User Manual for SPALLOOP

Adrian S. Sabau
Bldg. 5700, MS 6085
Oak Ridge National Laboratory
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 tube blockage due to exfoliation events in order to inform power plant operators of possible tube blockages. The SPALLation LOOP (SPALLOOP[1]) code contains modules developed for oxide growth, area exfoliated during boiler shut-downs, by taking into account the following phenomena and features:

- Temperature gradient through the tube circumference
- Non-uniform thermal expansion coefficient of oxides and metal substrates,
- Plant operation schedule with periodic alternate full-load and partial-load regimes,
- Flue temperature and/or heat flux distribution due to flue gases along and entire length of a boiler tube.
- Axisymmetric formulation for cylindrical tubes,
- Multiple oxide layers,
- Different growth temperatures for the spinel and magnetite,
- Oxide-growth induced stresses,

- Blockage model providing estimates for the amount of tube cross-sectional area blocked by the exfoliated oxide, and
- One or two tube bends, differentiating between blockages in the inlet loop bend and outlet loop bend.

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.

The code was used to obtain the data in Sabau et al. (2012), where the model implementation for the blockage module is described. Sabau A.S., I. G, Wright, and Shingledecker, J.P., "Oxide Scale Exfoliation and Regrowth in TP347H Superheater Tubes," Materials and Corrosion, Vol. 63, pp. 896-908, 2012. The code implementation of the creep behavior is described in detail in Sabau and Wright (2010). Sabau, A.S. and Wright, I.G. "Influence of Oxide Growth and Metal Creep on Strain Development in the Steam-side Oxide in Boiler Tubes," Oxidation of Metals, Vol. 73, pp. 467-492, 2010.

1 INSTALLATION

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 `spalloop1o/options` directory, as follows.

- blas and lapack libraries are needed; change file `spalloop1o.libs` accordingly. For most operating systems, blas and lapack routines are installed. If they are not already installed, which would be a rare occurrence, 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 `spalloop1o.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 spalloop1o
```

Other compiler options are available by reading `spalloop1o/GNUmakefile`. The code is build in `spalloop1o/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 `spalloop1o/src`. The source code files are grouped according to the physics and/or their function. Test problems are supplied with this distribution in the `spalloop1o/tests` or `spallation/tests/problems` directory. Then, either the binary `spalloop1o` is copied to the `spalloop1o/tests` directory or the directory `spalloop1o/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 `spalloop1o/tests/problems` directory, can be used to update the code after changes are made to the surce code.

```
cd ../..
gmake spalloop1o
cd tests/problems
cp ../../bin/spalloop1o .
```

A problem is run by typing the following command line:

```
spalloop1o problem_name
```

or,

```
spalloop1o
```

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.

An example of input file is provided in `spalloop1o_input_files_examples` directory.

Based on the temperature and pressure schedule, the operation is a succession 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
OXIDE_LAYER	how each oxide layer is made
OXIDE_GROWTH	how each oxide layer is made
DAMAGE	damage parameters, not used in spalloop
SOLVER	solver, physics, mesh, spallation map
PROPERTY	properties of each material
BOILER	Operation schedule
OUTPUT	specifies output files for strain, energy, outage, stress, and general output
WATERWALL	specifies tube length, radius of tube, tube thickness, tube orientation, output locations, steam inlet, and flue gas conditions along the tube length,
BLOCKAGE	specifies how exfoliated area is calculated as a function of elastic energy in the oxide, porosity of deposits from exfoliated oxide, oxide regrowth after exfoliation events, and bend geometry for obtaining the tube blockage in those tube bends.

Several examples of the input files will be provided (not yet included in the distribution) 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 DAMAGE - namelist to be removed from spalloop code

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.

Table 2: 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		

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

4 OXIDE_GROWTH

Table 3: 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.	

Typical recommended values: if_temp_steam_or_oxide .false., pbr_bernstein_factor = 0.1, growth_induce_stress = 'no' for NOT including the oxide-growth induced stresses, if_th_oxide_growth = .false. (otherwise, the coefficient of thermal expansion (CTE) is taken into account in oxide growth kinetics). Other notes: if_th_tube = .false., rad_int(1) = tube_outer_radius, rad_int(2) = rad_int(1) - tube_thickness. Otherwise, the CTE of metal should be considered for the tube dimensions. if_exclude_th_strain_trace = .true., then the trace_strain will exclude the thermal expansion component (i.e., e_r + e_theta + e_z - 3 * alfa * delta_T).

5 PROPERTY - NAMELIST

This namelist defines each material and its properties.

Table 4: 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			

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

6 OXIDE_LAYER - NAMELIST

This namelist describes each oxide layer.

Table 5: 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	

7 BOILER - NAMELIST

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

Table 6: 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 full-load, full-load, transition from the full-load to low-load, and a low-load. The initial idle period before starting regular cycle is considered to be low-load.

Table 7: Duration of each load phase in a cycle schedule: variables used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
time_low_load	low-load period [h]	real		
time_low2full_load	transition period low- to high-load [h]	real		
time_full_load	high-load period [h]	real		
time_full2low_load	transition period high- to low-load [h]	real		
number_load_cycles	number of total cycles	integer		

For a weekly cycle, time_low_load = 21, time_low2full_load = 1.0, time_full2low_load = 1.0, and time_full_load = 145 [h].

The array outage_time_interval provides the interval between outages. The **outage** variables for the temperature and pressure are similar, only that _full_load is replaced with keyword 'outage'. In addition, the outage duration and transition to/from outage to full load is given by time_outage_duration = 20.0 [h], time_full2outage = 100.0 [h], time_outage2full = 6.0 [h] (values given as an example).

Table 8: Low-load operation schedule variables used in the BOILER namelist

Variable	Description	Type	Valid Options	Default values
steam_temperature_low_load	steam temperature [^{o}C]	real		
heat_flux_fraction_low_load	heat flux reduction from full-load [fraction]	real		
furnace_temperature_low_load	gas temperature [^{o}C]	real		
pressure_steam_low_load	pressure at the inner surface of the tube [MPa], i.e., steam pressure	real		
pressure_furnace_low_load	pressure at the outer surface of the tube [MPa], i.e., hot gas pressure	real		

Table 9: Full-load operation schedule variables used in the BOILER namelist.

Variable	Description	Type	Valid Options	Default values
steam_temperature_full_load	steam temperature [^{o}C]	array real		
furnace_temperature_full_load	gas temperature [^{o}C]	array real		
pressure_steam_full_load	pressure at the inner surface of the tube [MPa], i.e., steam pressure	array real		
pressure_furnace_full_load	pressure at the outer surface of the tube [MPa], i.e., hot gas pressure	array real		

8 SOLVER - NAMELIST - to be updated - for now see input file provided

Table 10: 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 [^oC]	real		
temp_gas_last_idle	gas temperature [^oC]	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		

Table 11: Output variables used in the BOILER namelist

Variable	Description	Type	Valid Options
cycle_number_output_full2low	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	

9 BLOCKAGE

length_deposit_over_id_tube = gives the length of a deposit as: length_deposit= length_deposit_over_id_tube * id_tube

Table 12: 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 13: 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	

10 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. 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 Excel, KaleidaGraph, and any other spreadsheet or plotting package that can accept import of data from a text file.

Table 14: 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 15: 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	

The following variables are available at the output. More detail on the stress-strain equations and the variable definition, please see the paper by Sabau and Wright (2008), distributed with this manual.

- ϵ^s_θ - 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., $\sigma_{th,\theta} = E(\epsilon_{th,metal} - \epsilon_{th,oxide})/(1-\nu)$

The script **get_block2** is used to post process the data, i.e., to obtain the output for the following variables:

- the maximum elastic energy during a shut-down event and the area exfoliated. The code writes this info using this format:

```
! j = 1 indicates the total energy; as j=1 would indicate metal
! j = 1 the entire scale can exfoliate (all layers)
! j = 3 only magnetite exfoliation (the top first layer)
! (j=2 is spinel; but spinel cannot exfoliate without magnetite)
   write(enrg_lun, 22) i, length_tube_oxide_output(i), &
    & (max_energy_elastic_event_block(i, j, 3), area_exfoliated_fr_block(i, j, 3),
    & j = 1, no_f2l_event_out), &
    & (max_energy_elastic_event_block(i, j, 1), area_exfoliated_fr_block(i, j, 1),
    & j = 1, no_f2l_event_out)
  end do
22  format('energy_armitt_areaex ', i2, 1x, 100(1pe13.6, 1x))
```

- temperature and oxide thickness at outages. The following variables, are written: time, height, total oxide thickness, thickness of 1-st oxide layer, thickness of 2-nd oxide layer, steam temperature, metal mean temperature, metal maximum temperature, flue gas temperature, and heat flux, using the format:

```
4 format('Tsteam_dox_height_event ', i2, ' t h dox d_sp d_mag ', 'Tst Tmw Tmax Tg hf')
  write(enrg_lun, 2)  n1, Time_Boiler_p(i_time), height_now, &
   & SUM(thickness_oxide_height_time(first_oxide_layer: no_oxide_layers, &
   &  j, id_time(n1) - 1)), &
   &  thickness_oxide_height_time(first_oxide_layer, j, id_time(n1) - 1), &
   &  thickness_oxide_height_time(no_oxide_layers, j, id_time(n1) - 1), &
   &  Temp_steam_height_time(j, id_time(n1), k), &
   &  Temp_metal_mean_height_out(n1, j), &
   &  Temp_metal_max_height_out(n1, j), Temp_gas_height(j, k), &
   &  heat_flux_height_out(n1, j)
2       format('Tsteam_dox_height_event ', i2, 20(1pe13.6, 1x))
```

- fraction of skin exfoliated and fraction of oxide skin remained after each outage.

```
 character(LEN = 1), dimension(0:9):: ch_index = (/'0','1','2','3','4','5', '6','7','8','9'/)
 write(enrg_lun, 91) 'exfol_fr_skinoutage height ', &
   & (' efr_skin'//ch_index(j), ' skin1_on'//ch_index(j), &
   & j = 1, MIN(9, no_f2l_event_out)), &
   & (' efr_skin'//'1'//ch_index(j-10), ' skin1_on'//'1'//ch_index(j-10), &
   & j = 1 + MIN(9, no_f2l_event_out), no_f2l_event_out)
 91 format(150(a))
    write(enrg_lun, 92) length_tube_oxide_output(j), &
     & (exfoliate_fr_skin_outage(j, i), &
     & fraction_skin_remain_outage(j, i), i = 1, no_f2l_event_out)
92    format('exfol_fr_skinoutage ', 100(1pe13.6, 1x))
```

- oxide thickness before and after each outage.

```
    ! magnetite before exfoliation
    dox_before_exfol(2, iheight, ioutage) = dthick_ox(iheight, ioutage_old, imagnetite) + &
       & thickness_new_ox_growth_init(iheight, ioutage_old, imagnetite)
    ! spinel before exfoliation
    dox_spinel = dthick_ox(iheight, ioutage_old, first_oxide_layer) + &
       & thickness_new_ox_growth_init(iheight, ioutage_old, first_oxide_layer)
    ! entire oxide thickness before exfoliation
    dox_before_exfol(1, iheight, ioutage) = dox_spinel + &
        & dox_before_exfol(2, iheight, ioutage)
    ! magnetite after exfoliation
    dox_after_exfol(2, iheight, ioutage) = dox_before_exfol(2, iheight, ioutage)
    ! entire oxide thickness after exfoliation
    dox_after_exfol(1, iheight, ioutage) = dox_before_exfol(1, iheight, ioutage)

  write(enrg_lun, 93) 'dox_outage height ', (' sp'//ch_index(j), &
    & ' mag_b'//ch_index(j), ' mag_a'//ch_index(j), &
    & ' dox_b'//ch_index(j), ' dox_a'//ch_index(j), j = 1, MIN(9, no_f2l_event_out)), &
    & (' sp'//'1'//ch_index(j-10), ' mag_b'//'1'//ch_index(j-10), &
    & ' mag_a'//'1'//ch_index(j-10), ' dox_b'//'1'//ch_index(j-10), &
    & ' dox_a'//'1'//ch_index(j-10), &
    & j = 1 + MIN(9, no_f2l_event_out), no_f2l_event_out)
  93 format(150(a))
     write(enrg_lun, 94) length_tube_oxide_output(j), &
       & (dox_before_exfol(1, j, i) - dox_before_exfol(2, j, i), &
       & dox_before_exfol(2, j, i), dox_after_exfol(2, j, i), &
       & dox_before_exfol(1, j, i), dox_after_exfol(1, j, i), &
       & i = 1, no_f2l_event_out)
 94    format('dox_outage ', 100(1pe13.6, 1x))
```

- average von Mises stresses and creep strain in the metal and oxide scale before and after exfoliation.

```
  ! store average quantities before exfoliation
  if (if_creep_metal)  then
    creep_strain_before_exfol(1, iheight, ioutage) = creep_strain_ave(1, 3)
    creep_strain_eq_before_exfol(1, iheight, ioutage) = creep_strain_eq_ave(1, 3)
  endif
  if (if_creep_ox_scale)  then
    creep_strain_before_exfol(2, iheight, ioutage) = creep_strain_ave(imagnetite, 3)
    creep_strain_eq_before_exfol(2, iheight, ioutage) = creep_strain_eq_ave(imagnetite, 3)
  endif

  character(LEN = 1), dimension(0:9):: ch_index = (/'0','1','2','3','4','5','6','7','8','9'/)
  write(enrg_lun, 93) 'vm_creep_outage height ', (' vm_met'//ch_index(j), &
    & ' vm_mag_b'//ch_index(j), ' vm_mag_a'//ch_index(j),
    & ' scr_hoop_met'//ch_index(j), &
    & ' scr_hoop_mag_b'//ch_index(j), ' scr_hoop_mag_a'//ch_index(j), &
    & j = 1, MIN(9, no_f2l_event_out)), &
    & (' vm_met'//'1'//ch_index(j-10), ' vm_mag_b'//'1'//ch_index(j-10), &
```

```
      & ' vm_mag_a'//'1'//ch_index(j-10),  ' scr_hoop_met'//'1'//ch_index(j-10), &
      & ' scr_hoop_mag_b'//'1'//ch_index(j-10), ' scr_hoop_mag_a'//'1'//ch_index(j-10), &
      & j = 1 + MIN(9, no_f2l_event_out), no_f2l_event_out)
 93 format(150(a))
     write(enrg_lun, 94) length_tube_oxide_output(j), &
       & (stress_eq_ave_before_exfol(1, j, i), stress_eq_ave_before_exfol(2, j, i), &
       & stress_eq_ave_after_exfol(1, j, i),
       & creep_strain_before_exfol(1, j, i)%hoop, &
       & creep_strain_before_exfol(2, j, i)%hoop,
       & creep_strain_after_exfol(2, j, i)%hoop, &
       & i = 1, no_f2l_event_out)
 94    format('vm_creep_outage ', 100(1pe13.6, 1x))
```

- min, max, average von Mises stresses and hoop creep strain in the metal and oxide scale at fixed locations along the tube.

```
 operation_type(1) = 'max'
 operation_type(2) = 'min'
 operation_type(3) = 'ave_sum'
 do i = i_cr_start, i_cr_end
     do k = 1, 3
       ! get maximum, minimum, and average von Misses stress
       ! and hoop creep strain in each layer
       call OPERAND_ARRAY(1, mgrid, creep_strain_new(i, :)%hoop,  &
             & 1, npr_st(i), operation_type(k), creep_strain_ave(i, k)%hoop, loc1)
       call OPERAND_ARRAY(1, mgrid, stress_eq(i, :),  &
             & 1, npr_st(i), operation_type(k), stress_eq_ave(i, k), loc1)
     end do
   end do

      write(enrg_lun, 18) jheight_creep, ioutage_creep, cycle_no, time_now, &
           & (rad_int(first_oxide_layer) - rad_int(N+1)) / oxide_thick2si, &
           & stress_eq_ave(1, 2), stress_eq_ave(1, 3), stress_eq_ave(1, 1), &
           & stress_eq_ave(N, 2), stress_eq_ave(N, 3), stress_eq_ave(N, 1), &
           & creep_strain_ave(1, 2)%hoop, creep_strain_ave(1, 3)%hoop, &
           & creep_strain_ave(1, 1)%hoop, creep_strain_ave(N, 2)%hoop, &
           & creep_strain_ave(N, 3)%hoop, creep_strain_ave(N, 1)%hoop
 18    format('vm_height_cr_allt ', 2(i2, 1x), i5, 1x, 70(1pe13.6, 1x))
```

- total oxide thickness, spinel thickness, magnetite thickness, steam temperature at each height for all outages.

```
   ch_repeat = ' d_ox d_sp d_mag Tst'
   write(enrg_lun, 3) (ch_repeat(1:20)//ch_index(j), j = 1, MIN(9, no_outages)), &
        & (ch_repeat(1:20)//'1'//ch_index(j-10),
        & j = MIN(9, no_outages) + 1, no_outages)
 3  format('Tsteam_dox_height_final j h Tg ', 100(a))

     write(enrg_lun, 1)  j, height_now, Temp_gas_height(j, k), &
```

```
        & (SUM(thickness_oxide_height_time(first_oxide_layer: no_oxide_layers, &
        & j, id_time(n1) - 1)), thickness_oxide_height_time(first_oxide_layer, &
        & j, id_time(n1) - 1), thickness_oxide_height_time(first_oxide_layer+1, &
        & j, id_time(n1) - 1), Temp_steam_height_time(j, id_time(n1), k), &
        & n1 = 1, no_outages + 1)
 1    format('Tsteam_dox_height_final ', i2, 1x, 100(1pe13.6, 1x))
```

- oxide thickness in each layer at each height for all outages.

```
!at each exfoliation event for investigating possible blockage store
! the oxide thickness
 oxide_thickness_block(k_height, id_next_outage, :) = &
          & oxide_thickness_event(id_next_outage, :)
    write(enrg_lun, 27) j, j - j, (INT(time_event(j)), j = 1, no_f2l_event_out)
  27 format('#oxide_vs_height_outage ', i2, 1x, 20(i6, 1x))
  do k = 1, no_height_oxide_output
     write(enrg_lun, 26) k, length_tube_oxide_output(k), &
       & (1.0e+6*SUM(oxide_thickness_block(k, j, first_oxide_layer:no_oxide_layers)), &
       & j = 1, no_f2l_event_out)
   end do
  26 format('oxide_vs_height_outage ', i2, 1x, 20(1pe13.6, 1x))
```

- final exfoliated volume and mass per each tube loop (bend).

```
if (ioutage == no_f2l_event_out .and. iheight == no_height_oxide_output)  then
    if (no_loops == 1)  then
      write(enrg_lun, 7) 'v_all1 v_mag1'
    else if (no_loops == 2)  then
      write(enrg_lun, 7)  'm_all_dry2 m_mag_dry2 m_all_wet2 m_mag_wet2 ', &
      &  'v_all1 v_mag1 v_all2 v_mag2'
    endif

  7 format('mass_vol_deposit_total_shd outage t[h] t[d] t[w] ', &
      &  'm_all_dry1 m_mag_dry1 m_all_wet1 m_mag_wet1 ', 2(a))

    ! final exfoliated volume, mass,
     do j = 1, no_f2l_event_out
       mass_dry_all =  volume_all1 * deposit_oxide_density(1) * 1.0e+3
       mass_wet_all =  mass_dry_all * (1.0 + &
         & deposit_fluid_density(1) * deposit_porosity_fraction(1) / &
         & (deposit_oxide_density(1) * (1.0 - deposit_porosity_fraction(1))))
       mass_dry_mag =  volume_mag1 * deposit_oxide_density(1) * 1.0e+3
       mass_wet_mag =  mass_dry_mag * (1.0 + &
         & deposit_fluid_density(1) * deposit_porosity_fraction(1) / &
         & (deposit_oxide_density(1) * (1.0 - deposit_porosity_fraction(1))))
       write(enrg_lun, 6) j, time_event(j), time_event(j) / 24.0,
         & time_event(j) / (24.0 * 7.0), &
         & (mass_dry_all(i), mass_dry_mag(i), mass_wet_all(i),
         & mass_wet_mag(i), i= 1, no_loops), &
```

```
        & (volume_all1(i),  volume_mag1(i), i= 1, no_loops)
     end do
  6 format('mass_vol_deposit_total_shd ', i2, 1x, 20(1pe13.6, 1x))
endif
```

- blockage fraction if the entire oxide scale will exfoliate or if only the magnetite will exfoliate. Each line is printed for each outage. For each outage the blockage is calculated for different lengths of the deposit.

```
 ! no_length_deposit_tube is the number of deposit lengths
 ! used to perform the blockage calculations,
 ! specified in length_deposit_over_id_tube data at input

   if (no_loops == 1)  then
      ch_repeat(1) = ' afb_tot afb_magL1'
      ch_repeat(2:no_length_deposit_tube) = ch_repeat(1)
      ch_repeat0(1) = ' afb0_tot afb0_magL1'
      ch_repeat0(2:no_length_deposit_tube) = ch_repeat0(1)
      write(enrg_lun, 4) (ch_repeat(j)(1:18)//'_d'//ch_index(j), &
        & j = 1, no_length_deposit_tube), &
        &  (ch_repeat0(j)(1:20)//'_d'//ch_index(j), j = 1, no_loops*no_length_deposit_tube)
    else if (no_loops == 2)  then
      ch_repeat(1:no_length_deposit_tube) = ' afb_tot afb_magL1'
      ch_repeat0(1:no_length_deposit_tube) = ' afb0_tot afb0_magL1'
      ch_repeat(1+no_length_deposit_tube:2*no_length_deposit_tube) = &
      & ' afb_tot afb_magL2'
      ch_repeat0(1+no_length_deposit_tube:2*no_length_deposit_tube) = &
      & ' afb0_tot afb0_magL2'
      write(enrg_lun, 4) (ch_repeat(j)(1:18)//'_d'//ch_index(j),  &
        & j = 1, no_length_deposit_tube), &
        & (ch_repeat(j)(1:18)//'_d'//ch_index(j-no_length_deposit_tube), &
        &  j = 1+no_length_deposit_tube, 2*no_length_deposit_tube), &
        & (ch_repeat0(j)(1:20)//'_d'//ch_index(j),  j = 1, no_length_deposit_tube), &
        & (ch_repeat0(j)(1:20)//'_d'//ch_index(j-no_length_deposit_tube), &
        &  j = 1+no_length_deposit_tube, 2*no_length_deposit_tube)
    endif
 endif
 4  format('#blockage_shd_lengthdep outage time_outage', 20(a))

   do ioutage = 1, no_f2l_event_out ! no_total_outage
    write(out_lun, 5) ioutage, time_event(ioutage), &
      & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &  ! all layers
      & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &   ! only magnetite layer
      & ileng = 1, no_loops*no_length_deposit_tube), &
      & (area_fraction_blocked_tube0(ioutage, 1, ileng, 1), &
      & area_fraction_blocked_tube0(ioutage, 3, ileng, 1), &
      & ileng = 1, no_loops*no_length_deposit_tube)
    write(enrg_lun, 5) ioutage, time_event(ioutage), &
      & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &
```

```
      & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &
      & ileng = 1, no_loops*no_length_deposit_tube), &
      & (area_fraction_blocked_tube0(ioutage, 1, ileng, 1), &
      & area_fraction_blocked_tube0(ioutage, 3, ileng, 1), &
      & ileng = 1, no_loops*no_length_deposit_tube)
5 format('blockage_shd_lengthdep ', i2, 1x, 50(1pe13.6, 1x))
```

- blockage fraction if the entire oxide scale will exfoliate or if only the magnetite will exfoliate. Each line is printed for all outages at a different length of the deposit.

```
   ch_repeat(1) = ' afb_tot afb_mag'
   ch_repeat(2:10) = ch_repeat(1)
   write(out_lun, 9) (ch_repeat(j)(1:16), j = 1, no_f2l_event_out)
   write(enrg_lun, 9) (ch_repeat(j)(1:16), j = 1, no_f2l_event_out)
9  format('#blockage_lengthdep_shd outage time_outage', 10(a))

 do ileng = 1, no_length_deposit_tube
   ! first loop
   write(out_lun, 10) ileng, length_deposit_tube(ileng) / &
     & (2.0*radius_deposit_tube_loop(1)), &
     & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &
     & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &
     & ioutage = 1, no_f2l_event_out)
   write(enrg_lun, 10) ileng, length_deposit_tube(ileng) / &
     & (2.0*radius_deposit_tube_loop(1)), &
     & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &
     & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &
     & ioutage = 1, no_f2l_event_out)
10 format('blockage_lengthdep_shd ', i2, 1x, 50(1pe13.6, 1x))
 end do
 do ileng = 1 + no_length_deposit_tube, no_loops*no_length_deposit_tube
   ! second loop
   write(out_lun, 10) ileng, length_deposit_tube(ileng) / &
     & (2.0*radius_deposit_tube_loop(2)), &
     & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &
     & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &
     & ioutage = 1, no_f2l_event_out)
   write(enrg_lun, 10) ileng, length_deposit_tube(ileng) / &
     & (2.0*radius_deposit_tube_loop(2)), &
     & (area_fraction_blocked_tube(ioutage, 1, ileng, 1), &
     & area_fraction_blocked_tube(ioutage, 3, ileng, 1), &
     & ioutage = 1, no_f2l_event_out)
 end do
```

The script **get_block2** is used as follows: `./get_block2 energy_filename1 runID alloyName` where the following characters/names/strings are given on the command line to specify the main result file used to extract the data, the specific problem ran and specific alloy. These strings can be changed according to your naming convention.

- `energy_filename1` points to the actual output file `energy_filename1.txt` , which was specified in the input file as `energy_output_file = 'energy_filename1.txt',`
- `runID` is a name that helps identify the individual input run (i.e., a string of characters) when several cases were ran for the same alloy,
- `alloyName` is alloy name to keep results organized by alloy name,

10.1 List of files provided for output processing

All the input files, header files, script files, are provided in the `spalloop1o_output_files_examples` directory.

All the final postprocessing files, which have "demorun1" in their name, were obtained with the line command:

```
 ./get_block2 energy_347h_bp1_a23 demorun1 steel1

ss347h_bp1_a23.inp
dil_energy_347h_bp1_a23.txt  gen_strain_347h_bp1_a23.txt  run_347h_bp1_a23.txt
dis_energy_347h_bp1_a23.txt  outage_347h_bp1_a23.txt   strain_347h_bp1_a23.txt
energy_347h_bp1_a23.txt      pl_outage_347h_bp1_a23.txt
# comment: "demorun1" are postprocessed files
block_len_shd_steel1_demorun1.txt  oxthick_steel1_demorun1.txt
temp_ho_steel1_demorun1_h2.txt  vm_str_h2_steel1_demorun1.txt
block_steel1_demorun1.txt    skin1_exfol_steel1_demorun1.txt
temp_ho_steel1_demorun1_h3.txt
dox_ho_steel1_demorun1.txt    tempfho_steel1_demorun1.txt
temp_ho_steel1_demorun1_h4.txt  vm_str_steel1_demorun1.txt
e_armitt_af_demorun1.txt    temp_ho_steel1_demorun1_h10.txt
mvt_steel1_demorun1.txt    temp_ho_steel1_demorun1_h1.txt
end{verbatim}

\newpage
\subsection {UNIX scripts for output processing}
 Listing of script {\bf get\_block2}.  Comments were added below to explain the variables writ
\begin{verbatim}
#comment1: just remove prior data; in case the problem was ran before
rm -f temp_ho_$3_$2_h1.txt temp_ho_$3_$2_h2.txt temp_ho_$3_$2_h3.txt temp_ho_$3_$2_h4.txt temp_
rm -f block_shd_len_$3_$2.txt block_len_shd_$3_$2.txt
rm -f e_armitt_af_$2.txt vm_str_$3_$2.txt vm_str_h2_$3_$2.txt vm_str_h6_$3_$2.txt vm_str_h12_$3
rm -f skin1_exfol_$3_$2.txt dox_ho_$3_$2.txt
#comment 2: get the maximum elastic energy during a shut-down event and the area exfoliated
grep -a energy_armitt_areaex $1.txt > e_armitt_af_$2.txt
# comment 3: get the temperature distribution along the tube height at outage 1, 2, 3, 4, and 1
# oxide thickness and heat flux is also provided
```

```
grep -a 'Tsteam_dox_height_event  1' $1.txt > temp_ho_$3_$2_h1.txt
grep -a 'Tsteam_dox_height_event  2' $1.txt > temp_ho_$3_$2_h2.txt
grep -a 'Tsteam_dox_height_event  3' $1.txt > temp_ho_$3_$2_h3.txt
grep -a 'Tsteam_dox_height_event  4' $1.txt > temp_ho_$3_$2_h4.txt
grep -a 'Tsteam_dox_height_event 10' $1.txt > temp_ho_$3_$2_h10.txt
# comment 4: fraction of skin exfoliated and fraction of oxide skin remained after each outage
grep -a exfol_fr_skinoutage $1.txt > skin1_exfol_$3_$2.txt
# comment 5: oxide thickness before and after each outage
grep -a dox_outage $1.txt > dox_ho_$3_$2.txt
# comment 6: average von Mises stresses and creep strain in the metal and oxide scale before an
grep -a vm_creep_outage $1.txt > vm_str_$3_$2.txt
# comment 7: min, max, average von Mises stresses and hoop creep strain in the metal and oxide
#         at fixed locations along the tube
grep "vm_height_cr_allt  2" $1.txt > vm_str_h2_$3_$2.txt
grep "vm_height_cr_allt  6" $1.txt > vm_str_h6_$3_$2.txt
grep "vm_height_cr_allt 12" $1.txt > vm_str_h12_$3_$2.txt
grep "vm_height_cr_allt 16" $1.txt > vm_str_h16_$3_$2.txt
# comment 8: total oxide thickness, spinel thickness, magnetite thickness, steam temperature at
grep Tsteam_dox_height_final $1.txt > tempfho_$3_$2.txt
# comment 9: oxide thickness in each layer at each height for all outages
grep -a oxide_vs_height_outage $1.txt > oxthick_$3_$2.txt
# comment 10: final exfoliated volume and mass per each tube loop (bend).
grep -a mass_vol_deposit_total_shd $1.txt > mvt_$3_$2.txt
# comment 11: blockage fraction if the entire oxide scale will exfoliate or if only the magneti
#          Each line is printed for each outage.
#          For each outage the blockage is calculated for different lengths of the deposit.
grep blockage_shd_lengthdep $1.txt > block_$3_$2.txt
# comment 12: blockage fraction if the entire oxide scale will exfoliate or if only the magneti
# Each line is printed for all outages at a different length of the deposit.
grep blockage_lengthdep_shd $1.txt > block_len_shd_$3_$2.txt
```