

Using the TI980 SRJ LDF Generator and Analysis Software

Kevin M. Schmalbach

Supplement to: <https://doi.org/10.1007/s11340-022-00833-x>

Load Function Generator

Installation

This version of the load function generator uses Python 3. There are two versions: one using Matplotlib to preview the load function and one without.

1. Install [Anaconda 3](#) to your system. This is an open-source Python distribution.
 - a. Additional installation help can be found [here](#).
 - b. Installing Anaconda to the system path can be useful for getting Matplotlib to work. If you already know how to get Matplotlib working otherwise, proceed accordingly.
2. Install [Matplotlib](#). This is required so that a plot of the load function can be generated before creation (see below).

Running the Software

If you have an IDE, e.g. [Visual Studio Code](#), the software can be run from there. An IDE would allow you to see any error messages that may arise (for cases that I haven't thought of!). If you send me screenshots of inputs and errors, I'm happy to go through and fix! Otherwise, open Anaconda Prompt, which will open a command line-type console. Navigate to the directory containing the .py file and use the command "python SRJ_Generator".

Description of Inputs

General Parameters

Open loop/ load control: Whether feedback should be used to control the load. The DMA oscillation can sometimes cause difficulties with the controller (sometimes causing phase lag in non-viscoelastic materials).

Low load/high load: Selection indicates whether the load function is to be used with a low load/standard or high load transducer. The default is low load.

Berkovich/Spherical: Chooses which loading profile to use based on tip geometry. Berkovich

controls load using $\dot{\epsilon} = \frac{\dot{P}}{2P}$ [1]. Spherical uses $\dot{\epsilon} = \frac{\frac{4}{9\pi} \left(\frac{3}{4}\right)^{\frac{1}{3}} \dot{P}}{E_r^{\frac{1}{3}} R^{\frac{2}{3}} P^{\frac{2}{3}}}$ [2].

References:

1. Lucas BN, Oliver WC (1999) Indentation power-law creep of high-purity indium. Metall Mater Trans A 30:601–610. <https://doi.org/10.1007/s11661-999-0051-7>
2. Feldner P, Merle B, Göken M (2017) Determination of the strain-rate sensitivity of

Number of strain rates: Designates the number of different strain rates to be used in the strain rate jump test. For instance, a simple SRJT with strain rates of 0.05, 0.005, and back to 0.05 s⁻¹ would require three strain rates. A constant strain rate experiment can be run with a single strain rate. Clicking the “add strain rates” button opens another window to add in the desired strain rates and the end loads for each strain rate segment. The pop in load box allows the user to add a linear segment at the beginning of the test either to increase the starting load for the first constant strain rate segment, thereby shortening the load function, or to get to a load high enough to induce the first pop in, if applicable.

Oscillation frequency: The frequency of the oscillatory load. It is important to test at a frequency which avoids the resonant frequency of the transducer. It is also important to test at a frequency which shows constant dynamic compliance with respect to depth, which can be seen in the dynamic calibration.

Points per loading segment: The number of data points recorded per loading segment. A lower number of points per loading segment translates to shorter loading segments, thereby more closely approximating the exponential curve. However, there are also therefore more loading segments; Triboscan software can only handle approximately 2000 loading segments.

Oscillation amplitude: The amplitude of the oscillatory load. A higher value will cause a higher displacement amplitude, which should be kept in the range of 1-2 nm. Note that this does not contain a unit; a value of 1 does not necessarily create an amplitude of 1 nm. It may be higher or lower, depending on material.

File name: The file name for the load function. Saves in the directory where the Python file is saved.

Scaling exponent: Indicates the power to which the load amplitude should be scaled to create a constant displacement amplitude. Scaling factors should be given as decimals. A typical scaling factor is ½ but should be lowered if the displacement amplitude rises throughout the test.

Show plot: If checked, gives a preview of the load function prior to load function creation to allow the user to change parameters before load function creation. The default is unchecked.

.prm/ Read from prm: When a load function is generated, two files are created. The first is the load function (FILENAME.lbf). The second is the parameters file (FILENAME.prm). The parameters file is a text file displaying all of the parameters used to generate the load function to increase test repeatability. The information in the parameters file is sufficient to completely recreate the load function. The load function generator can load in a parameters file via the “read from .prm” button, which will autofill the contents into the appropriate boxes to allow the user to recreate the original load function or to modify a parameter.

Spherical parameters

These parameters are necessary for defining strain rate for a spherical indenter. An error will appear if the boxes are not filled when the “spherical” radio button is selected.

Reduced modulus: The reduced modulus of the sample (in GPa) for the tip material. This can be obtained by other methods.

Tip radius: The radius of the spherical tip (in nm).

Pre-times

Lift and reseek segments are defined the same as in base Triboscan.

Lift: Time of the lift segment in seconds.

Reseek: Time of the reseek segment in seconds.

Dwell: Time of the dwell segment in seconds. This is a hold at the preload as the dynamic signal turns on prior to loading. Default is two seconds.

Approach Offsets and Lift Parameters

Same as defined in Triboscan. Be aware that units for the approach offsets are in microns, not millimeters (as they are in Triboscan).

Troubleshooting

Error:

“Too many load segments: please make changes and try again”

Solution:

There are two options. The preferred solution is to increase the number of points per loading segment. At high strain rates, this can cause some drops in hardness, as discussed in the paper. The other solution is to reduce the frequency. This effectively decreases the frequency of data acquisition, making loading segments longer. However, changing frequency can have other issues related to dynamic compliance and resonance.

Error:

“Frequency higher than maximum”

Solution:

Decrease the frequency. As of writing, the low load nanoDMA transducer has a maximum frequency of 301.5 Hz, and the high load transducer has a maximum frequency of 101.5 Hz.

Analysis Functions

Here, the various analysis and correction functions are discussed. An example workflow is presented in ExampleWorkflow.m. For inputs and outputs, the format is:

VariableName [data type]: description

These functions are written to accept **cell arrays only**, where each cell is a new test number. It helps reduce errors due to changing dimensions, especially for functions that change data size dimensions (CutBefore, CutAfter, SRJ_Analysis). Otherwise, it is likely that columns from different tests will end up different sizes and result in an error.

To not cause errors, it is important to remove the unnecessary data and organize it into a format on which it could easily be operated. Although these functions do not perform any of the calculations necessary to understand the strain rate dependence of material properties, they are crucial for reproducible and efficient analyses. There is one data correction that must be performed before these data curation processes: contact point/zero point correction. If not performed before these processes, the contact point will likely be removed from the data, making the correction impossible in later steps.

The parameters file (".prm") contains the parameters necessary to perform these data curation processes. The first function used is called "*ReadPRM*", which reads the parameters file for the relevant information, such as the times at which each of the strain rate segments is set to end. The last value of end times can then be used with another function, "*CutAfter*", which removes the data from after the last strain rate segment. The data after the last strain rate segment are from the load hold and unloading process, where the strain rate is no longer being controlled (and CSM disabled) and thus will be unimportant for the analysis.

The next step, "*CutBefore*", removes the data from low displacements (as defined by user input, either from a specified time or displacement setpoint), where the properties are dominated primarily by effects such as surface roughness and probe imperfection effects. This must be performed after the "*CutAfter*" function because it searches for the last sequential point at a displacement less than the given value, which will be the last data point in the file if the unloading data are not removed.

ReadARA

Usage:

This function reads the area function, which is necessary for several of the following corrections.

Inputs:

ARAdir [char]: The directory where the area function file is saved.

Outputs:

Coeffs [double]: The coefficients for the area function.

ReadPRM

Usage:

This function reads values for number of strain rates, strain rates, and strain rate end times from the parameters file.

Inputs:

PRMdir [char]: The directory where the parameters file is saved.

Outputs:

SRNum [double]: The number of strain rates in the strain rate jump test.

Strains [double]: The values of the strain rates in the strain rate jump test in s^{-1} .

EndTimes [double]: The time at which each strain rate ends. Useful for cutting the holding and the unload segments with CutAfter. Also used in the SRJ analysis to separate into individual strain rate segments.

PPSeg [double]: Number of points per loading segment used during the SRJ load function. Used during the SRJ analysis for averaging.

Freq [double]: Applied oscillation frequency. Useful for resetting the time if things like approach offsets are used.

ZeroHLIndentLoadDisp/ManualRezeroLD

Usage:

This function resets the zero point for the load and displacement. In most load functions, the probe makes contact, then performs a lift segment. In some cases, TriboScan has trouble automatically finding the point when the probe contacts the surface. This function is used to reset the load and displacement to the actual zero point via an interactive figure (for manual version).

Automatic version: Uses a deviation from flatness criterion to find the zero point. May require manual tuning for reliability.

Manual version: A graph of the load-displacement data is displayed. There are two input boxes for the displacement (in nm) and the load (in μN) of where the zero point currently is. This point is the new zero or contact point.

Inputs:

Disp [cell]: The displacement data.

Load [cell]: The load data.

Stiff [cell]: The stiffness. I use storage stiffness since I am using metals. Needed to recalculate hardness and modulus (in my case storage) using the measured stiffness.

ARAdir [char]: The directory where the area function is saved. Needed because this function calls ReadARA to get the coefficients needed to recalculate the properties.

Outputs:

ZeroedDisp [cell]: The new displacement data, after rezeroing.

ZeroedLoad [cell]: The new load data, after rezeroing.

NewMod [cell]: The new modulus data, after rezeroing.

NewHard [cell]: The new hardness data, after rezeroing.

CutBefore

Usage:

Used to remove data from before a certain point. For instance, one can pass in displacement data as the BasisData with a condition of 100 nm to cut out the data from before 100 nm. Needs to be called for each set of data to be cut (ex. Cutting time and displacement each require separate calls).

Inputs:

DataToCut [cell]: The data which should be cut.

BasisData [cell]: The data to which to compare the condition.

Condition [double]: Condition to compare data; designates where the data should be cut.

Outputs:

CutData [cell]: The data with points removed outside of the desired range.

CutAfter

Usage:

Used to remove data from after a certain point. For instance, one can pass in time data as the BasisData with a condition of EndTimes(end) to cut out the data from after the last strain rate segment ends. Needs to be called for each set of data to be cut (ex. Cutting time and displacement each require separate calls).

Inputs:

DataToCut [cell]: The data which should be cut.

BasisData [cell]: The data to which to compare the condition.

Condition [double]: Condition to compare data; designates where the data should be cut.

Outputs:

CutData [cell]: The data with points removed outside of the desired range.

NixGao/NixGaoCSM

Usage:

Used to remove the indentation size effect. Uses previously-obtained constant strain rate or partial load-unload data to calculate the parameters in the Nix-Gao model (h^* and H_0). The two functions have slightly different inputs, but perform the same steps.

Inputs (NixGao):

DataDir [char]: the directory where the partial load-unload data are saved

Row [double]: the row in the file where to start reading the data

Inputs (NixGaoCSM):

Datadir [char]: the directory where the constant strain rate data are saved

ARADir [char]: the directory where the area function is saved (because there is natively a rezeroing function for the constant strain rate data).

Outputs:

hstar [double]: the characteristic length scale of the Nix-Gao model.

H0 [double]: the hardness at infinite depth, according to the Nix-Gao model.

For more information on these parameters, see:

Nix WD, Gao H (1998) Indentation size effects in crystalline materials: A law for strain gradient plasticity. J Mech Phys Solids 46:411–425. [https://doi.org/10.1016/S0022-5096\(97\)00086-0](https://doi.org/10.1016/S0022-5096(97)00086-0)

PlasErrCorr

Usage:

Used to correct the hardness for plasticity errors. Uses a known reduced modulus (from other experiments) to solve for contact depth and contact area to recalculate the hardness.

Inputs:

ARADir [char]: The directory where the area function file is saved.

Disp [cell]: Displacement data.

Load [cell]: Load data.

Er [double]: A known reduced modulus in GPa. Can be taken from previous nanoindentation experiments.

Outputs:

hc [cell]: The newly calculated contact depth.

H [cell]: The newly calculated hardness, corrected for plasticity error.

Ac [cell]: The newly calculated contact area.

SRJ_Analysis

Usage:

Used to calculate the strain rate sensitivity and activation volume from the strain rate jump test data. There are several optional inputs that can change plotting defaults.

Required Inputs:

PRMDir [char]: The directory where the parameters file is stored.

AnMode [char]: The analysis mode. Ideally, they give similar results. There are two options:

“regression”: Uses a linear regression on the hardness vs. depth profile for each strain rate segment. It extrapolates the hardness back to the jump depth to find the difference in hardness at the displacement of the jump, where the material microstructure should be the same.

“average”: Rather than linear regression, just averages the last 10% of the hardness vs. depth profile to get the hardness from that strain rate segment.

Hardness [cell]: Hardness data to process.

StorMod [cell]: Modulus data to process.

Time [cell]: Time data to process.

Load [cell]: Load data to process.

Disp [cell]: Displacement data to process.

Optional Inputs (indicated via Name-Value Pairs):

DistIn [double]: Can be used if AnMode is set to “regression”. This is a (number of strain rates – 1) by 1 vector of decimal values for how much of the data from a given strain rate segment to ignore. This allows the user to ignore the transient hardness at the very beginning of the strain rate segment. If it is not input and AnMode is set to “regression”, a figure will open that will plot the data from each strain rate segment with a slider and a text box. When the slider is moved (or new value entered in the text box), the regression will rerun. Clicking the “Save” button will use the current value of the slider/text box for the analysis of that jump.

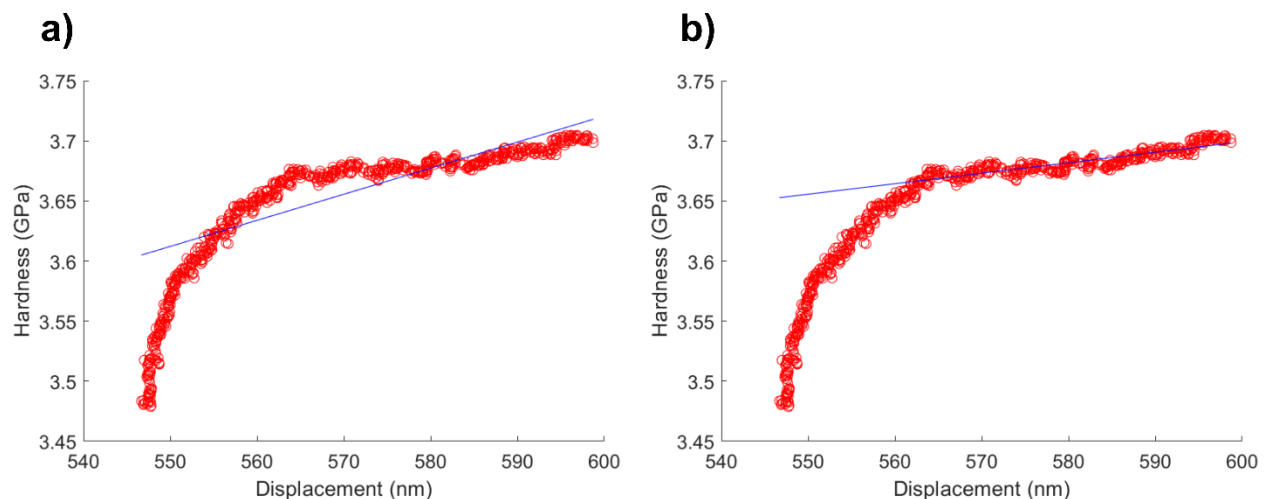


Figure 1: a) improperly tuned DistIn parameters. The regression is using too many of the early points; increase the value of DistIn by moving the slider to the right; b) properly tuned DistIn parameters. The blue regression line captures the steady-state behavior.

Temp [double]: Test temperature in Kelvin. Default is 300 K.

SegmentPlot [logical]: Displays the hardness vs. depth profile for each strain rate in a separate figure, as well as the linear regression results. Used for confirming the DistIn settings. Default is false.

ReprodPlot [logical]: Displays the hardness vs. depth profile averaged across all tests. Includes a shaded area indicating the standard deviation. Default is false.

Binning [logical]: Bins the data into 60 bins per strain rate segment. Default is false.

ExamplePlot [double]: Displays the hardness vs. depth and strain rate vs. depth profiles for the test number given by ExamplePlot. Default is 0, meaning none will be plotted if nothing is passed.

Required Outputs:

VStar [double]: The activation volume in m^3 . This is a JumpNumber by TestNumber matrix with the activation volume from each jump and each test. It is important to note that this code always assumes C^* (the Tabor/constraint factor) to be 3! If it is expected to be different, this output should be scaled accordingly.

SRsens [double]: The strain rate sensitivity. This is a JumpNumber by TestNumber matrix with the strain rate sensitivity from each jump and each test.

Optional Outputs:

Table [cell]: Contains 4 components:

- 1) ExpHard: A matrix containing the extrapolated hardness from each jump and each test, if desired.
- 2) ExpMod: A matrix containing the extrapolated modulus from each jump and each test, if desired.
- 3) Strains: The strain rates used.
- 4) DistIn: How far into each strain rate segment was used for the fitting. Only exports if no DistIn was input.