Documentation for gleeble\_relax\_combiner\_GUI

current version: v9.0

Documentation updated: 23.08.2022

Simple and fast guide

* 1. open GUI from terminal:
  + navigate to directory with the script 🡪 run command *python relax\_reader.py*
* 2. set directories for loading and saving with *<choose load|save dir>*-buttons
  + save directory can be anything
  + load directory must include only the files you want to load. If there are any other files, loading workbooks will fail.
* 3. load workbooks 🡪 *<load workbooks>*
  + modify *variables, data thinning* and *c|r variables* if needed
* 4. adjust compress and relax start times if needed 🡪 *<adjust c|r start>*
  + new window will pop up 🡪 change the starting times (change values 🡪 *<Update tsd>*) until you are satisfied with the result, then move to the next one *<Accept → Next! >*
* 5. draw helping lines and fit SRX equations 🡪 *<Draw lines>*
  + new window will pop up 🡪 Draw Perttula-type lines and fit JMAK and Zurob equations
* 6. save workbooks 🡪 *<save workbooks>*
  + with *checks* you can control what type of data combinations will be saved

|  |
| --- |
|  |
| fig 1. GUI main view |

|  |
| --- |
|  |
| fig 2. c|r adjusting window |

|  |
| --- |
|  |
| fig 3. SRX window |

Detailed documentation

Main window

* Buttons
  + *<Choose load|save dir>*
    - opens a dialog window where user chooses directories for loading and saving
    - HOX! If load dir includes *ts\_params.txt*, test series variables will be overwritten for more convenient test series updating. Example file includes the following lines:
      * *temps;[850,900,950,1000,1050]*
      * *strains;[0.2,0.3,0.4,0.5]*
      * *hold\_times;[200]*
      * *SRs;[10]*
  + *<load workbooks>*
    - Create test series dictionary and load all the data that can be found in the load directory (missing cases don’t raise an error)
  + *<adjust c|r start>*
    - Open separate window with figure of compress and relax start times. If automation hasn’t worked, both can be changed by modifying the value and pressing *<update tsd>*. When ready, continue to the next case with *<Accept → Next!>*
  + *<Draw lines>*
    - Open separate window to analyze SRX. The explanation for it quite long, so it will be explained in detail below.
  + *<save workbooks>*
    - Save workbooks with the chosen checks.
* Variables (Temperature, strain, hold time, strain rate)
  + These affect the test series dictionary python build 🡪 each case is taken into account, even if it does not exist in the loading directory (this allows the user to examine data when the full test series hasn’t been completed)
  + it is important to include the brackets ([]) and use comma as a separator for each case, this is the expected code syntax and when not followed, will cause the loading to crash
* c|r variables
  + controls compress|relax automation variables. *c\_lim* is the value that is compared when trying to find the compression start time and *r\_lim* is the equivalent for relaxation
  + automation logic: look at the normalized data (stress or strain, stress seems more reliable) 🡪 when c|r\_lim values are first encountered in the data list, that point should be the corresponding time.
  + This logic seems to work nicely for relax start, but not so much for compress start. For this reason, the user is given a chance to force the compress time into a single value, *c\_forced\_time*. When this is set to -1, the automation will try to find the best time. When any other value is used, that value will be the initial guess for compress start, unless that time is not found (too short data). In this case, the first time found will be used.
* Data handling
  + Data thinning 🡪 thin out data by choosing only every *n*th datapoint after relaxation starts
    - *Thin\_skip* 
      * Control how many datapoints to skip
      * Takes full compression data regardless of this value
  + Point averaging 🡪 average each point by taking a range of points around it (= moving average?)
    - *pavg\_range*
      * control the full range of averaging.
      * e.g. pavg 15 🡪 the range will be chosen as [i-7 … i+7]
* Checks
  + *all\_data\_pavg|orig\_chk*
    - save a workbook including
      * Variables-worksheet
        + corresponding case for each number
        + compress and relax starts
      * gleeble datas for each case
        + little charts with basic information to check that everything is fine
        + time-based are log10-scaled, strain based normal scale
  + *const\_temp|strain|SR|t\_hold\_chk*
    - save workbook with time-strain data for each value as a constant and other two as sub-values so that only one variable changes in each worksheet
  + *compr|relax\_chk* note! these checks don’t exist anymore, instead the data is automatically saved
    - add separate workbooks, where the same combination workbooks start from the determined compress and relax start times
    - The idea is to compare different strains and strain rates from the same base line
    - All charts are log10-scaled
  + *str-str\_chk*
    - save stress-strain data for all the cases
  + *all\_SRX\_chk*
    - save a general workbook for all SRX data.
      * Variables-worksheet
      * SRX data worksheet for each case

SRX window

* Buttons
  + *<Accept|Skip -> Next!>*
    - either save fitting variables and SRX data (=Accept) or reject (=skip) changes and move to next case
  + *<Update SRX>* 🡪 is this useful anymore?
    - Update only SRX figure
  + *<Update all>*
    - Update figure limits and both figures
  + *<fit JMAK SRX>*
    - Fit JMAK equation with chosen fitting options and the current values as initial values. Note! you can sometimes control the fitting algorithm by testing different initial values.
    - Also update SRX figure
  + *<fit Zurob>*
    - Fit Zurob equation with the current values as initial values.
    - Also update SRX figure
* Combobox
  + allows to swap between each test case
* Perttula line variables
  + *a1, b1* 🡪 used for pert\_rec
  + *a2, b2* 🡪 used for pert\_gg
* checks\_draw
  + *draw\_JMAK|Zurob*
    - choose whether to draw the selected SRX curve
* Perttula-JMAK fitting
  + *t50, k\_JMAK, n*
    - JMAK equation () variables
  + fit options
    - *full* 🡪 use full length data and basic form of JMAK equation
    - *t50\_lock* 🡪use full length data and t50 form of JMAK equation,
    - *20-80\_SRX* 🡪 use data with range 20-80 % SRX
* Zurob fitting
  + Zurob equations too long to open here, see Zurob article, Aarne article
  + *U\_a, V\_a* 🡪 fitting parameters for stress recovery
  + *Q\_d, k\_Z* 🡪 fitting parameters for SRX
  + fitting data interpolation
    - time step needs to be near constant and small enough for Zurob equation to properly work (since it uses numerical differential solving, which can be heavily dependent on timestep). The script makes sure that time step is constant by interpolating between data points when necessary. This virtual fitting data is only used when computing the equation, the output time-step is the same as non-interpolated data for easier compatibility when saving workbooks.
    - *ipol\_lim* 🡪 control maximum time, where the interpolation algorithm operates.