KADMOS: a Finite Element code for the calculation of apparent K-Ar ages in minerals

Evangelos Moulas[1*] & Mark T Brandon[2*]

[1*]Institute of Geosciences & Mainz Institute of Multiscale Modeling (M^3ODEL), Johannes-Gutenberg University Mainz, Germany
(evmoulas@uni-mainz.de)

[2*]Department of Earth and Planetary Sciences, Yale University, United States of America
(mark.brandon@yale.edu)

This documentation describes the main features of KADMOS (**K**-**A**r **D**iffusion **MO**delling **S**oftware). KADMOS is set of MATLAB routines that can be used to calculate the apparent ^{40}K-^{40}Ar (and the associated ^{40}Ar-^{39}Ar) ages as a function of a sample's thermal history. KADMOS is written in MATLAB language and utilizes the Finite-Element Method (FEM) with grid refinement. The advantage of KADMOS is that it has been optimized for accuracy, performance and for maximum flexibility with respect to the modelled scenarios. KADMOS can be used to evaluate the apparent ages of various crystals and various geometries (planar, cylindrical & spherical) simultaneously.

Despite the fact that the method that is followed here is general, we chose to focus on a particular case (^{40}K-^{40}Ar dating) and show specific examples. The software and the present documentation are provided free of charge[1]. At this point, we have also tested for compatibility with OCTAVE.

Current Version: 1 (06-Oct-2022) – doi: 10.5281/zenodo.7358138

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

[1] Creative Commons Attribution 4.0 International

Contents

Introduction to the K-Ar and Ar-Ar methods

The purpose of the present document is not to provide a detailed review of the Ar-Ar method but rather to present the major aspects of the KADMOS software. For a more comprehensive introduction to the topic, the reader is guided to the available literature (e.g. Braun and others, 2006; McDougall and Harrison, 1999; Reiners and others, 2017).

Governing Equations

The K-Ar and Ar-Ar methods are based on the radiogenic production of ^{40}Ar in K-bearing minerals. In a closed system, the decay of ^{40}K to ^{40}Ar leads to the increase of the concentration of ^{40}Ar with time. The increase of radiogenic argon is thus given by:

$$\frac{dAr}{dt} = \lambda_{Ar} K \tag{1}$$

where the left-superscripts of ^{40}Ar and ^{40}K have been omitted and λ_{Ar} is the decay constant of ^{40}K to ^{40}Ar (0.581·10^{-10}yrs). K represents the currently available amount of ^{40}K. The latter is given by:

$$K = K_0 e^{-\lambda_T t} \tag{2}$$

where K_0 is the initial amount of ^{40}K, λ_T is the total decay constant of ^{40}K (5.543·10^{-10}yrs) and t is time[2] (McDougall and Harrison, 1999, p. 17). The total decay constant considers also the decay of ^{40}K to ^{40}Ca. Replacing eq. (2) into eq. (1) yields an expression of argon production:

$$\frac{dAr}{dt} = \lambda_{Ar} K_0 e^{-\lambda_T t} \tag{3}$$

Equation (3) is a separable differential equation and can be solved with respect to Ar:

$$Ar = Ar_0 + \frac{\lambda_{Ar}}{\lambda_T} K_0 \left(1 - e^{-\lambda_T t}\right) \tag{4}$$

where Ar_0 is the initial amount of ^{40}Ar that is commonly assumed to be zero. The factor $\frac{\lambda_{Ar}}{\lambda_T}$ is the normalized constant for the decay of ^{40}K to ^{40}Ar can be expressed as:

$$f_{Ar} \equiv \frac{\lambda_{Ar}}{\lambda_T} \equiv \frac{1}{k_e} = \frac{1}{9.54} \tag{5}$$

Equation (4) can also be recast as a function of the current amount of ^{40}K, yielding:

[2] if the values of the decay constants are used as given (in years), then time should be given in years as well

$$Ar = Ar_0 + \frac{\lambda_{Ar}}{\lambda_T} K\left(e^{\lambda_T t} - 1\right) \tag{6}$$

With the use of eq. (5), eqs. (4-6) can be solved with respect to time. The resulting expressions are (assuming Ar_0 equal to zero):

$$t = \frac{1}{\lambda_T} ln\left(k_e \frac{Ar}{K} + 1\right) \tag{7a}$$

$$t = -\frac{1}{\lambda_T} ln\left(1 - k_e \frac{Ar}{K_0}\right) \tag{7b}$$

Equations (7) are the main formulas that are used to calculate the apparent age of a mineral once its ^{40}Ar to ^{40}K ratio is known. When minerals produce radiogenic ^{40}Ar, concentration gradients of ^{40}Ar may develop. In that case, ^{40}Ar will diffuse following the concentration gradients and its evolution over a given domain will be described by the following diffusion-production equation (in 1-dimension):

$$\frac{\partial Ar}{\partial t} = D \frac{\partial^2 Ar}{\partial x^2} + \lambda_{Ar} K_0 e^{-\lambda_T t} \tag{8}$$

where we have assumed that the diffusion coefficient is independent of concentration and the diffusion takes place at constant temperature and pressure. Division of eq. (8) by K_0 yields:

$$\frac{\partial}{\partial t}\left(\frac{Ar}{K_0}\right) = D \frac{\partial^2}{\partial x^2}\left(\frac{Ar}{K_0}\right) + \lambda_{Ar} e^{-\lambda_T t} \tag{9}$$

Equation (9) is the main equation that is solved in KADMOS. However, different variants and approximations of this form exist in the literature. For example, one could divide the previous equation by λ_{Ar}, and obtain:

$$\frac{\partial}{\partial t}\left(\frac{Ar}{\lambda_{Ar} K_0}\right) = D \frac{\partial^2}{\partial x^2}\left(\frac{Ar}{\lambda_{Ar} K_0}\right) + e^{-\lambda_T t} \tag{10}$$

which, for sufficiently small λ_T, makes the last term of eq. (10) equal to unity. In addition, for relatively small timescales and for negligible accumulation of ^{40}Ar, equation (7b) can be approximated by the apparent age (τ):

$$t \approx \tau = \frac{k_e}{\lambda_T} \frac{Ar}{K_0} = \frac{Ar}{\lambda_{Ar} K_0} \tag{11}$$

In the latter case, eq. (10) can be approximated by the following form (Wheeler, 1996, p. 922):

$$\frac{\partial \tau}{\partial t} = D \frac{\partial^2 \tau}{\partial x^2} + 1 \tag{12}$$

Numerical Solution of the Production-Diffusion Equations

Main Routine of KADMOS

In the current version of KADMOS, a main routine is provided (“*KADMOS_Path.m*”). *KADMOS_Path* can be used to calculate the radiogenic production of ^{40}Ar from ^{40}K in minerals as a function of their P-T history. In this routine, several P-T-t histories can be simulated simultaneously for variable crystal sizes. It is also possible to simulate the apparent ^{40}K-^{40}Ar ages that result from arbitrary P-T-t paths. A typical result of the *KADMOS_Path* is shown in Fig. 1.

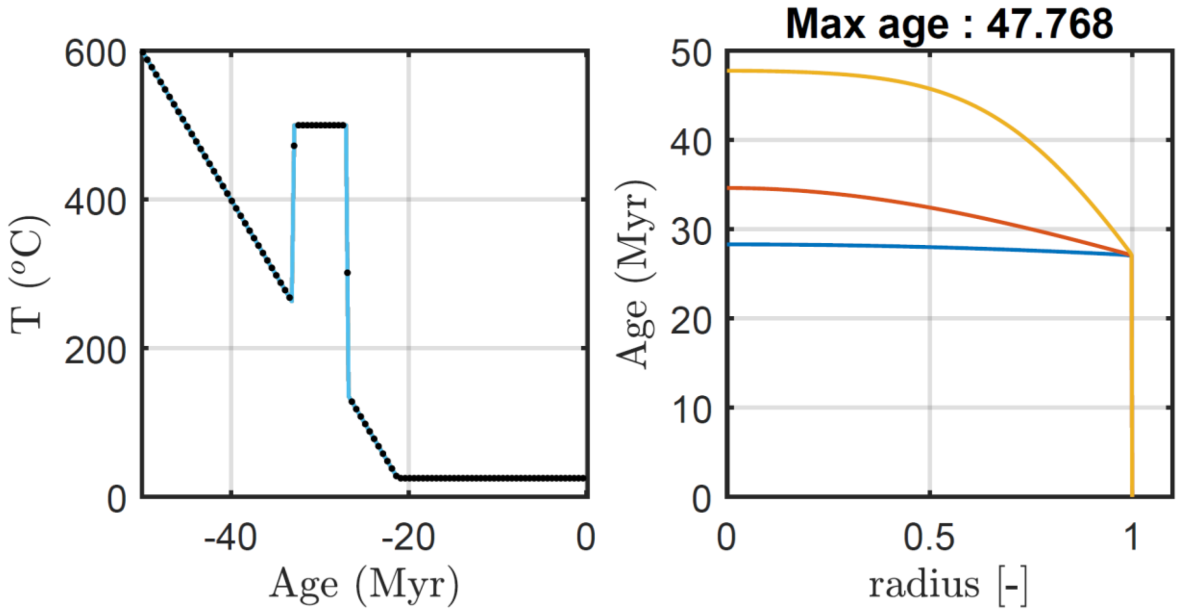


Fig. 1. Results from the KADMOS_Path program. Left diagram: Sample’s thermal history (dots indicate current position on path while plotting). Three spherical muscovite crystals are cooled from 600°C to 25°C with a constant cooling rate of 20°C/Myr. A thermal pulse of 500°C is added around 30Myr. Pressure remains constant at 2,000 MPa. Right diagram: The apparent ages of the crystals are plotted as a function of their radius (normalized by the size of each crystal). For this calculation three crystal sizes have been considered (100,200 and 500 µm).

The results shown in Fig. 1 demonstrate the effect of the thermal pulse on the apparent ages of the three crystal classes. It is expected that the smaller crystals (blue line in Fig. 1 – right diagram) will be largely affected by argon diffusion during the thermal pulse at ca 30 Myr. In contrast, the apparent age at the core of the largest crystal is almost unaffected (yellow line in Fig. 1 - right diagram).

Main Equations

KADMOS utilizes the Galerkin, Finite-Element-Method (FEM) in order to solve eq. (9) in 1-dimension (1-d). In particular, the following form of the equation is solved:

$$x^{n-1}\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(x^{n-1}D\frac{\partial C}{\partial x}\right) + x^{n-1}S \tag{13}$$

where C represents the Ar/K_0 ratio, S represents a source term ($S = \lambda_{Ar}e^{-\lambda_T t}$), and n is an integer $n \in [1,2,3]$. The integer can be chosen so that different geometries can be modelled. If the geometry is spherical $n = 3$, if the geometry is cylindrical $n = 2$, and if it is planar $n = 1$. For the planar case the "radius" corresponds to the half-width of the planar geometry (e.g. Lister and Baldwin, 1996, p. 86).

FEM discretization and solution

For the solution of equation (13) in 1-dimension we utilize linear shape functions (N). The linear shape functions are given by:

$$N_i(x) = 1 - \frac{x - x_i}{x_{i+1} - x_i} \tag{14a}$$

$$N_{i+1}(x) = \frac{x - x_i}{x_{i+1} - x_i} \tag{14b}$$

In its discretized general form, the following local system of equations (i.e. for each element) yields:

$$\left[\frac{M}{\Delta t} + \theta K\right]\begin{Bmatrix} C_i^j \\ C_{i+1}^j \end{Bmatrix} = \left[\frac{M}{\Delta t} - (1-\theta)K\right]\begin{Bmatrix} C_i^{j-1} \\ C_{i+1}^{j-1} \end{Bmatrix} + \theta\{F^j\} + (1-\theta)\{F^{j-1}\} \tag{15}$$

where Δt is the time increment, C_i^j is the main unknown (degree of freedom) at the point i and at the time instance j, M is the "mass" matrix, K is the "stiffness" matrix and F is the "force" vector. The superscript j indicates that K and F are evaluated at a given time instance. M,K and F are given below:

$$M = \int_{x_i}^{x_i+1} x^{n-1}\begin{bmatrix} N_iN_i & N_iN_{i+1} \\ N_{i+1}N_i & N_{i+1}N_{i+1} \end{bmatrix} dx \tag{16a}$$

$$K^j = \int_{x_i}^{x_i+1} D^j \cdot x^{n-1}\begin{bmatrix} \frac{\partial N_i}{\partial x}\frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial x}\frac{\partial N_{i+1}}{\partial x} \\ \frac{\partial N_{i+1}}{\partial x}\frac{\partial N_i}{\partial x} & \frac{\partial N_{i+1}}{\partial x}\frac{\partial N_{i+1}}{\partial x} \end{bmatrix} dx \tag{16b}$$

$$F^j = \int_{x_i}^{x_i+1} x^{n-1} \begin{Bmatrix} N_i \\ N_{i+1} \end{Bmatrix} S^j \, dx \tag{16c}$$

The advantage of the previous form is that the integrals in eqs. (16) can be evaluated analytically. In that case KADMOS uses the analytical expressions to evaluate eqs. 16. For the solution of the global system of equations all the matrices are assembled in the following form:

$$\boldsymbol{KM}\{C_i^j\} = \boldsymbol{R} \tag{17}$$

where $\boldsymbol{KM}$ is the global left-hand side matrix and $\boldsymbol{R}$ is the global right-hand-side vector. $\boldsymbol{KM}$ and $\boldsymbol{R}$ are assembled from the local KM and R matrices following eq. (15), that is:

$$\begin{gathered} KM = \left[\frac{M}{\Delta t} + \theta K\right] \\ R = \left[\frac{M}{\Delta t} - (1-\theta)K\right] \begin{Bmatrix} C_i^{j-1} \\ C_{i+1}^{j-1} \end{Bmatrix} + \theta\{F^j\} + (1-\theta)\{F^{j-1}\} \end{gathered} \tag{18}$$

At this point we note that another advantage of the previous formulation is that the matrix $\boldsymbol{KM}$ is symmetric. For boundary conditions we consider a no-flux boundary condition in the left side of the modelled domain ($x = 0$) and $C = 0$ at the right side of the domain ($x = L$) which represents the edge of the crystal. During the application of the boundary conditions, the symmetry of the $\boldsymbol{KM}$ matrix is enforced (c.f. Dabrowski and others, 2008; Räss and others, 2017).

The time integration of eq. (15) can be performed by the fully implicit ($\theta = 1$) or the semi-implicit ($\theta = 0.5$) method as it is typically done in diffusion applications (e.g. Simpson, 2017). In the current version of KADMOS programs, we have found that a combination of the fully implicit method ($\theta = 1$) together with the separation of production-diffusion terms in eq. (15) yields more accurate and stable results. This way of solution resembles the separation of terms as it was done in the finite-difference approach of Lister and Baldwin (1996). More specifically, the concentration is first updated according to the following rule (for $\theta = 1$):

$$C_i^j = C_i^j + \Delta t\left[\theta S^{j+1} + (1-\theta)S^j\right] \tag{19}$$

and the source term (S) is given by:

$$S^j = \frac{1}{k_e}\left[\lambda_T \cdot exp\left(-\lambda_T t^j\right)\right] \tag{20}$$

After updating the concentrations, the boundary condition (at $x = L$) is imposed and the solution of the diffusion part is obtained by solving eqs. (15) with $F = 0$ and $\theta = 1$. When pressure and temperature (P-T) change with time, a small timestep is chosen, and at each time instance, P and T are updated using linear interpolation on a pre-computed P-T-t path. At the given P-T conditions, the diffusion coefficient of Ar is evaluated and the production/diffusion equations are solved. The last steps are repeated until the end of the simulation.

The variable grid

The particular form of the diffusion/production equation (eq. 10) results to solutions that have very large spatial derivatives at the edge of the crystal (Fig. 2). Especially in the case where the diffusion is limited, and the solution is dominated by the production term, the sharp compositional gradients that develop at the edge of the crystals may render the results inaccurate near the crystal edge.

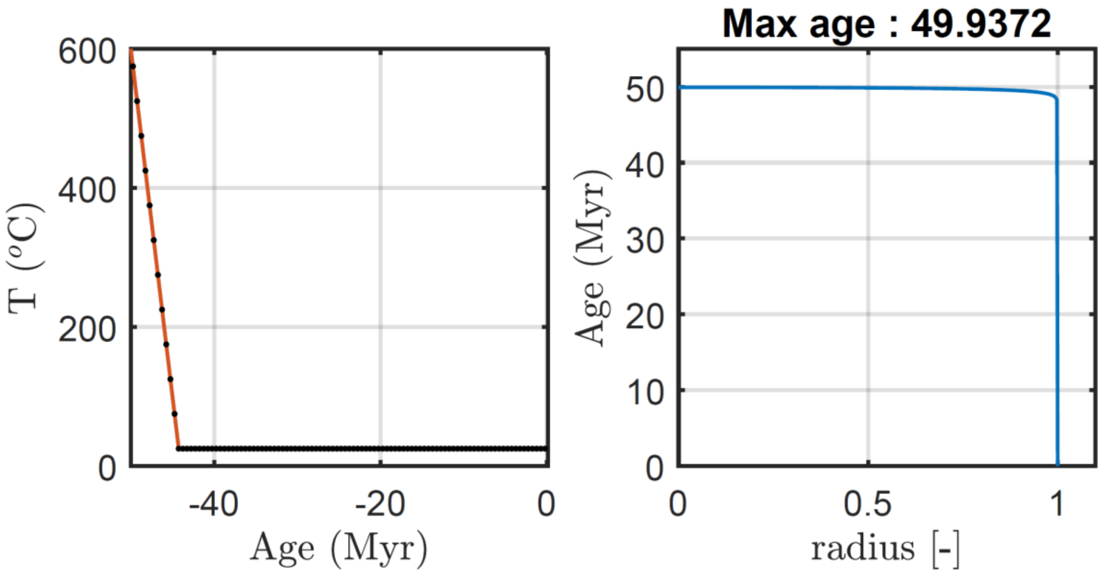


Fig. 2. Results from the *KADMOS_Path* program. Left diagram: Thermal history of a 500µm sized spherical muscovite crystal. The crystal is cooled from 600°C to 25°C with a constant cooling rate of 100°C/Myr. Right diagram: The apparent age of the crystal is plotted as a function of its radius (normalized by the size of the crystal).

Having a FEM formulation allows for the adaptation of a variable grid. The variable grid is used to calculate the solution in a very accurate way near the edge of the crystal. To construct the variable grid, we proceed as follows. First, we assume that the grid spacing (Δx) is changing by a constant factor (α) between neighboring increments. We thus have:

$$\Delta x_{i+1} = \alpha \Delta x_i \tag{21}$$

where the subscript i indicates the number of the element. In addition, we know that the sum of the spatial increments (Δx_i) is equal to the length of the domain (L). That is:

$$\sum_{i=1}^{k} \Delta x_i = L \tag{22}$$

where k is the number of the spatial increments. Equations (21) and (22) can be rearranged to produce a closed system of linear equations as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ -\alpha & 1 & 0 & 0 & \dots & 0 \\ 0 & -\alpha & 1 & 0 & \dots & 0 \\ 0 & 0 & -\alpha & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & -\alpha & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta x_4 \\ \dots \\ \Delta x_k \end{bmatrix} = \begin{bmatrix} L \\ 0 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \tag{23}$$

The solution of the linear system results to the necessary values of the spatial increments (Δx_i). Finally, we have chosen to express the grid refinement by a numerical factor (RdX)[3] that represents the ratio between the first and last spatial increment. That is:

$$\Delta x_1 = RdX \cdot \Delta x_k \tag{24}$$

As an example, an RdX parameter equal to 1, results to the generation of a regular grid while an RdX parameter equal to 10 results to the generation of a variable grid where the last element (at the edge of the crystal) is 10 times smaller than the first element (at the core of the crystal).

Generation of P-T-t paths and thermal histories

In order to allow the investigation of various P-T-t paths, several "ideal" P-T-t paths are considered. In general, every P-T-t path has an initial P-T (P0,T0) and a final P-T (Pmin,Tmin). In addition, a total duration of the calculation is assumed (tlim). Note that the temperature and pressure are not allowed to go below the values set by Pmin and Tmin. For example, if a cooling history with a constant cooling rate is defined, temperature remains at Tmin and is not allowed to decrease further. This is done so that negative absolute temperatures are avoided. The different forms of P-T paths that are generated in the *KADMOS_Path* routine are:

[3] Keywords that refer to variables in the codes are written in red.

- lin-CR: This is the simplest path that assumes a simple cooling history with a constant cooling rate (SCR: in °C per Myr). Pressure is considered equal to P0 during the whole duration of the calculation.

- asy-CR: This path assumes the asymptotic cooling of a sample following the formulation of Lasaga (Lasaga, 1983, p. 87). The pressure is considered constant in this case and it is equal to P0. An initial cooling rate is also needed in order to define the path.

- lin-P-T: This path assumes linear (i.e. with constant rate) cooling and decompression from P0-T0 to Pmin-Tmin.

- iso-T: This path assumes a constant temperature (equal to T0) and a constant pressure (equal to P0) for the whole duration of the simulation.

- loaded: This option allows the loading of a custom path.

As an example, the P-T-t path can be generated with the following function as follows:

```
[Pres_A,TempK_A,time_A] = make_PTtpath(‘lin-CR’,T0, SCR,P0,Pmin,Tmin,tlim);
```

The previous command generates a cooling path with a constant cooling rate (defined as positive). Once the temperature reaches the Tmin value, it will remain at Tmin until the simulation completes at tlim. Pressure will remain constant and equal to P0 in this case. Another example follows below:

```
[Pres_A,TempK_A,time_A] = make_PTtpath(‘asy-CR’,T0, SCR,P0,Pmin,Tmin,tlim);
```

The previous command generates an asymptotic cooling path that has an initial cooling rate equal to SCR. If the temperature reaches the value of Tmin, it will stay at that value; if the temperature does not reach the value of Tmin, it will keep dropping asymptotically until time reaches the value of tlim. Pressure will also remain constant and equal to P0 in this case.

The resulting arrays (Pres_A, TempK_A, time_A) are the pressure (in MPa), the temperature (in K) and the time (in Myr) and are given as rows.

Main Input Keywords

With the function make_PTtpath (included in the file “Inputs_Path.m”) it is possible to set up a great number of P-T-t histories and simulate the resulting apparent ages. However, a few more things need to be configured with the use of a few keywords in order to start the simulation. These keywords can be modified in the respective file “Inputs_Path.m”. These are:

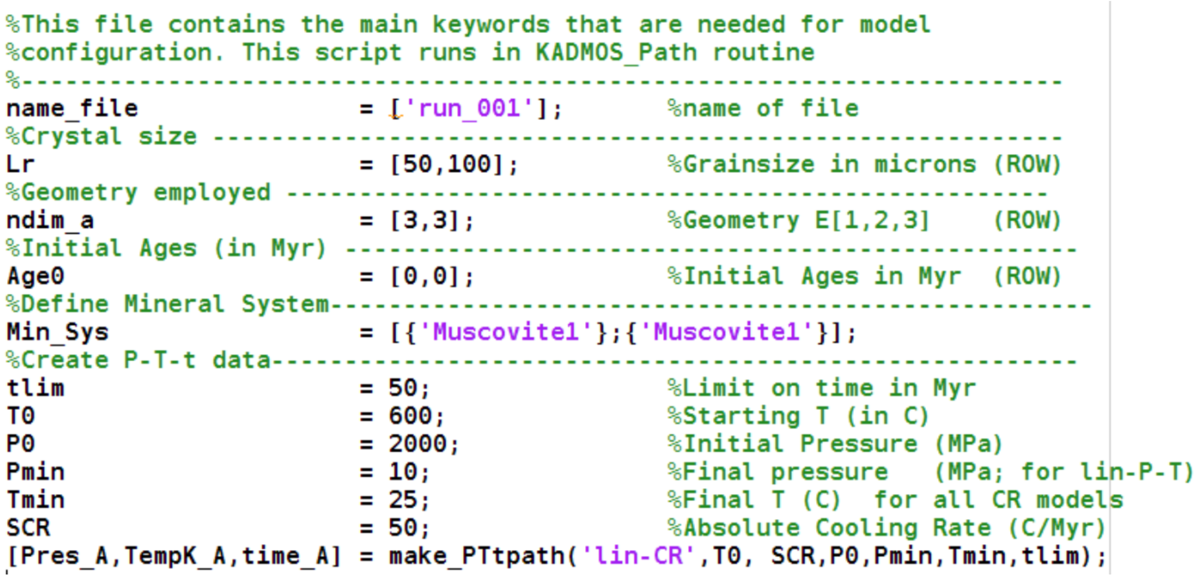
- name_file: The name of the file (e.g. [‘example01’])[4].
- Lr: The classes of the different grain sizes (e.g. [100,200,500]). The values correspond to the length (in micrometers) of the characteristic radius of the sphere, cylinder, or the half-thickness of the plane.
- ndim_a: An index that is used to select the geometry (spherical=3, cylindrical=2 or planar=1; e.g. [3,3,3,3] represents four spherical crystals).
- Min_Sys: As an example, this value is set to {'Muscovite1'} using the diffusion coefficients of argon in muscovite from Harrison and others (2009). For more details on the diffusion data see the following section “Diffusion Parameters”
- Age0: This is an array that contains the initial age of the modelled crystals. If this value is not set to zero for each grain-size class, then the specific grain sizes will have an initial (homogeneous) age (e.g. setting Age0 = [1,5,10]; will lead to three different size classes that have 1, 5 and 10Myr as initial apparent ages at the beginning of their modelled thermal history).

Note that phi, Lr, ndim_a, Age0 and Min_Sys must have the same size since their corresponding values apply for each size class individually.

A screenshot with a random set of inputs is given in Fig. 3 below. For example, the saved file will be named *run_001*, two size classes will be modelled (50 & 100 μm; Lr=(50,100]), the crystals will be spherical (ndim_a = [3, 3]), and their initial age will be zero (Age0 = [0, 0]). The diffusion parameters are taken from Harrison and others (2009) by selecting {‘Muscovite1’} in the keyword **Min_Sys** (more details are given in the following section). The simulation will run for 50 (modelled) million years (tlim = 50) starting at 600°C and 2,000MPa as starting conditions (T0 =600 ; P0 = 2000). A cooling thermal history (with constant cooling rate) is

[4] Examples of inputs are given in blue color

chosen with an initial cooling rate of 50°C per Myr (SCR = 50). The cooling path is generated via the use of the make_PTtpath function (see previous section for details).



```
%This file contains the main keywords that are needed for model
%configuration. This script runs in KADMOS_Path routine
%-----------------------------------------------------------------------
name_file               = ['run_001'];       %name of file
%Crystal size ----------------------------------------------------------
Lr                      = [50,100];          %Grainsize in microns (ROW)
%Geometry employed ----------------------------------------------------
ndim_a                  = [3,3];             %Geometry E[1,2,3]    (ROW)
%Initial Ages (in Myr) -------------------------------------------------
Age0                    = [0,0];             %Initial Ages in Myr  (ROW)
%Define Mineral System----------------------------------------------------
Min_Sys                 = [{'Muscovite1'};{'Muscovite1'}];
%Create P-T-t data-------------------------------------------------------
tlim                    = 50;                %Limit on time in Myr
T0                      = 600;               %Starting T (in C)
P0                      = 2000;              %Initial Pressure (MPa)
Pmin                    = 10;                %Final pressure   (MPa; for lin-P-T)
Tmin                    = 25;                %Final T (C)  for all CR models
SCR                     = 50;                %Absolute Cooling Rate (C/Myr)
[Pres_A,TempK_A,time_A] = make_PTtpath('lin-CR',T0, SCR,P0,Pmin,Tmin,tlim);
```

Fig. 3. A screenshot example of the keywords in the file Input_Path.m.

Diffusion Parameters

In the current version of KADMOS the diffusivity of Ar is taken from various studies, the currently available choices are given in the table that follows (Table 1). Note that in some applications, the data have been augmented by including values for activation volume that were not included in the original datasets (e.g. Lister and Baldwin, 1996; Skipton and others, 2018). In addition, the reported grain size does not necessarily correspond to the actual grain size and smaller values may be used (e.g. in the presence of cracks). Thus, the reported geometry is does not always correspond to the inferred geometry of a single crystal. For example, Lister and Baldwin (1996) use a planar geometry (slab) in order to describe diffusion of muscovite while, in contrast, Harrison and others (2009) use spherical geometry to fit their diffusion data. Thus, the user is encouraged to check the literature before choosing the optimal choice of parameters.

An additional feature of KADMOS is that custom diffusion data may be used, these data can be implemented with the choice “custom” in **Min_Sys** keyword. To use custom, data a simple data file must be created (i.e. with the use of a simple text editor). The file must be named “Custom_Diff.dat”. In this file, the user can add the values of the frequency factor (in cm^2/sec) in the fist line, the value of activation energy (in kj/mol) in the second line, the activation volume (in cm^3) in the third line, and the value of the reference pressure (in MPa) in the last line. The value of the reference pressure is normally taken as 0.0001 (MPa) but larger values may also be used based on the experimental calibration.

Table 1. Table of diffusivity data included in KADMOS.

Name (case sensitive)	Mineral	Grainsize (microns)	Geometry employed	Reference
Muscovite1	muscovite	24-87	spherical	(Harrison and others, 2009), note that spherical geometry was used to fit the experiments
Muscovite2	muscovite	5.9	planar	(Robbins, 1972)[5,6]
Hornblende1	hornblende	80	spherical	(Harrison, 1982), parameters from experiments only[5,6]
Hornblende2	hornblende	80	spherical	same as above, but parameters are given as suggested in the original study of Harrison (1982)[5,6]
Biotite1	Biotite			(Grove and Harrison, 1996)[7]
Biotite2	Biotite	150	cylindrical	(Harrison and others, 1985)[5,6]
Biotite3	Phlogopite	150	cylindrical	(McDougall and Harrison, 1999)[5,6]
Biotite4	Biotite	-	-	(Grove and Harrison, 1996)[8]
Orthoclase	Orthoclase	-	-	(Foland, 1994)[9]
custom	-	-	-	-

[5] Geometry and grain size as used in Lister and Baldwin, (1996)

[6] Activation volume after Lister and Baldwin, (1996)

[7] Biotite with Xann : 0.71. Activation volume is considered zero. Skipton and others, (2018) used 14cm^3/mol

[8] Data for Cooma biotite (Xann = 0.54). Activation volume is considered zero.

[9] Activation volume is considered zero.

Other software keywords

In this section we briefly describe the remaining keywords that can be found in the main file of “KADMOS_Path.m”. Note that the program is optimized for accuracy and performance and therefore the user is advised to proceed with caution.

- flag_save: This parameter can be 0 or 1. When it is set to 1, the program will save the results in a “mat” file at the end of the simulation.
- flag_plot: This parameter can be 0 or 1. When it is set to 1, the program will plot the results while computing them and at the end of the simulation.
- flag_echo: This parameter can be 0 or 1. When it is set to 1, the program prints the progress of the calculation in the command window.
- chol_sol: This parameter can be 0 or 1. When it is set to 1 the solver will use Cholesky factorization and back substitution to solve the global system of equations. If this value is not equal to 1, MATLAB will use the default backslash solve to solve the system of equations. Note that for the particular case where the diffusion coefficients remain constant (i.e. at constant P-T), the code can be modified so that the global left-hand-side matrix ($\boldsymbol{KM}$) is not re-assembled at each timestep. This will improve the code’s performance dramatically. However, we will not enter in such details in this first version.
- theta: This parameter can be between 0 and 1. However, values less than 0.5 can lead to unstable calculations. For optimal performance this parameter is set to 1.
- nt: This number must be an integer and denotes the number of time increments employed by the calculation. Increasing this number may increase accuracy but making this number too large will increase the time needed for the calculation.
- nels: This number must be an integer and represents the number of elements employed by the FEM method. Increasing this number may increase accuracy but will increase the time needed for the calculation.
- RdX: This number is the grid refinement factor (see section “The variable grid”).

Code testing

In order to ensure that the results produced by *KADMOS_Path* are meaningful we calculated some examples that demonstrate the verification of our code. In these examples we compare our numerical results with results from analytical solutions and we also proceed with a convergence test that shows that the algorithm converges to a unique solution with increasing numerical resolution.

Analytical benchmark

As a first example we consider the degassing of a muscovite crystal at constant temperature and pressure for a given period of time (Fig. 4). In this case, we can calculate the analytical solution of the concentration field, the total (integrated) outward flux as a function of time, and the fractional mass loss as a function of time. We consider a spherical crystal of 50µm radius that is degassed at 700°C (and 1,000MPa) for 1,000 hours. The expressions that were used for the calculation of the analytical solutions (for spherical geometry) are given in eqs. (25):

$$C(r,t) = -\frac{2LC_0}{\pi r}\sum_{n=1}^{n=\infty}\frac{(-1)^n}{n}exp(-n^2X)sin\left(\frac{n\pi r}{L}\right) \tag{25a}$$

$$q(t) = 8\pi LC_0D\sum_{n=1}^{n=\infty}exp(-n^2X) \tag{25b}$$

$$f(t) = 1-\frac{6}{\pi^2}\sum_{n=1}^{n=\infty}\frac{1}{n^2}exp(-n^2X) \tag{25c}$$

where C is the concentration, q is the total outward flux (integrated over the total surface of the sphere) and f is the fractional mass loss (McDougall and Harrison, 1999, p. 141)[10]. X is also a function of time and (for constant diffusivity) is given by:

$$X = \frac{\pi^2}{L^2}\int_0^t D(t')dt' = \frac{\pi^2 Dt}{L^2} \tag{26}$$

The infinite sums in eqs. (25) have been calculated for the first 500 terms (i.e. $n = 500$).

[10] Note that in some expressions of McDougal and Harrison (1999), the diffusivity is omitted (e.g. for flux expressions). This does not influence the results if normalized fluxes are calculated (e.g. for the calculation of an age spectrum). However, this omission can cause dramatic errors when absolute values are needed.

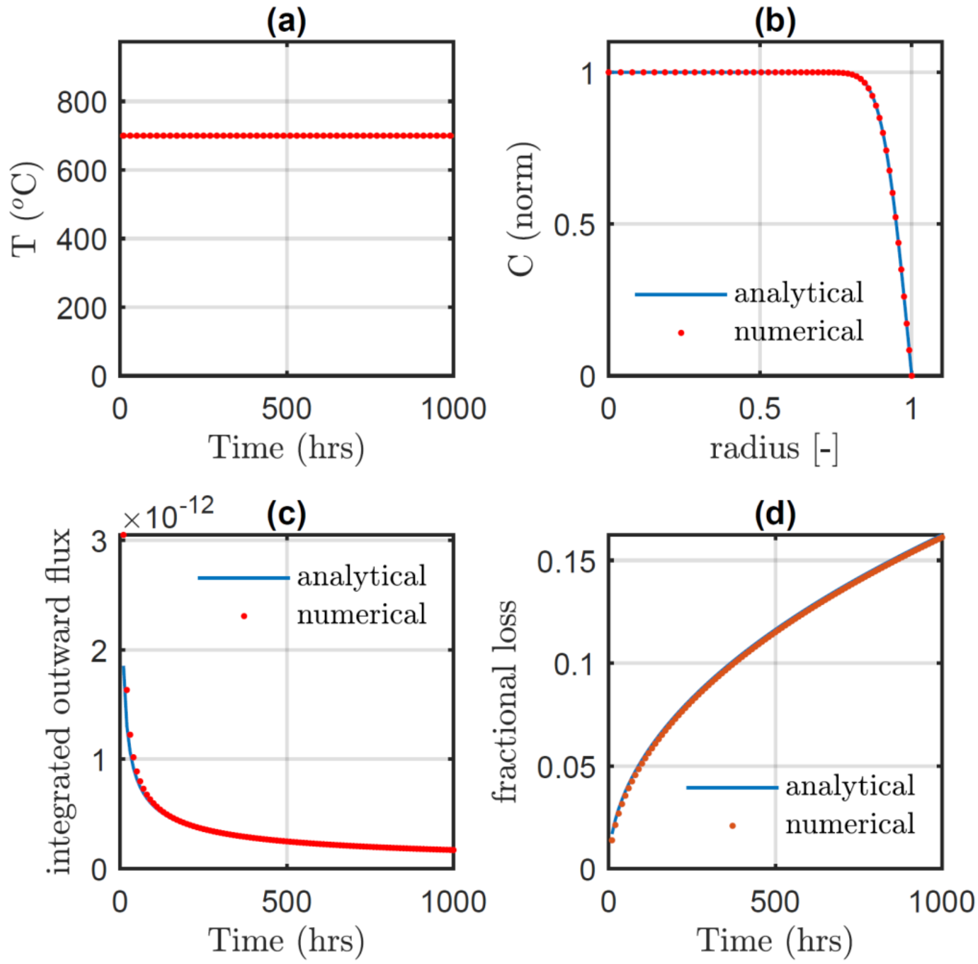


Fig. 4. Numerical and analytical solutions for the constant temperature degassing problem. (a) Temperature as a function of time. (b) Concentration (normalized) as a function of distance (normalized). (c) integrated outward flux as a function of time. (d) Fractional mass loss as a function of time.

In the previous example, the diffusion coefficient had to be calculated at the temperature and pressure of interest. For the numerical calculation, the concentration as a function of the non-dimensional distance is directly evaluated as a natural output of the KADMOS software (see FEM discretization and solution). To calculate the total outward flux numerically, we considered the equation of mass balance over a given volume as follows:

$$\frac{d}{dt}\iiint CdV = -\oiint JdA \tag{27}$$

where J represents the outward flux at the edge of the crystal. The first term of eq. (27) represents a volume integral and the second term represents a surface integral. Thus, by considering a finite time increment Δt, the total outward flux (q) can be evaluated via:

$$q = \oiint J dA = -\frac{d}{dt}\iiint C dV \approx \frac{\iiint C^j dV - \iiint C^{j-1} dV}{\Delta t} \tag{28}$$

where $\iiint C^j dV$ is the volume-integrated concentration at a given instance j. This method of flux calculation is less sensitive to numerical errors since there is no need for the direct calculation of spatial derivatives.

Numerical resolution test

In this section we show the results of a numerical convergence test, i.e. the converge of the solution with increasing numerical resolution. As an example, we consider the same model configuration that was used to produce Fig. 2. We monitor the age of the core of a crystal as a function of the number of elements (nels keyword). The results are plotted in Fig. 5 and show that for calculations with 1,000 number of elements, the apparent age in the core of a spherical crystals is essentially the same. Moreover, our results show that even a calculation with 50 elements yields accurate apparent age with the variation being in the order of 1,000 yrs.

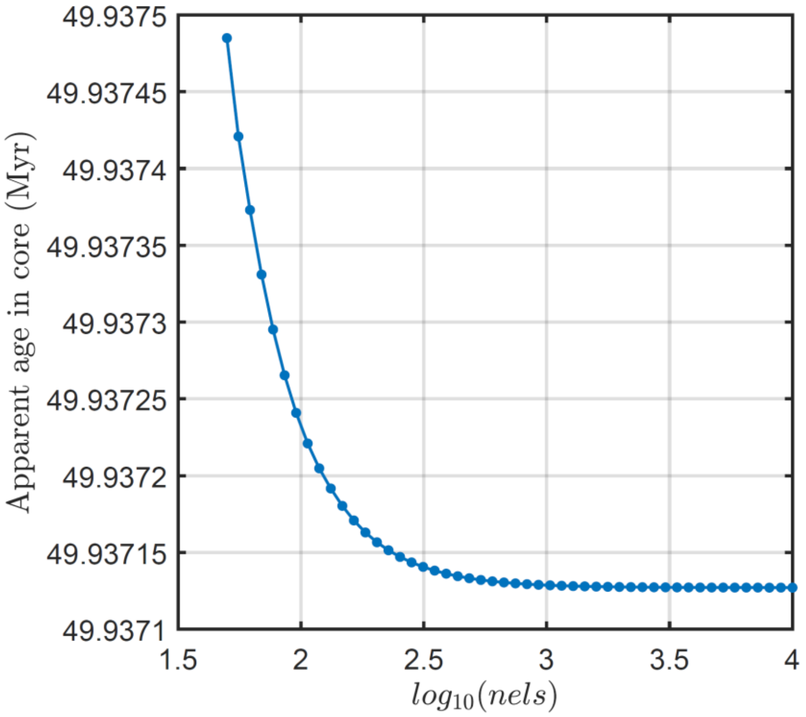


Fig. 5. Results of the numerical resolution test. *nels* represents the number of elements in the FEM model. Solution parameters as for Fig. 2.

Using KADMOS in MATLAB

The use of KADMOS software is straightforward. First the user needs to copy and paste in the same folder the two ".m" files from the KADMOS_MATLAB folder. These are:

- KADMOS_Path.m (contains the computational keywords and options)
- Inputs_Path.m (contains keywords related to the problem definition)

Once MATLAB opens in the folder where these files are located, the user can run the program KADMOS_Path.m either by right-clicking on it, or by typing:

```
KADMOS_Path
```

in the command window.

Using KADMOS in OCTAVE

Although KADMOS was written originally in MATLAB we have made sure that there is also compatibility with OCTAVE. The files that are needed to run the OCTAVE-friendly version are located in the folder KADMOS_OCTAVE. The main difference from the MATLAB version is that all the functions used must be present as separate files (and not included within other .m files) in the same folder. Therefore, the user needs to copy all the contents of the KADMOS_OCTAVE folder in a directory. These files are:

- KADMOS_Path_OCTAVE.m (contains the computational keywords and options)
- Inputs_Path_OCTAVE.m (contains keywords related to the problem definition)
- make_PTtpath.m (a function needed in Inputs_Path_OCTAVE)
- linspace_adapt.m (a function needed in KADMOS_Path_OCTAVE.m)
- fill_Matrix_FEM.m (a function needed in KADMOS_Path_OCTAVE.m)
- diff_system.m (a function needed in KADMOS_Path_OCTAVE.m)
- check_ndomains.m (a function needed in KADMOS_Path_OCTAVE.m)

Once OCTAVE opens in the folder where these files are located, the user can run the program KADMOS_Path_OCTAVE.m either by right-clicking on it, or by typing:

```
KADMOS_Path_OCTAVE
```

in the command window.

References

Braun, J., Beek, P. van der, and Batt, G., 2006, Quantitative Thermochronology: Numerical Methods for the Interpretation of Thermochronological Data: Cambridge University Press, Cambridge, 258 p.

Dabrowski, M., Krotkiewski, M., and Schmid, D. W., 2008, MILAMIN: MATLAB-based finite element method solver for large problems: Geochemistry, Geophysics, Geosystems, v. 9.

Foland, K. A., 1994, Argon Diffusion in Feldspars, *in* Parsons, I. ed., Feldspars and their Reactions: Springer Netherlands, Dordrecht, p. 415–447.

Grove, M., and Harrison, T. M., 1996, 40Ar* diffusion in Fe-rich biotite: American Mineralogist, v. 81, p. 940–951.

Harrison, T. M., 1982, Diffusion of 40Ar in hornblende: Contributions to Mineralogy and Petrology, v. 78, p. 324–331.

Harrison, T. M., Célérier, J., Aikman, A. B., Hermann, J., and Heizler, M. T., 2009, Diffusion of 40Ar in muscovite: Geochimica et Cosmochimica Acta, v. 73, p. 1039–1051.

Harrison, T. M., Duncan, I., and McDougall, I., 1985, Diffusion of 40Ar in biotite: Temperature, pressure and compositional effects: Geochimica et Cosmochimica Acta, v. 49, p. 2461–2468.

Lasaga, A. C., 1983, Geospeedometry: An extension to geothermometry, *in* Saxena, S. K. ed., Kinetics and Equilibrium in Mineral Reactions: Springer New York, p. 81–114.

Lister, G. S., and Baldwin, S. L., 1996, Modelling the effect of arbitrary P-T-t histories on argon diffusion in minerals using the MacArgon program for the Apple Macintosh: Tectonophysics, v. 253, p. 83–109.

McDougall, I., and Harrison, T. M., 1999, Geochronology and Thermochronology by the 40Ar/39Ar Method: Oxford University Press, 269 p.

Räss, L., Duretz, T., Podladchikov, Y. Y., and Schmalholz, S. M., 2017, M2Di: Concise and efficient MATLAB 2-D Stokes solvers using the Finite Difference Method: Geochemistry, Geophysics, Geosystems, v. 18, p. 755–768.

Reiners, P. W., Carlson, R. W., Renne, P. R., Cooper, K. M., Granger, D. E., McLean, N. M., and Schoene, B., 2017, Diffusion and thermochronologic interpretations, *in* Geochronology and Thermochronology: John Wiley & Sons, Ltd, p. 83–126.

Robbins, G. A., 1972, Radiogenic argon diffusion in muscovite under hydrothermal conditions:

Simpson, G., 2017, Practical Finite Element Modelling in Earth Science Using Matlab: Wiley-Blackwell, 248 p.

Skipton, D. R., Warren, C. J., and Hanke, F., 2018, Numerical models of P–T, time and grain-size controls on Ar diffusion in biotite: An aide to interpreting 40Ar/39Ar ages: Chemical Geology, v. 496, p. 14–24.

Wheeler, J., 1996, Diffarg: A program for simulating argon diffusion profiles in minerals: Computers & Geosciences, v. 22, p. 919–929.