

## **FRActured Conduits: permeability ( $k$ ) Reduction**

This document accompanies "*Time-dependent permeability evolution in compacting volcanic fracture systems and implications for gas overpressure.*" by Farquharson, J. I., Wadsworth, F. B., Heap, M. J., and Baud, P. It describes the format of the model discussed in the article, and provides instructions for its use.

In order to explore the time-dependent evolution of equivalent permeability of a fractured conduit margin, the series of equations outlined in the text have been combined into a multi-stage algorithm provided as Supplementary Material 1. The model (FRACkR.m: **FRActured Conduits: permeability ( $k$ ) Reduction**) has been written and optimized for MATLAB<sup>®</sup>, a multi-paradigm numerical computing environment and programming language. In order to run the model, the FRACkR folder (Supplementary Material) must be unzipped and set as the current folder. (To identify the current folder, type `pwd` into the MATLAB<sup>®</sup> Command Window.) The model FRACkR.m must then be called (`run FRACkR.m`). The user will be prompted to input a filename, which defines an .xls file into which the modelled data are saved.

The user is first prompted to define the timescale in days over which to compute the permeability evolution of their system. To minimize runtime, the model employs a fixed number of timesteps. Thus, there is a tradeoff between resolution and timescale, which is mitigated by employing a logarithmic time timespan distribution (this means that the first timestep is the smallest, with steps getting progressively larger as the model continues). The user is next prompted to input the temperature of the system (in °C), and then to define the "fracture window", *i.e.* the range of depths relevant to the scenario (for example, the user may be interested in fractures generated between 1600 and 1400 m depth). The user must next input a fracture density (the mean number of fractures per metre) and a mean fracture width in metres. The user will be alerted if the cumulative fracture width exceeds half of the total lengthscale of interest, and the script will request a narrower mean fracture width. If the

## **FRActured Conduits: permeability ( $k$ ) Reduction**

crystal content is known, the user can also define this at this stage. The user will be alerted if the input crystal fraction exceeds 1 or is less than 0, and script will request a new crystal fraction.

The model then generates random depths up to the user-defined fracture density. If the distance between any two fractures is less than half of the mean fracture width, the fracture generation function reiterates until this is not the case, avoiding fracture overlap. The user is given the option of calculating the equilibrium water content for each fracture, or inputting a single value in wt. %.

Once the model has been run, the script will display three figures, as follows:

Figure 1: a four panel figure plotting fracture porosity against time, fracture permeability against time, fracture permeability against fracture porosity, and equivalent permeability evolution with time.

Figure 2: Equivalent permeability evolution with time, contoured for a range of host rock permeabilities ranging from  $10^{-22}$  to  $10^{-10} \text{ m}^2$ .

Figure 3: A schematic of the model, indicating the generated fracture positions with respect to depth.

Further, an excel (.xls) file will be saved (under the filename designated by the user) containing three sheets. The first of these contains metadata pertaining to the model run, namely the user-input data. Sheet two contains the timesteps in seconds, and the equivalent permeability of the system for host rock permeabilities ranging from  $10^{-22}$  to  $10^{-10} \text{ m}^2$ . Sheet three contains the depths of all fractures generated.

### **Altering the model**

## FRActured Conduits: permeability ( $k$ ) Reduction

Prompting user inputs for every conceivable parameter would greatly diminish the usability of our model. Nevertheless, we recognize that users may desire to make use of experimentally- or literature-derived constants or equations rather than the default values. As such, the accompanying MATLAB<sup>®</sup> program (see Supplementary Material 1) has been structured such that it is easily adaptable to suit user requirements. The constituent models and the parameters therein may be adjusted to reflect new experimental data, for example, or may be substituted should a different model better account for the specific problem parameters of the user. For instance, we employ a two-slope porosity-permeability model (after Heap *et al*<sup>[1]</sup>): a single-slope model may be implemented simply by equating  $A$  and  $C$ , and  $B$  and  $D$ . The empirical equation governing compaction is taken from Russell and Quane<sup>[2]</sup>. Constants are stored in a function:

```
[alpha, phi_i, rho, g, A, B, C, D, phi_c] = constants();
```

including sintering constants  $\alpha$  ( $\alpha$ ),  $\phi_i$  ( $\phi_i$ ),  $\rho$  ( $\rho_b$ ), and  $g$ . Permeability-porosity relation constants are defined as  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $\phi_c(x^*)$ .

Please refer to Notation Used (Table 1) in the accompanying article for definitions of these constants.

### Functions

All functions employed by the model are presented below, along with a brief description.

The function `inputs(nt)` allows the user to input values for the timescale to compute, the temperature of the system, the depth range or "window" of interest, the mean fracture width and fracture density, and the crystal cargo of the compacting magma. Embedded in `inputs(nt)` are the following three functions:

```
fracture_width(nf,1);
```

## FRACTured Conduits: permeability (*k*) Reduction

```
fracture_generator(window_min, window_max,nf, mean_wf);  
Crystal();
```

The first of these [`fracture_width(nf,1)`] ensures that the cumulative fracture width is not greater than the overall length of intact material. If this assumption is violated, the user will be prompted to input a smaller value for the mean fracture width.

`fracture_generator(window_min, window_max,nf, mean_wf)` employs a uniform random probability distribution to randomly assign depths to fractures within the depth range imposed by the user, up to the user-defined fracture density. The function will run iteratively until the distance between each fracture is at least half of the mean fracture width, in order to avoid overlap.

The third function [`Crystal()`] ensures that the crystal fraction imposed by the user is between 0 and 1 (i.e. between 0 and 100 vol.%). If the input value is out of this range, the user will be prompted to re-enter their value.

The initial viscosity is determined by the function `viscosity(T, Xc, rho, g, w, sizew)`, which calculates the lithostatic stress driving compaction. The function also allows the user to calculate the equilibrium water content using `equilibrium(T,sigma)`, which employs the solubility law proposed by Liu *et al.*<sup>[3]</sup>, or to input their own value. It then calls the function `HD(T, H2O, Xc)`, described below.

`HD(T, H2O, Xc)`, is the viscosity model of Hess and Dingwell<sup>[4]</sup>, developed for hydrous calc-alkaline rhyolites, and allows the non-Arrhenian temperature dependency of melt viscosity to be calculated. To account for the influence of crystal content on rheology, the function `HD(T, H2O, Xc)` calls `Xcontent(eta_0,Xc,Xm)`, which comprises the particle-suspension model of Mueller *et al.*<sup>[5]</sup>.

`timescale(T, mean_wf, A, B, phi_i, eta_0)` calculates the ratio between the mean fracture width and the compaction lengthscale, given by  $\delta$  ( $\delta$ ). This requires the

## FRActured Conduits: permeability ( $k$ ) Reduction

pore fluid viscosity  $\nu$  ( $\nu$ ) to be known, which is determined by `pore_fluid(sigma, T)`. In order to compute the pore fluid viscosity, the function `pore_fluid(sigma, T)` calls the auxiliary functions `region(sigma, T)`, `density(sigma, T, region)` and (depending on the imposed thermodynamic region) `enthalpy(p)`. Embedded in `region(sigma, T)` are functions `B23(T)` and `saturation_pressure(T)`, which calculate boundary conditions within  $p$ — $T$  space as described in the supporting information (Appendix A). These functions are based on the constants and equations outlined by IAPWS<sup>[6 - 8]</sup>. Using the pore fluid viscosity,  $\delta$  is then calculated as described in the accompanying article. A further function [`Darcy_compaction(eta_0, A, B, phi_i, alpha, T, mean_wf)`] computes the Darcy compaction number, which delimits whether or not pore pressure can increase within sintering fractures. If this value is less than one, the user is warned that the pore pressure may not be in equilibrium, and given the option on whether they wish to continue with the computation (refer to accompanying article for explanation). This function outputs the critical permeability threshold  $k_{cr}$  ( $k_{cr}$ ), below which pore pressure may increase.

The final function [`figures(tt, P, time, KK, t, n, ke, w, kk, window_max)`], controls figure formatting and output.

We highlight that the constituent functions `HD.m`, `equilibrium.m`, and `Xcontent.m` summarize the works of Hess and Dingwell<sup>[4]</sup>, Liu *et al.*<sup>[3]</sup>, and Mueller *et al.*<sup>[5]</sup>, respectively, and should be cited as such if used independently of our presented model. Similarly, the functions `region(sigma, T)`, `density(sigma, T, region)`, `enthalpy(p)`, `B23(T)` and `saturation_pressure(T)`, rely on the works of IAPWS<sup>[6 - 8]</sup>, which should similarly be referenced if used independently of our presented model.

### Using functions independently of the model.

## FRActured Conduits: permeability (*k*) Reduction

It may often be useful to calculate specific parameters separately from the time-dependent permeability evolution of a fractured volcanic conduit. For this reason, FRACKR.m has been presented as a series of functions which may be used autonomously. For example, `pore_fluid(sigma, T)` may be used to calculate the viscosity of water vapour under a range of pressure and temperature conditions: if one were interested in the viscosity of water at 30 MPa and 800 °C, then the command `pore_fluid(30000000, 800)` would yield a value of  $4.1844 \times 10^{-5}$  Pa.s. Similarly, `HD(T, H2O, Xc)` may be used to determine melt viscosity for a given temperature, water content, and crystallinity. If pressure in Pa is known, then the equilibrium water content may be incorporated as well, for example: `HD(800, equilibrium(800, 30000000), 0.15)` would calculate a melt viscosity of  $3.3550 \times 10^6$  Pa.s. An additional function `Diffusivity_time(H2O, T, a)` is also included so that the user may estimate the characteristic time for molecular diffusion of water (after Zhang *et al.*<sup>[9]</sup>), as described in the accompanying manuscript, the inputs being the water content, temperature, and a characteristic particle radius (*a*). Further, a function `GRD(T, H2O, Xc)` is included, which allows users to substitute the viscosity model of Hess and Dingwell<sup>[4]</sup> for that of Giordano *et al.*<sup>[10]</sup>. To employ this model, the oxide fractions of the melt must to be known, and the `HD.m` model must be commented out in the `viscosity.m` function.

## References cited

- [3] Liu, Y., Zhang, Y. and Behrens, H., 2005. Solubility of H<sub>2</sub>O in rhyolitic melts at low pressures and a new empirical model for mixed H<sub>2</sub>O–CO<sub>2</sub> solubility in rhyolitic melts. *Journal of Volcanology and Geothermal Research*, 143(1), pp.219-235.
- [4] Hess, K.U. and Dingwell, D.B., 1996. Viscosities of hydrous leucogranitic melts: A non-Arrhenian model. *American Mineralogist*, 81(9-10), pp.1297-1300.

## **FRActured Conduits: permeability ( $k$ ) Reduction**

- [5] Mueller, S., Llewellyn, E., and Mader, H., 2010, The rheology of suspensions of solid particles. *Proc. R. Soc. A*, 466, (2116) 1201-1228.
- [6] IAPWS, 2014. Revised Supplementary Release on Backward Equations for the Functions  $T(p,h)$ ,  $v(p,h)$  and  $T(p,s)$ ,  $v(p,s)$  for Region 3 of the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. IAPWS meeting June 2014: Moscow, Russia.
- [7] IAPWS, 2008. Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance. IAWPS meeting September 2008: Berlin, Germany.
- [8] IAPWS, 2012. Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam. IAPWS meeting August 2007: Lucerne, Switzerland.
- [9] Zhang, Y., Stolper, E.M. and Wasserburg, G.J., 1991. Diffusion of water in rhyolitic glasses. *Geochimica et Cosmochimica Acta*, 55(2), pp.441-456.
- [10] Giordano, D., Russell, J.K. and Dingwell, D.B., 2008. Viscosity of magmatic liquids: a model. *Earth and Planetary Science Letters*, 271(1), pp.123-134.