

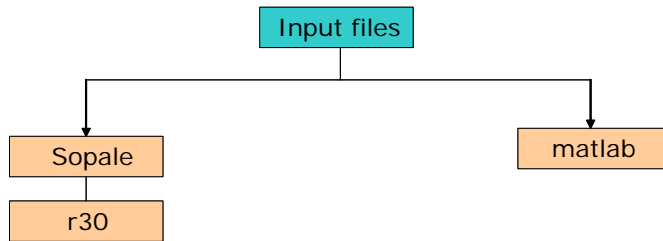
The common steps for submitting a SOPALE model are as follows:

- A) Designing and setting up a SOPALE model
- B) Submit a model
- C) Restarting a model

---

## INTRODUCTION

The directory structure of a SOPALE model is shown in Figure 1. Currently, members of the Dalhousie Geodynamics Group (DGG) run models on the P690 system. The input files are stored in a P690 directory, which usually has the same directory name as the model name. A fixed disk space was allocated to each member of the DGG group. DGG members need to keep track of the disk space to make sure that models will not die because of no available disk space. Output files will be written to the subdirectories *Sopale/r30* and *matlab*.



**Figure 1** Directory structure for a SOPALE model.

SOPALE produces various binary files as output. These SOPALE output will be stored in the subdirectory *Sopale/r30*. The outputs are large and include many fields such as coordinates, velocity, viscosity, strain rate, pressure, stress, mechanical materials, thermal materials. See [http://geodynamics.oceanography.dal.ca/bonny/docs/sopale\\_overview.txt](http://geodynamics.oceanography.dal.ca/bonny/docs/sopale_overview.txt) for the explanation of SOPALE output.

If you do not need to analyze all of these fields and want to save disk space, you can ask for MatLab output with only selected fields. The output files will be much smaller because not all fields are written on the outputs. The input file “matlab\_i” specifies the parameters which will be saved in the MatLab outputs and the numbers of saved outputs. MatLab outputs are stored in the subdirectory “matlab” (Figure 2).

## A DESIGNING AND SETTING UP A SOPALE MODEL

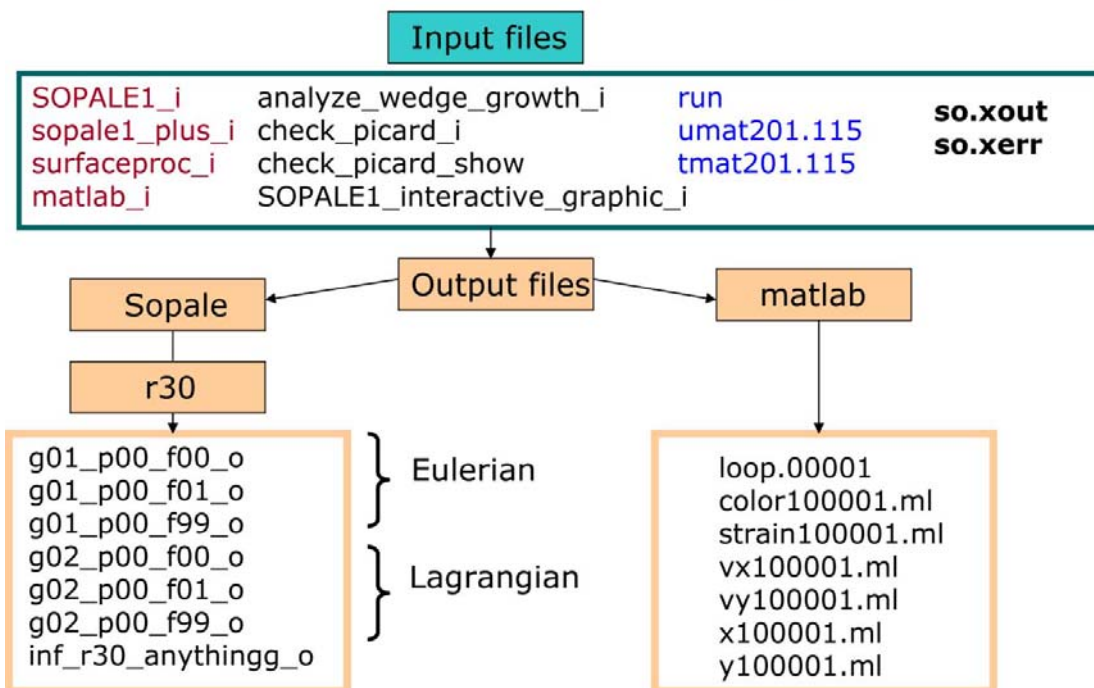
- 1) SOPALE code: The first step is to decide the version of the SOPALE code for the running the model. The path to the executable code can be stored in a file ‘runsopale’ so you have a record of which code was used.
- 2) Eulerian and Lagrangian Grids: Decide the model dimensions and the resolution of the Eulerian and Lagrangian grids.

- 3) Matrix files: The matrix files include mechanical, thermal files. For mechanical models, you need the mechanical matrix file (e.g. umat 801.79 is the mechanical matrix file of a model with 801 horizontal nodes and 79 vertical nodes). For thermal-mechanical models, you will need both thermal and mechanical matrix files (e.g. umat801.79 and tmat801.79).

Notes: The Fluid models will need the fluid and mechanical matrix files (e.g. fmat 801.79 and umat801.79). Bonny also has a new version of code which can generate the matrix files so you do not need to make the matrix files before running the model.

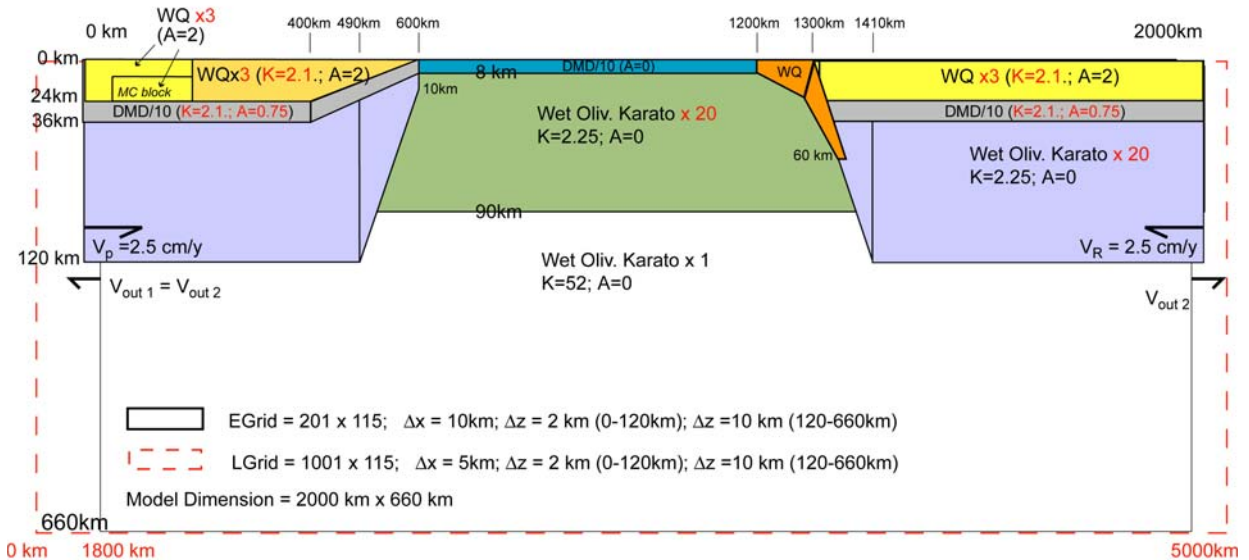
- 4) Edit input files: Most of the time, we prepared new input files by editing the existing set of input files from a previous model experiment. The boundary conditions, model geometry, thermal materials, mechanical materials and parameters of surface process (erosion, sedimentation) can be specified in the files SOPALE1\_i, sopale1\_plus\_i, surfaceproc\_i. A complete document of the User Guide for SOPALE can be found on the website <http://geodynamics.oceanography.dal.ca/bonny/docs/>. Figure 2 shows examples of input files.

## Model Inputs & Outputs



**Figure 2** Directory structure of the Sopale input and output (binary format). The SOPALE outputs are stored in the subdirectory “Sopale/r30/”. The restart files are g01\_p00\_f00\_o and g02\_p00\_f00\_o. The file g01\_p00\_f01\_o is the Eulerian output of frame 1 and g02\_p00\_f01\_o is the Lagrangian output of frame 1.

- 5) Diagrams should be drawn to illustrate the model set up of a new model series (Figure 3). The figure was based on the initial condition (time step 1) but with the vertical scale exaggerated.



**Figure 3** Diagram showing the model set up of a SOPALE Deep Model.

## B SUBMITTING A SOPALE MODEL

### 1) Submitting models on P690

After you prepare the inputs and put them in a directory on your P690 account, you can submit the model by enter the following command lines.

```
% cd 'model directory' (Change directory to the model directory)
% nohup /'pathname of the code'/SOPALE1.x (submitting model)
```

You can store the command lines in a script (e.g. 'run SOPALE') and then run it. This file will help you in the future to remember which version of the code you used to get a specific set of model outputs.

```
% ./run SOPALE
```

Here are the examples of the files *run SOPALE* and *SOPALE.cmd* for running models.

#### Example of the file *run SOPALE*

```
cp /home/mai/SOPALE/Model-inputs_Model-outputs/umat801.79 .
mkdir Sopale
mkdir Sopale/r30
mkdir matlab
nohup ./SOPALE.cmd 1>so.xout 2>so.xerr &
```

By running this script, the matrix file *umat801.79* will be copied to the current directory and the subdirectories for storing Sopale and MatLab outputs will be created before the

model runs. The error messages, if there are any, will be stored in the file `so.xerr`. The last line is the command for submitting model runs using the file `sopale.cmd`.

#### Example of the file `sopale.cmd`

```
echo `date`  
/home/beamnt1/sopale/code-feb07-06-adiabat/SOPALE1.x  
echo `date`
```

The time and date, when the model starts and stops, will be recorded in the file `so.xout`. Users will need to enter the correct path where the code is stored and to modify this file each time you use the new version of the SOPALE code.

## **2) Submitting models on SP**

Although most members of the Dalhousie GeoDynamics Group run models on P690, the `sp.dal.ca` system is also available for submitting jobs.

- Prepare the input files in users' home directory on `sp`.

- Commands to run or cancel a new model are as follows:

```
llsubmit *.cmd
```

```
llcancel *.cmd
```

- The LoadLeveler will assign the node where the model inputs will be copied to and the output files will also be stored in that node.

#### Example of a command file

```
#!/bin/ksh  
# template.job is used to generate temp.job automatically. Do  
# not modify.  
#@ output = /usertmp/mhnguyen/HTRV20-176-96-outputs/so.xout  
#@ error = /usertmp/mhnguyen/HTRV20-176-96-outputs/so.xerr  
#@ notification = Always  
#@ job_type = serial  
#@ class = long  
#@ min_processors = 1  
#@ max_processors = 1  
#@ queue  
# Make directory on /usertmp for the run  
JOBDIR=HTRV20-176-96-outputs  
if [[ -d /usertmp ]] then  
cd /usertmp  
if [[ ! -d mhnguyen ]] then  
mkdir mhnguyen  
fi  
cd mhnguyen/$JOBDIR  
# if [[ -d $JOBDIR ]] then  
# mkdir ${JOBDIR}_$$  
# cd ${JOBDIR}_$$  
# else  
# mkdir $JOBDIR  
# cd $JOBDIR  
# fi  
cp /home/beamnt1/ptt_jan2000/code/nonzero_vr/spmicx .  
cp /home/mhnguyen/microfem/HTRV20/${JOBDIR}/* .  
./spmicx  
fi
```

## C RESTARTING A MODEL

After you submit a model, it is important to check whether there are any error messages in the files `so.xerr` and `so.xout`.

Models might die early because there is no disk space left for writing the new output files. In this case, you will want to restart the model to continue to run the model after you clear up some disk space.

For some model runs, you might want to stop a model run on purpose and then restart it with the new modified input files.

Common steps for preparing a restart model:

- Make sure that the restart files (`/Sopale/r30/g01_p00_f00_o` and `/Sopale/r30/g02_p00_f00_o`) are at the correct timestep where you want to restart the model. You can check the file `/Sopale/r30/inf_r30_anythingggg_o` for the timestep of the restart files. For example, if the restart files are at 2000 timesteps and you want to restart your model at timestep 1500 (frame 3), you can copy the files `g01_p00_f03_o` to `g01_p00_f00_o` and `g02_p00_f03_o` to `g02_p00_f00_o`;
- Set the parameter `'rres' = 1` in the file `SOPALE1_i`;
- Rename the `so.xout` of the 'model before restart' to different name (e.g. `'so.xout_before-restart'`). The file `'so.xout'` will be overwritten by the new file `'so.xout'` when you run the restart model;
- Submit the restart model.

Here are the first few lines of a file `SOPALE1_i`.

```
1 ! number of comment lines to skip
This is a shortened SOPALE1_i input file (contains only one model)
1
30
B ! Test model run -- salt model template with flexure
30 6 ! amodel (model #), benchmark #
20 0 0 ! wres,rres,align_time
```

You will need to set the parameter `'rres' = 1` for running a restart model. The parameter `'align_time'` can be set to 0 or 1 depend on your decision. See the following notes from the SOPALE User Guide.

([http://geodynamics.oceanography.dal.ca/bonny/docs/sopale\\_inputs\\_current.html](http://geodynamics.oceanography.dal.ca/bonny/docs/sopale_inputs_current.html))

```
wres = number of timesteps between writing of 2 restart files
rres = determines whether to read restart file
      rres = 0 - don't read restart file
      rres = 1 - read restart file
align_time (only has effect when rres = 1)
= 0      on restart, set start time to zero and timestep to 1
= 1      This is appropriate when you're using a restart file
         which just establishes the state you want.
```

= 1 on restart, set start time to time in restart file  
and set timestep to timestep in restart file.  
This is good for restarting an interrupted run.

## D ANALYZING MODEL RESULTS

After you submit models and get the model output, you can analyze/interpret the model results by viewing the numerical values or post-processing the output. See the post-processing documentations for further information.

