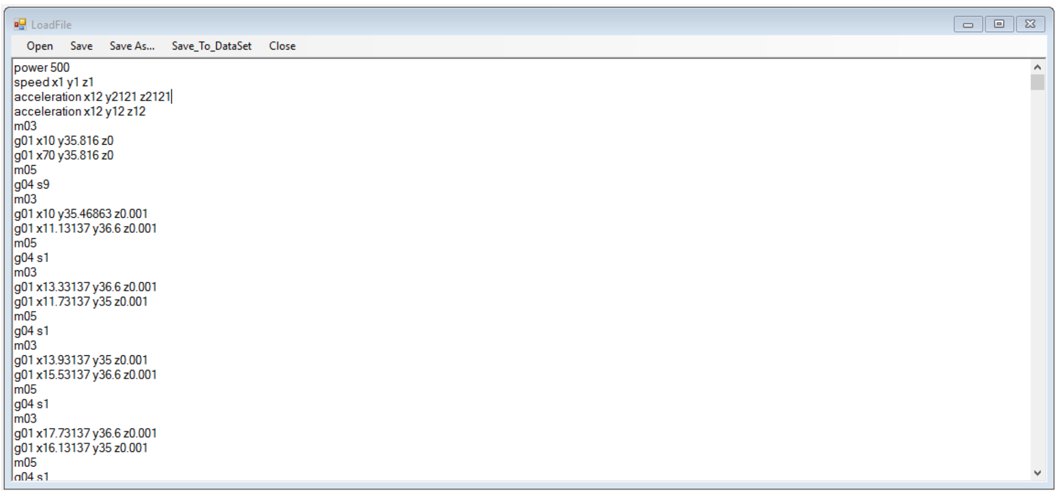


Figure 5-42: Showing, editing and saving the created depositpath.txt in the Editor zone

5.7. Step 7: Run LAMPS

The sixth step is to run the LAMPS simulation after confirming the input data. Click the "Solving" item and select "Run" sub-menu from the main menubar in Figure 5-37 or from the side menubar in Figure 5-38 to run the simulation. The LAMPS runs in a new window as shown in Figure 5-43.

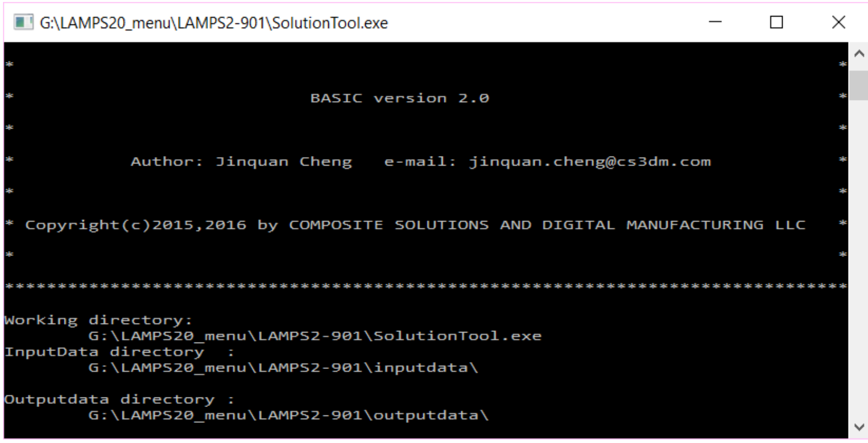


Figure 5-43: Running interface after clicking the "Run" sub-menu in the main menubar or the side menubar.

6. Keywords and Sub-keywords for the CaseType File

CaseType file is to create the modeling type and the file name of the base_data for the simulation, which name is name as "casetype.txt". There are only two parameters (two lines) set up as shown in following format:

For the additive manufacturing modelling,

Line No. 1: am Line No. 2 : Basedata.txt

For the reactive film multilayer modelling,

Line No. 1: rmf Line No. 2 : Basedata.txt

where the input data is obtained from the Step 1 as shown in Figure 5-3. Line No. 1 gives the modelling type. The Line No. 2 is the real file name of the base_data file which is used to store the relevant input data for the simulation.

Noted that the LAMPS software will only inquire the casetype.txt and license.txt for further simulation. If the base_data is not stored in the file named in Line No.2, please revise the base_data file name to that named in Line No. 2.

7. Keywords and Sub-keywords for the Base_data Input File

In the current version of LAMPS software, the input parameters are using two-level data input format, i.e. the **keyword** level and the **sub-keyword** level as defined in Chapter 4. In the keyword level, one keyword can be called or cited by the other keyword. The sub-keywords are used to set the detailed contents for the relevant keyword.

7.1. Part Base File Keyword

Th Part Base File keyword is given by the keyword ***~part** in the base_data file named by the CaseType input and used to define the file name of the materials_data, sections_data and layers_data. Figure 7-1 displays a sample of the format for the keyword *~part,

```
*~part
Materials = materials.txt
Layers = layers.txt
Sections = sections.txt
*~end
```

Figure 7-1: input sample for the keyword *~part to define the data files of materials_data , sections_data and layers_data

7.2. Ambient Keyword

Ambient keyword is given by ***~ambient** in the base_data input file and used to set up the deposit environment conditions. They include the ambient media, the ambient temperature, the media conductivity which could be linear temperature–dependent and the surface convection coefficients for the surfaces. Figure 7-2 gives an input sample of the ambient keyword:

```
*~ambient
name = air
type = air
temperature  = 100.0
conductivity = 0.9414  0.0058
surface_convection = 0.0  30.0  30.0  30.0  30.0  30.0
*~end
```

Figure 7-2: Ambient keyword input

In Figure 7-2, each sub-keyword is explained as

name is to set up the name of current ambient condition sets and its input is characters as,

name ␣ = ␣ media name

Type is to give the ambient media; its input is characters as follows,

type ␣ = ␣ media

Temperature is to give the ambient temperature T_0, which is also the final temperature of the part. And its unit is Kelvin. The input is a number value as follows,

temperature ␣ = ␣ T_0

Conductivity is to enter the conductivity of the ambient media. It can be a constant or temperature dependent presented by Equation (6-1)

$$k = k_0 + CoK * (T - T_0) \qquad (7\text{-}1)$$

where k_0 is the constant conductivity of the ambient media, *CoK* is the temperature dependence coefficient of the conductivity. Thus, the conductivity input is the value of k_0 and *CoK* in the following format:

conductivity ␣ = ␣ k_0 ␣ CoK

For constant case, the *CoK* input will be 0.0 as

conductivity ␣ = ␣ k_0 ␣ 0. 0

surface_convection is to set up the surface convection coefficients for six surfaces of the processing part as shown in Figure 7-2, including the top surface, bottom surface, left surface, right surface, front surface and back surface. Thus the input data are six numbers for the six surfaces in the following order:

surface_convection ␣ = ␣ top ␣ bottom ␣ left ␣ right ␣ front ␣ back

where the convection coefficients of the part surfaces are constant in the present version.

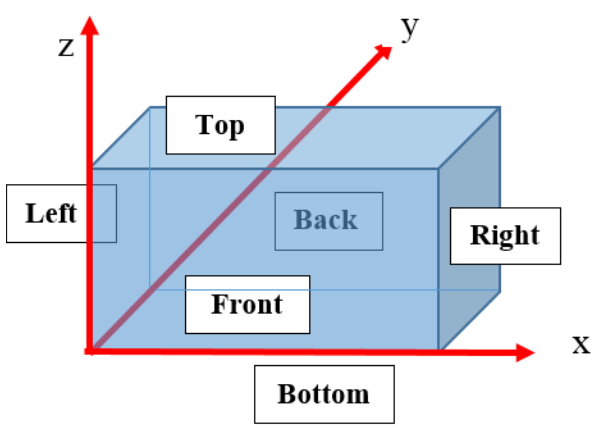


Figure 7-3: Schematic of the six surfaces

7.3. Process Keyword

Process keyword is presented by ***~process** keyword in the basedata.txt input file. It is utilized to set the process type, the tool used in the process and the relevant process parameters, including the scan speed, acceleration and deceleration, the track interval time between two adjacent tracks deposition, the layer interval time between two adjacent layers deposition and the powder feed rate etc. It is noted that some of the input parameters will be modified in the layerdata.txt and depositpattern.txt. Figure 6-3 displays an input sample for the process parameters.

```
*~process
name = LENS
type = powderbed
tool = laser
**speed unit is m/s in three direction
speed = 0.01 0.0 0.0
**acceleration unit is m/s^2 in three direction
acceleration = 1.0 1.0 0.0
**deceleration: unit is m/s^2 in three direction
deceleration = 1.0 1.0 0.0
**track interval: unit is s
track_interval = 0.5
**layer interval: unit is s
layer_interval = 2.0
**powder feed rate : g/min
powder_feed = 4
*~end
```

Figure 7-4: Sample of process keyword input

In Figure 7-4, each sub-keyword is described as the following:

name is to set the name of the process given by the user, which input is characters as

name ␣ = ␣ process name

type is to give which type of the process will be processed, which include powder cladding and powder bed. The selection characters for the type are cited as “powderbed” and “cladding”. The input format could be as follows

type ␣ = ␣ powderbed

or

type ␣ = ␣ cladding

If the process type is set as *powderbed*, the next *powderbed* keyword need more input information for further simulation of the powderbed processing, which is shown in the next Section 6.3. Otherwise, the cladding information will be given beyond the processing parameters.

tool is to give which type of processing tool (heat source) will be used in the process. Currently, there are two tools available to choose, one is the laser and the other is electron beam. Thus, the input formation for tool sub-keyword is as follows tool ␣ = ␣ laser

or

tool ␣ = ␣ EB

speed is the tool moving speeds (v_x, v_y, v_z) in three direction (x, y, z) and the unit is *m*/s. The input format for the speed is as listed

speed ␣ = ␣v_x ␣ v_y ␣ v_z

Acceleration is to input the acceleration of the tool to reach the demanded speed in the three directions(x, y, z) and its unit is m/s^2,. The input format is same to that of the speed as

acceleration ␣ = ␣ a_x ␣a_y ␣ a_z

Deceleration is to give the deceleration in the x-, y- and z- directions for the tool to reduce the speed to zero during the processing and its unit is m/s^2. The input formation is same to that of the deceleration. Noted the deceleration input in the input file is the absolution value of real deceleration in the processing as follows

deceleration ␣ = ␣ a_x ␣a_y ␣ a_z

track_interval is to set the time interval between two adjacent tracks' deposition during the process and its unit is second. The input format is the number of its value,

track_interval ␣ = ␣ value

layer_interval is the time interval between the two adjacent layers depositing and its unit is second. The input format is the number of its value,

layer_interval ␣ = ␣ value

It is noticed that the times for track_interval and layer_interval can be reset in the file of depositpath.txt with considering the scan speed, acceleration and deceleration.

7.4. Powderbed Keyword

Powderbed keyword is presented by ***~powderbed** in the basedata.txt input file. The Powderbed input is for the powder bed processing parameters, including the key parameters: powder bed thickness and powder bed densification in current version. Figure 6-4 shows the main input parameters for the powder bed processing: (1) powder bed thickness and (2) densification.

```
*~powderbed
**bed thickness unit: m
thickness = 100.0e-6
**densification unit: *100%
densification = 0.65
*~end
```

Figure 7-5: An input sample for powder bed keyword.

In Figure 7-5, each one of the sub-keywords is presented by:

thickness is to give the thickness of the powder bed layer to be deposited in the previous deposited part. Its unit is m and the input format is

thickness ␣ = ␣ value

densification is to set the densification of the powder bed which is strong dependent on the powder size and geometry and layer thickness. Its unit is the percentage and 100 percentage denotes the bulky material. The input format is value×100% as follows

densification ␣ = ␣ value

It is noted that the number can't exceed 1 and will be 0 < number < 1. For the powder bed process, the value is round 0.6 or more.

7.5. Cladding Keyword

Cladding keyword is presented by *~cladding in the basedata.txt input file. The Cladding input is for the cladding processing parameters, including the key parameters: powder feed rate, nozzle standoff distance, nozzle powder jet angle, nozzle bias angle, nozzle number, nozzle radius and nozzle central distance in current version. Figure 7-5 shows the main input parameters for the cladding processing as following as

```
*~cladding
**powder feed rate : g/min
powder_feed = 4.0
density = 7800
standoff = 0.10
jet_angle = 34
bias_angle = 5
nozzle_number = 3
nozzle_radius = 0.0015
central_distance = 0.01
*~end
```

Figure 7-6: an input sample for the cladding keyword

Powder_feed is to give the powder feed rate for cladding processing. The powder feed rate is referred to the total powder feed rate from all the nozzles, its unit is g/min. The input formation for the *powder_feed* sub-keyword is limited in the following format as

powder_feed ␣ = ␣ value

density is to present the density of the jetting material. Its unit is kg/m^3 and the relevant input formation is

density ␣ = ␣ value

standoff is the perpendicular distance from the nozzle tip center to the deposition top surface, as shown in Figure 7-7 about nozzle system for cladding. The standoff distance unit is m and in the following formation:

standoff ␣ = ␣ value

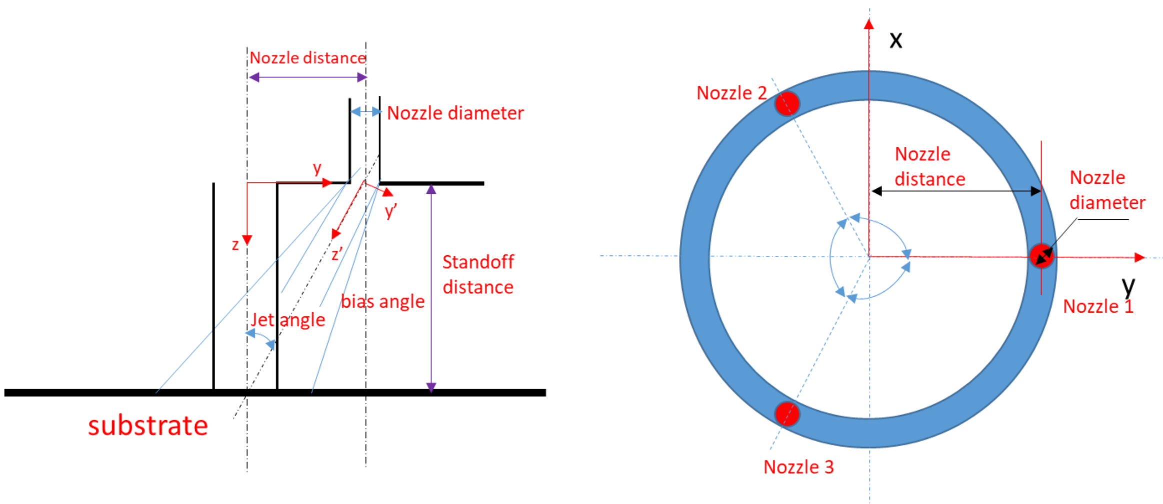


Figure 7-7: The illustration of the nozzle system for a cladding process

jet_angle is the powder jetting angle with respect to the nozzle system central axis (z-axis) as indicated in Figure 7-7. The unit for jet_angle is degree and the input format is

jet_angle ␣ = ␣ value

bias_angle is the nozzle tip bias angle with respect to the nozzle central axis (z'-axis) as indicated in Figure 7-7. The unit for bias_angle is degree and the input format is

bias_angle ␣ = ␣ value

nozzle_number is the total number of the nozzles used in the whole cladding system, its unit is piece of nozzle and its input number is nozzle_number ␣ = ␣ number

nozzle_radius is the radius of the nozzle as shown in Figure 7-7, its unit is m and relevant input is

nozzle_radius ␣ = ␣ value

central_distance is referred to the distance between one nozzle center to the cladding system central axis (z-axis) as shown in Figure 7-7. Its unit is m and the input formation is

central_distance ␣ = ␣ value

7.6. Tool Keyword

Tool keyword is shown as ***~tool** in the basedata.txt input file. It is utilized to set up the tool information used in the additive manufacturing. The input parameters include the type, name, power, heat source radius, heat source penetration depth from the top layer, efficiency of heat source absorbed by the part/particle, starting location in the part and the part geometry profile. Figure 7-8 displays an input sample for the tool keyword.

```
*~tool
type = laser
name = laser
** Power unit : w
power = 25
** radius unit: m
radius = 0.001
** penetration : m
penetration = 0.005
** efficiency *100% for eb 0.99 for laser 0.4
efficiency = 0.4
**location : (x (m),y(m),z(m))--start point
location = 0.045 0.045 0.01
**shape is the profile of the input energy
shape = square
*~end
```

Figure 7-8: Sample of the tool input

In Figure 7-8, each sub-keyword is illustrated by:

Type is to set the heat source type, i.e. laser or electron beam. The two types of heat source supplier are quite different. Thus the additional input is required for the different types of tool in the selections: *laser* or *eb*. The input formation for the *type* sub-keyword is limited in the following format as

type ␣ = ␣ laser

or

type ␣ = ␣ eb

where *laser* denotes the input heat source supplier is laser and *eb* is for the electron beam.

name is to give the name of the tool, which is dependent on the customer' definition. It can benefit for setting up the multiple types of heat sources in the processing. Its input format can be

name ␣ = ␣ laser

or

name ␣ = ␣ laser1

power is to define the input energy of the heat source. Its unit is W and it can be reset in the depositpath.txt to realize the control. The power input is the value of the power and its format is

power ␣ = ␣ value

7.7. Boundary_condition Keyword

The boundary keyword is defined by *~boundary_condition keyword and to enter the mechanical boundary conditions for the six surfaces of laminated plate model for one additive manufacturing part, as indicated in Figure 7-2. There are three types of the in-plane boundary condition considered in the model: (1) simply_supported condition (2) clamped condition and (3) free condition. Their details can be referenced to Chapter 1.2.2. For the out-plane condition, there are two types of given surface conditions can be applied: (1) given stresses (forces) -$\sigma_{xz}, \sigma_{yz}, \sigma_{zz}$ and (2) given displacements u, v, w .

```
*~boundary_condition
** left,right, front, back boundaries are inplane condition free, clamped, simply-supported
left = clamped
right = clamped
front = clamped
back = clamped
** top, bottome:: sigmaxz,sigmayz,sigmazz, or  uxx,vyy,wzz
top = sigmaxz sigmayz sigmazz
top_value = 0.0 0.0 0.0
bottom = sigmaxz sigmayz sigmazz
bottom_value = 0.0 0.0 0.0
*~end
```

Figure 7-9: Sample of the data input for the boundary_condition keyword

In Figure 7-9, the sub-keywords are introduced to set the data as follows:

left is to set the boundary condition for left boundary surface of the part, which is presented in following format:

left ␣ = ␣prescribed condition

right is to set the boundary condition for right boundary surface of the part, which is presented in following format:

right ␣ = ␣prescribed condition

front is to set the boundary condition for front boundary surface of the part, which is presented in following format:

front ␣ = ␣prescribed condition

back is to set the boundary condition for back boundary surface of the part, which is presented in following format:

right ␣ = ␣prescribed condition

top is to set the boundary condition for top boundary surface of the part, which is presented in following format:

top ␣ = ␣σ_{xz}␣ σ_{yz}␣σ_{zz}

or

top ␣ = ␣u_{xx}␣ v_{yy}␣w_{zz}

top_value is to set the detailed value for the given boundary condition in the sub-keyword *top* . The unit for the top_value is dependent on the type of the prescribed condition. If the prescribed condition is stress, the unit will be Pa. For displacement type prescribed condition, it becomes meter. The top_value input data format is presented in following format:

top_value ␣ = ␣value1␣value2␣value3

bottom is to set the boundary condition for bottom boundary surface of the part, which is presented in following format:

bottom ␣ = ␣σ_{xz}␣ σ_{yz}␣σ_{zz}

or

bottom ␣ = ␣u_{xx}␣ v_{yy}␣w_{zz}

bottom_value is to set the detailed value for the given boundary condition in the sub-keyword *bottom*. The unit for the bottom_value is dependent on the type of the prescribed condition.

If the prescribed condition, the unit will be Pa. For displacement type prescribed condition, it becomes meter. The bottom_value input data format is presented in following format:

bottom_value ␣ = ␣value1␣value2␣value3

7.8. Simulation Keyword

The simulation keyword is presented by *~simulation in the basedata.text input file. The function of the simulation keyword main focuses on setting the simulation type, starttime and duration for the transient simulation, the time step number for a duration and the number of the series terms along x and y direction. Figure 7-10 shows an input sample for the simulation keyword.

```
*~simulation
**type = steady
type = transient
start_time = 0.0
duration = 0.4
time_steps = 9
terms = 21
*~end
```

Figure 7-10: A sample of the simulation keyword input

In Figure 7-10, the sub-keywords are described as follows

type is the sub-keyword to set up the simulation type: *steady* or *transient*. The steady simulation type will let the software run the steady –state simulation for a give structure. The transient simulation type will set up an un-steady state simulation for an additive manufacturing procedure. The input for the type sub-keyword is characters as follows

type ␣ = ␣steady

or

type ␣ = ␣transient

starttime is the sub-keyword to set the start time of the simulation for the additive manufacturing procedure. In LAMPS Version 1.0, it will be always set as 0.0. In the future version, it will be set as any other time to restart the simulation. Thus, in the current version, its input format is a constant value as

starttime ␣ = ␣ 0.0

duration is the sub-keyword to set a duration for the block marching time. The duration input is value and its unit is second. As shown in Figure 7-10, the value of the duration is set as 0.4s. The input format is as follows

duration␣ = ␣ value

where it is noted that the duration selection is dependent on the beam size and scanning speed. It may be 0.1s or 5s, which is based on the input processing parameters. In Chapter 10, same samples well illustrate the selection.

timestep is the sub-keyword to give the time step for a duration. The timestep sub-keyword input is an integral number and its input is presented by

timestep ␣ = ␣ number

In general, the number of timestep is set as 7 or 9 which has good simulation accuracy in temporal simulation.

terms is the sub-keyword to set the term number of the series expand for each -axis. The input of the terms sub-keyword is a positive integral number and its formation is as follows

terms ␣ = ␣ number

where it is noted that the numbers for terms could be set from 5 to 21, which can reach a good numerical convergence characteristic. But sometime it is also dependent on the simulation case.

7.9. Outputdata Keyword

Outputdata keyword is to define the output data formation, location and time step for the simulation results, given by ***~outputdata** in the basedata.txt input file. Thus, it includes the following seven sub-keywords: (1) temperature, (2) mechanics, (3) format, (4) zlocation, (5) section (6) timestep and (7) filename. Figure 6.13 presents a sample of the of the keyword outputdata input data.

```
*~outputdata
**two fields temperature, mechanics, both
temperature = yes
mechanics = yes
** two types: matrix or point
format = point
** zlocal: top, deposited or all
zlocation = all
** section: deposited or all
section = deposited
**timestep: all- for all of time step in a duration , last-for the last time step of a
duration, all_last-for the last step for all simulation
timestep = temperature all
timestep = mechanics all_last
filename = al11n
*~end
```

Figure 7-11: A sample of the keyword outputdata input data

In Figure 7-11, each sub-keyword is described as follows

Temperature is the sub-keyword to set whether the software simulate the temperature or not. For additive manufacturing, it must be set to yes since the temperature is the key issue for the distortion and residual stress. Thus, its input format will be in the following format:

temperature ␣ = ␣ yes

mechanics is the sub-keyword to set whether the model will simulate the mechanical field or not. If user want to see the distortion and residual stress, it must set as yes. Thus, its input format is as follows:

mechanics ␣ = ␣ yes

or

mechanics ␣ = ␣ no

format is the sub-keyword to define the simulation result output format. There are two choices: one is *point* and the other *matrix*. The point format lets the simulation results output in point-by-point with the information time, point's coordinate (x, y, z) and the relevant temperature and temperature gradient or cooling rate if the simulation type is set as transient. For *matrix* format output, it only output temperature at each in-plane point (x,y)'s simulation results with the plane's z coordinate. In version 2.0, the data output format for point and matrix is always at vtu format of VTK. Thus the input format for the sub-keyword *format* is set as

format ␣ = ␣ point

or

format ␣ = ␣ matrix

zlocation is the sub-keyword to set up which layer/z-location's data is required to output. In Version 1.0 and 2.0, there are two selections: (1) "*top*" and (2) "*all*". The selection "*top*" denotes only the top surface's information of the part is required to output. And the selection "*all*" means to output all the part's data. Thus, the input format of the location sub-keyword is

location ␣ = ␣ top

or

location ␣ = ␣ all

timestep is the sub-keyword to define which time step is required to output the simulation for temperature simulation and mechanics. In Version 2.0, there are three choices: (1) "last", (2) "all", (3) "all_last". The choice "last" denotes only the last time-step's simulation results are output during a simulation time duration. The choice “all” allows one to output all the data for each time step. The choice “all_last” denotes only of the last step simulation for whole part is output. In order to conveniently set the different simulation field, the timestep sub-keyword has two parameters to set: (1) field (2) timestep choice. Therefore, the input for the sub-keyword-*timestep* is characters as follows

timestep ␣ = ␣ temperature ␣ choice of timestep

timestep ␣ = ␣ mechanics␣ choice of timestep

It is noticed that user can only simulate one type of field, i.e. temperature.

filename is the sub-keyword to set the file name suffix for all the output data required. The final output filename is defined in Chapter 3.2 about the output file.

filename ␣ = ␣ file’s name suffix

8. Keyword and Sub-keyword of the Materials_data File

8.1. Material Keyword

Material keyword is given by ***~material** in the basedata.txt input file. The material keyword is used to set up the name of material and the type of the material, thus its sub-keyword includes name and type as shown in Figure 8-1 for defining an aluminum material.

```
*~material
name = aluminum
type = isotropic
*~end
```

Figure 8-1: Sample of the material input.

In Figure 8-1, each sub-keyword is interpreted as follows:

name is to give the name of the material and used to create a new material. The material name will be cited by the thermophyical keyword and elastic keyword to determine its detail properties. Its input format is as follows

name ␣ = ␣ material's name

type is to set the material elastic behavior type, i.e. *isotropic*, *cubic*, *transverse* (transversely isotropic), *orthotropic* and *monoclinic*. The type will determine the input numbers of the elastic material parameters in engineering parameters. Thus, the input for type will be character of the material type as

type ␣ = ␣ material's type

where the material's type must be one of these: *isotropic*, *cubic*, *transverse* (transversely isotropic), *orthotropic* and *monoclinic*.

8.2. Elastic Keyword

Elastic keyword is given by ***~elastic** in the basedata.txt input file. The elastic keyword is to give the elastic properties of the material which name is given by the sub-keyword material. Thus, the elastic keyword includes the following sub-keywords: material, property, elastic-input and elastic_entry. The elastic keyword must be put behind all the material keyword in order to set up the given material's properties. Figure 8-2 presents a sample of the elastic keyword.

```
*~elastic
material = aluminum
property_type = isotropic
**input type is engineering, stiffness
elastic_input = engineering
** Modulus E and poisson ratio v
elastic_entry = 69.0E+9 0.33
**thermal expansion for three directions x,y,z
thermal_expansion = 13.0E-6 13.0E-6 13.0E-6
*~end
```

Figure 8-2: Sample of the elastic keyword input

In Figure 8-2, each sub- keyword is described in the following:

material is the sub-keyword to give the material name which the elastic properties are belonged to. The input format is character of the material's name as

material ␣ = ␣ material's name

It is noted that the material name should match the material name given by the material keyword part, like the material name in Figure 8-2 is matching that in Figure 8-1.

property is to set the properties type of the material and should be as one of the following five types: <u>*isotropic*</u>, <u>*cubic*</u>, <u>*transverse*</u> (transversal isotropic), <u>*orthotropic*</u> and <u>*monoclinic*</u>, which is same to the type of material given in the material keyword. Its input is characters as shown in above five types.

property ␣ = ␣ type

elastic_input is to define the input data format, i.e. engineering constant input format or stiffness constant input formation. Thus, the elastic_input term has two input choices: (1) <u>engineering</u> or (2) <u>stiffness</u> as shown in

elastic_input ␣ = ␣ engineering

or

elastic_input ␣ = ␣stiffness

elastic_entry is to input the elastic constant values based on the given properties type and elastic_input format in the above sub-keyword. Thus the number of the input data is dependent on the material behavior as shown in Table 8.1. The elastic_entry input format is variable as

elastic_entry ␣= ␣data1␣data2␣ data3….

Table 8.1: The number of elastic_entry data requirement v.s. material property.

Properties / Format	Isotropic	Cubic	transversal	orthotropic	monoclinic
Engineering constants	2	3	5	9	13
Stiffness constants	2	3	5	9	13

thermal_expansion is to input the thermal expansion coefficients in the x-y-and z-axes. For different material, the values can be different from each other. The unit is (*100%/K). The thermal_expansion input format is as follows:

thermal_expansion ␣= ␣Xvalue ␣Yvalue␣Zvalue

8.3. Inherent Keyword

The inherent keyword is given by *~inherent and to input the key parameters for the four-linear-line inherent shrinkage model for additive manufacturing as shown in Figure 1-2. Figure 8-3 displays a sample input of the inherent keyword.

```
*~inherent
material = aluminum
** the shrinakge is for the material
thermal_shrinkage = -0.01 -0.01 -0.03
** temperature unit K
temperature =  523 673
*~end
```

Figure 8-3: Sample of the inherent keyword input

In Figure 8-3, each sub-keyword of the keyword ***~inherent** is explained as follows:

material is the sub-keyword to give the material name which the elastic properties are belonged to. The input format is characters of the material's name as

material ␣ = ␣ material's name

It is noted that the material name should match the one given by the material keyword part, like the material name in Figure 8-3.

thermal_shrinkage is the sub-keyword to enter the shrinkage ratio along x-, y-and z- direction due to cooling down solidification after melting for metal or polymer during the additive manufacturing. It is can be regarded as permanent plastic deformation but it results from the temperature change circle as shown in Figure 1-2. Thus the unit for the shrinkage ratio is *100%. Their values will be always negative. The input format is given by

thermal_shrinkage ␣= ␣XShrinkage ␣YShrinkage␣ZShrinkage

temperature is the sub-keyword to input the two yielding temperatures for the four-linear-line model for the inherent shrinkage. The unit for the yield temperatures is Kelvin. The input format for the yielding temperatures is as follows

temperature ␣= ␣yielding1 ␣yielding2

It is noted that the first yielding temperature is less than the second one.

8.4. Thermophysical Keyword

The thermophysical keyword is given by *~thermophysical in the basedata.txt input file. It is to set up the relevant thermo-physical properties of the materials defined in *~material keyword for heats transfer simulation in additive manufacturing procedure. Thus, it includes the following necessary parameters input in Table 8.2.

Table 8.2: Thermo-physical input parameters and their units

Input Parameter	**Unit**	**Input parameter**	**Unit**
Solidus temperature	K	Thermal conductivity	W/(m*K)
Liquidus temperature	K	Solid density	kg/m^3
Evaporation temperature	K	Liquid density	kg/ m^3
Solid specific heat	J/(kg*K)	Latent heat of fusion	J/kg
Liquids specific heat	J/(kg*K)	Latent heat of evaporation	J/kg
Dynamic viscosity	N/(m*s)	Convection coefficient	W/(m^2 *K)

In Figure 8-4, each sub-keyword is explained as follows:

material is to give the material name which the thermos-physical properties belong to. Thus, its input is a string and must fit the material name given under the material keyword. The following data set of the thermophysical properties is assigned to the material named by this sub-keyword and the format is listed as

material ␣= ␣ material's name

```
*~thermophysical
material = Ti-64
solidus = 1877.0
liquidus = 1923.0
boiling = 3315.0
evaporation = 3533.0
latentfusion = 2.86e5
latentevaporation = 9.83e6
**the n linear segment relationship: temp, a00, a01(mill, 2002)
conductivity_entry = 1268.0 1.2595 0.0157
conductivity_entry = 1923.0 3.5127 0.0127
conductivity_entry = 1923.0 -12.752 0.024
**the n linear segment relationship: temp, a00, a01 (mills 2002)
viscosity_entry = 1923 0.01  -4.0e-6
**the n linear segment relationship: temp, a00, a01
density_entry = 1268 4465.892 -0.154
density_entry = 1923 4465.892 -0.154
density_entry = 1923 7227.64 -0.68
** the n linear segment relationship: temp, a00, a01
specificheat_entry = 1268 483.04 0.215
specificheat_entry = 1923 412.7 0.1801
specificheat_entry = 1923 831.0 0.0
**the combined coefficient for radiation and convection based on Kumar
paper: temp , a00, a01
convradiation_entry =  1268 -5.7431 0.0225
convradiation_entry =  1923 -21.331 0.0352
convradiation_entry =  1923 -36.533 0.0431

*~end
```

Figure 8-4: Input sample of thermophysical keyword

solidus is to set up the solidus temperature of the material. Its unit is Kelvin (K) and the input format is

solidus ␣= ␣ value

liquidus is to set up the liquidus temperature of the material which is the temperature point of material becoming liquid phase from solid phase. Its unit is Kelvin (K) and the input format is liquidus ␣= ␣ value

boiling is to input the boiling temperature of the material which is the temperature point of liquid material starting to boil. Its unit is Kelvin (K) and the input format is

boiling ␣= ␣ value

evaporation is to give the evaporation temperature of the material. Its unit is Kelvin (K) and the input format is

evaporation ␣= ␣ value

latentfusion is to input the latent heat of fusion for the material and its unit is J/(kg*K). Its input format is given by

latentfusion ␣= ␣ value

latentevaporation is to set the latent heat of evaporation for the material and its unit is J/(kg*K). Its input format is given by

latentevaporation ␣= ␣ value

conductivity_entry is to give the heat conductivity of the material. Its unit is W/(m*K). As is well-known, the heat conductivity can be temperature dependent, especially for the metal at high temperature range. Thus, it can be constant or temperature dependent presented by Equation (8-1)

$$k = k_0 + Cok * (T - T_0) \quad (8\text{-}1)$$

where k_0 is the constant heat conductivity of the material, *Cok* is the temperature dependence coefficient of the conductivity and T_0 is the start temperature point for the relevant coefficient. Thus, the conductivity_entry input requires three parameters in the following format:

conductivity_entry ␣ = ␣ T_0␣ k_0 ␣ Cok

For the constant case, the K input will be 0.0 and T_0 will be any value as follows

conductivity_entry ␣ = ␣ T_0␣ k_0 ␣ 0. 0

In Figure 8-4, the data of conductivity_entry for Ti-64 is based on the data of Mills 2002's book[6].

viscosity_entry is to give the dynamical viscosity of the material in liquid phase. Its unit is N/(m*s). It is well-known that the dynamical viscosity could be temperature dependent, Thus, it can be a constant or temperature dependent presented by Equation (8-2)

$$Visco = Visco_0 + CoV * (T - T_0) \quad (8\text{-}2)$$

where $Visco_0$ is the constant dynamic viscosity of the material, *CoV* is the temperature dependence coefficient of the viscosity and T_0 is the start temperature point for the relevant coefficient. Thus, the viscosity_entry input requires three parameters in the following format:

viscosity_entry ␣ = ␣ T_0␣ $Visco_0$ ␣ CoV

For the constant case, the K input will be 0.0 and T_0 will be any value as follows

viscosity_entry ␣ = ␣ T_0␣ $Visco_0$ ␣ 0. 0

In Figure 8-8, the data of viscosity_entry for Ti-64 is on the basis of the data in Mills 2002's book[6].

density_entry is to give the density of the material at different temperature ranges. Its unit is Kg/m^3. It is seen that the density is temperature dependent, especially for high temperature range. Thus, it can be constant or temperature dependent presented by Equation (8-3)

$$\rho = \rho_0 + Co\rho * (T - T_0) \tag{8-3}$$

where ρ_0 is the constant density of the material, Coρ is the temperature dependence coefficient of the density and T_0 is the start temperature point for the relevant coefficient. Thus, the format of the density_entry input is same to the above conductivity entry as follows:

density_entry ␣ = ␣ T_0␣ ρ_0 ␣ Coρ

For the constant case, the K input will be 0.0 and T_0 will be any value as follows

density_entry ␣ = ␣ T_0␣ ρ_0 ␣ 0. 0

In Figure 8-4, the density_entry data for Ti-64 is on the basis of the data in Mills 2002's book[6].

specificheat_entry is to set the material specific heat in different temperature ranges. Its unit is J/(kg*K). It is seen that the specific heat of the material could be temperature dependent. Thus, it can be a constant or temperature dependent presented by Equation (8-4)

$$C_\rho = C_{\rho 0} + CoC * (T - T_0) \tag{8-4}$$

where $C\rho_0$ is the constant specific heat of the material, CoC is the temperature dependence coefficient of the specific heat and T_0 is the start temperature point for the relevant coefficient. Thus, the specificheat_entry requires three input data in the following format:

specificheat_entry ␣ = ␣ T_0␣ $C\rho_0$ ␣ CoC

For a constant case, the CoC input will be 0.0 and T_0 will be any value as follows

specificheat_entry ␣ = ␣ T_0␣ $C\rho_0$ ␣ 0. 0

In Figure 8-4, the data of specificheat_entry for Ti-64 is on the basis of the data in Mills 2002's book[6].

convradiation_entry is to set the combined convection and radiation coefficient in additive manufacturing procedure. The combined convection and radiation coefficient is introduced here to simplify the simulation complexity of the convection and radiation effect since the radiation is four order power function of the temperature. The details can see the Reference of Kumar et al.'s paper[7]. In order to better simplify, a three-linear relationship is used to represent the nonlinear relationship like other input parameters. It is presented by Equation (8-5)

$$ConRad \quad = ConRad_0 + CoCR\ (T - T_0) \tag{8-5}$$

where $ConRad_0$ is the constant convradiation_entry of the material, CoCR is the temperature dependence coefficient of the combined convection and radiation coefficient and T_0 is the start temperature point for the relevant coefficient. Thus, the convradiation_entry need three input data as follows:

convradiation_entry ␣ = ␣ T_0␣ $ConRad_0$ ␣ CoCR

If neglecting the radiation effect, the $ConRad_0$ input will be 0.0 and T_0 will be initial value as follows

convradiation_entry ␣ = ␣ T_0␣ 0.0 ␣ CoCR

In Figure 8-4, the data of convradiation_entry for Ti-64 is in terms of the data of Kumar 2014's paper [6].

It is noted that the above some parameters are temperature dependent and isotropic thermophysical property. However like composite material, the thermophysical properties are anisotropic i.e. in some directions, the thermo-physical values are greater than the others. Further in additive manufacturing, the powder and bulky material properties are different and solidified part is also different from host material. Thus the input parameter format should be different from the isotropic one as described above. Some new sub-keywords have to introduce to describe these differences as discussed in the next Section 8.5.

8.5. Sub-keywords for Constant Anisotropic Thermo-physical Behavior

In order to consider the anisotropic thermos-physical behavior of a material, the new input format is introduced to allow the LAMPS can handle with anisotropic thermos-physical behavior. Figure 8-5 display a sample of how to set up the anisotropic thermos-physical behavior problem.

```
*~thermophysical
material = cfrp
**conductivities in three directions are constant k11,k22,k33
conductivity = 26.8 0.96 0.96
density = 1097
specificheat = 800
*~end
```

Figure 8-5: A sample of the input for an anisotropic thermophysical behavior problem

In Figure 8-5, some new sub-keywords are introduced to set the data as follows:

conductivity is to set the conductivities (k_x, k_y, k_z) in the three axes (x, y, z). For the isotropic thermophysical behavior material, the conductivities in three directions are same i.e. kx=

ky= ky, for anisotropic material, they must be different. The input format for conductivity can be

conductivity ␣ = ␣ vaule of k_x ␣ value of k_y ␣ value of k_z

density is to set the constant density of the material, thus its input is only one value as follow

density ␣ = ␣value

specificheat is to set the constant specific heat of the material and its input formation is complete different from the specificheat_entry as follow

specificheat ␣ = ␣value

9. Keywords and Sub-keywords for the Layers_data File

Since the layerwise additive manufacturing is a procedure of line-by-line and layer-by-layer depositing the material into the base plate/part, the input data requires that each layer information to fabricate the whole part. Thus, the keyword for the layerdata.txt is ***~layer** as shown in Figure 9-1 with the certain sub-keywords, include type, number, zone and length etc. parameters required by running the simulation.

```
*~layer
type = base
number = 0
length = 0.09
width = 0.09
height = 0.0003
globalposition = 0.0 0.0 0.0
slicethickness = 0.0003
material = aluminum
ambient = 0
*~end
```

Figure 9-1: A sample of the layerdata.txt input file.

In Figure 9-1, each sub-keyword is listed as follows:

type is to set the layer type. The definition of the layer type means which part section of the deposited part the layer belongs to, there are two choices: (1) base and (2) deposit. The base layer will be used to build up the baseplate for deposition. Since the baseplate material properties and geometry have a significant effect on the temperature distribution, it is very important to know them. The current version software can handle with the laminated baseplate through the type. Thus, the input formation for the type sub-keyword is

type ␣ = ␣base

or

type ␣ = ␣deposit

number is to set whether layer in the relevant part section this layer input belongs to. Thus, the number input is a positive number from 0 as shown as follows

number ␣ = ␣ layer No.

length is to set the length of the layer. In the current version, only the rectangular shape layer can be processed and its unit is *m*. But in the future, the software can handle the different shapes of the deposit layer. The input format for the length is

length ␣ = ␣ value

width is to set the width of the rectangle layer and its unit is *m*. The input format is as

width ␣ = ␣ value

height is to give the height of the rectangle layer and its unit is *m*. The input format is

height ␣ = ␣ value

globalposition is to set the start absolute position in the global coordination system of the model and its unit is *m*. This is convenient to allow us to set the deposition start point in the deposition layer with respect to the baseplate section. The input format for the globalposition sub-keyword is the position of the start point (x, y, z) in the following format:

globalposition ␣ = ␣ x ␣ y ␣ z

slicethickness is the thickness of the single depositing layer and its unit is *m*. It is used to determine how many single depositing layers are deposited for the part layer. The input format of the slicethickness sub-keyword is

slicethickness ␣ = ␣ value

material is to set the name of the layer material which detailed properties have been given in the basedata.txt input file. The input of the material sub-keyword is character as follows

material ␣ = ␣ material's name

ambient is to set the ambient condition for the layer. The input is the number of the ambient based on the basedata.txt input file. In the current version, the value of the ambient sub-keyword is always set as 0 since there is only one ambient condition in the basedata.txt input file as follows

ambient ␣ = ␣ 0

10. Keywords and Sub-keywords for the Sections_data File

Since the deposition is based on the 2D profile to deposit and build up the whole part layer-by-layer, it is necessary to create the section information for running the simulation. Thus, the sections_data file will provide the section information, including the section profile and deposition or not etc. Figure 10-1 shows a sample of the data in a sections_data file, which include 5 sections and total 20 points with quadratic 8-node model. In the file of sectiondata.txt, it contains two keywords: (1) ***~point** and (2) ***~section**, which details are explained in the next sections.

10.1. Point Keyword

Since a block technique is applied to model the deposition part in current software, the technique only requires a few points to model and simulate the thermo-mechanical behavior of the deposition part. For example, a 5-section model only needs 20 points to achieve the simulation. As shown in Figure 10-1, the sub-keywords for the Point keyword are explained as follows:

totalsection is to set the total number of the sections in one layer. Thus, before inputting the point information, the user must know how many sections will be used to model the part. The input format for the sub-keyword *totalsection* is an integer number as follows:

totalsection ␣ = ␣ total number of sections

totalpoint is to set the total number of the points used to model the above sections. The total number of points is dependent on the selected section model: 8-node model and 12-node model. The input format for the sub-keyword *totalpoint* is an integer number in the following format:

totalpoint ␣ = ␣ total number of points

point is to set the point coordinate for one point in a line. The unit for point coordination is meter. The input format for the sub-keyword *point* is in the following format:

point ␣ = ␣ xCoordination␣ yCoordination

where the total lines of sub-keyword point must be equal to the input of the sub-keyword totalpoint.

```
*~Points
totalsection = 5
totalpoint = 20
point = 0.0 0.0
point = 0.09 0.0
point = 0.07 0.035
point = 0.01 0.035
point = 0.045 0.0
point = 0.08 0.0175
point = 0.04 0.035
point = 0.005 0.0175
point = 0.09 0.09
point = 0.07 0.0366
point = 0.09 0.045
point = 0.08 0.0633
point = 0.07 0.0358
point = 0.0 0.09
point = 0.01 0.0366
point = 0.045 0.09
point = 0.005 0.0633
point = 0.04 0.0366
point = 0.0 0.045
point = 0.01 0.0358
*~end

*~section
sectionnumber = 0
type = base
pointnumber = 8
pointset = 0 1 2 3 4 5 6 7
** front back left right
front  = -1 clamped
back = 4 front
left = 3 right
right = 1 left
*~end
```

Figure 10-1: A sample of the data for sectiondata.txt

10.2. Section Keyword

The section keyword is used to set up a section based on previously input point information, as shown in Figure 10-1. There are total 8 sub-keywords to build up a section, (1) sectionnumber (2) type, (3) pointnumber, (4) pointset, (5) front, (6) back, (7) left and (8) right, which meanings are introduced as following:

sectionmumber is to the No. of the section in the total sections. Thus, its input format is a positive integer as follows:

sectionnumber ␣ = ␣ Number of the section

type is the sub-keyword to set the type of the section and includes two choices: base and deposit. The type depends on whether the section is a deposition one or not. If the section is a deposition one, the type will be set as deposit otherwise, it will be set as base. Thus, the input format of the sub-keyword is represented as

type ␣ = ␣ base

or

type ␣ = ␣ deposit

pointnumber is the sub-keyword to set the number of the point used to model a section. For a simulation, it must keep same for any sections. Current, since there are only two types of section models developed, pointnumber must be 8 or 12 like the following format for input:

pointnumber ␣ = ␣ 8

or

pointnumber ␣ = ␣ 12

pointset is the sub-keyword to set up the point set to build up a section based on the selected section model. The point set is in order as shown in Figure 8-2 for the selected section model, and the point number is based on the point order of the points in the keyword point. Thus, the input format for the sub-keyword *pointset* is in the following format:

pointset ␣ = ␣No.1 ␣ No.2␣No.3 ␣ No.4␣No.5 ␣ No.6␣No.7 ␣ No.8

or

pointset ␣ = ␣No.1 ␣ No.2␣No.3 ␣ No.4␣No.5 ␣ No.6␣No.7 ␣ No.8␣No.9 ␣
No.10␣No.11 ␣ No.12

front is the sub-keyword to set the boundary condition of the front edge in a section which is defined as the curve between No.1 and No.2 as shown in Figure 10-2. The sub-keyword is followed by two values: first is the edge adjacent section number and second is which edge is the edge inside the adjacent section. If the edge is a boundary edge, the first value will be set as -1, and the second value will be the real boundary condition one of three general boundary conditions: (1) simply_supported condition, (2) clamped condition and (3) free condition. Thus, the input format for the sub-keyword *front* is as

front ␣ = ␣No. of adjacent section ␣ edge

or

front ␣ = ␣-1 ␣ boundary condition

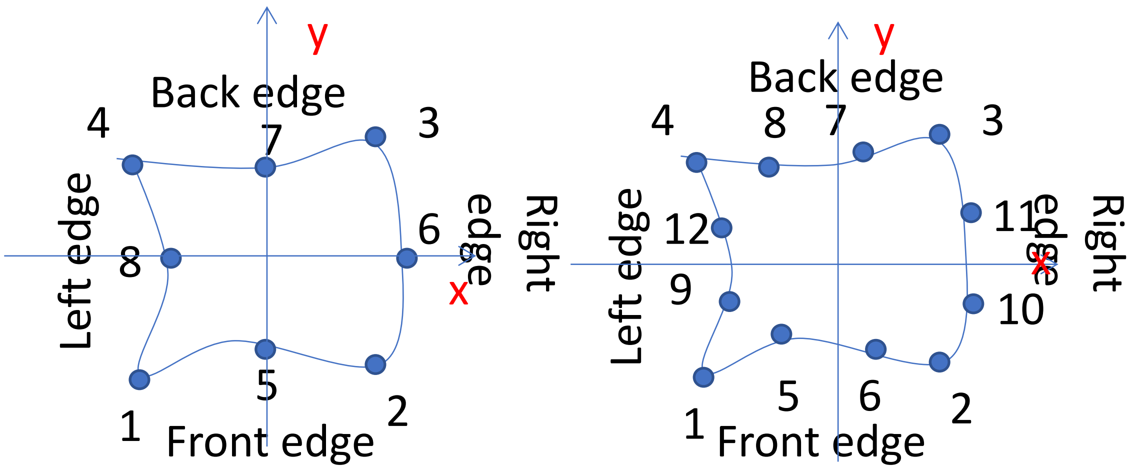


Figure 10-2: Order of the point set for one section based on (a) the 8 node model and (b) the 12-node model.

back is the sub-keyword to set the boundary condition of the back edge in a section which is defined as the curve between No.3 and No.4 as shown in Figure 10-2. Its input format is in a same manner to the sub-keyword *front* as follows:

back ␣ = ␣No. of adjacent section ␣ edge

or

back␣ = ␣-1 ␣ boundary condition

left is the sub-keyword to set the boundary condition of the left edge in a section which is defined as the curve between No.4 and No.1 as shown in Figure 10-2. The input format of the sub-keyword *left* is in a same manner to the sub-keyword *front* as follows:

left ␣ = ␣No. of adjacent section ␣ edge

or

left␣ = ␣-1 ␣ boundary condition

right is the sub-keyword to set the boundary condition of the right edge in a section which is defined as the curve between No.2 and No.3 as shown in Figure 10-2. The input format is in a same manner to the sub-keyword front as follows:

right ␣ = ␣No. of adjacent section ␣ edge

or

right␣ = ␣-1 ␣ boundary condition

It is noticed that the above relationship among the sections can be built up by using the Sectiondata GUI Creation as shown in Chapter 5.4.

11. Keywords and Sub-keywords for the Depositpath_data File

As above mentioned, the additive manufacturing is a procedure of line-by-line and layer-by-layer depositing the material into the base plate/part. Therefore, the deposit path/pattern is very important. In order to emulate the deposition path/pattern, a depositpath_data input is created based on the G-Code formatting for the current version LAMPS. For more information about the G-Code, please refer to the wiki link via https://en.wikipedia.org/wiki/G-code. Table 11.1 presents some of the G-Codes used in the depositpath_data input file.

Table 11.1. G-code and M-code used in depositpath_data file

Code	Description	Corollary information
G00	Rapid positioning	On 2- or 3-axis moves, G00 (unlike G01) traditionally does not necessarily move in a single straight line between start point and end point. It moves each axis at its max speed until its vector is achieved.
G01	Linear interpolation	The program specs the start and end points, and the control automatically calculates (interpolates) the intermediate points to pass through that will yield a straight line (hence "linear").
G02	Circular interpolation, clockwise	the interpolation generates a circle rather than a line. As with G01, the actual toolpath of the machining takes place with the given feedrate on a path that accurately matches the ideal (in G02's case, a circle) to within very small limits.
G03	Circular interpolation, counterclockwise	Same corollary info as for G02.
G04	Dwell	Takes an address for dwell period (may be X, U, or P). The dwell period is specified by a control parameter, typically set to milliseconds. Some machines can accept either X1.0 (s) or P1000 (ms), which are equivalent.
M03	Power on	This indicates to switch on the power supplier of the laser/EB heat source.
M05	Power off	This indicates to switch off the power supplier of the laser/EB heat source.

F	Define the feed rate	This is used to define the input power in the current deposit. This is easy to realize the power control in simulation.

Based on the above G-Code information, the depositpath.txt input can be set up. Figure 11-1 shows a sample of the G-Code for LAMPS simulation. There are some key input parameters for running the LAMPS software as follows:

```
Line 1:   power 25.0
Line 2:   ** speed (x,y,z) unit : m/s
Line 3:   speed  x0.01 y0.01 z0.01
Line 4:   acceleration x0.1 y0.1 z0.1
Line 5:   deceleration x0.1 y0.1 z0.1
Line 6:   ** g0 location (x,y,z, power on/off: m03 - on; m05- off )
Line 7:   gcode
Line 8:  m03
Line 9:   g01  x0.0 y0.0 z0.0 F25.0
Line 10:   g01  x0.09 y0.0 z0.0 F25.0
Line 11:  m05
Line 12:  g00  x0.0 y0.00 z0.0
Line 13:  g00  x0.0 y0.003 z0.0
Line 14:  m03
Line 15:  g01 x0.09 y0.003 z0.0 F25.0
Line 16:  m05
Line 17:  g00  x0.09 y0.006 z0.0
Line 18:  g00 x0.0 y0.006 z0.0
```

Figure 11-1: A sample of the part of a depositpath.txt input file.

Power in the 1^{st} line

Power is to set the input power for the deposit but it can be changed during the deposit by using F function in g01 code line.

Speed in the 3th line

Speed is to set the speed in the three axes (x, y, z). Following the speed keyword, there are the speed in each direction denoted by the first letter. Thus, the setting format for the speed is in the following format

Speed ␣ X*** ␣Y***␣Z***

For example,

Speed ␣X0.01 ␣Y0.01␣Z0.01

where X0.01 means the speed along x-axis is 0.01m/s, y0.01 indicates the speed along y-axis is 0.01m/s, z0.01 represent the speed along z-axis is 0.01m/s.

Acceleration in the 4th line

Acceleration is to set the acceleration of the head moving from a static status to a desired speed at the three directions (x, y, z). Following the acceleration keyword, there are the respective acceleration in each direction denoted by the first letter. The input format is

acceleration ␣ X*** ␣Y***␣Z***

deceleration in the 5th line

deceleration is to set the deceleration of the head reducing the speed to zero at the three directions (x, y, z). Following the deceleration keyword, there are the respective acceleration in each direction denoted by the first letter. The input format for the deceleration is in the following format:

deceleration ␣ X*** ␣Y***␣Z***

m03 in the 7th line

m03 is the keyword to switch on the power of heat source. The following procedure will have energy input on to the part till the *m05* keyword occurs.

g01 in the 8th line

g01 is the keyword to allow the header linearly move from the previous position to the target position defined by the following x***, y***, z*** and the power input value will be re-set by F*** in terms of the pre-set speed, acceleration and deceleration. Thus, the input format is

g01␣ X*** ␣Y***␣Z***

or

g01␣ X*** ␣Y***␣Z***␣F***

where the values behind x, y and z characters present the target position coordination. F*** is to re-set the input power value for the heat source. If there are not F*** following the new position coordinate, the input power will not be reset.

m05 in the 10th line

m05 is the keyword to switch off the power supplier of the heat source.

G00 in the 11th line

g00 is the keyword to allow the header linearly move from the previous position to the target position defined by the following x***, y***, z*** in a high speed which is set to 20 time of the speed given by the speed line. But it can be reset.

12. Validation and Samples

This Chapter presents some simulation samples to numerically validate the developed LAMPS model and software and further confirm its performance and capability for envisioning the additive manufacturing procedure. It is noted that current version software works for constant thermos-physical behavior material cases. For Nonlinear problem will be added in the future version.

12.1. Application for steady-state simulation

In order to verify the developed layerwise analytical model, the first sample is a laminated hybrid plate that is subjected to a locally concentrated heat flux as shown in Figure 12-1. The details of the geometries of the plate are a = 0.04m and b = 0.01m and Table 10.1 presents the construction and thermal conductivities of the five layers laminated plate. Along the edges and at the bottom surface, the convections are set to α_c = 30W/m^{2}K and T_∞= 0°C. At the top surface adiabatic conditions were assumed outside the heat flux of q= 100kW/m^2 [8]. First, the convergence characteristic of the number if the xterms and y terms can be studied. The input files is shown in the installation folder "..\samples\compoiste_plate_steady_problem". The user can change the parameters and testify it, like free-edge effect, energy intensity effect, absorption efficiency effect, and deposit location etc. factor's effect on the temperature distribution. Please check Table 12.1 for more information.

Table 12.1. Construction and thermal conductivities of hybrid plate

Layer	**Thickness (mm)**	**Material**	
1	0.3	Aluminum	
2	0.5	CFRP[0°,90°]$_s$	
3	0.3	Aluminum	
4	0.5	CFRP[0°,90°]$_s$	
5	0.3	Aluminum	
Material	**k_{xx}(W/mK)**	**K_{yy}(W/mK)**	**k_{zz}(W/mK)**
UD-CFRP layer	26.208	0.96	0.96
Aluminum	235.0	235.0	235.0

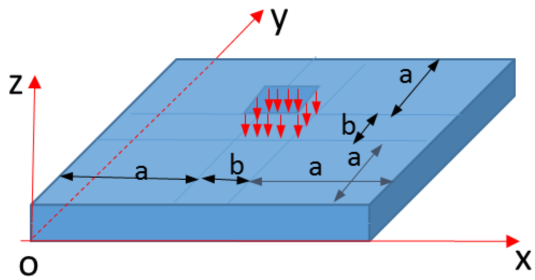


Figure 12-1: Schematic view of a laminated plate subjected to a heat flux.

12.2. Application for unsteady-state simulation by moving heat source problem

In terms of Chapter 12.1's sample structure, this sample focuses on the moving heat source problem in the composite materials/structures. The three input files are included in the folder "..\samples\compoiste_plate_transient_moving_heat". The user can test it to check the effect of the process parameters and material properties and structure geometry effect on the transient heat transfer problem in the composite structure. Please check Table 12.2 for more information.

Table 12.2. The simulation capability of LAMPS in Version 1.0

Processing parameters	steady		transient		
	temperature	Temperature gradient	temperature	Temperature gradient	Cooling rate
Scan speed			√	√	√
Input energy	√	√	√	√	√
Beam radius	√	√	√	√	√
Absorption efficiency	√	√	√	√	√
Deposit materials	√	√	√	√	√
Part geometry	√	√	√	√	√
Deposition location	√	√	√	√	√
Deposit strategy /path			√	√	√

Table 12.3. The simulation capability of LAMPS on mechanical fields in Version 2.0

Processing parameters	steady			transient		
	displacement	strain	stress	displacement	strain	stress
Scan speed				√	√	√
Input energy	√	√	√	√	√	√
Beam radius	√	√	√	√	√	√
Absorption efficiency	√	√	√	√	√	√
Deposit materials	√	√	√	√	√	√
Part geometry	√	√	√	√	√	√
Deposition location	√	√	√	√	√	√
Deposit strategy /path				√	√	√

12.3. Application for studying Additive Manufacturing procedure

On the basis of the Sample 1's information and structure, this sample will show how to use the LAMPS software to study the additive manufacturing through adding the depositing layer information and deposit strategy to input file. The detailed input files are listed in the folder: "..\samples\composite_plate_additive_manufacturing". The user can modify and test the problem. In order to save the computation time, the user can use lower number of the terms and increase the duration value in the *~simulation keyword to speed the simulation.

13. References

1. Mani Mahesh, Feng shaw, Lane Brandon, Donmez, Moylon Shawn, Ronnie Fesperman, Measurement Science needs for real-time control of additive Manufacturing powder bed fusion processes, NISTIR8036, 2015.
2. Van Elsen Maarten, Complexity of selective laser melting: a new optimization approach, Ph.D. dissertation, Katholieke Universiteit LEUVEN, Belgium, 2007.
3. Zhao CX, Kwakernaak C., Pan Y., Richardson I.M, Saldi Z, Kenjeres S., Kleijn, CR. The effect of oxygen on transitional Marangoni flow in laser spot welding, Acta Materialia, 58: 6345-6357, 2010.
4. Shen NG, Chou K, numerical thermal analysis in electron beam additive manufacturing with preheat effects, MS&T 2013, Montreal, Canada, Oct 27 - Oct. 30 2013.
5. Leonard F., Tammas-williams S., Prangnell PB, Todd I., Withers PJ., Assessment by X-ray CT of the effects of geometry and build direction on defects in Titanium ALM part, MS&T 2013, Montreal, Canada, Oct 27 - Oct. 30 2013.
6. Mills, K. C. (2002). Recommended Values of Thermophysical Properties for Selected Commercial Alloys, Woodhead Publishing Ltd, ISBN 978-1855735699, Cambridge.
7. Kumar A, Paul CP, AL Pathak, et al., A finer modeling approach for numerically predicting single track geometry in two dimensions during laser rapid manufacturing, Optics & laser Technology, 44: 555-565, 2012.
8. Kambly K., Characterization of curing kinetics and polymerization shrinkage in ceramic-loaded photocurable resins for large area markless photopolymerization (LAMP), Master thesis for master of Science in Mechanical Engineering, Georgia institute of Technology, 2009
9. Jang C.D, Kim Y.T., Jo Y.C., Ryu H.S., Welding distortion analysis of hull blocks using equivalent load method based on inherent strain, 10^{th} International Symposium on practical Design of ships and other floating structures, Houston Texas, 2007.
10. Liu H, Sparks T., Liou F., Dietrich D.M., Resident stress and Deformation modeling for metal additive Manufacturing Processes, Proceeding of the world congress on Mechanical, Chemical, and material Engineering (MCM 2015)., Page : 245-1~245-9
11. Cheng B., Ping L, and Chou K, 2014, Thermomechanical investigation of overhang fabrication in electron beam additive manufacturing, Processing of ASME 2014 international Manufacturing Science and Engineering Conference(MSEC) Detroit, MI, June 9 -13, 2014, MSEC2014-4063.