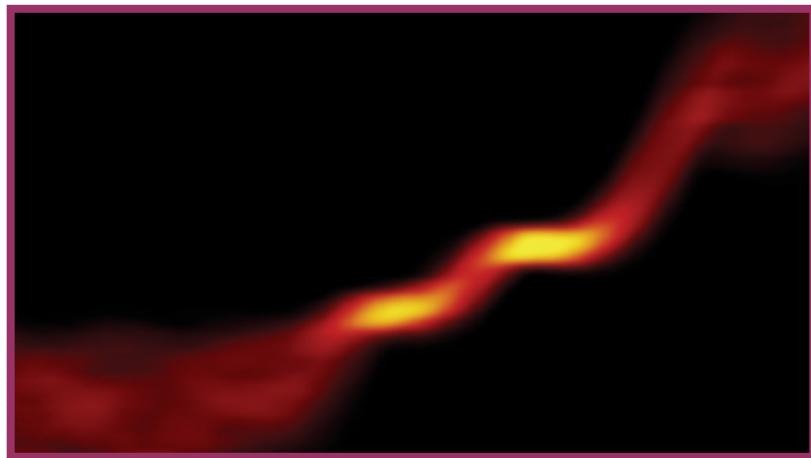


# Arvert 7.0.1

## Inversion of $^{40}\text{Ar}/^{39}\text{Ar}$ Age Spectra and associated mineral ages for thermal history



## User's Manual

Peter Zeitler

Earth & Environmental Sciences  
Lehigh University

[peter.zeitler@lehigh.edu](mailto:peter.zeitler@lehigh.edu)  
[www.ees.lehigh.edu/geochronology.html](http://www.ees.lehigh.edu/geochronology.html)

*release 7.0.1  
updated 16 July, 2023*

## **License**

Arvert is open-source software licensed under the GNU General Public License Version 3. Details about this license and the Arvert code can be found at <https://github.com/OpenThermochronology/Arvert/tree/main>.

# Contents

<b>1. Introduction.....</b>	<b>4</b>
1.1 What is Arvert? .....	4
1.2 How Does Arvert 7.0 Work?.....	4
1.3 How is Arvert Best Used? .....	8
1.4 What's Different About Arvert 7.0? .....	10
<b>2. Obtaining and Installing Arvert.....</b>	<b>14</b>
2.1 Obtaining Code or Applications.....	14
2.2 Overview of Running the Application .....	14
2.3 Compiling the Code.....	15
<b>3. Using Arvert .....</b>	<b>17</b>
3.1 Program Inputs and Outputs.....	17
Inputs (file <i>crs.in</i> ) .....	17
Inputs (file <i>domains.in</i> ) .....	33
Inputs (file <i>goal.in</i> ).....	35
Inputs (file <i>minerals.in</i> ).....	36
Outputs .....	40
3.2 Viewing Results .....	42
3.3 Modeling Considerations .....	42
While modeling .....	42
When you're done .....	43
3.4 Warnings and Issues .....	44
<b>4. Special Topics.....</b>	<b>49</b>
4.1 History of Arvert .....	49
4.2 Performance.....	50
4.3 Future Versions of Arvert.....	50
<b>5. References and Suggested Reading .....</b>	<b>51</b>
<b>6. Appendix – Convergence Sequence, Sample Results.....</b>	<b>53</b>

## Section One – Introduction

---

### 1.1 What is Arvert?

---

Arvert is a numerical model that inverts K-feldspar  $^{40}\text{Ar}/^{39}\text{Ar}$  age spectra and/or mineral ages for thermal history. Given a measured age spectrum together with kinetic and domain parameters obtained from the way  $^{39}\text{Ar}$  is released during step-heating, and/or mineral ages along with known diffusion kinetics, Arvert will try to find all those thermal histories that would result in an age spectrum and mineral ages like the ones measured.

This latest version of Arvert can address multiple samples and multiple mineral-age data in the inversion (one age spectrum and up to ten minerals for each of ten samples) and includes other significant changes as well – see notes below.

This manual assumes that readers and prospective users of Arvert have an understanding of geochronology and know the basic principles of thermochronology, including the multi-domain model for argon diffusion in feldspars and radiation-damage models for He diffusion in apatite and zircon.

If these paragraphs make no sense to you, now is a good time to stop.

### 1.2 How Does Arvert 7.0 Work?

---

Being an inverse model, Arvert attempts to determine fundamental controlling parameters (i.e., a thermal history) from observations of observed thermochronological data (i.e., an age spectrum and/or mineral ages). The program has two main components: forward models that can calculate an age spectrum or mineral ages for a given thermal history; and algorithms that reshape a pool of initially random thermal histories into a set of solutions that increasingly minimize the mismatch between the calculated and observed age spectra and/or mineral ages.

The core forward model for age spectra is a C++ translation of Oscar Lovera's original Fortran code for the calculation of age spectra from samples incorporating multiple diffusion domains (Lovera et al., 1989). Thus, at its heart, for age spectra Arvert is completely dependent on the assumptions that go into the multi-diffusion domain (MDD) model. There is an extensive literature on this subject and some excellent work and reviews by the UCLA group on the validity of this conceptual model as a description for Ar diffusion in feldspar, and some more recent articles (and comments) exploring possible limitations of the MDD model. If for some

reason you reject the model, then Arvert is not the program for you. The text by McDougall and Harrison (1999) is an excellent place to start for a discussion of  $^{40}\text{Ar}/^{39}\text{Ar}$  thermochronology and the MDD model.

To be clear, here are the assumptions upon which Arvert is based:

- *Ar diffusion in feldspars occurs by volume diffusion*, with similar mechanisms and kinetics operating in nature as during  $^{39}\text{Ar}$  extraction in the laboratory.
- *Feldspars are divided into discrete diffusion domains of differing sizes* (or diffusivity, or both). Although the code can handle domains of differing activation energy, experience shows this to be unnecessary (the code expresses variation among domains in terms of size variation but mathematically, variations in D have exactly the same influence as variations in size).
- *Arvert assumes that variations within an age spectrum are entirely due to accumulation of  $^{40}\text{Ar}$  by radioactive decay and loss of  $^{40}\text{Ar}$  by thermally activated diffusion acting within the sample's domains*: it is assumed that the sample contains no extraneous Ar components or that these have been corrected for.
- *For minerals, volume diffusion controls the accumulation and loss of daughter products*, modified in certain phases by the thermal-history-dependent accumulation of radiation damage. The code cannot address age dispersion due to either geologic factors among samples or complications in diffusion systematics within them.

The initial “classic” portion of the Arvert inversion uses the controlled random search (CRS) algorithm of Price (1977) as originally adapted by Willet (1997) for the inversion of apatite fission-track data. This algorithm retains the advantages of a Monte-Carlo approach in widely searching parameter space for true minima, while converging far more rapidly due to the “learning” component inherent in the CRS method. It is beyond the scope of this guide to discuss inverse methods as they have been applied to thermochronometric data; as a start, see the articles by Willett (1997), Gallagher (2012), and Gallagher et al. (2009) as well as references therein.

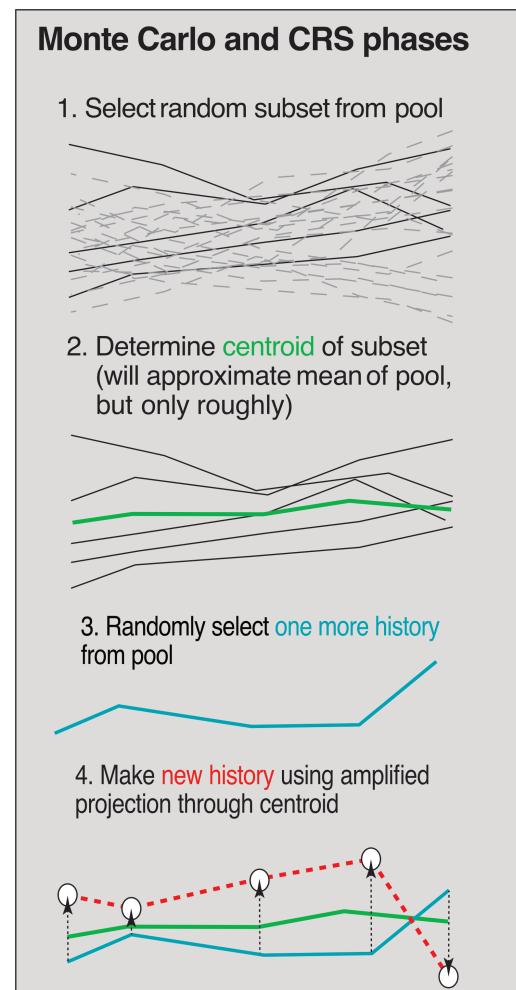
Arvert minimizes the misfit between a measured age spectrum and/or mineral ages, and the spectrum and ages that it calculates. In Arvert this “objective function” can take two main forms, either a simple mean-percent deviation between the observed and modeled spectrum and mineral age(s), or a mean square of weighted deviates (MSWD) that takes into account assigned uncertainties on each heating step and mineral age(s).

In detail, what I am now calling the “classic” Arvert works as follows. Initially, a pool of about 150 to 300 thermal histories is generated at random, subject to any constraints supplied by the operator (“explicit” constraints take the form of maximum and minimum temperatures for the problem space; “implicit” constraints take the form of maximum and minimum heating and cooling rates). Additionally, if there are multiple samples, Arvert can apply specified temperature offsets between samples or let the offsets vary according to the CRS algorithm. An age spectrum, and if used, mineral ages, are calculated for each thermal history (and any T-offset) and a fit parameter is determined for that thermal history. Next, the program goes into its main loop, and begins to sequentially create new thermal histories (and T-offsets), keeping or discarding them depending on whether they provide a better fit than the current worst member of the pool; if a good fit is found, the worst member of the pool is discarded. The program continues until it reaches the specified number of iterations, or has brought all histories in the pool to converge within the specified limits.

In this classic CRS mode, new CRS thermal histories are made as follows. A subset of about 15 to 30 thermal histories is randomly selected from the main pool, plus one more. The histories in the subset are averaged, and then a new history is made by reflecting the additional selected history through the averaged values, subject to an amplification factor that typically might range between 1.1 and 1.5. **Figure 1** gives what is probably a clearer depiction of this process. Arvert’s implementation of the CRS algorithm follows Willett (1997) in differing slightly from the initial Price (1977) algorithm by introducing the amplification factor into the production of a new CRS history.

**Figure 1.** Core CRS algorithm for creating a new thermal history (schematic).

**Arvert 7 (like Arvert 6) differs from earlier versions** in that for the Monte-Carlo stage, a more limited number of time nodes are randomly created for each thermal history, and then these are interpolated into



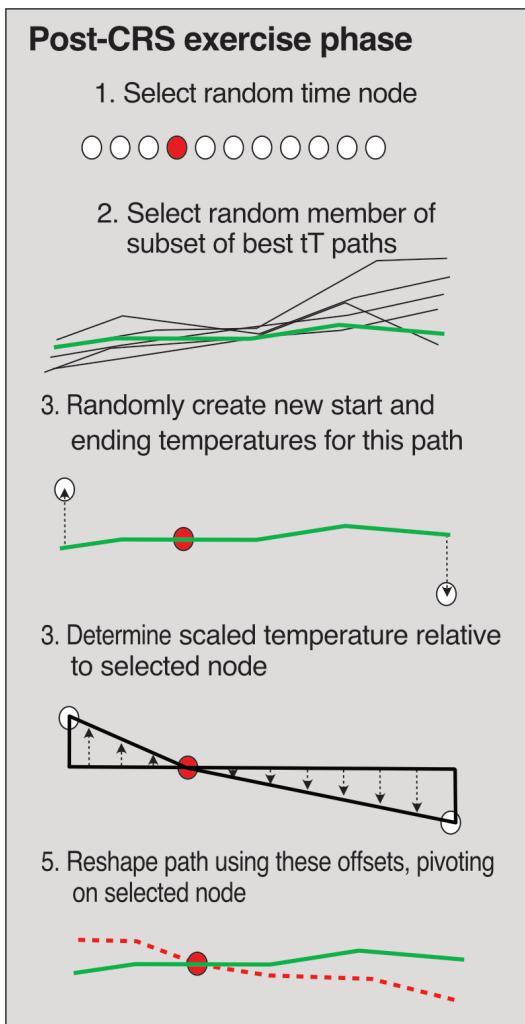
a much larger set of uniformly spaced time nodes that are used for generating the CRS thermal histories. Previous versions used a set of fixed time nodes for both stages; this could lead to bias depending on the location of time nodes, especially if they were fewer in number.

Note that unlike Rich Ketcham's widely used original HeFTy model (Ketcham, 2014), Arvert attempts to coax all histories in its pool to be good-fit histories, whereas classic HeFTy is a pure Monte Carlo model that sifts the problem space to try to find good-fitting histories. While the end results might look similar, it's important to understand this distinction. Note also that Arvert just stops once it's happy with its pool, whereas in Kerry Gallagher's QTQt model (Gallagher, 2012), an ensemble of good-fitting solutions is just the first step in a process that permits Bayesian inference about the statistical bounds of the pool.

## Post processing (NEW to Arvert 7)

Arvert 7 introduces a new inversion option that can be engaged once the CRS portion of the inversion is finished, as a post-processing routine, or earlier as an alternative inversion scheme after only relatively few CRS iterations. One problem we've observed is that Arvert models typically do not converge to statistically good fits when just analytical uncertainty is accounted for, and lacking this stopping point, the CRS algorithm can simply continue to churn. Because of the way good-fitting histories tend to be similar, as a side effect, models often over-converge in regions that should not have constraining power. In practice I have found this new routine to be extremely efficient at finding best fits in regions where data constrain the thermal history while not overconverging outside the constrained region (and in fact better exploring and defining this region).

To overcome this and to exercise a CRS solution set, Arvert can “post-process” any CRS set of solutions. This tends to clarify where the thermal history is and is not constrained by data. The post-processing algorithm is ad-hoc, and works as shown in **Figure 2**. Members of a best-fitting subset of the histories are exercised, and these are perturbed more at the beginning and ends of the models, where there is less likely to be constraining power. Choosing random nodes as the pivot point for this perturbation, choosing random members of the best-fit cohort, and a choosing a random temperature deflection ends up producing a range of tweaks to good-fitting models. By altering the balance between the number of classic CRS versus post-processing iterations, the user can blend different styles of parameter-space investigation (initial MC, CRS, and exploration of CR result).



parameters can bias your results, sometimes in subtle ways, especially when they interact.

Arvert is best used first to investigate what classes of thermal history might serve as explanations for a particular data set, given whatever firm and accurately framed geological constraints you have in hand. Then, once you know what the possibilities are, you can refine inversion options to develop an assessment of your confidence about the various parts of the derived thermal history. With careful use and an understanding of some of its pitfalls, Arvert can provide this information (but see Section 3).

**What you should NOT do** is just run Arvert once for countless iterations, and then take this single result as the truth. You must make some effort to see how different constraints influence the model, and for MDD age spectra, how well your derived diffusion-domain structure is permitting the model to function. Ill-defined kinetic and domain data as well as mineral-age dispersion can make it difficult for

**Figure 2.**  
Post-CRS exercise algorithm (schematic).

As noted, the post-processing routine is *ad hoc*. It seems to work very well, but has not been vetted via a peer-reviewed publication. So, once again, *caveat emptor*. Using the option also requires what I'd called a skilled user who is thinking about the data and about possible biases, and using the option requires more time since you will need to do multiple comparative runs using the restart option. That said, I strongly recommend you include this option in your workflow.

### 1.3 How is Arvert Best Used?

---

Unless you start tinkering with its code, or until you use it for a while, Arvert will appear to you as the quintessential black box, and it will be tempting to treat it as such. *Avoid this temptation!* Make an effort to understand what the main inversion parameters do, and how constraints and

Arvert to fit parts of an age spectrum or mineral ages, causing over-convergence to false thermal-history precision to occur for parts of the thermal history. It is important to remember that in many applications of thermochronology, your goal is to understand process and timing at the realistic level of precision that complex geologic data and usually under-determined thermal models can provide.

Given the current and improving speed of most personal computers and workstations, if you devote a few hours to multiple background runs of Arvert, you should be in good shape. Start by running multiple models for only a few thousand CRS iterations each, perhaps changing the duration of the model space to permit different prehistories to come into play. Try allowing some (re)heating, unless you are certain that any reheating is ruled out by the geology. Alter the number of both MC and CRS time nodes to see whether time-node distribution could be introducing an artifact for your case. Try making a very different Monte Carlo pool at the start of the model, by changing the heating- and cooling-rate constraints. For the Monte-Carlo phase. Alter the amplification factor. After these variations, check that the core result in the region constrained by measured ages does not vary significantly.

If you don't achieve complete statistical convergence (this is common) check to see if there are widespread misfits or if just one age-spectrum step or mineral age is dominating the misfit (Arvert provides an output file with this information named **aa-stepfit.txt**). Widespread misfits might be due to an error in MDD domain structure, some pathology in your age spectra or your mineral ages, or an attempt to model a sample's age spectrum past its breakdown point. Smaller misfits might be due to one anomalous step or mineral age, or if you are using the MSWD option, an unrealistically small uncertainty on a step or mineral age.

If you are having trouble with fitting and the CRS portion of the model seems to be leading to overconvergence outside age-constrained regions, you should try several things. First, using the restart option, you can alter the CRS amplification factor (use small values like 1.05 to look for subtle improvements; use very large values nearing 2.0 to force wider exploration). Next you can then use Arvert7's post-processing option to exercise your best-fit tT paths to better delineate constrained and unconstrained regions. If you are in this situation, it is best to engage the post-processing option earlier in the run sequence, rather than letting the classic CRS routine create overconvergence.

After you do all this, you should then run a final model with higher-resolution convergence criteria for age spectra and mineral-age calculations, seeing if that helps you achieve statistical convergence (it probably won't!).

**One final important note:** NEVER become obsessed with just the time-temperature output! Be sure you keep an eye on how well the predicted age spectrum and mineral ages match your data. Arvert tries to minimize misfits but **Arvert will not protect you from a best fit that remains a bad fit!**

## 1.4 What's Different About Arvert 7.0?

---

Arvert 7 is based on version 6.1.3 but includes several very significant changes, bug fixes, and new features, some that break backward compatibility with older data files. Here are differences between Arvert 7 and older versions, loosely grouped into several categories:

### The inversion itself

*Reheating:* Like Arverts 5 and 6, Arvert 7 tries to prevent thermal histories from oscillating too much. In other words, the code only allows the specified number of reheating episodes in both the creation of original Monte-Carlo thermal histories, and later CRS daughters. Allowing some reheating is often an important component of nice and permissive inversions, but the approach used by earlier Arvert versions meant that silly multiple oscillations could occur and result in good-fitting but geologically implausible models.

Because of changes in the way time nodes are handled, if you allow reheating you may see a occasional warnings about violating the constraint on the number of “flips” – don’t worry about these: these warnings should be rare and represent small deviations stemming from the need to obey other constraints

*Time nodes:* Arvert 7 retains my newer approach to making random thermal histories at the model start. Initially, only a modest number of random time nodes are created in the Monte Carlo stage, and then a larger number of evenly spaced time nodes is used for the CRS stage. This helps to greatly reduce the dependence of the model on the exact location of time nodes, and produces more smooth and realistic thermal histories in the classic CRS stage.

There are some costs to this. Models run are slightly slower because with more modes, the chances of producing “illegal” CRS histories is higher, so that the flag for allowable CRS attempts had to be significantly increased, and this in turn means the code might churn for a while looking for a set of good histories. It also means that for some combinations of parameters, with reheating allowed, you still might have the code abort because too many attempts have been made to create a

legal thermal history. In that rare case, your options are to try again, allowing even more attempts at making a CRS history, or to modify your parameters controlling the number of nodes and the degree of reheating.

*Radiation-damage models for mineral ages:* Like versions 5 and 6, Arvert 7 includes Rich Ketcham's implementation of the RDAAM model for helium-diffusion kinetics in apatite (Flowers et al., 2009) and a variant (dubbed 'ZRDAAM') for He kinetics in zircon (Guenther et al., 2013). The user also has the option of using a volume-diffusion model with user-supplied kinetics. In the case of helium in apatite and zircon, the RDAAM models are the same basic codes include in the classic HeFTy model, circa 2014 (Ketcham, 2014)). Note that for apatite, the *traprmr0* parameter is set to 0.83, for "standard apatite" – for Durango-like apatites, change this to 0.79 (or whatever value you can justify) and recompile.

*Multiple samples with temperature offsets:* Arvert 7 now allows you to model multiple samples at once. You can have up to ten samples, each having one MDD age spectrum and up to 10 mineral ages. These samples are related to one another via constant temperature-offset values supplied by the user or generated by the code via the CRS algorithm. Thus, you can simulate samples from a borehole or elevation profile, or, if you set all the temperature offsets to be zero, you can simply model multiple analyses from one site or region, including multiple MDD spectra. The first sample is the reference sample that must have a zero offset. The other samples have their thermal histories uniformly offset from the reference thermal history. These offsets can be fixed as supplied by the user, or randomized and made part of the inversion if you are unsure of the offset (these can vary progressively as in a simple borehole, or deflect around the reference sample's temperature). At this point it is not possible to have the temperature offset evolve through time. Note that you can still run a classic Arvert single MDD sample!

*Data blends:* As a necessary part of addressing multiple samples, Arvert7 will work with any blend of samples having an age spectrum or not, and different mineral ages; no MDD age spectra at all are required. Currently, the code is still limited to minerals that can be modeled either as simple volume diffusion (with and without alpha distribution) or radiation-damage impacted volume diffusion (for zircon and apatite). Future releases of Avert may support  $^4\text{He}/^3\text{He}$  and fission-track data; support for CRH He-release spectra will be added if experiments with a  $^4\text{He}/^3\text{He}$  variant show promise.

## The code itself

Arvert 7 required a significant number of changes to handle multiple numbers of samples with differing numbers of minerals (including no minerals or spectra for some samples in a set). Among other things this meant additional loops to churn through samples, and changing arrays to handle the larger and more complex data spread across multiple samples. All of that required changes to function calls and prototypes. The sample post-processing block was another addition. Trying to be a little more efficient, for each MDD sample, the code now uses a single call to a separate `lab_lovera()` routine to calculate the laboratory  $^{39}\text{Ar}$  losses just one time. The `geolovera()` routine handles the evolution of  $^{40}\text{Ar}$  in nature for each thermal history. Finally, I also chose to replace the old plotting block with a routine to write out all needed plotting files, and created a complex gmt script that then is called to do the plotting (see below). This allows to plots to be recreated or altered after the inversion is finished.

Collectively, all those changes are one reason Arvert 7.0.1 should be viewed as something of a beta release. The testing I've done using just single MDD samples as well as blended data sets suggests that the core calculations are all ok. I'm just more concerned with non-fatal, subtle bugs that might occur with respect to weighting fits, etc.

## Input files

Many input files had to be changed in detail to handle the multiple-sample nature of Arvert 7 and its new features. Additionally, I slightly expanded the use of text prompts to make it easier to edit the `crs.in` and `mineral.in` input files. The other input files `goal.in` and `domains.in` do not have text prompts and just stack the multi-sample info into one file. Because of the changes, older Arvert input files will not be directly compatible with Arvert 7.

## Inversion output

Arvert7 no longer generates on-the-fly gmt plot commands created in C++ code. Instead, it writes required plot files to a “PLOTTING” directory, copies a gmt6-compatible shell script to the directory, and launches the script to produce a pdf plot. The decision to offload the plot script was partly due to the complex and variable output needed for displaying data from multiple samples, and also because I felt it was better to preserve the plotting data and script, which the user can then rerun and possibly modify. This means that along with input files when running Arvert, a copy of the plotting script (`plot_arvert.sh`) must be present. As

before, a working version of gmt5 or gmt6, plus ghostscript, must be installed in order to use this plotting info. Also, Arvert 7 no longer uses an **ornament.in** file to change plot parameters, as the user can simply make changes to the local plotting script to change the plot appearance.

Be warned that the plot script is very complex, so if you edit it, make changes carefully and keep an original copy! Also, because of the large amount of file information that must be parsed by gmt, bash math, and other UNIX utilities, the plot script can be slow if you have many samples with lots of data, and you chose the “full” plotting option that shows lots of lines and points.

Output files are now written with a file suffix of **.txt**, which in 2022 are easier to “open-with” using Excel (not so long ago, Excel would just open **.xls** files, but now this is a two-step process or more, so it’s faster to just use “open with” on text files).

## Bug fixes

In making the significant changes needed for Arvert7, I found several “non-trivial” bugs in Arvert 6.1.3. None of these seem to impact any ultimate CRS results themselves, but they do change how the code was working relative to the user’s expectations. I strongly recommend doing all future work in Arvert 7, which handles single samples just fine. Given the small Arvert user base, I’m not going to spend the time to patch older versions. Here are some of the bugs:

- In MontetT(), Arvert 6 was passing wrong values because it filled the upperlimits[] and lowerlimits[] arrays with ttpoints not rtppoints number of time nodes. Those xxxlimit[] arrays need to be built inside MontetT() because each MC history has different time nodes and so different limiting arrays.
- Arvert 6 code was not always correctly forcing the starting temperature point to be safe for lovera() (safe, meaning hot and open to diffusion at the start). If MDD data are present, this is now accomplished by setting the initial temperature to be 200 °C above the largest-domain closure temperature for all samples. If this value is lower than the user-specified first minimum temperature constraint, the first constraint is raised and a warning is issued.
- Arvert 6 (and probably earlier): The run-summary file incorrectly reports Do for each domain as  $Do/a^2$ , and in units of 1/m.y. This was corrected to report  $Do/a^2$ , in units of 1/sec.

## Section Two – Obtaining and Installing Arvert

---

### 2.1 Obtaining Code or Applications

---

Arvert source code, installation instructions, this guide, examples of inputs and outputs, and helper applications can be obtained at the Arvert repository at the GitHub site [OpenThermochronology](https://github.com/OpenThermochronology):

<https://github.com/OpenThermochronology>

Helper applications include an implementation of Oscar's Lovera's domain-finding algorithm, a simple implementation of his code to calculate the cross-correlation between MDD age spectra and R/R<sub>0</sub> plots, and a standalone implementation of the lovera() routine for calculating age spectra (useful for forward modeling).

I use and develop Arvert under macOS. Arvert 7 runs as a UNIX command-line application under macOS and should be straightforward to port to other \*nix platforms. If you wish to use Arvert on a different platform, your biggest challenge will probably be installing gmt and getting the plot script to run (see Section 2.3).

### 2.2 Overview of running the application

---

Arvert runs as a command-line program in a terminal window. Until computer speeds increase by another factor of 20 or so, Arvert can't really work as an interactive program, so lack of a GUI isn't a big deal.

To run Arvert, you invoke it in standard UNIX fashion. On MacOS, launch Terminal.app, navigate to the directory containing both the executable and all needed input and helper files, and type

```
. /arvert7 (assuming you named the executable arvert7!)
```

*In macOS, note that double-clicking the executable in the Finder will not work; you will cause Terminal.app to launch and the program to start, but it will terminate in a “bus error”.*

You can choose to place the Arvert executable in any directory of your choice that is in your PATH (type `echo $PATH` in the console to see what's currently there). In this case, you can launch Arvert from any directory just by typing its name. This will save you the hassle of having to drag a copy of Arvert into your

project directory. Just remember that if you recompile and update Arvert, you need to refresh this central copy!

When Arvert starts, you are presented with a list of what Arvert thinks your inputs are. You should check these for blunders or silly values to avoid wasting time. Arvert does use some assertions to trap bad input values, but there are still many ways to launch useless runs. Once you've checked your inputs, your only options are to abort the run or to proceed.

When the program is running, you can quit the application in the standard UNIX way (cntrl-C), but you will effectively lose your calculated data. Arvert is an ok citizen with respect to multitasking, and you can switch away from it to do other work. Note that Arvert is processor intensive and may slow down the operation of other tasks, depending on how many cores you have, so your game playing might not be as enjoyable. Likewise, if you load your cores with many other jobs, you will slow down Arvert's progress as well. However, on modern multi-core machines, you should have computing power to spare. Because several of the main routines in Arvert are not easily amenable to parallel processing, I have done nothing to take advantage of modern multiple-core CPUs, so you will almost certainly find Arvert dominating just one processor core, although this will depend on how smart your compiler and your OS are.

You communicate with Arvert using text files (See Section 3.1, below). An obvious hint is to work in a separate directory for each sample you want to model so that Arvert puts its output directories in a logical place and your different input files are kept organized. Note: Be sure that the input text files have the correct line terminators (UNIX-style: the line terminator is line feed).

If you use numerous mineral-age constraints for mutliple samples, Arvert 7 can be noticeably slower due to the added computations involved. If you run into this I suggest you use “good” level of precision for modeling, and then end with a single replicate that switches to “best” precision.

## 2.3 Compiling the Code

---

You will need to compile the Arvert source code to get a working program – this will produce an executable best suited to your specific machine. Much of Arvert was written in standard C (not ANSI strict), and then later modestly tweaked to compile under and take advantage of C++. The source code consists of several

code and header files that will have to be compiled and linked as a project. This should be straightforward.

As usual, several minor areas may require attention for a few users. There are some calls to standard timing routines that should work but could cause a glitch on some systems. These can be expunged without any trouble, or rewritten if you really want timing feedback.

Depending on the system, compiler, and any IDE that you are using, you may need to take steps to cause standard output using printf() to go to some sort of console window or the command line (under macOS, Arvert 7 just reports to the command line in Terminal.app).

Arvert's current release (Arvert 7.0.1) has been built and tested using the g++ compiler. Performance optimization can be a minefield: I have detected no degradation in accuracy using fairly aggressive optimization. Using gcc 13.0.1 I am currently compiling Arvert 7 as listed below. For those of you new to this, note that in addition to all the component files being present in your compile directory, the header files **arvert.h**, **ZRDAAM.h**, **nr3.h**, **ran\_pz.h** and **tchar.h** must also be present as well. Finally, to obtain plotting, you need to have the **plot\_arvert.sh** file present in the working directory that holds the Arvert executable and the input files. Good luck!

Note: the following compilation command is all one line

```
g++ ZRDAAM.cpp main.cpp averages.cpp fits.cpp  
heatsched.cpp mineral.cpp lablovera.cpp geolovera.cpp  
monteTt.cpp newhistoryCRS.cpp ran3.cpp selectsubset.cpp  
sort.cpp -o arvert7M2 -Ofast -march=native -w
```

As of this writing Arvert7 is compiling and running natively under M2 Apple Silicon.

---

## Section Three – Using Arvert

### 3.1 Program Inputs and Outputs

**Inputs.** To work, Arvert requires at four input files to be present in the same directory as the application; these text files must retain their exact names. When editing the files, it is best to use tabs not spaces to delimit any parameters placed on the same line (this makes pasting to and from spreadsheets easier). **To repeat some critical advice:** the input text files must have the correct UNIX line terminator, which is a simple line feed (LF) (some text editors refer to this as ‘UNIX’ format). Input files in other formats will not work and will lead Arvert to crash because it cannot read the input files correctly. Be sure your text editor can handle the proper line breaks and also does not add invisible formatting codes (e.g., do not use Apple’s TextEdit).

What follows is a lengthy discussion of the contents of the input files and, where needed, a discussion of what the parameters do and what ranges of values to use. A fundamental description of each parameter is given in bold face; additional comments on the parameter(s) are in italics. The four files are **crs.in**, **domains.in**, **goal.in**, and **minerals.in**.

In Arvert 7, the files **crs.in** and **minerals.in** have been changed from earlier versions and are not backwards compatible. The files are formatted as templates with text that helps you know which parameter is which. This helper text is read and discarded when Arvert runs. You **must** preserve the template text when editing these two input files, or Arvert will not be able to read your parameters correctly (the key is to have a space between the helper text and any values, and no white space (tabs or spaces) within the template text).

---

**(File 1) crs.in** – this file contains all the controlling parameters for the inversion, and is the file you will most often modify between runs. Except for the first line, which has no template text, you must retain the template text and the space after each entry. The file format is as follows:

- **String giving info about run (maximum 100 characters; no template text)** Just provides a record to help you distinguish between runs. The entire line is read and passed along to output.
- **File suffix appended to output files (maximum 10 characters)**  
Not critical but very helpful in sorting output from different runs. Used as a file suffix.
- **Number of samples being modeled**  
Minimum of 1, maximum of 10. Value entered here guides how many data blocks are needed for some parameters in this and other files.
- **Temperature offsets between samples**  
The first value is for the reference sample and must be 0.0. As many lines are needed as there are samples being modeled. Values are needed even if Toffsets are later allowed to vary programmatically.
- **Minimum and Maximum values for T-offset, °C**  
These values limit the range over which new T-offsets are made. See next entry for how these values are interpreted.
- **Type of T-offset creation**  
Option #0 means that the input values of T-offset are fixed across all models.  
  
Option #1 means that progressive T-offsets change monotonically. The max and min offset apply to the increment between any two successive samples. Arvert tries to use a CRS-style algorithm to generate values. It's not clear how effective this parameter is in how it trades off against the actual thermal histories.  
  
Option #2 means that progressive T-offsets can fluctuate ± around the reference value. The max and min offset apply to the entire possible range that the T-offset can fluctuate across; the values apply equally to all samples.  
  
Option #0 means you have a firm idea of what, say, the effective geothermal gradient was for a borehole. Option #1 means that you have a sense for what trends T-offsets should have, but you don't know if the offsets were constant or variable or what their size was. Option #2 means you are

*allowing samples to have been warmer or cooler than the reference, say for example, when you have multiple surface samples across a locality and you aren't sure the current distribution at the surface matches their past distribution with depth.*

- **Model duration in millions of years**

*You must be sure to leave adequate model time before initial closure to diffusion, especially if you wish to permit reheating. Otherwise, your model results may be unduly influenced by your chosen starting constraints. On the other hand, it would be silly to run a 1000 m.y. model for a sample that was only 10 Ma in maximum age, as this would waste compute time and also lower model resolution because time nodes would be wasted outside the region where the sample has constraining power. Warning: you need to make sure that the model duration and the time of the first explicit temperature constraint are the same.*

*Unless you have data such as a U-Pb mineral age to constrain your sample's higher-temperature history, you might want to leave at least 50 m.y. between the start of the model run and the time of first closure; more if you are constraining cooling rates to be low. Tip: Because Arvert always uses 0 Ma as one of its model boundaries, if you are modeling a very old sample and know that more recent thermal events are unlikely, you might consider using a static age shift with your data, lowering all the ages in your goal age spectrum by the same amount. This will help focus the model on the period of interest to your age spectrum. However, if you do this you will need to keep the absolute uncertainties on the ages unchanged.*

- **Number of Monte-Carlo and CRS time nodes (MC: minimum of 5, maximum of 20; CRS: minimum of 20, maximum of 100)**

*Arvert creates and uses time nodes in two stages. In the initial Monte Carlo stage used to build the starting thermal-history pool, Arvert fixes the first and last nodes to the model start and end, and then generates random time nodes in between for each member of the pool. (This is a change from previous versions 5 and lower). This introduces more diversity into the pool, which can be important because insufficient sampling in time can lead to biases associated with sparse, fixed time nodes.*

*To prepare for the CRS and post-processing phases, Arvert transforms the Monte Carlo thermal histories into new forms that have a higher number of regularly spaced CRS time nodes. The CRS and post-processing algorithms*

*use those. The advantage is that with higher numbers of nodes, thermal histories are smoother. However, because the nodes are regularly spaced, you should be aware that using too-few CRS time nodes can lead to problems in fitting, since thermal histories will have only limited points at which to inflect.*

*Using between 10 and 20 MC time nodes and at least 25 to 50 CRS time nodes seems to work well. You may want to experiment with one or both of these if you are not happy with the form of the final CRS pool or if you are having trouble with fitting.*

*Be aware that as the number of CRS time nodes climbs, it becomes increasingly difficult for Arvert to make CRS histories that completely satisfy the implicit constraints. Arvert tries up to the specified number of times to make a legal history, and then gives up and aborts your run. In general, when the number of time nodes increases, the program will sometimes slow a little as it churns through possible histories in order to find a legal one.*

*A few more notes about time nodes and thermal histories. To try and minimize progressive bias when generating the Monte-Carlo histories, Arvert uses an alternating-direction approach developed by Sean Willett. Rather than start from one end and make temperatures at sequential time nodes, Arvert first chooses a time node at random and then works up and then down as it makes each history. Because the Lovera algorithm for age-spectrum prediction requires thermal histories to start above the closure temperature of the largest domains, Arvert checks to be sure the initial temperature constraint is 200°C above the closure temperature of the largest domain, overriding user input if this is not so. Also, Arvert tries a bit to honor your flip criterion when building the MC pool, but it is not slavish about this, so you might see an extra slope reversal at times.*

*Finally, be **VERY** aware that the constraints you place on the initial pool can produce patterns that survive into the CRS analysis and bias your results, or at least their appearance (e.g., disallowing heating in the Monte Carlo pool tends to produce sigmoidal histories because once a history gets cold, it can't climb back up). Therefore, it is wise to try starkly different initial pools so see if their shapes matter to the final inversion; it is especially useful to feed the CRS algorithm the opposite set of initial trends from what it is tending to predict: e.g., if you are getting predictions of fast cooling, limit the Monte-Carlo pool to only slow cooling, thus forcing the*

*inversion to engineer the fast cooling rather than simply adopting it. Finally, remember that the survival of initial trends is most evident in the regions outside those that are constrained, so when depicting Arvert results it is often advisable to mask out unconstrained regions to avoid consumers of your model picking up this possible visual bias. See examples at the end of this document.*

- **Number of constraining brackets for thermal histories**

*You can place explicit constraints on portions of the problem space by specifying permissible temperature ranges at specific times (between these specified times, Arvert just extrapolates linearly). You must place at least two constraints on the model (start and end), and no more than 10. The first and last constraints must be at the model start and end (at times <modelduration> and 0 m.y.). If you don't want to enter any constraints, then simply choose to create two and make their minimum and maximum temperatures be something like 0 °C and 600 °C.*

- **N constraints, format: <time> <min Temp °C> <max Temp °C>**

*Enter as many rows of constraints as you have chosen, using the format given above. Be sure that the time values decrease progressively. Also, be sure that the constraints don't conflict with the implicit (rate) constraints you specified, or Arvert will get confused. Remember that Arvert interpolates linearly between explicit constraints, so you might need more constraints than you think to define your constraint-space (e.g., if you have a "constraint-box" world view).*

- **Maximum heating and cooling rates for Monte-Carlo histories**

*You must specify what the maximum values for heating and cooling rate are during the generation of the initial Monte-Carlo pool of thermal histories. To rule out either cooling or heating, enter a value of 0.0 (the units of this parameter are °C/m.y.). The permitted range is between 0.0 and 1000.0 °C/m.y.*

*Note that for typical model runs of 10 m.y. or more with upper constraints of 500 °C or so, you will never be able to use a rate of 1000 °C/m.y., since this would be equivalent to only 0.5 m.y. in time, and for older models Arvert won't distribute time nodes with that resolution.*

*Let me repeat an important point I made above. Allowing high rates during the Monte Carlo routine can introduce a potential bias into the model,*

*especially if you rule out, say, heating, but do allow fast cooling. What happens is that whenever Arvert makes a low-temperature value, the history becomes trapped at low temperatures because heating isn't permitted. Thus, Monte-Carlo histories generated under such a model will be dominated by those that plunge to low temperatures, and the CRS pool will tend to inherit this sigmoidal shape. It is probably healthiest to make the inversion find the rate you suspect, rather than pre-supplying it with such histories. A useful technique is to run models in which the initial Monte-Carlo pool is run once allowing only low rates, and then again allowing high rates, and then see if the CRS results agree. (see also Section 3.5, below).*

- **Maximum heating and cooling rates for CRS histories**

*You must also specify heating and cooling-rate constraints for the CRS histories. These can be identical to the Monte-Carlo values if you prefer, or not. Note that for runs having many time nodes, these implicit rate constraints can slow the model because it becomes hard to generate a new legal CRS history (see discussion of time nodes, above).*

- **CRS amplification factor (usually 1.1 to 1.5)**

*This parameter controls how aggressively Arvert searches parameter space because it determines how much amplification is used to generate a new CRS thermal history (see discussion in Section I, and Figure 1). Typical values are between 1.1 and 1.5, with values of 1.1 producing a subtle model that converges more slowly, and values of 1.5 producing a rather noisy model that in some cases converges more rapidly. Both sorts of values can be useful in exploring subtleties or stirring up a better search of temperature space. Note that you are allowed to enter amplification factors between 0.5 and 2.0, but values below 1.0 will collapse the inversion and very high values tend to slam new histories up against the explicit constraints, or violate the implicit constraints.*

- **Number of histories in subset (min 5, max 50) and in main pool (max 300)**

*Interplay between these two seemingly simple parameters can change the course of the inversion. Consider that if the pool size is very large compared to the subset, convergence will take far longer because there are simply more histories to churn through and replace, plus the subset average will less closely represent the pool average. **The latter is an important point, as this interplay controls the tension between random exploration, learning, and convergence in this algorithm.** Too much randomness and the model*

*will converge too slowly, but too much learning and the model will falsely converge because all available variance will be expunged from the pool (consider the degenerate case where the pool and subset size are the same: such a model must collapse to false convergence).*

*The subset is constrained to be within the range 5 to 100 and less than a third the size of the main pool. The main pool can have as many as 300 members, but must be at least three times greater than the subset size. I'd suggest starting out with about 10:1 ratio between pool and subset sizes.*

- **Fitting cutoff** (the meaning of this value depends on the nature of the fitting method established by the next input)

*In theory, the value of this parameter will terminate the model and flag that it has converged (this rarely happens in practice!). Note that unlike some other thermochronology inversion schemes, Arvert tries to get all thermal histories in the pool to fit, so even if the model does not converge, there still might be a number of good-fitting histories.*

*There are two main options for this parameter, with the first having two variants.*

*The mean-percent deviation is simply the average, over the fitted heating steps and mineral ages, of the absolute values of the percent deviations between observed and measured ages. It is simple, and most of the original testing and development of Arvert used this parameter. One drawback is that it can be hard for the model to reach very low values because even relatively minor areas of mismatch can pollute the average fit: fits that are overall pretty good on visual inspection may not yield low values, as misses both above and below the goal spectrum accumulate. Another more major drawback is that the uncertainty on the age is not taken into account. As a variant, you can specify a percent-deviation fit that is weighted towards the early steps in the age spectrum, which in some cases can record more of a sample's thermal history. This option has not been tested very much, so be (very) wary.*

*The other available fitting option is a type of **mean square of the weighted deviates** (MSWD) (reduced chi square). MSWD is commonly used by geochemists when comparing observed and predicted data, as in regression or determination of means. For Arvert 7, MSWD is calculated by summing the squared differences between the model age spectrum to the goal age*

*spectrum at each step, weighting each term by the reciprocal of the variance, and then averaging by dividing by the number of fitted steps (not N-1, since we are not using the data to get a mean age) If there are mineral ages, these are included as well, with each age having the same weight as a single MDD step. MDD steps are NOT explicitly weighted by size, since due to the nature of diffusion and age spectra, early small steps have as much or more tT power as later larger steps that reflect mostly the stabilization of the  $^{40}\text{Ar}$  and  $^{39}\text{Ar}$  fluxes. There is possibly some implicit MDD weighting in that larger steps will have larger uncertainties compared to very small steps.*

*Arguing by analogy with MSWD as used for geochemical data, a value of about 1.0 means that the deviation between model and goal spectrum and mineral ages is just consistent with the assigned uncertainty, and values greater than 2-3 would indicate the model is most likely a mismatch. So, using a value of 3 as the cutoff criterion would mean that the model would terminate when the worst-fitting thermal history was just about acceptable. That seems like a reasonable way to proceed. Such MSWD rules-of-thumb are not that rigorous and true confidence levels require that the number of steps be taken into account. For our purposes, MDD work usually involves fitting 15 or many more steps, so the rules-of-thumb won't be too bad.*

**Important Note:** you need to specify the internal absolute uncertainty on each step if you plan to use the MSWD option for just a single MDD sample. Do not include the uncertainty due to the J-factor or other systematic errors, because you are looking at the internal systematics of the sample. However.... if you use mineral ages as an additional constraint, or multiple samples, recognize that total uncertainties might be more appropriate. Arvert 7 ignores this issue and leaves it to you to manage.

*Note that both types of fitting parameters apply to the overall fit. It is possible to have good fits over large parts of the age spectrum, and just a few localized mismatches. It is up to you to decide if this is something you will view as only a second-order glitch, or as a sign that some assumption about MDD domain structure or correction for excess Ar has been violated. One thing you can do is run subset models in which you fit just parts of the age spectrum. However, I strongly urge moderation in trying this: do not arbitrarily cut out parts of the goal spectrum, other than to focus on the low or high-temperature parts, unless you have documented reasons why a particular step reflects an analytical glitch.*

To allow you to see where misfits are happening, Arvert writes out a file called **aa-stepfits.txt**. This file lists the individual misfits for each age-spectrum step and if used, mineral age(s). Particularly for mineral ages, if one age is giving a very large misfit, you will have to decide how to handle this. Removing this could greatly improve the inversion since the it will not be struggling to deal with this point, but it does mean you are culling data. I recommend that you report the age, and then make clear on what grounds you are omitting it, if you do omit it..

Typical values for mean-percent deviation might be 1-2%, whereas values for MSWD would be between 1 to 3. In many cases, the scatter due to “geological” parameters and age dispersion mean that you will not get convergence to such values (based on analytical uncertainties alone).

- **Type of fit – (0 – early-weighted mean-percent deviation (mpd); 1 – unweighted mean-percent deviation (mpd); 2 - mean square of weighted deviates (mswd))** This parameter flags the type of fit to be used in the CRS algorithm, and in testing for convergence (see discussion immediately above). For Fit Type 0, you can specify the cutoff that defines “early” in the spectrum and the weight to give steps below this cut-off (relative to a weight of 1.0 for each step in the rest of the spectrum); these two parameters are specified later in this input file.
- **MDD diffusion geometry – 1 for sphere, 2 for infinite-slab**  
This is an essential parameter, but not one that will change from model to model. I repeat: it is **essential** to get this right, as you must choose the same geometry here that was used in determining your sample’s kinetic and domain information! **If you do not understand this point you should not be using this code.**
- **Maximum number of attempts to make a new CRS history**  
A value of 20000 should be enough but if you are running a sample with more CRS time nodes and that has reheating and tight constraints, you might need to increase this value.
- **Temperature step in °C for discretization of temperature histories**  
This parameter determines how the Lovera forward model breaks apart thermal histories for its use (the routine does this at regular temperature intervals rather than regular time intervals, to ensure adequate

*discretization during periods of rapid heating or cooling). A value of 10 or even 20°C will work for routine and quick models, but you will want to reduce this to 1 or 2°C for final runs. Permissible values range from 1°C (slower, better resolution) to 20°C (faster, poorer resolution). Note that if you are modeling over long Gy intervals, Arvert may spit up at you if there is a large temperature oscillation and you are discretizing at a fine interval like 1°C, because in this case you are producing too many calculation time nodes. The workaround is to increase this parameter until Arvert runs.*

- **Fractional cut-off for terminating infinite series in Lovera routine**

*This is a convergence criterion for the slow-converging series involved in the Lovera algorithm, expressed as a fractional change in successive values. The series converges **very** slowly for the low values of  $Dt/a^2$  associated with early heating steps. For routine assessment work you can keep this value as high as 1e-3 (0.1%). However, for good accuracy for all steps and adequate coverage for small steps, a value of 1e-7 or smaller is recommended. The permissible range is 1e-3 (fast, less accurate) to 1e-8 (slower, more accurate). Even then, it is possible that if you specify extremely small fractional losses for your first step or two, the lovera() routine will not have converged or will have reached the double-precision limit, so in such cases keep an eye on the ages and losses calculated by the model for such tiny early steps. However, it's unlikely you'll be doing useful work with such small initial heating steps.*

- **Flag for number of reports during run: 0 for short, 1 for verbose**

*Arvert will always report its progress to the console every 500 CRS histories, and every 500 CRS histories it will update an interim report file call <CRSRolling> that captures the current pool of thermal histories. If you set this report flag to verbose mode, Arvert will issue a progress report every 50 CRS histories, and Arvert will also write several additional thermal history files at the start of the run, which are informative about the early convergence of the model. This will happen whether in ‘restart’ mode or not. To reduce the files Arvert generates, choose the ‘short’ mode.*

- **Flag for generating plots with gmt: 0 for no, 1 for yes, 2 for yes with Monte Carlo**

*If you have version 5+ of gmt installed (The Generic Mapping Tools; <https://www.generic-mapping-tools.org/>), you can opt to have Arvert 7 create summary pdf plots of your final age-spectra, mineral-ages, T-offsets,*

*and the reference-sample thermal-history pool. This saves you the trouble of having to open Excel and manually plot each file, which can get VERY tedious when you are exploring different parameters and doing lots of runs. The pdf plots themselves should open gracefully in vector image-editing programs.*

*Option #0 creates no plots. You're nuts to choose this since reviewing the data will be tedious. If you have your own plotting solution, better to let Arvert generate the plot files so that you can at least work with those.*

*Option #1 creates a simpler set of plots than the full Option #2, but just plots the best and worst histories, spectra, and mineral ages. This makes plotting faster and less clutter.*

*Option #2 plots the initial Monte Carlo tT pool under the thermal history plot. This makes for a busier graphic but is recommended at least for trial runs to get a feeling for how the final solution and initial MC pool are related. This option also plots all members of the age-spectra and mineral-age pools. It's LOTS slower!*

*On the plots, the temperature axis is scaled to the highest temperature constraint you entered. The age axis is scaled to accommodate the maximum good goal age (i.e., fitted step or mineral age). Using the defaults, the best 15 models are shown in green, the worst 15 models are shown in red, and all the others are shown in the medium blue (they are shown only if you request "full" output). For the measured age spectrum, the fitted steps are shown in blue. The model age spectra are color coded like the thermal histories. Finally, if you selected plotting of the Monte Carlo pool, these histories will be shown in faint gray on the thermal-history plot. Uncorrected measure mineral ages are shown as blue dots; predicted mineral ages follow the same color scheme as age spectra and the tT plot. On full-output plots, the regions shaded in pale blue and green show the ranges over which age spectra and mineral ages provide a direct constraint.*

*Keep in mind that the significance of this color scheme is different than that used in, for example, the classic HeFTy code or David Shuster's  $^4\text{He}/^3\text{He}$  inversions – these more simple Monte-Carlo inversions record all trials and use color to display statistically "acceptable" and "good" fits.*

You can specific how many of the best and worst fits are plotted in color (see below).

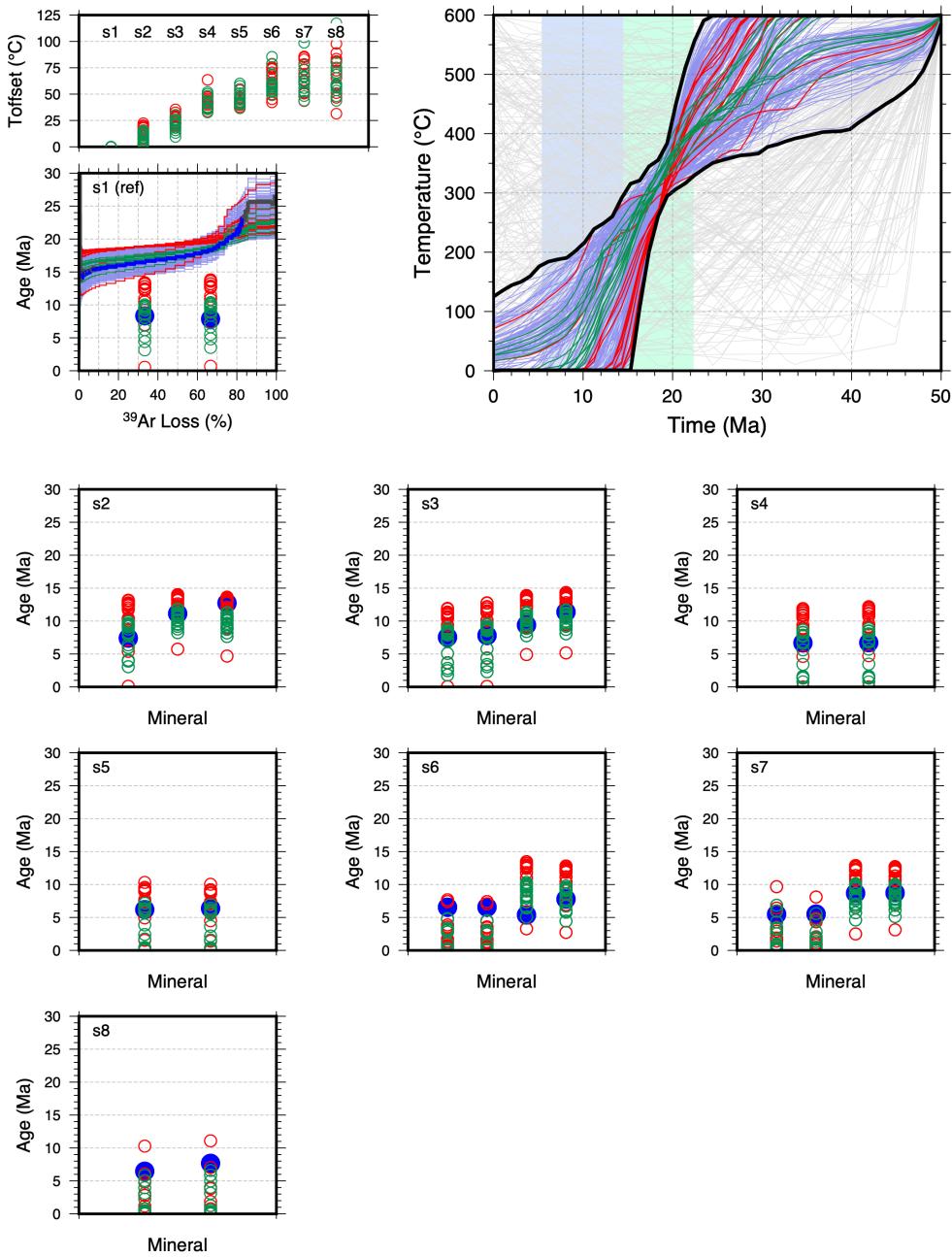


Figure 3. Example of Arvert7 “full” output for a model having multiple samples and multiple age types.

- **NBEST:** number of best-fitting histories to plot in color (green)
- **NWORST:** number of worst-fitting histories to plot in color (red)

- **FLIPS:** number of reheating events that are allowed

*Advanced option. A value of '1' means that one reheating peak is permitted in the thermal history. The code to do this is annoying and complex, and has not been deeply tested for values above 1. The idea is to trim out geologically unreasonable oscillating thermal histories. Modify at your own risk (meaning, keep a close eye on what is happening if you change this).*

- **EARLY\_STEP** and **EARLY\_WEIGHT:** cut-off below which MDD steps will receive extra weight, if fitting option '0' is chosen, plus weight for early steps located below cut-off

*Values are in fractional  $^{39}\text{Ar}$  loss, and values relative to 1.0 for a higher-loss step or a mineral age. **IMPORTANT:** even if you do not have MDD data for all or any samples, and even if you are not choosing Option #0. you need to have as many dummy lines here as there are samples.*

- **Use minerals?**

*Flag that determines how mineral ages will be used.*

*Option #0 turns off use of minerals. Option #1 turns on their use. Option #2 turns on use of mineral ages but turns off use of any MDD data.*

*Not that unlike previous versions of Arvert, in Arvert 7 you do not need dummy mineral or MDD files if you turn off mineral ages or MDD spectra – the code should skip around appropriately based on this parameter.*

- **Number of iterations for the CRS algorithm**

*Most runs will terminate based on this parameter. Resist the urge to choose too many iterations until you are ready, as you can always keep restarting the model after having a quick check of results. Usually, a total of 5000 to 20000 iterations will be sufficient, and values of 1000 to 2000 iterations can be useful to check model setup or cue up the post-processing algorithm.*

*For RESTART option 2 (post processing; see below) if you enter other than a '0', Arvert 7 will first do additional CRS iterations before shifting to post-processing mode.*

- **Post-processing iterations**

*Number of iterations for post-processing. Only matters if RESTART option 2 is chosen*

- **Post-processing subset size**

*Number of best thermal histories from the pool that will be post-processed. Constrained to be more than ‘1’ and no larger than the full poolsize. Start with modest numbers like 10 to 25 to focus the post processing on the best fits determined by the CRS algorithm. Low values means you will be exploring just a few better fits; higher values means you will be exploring using more of the pool – this will be keep in some poorer fits but might also retain a bit of diversity in the exploration.*

- **Range of post-processing temperature deflection (°C)**

*Maximum range over which starting and ending temperatures are deflected (can be above or below original value; range symmetrically spans the original temperature).*

*Larger values explore more but lead to more frequent failures; smaller values explore less but are better finding improved histories.*

- **Restart option –**

**0: start from fresh Monte Carlo pool followed by CRS processing**

**1: restart from previous CRS pool for additional CRS (and T-offset) processing**

**2: restart with previous pool, using additional tT (and T-offset) “post-processing”**

*A value of ‘0’ begins a fresh model run that includes generation of Monte-Carlo histories as a starting point, followed by the specified number of CRS iterations. This is the basic “classic” mode for Arvert.*

*A value of ‘1’ means that the model will restart using the state of the CRS pool when the model last ended. This option allows you to run the model in stages, and with care, make changes to several inversion parameters as you go along. Of greatest interest, you could make changes to the implicit and explicit constraints, the amplification factor, fitting criteria, and the controls that govern the operation of the Lovera routine (see below). For example,*

*you could run a faster set of runs using a modest amplification factor, then increase the model accuracy or amplification factor.*

*Note that you cannot change most of the other model parameters, like model duration, time nodes, pool size, etc. Such items impact array sizes and the nature of the data in the restart file, and changes in them will produce erratic behavior or most likely, a crash. Similarly, you can't make changes to the goal-spectrum file or the domain-structure file. This version of Arvert provides **NO PROTECTION** against making such fatal changes, so it is up to you to use the restart option with care!*

*A value of '2' means that the model will restart with the previous CRS pool, but then use the new post-processing algorithm for the specified number of additional iterations. If relevant, T-offset processing uses a CRS-style algorithm. **Important:** if you just want to do post-processing, remember to set the number of CRS iterations to 0! Otherwise, the code will first do more CRS iterations and then do post-processing. I have not tested what happens if you alternate CRS and post-processing!*

***Post-processing honors the initial choice of how T-offsets will be treated. If you start with them fixed, you should stay that way across various restart selections. The same goes for the random options for T-offset. There is no protection against mixing and matching, so please avoid that!***

*Comments on these updated restart options.* This is probably the most untested part of Arvert 7, and an area where user savvy is really required. You can definitely end up doing lots of wasted runs and falling into a rabbit whole if you don't think about what you are doing. That said, used carefully, I think these options offer some real improvements.

*Here's how I think you might typical use this set of options, once you have explored the model with some quick initial runs.*

1. Use options 0 and then 1 to run a standard model for a few thousand iterations.
2. Deploy the post-processing option for a few thousand iterations; repeat if desired.
3. Repeat steps #1 and #2 with better levels of precision.

The following is a valid Arvert 7 input file, **crs.in**

```
Kohistan-SN 40C gradient
SUFFIX kohsn
NUM-SAMPLES      8
***TOFFSET       0
***TOFFSET       3
***TOFFSET       5
***TOFFSET       10
***TOFFSET      14
***TOFFSET      26
***TOFFSET      36
***TOFFSET      43
min-offset_max-offset   0.    20.0
Toffset-0fixed_1mono_2random 1
MODEL-DURATION    50
MC-TIME-NODES_CRS-TIME-NODES  5 50
CONSTRAINTS       3
***TIME_minTEMP_maxTEMP 50 500 600
***TIME_minTEMP_maxTEMP 45 0 600
***TIME_minTEMP_maxTEMP 0 0 600
MONTE_maxHEATRATE_maxCOOLRATE 25.0 25.0
CRS_maxHEATRATE_maxCOOLRATE   0.0 150.0
AMPLIFICATION     1.3
SUBSET_POOLSIZE   30 250
FIT-VALUE         3.0
FIT-TYPE-0wmpd-1mpd-2mswd   2
DIFF-GEOM-1sph-2slb        2
MAX-CRS-TRIALS          20000
DISCRETIZATION-TEMP       2
LOVERA-CUTOFF            0.0001
REPORT-FLAG-0short-1verbose 1
PLOT-OPTION-0no-1short-2full 1
NBEST 15
NWORST 15
tT-FLIPS-ALLOWED 1
***EARLY-CUTOFF_WEIGHT 0.10 2.0
***EARLY-CUTOFF_WEIGHT 0.10 2.0
```

```

***EARLY-CUTOFF_WEIGHT 0.10 2.0
USE-MINERALS-0no-1yes-2noMDD      1
CRS-ITERATIONS          2000
POST-PROCESSING-ITERATIONS  0
POST-PROCESSING-SUBSET 50
POST-PROCESSING-RANGE 400.
RESTART-0no-1CRS-2ptT   0

```

**(File 2) domains.in** – this file contains the domain structure of any MDD data in your sample(s). Once you have created it you normally would not modify this file during a series of runs. You will need as many sequential blocks as you have samples, in a format is as follows:

- **number of diffusion domains**

*The one comment to make here is that is has been shown by Lovera and others that having “extra” domains does no harm, but skimping creates problems. So, be sure that you have adequately analyzed your sample’s Arrhenius behavior and include sufficient domains to account for subtleties in its R/R<sub>0</sub> plot. Let me reiterate this: you must properly characterize your sample’s domain distribution or Arvert will not give reliable results. The data in this file **must be derived for the same diffusion geometry** you declare in the **crs.in** file.*

*If the you do not have MDD data for a sample, enter ‘0’ for the number of diffusion domains, and do not enter any other data for this block.*

- **groups of three lines giving the kinetic parameters for the number of domains specified. The sequence is activation energy (kcal/mol),  $\log_{10}(D_o/a^2)$ , and volume fraction.**

*The number of groups must equal the number for domains. Be sure to use the 10-based log of the frequency factor ( $D_o/a^2$ ) for each domain. The volume fractions should total to 1.0.*

The following is a valid Arvert input file, **domains.in**. It describes an eight-sample model in which the first sample has MDD data (with 10 domains) and the other seven samples have no MDD data.

```
10
46.9961
5.5653
0.097056
46.9961
4.8533
0.103870
46.9961
4.6554
0.104187
46.9961
4.2134
0.099705
46.9961
3.9946
0.102323
46.9961
3.3642
0.085057
46.9961
3.3618
0.091641
46.9961
2.2401
0.075573
46.9961
1.6076
0.061845
46.9961
0.9416
0.178742
0
0
0
```

0
0
0
0

---

---

**(File 3) goal.in** – this file contains the measured age spectra of your sample(s), which will serve as a goal for the inversion. This is another file you normally would not change during a series of inversion runs. If you are just modeling minerals ages and have no indicated that you have no age spectra, you can omit creating this file.

The format is as follows:

- **number of heating steps, less one**

*By convention, the last heating step for an age spectrum reaches a fractional loss of 1.00, for which one cannot do diffusion calculations. So, if you have measured 48 heating steps, you would enter 47 for this parameter.*

- **N steps, format: <fractional loss> <age (Ma)> <error in age (Ma)> <fitting flag, 0=no 1=yes>**

*Remember to express the  $^{39}\text{Ar}$  loss as a fractional loss, not a percentage loss. Age and error in age are obvious; if you choose to use mean-percent deviation than age error is not important but a placeholder value still needs to be entered. Note that you should specify the age error without the error in the J-factor if you are modeling a single age spectrum as the uncertainty in J is systematic for one age spectrum. If you are modeling several spectra and mineral ages, it makes more sense to include the uncertainty in J for any spectra.*

*The fitting flag allows you to indicate which steps of an age spectrum should be used by the inversion. You need to omit certain steps early in the age spectrum if there are problems with excess or fluid-inclusion hosted argon, and late in the release you **must** omit steps in which Ar was released by partial melting, not diffusion. Or, you can specify a block of steps early or late in an age spectrum to focus Arvert on just a particular portion of the thermal history. If you do this, you should have a rationale for omitting steps and some consistent set of criteria, e.g., the step has extraction temperature above 1150 C, step is first step of isothermal replicate, etc. Don't farnarkle around and fall into the trap of cherry-picking steps that "look good."*

*Simply flag any steps you want to omit with a ‘0’. Only those steps flagged with a ‘1’ will be used in Arvert’s fitting routines.*

*If you have more than one sample, you can just stack the goal spectrum data sequentially, just ignoring samples with noMDD data. Arvert 7 uses the domain input to record whether a sample has MDD data or not*

The following is a valid Arvert input file, **goal.in**: (this just shows a single short spectrum).

16			
0.0015	5.31	1.1	1
0.0165	5.63	1.0	1
0.0423	5.78	0.8	1
0.0473	6.02	0.6	1
0.0923	6.08	0.6	1
0.1022	6.13	0.6	1
0.1053	6.89	0.6	1
0.1141	7.59	0.8	1
0.1367	9.02	0.9	1
0.1882	11.86	0.9	1
0.2955	16.38	1.1	1
0.5023	21.32	1.2	1
0.7833	23.35	1.1	1
0.9604	23.46	1.1	1
0.9986	23.46	1.8	0
0.9999	23.46	2.3	0

---

---

**(File 4) minerals.in** – this file contains information about mineral ages that you want to add as a constraint. The Arvert 7 format is as follows; it uses a template format like **crs.in** so be sure not to overwrite or delete the template info. This file can be quite lengthy if you have multiple samples with multiple minerals.

The file must have MINERALS blocks equal to the number of samples. If MINERALS is set to 0 because the sample has no mineral info, then you can omit the complete mineral info for that sample – no dummy data are needed.

Here is what is required for each compete block.

- **number of minerals in the sample (0 to 10)**

*The number of mineral analyses for this sample for which you have a date. You can have multiple ages for a particular mineral; this makes most sense if you analyzed different grain sizes or different eU values in the case of radiation-damage models.*

*There follow four data lines for each analysis, unless the number of minerals is 0.*

- **Mineral analysis type (0, 1, 2, 3, 4, 5)**

*Type of analysis:*

*0 – He in apatite, volume diffusion, alpha-loss included*

*1 – He in zircon, volume diffusion, alpha-loss included*

*2 – He in titanite, volume diffusion, alpha-loss included*

*3 – other mineral, volume diffusion, no alpha-loss included*

*4 – He in apatite, RDAAM, alpha-loss included*

*5 – He in zircon, ZrDAAM (Guenther et al.), alpha-loss included*

*Use option 3 for argon and U-Pb mineral data. If you want to use a radiation-damage model without alpha-ejection correction, or a different mineral that requires a different stopping distance, you will have to modify the code and recompile.*

*The stopping distances used are from Ketcham et al. (2011). Radiation-damage code provided courtesy of Rich Ketcham.*

- **Sample age, and error in age (m.y., IMPORTANT: not alpha corrected)**

*Arvert solves the accumulation-loss function with the effects of alpha ejection included, because alpha-ejection modification of the diffusion profile could be important in some special cases, as for a sample lingering in the partial retention zone; the radiation-damage routines do so as well. Therefore, its direct output is an uncorrected age, and this is also what is directly measured for unknowns. Therefore, it makes most sense to perform the inversion using the*

*uncorrected age as the constraint. The only problem with this is that is harder to relate the uncorrected age directly to the thermal history, at least by eyeball. Get used to it.*

*If you are modeling a non-helium mineral with no alpha loss, then obviously you just use the measured age and uncertainty.*

*In its volume-diffusion calculations for minerals, Arvert just uses the alpha-amplified  $^{238}\text{U}$  decay constant, even for non-U-He minerals. This should not be an issue except for long-duration models where the sample spends a long time in the PRZ. If this really bugs you, edit the decay constant in the mineral.cpp file. Also, in volume-diffusion models (options 1, 2, 3) Arvert only models ejections using  $^{238}\text{U}$ , assuming  $^{232}\text{Th}$  and  $^{147}\text{Sm}$  to be zero. Someday maybe I'll improve this. Or not.*

- **Weighting factor for mineral-age constraints (dimensionless; generally 0.5 to 2.0; could be higher)**

*When calculating fits to the observed data, Arvert averages fits for each age-spectrum step and for any mineral ages, each weighted equally. If you want to weight a mineral age's impact on the total fit more heavily (it has more influence), specify a value greater than 1.0; if you want to downweight the impact of the mineral age, use a value of less than 1.0. A value of 1.0 means that the mineral age has the same impact in the total mean fit as a single heating step. What this does to any statistical value of the reported MSWD option is murky at best.*

- **Grain radius, spherical (microns)**

*All mineral-age calculations in Arvert 7 use spherical geometry. Following the recommendation of Ketcham et al. (2011), it would be best practice to supply the radius of the Ft-equivalent sphere, but an alternative would be to supply the radius of a sphere having the same surface-to-volume ratio as your analyzed sample.*

- **Activation energy (kcal/mol) and diffusion coefficient ( $\text{cm}^2/\text{s}$ )**

*Arvert's diffusion routine uses spherical geometry. The kinetic data you supply must have been derived using this geometry. Make sure you supply  $D_o$ , not  $D_o/a^2$ : if you have actual kinetic data for your sample that involves the frequency factor, you will have to break out apart your  $D_o/a$  value to obtain separate*

*values for diffusion coefficient and size. These data are required as placeholders even if you are using a radiation-damage model.*

- **U, Th, and Sm values in ppm**

*These values are used in the radiation-damage models; these data are required as placeholders even if you are using a volume-diffusion model.*

- **Precision (0 = good; 1 = better; 2 = best). Once only, at end of all blocks of mineral info for all samples.**

*This sets up, HeFTy-style, the precision for volume diffusion and radiation-damage models. For damage models, good precision involves 128 distance nodes, better precision involves 256 distance nodes, and best precision involves 512 distance nodes, the coarseness of temperature steps also varies. For volume diffusion, good precision is 100 and 100 distance and time nodes, better is 250 and 250 distance and time nodes, and best is 1000 and 500 distance and time nodes. Best-precision calculations can be quite a bit slower, so the wisest choice is to use this precision only for final runs.*

The following is a valid Arvert input file, **minerals.in**: for two samples, the second of which has no mineral data.

```
*****MINERALS      2
**MINERAL          4
UNCORR-AGE_ERROR   8.33  0.23
WEIGHT             2
RADIUS-SPHERICAL  87.41
EACT_DIFZERO      30     50
U_TH_SM            19.9   25.6  72.4
**MINERAL          4
UNCORR-AGE_ERROR   7.87  0.21
WEIGHT             2
RADIUS-SPHERICAL  103.04
EACT_DIFZERO      30     50
U_TH_SM            30.2   41.4  119.6
*****MINERALS      0
PRECISION_0GD_1BE_2BST 0
```

## Outputs

Arvert creates a number of text files to record its outputs, and also issues a rolling status message while running. The file suffix that you specify is appended to most of these files.

The status message reports the number of CRS histories that have been processed, as well as the current best and worst fits of the data in the CRS pool. This updates every 500 histories to convince you that Arvert is alive (or every 50 histories if you are in verbose mode).

Arvert uses four utility files to carry out the restart option. The main file involved is `CRSrestart`, which contains the state of the final thermal history pool at the end of the previous run. The file `CRScount.in` attempts to track the total number of CRS iterations that have been run for a given model, assuming that the restart option has been used. The file `MONTErestart` is also written to the input directory to keep track of the original Monte-Carlo pool, should the user want to use the restart option and still plot these starting data. The file `TOFFSETrestart` is a record of the final T-offset array. These files are placed in the same working directory as Arvert itself and **must be there** if you want the restart option to work! They are overwritten with each progressive model run.

A summary of the model run (including inputs and performance stats) is placed in the text file `Modelinfo_suffix.txt`. This serves as the record of your numerical experiment.

The main Arvert output consists of files named `CRSTT_nnnn.txt` and `CRSage_nnnn.txt`, where `nnnn` is either the relevant number of CRS iterations or ‘final\_suffix’. The file `CRSTT_rolling.txt` contains the most recent set of thermal histories, and is updated every 500 CRS histories; this provides a record of what was happening should the program crash, and provides a means of peeking at model progress for longer runs. There are also two files, `MONTEtt_suffix.txt` and `MONTEage_suffix.txt`, that record the original Monte-Carlo pool used to start the inversion. It can be important to look at the former as the nature of the Monte Carlo pool can condition the path the inversion takes towards convergence, and can also influence the appearance of the thermal histories outside the region of convergence. Other analogous files have been created for postprocessing information.

If the verbose-output option is requested, you will see output files for 500, 1000, and 2000 CRS iterations as well as the final summary files; you will also see other files with this spacing if you have used the restart option. This provides a means of viewing the interesting initial stages of the inversion process. Otherwise, these files will not be written and the output directory will not be so crowded.

Finally, the file `goalspec.txt` rewrites the goal age spectrum into a format that can be plotted by Excel and compared to the model results, Arvert 7 also includes this goal spectrum data into its ‘age’ output files.

Arvert places all output except the utility files in a directory named `Results.suffix`. If you rerun a model without changing the file suffix, Arvert does not overwrite earlier results but instead starts creating numbered directories having the same name.

Additional files used in plotting are placed in a `PLOTTING` subdirectory in the `Results` directory.

In Arvert 7 all output files except the utility files and the summary file are tagged with the “`.txt`” extent. This allows you to open these tab-delimited directly into Excel (or other apps) by control-selecting and choosing “open with”.

**Output file format.** The core output files are in tab-delimited format, and can be plotted in Excel or many other (better) plotting programs.

For the age-spectrum files, the first column gives the  $^{39}\text{Ar}$  losses, the next column gives the ages for the goal spectrum, the third column gives the average of all the modeled spectra, and the subsequent columns give the ages for the individual modeled spectra, sorted in order best to worst fit.

For the thermal-history files, the first column gives the time nodes in m.y., the second column gives the average temperature over all the histories, and the third and fourth columns give the high- and low-temperature envelope around the model histories. The remaining columns give the temperatures in  $^{\circ}\text{C}$  for each history, again sorted in order best to worst fit (the fit belonging to its associated age spectra and mineral-age data). Thus, you can easily plot just summary info, or summary info plus the best fits.

It is important to note that **the average thermal history and certainly the temperature envelopes are not necessarily good-fit solutions**. Willett (1997) has found that the average history is usually not too bad a representation of the CRS pool, but the temperature envelopes must be viewed more as boundaries between temperature spaces that are not permitted in any circumstances (given model boundary conditions), and spaces that might hold acceptable solutions.

## 3.2 Viewing Results

---

If you have gmt installed and enable the plotting options, you can have Arvert 7 automatically produce summary plots. The plot will be named `arvert_results.pdf` and will be located in the PLOTTING subdirectory of your model's RESULTS directory.

The PLOTTING directory will have numerous data and helper files that make it easier for the plot script to work and do only the minimum math needed to operate. If you are using other plotting software you can make use of these files – you'll have to parse the plot script to see exactly what each is, but they are generally named to correspond to portions of the pdf plot (“reference” refers to the primary sample, “member” to the other samples).

## 3.3 Modeling Considerations

---

### While modeling

Many pages ago I discussed the importance of not treating this model as a black box, and of taking the time to work through a series of models to explore the significance of your sample. There's not much to add in this regard, except to reiterate that you should first explore what the options are with some quick runs, then do experiments to see how robust the solutions are, and then go through some final runs more carefully. A very useful and comforting thing to do is to rerun some models with a few minor changes in parameters, just to see that despite different randomly generated starting points, the model does (or does not...) converge to the same result.

Keep in mind that Arvert will try to bring *all* histories in the CRS pool to agreement with the observed data. There is nothing magic about this convergence, however. So, if you are having trouble getting the model to converge, and find that attempts to do so are causing over-convergence, you can run the model for fewer iterations, and then look at the sorted results for just those that are acceptable fits.

Also, if for geological reasons you are unhappy about some of the results, you could write code to parse the output and extract only those histories that make geological sense (you could write a program, or do this directly in Excel). And, of course, with Arvert 7, you could try post processing to better clarify what parts of the thermal history are well constrained.

**Important:** Be sure to look not only at your time-temperature results, but also your age spectra (and list of mineral ages, if you're making use of this constraint). If you have a gnarly sample, or you have not done a good job in assessing an MDD sample's domain structure, or you have made a blunder in input parameters, or you have entered illogical constraints, Arvert will still chug along happily, doing the best it can to minimize misfits, even if this effort is not very good. If the inversion gets stuck at a rather high fit value but then is allowed to continue to run, you can get into a situation where the time-temperature results look very tightly converged in places, but the model age spectra diverge widely from the observed spectrum.

**When you're done.** I strongly urge you to look at the paper by Willett (1997) for a clear discussion of what the results from a model like this mean. I will parrot some of what he says here in abbreviated form, relevant to what you might do once you have a bundle of thermal histories in hand.

Let's say you have a bundle of thermal histories you are happy with. How do you use and describe them? First, Willett (1997) has shown that to a first approximation, the average thermal history of the converged bundle isn't a bad representation of the solutions, *but it is important to realize that this average may not be a best-fit solution*. A simple way to assess or depict the constraining power of your sample is to plot the average history and then the envelopes surrounding the total bundle of results. However, these envelopes are **definitely not** a solution, and more accurately should be thought of as dividing regions of temperature-time space that **are unlikely** to be part of the solution from regions that **may** be part of the solution ("unlikely" and "may" well depend on what is geologically possible and how you have constrained the inversion).

If you want a more quantitative measure of likelihood, you will probably want to turn to other models, like Kerry Gallagher's QTQt which provides Bayesian inference (this code includes K-feldspar MDD modeling among with many other systems).

My last bit of advice is that before you embark on inverse modeling, and in fact before you even start analyzing feldspars and associated mineral data, you should

ask yourself what you are trying to accomplish and what sort of resolution is required to answer the questions you are interested in. Do you need precise time-temperature results, or just general timing of inflections in cooling rate, as a measure of tectonic or erosional processes? Do you need accurate and precise information about paleotemperatures, or will hot – medium – cool be enough? If one of the talents of a modeler is to know when to do it, another important talent is to know when to stop! And above all, don't forget geological constraints and intuition, and data from other dating systems. In this game, the more tangled the web you weave, the less likely you'll be deceived.

## 3.4 Warnings and Issues

---

Yes, yes, Arvert continues to be wonderful, but like any model it comes with baggage and pitfalls. If you don't become aware of these and recognize them in your work, you could end up being embarrassed. Here are some important things to keep in mind.

### 3.4.1. Over-convergence

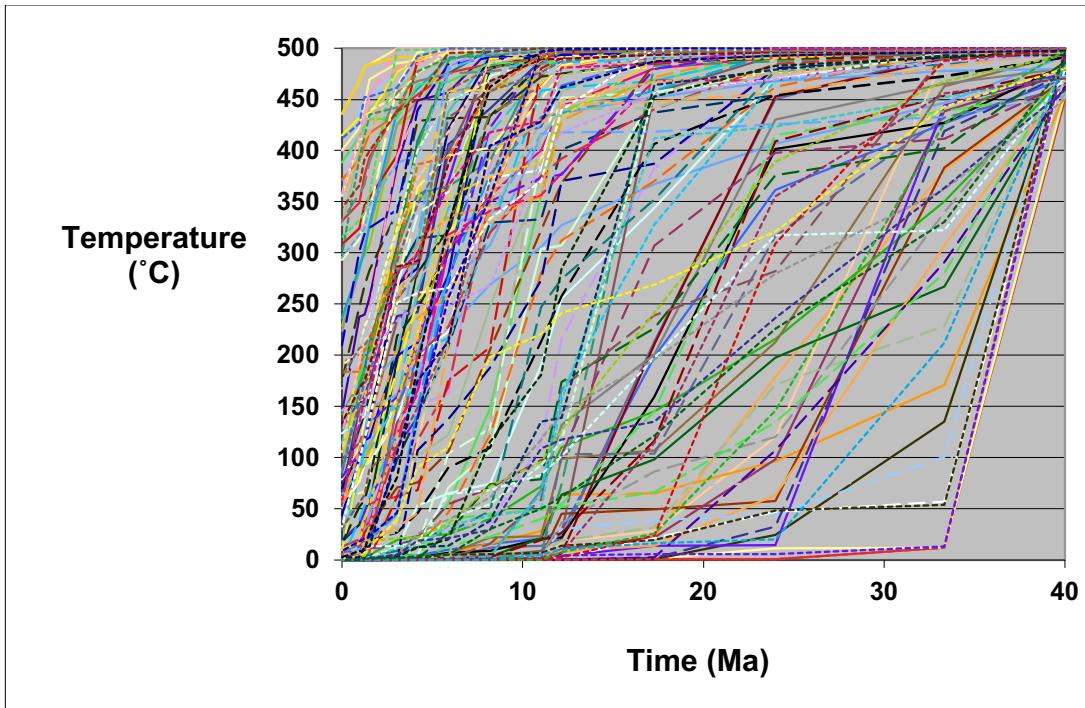
Like all its previous incarnations, the current version of Arvert sometimes has a hard time meeting the specified fitting criterion, usually because of minor errors in domain structure, a few steps of an age spectrum that aren't ideal, or mineral-age data that display overdispersion. For example, to a degree determined by how much overlap there is among domains, most parts of a thermal history affect most parts of an age spectrum, even if there is also some degree of independence. So, if after making some progress Arvert starts fussing with a piece of an age spectrum it can't match, it just keeps going, generating new CRS histories and trying in vain to fix its misfit. In the course of doing this, it may discover histories that are incrementally better due to changes in regions *outside* the problem area. So, what happens is that the model over-converges as it gradually makes tiny overall improvements. This reduces diversity in the CRS pool, and can result in the convergence of parts of the history that are outside any reasonable region that could be constrained by the actual age spectrum (e.g., at very low or high temperatures, or at times well outside that bracketed by the age spectrum). Such behavior can also be induced by a starting Monte-Carlo pool that offers insufficient diversity for the model to use to generate thermal histories of the shape it needs; the number of time nodes can also play a role here (try specifying just two nodes and see what happens!).

With Arvert 7, post processing can help a great deal, but this step cannot easily remove over-convergence if it is too far gone. Thus you will have to decide whether to model a sample using more CRS or more post-processing iterations.

**An essential point is that you should keep your eye on just that part of the thermal history your data have a chance of constraining.** You should not think that running the model forever will eventually get you to a good place: more is not always better! If you see this problem, re-examine the fits to your observed data to see why Arvert can't succeed: is it a highly finicky thermal history with complexities like rapid changes in rate, or is there a problem with some steps in an age spectrum or your mineral data? To see how important the problematic area is, you can always turn off fitting in the problem region and see what kind of history you then get. I'm belaboring this point because too often I have seen people ignore the data they worked so hard to collect!

### 3.4.2a. Biases from constraints

Earlier, I discussed how it's possible to produce a rather odd-looking pool of Monte Carlo histories by allowing fast cooling but no or little heating. An example is shown in Figure 4. Arvert can cope with such a start, but if you were expecting there to have been a fast-cooling pulse, you must recognize that you are *serving up this result to the model*. Also, a consequence of using a set of histories as shown in Figure 4 is that in the regions close to the directly constrained region, the model will give the appearance of a trend in temperatures that is merely inherited from the starting pool. *The preceding is a subtle but important point!*



**Figure 4.** Starting Monte-Carlo thermal-history pool, generated with constraints allowing no heating and cooling at rates of up to 100°C/m.y. Note potential bias in these starting histories towards fast-cooling scenarios, and inflections in rate. If those inflections fall outside the constrained region, they will end up in final poll and give the appearance of complex history.

Figure 4 gives a simple, common, and obvious example of how you might introduce possible biases, but similar features can creep in in other ways. You might specify a model duration that starts the model too close to initial cooling to allow it to explore temperature space. Perhaps you might have chosen some explicit constraints that while legal act in a subtle way act to rule out certain histories due to interactions with rate constraints. It can be hard to anticipate all such problems, and you might question the whole premise that inverse modeling can overcome the operator biases that can pollute forward modeling.

The answer is, once again, to run multiple models with different starting conditions, and see what you get. Try feeding the model a starting set like that shown in Figure 4, and then try the opposite, starting it with a pool showing mostly shallow slopes. Overlay the two or more results to get a more robust picture of what your sample can and cannot tell you.

**3.4.2b. Biases from time nodes.** While having too many time nodes can slow the model and create an underdetermined situation, the location of time nodes relative

to your specific age-spectrum and mineral-age constraints could lead to bias, should there be insufficient time nodes to define segments that are needed to represent the correct thermal history. It is **strongly advisable** to run several models where all you change are the number of time nodes – be sure to check that when you do this, no major changes occur in the thermal history. This should be much less of a problem in Arvert 7 because it tends to use a larger number of CRS time nodes, but you should still keep an eye on possible bias, especially if you really cut back on the number of CRS time nodes if you have a model where the exact time of a major cooling-rate change is important.

**3.4.3. Choosing MDD Steps to Fit.** You will have to make some informed judgments about which steps from your age spectrum can be used for fitting.

At high levels of  $^{39}\text{Ar}$  release, KF spectra often go a bit wild just when incongruent melting occurs and the kinetic properties of the sample become undefined. In other samples, the age spectrum seems to sail nicely right past this point. The proper thing to do is to fit only steps below the incongruent melting point because only these steps will have been released by the process of volume diffusion, which is the only process Arvert models. Keep in mind that the presence of impurities (quartz in myrmekite, albite in exsolution lamellae) will likely cause melting to happen at lower temperatures than you'd expect for pure K-feldspar. It is usually quite obvious from R-Ro plots when such breakdown is happening.

At low levels of  $^{39}\text{Ar}$  release, the main problem is usually fluid-inclusion hosted  $^{40}\text{Ar}$ , and if you use isothermal replicates to explore this phenomenon, you will likely have a spectrum that initially descends while oscillating in detail. Some of these often-small steps have relatively large uncertainties. If no correction for Cl-correlated Ar is possible, then you will have to decide which if any of the younger ages in the replicate zone might be reliable. This can be problematic across the transition from old inclusion-related ages to pure cooling ages, since ages will drop and then rise, and thus define a flat segment for a bit. If you retain those steps uncritically, Arvert will interpret that flat segment as fast cooling when in fact it is an artifact. Having multiple replicates from multiple samples can help here.

What about fitting non-contiguous steps? My own view is that early in an age spectrum, if there is clear evidence for fluid-inclusion-hosted Ar, then it is ok to fit alternate steps, although this does introduce an element of subjectivity. In the middle and later parts of an age spectrum, I would be extremely wary of starting to omit the odd step, as this is a dangerous and subjective game. If you have a single

step that seems to represent some sort of analytical glitch, I suppose it would be ok to flag it for omission.

**3.4.4. Crashes and unsociable behavior.** At this point, Arvert 7 seems fairly robust and I have not seen a crash in quite some time that was not due to input-file error. I believe all issues with memory including leaks, array indices, pointers and the like have been beaten out of the program.

However, there are some issues that could appear. Despite extensive use and testing, keep in mind that Arvert uses large amounts of random numbers for a number of procedures and creates new data out of complex combinations of earlier data. This makes some problems very hard to debug, so it is possible that on occasion a random sequence might produce a glitch or error that is not caught, creating a divide-by-zero or out-of-bound array indices. Given the nature of C and C++, the latter in particular can lead to unexpected results but not run-time errors.

If you experience a bad run, or odd-looking data, please double-check your input parameters. If these are ok, then try re-running the model a few times with exactly the same parameters. If the problem persists, let me know. If it goes away (it usually does), you can assume it was a rare glitch related to random-number generation.

**Best-fit worst-fit bug.** Rarely at this point, you may notice that the progress message that reports CRS iterations will tell you that the best-fit history has a worse fit than the worst-fit history. I have tried for the last two Arvert versions to fix this bug, thinking several times it was squashed, but sometimes it reappears. Usually the inversion continues and heals itself. If this worries you, you can also restart the run. This issue seems to have been fixed as a byproduct of coding Arvert 7, but we will see.

**Rare “Segmentation fault: 11” Error.** Very rarely, you may see this error during the Monte-Carlo phase. It probably has to do with an unusual thermal history being generated that when interpolated, results in an array index that is too big. This error has not surfaced in quite some time, so here’s hoping it’s been eliminated.

---

## Section Four – Special Topics

### 4.1 History of Arvert

---

It's been a long, strange trip. You really don't want to know the history of Arvert in any detail. What follows is already much more than too much.

Back in the mid-1980's when I was a research fellow at RSES/ANU in Canberra, I took some Crank-Nicolson diffusion code for thermal modeling that ultimately traces back via Mark Harrison to Garry Clarke at UBC, and converted it to model Ar diffusion profiles. By the time I left Canberra in 1988, this code could generate age spectra for any given thermal history and domain distribution. At Lehigh, I was bothered by the under-determined nature of age spectra and the potentials for operator bias during forward modeling, so I stumbled ahead to create a crude, purely Monte-Carlo inverse model. We actually got some interesting results but in retrospect this approach was hilariously futile and naïve, given the number of possible random thermal histories that can exist and the relatively small proportion that are solutions.

During a visit to Dalhousie in the mid-1990's, Sean Willett introduced me to his application of the CRS algorithm to the inversion of fission-track data. After a little effort, Arvert was born, using the Crank-Nicolson code as a core forward model and the CRS algorithm to guide the inversion. A few years went by tweaking this model and learning its ins and outs, all the work being concentrated in occasional patches when I'd find the time and the interest to push things a little further (usually when I was procrastinating from something like grading).

A few years ago, I decided to abandon the finite-difference diffusion function in favor of Oscar Lovera's code, which is in principle more accurate. (As a footnote, it turns out that for routine work, the finite-difference code is as fast and not that different in accuracy). There followed some excursions into LabView programming of helper apps, and then a descent into darkness as I tried to make the inversion more reliable and easier to use; over the years this included user-specified time nodes (hard to input, and prone to user bias), randomly variable time nodes (nice idea but the CRS routine would tend to grab hold of certain nodes and fatally select against the others), and then Chebyshev-type curves where the CRS routine worked on polynomial coefficients, not  $tT$  points (it's hard to corral these coefficients to produce non-wacko histories).

The more recent versions of Arvert returned to simplicity, incorporated mineral-age constraints, and seem to work reliably, at least in the hands of an experienced user who doesn't expect too much. Feel free to contact me if for some inexplicable reason you want to learn more about the guts of this code and how it got here.

## 4.2 Performance

---

Avert is still many generations of processor upgrades away from being real-time interactive, but its performance is much improved from the days not long ago when overnight runs were the norm and input typos were cause for tears. On a gigahertz machine you should be able to run a typical survey model of a few thousand histories in a few minutes or less.

Arvert's performance scales pretty much linearly with the number of domains, the number of heating steps, and the number of mineral ages. The CRS, bookkeeping, and other routines account for only a small part of the total CPU usage, although under some conditions you can make the CRS routine struggle as it tries to create a new thermal history that's legal. Including multiple mineral ages, especially with better precision and using radiation-damage models, will also slow things down, potentially a lot.

I have not looked deeply into parallelizing Arvert. That would seem to be the way to go given the nature of modern CPUs and the availability of GPU cores. The CRS algorithm is not really amenable to using simple approaches like OpenMP and OpenACC because of the need to share the data in the CRS pool; all the other routines are already fast enough that the overhead in instantiating parallelized loops tends to outweigh any local performance gains. What might be fruitful (volunteers?) would be to write a small cluster script that runs multiple instances of Arvert in “embarrassingly parallel” mode (one model per node) – this would be a way of more quickly testing different parameter sets.

## 4.3 Future Versions of Arvert

---

For now, it's time to see if the updates that amount to Arvert 7.0 are stable and useful. If there is user demand, I'm open to making some additional changes. Potentially on deck are:

- ${}^4\text{He}/{}^3\text{He}$  ratio-evolution data
- fission-track data

- using the actual heating schedule and the domain info to predict lab  $^{39}\text{Ar}$  losses (instead of creating a faux schedule that matches up loss values).
- revisiting using finite difference code for age spectra instead of Lovera(), especially if a stepheating F.D. module gets added anyway for 4/3 data – might this be more efficient?
- $^4\text{He}/^3\text{He}$  CRH release spectra for He in apatite, including adding a diffusion-jump module that can handle He trapping in pores (this will depend on how research in this area goes).

What might dissuade me from adding more to Arvert is that I think Bayesian analysis is the way to go, because of the way it allows for more rigorous assessment of results. I've helped Kerry Gallagher get MDD analysis into his QTQt model, which has many advantages and is set up for more mineral systems than Arvert, including  $^4\text{He}/^3\text{He}$  spectra. I strongly recommend getting up to speed on this model.

There seems to be general interest in the community for folding spectral methods like  $^4\text{He}/^3\text{He}$  and MDD into kinematic, geodynamic, and other models, but that would represent essentially new code. I think there would be merit in the community producing rock-solid open-source modules that could be shared and then more easily added to other sorts of models. Our OpenThermochronology GitHub site is an attempt to move in this direction.

---

## Section Five – References and Suggested Reading

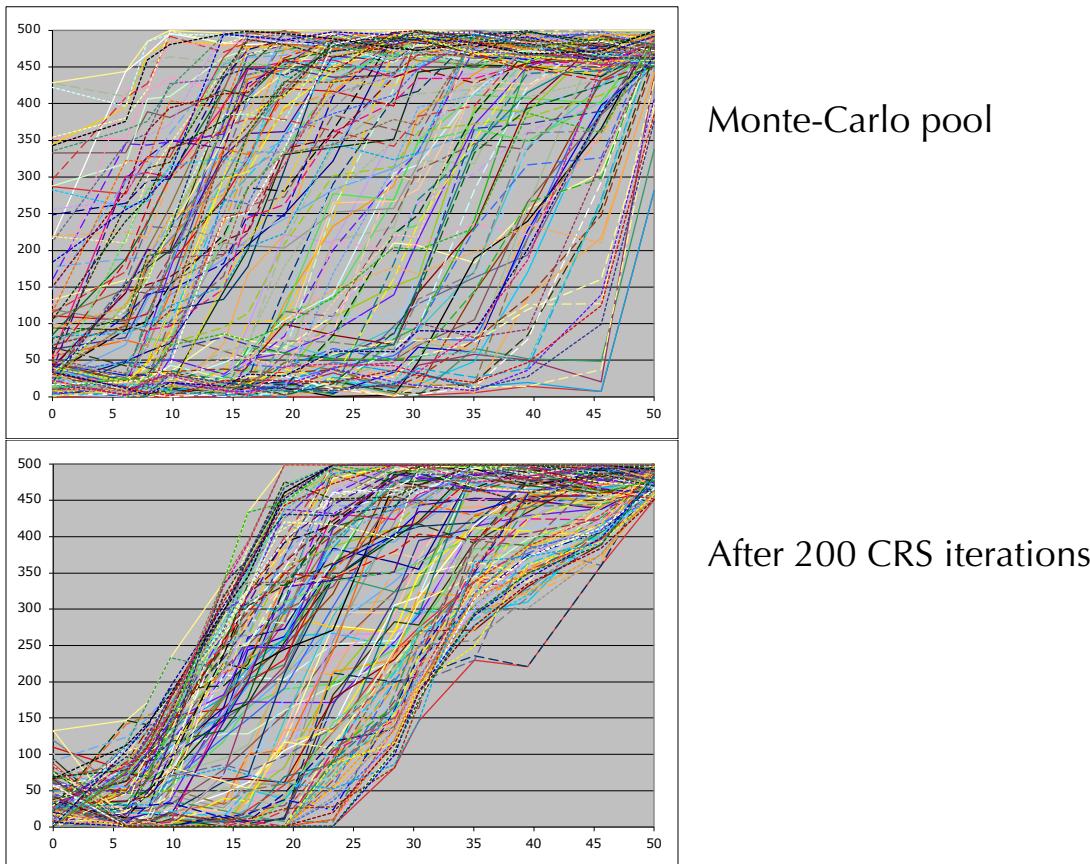
- Flowers, R. M., Ketcham, R. A., Shuster, D. L., and Farley, K. A., 2009. Apatite (U–Th)/He thermochronometry using a radiation damage accumulation and annealing model. *Geochimica et Cosmochimica Acta*, 73, 2347–2365.
- Gallager, K., 2012. Transdimensional inverse thermal history modeling for quantitative thermochronology. *Journal of Geophysical Research*, 117, B02408, doi: 10.1029/2011JB008825.
- Gallagher, K., Charvin, K., Nielsen, S., Sambridge, M., and Stephenson, J., 2009. Markov chain Monte Carlo (MCMC sampling methods to determine optimal models, model resolution and model choice for Earth Science problems. *Marine and Petroleum Geology*, 26, 525–535.
- Guenther, W. R., Reiners, P. R., Ketcham, R. A., Nasdala, L., and Giester, G., 2013. Helium diffusion in natural zircon: radiation damage, anisotropy, and the

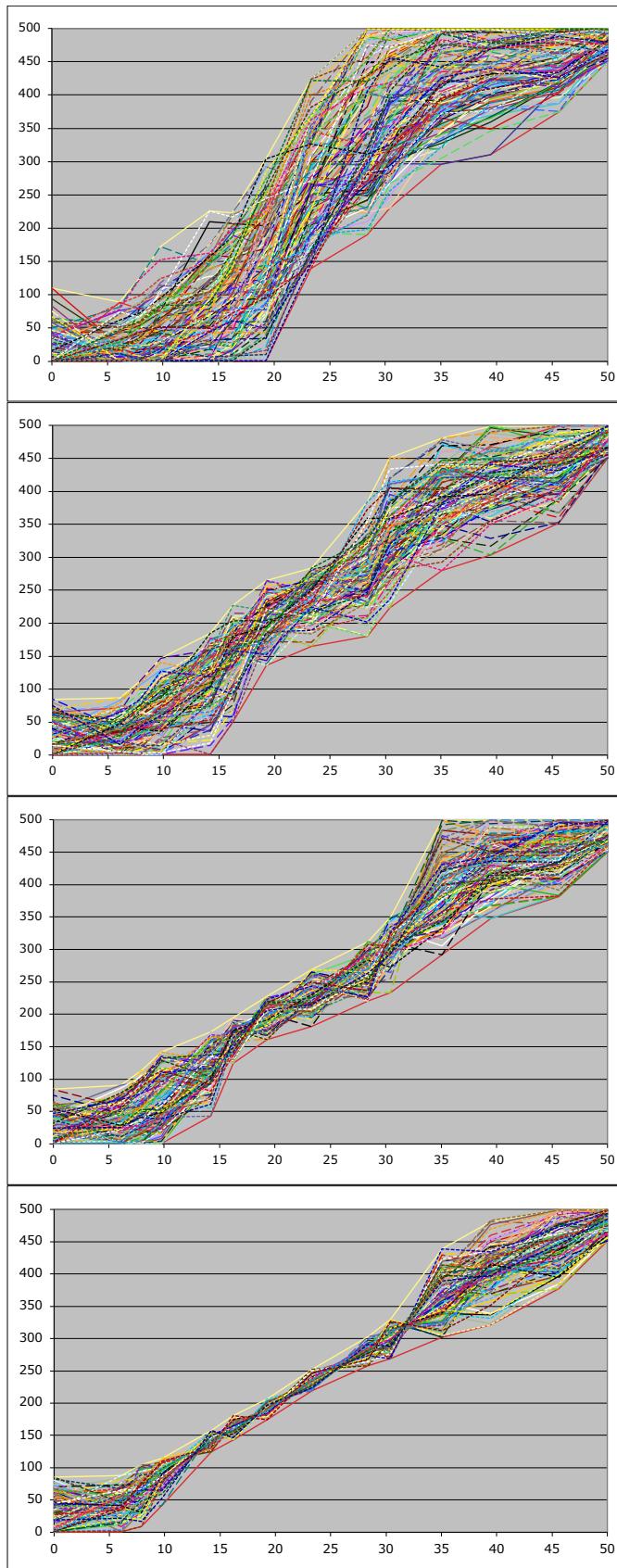
- interpretation of zircon (U-Th)/He thermochronology. American Journal of Science, 313, 145-198.
- Kaelo, P., and Ali, M. M., 2006. Some variants of the controlled random search algorithm for global optimization. Journal of optimization theory and applications, 130(2), 253-264.
- Ketcham, R.A., 2014. HeFTy Versions 1.8.3 – Users Manual.
- Ketcham, R. A., Gautheron, C., and Tassant-Got, L., 2011. Accounting for long alpha-particle stopping distances<sup>[1]</sup> in (U–Th–Sm)/He geochronology: Refinement of the baseline case. Geochimica et Cosmochimica Acta, 75, 7779–7791.
- Lovera, O.M., Richter, F.M., and Harrison, T.M., 1989. The  $^{40}\text{Ar}/^{39}\text{Ar}$  thermochronometry for slowly cooled samples having a distribution of diffusion domain sizes. Journal of Geophysical Research, 94 (12), 17,917-17,935.
- Meesters, A., and Dunai, T.J., 2002. Solving the production-diffusion equation for finite diffusion domains of various shapes; Part II, Application to cases with alpha -ejection and nonhomogeneous distribution of the source. Chemical Geology, 186 (1-2), 57-73.
- McDougall, I., and Harrison, T.M., 1999. Geochronology and thermo-chronology by the  $^{40}\text{Ar}/^{39}\text{Ar}$  method. Oxford University Press (Oxford), 2nd edition, 269 pp.
- Price, W. L., 1977. A controlled random search procedure for global optimization. The Computer Journal, 20(4) 367-370.
- Willett, S.D., 1997. Inverse modeling of annealing of fission tracks in apatite; 1, A controlled random search method. American Journal of Science, 297 (10), p. 939-969.

## Section Six – Appendices – Convergence Sequence and Sample Results

The on-line Avert distribution contains a suite of sample input and output files you can use to check your installation. Below, I provide a venerable example of a typical Arvert convergence sequence, and some sample results for synthetic data. While it's gratifying to see Arvert work so well, keep in mind that for the synthetic data, Arvert damn well better work, since the synthetic data were created by the versions of the same lovera() and mineral() routines that are at the core of Arvert!

The first sequence of images gives the convergence sequence and summary results from a model of synthetic data; these were calculated for a multi-domain sample experiencing linear slow cooling at 10 °C/m.y.



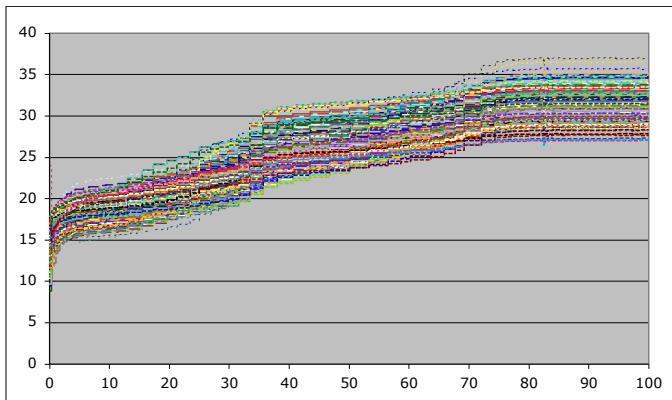


500 CRS iterations

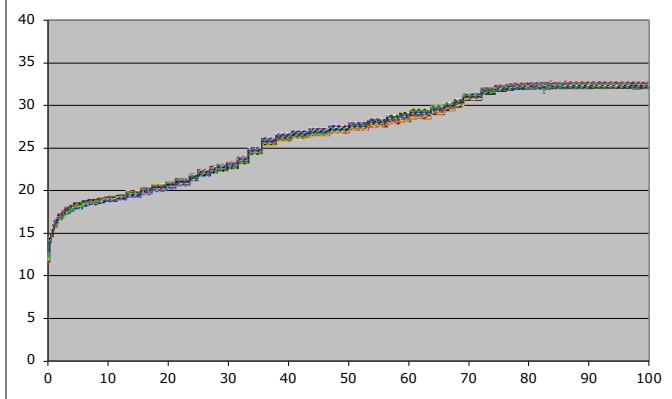
1000 CRS iterations

2000 CRS iterations

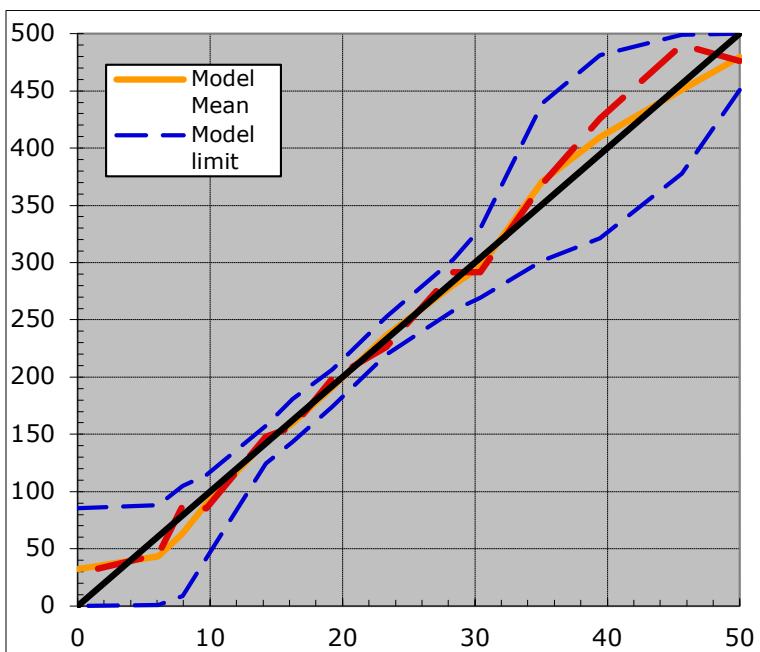
Final pool – 5599 CRS iterations



Age spectra after 1000 CRS iterations



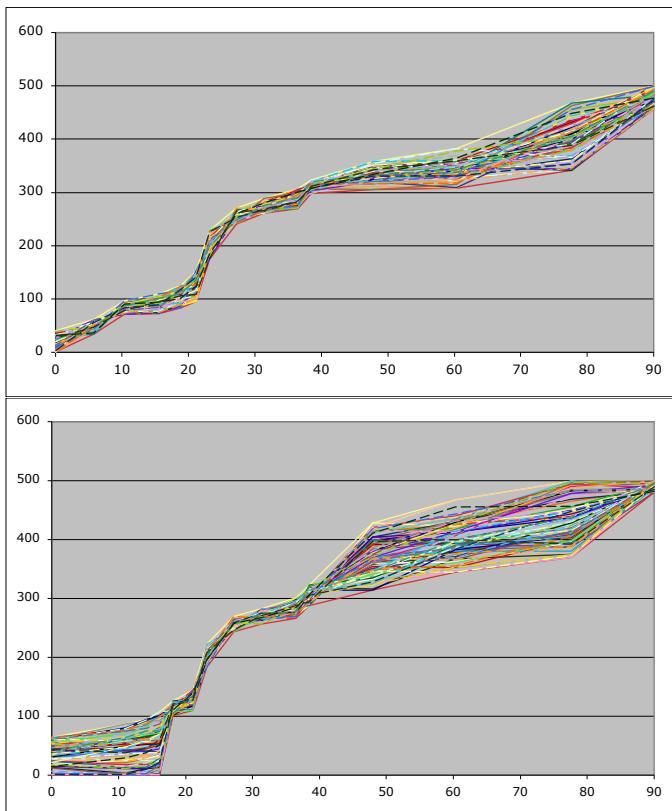
Final age spectra after 5599 CRS iterations



Summary of results for linear model. Note that Arvert output files contain this sort of information in the first few columns of data, making it easy to produce summary plots like this.

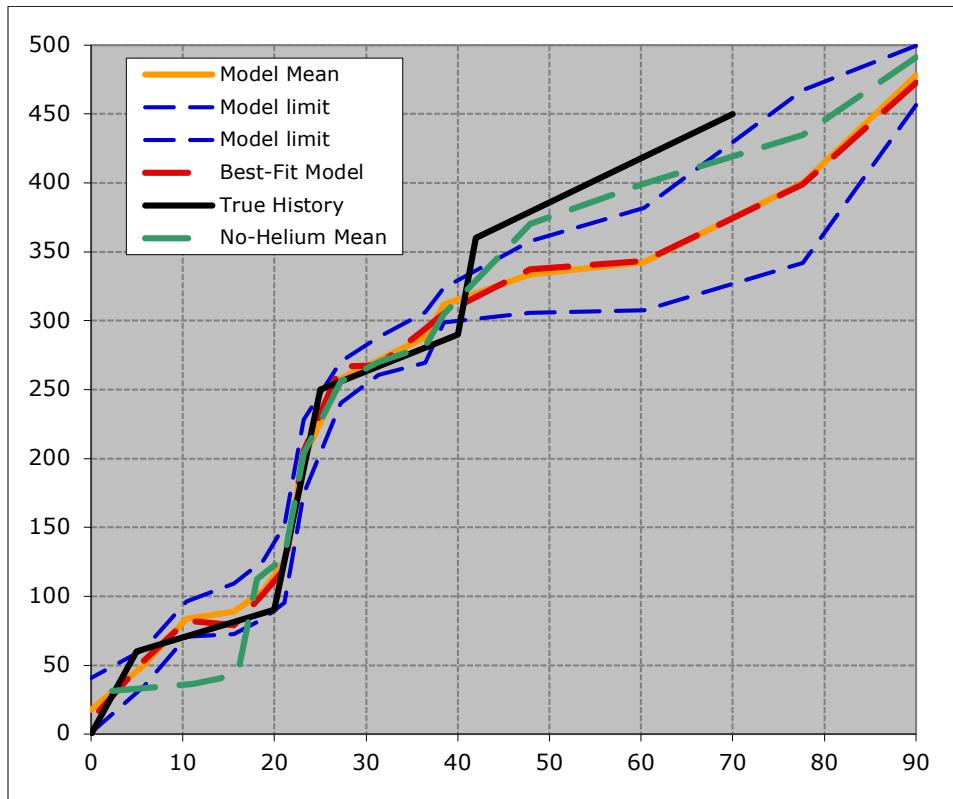
In the sequence above, note how the progressive tT histories inherit the sigmoidal shape of the original Monte-Carlo pool. Gratifyingly, this has largely been expunged by the end of the model, but it's worth keeping an eye on.

The next sequence of images shows summary data and summary results from a model of synthetic data that include a He age as a constraint and that were calculated for a non-linear cooling history. Also shown are results of an inversion where the He data are not included as a constraint.



With helium age as constraint

Without helium age as constraint



Summary of model runs for synthetic data, non-linear cooling history