User’s Manual for

Diffuser (Numerical)
Beta 2.0

Diffusion modeling using numerical methods

Released May 24, 2022

www.geoapp.cn

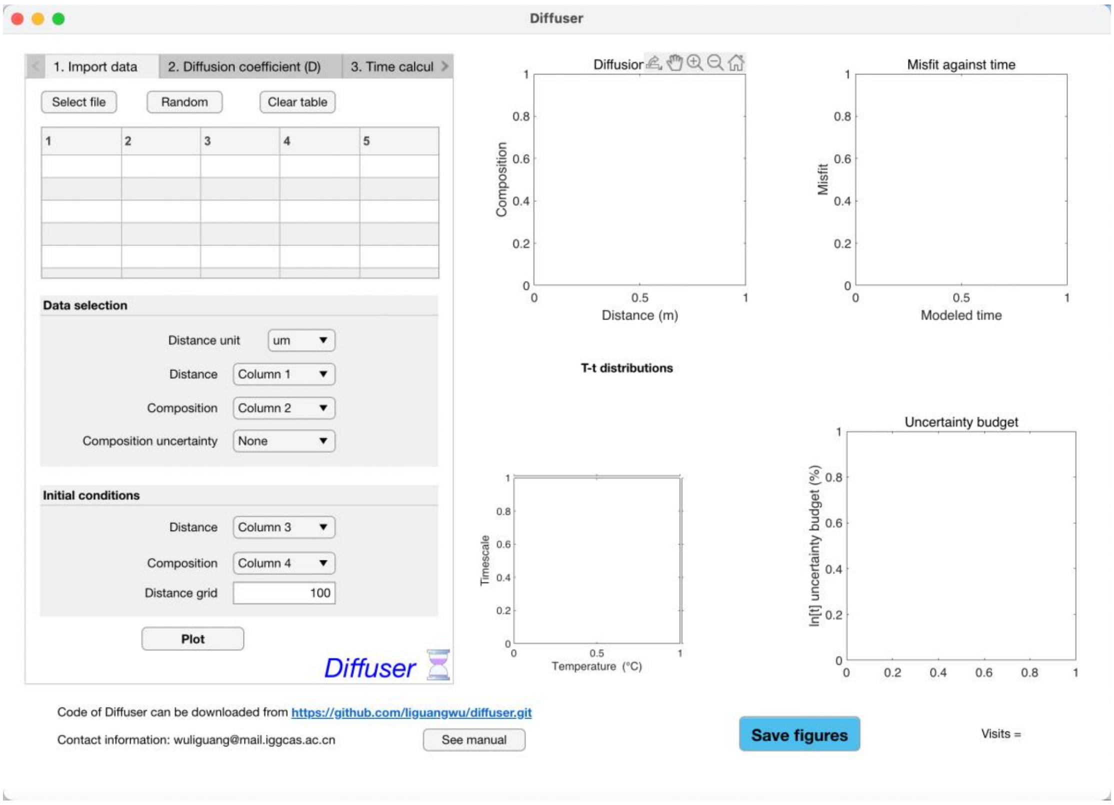
Li-Guang Wu, Yang Li, Michael C. Jollands, Pieter Vermeesch, Xian-Hua Li, 2022. Diffuser: a user-friendly program for diffusion chronometry with robust uncertainty estimation. Computers and Geosciences. https://doi.org/10.1016/j.cageo.2022.105108

Contact: wlg@cugb.edu.cn (L.G. Wu) and geoliy@outlook.com (Y. Li)

Table of Contents

Diffuser (Numerical) is a program written in MATLAB to model elemental diffusion in minerals. It uses numerical methods to calculate time with uncertainty propagation of curve fitting, temperature, and experimentally determined diffusion coefficients. Distinguished from the first version of Diffuser which uses analytical solutions, it is designed for solving more complex diffusion problems, i.e.,
(1) modeling concentration-dependent diffusion
(2) calculating cooling rate and timescale.
Users are strongly recommended to use offline version (downloaded from https://github.com/liguangwu/DiffuserNumerical), which uses parallel computing to accelerate calculation. A web version (Figure 1) is available at http://www.geoapp.cn/. But this online version estimating uncertainties is much slower than offline version due to limited server performance (unable to use parallel computing).

The current app is a beta version. Any bug report can be sent to wlg@cugb.edu.cn.



Distance unit: um
Distance: Column 1
Composition: Column 2
Composition uncertainty: None

Distance: Column 3
Composition: Column 4
Distance grid: 100

Code of Diffuser can be downloaded from https://github.com/liguangwu/diffuser.git
Contact information: wuliguang@mail.iggcas.ac.cn

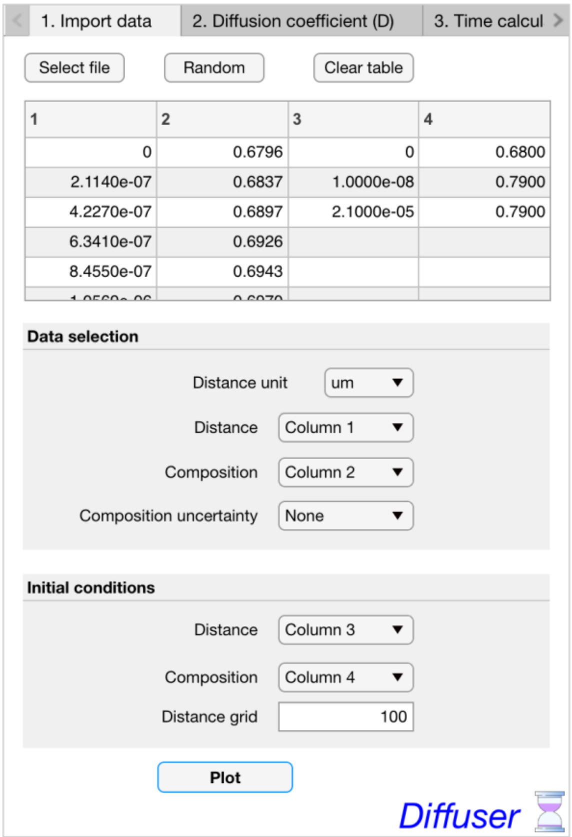
Figure 1. A screenshot of Diffuser (Numerical)

A user manual is given below.

1 Import data

1.1 Upload file

After opening the graphical user interface (GUI) of Diffuser (Numerical), the user can click the ‘**Select file**’ button from the import data panel to upload a data file (Figure 2). The measured diffusion profile can be fed into the software through a delimited text (e.g., txt or csv format) or spreadsheet file (Microsoft® Excel).



1	2	3	4
0	0.6796	0	0.6800
2.1140e-07	0.6837	1.0000e-08	0.7900
4.2270e-07	0.6897	2.1000e-05	0.7900
6.3410e-07	0.6926		
8.4550e-07	0.6943		

Data selection

Distance unit: um

Distance: Column 1

Composition: Column 2

Composition uncertainty: None

Initial conditions

Distance: Column 3

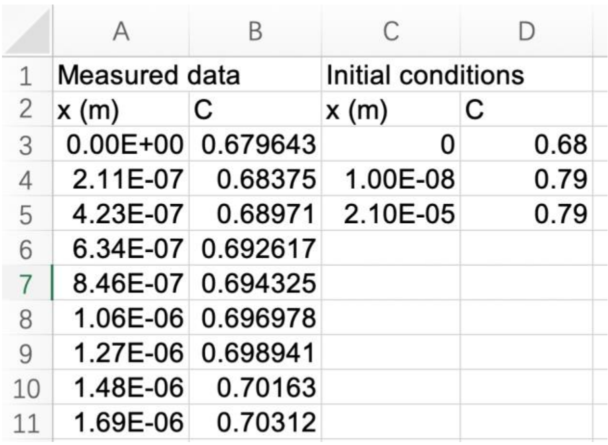
Composition: Column 4

Distance grid: 100

Figure 2. Import data panel of Diffuser (Numerical)

1.2 Select what each column represents

The diffusion data should be in columns (Figure 3) and the user can select which column is the measured distance, composition, composition uncertainty (optional), and initial conditions including distance and composition (Figure 2).



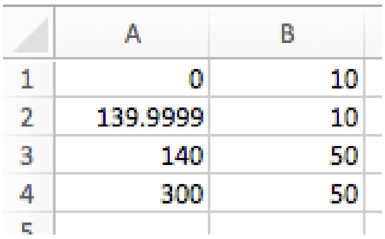
	A	B	C	D
1	Measured data		Initial conditions	
2	x (m)	C	x (m)	C
3	0.00E+00	0.679643	0	0.68
4	2.11E-07	0.68375	1.00E-08	0.79
5	4.23E-07	0.68971	2.10E-05	0.79
6	6.34E-07	0.692617		
7	8.46E-07	0.694325		
8	1.06E-06	0.696978		
9	1.27E-06	0.698941		
10	1.48E-06	0.70163		
11	1.69E-06	0.70312		

Figure 3. An example of diffusion data to be imported

Note:

1) The initial condition needs to contain sufficient points to describe the whole profile and the two boundary conditions. For example, if the measured distance is between 0 µm and 21 µm, the distance of initial conditions should be wider, with minimum <=0 µm and maximum >=21 µm. However, the initial conditions do not need to be the same length as the measured data (e.g., Figure 3).

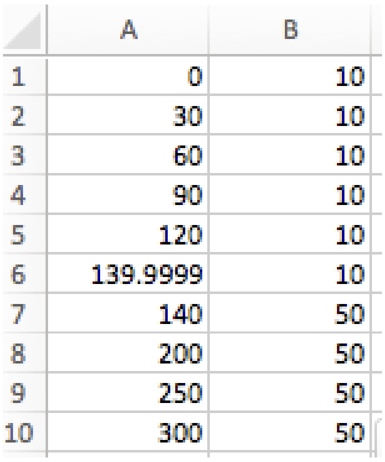
2) The distance of the initial condition cannot have any duplicate values. For example, consider a case where we want to simulate a step function, with extremities at 0 and 300, with the step at 140. The value between 0 and 140 is 10, and the value between 140 and 300 is 50. As stated above, the distance cannot have any duplicates, which means we must define the step using two values close to, but not exactly, the same, such as 139.9999 and 140. A spreadsheet with the initial conditions would look like this:



	A	B
1	0	10
2	139.9999	10
3	140	50
4	300	50
5		

Figure 4. An example of initial conditions

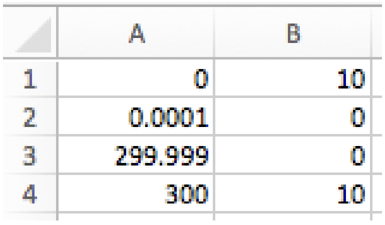
Diffuser (Numerical) will assume that the concentration at position 0 is always 10, and the concentration at position 300 is always 50. The minimum number of points required to describe this step is four. If extra points are added, the outcome will not be changed. For example, the initial conditions in Figure 5 will give the same result.



	A	B
1	0	10
2	30	10
3	60	10
4	90	10
5	120	10
6	139.9999	10
7	140	50
8	200	50
9	250	50
10	300	50

Figure 5. An example of initial conditions that has the same result as Figure 4

For a second example, let's assume we want the initial conditions to be a homogeneous zero concentration with a length of 300, with interfaces at positions 0 and 300 with concentrations of 10. Again, because duplicates are not permitted, a potential input could be:



	A	B
1	0	10
2	0.0001	0
3	299.999	0
4	300	10

Figure 6. An example of initial conditions

3) If composition uncertainty (σ) is assigned 'None' (Figure 2), all composition data will be treated with equal weights.

1.3 Set the distance unit and grid

The user must choose the unit of the distance, e.g., nm, um, mm, or m. This works for both the measured distance and initial conditions. So the user has to prepare them in the same unit.

The user also has to set the grid points for dividing the distance of initial conditions. For example, 100 in Figure 2 (as default) means the software will divide the distance of initial conditions into 100 grids. The composition at each grid point is interpolated **linearly** based on the initial conditions.

Note:

The grid size is better to be set close to the measured data. For example, if the interval of two neighbor points along the measured distance is 2 μm, a similar or smaller grid size can ensure the precision of calculation. A smaller grid size will take more time to compute.

1.4 Plot the diffusion profile

The user can click the ‘**Plot**’ to see the measured diffusion profile (Figure 7) based on the above settings.

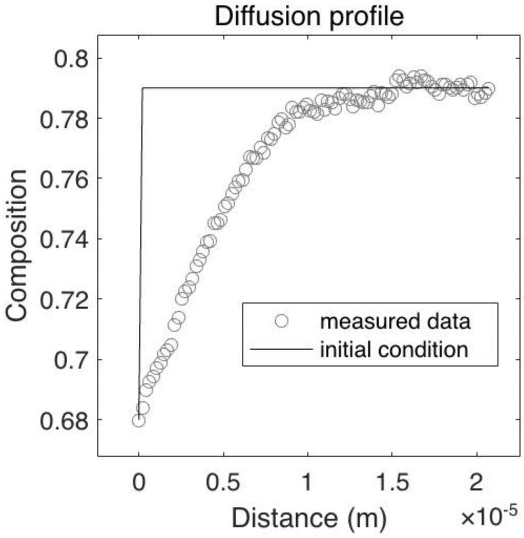


Figure 7. A designed plot of the diffusion profile

1.5 Other options

To play with Diffuser (Numerical), the user can click the ‘**Random**’ button that will generate a random diffusion profile. Then the user can deal with the example data in the same way. For clearing the table and plot of diffusion data, just click the ‘**Clear table**’ button.

2 Set the diffusion coefficient

To calculate the diffusion time of a given element in a specific mineral, the user should select the mineral, element, and data reference on the diffusion coefficient panel (Figure 8).

1) A simple Arrhenius equation (D is a function of temperature) has been used in the first version of Diffuser and is implemented directly here.

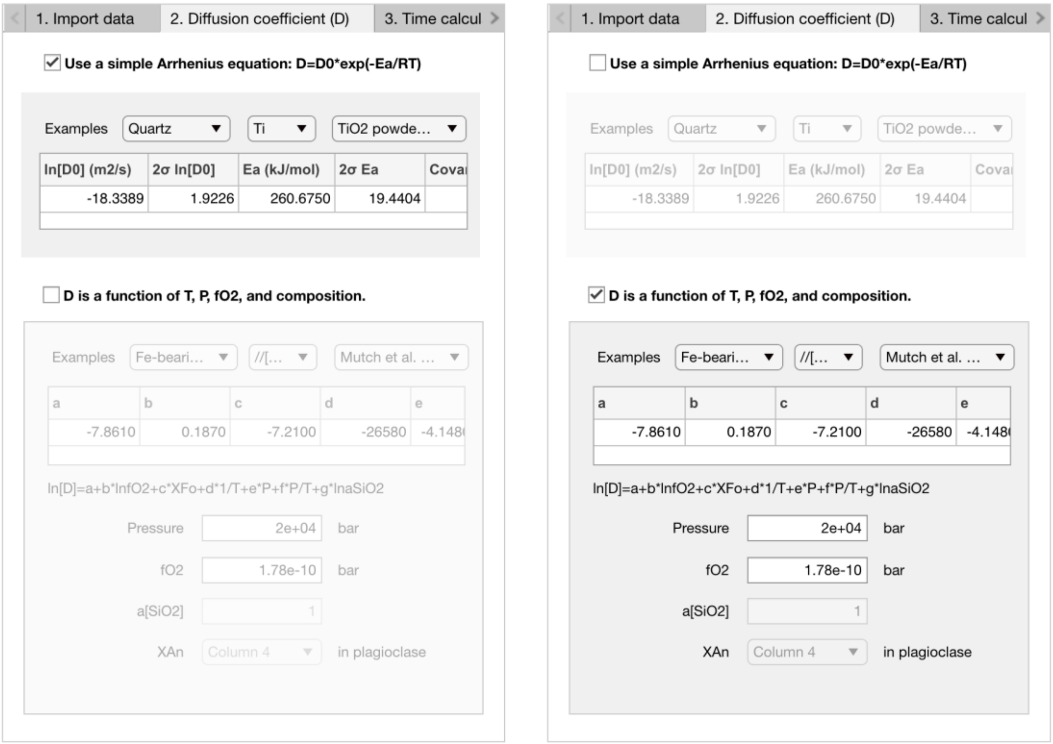
2) In addition, when D is a function of temperature, pressure, oxygen fugacity, and composition (whether itself or other elements), a unified equation form is used for calculation (Mutch et al., 2021). Currently, Diffuser (Numerical) implements the diffusion coefficient of **olivine** (Fe-Mg, Ni, Mn) and **plagioclase** (Mg, Sr, Ba) that has

been compiled by Mutch et al. (2021) (Figure 9), which considers the covariance matrix of the parameters in the equation of diffusion coeffcient. Diffusive anisotropy of olivine is taken to be six times faster along the [001] axis than the [010] and [100] axes for Fe-Mg and Mn (Chakraborty, 2010), and 10.7 times faster for Ni (Spandler & O'Neill, 2010).

Note:

1) Whether the pressure, fO_2 (oxygen fugacity), a[SiO_2] (SiO_2 activity), and X_{An} (anorthite content in the plagioclase along the diffusion profile) should be set depends on the diffusion coefficient. Diffuser (Numerical) reminds the user to input them by making them editable on the panel. For example, if Fe-Mg in olivine is selected, the pressure and fO_2 are editable and should be input (Figure 9).

2) The X_{An} option is only editable when the plagioclase is chosen and the diffusivity is dependent on it. If Mn or Ni in olivine is selected that is dependent on X_{Fo} (forsterite content of the olivine), the text of this option will be changed to X_{Fo} automatically. The X_{An} or X_{Fo} is mole fraction between 0 and 1 and should be imported in the same way as the initial composition (see section 1.2).



1. Import data | 2. Diffusion coefficient (D) | 3. Time calcul

☑ Use a simple Arrhenius equation: D=D0*exp(-Ea/RT)

Examples: Quartz ▼ | Ti ▼ | TiO2 powde... ▼

ln[D0] (m2/s)	2σ ln[D0]	Ea (kJ/mol)	2σ Ea	Cova
-18.3389	1.9226	260.6750	19.4404	

□ D is a function of T, P, fO2, and composition.

Examples: Fe-beari... ▼ | //[... ▼ | Mutch et al. ... ▼

a	b	c	d	e
-7.8610	0.1870	-7.2100	-26580	-4.148(

ln[D]=a+b*lnfO2+c*XFo+d*1/T+e*P+f*P/T+g*lnaSiO2

Pressure: 2e+04 bar

fO2: 1.78e-10 bar

a[SiO2]: 1

XAn: Column 4 ▼ in plagioclase

1. Import data | 2. Diffusion coefficient (D) | 3. Time calcul

□ Use a simple Arrhenius equation: D=D0*exp(-Ea/RT)

Examples: Quartz ▼ | Ti ▼ | TiO2 powde... ▼

ln[D0] (m2/s)	2σ ln[D0]	Ea (kJ/mol)	2σ Ea	Cova
-18.3389	1.9226	260.6750	19.4404	

☑ D is a function of T, P, fO2, and composition.

Examples: Fe-beari... ▼ | //[... ▼ | Mutch et al. ... ▼

a	b	c	d	e
-7.8610	0.1870	-7.2100	-26580	-4.148(

ln[D]=a+b*lnfO2+c*XFo+d*1/T+e*P+f*P/T+g*lnaSiO2

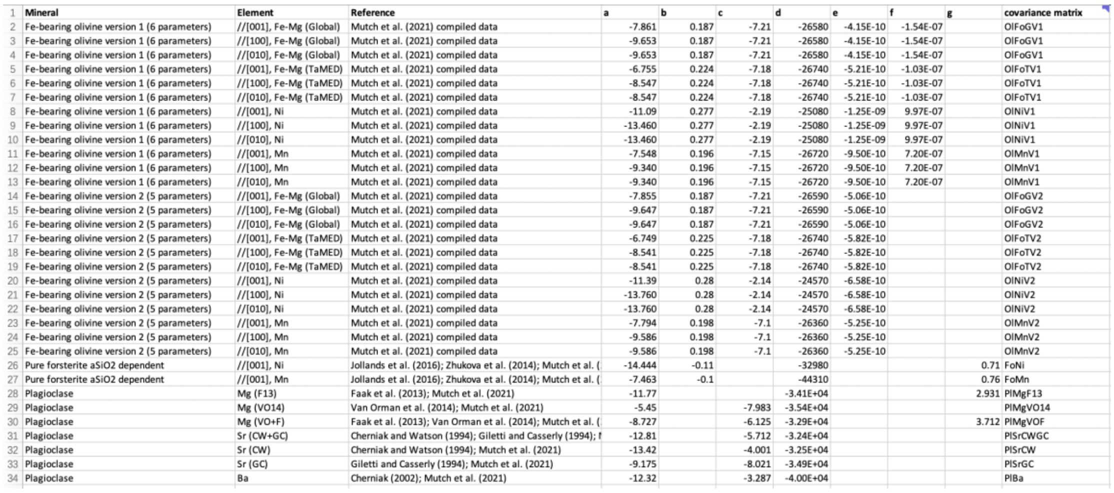
Pressure: 2e+04 bar

fO2: 1.78e-10 bar

a[SiO2]: 1

XAn: Column 4 ▼ in plagioclase

Figure 8. Diffusion coefficient panel of Diffuser (Numerical)



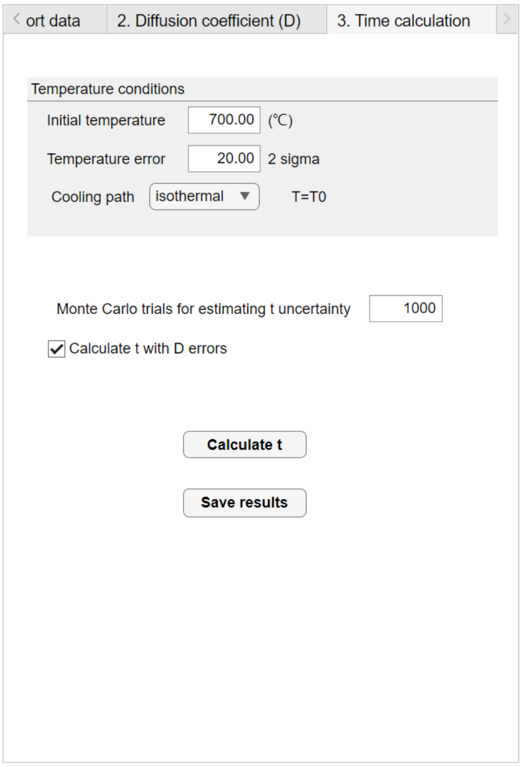
	Mineral	Element	Reference	a	b	c	d	e	f	g	covariance matrix
1	Mineral	Element	Reference	a	b	c	d	e	f	g	covariance matrix
2	Fe-bearing olivine version 1 (6 parameters)	//[001], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-7.861	0.187	-7.21	-26580	-4.15E-10	-1.54E-07		OlFoGV1
3	Fe-bearing olivine version 1 (6 parameters)	//[100], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-9.653	0.187	-7.21	-26580	-4.15E-10	-1.54E-07		OlFoGV1
4	Fe-bearing olivine version 1 (6 parameters)	//[010], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-9.653	0.187	-7.21	-26580	-4.15E-10	-1.54E-07		OlFoGV1
5	Fe-bearing olivine version 1 (6 parameters)	//[001], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-6.755	0.224	-7.18	-26740	-5.21E-10	-1.03E-07		OlFoTV1
6	Fe-bearing olivine version 1 (6 parameters)	//[100], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-8.547	0.224	-7.18	-26740	-5.21E-10	-1.03E-07		OlFoTV1
7	Fe-bearing olivine version 1 (6 parameters)	//[010], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-8.547	0.224	-7.18	-26740	-5.21E-10	-1.03E-07		OlFoTV1
8	Fe-bearing olivine version 1 (6 parameters)	//[001], Ni	Mutch et al. (2021) compiled data	-11.09	0.277	-2.19	-25080	-1.25E-09	9.97E-07		OlNiV1
9	Fe-bearing olivine version 1 (6 parameters)	//[100], Ni	Mutch et al. (2021) compiled data	-13.460	0.277	-2.19	-25080	-1.25E-09	9.97E-07		OlNiV1
10	Fe-bearing olivine version 1 (6 parameters)	//[010], Ni	Mutch et al. (2021) compiled data	-13.460	0.277	-2.19	-25080	-1.25E-09	9.97E-07		OlNiV1
11	Fe-bearing olivine version 1 (6 parameters)	//[001], Mn	Mutch et al. (2021) compiled data	-7.548	0.196	-7.15	-26720	-9.50E-10	7.20E-07		OlMnV1
12	Fe-bearing olivine version 1 (6 parameters)	//[100], Mn	Mutch et al. (2021) compiled data	-9.340	0.196	-7.15	-26720	-9.50E-10	7.20E-07		OlMnV1
13	Fe-bearing olivine version 1 (6 parameters)	//[010], Mn	Mutch et al. (2021) compiled data	-9.340	0.196	-7.15	-26720	-9.50E-10	7.20E-07		OlMnV1
14	Fe-bearing olivine version 2 (5 parameters)	//[001], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-7.855	0.187	-7.21	-26590	-5.06E-10			OlFoGV2
15	Fe-bearing olivine version 2 (5 parameters)	//[100], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-9.647	0.187	-7.21	-26590	-5.06E-10			OlFoGV2
16	Fe-bearing olivine version 2 (5 parameters)	//[010], Fe-Mg (Global)	Mutch et al. (2021) compiled data	-9.647	0.187	-7.21	-26590	-5.06E-10			OlFoGV2
17	Fe-bearing olivine version 2 (5 parameters)	//[001], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-6.749	0.225	-7.18	-26740	-5.82E-10			OlFoTV2
18	Fe-bearing olivine version 2 (5 parameters)	//[100], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-8.541	0.225	-7.18	-26740	-5.82E-10			OlFoTV2
19	Fe-bearing olivine version 2 (5 parameters)	//[010], Fe-Mg (TaMED)	Mutch et al. (2021) compiled data	-8.541	0.225	-7.18	-26740	-5.82E-10			OlFoTV2
20	Fe-bearing olivine version 2 (5 parameters)	//[001], Ni	Mutch et al. (2021) compiled data	-11.39	0.28	-2.14	-24570	-6.58E-10			OlNiV2
21	Fe-bearing olivine version 2 (5 parameters)	//[100], Ni	Mutch et al. (2021) compiled data	-13.760	0.28	-2.14	-24570	-6.58E-10			OlNiV2
22	Fe-bearing olivine version 2 (5 parameters)	//[010], Ni	Mutch et al. (2021) compiled data	-13.760	0.28	-2.14	-24570	-6.58E-10			OlNiV2
23	Fe-bearing olivine version 2 (5 parameters)	//[001], Mn	Mutch et al. (2021) compiled data	-7.794	0.198	-7.1	-26360	-5.25E-10			OlMnV2
24	Fe-bearing olivine version 2 (5 parameters)	//[100], Mn	Mutch et al. (2021) compiled data	-9.586	0.198	-7.1	-26360	-5.25E-10			OlMnV2
25	Fe-bearing olivine version 2 (5 parameters)	//[010], Mn	Mutch et al. (2021) compiled data	-9.586	0.198	-7.1	-26360	-5.25E-10			OlMnV2
26	Pure forsterite aSiO2 dependent	//[001], Ni	Jollands et al. (2016); Zhukova et al. (2014); Mutch et al. (	-14.444	-0.11		-32980			0.71	FoNi
27	Pure forsterite aSiO2 dependent	//[001], Mn	Jollands et al. (2016); Zhukova et al. (2014); Mutch et al. (	-7.463	-0.1		-44310			0.76	FoMn
28	Plagioclase	Mg (F13)	Faak et al. (2013); Mutch et al. (2021)	-11.77			-3.41E+04			2.931	PlMgF13
29	Plagioclase	Mg (VO14)	Van Orman et al. (2014); Mutch et al. (2021)	-5.45		-7.983	-3.54E+04				PlMgVO14
30	Plagioclase	Mg (VO+F)	Faak et al. (2013); Van Orman et al. (2014); Mutch et al. (	-8.727		-6.125	-3.29E+04			3.712	PlMgVOF
31	Plagioclase	Sr (CW+GC)	Cherniak and Watson (1994); Giletti and Casserly (1994);	-12.81		-5.712	-3.24E+04				PlSrCWGC
32	Plagioclase	Sr (CW)	Cherniak and Watson (1994); Mutch et al. (2021)	-13.42		-4.001	-3.25E+04				PlSrCW
33	Plagioclase	Sr (GC)	Giletti and Casserly (1994); Mutch et al. (2021)	-9.175		-8.021	-3.49E+04				PlSrGC
34	Plagioclase	Ba	Cherniak (2002); Mutch et al. (2021)	-12.32		-3.287	-4.00E+04				PlBa

Figure 9. Olivine and plagioclase diffusion coefficients used in Diffuser (Numerical)

3 Time calculation

3.1 Set the temperature conditions

Next, the **initial temperature with its uncertainty, and cooling path (isothermal, linear, exponential, or parabolic)** should be set on the time calculation panel (Figure 10).



Temperature conditions

Initial temperature 700.00 (℃)

Temperature error 20.00 2 sigma

Cooling path isothermal ▼ T=T0

Monte Carlo trials for estimating t uncertainty 1000

☑ Calculate t with D errors

Figure 10. Time calculation panel of Diffuser.

The linear, exponential, or parabolic cooling path has an equation form of:

T=T_0-At for linear cooling
T=T_0e^{-At} for exponential cooling
T=T_0-At^2 for parabolic cooling

whether T is the real-time temperature, T_0 is the initial temperature, A is a constant decribing the cooling rate, and t is the time. Temperature unit is Kelvin for calculating A. As such, the unit of A is °C/s (or K/s), 1/s, and °C/s^2 (or K/s^2) for the three equations, respectively.

3.2 Set the trials for the Monte Carlo calculation

When the uncertainty of the temperature or parameters of *D* (see below) is propagated into the uncertainty of time using the Monte Carlo method, the user needs to input the trials for the Monte Carlo calculation (Figure 10).

3.3 Propagate the uncertainty of parameters of *D*

The user can choose uncertainty propagation of the experimentally determined parameters of *D* (by ticking the box of '**Calculate *t* with *D* errors**'; Figure 10) to see the contributions to the uncertainty of the calculated timescale compared with curve fitting and temperature.

3.4 Calculate diffusion time

After all parameters are set, the user can start calculating the timescale by clicking '**Calculate *t***' on the panel (Figure 10).

3.5 Calculate cooling rate

Usually, diffusion continues with cooling in natural samples while the cooling rate is unknown. In this case, Diffuser will calculate cooling rates and corresponding timescales automatically. To do this, the user can choose a cooling path other than isothermal, e.g., linear, exponential, or parabolic (section 3.1, Figure 10), and click '**Calculate *t***' button.

4 Output

4.1 Curve fitting

After calculation, a figure will plot the best fit and its 95% confidence level (95c.l.) along with the diffusion profile (Figure 11).

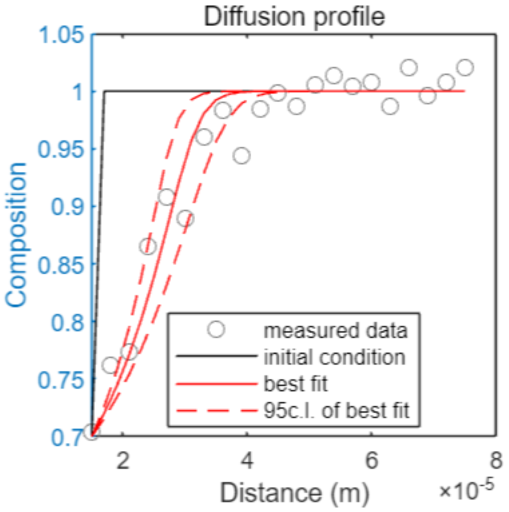


Figure 11. An example of curve fitting in Diffuser (Numerical)

4.2 Plot residual sum of squares (RSS) vs. time or RSS vs. cooling rate

Diffuser (Numerical) will find the best fit of the diffusion profile automatically and plot the data of the residual sum of squares (RSS) against time (Figure 12). The time at the minimum RSS represents the best fit. The 95c.l. definition of best fit can be found in the section of algorithms.

If the user chooses to calculate the cooling rate, this figure will plot RSS vs. cooling rate (Figure 13).

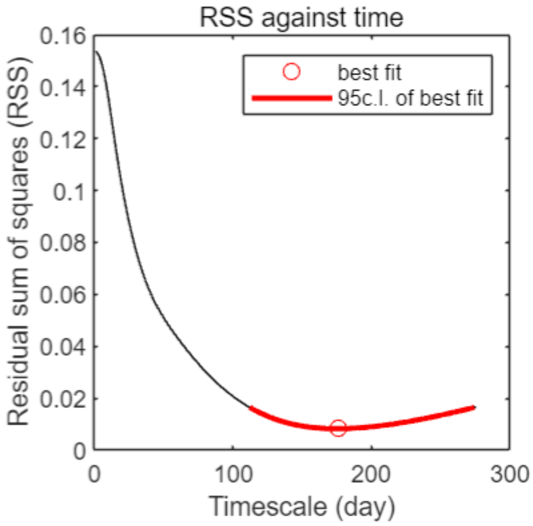


Figure 12. An example of the residual sum of squares (RSS) against time

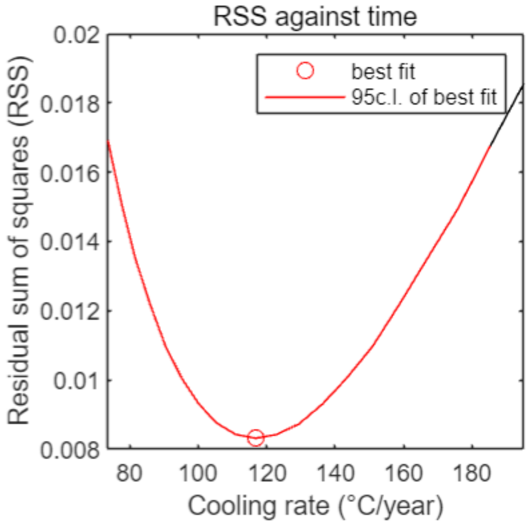


Figure 13. An example of the residual sum of squares (RSS) against cooling rate

4.3 T-t distributions or Cooling rate-t distributions

If the temperature is assigned a non-zero error, a marginal plot will show distributions of the temperature and diffusion timescale and the trade-off between these two parameters (Figure 14). The final calculated timescale will be shown in the top right corner of this figure.

If the user chooses to calculate the cooling rate, this figure will plot cooling rate vs. timescale (Figure 15).

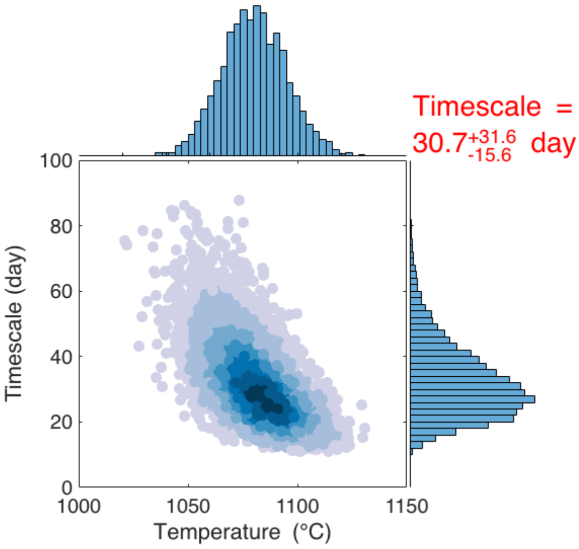


Figure 14. An example of the distributions of the temperature and diffusion timescale.

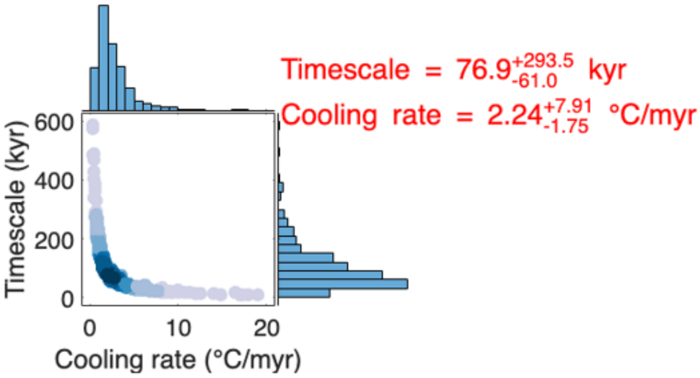


Figure 15. An example of the distributions of the cooling rate and corresponding timescale.

4.4 Uncertainty budget

A histogram will also show the uncertainty budget of the modeled timescale (Figure 16), so that the user can evaluate the main contributions of the timescale uncertainty.
A dialog will appear saying that 'Trials are not enough to estimate ln[t] uncertainty budget' when the calculated uncertainty of the timescale using propagation of three error sources (curve fitting, temperature, and parameters of *D*) is even smaller than that using propagation of two error sources (or two sources<one source). In this case, the user should consider using a larger number of trials for Monte Carlo modeling.
If the user chooses to calculate the cooling rate, this figure will plot the uncertainty budget of cooling rate and timescale (Figure 17).

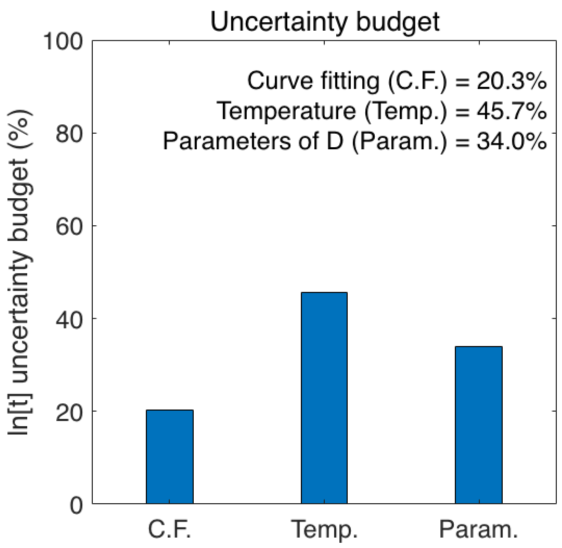


Figure 16. Histogram showing the uncertainty budget of the modeled timescale.

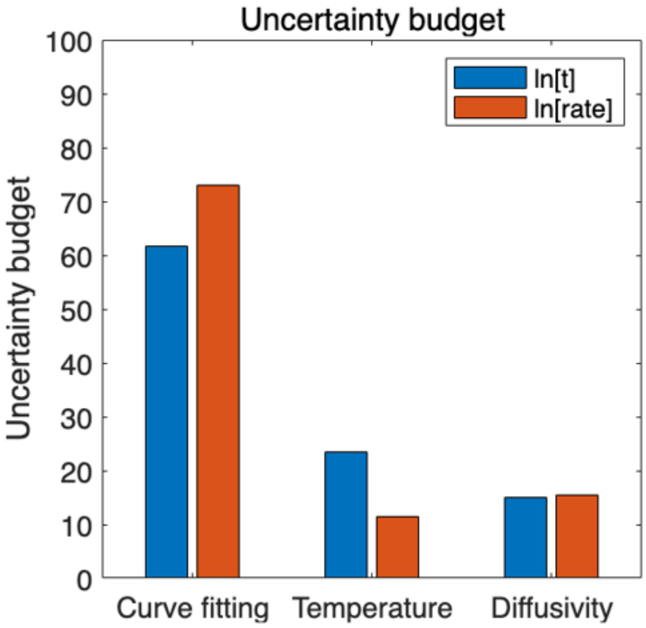


Figure 17. Histogram showing the uncertainty budget of the modeled cooling rate and timescale.

4.5 Monte Carlo modeling results

After calculation, the user can save the Monte Carlo modeling results by clicking the ‘Save results’ button (Figure 10).

5 Algorithms of Diffuser (Numerical)

Diffuser (Numerical) uses the Crank-Nicolson scheme (Crank and Nicolson, 1947) to solve diffusion equations by the finite differences method. Additionally, there are two specific algorithms designed for Diffuser (Numerical), which are given below:

5.1 Definition of best fit and its 95% confidence level (95c.l.)

The misfit between the modeled diffusion curve (Cp) and measured diffusion profile (Cm) is calculated as residual sum of squares (RSS):

$$\text{RSS} = \sum_1^n \frac{1}{{\sigma_i}^2}(C_{pi} - C_{mi})^2$$

As time elapses and diffusion continues, RSS will reach the lowest value when Cp best matches Cm (Figure 12). This timescale is considered as the best estimation. If the user chooses to calculate the cooling rate, the best fit is defined by a suitable cooling rate and timescale that freezes the diffusion curve to best fit the measured diffusion profile. The 95% confidence level of best fit is defined as: when RSS gets larger than the best fit value (Figure 12), at a certain RSS value (i.e., a certain timescale or cooling rate), two diffusion curves that have this RSS value (lying at two sides of the best fit; Figure 11) cover ~95% points of the total useful measured points (the points lying outside the range of initial concentration conditions are not useful since diffusion curve will never reach these points).

5.2 When the diffusion profile is frozen?

If the user chooses to calculate the cooling rate, Diffuser will try various cooling rate scenarios to identify suitable rates that can effectively lower the temperature to a threshold value capable of freezing the diffusion profile to the measured data. Diffuser will try the maximum of cooling rate (Am) first, and reduce it step by step to find suitable values that can make the finally frozen diffusion curve lie between the 95c.l. of best fit. Taking the linear cooling as an example, Am is defined as:

$$\text{Am} = \frac{T_m}{t_m}$$

where Tm is the initial temperature (maximum temperature variation for cooling) and tm is the timescale when modeling the diffusion process under isothermal conditions (minimum timescale for cooling).

But when we can consider a diffusion profile is frozen? Diffuser's definition is that if RSS tends to be 'nearly' constant when T drops to the end (Figure 18), the diffusion curve is frozen. Then we can define a tolerance that Diffuser will stop modeling when the misfit of the modeled diffusion curve between time step n and n-1 reaches:

$\sum_1^n \frac{1}{\sigma_i{}^2}(C_i^n - C_i^{n-1})^2 \ < \ \sum_1^n \frac{1}{\sigma_i{}^2}(C_i^{n-1} * tolerance)^2$ (default tolerance is 1e-8 in Diffuser),
which we can simply imagine that at the time step n, RSS is the same between step n and n-1, i.e., the diffusion curve is the same between the two steps.

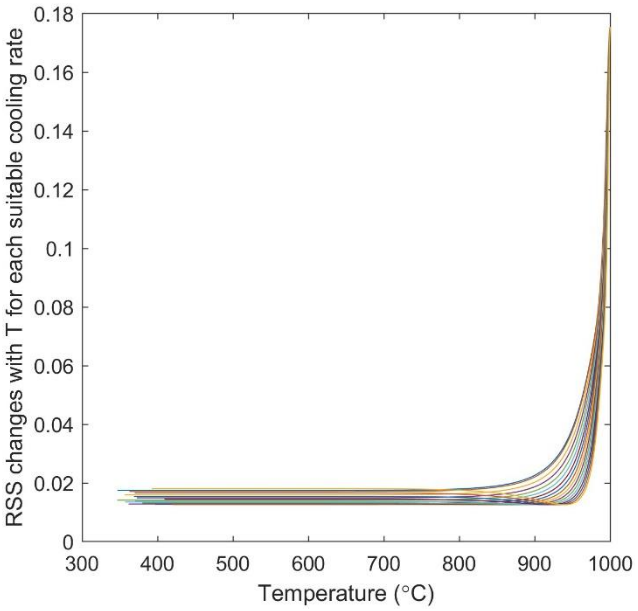


Figure 18. RSS vs. temperature during cooling. When the RSS is nearly constant, Diffuser considers the diffusion curve to be frozen.

Deconvolute the beam effect (optional)

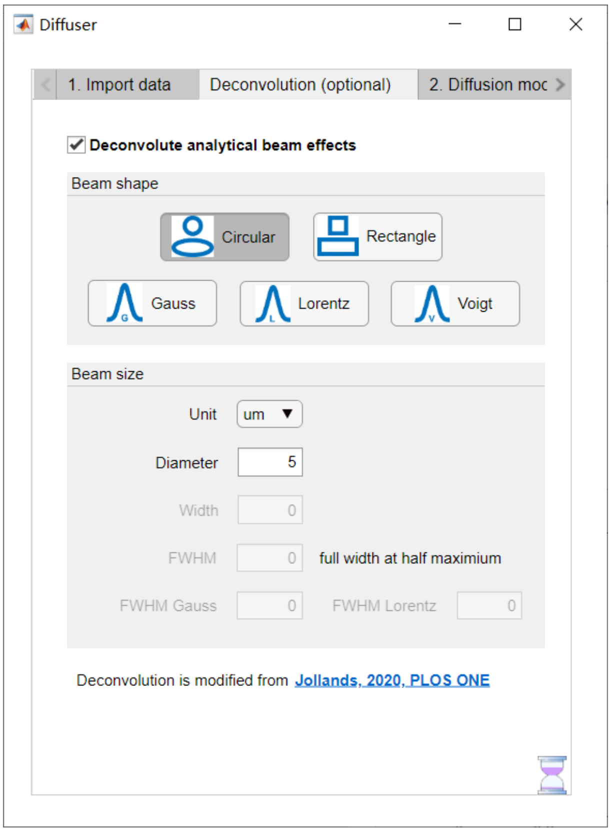
Why and when the user needs deconvolution: All in-situ analytical techniques have a non-zero beam size, so the measured diffusion profiles will suffer from convolution to some degree, especially when the diffusion length approaches the resolution of the analytical technique. Jollands (2020) has developed a program for numerically deconvoluting diffusion profiles acquired using techniques with Gaussian, Lorentzian, (pseudo-)Voigt, circular/elliptical, or square/rectangular interaction volumes (PACE-the Program for Assessing Convolution Effects), which has been incorporated into Diffuser (Numerical).

1 Choose deconvolution

The user can evaluate the convolution effect and determine whether deconvolution is necessary by ticking the box of ‘**Deconvolute analytical beam effects**’ on the ‘Deconvolution’ panel (Figure 19).

2 Set the beam shape and size

If deconvolution is chosen, the beam shape and size and the **size unit** should be set (Figure 19). If a **circular/elliptical** beam shape is selected, the **diameter** of the beam is required. If a **square/rectangular** beam shape is selected, the **width** of the beam is required. If a **Gaussian or Lorentzian** beam shape is selected, the **full width of half maximum** (FWHM) of the beam is required. If a **pseudo-Voigt** beam shape is selected, the **FWHM of both the Gaussian and Lorentzian** components is needed. After the beam shape and size are set, Diffuser (Numerical) will deconvolute the diffusion profile to calculate the timescale.



☑ **Deconvolute analytical beam effects**

Unit: um

Diameter: 5

Width: 0

FWHM: 0 full width at half maximium

FWHM Gauss: 0

FWHM Lorentz: 0

Deconvolution is modified from Jollands, 2020, PLOS ONE

Figure 19. Deconvolution panel of Diffuser (Numerical)

References

Chakraborty, S., 2010. Diffusion Coefficients in Olivine, Wadsleyite and Ringwoodite. Reviews in mineralogy and geochemistry 72, 603–639. https://doi.org/10.2138/rmg.2010.72.13

Crank, J., Nicolson, P., 1947. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Mathematical Proceedings of the Cambridge Philosophical Society 43, 50–67. https://doi.org/10.1017/S0305004100023197.

Jollands, M.C., 2020. Assessing analytical convolution effects in diffusion studies: Applications to experimental and natural diffusion profiles. PLOS ONE 15, e0241788. https://doi.org/10.1371/journal.pone.0241788

Mutch, E.J.F., Maclennan, J., Shorttle, O., Rudge, J.F., Neave, D.A., 2021. DFENS: Diffusion chronometry using finite elements and nested sampling. Geochemistry, Geophysics, Geosystems 22, e2020GC009303. https://doi.org/10.1029/2020GC009303

Spandler, C., O'Neill, H.S.C., 2010. Diffusion and partition coefficients of minor and trace elements in San Carlos olivine at 1,300°C with some geochemical implications. Contributions to Mineralogy and Petrology 159, 791–818. https://doi.org/10.1007/s00410-009-0456-8

Wu, L.-G., Li, Y., Jollands, M.C., Vermeesch, P., Li, X.-H., 2022. Diffuser: A user-friendly program for diffusion chronometry with robust uncertainty estimation. Computers & Geosciences 163, 105108. https://doi.org/10.1016/j.cageo.2022.105108