

# Final Report

I PRO 330 – Operator Information Tool to Manage Heat Treat Furnace Parts for A.Finkl  
& Sons  
Illinois Institute of Technology, Spring 2006

## **Project Objectives:**

- To generate an accurate survey of temperature distribution within the furnace.
- To determine a method of stacking that provides the most uniform temperature distribution.
- To create an efficient and intuitive tracking system that allows the exact location of each part in the furnace to be recorded and catalogued for future reference.
- To put this knowledge together to create a software package that can be used to quickly and easily generate a stacking sequence that will result in a proper heat treatment for each piece in the furnace each time.
- To fulfill all I PRO requirements

## **Project Background:**

The engineers at Finkl operating the heat treatment facility need to optimize the furnace load configurations in order to maximize the number of pieces which may be heat treated without the necessity of an additional heat treatment due to inadequate tempering.

The engineers would like to have a simple way to visualize and manipulate the loading configuration of pieces of steel inside a virtual furnace. The heat treat Foreman could then key in a furnace load number at his computer or handheld device and see a block diagram of the optimized configuration to load the furnace. The load configuration file provides electronic recordkeeping that is helpful in recalling past heat treat jobs and reviewing how the pieces were arranged inside the furnace.

This project will consist of two parts. The first part is to create a software package to represent the 3-D world inside a furnace in 2-D in order to plan the stacking of parts in a fixed space so that the heat distribution is more or less uniform over each piece in the load. This representation has to be presented to the operator visually so that he can select and manipulate graphical representations of the part shapes available from a menu of shapes. Since the operator needs this ability during the stacking operation, the program should be run on a PDA or a local PC. Possible 3-D solutions could involve defining a surface or solid model in a file, which can be generated by a CAD package (Pro/Engineer, Unigraphics, SolidEdge, etc.). This file permits verification and review of heat treatment conditions and troubleshooting and will also be used in the next step of the process to simulate the heat treatment.

In order to simulate the heat treatment process it is necessary to have real data on the time-temperature profile of parts in different configurations. These data will be obtained by thermo-coupling test blocks and running them through real heat treatments with other parts. The data will be analyzed to obtain heat transfer coefficients related to position in the furnace and load configuration.

## **Methodology:**

The team will work closely with Finkl employees on this project throughout the semester and several trips to the Finkl plant in Chicago are planned. The team has identified the specific tasks at hand as follows:

- 3D CAD drawings of the furnaces, the loading zone, and the actual parts being treated must be generated.
- Furnace surveys must be conducted and used in conjunction with the drawings to determine ideal loading conditions.
- A basic software program must be written that can take these drawings and use them to catalogue the stacking sequence in the furnace for a given treatment cycle.
- Ultimately, this software must be adapted to allow a stacking sequence to be determined prior to loading that guarantees a proper heat treat for each item in the furnace.

The first three tasks are being tackled simultaneously, while the final item consists of a synthesis of these three tasks. All the data and information required to accomplish this is either in the team's possession now or can be found within Finkl's database. Therefore, we are confident that we will be able to accomplish all these tasks successfully and in full.

## **Results:**

- The temperature distribution in an empty furnace has been successfully surveyed:

Two team members went to Finkl and mounted 19 thermocouples inside the empty furnace. The thermocouples were distributed so as to get sample air temperatures throughout the entire furnace. Data was obtained at four different temperatures.

The data was then analyzed in Matlab using a program written by one of the team members. Temperature distributions were plotted in 3D using several slices across the furnace. This can be seen in Figure 1. The data obtained suggest that the distribution inside the furnace is quite even and should not generally be the source of a mistreat. As such, the focus of the project has shifted from trying to correct the temperature distribution inside the furnace to generating a stacking software package for use by Finkl operators. A furnace survey of a loaded furnace was completed as well. Temperature distributions, combined with

tempering curves obtained from Finkl for their allows, indicate that nothing is wrong with this furnace.

- CAD drawings of the furnace  
Team members went to Finkl and obtained dimensions for the furnace and for the loading zone. We used this information to generate CAD drawings and 3D models of the furnace setup. We can then reference these drawings when developing the representation of the furnace in our stacking software.
- Researched different visualization software☺

After extensive research into the options below, we resolved to create the software package from scratch using the VTK library for C++.

#### Visualization Software – AnCAD MATFOR® GUI System

- A. <http://www.ancad.com/overview.html#overviewmenu>
- B. Must get license from this company – are in contact already
- C. Capabilities:
  1. Can see actual geometry of the furnace and structure – brick, steel, etc.
  2. Can change parameters such as material, size, position, can also add menus file, output
  3. Can operate using manual options in the program
  4. Can do animations
  5. Ability to use Matlab kernels we have written and develop them into Fortran or C/C++, and compile using the program.
- D. On website, has module to convert Matlab code into Fortran or C++
  1. <http://www.ancad.com/codeconverter.php>

#### Visualization Software - OpenInventor

- A. <http://oss.sgi.com/projects/inventor/>
- B. Free 15 day trial for academic institutions
  1. Open source – might be an issue for Finkl
  2. Can perhaps look into a non-open source version.
- C. Capabilities:
  1. Can build 3D solid models – does it already have collision detect?
  2. Has built in sets of objects such as cubes, polygons, text,
  3. Can change materials
  4. Has cameras, lights, trackballs, handle boxes, 3D viewers, and editors
  5. Windows system, but platform independent
- Wrote an late-beta version of the software.  
In the first attempt at developing the Stacking Software, we used C# and the .NET Framework. C# combines the power of C/C++ and the productivity of Visual Basic and greatly reduced development time. It seemed like a good idea to develop our graphical application using C# and SharpGL (an OpenGL library for .NET). However, after only a few days into development, we faced problem

which could not be easily solved with the standard OpenGL Library. The outer axes of the furnace had to be marked with graphical text representing X Y and Z axes. For this to work properly, the axes had to be mapped on the outer boundaries of the furnace. After discussing this problem with Dr.Hu, it was decided that we will have to use VTK (Visualization Toolkit) to render all furnace data and the outer axes. VTK provides a great framework for rendering polygons and volumes, however, in order to use it the software had to be written in C++. Taking into consideration the tools and flexibility offered by VTK, we made the switch from C# to C++.

- **Compiled literature on heat treatment:**  
To better understand the problem at hand, the team has gathered background information on the theory and application of steel heat treatments. Such references are requisite to making qualified claims concerning the possible sources of mistreat. The most comprehensive sources for this particular project are Physical Metallurgy Principles (Reed-Hill and Abbaschian, PWS Publishing Company, 1994) and Mechanical Metallurgy (Dieter, McGraw-Hill, 1986). The team has now moved on to collecting information more specific to the tempering process.

### **Future projects:**

The following are areas into which this project can be expanded:

- Portable Handset Display
  - Find and develop a handset that can run the stacking software program quickly and easily.
  - Has to have real-time capabilities with uploading and downloading.
- Enhance Shape Data Base
  - Design all of the different shapes of pieces that are used today.
  - Increase sensitivity in collision awareness for odd shaped pieces
  - Design a method for user to make new shapes easily.
- Furnace Testing
  - Run more temperature distribution tests in different furnaces.
    - Small Furnaces (size)
    - Large Furnaces (size)
  - Run tests using different stacking sequences.
  - Look into changing location of controller thermocouple.

## **Software Accomplishments:**

### **AutoStack**

The software is designed to graphically represent in three dimensions the stacking of steel parts to be heat treated in a furnace. This work is to be used to automate stacking of these very large parts as well as provide a data base of heat treated parts for tracking. Since parts are occasionally out of specification, the ability to determine if a furnace stacking problem is responsible will prove invaluable. The visualizations will also help facilitate optimum furnace operation and heat distribution among all parts.

Models residing in software will track the internal conditions of the part as it travels through the furnace and determine the optimum furnace parameters. The system will use real-time calculated conditions of the part to fine-tune furnace parameters to attain optimum furnace conditions and obtain quality parts.

At this moment, it is virtually impossible to tell the actual difference in the furnace and part temperatures during an operation, and this system will solve this problem by providing a temperature map of each individual part within the furnace at any point in the heating/cooling cycle. The part profile is continuously updated based on the furnace data. Determination of furnace operating parameters can be done on-line; the system will calculate the parameters dynamically by feeding all design inputs and operating data to the models.

The furnace is represented as a wire cube with arbitrary dimensions ranging from 0 to 250 in the X Axis, 0 to 100 in the Y Axis and 0 to 50 in the Z Axis. Steel blocks are represented as solid cubes with user specified dimensions. Every part is represented in 3 Dimensions. When a new part is introduced in the furnace, its outer bounds are checked against the furnace boundaries. If the part falls outside the furnace in any of the three dimensions, a warning is issued. Moreover, each new block is checked against all existing blocks for possible collisions. When a collision is detected, the user is informed accordingly.

As of now, the placement of each part is defined by the user. If a part does not fit in the furnace, it can be placed diagonally. The ultimate goal to analyze all the parts inside the furnace and re-arrange them so that every part gets equal heat treatment.

The Overview screen provides a picture of the overall process. Background graphics, animation and informative displays will accurately depict the heat treating process in real time. Other screen details include the name of the furnace, number of heating and cooling zones, the current user and their access privileges. The Faceplate screen allows a user to modify process parameters to control the furnace temperature and the position of all the blocks.

## Collision Detection

AutoStack uses a three-stage collision detection architecture:

- An N-body test finds possibly colliding pairs of objects
- A hierarchical oriented bounding box test finds possibly colliding pairs of triangles
- An exact test determines whether or not a pair of triangles actually overlaps

To determine whether two objects are close enough to potentially collide, we check whether the axis aligned bounding boxes of the respective objects overlap. If they do, further investigation is necessary. If not, the objects can't possibly collide and the algorithm can move on. To determine whether two bounding boxes overlap, the algorithm reduces the 3D problem to three simpler 1D problems. It does so by determining the intervals occupied by the bounding volume along each of the x,y and z axes.

If and only if the intervals of two bounding volumes overlap in all of the three dimensions, the objects corresponding to these bounding volumes must overlap. To determine which intervals of the objects along an axis overlap, the list of the intervals is sorted.

Normally, using quick-sort, this is a fairly quick process. However, by exploiting frame coherence (the similarity between situations in two subsequent frames) we can sort the lists using insertion sort. Another difficult part in this approach is the maintenance of the bounding volume. If the objects in the scene move or rotate, the previously calculated bounding boxes are invalid. It is important to be able to update the boxes as quickly as possible. Again, we can do this by exploiting frame coherence.

### Technologies Used:

The software is written entirely in C++ with extensive use of Visualization Toolkit (VTK) and MFC. VTK is a high-level OpenGL library which allows programmer's to build complex shapes without too much work. It is an open source, freely available software system for 3D computer graphics, image processing, and visualization. VTK supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as implicit modelling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. In addition, dozens of imaging algorithms have been directly integrated to allow the user to mix 2D imaging / 3D graphics algorithms and data. The design and implementation of the library has been strongly influenced by object-oriented principles.

The GUI front-end of the software is based on the Microsoft Whidbey model. We used the Xtreme Toolkit by Codejack to develop the front face of the application. This toolkit extends MFC to give the ability to create professional applications that have the latest visual styles and advanced features that are seen in such products as Microsoft Office and Microsoft Outlook.

## Furnace Evaluation Results:

The mechanical properties of tool and die steels are highly dependant on their thermal history, and great care is taken to ensure that parts are properly heat treated. Occasionally, however, treated components do not pass a hardness tests and must be retreated before the material can be put into service. It was suspected that the stacking of components in the tempering furnaces led to improper heat treatment. To determine if this was true, two surveys were carried out on furnaces at Finkl steel to determine the actual temperature distributions. Both surveys were carried out in the same furnace; the first survey was performed on an empty furnace to get a base line 3-D temperature profile. The second survey was carried out with the furnace loaded with components.

The approximate furnace loading zone dimensions are  $21 \times 6 \times 4.5$  [ft], and the overall dimensions are  $24 \times 9 \times 6$  [ft]. Two natural gas burners heat the furnace, are located in opposite corners, and are aimed into the firing lanes along the furnace edges. A schematic of the furnace can be seen in Fig \_\_\_\_\_. The furnace controller uses readings from 4 separate thermocouples located in the 4 walls of the furnace to control the burners.

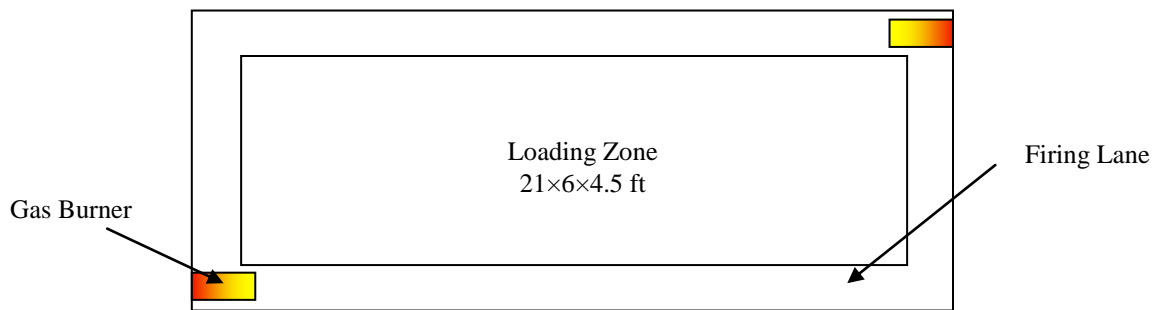


Figure 1: Furnace Schematic

In each survey, approximately 20 thermocouples were arranged throughout the loading zone volume. For the empty furnace survey, the thermocouples were placed on stands in a rectangular grid arrangement. In the loaded furnace survey, the majority of the thermocouples were placed directly onto the components being heat treated, while some were again placed on stands or on the furnace hearth. The temperatures measured by the thermocouples in contact with the components correspond to the component temperatures, which can differ significantly from the air temperature.

Temperatures were recorded by a digital data acquisition system once a minute for approximately 24 hours for both surveys. For the empty furnace survey, the set point was stepped from  $500^{\circ}\text{F}$  to  $1100^{\circ}\text{F}$  in  $200^{\circ}\text{F}$  increments. The furnace was allowed to soak for 4 hours once it reached its set point, and the temperatures for each thermocouple were averaged over this time period. A MATLAB script file was used to analyze the data and generate 3-D temperature distribution plots. The loaded furnace survey was carried

out in a similar manner, however the set point temperature was ramped from room temperature to 1150°F. While the entire heating cycle took over 24 hours, the soak period lasted for only 12 hrs; the temperatures were averaged over this period, and fed into the same plotting algorithm for analysis.

Figure 2 shows a sample temperature distribution for an empty furnace. The temperatures displayed represent averages over a 4 hour period. The plotting algorithm interpolates the temperature throughout the volume based on the 20 measured temperatures, and then shades planes to represent the local temperatures. Planes can be added at any location and at any orientation. Because the largest temperature variation occurred across both the length and height of the furnace, planes parallel to the XZ plane are shown here.

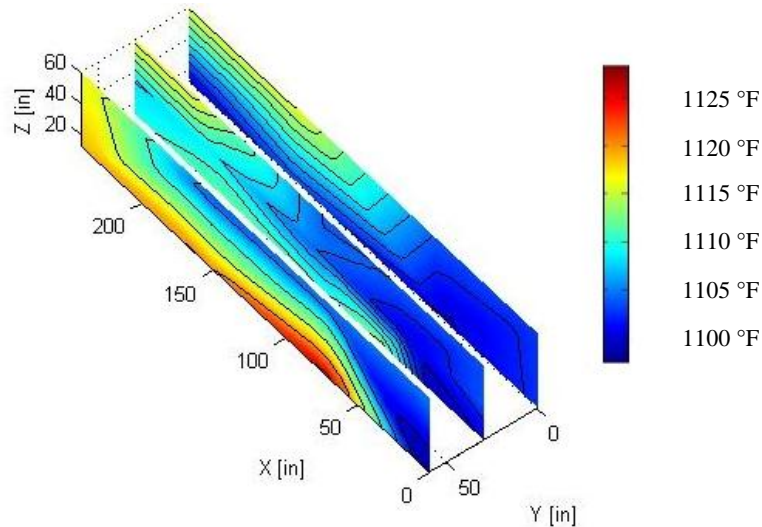


Figure 2: Empty Furnace Temperature Distribution at 1100°F Set point

Note that the temperature throughout the majority of the furnace is close to the set point temperature of 1100°F, especially near the front ( $X = 0$ ). There is a high temperature region, approximately 1125°F, along the lower edge of the furnace. This location is slightly outside of the loading zone, and instead corresponds with the location of the firing lane. The other side of the furnace does not show this behavior, as the thermocouples are fully within the loading zone. Within the loading zone, the temperatures are relatively even, and agree well with the set point, however temperatures outside of the loading zone can be significantly higher than desirable.

Figure 3 shows the average temperature distribution measured for the loaded furnace. Here, the approximate component locations are shown along with the temperature distribution. Note that the plotting algorithm can only interpolate temperatures within the 3-D arrangement of thermocouples. Because of the irregular thermocouple placement in this configuration, there is not a regular grid arrangement of thermocouples, and the subsequent interpolation does not occur over a prismatic volume. The resulting plot is still instructive however, as it shows that there are no locations in the furnace above the 1150°F set point. This indicates that the components would be over heated, and none should have lower than desirable hardness as long as they were properly quenched before the tempering stage. In the front region of the furnace, the temperatures are very even throughout the volume, and within 5°F of the set point. There is a cool



region in the back portion of the furnace, that corresponds to 3 thermocouples that were located on the floor of the furnace rather than on blocks themselves. This indicates that the furnace hearth is a strong heat sink, however the component temperatures were very close to the set point, both on the top and the bottom of the blocks. When components are placed in the furnace, they rest on setters, which hold them approximately 6 to 8 inches above the floor. This allows for air circulation below the components, and results in an even temperature component temperature distribution even though the floor is cooler than the set point. This combination of furnace surveys shows that if components are properly placed on setters completely within the loading zone, they should reach the proper temperature without being over heated.

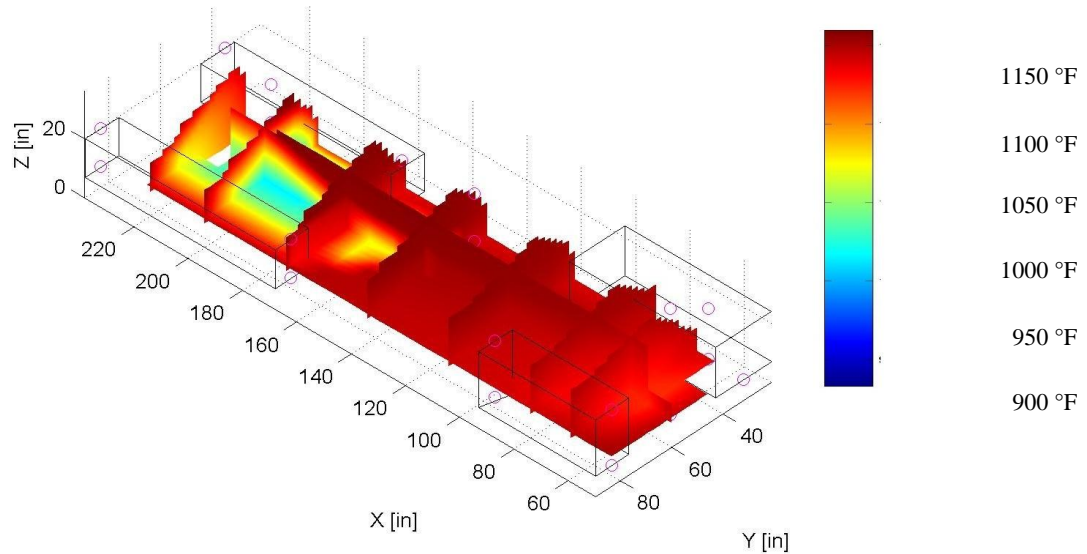


Figure 3: Loaded Furnace Temperature Distribution at 1150°F Set point

A tempering curve for Finkl Sheldie steel is shown in Fig. 4. This curve shows the change in hardness in the material after tempering at various temperatures. From the figure, there is very little change in hardness for specimens tempered at temperatures below approximately 900°F. This indicates that if parts are rejected because they are too soft, they were almost certainly overheated during tempering. Under-heating a component does not necessarily result in a desirable final product, however, depending on the desired tempering temperature, the component hardness could still be high enough so that it would pass the requisite quality control checks.

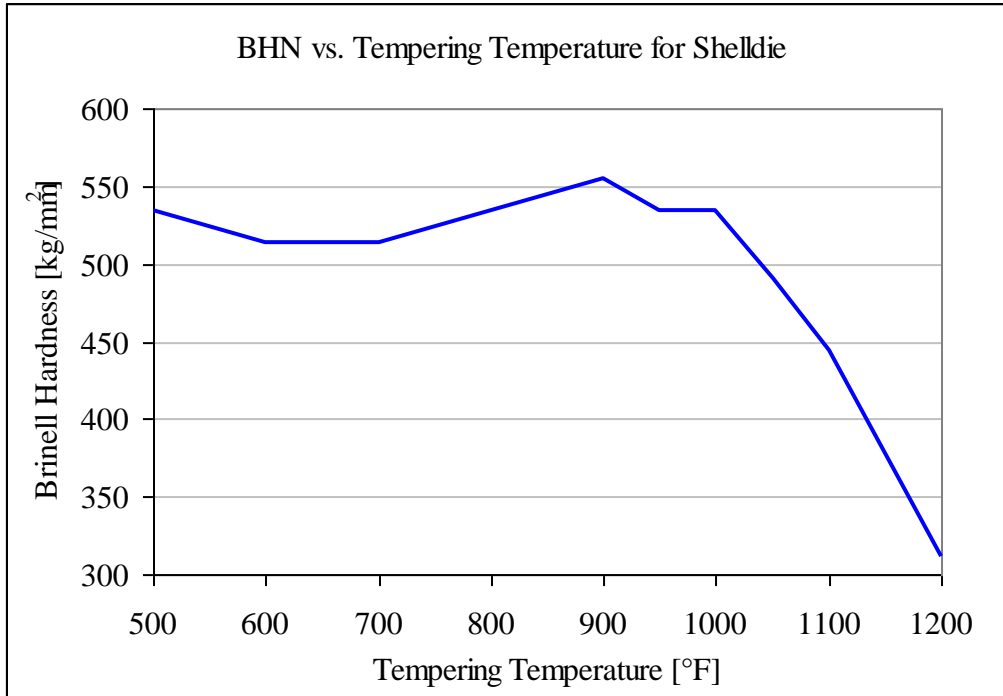


Figure 4: Tempering Curve for Sheldie Steel

This information in conjunction with the loaded and empty furnace temperature profiles indicates that overheating is the likely cause of failure. In all instances, the temperature within the loading zone was at or below the furnace temperature set point at all locations, and the only thermocouples with temperatures hotter than the set point were near the firing lanes outside of the loading zone as shown in Fig 1. If parts are improperly stacked such that they extend beyond the loading zone, they could be improperly heat treated via overheating.

The furnace surveyed had an acceptable temperature distribution within the loading zone both in the empty and loaded configurations. It was shown that the importance of properly loading components is paramount, and a system to keep track of the furnace loads is critical. If components fail, it is desirable to be able to recall the position of the component in the furnace, and use that information to change future loading configurations. This is the impetus for development of software to keep track of component location and orientation within each furnace load.

# Images and Appendix:

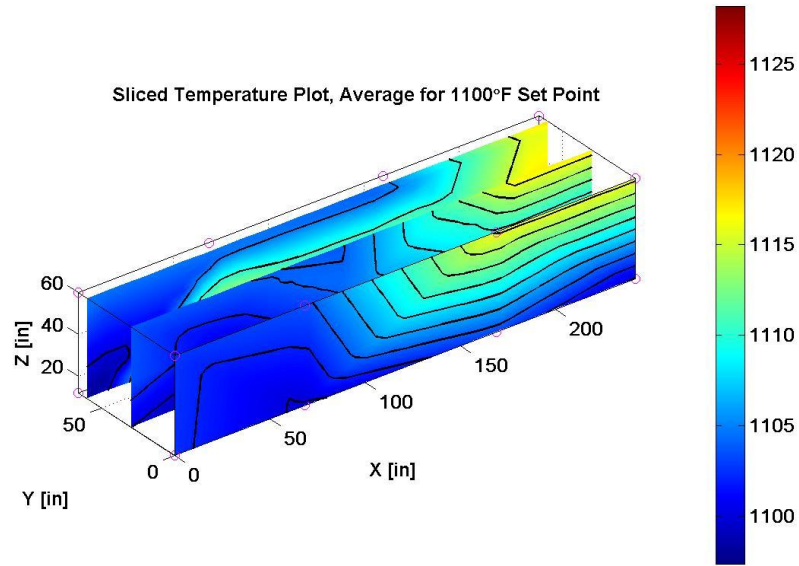


Figure 1: Temperature distribution for 1100 F nominal temperature.

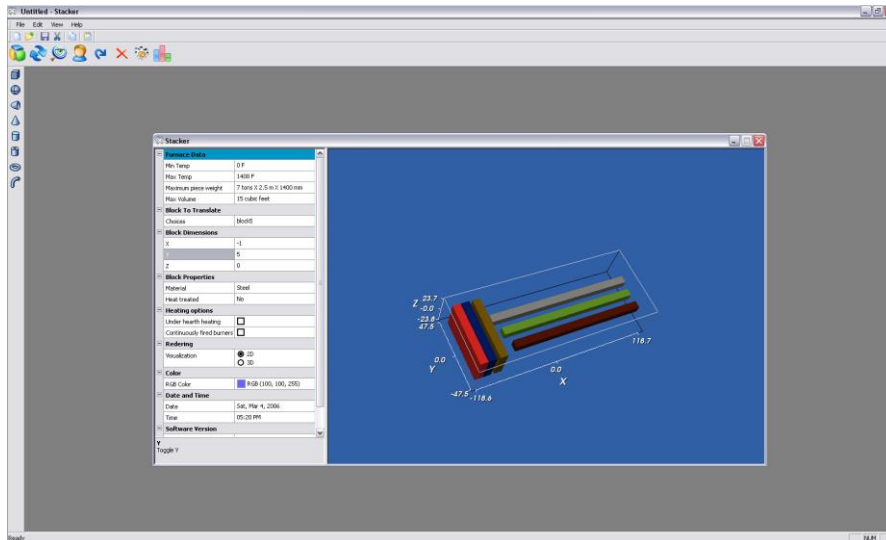
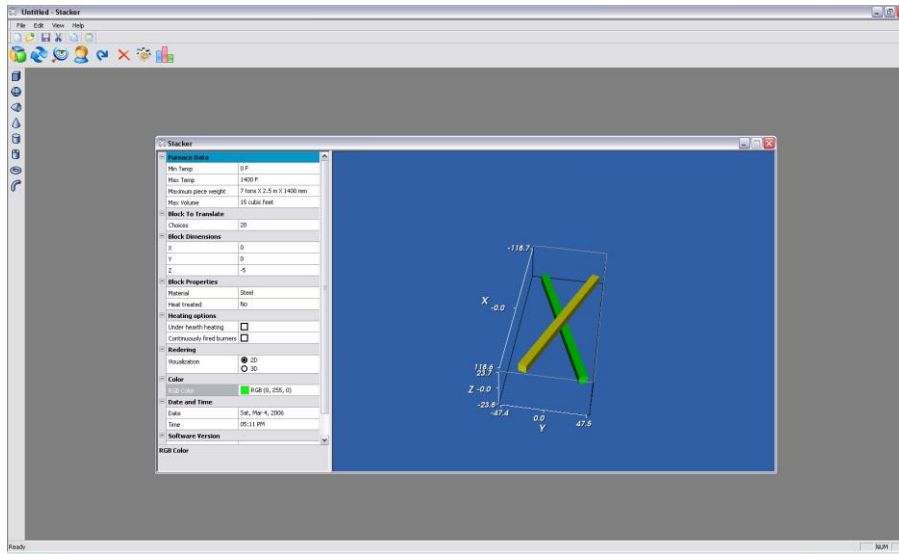


Figure 2: Typical workspace in the current build of Stacker



**Figure 3: Stacker is capable of laying objects at any angle.**