9 Determination of Material Parameters from Experimental Data

Chapter Outline

9.1 Introduction

Every finite element model requires a calibrated material model for each material to be simulated. Each material model, in turn, consists of two parts: a constitutive model specifying the equations that govern the material response, and a set of parameters that go into the equations and are specific for each material. As an example, the linear elastic material model is given by Equation (5.7) and takes two material parameters: a Young's modulus and a Poisson's ratio.

Different techniques can be used to determine the necessary material parameters. All methods, however, require experimental test data and the selection of an appropriate constitutive model, followed by some procedure for determining the parameters for the model from the experimental data. So far in this book we have focused on the first two of these topics, but have not spent much time discussing different procedures that can be used for calibrating the selected material model; and clearly, if the material parameter extraction is not done properly then the finite element results will not be very accurate either.

Mechanics of Solid Polymers.
DOI: http://dx.doi.org/10.1016/B978-0-323-31150-2.00009-1

It is interesting to note that material parameter extraction is a topic that is not that well covered in the literature, perhaps because on the surface the theory is not very difficult. The difficulties involved with finding the best set of material parameters for a material model, however, are real and often a major challenge when it comes to using an advanced material model for polymers.

There are various commercial software packages that can be used to calibrate a material model from experimental data, and most FE programs contain some functionality for material model calibration. None of the major FE programs, however, include a general purpose material model calibration tool, or even tools for all the material models that are included in their library of material models. One approach that is sometimes used to overcome this problem is to rely on graphical techniques, and trial-and-error techniques. These approaches are useful for learning how a material model behaves, but it is difficult and time-consuming to use these approaches in general.

This chapter will introduce the theory behind material parameter extraction, and discuss different techniques that can be used to find an appropriate set of parameters for both simple and advanced material models.

9.2 Mathematics of Material Parameter Determination

Determining the most appropriate material parameters for use with a material model require: (1) a set of experimental data; and (2), the selection of a constitutive model. As was discussed in Chapter 2, different constitutive models need different types of experimental data for the purpose of calibration. For example, a hyperelastic material model will require, at a minimum, monotonic loading to a final strain, in one or more loading modes. A hyperelastic model does not need experimental data at different strain rates, as this class of models predicts the same stress for a given strain independent of applied strain rate. To calibrate a viscoplastic model, on the other hand, requires also data at different strain-rates.

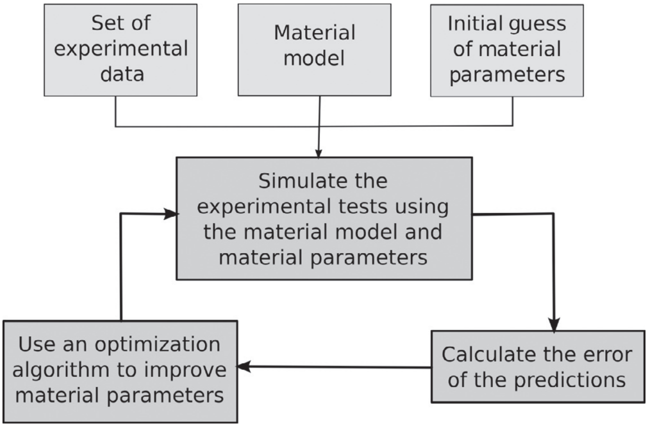

Set of experimental data
Material model
Initial guess of material parameters
Simulate the experimental tests using the material model and material parameters
Use an optimization algorithm to improve material parameters
Calculate the error of the predictions

Figure 9.1 Flow diagram for material parameter extraction.

To start the material parameter extraction, many techniques also require an initial guess of the material parameters. This guess often does not have to be very close to the optimal set of parameters, and prior knowledge of the parameters of similar materials is often sufficient.

A flow chart outlining a procedure for determining the material parameters is given in Figure 9.1.

As illustrated, the first step is to simulate the experimental loading conditions using the selected material model with an initial guess of the material parameters. The model predictions are then compared to the experimental data, and the difference is used to calculate a scalar error value of the predictions. Finally, a numerical minimization algorithm is used to determine a new guess of the material parameters and the procedure is repeated until no further improvement is achieved.

This procedure can be written in the form of a mathematical minimization problem:

$$\min_{\boldsymbol{\xi}} \sum_{i=1}^{N} f\left(\mathbf{M}^i(\boldsymbol{\xi}) - \mathbf{E}^i\right), \tag{9.1}$$

where $\boldsymbol{\xi}$ is the vector of material parameters to optimize, N is the number of experimental data sets, $f(\cdot)$ is a function for calculating the error value of a prediction, $\mathbf{M}^i(\boldsymbol{\xi})$ is the model prediction of experimental data set i, $\mathbf{E}_i$ is the experimental data set i. Both $\mathbf{M}^i$ and $\mathbf{E}^i$ are vectors of either stress or strain as discussed in Section 9.4. This formulation, which is also called a non-linear programming problem, needs to be solved once for each material.

To solve this non-linear optimization problem it is necessary to: (1) have a good strategy for determining the initial material parameters; (2) select an appropriate error measurement function $f(\cdot)$; and (3) use a good non-linear optimization finder. These topics are discussed in more detail in the following sections.

9.3 Initial Guess of the Material Parameters

Finding a good initial guess of the material parameters is important in order to ensure that the material parameters that are determined by the minimization algorithm are close to the global optimum. A poor initial guess can cause the minimization algorithm to get stuck at an undesirable local minimum, and it can also significantly slow down the parameter extraction procedure.

As an example, Figure 9.2 shows experimental data for a chloroprene rubber tested in uniaxial tension. If one tries to fit an elastic-plastic material model with isotropic hardening using three pairs of yield stress and plastic strain values, but use an initial guess of the yield stress values that are too high, then the material parameter optimization can get stuck at the solution shown in Figure 9.2.

By selecting a better initial guess of the yield stress values, the same material model and optimization algorithm can give the predictions shown in Figure 9.3, which is significantly better.

This example is somewhat trivial since it is well-known that the predicted stress-strain curves from the J_2-plasticity model with isotropic hardening is piecewise linear, and hence the predictions in Figure 9.2 are clearly not optimal. The problem illustrated in the example, however, can be a significant hurdle when working with a highly non-linear material model for which the behavior of the material model may not be clear a priori.

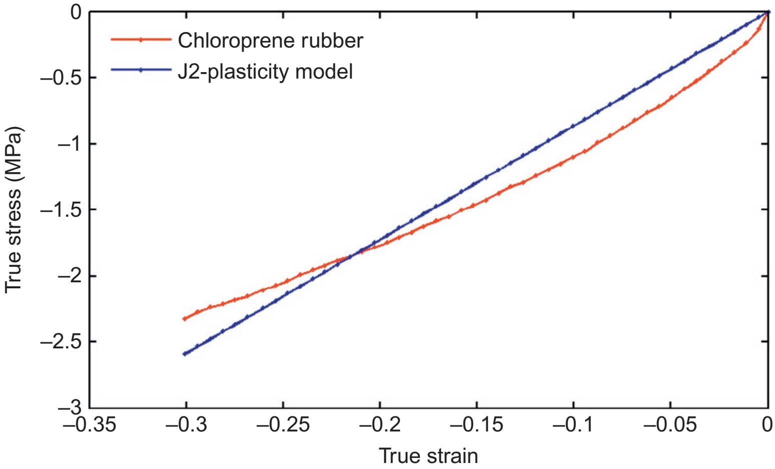


Figure 9.2 Example of how a bad initial guess of the material parameters can give poor model predictions. In this case the yield stress values were too high.

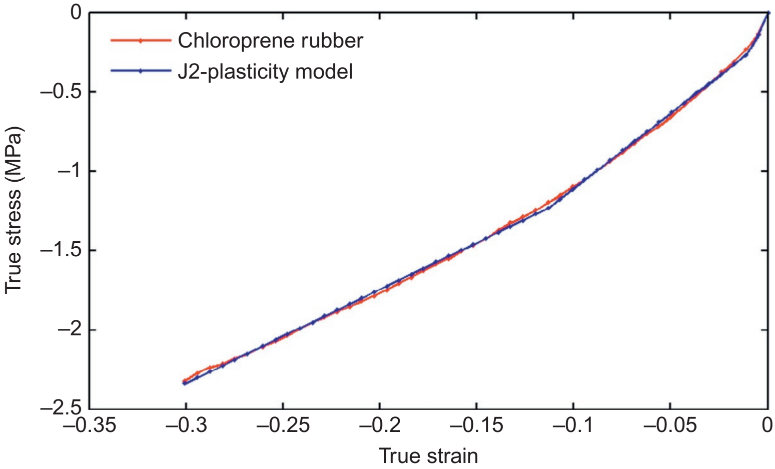


Figure 9.3 Example of how a better initial guess of the material parameters can give good model predictions. The example in this figure uses the same material model and optimization algorithm as in Figure 9.2.

The two most common approaches for finding a good initial guess are:

- **The Monte Carlo method**
  The Monte Carlo method [1, 2] is an optimization method in which each material parameter is first restricted to be in a certain pre-defined interval, and then for each parameter a random value in the specified range is generated. The set of parameters is then evaluated using an error measurement function, as discussed in Section 9.4. The whole procedure is then repeated a number times and the best value is used as the initial guess of the material parameters.
- **Prior knowledge of similar materials**
  Having prior knowledge of the material parameters of similar materials is typically the best way to construct a good set of initial material parameters. Not only does it allow for the determination of a good set of parameters, but it is also computationally fast.

9.4 Error Measurement Functions

Once the material model has been used to simulate an experimental test, the next step is to evaluate the magnitude of the error of the model predictions. If the experiment was run in strain-control mode, then the simulation will follow the exact same strain history, and the difference between the model predicted and experimental *s*tress values can be used to evaluate the accuracy of the model predictions, see Figure 9.4. That is, the error of the prediction is given by:

$$error = f\left(\boldsymbol{\sigma}^{\text{pred}} - \boldsymbol{\sigma}^{\text{exp}}\right), \tag{9.2}$$

where $\boldsymbol{\sigma}^{\text{pred}}$ is the predicted stress vector, and $\boldsymbol{\sigma}^{\text{exp}}$ is the experimentally determined stress vector.

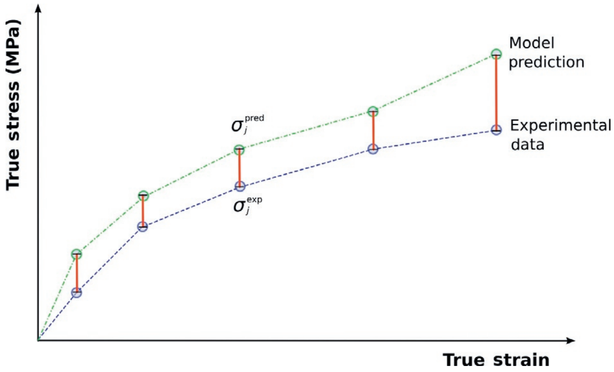


Figure 9.4 Residual error for a strain-controlled experiment.

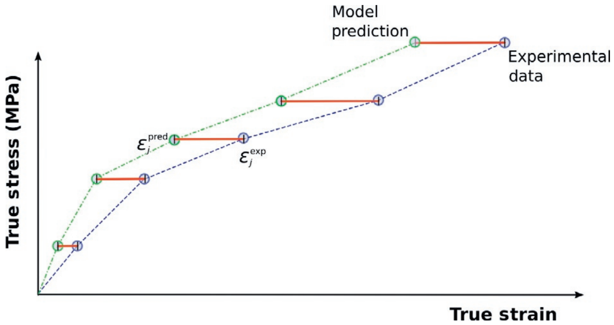


Figure 9.5 Residual error for a stress-controlled experiment.

On the other hand, if the experiment was run in load-control mode, then the simulation will follow the exact same stress history, and the difference between predicted and experimental strain values can be used to evaluate the accuracy of the model predictions, see Figure 9.5. That is, the error of the prediction is given by:

$$error = f\left(\boldsymbol{\epsilon}^{\text{pred}} - \boldsymbol{\epsilon}^{\text{exp}}\right), \tag{9.3}$$

where $\boldsymbol{\epsilon}^{\text{pred}}$ is the predicted strain vector, and $\boldsymbol{\epsilon}^{\text{exp}}$ is the experimentally determined strain vector.

The function $f(\cdot)$ in Equations (9.2) and (9.3) is typically selected from one of the following:

- **Normalized Root-Mean Square Difference**
  This function is defined by

$$f(\mathbf{y}^{\text{pred}}, \mathbf{y}^{\text{exp}}) = \sqrt{\left\langle (\mathbf{y}^{\text{pred}} - \mathbf{y}^{\text{exp}})^2 \right\rangle} \Big/ \sqrt{\langle (\mathbf{y}^{\text{exp}})^2 \rangle}. \tag{9.4}$$

- **Normalized Mean Absolute Difference**
  This function is defined by

$$f(\mathbf{y}^{\text{pred}}, \mathbf{y}^{\text{exp}}) = \langle |\mathbf{y}^{\text{pred}} - \mathbf{y}^{\text{exp}}| \rangle \Big/ \langle |\mathbf{y}^{\text{exp}}| \rangle. \tag{9.5}$$

- **Coefficient of Determination, R^2**
  This function is defined by

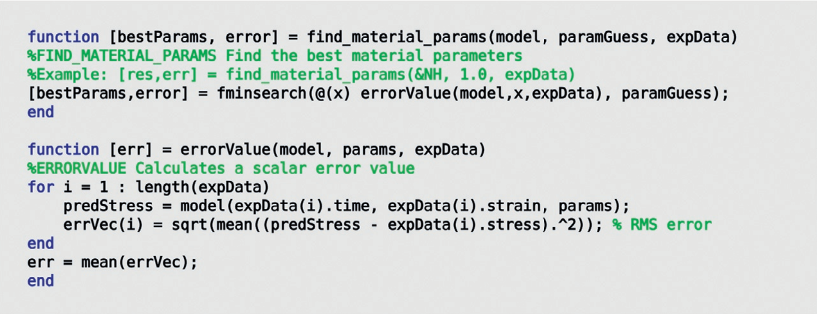
$$f(\mathbf{y}^{\text{pred}}, \mathbf{y}^{\text{exp}}) = 1 - \sum_{i=1}^{n} \left(y_i^{\text{pred}} - y_i^{\text{exp}} \right)^2 \Big/ \sum_{i=1}^{n} \left(y_i^{\text{exp}} - \langle \mathbf{y}^{\text{exp}} \rangle \right). \tag{9.6}$$

In these equations $\langle \cdots \rangle$ is the arithmetic mean, $\mathbf{y}^{\text{pred}}$ and $\mathbf{y}^{\text{exp}}$ are either stress or strain vectors depending on if the experiment was run in displacement control or load control. It is important to note that all of these equations are normalized. The normalization is introduced in order to give equal weights to experimental data with different magnitudes of stress and strain. It is also possible to apply a weight function to portions of experimental data curves, or different weights to different experimental tests.

9.5 Algorithms for Parameter Extraction

One way to find the optimal material parameters giving the smallest difference between predicted and experimental data is to use a custom computer program written in any mathematical software or general purpose language. As an example, the Matlab function `find_material_params`, shown below, uses the root mean square error to evaluate the fitness of a set of model predictions. The function uses an optimization algorithm based on the Nelder-Mead simplex method.

Matlab File Name: find_material_params.m



```
function [bestParams, error] = find_material_params(model, paramGuess, expData)
%FIND_MATERIAL_PARAMS Find the best material parameters
%Example: [res,err] = find_material_params(&NH, 1.0, expData)
[bestParams,error] = fminsearch(@(x) errorValue(model,x,expData), paramGuess);
end

function [err] = errorValue(model, params, expData)
%ERRORVALUE Calculates a scalar error value
for i = 1 : length(expData)
    predStress = model(expData(i).time, expData(i).strain, params);
    errVec(i) = sqrt(mean((predStress - expData(i).stress).^2)); % RMS error
end
err = mean(errVec);
end
```

The example shown above is using the `fminsearch` function which is a built-in Matlab implementation of the Nelder-Mead Simplex method [3]. The Simplex method is often used for material parameter extraction because of its robustness. The main weakness of this optimization algorithm is that it can be slow and have a tendency of getting stuck at a local minimum. There are a number of alternative optimization methods that have been designed to overcome these limitations. One interesting method is the Powell method [4], another is a class of optimization algorithms called genetic algorithm [5, 6].

Another option is to use a commercial material parameter extraction tool, such as the MCalibration software [7]. The goal of this chapter is to show that it is very important to have a general purpose material parameter extraction tool, and that a material parameter extraction tool can be immensely valuable for a finite element simulation engineer.

9.6 Exercises

1. What are the different steps that are required for calibrating a material model?
2. What methods are commonly used to determine an initial site of material parameters? What are the strengths and limitations of these methods?

3. What error measurement functions are typically used for material parameter extraction? Which error measurement function do you think is most useful?
4. Write a material parameter extraction tool in your favorite mathematical software language, and use the tool to determine the optimum Young's modulus for a linear elastic material model.

References

[1] R.Y. Rubinstein, D.P. Kroese. Simulation and the Monte Carlo Method. Wiley, New York, 2007.

[2] C.P. Robert, G. Casella. Monte Carlo Statistical Methods. Springer, New York, 2005.

[3] J.A. Nelder, R. Mead. A Simplex Method for Function Minimization. Computer J., 7 1965 308-313.

[4] R.P. Brent. Algorithms for Minimization Without Derivatives. Prentice-Hall, Englewood Cliffs, NJ, 1993.

[5] M. Michelle. An Introduction to Genetic Algorithms. The MIT Press, Cambridge, 1998.

[6] D.E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston, MA, 1989.

[7] Veryst Engineering. http://PolyUMod.com.