INTRODUCTION TO

Optimization Module

6.2

COMSOL

Introduction to the Optimization Module

Version: COMSOL 6.2

Contact Information

Visit the Contact COMSOL page at www.comsol.com/contact to submit general inquiries or search for an address and phone number. You can also visit the Worldwide Sales Offices page at www.comsol.com/contact/offices for address and contact information.

If you need to contact Support, an online request form is located on the COMSOL Access page at www.comsol.com/support/case. Other useful links include:

- Support Center: www.comsol.com/support
- Product Download: www.comsol.com/product-download
- Product Updates: www.comsol.com/product-update
- COMSOL Blog: www.comsol.com/blogs
- Discussion Forum: www.comsol.com/forum
- Events: www.comsol.com/events
- COMSOL Video Gallery: www.comsol.com/videos
- Support Knowledge Base: www.comsol.com/support/knowledgebase

Part number. 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 models that introduce the workflow. The first tutorial solves a curve fitting problem, and the second example contains a topology optimization of a structural beam.

You can use the Optimization Module in combination with essentially any of the other products in the COMSOL product suite. The module 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 such as 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.

Problems that may benefit from the systematic exploratory processes provided by the Optimization Module can be divided broadly into two classes:

- *Design problems* with a single objective.

  Here, the problem is to find the values of control variables or design variables that yield the best performance of a model, quantified by means of an objective function. Problems of this kind arise, for example, in structural optimization, antenna design, and process optimization. In many cases, improving the objective function in a robust way is more important than finding the absolute optimum.

- *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.

Problems of the above types can often be formulated more generally as optimization problems. The Optimization interface, Optimization study step, and Parameter Estimation study step in COMSOL Multiphysics are useful for solving design problems as well as inverse problems and parameter estimation.

Workflow and Examples

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

- For classical optimization, not involving any physics, add a Stationary study and an Optimization study step to an empty model. Define parameters and help variables under Global Definitions, then set up an objective function, control variables, bounds, and constraints in the Optimization study step. Constraints and objectives can be written as explicit functions of the control variables.
- For PDE constrained optimization, first create the *forward model* containing the geometry and the physics. Define parameters under Global Definitions, or by adding control variables to an Optimization interface. Make sure the forward model solves properly for some reasonable values of the control variables before proceeding with defining the objective function and constraints, and then finally solve the optimization problem.

Note that if your optimization problem only requires global scalar control variables, objective functions and constraint expressions, you can set everything up directly in the Optimization study step. The Optimization interface is only needed if constraints must be applied at each mesh node individually, or if the objective function is of a nonglobal least-squares type.

Examples using the Optimization Module exist in the application library for the Optimization Module, as well as in the Application Libraries for several other modules. Search for *optimization* in the Application Libraries window search field to find models that use the Optimization Module in the products that your license includes. The models illustrate the use of the Optimization interfaces

- The General Optimization interface
- The Topology Optimization interface
- The Shape Optimization interface
- The Parameter Estimation interface

The interfaces can be used across the different physics, so the absence of an example for particular combination of interface and physics does not indicate incompatibility, but this can be the case. In example, the shape and topology optimization interfaces rely heavily on gradient-based optimization and this is currently unsupported for explicit time stepping.

An Overview of Optimization Terminology

Assuming that you have solved a COMSOL Multiphysics model of a physical design, it is common to ascribe to it some figure of merit, a cost function, or —

using a more general term — an *objective function*. This objective function can quantify almost anything about the performance of the system.

It is natural to question how this objective function changes when you vary some parameters or *control variables*. These variables or parameters can control any part of the design: dimensions, loads, boundary conditions, material properties, material distribution, and so forth. The control variables often have a set of *bounds* and *design constraints* associated with them: dimensions must be within some limits, geometric features must not come to close to each other, only certain materials are possible, and so on.

When optimizing a system modeled as a Multiphysics model, there may also be *performance constraints* present, which depend on the model solution. For example, when minimizing the weight of a part, it is often important to keep the maximum stress or temperature below a predefined safety limit.

In general, an *optimization problem* is when you want to improve the *objective function* by varying the *control variables* within some set of *constraints*. Optimization problems can be further categorized as *topology*, *size*, *shape*, and *parameter* optimization.

Optimization Algorithms

The Optimization Module contains the framework and functionality needed to perform*gradient-based* and*gradient-free* optimization on an existing COMSOL Multiphysics model.

There are four optimization algorithms for gradient-based optimization available in the 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, frequency-, or time-dependent, analytic sensitivities of the objective function with respect to the control variables can be used.

The second algorithm is the IPOPT solver, which has similar characteristics as SNOPT. The main authors are Andreas Waechter and Carl Laird from IBM, but the code is still developed as an open-source project with many contributors.

The third algorithm is the MMA solver, which is based on the method of moving asymptotes by Krister Svanberg of KTH (Royal Institute of Technology, Stockholm). The MMA solver can handle objective functions and constraints of the same very general form as SNOPT. It can solve min-max optimization problems, and it is well suited to handle problems with a large number of control variables, such as topology optimization. The globally convergent version of the solver is often referred to as GCMMA. This is the default, but the original version of the MMA algorithm is also available. Both MMA and GCMMA have support for limiting the step length via move limits.

The fourth algorithm is a Levenberg–Marquardt solver. When this solver is used, the objective function must be of least-squares type. Also, constraints are not supported. Since the Levenberg-Marquardt method is derived to solve problems of least-squares type, it typically converges faster than SNOPT, IPOPT, and MMA for such problems. The solver can compute confidence intervals for the control variables, but this assumes that the measurement errors are small and follow a normal distribution.

In addition, the Optimization Module provides a number of gradient-free (derivative-free) optimization algorithms. Currently Nelder–Mead, BOBYQA, COBYLA, and a coordinate search are supported. These methods can optimize a model with respect to design parameters (model parameters) including those that control the geometry sequence that defines the model geometry. Coordinate search, Nelder–Mead, and COBYLA support strict enforcement of constraints that only depend on the control variables (design constraint).

All optimization solvers are accessible from the same Optimization study step, which contains the ordinary solver sequence over which the optimization method iterates. The gradient-free methods can contain any other study sequence, including Study Reference links, which make it possible to evaluate different objective function contributions and constraints in different studies. The gradient-based methods are limited to optimizing over a single study step for types supporting the computation of analytic sensitivities: currently Stationary, Time Dependent, and Frequency Domain studies.

Optimization Module Applications

The Optimization Module provides versatile tools for optimization, curve fitting, parameter estimation, and inverse modeling. Optimization can involve solving 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 1 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 application 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.

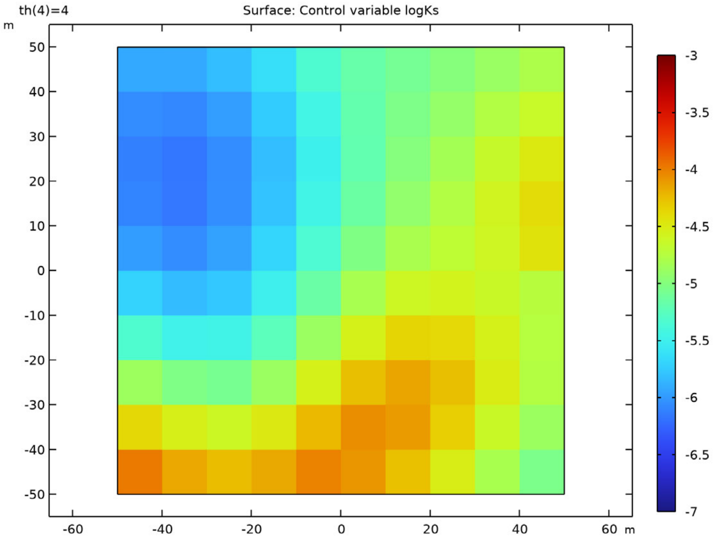


Figure 1: Hydraulic conductivity obtained by inverse modeling in the Aquifer Characterization model.

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 2 shows the *x*-component of the velocity field for the optimal solution and the distribution of the filling material in the open channel:

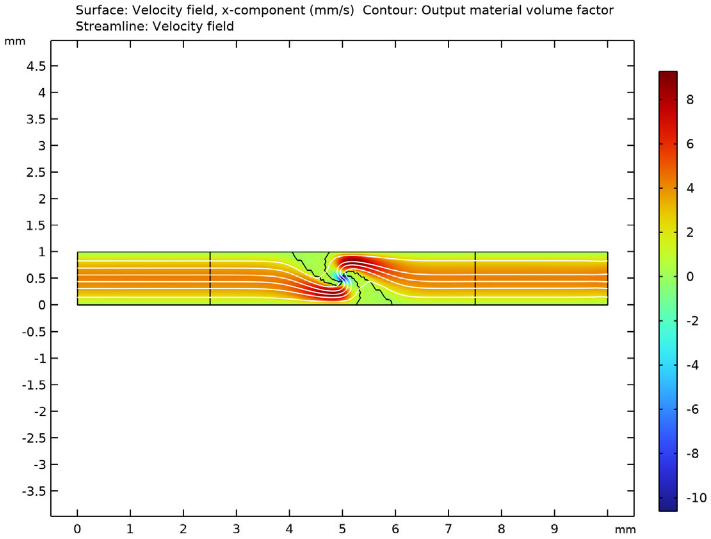


Figure 2: The horizontal velocity (surface plot) and velocity field (streamlines) after optimization. In addition, the contour γ *= 0.5 indicates the border between open channel and filling material.*

This model is in the Optimization Module Application 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.

About Parameter Estimation

You can use the Optimization Module to perform parameter estimation: to determine input values so that the output matches some experimental data — that is, to solve an *inverse problem*. The first tutorial (Tutorial Model — Curve Fitting) describes how to use the Optimization Module for parameter estimation. When solving an inverse problem, there is some set of experimental data that can be compared to the output of the model. Parameter estimation is used to adjust model inputs or properties 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 often ill-posed in the sense that they have multiple locally optimal solutions where the optimization algorithm may get stuck, or entire subspaces of equal merit may exist where the algorithm wanders around forever. In addition, the particular local optimum that is found depends strongly on the initial guess and on small perturbations in the experimental data. For these reasons, it is best to approach inverse problems with a good understanding of the underlying physics and mathematics of the problem. In particular, some problem-dependent regularization strategy is usually needed to make the optimization process robust.

Consider a COMSOL Multiphysics 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 using a Mooney–Rivlin model, which requires empirically determined material constants to describe the structural deformation. Assuming that there are experimental results relating load and deformation, as shown in Figure 3, it is possible to use parameter estimation to determine the material model constants that best fit this data. The curve shown in Figure 3 is computed based on the results of a parameter estimation; it is a least-squares fit of a COMSOL Multiphysics model with Mooney–Rivlin material parameters to the experimental data.

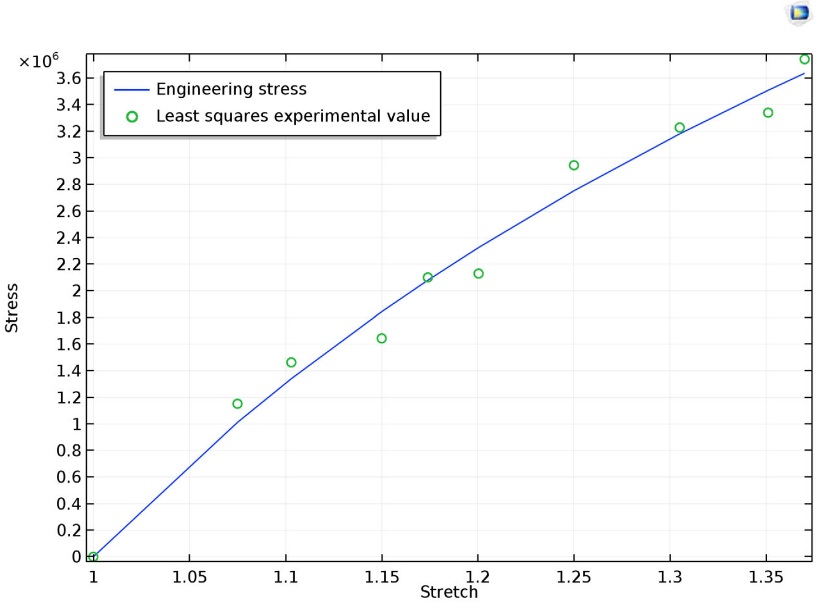


Figure 3: Experimental data and a best fit curve of the Mooney–Rivlin material model based upon a finite element model of a test sample.

About Topology Optimization

The second tutorial model in this guide (Tutorial Model — Topology Optimization) shows how to solve a model using topology optimization.

Consider a COMSOL Multiphysics model of a physical system, such as the bracket shown in Figure 4. 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). A closely related problem which is often easier to solve can be stated as: Minimize the compliance (the deflection) of the structure under the condition that the total mass must be reduced by at least some predefined factor.

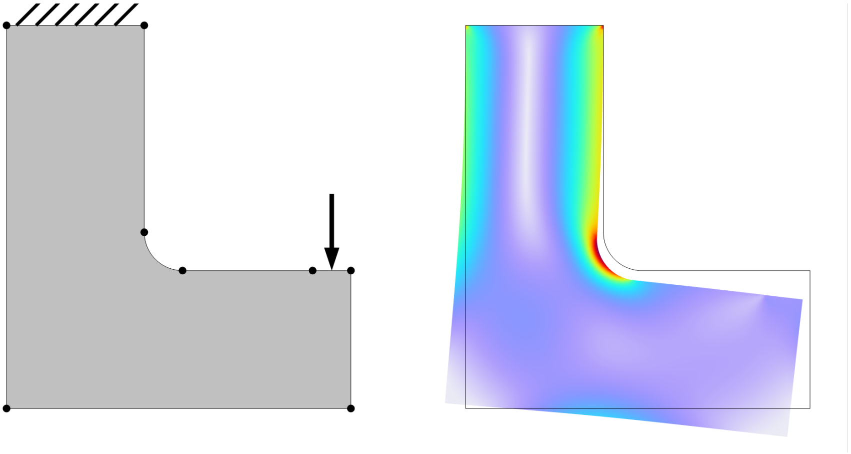


Figure 4: A simple bracket design, and the solved finite element model showing stresses and deflections. This design has room for improvement.

One way to address this is to use *topology optimization*. Topology optimization treats the material distribution as the design variable. Design variables of this type are treated in the same way as the solution variables; they are interpolated using shape functions on the finite element mesh. Therefore, there is one design variable per mesh node or per mesh element, associated with a position in the geometry. We call this a *control variable field*.

The Optimization Module computes the *derivative*, or the *sensitivity*, of the objective function with respect to these discrete design variables. For topology optimization, the *adjoint method* is required, because all other methods are too slow. The optimization solver uses the sensitivity information to manipulate the design variables, within the limit of any user-defined constraints, in the direction of improved objective function values.

This is an iterative procedure, as illustrated in Figure 5. Starting with the initial design, the solver adjusts the control variables representing a fictitious material density in each element. This is repeated as long as some improvement in the objective function is possible and the constraints are not violated.

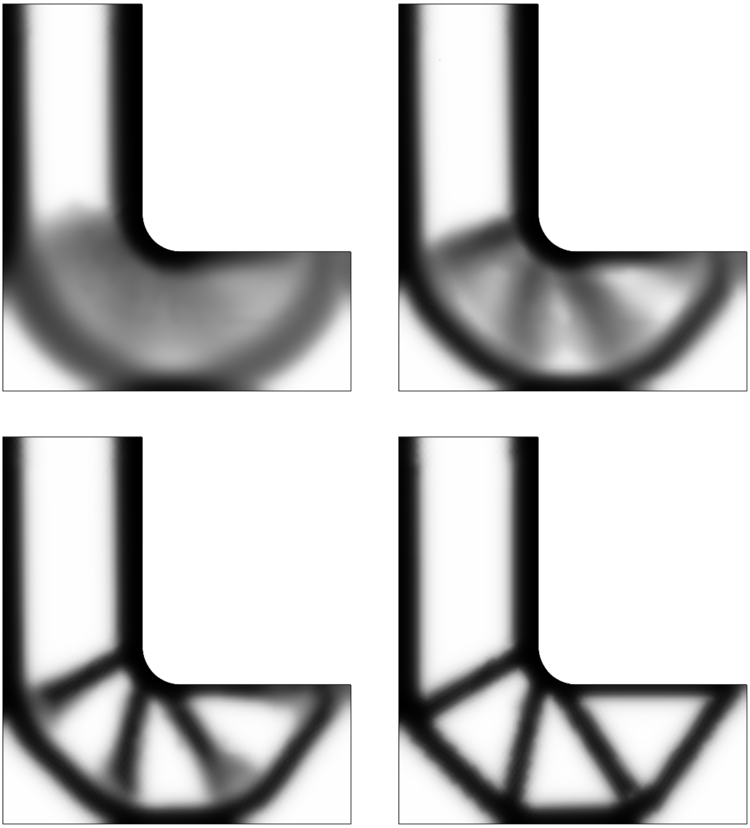


Figure 5: Topology optimization of the bracket, the material distribution, as defined by the finite element mesh, is modified to minimize the weight while satisfying a constraint on peak displacement. The design evolution is shown after 8, 12, 16, and 60 iterations.

A structural engineer will recognize that the resulting design is essentially 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 resulting design does have some dependency on the mesh used. There are techniques known as *regularization* that you can use to avoid this and the topology model example in this guide includes regularization. Second, it is difficult to incorporate geometric constraints into the optimization algorithm and this can lead to impractical designs, but COMSOL does support the use of milling constraint for topology optimization. This can be used to reduce the design freedom, so that more practical designs are found. In any case, topology optimization should be used early in the design process as a guide for the overall structure of the system. The method 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.

Optimization Module Functionality

The Optimization Module is a general purpose add-on to COMSOL Multiphysics and its other modules. It consists of a physics interface, special study steps, solvers, a density topology feature, and an application library.

The Optimization Interface

The Optimization Module adds the Optimization interface ( )to the list of available physics interfaces. The interface can be found under the Mathematics>Optimization and Sensitivity branch when you are adding physics to your model.

The Optimization interface includes a wide range of advanced 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)

The Optimization Study Step

The Optimization study step ( ) acts as the central control panel for any optimization problem. It can be added to any study, where it will appear as the first step. The remaining steps in the study define the simulation that is being optimized. Note that when a gradient-based optimizer is used, the Optimization study step replaces and cannot be used together with some other study steps, notably Parametric Sweep and its variants.

In the Optimization study step, you can set up objective function contributions, select control variables, and set constraints. Objectives, constraints, and control variables defined in an Optimization interface are displayed in the study step, where they can be disabled or enabled. Note that most optimization tasks can be specified using the Optimization study step alone, needing no Optimization Interface.

The Optimization study step also contains an optimization solver choice, including the most important solver settings, as well as extensive settings for controlling graphics and table output during the solution process.

The Parameter Estimation Study Step

The Parameter Estimation study step () is a specialized version of the Optimization study step, intended to simplify some typical parameter estimation tasks. In particular, it contains a user interface for specifying a least-squares objective function based on time-dependent measurements entered or imported into an interpolation function.

The Shape and Topology Optimization Study Steps

Both the Shape Optimization study step () and the Topology Optimization study step () are stripped-down versions of the Optimization Study step. They only allow gradient-based optimization, and they are incompatible with objectives and constraints in the Optimization interface. They both default to the MMA optimization solver, and they allow limiting the absolute change of individual control variables using a move limit (only for MMA). The move limit is enabled by default and set to 0.1 for the Shape Optimization study step in order to reduce the likelihood of problems with inverted elements.

Topology Optimization with the Density Method

The Density Method is a topology optimization feature located under component definitions. The method allows the user to impose a Helmholtz filter with a custom radius, such that a minimum length scale can be imposed.

The user has to couple the physical material properties with the output of the filter manually, but the feature supports projection and common interpolation functions, so this typically only involves a multiplication.

The Optimization Solvers

The Optimization Module contains the following optimization solvers:

SNOPT: A robust, gradient-based, general-purpose, sequential quadratic programming algorithm for nonlinear constrained problems.

IPOPT: A gradient-based, general-purpose interior point algorithm for nonlinear constrained problems.

MMA: A general-purpose, interior point method based on successive convex approximations constructed from objective and constraint gradient information, particularly suited to topology optimization.

Levenberg–Marquardt: A special-purpose solver for least-squares fitting, using the special problem structure to estimate second-order derivatives from first order gradient data. Constraints are ignored.

Nelder–Mead: A robust derivative-free, heuristic, simplex search algorithm, including a penalty method for constraint handling.

BOBYQA: An algorithm sampling the objective function to construct and maintain a quadratic approximation of the objective inside a moving trust region. General constraints can be handled in an augmented Lagrangian wrapper, solving a sequence of unconstrained problems.

COBYLA: An algorithm sampling both objective function and constraint values to create and maintain a linear approximation inside a moving trust region. General constraints are accepted but may be violated at intermediate solution steps.

Coordinate search: A method which searches for the optimum by successive sampling along the control variable axes, typically changing one control variable at a time.

Tutorial Model — Curve Fitting

This tutorial model demonstrates how to use the Optimization Module to estimate unknown function parameters based on measured data. The two-parameter Mooney–Rivlin solid material model is used as an example, but the procedure is generally applicable when you need to fit a parameterized analytic function to measured data.

The two-parameter incompressible Mooney–Rivlin material model describes the local behavior of rubber-like materials. The model assumes that the local strain energy density in an incompressible solid is a simple function of local strain invariants.

In a standard tensile test, a rotationally symmetric test specimen is pulled in such a way that it extends in one direction and contracts symmetrically in the other two. For this case of uniaxial extension, the relationship between an applied force, F, and the resulting extension, ΔL, of a true Mooney–Rivlin material is

$$\frac{F}{A_0} = 2\left(C_{10} + C_{01}\frac{L_0}{L_0 + \Delta L}\right)\left(\frac{L_0 + \Delta L}{L_0} - \left(\frac{L_0}{L_0 + \Delta L}\right)^2\right)$$

where A_0 is the original cross-sectional area of the test specimen and L_0 is its reference length. The constants C_{10} and C_{01} are material parameters that must be determined by fitting the above equation to the experimental data from the tensile test.

In practice, tensile test data is delivered in a form which is independent of the geometry of the test specimen used. There are multiple possible formats. The one used here contains corresponding measured values of engineering stress, P_i, representing force per unit reference area

$$P = \frac{F}{A_0}$$

and stretch, λ_i, representing relative elongation

$$\lambda = \frac{L_0 + \Delta L}{L_0}$$

The expected relationship between these variables for a Mooney–Rivlin material is

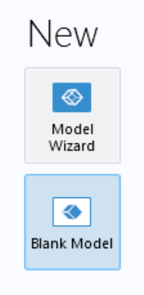
$$P(\lambda) = 2\left(C_{10} + \frac{C_{01}}{\lambda}\right)\left(\lambda - \frac{1}{\lambda^2}\right)$$

Given N pairs of measurements (λ_i,P_i), $i = 1,\ldots,N$, the values of C_{10} and C_{01} that best fit the measured data are considered to be those which minimize the total squared error

$$e = \sum_{i=1}^{N} (P(\lambda_i) - P_i)^2$$

The curve fitting problem is therefore identical to an optimization problem.

Creating a New Model



When starting COMSOL Multiphysics, you are normally greeted by the New window. Otherwise, from the File menu, choose New.

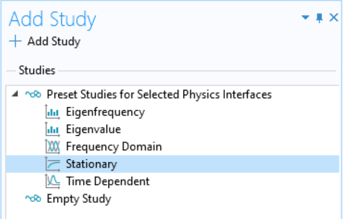
Note: These instructions are for the user interface on Windows, but apply with minor differences also to Linux and Mac. In particular, the Home toolbar refers to the fixed set of controls near the top of the Desktop

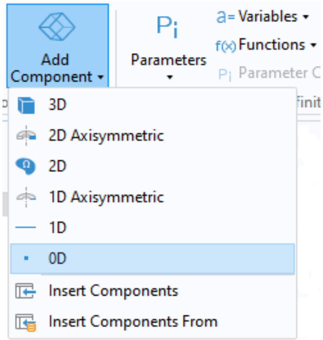
- In the New window click the Blank Model button .

Add a Component

This pure optimization example does not refer to any geometry. Any model must, however, contain at least one component. Zero-dimensional components are useful for this type of optimization problem, as well as, for physics modeled using only ordinary differential equations (ODEs). The Stationary study is necessary as a placeholder signaling the problem type, even if there are no equations to solve in this case.

1. In the Home toolbar, click the Add Component button and select 0D ·.
2. In the Study toolbar, click Add Study and a Stationary study step.





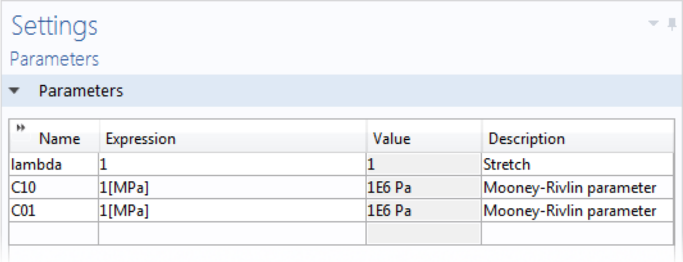
Global Definitions

When your control variables are independent of geometry, it is easiest to set them up initially as global parameters. Also add the stretch parameter `lambda` that has been varied in the measured data as a global parameter.

Parameters

Note that you can click the double-arrow button in the upper-left corner of parameter and variable tables to expand them, making editing easier.

1. In the Home toolbar click Parameters Pi .
2. In the Settings window locate the Parameters section.
3. In the table, enter the following settings:

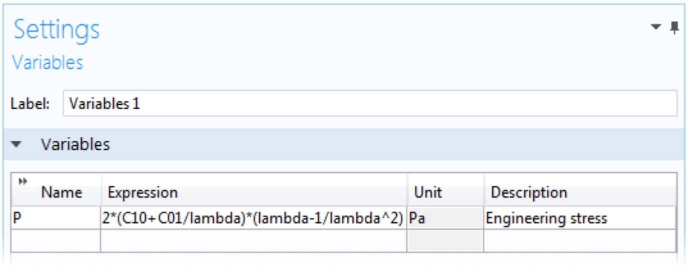


Name	Expression	Value	Description
lambda	1	1	Stretch
C10	1[MPa]	1E6 Pa	Mooney-Rivlin parameter
C01	1[MPa]	1E6 Pa	Mooney-Rivlin parameter

Variables 1

1. In the Home toolbar click Variables a= and choose Global Variables a= .
2. In the Settings window for Variables locate the Variables section.

3 In the table, enter the following settings:



Settings

Variables

Label: Variables 1

Variables

Name	Expression	Unit	Description
P	2*(C10+C01/lambda)*(lambda-1/lambda^2)	Pa	Engineering stress

Optimization

You can import tensile test data from file in the form of comma-separated values or, as demonstrated here.

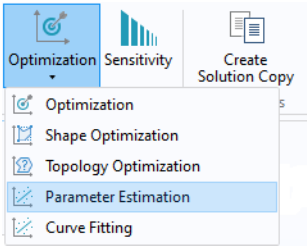
Parameter Estimation

1 In the Study toolbar, click the Optimization button and choose Parameter Estimation.

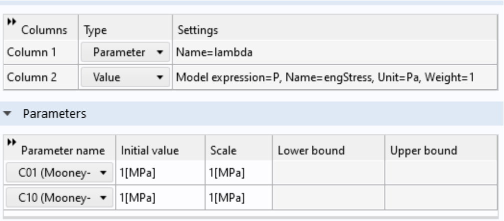
2 In the Parameter Estimation study step, locate the Data Source section and input `curve_fit_mooney_rivlin.csv` as the filename.

3 Setup the first column to be a Parameter column for the `lambda` parameter and the second column to be Value column for the `P` variable.

4 Locate the Control Variables and Parameters section and click Add + twice.



5 In the table, enter the following settings:



Columns	Type	Settings
Column 1	Parameter	Name=lambda
Column 2	Value	Model expression=P, Name=engStress, Unit=Pa, Weight=1

Parameters

Parameter name	Initial value	Scale	Lower bound	Upper bound
C01 (Mooney-	1[MPa]	1[MPa]		
C10 (Mooney-	1[MPa]	1[MPa]		

6 Locate the Parameter Estimation method section and change the method to Levenberg–Marquardt.

7Change the Least-squares time/parameter method to From least-squares objective.

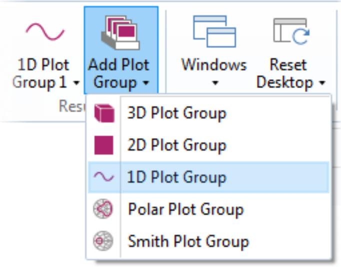
8Click the Compute = button.

Results

Add a plot of the least-squares fitted stress-strain curve together with the measured data.

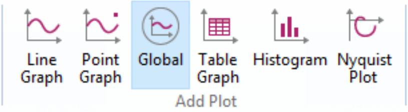
1D Plot Group 1

In the Home toolbar, click Add Plot Group and choose 1D Plot Group .



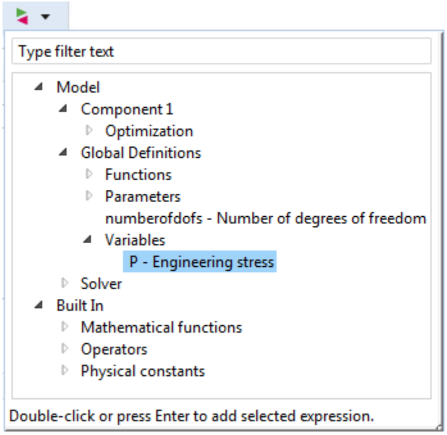
Global 1

1 On the 1D plot group toolbar, click Global .



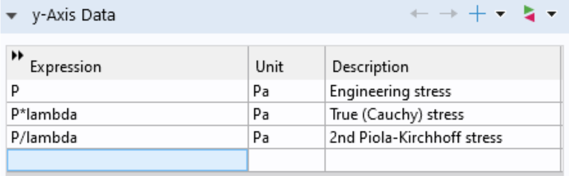
2 In the Settings window for Global, locate the y-axis data section.

3 Click Replace Expression in the upper-right corner of the y-axis data section. From the menu, choose Model>Global Definitions>Variables> P Engineering stress.



Double-click or press Enter to add selected expression.

4 Add two more expressions to plot the true (Cauchy) stress as well as the 2nd Piola–Kirchhoff stress as shown below:
stress.



y-Axis Data

Expression	Unit	Description
P	Pa	Engineering stress
P*lambda	Pa	True (Cauchy) stress
P/lambda	Pa	2nd Piola-Kirchhoff stress

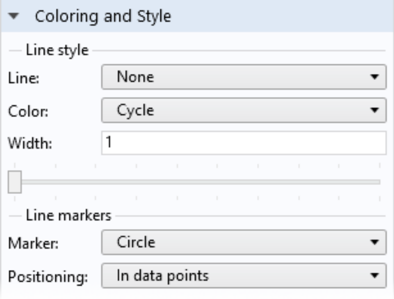
Global 2

1 On the 1D Plot Group 1 toolbar, click Global.

2 In the Settings window for Global, locate the y-axis data section.

3 Click Replace Expression in the upper-right corner of the y-axis data section. From the menu, choose Model>Solver>Parameter Estimation> opt.glsobj.engStress.data - Least squares experimental value.

4 Click to expand the Coloring and style section.



Coloring and Style

Line style

Line: None

Color: Cycle

Width: 1

Line markers

Marker: Circle

Positioning: In data points

5 Find the Line style subsection. From the Line list, choose None.

6 Find the Line markers subsection. From the Marker list, choose Circle.

7 From the Positioning list, choose In data points.

8 On the 1D plot group toolbar, click Plot .

Finish the plot by adjusting the title, axis labels, and legend positioning.

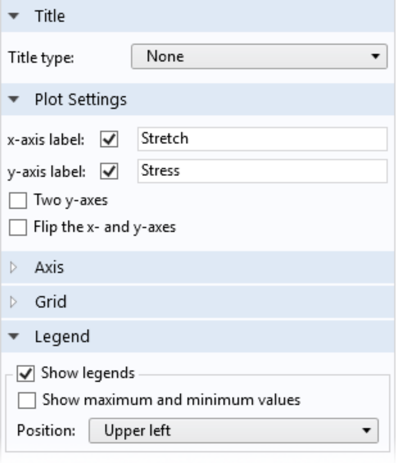
9 In the Model Builder click 1D Plot Group 1 .

10 In the Settings window for 1D Plot Group 1 click to expand the Title section. From the Title type list, choose None.

11 Locate the Plot Settings section.

- Select the x-axis label check box.
- In the `x-axis` label text field, type `Stretch`.
- Select the y-axis label check box.
- In the y-axis label text field, type `Stress`.

12 Click to expand the Legend section. From the Position list, choose Upper left.



▼ Title

Title type: None

▼ Plot Settings

x-axis label: ☑ Stretch

y-axis label: ☑ Stress

☐ Two y-axes

☐ Flip the x- and y-axes

▷ Axis

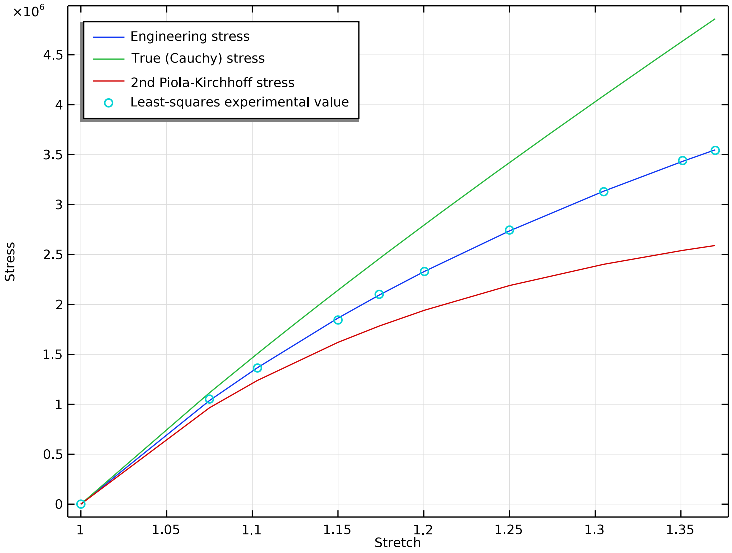
▷ Grid

▼ Legend

☑ Show legends

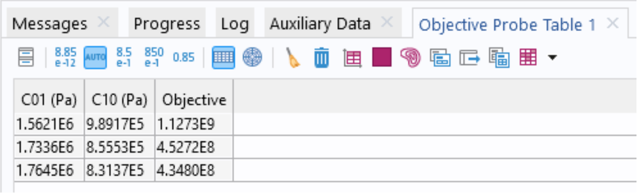
☐ Show maximum and minimum values

Position: Upper left



Objective Probe Table 1

You can see the values of the optimized parameters in the last row of the automatically created Objective Probe table.



C01 (Pa)	C10 (Pa)	Objective
1.5621E6	9.8917E5	1.1273E9
1.7336E6	8.5553E5	4.5272E8
1.7645E6	8.3137E5	4.3480E8

Tutorial Model — Topology Optimization

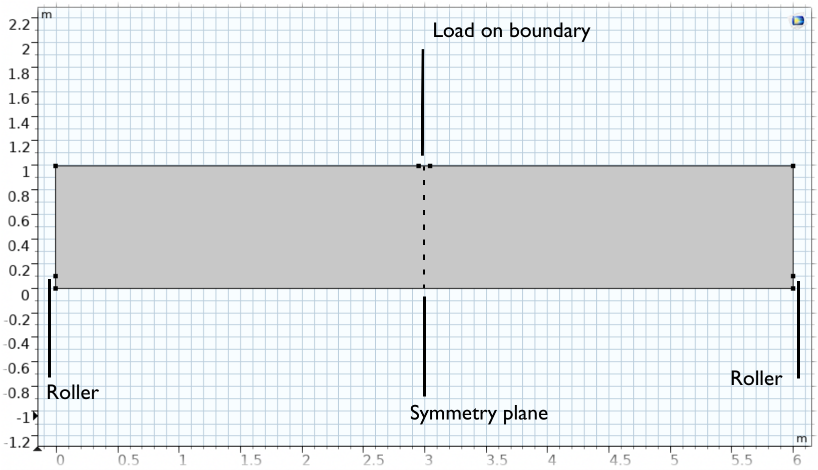
Topology optimization in a structural mechanics context can answer the question: Given that you know the loads on the structure, which distribution of the available material maximizes stiffness? Or, conversely, how much material is necessary to obtain a predefined stiffness, and how should it be distributed? Such investigations typically occur during the concept design stages.

The conflicting goals of stiffness maximization and mass minimization lead to a continuum of possible optimal solutions, depending on how you balance the goals against each other. This topology optimization example demonstrates how to use a penalization method (SIMP) to obtain an optimal distribution of a fixed amount of material such that stiffness is maximized. Changing the amount of material available leads to a different solution that is also Pareto optimal, representing a different balance between the conflicting objectives.

The geometry and load pattern are similar to the Messerschmitt–Bölkow–Blohm beam (MBB) commonly used as a validation test for topology optimization.

Model Definition

The model studies optimal material distribution in 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 an original total weight of 23,550 kg. The beam is symmetric about the plane $x = 3$. Both ends rest on rollers while an edge load acts on the top middle part.



The objective functional for the optimization, which defines the criterion for optimality, is the total *strain energy*. Note that the strain energy exactly balances the work done by the applied load, so minimizing the strain energy minimizes the displacement induced at the points where load is applied, effectively minimizing the compliance of the structure — maximizing its stiffness. The other, conflicting, objective is minimization of total mass, which is implemented as an upper bound on the mass of the optimized structure.

Another condition that is more difficult to enforce is the requirement that the optimized topology is truly binary — any given point is either solid or void, nothing in between — but still without excessively fine details or “checkerboard” patterns. The SIMP penalization method accomplishes the first part. In the SIMP model, the stress tensor is considered to be a function of the true Young’s modulus E_0 and the artificial density, θ_c, which acts as control variable in the optimization problem:

$$\begin{aligned}\theta_p &= \theta_{\min} + (1-\theta_{\min})\theta(\mathbf{x})^p \\ E(\mathbf{x}) &= E_0\theta_p\end{aligned}$$

Since the stiffness must, for numerical reasons, not vanish completely anywhere in the model, the density parameter is constrained such that $\theta_{\min} \leq \rho \leq 1$. The exponent $p \geq 1$ is a penalty factor that makes intermediate densities provide less stiffness compared to what they cost in weight.

When there is an overall limit on the fraction $V_{\mathrm{frac}} = 0.5$ of the domain area, A, occupied by solid material

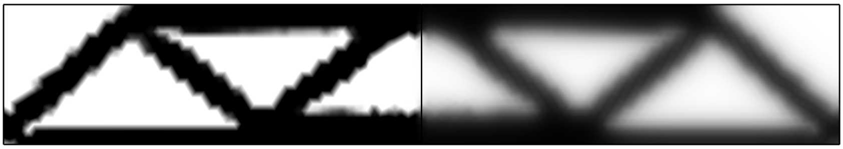
$$0 \leq \theta_{\mathrm{avg}} = \int_{\Omega} \theta(\mathbf{x}) d\Omega \leq V_{\mathrm{frac}}$$

the SIMP penalization forces θ_c toward either of its bounds. Increasing p therefore leads to a sharper solution.

At the same time, the solution must not contain too much fine detail, since that would compromise the reliability of the final design. Also, it is desirable to make the solution as independent of the mesh resolution as possible. To accomplish this, some kind of regularization is needed. There are many possibilities. One of the easiest to implement is based on a combination of successive mesh refinements and minimum length scale imposed via a Helmholtz filter.

$$\theta_f = R_{\min}^2 \nabla^2 \theta_f + \theta_c$$

The equation essentially just smears out the design as illustrated below with and θ_c and θ_f to the left and right, respectively.

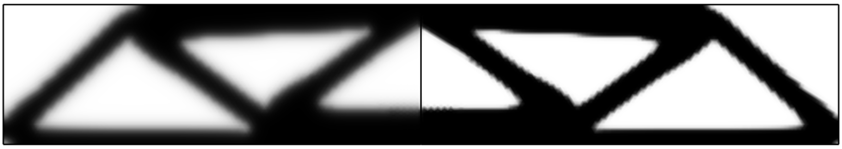


The material volume factor (right) has a lot of gray scale.

This filtered design variable can then be projected to reduce the extent of the gray scale region. COMSOL Multiphysics supports projection based on the hyperbolic tangent function (see Ref. 1)

$$\theta = \frac{(\tanh(\beta(\theta_f - \theta_\beta)) + \tanh(\beta\theta_\beta))}{(\tanh(\beta(1-\theta_\beta)) + \tanh(\beta\theta_\beta))}$$

This is illustrated for $\theta_\beta = 0.5$and $\beta = 8$ below with θ_f and θ to the left and right.

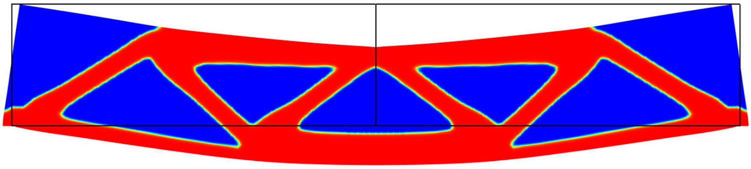


The (projected) material volume factor shown to the right has no visible gray scale.

High values of β are associated with strong projection and less gray scale, but also slows down the optimization process, so initially we will not use it and thus set $\theta = \theta_f$.

Results

The following plot shows the stiffness distribution for the optimized solution. The resulting design is an approximation of a truss structure, which is expected for this beam size. Note that the optimal design for a longer or shorter beam may be quite different, and also that there may exist stiffer solutions using the same amount of material. Such solutions typically consist of thinner members that have been eliminated here by the regularization and refinement procedure.



Notes About the COMSOL Implementation

A Solid Mechanics interface represents the structural properties of the beam, while an Optimization interface allows adding the control variable, objective, and constraints for the optimization problem. The total internal strain energy is a predefined variable, `solid.Ws_tot`, readily available to use as the objective function for the optimization problem.

Only the left half of the beam geometry must be modeled, because of the assumed symmetry, which is implemented as a Roller condition on the symmetry plane. Plots for postprocessing and inspection of the solution while solving are set up using a Mirror dataset, to show the complete solution. The geometry is parameterized, making it easy to experiment with different beam sizes.

The optimization solver is selected and controlled from a Topology Optimization study step. For topology optimization, either the MMA, IPOPT, or the SNOPT solver can be used. They have different merits and weaknesses. MMA tends to be braver in the beginning, proceeding quickly toward an approximate optimum. SNOPT and IPOPT are more cautious, but converges more efficiently close to the final solution due to its second-order approximation of the objective.

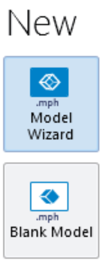
This example demonstrates a strategy based on using MMA with a limited number of iterations on successively finer meshes. It quickly and reliably produces trustworthy topologies showing good improvement of the objective function value. These solutions are not necessarily globally optimal, which may, however, be of less importance in practice.

Reference

1. F. Wang, B.S. Lazarov, and O. Sigmund, “On projection methods, convergence and robust formulations in topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 43, pp. 767–784, 2011.

Creating a New Model

When starting COMSOL Multiphysics, you are normally greeted by the New window. Otherwise, from the File menu, choose New.



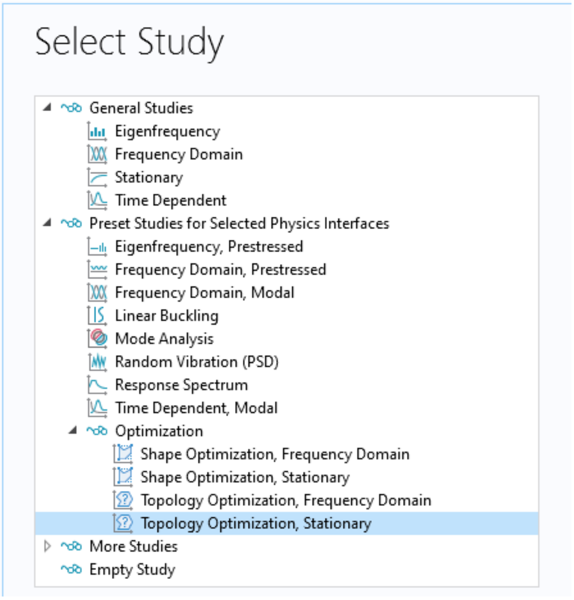
Note: These instructions are for the user interface on Windows but apply with minor differences also to Linux and Mac. In particular, the Home toolbar refers to the fixed set of controls near the top of the Desktop

1. In the New window click the Model Wizard button .

Model Wizard

1. In the Model Wizard window, click 2D .
2. In the Select Physics tree under Structural Mechanics, click Solid Mechanics (solid) .
3. Click Add.
4. Click the Study button .
5. In the tree under Preset Studies for Selected Physics Interfaces, Expand the Optimization folder , and click Topology Optimization, Stationary .

6 Click Done ☑.



Select Study

- General Studies
  - Eigenfrequency
  - Frequency Domain
  - Stationary
  - Time Dependent
- Preset Studies for Selected Physics Interfaces
  - Eigenfrequency, Prestressed
  - Frequency Domain, Prestressed
  - Frequency Domain, Modal
  - Linear Buckling
  - Mode Analysis
  - Random Vibration (PSD)
  - Response Spectrum
  - Time Dependent, Modal
  - Optimization
    - Shape Optimization, Frequency Domain
    - Shape Optimization, Stationary
    - Topology Optimization, Frequency Domain
    - Topology Optimization, Stationary
- More Studies
- Empty Study

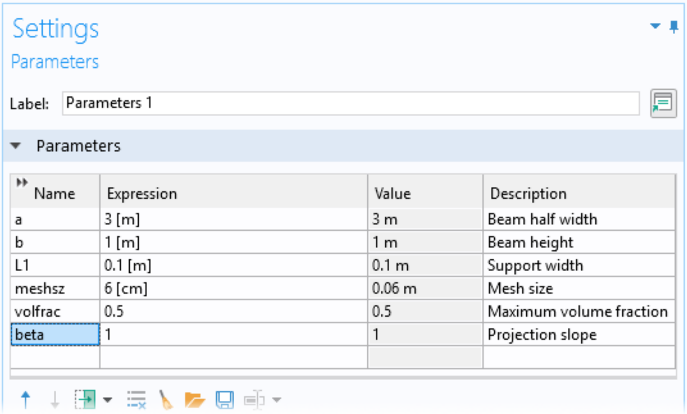
Global Definitions

Set up named parameters for any quantity that you might perceivably want to change later on. The global Parameters node then gives you a direct overview of the values that are used in the model, making it easier to reproduce and modify.

Parameters

1 In the Home toolbar click Parameters Pi and select Parameters 1 Pi .

2 In the Settings window for Parameters locate the Parameters section and enter the following settings:



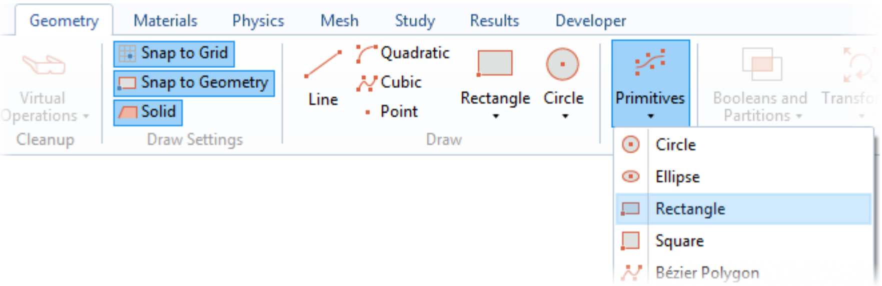
Name	Expression	Value	Description
a	3 [m]	3 m	Beam half width
b	1 [m]	1 m	Beam height
L1	0.1 [m]	0.1 m	Support width
meshsz	6 [cm]	0.06 m	Mesh size
volfrac	0.5	0.5	Maximum volume fraction
beta	1	1	Projection slope

Geometry 1

The actual beam geometry consists of a rectangle of width $2a$ and height b; however, because of the mirror symmetry, you only need to draw the left half. Also insert two points to delimit boundary segments where the central load and the roller condition in the lower-left corner can be applied. Applying the boundary conditions on small boundaries of fixed size avoids the mesh dependency caused by point conditions.

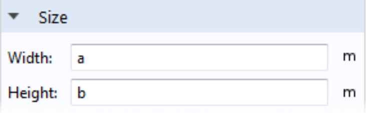
Rectangle 1 (r1)

1 In the Geometry toolbar, click Primitives and choose Rectangle ▭ .



2 In the Settings window for Rectangle locate the Size section.

- In the Width text field, type `a`.
- In the Height text field, type `b`.



Size

Width: a m

Height: b m

3 Click the Build Selected button .

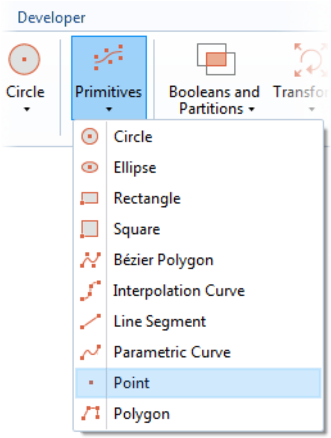
Point 1 (pt1)

4 In the Geometry toolbar, click Primitives and choose Point · .

5 In the Settings window for Point locate the Point section. In the y text field, type `L1`.

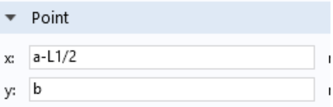
Point 2 (pt2)

1 In the Geometry toolbar, click Primitives and choose Point · .



2 In the Settings window for Point locate the Point section.

- In the x text field, type `a-L1`.
- In the y text field, type `b`.



Point

x: a-L1/2

y: b

3 Click the Build Selected button .

Add Material

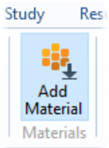
Add structural steel as the domain material in the beam. Note that Young's modulus and Poisson's ratio are defined in their own property group subnode under the material.

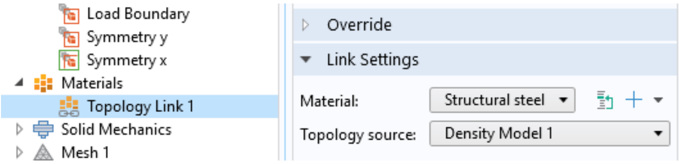
1 In the Home toolbar click Add Material .

2 In the Add Material window tree under Built-In, click Structural steel. Click Add to Global Materials.

3 In the Model Builder Window, under Component 1 (comp1)>Materials click Topology Link 1

4 Select Structural steel in the Material list.





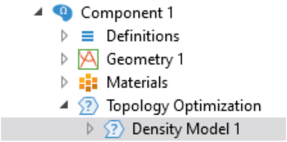
Material: Structural steel

Topology source: Density Model 1

Definitions

Set up the Density Model feature to use Hyperbolic tangent projection with a projection slope equal to the beta parameter.

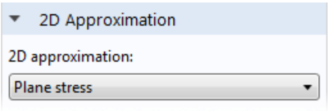
Density Model 1 (dtopo1)



1. In the Model Builder Window, under Component 1 >Topology Optimization click Density Model.
2. Enable Hyperbolic tangent projection and set the projection slope to `beta`.
3. Set the initial value to `volfrac` in the settings window for the Density model.

Solid Mechanics (solid)

1. In the Model Builder under Component 1, click Solid Mechanics.
2. In the Settings window for Solid Mechanics locate the 2D Approximation section. From the 2D approximation list, choose Plane stress.

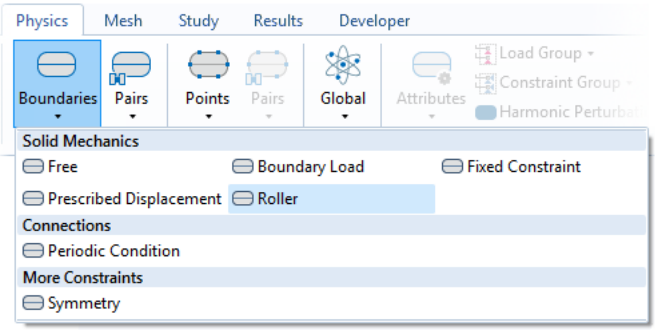


2D approximation: Plane stress

Roller 1

Use a Roller condition to represent the mirror symmetry boundary. If you have access to the Structural Mechanics Module, you can use a Symmetry condition instead.

1 In the Physics toolbar click Boundaries and choose Roller.



2 Select Boundary 6 only.

Note: 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 boundary number in the Selection field. For more information about selecting geometric entities in the Graphics window, see the *COMSOL Multiphysics Reference Manual*.

Prescribed Displacement 1

Add a prescribed displacement condition for the lower-left corner of the beam.

1 In the Physics toolbar click Boundaries and choose Prescribed Displacement.

2 Select Boundary 1 only.

3 Check the box with the label Prescribed in y direction.

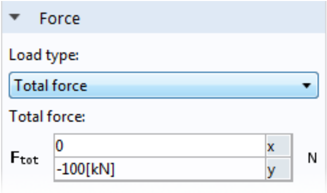
This will support the beam in the corner, but due the vertical alignment of the boundary, there will be no support for bending in the corner.

Boundary Load 1

Since this is a linear problem, the magnitude of the applied force does not affect the optimal topology.

1 In the Physics toolbar click Boundaries and choose Boundary Load.

2 Select Boundary 5 only.



Force

Load type:

Total force

Total force:

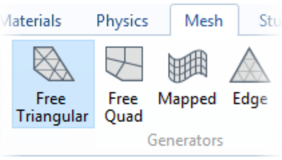
F_{tot} 0 x
-100[kN] y N

3. In the Settings window for Boundary Load locate the Force section.
   - From the Load type list, choose Total force.
   - Specify the $\mathbf{F}_{\mathrm{tot}}$ vector as `-100[kN]` in the y direction

Mesh 1

For topology optimization, it can be necessary to take manual control over the mesh size, but it this case the predefined meshes work fine, but in this model we will show how one can save computational time by solving the topology optimization problem on a coarse mesh and then use the result as an initial guess for an optimization on a finer mesh.

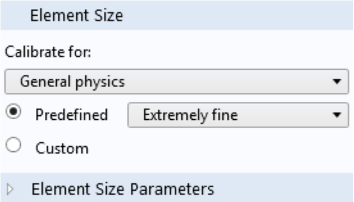
1. In the Mesh toolbar click Free Triangular .



Size

Pick the predefined Extra fine option as the element size.

1. In the Model Builder under Mesh 1, click Size .
2. In the Settings window for Size locate the Predefined list and choose Extremely fine.



Element Size

Calibrate for:

General physics

● Predefined Extremely fine

○ Custom

▷ Element Size Parameters

3. Click the Custom button and set the Maximum element size to `meshsz`.
4. Click the Build All button .

Study 1

1. Select the first study in the Model builder window.

2 Change the label to Optimization.

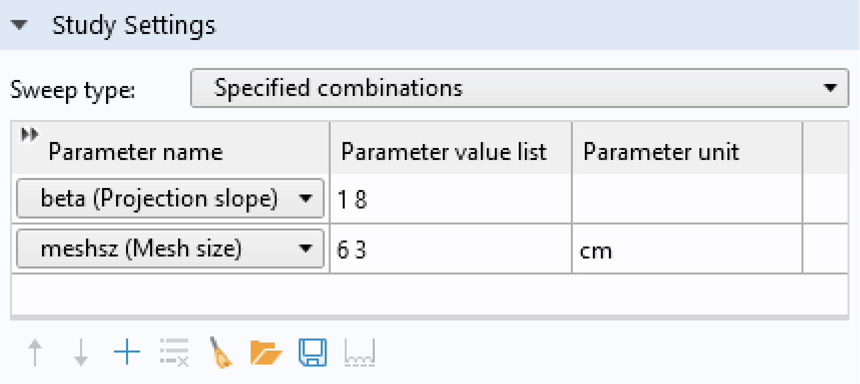
3 Clear the Generate Default Plots check box.

Parametric Sweep

First initialize a the study to create a dataset that can be used for setting up suitable plots.

1 In the Study toolbar click Parametric Sweep

2 Click Add + twice and change the table to sweep the `meshsz` and `beta` parameters from 3[cm] to 6[cm] and 1 to 8, respectively.



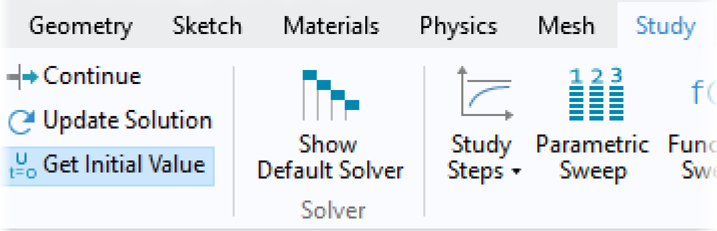
Study Settings

Sweep type: Specified combinations

Parameter name	Parameter value list	Parameter unit
beta (Projection slope)	1 8	
meshsz (Mesh size)	6 3	cm

3 Expand the Advanced Settings section and enable Reuse solution from previous step.

4 In the Study toolbar click Get Initial Value.



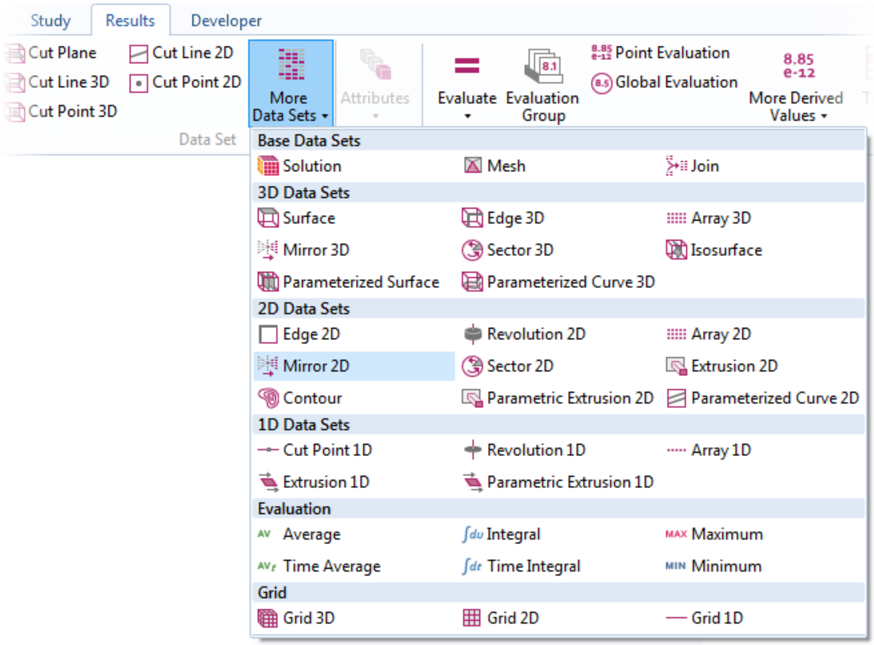
Results

Use a Mirror dataset to plot the whole beam rather than just the left half. Also, customize one of the plots for plotting the topology as it evolves when solving using an optimization solver.

Datasets

For the Mirror dataset, specify two points on the symmetry plane.

1 In the Results toolbar click More Datasets and choose Mirror 2D .

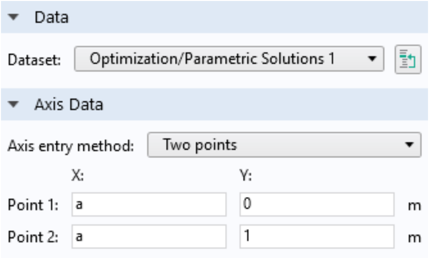


2 In the Settings window for Mirror 2D choose Optimization/Parametric Solution 1 from the Dataset list.

3 In the Settings window for Mirror 2D locate the Axis Data section.

- In row Point 1, set X to a.
- In row Point 2, set X to a.

4 Locate the Advanced section. Select the Define Variables check box.



▼ Data

Dataset: Optimization/Parametric Solutions 1

▼ Axis Data

Axis entry method: Two points

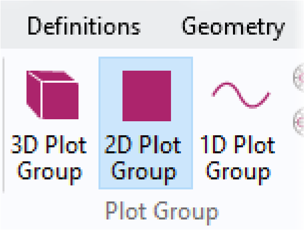
	X:	Y:	
Point 1:	a	0	m
Point 2:	a	1	m

5 Press CTRL+D to duplicate the dataset.

6 Choose Optimization/Solution from the Dataset list in the 2nd mirror dataset.

Topology

Create a custom plot to show the control material volume factor and filtered material volume factor in the same plot.



1 In the Results toolbar click 2D Plot Group .

2 In the Settings window for 2D Plot Group locate the Data section. From the Dataset list, choose Mirror 2D 2.

3 Click 2D Plot Group 2 and press F2.

4 Go to the Rename 2D Plot Group dialog box and type `Topology` in the New name text field. Click OK.

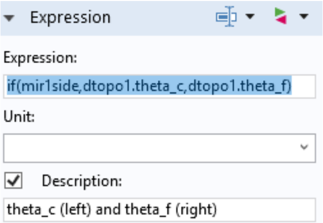
5 In the Topology toolbar click Surface .

6 Change the expression to `Topology`.

Surface

1 In the Model Builder expand the Results>Topology node, then click Surface 1 .

2 In the Settings window for Surface locate the Expression section. In the Expression text field, type `if(mir1side,dtopo1.theta_c,dtopo1.theta_f)`.



Expression

Expression:

if(mir1side,dtopo1.theta_c,dtopo1.theta_f)

Unit:

☑ Description:

theta_c (left) and theta_f (right)

3 Select the Description check box. Write `theta_c (left) and theta_f (right)` in the associated text box.

4 Select the reverse color table check box.

5 Select the `Topology` Plot group in the Model Builder window and press CTRL+SHIFT+D to duplicate it.

Topology 2

1 Rename the new plot group to `Topology 2`.

2 From the Dataset list, choose Mirror 2D 1.

3 Expand the new plot group

Surface

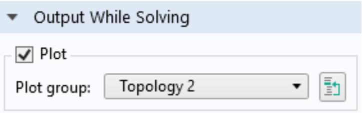
1 In the Model Builder expand the Results>Topology 2 node, then click Surface 1 .
2 In the Settings window for Surface locate the Expression section. In the Expression text field, type `if(mir1side,dtopo1.theta_f,dtopo1.theta)`.
3 Select the Description check box. Write `theta_f (left) and theta (right)` in the associated text box.

Study 1

The topology optimization study step was added from the model wizard, so the solver, objective function and constraint is already setup, so the only thing left to do is to reduce the number of iterations and switch on plotting while solving.

Topology Optimization

1 Select the Topology Optimization study step in the Model Builder window, .
2 Set the Maximum number of iterations to `50`.
3 Locate the Output While Solving section.
   - Select the Plot check box.
   - From the Plot group list, choose Topology 2.

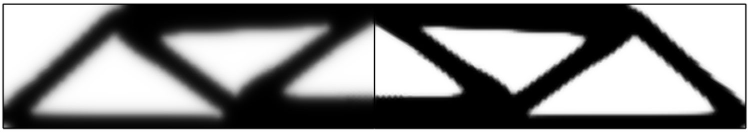


▾ Output While Solving

☑ Plot

Plot group: Topology 2

4 In the Home toolbar click Compute = .



The default plots show the material distribution in the optimized structure, clearly indicating that the load paths follow the optimized topology. Switch to the other plot group to display the stiffness distribution of the final design.