Understanding Physical Aging and its Mechanical Characterization

Luigi Casagrande[1]⋆
[1] *Shanghai Jiao Tong University, Materials Science and Engineering, Shanghai, China*

12 December 2024

Abstract
This study investigates the mechanical behavior of glassy polymers under physical aging by characterizing stress relaxation modulus and creep compliance. Aging shift factors and rates were extracted using RMS error minimization to align aging curves to a reference curve. The stress relaxation modulus data were fitted to quadratic polynomials, and viscoelastic theory was applied to compute creep compliance via Laplace transforms. Both isothermal and nonisothermal aging models were analyzed to predict long-term creep behavior.
However, the effective time theory, based on experimental data, was not able to mathematically capture the behavior under nonisothermal conditions correctly due to limitations in the shift factor decay model. Results showed that aging shift factors effectively describe the material's time-dependent behavior, with isothermal models providing robust predictions. These findings enhance the understanding of physical aging and its impact on the viscoelastic properties of polymers, offering insights for long-term material performance in engineering applications.

Key words: aging shift factor – creep – long-term creep – effective time – nonisothermal physical aging

1 Introduction

Physical aging profoundly impacts the mechanical behavior of many engineering materials, including metallic glasses, composites, and polymers. This phenomenon is particularly critical in amorphous polymers, where the material transitions between distinct states depending on its thermal history. At temperatures above the glass transition temperature (T_g), amorphous polymers exist in a rubbery state, characterized by sufficient molecular mobility to enable rapid attainment of thermodynamic equilibrium after a temperature change [1].

However, when cooled below T_g, these materials enter a **non-equilibrium glassy state**. In this state, molecular rearrangements slow dramatically, and the material evolves gradually toward equilibrium over time. The duration of this evolution depends on how far the material's temperature lies below T_g, with the equilibration process becoming slower as the temperature decreases further. During this process, the material undergoes time-dependent changes in various properties, including specific volume, enthalpy, mechanical characteristics, and dielectric response.

The term **physical aging** was introduced to describe this phenomenon, initially identified by Kovacs in 1963 and later extensively studied by Struik in 1978 in the context of glassy polymers. The prefix *"physical"* differentiates this reversible process from chemical degradation or biological aging, both of which involve irreversible changes in material composition or structure.

Notably, physical aging can be erased entirely through a process called **rejuvenation**, wherein heating the material above T_g restores it to its initial equilibrium state. This reversible nature makes physical aging a significant subject of study in materials science, as it directly influences the mechanical stability and reliability of engineering components.

Understanding the influence of physical aging on the mechanical properties of materials requires precise characterization, typically achieved through carefully designed mechanical tests. These tests assess how mechanical behavior evolves with aging time, thermal history, and applied stress or strain conditions. Two fundamental techniques employed for this purpose are **creep tests** and **stress relaxation tests**, each offering unique insights into the time-dependent viscoelastic properties of materials.

1.1 Creep Tests

Creep tests involve subjecting the material to a constant stress (σ_0) during a *"load"* step, followed by a *"unload"* step where the applied stress is reduced to zero. During both phases, the strain response is continuously monitored, allowing for the extraction of viscoelastic properties at discrete aging times. To ensure the aging state remains effectively constant during testing, the duration of each load step is kept much shorter than the elapsed aging time.

The strain contribution during a given load step is determined by subtracting the extrapolated strain from the previous unload step ($\varepsilon_{\text{unload}}$) from the measured strain (ε). The momentary tensile compliance, $D(t)$, is then defined as:

$$D(t) = \frac{\varepsilon(t) - \varepsilon_{\text{unload}}(t)}{\sigma_0} \qquad , \qquad t = t_e - t_{ei}$$

where t is the time elapsed since the start of the load step, and t_e denotes the aging time at the onset of the load step.

This compliance data is often modeled using the shifted three-

⋆ luigi.casagrande@sjtu.edu.cn (J12405099002)

parameter **Kohlrausch function**:

$$D(t) = D_0 \cdot \exp\left[\left(\frac{t}{\tau}\right)^{\beta}\right]$$

where D_0 represents the initial compliance, τ is the relaxation time, and β is the shape parameter indicative of the distribution of relaxation times. The Kohlrausch function effectively captures the broad spectrum of relaxation dynamics characteristic of glassy materials.

1.2 Stress Relaxation Tests

Stress relaxation tests provide a complementary approach to creep tests, involving the application of a constant strain (ε_0) during each load step. The stress response is monitored over time, typically decreasing during load steps and increasing during unload steps. A critical distinction between stress relaxation and creep tests is that, during the unload step, the strain is not reduced to zero. This introduces compressive stresses, which can lead to specimen buckling if not carefully managed.

The momentary modulus, $E(t)$, is defined analogously to compliance in creep tests:

$$E(t) = \frac{\sigma(t) - \sigma_{\text{unload}}(t)}{\varepsilon_0}$$

where $\sigma(t)$ is the stress during the load step, $\sigma_{\text{unload}}(t)$ is the extrapolated stress from the previous unload step, and ε_0 is the strain responsible for the stress variation.

Similar to compliance, the modulus is often modeled using the Kohlrausch function:

$$E(t) = E_0 \cdot \exp\left[-\left(\frac{t}{\tau}\right)^{\beta}\right]$$

where E_0, τ, and β represent the initial modulus, relaxation time, and shape parameter, respectively.

Both test methods are performed under conditions designed to minimize aging effects during the individual load steps, ensuring that the observed responses reflect the material's aging state at specific moments.

Collectively, these techniques enable the quantitative analysis of physical aging effects, providing valuable insights into the viscoelastic behavior, relaxation dynamics, and time-dependent mechanical stability of glassy materials.

1.3 Effective Time Theory

Effective time theory provides a comprehensive framework for analyzing the time-dependent mechanical behavior of materials undergoing physical aging, particularly in the context of isothermal and non-isothermal conditions [2].

This theory introduces the concept of **effective time** (λ), which accounts for the cumulative effects of aging and enables the prediction of long-term mechanical responses, such as creep or stress relaxation, beyond the constraints of Momentary Master Curve (MMC) measurements.

In an isothermal creep test, the material begins aging at an initial aging time t_{e0}, defined as the time elapsed since the material was quenched below the glass transition temperature (T_g). At any later moment during the test, the aging time is given by $t_e = t_{e0} + t$, where t is the elapsed time from the onset of the load step. As t approaches or exceeds t_{e0}, deviations in the material's long-term response from its momentary behavior become apparent due to the evolving aging state.

To capture this evolution, the reference aging time

$$t_{e,\text{ref}} = t_{e0}$$

is used to define the MMC, and a shift factor $a_e(t)$ is introduced to account for changes in the aging state. The shift factor is expressed as:

$$a_e(t) = \left(\frac{t_{e0}}{t_{e0} + t}\right)^{\mu}$$

where μ is the aging shift rate, reflecting the rate at which material properties change with aging.

The effective time increment $\mathrm{d}\lambda$ for a real-time increment $\mathrm{d}t$ is then given by:

$$\mathrm{d}\lambda = a_e(t)\,\mathrm{d}t$$

Integrating this expression over the total testing time yields the total effective time $\lambda(t)$:

$$\lambda(t) = \int_0^t a_e(\xi)\,\mathrm{d}\xi$$

Replacing the real time t with the effective time $\lambda(t)$ in the Kohlrausch model for compliance allows the long-term creep response to be described as:

$$D(t) = D_0 \cdot \exp\left[\left(\frac{\lambda(t)}{\tau}\right)^{\beta}\right]$$

where D_0 is the initial compliance.

1.3.1 Non-Isothermal Aging and Shift Rate

Non-isothermal aging introduces additional complexity, as the aging shift rate $\mu^*(\xi)$ becomes time-dependent due to temperature variations. In such cases, the shift factor $a_e(t)$ deviates from a simple log-linear relationship, and $\mu^*(\xi)$ must account for the effects of temperature history.

The effective time for non-isothermal aging, $\lambda^*(t)$, is determined by:

$$\lambda^*(t) = \int_0^t a_e(\xi)\,\mathrm{d}\xi$$

with:

$$a_e(\xi) = \left(\frac{t_{e0}}{t_{e0} + \xi}\right)^{\mu^*(\xi)}$$

The shift rate $\mu^*(t)$ is defined as:

$$\mu^*(t) = -\frac{\mathrm{d}\log a_e(t)}{\mathrm{d}\log t_e}$$

where $t_e = t_{e0} + t$ represents the aging time since the last temperature jump.

Over long durations, $\mu^*(t)$ asymptotically approaches the isothermal shift rate μ_{iso}, demonstrating the "fading memory" effect, wherein the material's behavior gradually reflects only the current temperature conditions.

2 Method

2.1 Data Extraction

The graphs given represent the creep compliance, stress relaxation modulus, and shift factors derived from short-term creep tests, each of which provides valuable insights into the material's response to thermal aging.

The MATLAB code (referred to the compliance graph at 110°C) presented in the Appendix A outlines the process of extracting the raw data, organize it and replot it again for better clarification.

The figures resulting from the MATLAB code, that will be identical as the one provided, are included in Section 3.1 of the Results.

2.2 Characterization of Isothermal Physical Aging

To investigate the material's behavior at different aging times and temperatures, the aging shift factors and aging shift rates were extracted by analyzing the stress-strain data provided in Section 2.1.

The primary objective was to identify the horizontal shift required to align the various aging curves with the reference curve, thereby enabling a consistent comparison.

Firstly, the data is loaded from the Excel file obtained from the script presented in the Appendix A. Each curve corresponds to a different aging time. The data is then processed to extract the logarithmic values of time and stress for each aging time.

```
x_data = zeros(limit_value,7);
y_data = zeros(limit_value,7);

%% Load DataSet
Dataset = '>..directory/Data110C.xlsx';
data = readtable(Dataset,'ReadVariableNames',false);

num_curves = width(data) / 2;  % Number of curves

for i = 1:num_curves
    x_data(:,i) = log10(data{1:limit_value, 2*i-1});
    y_data(:,i) = log10(data{1:limit_value, 2*i});
end
```

To determine the aging shift factors, the horizontal and vertical shifts are computed by minimizing the RMS error between the reference curve and the other test curves.
The reference curve is taken as the first curve in the dataset (with the lowest aging time). The RMS error function is defined later for horizontal and vertical shifts, and the 'fminsearch' function is used to minimize the error and find the optimal shift values.

```
%% RMS Error Minimization
a_T_o_store = zeros(1,num_curves-1);
a_T_v_store = zeros(1,num_curves-1);
log_t_e = log10([5/8 5/4 5/2 5 10 20]');

options = optimset('MaxFunEvals', 10000, 'MaxIter', 10000);

% Reference curve (first one)
x_ref = x_data(:,1);
y_ref = y_data(:,1);

% Shift Factor Calculation
for j = 2:num_curves
    % Test curve
    x_test = x_data(:,j);
    y_test = y_data(:,j);

    % Definition of RMS error function
    rms_error_o = @(a_T_o) compute_rms_error_o(a_T_o, x_ref,
         x_test);
    rms_error_v = @(a_T_v) compute_rms_error_v(a_T_v, y_ref,
         y_test);

    % Initial guess for a_T
    initial_guess = 5;

    % Minimize the RMS error with custom options
    optimal_a_T_horizontal = fminsearch(rms_error_o,
         initial_guess, options);
    optimal_a_T_vertical = fminsearch(rms_error_v,
         initial_guess, options);
    a_T_o_store(:,j-1) = optimal_a_T_horizontal;
    a_T_v_store(:,j-1) = optimal_a_T_vertical;

    % Output
    disp(['Optimal a_T between reference curve 1 and curve ',
         num2str(j),' is: ', num2str(optimal_a_T_horizontal)])
         ;
end
```

For horizontal shifts a_T°, the shifted test curve

$$T_j' = (x_{\text{shifted},j,\text{test},j})$$

is given by:

$$x_{\text{shifted},j} = x_{\text{test},j} + a_T^{\circ}$$

The RMS error for horizontal alignment is defined as:

$$\text{RMS}_{\text{o}}\left(a_T^{\circ}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[x_{\text{ref}}(i) - x_{\text{shifted},j}(i)\right]^2}$$

In the same way, for the vertical shift:

$$T_j' = (x_{\text{test},j}, y_{\text{shifted},j})$$

$$y_{\text{shifted},j} = y_{\text{test},j} + a_T^{\vee}$$

$$\text{RMS}_{\text{v}}\left(a_T^{\vee}\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[y_{\text{ref}}(i) - y_{\text{shifted},j}(i)\right]^2}$$

The optimal shift factors are determined by minimizing the respective RMS errors:

$$a_T^{\circ} = \min_{a_T^{\circ}} \text{RMS}_{\text{o}}\left(a_T^{\circ}\right)$$

$$a_T^{\vee} = \min_{a_T^{\vee}} \text{RMS}_{\text{v}}\left(a_T^{\vee}\right)$$

This is achieved numerically using the **Nelder-Mead optimization algorithm**, implemented in MATLAB as '`fminsearch`'.

After computing the aging shift factors, the shift rate is determined by fitting the horizontal shift factors ('`a_T_o_store`') to a linear regression model using '`poly1`' (a linear polynomial). The negative slope of the regression provides the shift rate for each curve.

```
% Fit data to a linear regression model
regression = fit(log_t_e, (a_T_o_store)', 'poly1');
shift_rate = - regression.p1 * ones(length(log_t_e));
```

Three figures are generated to visualize the results (Section 3.2). The first figure shows the shifted test curves, with the reference curve and shifted curves overlaid. The second figure shows the aging shift factors as a function of time, and the third figure illustrates the shift rate.

```
%% Figure 1 - Shifted Curves
figure(1);
hold on;
set(gcf, 'Position', [10, 220, 700, 500]);

% Plot - Reference curve
plot(x_data(:,1), y_data(:,1), '*r', 'DisplayName', 'Reference
      Curve', ...
    'MarkerSize', 8, 'LineWidth', 1);

% Plot - Shifted curves
for k = 1:(num_curves-1)
    % Plot the shifted test curves
    plot(x_data(:,k+1) - a_T_o_store(1,k), y_data(:,k+1) -
         a_T_v_store(1,k), ...
            'LineWidth', 1.5, 'DisplayName', ['Shifted Curve '
                 , num2str(k+1)]);
end

xlabel('log(t)');
ylabel('Log Stress');
title_srt = sprintf('Shifted Data (%s)', title_setup);
title(title_srt);
legend('show', 'Location', 'best');
grid on;
hold off;

%% Figure 2 - a_T_o_store vs log_t_e
figure(2);
hold on;
set(gcf, 'Position', [400, 320, 600, 430]);

plot(log_t_e, a_T_o_store, 'k.', 'LineWidth', 1.5, 'MarkerSize
     ', 20, ...
    'DisplayName', 'Shift Factors (Horizontal)');

title_srt2 = sprintf('Shift Factors (%s)', title_setup);
title(title_srt2);
xlabel('log(t)');
ylabel('Shift Factors (a_T)');
grid on;
hold off;

%% Figure 3 - shift_rate vs log_t_e
figure(3);
hold on;
set(gcf, 'Position', [900, 320, 600, 430]);

plot(log_t_e, shift_rate, 'g.', 'LineWidth', 1.5, 'MarkerSize'
     , 20, ...
    'DisplayName', 'Shift Rate (Horizontal)');

title_srt3 = sprintf('Shift Rates (%s)', title_setup);
title(title_srt3);
xlabel('log(t)');
ylabel('Shift Rate');
grid on;
hold off;
```

As discussed previously, the RMS error function is defined at the end of the script as follows:

```
%% Function definition
function error = compute_rms_error_v(a_T_v,y_ref,y_test)
    % Shift x_data by a_T
    shifted_y = y_ref+a_T_v;

    % Compute the RMS error
        error = sqrt(mean((shifted_y - y_test).^2));
end
function error = compute_rms_error_o(a_T_o,x_ref,x_test)
    % Shift x_data by a_T
    shifted_x = x_ref + a_T_o;

    % Compute the RMS error
    error = sqrt(mean((shifted_x - x_test).^2));
end
```

2.3 Relationship between Creep and Stress Relaxation

The goal of this script is to calculate the theoretical creep compliance $D(t)$ from experimentally measured stress relaxation modulus $E(t)$ using a mathematical relationship derived from viscoelastic theory.

The script simply starts by clearing the workspace, initializing variables and data loading.

```
%% Luigi Casagrande
clc;
clear;
close all;

limit_value = 20;
title_setup = '120°C';

x_data = zeros(limit_value,7);
y_data = zeros(limit_value,7);
x_data_c = zeros(limit_value,7);
y_data_c = zeros(limit_value,7);
D_t=zeros(limit_value,7);

%% Load DataSet
Dataset_C = '>..directory/Data120C.xlsx';
data_compliance = readtable(Dataset_C,'ReadVariableNames',
     false);
Dataset_M = '>..directory/Data120M.xlsx';
data_modulus = readtable(Dataset_M,'ReadVariableNames',false);

num_curves = width(data_modulus) / 2;  % Number of curves
coeffs=zeros(3,num_curves);

for i = 1:num_curves
    x_data(:,i) = data_modulus{1:limit_value, 2*i-1};
    y_data(:,i) = data_modulus{1:limit_value, 2*i};
end
for i = 1:num_curves
    x_data_c(:,i) = data_compliance{1:limit_value, 2*i-1};
    y_data_c(:,i) = data_compliance{1:limit_value, 2*i};
end
```

The stress relaxation modulus data for each curve is fitted to a quadratic polynomial:

$$E(t) = p_1 t^2 + p_2 t + p_3$$

This is done using MATLAB's '`fit()`' function with a second-order polynomial ('`poly2`'). The coefficients p_1, p_2, and p_3 are stored in the matrix '`coeffs`'.

```
x_in = x_data;
y_in = y_data;

%% Fit the data
for j = 1:num_curves
    x_cf = x_in(:,j);
    y_cf = y_in(:,j);
    regression = fit(x_cf,y_cf,'poly2');
    coeffs(:,j) = [regression.p1,regression.p2,regression.p3];
end
```

Using viscoelastic theory, the Laplace transform relationship between modulus and compliance is:

$$D(s) = \frac{1}{s^2 \cdot E(s)}$$

where $E(s)$ is the Laplace transform of $E(t)$. For each curve, the modulus $E(t)$ is defined symbolically using the polynomial coefficients. The script computes the Laplace transform $E(s)$, derives $D(s)$ using the above formula, and applies the inverse Laplace transform to find $D(t)$.

```
% Laplace transformation
for k = 1:num_curves
    syms x s

    % Define the creep compliance polynomial function
    creep = coeffs(1,k)*x.^2 + coeffs(2,k)*x + coeffs(3,k);

    % Laplace transform of creep
    E_laplace = laplace(creep,x,s);

    % Inverse Laplace relationship
    D_laplace = (1/((s.^2)*E_laplace));
    D_t_sym = ilaplace(D_laplace,s,x);

    % Substitute x_in values into D(t) to get the creep
         compliance over time
    D_t_vals = double(subs(D_t_sym,x,x_in(:,k)));
    D_t(:,k) = D_t_vals;
end
```

The computed creep compliance $D(t)$ is compared with experimental creep compliance data $D_{\text{exp}}(t)$. The plot visualizes both data sets to assess their agreement. The x-axis uses a logarithmic scale to better represent the time range.

```
% Reference creep data for comparison
x_ref = x_data_c(:,1);
y_ref = y_data_c(:,1);

%% Plot
figure(1);
hold on;
set(gcf, 'Position', [300, 320, 700, 450]);
plot(x_in(:, 1), D_t(:, 1), 'r.', 'MarkerSize', 25, 'LineWidth', ...
    1.5, 'DisplayName', 'Computed Creep Compliance');
plot(x_ref, y_ref, 'gx', 'MarkerSize', 10, 'LineWidth', 1.5, ...
    'DisplayName', 'Experimental Creep Compliance');

set(gca, 'XScale', 'log');
grid on;
xlabel('Time [s]');
ylabel('Creep Compliance [Pa^{-1}]');
title_srt = sprintf('Creep Compliance - Stress Relaxation Conversion (%s)', title_setup);
title(title_srt, 'FontSize', 13);
legend('show', 'Location', 'northwest');

hold off;
```

Again, as before, the results for this task are analysis in Section 3.3.

2.4 Long-term Nonisothermal Physical Aging

The modeling process starts with initializing arrays to store the variables, experimental data, and computed results.

```
x_data_noniso = zeros(7,1);
y_data_noniso = zeros(7,1);
x_data_iso = zeros(7,1);
y_data_iso = zeros(7,1);

t_int=(linspace(1,50,1000))';
lambda_iso = zeros(length(t_int),1);
lambda_noniso = zeros(length(t_int),1);
D_t_noniso = zeros(length(t_int),1);
D_t_iso = zeros(length(t_int),1);

tau = 0.4425; % hr
D_0 = 0.460; % GPa
beta = 0.417;
t0 = 0.5;

%% Load DataSet
Dataset_noniso = '>..directory/Multi_T_step.csv';
data_noniso = readtable(Dataset_noniso,'ReadVariableNames', false);

% First curve from "Multi step" graph (27-73-27-73)
for i = 3:9
    x_data_noniso(i-2,1) = log10(data_noniso{i,1});
    y_data_noniso(i-2,1) = log10(data_noniso{i,2});
end

Dataset_iso = '>..directory/Single_T_step.csv';
data_iso = readtable(Dataset_iso,'ReadVariableNames',false);

% Isothermal curve from the "Single step" graph (73)
for i = 3:9
    x_data_iso(i-2,1) = log10(data_iso{i,7});
    y_data_iso(i-2,1) = log10(data_iso{i,8});
end
```

Next, a linear regression is performed on the isothermal data representing the shift factor and rate using the '`fit()`' function with the '`poly1`' option for a first-degree polynomial fit:

$$y = p_1 x + p_2$$

The shift rate, μ_{iso}, is determined as the negative of the slope of the linear fit:

$$\mu_{\text{iso}} = -p_1$$

Subsequently, the shift factor, aging parameter, and creep compliance are calculated using the formulas outlined in Section 1.

```
%% Regression for Isothermal Shift Factor/Rate
fitting_iso = fit(x_data_iso,y_data_iso,'poly1');
coeffs_iso = [fitting_iso.p1 fitting_iso.p2];
a_t_iso = coeffs_iso(1,1).*(x_data_iso) + coeffs_iso(1,2);

shift_rate_iso = - coeffs_iso(1,1);
a_u_iso_int = @(x) ((t0)./(t0 + x)).^(shift_rate_iso);

for i = 1:length(t_int)
    lambda_iso(i,1) = integral(a_u_iso_int,0,t_int(i,1));
    D_t_iso(i,1) = D_0 * exp((lambda_iso(i,1)./tau).^beta);
end
```

Simultaneously, a quadratic regression is performed on the nonisothermal data:

$$y = p_1 x^2 + p_2 x + p_3$$

The shift rate is again determined as the negative of the first derivative of the fitted polynomial:

$$\mu_{\text{noniso}}(x) = -\frac{\partial \log a_e(x)}{\partial \log x} = -\left(2p_1 x + p_2\right)$$

The shift factor, aging parameter, and creep compliance are then calculated using the formulas already presented in Section 1:

$$a_{\text{noniso}} = \left(\frac{t_0}{t_0 + x}\right)^{-(2p_1 x + p_2)}$$

$$\lambda_{\text{noniso}} = \int_0^t a_{\text{noniso}} \qquad D_{\text{noniso}} = D_0 \cdot \exp\left(\frac{\lambda_{\text{noniso}}}{\tau}\right)^{\beta}$$

The resulting script for the nonisothermal analysis is as follows:

```
%% Regression for Nonisothermal Shift Factor/Rate
fitting_noniso = fit(x_data_noniso,y_data_noniso,'poly2');
coeffs_noniso = [fitting_noniso.p1,fitting_noniso.p2, fitting_noniso.p3];
a_t_noniso = (coeffs_noniso(1,1)*(x_data_noniso).^2+ coeffs_noniso(1,2)*x_data_noniso+coeffs_noniso(1,3));

shift_rate_noniso = - (2.*coeffs_noniso(1,1).*x_data_noniso + coeffs_noniso(1,2));
a_u_noniso = @(x) ((t0)./(t0+x)).^(-2.*coeffs_noniso(1,1).*x - coeffs_noniso(1,2));

for j = 1:length(t_int)
    lambda_noniso(j,1) = integral(a_u_noniso,0,t_int(j,1));
    D_t_noniso(j,1) = D_0 * exp((lambda_noniso(j,1)./tau).^beta);
end
```

As usual, the results were plotted to visualize the findings; an example of the plot is shown below.

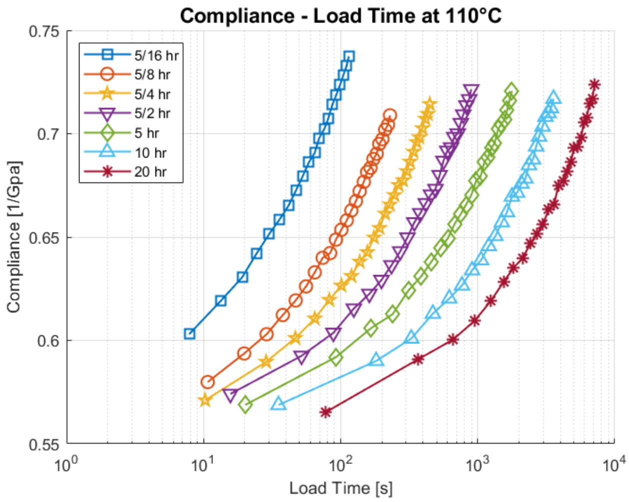
```
% Effective time vs. Real time
figure(6);
set(gcf, 'Position', [510, 320, 600, 430]);
hold on;
plot(t_int, lambda_iso, 'b-', 'LineWidth', 2.5, 'DisplayName', 'Isothermal');
plot(t_int, lambda_noniso, 'r-', 'LineWidth', 2.5, 'DisplayName', 'Nonisothermal');
title('Effective time vs. Real time', 'FontSize', 13);
xlabel('Real Time [h]');
ylabel('\lambda');
legend('show', 'Location', 'northwest');
grid on;
hold off;
```

3 Results

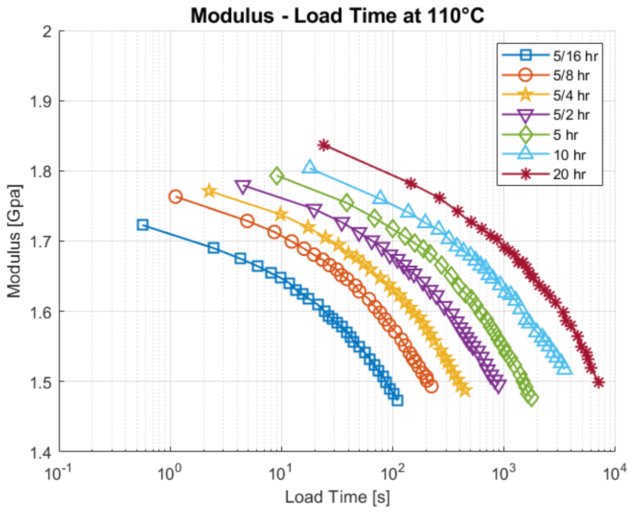
From the MATLAB script provided in Section 2, these are examples of the resulting data and figures obtained from the simulation.

3.1 Data Extraction

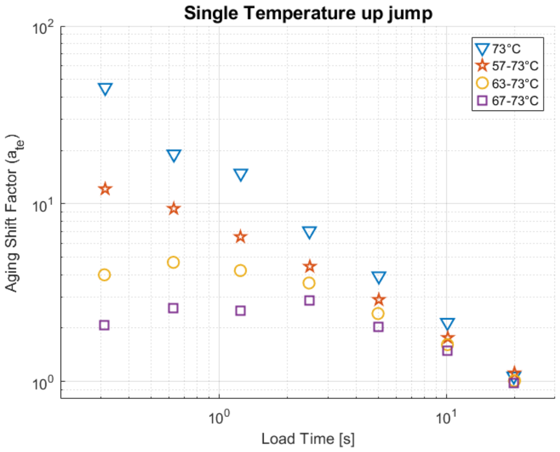
Creep Compliance graphs illustrates the material's compliance behavior at various aging times and temperatures, revealing how it deforms over time under constant stress. It allows for an assessment of the material's long-term strain behavior under elevated temperatures.

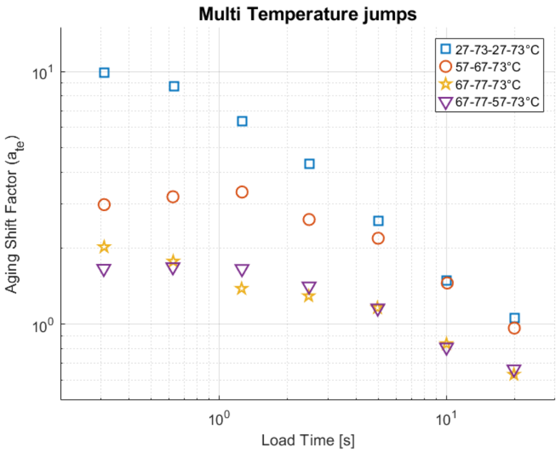


Stress Relaxation Modulus graphs shows the stress relaxation characteristics at different aging times and temperatures. It demonstrates how the material's internal stresses relax over time, highlighting its stiffness and recovery properties under thermal exposure.



The third graphs presents the **aging shift factors** obtained from short-term creep tests, in particular for a single jump and multi-temperature jumps. These factors are critical for predicting long-term material behavior, especially in non-isothermal conditions.





3.2 Characterization of Isothermal Physical Aging

The aging shift factors and shift rates for all six datasets (three temperatures and two datasets, compliance and modulus) were determined using the RMS error minimization technique described in Section 2.2.

The Table 1 list the calculated aging shift factors and corresponding shift rates for each dataset.
For the six curves, three graphs each were obtained:

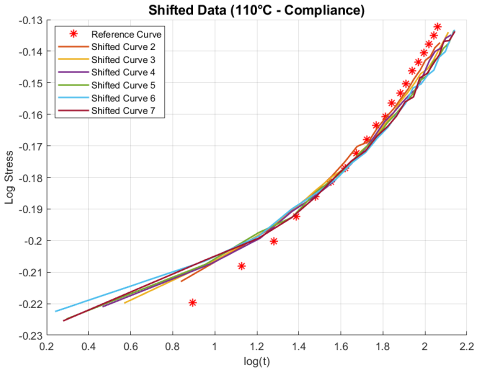
- **Shifted Data (log Stress vs. log Time)** This plot demonstrates the alignment of the aging curves after applying the calculated horizontal and vertical shift factors.
  The reference curve (lowest aging time) was chosen as the anchor, and subsequent curves were shifted horizontally and vertically to minimize the RMS error.
- **Aging Shift Factors (log Time vs. Shift Factor)** This plot displays the calculated aging shift factors as a function of aging time in a log-log space.
  The linear trend observed in each dataset reflects the logarithmic dependence of aging shift factors on time, consistent with the theoretical framework.
- **Aging Shift Rates (log Time vs. Shift Rate)** The aging shift rates, derived from the slope of the shift factor vs. log aging time, are plotted in this figure.
  These rates indicate the speed at which the material's properties evolve with aging.

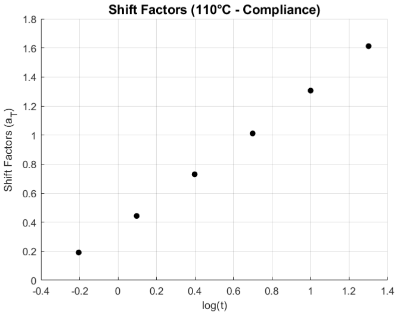
Table 1. Aging shift factors and aging shift rate along the x-axis for the six datasets analyzed. The curves have been renamed, for simplicity, according to the temperature they refer to, with "C" indicating the creep compliance curve and "M" indicating the stress relaxation modulus.

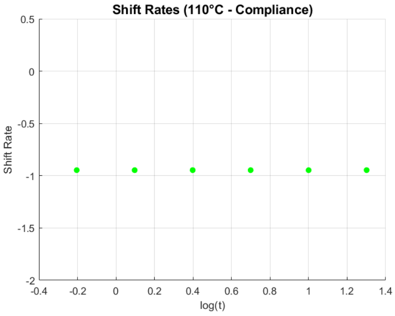
	Shift Factor						**Shift Rate**
Plot	**Curves 1-2**	**Curves 1-3**	**Curves 1-4**	**Curves 1-5**	**Curves 1-6**	**Curves 1-7**	**along x**
110C	0.1914	0.4429	0.7299	1.0111	1.3061	1.6119	-0.9467
110M	0.3185	0.6229	0.9227	1.2418	1.4899	1.7681	-0.9651
120C	0.1703	0.4385	0.7239	1.0155	1.3109	1.6150	-0.9617
120M	0.3210	0.6102	0.9187	1.1896	1.5174	1.8102	-0.9912
130C	0.1371	0.3052	0.5839	0.8471	1.1337	1.4436	-0.8809
130M	0.3013	0.5869	0.9206	1.1969	1.5039	1.8007	-0.9989

3.2.1 Creep Compliance at 110°C

This figures demonstrates the alignment of the aging curves for creep compliance after applying the necessary horizontal shifts. The shift factor at 110°C is also shown.

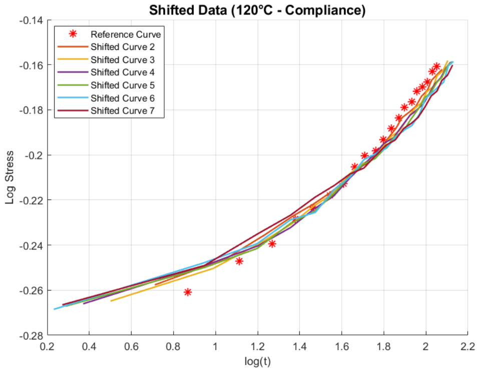


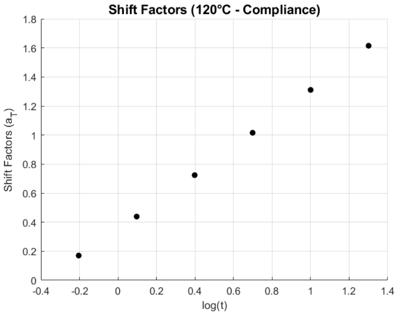


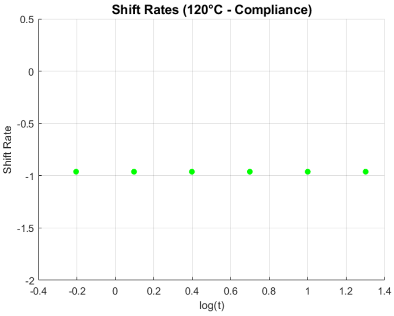


3.2.2 Creep Compliance at 120°C

This figures demonstrates the alignment of the aging curves for creep compliance after applying the necessary horizontal shifts. The shift factor at 120°C is also shown.

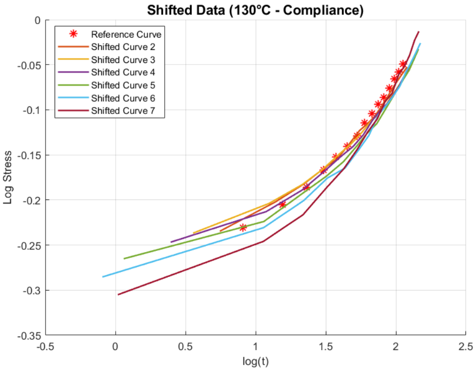


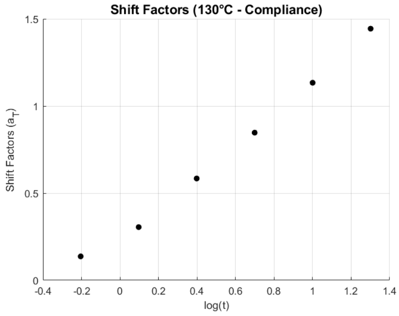




3.2.3 Creep Compliance at 130°C

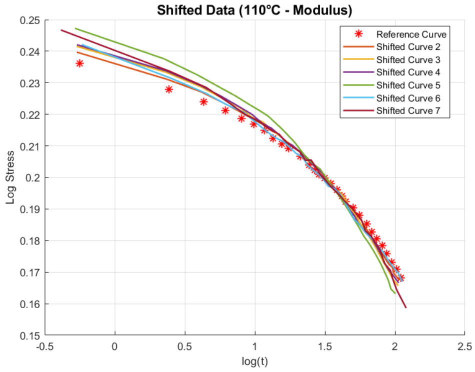
This figures demonstrates the alignment of the aging curves for creep compliance after applying the necessary horizontal shifts. The shift factor at 130°C is also shown.

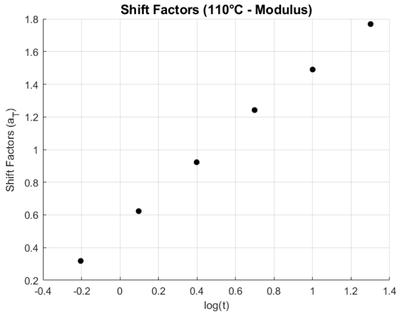




3.2.4 Stress Relaxation Modulus at 110°C

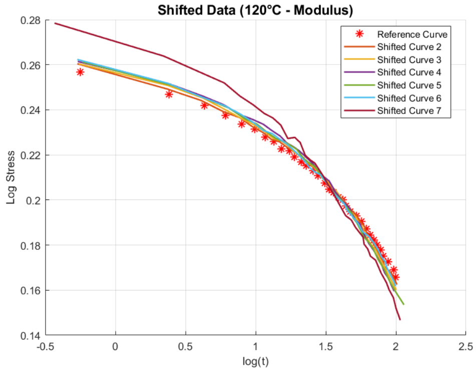
This figures demonstrates the alignment of the aging curves for stress relaxation modulus after applying the necessary horizontal shifts. The shift factor at 110°C is also shown.

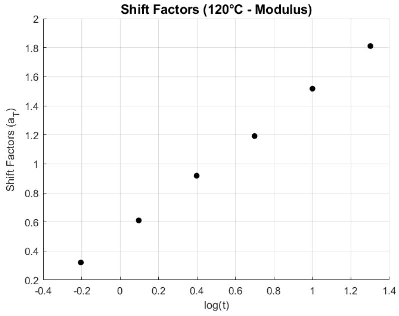




3.2.5 Stress Relaxation Modulus at 120°C

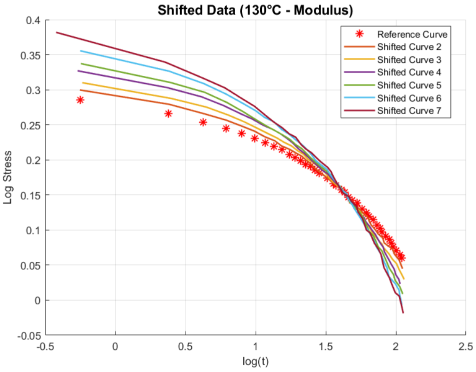
This figures demonstrates the alignment of the aging curves for stress relaxation modulus after applying the necessary horizontal shifts. The shift factor at 120°C is also shown.

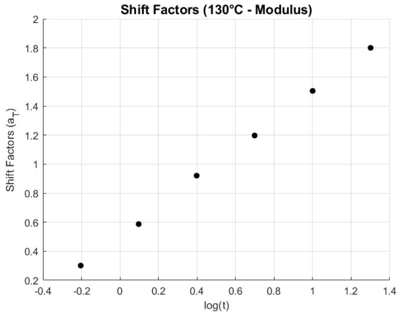




3.2.6 Stress Relaxation Modulus at 130°C

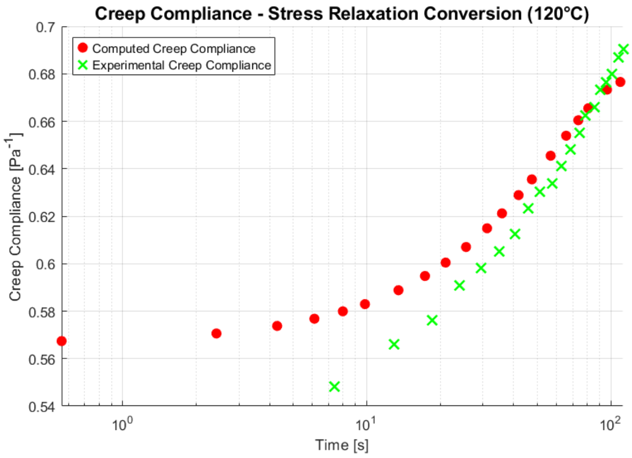
This figures demonstrates the alignment of the aging curves for stress relaxation modulus after applying the necessary horizontal shifts. The shift factor at 130°C is also shown.





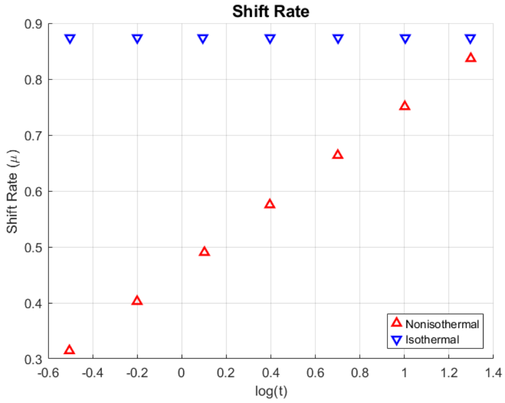
3.3 Relationship between Creep and Stress Relaxation

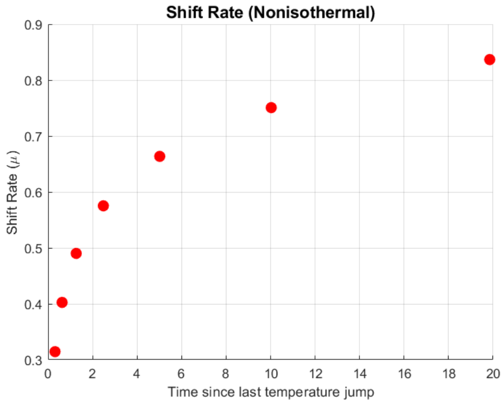
From the script presented in Section 2.3, the graphs obtained for 120°C is shown below.
This graph compare the computed creep compliance, derived from the stress relaxation modulus data, with the experimental creep compliance.



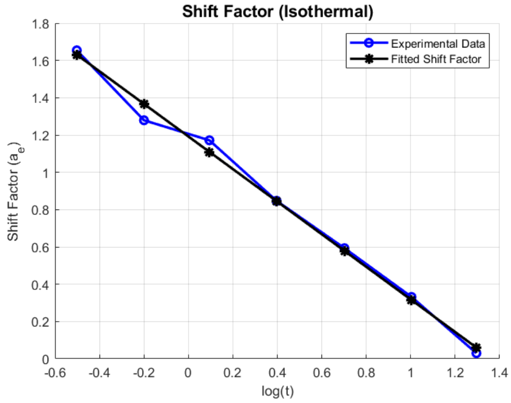
3.4 Long-term Nonisothermal Physical Aging

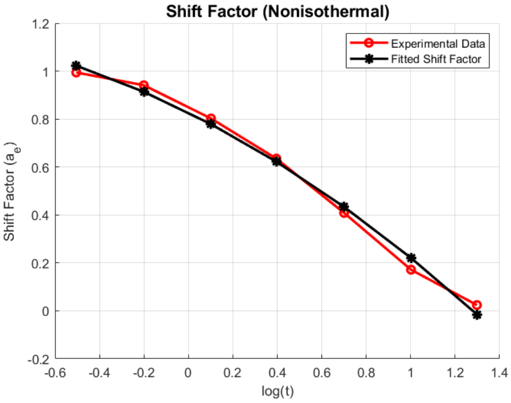
The shift rate for both conditions is shown, illustrating the rate at which the material undergoes aging under different thermal regimes.



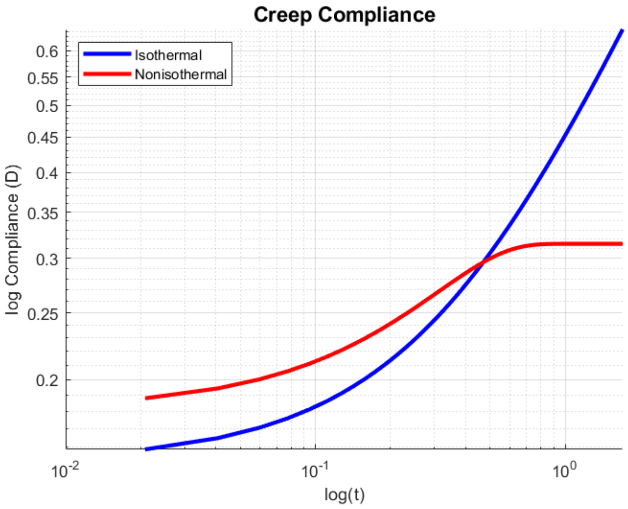


The fitted shift factors for isothermal and nonisothermal aging are also provided, highlighting the distinct behaviors of the material over time.

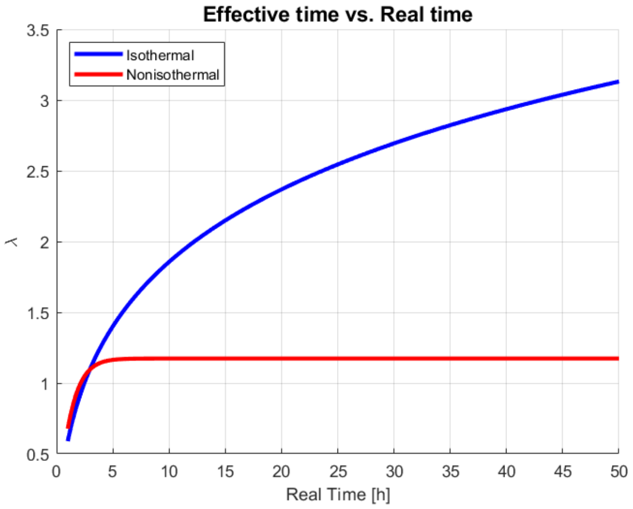




The creep compliance for both isothermal and nonisothermal conditions is plotted in a log-log scale, allowing for an assessment of the material's time-dependent viscoelastic properties.



Finally, the comparison between effective time and real time for both isothermal and nonisothermal conditions highlights the influence of the aging process on the material's behavior with respect to the time scale under varying thermal conditions.



4 Discussion & Conclusion

4.1 Characterization of Isothermal Physical Aging

The results presented in Section 3.2 (Table 1) confirm the validity of the isothermal physical aging characterization method, which was applied to both creep compliance and stress relaxation modulus data sets. The aging shift factors, determined by minimizing the root-mean-square (RMS) error upon horizontal shifting, show that the curves for different aging times effectively overlap when shifted. This confirms that the aging shift factors provide a consistent way to describe the changes in the material's behavior due to aging, demonstrating the robustness of the approach.

For both the creep compliance $D(t)$ and the stress relaxation modulus $G(t)$, the aging shift factors a_t plotted against the logarithm of time exhibit a linear relationship, expressed mathematically as:

$$\log(a_t) = m \cdot \log(t) + c$$

where m is the slope (shift rate) and c is the intercept. This indicates that the shift factor scales predictably with the logarithmic aging time, which is a hallmark of physical aging in polymer materials.

The shift factor a_t is fundamentally connected to molecular mobility, representing how the effective time scale of molecular relaxation changes as the material undergoes aging. The observed linear dependence suggests that the underlying physical processes, such as molecular rearrangement and redistribution of free volume, occur at a consistent rate, leading to a uniform time-scale transformation. This is consistent with the principles of physical aging, where material properties evolve logarithmically with time.

Additionally, when the shift rate

$$\mu = -\frac{\partial \log(a_t)}{\partial \log(t)}$$

is calculated and plotted against $\log(t)$, it appears as a horizontal line:

$$\mu = \text{constant}$$

This constant behavior implies that the aging process does not deviate over time, reinforcing the notion of a stable and uniform transformation of the material's time scales.

4.2 Relationship between Creep and Stress Relaxation

The goal was to convert the stress relaxation modulus data at 120°C to creep compliance using the established theoretical relationship between stress relaxation and creep compliance. This was achieved by applying the Laplace transform to the stress relaxation modulus and then obtaining the corresponding creep compliance as a function of time (as detailed in Section 2.3).

While the calculated creep compliance results show a similar trend as the experimental creep data, the curves slightly diverge, with the calculated creep compliance starting to flatten out at both higher and lower times.

This discrepancy can be attributed to several factors related to the underlying physical processes governing stress relaxation and creep in polymers:

a. Finite Time Effects: Experimental data may reflect prolonged molecular rearrangements and stress relaxation over time, which are not fully captured in the stress relaxation modulus data. The calculated curve's flattening may indicate a faster equilibrium state than observed experimentally.

b. Viscoelastic Behavior and Aging Effects: Time-dependent phenomena like physical aging or time-temperature superposition can alter the relaxation modulus over time, causing deviations in creep compliance predictions. These effects are often not included in simplified theoretical models.

c. Mathematical Approximation Limitations: The Laplace transform relies on approximations, such as assuming constant or ideal relaxation times, which may not accurately represent the viscoelastic response of polymers. This simplification could explain the observed flattening in the calculated compliance.

4.3 Long-term Nonisothermal Physical Aging

The fit for the isothermal model was quite accurate, confirming that under constant temperature conditions, the aging shift factor does not change with time, but instead behaves in a manner consistent with the Kohlrausch function parameters. The consistency of the shift factor in the isothermal case makes sense, as the material experiences the same temperature, so the rate of aging does not need to adjust for temperature fluctuations.

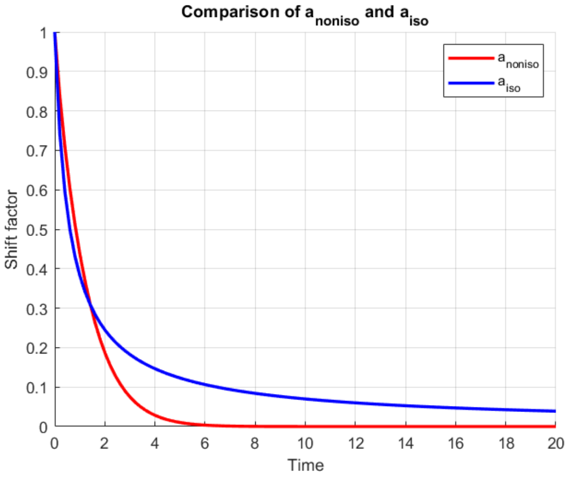
In contrast, the nonisothermal prediction involved the application of a temperature-dependent aging shift factor. In the graphs shown in Section 3.4, as the material undergoes thermal cycling, the shift factor adjusts in a linear fashion, as opposed to the constant behavior in the isothermal case. This is because, in nonisothermal conditions, the aging of the material is influenced not just by the cumulative time spent at each temperature, but also by the transitions between temperatures, leading to a more complex and temperature-sensitive shift factor.

The **creep compliance** was analyzed in log-log space for both isothermal and nonisothermal conditions:

- **Isothermal Case:** The creep compliance under isothermal conditions exhibited the expected growing trend, which is a characteristic of viscoelastic materials experiencing time-dependent deformation. This growing behavior reflects the material's continued response to stress over time, with the compliance increasing as the material undergoes more creep deformation. This is consistent with the idea that, at constant temperature, the material's response follows the usual time-dependent creep pattern, where compliance increases as time progresses.

- **Nonisothermal Case:** In the nonisothermal case, the creep compliance initially increased in a manner similar to the isothermal case, but then it reached a plateau after a certain period. This plateau suggests that the creep compliance was limited by the material's temperature cycling.
  A more careful investigation reveals that the issue stems from the fitting method used in MATLAB's `'poly2'` interpolation, specifically with the decay of the shift function a_u. The rapid decay of a_u is due to an overfitted exponential decay term, causing a_u to decay too quickly. As a result, when a_u is integrated to compute the effective time λ, the integral plateaus, as the value of a_u reaches zero after a certain point (see the Figure below).
  This behavior is not a physical limitation of the material but a mathematical issue arising from the fitting procedure. The rapid decay of a_u is a result of the specific data and the chosen fitting model, not the material's intrinsic properties. In reality, the creep compliance in a nonisothermal system should grow exponentially at first, influenced by increasing temperature, but should eventually reach a steady-state creep rate where the compliance stabilizes, due to the balance between deformation and recovery mechanisms (which are temperature-dependent).
  The plateau observed in λ is simply a consequence of the incorrect behavior of a_u under the current fitting model.

The second-order polynomial fit was chosen because it provides a smoother, less extreme variation that avoids the rapid decays or blow-ups associated with exponential functions, allowing for more stable integration, compared to an exponential one.



The relationship between **effective time and real time** was also explored to better understand the material's creep response under both isothermal and nonisothermal conditions.

- **Isothermal Case:** In the isothermal scenario, the relationship between real time and effective time follows the expected trend, with a logarithmic-like increase.
  This logarithmic behavior arises because the aging shift factor is a function of time, and in isothermal conditions, the material's response to stress becomes progressively slower as time increases.
- **Nonisothermal Case:** In contrast, for the nonisothermal case, the effective time initially grows as expected but then quickly plateaus. This rapid leveling off of effective time is definitely due to the same issue encountered in the creep compliance fit – namely, the inappropriate decay of the aging shift factor in the fitting procedure.
  The overestimated decay rate of a_u leads to an overestimated aging shift factor, causing the effective time to quickly stabilize. In theory, the effective time should continue increasing as the material responds to the varying temperature over time, but this fast plateau prevents a true reflection of the temperature cycling's impact on the material.

For nonisothermal conditions, as the material undergoes cyclic temperature changes, the effective time should eventually closely align with real time. This is because, at long times, the effect of previous thermal history should become negligible, and the material would reach a steady-state deformation rate regardless of the specific thermal history.
Hence, while the effective time may initially deviate due to temperature fluctuations, it should eventually coincide with the real time in long-term creep.

References

[1] Yunlong Guo et al. "Modeling mechanical aging shift factors in glassy polymers during nonisothermal physical aging. I. Experiments and KAHR-ate model prediction". In: *Journal of Polymer Science Part B: Polymer Physics* 47.3 (2009), pp. 340–352.

[2] Yunlong Guo and Roger D Bradshaw. "Long-term creep of polyphenylene sulfide (PPS) subjected to complex thermal histories: The effects of nonisothermal physical aging". In: *Polymer* 50.16 (2009), pp. 4048–4055.

Appendix A: DATA EXTRACTION

```
%% Luigi Casagrande
clc;
clear;
close all;

temps = [110, 120, 130];
aging_times = [5/16, 5/8, 5/4, 5/2, 5, 10, 20];

%% 110C Dataset

% Read the CSV file
Dataset110C = '>..directory/110C_Dataset.csv';
data110C = readtable(Dataset110C,'ReadVariableNames',false);

numPairs = width(data110C) / 2;  % Number of (X,Y) pairs

for i = 1:numPairs
    % Extract X and Y columns for the current pair
    X = data110C{:, 2*i-1};  % X column (odd index)
    Y = data110C{:, 2*i};    % Y column (even index)

    % Sort X and re-order Y accordingly
    [X_sorted, idx] = sort(X);
    Y_sorted = Y(idx);

    % Store the sorted data back into the table
    data110C{:, 2*i-1} = X_sorted;
    data110C{:, 2*i} = Y_sorted;
end

writetable(data110C, 'DataSet/Data110C.xlsx', 'WriteVariableNames', false);

% Extract datasets
% '5/16 hr' data
x1110C = data110C{1:20, 1};
y1110C = data110C{1:20, 2};
% '5/8 hr' data
x2110C = data110C{1:end, 3};
y2110C = data110C{1:end, 4};
% '5/4 hr' data
x3110C = data110C{1:end, 5};
y3110C = data110C{1:end, 6};
% '5/2 hr' data
x4110C = data110C{1:end, 7};
y4110C = data110C{1:end, 8};
% '5 hr' data
x5110C = data110C{1:end, 9};
y5110C = data110C{1:end, 10};
% '10 hr' data
x6110C = data110C{1:end, 11};
y6110C = data110C{1:end, 12};
% '20 hr' data
x7110C = data110C{1:end, 13};
y7110C = data110C{1:end, 14};

% Plot Creation
figure(1);
set(gcf, 'Position', [10, 320, 600, 430]); % Adjust the size & position
set(gca, 'XScale', 'log'); % Logaritmic x-axis
hold on;

% Plot each dataset
plot(x1110C, y1110C, '-s', 'DisplayName', '5/16 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x2110C, y2110C, '-o', 'DisplayName', '5/8 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x3110C, y3110C, '-p', 'DisplayName', '5/4 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x4110C, y4110C, '-v', 'DisplayName', '5/2 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x5110C, y5110C, '-d', 'DisplayName', '5 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x6110C, y6110C, '-^', 'DisplayName', '10 hr', 'LineWidth', 1.2, 'MarkerSize', 8);
plot(x7110C, y7110C, '-*', 'DisplayName', '20 hr', 'LineWidth', 1.2, 'MarkerSize', 8);

% Customize the plot
xlabel('Load Time [s]');
ylabel('Compliance [1/Gpa]');
xlim([1, 1e4]); % X-axis range
ylim([0.55, 0.75]); % Y-axis range

legend('show', 'Location', 'northwest');
grid on;
title('Compliance - Load Time at 110°C', 'FontSize', 13);
legend;
hold off;
```