Introduction to Optimization Module

VERSION 4.3

COMSOL

Introduction to the Optimization Module

Version: May 2012 COMSOL 4.3

Contact Information

Visit www.comsol.com/contact for a searchable list of all COMSOL offices and local representatives. From this web page, search the contacts and find a local sales representative, go to other COMSOL websites, request information and pricing, submit technical support queries, subscribe to the monthly eNews email newsletter, and much more.

If you need to contact Technical Support, an online request form is located at www.comsol.com/support/contact.

Other useful links include:

- Technical Support www.comsol.com/support
- Software updates: www.comsol.com/support/updates
- Online community: www.comsol.com/community
- Events, conferences, and training: www.comsol.com/events
- Tutorials: www.comsol.com/products/tutorials
- Knowledge Base: www.comsol.com/support/knowledgebase

Part No. CM021702

Contents

Introduction

This introduction is intended to give you a jump-start in your modeling work. It contains examples of the typical use of the Optimization Module and two tutorial examples that introduce the workflow. The first tutorial solves a curve fitting problem and the second example contains a topology optimization for an MBB beam.

You can use the Optimization Module throughout the COMSOL Multiphysics product family—it is a general interface for computing optimal solutions to engineering problems to, for example, improve a design so that it minimizes energy consumption or maximizes the output. Any model inputs—geometric dimensions and part shapes, material properties, and material distribution—can be treated as design variables, and any model output can be an objective function.

There are two optimization algorithms available in the Optimization Module. The first algorithm is based on the SNOPT code developed by Philip E. Gill of the University of California San Diego, and Walter Murray and Michael A. Saunders of Stanford University. When using SNOPT, the objective function can have any form and any constraints can be applied. The algorithm uses a gradient-based optimization technique to find optimal designs, and when the underlying PDE is stationary, analytic sensitivities of the objective function to the design variables can be used.

The second algorithm is a Levenberg-Marquardt solver. You can use this solver when the objective function is of least-squares type. Because the Levenberg-Marquardt method is derived to solve problems of least-squares type, it typically converges faster than SNOPT for such problems.

The following important classes of problems have resolutions that rely on a systematic exploratory process and therefore benefit from the Optimization Module:

- *Design problems* with a single objective.

  Here the problem is to find the values of some decision variables that yield the best performance of the output of a simulation model when the latter is quantified by means of a single function. Problems of this kind arise in, for example, structural optimization, antenna design, and process optimization.

- *Inverse problems*, and in particular *parameter estimation* in PDEs.

  Here the problem is to reliably determine the values of a set of parameters that provide simulated data that best matches measured data. Such problems arise in applications such as geophysical imaging, nondestructive testing, biomedical imaging, and weather data assimilations. *Curve fitting* also belongs to this category.

It is often possible to reformulate problems of the above type as optimization problems. The Optimization interface in COMSOL Multiphysics is useful for solving design problems as well as inverse problems and parameter estimation.

The work flow in the Optimization Module is quite straightforward and can be described by the following steps:

- For standalone optimization, use a "0D model" and define the global objective function, constraints, and optimization variables.
- For multiphysics optimization, define the geometry and the physics, and then add an Optimization interface where you define the objective, constraints, and control variables, which can be variables from the physics interfaces in the model.

Then define the finite element mesh, select a solver, and visualize the results. All these steps are accessed from the COMSOL Desktop. The mesh and solver steps are usually carried out automatically using default settings, but to activate the optimization solver, select the Optimization check box in the Study Extensions section for the Stationary and Time Dependent study step settings.

Example models using the Optimization Module are in the COMSOL Multiphysics Model Library and in several of the modules. Search for *optimization* in the Model Library window's search field to find the models that use the Optimization Module in the products that your license includes. These models describe the Optimization interface and its different features through tutorial and benchmark examples for various types of optimization. The set includes models of topology optimization, shape optimization, flow minimization, maximization of the reaction rate, and inverse modeling.

An Overview of Optimization Techniques

Assuming that you have solved a COMSOL model of a physical design, it is common to ascribe to it some figure of merit, or *cost function*. This cost function can quantify almost anything about the performance of the system. It is natural to question how this cost function changes when you vary some design parameters. These parameters can be any part of the design: dimensions, loads, boundary conditions, material properties, material distribution, or other scalar model parameters. These design parameters also often have a set of constraints associated with them: dimensions must be within some limits, only certain materials are possible, and so on. An *optimization problem* is when you want to improve the cost function by varying the design parameters within some set of constraints.

The Optimization Module contains the framework and functionality needed to perform gradient-based optimization on an existing COMSOL model. The capabilities can be further categorized as topology, size, shape, and parameter optimization.

About Parameter Estimation

You can use the Optimization Module to perform parameter estimation: To determine what the correct inputs are for a model based upon some experimental data—that is, to solve an *inverse problem*. The first tutorial, which starts on page 14, describes how to use the Optimization Module for parameter estimation. When solving an inverse problem, the shape of the structure is known, and there is some set of experimental data that can be compared to the output of the model. Parameter estimation is used to adjust the other model inputs such that the results agree with the experimental data. Typical objectives include estimating material properties or loads.

For several reasons, inverse problems are inherently the most difficult problems to solve. They are usually ill-posed in that there is not one unique solution to the problem but a set of possible solutions. The solution or solutions can be very sensitive to small perturbations in the experimental data or the structure. It is also easy to pose an inverse problem that has no solution. For these reasons, it is best to approach inverse problems with a good understanding of the underlying physics and mathematics of the problem.

Consider a COMSOL model of a physical system, such as a sample of a rubber-like material in simple uniaxial tension. One way to model rubber-like materials is with a Mooney-Rivlin model, which uses empirically determined material constants to describe the structural deformation. Assuming that there are experimental results for load and deflection, as shown in Figure 1, it is possible to use parameter estimation to determine the material model constants that best fit this data. The curve shown in Figure 1 is computed based upon the results of a parameter estimation; it is a least-squares fit of a COMSOL model with Mooney-Rivlin material parameters to the experimental data.

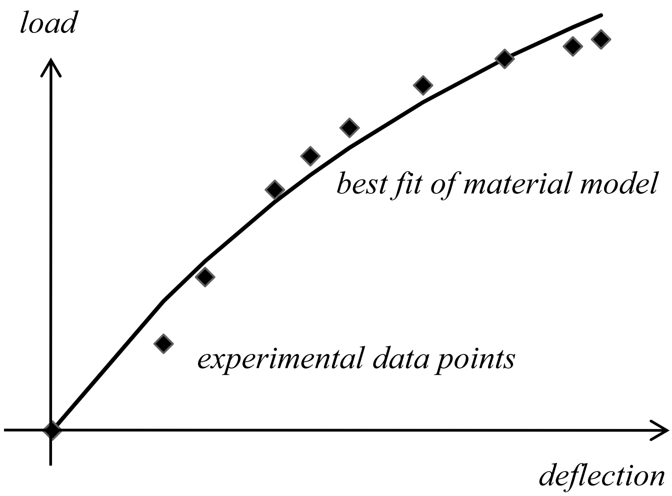


Figure 1: Experimental data and a best fit curve of the Mooney-Rivlin material model based upon a finite element model of a test sample.

About Topology Optimization

The second tutorial model in this guide, which starts on page 23, shows how to solve a model using topology optimization.

Consider a COMSOL model of a physical system, such as the bracket shown in Figure 2. The part is constrained at the top and loaded on the side. The part must support a known load while staying within acceptable limits on the deflection. However, an experienced structural engineer can quickly recognize that the part is overdesigned. It is possible to remove material from the part, which would reduce the total mass (the cost) of the part, while still staying within the limits on the deflection (the constraints). This optimization problem can alternatively be stated as: Minimize the compliance (the deflection) of the structure such that the total mass of the part is within some limit.

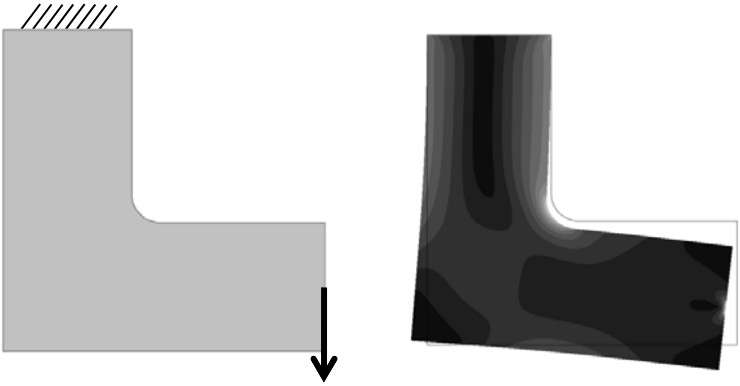


Figure 2: A simple bracket design, and the solved finite element model showing stresses and deflections. This design has room for improvement.

One way to address this is to use *topology optimization*. Topology optimization treats the material distribution, as defined by the finite element mesh, as the design variables. Each element in the finite element mesh has an additional scalar variable that is used to track the presence of the material within that element. The set of these scalar variables, one per element, are the design variables to be changed in an attempt to improve the cost function, while satisfying the constraints.

The Optimization Module takes the *derivative,* or the *sensitivity*, of the objective function with respect to these design variables using the *adjoint method*. It then alters these variables, within the set of user-defined constraints. This is an iterative procedure, as illustrated in Figure 3. Starting with the initial design, the variables defining the presence of material are adjusted on an element-wise basis. This is repeated as long as some improvement in the objective function is possible and the constraints are not violated.

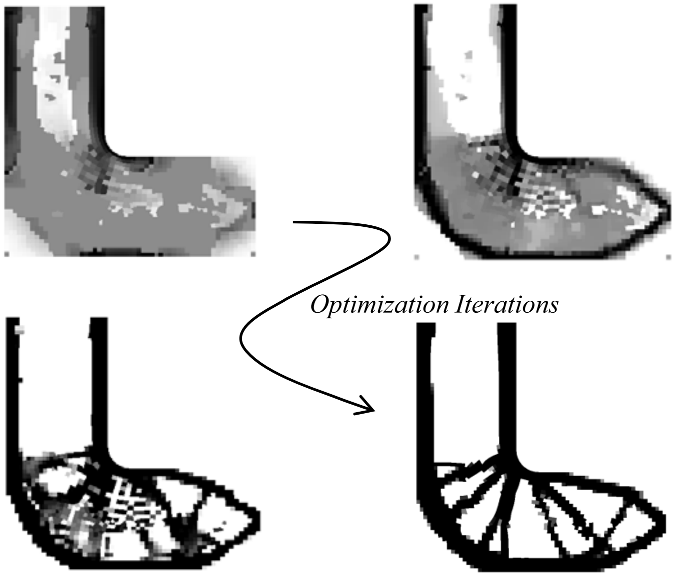


Figure 3: Topology optimization of the bracket, the material distribution, as defined by the finite element mesh, is iteratively refined to reduce the mass, while satisfying a constraint on peak displacement.

A structural engineer also recognizes that the resultant design is an approximation of a truss structure. This result also highlights an issue with topology optimization: The optimal design is often not a candidate for manufacture, it is only an approximation of a better design, as found via the current finite element mesh. With a modicum of engineering knowledge it is still possible to use the results of a topology optimization as a design guide to build the next design iteration.

Although topology optimization is quick to set up, it has a couple of drawbacks. First, the resultant design does have some dependency on the mesh used. There are techniques known as *regularization* that you can use to avoid this, but they do make the setup more complex. The topology model example in this guide includes regularization. Second, it is difficult to incorporate geometric constraints (constraints on bend radius or feature thickness, for example) into the optimization algorithm. Without incorporating such constraints, topology optimization can often come up with designs that are impractical. Therefore, topology optimization should be used early in the design process as a guide for the overall structure of the system. It becomes more difficult to use efficiently when the design is already quite rigid and only minor modifications to the design are possible. In such situations, *size optimization* and *shape optimization* become more useful.

The Use of the Optimization Module

The Optimization Module provides versatile tools for optimization, curve fitting, parameter estimation, and inverse modeling. The optimization can be to solve a basic linear or quadratic programming problem but you can also seamlessly incorporate optimization and parameter estimation as additions to, for example, models of structural mechanics, fluid flow, or chemical reactions. The optimizations can be an optimal shape or topology, or a maximization or minimization of some quantity, just to name a couple of examples.

The plot in Figure 4 shows the hydraulic conductivity obtained by inverse modeling using 24 observations in a model where the aim is to characterize an aquifer. The model is in the Subsurface Flow Module Model Library and combines the Optimization interface with a Darcy's Law interface for the fluid flow in the aquifer. The inverse problem in this model is to estimate the hydraulic-conductivity field in the aquifer using experimental data in the form of hydraulic-head measurements from four dipole-pump tests.

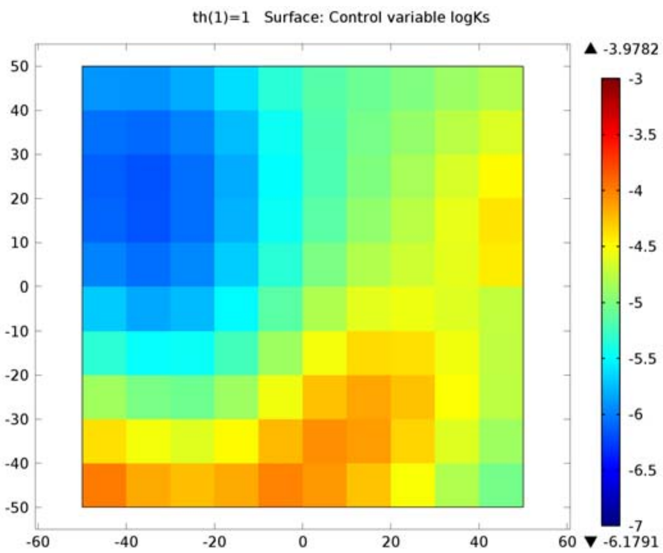


Figure 4: Hydraulic conductivity obtained by inverse modeling in the Aquifer Characterization model.

Another application of the Optimization Module is to find the optimal distribution of porous material in a microchannel, where the objective is to minimize the horizontal velocity at the center of the channel. The plot in Figure 5 shows the x-component of velocity field for the optimal solution and the distribution of the filling material in the open channel:

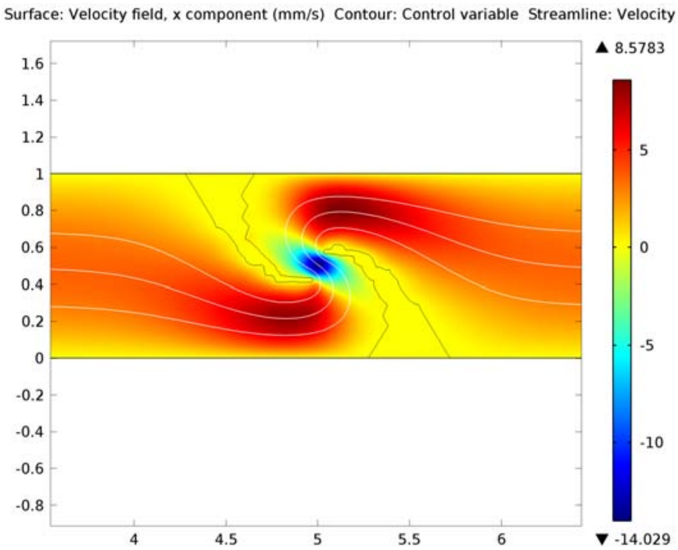


Figure 5: The horizontal velocity (surface plot) and velocity field (streamlines) after optimization. In addition, the contour γ = 0.5 indicates the border between open channel and filling material.

This model is in the COMSOL Multiphysics Model Library. It combines a Laminar Flow interface for the single-phase flow in the channel and an Optimization interface, where the objective is a scaled x-direction velocity that the optimization solver minimizes. The design variable `gamma` (γ) can be interpreted as the local porosity, ranging between 0 (filled) and 1 (open channel), so the contour $\gamma = 0.5$ indicates the border between the open channel and the filling.

Opening the Model Library

To open a COMSOL Multiphysics model, select **View>Model Library** from the main menu in COMSOL Multiphysics. In the Model Library window that opens, expand the available folders and browse or search the contents. Click **Open Model and PDF** to open the model in COMSOL Multiphysics and a PDF to read background theory about the model including the step-by-step instructions on how to build it.

The Optimization Interface

The Optimization Module adds one interface to the Model Wizard's list of physics interfaces: the Optimization interface ( ), which is found under the **Mathematics>Optimization and Sensitivity** branch.

The Optimization interface includes a wide range of features for defining an optimization problem, separately or in connection to a physics model:

- Global objectives, including least-squares objectives
- Global constraints
- Global control variables
- Integral objectives on all geometry levels: domains, boundaries, edges, and points
- Integral constraints on all geometry levels
- Pointwise constraints on all geometry levels
- Control variable fields on all geometry levels
- Probe objectives in domains (an objective defined by a point evaluation of an expression)

Tutorial Example—Curve Fitting

This tutorial model uses the Optimization interface for curve fitting (parameter estimation). In this example, the task is to fit two parameters that define a nonlinear material model in solid mechanics to some measured data. The material model is the hyperelastic Mooney-Rivlin material model, which is available in the Nonlinear Structural Materials Module. There are two parameters, C_{10} and C_{01}, that define the Mooney-Rivlin model. The objective is to fit the values of these parameters to measured data. You can always extend the concepts shown here to multivariate optimization as long as a suitable objective function is available.

MODEL WIZARD

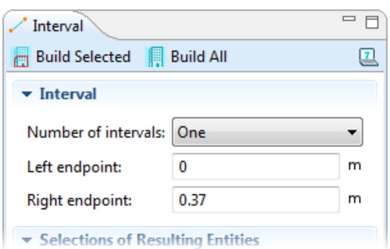
1. Open COMSOL Multiphysics. In the **Model Wizard** window, click the **1D** button. Click **Next**.
2. In the **Add physics** tree, under **Mathematics>Optimization and Sensitivity** double-click **Optimization (opt)** to add it to the **Selected physics** list. Click **Next**.
3. In the **Studies** tree under **Preset Studies**, click **Stationary**.
4. Click **Finish**.

GEOMETRY 1

The 1D geometry is a segment that represents the range of the strain: 0–0.37.

Interval 1

1. In the **Model Builder**, right-click **Geometry 1** and choose **Interval**.
2. Go to the **Interval** settings window. Under **Interval** in the **Right endpoint** field, enter `0.37`.
3. Click the **Build All** button.
4. If needed, click the **Zoom Extents** button on the **Graphics** toolbar.



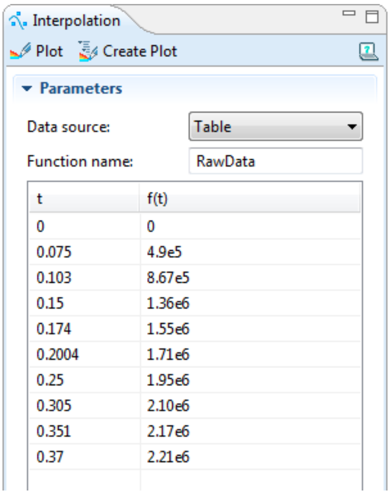
DEFINITIONS

Use the Definitions node features to add an Interpolation function, an Integration model coupling, and define Variables.

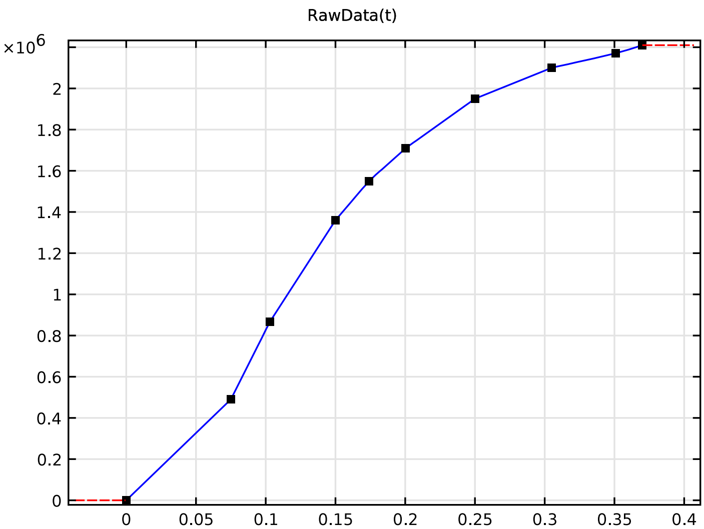
Interpolation 1

1. In the **Model Builder** under **Model 1**, right-click **Definitions** and choose **Functions>Interpolation**.
2. Go to the **Interpolation** settings window. Under **Parameters** in the **Function name** field, enter `RawData`.

   The interpolation function uses measured stress-strain data as the raw data for the curve fitting.
3. In the table, enter the settings as in the figure to the right.
4. Click the **Plot** button.

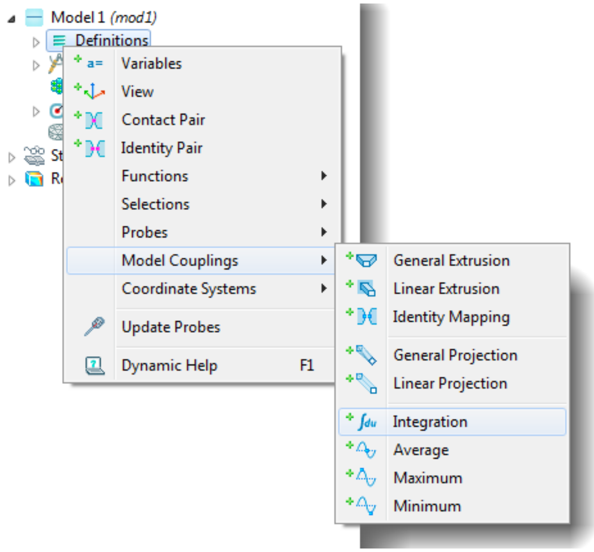


t	f(t)
0	0
0.075	4.9e5
0.103	8.67e5
0.15	1.36e6
0.174	1.55e6
0.2004	1.71e6
0.25	1.95e6
0.305	2.10e6
0.351	2.17e6
0.37	2.21e6



Integration 1

1 In the **Model Builder**, right-click **Definitions** and choose **Model Couplings>Integration**.



2 Go to the **Integration** settings window. Under **Source Selection** from the **Selection** list, select **All domains**.

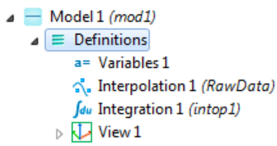
Variables 1

1 In the **Model Builder**, right-click **Definitions** and choose **Variables**.

Define the variables `L` and `P` corresponding to lambda and P, respectively, in the parametric function that models the stress using a Mooney-Rivlin hyperelastic material model. You also define the error as a least-squares error for the difference between the model and the measured data.

2 Go to the **Variables** settings window. Under **Variables** in the table, enter (or copy and paste) the following settings:

NAME	EXPRESSION	DESCRIPTION
`L`	`1+x[1/m]`	Mooney-Rivlin parameter, lambda
`P`	`2*(L-1/L^2)*(C10+C01/L)`	Mooney-Rivlin parameter, P
`SquareDiff`	`intop1((RawData(x[1/m])-P)^2)`	Least-squares error



The node sequence under **Definitions** should match this figure. The unit evaluation shows that the units are undefined because the variables C10 and C01 have not yet been defined.

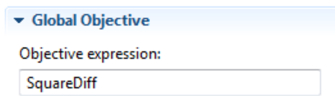
OPTIMIZATION

The global objective is to minimize is the least-squares error between the model and the measured data with the two model parameters C01 and C10 as the control variables.

Global Objective 1

1 In the **Model Builder** under **Model 1**, right-click **Optimization** and choose **Global Objective**.

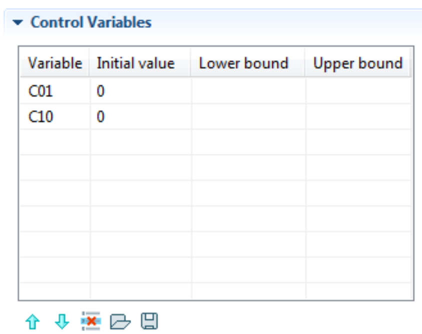
2 Go to the **Global Objective** settings window. Under **Global Objective** in the **Objective expression** field, enter `SquareDiff`.



Global Control Variables 1

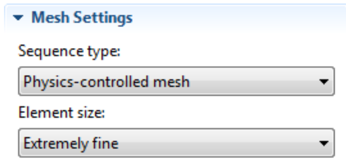
1 In the **Model Builder**, right-click **Optimization** and choose **Global Control Variables**.

2 Go to the **Global Control Variables** settings window. Under **Control Variables** in the table, enter the settings as in the figure.



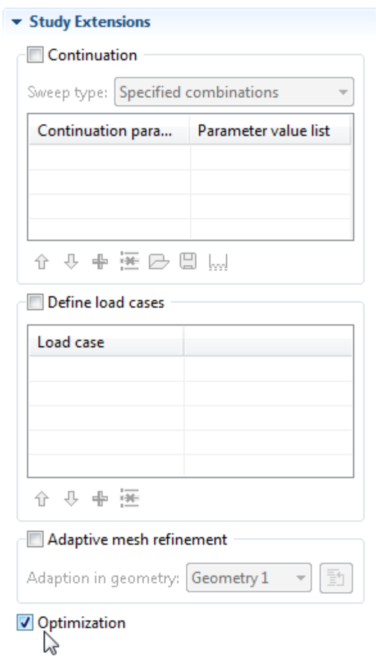
MESH 1

1 In the **Model Builder** click **Mesh 1**.
2 Go to the **Mesh** settings window. Under **Mesh Settings** from the **Element size** list, select **Extremely fine**.
3 Click the **Build All** button in the settings toolbar.



STUDY 1

1 In the **Model Builder**, expand the **Study 1** node and click **Step 1: Stationary**.
2 Go to the **Stationary** settings window. Click to expand the **Study Extensions** section.
3 Select the **Optimization** check box.
4 In the **Model Builder**, right-click **Study 1** and choose **Compute**.



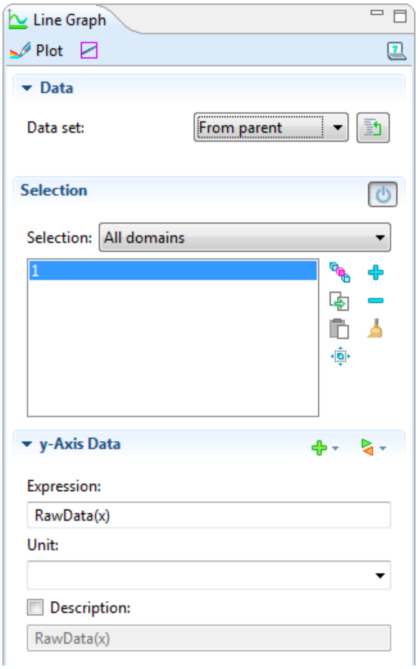
RESULTS

1D Plot Group 1

1 In the **Model Builder**, right-click **Results** and choose **1D Plot Group**. Right-click **1D Plot Group 1** and choose **Line Graph**.

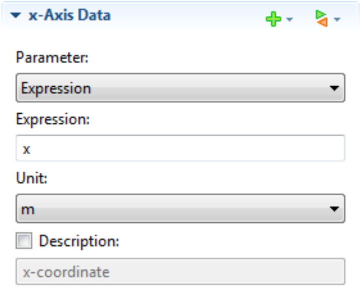
2 Go to the **Line Graph** settings window. Under **Selection** from the **Selection** list, select **All domains**.

3 Under **y-Axis Data** in the **Expression** field, enter `RawData(x)`.



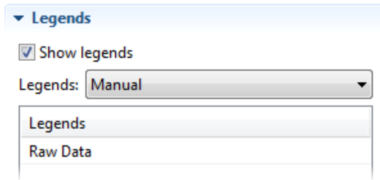
4 In the upper-right corner of the **x-Axis Data** section, click **Replace Expression**. From the menu, choose **Geometry and Mesh>Coordinate>x-coordinate (x)** (or enter x in the field).

5 Click to expand the **Coloring and Style** section. Find the **Line style** subsection. From the **Color** list, select **Blue**.



6 Click to expand the **Legends** section.

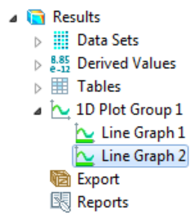
- Select the **Show legends** check box.
- From the **Legends** list, select **Manual**.
- In the table, enter `Raw Data` in the **Legends** table.



7 Click the **Plot** button.

A **Line Graph** with the raw data plotted is displayed in the **Graphics** window. Next add a second Line Graph.

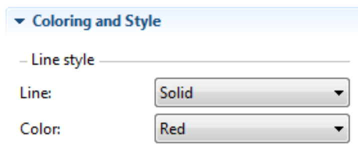
8 Right-click **Line Graph 1** and choose **Duplicate**. A second **Line Graph** node is added to the **Model Builder**.



9 Go to the **Line Graph** settings window. In the upper-right corner of the **y-Axis Data** section, click **Replace Expression**.

10 From the menu, choose **Definitions>Mooney-Rivlin parameter, P (P)** (or enter `P` in the **Expression** field).

11 Under **Coloring and Style** find the **Line style** subsection. From the **Color** list, select **Red**.



12 Under **Legends** in the table, enter `Fitted model` in the **Legends** table.

13 Click the **Plot** button.

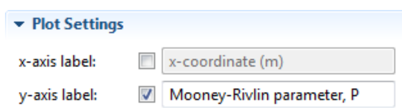
The combined **Line Graph** plots plotting the Raw Data and Fitted Model display in the **Graphics** window.

Finish the plot by adjusting the title, y-axis label, and legend positioning.

1 In the **Model Builder**, click **1D Plot Group 1**.

2 Go to the **1D Plot Group** settings window. Click to expand the **Title** section. From the **Title type** list, select **None**.

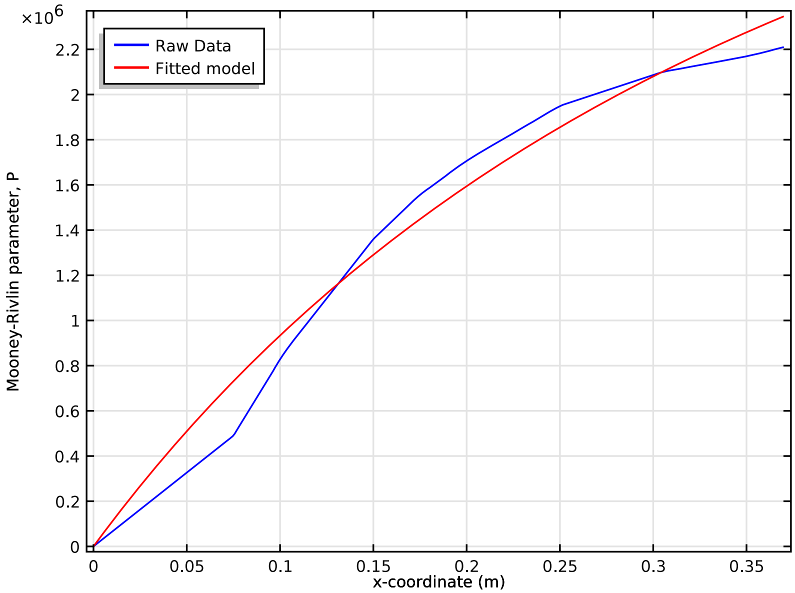
3 Under **Plot Settings** select the **y-axis label** check box. Enter `Mooney-Rivlin parameter, P` in the field.



4 Click the **Plot** button.

5 Click to expand the **Legend** section. From the **Position** list, select **Upper left**.

6 Click the **Plot** button.



DERIVED VALUES

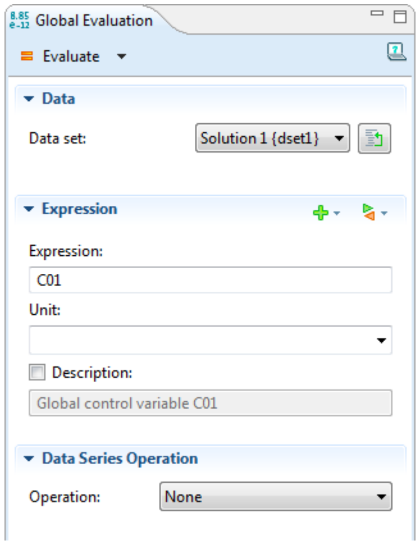
Finally, evaluate the resulting values for the model parameters C01 and C10.

1 In the **Model Builder** under **Results**, right-click **Derived Values** and choose **Global Evaluation**.

2 Go to the **Global Evaluation** settings window. In the upper-right corner of the **Expression** section, click **Replace Expression** .

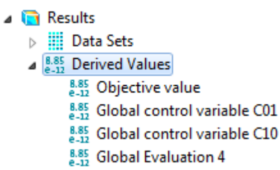
3 From the menu, choose **Optimization>Global control variable C01 (C01)** (or enter C01 in the **Expression** field).

4 Click the **Evaluate** button .



5 Go back to the **Global Evaluation** settings window. In the upper-right corner of the **Expression** section, click **Replace Expression** .

6 From the menu, choose **Optimization>Global control variable C10 (C10)** (or enter C10 in the **Expression** field). Click the **Evaluate** button .



Tutorial Example—Topology Optimization

This tutorial shows you how to use the Optimization Module for topology optimization: Finding the location and number of voids within a structure given a prescribed set of limitations and a clear objective. Topology optimization in a structural mechanics context can answer the question: Given that you know the loads on the structure, distribute the available material while making sure that the stiffness is maximized. Such investigations typically occur during the concept design stages.

This topology optimization example is for an MBB (Messerschmitt-Bölkow-Blohm) beam and uses the SIMPS (solid isotropic material with penalization) method to determine the optimal topology. In this model, this means using a model for Young's modulus E such that $E = \mu(x)^p E_0$, where $0 < \mu(x) \le 1$. The exponent p is a parameter such that $p = 1$ corresponds to ignoring the binary nature of μ whereas higher values of p yields more binary structures. In the model, p is set to 5. The variable for the penalized Young's modulus is called `E_SIMP`.

Defining the Optimization Problem

For the physics, a Solid Mechanics interface represents the structural properties of the beam, and an Optimization interface adds the control variable, objective, and constraints for the optimization problem. The model uses a stationary study of the beam, which consists of a linear elastic material, structural steel. The dimensions of the beam region—6 meters by 1 meter by 0.5 meters—means a total weight of 23,550 kg. In the material, the stress tensor is considered a function of the elasticity tensor and the density. The beam is fixed in the bottom left part and rests on a roller in the bottom right part. An edge load affects the beam in the top middle part. The structure is fully defined by the elasticity tensor E, the density ρ, and the domain, which is fixed, so the design variables are E and ρ. The objective functional for the optimization, which defines the criterion for optimality, is the *strain energy* in this model. The integrand for the strain energy, the strain energy density, is a predefined variable, `solid.Ws`, in the Solid Mechanics interface. The control variable is ρ_{design} (rho_design in the model), which is constrained, using a pointwise constraint, to a value between 10^{-4} and one (1) and, using an integral constraint to be smaller than the area of the design domain (the entire beam geometry) times an area fraction, which is a parameter `area_frac` set to 0.5.

MODEL WIZARD

1 Start a new session of COMSOL or click the **New** button on the main toolbar.
2 In the **Model Wizard** click the **2D** button. Click **Next**.
3 In the **Add Physics** tree under **Structural Mechanics**, double-click **Solid Mechanics (solid)** to add it to the **Selected physics** list.
4 In the **Add Physics** tree under **Mathematics>Optimization and Sensitivity**, double-click **Optimization (opt)** to add it to the **Selected physics** list. Click **Next**.
5 In the **Studies** tree, under **Preset Studies for Selected Physics** click **Stationary**.
6 Click **Finish**.

GLOBAL DEFINITIONS - PARAMETERS

1 In the **Model Builder**, right-click **Global Definitions** and choose **Parameters.**
2 Go to the **Parameters** settings window. In the **Parameters** table, enter (or copy and paste) the following settings:

NAME	EXPRESSION	DESCRIPTION
`F_load`	`50[kN/m]`	Edge load
`area_frac`	`0.5`	Area fraction
`p`	`5`	Exponent in model for Young's modulus
`reg_param`	`100`	Regularization parameter

GEOMETRY 1

Rectangle 1

1 In the **Model Builder** under **Model 1**, right-click **Geometry 1** and choose **Rectangle**.
2 Go to the **Rectangle** settings window. Under **Size** in the **Width** field, enter `6`.
3 Click the **Build Selected** button.

Add Four Point Nodes

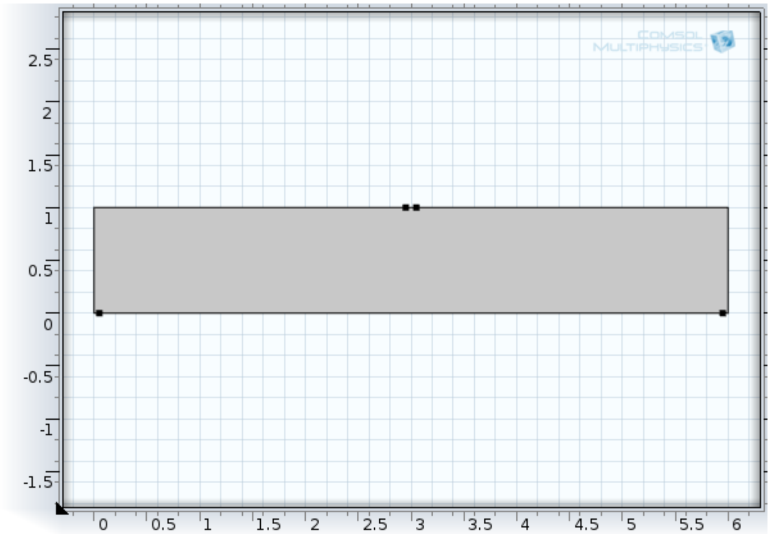
1 In the **Model Builder**, right-click **Geometry 1** and choose **Point**.
2 Go to the **Point** settings window. Under **Point** in the **x** field, enter `0.05`.

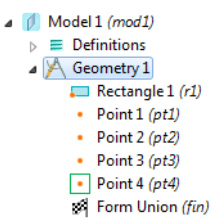
3 Click the **Build Selected** button.

4 Repeat these steps to add three more **Point** nodes to the **Geometry** sequence, with the settings for each node as follows:

- **Point 2**: In the **x** field, enter `2.95`. In the **y** field, enter `1`.
- **Point 3** In the **x** field, enter `3.05`. In the **y** field, enter `1`.
- **Point 4** In the **x** field, enter `5.95`. In the **y** field, leave the default at `0`.

5 Click the **Build All** button.





The **Geometry** sequence in the **Model Builder** should look match the figure.

DEFINITIONS - INTEGRATION 1

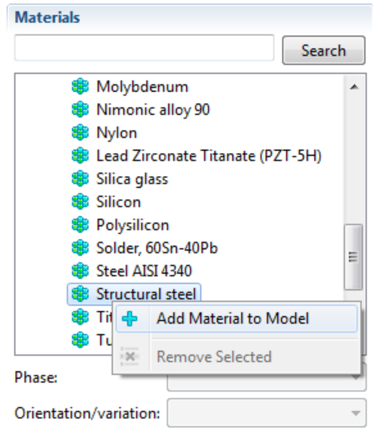
1 In the **Model Builder** under **Model 1**, right-click **Definitions** and choose **Model Couplings>Integration**.

2 Go to the **Integration** settings window. Under **Source Selection** rom the **Selection** list, select **All domains**.

MATERIALS

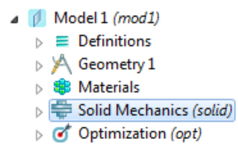
1 Select **View>Material Browser** from the main menu.

2 In the **Material Browser**, under **Built-In**, right-click **Structural steel** and choose **Add Material to Model**.



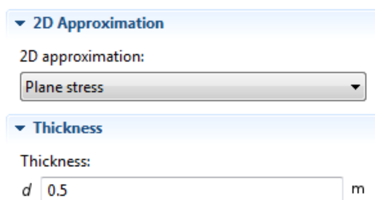
SOLID MECHANICS

1 In the **Model Builder**, click **Solid Mechanics**.



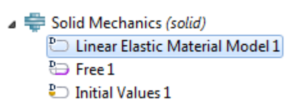
2 Go to the **Solid Mechanics** settings window. Under **2D Approximation** from the **2D approximation** list, select **Plane stress**.

3 Under **Thickness** in the d field, enter `0.5`.

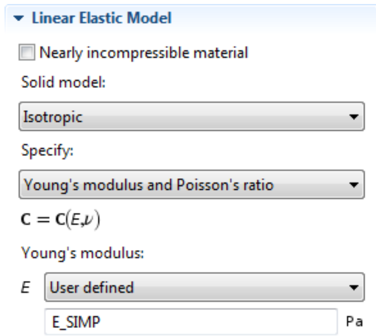


Linear Elastic Material Model 1

1 In the **Model Builder**, expand the **Solid Mechanics** node, then click **Linear Elastic Material 1**.



2. Go to the **Linear Elastic Material** settings window. Under **Linear Elastic Model** from the **Young's modulus** E list, select **User defined**. In the associated field, enter `E_SIMP`.

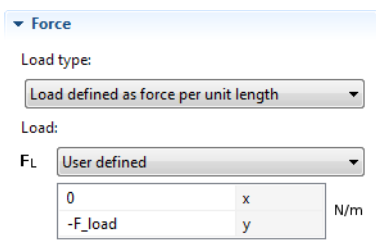


Boundary Load 1

1. In the **Model Builder**, right-click **Solid Mechanics** and from the boundary level, choose **Boundary Load**.
2. Go to the **Boundary Load** settings window. Under **Boundary Selection**, select Boundary 5 only (the short middle segment of the top boundary).

There are many ways to select geometric entities. When you know the geometric entity to add, such as in this exercise, you can click the **Paste Selection** button and enter the information in the **Selection** field. In this example, enter 5 in the **Paste Selection** window. For more information about selecting geometric entities in the **Graphics** window, see the *COMSOL Multiphysics User's Guide*.

3. In the **Boundary Load** settings window, under **Force**:
   - From the **Load type** list, select **Load defined as force per unit length**.
   - Under **Load**, enter values for the $\mathbf{F}_L$ vector in the table as in the figure to the right.

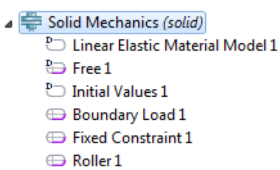


Fixed Constraint 1

1. In the **Model Builder**, right-click **Solid Mechanics** and from the boundary level, choose **Fixed Constraint**. Select Boundary 2 only (the left segment of the lower boundary).

Roller 1

1. In the **Model Builder**, right-click **Solid Mechanics** and from the boundary level, choose **Roller**. Select Boundary 7 only (the right segment of the lower boundary).



The node sequence in the **Model Builder** under **Solid Mechanics** should match the figure.

Control Variable Field 1

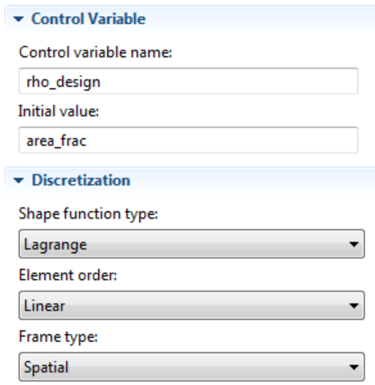
1 In the **Model Builder**, right-click **Optimization** and choose **Control Variable Field**.

2 In the **Control Variable Field** settings window under **Domain Selection** select **All domains** from the **Selection** list.

3 Under the **Control Variable**:

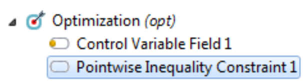
- In the **Control variable name** field, enter `rho_design`.
- In the **Initial value** field, enter `area_frac`.

4 Under **Discretization** from the **Element order** list, select **Linear**. This is to make sure that the changes in the design variable can be sharp.



Pointwise Inequality Constraint 1

1 In the **Model Builder**, right-click **Optimization** and at the domain level choose **Pointwise Inequality Constraint**.

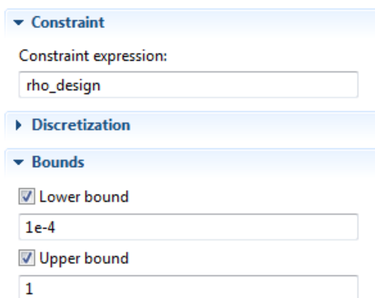


2 In the **Pointwise Inequality Constraint** settings window under **Domain Selection** select **All domains**.

3 Under **Constraint** in the **Constraint expression** field, enter `rho_design`.

4 Under **Bounds** in the:

- **Lower bound** field, enter `1e-4`.
- **Upper bound** field, enter `1`.



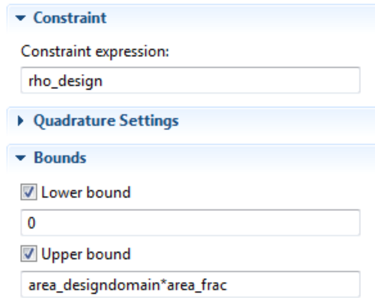
Integral Inequality Constraint 1

1 In the **Model Builder**, right-click **Optimization** and at the domain level choose **Integral Inequality Constraint**.

2 In the **Integral Inequality Constraint** settings window under **Domain Selection** select **All domains**.

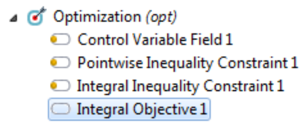
3 Under **Constraint** in the **Constraint expression** field, enter `rho_design`.

4 Under **Bounds** in the **Upper bound** field, enter `area_designdomain*area_frac`.



Integral Objective 1

1 In the **Model Builder**, right-click **Optimization** and at the domain level choose **Integral Objective**.



2 In the **Integral Objective** settings window under **Domain Selection**, select **All domains**.

3 Under **Objective** in the **Objective expression** field, enter `solid.Ws`.

Next, add an integral inequality constraint that provides regularization to avoid a checkerboard pattern in the optimized solution.

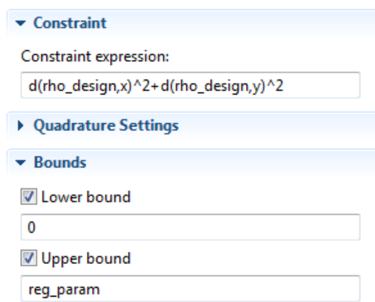
Integral Inequality Constraint 2

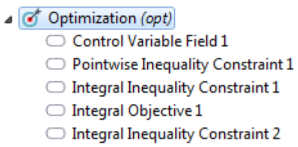
1 In the **Model Builder**, right-click **Optimization** and at the domain level choose **Integral Inequality Constraint**.

2 In the **Integral Inequality Constraint** settings window under **Domain Selection** select **All domains**.

3 Under **Constraint** in the **Constraint expression** field, enter `d(rho_design,x)^2+d(rho_design,y)^2`.

4 Under **Bounds** in the **Upper bound** field, enter `reg_param`.





The final node sequence in the **Model Builder** under **Optimization** should match the figure.

GLOBAL DEFINITIONS - VARIABLES

1. In the **Model Builder**, right-click **Global Definitions** and choose **Variables**.
2. Go to the **Variables** settings window. Under **Variables** in the table, enter the following settings:

NAME	EXPRESSION	DESCRIPTION
`E_SIMP`	`2e11[Pa]*mod1.rho_design^p`	Penalized Young's modulus
`area_designdomain`	`mod1.intop1(1)`	Area of design domain

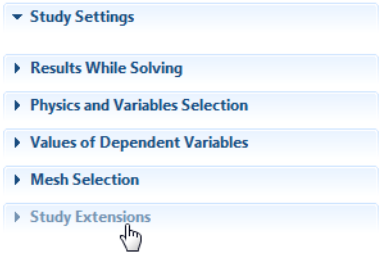
MESH 1

1. In the **Model Builder**, click **Mesh 1**.
2. Go to the **Mesh** settings window. Under **Mesh Settings** from the **Element size** list, select **Extra fine**.
3. Click the **Build All** button.

STUDY 1

1. In the **Model Builder**, expand the **Study 1** node, then click **Step 1: Stationary**.

2 Go to the **Stationary** settings window. Click to expand the **Study Extensions** section.
3 Select the **Optimization** check box.
4 In the **Model Builder**, right-click **Study 1** and choose **Compute** .



RESULTS

To create the plot in Figure 6 plot the `rho_design` variable.

1 Right-click **Results** and select **2D Plot Group** .
2 Right-click **2D Plot Group** and select **Surface** .
3 Click the **Plot** button .

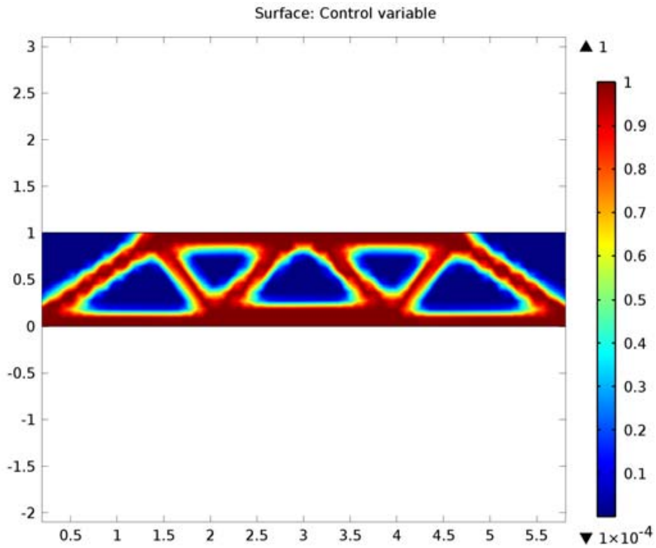


Figure 6: A plot of the control variable ρ_{desig} *for the optimized solution. The resultant design is an approximation of a truss structure.*

As a final step, pick a plot to use as a model thumbnail.

1. In the **Model Builder** under **Results** click **2D Plot Group**.
2. From the **File** menu, choose **Save Model Thumbnail**.

To view the thumbnail image, click the **Root** node and look under the **Model Thumbnail** section. Make adjustments to the image in the **Graphics** window using the toolbar buttons until the image is one that is suitable to your purposes.