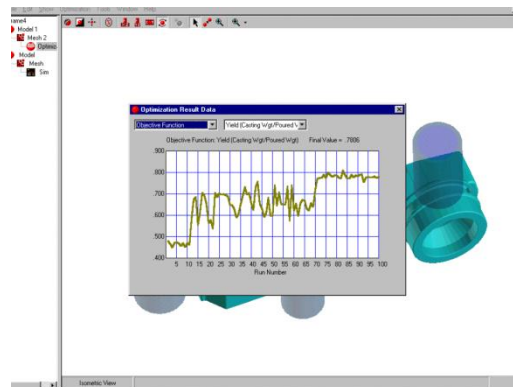
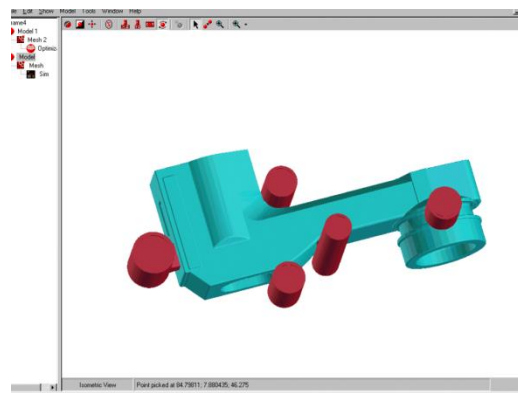
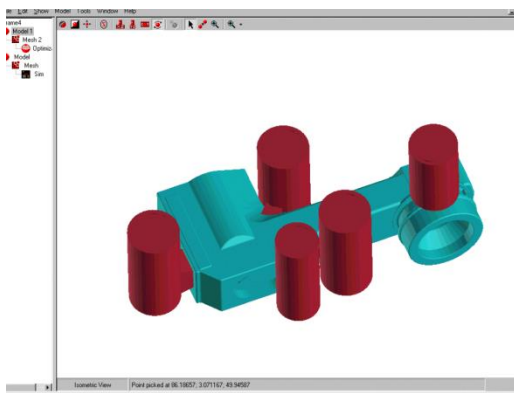


# OPTI CAST

PC-Based Casting/Process Optimization Software



Training Course Workbook  
Copyright © 2014



Version 8.3.0

COURSE SYLLABUS

---

- UNIT 1: Overview of the Optimization Process
- UNIT 2: System Installation and System Considerations
- UNIT 3: Setting Up an Optimization Run
- UNIT 4: Selecting Geometric Design Variables
- UNIT 5: Creating Linked Geometric Design Variables
- UNIT 6: Putting Limits on Model Data
- UNIT 7: Setting Process Constraints
- UNIT 8: Setting the Objective Function
- UNIT 9: Simulation Parameters
- UNIT 10: Advanced Optimization Parameters
- UNIT 11: Running the Optimization
- UNIT 12: Reviewing Optimization Results
- UNIT 13: Optimization Strategies

## UNIT 1: Overview of the Optimization Process

---

Solidification modeling, or computer simulation of the casting process, is being used by more and more foundries around the world to design the process for production of castings before castings are made or before equipment is built or altered. The primary objectives of using casting modeling are generally as follows:

1. To improve the quality of the casting produced, both in the first castings made and over the production life of the part.
2. To reduce the amount of time for new parts to enter production.
3. To reduce costs such as melting and handling by improving yield and reducing cleaning operations.
4. To quickly resolve design-related quality problems which may occur during production.

The traditional method of designing casting processes (and which, frankly, is still in use in the majority of foundries today) is to use either experience or a set of rules (or some combination thereof) to arrive at an initial design for gating and feeding a casting. Pattern or mold equipment is then produced according to this initial design, and test castings are made. If the test castings exhibit some unacceptable defect, then the pattern equipment is modified and new test castings are poured. This sequence may be repeated several times until suitable castings are made, or until time pressures from the customer force the foundry to begin shipping castings even if the design is sub-optimal. Often, in practice, this trial-and-error method of improving the process design continues even while castings are being shipped to the customer, which may have adverse effects on the relationship between the customer and the foundry.

Computer modeling offers the potential of evaluating alternate potential process designs in much less time, and at much less cost, than building equipment and producing sample castings.

The casting modeling systems which are available for foundry users today are, for the most part, software programs which accept a user's design for a casting production system and then analyze the design to make a prediction of the likelihood of defects. The user provides both the geometric data (the shape of the casting, feeders and gating system) and the material property data for the model to be analyzed. Once an analysis has been completed, the user views the results of the analysis, typically by viewing various graphic images and interpreting these images. If an area of potential defect is found within the casting (such as internal shrinkage porosity), then the user needs to make a determination as to whether the defect is acceptable or not within the area shown.

If the defect is deemed to be unacceptable, then the user must make some determination as to why the defect formed in the area shown. This may involve viewing a variety of output data in order to develop an understanding of the solidification dynamics involved. Once this is accomplished, the user must then make a decision as to what change in the process or in the design might improve the situation, and when this decision is made the model must be altered and a new analysis run in order to verify whether the change had the intended effect. If not, then the process must be repeated until the desired result is obtained.

The effect of this process is that the design must be modified through a trial-and-error sequence until the analysis shows that the desired result has been achieved. Thus, in effect, we have replaced trial-and-error on the foundry floor (the traditional method) with trial-and-error on the computer. The advantage of doing this is that the time and cost have been reduced. However, we are still dependent upon the foundry engineer to interpret the results of a given analysis and to make a decision as to what changes are required for the next design iteration. And, once an acceptable result has been achieved, we still do not know if the

result is an optimum. For example, is this the smallest riser size that would produce a sound casting, or could we have gone smaller?

In an effort to advance beyond the trial-and-error stage, we have developed the OPTICast™ software module to apply the methodology of optimization to the solidification modeling process. Optimization is a technology wherein a problem is structured such that input variables and constraints are identified, and an objective is specified, and we then search for an optimum solution to the problem using an optimization algorithm. There are numerous techniques for performing this search for the optimum condition, some of which are more efficient than others. The specific technique which we have settled on for performing optimization with casting solidification modeling is known as multi-variable response surface optimization, which basically considers the amount of change which results from modifying one or more design variables and determines whether this brings us closer to, or farther away from, the desired objective.

The effort involved applying Response Surface Methodology to the metal casting rigging process. This is the first time, to our knowledge, that this methodology has been successfully applied to the casting process.

## APPLYING OPTIMIZATION TECHNOLOGY

The question to be considered was how to apply this optimization technology to casting modeling, so that the design of a given casting with its rigging could be automatically modified to produce an optimum condition. Optimization requires the identification of three basic parameters:

### 1. DESIGN VARIABLES

These are features of a particular design that are allowed to vary while the system searches for an optimum condition. In the case of a computer model of a casting process, design variables may be geometric features such as the diameter and height of a riser or a riser/sleeve combination. They may also be process specifications such as the pouring temperature of the casting alloy. It is necessary to specify a minimum, a maximum and a nominal value so that the optimization system knows within what “envelope” it is allowed to operate.

The aspect of design that is most often varied when optimizing a casting process design is probably geometry. Therefore, it is necessary to develop a methodology by which geometry could be automatically varied. This was complicated by the fact that in model building, shape geometry can be represented through various types of parametric shapes as well as non-parametric geometry imported from CAD systems. It was decided that a universal “scaling” parameter could be applied to any shape, independently in the horizontal and vertical directions, in order to vary any type of shape. This would mean that a

system user could select some geometric feature and specify the limits of scaling of this feature horizontally and vertically, and thus create a design envelope for the feature within which the optimization module could vary the size and shape of the feature. There are other, more sophisticated ways of modifying geometry that could be implemented later, but as a first iteration it was felt that the scaling technique would suffice for most foundry applications.

A second type of design variable required is process data, such as initial temperatures or pouring times. The optimization routine was therefore set up to allow the user to select specific process data and specify the allowable range for that item.

### 2. CONSTRAINTS

Constraints are values of some aspect of process data above or below which a result is not allowed. Constraints may be specified as a minimum condition, in which case the result value must be at or above the given constraint value, or as a maximum condition where the result value must be at or below the given constraint value. One or more constraints may be specified for each optimization run. An example constraint would be a maximum allowable porosity level.

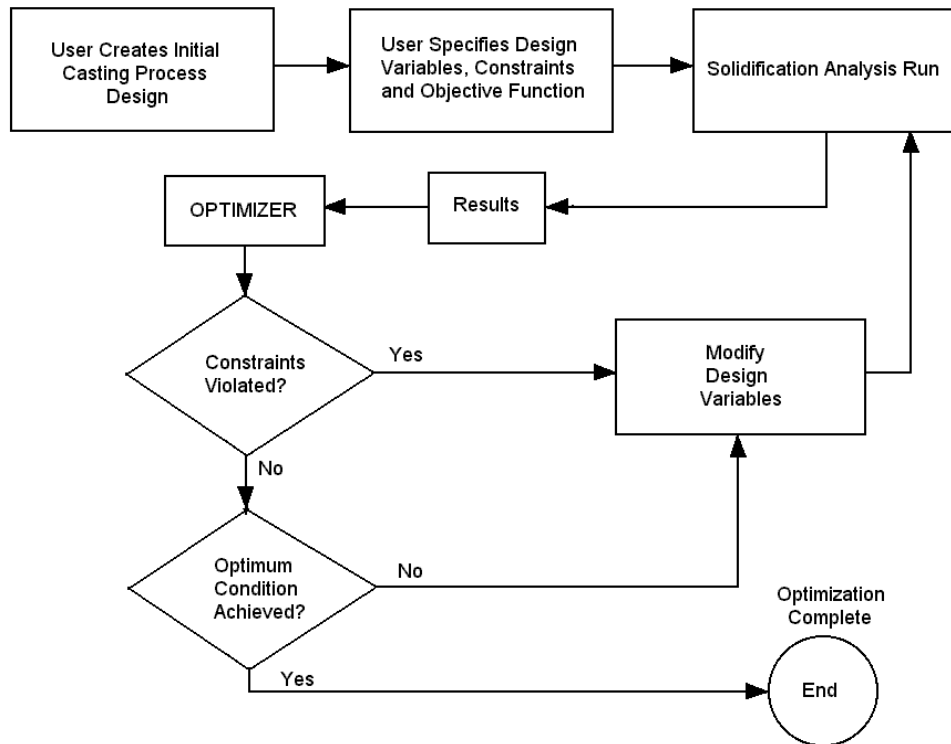
### 3. OBJECTIVE FUNCTION

The objective function specifies what, ultimately, is trying to be achieved with a given process design. Selection of an objective function tells the optimization system what process result is to be used to judge whether you have achieved an optimum condition. The system user selects an objective function, and specifies whether the value of that function is to be minimized or maximized. For example, you might select predicted shrinkage porosity as an objective function, in which case you would want to minimize its value. On the other hand, you might select material yield (the ratio of casting weight to poured weight) as an objective function and try to maximize its value.

Only one objective function can be specified for each optimization run.

## OPTIMIZATION SEQUENCE

The overall sequence of events which occurs when optimizing a design is shown in the following process flow diagram:



As shown in the diagram, the sequence of operation of an optimization run is, first of all, for the user to create an initial process design, i.e., a three-dimensional model of the casting with gating and feeding, and all relevant material data. This is the same data required for any casting simulation. The user then selects the design variables, constraints and objective function and launches an optimization run. Optimization basically consists of running a series of simulations automatically, varying the values of the design variables, checking to make sure that constraints are not violated, and searching for a maximum or minimum value for the selected objective function.

In order to allow this process to occur within a reasonable amount of time, the time for each successive simulation must be as short as possible. This means that the simulation algorithm must perform the minimum number of computations to arrive at an accurate result.

There are two basic methods of computing successive heat loss and solidification of castings. The first is the Finite Difference Method (FDM), in which small brick-shaped elements approximate the shape of the casting, and progressive temperatures and material conditions are solved by wave after wave of calculation increments through the mesh which solve for temperatures in each element by considering only the neighboring elements.

The second method is the Finite Element Method (FEM), in which elements of different sizes and shapes more closely approximate the casting geometry. In FEM, a simultaneous solution of temperatures at all nodes is required at each time step; in effect, FEM requires solution of a very large system of simultaneous equations. Although the time steps in FEM can be larger than in FDM, the FEM method is very calculation-intensive and, in general, the FDM method is faster to produce results. Thus, it is our feeling that FDM is a preferred calculation route for the optimization process.

Another difference between FDM and FEM involves the creation of the mesh, which is the set of elements that represents the casting geometry. An FEM mesh, as stated, involves elements of varying size and shape. Creation of an FEM mesh is a relatively complex process and often requires user interaction. To modify geometry of a casting model in an automatic way and avoid user interaction, as is required by optimization, would generally mean modifying the mesh by compressing or stretching the elements. This could potentially result in invalid elements that would void the simulation calculations. This intractable problem can be avoided by focusing on the FDM method. In this method, the basic geometry of the casting model can be adjusted programmatically and a new mesh can be created for each simulation, because the meshing process is entirely automated and does not require any user intervention.

Other features which can considerably speed up the calculation process, and which have been used in simulations for optimization, include the use of uniform-shaped (cubical) elements and the use of average material properties. Using a variety of such techniques, we have been able to produce simulations of casting processes which used a million or more elements (thus representing the casting geometry quite accurately) and which have run in less than 30 minutes on up-to-date personal computers. Using these calculations, an optimization requiring, say, 16 successive simulation calculations could run in one day. The same series of simulations in some other types of software systems could conceivably take weeks to perform.

## REFINEMENT OF PARAMETER SELECTIONS

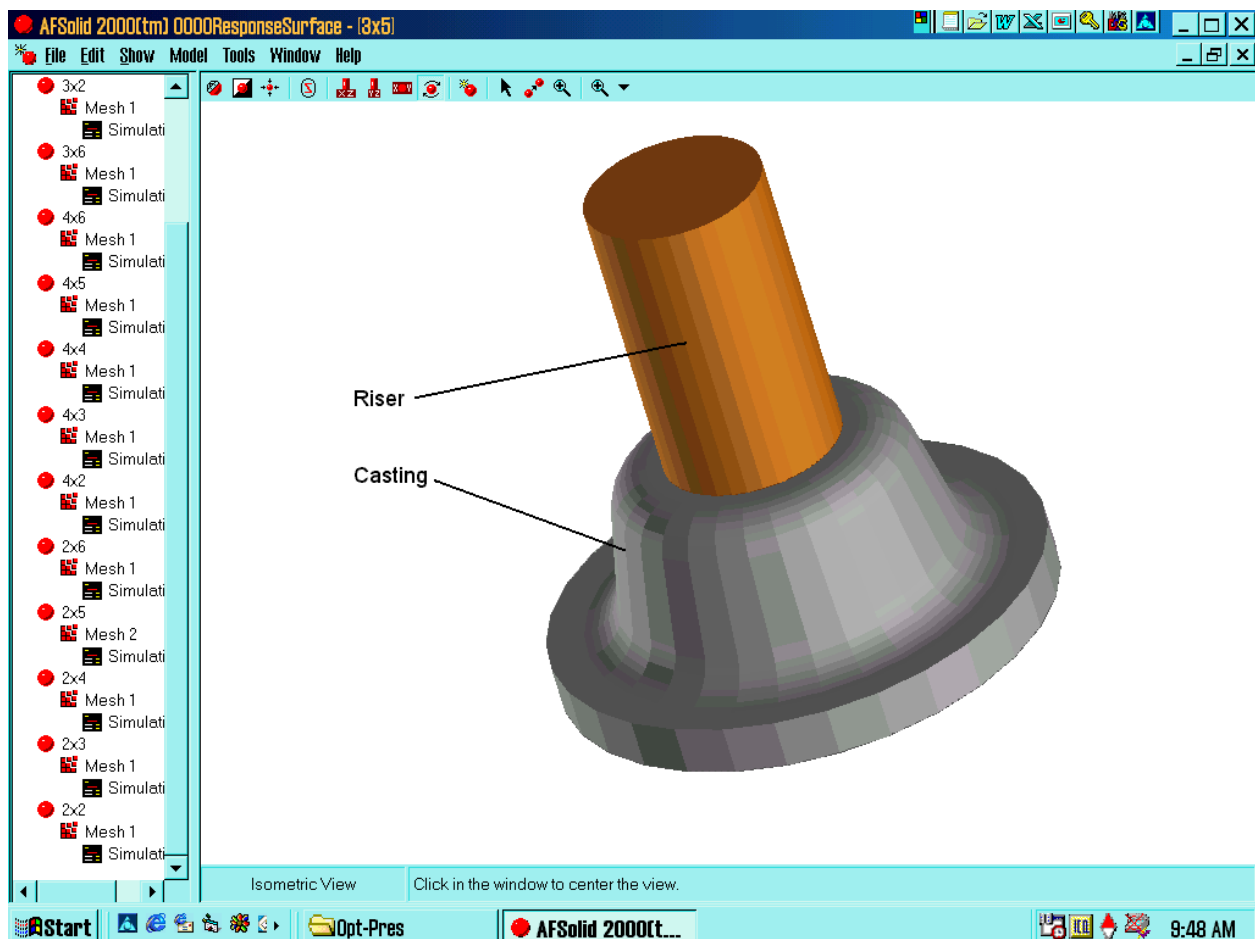
In order to develop a procedure for optimization, it is necessary first of all to understand what goals you are trying to achieve. Today, many casting simulations are run for the purpose of predicting shrinkage in castings (either in the form of macroporosity or microporosity), with the goal being to produce a sound casting. Therefore, our initial efforts in this area have been to optimize castings with respect to shrinkage formation.

Macroporosity and microporosity form under specific, and sometimes different, circumstances. In our work with simulation, we have found that macroporosity can be predicted by simulating the volumetric changes occurring in solidifying metals, and then simulating the flow of liquid feed metal in response to these volumetric changes. Microporosity, on the other hand, often seems to form in areas of poor directional solidification, which can be measured by consideration of criteria such as temperature gradients, local solidification times and the velocity of the solidification wavefront through various parts of the casting. In order to optimize castings for soundness, it is necessary to select the proper data for constraints and the objective function to achieve a sound casting in the most efficient way possible.

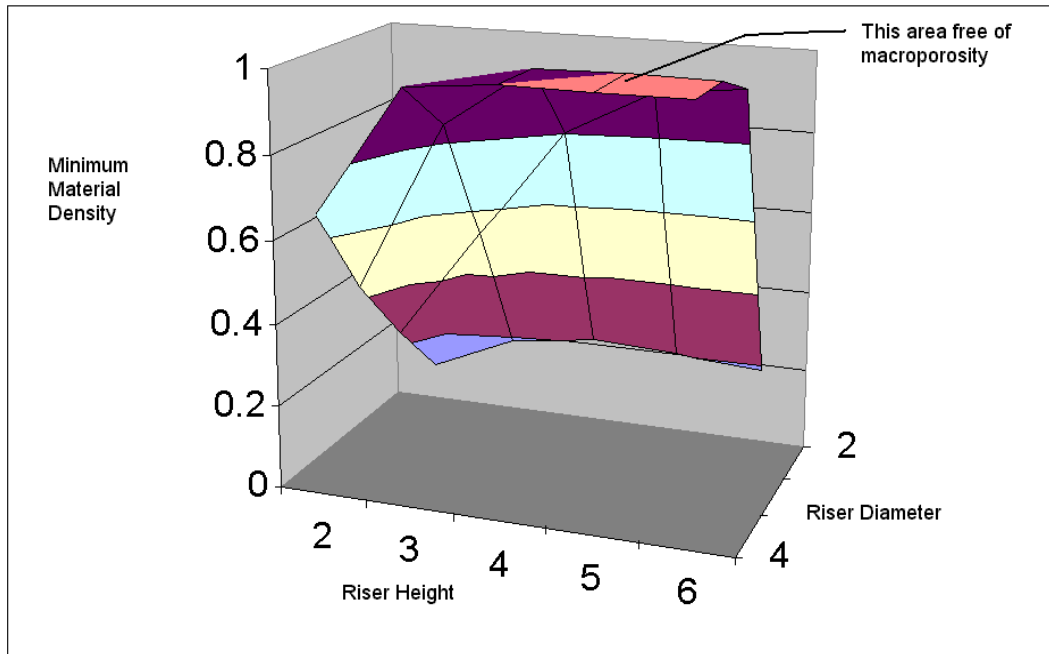


Foundry engineers are also concerned with the cost of producing a casting. One of the major costs is the energy involved in melting the metal to make a particular casting; the less metal required to be poured, the lower the melt cost. Therefore, when dealing with the concept of optimizing a casting design, you should take into account the total amount of metal which is required vs. the net metal in the casting, i.e., the process yield. It may be that several alternate riser designs could result in a sound casting, and the optimization system should be capable of identifying which of these designs is preferable from a cost standpoint.

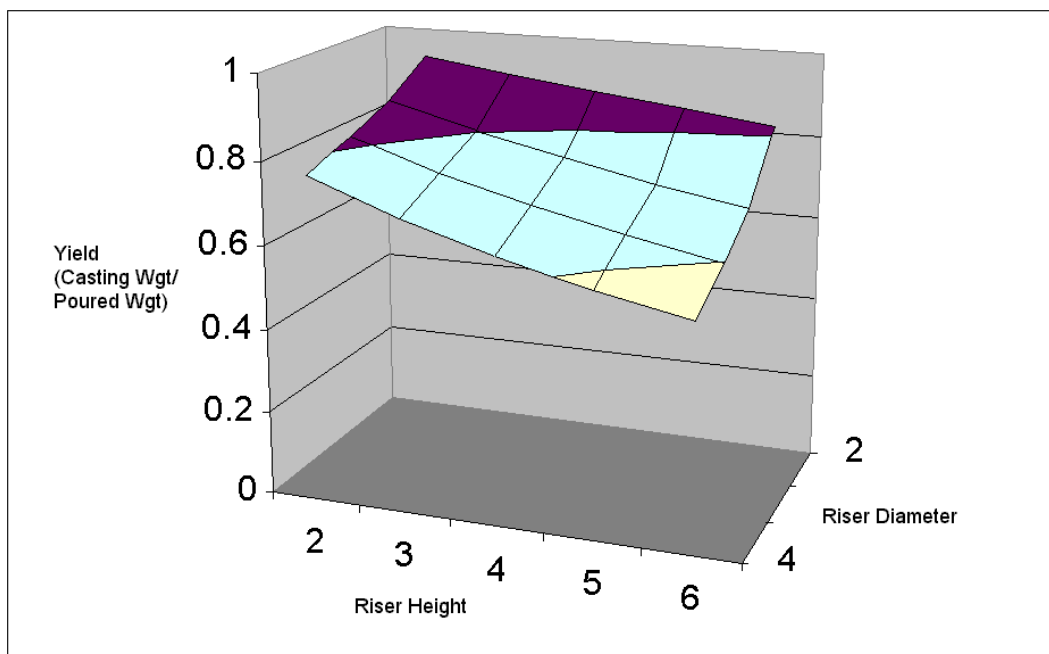
In order to provide a direction for refinement of the optimization process, a simple casting with a single riser was considered, as shown in the figure below:



Multiple simulation runs were performed with this casting, varying riser height and diameter, and several response surfaces were mapped out (a response surface is merely a plot of a particular process result over several combinations of process inputs). One of the features of the response surfaces that became obvious from this exercise was that, when considering measures of the soundness of the casting (either macroporosity or microporosity), there are regions where the casting is sound over several combinations of riser height and diameter, as can be seen in the response surface pictured on the next page:



This means that, if we were to use a soundness criterion as our objective function, once the optimization system entered this region it would be unable to differentiate between alternate riser designs; in other words, there are several combinations of riser height and diameter that would result in a sound casting, and the optimization system has no reason to pick one over the other. On the other hand, a response surface which is generated showing process yield mapped over the design variables seems to have a continuous slope as the process inputs are varied, as can be seen in the following image:



This means that it is easier to locate a maximum or minimum point on such a surface.

The preceding results suggest that a rational method for optimizing a given casting for soundness would be to specify one or more porosity predictors as constraints, and then maximize the process yield. This would allow the system to identify those designs which result in a sound casting, and then to select the least-cost design as being that which maximizes the material yield of the process. This scheme would resolve a further problem, in that only one objective function may be specified for an optimization run; it is not possible, for example, to specify criteria for both macro- and microporosity as being objective functions within one optimization run. Considering both of these criteria as constraints allows both of them to be considered when searching for an overall optimization. Of course, it is possible that for a given initial design a sound casting may not be possible, in which case such constraints could not be satisfied; the optimization system would report this, and some consideration would have to be given to revising the process design so that a sound, optimized result could be obtained. It may be possible, in the future, to structure a sort of "hyper-optimization" such that the system is able to search for constraints that would be able to produce a valid, sound result.



## UNIT 2: System Installation and System Considerations

---

### Hardware

Minimum Requirements (SOLIDCast without OPTICast or FLOWCast, or for small mesh node counts)

Intel Core 2 Duo E6300 or better, or AMD Athlon II or better

2 GB RAM

500GB Hard Drive

Midrange (\$200-\$300) DirectX 9 or 10 compatible video card

Windows XP Professional or Windows 7 Professional, Enterprise, or Ultimate

Recommended Requirements (SOLIDCast with FLOWCast and/or OPTICast, or very large node counts):

Intel Core i7 900 series or better, or AMD Phenom II

8 GB RAM or larger

1 TB Hard Drive

Midrange (\$200-\$300) DirectX 9 or 10 compatible video card

Windows 7 Professional, Enterprise, or Ultimate, 64-bit edition

A note about video cards: SOLIDCast makes use of standard Microsoft video driver technology to display its images. Any current midrange video card available which advertises compatibility with Direct X 10 will work for the current version and upcoming versions of SOLIDCast. An existing video card which is compatible with DirectX 9 will work with the current version of SOLIDCast.

### System Installation

New systems will come on a CD, DVD or USB Drive with several folders. These folders are typically labeled Software, Software Demos and Training Materials.

Note: Version 8 of SOLIDCast REQUIRES Windows XP, Windows Vista, or Windows 7. If you are running Windows XP, an update to Service Pack 3 (through Windows Update) is required for support of all of SOLIDCast's features.

To INSTALL the package:

1 - Insert the CD, DVD or USB Drive.

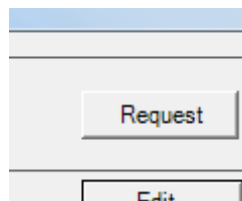
2 - Go to the Software Folder and double-click the file name SOLIDCastInstall8-2-5.exe to run the installation program.

3 - Follow on-screen instructions. FLOWCast and OPTICast are included in the package installation, and will activate with SOLIDCast if your organization has licensed them.

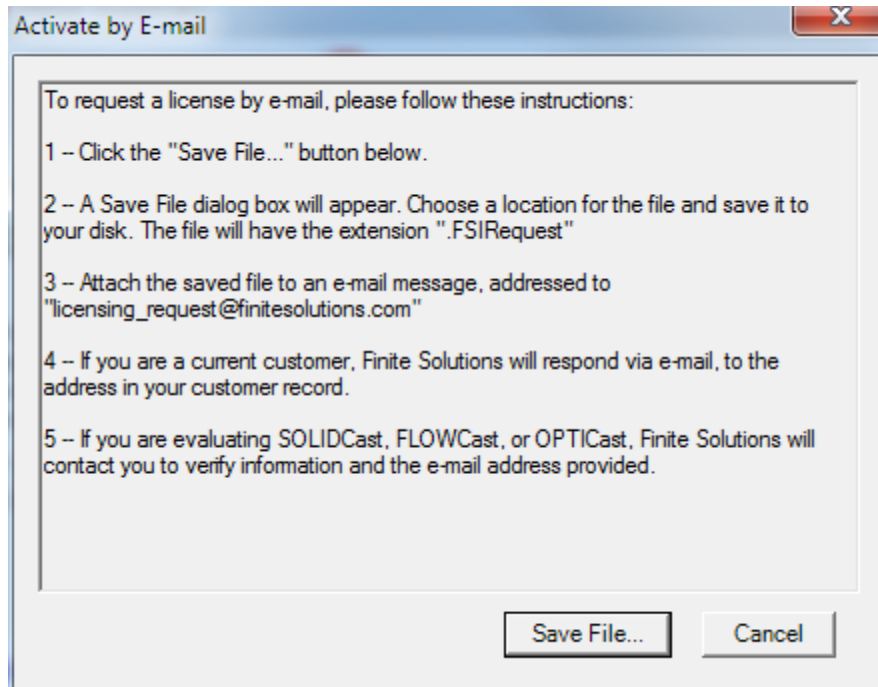
4 - To activate the system for normal operations, a license file is required, which will be supplied by Finite Solutions Inc. Start SOLIDCast, then click on Help...About SOLIDCast. This will bring up a window with the license agreement, and a "Licensing..." button. Click the "Licensing..." button.

You will be prompted to enter registration information. If you are evaluating SOLIDCast, click the "Evaluation Request" button and enter all required fields. If you are a current customer, click the "Current Customer" button and fill out Name, Company, and E-mail.

5a - If your computer is attached to the Internet, click on the button marked "Request". Clicking "Request" will send a license request to Finite Solutions; nothing more is needed from your end.



5b - If your computer is not on the Internet you will see a "Save File" dialog upon clicking "Request" E-mail the file according to the instructions on the prompt.



If you do not see the Request button, back out one level, then hold down the CTRL key as you click the Licensing button. This will 'force' the Request button to appear.

6 - Close SOLIDCast.

7 - You will receive a license file from Finite Solutions, emailed to the contact name in your customer record, or to the email provided to us in the request. Copy that file to your hard drive and double-click it. This will load the file onto each your machine. One license file may activate one or more machines, but each machine must send in a request in order to be active in the file. Again, be sure that SOLIDCast is NOT running when you activate the license by double-clicking the license file.

8 - SOLIDCast is now activated for your use. Note that a single activation request will cover all modules, including SOLIDCast, FLOWCast and OPTICast.

The media also contains other support files, including the training workbooks, version log file and others. If you have any questions about these files, read the more detailed information in the workbook files for each program, or contact Finite Solutions Inc.

David Schmidt  
Finite Solutions Inc  
Tel 262 644 0785  
Fax 262 364 2537  
Email [Dave@finitesolutions.com](mailto:Dave@finitesolutions.com)

## Updates

SOLIDCast is under continuous development, as we add new features and functions to the system. These updates can be downloaded from our web site. To obtain an update, navigate to the following site using your Web browser:

<http://www.finitesolutions.com>

You will see the FSI download page here. There are several useful items here, including the full install set, the latest update, and a copy of this training manual, in compressed format. All training manuals are saved as pdf files, and have been compressed using WinZip to reduce file size. Just click on a link to download the file to your computer.

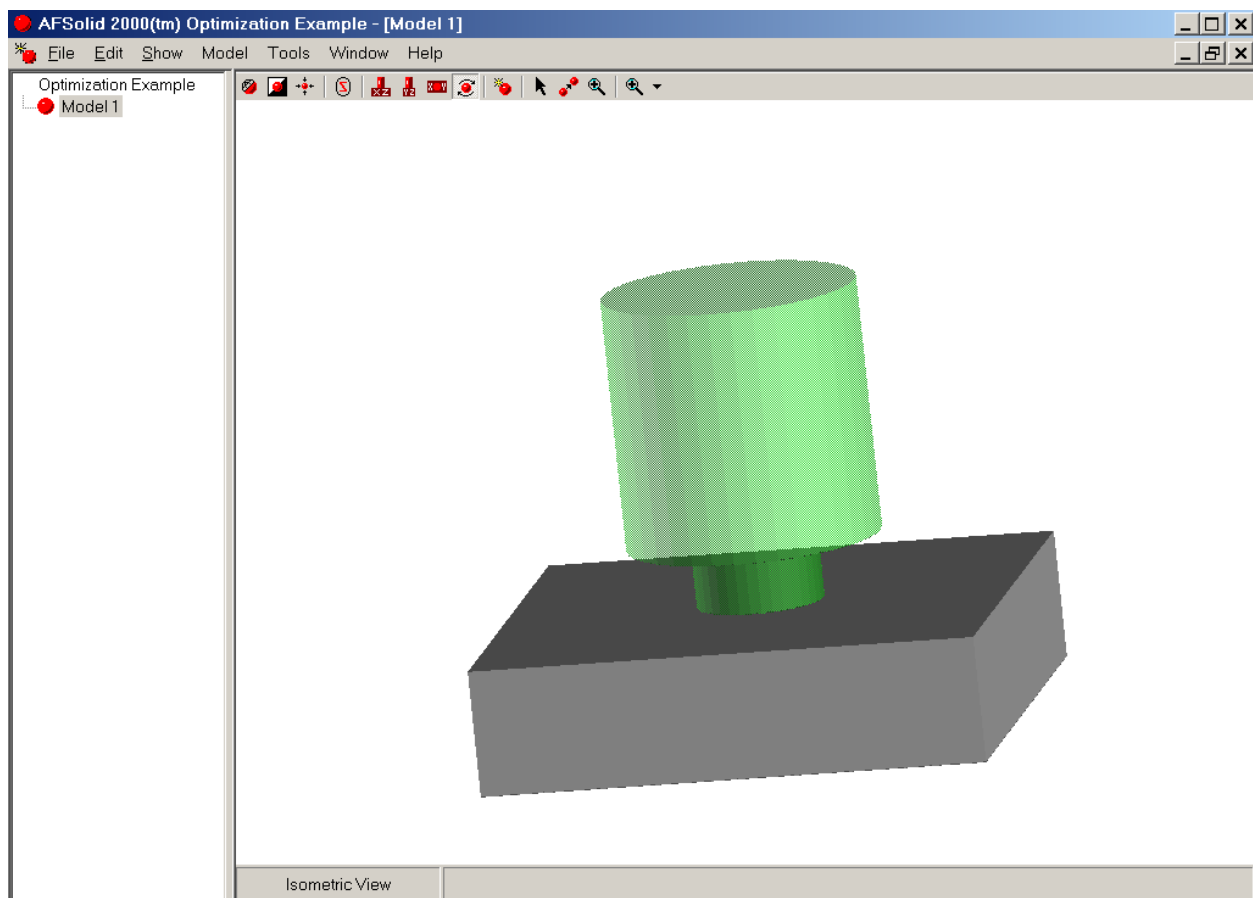
Please be sure to fill out the license agreement completely and return it to Finite Solutions Inc via fax or email. Having a complete record on file is the way to insure that we keep you posted about what is new with the software, and let you know when an update is available for download.



## UNIT 3: Setting Up an Optimization Run

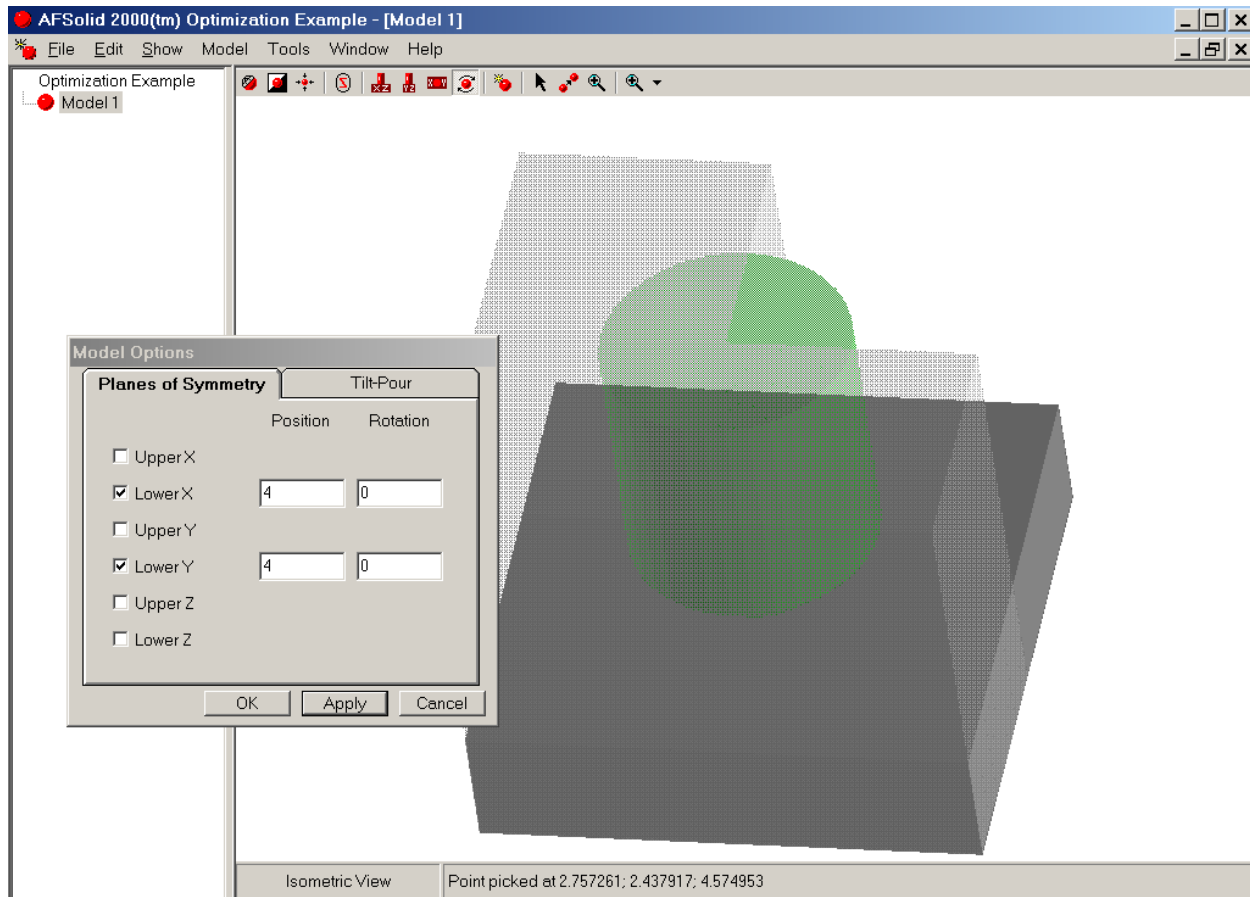
Before an optimization run can be set up, you need to load a model. This is done by loading the project, then double-clicking the model name. The solid view of the model will then be displayed in the model building window.

For example, the figure below shows a simple plate casting with a top riser on it. Note that the riser is made out of Riser Material, not Casting Material. This is important, because we will be trying to optimize the yield of this process, so the casting and the risering MUST be made of separate materials. If you will be importing STL files from CAD, be sure to save the gating as a separate file from the casting.



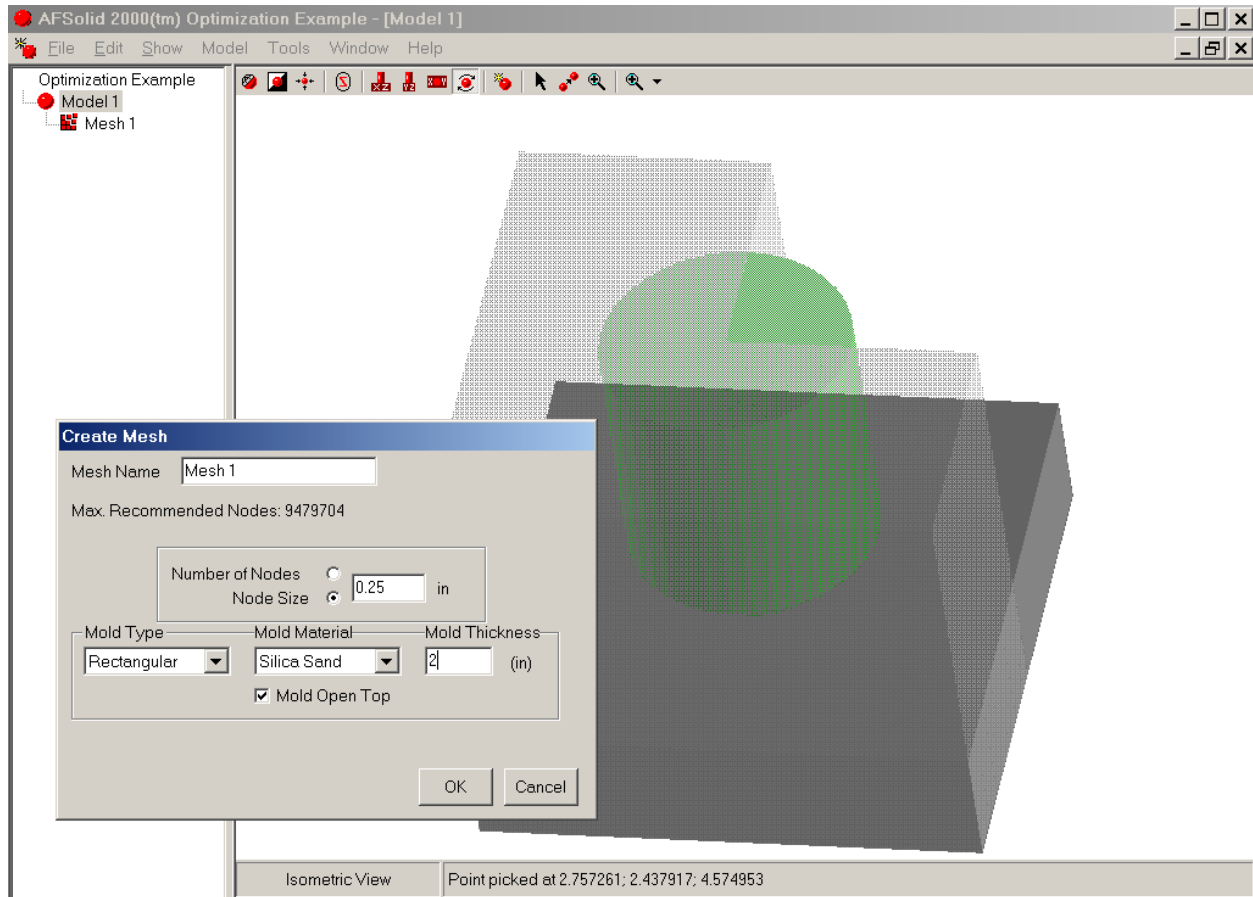
Since this part is symmetrical, it will be much more efficient to use planes of symmetry. By only running a simulation on a quarter section, we will be able to get results four times as quickly as if we had run the full model. That can become extremely important as the number of variables, and thus simulation runs, increases.

The following figure shows the casting model with a lower X and a lower Y plane of symmetry applied.

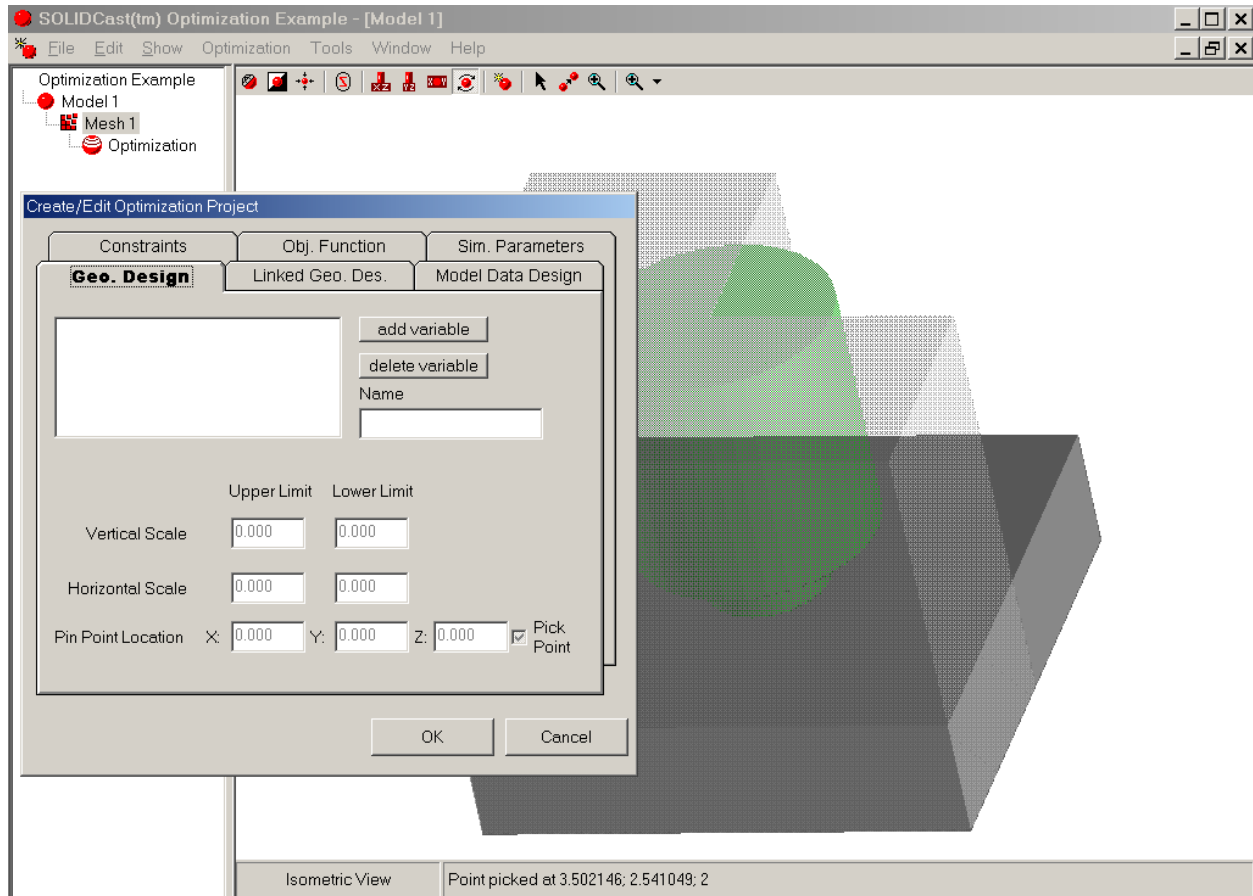


The second step is to create a mesh of the model. This is done by selecting Model...Create Mesh. An important consideration here is to ALWAYS USE THE NODE SIZE OPTION. This is because the system will be automatically modifying your model as the optimization run progresses, and if the mesh is given a constant number of nodes, rather than a constant node size, the mesh 'box' will be constantly changing. This introduces a great deal of 'noise' into the process, and you may not be able to get a solution because of this.

The following figure shows the plate casting meshed with 0.25 inch nodes, and a rectangular sand mold 2 inches thick. The riser is left open to the atmosphere. Because of planes of symmetry, the mesh uses only about 21,000 nodes, rather than 84,000.



Once the mesh has been created, we can start the process of setting up the optimization run. Highlight the mesh name on the Project Tree, then select Mesh...Create New Optimization Project. This will add an optimization icon beneath the mesh name on the Project Tree and will open the Create/Edit Optimization Project window, allowing you to set up the optimization run. This window is shown on the next page.

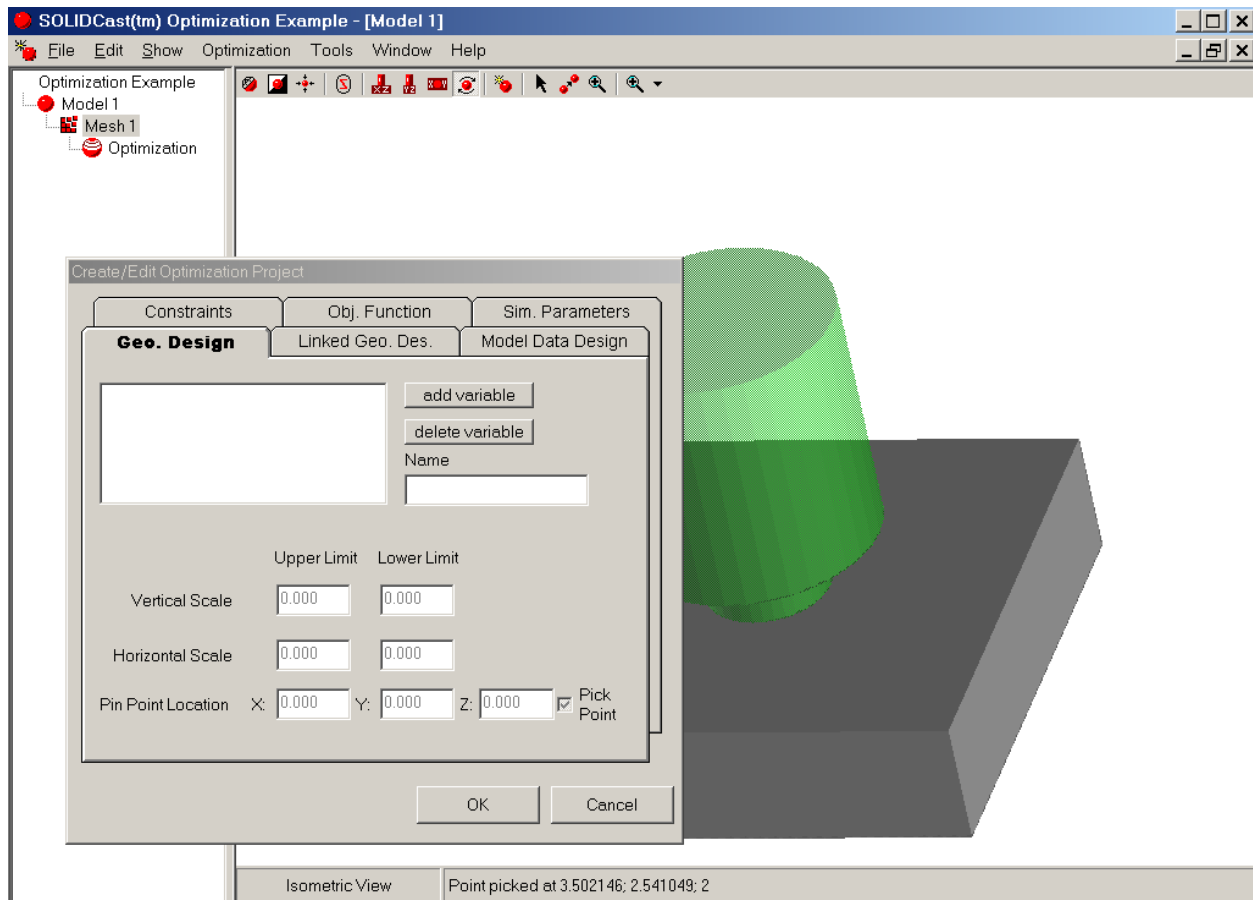


The six tabs shown in the Create/Edit Optimization Project window will be described in the next six units.

## UNIT 4: Selecting Geometric Design Variables

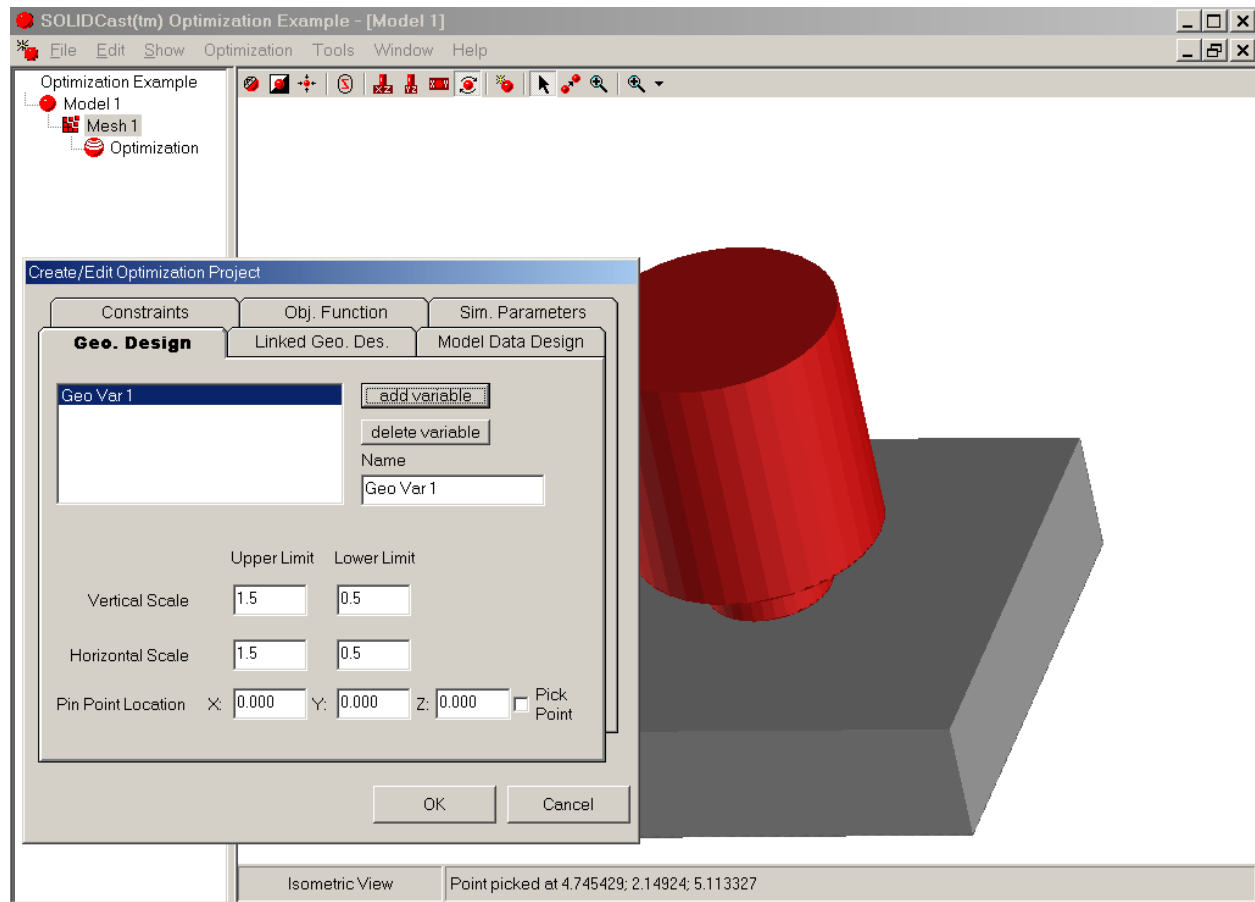
Geometric design variables are those parts of the model that you will allow to change in size during the optimization process. The most obvious choice for this would be riser/feeder size, but could also include things such as gate size, or even a part of the casting geometry itself, if you can adjust that feature separately.

When you select the Geo. Design Tab from the Create/Edit Optimization Project window, you will see something similar to this:



The general procedure is to first use the model building tools to select a part of your model that you want to be a geometric design variable, then click on the add variable button on the Geo. Design tab. You can choose to have the geometric design vary in the horizontal and/or vertical direction. You need to tell the system where the design is locked, or pinned, by clicking with the mouse. Let's see how this would work with our plate casting.

As an example, let's create a geometric design variable that will consist of both the riser and the riser contact. First, on the Model Building Icon Menu, click on the arrow to choose Select Shape Mode. Then, holding down the CTRL key, click on both the riser and the riser contact to select them at the same time. The riser and contact should change color, as shown in the figure on the next page, to indicate that they have both been selected.



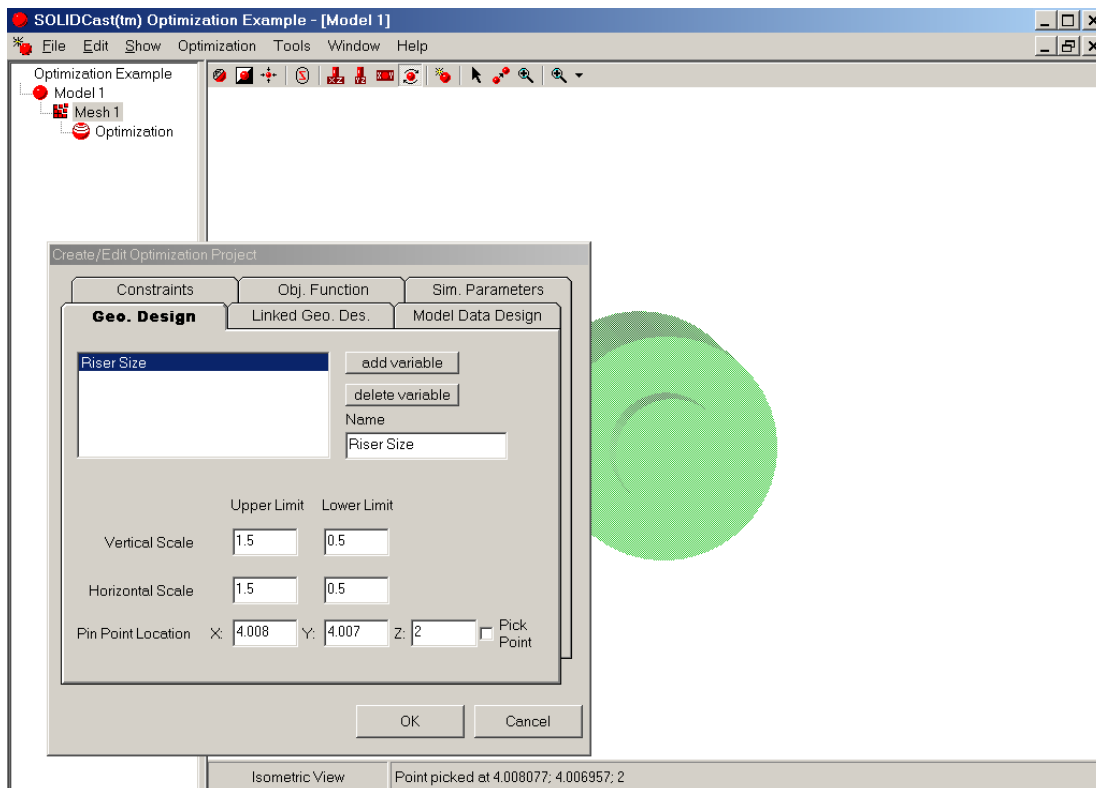
Once the features are selected in the Model Builder, click on the add variable button. The system will add a new Geometric Variable to the list with a default name that you can rename if you like. The system will also set up defaults, including high and low limits for the change in vertical and horizontal scales, as well as the Pin Point Location, about which the scaling takes place.

The default values for upper and lower limits for the vertical and horizontal scales are 1.5 and 0.5 times the current size of that dimension. So, for example, if the riser was originally 4 by 4 inches in height and diameter, the system would be able to modify the riser size between 2 by 2 inches and 6 by 6 inches.

To change the name of the variable, simply type over the default name listed in the Name box. This will automatically change the name listed in the Variable List.

The Pin Point Location can be specified in two ways. First, if you know where you want the Pin Point, you can simply type in the X, Y and Z coordinates for the point. If you do not know the exact location, you can use the Pick Point function to find it.

To use the Pick Point function, you need to first expose the location you want to choose in the Model Builder. For example, we'd like the Pin Point to be at the center of the bottom of the riser contact. To see this location we need to rotate the model up, select the plate section, and select Show...Hide, to temporarily hide the casting, so we can see the bottom of the contact. Then, check the Pick Point box on the Geo. Design tab and click the mouse on the bottom center of the contact. The coordinates of that location will then be entered onto the tab. The figure on the next page shows typical results.



This example shows that the default name was changed to Riser Size, and that the Upper and Lower Limits for the riser height and diameter were left at their default values. The Pin Point Location was selected using the mouse. Notice that the location is very close to 4, 4, 2, which would be the exact location. Such small variations will not make any difference in the simulations and can be safely ignored.

It should be noted that Select Shape Mode is a 'sticky' function; i.e., that mode stays in effect until it is de-selected using the mouse. When removing the casting from view to pick the base of the riser contact, you may need to move into and/or out of Select Shape Mode more than once. If the Create/Edit Optimization Project window gets in the way, you can move it by dragging on the title bar until the part of the model you need is visible.

If more variables are desired, they are added in the same way, by selecting the geometric feature, then choosing add variable from the Geo. Design Tab. To reduce optimization time, keep the number of variables to a minimum. And, if possible, use linked geometries, as described in the next section.

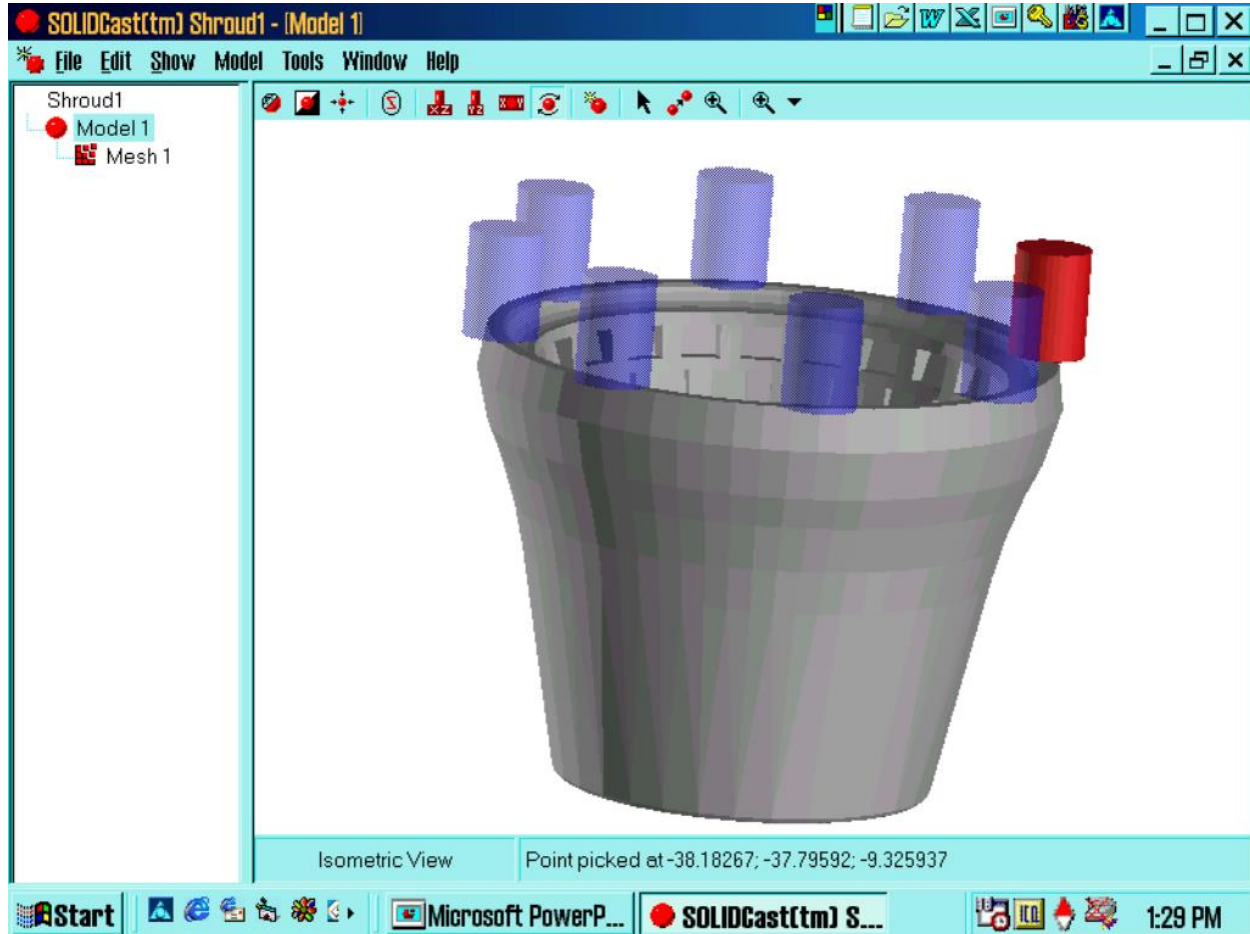




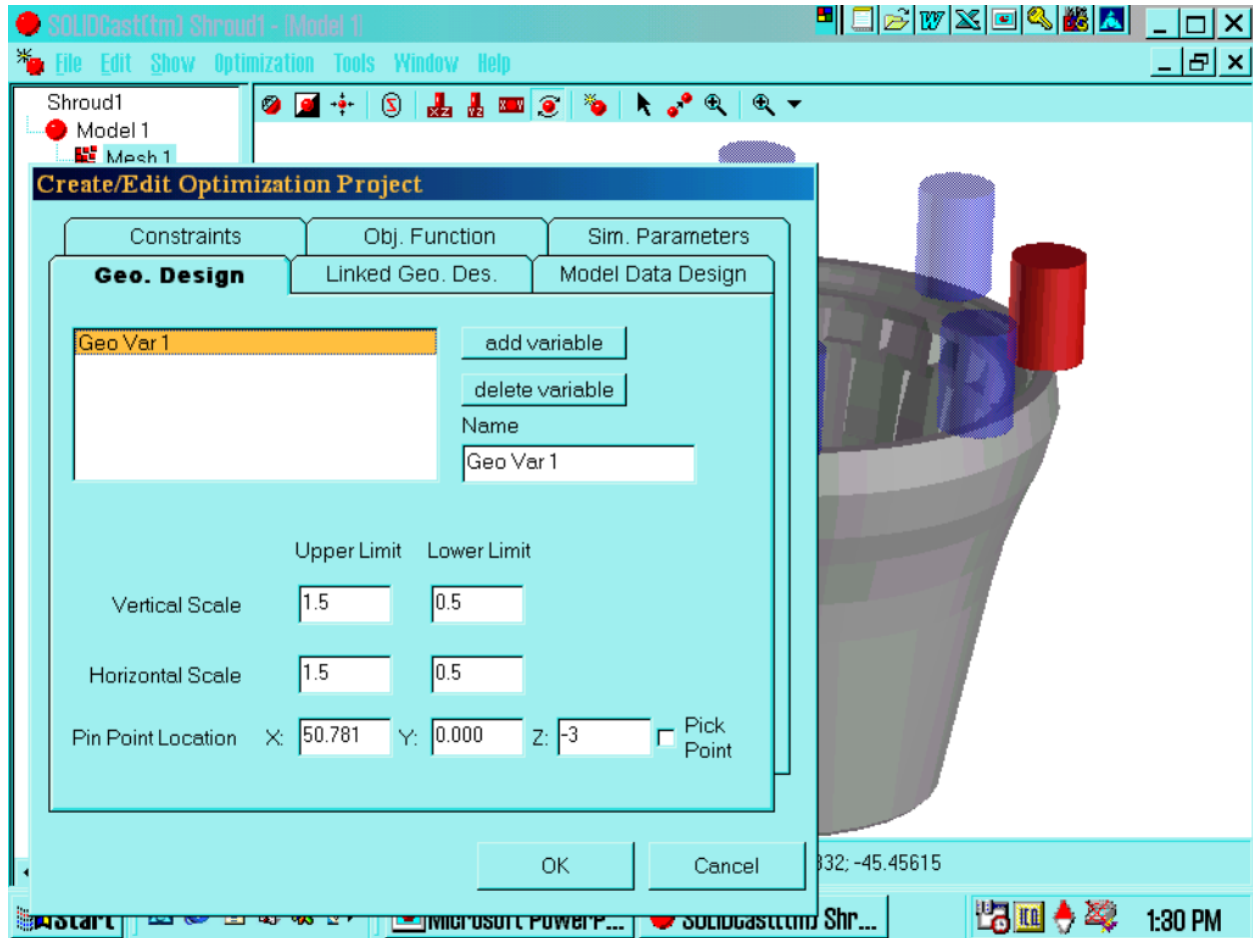
## UNIT 5: Creating Linked Geometric Design Variables

If you have a model where a pattern of risers must be symmetrical, you can use the Linked Geometry feature to link additional risers to a defined Design Variable. This will allow multiple risers to vary in shape, while only the original riser will be defined as a Design Variable.

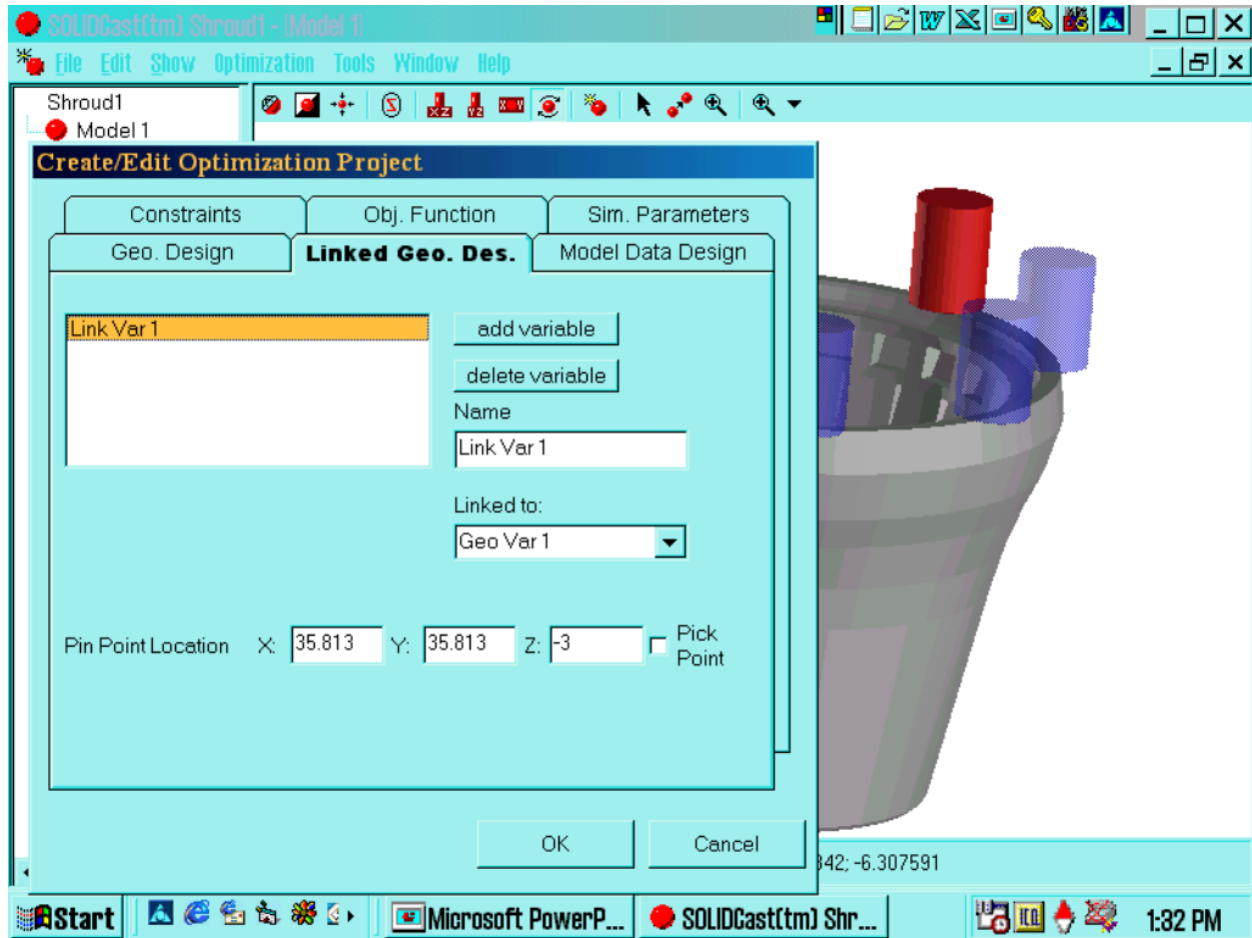
For example, the model shown below has 8 symmetrical risers.



If we designate one riser as a Geometric Design Variable, and allow it to vary both vertically and horizontally, we create two actual design variables, as shown on the next page.



Then we can select and designate another riser as a Linked Geometric Design Variable. This riser will have its own Pin Point, which we must specify. An example is shown on the next page.

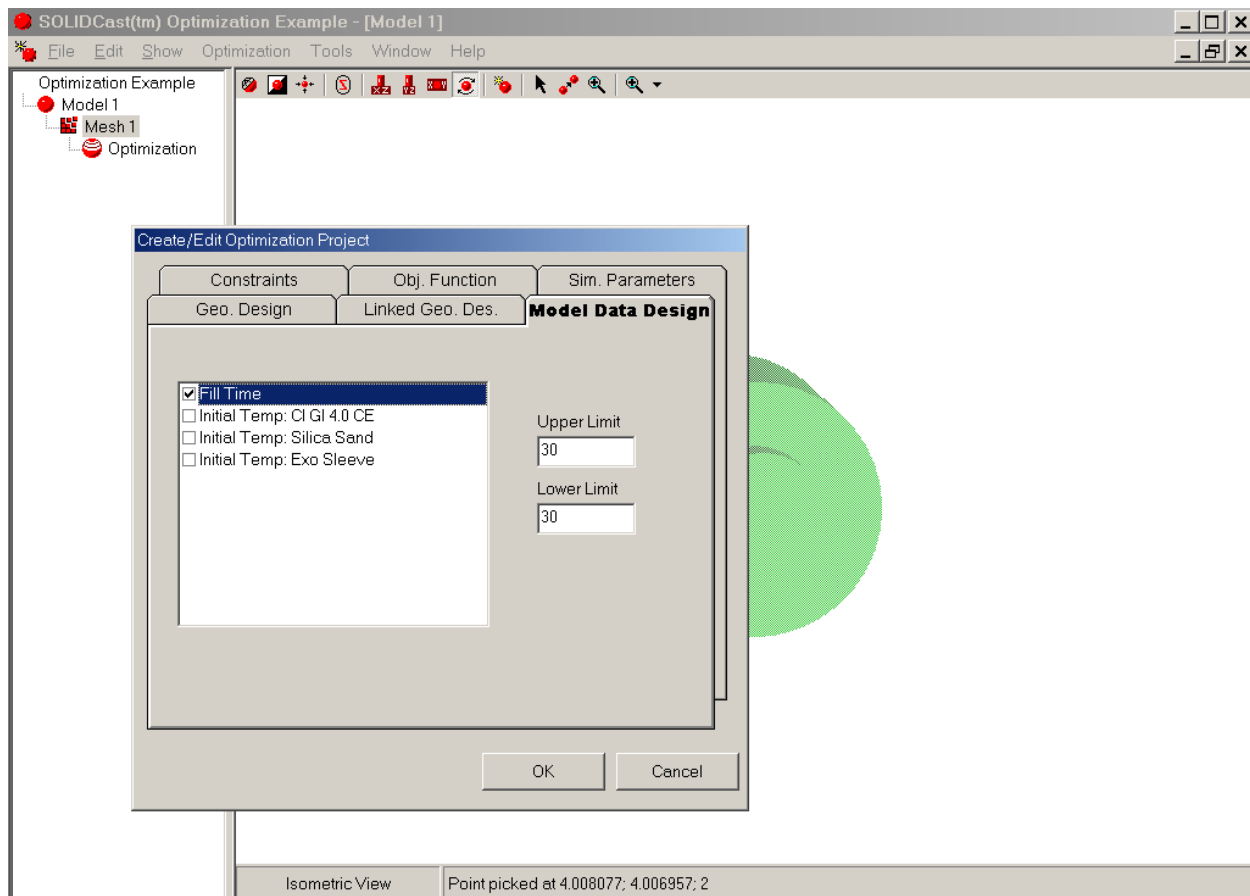


By continuing to use the Linked Geo. Design Tab, we can link all of the other risers to the original Design Variable. Then, as OPTICast varies the size of the first riser, all of the other risers will vary with exactly the same scale factors. In this way we still have only 2 Design Variables but are varying all 8 risers.



## UNIT 6: Putting Limits on Model Data

You may want to investigate variables that are process-related, rather than geometric in nature. For example, you may want to vary the pouring temperature or time, or perhaps the preheat temperature of an investment casting shell or permanent mold die. This is done using the Model Data Design Tab. A sample screen is shown below:

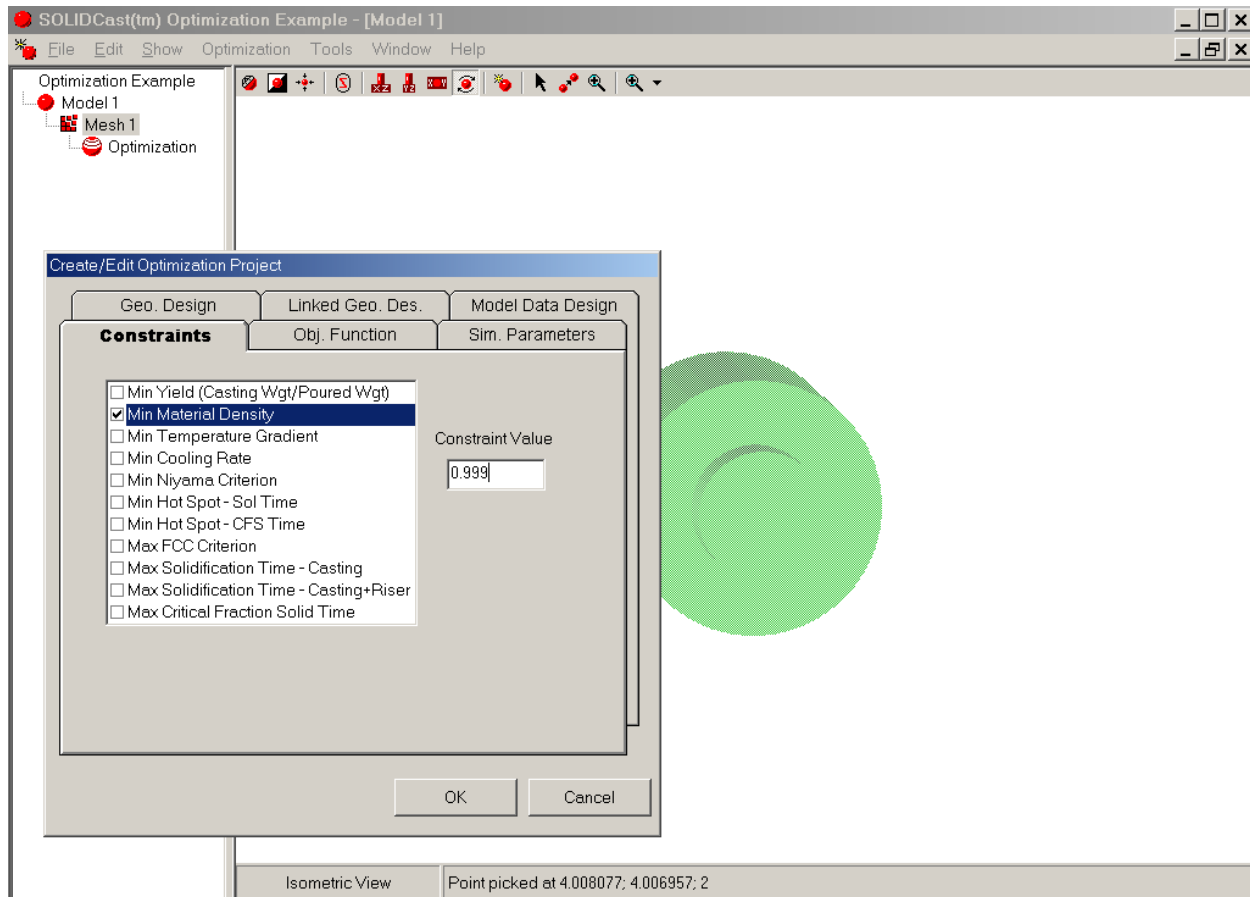


When you check one of the boxes, an upper and lower limit are shown for that variable, and you can adjust them. An example might be to vary the pouring time between 15 and 25 seconds for a particular casting, or to vary the pouring temperature up and down to see how casting quality is affected.



## UNIT 7: Setting Process Constraints

Process constraints are the way in which the system can tell if it has made a 'good' or a 'bad' part. Any one or all of the 11 constraints listed in the figure below can be used in a given optimization run.



In this example, the Minimum Material Density box was checked, and a Constraint Value of 0.999 was entered. In other words, all parts of the casting must have a material density value equal to or greater than 0.999 in order to be considered 'good'.

This is why it is important to create the gating system out of Riser Material. If the risers were made of casting material, there would always be porosity present and the optimization would fail.

One interesting technique is to create the riser contact out of casting material. This will, in effect, create a safety factor in the model, since the riser contact will have to be sound to satisfy the constraint. As long as the contact is created as a separate object from the rest of the riser, this can be done easily.

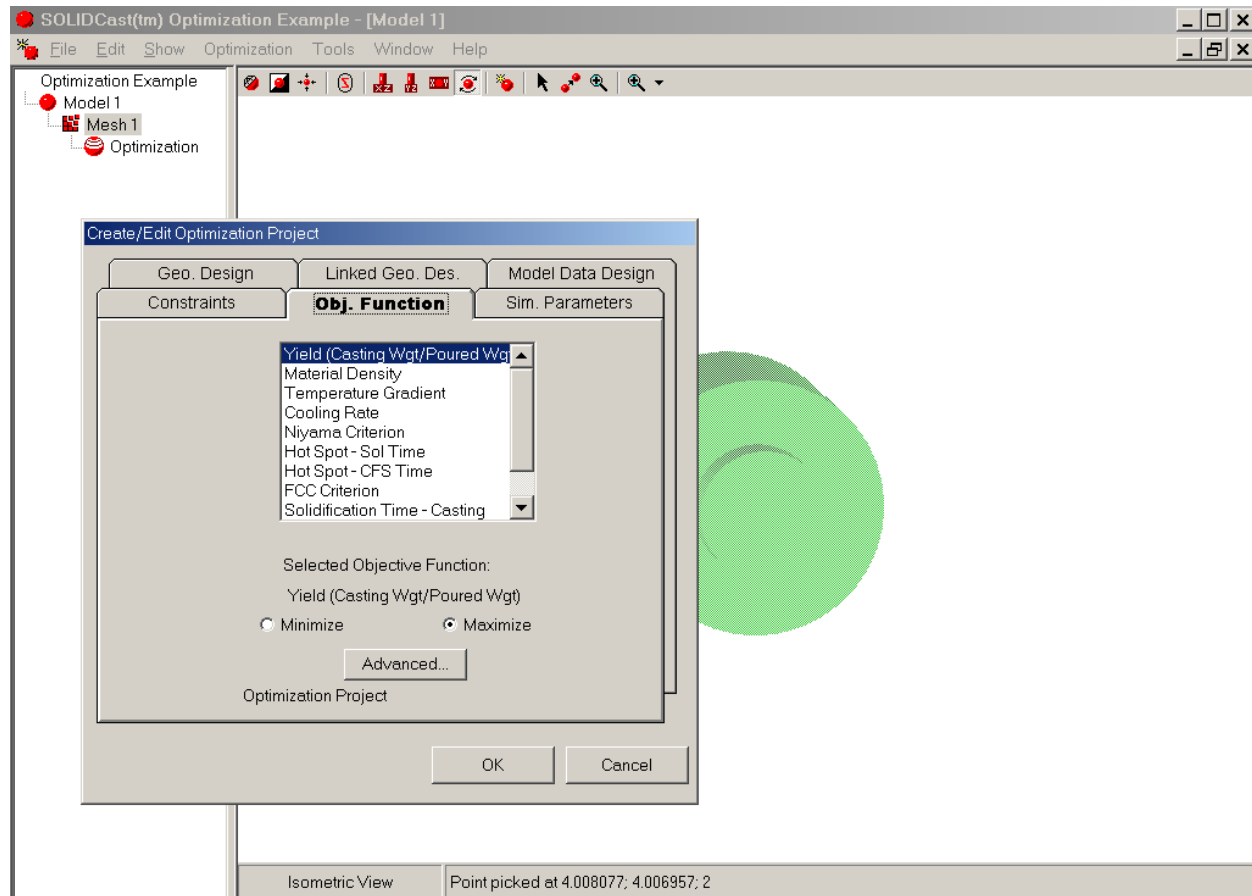
If you choose more than one constraint, each constraint must be satisfied in order for the part to be considered 'good'.





## UNIT 8: Setting the Objective Function

The objective is the goal for the optimization process. It is the end result that you are trying to achieve. While you can have many constraints, you can have only one objective. That objective will be to minimize or maximize one of the eleven functions listed on the Obj. Function Tab, shown below:



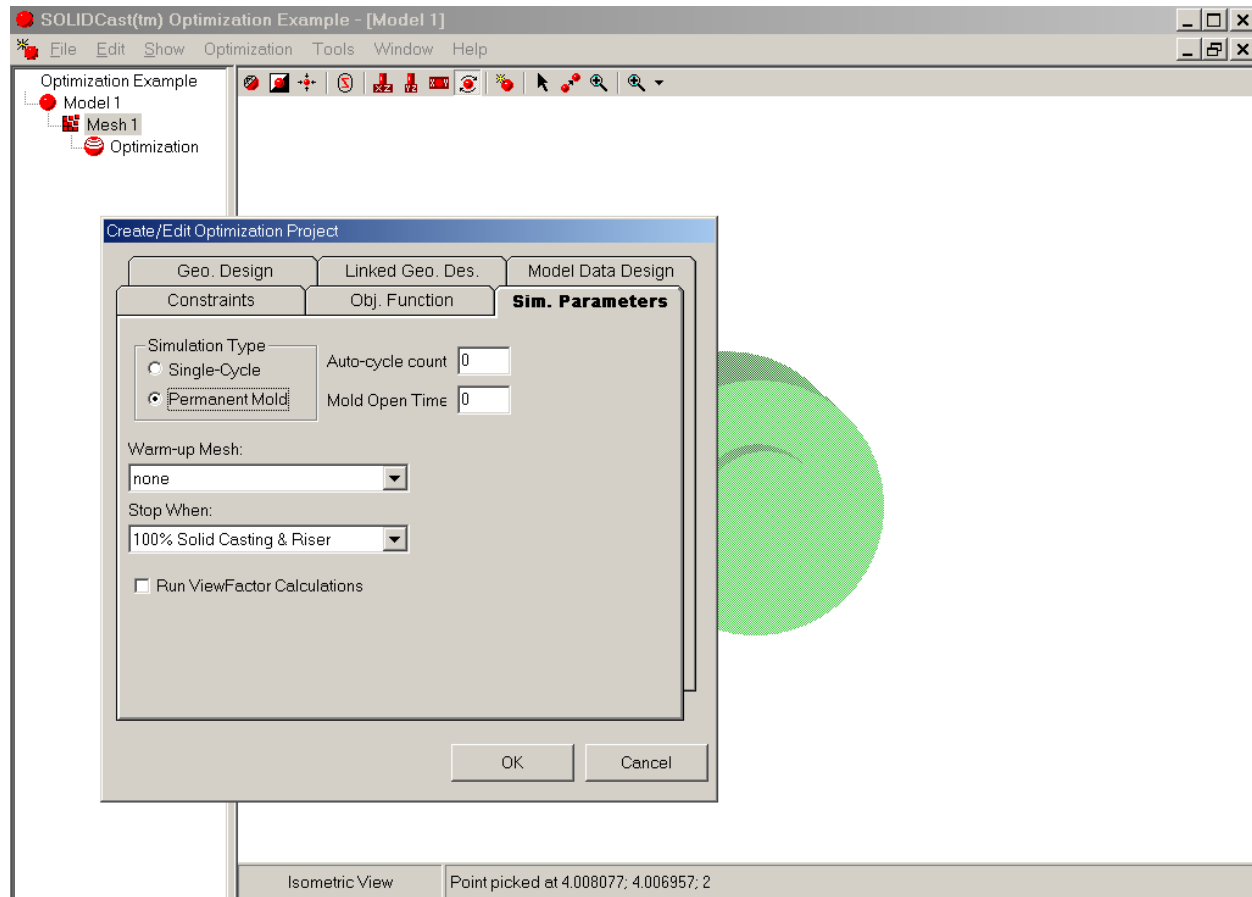
A common objective will be to maximize the yield, or the ratio of casting weight to poured weight. This is another reason why it is important to create the gating system out of Riser Material. If all metal were made of Casting Material, there would be no change in yield and that type of optimization could not be done.

The Advanced... button on the Obj. Function Tab allows you to set other parameters that control an optimization run. These will be further discussed in Unit 10.



## UNIT 9: Simulation Parameters

The Sim. Parameters Tab controls how a simulation will be run. Setting these parameters properly is important, because once an optimization run has started, there is no further operator intervention. If a simulation is run improperly, it will affect ALL simulations in the run.

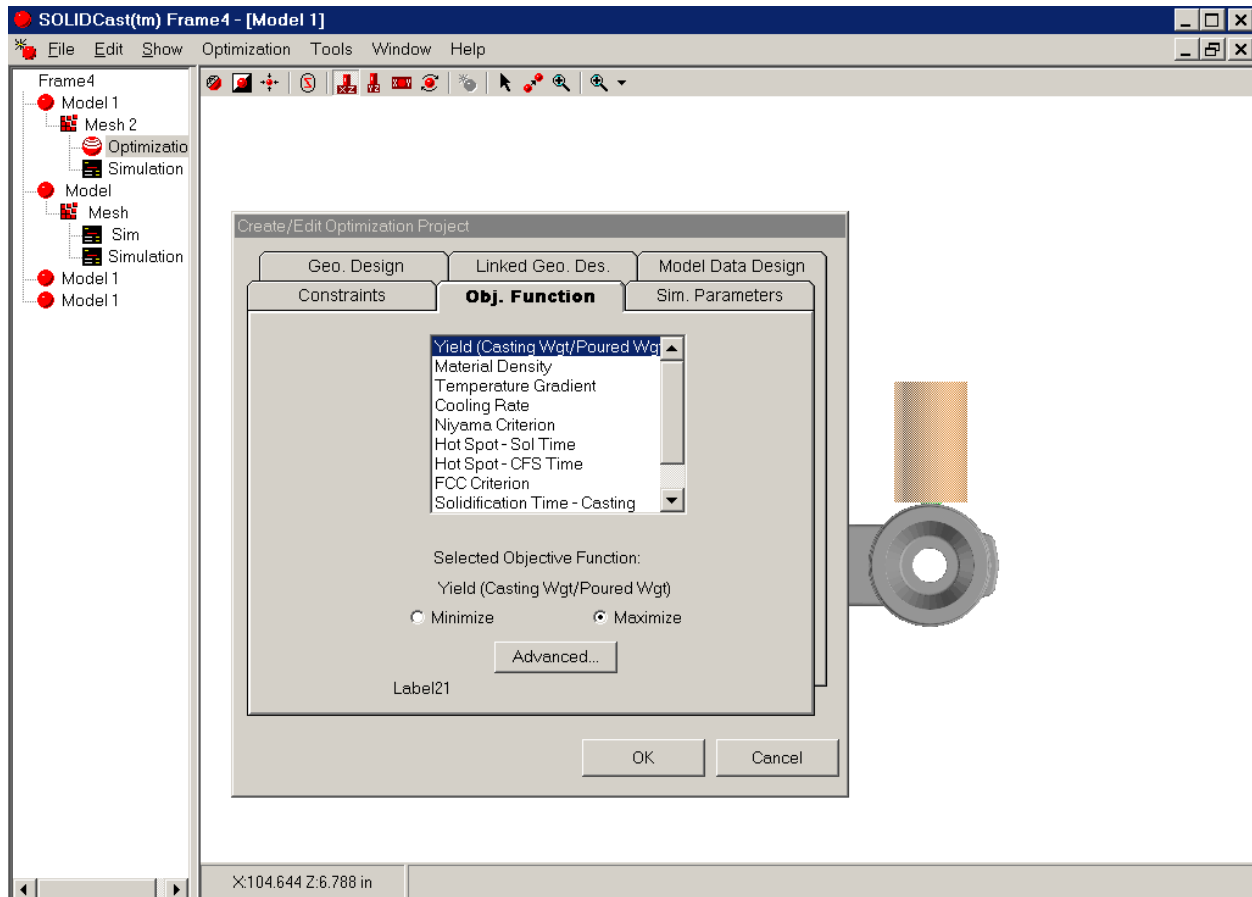


You can choose between single cycle and multiple cycle simulations, when to stop a simulation, and whether or not to run radiation view factor calculations on each mesh. One technique you can use to speed up an optimization run is to Stop When: 100% Solid Casting is selected. In this way you will not be running calculations just to find out how long it takes the risers to freeze. When the casting is solid, the evaluation will be done and the next simulation set up and run, if needed.

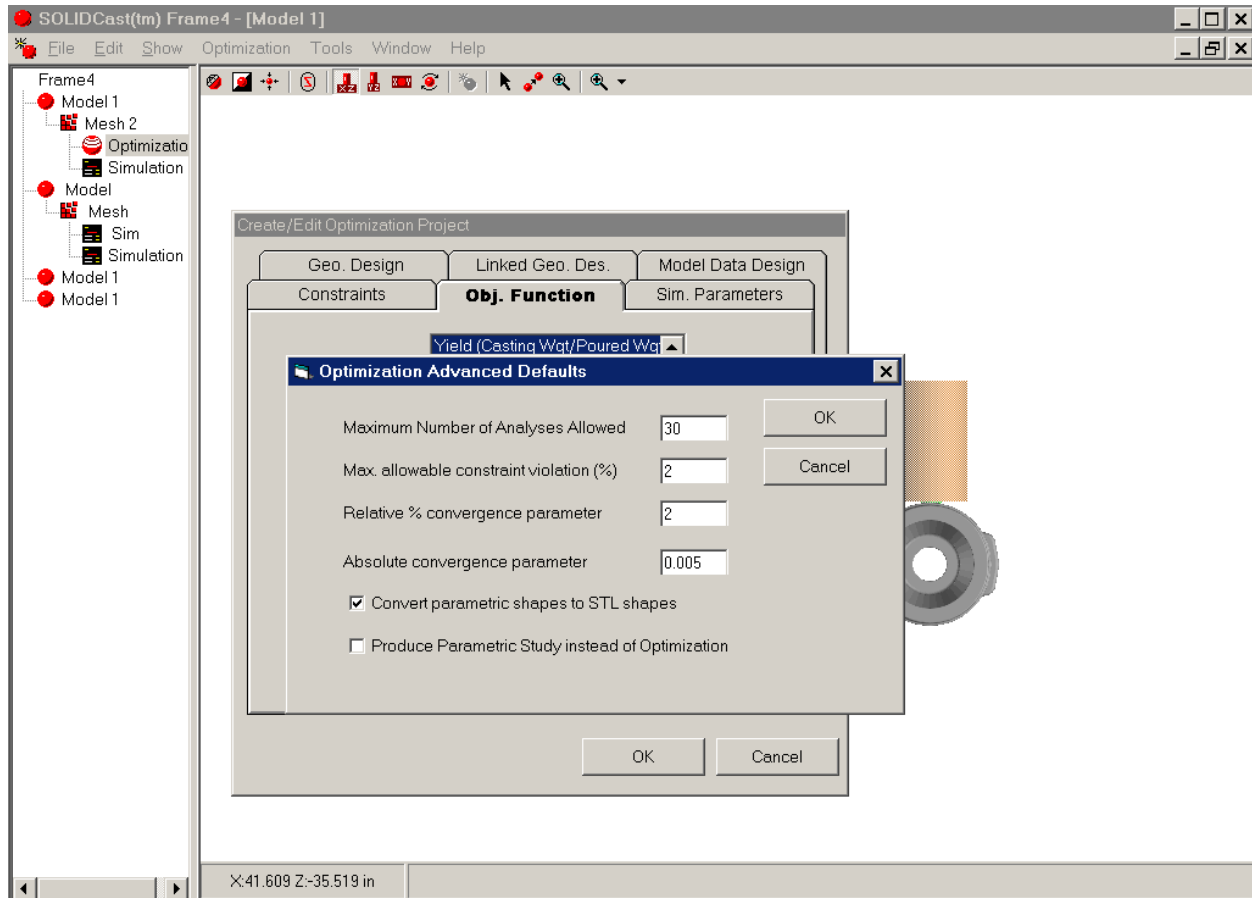


## UNIT 10: Advanced Optimization Parameters

When creating or editing an Optimization Project, and you are on the Obj. Function Tab, one of the options is to set up Advanced Optimization Parameters. This is done by clicking on the Advanced... button, as shown below:



This will bring up a window with the Optimization Advanced Defaults, as shown on the next page. These settings are described on the following pages.



**MAXIMUM NUMBER OF ANALYSES ALLOWED** - The parameter tells OPTICast how many simulations it may run, as a maximum, in searching for an optimum solution. As a very rough rule of thumb, the number of simulations required is the square of the number of Design Variables. If you have 4 DV's, the system may take around 16 simulations to find an optimum. It is a good idea to set this parameter a little higher, so a good setting with 4 DV's might be 25-30.

Remember that if you are allowing Vertical Scale and Horizontal Scale to vary for a riser, this makes 2 DV's for that riser.

**MAX. ALLOWABLE CONSTRAINT VIOLATION (%)** - OPTICast allows constraints to be violated by a small amount. This parameter gives OPTICast a "band" to work in to evaluate whether a design is feasible, i.e., whether it violates a constraint or not.

A good working number for this would typically be in the range of 1.5% to 2.0%.

Too small a percentage may make it difficult for OPTICast to find a solution.

**RELATIVE % CONVERGENCE PARAMETER** - When searching for an optimum condition, OPTICast looks at the Objective Function value on each subsequent simulation. If subsequent values are within this percentage, then OPTICast decides that it has found an optimum condition.

A good working number for this would typically be in the range of 1.5% to 2.0%.

If this value is too small, OPTICast may have trouble finding a true optimum.

**ABSOLUTE CONVERGENCE PARAMETER** - This is another parameter which OPTICast uses to look at subsequent values of the Objective Function and determine whether an optimum has been reached. Instead of a percentage, this parameter looks at the absolute difference in values from one simulation to the next.

A suggested starting value might be 0.005.

**CONVERT PARAMETRIC SHAPES TO STL SHAPES** - In SOLIDCast, model shapes can be Parametric (such as a cylinder) or STL shapes (as imported from CAD in STL format). When stretching shapes vertically and horizontally, there may be situations where a Parametric shape may not stretch correctly.

For example, a cylinder is always perfectly round. If a horizontal cylinder is scaled by one factor vertically and another factor horizontally, the result may not be correct since the cylinder will refuse to be elliptical in shape.

This parameter converts ALL shapes in the model to STL shapes, which can be scaled in ANY direction. We recommend turning "ON" this parameter.

**PRODUCE PARAMETRIC STUDY INSTEAD OF OPTIMIZATION** - There are times when you might like to evaluate how much effect a single Design Variable has on a given Objective Function (simulation output). Instead of running an optimization, you can select a Single-Variable Parametric Study. You must have specified only ONE Design Variable (Remember: Vertical and Horizontal Scale Factors are EACH DV's).

By selecting a Parametric Study, the system will ask how many study points to include. For example, six study points will cause OPTICast to run six simulations with the DV varying in even increments. You can then see how this single variable affects the output.





## UNIT 11: Running the Optimization

---

Once you have set up the optimization run, selected variables, constraints and your objective, you are ready to run the optimization process.

First of all, click OK on the Create/Edit Optimization Project window to close it. Highlight the optimization icon and click on the Optimization Menu. You can still edit variables at this point.

If you are ready to go ahead, click on Start Optimization Run. The system will begin to mesh the first model, then set up and run simulations based on your inputs. As the run continues, you can do other things with the computer, and the simulations will run in the background.

If you want to monitor progress, you can maximize the SOLIDCast™ window, highlight the Optimization icon, then click on Optimization...Report. If you click on View Graphs... you can look at the results of the simulations so far in a graphical format.

The typical number of simulations needed to get a good solution, if one can be found, is about the square of the number of variables. For example, if you have five variables, expect about twenty-five simulations to achieve an optimized solution. It is clear that you want to keep the number of variables as low as you can. It is also good practice to keep the meshes down to a minimum size, and use every available technique to keep run times down. Of course, optimization runs can be run at night, too, so in many cases even runs with up to 100 simulations could be done in a single night!



## UNIT 12: Reviewing Optimization Results

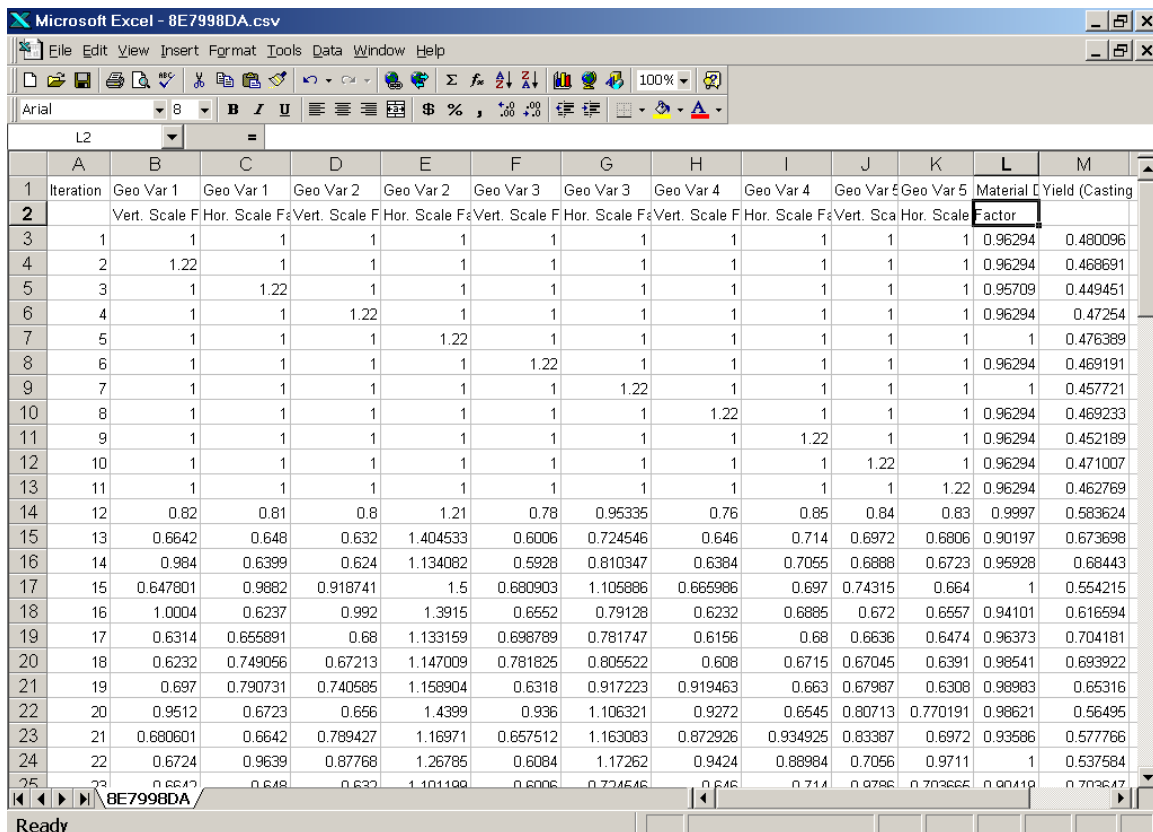
While an optimization run is in progress, no operator involvement is necessary. When the run is complete, the system will bring you back to the SOLIDCast Main Window, just as if you had completed a regular simulation.

At this point the Project Tree will have two sets of entries. The first will be your original model and mesh, along with the Optimization Icon. The second will be the final model, mesh and simulation of the optimized casting/process. This data can be examined and plotted just like any other simulation. Those procedures are covered in detail in the SOLIDCast training workbook.

In addition to the normal simulation results, you can view a summary of the iteration history, which is contained in an Excel spreadsheet, and also a series of graphs, which plot the variables, constraints and objective function over the range of simulation runs.

To view these results, load the project, if it isn't already, and click on the Optimization Icon. Then click on the Optimization Menu and choose Report.

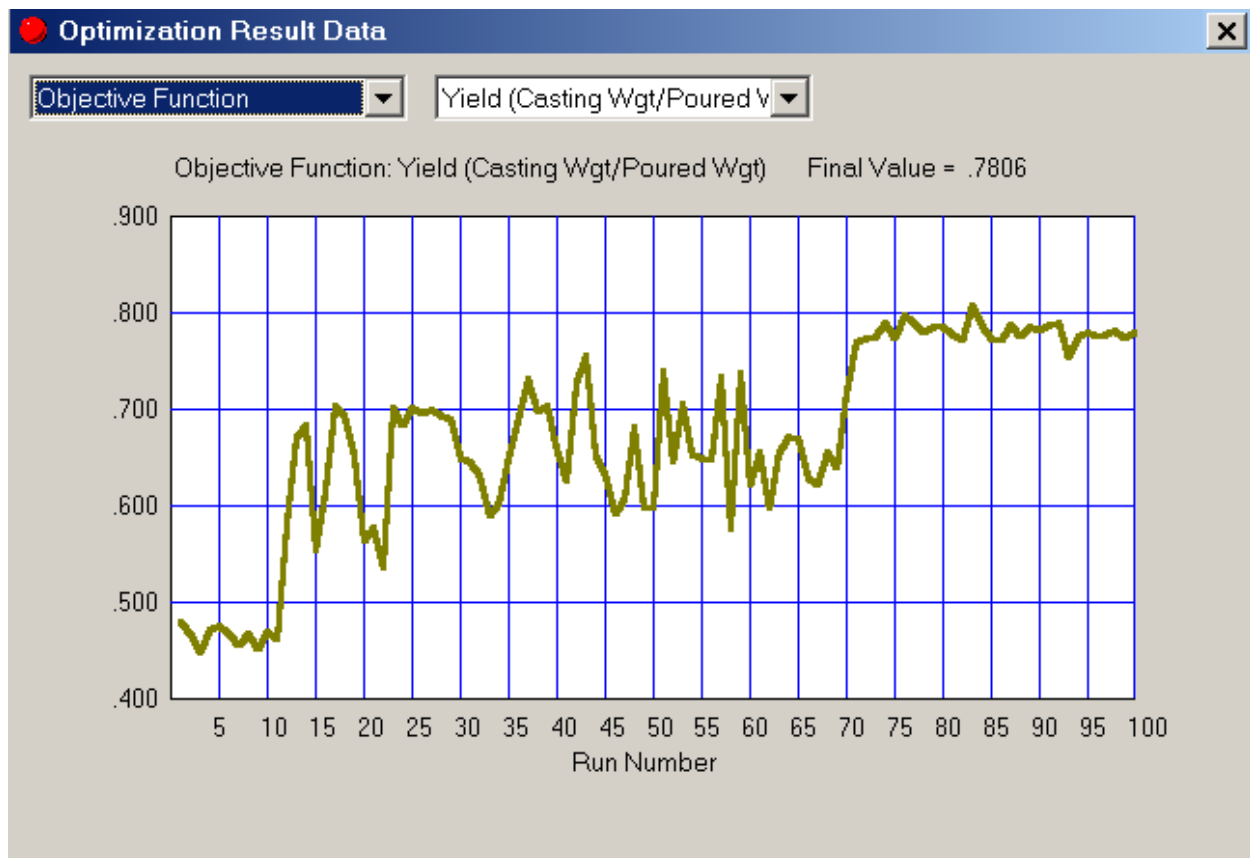
To look at the summary of each iteration, click on View Iteration Data(.CSV file)... This will automatically start Excel and load the data into the spreadsheet. Note: This requires that you have a copy of Excel installed on your computer. If you do not have Excel, the .CSV file can be loaded manually into any other spreadsheet. A typical example of what the file looks like is shown here:



Iteration	Geo Var 1	Geo Var 1	Geo Var 2	Geo Var 2	Geo Var 3	Geo Var 3	Geo Var 4	Geo Var 4	Geo Var 5	Geo Var 5	Material Yield (Casting Factor)		
1													
2		Vert. Scale F	Hor. Scale F	Vert. Scale F	Hor. Scale F	Vert. Scale F	Hor. Scale F	Vert. Scale F	Hor. Scale F	Vert. Scale F			
3	1	1	1	1	1	1	1	1	1	1	0.480096		
4	2	1.22	1	1	1	1	1	1	1	1	0.468691		
5	3	1	1.22	1	1	1	1	1	1	1	0.449451		
6	4	1	1	1.22	1	1	1	1	1	1	0.47254		
7	5	1	1	1	1.22	1	1	1	1	1	0.476389		
8	6	1	1	1	1	1.22	1	1	1	1	0.469191		
9	7	1	1	1	1	1	1.22	1	1	1	0.457721		
10	8	1	1	1	1	1	1	1.22	1	1	0.469233		
11	9	1	1	1	1	1	1	1	1.22	1	0.452189		
12	10	1	1	1	1	1	1	1	1	1.22	0.471007		
13	11	1	1	1	1	1	1	1	1	1	0.462769		
14	12	0.82	0.81	0.8	1.21	0.78	0.95335	0.76	0.85	0.84	0.9997	0.583624	
15	13	0.6642	0.648	0.632	1.404533	0.6006	0.724546	0.646	0.714	0.6972	0.6806	0.90197	0.673698
16	14	0.984	0.6399	0.624	1.134082	0.5928	0.810347	0.6384	0.7055	0.6888	0.6723	0.95828	0.68443
17	15	0.647801	0.9882	0.918741	1.5	0.680903	1.105886	0.665986	0.697	0.74315	0.664	1	0.554215
18	16	1.0004	0.6237	0.992	1.3915	0.6552	0.79128	0.6232	0.6885	0.672	0.6557	0.94101	0.616594
19	17	0.6314	0.655891	0.68	1.133159	0.698789	0.781747	0.6156	0.68	0.6636	0.6474	0.96373	0.704181
20	18	0.6232	0.749056	0.67213	1.147009	0.781825	0.805522	0.608	0.6715	0.67045	0.6391	0.98541	0.693922
21	19	0.697	0.790731	0.740585	1.158904	0.6318	0.917223	0.919463	0.663	0.67987	0.6308	0.98983	0.65316
22	20	0.9512	0.6723	0.656	1.4399	0.936	1.106321	0.9272	0.6545	0.80713	0.770191	0.98621	0.56495
23	21	0.680601	0.6642	0.789427	1.16971	0.657512	1.163083	0.872926	0.934925	0.83387	0.6972	0.93586	0.577786
24	22	0.6724	0.9639	0.87768	1.26785	0.6084	1.17262	0.9424	0.88984	0.7056	0.9711	1	0.537584
25	23	0.6642	0.648	0.632	1.101198	0.6006	0.724546	0.646	0.714	0.6972	0.703665	0.90197	0.703647

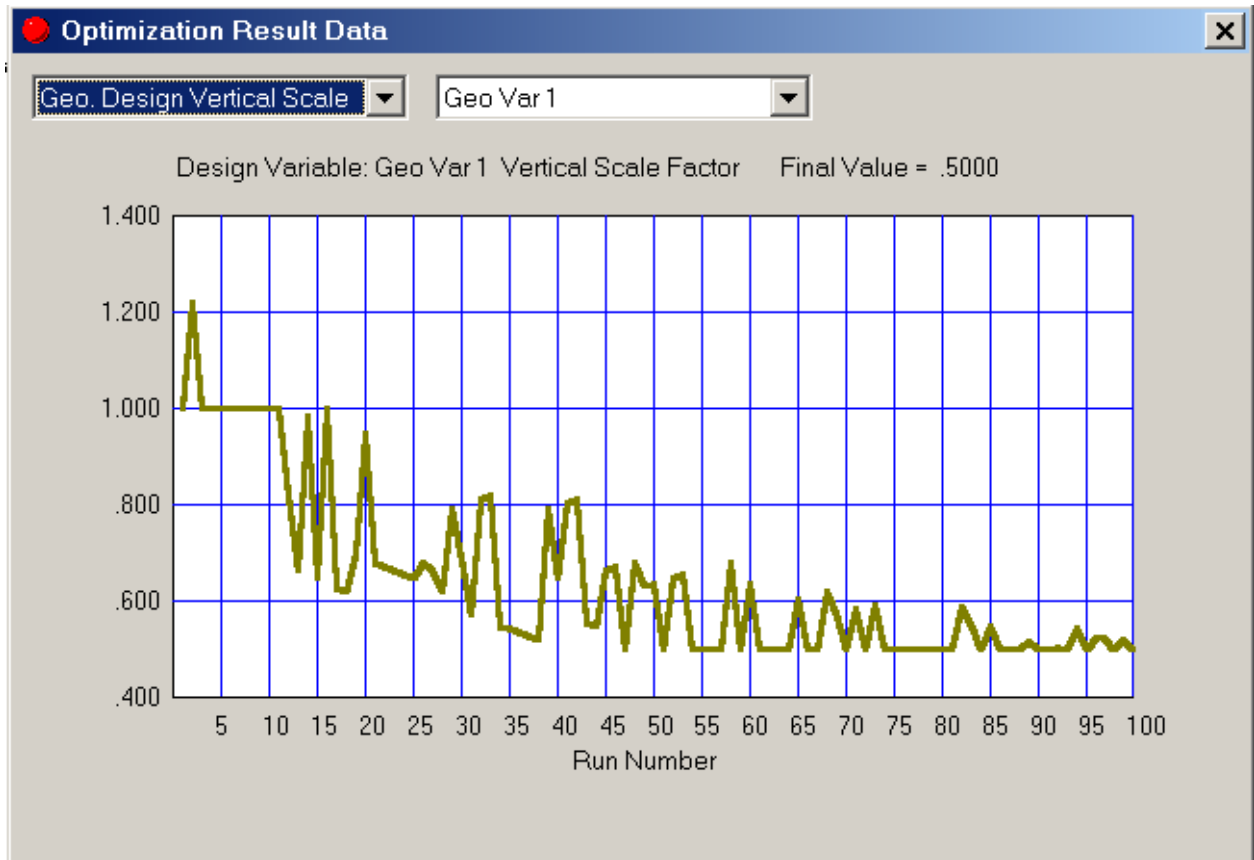
There will be an entry for each iteration, which will list the values of each variable, the constraints, and the objective function. This is a quick way to see how the optimization run progressed. As you can see from the sample file, the first simulations are single variable changes, to find the slope of the response surface of each variable. Once each contribution is found, the optimization runs begin in earnest.

The other way to view the data is in graphical format. To do this, go to the Optimization Menu and select Report...View Graphs. The first graph to be displayed will be the Objective Function, as shown here:

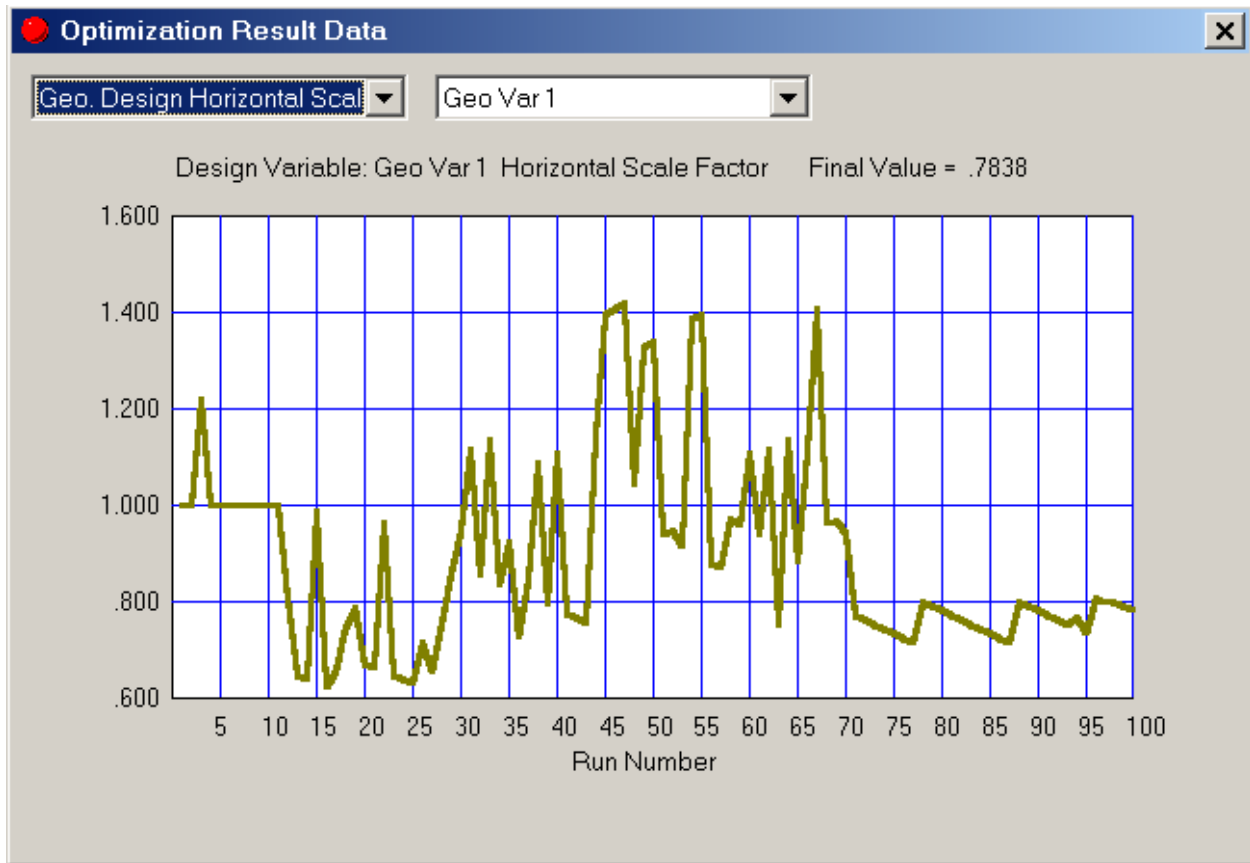


In this example, the casting yield was improved from about 48% to over 78%, all without operator involvement!

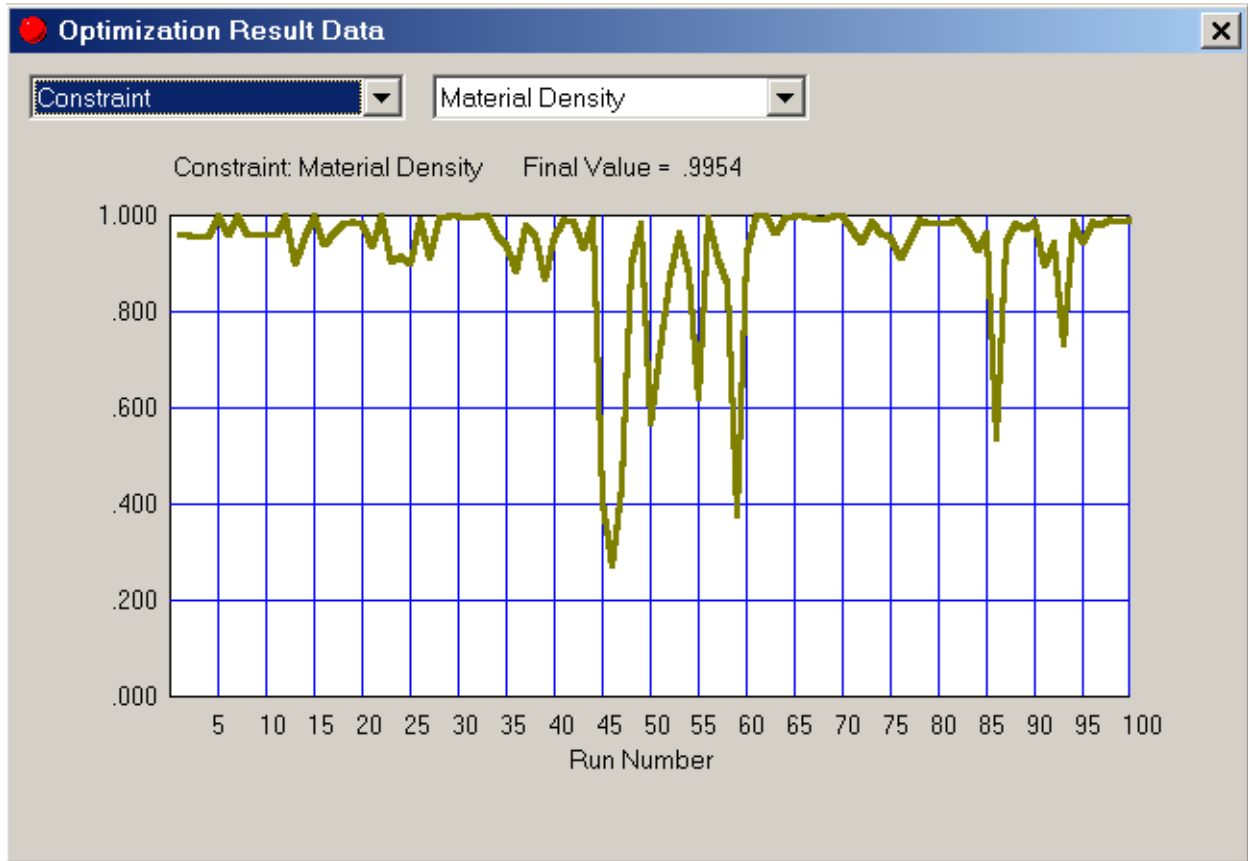
Other data that can be plotted include the design variables, model variables and constraints. Several examples of these are shown on the following pages.



The figure above shows the progression of the vertical scale, or height, of a riser. Note that the final height is one half(0.500) of the original height.

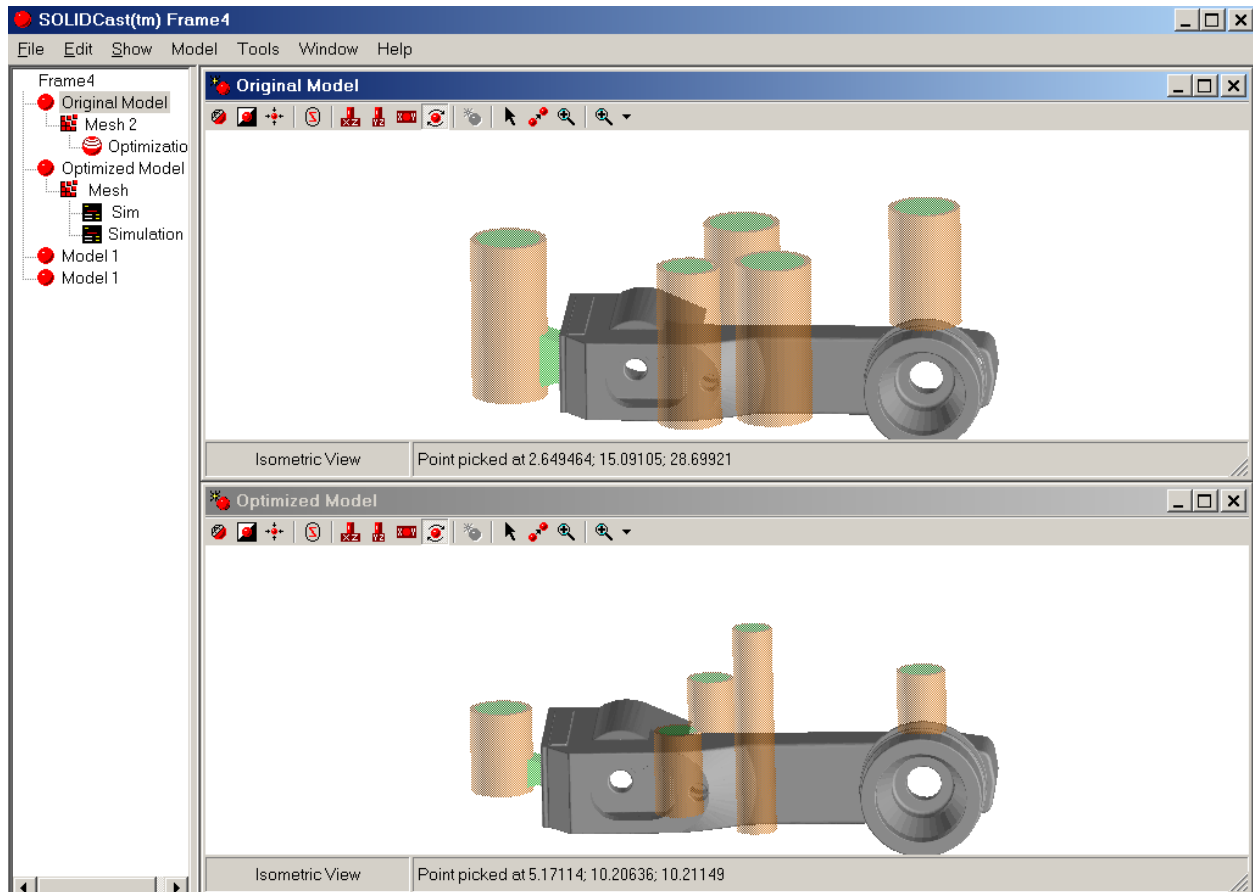


This chart shows the progression of a riser horizontal scale, or diameter, through the 100 runs. The final diameter was reduced to about 78% of the original diameter.



This chart shows the constraint of Material Density. The final value was 0.9954, which was acceptable in this particular run. The minimum value specified was 0.994.

A final comparison can be made by visual inspection. The screen below shows the first and last model, side by side, so you can compare the rigging systems.



All changes to the model, meshing and simulation work was done completely without operator help, once the optimization run was set up.



## UNIT 13: Optimization Strategies

---

There are a number of things to take into consideration when running optimizations, as opposed to simply running a single simulation. These items are presented in this section, in Question and Answer format.

**QUESTION:** How do I get started?

**ANSWER:** You need an initial casting/process design in SOLIDCast, just as if you were running a typical simulation.

**QUESTION:** Is there anything special about the construction of a model for use with OPTICast?

**ANSWER:** YES! When looking at the results of simulations, OPTICast evaluates only the Casting Material in a model. Gates and risers should therefore be constructed of Riser Material only, so their material will not be included in the evaluation of results.

**QUESTION:** What if I am importing geometry from a CAD system in the form of STL files?

**ANSWER:** It is necessary for each riser that will be a Design Variable to be a separate shape. This means that each riser and each shape that will vary must be stored and imported as a SEPARATE STL file into SOLIDCast.

**QUESTION:** How can I apply a "factor of safety" to the results obtained from OPTICast?

**ANSWER:** One method is to construct a lower portion of a riser out of Casting Material. Then, when the results are evaluated (for example, Material Density) a "safety zone" will be created by this lower portion of the riser.

**QUESTION:** Is there anything special about how the model should be meshed?

**ANSWER:** In general, it is best to specify "Node Size" rather than "Number of Nodes" when meshing, so that all simulations will be evaluated using the same node size.

**QUESTION:** What are we doing when we create an Optimization Project?

**ANSWER:** We are doing several things. First of all, we're defining Design Variables. These are any aspects of the model that are allowed to vary while searching for an optimum. These might be Geometric Design Variables, where a shape varies, or Model Data Design Variables, where some data like initial temperature varies.

Secondly, we're defining Constraints. These are results that we want to specify as a maximum or minimum acceptable condition. For example, we might set a value of 0.995 as a minimum acceptable value for Material Density.

Third, we're defining an Objective Function. This is the one result that we are trying to achieve. This might be maximization of yield, or minimization of the FCC Criterion (microporosity).

In addition, there are some Optimization Parameters and Simulation Parameters that we can set.

**QUESTION:** Is there a limit on the number of Design Variables in an Optimization Project?

**ANSWER:** No, but you should in general try to keep the number of Design Variable as small as possible, to reduce run times. If the riser height can't vary, don't make it a Design Variable. More DV's increase the number of simulations required.

**QUESTION:** Can a model shape be included in more than one Design Variable?

**ANSWER:** No. A Specific shape in a model can be included in ONLY ONE Design Variable.

**QUESTION:** Does OPTICast save the results of every simulation that it runs?

**ANSWER:** No. Only the final model and simulation results are available for loading and plotting in SOLIDCast. The intermediate results can be seen with the charts and spreadsheet reports.

**QUESTION:** Will OPTICast always be able to find an optimum solution?

**ANSWER:** No. In some cases OPTICast may not be able to find a feasible solution, no matter what size the risers are extended to. For example, if Material Density of 0.998 is set as a Constraint, it is possible that given an initial design, OPTICast may not find ANY design which satisfies this Constraint.

**QUESTION:** How can I establish a better initial design?

**ANSWER:** One good way is to run a simulation of the casting without any gates or risers in SOLIDCast, and locate the last areas to solidify in each part of the casting. Feeding these areas will provide the best chance of a successful design. The Gating and Riser Design Wizards included in SOLIDCast provide the best single source to create a good initial rigging design.

**QUESTION:** What factors make the best Constraints and Objective Functions to select?

**ANSWER:** We have generally found that for ferrous castings, a good starting point is to select Material Density close to 1.000 as a constraint, and maximizing Yield as an Objective Function.

For non-ferrous castings, specifying the FCC Criterion (microporosity) as a Constraint is also a good idea.

**QUESTION:** Are there occasions when maximizing Yield is not necessarily the Objective Function to select?

**ANSWER:** Yield maximization is generally a goal for medium to high-volume production (given a sound casting as a constraint). For low-volume or prototype production, producing a sound part (maximizing Material Density or minimizing FCC Criterion or Niyama) might be a more reasonable objective.

**QUESTION:** Can OPTICast move a riser to change its location on the casting?

**ANSWER:** No. Not in this release. However, we will be considering a method of incorporating this capability in a future release.

**QUESTION:** Can OPTICast be used to minimize porosity in just a "zone" in the casting rather than the whole casting?

**ANSWER:** Since OPTICast considers result data in Casting Material only, you could create a casting model using Casting Material for the zone of interest, and Riser Material for the rest of the casting. This would distort the "Yield" calculation; however, you might be able to use "Minimize Solidification Time" as a replacement objective for "Yield Maximization".

**QUESTION:** What type of computer hardware is required for OPTICast?

**ANSWER:** The best and fastest that you can get! We recommend a multi-core machine with 4GB or more of RAM and a 500 GB hard drive or larger. Also, we recommend a DEDICATED computer, not one that will be used concurrently for other tasks.