PyFLOWGO: an open-source platform for simulation of channelized lava thermo-rheological properties

Magdalena Oryaëlle Chevrel[a], Jérémie Labroquère[b], Andrew Harris[a], Scott Rowland[c]

[a] *Université Clermont Auvergne, CNRS, IRD, OPGC, Laboratoire Magmas et Volcans, f-63000 Clermont-Ferrand, France*

[b] *Private work. Formally working at THALES SERVICES SAS, 290 Allée du Lac, 31670 Labège, France*

[c] *Department of Geology and Geophysics, University of Hawai'i at Mānoa, Honolulu, Hawaii, USA*

Abstract

Lava flow advance can be modeled through tracking the evolution of the thermo-rheological properties of a control volume of lava as it cools and crystallizes. An example of such a model was conceived by Harris and Rowland (2001) who developed a 1-D model, FLOWGO, in which the velocity of a control volume flowing down a channel depends on rheological properties computed following the thermal path estimated via a heat balance box model. We provide here an updated version of FLOWGO written in Python that is an open-source, modern and flexible language. Our software, named PyFLOWGO, allows selection of heat fluxes and rheological models of the user's choice to simulate the thermo-rheological evolution of the lava control volume. We describe its architecture that offers more flexibility while reducing the risk of making error when changing models in comparison to the previous FLOWGO version. Three cases are tested using actual data from channel-fed lava flow systems and results are discussed in terms of model validation and convergence. PyFLOWGO is open-source and packaged in a Python library to be imported and reused in any Python program.

Keywords: FLOWGO, Python, lava flow, heat budget, rheology

1. Introduction

The thermo-rheological properties of lava flowing in a channel depend on the evolution of the moving volume, where viscosity and yield strength are increasing due to cooling and crystallization (e.g. Lipman and Banks 1987, Crisp et al.

1994, Cashman et al. 1999). Harris and Rowland (2001) produced a 1-D model called FLOWGO in which velocity of a lava control volume flowing down a channel is computed via the Jeffreys (1925) equation as modified for a Bingham fluid by Moore (1987). In this approach velocity depends on the lava rheological properties computed according to the cooling and crystallization path of the control volume as estimated via a heat balance box model (Fig. 1). FLOWGO is thus a framework within which thermo-rheological models can be integrated to test fits between output parameters and natural data. By selecting appropriate models to place within this framework Harris and Rowland (2001) succeeded in simulating the down flow heat budget, cooling, crystallinity, viscosity, yield strength, velocity, channel width and maximum length of several lava flows including those of Mauna Loa 1984, Pu'u 'Ō 'ō 1997 and Etna 1998. Rowland et al. (2004) later adapted the models contained within FLOWGO to run in a Martian environment allowing cooling-limited, channelized lava flows on Mars to be simulated and their emplacement properties to be inferred (Ramsey et al. 2016).

During the 14 years since the inception of this approach, the basic physical principles on which FLOWGO is based have not changed. But recently, Harris and Rowland (2015) and Harris et al. (2015) incorporated an alternative model to compute the melt phase viscosity that is based on lava composition, rather than on a given assumed viscosity as originally proposed. They also introduced a three phase rheological model to estimate the effect of crystals and bubbles on viscosity. To correctly simulate the evolution of thermo-rheological parameters down flow using FLOWGO the user thus is allowed a degree of flexibility so as to best-fit the natural cases, while changing thermo-rheological models and variables within plausible limits (e.g. Harris et al. 2007). Originally, Harris and Rowland (2001) wrote FLOWGO in the programming language IDL (Interactive Data Language) but due to license price and other computing issues this code was set aside and an Excel version was written (officially published in Harris et al. 2015). This was freely shared when needed by other scientists. Although Excel is a convenient tool and is easily and widely used by geologists, it has limited applications, a poor flexibility for model evolution, and when many equations and input parameters are stacked in sequence, it becomes too easy to key in a hidden (or very-hard to find) error. Besides, it cannot be easily incorporated into other software. Lava modeling capabilities and computer processing power has improved over the past decade, and FLOWGO remains often cited, being recognized as the only thermo-rheological-based model. Some authors have therefore used it to reproduce natural flow evolution of past (e.g. Riker et al. 2009; Wantim et al. 2013) or ongoing eruptions (Harris et al. 2011; Wright et al. 2008), as well as producing hazards maps (Rowland et al. 2005) and applying FLOWGO as a reference to compare results of other models against (e.g. Cordonnier et al. 2015) or as input to develop new probabilistic models (e.g. QLAVAH, Mossoux et al. 2016).

The focus of the present work is thus to provide FLOWGO in a modern and flexible language. We chose Python because it provides useful libraries, is open-source, and its object-oriented approach allows for great flexibility. Python also has been widely adopted in scientific computing during the recent years and has been described as "the next wave in Earth Sciences Computing because it simply enables users to do more and better science" (Lin 2011). Furthermore, Python can be run on any operating system which guarantees portability. Here, we describe the architecture of our new open-source code, named PyFLOWGO, explaining the various models (heat flux, rheology, crystallization rate, crust temperature, crust cover fraction) that can currently be chosen to set-up a lava flow simulation. So that the model can be trusted as an operational tool with known uncertainty we tested the output against previous iterations of the model. As validation, we followed three cases for which appropriate natural data are available and have been previously tested: Mauna Loa 1984 (Hawaii), Mauna Ulu 1974 (Kilauea, Hawaii) and Piton de la Fournaise 2010 (La Réunion), and results are discussed in terms of model convergence and error.

2. Model architecture in Python

PyFLOWGO is developed in Python v3 which is an object-oriented programming language. The code has been designed to allow the user to switch between any existing models and add new models as they become available, without modifying the architecture of the code. The software acts as a framework that provides interfaces to implement multiple models, and calls them in the correct sequence to build the lava flow differential equations and solve them using a numerical approach. The interfaces basically define the methods necessary for the solver to work and can be implemented with specific models depending on the desired simulation. The top level of the architecture is the *integrator* which solves the differential equations depending on heat fluxes and on input physical characteristics of the lava (described by the *material lava class*), *terrain conditions* and a given crystallization rate model (Fig. 2). The *integrator* solves the differential equations and updates accordingly the current lava state (temperature, crystallization, position, etc.) which is then used for the next integration step. This process is iterated until termination conditions are reached. The *material lava class* is composed of multiple models such as the melt viscosity model, the relative viscosity model, the yield strength model and the vesicle fraction model. Each model is defined by the same interface that governs inputs and outputs delivered to and from the model. As an example, to compute the melt viscosity, all the models available to the user share a common interface called *base melt model viscosity* (Fig. 3). This interface makes sure that the model receives the state of the lava in order to deliver the viscosity value in the expected unit, that is Pa s. In the same way, all heat fluxes that compose the differential equation share the same interface called *base flux* (Fig. 4). In this case, the interface provides a unique method to compute and return the flux in W/m based on the state and channel dimensions as input parameters. With this architecture, new physical models or fluxes can then easily be added by implementing the given interface it depends from. Communication is carried out only between the interfaces, and models can be switched from one to another with no modification of the code structure, thus avoiding implementation errors and allowing a great flexibility.

3. Modeling

3.1. Differential equation for heat budget and crystallization down flow

PyFLOWGO is built around the main differential equation established in FLOWGO which is based on the heat budget for a control volume of lava within a channel (e.g. Danes 1972; Park and Iversen 1984; Crisp and Baloga 1994; Keszthelyi 1995b; Keszthelyi and Self 1998). The thermal budget (ΔH) represents the balance of the heat fluxes flowing in (gain) and out (loss) of a box model as illustrated in Figure 1. The change in heat content of a unit length per unit time (ΔH in J $s^{-1}m^{-1}$) is therefore described by:

$$\Delta H = Q_{rad} + Q_{conv} + Q_{rain} + Q_{cond} - Q_{cryst} - Q_{visc} \tag{1}$$

where Q_{rad}, Q_{conv} and Q_{rain} represent the heat loss from the surface due to radiation, forced atmospheric convection and rain vaporisation; Q_{cond} is the heat loss by conduction through the flow base and levées; and Q_{cryst} and Q_{visc} are the heat gains due to crystallization and viscous dissipation. Note that heat loss due to entrainment of cold material from the crust into the hotter flow interior could also be involved (Crisp and Baloga 1994). This case is not treated here but all details may be found in Harris and Rowland (2001). Following Keszthelyi (1995b), heat gain from crystallization, Q_{cryst} can be written as:

$$Q_{cryst} = \frac{\partial T}{\partial x} \rho_{bulk} L_{cryst} E_r \frac{\partial \phi}{\partial T_{cool}} \tag{2}$$

where $\partial T/\partial x$ is the cooling per unit length (in K/m), ρ_{bulk} (kg/m^3) is lava bulk density, L_{cryst} (J/kg) is latent heat of crystallization, E_r (m^3/s) is effusion rate, and $\partial \phi/\partial T_{cool}$ (K^{-1}) is the increase of crystal volume fraction per degree of cooling (crystallization down flow, $-\partial \phi/\partial T$). Combining and re-arranging Eq. 1 and Eq. 2 the cooling per unit length (as function of distance down flow) can be isolated and the following differential equation established (Harris and Rowland 2001; Harris and Rowland 2015; Harris et al. 2015):

$$\frac{\partial T}{\partial x} = \frac{-Q_{rad} - Q_{conv} - Q_{cond} - Q_{rain} + Q_{visc}}{E_r \rho_{bulk} L_{cryst} \partial \phi / \partial T_{cool}} \tag{3}$$

where the fluxes are expressed in W/m and detailed in the supplementary material (Appendix A) together with the thermal conditions of the lava (crust temperature and coverage). Each heat flux (Q_i) can be written as a function of the state (Y) and the position (x), such that: $Q_i(Y,x)$. This allows rewriting Eq.3 in the following form:

$$\frac{\partial T}{\partial x} = F\left(Y, x, \frac{\partial \phi}{\partial T_{cool}}\right)$$

$$\text{with:} \qquad F\left(Y, x, \frac{\partial \phi}{\partial T_{cool}}\right) = \frac{1}{E_r \rho_{bulk} L_{cryst}} \frac{1}{\partial \phi / \partial T_{cool}} \sum_{fluxes} Q_i(Y,x) \tag{4}$$

The increase of down flow crystal fraction ($\partial \phi / \partial x$) is then computed via:

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial T_{cool}} \frac{\partial T}{\partial x} \tag{5}$$

where the crystallization rate per degree of cooling, $\partial \phi / \partial T_{cool}$, can be calculated from one of the models presented in the supplementary material (Appendix A).

3.2. Conservation of volume

Based on mass conservation, if the effusion rate and channel depth are kept fixed down flow, then width can be computed at each down flow step from:

$$w = \frac{E_r}{d V_{mean}} \tag{6}$$

with w and d being the channel width and depth (in m), and V_{mean} being the flow velocity (in m/s) which in turn is controlled by the underlying slope and the rheological properties of the lava. The effusion rate is computed at the vent from initial flow geometry (width x depth), rheology and velocity.

3.3. Flow velocity

The velocity model originally used by FLOWGO is derived from the Jeffreys (1925) equation that was later adapted for a Bingham rheology by Moore (1987). Although this equation is based on the mean of the velocity gradient inside the channel, here it is used as to estimate a single value to best characterize the velocity of the control volume at each step. Note that this is not the rate of advance of the flow front and is expressed as:

$$V_{mean} = \left[\frac{\rho_{bulk}\, g\, d^2 \sin\theta}{n\, \eta_{bulk}}\right] \left[1 - \frac{3}{2}\frac{\tau_0}{\tau_b} + \frac{1}{2}\left(\frac{\tau_0}{\tau_b}\right)^3\right] \tag{7}$$

where n is the channel shape factor, θ is the underlying slope in radians, g (m/s^2) is acceleration due to gravity, ρ_{bulk} (kg/m^3) is the lava bulk density, η_{bulk} (Pa·s) is the bulk viscosity of the lava mixture (considering the melt phase and the effect of particles, see supplementary for details), and τ_0 (Pa) and τ_b (Pa) are respectively the lava yield strength and the basal shear stress. Channel shape factor can be obtained via $3(1+d/w)^2$ (Wilson and Parfitt 1993) and reduces to 3 when the channel is much wider than deeper. Solution of the velocity model now requires definition of ρ_{bulk}, η_{bulk}, τ_0 and τ_b where η_{bulk}, τ_0 are functions of $\partial\phi/\partial T_{cool}$ and $\partial T/\partial x$ and hence dependent on the thermal box model. Details about the models to calculate these variables are given in Appendix A.

4. Equation solving and numerical methods

4.1. Numerical integration

PyFLOWGO uses the FLOWGO solution (Harris and Rowland 2001; Rowland et al. 2005; Harris and Rowland 2015) to solve the lava flow temperature and crystal fraction equations, i.e., Eq. 4 and Eq. 5, respectively. Theses two equations are discretized in the space domain using a linearization at the position x_i. By denoting $\delta_1^2\Lambda = \Lambda_2 - \Lambda_1$ as the variation in variable Λ from state 1 to state 2, and $\Lambda_i = \Lambda(x_i)$ as the variable value at position x_i, we obtain the following first order integration (neglecting higher order components):

$$\frac{\delta_i^{i+1}T}{\delta_i^{i+1}x} \approx F\left[Y_i, x_i, \left(\frac{\partial\phi}{\partial T_{cool}}\right)_i\right] \tag{8}$$

which leads to the common Euler scheme for temperature:

$$T_{i+1} = T_i + (\delta_i^{i+1}x)F\left[Y_i, x_i, \left(\frac{\partial\phi}{\partial T_{cool}}\right)_i\right] \tag{9}$$

with $\delta_i^{i+1}x$ being the step distance between i and $i+1$. Then the second equation, for the crystal fraction, is solved using the same approach:

$$\phi_{i+1} = \phi_i + (\delta_i^{i+1}x)\left(\frac{\partial\phi}{\partial T_{cool}}\right)_i F\left[Y_i, x_i, \left(\frac{\partial\phi}{\partial T_{cool}}\right)_i\right] \tag{10}$$

The values of the state (Y_{i+1}) now allow computation of the rheology, and hence the velocity of the control volume, at position x_{i+1}, which is then used to estimate channel width by considering a constant effusion rate E_r and a constant depth in the conservation of volume equation (Eq. 6).

The thermal conditions and the crystal fraction are thus integrated by propagating the initial state conditions at x_0 down-channel, and by updating velocity and control volume thermo-rheological properties at each step. In practice the step size is fixed such that $\forall i, \delta_i^{i+1} x = \Delta x$. Determination of this step is model and case dependent, and must be chosen to be small enough to remove any numerical error and provide enough accuracy in the variables of interest. A convergence study must therefore be performed for every new studied case.

4.2. Data interpolation

At multiple places in PyFLOWGO, interpolation is needed when data are provided as discrete values. In particular, the line-of-steepest-descent down which the control volume is moved is extracted from a Digital Elevation Model (DEM) where data are usually provided for every 1, 10, 50 or even every 100 meters (depending on the spatial resolution of the DEM). The same holds for the MELTS data used to estimate crystallization as a function of temperature (see crystallization rate model in supplementary material). In this case, data are usually given with a temperature step of 0.1, 1 or 10 °C. To be independent of data discretization, and thus able to read any type of data, the data has to be interpolated to be set at a common step value. In PyFLOWGO, we use a linear interpolation to reconstruct any missing data during the integration of the differential equations.

5. Description of the software package

PyFLOWGO is packaged as a module (or library) to be imported and reused in any Python program. The models and simulation configurations are chosen using a single configuration file (json format) that contains all the necessary numerical and model parameters (Table A.1 in appendix). The user can therefore select which heat fluxes to consider and the associated models for lava thermal condition down flow including effective crust cover fraction, crust and uncrusted surface temperature (see Table A.2 in appendix). The package also include models for crystallization per degree of cooling, density, melt viscosity, relative viscosity, vesicle fraction and for yield strength and basal shear stress (see Appendix A for details). As described above (section 2), any new model can be added by implementing the interface it depends from.

The line-of-steepest-descent down which the control volume is moved has to be previously extracted from a DEM and input as a x, y text file where x is distance down flow and y is the slope (in degrees) at that point. All computed variables (as a function of distance) can be stored for every step in an output file (CSV format). Some tools are provided to plot graphs such as crystallization rate, velocity, evolution of the bulk viscosity (interstitial melt + effect of particles), crystal fraction, yield strength, channel width as a function of the distance. Ground-truth data can also be added and plotted with the model results.

6. Model Verification

Three lava channels, Mauna Loa 1984 (ML84), Mauna Ulu 1974 (MU74) and Piton de la Fournaise 2010 (PdF2010) have been chosen as test cases from previously published FLOWGO papers. These flows were chosen because their input parameters are well constrained and because they cover a large range of lava flow characteristics: ML84 being a long channel of more than 25 km, MU74 being a mature channelized flow of about 8 km in length and PdF2010 being slower and cooler than the Hawaiian cases and of only 1-2 km in length. For these cases, PyFLOWGO was run using the same input parameters as provided in Harris and Rowland (2015), Robert et al. (2014) and Harris et al. (2015), respectively for the three test cases (see table A.3 in appendix).

6.1. Convergence analysis

Numerical convergence is a necessary step to verify of the numerical solution. Indeed, the integration scheme propagates a numerical error which can then exceed the model error. The numerical error of an Euler integration scheme is controlled by the integration step size. Convergence analysis consists of reducing the step size and then tracking the value returned for the variable of interest. When the variable seems to stabilize within an acceptable range, then the step size is sufficiently small.

A convergence analysis is performed here for each case by reducing the iteration step size from 100 m down to 1 m. Figure 5 shows the maximum distance attained by the control volume (i.e. the point where mean velocity equals 0 m/s) and the lava core temperature near the end of the flow with respect to the step size. For the three cases the convergence for distance is reached at step size less than 10 m with an error of less than 50 m. For ML84 and MU74, the temperature convergence is reached at step sizes of less than 10 m, with an error within $\sim$ 2-3 °C, but for PdF2010, temperature convergence seems to be reached only at smaller step sizes (Fig. 5). For these three test cases, we conclude that a 10 m step is a maximum step size that must be used to run PyFLOWGO in order to guarantee small numerical errors on simulated lava flow properties and dimensions. Initial convergence tests by Harris and Rowland (unpublished) and by us here indicate 10 m as being an optimum step size in terms of errors and run time. As mention in the previous section a convergence study must be performed for every new studied case.

6.2. Results and validation against FLOWGO

Figure 6 plots lava core temperature, bulk viscosity and mean velocity computed with PyFLOWGO against the results from FLOWGO Excel spreadsheets prepared for the same test cases by Harris and Rowland (2015), Robert et al. (2014) and Harris et al. (2015), using identical input parameters (see table A.3 in appendix). For all the cases, PyFLOWGO reproduces the FLOWGO results perfectly. Note that the oscillations in mean velocity for ML84 and Pdf2010 come from the small spatial resolution of the line of steepest descent. To obtain less noisy results, one could filter the DEM data. For Mauna Ulu 1974, we used the same slope path as the one provided by Robert et al. (2014), specifically one value every 200 m. The PyFLOWGO run at this 200 m step size thus reproduces very well the results of Robert et al. (2014), but a comparison with PyFLOWGO run at a step size of 10 m shows that convergence was not actually reached using the 200 m step size. Here one can see that using an appropriate step size is necessary to avoid large errors (in this case the distance reached differs by 1 km on a total distance of 6.5 km).

7. Conclusion

This paper describes PyFLOWGO, a software written in Python to run FLOWGO, a thermo-rheological framework for lava flowing in a channel as originally presented by Harris and Rowland (2001). PyFLOWGO is constructed in a similar manner as FLOWGO to allow estimation of all parameters involved in the thermo-rheological evolution of a control lava volume flowing down a channel. We present here the architecture of the code, as well as the discretized formulation of the channelized lava flow problem and the various models that can be selected according to the study case. This new code is written with the object-oriented programming language Python v3, and offers more flexibility while reducing the risk of making error when changing models in comparison to the previous FLOWGO version which was written in Excel. The user can run PyFLOWGO using already implemented models, or extend the code with new models by simply implementing the base classes. Communication through interfaces allow tests of different models on the same study case, without modifying the code architecture. This software, used as a model testing platform, also allows the user to easily and quickly set up new complex cases of lava flow simulation to test.

PyFLOWGO has been successfully validated against FLOWGO via three test cases (Mauna Loa 1984, Mauna Ulu 1974 and Piton de la Fournaise 2010). For each test case, a convergence study has been performed, which is an essential practice that must be conducted for each new study. PyFLOWGO has also been unit tested and packaged in a Python library form to allow ease of installation. This software is open-source, thus available at any location and institution.

Acknowledgements

The authors greatly acknowledge Simone Tarquini and an anonymous reviewer for revisions that improve the clarity of the manuscript. M.O.C acknowledges the Auvergne fellowship program for full support. This is ClerVolc publication number 270.

References

Bailey, J., Harris, A., Dehn, J., Calvari, S., Rowland, S., 2006. The changing morphology of an open lava channel on mt. etna. Bulletin of Volcanology 68, 497–515.

Calvari, S., Cotteli, M., Neri, M., Pompilio, M., Scribano, V., 1994. The 1991-1993 etna eruption: chornology and lava flow-field evolution. Acta Vulcanol. 4, 1–14.

Cashman, K.V., Thornber, C., Kauahikaua, J.P., 1999. Cooling and crystallization of lava in open channels, and the transition of pahoehoe lava to aa. Bulletin of Volcanology 61, 306–323.

Castruccio, A., Rust, A.C., Sparks, R., 2010. Rheology and flow of crystal-bearing lavas: Insights from analogue gravity currents. Earth and Planetary Science Letters 297, 471–480.

Chevrel, M., Platz, T., Hauber, E., Baratoux, D., Lavallee, Y., Dingwell, D., 2013. Lava flow rheology: A comparison of morphological and petrological methods. Earth and Planetary Science Letters 384, 109–120.

Cimarelli, C., Costa, A., Mueller, S., Mader, H.M., 2011. Rheology of magmas with bimodal crystal size and shape distributions: Insights from analog experiments. Geochem. Geophys. Geosyst. 12, Q07024.

Cordonnier, B., Lev, E., Garel, F., 2015. Benchmarking lava-flow models. Detecting, Modeling and Responding to Effusive Eruptions. Eds, Harris AJL, De

Groeve T, Garel F and Carn SA, Geological Society, London, Special Publications 426.

Costa, A., Caricchi, L., Bagdassarov, N., 2009. A model for the rheology of particle-bearing suspensions and partially molten rocks. Geochemistry Geophysics Geosystems 10, Q03010.

Costa, A., Macedonio, G., 2003. Viscous heating in fluids with temperature-dependent viscosity: implications for magma flows. Nonlinear Processes in Geophysics 10, 545–555.

Crisp, J., Baloga, S., 1990. A model for lava flows with two thermal components. Journal of Geophysical Research 95(B2), 1255–1270.

Crisp, J., Baloga, S., 1994. Influence of crystallization and entrainment of cooler material on the emplacement of basaltic 'a'a lava flows. J. Geophys. Res. 99, 11,819 – 11,831.

Crisp, J., Cashman, K.v., Bonini, J.A., Hougen, S., Pieri, D., 1994. Crystallization history of the 1984 Mauna Loa lava flow. Journal of Geophysical Research 99, 7177–7198.

Danes, Z., 1972. Dynamics of lava flows. Geophys. Res. Lett. 77, 1430–1432.

Dingwell, D.B., 1996. Volcanic dilemma flow or blow. Science 273, 1054–1055.

Dragoni, M., 1989. A dynamical model of lava flows cooling by radiation. Bulletin of Volcanology 51, 88–95.

Flynn, L., Mouginis-Mark, P., 1994. Temperature of an active lava channel from spectral measurments, kilauea volcano, hawaii. Bulletin of Volcanology 56, 297–301.

Fulcher, G., 1925. Analysis of recent measurements of the viscosity of glasses. J. Am. Ceram. Soc. 8, 339–355.

Ghiorso, M.S., Sack, O., 1995. Chemical mass transfer in magmatic processes IV.A revised and internally consistent thermodynamic model for the interpolation and extrapolation of liquid-solid equilibria in magmatic systems at elevated temperatures and pressures. Contrib. Mineral. Petrol. 119, 197–212.

Giordano, D., Russell, J.K., Dingwell, D., 2008. Viscosity of magmatic liquids: A model. Earth and Planetary Science Letters 271, 123–134.

Greeley, R., Iversen, J.D., 1987. Measurements of wind friction speeds over lava surfaces and assessment of sediment transport. Geophysical Research Letters 14, 925–928.

Harris, A., Dehn, J., Patrick, M., Calvari, S., Ripepe, M., Lodato, L., 2005. Lava effusion rates from hand-held thermal infrared imagery: an example from the june 2003 effusive activity at stromboli. Bulletin of Volcanology 68, 107–117.

Harris, A., Favalli, M., Mazzarini, F., Pareschi, M., 2007. Best-fit results from application of a thermo-rheological model for channelized lava flow to high spatial resolution morphological data. Geophysical research letters 34, L01301.

Harris, A., Favalli, M., Wright, R., Garbeil, H., 2011. Hazard assessment at Mount

Etna using a hybrid lava flow inundation model and satellite-based land classification. Natural Hazards 58, 1001–1027.

Harris, A., Flynn, L., Keszthelyi, L., Mouginis-Mark, P., Rowland, S., Resing, J., 1998. Calculation of lava effusion rates from landsat tm data. Bulletin of Volcanology 60, 52–71.

Harris, A.J.L., Rhéty, M., Gurioli, L., Villeneuve, N., Paris, R., 2015. Simulating the thermorheological evolution of channel-contained lava: FLOWGO and its implementation in EXCEL. Detecting, Modelling and Responding to Effusive Eruptions. Eds, Harris AJL, De Groeve T, Garel F and Carn SA, Geological Society, London, Special Publications 426.

Harris, A.J.L., Rowland, S.K., 2001. FLOWGO: a kinematic thermo-rheological model for lava flowing in a channel. Bulletin of Volcanology 63, 20–44.

Harris, A.J.L., Rowland, S.K., 2015. FLOWGO 2012: An updated framework for thermorheological simulations of channel-contained lava. Hawaiian Volcanoes: From Source to Surface, Geophysical Monograph 208, Eds, Carey R, Cayol V, Poland M, and Weis D, American Geophysical Union .

Hon, K., Kauahikaua, J., Denlinger, R., Mackay, K., 1994. Emplacement and inflation of pahoehoe sheet flows: Observations and measurements of active lava flows on Kilauea Volcano, Hawaii. Geol. Soc. Am. Bull. 106, 351–370.

Hulme, G., 1974. The interpretation of lava flow morphology. Geophys. J. R. Astron. Soc. 39, 361–383.

Jeffreys, H., 1925. The flow of water in an inclined channel of rectangular section. Phil. Mag. serie 6, 49, 293, 793–807.

Keszthelyi, L., 1995a. Measurements of the cooling at the base of pahoehoe flows. Geophysical Research Letter 22, 2195–2198.

Keszthelyi, L., 1995b. A preliminary thermal budget for lava tubes on the earth and planets. Journal of Geophysical Research 100, 20,411–20,420.

Keszthelyi, L., Denlinger, R., 1996. The initial cooling of pahoehoe flow lobes. Bulletin of Volcanology 58, 5–28.

Keszthelyi, L., Harris, A.J.L., Dehn, J., 2003. Observations of the effect of wind on the cooling of active lava flows. Geophysical Research Letters 30, 1944–8007.

Keszthelyi, L., Self, S., 1998. Some physical requirements for the emplacement of long basaltic lava flows. Journal of Geophysical Research B11, 27,447–27,464.

Kolzenburg, S., Giordano, D., Cimarelli, S., Dingwell, D.B., 2016. In situ thermal characterization of cooling/crystallizing lavas during rheology measurements and implications for lava flow emplacement. Geochimica et Cosmochimica Acta 195, 244–258.

Krieger, I., 1972. Rheology of monodispersed latices. Adv. Colloid Interface Sci. 3, 111–136.

Krieger, I.M., Dougherty, T., 1959. A mechanism for non-Newtonian flow in suspensions of rigid spheres. Journal of Rheology 3, 137.

Lin, J., 2011. Why python is the next wave in earth sciences computing. Bull. Am. Meteorol. Soc. 93, 1823–1824.

Lipman, P.W., Banks, N.G., 1987. Aa flow dynamics, mauna loa 1984. U.S. Geol. Surv. Prof. Pap 1350 , 1527–1567.

Llewellin, E.W., Manga, M., 2005. Bubble suspension rheology and implications for conduit flow. J. Volcanol. Geotherm. Res. 143, 205– 217.

Mader, H., Llewellin, E., Mueller, S., 2013. The rheology of two-phase magmas: A review and analysis. Bulletin of Volcanology 257, 135–158.

Maron, S.H., Pierce, P.E., 1956. Application of Ree-Eyring generalized flow theory to suspensions of spherical particles. J. Colloid Sci. 11, 80–95.

Moitra, P., Gonnermann, H.M., 2015. Effects of crystal shape- and size-modality on magma rheology. Geochemistry, Geophysics, Geosystems 16, 1–26.

Moore, H.J., 1987. Preliminary estimates of the rheological properties of 1984 Mauna Loa lava. U.S. Geol. Surv. Prof. Pap 1350 99, 1569–1588.

Mossoux, S., Saey, M., Bartolini, S., S., P., Canters, F., Kervyn, M., 2016. Q-LAVHA: A flexible GIS plugin to simulate lava flows. Computers and Geosciences 97, 98–109.

Mueller, S., Llewellin, E.W., Mader, H.M., 2010. The rheology of suspensions of solid particles. Philos. Trans. R. Soc. Lond. A 466, 1201–1228.

Pabst, W., 2004. Fundamental considerations on suspension rheology. Ceram-Silikaty 48, 6–13.

Pal, R., 2003. Rheological behavior of bubble-bearing magmas. Earth Planet. Sci. Lett. 207, 165 – 179.

Park, S., Iversen, J.D., 1984. Dynamics of lava flow: Thickness growth characteristics of steady two-dimensional flow. Geophys. Res. Lett. 7, 641–644.

Phan-Thien, N., Pham, D.C., 1997. Differential multiphase models for polydispersed suspensions and particulate solids. J. Non-Newtonian Fluid Mech. 72, 305–318.

Pieri, D.C., Baloga, S.M., 1986. Eruption rate area, and length relationships for some hawaiian lava flows. Journal of Volcanology and Geothermal Research 30, 29 – 45.

Pieri, D.C., Glaze, L.S., Abrams, M.J., 1990. Thermal radiance observations of an active lava flow during the june 1984 eruption of mount etna. Journal of Volcanology and Geothermal Research 18, 1018 – 1022.

Pinkerton, H., Stevenson, R.J., 1992. Methods of determining the rheological properties of magmas at sub-liquidus temperatures. J. Volcanol. Geotherm. Res. 53, 47–66.

Ramsey, M.S., Harris, A., Crown, D., 2016. What can thermal infrared remote sensing of terrestrial volcanoes tell us about processes past and present on Mars? Journal of Volcanology and Geothermal Research 311, 198–216.

Riker, J., Cashman, K., Kauahikaua, J., Montierth, C., 2009. The length of channelised lava flows: insight from the 1859 eruption of Mauna Loa Volcano, Hawaii. Journal of Volcanology and Geothermal Research 183, 139–156.

Robert, B., Harris, A., Gurioli, G., Medard, E., Sehlke, A., Whittington, A., 2014.

Textural and rheological evolution of basalt flowing down a lava channel. Bulletin of Volcanology 76, 824.

Rowland, S.K., Garbeil, H., Harris, A., 2005. Lengths and hazards from channel-fed lava flows on Mauna Loa, Hawai-i, determined from thermal and downslope modeling with flowgo. Bulletin of Volcanology 67, 634–647.

Rowland, S.K., Harris, A., Garbeil, H., 2004. Effects of martian conditions on numerically modeled, cooling-limited, channelized lava flows. Journal of Geophysical Research 109, E100101.

Ryerson, F.J., Weed, H.C., Piwinskii, A.J., 1988. Rheology of subliquidus magmas: I Picritic compositions. J. Geophys. Res. 93, 3421– 3436.

Shaw, H., 1965. Comments on viscosity, crystal settling, and convection in granitic magmas. Am. J. Sci. 263, 120–152.

Shaw, H.R., 1972. Viscosities of magmatic silicate liquids: An empirical method of prediction. Am. J. Sci. 272, 870–893.

Tammann, G., Hesse, W., 1926. Die Abhängigkeit der Viskosität von der Temperatur bei unterkühlten Flüssigkeiten. Z. Anorg. Allg. Chem. 156.

Vogel, D., 1921. Temperaturabhängigkeitsgesetz der Viskosität von Flüssigkeiten. Phys. Zeit 22, 645–646.

Wantim, M., Kervyn, M., Ernst, G., del Marmol, M., Suh, C., Jacobs, P., 2013. Numerical experiments on the dynamics of channelised lava flows at mount cameroon volcano with the flowgo thermo-rheological model. Journal of Volcanology and Geothermal Research 253, 35–53.

Wilson, L., Parfitt, E., 1993. The formation of perched lava ponds on basaltic volcanoes: the influence of flow geometry on cooling-limited lava flow lengths. Journal of Volcanology and Geothermal Research 56, 113–123.

Wright, R., Garbeil, H., Harris, A.J., 2008. Using infrared satellite data to drive a thermo-rheological/stochastic lava flow emplacement model: A method for near-real-time volcanic hazard assessment. Geophysical Research Letters 35, 1–5.

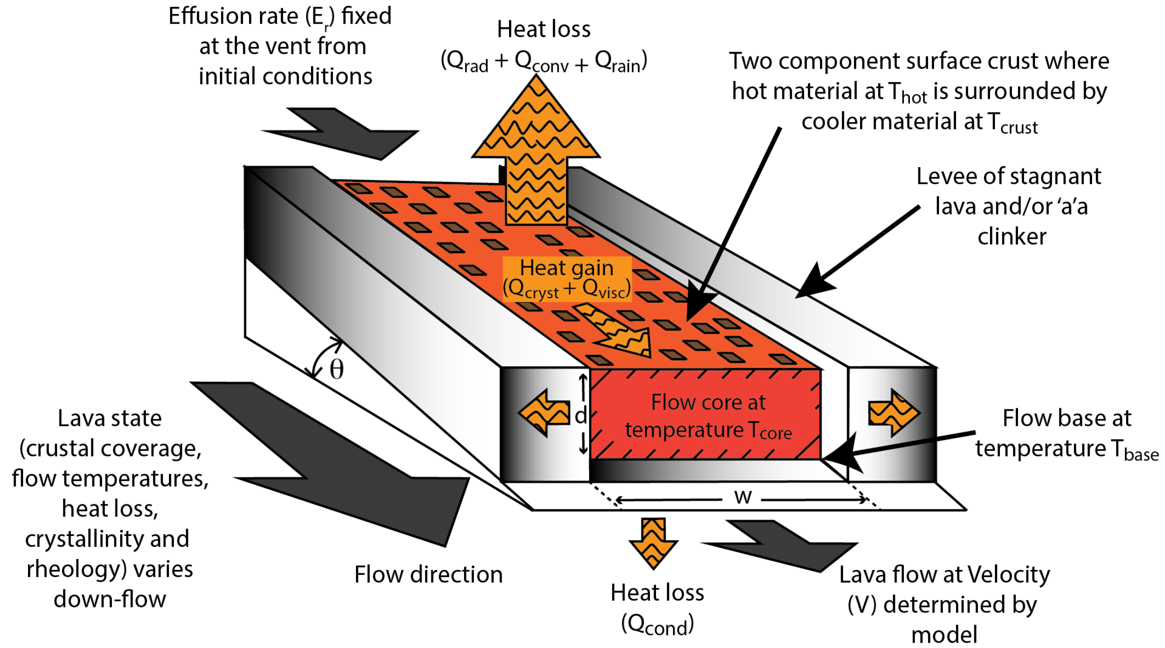
Effusion rate (E_r) fixed at the vent from initial conditions
Heat loss (Q_{rad} + Q_{conv} + Q_{rain})
Two component surface crust where hot material at T_{hot} is surrounded by cooler material at T_{crust}
Levee of stagnant lava and/or 'a'a clinker
Heat gain (Q_{cryst} + Q_{visc})
θ
d
Flow core at temperature T_{core}
Flow base at temperature T_{base}
w
Lava state (crustal coverage, flow temperatures, heat loss, crystallinity and rheology) varies down-flow
Flow direction
Heat loss (Q_{cond})
Lava flow at Velocity (V) determined by model

Figure 1: Schematic view of the thermo-rheological model FLOWGO illustrating the heat box model of the control volume of lava advancing through a channel (modified from Harris and Rowland, 2001). The lava viscosity and yield strength are estimated within the control volume according to the lava state within the box (including thermal state: heat budget, temperature of core, base, surface, crust; and physical state: crustal coverage, crystallinity, vesicularity) in order to compute the velocity and corresponding channel width (for a fixed effusion rate) used for the next step. This model assumes a ”cooling limited” lava flow behavior: the lava stops flowing because it has cooled to such an extent that its rheological behavior impede motion. Q_{rad}, Q_{conv} and Q_{rain} are heat losses into the atmosphere due to radiation, forced convection due to heating of the air above the lava surface and effect of rain, respectively. Q_{cond} is the heat loss by conduction into the cooler base and levees. Q_{cryst} and Q_{visc} are the heat gain due to crystallization and viscous dissipation, respectively. Dimensions d and w are the channel depth and width, and θ is the underlying ground slope.

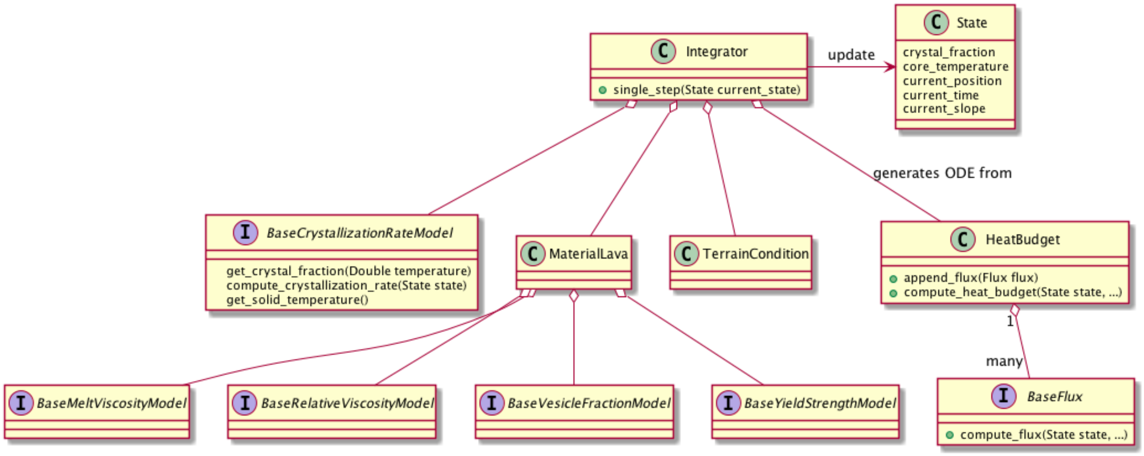

Figure 2: PyFLOWGO UML class diagram - top level. The interfaces (labeled ”I” and with the prefix ”base”) provide parameters to main classes (labeled ”C”) that enable the *Integrator* to update the flow *State* at discrete positions along a slope.

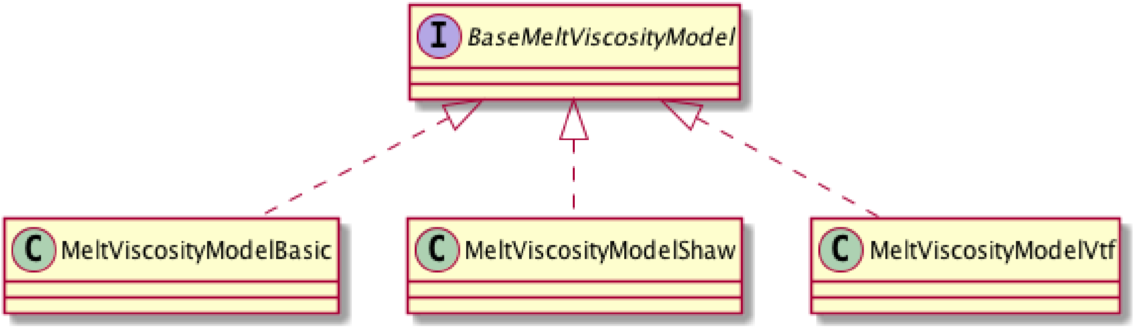

Figure 3: Example of PyFLOWGO UML class diagram describing the interface, *base melt viscosity model, for the various models to compute the melt viscosity (see Appendix A for details about the models). The user is free to chose the model of his choice (see Table A.2 in appendix for the available models at this date) or implement a new model.*

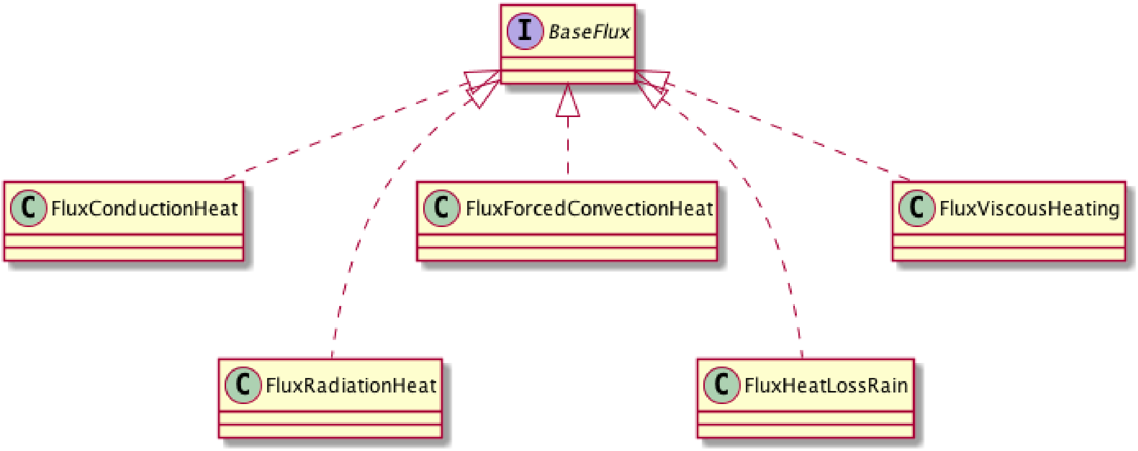

Figure 4: Example of PyFLOWGO UML class diagram describing the interface, *base flux*, for the various heat fluxes (see Appendix A for details about the fluxes). The user is free to chose which fluxes to consider (see Table A.2 in appendix for the available fluxes at this date) or implement a new flux.

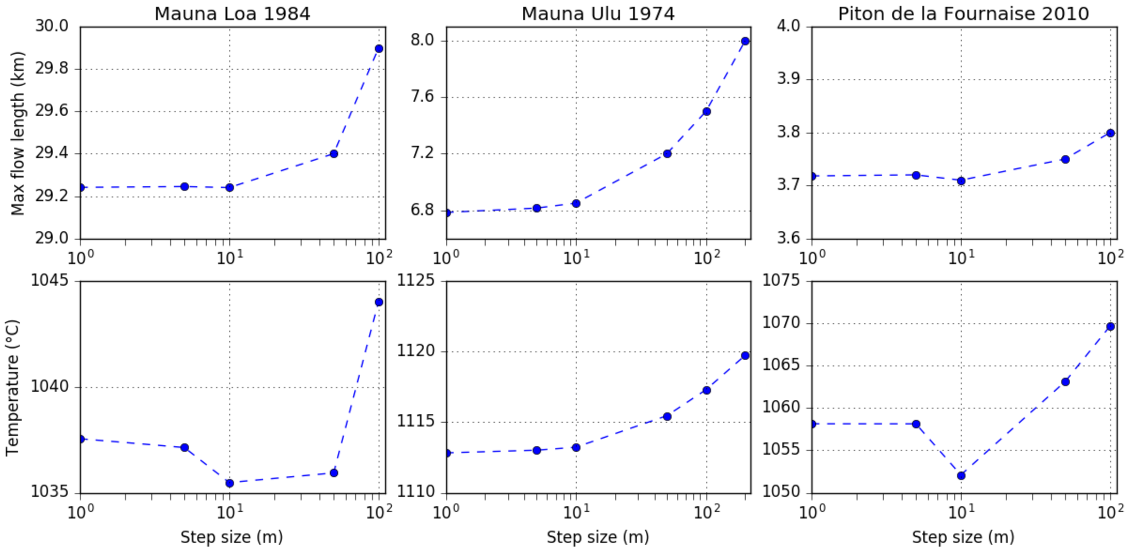

Figure 5: Convergence of the PyFLOWGO runs at 1, 5, 10, 50, and 100 m or 200m step size. The maximum channel length (top) represent the distance reached at v_{mean} = 0 m/s. The convergence of the temperature (bottom) is shown for distances of 29.2, 6.6 and 3.7 km for the three lava flows, respectively. Note that the ML84 results here are for the ”cold” lava as presented in Harris and Rowland (2015) and the PdF2010 is with the LiDAR-derived slope as presented in Harris et al. (2015)

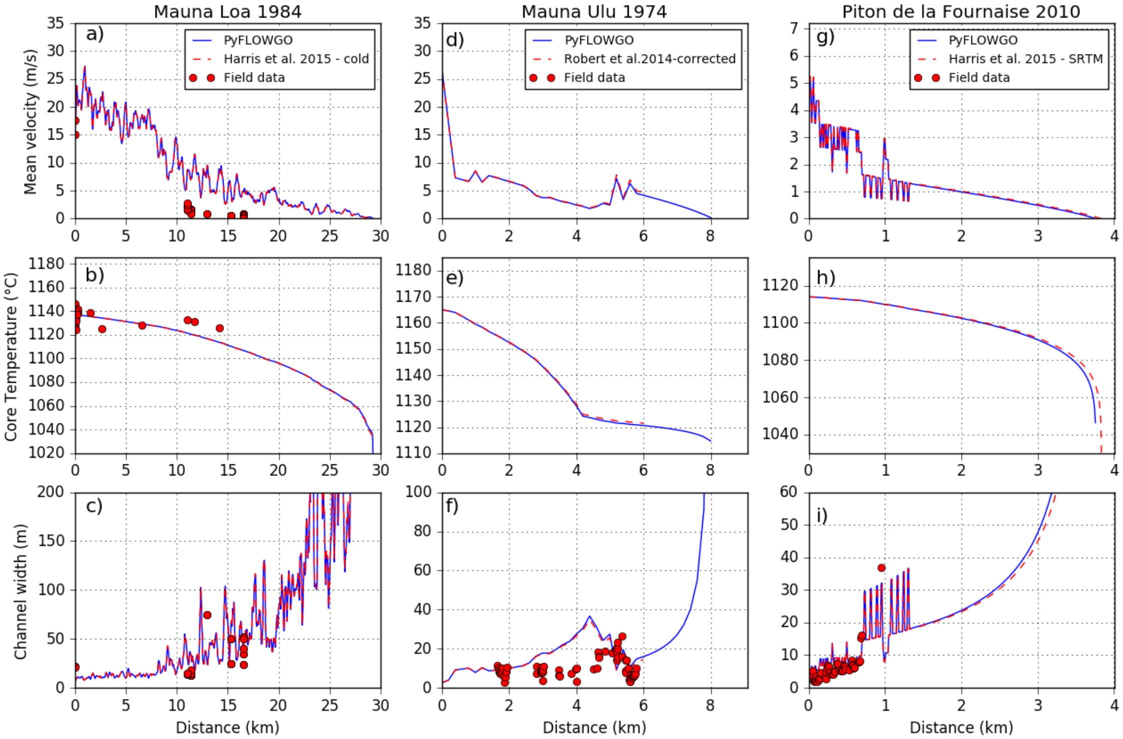

Figure 6: Validation of PyFLOWGO (blue line) against the Excel version of FLOWGO (dashed red line). Here are shown examples of output data (velocity, core temperature and channel width) obtained for Mauna Loa 1984 (a, b, c) according to ”cold” regime as given by Harris and Rowland (2015); for Mauna Ulu 1974 (d, e, f) according to a corrected data set from the published version of Robert et al. (2014); and for Piton de la Founaise 2010 (g, h, i) using the SRTM acquisition slope path from Harris et al. (2015). The input parameters are given in Table A.3 in appendix. Field data are also plotted for comparison. Note also that the line-of-steepest-descent for Mauna Ulu 1974 and for Piton de la Founaise 2001 is given only until 6000 m and 1000 m, respectively; the last section of the slope is therefore equal to the last slope value.

Appendix A. Models

Appendix A.1. Crystallization rate models

Crystallization rate per degree of cooling down flow ($\partial\phi/\partial T_{cool}$) can be calculated via different models. Here we provide four models that the user is free to chose to run in PyFLOWGO.

Appendix A.1.1. Basic model

The basic model proposed by Harris and Rowland (2001) takes into account the amount of crystallization during flow (ϕ_{grown}). This is defined as the amount of crystallization occurring between the eruption temperature (T_{erupt}) and the temperature at which the lava cannot flow anymore (T_{solid}):

$$\frac{\partial\phi}{\partial T_{cool}} = \frac{\phi_{grown}}{T_{erupt} - T_{solid}} \tag{A.1}$$

Appendix A.1.2. Bimodal model as function of distance

This bimodal model was proposed by Robert et al. (2014) and Harris and Rowland (2015) and allows the crystallization rate to be changed after a given distance (x_{critic}).

$$\begin{aligned} &\text{If } x \leqslant x_{critic}\text{: } (\partial\phi/\partial T_{cool}) = \mathrm{C}_1 \\ &\text{If } x > x_{critic}\text{: } (\partial\phi/\partial T_{cool}) = \mathrm{C}_2 \end{aligned} \tag{A.2}$$

where x_{critic}, and the constants C_1 and C_2 are of the user's choice.

Appendix A.1.3. Bimodal model as function of temperature

This bimodal model allows the crystallization rate to be changed after a given temperature (T_{critic}) is reached.

$$\begin{aligned} &\text{If } T_{core} \geqslant T_{critic}\text{: } (\partial\phi/\delta T_{cool}) = \mathrm{C}_1 \\ &\text{If } T_{core} < T_{critic}\text{: } (\partial\phi/\delta T_{cool}) = \mathrm{C}_2 \end{aligned} \tag{A.3}$$

where T_{critic}, and the constants C_1 and C_2 are of the user's choice.

Appendix A.1.4. MELTS model

The MELTS model allows the crystallization rate per degree of cooling to be set from a MELTS-based look-up table as suggested by Harris and Rowland (2001), Harris and Rowland (2015) and Riker et al. (2009). The look-up table is a file containing the amount of crystals (fraction) as a function of temperature (in °C) that must be previously built using the MELTS software of Ghiorso and Sack (1995). A linear interpolation of these data is computed by PyFLOWGO and gives a function (ϕ_{interp}) that represents the fraction of crystals grown as a function of temperature. The fraction of crystals grown per degree of cooling is then computed using the finite differences via the interpolated function:

$$\frac{\partial\phi}{\partial T_{cool}} \approx -\frac{\phi_{interp}(T_{core}+\Delta T) - \phi_{interp}(T_{core}-\Delta T)}{2\Delta T} \tag{A.4}$$

with ΔT being the temperature step that is chosen to be small enough (for example 10^{-6}). Note that this model considers that crystallization happens under equilibrium conditions, which is probably not the case during lava emplacement (e.g., Chevrel et al. 2013; Kolzenburg et al. 2016).

Appendix A.2. Heat flux models

Appendix A.2.1. Radiative heat flux

Heat loss due to radiation from the lava surface to the atmosphere is expressed as:

$$Q_{rad} = \sigma \varepsilon T_{eff}^{4} \, w \tag{A.5}$$

where σ (W/m^2 K^4) is the Stefan − Boltzmann constant, ε is emissivity, w is the channel width and T_{eff} (K) is the effective surface temperature, which is calculated using a two-component model for the lava surface (Pieri and Baloga 1986, Crisp and Baloga 1990, Pieri et al. 1990):

$$T_{eff} = \left[f_{crust}(T_{crust}^{4} - T_{atmo}^{4}) + (1 - f_{crust})(T_{hot}^{4} - T_{atmo}^{4}) \right]^{0.25} \tag{A.6}$$

where T_{atmo} is the temperature of the surrounding atmosphere, f_{crust} is the fraction of crusted lava, T_{crust} is the cool crust temperature, $1 - f_{crust}$ represents the fraction of exposed uncrusted hot lava and T_{hot} is the hot component temperature. The different models used to calculate f_{crust}, T_{crust} and T_{hot} are described in sections Appendix A.7, Appendix A.8 and Appendix A.9, respectively.

Appendix A.2.2. Forced convection heat flux

Heat loss due to forced atmospheric convection from the lava surface is calculated via (e.g. Keszthelyi et al. 2003):

$$Q_{conv} = h_{conv}(T_{conv} - T_{atmo}) \, w \tag{A.7}$$

where h_{conv} is the convective heat transfer (in W/m^2 K) and T_{conv} (K) the characteristic surface temperature. The convective heat transfer depends on atmospheric conditions and can be defined as:

$$h_{conv} = U\, C_H\, \rho_{atmo}\, Cp_{atmo} \tag{A.8}$$

where U is wind speed (m/s), C_H the wind friction factor as defined by Greeley and Iversen (1987), ρ_{atmo} (kg/m^3) is atmospheric density and Cp_{atmo} the heat capacity of the air (J/kg K) in contact with the lava surface. The characteristic surface temperature is calculated via:

$$T_{conv} = \left[f_{crust} T_{crust}^{1.33} + (1 - f_{crust}) \left(T_{hot}^{1.33} \right) \right]^{0.75} \tag{A.9}$$

Appendix A.2.3. Heat flux due to rain

The heat flux due to vaporization of rainwater falling onto the lava surface is expressed by:

$$Q_{rain} = \frac{\partial R}{\partial t} \rho_{H_2O}\, L_{H_2O}\, w \tag{A.10}$$

where $\partial R / \partial t$ (m/s) is the rainfall rate and ρ_{H_2O} (kg/m^3) and L_{H_2O} (J/kg) are, respectively, the density and latent heat of vaporisation of water.

Appendix A.2.4. Conductive heat flux

The heat flux through the base and the levées of the flow occur via conduction and is expressed as (after Keszthelyi 1995a):

$$Q_{cond} = \kappa_{lava} \frac{T_{core} - T_{base}}{h_{base}}\, w \tag{A.11}$$

where κ_{lava} is the thermal conductivity of the lava (in W/mK), T_{core} the lava core temperature (in K), T_{base} (K) the temperature at the base of the basal layer and h_{base} (m) the thickness of the basal layer that is defined between the underlaying surface and the thermal boundary when T_{core}) is reached. It is usually calculated via:

$$h_{base} = d\,H_b/100 \tag{A.12}$$

where H_b is the proportion occupied by the basal layer in respect to the entire flow thickness (d, in m).

Appendix A.2.5. Viscous heating

Viscous heating in the lava channel is expressed here, for a channel that is wider than it is deep ($w > d$) following Costa and Macedonio (2003):

$$Q_{visc} = \eta_{bulk}(V_{mean}/d)^2\, w \tag{A.13}$$

where η_{bulk} (Pa·s) is the bulk viscosity of the molten lava as calculated in section Appendix A.4 and V_{mean} is the mean velocity of the lava as calculated in section 3.3.

Appendix A.3. Density model

PyFLOWGO provides one model to calculate the bulk density:

$$\rho_{bulk} = \phi_b \rho_{DRE} \tag{A.14}$$

where ρ_{DRE} is the density of the dense rock equivalent and ϕ_b is the volume fraction of bubbles in the lava obtained via the methods described in section Appendix A.5.

Appendix A.4. Viscosity models

Magma and lava are complex systems composed of a polydispersed particle mixture of crystals and bubbles of various shapes and sizes in a liquid phase (the silicate melt). The viscosity of this mixture may be defined as:

$$\eta_{bulk} = \eta_{melt}\eta_r \tag{A.15}$$

where the viscosity of the interstitial melt, η_{melt} (Pa s) is Newtonian and depends on temperature and composition, and the relative viscosity, η_r (dimensionless) is obtained by the ratio η_{bulk}/η_{melt}, and depends on the volumetric abundance and aspect ratio of the particles (bubbles and crystals) in the mixture as well as on the strain rate of the flow. PyFLOWGO offers the possibility of calculating the bulk viscosity of the lava using one of four melt viscosity models which can be combined with one of five relative viscosity models.

Appendix A.4.1. Melt viscosity models, η_{melt}

Dragoni and basic model

The Dragoni model calculates the viscosity of the melt at the lava temperature (T_{core}) using the relation proposed by Dragoni (1989):

$$\eta_{melt} = \eta_0 \exp^{0.04(T_0 - T_{core})} \tag{A.16}$$

where η_0 (Pa s) is the viscosity of the lava at the liquidus temperature T_0.

The basic model, as proposed in the original FLOWGO version, is adapted from Dragoni (1989) where instead of liquidus viscosity and temperature, it is the eruption viscosity (η_{erupt}) and temperature (T_{erupt}) that are used in Eq.A.16.

Shaw model

This model calculates the melt viscosity according to the Arrhenian relationship proposed by Shaw (1972) and reformulated here as:

$$log(\eta_{melt}) = \left[s\frac{10^4}{T_{core}} - 1.5s - 6.4)/2.303 \right] - 1 \tag{A.17}$$

where s is the characteristic slope of the η_{melt} versus T_{core} relationship that needs to be computed from the melt chemical composition using Shaw (1972).

VFT model

This model is based on the Vogel-Fulcher-Tammann equation (Vogel 1921, Fulcher 1925, Tammann and Hesse 1926) and takes into account the non-Arrhenian behavior of the melt viscosity Dingwell (1996) and allows η_{melt} to be calculated via:

$$log(\eta_{melt}) = A + \frac{B}{C - T_{core}} \tag{A.18}$$

where A (Pa s), B (J/mol) and C (K) are fitting parameters that depend on chemical composition. These fitting parameters need to have been previously determined either from viscosity measurements at high and low temperature or from the melt chemical composition using for example the model proposed by Giordano et al. (2008).

Appendix A.4.2. Relative viscosity models

The first four relative viscosity models given here take into account the affect of crystals whereas the fifth model considers those of both crystals and bubbles. More complex formulations may take into account bimodal particle size distribution and shape (e.g. Castruccio et al. 2010; Cimarelli et al. 2011; Moitra and Gonnermann (2015)) or bubble content as a function of their ability to deform (Llewellin and Manga 2005; Pal 2003) but they are not presented here.

Einstein-Roscoe model

This model calculates the effect of crystals on viscosity according to the Einstein-Roscoe relationship, as first introduced by Shaw (1965) and as used in the original FLOWGO version:

$$\eta_r = (1 - R\phi)^{-2.5} \tag{A.19}$$

where here R = 1.51 for spherical solid particles, as suggested by Pinkerton and Stevenson (1992), this equation is therefore only applicable for spherical particles and for a volume fraction maximum of 0.66 (i.e., $1/R$).

Krieger-Dougherty model

This model calculates the effect of crystals on viscosity according to the Krieger-Dougherty relationship (Krieger 1972, Krieger and Dougherty 1959, Pabst 2004):

$$\eta_r = (1 - \phi/\phi_m)^{-b\phi_m} \tag{A.20}$$

where b is the Einstein coefficient (also termed intrinsic viscosity) and ϕ_m is the crystal maximum packing, both being fitting parameters that depend on particle shape. In theory, for spherical particles (aspect ratio of 1) this relationship reduces to Eq. A.19. For elongated particles of aspect ratio of approx. 9, Mueller et al. (2010) give b = 6.07 and ϕ_m= 0.343. See for more examples Mueller et al. 2010, Cimarelli et al. 2011, and Mader et al. 2013.

Maron-Pierce model

This model calculates the effect of crystal cargo on relative viscosity according to Maron and Pierce (1956):

$$\eta_r = (1-\phi/\phi_m)^{-2} \tag{A.21}$$

where ϕ_m is a fitting parameter that depends on particle shape (e.g. Mueller et al. 2010 and Mader et al. 2013). For example, Mueller et al. (2010) use ϕ_m = 0.633 for spherical particles and ϕ_m = 0.339 for elongated particles with an aspect ratio of approx. 9.

Costa model

The Costa model allows the effect of crystal fraction in an intermediate range of crystallinity (30 to 80 vol % crystals) to be calculated by taking into account applied deformation (strain rate) following Costa et al. (2009):

$$\begin{gathered} \eta_r = \frac{1+\left(\frac{\phi}{\phi_*}\right)^{\delta}}{(1-F)^{b\phi_*}} \\ \text{in which:} \\ F = (1-\xi)\mathrm{erf}\left[\frac{\sqrt{\pi}}{2(1-\xi)}\frac{\phi}{\phi_*}\left(1+\left(\frac{\phi}{\phi_*}\right)^{\gamma}\right)\right] \end{gathered} \tag{A.22}$$

here, ϕ_* is the critical solid fraction that is present at the onset of the exponential increase in η_r with ϕ; γ is the slope of the relation between η_r and ϕ as the crystal fraction approaches ϕ_*, and δ is the slope of the relations for values of ϕ greater than ϕ_*. ξ, γ and δ are all empirical parameters that depend on particle shape and applied strain rate. Two default models are offered in PyFLOWGO: *costa*1 that is only applicable for spherical particles (aspect ratio of 1) and $costa2$ that is for elongated particles (aspect ratio of 9). Both models can be used for strain rate set either at $1s^{-1}$ or $10^{-4}s^{-1}$ and use the values given in Cimarelli et al. (2011). For example, at $1s^{-1}$, $costa1$ produces: ϕ_*= 0.67, ξ = 0.01, γ =1.6 and δ =11.4, while $costa2$ produces: ϕ_*= 0.28, ξ =0.001, γ =8.55 and δ =4.45. More examples can be found in Costa et al. 2009, Cimarelli et al. (2011) and Chevrel et al. (2013).

Phan-Thien and Pham model

PyFLOWGO offers one model that allows the treatment as a three-phase mixture comprising a suspension of rigid spherical particles (ϕ) and bubbles (ϕ_b) following Phan-Thien and Pham (1997). This model is applicable only for $\phi + \phi_b < 1$. One of the three following cases can be applied:

Case *ptp1*, crystals are smaller than bubbles:

$$\eta_r = \left(1 - \frac{\phi}{1 - \phi_b}\right)^{-5/2} (1 - \phi_b)^{-1} \tag{A.23}$$

Case *ptp2*, crystals and bubbles are the same size:

$$\eta_r = (1 - \phi - \phi_b)^{\frac{5\phi - 2\phi_b}{2\phi - \phi_b}} \tag{A.24}$$

Case *ptp3*, crystals are larger than bubbles:

$$\eta_r = \left(1 - \frac{\phi_b}{1 - \phi}\right)^{-1} (1 - \phi)^{-5/2} \tag{A.25}$$

Appendix A.5. Vesicle fraction models, ϕ_b

Two vesicle state models are available. The first is a simple model whereby the vesicle fraction is held constant down flow and is equal to the initial (at vent) value.

The second is a model as proposed by Harris and Rowland (2015), which allows the vesicle fraction to be changed after a given distance, x_{critic}, and is intended to take into effect down flow degassing. In the PyFLOWGO framework these are the *constant* and *bimodal* model, respectively. In the bimodal model:

$$\begin{aligned} &\text{If } x \leqslant x_{critic}\text{: } \phi_b = \phi_{b1} \\ &\text{If } x > x_{critic}\text{: } \phi_b = \phi_{b2} \end{aligned} \tag{A.26}$$

where x_{critic}, ϕ_{b1} and ϕ_{b2} are the proximal and distal vesicularities, respectively and can be set using down flow assessments of lava density (e.g. Robert et al. 2014).

Appendix A.6. Yield strength and shear stress model

Velocity depends also on the yield strength of lava and on the basal shear stress (Eq. 7). PyFLOWGO provides one basal shear stress (τ_b) model (Hulme 1974):

$$\tau_b = dg\rho_{bulk}\sin(\theta) \tag{A.27}$$

where τ_b is in Pa.

Lava yield strength can instead be calculated as a function of temperature and crystallinity following Dragoni (1989), and Pinkerton and Stevenson (1992) as proposed in the original version of FLOWGO. PyFLOWGO uses this approach of allowing τ_0 to be calculated as function of lava temperature (T_{core}) using the liquidus temperature (T_0) and the lava crystal content (ϕ) as proposed by Ryerson et al. (1988) in:

$$\tau_0 = 0.01\left[exp^{0.08(T_0 - T_{core})} - 1\right] + 6500\phi^{2.85} \tag{A.28}$$

Following Harris and Rowland (2001), PyFLOWGO also allows a model that considers the eruption temperature (T_{erupt}) instead of T_0 in Eq.A.27. In the PyFLOWGO framework these are the *dragoni* and *basic* model, respectively.

Appendix A.7. Effective crust cover fraction model

The upper surface of the lava is partially covered by a cooler crust. The fraction of this crusted lava is termed as the effective crust cover fraction, f_{crust}, and varies between zero (crust free, no insulation: rare in nature) and one (complete crust coverage, well-insulated; but not equivalent to a lava tube). Effective crust cover fraction directly affects the effective surface temperature (Eq. A.6) and the characteristic surface temperature (Eq. A.9) which, in turn, influences the heat fluxes due to radiation and forced convection. PyFLOWGO offers two models to calculate f_{crust}.

The *basic* model, as proposed in the original version of FLOWGO, allows f_{crust} to vary down flow as function of velocity:

$$f_{crust} = f_{init} \exp^{\alpha V_{mean}} \tag{A.29}$$

where f_{init} is the initial (at vent) crust fraction and α is a coefficient that varies crust cover as function of V_{mean}: crust cover increases as flow velocity decreases. Based on examination of aerial photographs of active channels flowing at known velocities, Harris and Rowland (2001) derived f_{init} of 0.9 and α of -0.16, for poorly insulated flow, and f_{init} of 1.0 and α of -0.00756 for more heavily crusted flow. Alternatively, f_{crust} can be held constant down flow, equals f_{init} at all down flow location when $\alpha = 0$.

The second model, named *bimodal* in PyFLOWGO, allows the dependence of effective crust cover fraction with velocity to be changed after a given distance, x_{critic}, as proposed by Harris and Rowland (2015):

$$\begin{aligned} &\text{If } x \leqslant x_{critic}\text{: } f_{crust} = f_{init} \exp^{\alpha_1 V_{mean}} \\ &\text{If } x > x_{critic}\text{: } f_{crust} = f_{init} \exp^{\alpha_2 V_{mean}} \end{aligned} \tag{A.30}$$

where α_1 and α_2 are the crust cover growth coefficients for proximal and distal channel reaches, respectively and are determined by field observation; for which we need more measurements [i.e. for the relationship between V_{mean} and f_{crust} (Harris and Rowland 2015)].

Appendix A.8. Crust temperature models

PyFLOWGO provides three models to calculate the temperature of the crust, T_{crust}. The *constant* model allows the at-vent initial crust temperature to be held constant down flow. The *hon* model, as suggested in the original FLOWGO version, allows calculation of T_{crust} (in °C) following Hon et al. (1994):

$$T_{crust} = -140 \log\left(\frac{time}{3600}\right) + 303 + 273.15 \tag{A.31}$$

where *time* is in seconds and is calculated via:

$$time = \partial x / V_{mean} \tag{A.32}$$

in which ∂x is one down flow distance increment. This equation implies implicitly that the initial crust temperature is 1070°C and as it is an empirical relationship determined from Hawaiian pahoehoe lava, but - given that it is based on the Stefan cooling problem (Harris et al. 2005)- it can be adapted to any basaltic surface cooling due to radiation.

The third model, as suggested by Harris and Rowland (2015), allows T_{crust} to vary as function of time according to Hon et al. (1994) from the vent until a given distance, x_{critic}, and then to be held constant:

$$\begin{aligned} &\text{If } x \leqslant x_{critic}\text{: } T_{crust} = -140 \log\left(\frac{time}{3600}\right) + 303 + 273.15 \\ &\text{If } x > x_{critic}\text{: } T_{crust} = T_{init} \end{aligned} \tag{A.33}$$

In the PyFLOWGO framework, this is the *honbimodal* model.

Appendix A.9. Uncrusted surface temperature model

The temperature of the uncrusted lava surface, T_{hot}, will be lower than of the flow core (e.g. Calvari et al. 1994, Flynn and Mouginis-Mark 1994, Harris et al. 1998). PyFLOWGO accounts for this difference via:

$$T_{hot} = T_{core} - buffer \tag{A.34}$$

where *buffer* is the temperature difference between the maximum surface temperature and the core temperature, and is set by the user. Based on field measurements using thermocouples and radiometers, Harris and Rowland (2001) give *buffer* = 140. The buffer value may also be lower; Bailey et al. (2006) described an active channel on Etna with a maximum surface temperature of 1042 °C and a core temperature of 1065 °C .

Table A.1: Description of the input parameters contained in the json file

Input parameters in json file	Symbol	Definition	Unit	Constant for Earth
lava_name		name of the lava flow	n.a	
slope_file		file containing distance (m) and slope (°)		
step_size		step size for lava advance down flow	m	
terrain_conditions				
width	w	channel’s width	m	
depth	d	channel’s depth	m	
gravity	g	gravity of the planet	m/s^2	9.81
max_channel_length	L_{max}	maximum flow length*	m	
eruption_conditions				
eruption_temperature	T_{erupt}	temperature of the eruption	K	
viscosity_eruption	η_{erupt}	viscosity of the lava at T_{erupt}, only for ”basic” melt viscosity model	Pa s	
lava_state				
position	x	distance from the vent at which the iteration starts	m	
critical_distance	x_{critic}	distance when the bimodal models change, only when ”bimodal” models”	m	
time	t	time at which the iteration starts, only for ”hon” and ”honbimodal” crust temperature model	s	
crystal_fraction	ϕ	initial crystal fraction	n.a	
density_dre	ρ_{DRE}	dense rock equivalent density	kg/m^3	
vesicle_fraction	ϕ_b	initial fraction of vesicle at the vent	n.a	
liquidus_temperature	L_0	temperature of the liquidus , only for ”dragoni” melt viscosity model	K	
radiation_parameters				
stefan-boltzmann_sigma	σ	stefan-boltzmann constant	W/m^{2}K^4	5.669E-8
emissivity_epsilon	ε	emissivity	n.a.	0.98
conduction_parameters				
basal_temperature	T_{base}	temperature at the base of the flow	K	
core_base_distance	H_b	percentage of base layer over flow depth	%	
rain_parameters				
rainfall_rate	$\partial R/\partial t$	rainfall rate	m/s	
density_water	ρ_{H_2O}	density of the water	kg/m^3	958
latent_heat_vaporization	L_{H_2O}	latent heat of vaporisation of the water	J/kg	2800000
convection_parameters				
wind_speed	U	wind speed	m/s	
ch_air	C_H	value from Greeley and Iversen (1987)**	n.a.	3.599E-3
air_temperature	T_{atmo}	temperature of the air	K	
air_density	ρ_{atmo}	density of the air	kg/m^3	0.4411
air_specific_heat_capacity	Cp_{atmo}	heat capacity of the air	J/kg K	1099
thermal_parameters				
buffer	$buffer$	difference between T_{core} and T_{hot}	K	
crust_cover_fraction	f_{init}	initial crust cover fraction	n.a.	
alpha	α	coefficient for velocity dependence of the crust cover	n.a	
initial_crust_temperature	T_{init}	chilled crust temperature	K	
melt_viscosity_parameters				
shaw_slope	s	coefficient calculated from melt chemical composition only for ”shaw” melt viscosity model		
a_vft	A	coefficient calculated from melt chemical composition, only for ”vft” melt viscosity model	Pa.s	
b_vft	B	coefficient calculated from melt chemical composition, only for ”vft” melt viscosity model	J/mol	
c_vft	C	coefficient calculated from melt chemical composition, only for ”vft” melt viscosity model	K	
crystals_parameters				
crystals_grown_during_cooling	ϕ_{grown}	fraction of crystal grown during emplacement, only for ”basic” crystallization rate model	n.a	
solid_temperature	T_{solid}	temperature at which the lava cannot flow, only for ”basic” crystallization rate model	K	
crystallization_rate_1	C_1	crystallization rate, only for ”bimodal” crystallization rate model	crystals/°C	
crystallization_rate_2	C_2	crystallization rate, only for ”bimodal” crystallization rate model	crystals/°C	
latent_heat_of_crystallization	L	latent heat of crystallization	J/kg	350000
relative_viscosity_parameters				
max_packing	ϕ_m	maximum fraction of crystals, only for ”kd” and ”mp” relative viscosity models	n.a.	
einstein_coef	b	Einstein coefficient or intrinsic viscosity, only for ”kd” relative viscosity models	Pa.s	
strain_rate		strain rate, only for ”costa1” and ”costa2” relative viscosity models	s^{-1}	0.0001 or 1

*used in case the limiting conditions ($v_{mean} = 0$ or $\phi = \phi_{max}$ or $T_{core} = T_{solid}$) are not reached.

** $C_H = (U'/U)^2$ where U' is the fraction of wind speed according to Keszthelyi and Denlinger (1996).

Table A.2: Models' choice

Model's name	Symbol	What to write in the json file
heat_budget_models		
radiation	Q_{rad}	"yes" / "no"
conduction	Q_{cond}	"yes" / "no"
convection	Q_{conv}	"yes" / "no"
rain	Q_{rain}	"yes" / "no"
viscous_heating	Q_{visc}	"yes" / "no"
models		
crystallization_rate_model	$\partial\phi/\partial T$	"basic" / "bimodal" / "bimodal_f_temp" / "melts"
melt_viscosity_model	η_{melt}	"basic" / "dragoni" / "shaw" / "vft"
relative_viscosity_model	η_r	"er" / "mp" / "kd" / "costa1" / "costa2" / "ptp1" / "ptp2" / "ptp3"
yield_strength_model	τ_0	"basic" / "dragoni"
crust_temperature_model	T_{crust}	"basic" / "hon" / "bimodal"
effective_cover_crust_model	f_{crust}	"basic" / "bimodal"
vesicle_fraction_model	ϕ_b	"constant" / "bimodal"

Table A.3: Input parameters of the json files used for the test cases

”lava_name”	ML84_HR2015_cold	MU74_Robertetal2014	PdF2010_srtm
”slope_file”	ML84-slope_file.txt	DEM_maunaulu74.txt	DEM_pdf2010_srtm.txt
”step_size”	10.0	200.0	10.0
”models”			
”crystallization_rate_model”	basic	bimodal	basic
”melt_viscosity_model”	shaw	vft	vft
”relative_viscosity_model”	er	ptp2	er
”yield_strength_model”	dragoni	basic	basic
”crust_temperature_model”	hon	constant	constant
”effective_cover_crust_model”	basic	basic	basic
”vesicle_fraction_model”	constant	constant	constant
”heat_budget_models”			
”radiation”	yes	yes	yes
”conduction”	yes	yes	yes
”convection”	yes	yes	yes
”rain”	no	no	no
”viscous_heating”	no	no	no
”terrain_conditions”			
”width”	21	2.475	4.5
”depth”	3	2.475	1.4
”gravity”	9.8	9.8	9.8
”max_channel_length”	50000	10000	4000
”eruption_condition”			
”eruption_temperature”	1410.15	1438.15	1387.15
”lava_state”			
”critical distance”	0	4190	0
”position”	0	0	0
”time”	1	0	1
”crystal_fraction”	0.25	0.1097	0.104
”density_dre”	2724	2900	2970
”vesicle_fraction”	0.140	0.4866	0.64
”liquidus_temperature”	1473.15	1473.15	0
”radiation_parameters”			
”stefan-boltzmann_sigma”	5.67e-8	5.67e-8	5.67e-8
”emissivity_epsilon”	0.95	0.98	0.95
”conduction_parameters”			
”basal_temperature”	773.15	1273.15	773.15
”core_base_distance”	19	19	19
”convection_parameters”			
”wind_speed”	5.12259623	5.12259623	5
”ch_air”	0.0036	0.0036	0.0036
”air_temperature”	283.15	293.15	293.15
”air_density”	0.4412	0.4412	0.4412
”air_specific_heat_capacity”	1099	1099	1099
”thermal_parameters”			
”buffer”	140	0	140
”crust_cover_fraction”	0.9023	0.5	1
”alpha”	-0.1601	0	-0.0076
”crust_temperature”	698.15	1273.15	773.15
”melt_viscosity_parameters”			
”shaw_slope”	2.36	0	0
”a_vft”	0	-4.7	-4.52
”b_vft”	0	5429.7	5558
”c_vft”	0	595.5	582.9
”crystals_parameters”			
”crystals_grown_during_cooling”	0.45	0.89	0.89
”solid_temperature”	1243.15	1268.15	1237.15
”latent_heat_of_crystallization”	350000	350000	350000
”crystallization_rate_1”	-	0.003	-
”crystallization_rate_2”	-	0.025	-