Test case for the UP4 2017-18 Identification of a volcano reservoir from surface displacements

Rodolphe Le Riche, Victor Picheny, Valérie Cayol

March 28, 2018

Contents

1 Foreword

This directory contains data files with measured displacements at the surface of the Piton de la Fournaise volcano, along with files for modeling the displacements created by a specific magma reservoir. The source files are written

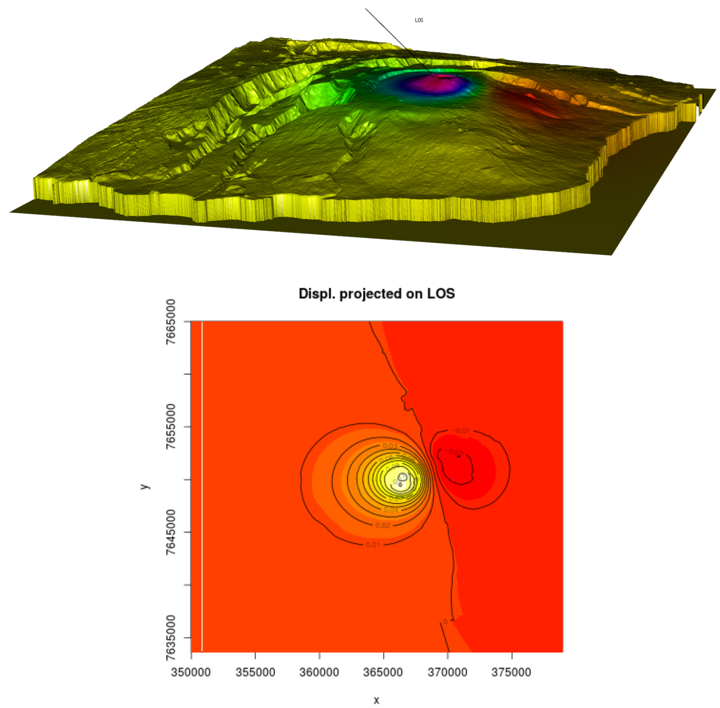

Figure 1: Digital terrain of the Piton de la Fournaise with displacements projected along the line of sight as colors (top) and (bottom) contour plot of the projected displacements (versus longitude x and latitude y). These projected displacements are the quantities to match in order to estimate the magma reservoir.

in the R programming language. By matching the modeled and measured displacements, one can identify the reservoir underneath the volcano. The test case was originally developped by Valérie Cayol, a volcanologist at CNRS, and re-implemented as an illustration for the course [1]. An illustration of the test case is provided in Figure 1.

2 General test case presentation

2.1 General formulation

The magma reservoir is the source of displacements at the surface of the volcano. The reservoir is assumed to have a spherical shape. It is described by $n = 5$ inputs:

- The position of the center of the sphere
  - the longitude and latiture in UTM meters, x^S and y^S, respectively,
  - the elevation of the source with respect to sea level in meters, z^S,
- the source radius in meters, a,
- and the source overpressure in MPa, p.

These 5 scalars are the unknowns, or variables, of our problem and will be gathered in the vector $\boldsymbol{\theta} \in [\boldsymbol{\theta}^{min}, \boldsymbol{\theta}^{max}] \subset \mathbb{R}^5$.

The volcano displacements are seen by InSAR radars mounted on a satellite as the projection of the displacements along a line of sight (LOS). Let $u_{LOS}(x, y; \boldsymbol{\theta})$ be the modeled displacement field parameterized by longitude and latitude x and y,

$$\begin{aligned} u_{LOS}(;\boldsymbol{\theta}) :\ & \mathbb{R}^2 \to \mathbb{R} \\ & (x, y) \mapsto u_{LOS}(x, y; \boldsymbol{\theta}) \end{aligned} \tag{1}$$

$u_{LOS}(;\boldsymbol{\theta})$ is expressed with the Mogi model [2, 3] which simulates the displacements created by a spherical magma chamber at the surface of a digital terrain. The displacements are measured at a finite set of m measurements points, (x^i, y^i), which yields $u^{\text{meas}}_{LOS}(x^i, y^i)$, $i = 1, \dots, m$. A weighted least squares distance between the measured and modeled displacements is expressed as

$$J^2_W(\boldsymbol{\theta}) \ = \ \frac{1}{2} \left(\mathbf{u}^{\text{meas}}_{\mathbf{LOS}} - \mathbf{u}_{\mathbf{LOS}}(\boldsymbol{\theta})\right)^\top \mathbf{W} \left(\mathbf{u}^{\text{meas}}_{\mathbf{LOS}} - \mathbf{u}_{\mathbf{LOS}}(\boldsymbol{\theta})\right) \ , \tag{2}$$

where the bold face version of u_{LOS} denotes its vectorized version at the measured points, e.g.,

$$\mathbf{u}_{\mathbf{LOS}}(\boldsymbol{\theta}) \ = \ \left[u_{LOS}(x^1, y^1; \boldsymbol{\theta}) \dots u_{LOS}(x^m, y^m; \boldsymbol{\theta})\right]^\top$$

and $\mathbf{W}$ is a *given* weight matrix, typically the inverse of a covariance matrix.

Normalizing a weighted least squares: the distribution of $J_W^2(\boldsymbol{\theta})$ when $\boldsymbol{\theta}$ is random within $[\boldsymbol{\theta}^{min}, \boldsymbol{\theta}^{max}]$ is far from being gaussian as it is bounded above 0 and has many very large values. The transformation $\log(1 + J_W^2(\boldsymbol{\theta}))$ makes it more gaussian and will be used in the following as it makes the function more compatible with a gaussian process model.

Model identification as an optimization problem: Finding the magma reservoir that corresponds to the measurements $\mathbf{u}_{\mathbf{LOS}}^{\text{meas}}$ amounts to minimizing the weighted least squares distance between the measurements and the Mogi model over the reservoir unknowns,

$$\boldsymbol{\theta}^\star \;=\; \arg \min_{\boldsymbol{\theta} \in [\boldsymbol{\theta}^{min}, \boldsymbol{\theta}^{max}]} J_W^2(\boldsymbol{\theta}) \; . \tag{3}$$

In the case of ill-posed problems, the solution $\boldsymbol{\theta}^\star$ may not be unique.

To sum up, this project aims at learning (a normalized version of) the n dimensional function $J_W^2(\boldsymbol{\theta})$ by carrying out a design of experiments (DoE), building a Gaussian process conditioned by the results of the DoE, and using it to minimize $J_W^2(\boldsymbol{\theta})$.

2.2 Installation, files provided and first steps

2.2.1 Software prerequisites

1. Have R available on your computer, cf. `https://www.r-project.org/`

2. Optionally (but really helpful) have rstudio installed, cf. `https://www.rstudio.com`

3. Install the "R.matlab" package to load the data that are in matlab format (`file_name.mat`), either from Tools / Install Package in rstudio or with the command `install.packages("R.matlab")`.

4. As above, install the following Dice suite of packages: "DiceKriging", "DiceOptim", "DiceView".

2.2.2 Files list

- data files

- `fullgrid_xyzulos.csv` : full digital terrain ((x, y, x)'s) and measured projected displacements in the ascii comma separated value format. Large file, only used for the plot demo (file `plots_3d_full_grid.R`).
- `data_nonoise.mat` : 220 measured (x, y, x)'s and associated u_{LOS}'s. Matlab format. Read and processed in the file `mainScript_DatascienceClass.R`.

- R source files:
  - `plots_3d_full_grid.R` : Loads the full digital model of the volcano (`fullgrid_xyzulos.csv`), make a 3D plot of it, makes 2D contour maps of the displacements (target and another vector of variables).
  - `mainScript_DatascienceClass.R` : utilities to load the data, and creates the function `compute_wls` which calculates a normalized J_W^2 (Eq. 2).
  - `wls_ulos.R` : calculates the weighted least squares distance J_W^2 (Eq. 2) between the model projected displacements and the target.
  - `mogi_3D.R` : calculate displacements on a digital terrain model from a point-wise spherical source.
  - `kernels.R` contains utilities to build covariance functions and matrices, used here only to generate the $\mathbf{W}$ weight matrix of Eq. 2.
- documentation: file `volcan_test_case.pdf`, compiled from `volcan_test_case.tex`, `biblio_paper.bib`, `contour_ulos.png` and `piton_ulos_2.png`.

During the project, you should only use the function `compute_wls` and should not change the rest of the provided code.

2.2.3 Running the demo step by step

1. Open with rstudio the file `plots_3d_full_grid.R` and "source" it, or start R in a console in the file directory and type `source("plots_3d_full_grid.R")`. A 3D window opens that shows the digital terrain of the volcano with a color projection of the displacements to be identified ($\mathbf{u}_{\mathbf{LOS}}^{\text{meas}}$). The line of sight is drawn at the top of the volcano. The 3D plot can be rotated. Contours plots are also given, where new set of variables $\boldsymbol{\theta}$ can be tried and compared through their $\mathbf{u}_{\mathbf{LOS}}(\boldsymbol{\theta})$ to the targeted variables.

2. Open in rstudio the file `mainScript_DatascienceClass.R`, or open the file with any text editor and start R in a console. The code is split in parts separated by a line of comments (`#### step #####...`). The first part loads libraries and utility files (e.g., the one coding the Mogi model). The second part provides the variables bounds $\boldsymbol{\theta}^{min}$ and $\boldsymbol{\theta}^{max}$ (encoded as `xmin` and `xmax`) along with other variables related data. The third part (`#### load data #####...`) loads the data, in particular the location of the measurements, (x^i, y^i) as `Glb_xi` and `Glb_yi`, and the vector of measured displacements as `Glb_ulos`. The fourth and last part provides functions for norming and unnorming the variables and norming the weighted least squares (J_W^2). It finishes with `compute_wls <- function(x)` whose input `x` is $\boldsymbol{\theta}$ and whose output is a normalized and centered $\log(1 + J_W^2(\boldsymbol{\theta}))$.

2.2.4 Overall goal of the project

From that point onwards, it will be your goal to complete the code, mainly using DiceKriging and DiceOptim to

1. build a statistical model of the function `compute_wls` (the normalized $\log(1 + J_W^2(\boldsymbol{\theta}))$),
2. validate it,
3. minimize it and
4. analyze the results

following the detailed instructions of each session.

3 Building a kriging model of the weighted least squares function (WLS)

Cf. text of "TP1 : plans remplissant l'espace et krigeage" by Victor Picheny.

4 Identification of the volcano source

The goal of the practical session is to estimate solutions to the identification problem of Eq. (3). Because in real world situations, a call to the volcano model (through the `wls_ulos` function) would be computationally expensive, we will use the EGO algorithm [4] in its DiceOptim version.

1. Write a script where the function `EGO.nsteps` is called with `compute_wls` as objective function.

2. Plot the results: `res$value` to look at the convergence of the objective function, `pairs(res$lastmodel@X)`, to look at the distribution of the points in the DoE (look at the best points and at the points added), `hist(res$value)` to look at the distribution of objective functions sampled by EGO.

3. Comment on the distribution of the best reservoir variables.

5

References

[1] Nicolas Durrande and Rodolphe Le Riche. Introduction to Gaussian Process Surrogate Models. Lecture at 4th MDIS Formater workshop, slides as HAL report no. cel-01618068, October 2017.

[2] N. Yamakawa. On the strain produced in a semi-infinite elastic solid by an interior source of stress. *J. Seismol. Soc. Japan*, 8:84–98, 1955.

[3] Kiyoo Mogi. Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them. 36, 01 1958.

[4] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.