GDIFF: a Finite Difference code for the calculation of multicomponent diffusion in garnet

Evangelos Moulas[1*]

[1*]Institute of Geosciences & Mainz Institute of Multiscale Modeling (M^3ODEL), Johannes-Gutenberg University Mainz, Mainz,Germany

(evmoulas@uni-mainz.de)

This documentation describes the main features of GDIFF (**G**arnet **Diff**usion). GDIFF is set of MATLAB routines that can be used to calculate the concentration profiles of garnet in 1 dimension. The routines must all be placed in a common folder. These routines are:

- diffusion_grt_v8_VECT.m
- GDIFF_time.m
- GDIFF_cooling.m

and they can be run by typing GDIFF_time or GDIFF_cooling in the MATLAB command window.

The chemistry of garnet that is considered is described by the mineral formula $[Ca,Mg,Fe,Mn]_3Al_2Si_3O_{12}$. GDIFF utilizes the conservative, finite-difference method for the solution of the diffusion problem in 1 dimension (also for spherical or cylindrical coordinates). The code has been written in general form using functions that would allow the more transparent presentation of the results. More technical details follow below. The software and the present documentation are provided free of charge[1]. At this point, all the provided routines have been tested for compatibility with OCTAVE.

Current Version: 1.1 (16-May-2023) – doi: 10.5281/zenodo.7805990

In case of questions please send an email to evmoulas@uni-mainz.de

[1] Creative Commons Attribution 4.0 International

Contents

Introduction

The purpose of the present document is not to provide a detailed review of the literature on garnet diffusion, but to provide a concise introduction on the methods that are implemented in GDIFF and to show the respective governing equations. I have tried to keep the text to a minimum and show how the program can be easily used. Results from this code have been used already in published work done by the author (Burg & Moulas, 2022; Cheng et al., 2020) and more details on the theoretical formulation can be found in the citing literature (Chakraborty & Ganguly, 1991, 1992; Lasaga, 1979).

Governing Equations

In one dimension, the partial differential equations governing the multicomponent diffusion in garnet are:

$$\frac{\partial C_i}{\partial t} = \frac{1}{x^{n-1}}\frac{\partial}{\partial x}\left(x^{n-1}D_{ij}\frac{\partial C_j}{\partial x}\right) \tag{1}$$

where t is time, x is the spatial direction, $C(x,t)$ indicates the concentration, $i,j \in \{1,2,3\}$ represent the i,j^{th} independent endmembers (Almandine-Pyrope-Spessartine) and $D_{ij}(x,t)$ represents the multicomponent matrix of diffusivities which is generally concentration dependent. For the case of garnet, Grossular is taken as the fourth (dependent) component and can be calculated from $C_4 = 1 - \sum_{i=1}^{3} C_i$. The dimension number $n \in \{1,2,3\}$ can be used to choose a planar ($n = 1$), cylindrical ($n = 2$), or spherical ($n = 3$), geometry.

The Diffusivity matrix is calculated following the approach of Lasaga (1979) which can be simplified as:

$$D_{ij} = D_i^*\delta_{ij} - \frac{D_i^* X_i}{\sum_{k=1}^{4} D_k^* X_k}\left(D_j^* - D_4^*\right) \tag{2}$$

where $D^*$is the tracer-diffusion coefficient and δ_{ij} is the Kronecker delta (1 when $i = j$ and 0 otherwise). X_i in eq. (2) represent molar fractions. In this work, the effect of density variations during diffusion is ignored and therefore eq. (1) can be written in terms of concentrations or molar factions. Within GDIFF, the tracer diffusion coefficients are taken from the experimental dataset of Chakraborty and Ganguly (1992). This choice of diffusion coefficients was made since this set of diffusion coefficients seemed to fit natural data better (Cheng et al., 2020). However, the code can be easily modified to use other values.

In the work of Chakraborty and Ganguly (1992) only the tracer diffusion coefficients of Fe, Mg and Mn are provided, while the tracer diffusivity of Ca is calculated implicitly as half of that of Fe (e.g. Fig. 4 in Chakraborty and Ganguly, 1992). The pressure and temperature dependence of the tracer diffusion coefficients is given by the following Arrhenius form:

$$D = D_0 exp\left(-\frac{Q + (P-1)\cdot \Delta V}{RT}\right) \tag{3}$$

where D_0 is the pre-exponential factor (usually given in cm^2/s), Q is the activation energy, P is pressure, ΔV is the activation volume, R is the gas constant and T is the absolute temperature. Within the GDIFF programs, D and D_0 are calculated in m^2/s and the numerator, denominator within the exponent of eq. (3) are calculated in kj/mol. An example of the calculation of tracer diffusion coefficients is shown in Fig. 1.

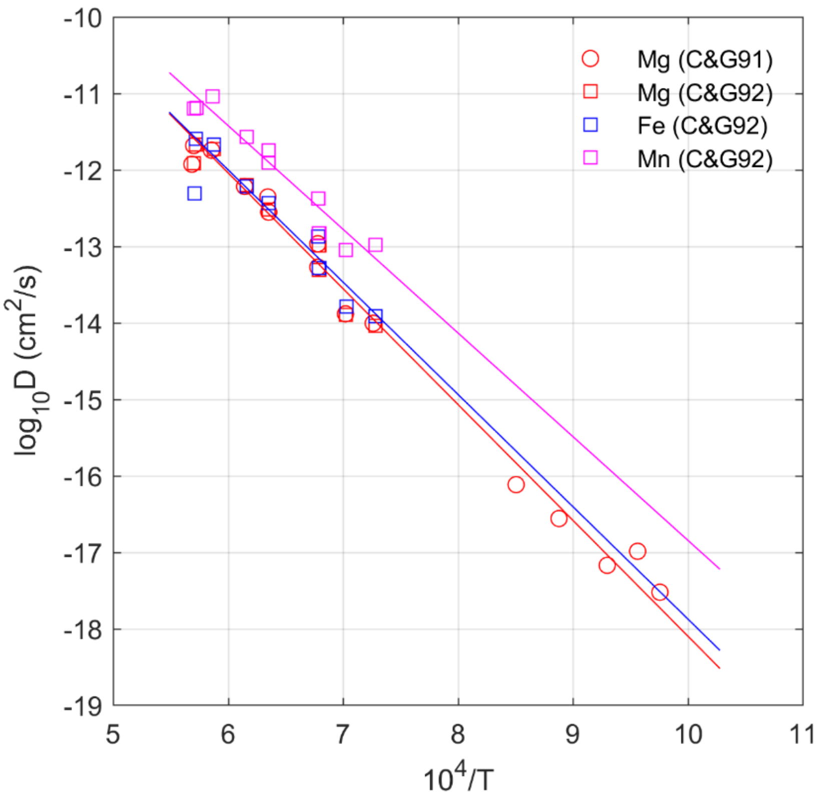


Fig. 1. Calculation of diffusion coefficients using the GDIFF program. The data come from Fig. 6a in Chakraborty and Ganguly (1992). Abbreviations: CG91: Chakraborty and Ganguly, (1991); CG92: Chakraborty and Ganguly, (1992).

Regarding the pressure dependence, the original values for the activation volume given by Chakraborty and Ganguly, (1992) do not seem to reproduce exactly their figure (Fig. 6b in Chakraborty and Ganguly, 1992). Nevertheless, the discrepancy is very small and within the data uncertainty that is given by the authors (Fig. 2). Therefore, GDIFF uses the original values that were suggested by the authors.

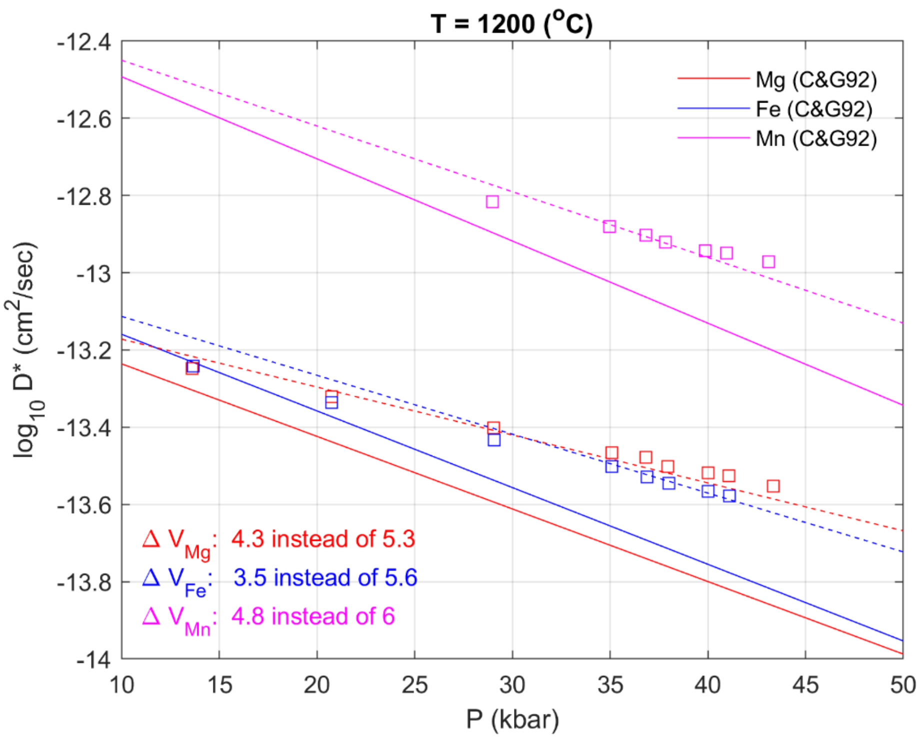


Fig. 2. Calculation of diffusion coefficients as a function of pressure using the GDIFF program. Solid lines indicate the diffusivities using the parameters that are given in the original paper by Chakraborty and Ganguly, (1992). Dashed lines indicate better fits through the data. The data come from Fig. 6b in Chakraborty and Ganguly (1992). Abbreviations: CG92: Chakraborty and Ganguly, (1992).

Numerical Solution of the Diffusion Equations

Solution procedure in GDIFF

In order to solve eq. (1), we can multiply it by x^{n-1} from both sides to avoid division by zero when $x = 0$ (e.g., in spherical coordinates). We thus have:

$$x^{n-1}\frac{\partial C_i}{\partial t} = \frac{\partial}{\partial x}\left(x^{n-1}D_{ij}\frac{\partial C_j}{\partial x}\right) \qquad (4)$$

Equations (4) are discretized following an implicit, conservative finite-difference scheme as follows:

$$x_m^{n-1}\frac{C_{i,m}^k - C_{i,m}^{k-1}}{\Delta t} = \sum_{j=1}^{3}\frac{1}{\Delta x^2}\left(x_{m+0.5}^{n-1}D_{ij,m+0.5}\left(C_{j,m+1}^k - C_{j,m}^k\right) - x_{m-0.5}^{n-1}D_{ij,m-0.5}\left(C_{j,m}^k - C_{j,m-1}^k\right)\right) \qquad (5)$$

where we have assumed that the grid is regularly spaced. The subscripts m indicate the position in the numerical grid and the superscripts k indicate the number of the timestep. The previous equation is re-arranged and the coefficients of the concentrations are assembled in a matrix of the form:

$$\boldsymbol{Ax} = \boldsymbol{b} \qquad (6)$$

where $\boldsymbol{x}$ is the vector of the unknowns, $\boldsymbol{A}$ contains the elements of the discretization matrix and $\boldsymbol{b}$ is the right-hand-side vector. In particular, the matrix is organized in the following block form:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}\begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix} \qquad (7)$$

where A_{ii} are the blocks containing the diagonal coefficients and $A_{ij}\big|_{i\neq j}$ are the blocks responsible for the coupling of the various components. Consequently, the subscripts 1,2,3 in the vector of unknowns $\boldsymbol{x_i}$ and in the right-hand-side vector $\boldsymbol{b_i}$ refer to the independent components.

It is noted that having a variable concentration generally leads to diffusivities that are variable in space, and require the evaluation of the, yet unknown, concentration because of eq. (2). For this reason, we preform direct (Picard) iterations on the solution of eq. (7) that leads to the solution of the following system of equations.

$$A(\boldsymbol{x}^{m-1})\boldsymbol{x}^{\boldsymbol{m}} = \boldsymbol{b} \qquad (8)$$

where the superscript m denotes the iteration number. The previous system is solved iteratively using MATLAB's backslash solver until the following infinity norm becomes:

$$\|\boldsymbol{x}^{m-1} - \boldsymbol{x}^{m}\|_\infty \leq 10^{-5} \qquad (9)$$

Currently the GDIFF program considers two kinds of boundary conditions. For the right side of the model domain the concentrations must be given as fixed values (Dirichlet boundary conditions). For the left side of the model domain the user can chose a Dirichlet or a Neumann (no-flux) boundary condition (NBC parameter 0 or 1). The latter is advantageous if one considers spherical coordinates. ***Thus, the modelled profile must always be oriented so that the outer edge of the crystal is towards the right side***.

Performance assessment

GDIFF utilizes a vectorized matrix assembly in order to solve the system of equations. This has a dramatic increase of performance in MATLAB (e.g. Dabrowski et al., 2008 and references therein). For example, a problem using planar geometry was solved for variable numerical grid resolution (up to 15,000 gridpoints leading to 45,000 degrees of freedom). To solve the problem, 10 time iterations were performed and for each time iteration 2-4 non-linear iterations were executed until the residuals converge (eq. 9). The same problem was solved following two approaches: in the first approach, the matrix assembly was performed using loops, while in the second approach, the matrix assembly was performed in a vectorized manner. Figure 3a shows that with increasing problem size, the time needed to solve the problem, with a loop-assembly, increases dramatically. For the problem of 15,000 gridpoints, more than 1,000 seconds were needed to complete the solution. On the contrary, using the vectorized assembly approach, the time needed to complete the solution was 3.8 seconds (Fig. 3b). The last result shows that for numerical results with high resolution, using a GDIFF offers a performance advantage.

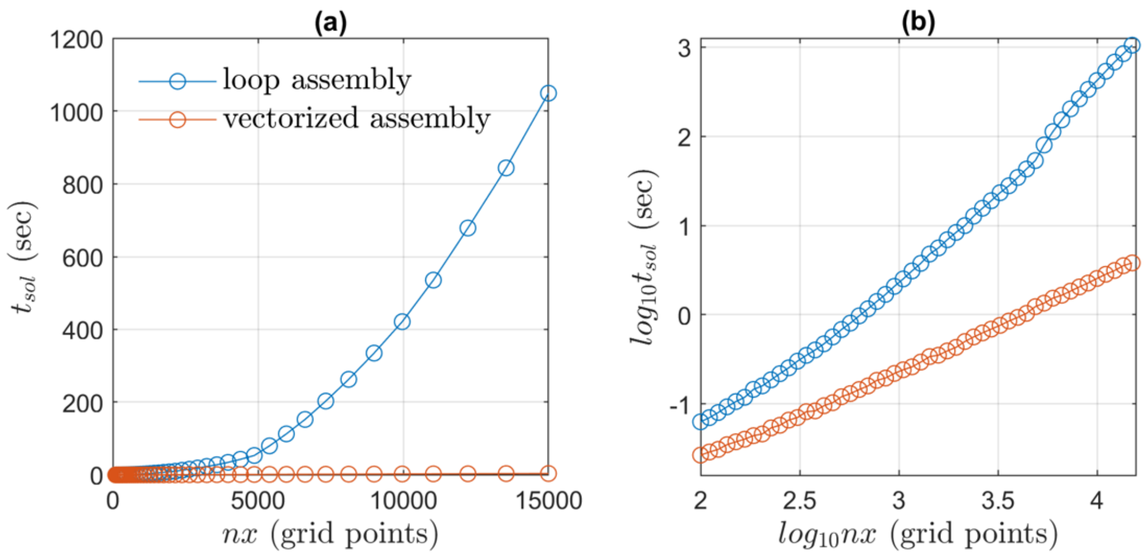
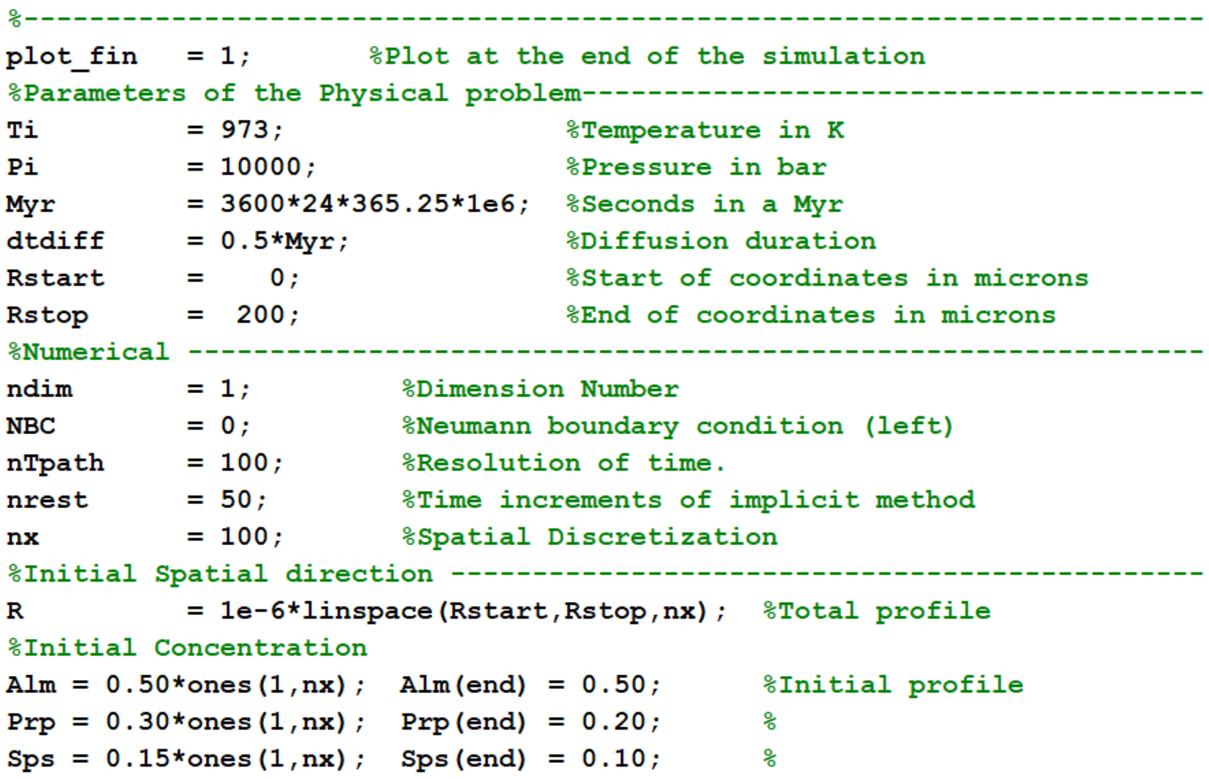


Fig. 3(a) Time needed in order to solve a given problem as a function of problem size (grid points). (b) As in (a) but using logarithmic values.

Worked Examples

Calculations at constant P-T conditions

The main routine of GDIFF is diffusion_grt_v8_VECT.m that takes an initial set of concentration arrays as input and produces the result of diffusion as output. To use this code, two additional routines were created to help the user in the model configuration. In the first routine, GDIFF_time.m, the user specifies an initial concentration profile, a pressure and a temperature value, and the duration of the diffusion process. A snapshot of the inputs is given in Fig. 4 below.



```
%-----------------------------------------------------------------------
plot_fin   = 1;       %Plot at the end of the simulation
%Parameters of the Physical problem-------------------------------------
Ti         = 973;                 %Temperature in K
Pi         = 10000;               %Pressure in bar
Myr        = 3600*24*365.25*1e6;  %Seconds in a Myr
dtdiff     = 0.5*Myr;             %Diffusion duration
Rstart     =    0;                %Start of coordinates in microns
Rstop      =  200;                %End of coordinates in microns
%Numerical -------------------------------------------------------------
ndim       = 1;         %Dimension Number
NBC        = 0;         %Neumann boundary condition (left)
nTpath     = 100;       %Resolution of time.
nrest      = 50;        %Time increments of implicit method
nx         = 100;       %Spatial Discretization
%Initial Spatial direction ---------------------------------------------
R          = 1e-6*linspace(Rstart,Rstop,nx);  %Total profile
%Initial Concentration
Alm = 0.50*ones(1,nx);  Alm(end)  = 0.50;     %Initial profile
Prp = 0.30*ones(1,nx);  Prp(end)  = 0.20;     %
Sps = 0.15*ones(1,nx);  Sps(end)  = 0.10;     %
```

Fig. 4 Code snapshot from GDIFF_time.m. The code inputs and the numerical parameters are given in consequential order. The initial profile of this problem is a constant composition of a garnet with $Alm_{50}Prp_{30}Sps_{15}Grs_{5}$ with boundary conditions at the edge $Alm_{50}Prp_{20}Sps_{10}Grs_{10}$.

Note that for the previous calculation the parameter nrest (time resolution) must be set to an integer >1. This is because, and despite the fact that the solver is stable for large timesteps, small time increments are needed to ensure that the solution converges. The reason for that is the fact that diffusivity is composition dependent, hence, small-time increments are needed for the accurate time integration. In order to check for convergence, the user can increase the number nrest until the final solution remains unchanged[2]. The solution of the previous problem is given in Fig. 5.

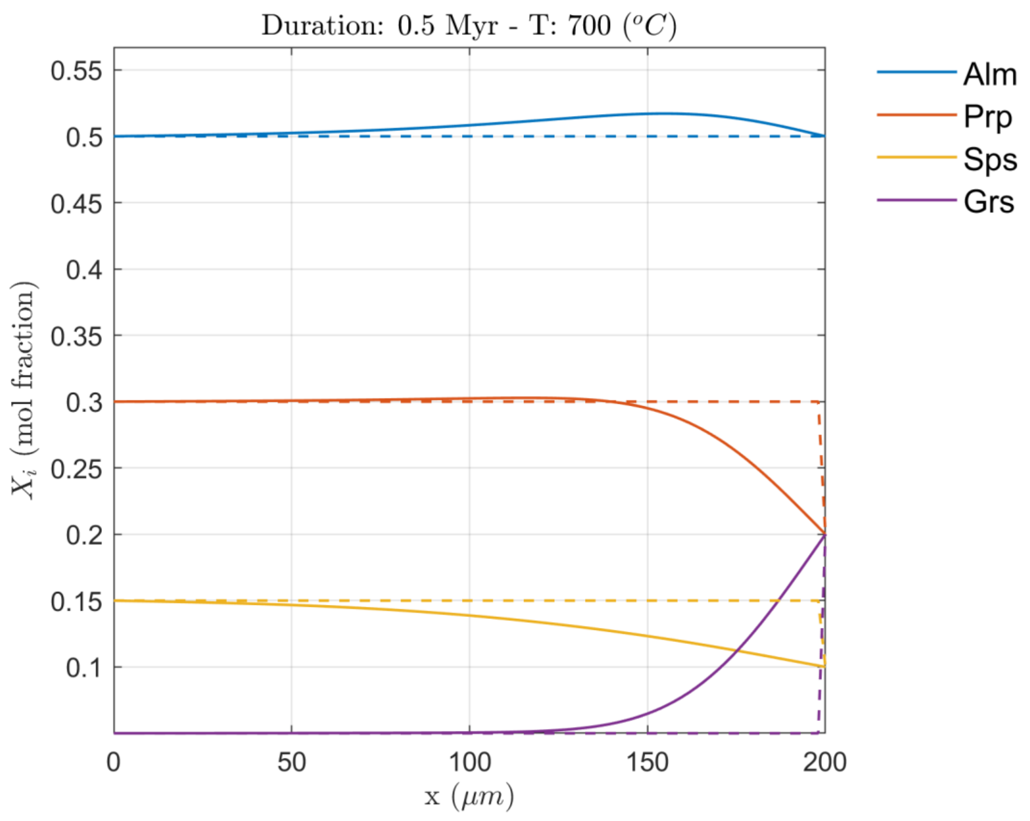
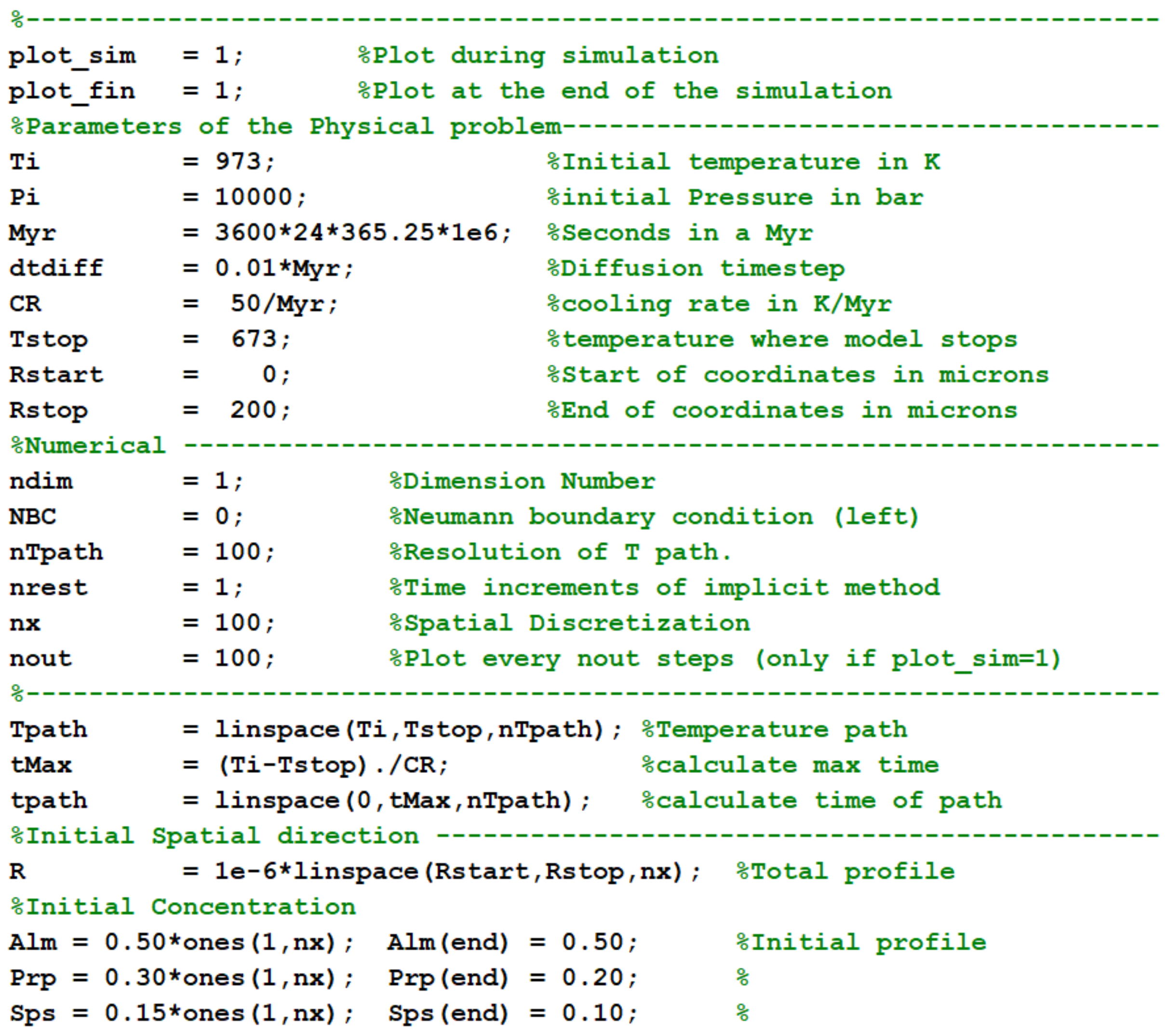


Fig. 5. Example of model results of code GDIFF_time.m. See text for further details.

[2] Note that increasing this number will make the code slower, so the user is advised to increase this number gradually.

Calculations using simple cooling histories

In order to perform simulations involving complex pressure and temperature (P-T) histories, one must integrate the time diffusion in time while updating the diffusion coefficients. A simple routine was created (GDIFF_cooling.m) in order to perform such operations. The routine assumes a simple cooling history where pressure and temperature are given as arrays. To simulate a cooling history the user must specify a cooling rate in K/Myr. An example of the inputs is shown in Fig. 6.



```
%------------------------------------------------------------------------
plot_sim   = 1;        %Plot during simulation
plot_fin   = 1;        %Plot at the end of the simulation
%Parameters of the Physical problem--------------------------------------
Ti         = 973;                   %Initial temperature in K
Pi         = 10000;                 %initial Pressure in bar
Myr        = 3600*24*365.25*1e6;    %Seconds in a Myr
dtdiff     = 0.01*Myr;              %Diffusion timestep
CR         =  50/Myr;               %cooling rate in K/Myr
Tstop      =  673;                  %temperature where model stops
Rstart     =    0;                  %Start of coordinates in microns
Rstop      =  200;                  %End of coordinates in microns
%Numerical --------------------------------------------------------------
ndim       = 1;          %Dimension Number
NBC        = 0;          %Neumann boundary condition (left)
nTpath     = 100;        %Resolution of T path.
nrest      = 1;          %Time increments of implicit method
nx         = 100;        %Spatial Discretization
nout       = 100;        %Plot every nout steps (only if plot_sim=1)
%------------------------------------------------------------------------
Tpath      = linspace(Ti,Tstop,nTpath); %Temperature path
tMax       = (Ti-Tstop)./CR;            %calculate max time
tpath      = linspace(0,tMax,nTpath);   %calculate time of path
%Initial Spatial direction ----------------------------------------------
R          = 1e-6*linspace(Rstart,Rstop,nx);  %Total profile
%Initial Concentration
Alm = 0.50*ones(1,nx);  Alm(end) = 0.50;      %Initial profile
Prp = 0.30*ones(1,nx);  Prp(end) = 0.20;      %
Sps = 0.15*ones(1,nx);  Sps(end) = 0.10;      %
```

Fig. 6. Code snapshot from GDIFF_cooling.m. The code inputs and the numerical parameters are given in consequential order. For this case, the cooling rate is set to 50 degrees per million years (parameter CR). The initial profile of this problem is a constant composition of a garnet with $Alm_{50}Prp_{30}Sps_{15}Grs_{5}$ with boundary conditions at the edge $Alm_{50}Prp_{20}Sps_{10}Grs_{10}$.

The parameters of GDIFF_cooling.m are similar like in the case of GDIFF_time.m shown previously. However, there are two major differences. Firstly, a cooling rate is needed (parameter CR). Furthermore, the routine will stop only if temperature goes below a threshold value (Tstop). Like previously, for the accurate time integration, the timesteps should not be very large. To improve time integration the user can decrease the timestep (dtdiff) gradually until the result remains essentially the same. For this kind of calculation, the parameter nrest can be as low as 1 (see Fig. 6). The solution of the previous problem is given in Fig. 7.

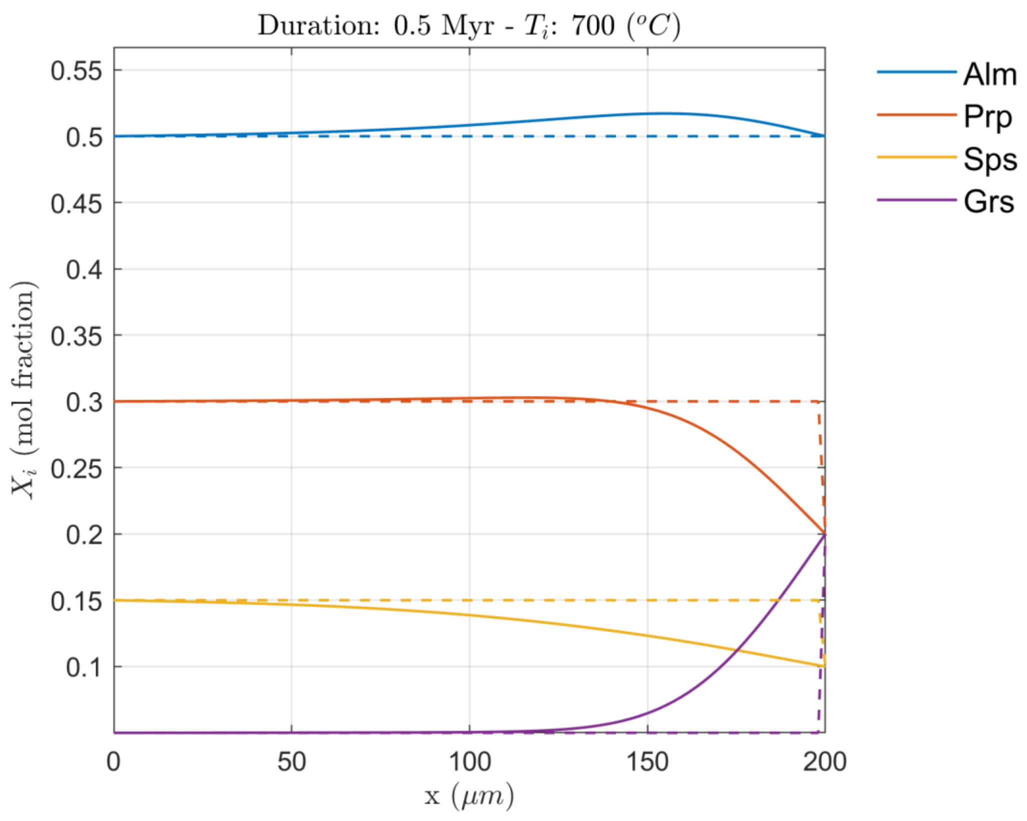


Fig. 7. Example of model results of code GDIFF_cooling.m. See text for further details.

Using GDIFF in OCTAVE

Although GDIFF was written originally in MATLAB, compatibility with octave has been checked and the codes work normally. The only issues are related to labeling during plotting. In that case, the option "'intepreter','latex'" and he symbol "$" must be deleted from the axis lables/titles.

References

Burg, J.-P., & Moulas, E. (2022). Cooling-rate constraints from metapelites across two inverted metamorphic sequences of the Alpine-Himalayan belt; evidence for viscous heating. *Journal of Structural Geology*, *156*, 104536. https://doi.org/10.1016/j.jsg.2022.104536

Chakraborty, S., & Ganguly, J. (1991). Compositional Zoning and Cation Diffusion in Garnets. In J. Ganguly (Ed.), *Diffusion, Atomic Ordering, and Mass Transport: Selected Topics in Geochemistry* (pp. 120–175). Springer US. https://doi.org/10.1007/978-1-4613-9019-0_4

Chakraborty, S., & Ganguly, J. (1992). Cation diffusion in aluminosilicate garnets: Experimental determination in spessartine-almandine diffusion couples, evaluation of effective binary diffusion coefficients, and applications. *Contributions to Mineralogy and Petrology*, *111*(1), 74–86. https://doi.org/10.1007/BF00296579

Cheng, H., Bloch, E. M., Moulas, E., & Vervoort, J. D. (2020). Reconciliation of discrepant U–Pb, Lu–Hf, Sm–Nd, Ar–Ar and U–Th/He dates in an amphibolite from the Cathaysia Block in Southern China. *Contributions to Mineralogy and Petrology*, *175*(1), 4. https://doi.org/10.1007/s00410-019-1644-9

Dabrowski, M., Krotkiewski, M., & Schmid, D. W. (2008). MILAMIN: MATLAB-based finite element method solver for large problems. *Geochemistry, Geophysics, Geosystems*, *9*(4). https://doi.org/10.1029/2007GC001719

Lasaga, A. C. (1979). Multicomponent exchange and diffusion in silicates. *Geochimica et Cosmochimica Acta*, *43*(4), 455–469. https://doi.org/10.1016/0016-7037(79)90158-3