

Downsampling inelastic cyclic stress-strain data with minimal points to preserve curvature

A Hartloper

July 20, 2021

Abstract

A downsampling algorithm for stress-strain data that uses few points but retains curvature within a pre-defined tolerance is developed in this document.

1 Introduction

The raw data (i.e., extensometer displacement, load cell force) in experiments used for material model calibration are typically sampled at 10 Hz in the EPFL test setup. As experiments range from around a few minutes (e.g., LP1) to hours (LP5), this results in thousands to hundreds of thousands of data points for each experiment. However, the sampled data is often denser than required to reasonably describe the stress-strain behavior of materials under the applied strain rates. When it comes to calibrating material models, the time required is proportional to the number of data points. Therefore, time can be saved by reducing the number of data points prior to running the calibration while maintaining the fidelity of the data.

Experience has shown that the fidelity of an experiment can be kept with around 10–100 times less points depending on the load protocol. Calibrating the UVC model with ten load protocols *without* reducing the number of points is, therefore, expected to take around several weeks to a year. A “good” method of reducing the number of data points, or *downsampling* the data, is essential to keep the calibration time reasonable (say, on the order of days to a week per material). The properties that define a good downsampling method in this context are now defined.

A good downsampler for inelastic, cyclic, stress-strain data of structural steels:

1. samples enough points in the initial elastic region to obtain a good estimate of the initial elastic modulus,
2. samples the upper yield point,

3. samples all the maxima and minima in each loading cycle (i.e., the *peaks* of the stress-strain graph),
4. samples enough points to retain the fidelity of regions in the stress-strain graph with high curvature,
5. samples as few points as possible in regions of low curvature,
6. can handle different load protocols,
7. can handle noise in the measurements,
8. can handle different strain rates,
9. has interpretable heuristic parameters,
10. is easy to use.

An algorithm that combines different techniques to satisfy these desirable properties is developed in this document.

2 Proposed algorithm

The proposed algorithm is composed of three parts. The first part is to find the peaks in the stress-strain data. The second part is to downsample the entire stress-strain history based on the curvature of the stress-strain data. The third part is to keep additional points in the initial elastic region. The final downsampled data is the unique set combining all three parts.

Most of the heavy lifting is done in the second part. The first part is essential to overcome *aliasing* in the sampled data. The third part is essential to ensure fidelity of the initial elastic region. Details are given for each part and a summary of the entire procedure is given afterwards.

2.1 Identifying peaks in the stress-strain graph

The peaks in the stress-strain graph are identified by the minima or maxima in each cycle. Cycles can be fairly reliably identified by the switching of signs in the stress signal. Therefore, the peaks are the minima and maxima between each change in sign of the stress signal. This method works well if the stress crosses zero in each cycle and there is no substantial noise in the stress signal around zero.

An attempt at identifying upper yield point, if it exists, is made in this first part. First, the 0.2 % offset yield stress is computed using the initial elastic modulus. The initial elastic modulus itself is computed using linear regression up to $0.65f_{y,n}$, where $f_{y,n}$ is the nominal yield stress of the material. Then, the upper yield point is selected as the point of maximum stress up to the 0.2 % offset yield stress. If there is no upper yield point (due to the

material or imperfections in the specimen), then the maximum will typically be at the 0.2 % offset yield stress, and this point is sampled instead.

Another important point in stress-strain data tested according to the RESSLab protocols is the first instance 2 % strain amplitude is crossed. This point is important because the strain rate changes by a factor of around 30 at this point. Therefore, this point is identified as well. The final point of importance is the final point. Unless otherwise specified, the final point is selected at 12.5 % strain due to limitations in the small extensometer used for M8 experiments. Any points past the final point are neglected.

To summarize, the sampled points from the first part are:

- the upper yield point (or 0.2 % offset yield stress point),
- the point where 2 % strain amplitude is crossed (not applicable for LPs 4 and 5),
- the peaks in the stress-strain graph, and
- the final strain point (typically 12.5 % strain).

2.2 Curvature-based downsampling

The idea in the second part of the overall procedure is to remove as many points as possible in the “straight” regions of the stress-strain graph and keep as many as necessary in the “curved” regions. The algorithm used to accomplish this task was based on <https://kaushikghose.wordpress.com/2017/11/25/adaptively-downsampling-a-curve/>. The key to this algorithm is to define what is “straight”, and to sample a point every time that this condition is violated.

The method, named the “maximum deviation downsampler”, is outlined in Algorithm 1. This method requires a set of points and a tolerance, ϵ . The algorithm starts at the first point and steps through each point in the set. A line is computed between the starting point and the current point. The perpendicular distance is computed between the line and each point in the range of the starting and current point. If the perpendicular distance is greater than the tolerance, the point before the current point is sampled; the sampled point is then set as the starting point.

The keys to making Algorithm 1 in the current context are to: compute an appropriate perpendicular distance, to use an appropriate tolerance, and to remove noise from the data. One challenge in stress-strain data is that the “distances” in stress and strain have different units. This is handled by normalizing the stress data by the range in stress ($\sigma_i^* = \sigma_i / (\max \sigma - \min \sigma)$), and the strain by the range in strain ($\epsilon_i^* = \epsilon_i / (\max \epsilon - \min \epsilon)$). Therefore, the normalized values have a range of 1.0 in both axes.

Given this normalization, a tolerance of $\epsilon = 0.001$ has been found to be appropriate. This tolerance can be interpreted that “straight” is defined

as 1/1000'th of the maximum “distance” spanned by the graph. Finally, noise in the data can cause spurious sampling of points. Therefore, a moving average filter is applied to the stress-strain data prior to normalization. The moving average filter requires a window length and an interpolation order. A window length of 55 is used for the data prior to 2 % strain (low strain rate), and a window length of 5 is used for the data past 2 % strain.

Filtering the stress-strain data removes the noise, but leads to aliasing. Recall that the peaks of the stress-strain signal are already included in part 1 of the overall procedure, therefore, aliasing is not an issue in part 2. For this reason, linear interpolation is used because it is most effective at reducing the noise (the trade-off is that it increases aliasing—which we have already dealt with). Therefore, the set of 2-D points $\{x_i\}$ input to Algorithm 1 are the filtered, normalized stress-strain data.

Algorithm 1 Maximum deviation downsampler.

```

1: input: Set of points,  $\{x_i\}_{i=0}^{N-1}$  with  $x_i \in \mathbb{R}^n$  and  $N \in \mathbb{Z}$ ; and tolerance,  $\epsilon \in \mathbb{R}$ .
2: output: Set of indices between  $[0, N)$  to keep,  $k_{sample}$ .
3:  $k_{sample} \leftarrow \emptyset$ 
4:  $i \leftarrow 0$ 
5: for  $j \in (0, \dots, N)$  do
6:      $l \leftarrow x_j - x_i$  ▷ Line between  $i$  and  $j$ 
7:     for  $k \in (i, \dots, j)$  do
8:          $d_k \leftarrow x_k \perp l$  ▷ Perpendicular line between  $x_k$  and  $l$ 
9:     end for
10:    if  $\max \|d_k\|_2 > \epsilon$  then
11:         $k_{sample} \leftarrow k_{sample} \cup \{k - 1\}$ 
12:         $i \leftarrow k - 1$ 
13:    end if
14: end for
15: return:

```

2.3 Overall algorithm

Algorithm 2 Overall summary of the proposed downsampling method.

```

1: hi

```
