Table of Contents

Introduction

The Sequence of Physical Processes (SPP) framework is a way of interpreting the transient data derived from oscillatory rheological tests. It is designed to allow both the linear and non-linear deformation regimes to be understood within a single unified framework.

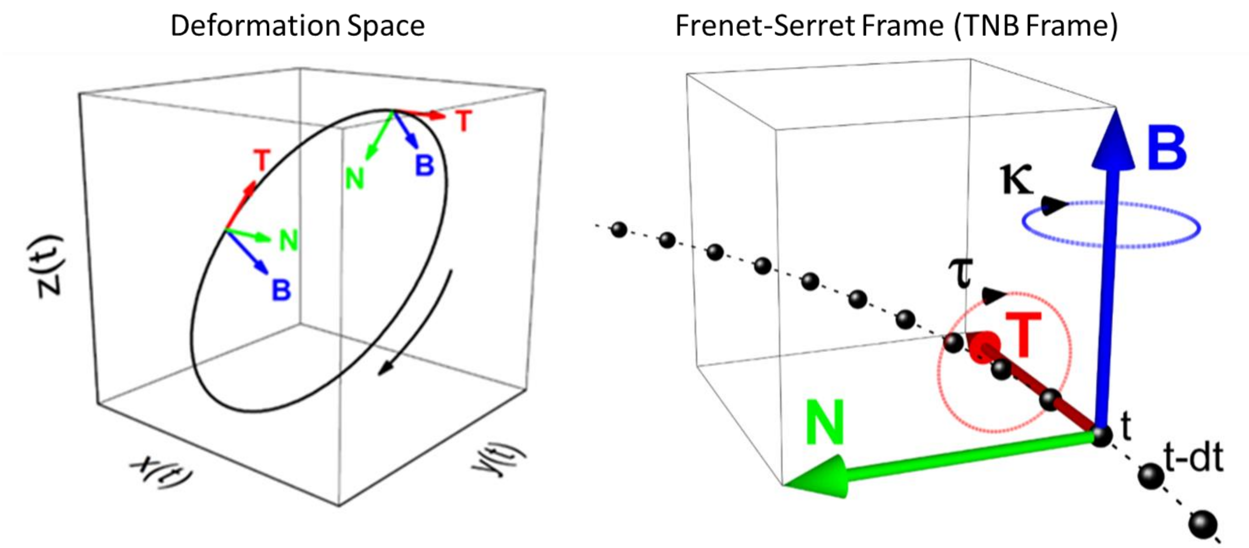
This code provides a convenient way to determine the SPP framework metrics for a given sample of oscillatory data. It will produce a text file containing the SPP metrics, which the user can then plot using their software of choice. It can also produce a second text file with additional derived data (components of tangent, normal, and binormal vectors), as well as pre-plotted figures if so desired.

The Sequence of Physical Processes Approach

A complete explanation of the SPP approach can be found in (Rogers, Rheol Acta, 2017); a brief introduction is provided here for convenience.

The key idea behind the SPP framework is that the dynamic moduli that define viscoelasticity are not required to remain constant throughout a test, and that their values change transiently over the course of a period of oscillation once deformation exits the linear regime.

The SPP framework is able to calculate these instantaneous, time-dependent moduli by utilizing the mathematics of trajectories from differential geometry. To begin, the measured material response is viewed as a trajectory within 3D deformation space. Under strain-controlled conditions, this space consists of strain on the x-axis, strain rate on the y-axis, and stress on the z-axis.



The framework then utilizes the Frenet-Serret Frame (FSF), or Tangent-Normal-Binormal Frame (TNB Frame), to track the differential motion of the trajectory through deformation space. This frame decomposes the motion of the trajectory into three orthonormal vectors: the tangent vector which indicates the current direction of the trajectory, the normal vector which indicates how that direction is changing, and the binormal vector which indicates the current osculating plane of the response. These vectors are then used to determine the various SPP analysis metrics.

What This Code Does

SPPplus is a MATLAB package which consists of a script that acts as a user interface and 5 associated functions which handle the processing and analysis. These are:

RunSPPplus_v2.m ---------------------------------- Script, user interface, sets input file and analysis parameters
SPPplus_read_v2.m --------------- Function, reads/reorganizes/truncates/differentiates data from input file
SPPplus_fourier_v2.m ------------------------------- Function, performs SPP analysis via Fourier domain filtering
SPPplus_numerical_v2.m ------------------------ Function, performs SPP analysis via numerical differentiation
SPPplus_print_v2.m --- Function, prints SPP analysis result files
SPPplus_figure_v2.m ------------------ Function, displays SPP results in figures, can cave figures as image files

All user inputs can be found inside RunSPPplus_v2.m

New in Version 2

Version 2 of SPPplus has several improvements and expansions over the previous versions of the code:

1) The number of ways to import data into the code have been expanded. In addition to the previously supported .txt files, the code can now also import .csv files. Functionality has been added to allow for reorganizing and truncating of the imported data in the code, so that less preprocessing is needed. Additionally, the rate variable can now be differentiated within the import code if the data is oscillatory at steady alternance.
2) The data produced by the code has been expanded. A number of metrics that were not calculated previously in the code have been added, as well as several more recently published metrics. The full set of time-dependent data now calculated by the code is:

Variable	Description	Output	Notes
G'_t	time-dependent storage modulus	standard	
G''_t	time-dependent loss modulus	standard	
$\lvert G^*_t \rvert$	magnitude of time-dependent complex modulus	standard	
$\tan(\delta_t)$	time-dependent tan(delta)	standard	
δ_t	time-dependent phase angle	standard	$3\pi/2 \geq \delta_t > -\pi/2$
σ_d	displacement stress	standard	
$\gamma_{eq,est}$	estimated equilibrium strain	standard	valid if $G'_t \gg G''_t$
$\dot{G}'_t$	derivative of t.d. storage modulus	standard	
$\dot{G}''_t$	derivative of t.d. loss modulus	standard	
$\lvert \dot{G^*_t} \rvert$	speed of t.d. complex modulus	standard	
$\tilde{\dot{\delta}}_t$	normalized phase angle velocity	standard	assumes sinusoidal strain
$\vec{T}$	tangent vector	extended	
$\vec{N}$	normal vector	extended	
$\vec{B}$	binormal vector	extended	

3) The number of figures produced by the code has been increased to display the additional calculated parameters.
4) The export of data has also been improved. Data can now be exported as either a .txt file or as a .mat structure, allowing for more flexibility post-analysis.

Input Data Requirements

In order for the SPPplus code to function properly, the input files containing data to be analyzed have some specific structural requirements. The files must be either tab delimited text (.txt) files, or comma separated variable files (.csv). All data should be laid out in column orientation.

If the file is .txt it must contain exactly 4 columns: Time, Strain, Strain Rate, and Stress. This order is preferred; if the file has a different order these can be reorganized in the SPPplus_read v2.m function by specifying the variable “var_loc” in RunSPPplus_v2.m . Any column headers should be removed from the .txt file prior to analysis to prevent the code from interpreting them as part of the data.

If the file is .csv, it needs to contain at least 4 columns, including Time, Strain, Strain Rate, and Stress. The specific column locations should be specified with "var_loc" in RunSPPplus_v2.m . Headers do not need to be removed from .csv files.

The preferred units for each of the columns are as follows: Time [s], Strain [-] (strain units), Strain Rate [1/s], and Stress [Pa]. If the units in the data differ from these units, the code can perform the appropriate unit conversion(s) if conversion factors are provided in the variable "var_conv". For oscillatory data, an integer number of periods must be covered with an even number of data points per period. The text file to be analyzed must be placed inside the SPPplus_v2 folder, in order for Matlab to properly locate it.

If necessary, the data for analysis can be taken from a portion of the overall data imput. This can be achieved through setting the variable "data_trunc" appropriately (see below).

If the data to be analyzed is oscillatory at steady alternance, SPPplus_read v2.m can differentiate the Rate data from supplied stress data if no rate data is available. In this case, .txt files need to have exactly 3 columns (Time, Strain, and Stress), while .csv files must contain at least these 3 columns.

Within the SPPplus folder, three sample files .txt containing properly formatted input data (0010.txt, 0100.txt, and 1000.txt) are provided for reference. These files contain data from responses of the Giesekus model to different conditions of applied deformation. The model parameters are $\lambda_1 = 1$ s; $\eta_s = 0.01$ Pa s; $\eta_p = 10$ Pa s; $\alpha = 0.3$.

Using the Code

To use SPPplus, open RunSPPplus_v1.m in MATLAB. Upon opening the script, navigate to the section of the file labeled "User-defined variables". (This is found on lines 51-114 of the file.)

The first portion of this section (lines 53-73) contain the data import settings variables. These must be specified for every analysis run. This section appears as follows in the file:

```
53      % DATA IMPORT SETTINGS
54  -   fname = "1000"; % Input data file name (do NOT include extension) (must be
55          %either .txt OR .csv) (Note restrictions on data mentioned above)
56  -   ftype = 1; % What type of file is being imported
57          %use 1 if .txt file
58          %use 2 if .csv file
59  -   var_loc = [1,2,3,4]; % Location of each data column in imported file (Time,
60          %Strain, Rate, Stress). All inputs MUST be positive integers, except
61          %Rate (3rd index), which can be set to zero to indicate that rate needs
62          %to be differentiated from strain. (This differentiation only supports
63          %closed-loop periodic data currently)
64  -   var_conv = [1,1,1,1]; % Conversion factors required to put the data in
65          %the assumed units: Time (s), Strain (-), Rate (1/s), and Stress (Pa),
66          %with each index corresponding the units of the four variables in that
67          %order. Conversion factors will multiply the data as follows:
68          %    (data in file)*(conversion factor)=(data in correct units)
69  -   data_trunc = [0,1,512]; % Determines if/how data will be truncated.
70          %First index denotes whether truncation of input data will occur
71              %if 1, data will be truncated
72          %Second and third indeces denote the 1st and last points to keep when
73          %truncating dataset
```

The variables in this section that the user can set are as follows:

fname: This string specifies the name of the file that contains the data for use in the SPP analysis. The code only reads the file during the extraction of the data; it will not edit the input file in any way. The file must match the requirements listed above in the input data requirements section, or the code will not be able to work properly. The file extension (.txt or .csv) should not be included

ftype: This scalar indicates the file extension for the import file specified by "fname". Accepted values are either 1 (.txt files) or 2 (.csv files).

var_loc: This vector indicates the position of the four data columns used by the SPP analysis in the input file. The indices correspond to the position of the Time, Strain, Strain Rate, and Stress vectors. Each should be set to a different positive integer. If there is no rate column in the input data, and the data is oscillatory at steady alternance, the third index (Strain Rate) should be set to zero; this indicates to the code to differentiate the strain vector by the time vector to get the rate vector.

var_conv: This vector contains the conversion factors required to convert the input data to the units required for the analysis. These units are: Time [s], Strain [-] (i.e. strain units), Rate [1/s], and Stress [Pa]. Each index of the vector corresponds to those quantities in that order. The conversion factors are specified so that they multiply the input data. The factors can be calculated using the following relation:

$$(\text{conversion factor}) = \frac{\text{required units}}{\text{input units}}$$

data_trunc: This vector controls if/how the input data will be truncated for analysis. The first index indicates whether truncation will be performed, entered as either 0 (off) or 1 (on). The second and third indices give the first and last points for the truncated data, respectively, and are only active if the first index is set to 1.

The second portion of this section (lines 75-99) contain the analysis settings variables. The first of these must always be specified, while the remainder only need to be specified for certain analysis types. This section appears as follows in the file:

```
% ANALYSIS SETTINGS
an_use = [1,1]; % Determines what analyses to run.
    %First index denotes whether a fourier series will be used
        %if 1, SPPbasic_fourier_v1 will be run
    %Second index denotes whether a numerical differentiation will be used
        %if 1, SPPbasic_numerical_v1 will be run
omega=3.16; %frequency of oscilation, in units of rad/s
    %For Fourier mode, this must be a single number
    %For Numerical mode, this can either be a single number OR a vector
        %with the same length as the data
% IF FOURIER DOMAIN FILTERING IS USED
M=39; % The number of harmonics to be used in reconstruction of stress.
    %(Must be an odd integer)
p=1; % Total number of periods of measuring time, which we have to know in
    %order to do FT. (Must be a positive integer)
% IF NUMERICAL DIFFERENTIATION IS USED
k=8; % Step size for numerical differentiation, default to be 1. (Must be
    %a positive integer)
num_mode=1; %method of numerical differentiation
    %1 = "standard" (does not make assumtions abot the form of the data,
        %uses forward/backward difference at ends and centered derivative
        %elsewhere)
    %2 = "looped" (assumes steady state oscilatory, uses centered
        %difference everywhere by connecting the data in a loop) (must be a
        %closed periodic curve to utilize)

```

The variables in this section that the user can set are as follows:

an_use: This vector specifies the analysis method(s) that will be used in the current analysis run. The first index indicates whether to use the Fourier domain filtering analysis on the current data. The second index indicates whether to use the numerical differentiation analysis on the current data. Both indices should be entered as either 0 (off) or 1 (on) numbers. At least one of the indices must have a value 1, otherwise no analysis procedure will be run.

omega: The frequency of the oscillation in radians/second. If the mode is set to Fourier, this must be a single value. If the mode is set to numerical, it can either be a single value or a vector the same length as the input data. If there is no known fixed oscillation frequency, enter an estimate.

For the Fourier domain filtering analysis:

M: The number of harmonics used to reconstruct the stress waveform in the SPP analysis. This number must be a positive, odd integer.

p: The total number of periods of measuring time in the input data. This number must be a positive integer.

For the numerical differentiation analysis:

k: The step size for the numerical differentiation. This number must be a positive integer.

num_mode: The procedure used for the numerical differentiation, indicated by a single number. There are currently two possible numerical differentiation procedures implemented in SPP basic. The first one, “standard differentiation” (called if num_mode = 1), makes no assumptions about the form of the data. It utilizes a forward difference to calculate the derivative for the first 2*k points of the data, a backward difference for the final 2*k points, and a centered difference elsewhere. The second, “looped differentiation” (called if num_mode = 2), assumes that the input data is taken under oscillatory conditions at steady alternance, and represents an integer number of periods. These assumptions allow a centered difference to be calculated everywhere by looping over the ends of the data.

The third portion of this section (lines 101-116) contain the output settings variables. These must be specified for every analysis run. This section appears as follows in the file:

```
101       % OUTPUT SETTINGS
102 -     out_type = 1; % What type of file is being exported
103           %use 1 if .txt file
104           %use 2 if .mat file
105 -     is_fsf = 1; % Print full frenet-serret frame data?
106           %if 0, only the standard SPP data file will be produced
107               %(waveforms, time dependent moduli, moduli rates, etc.)
108               %(ex. sample_file1_SPP_LAOS_FOURIER.txt)
109           %if 1, second extended data file will be produced
110               %(includes components of Tangent, Normal, & Binormal vectors)
111               %(ex. sample_file1_SPP_LAOS_FOURIER_FSF.txt)
112 -     save_figs = 1; % Save figures of the SPP metrics ?
113           %if 0, figures will be displayed but not saved
114           %if 1, all figures will be saved as image files
115           %(ex. sample_file1_SPP_LAOS_FOURIER_PLOT.jpg)
116           %note: the exact figures produced will depend on the chosen analyses
117
```

The variables in this section that the user can set are as follows:

out_type: This scalar indicates the file extension for the exported data. Accepted values are either 1 (tab delimited .txt files) or 2 (hierarchical structure .mat files).

is_fsf: This scalar specifies the output data files that will be given to the user for each of the analysis procedures in the current analysis run. The default value is 0, which Under these default conditions, the code will produce a file (ex. fname_SPP_analysistype.txt) containing the standard SPP metrics as determined by a given analysis method. If the value is set to 1, a second data file containing the components of the Frenet-Serret Frame vectors from the analysis is produced (ex. fname_SPP_analysistype_FSFRAME.txt).

save_ figs: This scalar specifies whether to save the figures produced by the analysis as image files, entered as either 0 (off) or 1 (on). The code will always display any figures associated with the current analysis method, it will only save these figures if this is set to 1.

The following table gives an example of the outputs produced by the code, assuming an input file "sample_file1.txt" (could be .csv), and that .txt is selected as the preferred output type for data (could choose .mat):

Analysis type	Fourier	Numerical
Default (always produced)	- sample_file1_SPP_FOURIER.txt	- sample_file1_SPP_NUMERICAL.txt
Extended data on	- sample_file1_SPP_FOURIER_FSFRAME.txt	- sample_file1_SPP_NUMERICAL_FSFRAME.txt
Figure saving on	- sample_file1_SPP_LAOS_FOURIER_PLOT.jpg - sample_file1_SPP_LAOS_FOURIER_SPEED.jpg - sample_file1_SPP_LAOS_FOURIER_MORE.jpg - sample_file1_SPP_LAOS_FOURIER_WAVECHECK.jpg - sample_file1_SPP_LAOS_FOURIER_HARMONICS.jpg	- sample_file1_SPP_LAOS_NUMERICAL_PLOT.jpg - sample_file1_SPP_LAOS_NUMERICAL_SPEED.jpg - sample_file1_SPP_LAOS_NUMERICAL_MORE.jpg -

Example Analysis Run

To demonstrate the SPPplus code in action, we will perform an analysis run on the data file 0100.txt that is included in the SPP basic folder. The data in this text file comes from the Giesikus model ($\lambda_1 = 1$ s, $\eta_s = 0.01$ Pa·s, $\eta_p = 10$ Pa·s, $\alpha = 0.3$) under steady-state, strain-controlled oscillatory conditions ($\omega = 3.16$ rad/s, $\gamma_0 = 1$ strain unit). The data is arranged into four columns: Time (s), Strain (-), Rate (1/s) and Stress (Pa), reflecting the applied strain-control. To analyze the data contained in this file, we enter in the required variables, as seen below:

```
% DATA IMPORT SETTINGS
fname = "0100"; % Input data file name (do NOT include extension) (must be
    %either .txt OR .csv) (Note restrictions on data mentioned above)
ftype = 1; % What type of file is being imported
    %use 1 if .txt file
    %use 2 if .csv file
var_loc = [1,2,3,4]; % Location of each data column in imported file (Time,
    %Strain, Rate, Stress). All inputs MUST be positive integers, except
    %Rate (3rd index), which can be set to zero to indicate that rate needs
    %to be differentiated from strain. (This differentiation only supports
    %closed-loop periodic data currently)
var_conv = [1,1,1,1]; % Conversion factors required to put the data in
    %the assumed units: Time (s), Strain (-), Rate (1/s), and Stress (Pa),
    %with each index corresponding the units of the four variables in that
    %order. Conversion factors will multiply the data as follows:
    %   (data in file)*(conversion factor)=(data in correct units)
data_trunc = [0,1,512]; % Determines if/how data will be truncated.
    %First index denotes whether truncation of input data will occur
        %if 1, data will be truncated
    %Second and third indeces denote the 1st and last points to keep when
    %truncating dataset
```

We enter in fname = ”0100” and ftype = 1 based on the file name and extension of 0100.txt . As the data is also formatted correctly in terms of column order and units, we enter var_loc = [1,2,3,4] and run_conv = [1,1,1,1]. As no truncation is needed, we set data_trunc(1) = 0 (the other indices only matter if the first is set to 1).

```
% ANALYSIS SETTINGS
an_use = [1,1]; % Determines what analyses to run.
    %First index denotes whether a fourier series will be used
        %if 1, SPPbasic_fourier_v1 will be run
    %Second index denotes whether a numerical differentiation will be used
        %if 1, SPPbasic_numerical_v1 will be run
omega=3.16; %frequency of oscilation, in units of rad/s
    %For Fourier mode, this must be a single number
    %For Numerical mode, this can either be a single number OR a vector
        %with the same length as the data
% IF FOURIER DOMAIN FILTERING IS USED
M=39; % The number of harmonics to be used in reconstruction of stress.
    %(Must be an odd integer)
p=1; % Total number of periods of measuring time, which we have to know in
    %order to do FT. (Must be a positive integer)
% IF NUMERICAL DIFFERENTIATION IS USED
k=8; % Step size for numerical differentiation, default to be 1. (Must be
    %a positive integer)
num_mode=1; %method of numerical differentiation
    %1 = "standard" (does not make assumtions abot the form of the data,
        %uses forward/backward difference at ends and centered derivative
        %elsewhere)
    %2 = "looped" (assumes steady state oscilatory, uses centered
        %difference everywhere by connecting the data in a loop) (must be a
        %closed periodic curve to utilize)

```

We want to analyze the data via both fourier series and numerical differentiation, so we enter run_state = [1,1]. Because of the conditions that the data was taken from, we enter in omega = 3.16 and p = 1. Based on the noise floor for the data, we set M = 39. In order to achieve a smooth numerical differentiation, we set k = 8, and num_mode = 2.

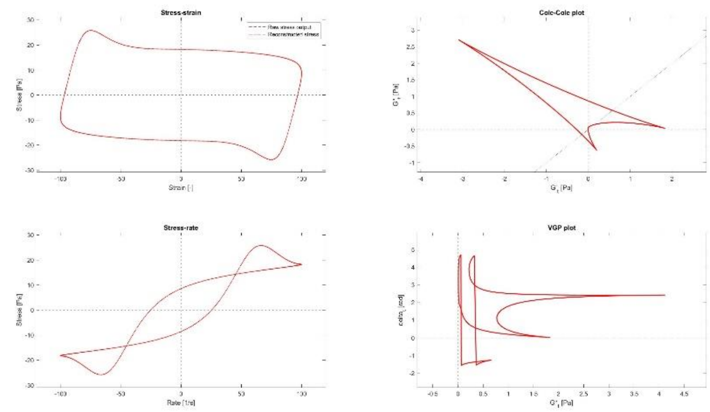
```
% OUTPUT SETTINGS
out_type = 1; % What type of file is being exported
    %use 1 if .txt file
    %use 2 if .mat file
is_fsf = 1; % Print full frenet-serret frame data?
    %if 0, only the standard SPP data file will be produced
        %(waveforms, time dependent moduli, moduli rates, etc.)
        %(ex. sample_file1_SPP_LAOS_FOURIER.txt)
    %if 1, second extended data file will be produced
        %(includes components of Tangent, Normal, & Binormal vectors)
        %(ex. sample_file1_SPP_LAOS_FOURIER_FSF.txt)
save_figs = 1; % Save figures of the SPP metrics ?
    %if 0, figures will be displayed but not saved
    %if 1, all figures will be saved as image files
    %(ex. sample_file1_SPP_LAOS_FOURIER_PLOT.jpg)
    %note: the exact figures produced will depend on the chosen analyses

```

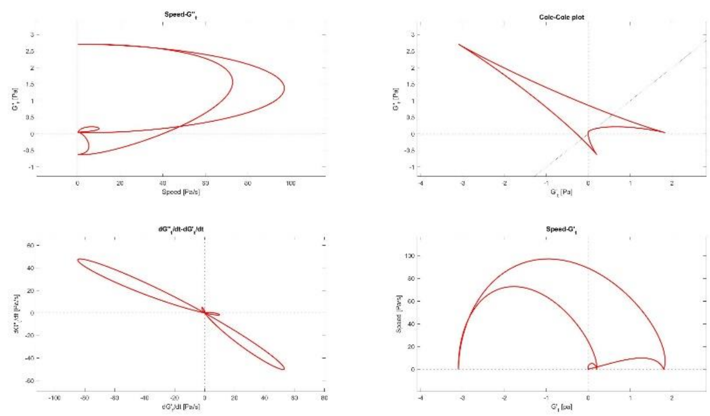
We would like to export text files, so we set out_type = 1. We also want to get all optional files, so we enter is_fsf = 1 and save_figs = 1.

Once we press run, the code runs both analyses, generating files listed in the table on page 7 from the results. The 8 figures produced by the code appear as follows:

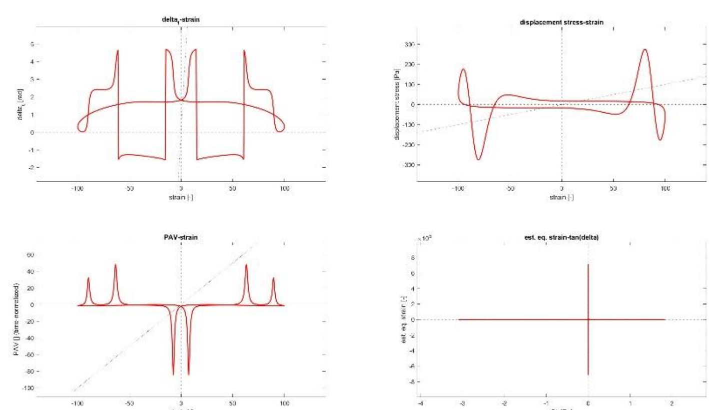
1) The SPP metrics plots from the Fourier analysis:



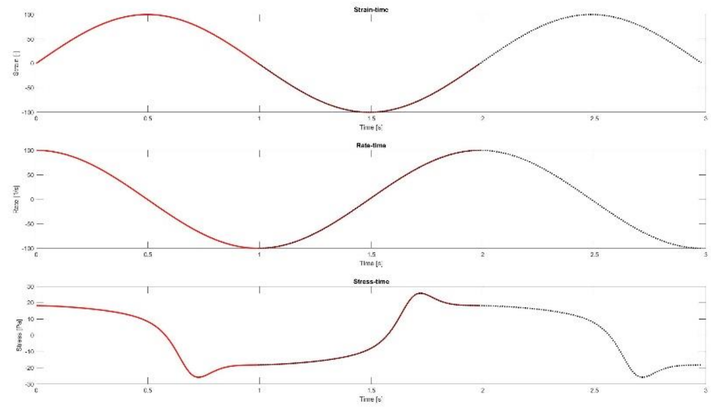
2) The derivatives of the SPP metrics plots from the Fourier analysis:



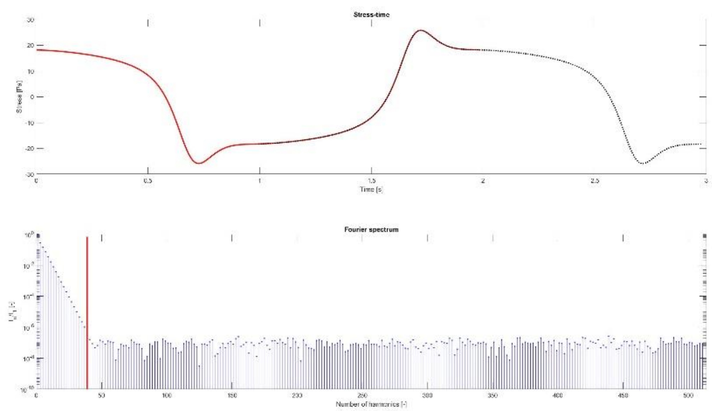
3) The additional SPP metrics plots from the Fourier analysis:



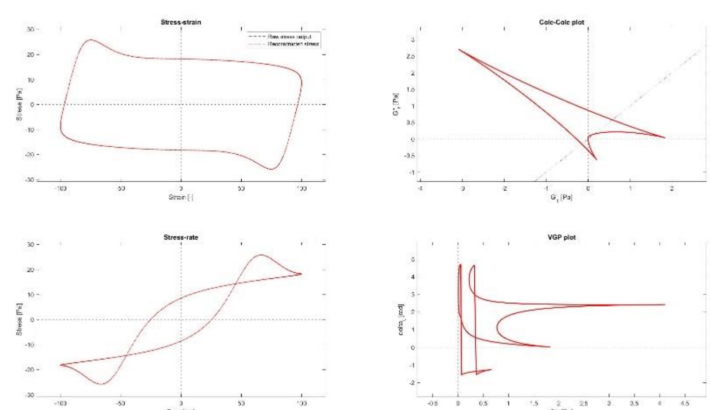
4) The reconstructed waveform comparison plot from the Fourier analysis:



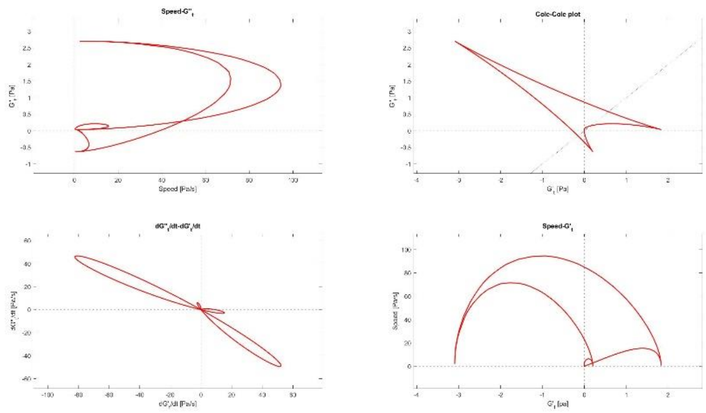
5) The plot of harmonics for the stress response form the Fourier analysis:



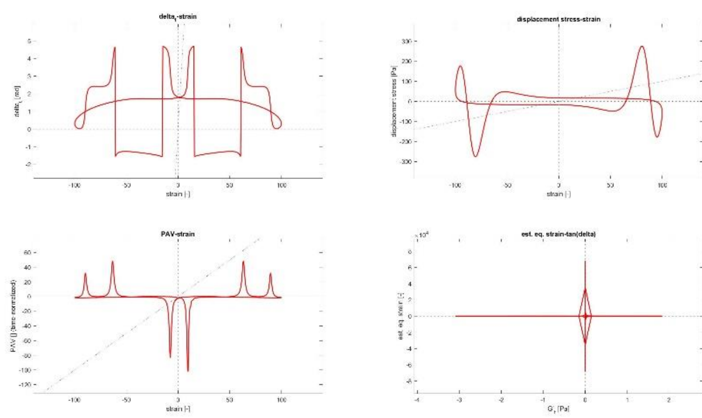
6) The SPP metrics plots from the numerical analysis:



7) The derivatives of the SPP metrics plots from the numerical analysis:



8) The additional SPP metrics plots from the numerical analysis:



Theory

The core theoretical features behind the Sequence of Physical Processes Framework will now be presented. A full derivation and discussion of the metrics discussed here can be found in Rogers (2017).

Within the fully quantitative SPP approach, a material's response to oscillatory shearing produces a trajectory in deformation space given by:

$$\boldsymbol{A} = [A_\gamma \quad A_{\dot{\gamma}/\omega} \quad A_\sigma] = [\gamma_0 \sin(\omega t) \quad \gamma_0 \cos(\omega t) \quad \sigma(t)] = [\gamma(t) \quad \dot{\gamma}(t)/\omega \quad \sigma(t)] \qquad (1)$$

The trajectory can be described by a set of three orthonormal vectors called the tangent ($\boldsymbol{T}$), principal normal ($\boldsymbol{N}$), and binormal ($\boldsymbol{B}$), vectors. The tangent vector points in the direction of instantaneous motion, and the principal normal points in the direction of the derivative of the tangent:

$$\boldsymbol{T} = \frac{\dot{\boldsymbol{A}}}{\|\boldsymbol{A}\|}, \tag{2}$$

$$\boldsymbol{N} = \frac{\dot{\boldsymbol{T}}}{\|\boldsymbol{T}\|}. \tag{3}$$

The tangent and principal normal vectors span the osculating plane, which can be thought of as the plane in which the trajectory sits on a local scale. The binormal vector is given by the vector cross product of the tangent and principal normal vectors and therefore defines the orientation of the osculating plane:

$$\boldsymbol{B} = \boldsymbol{T} \times \boldsymbol{N}. \tag{4}$$

The SPP framework defines two transient moduli (${G_t}'$ and ${G_t}''$) which are differential parameters that represent the orientation of the trajectory in deformation space. As discussed in Rogers (2017), a complete description of any trajectory requires information regarding the position of the osculating plane as well as the plane's orientation. We thus seek an equation of the form:

$$\sigma = {G_t}'\gamma + {G_t}''\dot{\gamma}/\omega + \sigma^d, \tag{5}$$

where the transient moduli (${G_t}'$ and ${G_t}''$) represent the orientation of the osculating plane, and the displacement stress (σ^d) represents its position in deformation space.

To determine the required form of the displacement stress, we start with the point-normal form of the equation of a plane, as the binormal vector is normal to the osculating plane by definition:

$$B_x(x - A_x) + B_y(y - A_y) + B_z(z - A_z) = 0. \tag{6}$$

We choose to determine the position of the osculating plane along the stress axis, and so we set the x and y-components of the plane to zero and solve:

$$\sigma^d = \frac{B_\gamma}{B_\sigma}\gamma + \frac{B_{\dot{\gamma}/\omega}}{B_\sigma}\dot{\gamma}/\omega + \sigma. \tag{7}$$

Substituting this form of the displacement stress back into equation 5 leads to a description of the trajectory that may be rearranged as follows:

$$\left({G_t}' + \frac{B_\gamma}{B_\sigma}\right)\gamma + \left({G_t}'' + \frac{B_{\dot{\gamma}/\omega}}{B_\sigma}\right)\dot{\gamma}/\omega = 0. \tag{8}$$

On the basis of equation 8, we will therefore define the transient moduli as:

$${G_t}'(t) = -\frac{B_\gamma(t)}{B_\sigma(t)}, \tag{9}$$

$${G_t}''(t) = \frac{B_{\dot{\gamma}/\omega}(t)}{B_\sigma(t)}. \tag{10}$$

In addition to defining transient moduli, the SPP framework also provides explicit definitions of their derivatives, which can be used to tell us not only whether the response is softening, stiffening, thickening, or thinning, but also when and by how much. The derivatives of the transient moduli have slightly more complex forms than the transient moduli themselves, and require the principal normal vector, the binormal vector, and the torsion ($\tau = -\|\dot{\boldsymbol{A}}\|\boldsymbol{N}\cdot\boldsymbol{B}$) which geometrically tells us how fast the osculating plane rotates around the axis given by the tangent vector:

$$\dot{G_t'} = \tau\|\dot{\boldsymbol{A}}\|\left(\frac{N_\gamma}{B_\sigma} + \frac{B_\gamma N_\sigma}{{B_\sigma}^2}\right), \tag{11}$$

$$\dot{G_t''} = \tau\|\dot{\boldsymbol{A}}\|\left(\frac{N_{\dot{\gamma}/\omega}}{B_\sigma} + \frac{B_{\dot{\gamma}/\omega} N_\sigma}{{B_\sigma}^2}\right). \tag{12}$$

In addition to defining the time-dependent moduli and their derivatives, the SPP framework is unique among oscillatory shear analysis techniques in that it allows for unrecoverable strain via the inclusion of a moving strain equilibrium position, and a yield stress that is not represented by the moduli. While the orientation of the osculating plane gives information regarding the local moduli, it is its displacement (σ^d) that contains information about the strain equilibrium position (γ_{eq}) and the yield stress (σ_y). The displacement stress, defined by equation 7, is physically interpreted as being equal to

$$\sigma^d(t) = \sigma_y(t) - {G_t}'(t)\gamma_{eq}(t) = \sigma(t) - ({G_t}'(t)\gamma(t) + {G_t}''(t)\dot{\gamma}(t)/\omega). \tag{13}$$

When the response is predominantly elastic (eg. ${G_t}'(t) \gg {G_t}''(t)$), this equality can be simplified to

$$\gamma_{eq}(t) = \gamma(t) - \frac{\sigma(t)}{{G_t}'(t)}, \tag{14}$$

allowing for a straightforward determination of the equilibrium position, and therefore also the recoverable and unrecoverable components of the strain.

Contact Info and References

If you have questions regarding the code, please email Dr. Simon Rogers at sarogers@illinois.edu .

The following papers can be used as reference for a thorough discussion and historical development of the Sequence of Physical Processes framework:

J. Choi. F. Netteshiem. S. A. Rogers. “The unification of disparate rheological measures in oscillatory shearing”, Phys. Fluids, 31 (2019), pp.073107 (2019)

G. J. Donley. J. R. deBruyn. G. H. McKinley. S. A. Rogers. “Time-resolved dynamics of the yielding transition in soft materials”, J. Non-Newtonian Fluid Mech. Volume 264 (2019), Pages 117-134

S. A. Rogers. “In search of physical meaning: defining transient parameters for nonlinear viscoelasticity” Rheol. Acta, 5,56 (2017), pp. 501-525

S. A. Rogers. “A sequence of physical processes determined and quantified in LAOS: An instantaneous local 2D/3D approach” J. Rheol. 56 (2012) p. 1129-1151

S. A. Rogers. M. P. Lettinga. “A sequence of physical processes determined and quantified in large-amplitude oscillatory shear (LAOS): Application to theoretical nonlinear models” J. Rheol. 56 (2012) p. 1-25

S .A. Rogers. B. M. Erwin. D. Vlassopolous. M. Cloitre. “A sequence of physical processes determined and quantified in LAOS: Application to a yield stress fluid” J. Rheol. 55 (2011) p. 435-458

S. A. Rogers. B. M. Erwin. D. Vlassopolous. M. Cloitre. “Oscilatory yielding of a colloidal star glass” J. Rheol. 55 (2011) p. 733-752

Patch Notes

Version 2.1:
- The previous version of the code incorrectly gave η'_t instead of G''_t, due to an improper conversion. This has now been fixed.
- The new version of the code now requires frequency to be user defined for both Fourier AND numerical modes. (Previous version only required frequency for Fourier mode.)
- For numerical mode, the frequency can be specified as either a single value OR as a vector of length L (where L is the length of the dataset). This should allow for variable frequency tests to be analyzed properly.
- If performing a test where no frequency can be defined, the prior version of the code can be used, as long as it is understood that the code is giving η'_t not G''_t. Some of the derived parameters will also be slightly changed as a result, and may no longer retain the same meaning. It can also be used for $\omega = 1$, as the two values are equal under those conditions, and all parameters will therefore retain meaning.
