LayerSlayer Transient User’s Manual

Matthew R. Begley
Mechanical Engineering, University of California, Santa Barbara
(Dated: 14 December 2012)

This document provides user instructions for the multilayer analysis code Layer Slayer Transient. The framework is general for an arbitrary number of layers, which are assumed to be elastic, isotropic layers described by a modulus, coefficient of thermal expansion, thermal conductivity, volumetric heat capacity, a reference (strain-free) temperature distribution (linear), growth strain distribution (linear), and prescribed thermal history of the top and bottom of the stack. The code predicts the following:

- transient temperature distribution throughout the layers
- time-dependent deformation of the multilayer, assuming plane strain deformation
- time-dependent stress-distributions throughout the stack, assuming plane strain deformation
- time-dependent energy release rate for debonding at all locations, assuming plane strain conditions everywhere
- response can

I. GETTING STARTED

LayerSlayer Transient is written in Mathematica. The source code (which defines commands used in the analysis) is LSTencrypt, which is read (“loaded”) into any Mathematica notebook. The source code is loaded using the Mathematica commands:

```
SetDirectory["/Users/Begley/Desktop/LST FINAL"]
<< LSTencrypt'
```

where the pathway in the SetDirectory command should be set to the name of the folder in which the file LSBencrypt is stored. **NOTE: reading in the encrypted files will generate error messages pertaining to function definitions: these can be ignored.** The file LSB Example is a conventional Mathematica notebook which illustrates how various analyses are conducted.

II. OVERVIEW

The analysis is conducted for a stack of blanket films, as shown in Figure 1. The films are assumed to be infinite in the \((x,z)\) plane, with a semi-infinite crack that grows along an interface that lies on a specific \((x,z)\) plane, with a crack tip that is aligned with the \(z\)-direction; energy release rates are computed for the steady-state growth scenario, i.e. when the crack is long enough so as to not influence the results. The problem is essentially one-dimensional, such that behavior in the \((x,y)\) plane is analyzed: in the third dimension (out-of-plane to the analysis), plane strain deformation (i.e. \(\varepsilon_z = 0\)) is enforced.
Temperatures are assumed to be uniform in the plane of the blanket, i.e. in the \((x,z)\) plane, with gradients occurring only in the through-thickness direction, i.e. in the \(y\) direction. The thermal analysis solves for the transient distribution through the intact stack. If finite conductance is specified at the interfaces, the temperature is not continuous across the interface but rather experiences jumps proportional to the inverse of the interface conductance: hence, the temperatures on either side of the interface (i.e. bottom of one layer and top of layer beneath it) are retained as possibly independent variables. Temperatures on either side of an interface will be identical if the interface conductance is sufficiently high.

The transient temperature distribution is found via finite element analysis of the one-dimensional heat transfer equation:

\[
c_V \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial t^2} \tag{1}
\]

where \(T\) is the temperature, \(t\) is time, \(c_V\) is the volumetric heat capacity, and \(k\) is the thermal conductivity. The transient problem being solved is one where the temperatures at the top and bottom of the stack are prescribed functions of time. (One can import temperature data into Mathematica and generate interpolation functions with a single command, thus avoiding form-fitting of specific time functions.) One can specify initial conditions as either a uniform temperature or as the linear steady-state temperature distribution that arises from prescribed surface temperatures. The code derives a set of coupled linear first order differential equations for the temperatures at the nodes, and solves them using the built-in Mathematica function NDSolve, which is a powerful differential equation solver with adaptive time-stepping.
When analyzing cracking, it is assumed that the crack has no influence on the temperature distribution, i.e. the intact stack is analyzed. While cracking would naturally alter the heat flow in the cracked region, it is assumed that cracking events are sufficiently fast so as to not allow for changes associated with debonding. That is, the scenario being analysis consists of a steady-state crack growing at speeds faster than thermal transport. (The analysis of initiation of crack growth, involving a sufficiently fast so as not to allow for changes associated with debonding. That is, the scenario being analysis consists of a perfect conduction, with (piece-wise) continuous temperatures throughout the stack.

The key assumption is that the strain distribution in bonded layers is given by:

\[ \epsilon_c(y) = \epsilon^o_x - \kappa y + \alpha (T(y) - T^o_y) + \epsilon^g_x(y), \]  

where \( \epsilon^o_x \) is the elongation in the \( x \)-direction at the bottom of the stack, \( \kappa \) is the curvature of the stack about the \( z \)-axis, \( \alpha \) is the coefficient of thermal expansion in the layer, \( T^o_y \) is a reference temperature that has a linear distribution in a given layer, \( \epsilon^g_x(y) \) is a growth strain that has a linear distribution in any given layer, and \( T(y) \) is the temperature distribution determined the finite element analysis. This strain distribution is used to compute the resultant moment and normal force on any stack (including sub-stacks created by cracking), which are equated to applied moments/forces. This fundamentally determines the elongation and curvature of the stack; the stresses and strain energy densities are then computed using those deformation variables and the above strain definition.

III. GEOMETRY AND PROPERTY DEFINITION

A. Individual layer definition

Each layer is given a name and the list of its properties, defined according to:

\[ \text{Layer} = \{ h, E, \nu, \alpha, \{ T^\text{bot}_o, T^\text{top}_o \}, \{ \epsilon^\text{el}_{g,o}, \epsilon^\text{el}_{o} \}, k, c_V, N_{el}, \{ T_b, T_t \} \}, \]  

where \( \text{Layer} \) is the name of the layer (can be anything!), \( h \) is the layer thickness, \( E \) is the layer modulus, \( \nu \) is the layer’s Poisson ratio, \( \{ T^\text{bot}_o, T^\text{top}_o \} \) are the bottom and top reference temperatures for that layer (i.e. there is a linear distribution of reference temperature, which defines the zero thermal strain temperature for that location), \( \{ \epsilon^\text{el}_{g,o}, \epsilon^\text{el}_{o} \} \) are the growth strains at the top and bottom of the layer, \( k \) is the thermal conductivity of the layer, \( c_V \) is the volumetric heat capacity of the layer, \( N_{el} \) is the number of finite elements in that layer, and \( \{ T_b, T_t \} \) are the current temperature of the layer at the top and bottom of the layer. The order must be as prescribed above.

B. Stack/multilayer definition

The multilayer is defined as a list of layers, as in:

\[ \text{Multilayer} = \{ \text{Substrate, BondCoat, Coating} \}, \]  

where \( \text{Substrate, BondCoat and Coating} \) are previously defined layers. The order must be from bottom to top, i.e. the order in the \( \text{Multilayer} \) list indicates the relative position of the layers. Note that you can define the multilayer with any name (including layers with any name), e.g. \( \text{CaseA} = \{ \text{Layer1, Layer2} \} \).

For the thermal analysis, the analysis incorporates interface elements with zero thickness to allow for temperature drops between layers associated with finite values of interface conductance. For the thermal analysis commands, one must provide a list of interface conductances, with the numbering/ordering being from bottom to top: e.g.:

\[ \text{Interfaces conductance} = \{ k^1_{12}, k^2_{23} \}, \]  

where \( k^1_{12} \) is the interface conductance between layers 1 and 2, \( k^2_{23} \) is the interface conductance between layers 2 and 3, and so forth. Again, note that this list of interface properties can be given any name, as in \( \text{IfaceEx} = \{ 10^6, 10^3 \} \), etc. Setting the interface conductances to a large number (in comparison to the effect layer conductances, characterized by \( k/h \)) results in nearly perfect conduction, with (piece-wise) continuous temperatures throughout the stack.
IV. THERMAL PROBLEM DEFINITION

A. Boundary Conditions

The problem is defined by prescribing the temperatures at the top and bottom of the stack as a function of time. This can be done either by specifying an explicit time-function for the temperatures, or by fitting an interpolation function to a data set. Here is an example:

\[
\begin{align*}
T_{\text{bot}}[t] &= 800 \exp(-2t) \\
T_{\text{top}}[t] &= 800 \exp(-t)
\end{align*}
\]

Here, the underscore after the \( t \) in the function definition means that Mathematica will explicitly define the function, such that derivatives can be taken. Any function can be specified, including a Piecewise function that is useful for ramps, hold times, and decay (for example). The command Interpolation applied to a data set of \( \{\{t_1, T_1\}, \{t_2, T_2\}, \ldots\} \) will generate this type of function definition as well, as illustrated in the example in this manual. The list ThermalBCs can be named anything, so long as it is a list with the first entry being the temperature function at the bottom, and the second entry being the temperature function at the top.

B. Initial Conditions

The command that runs the thermal analysis is:

\[
\text{GetThermalSolution[multilayer, iface, thermalbc, ic, tmax]}
\]

where \textit{multilayer} is the list of layers (each layer being a list of properties) being analyzed, \textit{iface} is the list of interface conductances, and \textit{tmax} is the maximum time sought. (\textit{NOTE: if an interpolation function is used, one should take care not to ask for a solution at times for which there is no data, as the interpolation function will not be defined beyond the fitted range.})

The variable \textit{ic} is the initial condition: if this is set to \(-1\), then the steady-state distribution arising from the boundary conditions at time equals zero is used as the initial temperature distribution. Otherwise, the number supplied for \textit{ic} will be the uniform temperature assumed as the initial condition throughout the stack.

V. SUBROUTINES/FUNCTIONS

A. Temperature analysis

The command that runs the thermal analysis is:

\[
\text{GetThermalSolution[multilayer, iface, thermalbc, ic, tmax]}
\]

where \textit{multilayer} is the list of layers (each layer being a list of properties) being analyzed, \textit{iface} is the list of interface conductances, and \textit{tmax} is the maximum time sought. (\textit{NOTE: if an interpolation function is used, one should take care not to ask for a solution at times for which there is no data, as the interpolation function will not be defined beyond the fitted range.})

The variable \textit{ic} is the initial condition: if this is set to \(-1\), then the steady-state distribution arising from the boundary conditions at time equals zero is used as the initial temperature distribution. Otherwise, the number supplied for \textit{ic} will be the uniform temperature assumed as the initial condition throughout the stack.

The command above generates a list of pairs, consisting of \( \{y_i, T_i\} \), the global position of the node and the associated temperature. The temperature in the list is actually an interpolation function with respect to time, such that it contains the temperatures at that location for all time between zero and the specified value for maximum time. One can generate a list of position-temperature number pairs by specifying a specific time (e.g. \( t = 2.3 \)), as in:

\[
\begin{align*}
\text{results} &= \text{GetThermalSolution[\ldots]} \\
\text{timeA} &= \text{results} / . t \to 2.3
\end{align*}
\]
Now, the list \( \text{timeA} \) contains a list of nodal positions and temperatures at the requested time \( t = 2.3 \).

A function has been provided to automate plotting of the temperature distribution throughout the stack at specified times; this function is:

\[
\text{PlotTemperature}[\text{results}, \text{timelist}] \quad (13)
\]

where \( \text{results} \) is the list of nodal positions and interpolation functions generated by executing the \( \text{GetThermalSolution} \) command, and \( \text{timelist} \) is a list of times of interest. The command generates a plot that is automatically scaled according to the dimensions of the stack and the temperature range of the solution. Here’s an example (where the initial condition is the steady-state temperature distribution arising from the temperatures at the top and bottom at time zero):

\[
\text{results} = \text{GetThermalSolution}[\text{multilayer}, \text{iface}, \text{thermalbc}, -1, \text{tmax}] \quad (14)
\]

\[
\text{tset} = \text{Table}[i, \{i, 0.0, 2.0, 0.1\}] \quad (15)
\]

\[
\text{PlotTemperature}[	ext{results}, \text{tset}] \quad (16)
\]

where \( \text{test} \) is a list of times spanning from \( t = 0 \) to \( t = 2 \) in increments of \( \Delta t = 0.1 \); hence, the temperature plot will have twenty curves. The output is shown in Figure 2.

\[
\text{FIG. 2. Example of temperature distributions for the three-layer system described in the next section, for the times in the list } \text{test}. \text{ The curves are always color coded from red (early times) to blue (later times).}
\]

An obvious question that often arises is whether or not a prescribed temperature history reaches steady-state (particularly for cases that involve a ramp and hold time): this can be checked by recovering the steady-state temperature profile for a set of fixed surface temperatures:

\[
\text{GetSteadyState}[\text{multilayer}, \text{iface}, \text{Tbot}, \text{Ttop}] \quad (17)
\]

where \( \text{multilayer} \) is the list of layers, \( \text{iface} \) is the interface conductances: \( \text{Tbot} \) and \( \text{Ttop} \) are the temperatures at the top and bottom of the stack, respectively. The command returns a list of pairs that have nodal positions and the steady-state temperatures at those positions. One can use the command \( \text{ListPlot} \) directly on the output of \( \text{GetSteadyState} \) to plot the steady-state temperature profile.

### B. Stress analysis

The stresses in the multilayer are obtained with a command that requires the multilayer definition, the output from \( \text{GetThermalSolution} \) (i.e. interpolation functions for nodal temperatures), the time of interest and a reference time that defines the stress-reference
temperature throughout the stack. The command is:

\[
\text{GetStressDistribution}[\text{multilayer}, \text{solution}, \text{time}, \text{reftime}],
\]

where \text{multilayer} is the name of the list defining the stack, \text{solution} is the name of the output from the thermal analysis, \text{time} is the time at which the stresses are desired, and \text{reftime} is the time that is used to define the stress-free reference state of the layers. The command returns an ordered pair of \{y_i, \sigma_i\} at the nodes, for all layers. Recall that the interfaces have a node associated with each side of the interface (i.e., one node in each layer), such that two stress values will be returned at each interface location. The command \text{ListPlot} can be used directly on the output from \text{GetStressDistribution} to plot the stress distribution at a given time.

NOTE: If \text{reftime} = -1, then the stress-free reference temperatures are taken to be linear in each layer and defined by the reference temperatures in the layer definitions. If \text{reftime} \geq 0, then the reference temperature is the temperature profile obtained by the thermal analysis.

For convenience, the command \text{PlotStress} has been defined to automatically plot the stress distributions associated with the temperature distributions for a prescribed set of times:

\[
\text{PlotStress}[\text{multilayer}, \text{solution}, \text{tset}, \text{tref}],
\]

plots \sigma, stress distributions based on the thermal analysis results \text{solution}, for the times listed in \text{tset}, using the time \text{tref} to define these stress-free state. Again, \text{tref} = -1 reverts to using the linear reference temperature distribution defined in the layer lists.

Figure 3 illustrates stress distributions arising from the temperature files in Figure 2; on the left are the stresses obtained with \text{PlotStress}[\text{Case}, \text{out}, \text{test}, -1], i.e. when a uniform stress-free reference state (in this case room temperature) is used. On the right are the stresses obtained with \text{PlotStress}[\text{Case}, \text{out}, \text{test}, 0], i.e. when the initial temperature distribution (in this case the steady-state solution) is used to define the stress-free reference temperatures. Note that these results are only for the first two seconds of the simulation: the stresses start at zero (red curves) for the case where the steady-state temperature distribution is the reference state. Eventually, the curves on the left asymptote to zero (when room temperature is reached, the layers are stress free), while on the right, the curves asymptote to the stresses that arise from cooling from the elevated steady-state profile (in which case they’ll be piece-wise linear).

C. Cracking analysis

There are two commands to get the energy release rates associated with a given temperature distribution. The first obtains the energy release rate at the interfaces between layers:

\[
\text{GetERRinterface}[\text{multilayer}, \text{results}, \text{iface}, \text{time}, \text{tref}],
\]
where `multilayer` is the list defining the stack, `results` has the results of the thermal analysis, `iface` is the interface number of interest (recall interface `i` lies between layers `i` and `i+1`), `time` is the time where the ERR is sought, and `tref` is the flag that defines the reference temperature for the analysis. When `tref = −1`, the reference temperatures are defined via the layer definitions; when `tref ≥ 0`, then the temperature profile arising at that the time value of `tref` is assumed to be stress-free.

A time-history of the energy release rate at a given interface can be plot simply by creating a list of ERRs at various times. Here’s an example:

\[
Gex1 = \text{Table}[[i, \text{GetERRInterface}[[\text{Case, out, 2, tex, -1}}], \{i, 0, 5, 0.5\}] \tag{21}
\]

\[
Gex2 = \text{Table}[[i, \text{GetERRInterface}[[\text{Case, out, 2, tex, 0.}}], \{i, 0, 5, 0.5\}] \tag{22}
\]

where `Gex1` is the list of `{ti, Gi}` pairs for the ERR at interface two, for the times from `t = 0` to `t = 5` in increments of `Δt = 0.05`, assuming the specified reference temperatures define the stress-free state. `Gex2` is the list of `{ti, Gi}` pairs for the ERR at interface two, for the times from `t = 0` to `t = 5` in increments of `Δt = 0.05`, assuming the initial condition define the stress-free state. A `ListPlot` of these results is shown in Figure 4.

![Graph of energy release rate](image)

**FIG. 4.** Example of the energy release rate as a function of time at the second interface (between second and third (top) layers, assuming either the specified reference state or the initial (steady-state) temperature distribution as the reference state. Assuming room temperature reference state, the ERR starts at a non-zero value, due to the stresses associated with the initial (steady-state) elevated temperatures: it asymptotes to zero because the final temperature is equal to the reference temperature. Assuming the stresses relax at high temperature - i.e. using the initial condition (steady-state temperature profile) as the reference state - implies the ERR starts at zero and asymptotes to the value associated with the total change from steady-state temperatures to room temperature. In both cases, the peak at short times is a consequence of differential cooling of the top layer (relative to the bottom two layers).

The energy release rates at all locations in the stack can be computed with the function:

\[
\text{GetERRDistribution}[[\text{multilayer, results, time, tref}\] \tag{23}
\]

where `multilayer` is the list defining the stack, `results` has the results of the thermal analysis, `time` is the time where the ERR is sought, and `tref` is the flag that defines the reference temperature for the analysis. When `tref = −1`, the reference temperatures are defined via the layer definitions; when `tref ≥ 0`, then the temperature profile arising at that the time value of `tref` is assumed to be stress-free. The command returns a list of ordered pairs, `{yi, Gi}`, where `yi` is the location of the delamination crack (in global `y` coordinates) and `Gi` is the energy release rate at that location.

The energy release rate distributions at various times can be plotted with the function:

\[
\text{PlotERR}[[\text{multilayer, results, tset, tref}\] \tag{24}
\]
where *multilayer* is the list defining the stack, *results* has the results of the thermal analysis, *tset* is the list of times where the ERR distribution is sought, and *tref* is the flag that defines the reference temperature for the analysis. The results for ERR distributions at the times shown in Figures 2 (the temperatures) and 3 (the stresses) are shown in Figure 5. It is interesting to note that if the substrate is brittle, delamination will occur about one coating thickness into the *substrate*!

![Graph showing energy release rate (ERR) distributions in the stack at various times.](image)

**FIG. 5.** Energy release rate distributions in the stack at various times (i.e. from $t = 0$ to $t = 2$ in increments of $\Delta t = 0.05$), assuming two different reference states. Using room temperature as the reference state implies the curves for later times will asymptote to zero. Using the elevated temperature distribution implies the distribution starts at zero, raises through a maximum at small times, and then asymptotes to a constant value. Figure 4 is a cross-plot of the values at the location $y_i = 0.0031$ (i.e. the interface underneath the top coating) as a function of time.
VI. ILLUSTRATION: HEATING-HOLD-COOLING CYCLE

Consider a three layer system, with the following layer properties:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Thickness</th>
<th>Modulus</th>
<th>Poisson's Ratio</th>
<th>CTE</th>
<th>Ref. Temp</th>
<th>Growth strain</th>
<th>Thermal conductance</th>
<th>Bot., Top Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer1</td>
<td>0.003</td>
<td>200x10^9</td>
<td>0.35</td>
<td>16x10^-6</td>
<td>{70,70}</td>
<td>{0,0}</td>
<td>25</td>
<td>{0,0}</td>
</tr>
<tr>
<td>Layer2</td>
<td>0.001</td>
<td>100x10^9</td>
<td>0.35</td>
<td>16x10^-6</td>
<td>{70,70}</td>
<td>{0,0}</td>
<td>10</td>
<td>{0,0}</td>
</tr>
<tr>
<td>Layer3</td>
<td>0.0005</td>
<td>40x10^9</td>
<td>0.15</td>
<td>11x10^-6</td>
<td>{0,0}</td>
<td>{0,0}</td>
<td>2</td>
<td>{0,0}</td>
</tr>
</tbody>
</table>

The multilayer is defined as follows:

Layer1 = {0.003, 200. x 10^9, 0.35, 16x10^-6, {70,70}, {0,0}, 25, {0,0}}; (25)
Layer2 = {0.001, 100. x 10^9, 0.35, 16x10^-6, {70,70}, {0,0}, 10, {0,0}}; (26)
Layer3 = {0.0005, 40. x 10^9, 0.2, 11x10^-6, {70,70}, {0,0}, 2, {0,0}}; (27)
Multilayer = {Layer1, Layer2, Layer3} (28)

Thus, the layers are named appropriately from bottom to top, i.e. Layer1 is on the bottom, and Layer3 is on the top.

The thermal interface conductances are defined according to:

IfaceC = {5 x 10^5, 5 x 10^6} (29)

The boundary conditions are specified according to:

Tbot[t_] := Piecewise[{{300, t < 1}, {300, 1 < t <= 3}, {300 - 3(t - 3), 3 < t < 4}, {0, t >= 4}}] (30)
Ttop[t_] := Piecewise[{{500, t < 1}, {500, 1 < t <= 3}, {-1000(t - 3.5), 3 < t < 3.5}, {0, t >= 3.5}}] (31)

This prescribed temperature history on bottom and top surfaces is depicted in Figure 6.

![FIG. 6. Prescribed temperature histories for the top and bottom face.](image)

The solution found for 0 < t < 10, assuming uniform reference temperature (as specified in the layer properties) via the following:

out = GetThermalSolution[Case, IfaceC, {Tbot, Ttop}, -1, 10.] (32)

We can plot the temperature distributions between times 3 < t < 5, i.e. just after the beginning of the cooling step, by typing:

tset = Table[1, {i, 3, 5, 0.05}] (33)
PlotTemperature[out, tset] (34)

The result of the PlotTemperature command is shown in Figure 7. Again, the results are color coded from early times (red) to later times (blue): in the list of times, the increment is \( \Delta t = 0.05 \). Hence, the upper red curve is the result for \( t = 3 \), after the
FIG. 7. Temperature profiles from the start of the down ramp $t = 3$ to just after the down ramp is completed, $t = 5$.

whole, in which case the temperature distribution is that of steady-state (because the hold time is long enough to reach this state). The bottom blue curve is the result for $t = 5$, at which point the stack has nearly cooled down to zero. The stress distribution can be found as follows:

\begin{align}
\text{PlotStress,Case,out,test,-1] } & \quad \text{(35)} \\
\text{PlotStress,Case,out,test,3.0] } & \quad \text{(36)}
\end{align}

The first of these calculates the stresses assuming the reference temperatures listed in the layer definition lists. The second of those plots the temperatures assuming the temperatures at $t = 3$ (i.e. the end of the hold period) define the stress-free state. The results of these two commands are shown in Figure 8. Note that for the uniform room temperature reference state, the stresses are asymptoting to zero (blue curves): they’re not quite there yet, because the stack is slightly heated at $t = 5$, which is the last time specified in test. When the steady-state temperature profile is used as the reference state, the stress starts at zero ($t = 3$, the red curves) and then increases as the stack is cooled. Eventually, at infinite time, the stress-state will asymptote to a piece-wise linear result, after the layers have completely cooled.

The energy release rate at the interface between layers two and three (i.e. just beneath the top layer) as a function of time is found as follows:

\begin{align}
\text{Gex1=Table[} \{\text{tex,GetERRinterface[Case,out,2,tex,-1]},\{\text{tex,3,8.,0.05}\}\} & \quad \text{(37)} \\
\text{Gex1=Table[} \{\text{tex,GetERRinterface[Case,out,2,tex,-1]},\{\text{tex,3,8.,0.05}\}\} & \quad \text{(38)}
\end{align}

The first of these calculates the ERR assuming the reference temperatures listed in the layer definition lists. The second of those plots the temperatures assuming the temperatures at $t = 3$ (i.e. the end of the hold period) define the stress-free state. The results of these two commands are shown in Figure 9. The energy release rate rises due to differential cooling of the top and bottom surfaces, until the top is completely cooled at $t = 3.5$. If room temperature is the reference state, the ERR asymptotes to zero because there is no stress in the layers. If the steady-state temperature profile is used as the zero stress reference, the ERR asymptotes to a fixed value associated with the temperature change from steady-state: note that the ERR is maximum in the transient, because of the differential cooling effect.

Finally, the ERR distribution throughout the stack can be computed as follows:

\begin{align}
\text{PlotERR[Case,out,tsett,-1] } & \quad \text{(39)} \\
\text{PlotERR[Case,out,test,3.0] } & \quad \text{(40)}
\end{align}
FIG. 8. Stress profiles from the start of the down ramp $t = 3$ to just after the down ramp is completed, $t = 5$ (in increments of $\Delta t = 0.05$; on the left, the reference temperature is taken to be room temperature (so the layers are heading towards the stress-free condition), on the right, the reference temperature is taken to be that at the start of the down ramp $t = 3$, and are headed to a steady-state linear profile associated with the change from steady-state.

FIG. 9. Energy release rate for the top layer as a function of time, assuming the reference state is the prescribed uniform room temperature, and assuming the steady-state defines the stress-free state (i.e. start of the down ramp, $t = 3$).

The first of these calculates the ERR assuming the reference temperatures listed in the layer definition lists. The second of those plots the temperatures assuming the temperatures at $t = 3$ (i.e. the end of the hold period) define the stress-free state. The results of these two commands are shown in Figure 10. The ERR at the interface plotted in Figure 9 is a cross-plot of the results at $y = 0.0031$, which is the location of the interface. Note that for both reference states, the ERR distribution initially rises due to transients(red to free): for the room temperature reference case (left), it then falls (free to blue) as the stack cools. For the steady-state temperature reference case, it initially rises (red to green), and then falls (green to light blue), but then eventually rises again (light blue to dark blue). Also, note that the energy release rate is actually maximum at a location in the substrate, just underneath the middle layer. For the properties listed here, appropriate to metal substrates, this has little significance, since this layer will be the toughest. However, for purely brittle laminates, delamination would occur by substrate splitting.
FIG. 10. Energy release rate distributions for various times, starting at the instant of down ramp \( t = 3 \) and ending at \( t = 5 \), in increments of \( \Delta t = 0.05 \); on the left, the room temperature state is assumed to be reference, on the right, the steady-state elevated temperatures are used as reference temperatures. Note the maximum energy release rate occur in the substrate, at the location of the neutral axis.