User Manual for SPALLMAP

Adrian S. Sabau
Bldg. 4508, MS 6083
Oak Ridge National Lab
Street Address: 1 Bethel Valley Road
Oak Ridge, TN 37831-6083
Ph: (865) 241-5145
Fax: (865) 574-3945

May 16, 2011

Abstract

Boiler tubes in steam power plants experience exfoliation of oxide grown on the inner side of the tubes. In extreme cases, the exfoliation cause significant tube blockages that lead to forced power plant outages. It is thus desired to predict through modeling the propensity of exfoliation events in order to inform power plant operators of possible tube blockages. The SPALLation MAP (SPALLMAP[1]) code contains modules developed for oxide growth by taking into account the following phenomena and features:

- Non-uniform thermal expansion coefficient of oxides and metal substrates,
- Plant operation schedule with periodic alternate full-load and partial-load regimes,
- Axisymmetric formulation for cylindrical tubes,
- Multiple oxide layers,
- Oxide-growth induced stresses, and
- Damage criteria from classical fracture mechanics.

The computer program is written in FORTRAN90. Its modular structure was sought for allowing the best flexibility in updating the program by implementing new constitutive equations due to availability of new material property data and/or new physical phenomena.

1 INSTALLATION

The computer program is contained in the archive file `spallation.tar.gz`. The following commands provide an expansion of different directories and files.

```
gunzip spallation.tar.gz
tar xvf spallation.tar
```

The program can be compiled using the gmake script files available in the GNUmakefile files. The compilation procedures are those from `Telluride`, a software that was developed at Los Alamos National Laboratory. The information on platform, libraries, and compiler must be changed in few files found in `spallation/options` directory, as follows.

- blas and lapack libraries are needed; change file `spallmap.libs` accordingly. For most operating systems, blas and lapack routines are installed. If they are not already installed, which would be a rare occurence, they can be downloaded from internet. Basically, only `DGETRF` and `DGETRS` subroutines are needed to solve linear systems.
- operating system; with the current distribution the OS is Linux, otherwise change the operating system in file `spallmap.env` accordingly
- FORTRAN compiler is needed; specify FORTRAN compiler in the line `FC =` in the appropriate platform file that starts with `cf` .
  For example, for a Linux box, use file `cf.linux.i686.intel` .

Once platform specific information is updated as instructed above, the code is compiled with:

```
gmake spallmap
```

Other compiler options are available by reading `spallation/GNUmakefile`. The code is build in `spallation/src/build` directory. The library and binary files are stored in `bin` directory. If new directories are added to the code, please change `spallation/src/build/GNUmakefile` accordingly. The source code is located in directories under `spallation/src`. The source code files are grouped according to the physics and/or their function. Test problems are supplied with this distribution in the `spallation/tests` or `spallation/tests/problems` directory. Then, either the binary `spallmap` is copied to the `spallation/tests` directory or the directory `spallation/src/build` is added to the PATH enviroment (UNIX jargon).

When the PATH environment is not changed, the following UNIX script `update`, which is provided in the `spallation/tests/problems` directory, can be used to update the code after changes are made to the surce code.

```
cd ../..
gmake spallmap
cd tests/problems
cp ../../bin/spallmap .
```

A problem is run by typing the following command line:

```
spallmap problem_name
```

or,

```
spallmap
```

and then provide the `problem_name` when prompted on console.

2 INPUT FILE

For each problem, an input file must be prepared with materials properties, options for physical phenomena to be considered, and operation schedule during oxidation. The input file must end with '.inp', i.e., problem_name.inp.

Based on the temperature and pressure schedule, the operation is a successsion of full-load and partial load-regimes. The full-load is sometimes referred to as high-load, while the partial-load may be referred to as low-load.

The input file contains a series of namelists, which are organized according to a common features. The following Fortran namelists must be included in the input file.

Table 1: Namelists used in input file.

Namelists	Features/Options
SOLVER	solver, physics, mesh, spallation map
DAMAGE	damage parameters
PROPERTY	properties of each material
OXIDE_LAYER	how each oxide layer is made
BOILER	Operation schedule

Several examples of the input files are included for the following problems:

- one oxide layer
  - simple plain strain, and
  - generalized plain strain.
- two oxide layers (spinel and magnetite).
- three oxide layers (spinel, magnetite, and hematite).

3 SOLVER - NAMELIST

Table 2: Physics related variables used in the SOLVER namelist

Variable	Description	Type	Valid Options	Default values
generalized_plane_strain	choose type of problem, simple or generalized plane strain problem	Logical	.true. or .false.	
internal_htc	impose internal htc, if .true.	Logical	.true. or .false.	
external_htc	impose external htc, if .true.	Logical	.true. or .false.	

Table 3: Spallation-map related variables used in the SOLVER namelist

Variable	Description	Type	Valid Options	Default values
thickness_spallation_map	minumum and maximum for the oxide thickness [microns] at which the spallation map will be computed	Real(2)	10.0, 510.0,	
no_thick_spall_map	number of data points for the spallation map	integer	less than 100	
interval_type_thick_spall_map	describe how points for the spallation map are spaced	character	'log' or 'linear'	
map_level	level of the spallation map as described by	integer	1	

Table 4: Discretization (spatial, temporal) related variables used in the SOLVER namelist

Variable	Description	Type	Valid Options	Default values
no_thick_interval	number of time intervals for oxide scale growth (load transition)	integer	10 or 20	
no_st_rad_points_tube	number of mesh points at which the stress-strain is computed in the substrate	integer	7	
no_st_rad_points_oxide	number of mesh points at which the stress-strain is computed in each oxide layer	integer	5	

Table 5: Thresholds and equivalent-oxide related variables used in the SOLVER namelist

Variable	Description	Type	Valid Options	Default values
if_average_oxide	use one oxide layers with equivalent average properties if .true.	Logical	.true. or .false.	
small_oxide_thickness	oxide thickness [micron] beyond which the thermo-mechanical solution will be obtained	real	1.0	

4 DAMAGE

Parameters that describe damage mechanisms are given below.

The failure mode considered are: through_scale_cracking, deflection_delamination, intefacial_crack_growth, buckling, crack_deflection, spalling. Only the first generation of damage map, or spallation map, was implemented. The data on amplitude roughness, wavelength roughness, separated area, porosity fraction, and number cracks per unit volume will be required for the subsequent generations of spallation maps.

Typical recommended values: flaw_oxide_size_factor = 0.2, flaw_defect_factor = 1.0, Relationships: defect size = flaw_oxide_size_factor $*$ oxide_thickness

Table 6: Variables used in the DAMAGE namelist

Variable	Description	Type	Valid Options	Default values
flaw_oxide_size_factor	ratio between the flaw size and oxide thickness	real	less than 1	
flaw_defect_factor	factor dependent on shape, size, and position of the void	real	equal or less than 1	
delamination_size_existent	length [micron] of delamination used in buckling and crack_deflection criteria	real positive		
delamination_size_factor	ratio between the length of delamination, used in buckling and crack deflection criteria, and oxide thickness	real positive		

5 OXIDE_GROWTH

Typical recommended values: if_temp_steam_or_oxide .false., pbr_bernstein_factor = 0.1

Table 7: Variables used in the OXIDE_GROWTH namelist

Variable	Description	Type	Valid Options	Default values
oxide_thick_units	units for the oxide thickness	character	'micron' or 'm'	
activ_energy_oxide	activation energy that describe the temperature variation of oxide growth	real		
const_energy_oxide	constant 'A' that describe the oxide growth	real		
const_energy_units	units for the constant 'A'	character	'kJ/mole'	
oxide_rate_value	if activ_energy_oxide < 0, then the oxide rate value is given directly as a function of temperature	real positive		
oxide_rate_units	units for the oxide_rate_value	character	'micron2_hour' or 'm2_hour'	
oxide_rate_temp	temperature ^oC at which oxide_rate_value are given	real positive		
pilling_bedworth_ratio	volumetric ratio of oxide to consumed metal, Pilling Bedworth Ratio	real	equal or larger than 1	
if_temp_steam_or_oxide	option for choosing the temperature at which the oxide grows, T_{gr}, .true. indicate $T_{gr} = T_{steam}$, .false. indicate $T_{gr} = T_{oxide}$	logical	.true. or .false.	

6 PROPERTY - NAMELIST

This namelist defines each material and its properties.

Note: Either surf_fracture_energy_mat or (and) fracture_toughness_mat must be given. The other quantity is obtained from: fracture_toughness $= \sqrt{2.0 * surf\_fracture\_energy * Youngs\_modul}$

Table 8: Variables used in the PROPERTY namelist

Variable	Description	Type	Valid Options	Default values
material_name	name of material	character		
rho_mat	density [Kg/m^3	real		
poisson_ratio_mat	Poissson ratio	real	< 0.5	
cond_value_mat	thermal conductivity [W/m K]	real(30)		
cond_temp_mat	temperature ^oC at which cond_value_mat is given	real(30)		
cp_value_mat	Specific heat [J/kg K]	real(30)		
cp_temp_mat	temperature ^oC at which cp_value_mat is given	real(30)		
Youngs_modul_mat	Youngs modulus [MPa]	real(30)		
Youngs_temp_mat	Temperature ^oC at which Youngs_modul_mat values are given	real(30)		
surf_fracture_energy_mat	surface fracture energy [J/m^2]			
fracture_toughness_mat	Fracture toughness [MPa $\sqrt{m}$]			
th_exp_coeff_mat	linear thermal expansion coefficient	real(30)		
th_exp_temp_mat	temperature ^oC at which th_exp_coeff_mat is given			

7 OXIDE_LAYER - NAMELIST

This namelist describes each oxide layer.

Table 9: Variables used in the OXIDE_LAYER namelist

Variable	Description	Type	Valid Options	Default values
layer_name	name of oxide layer	character		
layer_materials_name	name of materials in the current oxide layer	character(30)		
thickness_fr	fraction of layer_materials_name material in this oxide layer	real	<= 1	

8 BOILER - NAMELIST

This namelist describes the operation schedule of the assembly where the tubes are used.

Table 10: Variables for tube dimensions and heat transfer coefficients used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
tube_outer_radius	outer radius of tube [m]	real		
tube_thickness	tube thickness [m]	real		
htc_tube_inner	heat transfer coefficient [W/m^{2}K] at the tube-steam surface, i.e., oxide surface	real		
htc_tube_outer	heat transfer coefficient [W/m^{2}K] hot gas surface, outer tube surface			

A cycle includes a transition from the low-load to high-load, high-load, transition from the high-load to low-load, and a low-load.

For a daily cycle, time_boiler_idle = 8.0, time_idle_pulse = 1.0, time_pulse_idle = 1.0, and time_boiler_pulse = 14 [h].

Table 11: Initial idle period before starting regular cycle: paremeters used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
time_boiler_first_idle	initial idle time period [h]	real		
temp_steam_first_idle	steam temperature [^{o}C]	real		
temp_gas_first_idle	gas temperature [^{o}C]	real		
press_inner_first_idle	pressure at the inner surface of the tube [MPa], i.e., steam pressure	real		
press_outer_first_idle	pressure at the outer surface of the tube [MPa], i.e., hot gas pressure	real		

Table 12: Duration of each load phase in a cycle schedule: variables used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
time_boiler_idle	low-load period [h]	real		
time_idle_pulse	transition period low- to high-load [h]	real		
time_boiler_pulse	high-load period [h]	real		
time_pulse_idle	transition period high- to low-load [h]	real		
no_total_pulse	number of total cycles	integer		

9 OUTPUT

During a simulation run, the data is written in '.aux' and '.out' files, i.e., problem_name.aux and problem_name.out, respectively. This section describes what data is written in output files, the order in which the data is written, and how the data can be retrieved.

For each type of output, each line starts with a specific character string that indicates what kind of data is written. The output data can be changed by changing the respective `write` statements.

The 'high' (or 'hig') and 'low' keywords in the character string indicates that the data is for the full-load and low-load domain, respectively. Each column is separated by one or more character spaces. Several scripts are provided to help with retrieving some data. By running the scripts described in this section, the data for each specified output will be extracted into separate data files, which can be easily displayed into Microsoft Excell, Kaleida Graph, and any other spreadsheet or plotting package that can accept import of data from a text file.

The following variables are available at the ouput. More detail on the stress-strain equations

Table 13: High-load operation schedule variables used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
temp_steam_pulse	steam temperature [^{o}C]	real(2000)		
temp_gas_pulse	gas temperature [^{o}C]	real(2000)		
press_inner_pulse	pressure at the inner surface of the tube [MPa], i.e., steam pressure	real(2000)		
press_outer_pulse	pressure at the outer surface of the tube [MPa], i.e., hot gas pressure	real(2000)		

Table 14: Last time period after ending the regular cycle: paremeters used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
time_boiler_last_idle	last idle time period [h]	real		
temp_steam_last_idle	steam temperature [^{o}C]	real		
temp_gas_last_idle	gas temperature [^{o}C]	real		
press_inner_last_idle	pressure at the inner surface of the tube [MPa], i.e., steam pressure	real		
press_outer_last_idle	pressure at the outer surface of the tube [MPa], i.e., hot gas pressure	real		

and the variable definition, please see the paper by Sabau and Wright (2008), distributed with this manual.

- ϵ_θ^s - hoop strain component that generates stress,
- σ_θ - hoop stress [MPa],
- ϵ_θ) - total hoop strain,
- ϵ_{th} - thermal expansion strain,
- $\sigma_{th,\theta}$ - equivalent hoop stress [MPa] for flat plate assumption due only to thermal expansion, i.e., $\sigma_{th,\theta} = E(\epsilon_{th,metal} - \epsilon_{th,oxide})/(1-\nu)$

Table 15: Output variables used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
no_pulse_full2low_output	Number of cycle at which data is written in output files for (a) the transition full-load to partial-load, (b) spatial distribution of stress and strain	integer(20)		

9.1 Spallation map data

In the first generation of spallation maps, the critical strains, for each damage criteria, and the strain evolution in the oxide scale are plotted as a function of the oxide thickness.

9.1.1 Damage criteria

Data on damage criteria can be obtained from '.out' file by using the command lines below. Other damage criteria can be added and/or formulae changed in the subroutine `CRITICAL_ETA_OMEGA` .

- Damage criteria for the entire oxide layers, are obtained by averaging the mechanical properties for all oxide layers. This is obtained since most experimental data is for the entire oxide as a whole not for each individual oxide layer. Each line containing the results for the entire oxide scale starts with 'damage1_ave'.

  ```
  grep damage1_ave spmlgp3.out > dm_ave_spmlgp3.txt
  ```

  The resulting output file dm_ave_spmlgp3.txt has the following columns:

  1. damage1_ave character string,
  2. $d_{ox}[\mu m]$,
  3. $10^3\epsilon_c$ for through_scale_cracking, where $\epsilon_c = K_{Ic}/(E\ FDF\ \sqrt{\pi\ f\ d_{ox}})$ and E= Youngs modulus, FDF = flaw_defect_factor, f=flaw_oxide_size_factor,
  4. $10^3\epsilon_c$ for deflection_delamination, where $\epsilon_c = 2K_{Ic}/(E\ FDF\ \sqrt{\pi\ f\ d_{ox}})$,
  5. $10^3\epsilon_c$ for intefacial_crack_growth, where $\epsilon_c = -0.5K_{Ic}(1+\nu)/(E\ FDF\ \sqrt{\pi\ f\ d_{ox}})$,
  6. $10^3\epsilon_c$ for buckling, where $\epsilon_c = -1.22(d_{ox}/DS)^2/(1-\nu)^2$, with DS = delamination_size,
  7. $10^3\epsilon_c$ for crack_deflection, where $\epsilon_c = -3.6(d_{ox}/DS)^2$, and
  8. $10^3\epsilon_c$ for spalling, where $\epsilon_c = -\sqrt{2\gamma_s/d_{ox}\ E\ (1-\nu)}$, with γ_s= surf_fracture_energy.

- Damage criteria can be obtained for each individual oxide layer.

```
grep damage1_lay spmlgp3.out > dm_lay_spmlgp3.txt
```

The resulting output file dm_lay_spmlgp3.txt has the following columns. The first two columns are damage1_lay and $d_{ox}[\mu m]$. Then, for each oxide layer, the thickness of the oxide layer $[\mu m]$ and the six critical strains are given in the order specified above. Thus, for N layers, including the substrate, the total number of colums is $7N$ - 5.

9.1.2 Strain evolution

The data on strain evolution in the oxide scale can be obtained as follows.

- Maximum strain for the entire oxide scale. This is obtained since most experimental data is for the entire oxide as a whole not for each individual oxide layer. Each line containing the results for the entire oxide scale starts with 'strain_ave_map_low' or 'strain_ave_map_hig' for the partial load and full-load, respectively.

  ```
  grep strain_ave_map_low spmlgp3.aux > strain_ave_dm_spmlgp3l.txt
  grep strain_ave_map_hig spmlgp3.aux > strain_ave_dm_spmlgp3h.txt
  ```

  The resulting output files strain_ave_dm_spmlgp3l.txt and strain_ave_dm_spmlgp3h.txt have the following columns:

  1. character string,
  2. Time [h]
  3. $d_{ox}[\mu m]$,
  4. $10^3 MAX(\epsilon_\theta^s)$

- Maximum strain can be obtained for each individual oxide layer.

  ```
  grep strain_lay_map_low spmlgp3.aux > strain_dm_spmlgp3l.txt
  grep strain_lay_map_hig spmlgp3.aux > strain_dm_spmlgp3h.txt
  ```

  The resulting output files strain_dm_spmlgp3l.txt and strain_dm_spmlgp3h.txt have the following columns. The first two columns are character string and $d_{ox}[\mu m]$. Then, for each oxide layer two variables are available, the thickness of the oxide layer $[\mu m]$ and the maximum strain $10^3 MAX(\epsilon_\theta^s)$. Thus, for N layers, including the substrate, the total number of colums is $2N$ +1.

9.2 Data on spatial profile

Data on hoop strain and hoop stress as a function of radius is obtained as follows. The data is written out only at the outer surface, metal-oxide interface, and oxide-oxide interfaces. Since the hoop strains and stresses are double valued at the interface, two data lines are written for each

interface location, i.e., one for each side of the interface. Thus, for a total number of layers, N (one metal substrate and $N-1$ oxide layers), there will be $2N$ lines of data written out at each given time/cycle.

This data is available at the cycle numbers given by the no_pulse_full2low_output. Two data files are generated using the scripts below for each for each operation cycle, one for full-load and another one for partial-load. Thus, the number of output files is equal to twice the numbers of cycles given in no_pulse_full2low_output. The intermediate file names are 'ProblemName_lprof.txt' and 'ProblemName_hprof.txt' The resulting file names are 'ProblemName_lproff#.txt' and 'ProblemName_hproff#.txt' for the low-load and high-load, respectively, where # indicates the number of output.

Each line starts with either int_jump_profile_high or int_jump_profile_low character string for the full-load and low-load, respectively. Each line contains the relevant data in the 3-rd through 6-th column as follows (the first line below is the header):

character_string id Time oxide_thickness radius hoop_strain hoop_stress
character_string id Time[h] $d_{ox}[\mu m]$ $r[cm]$ $10^3\epsilon_\theta^s$ $\sigma_\theta[MPa]$

- one oxide layer example, problem name 'spslgp4'.

```
grep int_jump_profile_low spslgp4.aux | grep -v id_time > spslgp4_lprof.txt
awk '{print > ("spslgp4_lprof" int((NR+3)/4))".txt"}' spslgp4_lprof.txt
grep int_jump_profile_hig spslgp4.aux | grep -v id_time > spslgp4_hprof.txt
awk '{print >("spslgp4_hprof" int((NR+3)/4))".txt"}' spslgp4_hprof.txt
get_profile spslgp4 fig_prof_header_1layerl.txt fig_prof_header_1layerh.txt
```

- two oxide layers example, problem name 'spmlgp3'

```
grep int_jump_profile_low spmlgp3.aux | grep -v id_time > spmlgp3_lprof.txt
awk '{print >("spmlgp3_lprof" int((NR+5)/6))".txt"}' spmlgp3_lprof.txt
grep int_jump_profile_hig spmlgp3.aux | grep -v id_time > spmlgp3_hprof.txt
awk '{print >("spmlgp3_hprof" int((NR+5)/6))".txt"}' spmlgp3_hprof.txt
get_profile spmlgp3 fig_prof_header_1layerl.txt fig_prof_header_1layerh.txt
```

- three oxide layers example, problem name 'spmlg4'

```
grep int_jump_profile_low spmlgp4.aux | grep -v id_time > spmlgp4_lprof.txt
awk '{print >("spmlgp4_lprof" int((NR+7)/8))".txt"}' spmlgp4_lprof.txt
grep int_jump_profile_hig spmlgp4.aux | grep -v id_time > spmlgp4_hprof.txt
awk '{print >("spmlgp4_hprof" int((NR+7)/8))".txt"}' spmlgp4_hprof.txt
get_profile spmlgp4 fig_prof_header_1layerl.txt fig_prof_header_1layerh.txt
```

Files `fig_prof_header_1layerh.txt` and `fig_prof_header_1layerl.txt` contain one line to provide the header for files. It also provides the legend for plotting packages.

9.3 Output data as a function of oxide thickness

For obtaining the strain-stress variation as a function of oxide thickness, a UNIX script is used to extract the data at the cycles given in no_pulse_full2low_output for the maximum and minimum of selected variables in each layer. Each line starts with either table_prop_layer_hoop_gen_hig or table_prop_layer_hoop_gen_low character string for the full-load and low-load, respectively. This type of output described in this section is referred to as tabular output.

Table 16: Data format: Maximum and minimum data for different oxide thicknesses

Columns	Description
1-2	character string and an internal id
3-6	Time[h], $d_{ox}[\mu m]$, gas temperature [^{o}C], steam temperature [^{o}C]
7 through 4+2 N	$MAX(T)$ [^{o}C], $MIN(T)$ [^{o}C] per each oxide layer
5 +2 N through 2 + 4 N	($10^3 MAX(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$ per each oxide layer
3 + 4 N through 6 N	$MAX(\sigma_\theta)$, $MIN(\sigma_\theta)$ per each oxide layer
1 + 6 N through -2 + 8 N	$10^3 MAX(\epsilon_\theta)$, $10^3 MIN(\epsilon_\theta)$ per each oxide layer
-1 + 8 N through -4 + 10 N	$10^3 MAX(\epsilon_{th})$, $10^3 MIN(\epsilon_{th})$ per each oxide layer
-3 + 10 N through -6 + 12 N	$MAX(\sigma_{th,\theta})$, and $MIN(\sigma_{th,\theta})$ per each oxide layer

The 2-nd layer is the oxide grown next to the metal substrate and the N-th layer is the last oxide layer, whose surface is in contact with the steam.

For example, for one oxide layer (N=2), the following data is written in columns 7 through 18: $MAX(T)$ [^{o}C], $MIN(T)$ [^{o}C], $10^3 MAX(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $MAX(\sigma_\theta)$, $MIN(\sigma_\theta)$, $10^3 MAX(\epsilon_\theta)$, $10^3 MIN(\epsilon_\theta)$, $10^3 MAX(\epsilon_{th})$, $10^3 MIN(\epsilon_{th})$, $MAX(\sigma_{th,\theta})$, and $MIN(\sigma_{th,\theta})$. For two oxide layers, N=3, and the following data is written in columns 7 through 30, respectively: $MAX(T)$ [^{o}C], $MIN(T)$ [^{o}C], $MAX(T)$ [^{o}C], $MIN(T)$ [^{o}C], $10^3 MAX(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $MAX(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MAX(\sigma_\theta)$, $MIN(\sigma_\theta)$, $10^3 MAX(\epsilon_\theta)$, $10^3 MIN(\epsilon_\theta)$, $10^3 MAX(\epsilon_\theta)$, $10^3 MIN(\epsilon_\theta)$, $10^3 MAX(\epsilon_{th})$, $10^3 MIN(\epsilon_{th})$, $10^3 MAX(\epsilon_{th})$, $10^3 MIN(\epsilon_{th})$, $MAX(\sigma_{th,\theta})$, $MIN(\sigma_{th,\theta})$, $MAX(\sigma_{th,\theta})$, and $MIN(\sigma_{th,\theta})$.

The tabular scripts are used as follows:

- one oxide layer example, problem name 'spslgp4'
  `get_tab1 spslgp4 table_header_1layer.txt`
- two oxide layers example, problem name 'spmlgp3'
  `get_tab1 spmlgp3 table_header_2layers.txt`

- three oxide layers example, problem name 'spmlg4'
  `get_tab1 spmlgp4 table_header_3layers.txt`

The scripts will generate two new files. The file names are 'ProblemName_ltab.txt' and 'ProblemName_htab.txt' for the low-load and high-load, respectively.

9.4 Output data for the transition full-load to partial-load

The evolution of strain-stress variables during the transition full-load to partial-load can obtained using the scrips shown in this subsection. Here it is considered that the data is written at 60 equally time intervals during the transition full-to-partial load, i.e., with a time increment of time_pulse_idle/60 [h]. Each line starts with f2l_fig_data.

Table 17: Data format: Maximum and minimum data for different oxide thicknesses

Columns	Description
1- 4	character string, internal id, transition_time [h], $d_{ox}[\mu m]$
5 through 4 + N	$10^3 MAX(\epsilon^s_\theta)$ per each layer
5+N through 4+2 N	$10^3 AVE(\epsilon^s_\theta)$ per each layer
5+2N through 4+3 N	$10^3 MIN(\epsilon^s_\theta)$ per each layer
5+3N through 4+4 N	$MAX(\sigma_\theta)$ per each layer
5+4N through 4+5 N	$AVE(\sigma_\theta)$ per each layer
5+5N through 4+6 N	$MIN(\sigma_\theta)$ per each layer

where, $AVE(q)$ indicates an AVE of quantity q, defined as $AVE(q) = \sum_{k=1}^{M_i} q_k/M_i$.

The 1-st layer is the metal substrate, 2-nd layer is the oxide grown next to the metal substrate, and the N-th layer is the last oxide layer, whose surface is in contact with the steam.

For example, for one oxide layer (N=2), the following data is written in columns 5 through 16: $10^3 MAX(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $MAX(\sigma_\theta)$, $MAX(\sigma_\theta)$, $MAX(\sigma_\theta)$, $AVE(\sigma_\theta)$, $AVE(\sigma_\theta)$, $AVE(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MIN(\sigma_\theta)$. For two oxide layers, N=3, and the following data is written in columns 5 through 22, respectively: $10^3 MAX(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 MAX(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 AVE(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $10^3 MIN(\epsilon^s_\theta)$, $MAX(\sigma_\theta)$, $MAX(\sigma_\theta)$, $MAX(\sigma_\theta)$, $MAX(\sigma_\theta)$, $AVE(\sigma_\theta)$, $AVE(\sigma_\theta)$, $AVE(\sigma_\theta)$, $AVE(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MIN(\sigma_\theta)$, $MIN(\sigma_\theta)$.

The full-to-low load scripts can be used as follows:

- one oxide layer example, problem name 'spslgp4'

```
grep f2l_fig_data spslgp4.aux | grep -v id_f2l > spslgp4_f2l.txt
```

```
awk '{print >("spslgp4_f2l" int((NR+60)/61))".txt"}' spslgp4_f2l.txt
getf2l spslgp4 fig_f2l_header_1layer.txt
```

- two oxide layers example, problem name 'spmlgp3'

```
grep f2l_fig_data spmlgp3.aux | grep -v id_f2l > spmlgp3_f2l.txt
awk '{print >("spmlgp3_f2l" int((NR+60)/61))".txt"}' spmlgp3_f2l.txt
getf2l spmlgp3 fig_f2l_header_2layers.txt
```

- three oxide layers example, problem name 'spmlg4'

```
grep f2l_fig_data spmlgp4.aux | grep -v id_f2l > spmlgp4_f2l.txt
awk '{print >("spmlgp4_f2l" int((NR+60)/61))".txt"}' spmlgp4_f2l.txt
getf2l spmlgp4 fig_f2l_header_3layers.txt
```

This data is available at the cycle numbers given by the no_pulse_full2low_output. One data files is generated using the scripts below for each for each operation cycle. Thus, the number of output files is equal to the numbers of cycles given in no_pulse_full2low_output. The resulting file names are 'ProblemName_f2lf#.txt', where # indicates the number of output.

9.5 List of files provided for output processing

All the input files, header files, script files, and a help file, `help_spallmap.txt`, which contains all the command line instructions shown above are provided in the `problems` directory.

```
fig_prof_header_1layerl.txt  spmlgp4.inp
getf2l                       spslgp4.inp
fig_f2l_header_1layer.txt    get_profile                table_header_1layer.txt
fig_f2l_header_2layers.txt   get_tab1                   table_header_2layers.txt
fig_f2l_header_3layers.txt   help_spallmap.txt           table_header_3layers.txt
fig_prof_header_1layerh.txt  spmlgp3.inp                 update
```

9.6 UNIX scripts for output processing

- getf2l script file

```
i=1
while [ $i -lt 7 ]
do
  echo I is $i
  echo "file $1_f2l$i.txt - $1_f2lf$i.txt"
  cat $2 $1_f2l$i.txt > $1_f2lf$i.txt
  rm -f $1_f2l$i.txt
  i=‘expr $i + 1‘
done
```

- get_tab1 script file

```
# getting the table data for full load
grep table_prop_layer_hoop_gen_hig $1.aux | grep -v Time > $1_tab.tx
cat $2 $1_tab.tx > $1_htab.txt
rm -f $1_tab.tx
# getting the table data for partial load
grep table_prop_layer_hoop_gen_low $1.aux | grep -v Time > $1_tab.tx
cat $2 $1_tab.tx > $1_ltab.txt
rm -f $1_tab.tx
```

- get_profile script file

```
i=1
while [ $i -lt 7 ]
do
  echo I is $i
  echo "file $1_lprof$i.txt - $1_lproff$i.txt"
  cat $2 $1_lprof$i.txt > $1_lproff$i.txt
  rm -f $1_lprof$i.txt
  cat $3 $1_hprof$i.txt > $1_hproff$i.txt
  rm -f $1_hprof$i.txt
  i=‘expr $i + 1‘
done
```