response to associate editor

1. Clearly there is a difficult balance to strike between accessibility and providing full details of our new approach. We have moved much of the mathematical formulation into Appendices, but retain key information in the main text. We have also tried to provide insight into the (often cryptic) protocol and calculations implemented in the widely-used ISOPLOT software of Ken Ludwig.

2. In order to address the associate editor's suggestion, a comparison of application of different algorithms, including those used in ISOPLOT, is included in the revision in a new Appendix, using the data used for the old Fig. 6 (now Fig. 7). These data are now included in Appendix C as requested by reviewer 4, making it straightforward for the reader to see and assess the results.

   The robust algorithms in Ken Ludwig's ISOPLOT are given in the isoplot manual 3.75, (2008 revision) p. 25, with details in the papers cited. Strictly these are resistant algorithms, having high breakdown point, but low efficiency (e.g. Huber & Rochetti, 2009, Sect. 1.2.3). In our algorithm, and the code in our manuscript, such methods are used to provide a *starting point* for the iteration in SPINE, as now spelt out at the start of Appendix C. Including Siegel (1982) was the idea of reviewer 4, and that has been implemented in the code and included in the manuscript.

   Ken Ludwig was a reviewer of the Powell et al. (2002) paper, and was enthusiastic about the work. Subsequent communications between Ken and the 1st author were aimed at including the robust isochron calculation approach of the 2002 paper in ISOPLOT. Unfortunately this didn't happen, due to personal matters that Ken experienced immediately after that time. In the ISOPLOT documentation regarding robust regression, his first suggested method is superseded by the second (Siegel, 1982). Looking at Ken's `vba` ISOPLOT code, the bootstrap used with the Siegel (1982) data-fitting is a simple "cases", or structural-based resampling, as though the x-y data are drawn from some underlying (unknown) bivariate probability distribution (not the more appropriate, model-based approach used in the 2002 paper). However, any bootstrap is difficult and undeveloped for robust regression, e.g. Davison & Hinkley (1997), Sect. 6.5, and unreliable in general for median-based calculations. Significant effort went into developing the bootstrap for our current work - without success. In the 2002 paper the analytical uncertainties were discarded before the regression and the bootstrap were calculated, as already stated in the Discussion of the manuscript. Integrating the analytical uncertainties into the bootstrap contributed significantly to our difficulty.

3. We think that there may be a misunderstanding here regarding the sentence quoted from our response to review 3, "...such additional datasets show that the 95% confidence limit on the ages is 3.97 to 4.03 Ma with SPINE, but 3.91 to 4.09 Ma with YORK, a significant increase in reliability with SPINE". This is a 95% confidence interval on *all* the individual ages of such datasets, 4445 of them, that fail `mswd` but have acceptable spine width, using the cutoffs in Table 1 in the manuscript. Reliability is possibly not the best word to use, but this result flags how much better SPINE does than model 2 (or YORK) in this simulation.

   Philosphically isochron-errorchron and model 3 are orthogonal to each other. In the former, a general purpose calculation method is sought which includes a way to distinguish datasets that are more likely to have age significance (isochrons) from those which are less likely (errorchrons). In the latter, data scatter is parameterised with analytical uncertainties and additional contributions depending on the geological processes envisaged to be involved. In the latter, maximum likelihood can then be used to devise an algorithm to find the model parameters, if the imagined data uncertainty structure is completely specified. We, in our manuscript, aim for the former, whereas the reviewer has focused on the latter in recent publications.

   Model 3 calculations can be very interesting when there are geological reasons to make them plausible. However, commonly, identifying the cause of excess scatter is not easy to do or may indeed be impossible. Instead, our approach aims to calculate age information, discounting the potentially deleterious effects of excess scatter. Using a robust statistics logic, isochron calculations can be undertaken for a wider range of datasets. Importantly our approach is consistent with YORK for "good" datsets (e.g. with `mswd` less than some cutoff, as in model 1, accepting that the data (analytical) uncertainties account for data scatter). But, in addition, the approach extends seamlessly to datasets where `mswd` is

greater than the cutoff (e.g. model 1 with expanded errors, as well as model 2). This extension is considered to also give isochrons as long as the central spine of the data, s, is less than some cutoff. The cutoffs, depending on the number of datapoints, are given in Table 1 of the submitted manuscript, from a 95% confidence interval on s for simulated datasets with Gaussian-distributed data uncertainties. The use of these cutoffs is directly analogous to the use of cutoffs on mswd, also given in Table 1. The use of such s cutoffs avoids the rubbish-in rubbish-out problem, in the same way that mswd cutoffs do in YORK fitting, for strictly Gaussian data uncertainties.

We would contest the idea that maximum likelihood and the tests that can be formulated in that framework are always that useful, particularly for the small size of datasets that are typical of geochronology. As noted already in the response to review 4, even the 51-point dataset of Fig.6 is insufficient to distinguish a strictly Gaussian data uncertainty structure from a contaminated Gaussian one. In YORK calculations the adoption of a strictly Gaussian uncertainty structure is an assumption - it cannot be tested for. It is well known that even trivial departures from this assumption have a deleterious effect on the results from using classical statistical methods (see, for example, the introductory chapters of Huber, 1982, and Hampel et al., 1985). This may well apply in model 3 calculations too. Whether the assumptions in the parameterisation cause a problem in algorithms used for model 3 calculations could easily be tested with simulations, for example using very slightly contaminated Gaussian data, e.g. 5%3N.

Certainly, in the current manuscript we are not following the "cult of mswd". Far from it. We are advocating that this statistic is not useful and we do not use it. Mswd appears in the manuscript simply because it is needed to show that SPINE subsumes YORK for "good" datasets. Indeed, because of that, our algorithm is based on a minimis- ation that reduces to minimising mswd for "good" datasets. Of course, mswd is pivotal in model 3 calculations given that it is used to signal the existence of excess scatter. But, if, as suggested in Vermeesch (2018), excess scatter starts at mswd = 1 with increasingmswd, rather than at some cutoff from a confidence interval on mswd, then a model 3 calculation might well be trying to model something which is not there statistically, just noise? See Table 1 in the manuscript in the context of Appendix A, discussing ISOPLOT model 1 calculations. Even for a 50 datapoint dataset, a 1-sided 95% confidence interval on mswd extends up to 1.36. It doesn't stop at $\texttt{mswd} = 1$.

The main ideas in this section of our response are incorporated in the revision, but a discussion of model 3 calculations is beyond our remit there.

Robust Isochron Calculation

Roger Powell[1], Eleanor CR Green[1], Estephany Marillo Sialer[1], and Jon Woodhead[1]

[1]School Earth Sciences, The University of Melbourne, Vic 3010, Australia

Correspondence: Roger Powell (powell@unimelb.edu.au)

=== main changes in blue ===

Abstract.

The standard classical statistics approach to isochron calculation assumes that the distribution of uncertainties on the data arising from isotopic analysis is strictly Gaussian. This effectively excludes from consideration datasets that have excess scatter, even though many appear to have age significance. A new approach to isochron calculations is developed in order to circumvent this problem. This only requires that the central part of the data uncertainty distribution is Gaussian, significantly increasing the range of datasets from which age data can be extracted but also providing seamless integration with well-behaved datasets, and thus all legacy age determinations. A statistical test is provided to ensure that a central spine of the uncertainty distribution data is Gaussian. Then a robust line-fitting approach is adopted that is more reliable when used on data with excess scatter, but is coincident with the classical statistics approach for datasets without excess scatter. A calculation method for the algorithm is presented, accompanied by an implementation in Python.

1 Introduction

The ability to fit a straight line through a body of isotope ratio data in order to form an isochron is the cornerstone of many geochronological methods. In detail, however, this is a non-trivial task, since uncertainties are usually associated with all variables, and these are often correlated, precluding simple "least squares" line-fitting techniques. Most of the research in this area was conducted in the late 1960's and early 1970's, being dominated by a classical statistics approach in which data uncertainties, derived from the analytical methods, are taken to be strictly Gaussian-distributed (e.g. York, 1969; York et al., 2004, and references therein). This approach, referred to here as YORK, became entrenched in the geochemical community, particularly in the last two decades as the essential component of the very widely-used software, ISOPLOT, e.g. Ludwig (2012).

In this contribution we examine some of the problems inherent in these techniques and suggest an alternative approach. Our primary focus here will be on general-purpose isochron calculations, involving determining the age of an "event" that established the isotopic compositions of samples in a dataset. This involves what are called model 1 and 2 calculations in ISOPLOT - as described below. Approaches that try and extract detail within events, including ISOPLOT model 3 calculations, are not considered (but see e.g. Vermeesch, 2018).

1.1 On ISOPLOT

In order to show that there are significant problems in using ISOPLOT for general-purpose isochron calculations, and then to see how they can be addressed, it is first necessary to outline the ISOPLOT protocol, some details of which may not be apparent to the user. Central to this workflow, the main tool for considering data scatter is mswd, the mean standard weighted deviates (also called the reduced chi-squared statistic), see eq. 1. For strictly Gaussian distributed uncertainties $(n-2)$mswd is distributed as chi-squared (χ^2_{n-2}), meaning that if data uncertainties are correctly assigned, a strong statistical statement can be made about whether the data scatter of a particular dataset is solely consistent with the data uncertainties (i.e. with no geological scatter), for example in the form of a 95% confidence interval on mswd, Wendt and Carl (1991). Such a confidence interval is not fixed, but depends on the number of datapoints under consideration, so for example for $n = 10$, mswd < 1.94 (meaning that mswd extends from less than 1 through to 1.94), while for $n = 50$, mswd < 1.36. A dataset with mswd in the chosen range for the number of datapoints has data scatter that is consistent with the data uncertainties. This situation is commonly referred to as mswd "passes"; otherwise mswd "fails", in relation to χ^2_{n-2}. Mswd passing provides a "pure" interpretation of YORK, and, in ISOPLOT is referred to as a model 1 fit. This is depicted as the horizontal line in Fig. 1, indicating that in this range of mswd, corresponding to a confidence interval, the calculated uncertainty on an isochron age does not vary with mswd. Such a figure is drawn by taking an actual dataset and progressively modifying it to show what happens as mswd varies, as described in Appendix A.

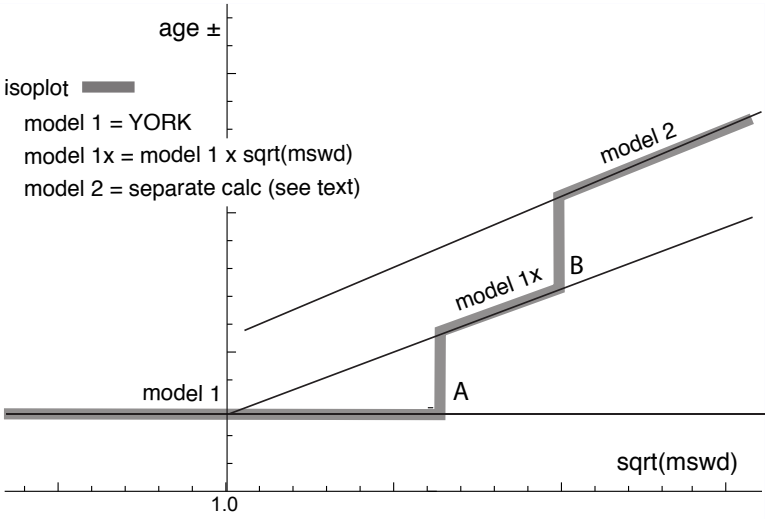

Figure 1. [new fig and caption] Age uncertainty (age±) plotted against $\sqrt{\text{mswd}}$ under the ISOPLOT protocol for a progressively modified dataset (see text, and Appendix A). Under the condition of a model 1 fit, the age uncertainty is constant with increasing data scatter (reflected in increasing mswd), until there is a step change in the data treatment at A when the age uncertainty is multiplied by sqrt(mswd). Then at B there is another step change in age uncertainty calculation with increasing data scatter forming ISOPLOT model 2 (see text)

What if mswd is greater than the upper limit of the chosen confidence interval? Then the data are considered to have *excess* scatter, in addition to that accounted for by the data uncertainties (assuming that they are strictly Gaussian). At this point, ISOPLOT, asks the user whether an alternative—model 2—calculation should be undertaken. This decision point is indicated at A in Fig. 1. If the user declines, ISOPLOT gives results that are referred to here as model 1x, as shown in Fig. 1. The model 1 age uncertainty is multiplied by $\sqrt{\texttt{mswd}}$, to reflect the data scatter being more than expected from the data uncertainties alone. With further scatter, the switch is made to model 2, at B in Fig. 1. If the switch to model 2 takes place at A in response to user input, then model 1x is not used, the vertical line at A extending up to the model 2 line in Fig. 1. The model 2 calculation in ISOPLOT is unrelated to YORK. The data uncertainties are discarded and the slope of the line through the linear trend is calculated as the geometric mean of the lines calculated by unweighted least squares of y on x and of x on y (see Appendix A).

In summary, then, in ISOPLOT the calculation of ages and their uncertainties involves a number of decision points based around the concept of mswd that impart significant (and in our view unwelcome) step-changes in the way that the data are handled, and algorithms applied. To assist in further discussion of these matters we depart from the language of ISOPLOT at this point, reintroducing the term errorchron, counterposed to isochron, following Brooks et al. (1972). The idea is that isochrons have a higher chance of having age significance, while errorchrons have a lower chance. In particular, it seems to be unhelpful for the results of model 2 calculations to be called isochrons as done in ISOPLOT, given that there is excess scatter in the data.

1.2 Replacing ISOPLOT

Given that ISOPLOT's implementation of model 1+2 line fits is the gold standard of isochron calculations presently, where are the problems, and then what can be done about them? A shortcoming in YORK stems from the assumption that data uncertainties are *strictly* Gaussian-distributed. In real-world application this appears to be too restrictive, with datasets that are likely to have age significance being labelled as errorchrons because mswd is too large. While using YORK guided by mswd is optimal statistically if data uncertainties are strictly Gaussian, this logic fails once uncertainties are even slightly non-Gaussian. In such circumstances, both mswd and least squares methods themselves, like YORK, become unreliable (e.g. Hampel et al., 1986; Huber, 1981).

Rather than being truly Gaussian, data uncertainties may well be Gaussian-distributed in their centres, but slightly fat-tailed distant from the centres. An isotopic dataset looks intuitively acceptable if the data has a central linear "spine", in which scatter is commensurate with stated analytical uncertainty, but this spine is flanked by data of somewhat larger scatter (i.e. excess scatter, from the "fat tail"). This excess scatter may originate in the isotopic analysis or as a result of geological disturbance. Age-significance in such data manifests primarily via the position of the spine. In the following, the focus is on this spine in the data.

Adopting this spine approach, a successful calculation method for a dataset that may not have strictly Gaussian-distributed uncertainties must, firstly, ascertain whether or not such a spine exists in the data—and hence whether calculations yield an isochron or an errorchron. Secondly, in the case of an isochron calculation, the successful method must reliably locate the

spine without being perturbed by vagaries in the more scattered data. Classical statistical methods can do neither of these things, tending to be excessively influenced by the data at the extremes of the scatter. However, the field of robust statistics offers calculation methods that can. When a dataset has no excess scatter, reflected in mswd lying within an appropriate χ^2-constrained confidence interval, such methods can be devised to retrieve identical results to classical statistic methods, but, in addition, provide reliable age and age-uncertainty estimates in the presence of excess scatter around a spine. This continuity of operation with increasing mswd contrasts with previous approaches and means that the steps in the ISOPLOT line in Fig. 1, which are certainly undesirable, are circumvented. Moreover the involvement of potentially unreliable least squares-based methods, like ISOPLOT model 2, are avoided when the data show excess scatter.

2 An Algorithm for Isochron Calculations

An algorithm is sought that finds a robust straight line through a 2-dimensional linear data trend, while converging with the classical statistical approach of YORK for datasets with consistent scatter (i.e. mswd passes). This section describes the nature of the problem and the theoretical basis for the robust statistical approach that will be adopted. The algorithm adopted

1. determines a preliminary resistant fit of the data, not dependent on vagaries of the data scatter
2. determines the spine width in relation to this preliminary fit
   - if the spine width is in an acceptable range: isochron
   - if the spine width is not in an acceptable range: errorochron
3. determines a robust fit of the data, starting from the preliminary resistant fit, it reducing to YORK for "good" data

This algorithm is fleshed out below, and then evaluated via simulated datasets and applied to a natural dataset. The central calculation in the algorithm is detailed in Appendix B, and a python implementation is given in Appendix C.

2.1 Uncertainty distributions and data fitting

Geochronological datasets are collected on the presumption that the isotopic compositions were established via an "event" the age of which is to be estimated. Given the focus here on data with linear trends, even if the effect of the event is recorded perfectly by the samples analysed—the isotopic compositions lying on a line—the actual data are measured with finite precision and so the data inevitably scatter about the trend. An uncertainty probability distribution can be used to describe the form of the data scatter.

Classical statistical methods assume that the underlying uncertainty distribution of a dataset is known, typically taken to be Gaussian. Under the Gaussian assumption, if the analytical uncertainty on the measurements have been appropriately inferred, mswd, the classical statistics parameter used in YORK to validate an isochron, tests that the scatter of datapoints is consistent with the inferred uncertainties. But, in general, there is no reason to suppose that a given analytical technique generates a

truly Gaussian uncertainty distribution. Even small amounts of geological disturbance destroy the optimality of YORK. If the uncertainty distribution is not strictly Gaussian then classical methods of data fitting become sub-optimal or worse.

While there are many possible non-Gaussian uncertainty distributions, this paper is concerned with a situation commonly occurring in datasets, in which the datapoints form a linear spine with Gaussian-like scatter, but additional scatter is seen in the tails of the distribution. Such a dataset still encodes meaningful age information in its spine, yet it will typically fail an `mswd` test owing to its departure from a Gaussian distribution. In this work, datasets of this nature are modelled using a contaminated Gaussian uncertainty distribution (Gaussian mixture), written $c\%d\mathrm{N}$, meaning that with a probability $(100-c)\%$ the distribution involves a standard deviation, σ, but with a probability $c\%$ the distribution has a standard deviation, $d\sigma$, both with a mean of zero (see Powell et al., 2002; Maronna et al., 2006, Sect. 2.1). An example is 25%3N, with $c=25$ and $d=3$, so that with 25% probability the uncertainty is drawn from $\mathrm{N}(0,3\sigma)$, and 75% probability drawn from $\mathrm{N}(0,\sigma)$, with the $\mathrm{N}(0,s)$ notation indicating a Gaussian distribution with a mean of zero and a standard deviation of s. Such distributions provide excess scatter suitable for developing and evaluating a robust line-fitting calculation. It does not matter if excess scatter in real data is drawn from a different contaminated Gaussian distribution. Note that with the sample sizes provided by most modern geochronological techniques, it is not possible to test for Gaussian behaviour, or such small departures from Gaussian behaviour.

2.2 Isochrons and errorchrons

In YORK, assuming that the data uncertainties are strictly Gaussian distributed, the probability distribution of `mswd` provides bounds that can be used to distinguish isochrons from errorchrons (e.g. Wendt and Carl, 1991). These bounds come from a 95% confidence interval on `mswd`, as discussed in Appendix A. Datasets whose scatter give `mswd` outside the bounds are deemed to be errorchrons, not isochrons. The focus in this paper is on `mswd` that is too large, indicating excess scatter. `Mswd` is defined with the residuals, r_k, the distance in y of the datapoint, k, to the line, e_k, weighted by the uncertainty on this distance, σ_{e_k}

$$\texttt{mswd} = \frac{1}{n-2}\sum_{k=1}^{n} r_k^2 \qquad \text{with} \qquad r_k = \frac{e_k}{\sigma_{e_k}} = \frac{a + b\,x_k - y_k}{b^2\,\sigma_{x_k}^2 + \sigma_{y_k}^2 - 2\,b\,\sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k}} \tag{1}$$

The line being fitted is $y = a + bx$; datapoint, k, is $\{x_k, y_k\}$; the analytical uncertainty on x_k, σ_{x_k}; the analytical uncertainty on y_k, σ_{y_k}; and the correlation between x_k and y_k, $\rho_{x_k y_k}$ (see derivation of eq. B4 in Appendix B). Note that the slope, b, appears in the denominator of r_k, as well as the numerator.

If, instead, data uncertainties are $c\%d\mathrm{N}$, with unknown c and d, or some other contaminated Gaussian distribution then there is no equivalent of the `mswd` argument to say which datasets should give isochrons rather than errorchrons. The approach advocated here is to use a measure that reflects whether the dataset has a linear spine of "good" data within it. The measure suggested, s, coined the spine width, is robust, and is defined as

$$s = \texttt{nmad}(r) = 1.4826\,\texttt{median}(|r_k - \texttt{median}(r)|) \tag{2}$$

with the constant normalising the result to be like the standard deviation for Gaussian-distributed r (e.g. Maronna et al., 2006, Sect. 2.4). Given that s is based on a median, its magnitude depends on that half of the data that have the smallest absolute

values of centred r, in other words those that would define a spine. If the data were in fact Gaussian-distributed, it is expected that s should be in a range about 1 in the same way that mswd is, given that r already involves the analytical uncertainties. The larger is s, greater than 1, the less pronounced is the linear spine in the data (or the uncertainties have been underestimated). Whereas the 95% confidence interval (95%ci) on mswd for Gaussian-distributed uncertainties comes from a well-established probability distribution, with $(n-2)\texttt{mswd} \sim \chi^2_{n-2}$ (e.g Wendt and Carl, 1991), the confidence interval on s needs to be found by simulation (see Appendix D), with the simulated datasets just involving Gaussian-distributed uncertainties. The intervals are given in this table, Table 1:

		95%ci	$\sqrt{\texttt{mswd}}$		95%ci	s
n	low	high	$*$	low	high	$*$
5	0.268	1.765	1.614	0.09	1.64	1.48
6	0.348	1.669	1.540	0.17	1.62	1.47
8	0.454	1.552	1.449	0.26	1.58	1.45
10	0.522	1.480	1.392	0.31	1.55	1.43
15	0.621	1.379	1.312	0.40	1.50	1.40
30	0.739	1.260	1.215	0.58	1.39	1.33
60	0.818	1.181	1.151	0.71	1.28	1.23

Whereas 1-sided confidence intervals are advocated in Appendix A—columns marked with an asterix in the Table—2-sided confidence intervals are also given in the table. Using the asterixed column, for example for a dataset with 10 datapoints ($n = 10$), the dataset is deemed to yield an isochron if the observed s is less than 1.43. If s is larger, the dataset gives an errorchron. For isochrons, the age uncertainty is calculated as in Appendix B. For errorchrons, the age uncertainty is not calculated.

2.3 A robust statistics approach to isochron calculation

We seek a statistical approach to isochron calculation that is *robust* (e.g. Huber, 1981; Hampel et al., 1986), meaning that it is not excessively affected by outliers in the data, while having desirable statistical properties, for example good efficiency (see below). In addition, we require the approach to converge to YORK for a "good" dataset, one with a near-Gaussian uncertainty distribution, allowing seamless compatibility with classical data interpretation. The overall approach adopted will be referred to as SPINE, involving the use of spine width for isochron-errorchron distinction, combined with robust line-fitting. The line-fitting is based on the approach of Huber (1981), as outlined in Maronna et al. (2006), Sect. 2.2.2. Whereas most robust line-fitting methods use the scatter of the data as a scale, data uncertainties having been discarded (e.g. Powell et al., 2002), here the data uncertainties are used. This is necessary in order to have continuity of the results with YORK, in which the data uncertainties are an integral part of the calculation.

In both the Huber approach in SPINE, and in YORK, a straight line is fitted to a dataset by minimising a function of the residuals, r_k. In the case of YORK this is just the mswd, (1). Since isochron data are generally bivariate with correlated

analytical uncertainties in x and y, the analytical uncertainty in datapoint k can be represented as an ellipse as in Fig. 2. The absolute value of the residual for datapoint, k, r_k, is in fact the scaling factor on the size of the ellipse required to expand it or reduce it until it touches the best-fit line (Fig. 2).

The function that is minimised to find the best-fit line can be written $\sum \rho(r_k)$ for both YORK and SPINE. Whereas in YORK, $\rho(r_k) = r_k^2$ for all r_k, in SPINE $\rho(r_k) = r_k^2$ near the centre of the uncertainty distribution (as in YORK), but downweights datapoints for which the absolute value of the residual is greater than a cut-off value, h. Thus, in SPINE, and in Fig. 3

$$\rho(r_k) = \begin{cases} 2\,h\,r_k - h^2 & r_k < -h \\ r_k^2 & \text{if} \quad -h < r_k < h \\ 2\,h\,r_k - h^2 & r_k > h \end{cases} \tag{3}$$

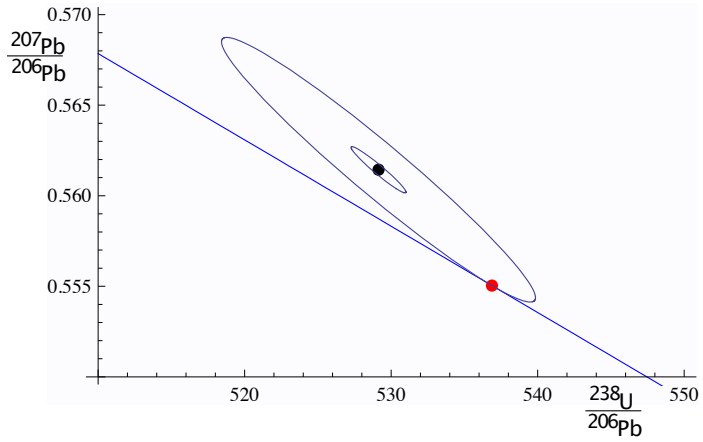

Figure 2. For an example datapoint, $\{x_k, y_k\}$, the inner ellipse is calculated with the analytical uncertainties, V_k, at the 1σ level (in black). Given a line, $y = a + bx$ (in blue), the ellipse must be drawn at the $|r_k|\sigma$ level (in red) to touch the line, in this case $|r_k| = 5.73$. The data point is $x_k = 529.14$, $y_k = 0.5614$, and $\sigma_{x_k} = 1.870$, $\sigma_{y_k} = 0.00127$ and $\rho_{x_k y_k} = -0.967$. The line is $y = 0.8108 - 0.0004764\,x$.

In SPINE, for residuals that have an absolute value less than an adjustable constant, h, the contribution to the sum being minimised is the same as for YORK, but it is linear in the residual for larger absolute value. Note that as h becomes larger and larger, SPINE converges to YORK. The value to use for h is discussed in Maronna et al. (2006), Sect. 2.2.2.

The iteration developed in Appendix B minimises $\sum_k \rho(r_k)$ with respect to the unknown, θ, a two-element column vector, $\{a, b\}^{\mathrm{T}}$ in the line equation, $y = a + bx$. The iteration is applicable to SPINE and also YORK. As a starting point of the iteration, a resistant estimate for θ is used (see Appendix B). However such methods are much less efficient than SPINE (see below), so SPINE is a better ultimate estimator. A full iteration is envisaged in Appendix B. The iteration converges in less than 5 iterations for all the simulations run (see Appendix D). Once θ is calculated, the measure of scatter used to distinguish an isochron from an errorchron can be calculated using Table 1. If an isochron is deemed to have been calculated, the uncertainty on θ, $\mathbf{V}_\theta$, can be found, as outlined in Appendix B.

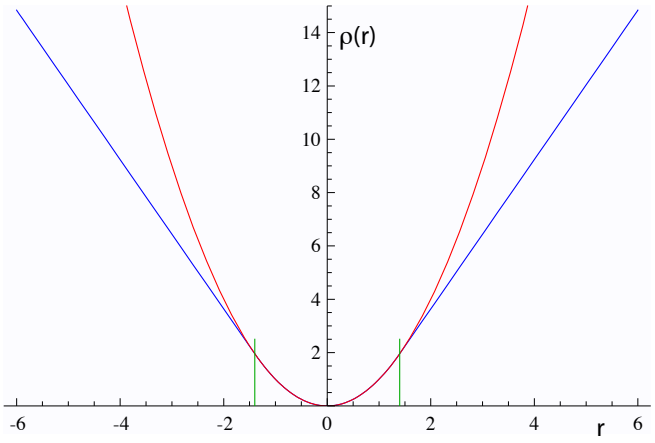

Figure 3. Plots of $\rho(r)$ against r for YORK in red (r^2), and for SPINE (eq. 3) in blue, with the two curves coincident for $|r| < h$, with $h = 1.4$ the vertical green lines. See text.

The SPINE algorithm can be summarised:

1. determine a preliminary resistant fit of the data using e.g. SIEGEL (Appendix A)
2. determine the spine width using `nmad`
   - if the spine width is in an acceptable range, from col. 6 in Table 1: isochron
   - if the spine width is not in the acceptable range: errorochron
3. determine a robust fit of the data, starting e.g. from SIEGEL, by minimising $\sum \rho(r_k)$, with ρ defined by (3)

2.4 Application of SPINE to simulated datasets

Assessing algorithms for data fitting is best done using simulated datasets. In this case, datasets were generated by drawing data points from a range of uncertainty distributions, all centred on a linear trend reflecting an age of 4 Ma. Full details are provided in Appendix D. Two features of the datasets are varied: the number of datapoints in the dataset, and the uncertainty structure adopted, the latter via varying c and d in $c\%d$N. The algorithm is assessed in terms of its ability to retrieve the specified age of the linear trend on which the simulated datasets are built, and on the uncertainty in the age.

Given that the datasets investigated have fat-tailed contaminated-Gaussian uncertainty distributions, the focus is on the effect of excess scatter in the data, in other words, data scatter over and above what is expected for Gaussian data uncertainties. Nevertheless a small proportion of datasets do have small scatter, giving s which is below the lower bound for that number of datapoints.

The analysis below compares the results of YORK, applied only to those simulated datasets that lie within the `mswd` bounds, with the results of SPINE, applied to those datasets that lie within the spine width (s) bounds. The greatest majority of the former

are included in the latter, e.g. $> 97\%$ for $n = 10$). Importantly, however, SPINE typically identifies the age information in many more datasets than YORK. In the following table, $m\%\text{excl}$ and $s\%\text{excl}$ are the percentage of simulated datasets excluded on the basis of the `mswd` and s bounds, respectively:

n	N		5%3N		25%3N		10%10N	
	$m\%\text{excl}$	$s\%\text{excl}$	$m\%\text{excl}$	$s\%\text{excl}$	$m\%\text{excl}$	$s\%\text{excl}$	$m\%\text{excl}$	$s\%\text{excl}$
5	2.5	2.5	8.7	4.0	30.2	9.8	32.5	9.5
6	2.5	2.5	9.6	3.9	34.6	13.8	37.3	10.9
8	2.5	2.5	12.7	4.2	44.7	14.5	46.0	10.4
10	2.5	2.5	14.2	4.0	51.8	15.2	53.5	9.7
15	2.5	2.5	17.4	4.2	65.2	17.1	68.2	9.1

Note that, for example, for $n = 10$, datasets drawn from 5%3N, in fact have $100(10^{0.95}) = 59.9\%$ of the datasets having all uncertainties Gaussian, and 40.1% having at least one uncertainty drawn from 3 times Gaussian (3N). For 25%3N, 5.6% are Gaussian only, and for 10%10N, 34.9%. The leftmost columns are 2.5% by definition.

The 95% confidence interval on ages derived using SPINE is the same or slightly less than those derived from using YORK for all dataset sizes and uncertainty structures studied, noting that this is from a (much) larger proportion of the dataset simulations. Such a 95% confidence interval is calculated from an ordered list of the ages, with the lower limit at the 2.5% point in the list, and the upper limit at the 97.5% point.

Even if the age comparison is favourable, it might be expected that the age uncertainty suffers from the excess scatter in the data. This appears not to be the case, but there is a small degradation in the age uncertainties retrieved caused by an unavoidable efficiency loss. Efficiency at the Gaussian distribution is the ratio of the variance obtained by the optimal estimator (YORK), divided by the variance using the chosen robust estimator (in this case, SPINE). Obviously, SPINE has optimal efficiency when all r in a dataset have $|r_k| < h$, when it is identical to YORK, but there is an efficiency loss associated with using SPINE for an isochron-yielding dataset with any $|r_k| > h$. In fact there is a trade-off between efficiency and robustness when the distribution is not strictly Gaussian, but is near Gaussian ([Maronna et al., 2006, Sect. 3.4).

The efficiency loss is illustrated in Figure 4 via kernel density estimate (`kde`) plots of the age uncertainties calculated for simulated datasets with $n = 10$. `Kde` plots are probability distributions akin to smoothed histograms (Wand and Jones, 1995). The red curve is the `kde` for datasets that have all $|r_k| < h$, for which efficiency is optimal. The blue curve is the `kde` for all datasets with at least one $|r_k| > h$. The efficiency loss is seen in the displacement of the blue curve to slightly higher age uncertainty than the red curve. The overall `kde`, in black, is the `kde` of all of the datasets in the red and blue `kde`, in observed proportion, about 30% to 70%. The relationships shown in Figure 4 for $n = 10$ can be seen for other n in Figure 5. The pairs of red and black lines, correspond to, and have the same meaning as the red and black lines in Figure 4. As expected, the distribution of age uncertainties moves towards larger values as the sample size n decreases.

The ability of SPINE to retrieve age uncertainty information for data with uncertainties from contaminated Gaussian distributions varies with the probability and scale of the contamination, as shown in Figure 6. Not unexpectedly the more seriously

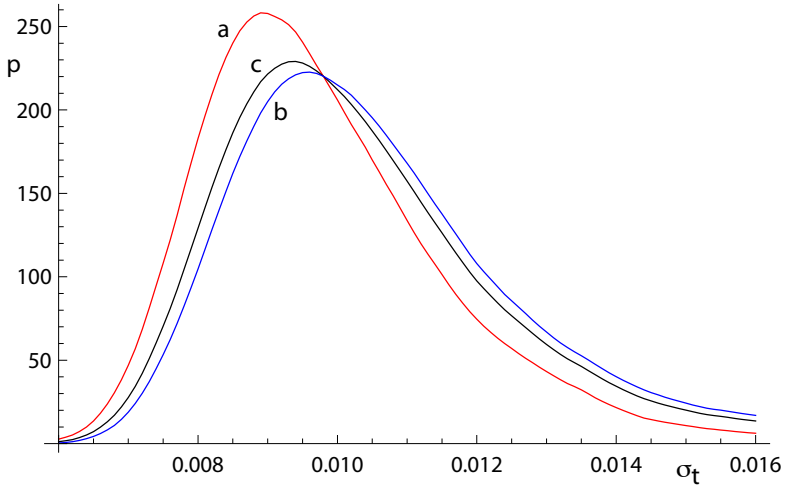

Figure 4. Kernel density estimates (kde) for age uncertainty calculated with SPINE on 10,000 simulated datasets with n=10 and Gaussian-distributed uncertainties. (a) Those datasets for which all $|r_k|<h$ (in red); (b) those datasets for which at least one $|r_k|>h$ (in blue), and (c) overall result combining a and b in observed proportion (in black).

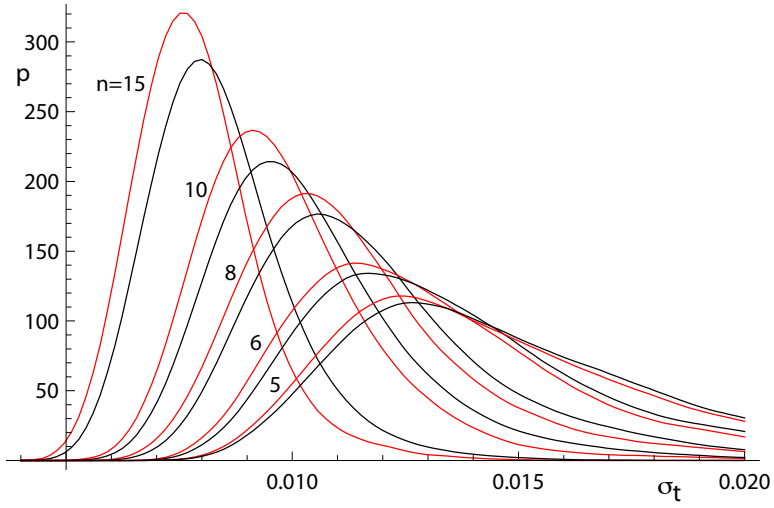

Figure 5. Kernel density estimates for age uncertainty calculated with SPINE on 10,000 simulated datasets with a range of n values and Gaussian-distributed uncertainties. In each case, the kde for those datasets for which all $|r_k|<h$ is in red and the overall kde is in black.

295 contaminated distributions (25%3N and 10%10N) involve a greater displacement of the kde to higher age uncertainty than the more weakly contaminated 5%3N distribution. Although the displacement of the blue curves from the black curve is real, nevertheless the ability of SPINE to retrieve age uncertainties from datasets with contaminated distributions is good.

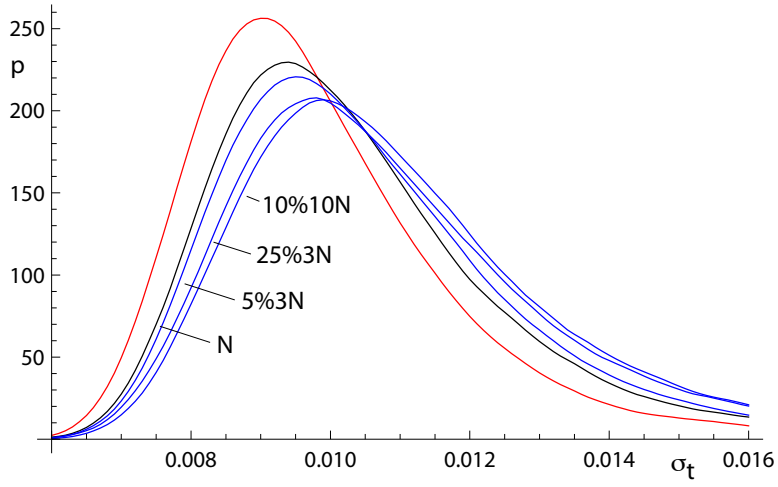

Figure 6. Kernel density estimates for age uncertainty calculated with SPINE on 10,000 simulated datasets with n=10 and several uncertainty structures. The `kde` for those datasets for which all $|r_k|<h$ is in red, the `kde` for datasets with Gaussian-distributed uncertainties is in black, and the `kde` for all of the datasets for 5%3N, 25%3N and 10%10N respectively are in blue.

2.5 Application of SPINE to a natural dataset

In order to show the real-world utility of SPINE, we show data for a carbonate flowstone from the Riversleigh World Heritage fossil site in Queensland, Australia (Sample 0708). Isotope dilution U-Pb data for the bulk sample were previously published by Woodhead et al. (2016) providing a Model 2 isochron with an age of 13.72 ± 0.12 Ma and with a `mswd` of 3.7. The new data presented here were obtained by laser ablation ICPMS on the same sample using methods outlined and published in Woodhead and Petrus (2019). Such datasets are typically larger with little error correlation (rounder error ellipses), but with larger uncertainties than isotope dilution data. These new data define an errorchron under the YORK assumptions, with `mswd` = 1.68, and a model 2 age of 13.68 ± 0.31 Ma. These data might therefore be rejected under the `mswd` criterion despite exhibiting a well-developed linear trend in Tera-Wasserburg isochron space. With SPINE, $s = 1.24$, within the s range for an isochron, the age is 13.69 ± 0.26 Ma ($\pm$ is 1.96σ). The data for 0708 are plotted in Fig. 7, with 95% confidence ellipses on the datapoints. Further calculations with this sample, comparing the results of our new algorithm with existing approaches are presented in Appendix A.

3 Discussion

This work was motivated by the belief that many isotopic datasets contain meaningful age information that cannot be identified using classical statistical methods. In such datasets, the age information is contained in a linear spine in the data, but the dataset also contains excess scatter that is inconsistent with a Gaussian uncertainty distribution. A statistical test based on the

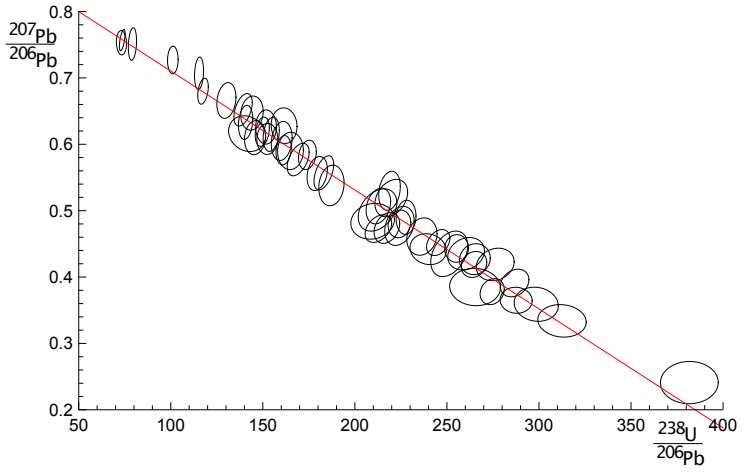

Figure 7. Laser ablation example 0708. See text.

spine width is devised, akin to using `mswd` in classical methods, allowing an isochron-errorchron distinction to be made. This distinction, conditional on the adoption of the spine width approach, means that such isochrons include many datasets that are errorchrons under a Gaussian uncertainty distribution assumption. A statistically-robust isochron calculation method is able to identify this age information and to provide appropriate uncertainty estimates. Calculated ages and age uncertainties are more reliable than ISOPLOT ones, given that least-squares-based methods are unreliable with excess scatter in the data.

Contaminated Gaussian distributions provide a model for a type of dataset with excess scatter relative to a strictly Gaussian-distributed one. The robust isochron method presented in this work can however be applied *in general* to data which is Gaussian-distributed only in the central spine of the uncertainty distribution, with non-Gaussian scatter occurring in the tails, arising from analytical or geological uncertainty.

In most robust statistics data fitting approaches, the formal uncertainties output during data measurement are ignored. Instead, the scale used in the data fitting is derived from the scatter in the data themselves, an approach adopted by Powell et al. (2002). The new approach followed here *does* include the data measurement uncertainties, and this allows the results to converge on those of YORK when the data have little excess scatter. This provides compatibility with older "good" datasets processed using the classical statistical approach but, going forward, allows age information to be extracted from a much wider range of datasets which might otherwise be rejected for having `mswd` greater than the isochron cutoff.

A problem with SPINE, shared with YORK, is that the effect of high-leverage data is not taken into account. Such data are easily recognised in x-y plots, when a small proportion of the data—even one datapoint—is separated from the main body of the data along the trend through the data. Data fitting tends to be overly constrained to fit high-leverage data, giving them small residuals, even if the best fit of the main body of the data alone would give the high-leverage data larger residuals. In 0708, the point at highest x is relatively high leverage (`hat` = 0.171). Robust approaches have been developed to handle high leverage data, e.g. Maronna et al. (2006), ch. 5, but are not yet developed for the situation where the data uncertainties are

taken into account, nor for the relatively small datasets that are typical of geochronological studies (c.f. Fig. 7). Huber and Rochetti (2009), ch. 7, have a counter view advocating data assessment, rather than aiming for a black-box method to try and automatically safeguard against the potentially deleterious effect of high leverage data, an approach we suggest here. In the case of the relatively high leverage datapoint in 0708, omitting this datapoint gives 13.75 ± 0.27 (compare row 13 with row 1 in the Table in Appendix A), within uncertainty of the age including this datapoint.

Appendix A: Algorithms and applications to sample 0708

Here are collected some results of calculations for sample 0708, used in Fig. 7, and some related algorithmic details.

thought experiment in Fig. 1

The thought experiment sketched in Fig. 1 aimed to show the consequence for ISOPLOT behaviour of the modification of the observed data in a dataset to reduce or increase the scatter of the data about the linear trend. The calculated equivalent of Fig. 1 for sample 0708 is shown in Fig. A1, including also the corresponding SPINE results. In calculating the Figure, the modification

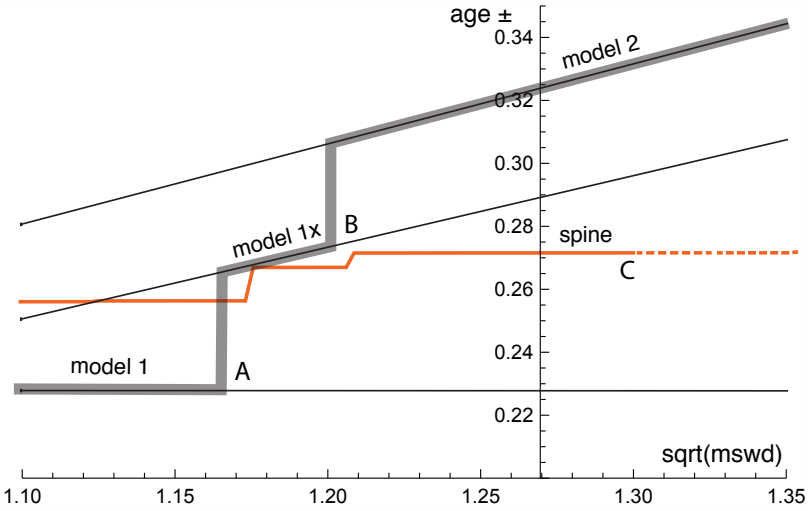

Figure A1. [new fig and caption] Age uncertainty (age±) plotted against $\sqrt{\texttt{mswd}}$ under the ISOPLOT protocol for the progressively modified dataset, 0708 (see text). In model 1, the age uncertainty is constant with increasing data scatter (reflected in increasing mswd), until there is a step change in age uncertainty at A when the ± is multiplied by $\sqrt{\texttt{mswd}}$. Then at B there is another step change with further increase of data scatter to model 2 (see text). The location of the step at A is based on a 95% confidence interval for mswd (discussed below), whereas the $\sqrt{\texttt{mswd}}$ position of the step at B is arbitrary. The y-axis is drawn at the $\sqrt{\texttt{mswd}}$ of the actual data, 1.27 (i.e. no modification of the data). See text.

of the dataset is done by first taking the data points with their attendant error ellipses (i.e. covariance matrices), and moving them all in to lie on the linear trend, considered as fixed by a YORK calculation. Then the points and ellipses are considered to

be displaced away from the trend. This is "move" in the Table below, going from -1, when the points lie on the trend, through 0, with the points as in the original data, to positive when displaced further away from the trend. "Move" varies more or less linearly with $\sqrt{\texttt{mswd}}$ from -0.133 at the left-edge of the Figure, to 0.064 on the right-edge; the spine width changes from 1.08 to 1.32 across the Figure. For these calculations the last line in the dataset is omitted as it is relatively high leverage (`hat` = 0.171), not wishing this datapoint to affect the results.

In the Figure, extending from the left, through `mswd` = 1, to A, the ISOPLOT age uncertainty (model 1, i.e. YORK) is constant because the data scatter is consistent with the data uncertainties. Through this range the SPINE age uncertainty is above the model 1 line because of the efficiency loss embodied in SPINE, as shown in Figs 4–5. However, after the age uncertainty steps with increasing `mswd`, to the right of the diagram, the SPINE age uncertainty is smaller than the model 1x and model 2 age uncertainty. This is because SPINE gives an isochron on the basis of spine width up to C at $\sqrt{\texttt{mswd}} \approx 1.3$, whereas model 2 is an errorchron, on the basis of the assumption of strictly Gaussian data uncertainties. The small steps in the SPINE age uncertainty line are an artefact of the approximation used in the calculation of the age uncertainty, see Appendix B.

In the top part of the following Table, the results for SPINE and for ISOPLOT are summarised. The Δ column gives the change to the age from the SPINE age, normalised by the uncertainty on the SPINE age. Below the double line in the Table, are some results from the above thought experiment, Fig. A1.

		move	$\sqrt{\texttt{mswd}}$	age	age±	Δ	notes
1	SPINE			13.685	0.257	–	s = 1.24
2	YORK		1.296	13.733	0.216	0.37	outside 95% c.l.
3	ISOPLOT model 1x			13.733	0.280		
4	ISOPLOT model 2			13.679	0.306	−0.05	
5	siegel			13.803		−0.95	
6	$L1$			13.518		1.23	
7	lsq			13.608	0.307	0.54	with eiv
8	SPINE	−0.133	1.1	13.769	0.252		$s = 1.08$
9	YORK			13.800	0.223	−0.39	inside 95% c.l.
10	SPINE	−0.055	1.2	13.756	0.262		$s = 1.18$
11	ISOPLOT model 1x			13.800	0.268	−0.39	
12	ISOPLOT model 2			13.832	0.300	−0.62	
13	SPINE	0	1.27	13.747	0.267	–	$s = 1.25\ \Delta = 0.366$
14	ISOPLOT model 2			13.836	0.317	−0.65	
15	SPINE	0.024	1.3	13.743	0.267		$s = 1.26$
16	ISOPLOT model 2			13.837	0.325	−0.66	

Additional algorithmic details for ISOPLOT follow next, in part related to the above thought experiment.

ISOPLOT model 1

In the ISOPLOT *model 1* calculation, i.e. in YORK, a decision has to be made about the confidence interval on `mswd` that is used to denote the range of data scatter (i.e. `mswd`) that is considered to be accounted for by the data uncertainties, without need to either multiply the age uncertainty by $\sqrt{\texttt{mswd}}$ (i.e. model 1x), or switch directly to an alternative calculation (which is model 2 in ISOPLOT). On the understanding that data uncertainties are correctly assigned, a one-sided confidence interval on `mswd` can be adopted, acknowledging that `mswd` is not being used to identify the case where assigned data uncertainties are too large or too small. The upper end of the confidence interval on `mswd` is where excess scatter is considered to start, a conventional choice being derived from a 95% confidence interval. Note that there is no argument that this should be at $\texttt{mswd} = 1$ (c.f. Dickin, 2005, p37), unless the number of datapoints is huge. Even for a dataset of 50 datapoints the 95% confidence interval on `mswd` extends to 1.358. In terms of probability of fit, as used in ISOPLOT, this is just $100 - 95 = 5\%$. It might be noted that the naming of probability of fit seems unhelpful - it is clearer to focus on `mswd`.

ISOPLOT model 2

The so-called *error-in-variables* (`eiv`) or measurement-error problem is avoided in YORK because the uncertainty in the x variable is taken into account explicitly. If it was not, then `eiv` results in the calculated slope being biased downwards and the approach being inconsistent (e.g. Fuller, 1987).

In the ISOPLOT *model 2* calculation, `eiv` is avoided, even though the data uncertainties are discarded, by making the slope of the line through the data be the geometric mean of the slopes of ordinary least squares of y on x, b_{yx}, and x on y, $1/b_{xy}$. These are

$$b_{yx} = \frac{\sum(x_k - \overline{x})(y_k - \overline{y})}{\sum(x_k - \overline{x})^2} \qquad \text{and} \qquad \frac{1}{b_{xy}} = \frac{\sum(x_k - \overline{x})(y_k - \overline{y})}{\sum(y_k - \overline{y})^2}$$

with $\overline{x} = \frac{1}{n}\sum x_k$ and $\overline{y} = \frac{1}{n}\sum y_k$. Then

$$b = \pm\sqrt{b_{yx} b_{xy}} = \pm\sqrt{\frac{\sum(y_k - \overline{y})^2}{\sum(x_k - \overline{x})^2}} \qquad \text{and} \qquad a = \overline{y} - b\overline{x}$$

with, in this case, the sign of the square root being negative. The calculation in ISOPLOT does not use this explicit formula, instead adopting an algebraic equivalent that allows the YORK iteration to be used.

ISOPLOT robust

In ISOPLOT, there is an option to use a *robust* isochron calculation method. The two available are, strictly, resistant methods, having high breakdown point but low efficiency (e.g. Huber, 1981, Sect. 1.2.3). The second method—(Siegel, 1982)—can be considered to supercede the first. In fact, here, SIEGEL is used in the implementation of SPINE as a possible starting point for the iteration, along with least absolute deviations, $L1$, Sadovski (1974) (see Appendix C). The fit of the data for sample 0708 with SIEGEL and $L1$, along with that for ordinary least squares, are given in lines, 5–7, of the Table.

Appendix B: SPINE iteration

The SPINE algorithm involves minimising $\sum_k \rho(r_k)$ with respect to the unknown, θ, a two-element column vector, $\{a, b\}^{\mathrm{T}}$ in the line equation, $y = a + bx$, in order to fit the data. The residual, r_k on datapoint k is defined below, and the function ρ is defined in (3) in the main text. The SPINE iteration subsumes YORK.

Writing the kth datapoint as $\{x_k, y_k\}$, generally the isotopic data used in isochron calculations involve uncertainties in both x_k and y_k, and commonly the x_k and y_k are also correlated. These can be represented by a covariance matrix, V_k,

$$V_k = \begin{bmatrix} \sigma_{x_k}^2 & \sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k} \\ \sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k} & \sigma_{y_k}^2 \end{bmatrix} \tag{B1}$$

in which σ_{x_k} is the standard deviation on x_k, σ_{y_k} the standard deviation on y_k, $\sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k}$ the covariance between x_k and y_k, and $\rho_{x_k y_k}$ the correlation coefficient between x_k and y_k. The covariance matrix can be represented by an ellipse around the data point in an x–y diagram, as illustrated in Fig. 2. The residual, r_k, a measure of the distance of the point $\{x_k, y_k\}$ to the line, is calculated from the coordinates of the data points, $\{x_k, y_k\}$, and their uncertainties in V_k, by

$$r_k = \frac{e_k}{\sigma_{e_k}} \tag{B2}$$

in which e_k is the distance of the datapoint from the line, $e_k = a + b\,x_k - y_k$, and σ_{e_k} is the standard deviation on e_k. The standard deviation, σ_{e_k}, is calculated by error propagation using V_k:

$$\sigma_{e_k}^2 = \left\{ \frac{\partial e_k}{\partial x_k}, \frac{\partial e_k}{\partial y_k} \right\} V_k \left\{ \frac{\partial e_k}{\partial x_k}, \frac{\partial e_k}{\partial y_k} \right\}^{\mathrm{T}} = b^2\sigma_{x_k}^2 + \sigma_{y_k}^2 - 2\,b\,\sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k} \tag{B3}$$

with the term in curly brackets evaluating to $\{b, -1\}$. The residual is then

$$r_k = \frac{e_k}{\sigma_{e_k}} = \frac{a + b\,x_k - y_k}{\sqrt{b^2\sigma_{x_k}^2 + \sigma_{y_k}^2 - 2\,b\,\sigma_{x_k}\sigma_{y_k}\rho_{x_k y_k}}} \tag{B4}$$

The minimisation of $\sum_k \rho(r_k)$ is iterative, starting from a resistant estimate of the line, for example using least absolute deviations, L_1, as advocated by Maronna et al. (2006), or Siegel (1982). At each iteration, an update of θ, $\Delta\theta$, is generated so that at the ith iteration, $\theta_i = \theta_{i-1} + \Delta\theta$.

The minimisation of $\sum \rho(r_k)$ is undertaken using the fact that, at the minimum, the derivative of $\sum \rho(r_k)$ with respect to θ is zero. Defining

$$2\,\psi(r_k) = \frac{\partial \rho(r_k)}{\partial r_k} \tag{B5}$$

this function, for (3) in SPINE, is

$$\psi(r_k) = \begin{cases} -h & \\ r_k & \text{if} \quad -h < r_k < h \\ h & \end{cases} \qquad \begin{array}{l} r_k < -h \\ \\ r_k > h \end{array}$$

For YORK, $\psi(r_k) = r_k$, equivalent to SPINE with large h. Then, at the minimum

$$\sum_k \frac{\partial \rho(r_k)}{\partial \theta} = 0 = \sum_k \left(\frac{\partial \rho(r_k)}{\partial r_k} \right) \left(\frac{\partial r_k}{\partial \theta} \right) = \sum_k \psi(r_k) \left(\frac{\partial r_k}{\partial \theta} \right) \tag{B6}$$

Defining the kth row of a matrix $\mathbf{C}$, C_k, to be the derivative of r_k with respect to θ then

$$C_k = \frac{\partial r_k}{\partial \theta} = \frac{1}{\sigma_{e_k}} \frac{\partial e_k}{\partial \theta} - r_k \frac{\partial \sigma_{e_k}}{\partial \theta} = B_k - r_k \frac{\partial \sigma_{e_k}}{\partial \theta} \tag{B7}$$

with B_k the kth row of $\mathbf{B}$, given by $1/\sigma_{e_k}\, \partial e_k / \partial \theta$, then at the minimum

$$\sum_k \psi(r_k) \frac{\partial r_k}{\partial \theta} = \sum_k \psi(r_k)\, C_k = 0 \tag{B8}$$

or in matrix form, $\mathbf{C}^{\mathrm{T}} \psi(\mathbf{r}) = 0$, in which $\psi(\mathbf{r})$ is a column vector whose kth element is $\psi(r_k)$. This constitutes two non-linear equations requiring iteration to solve.

Now, at iteration i, progressing towards the minimum, writing $\theta_i = \theta_{i-1} + \Delta\theta$

$$\psi_i(r_k)|_{|r_k|<h} = B_k(\theta_{i-1} + \Delta\theta) - \frac{y_k}{\sigma_{e_k}} = \psi_{i-1}(r_k) + B_k\, \Delta\theta \tag{B9}$$

and $\psi_i(r_k)|_{|r_k|>h} = \psi_{i-1}(r_k)$ otherwise. This can be written

$$\psi_i(r_k) = \psi_{i-1}(r_k) + \dot{\psi}_{i-1}(r_k)\, B_k\, \Delta\theta \tag{B10}$$

in which $\dot{\psi}(r_k) = \partial \psi(r_k)/\partial r_k$. So, for SPINE, $\dot{\psi}(r_k) = 1$ for $|r_k| < h$, and $\dot{\psi}(r_k) = 0$ otherwise. Substituting (B10) into (B8) gives

$$\sum (\psi_{i-1}(r_k) + \dot{\psi}_{i-1}(r_k) B_k \Delta\theta)\, C_k = 0 \tag{B11}$$

or, in matrix form, dropping iteration subscripts

$$\mathbf{C}^{\mathrm{T}} (\mathbf{I}'\, \mathbf{B}\, \Delta\theta + \psi(\mathbf{r})) = 0 \tag{B12}$$

with $\mathbf{I}' = \mathrm{diag}(\dot{\psi}(\mathbf{r}))$ a modified identity matrix with its kkth element equal to $\dot{\psi}(r_k)$. Equation (B12) can then be rearranged to give $\Delta\theta$ at the current iteration

$$\Delta\theta = -(\mathbf{C}^{\mathrm{T}} \mathbf{I}'\, \mathbf{B})^{-1} \mathbf{C}^{\mathrm{T}} \psi(\mathbf{r}) \tag{B13}$$

The iteration works because the changes in $\mathbf{B}$ and $\mathbf{C}$ between iterations are small, particularly when a good starting guess is used at the beginning of the iterations. This is the iteration implemented in the python code.

Accepting that an isochron has been calculated, the covariance matrix of θ, $\mathbf{V}_\theta$, can be calculated by error propagation of $\mathbf{r}$ to θ

$$\mathbf{V}_\theta = \left(\frac{\partial \theta}{\partial \mathbf{r}} \right) \mathbf{V}_\mathbf{r} \left(\frac{\partial \theta}{\partial \mathbf{r}} \right)^{\mathrm{T}} \tag{B14}$$

assuming that θ is approximately linear in r_k around the minimum in $\sum \rho(r_k)$. Then, using (B14) with (B13)

$$\mathbf{V}_\theta = \left(\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{B}\right)^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{V}_\mathbf{r}\mathbf{I}'\mathbf{C}\left(\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{B}\right)^{-\mathrm{T}} \tag{B15}$$

If it is assumed that the uncertainty on a datapoint has the form $c\%d\mathrm{N}$, with unknown c and d, then $\mathbf{V}_r$ is not specified. However those residuals with $|r_k| > h$ are likely to be those where $d > 1$, but these residuals are the ones with $\dot{\psi}(r_k) = 0$. Therefore a good approximation involves taking $\mathbf{V}_r = \mathbf{I}'$. This is identically true in the case of YORK, when $\mathbf{V}_r = \mathbf{I}$ as then $\mathbf{I}' = \mathbf{I}$. Taking $\mathbf{V}_r = \mathbf{I}'$, (B15) becomes

$$\mathbf{V}_\theta = \left(\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{B}\right)^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{C}\left(\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{B}\right)^{-\mathrm{T}} \tag{B16}$$

This result is equivalent to York (1969), whereas the equivalent result from Titterington and Halliday (1979), as outlined in York et al. (2004), involves replacing $\mathbf{B}$ in (B16) with $\mathbf{C}$, resulting in

$$\mathbf{V}_\theta = \left(\mathbf{C}^{\mathrm{T}}\mathbf{I}'\mathbf{C}\right)^{-1} \tag{B17}$$

The small steps in the age uncertainty (age$\pm$) curve in Fig A1 arise because of the assumption involved in using $\mathbf{I}'$ in (B16). When diagonal elements in $\mathbf{I}'$ change from one to zero as `mswd` increases in Fig A1, the small steps result.

In YORK (or if all $|r_k| < h$ in SPINE), then $\mathbf{I}' = \mathbf{I}$ and $\psi(\mathbf{r}) = \mathbf{r}$. So

$$\Delta\theta = -(\mathbf{C}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{r} \tag{B18}$$

and the covariance matrix becomes

$$\mathbf{V}_\theta = (\mathbf{C}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{C}(\mathbf{C}^{\mathrm{T}}\mathbf{B})^{-\mathrm{T}} \tag{B19}$$

or, following York et al. (2004)

$$\mathbf{V}_\theta = (\mathbf{C}^{\mathrm{T}}\mathbf{C})^{-1} \tag{B20}$$

If, in addition, all $\sigma_{x_k} = 0$ then $\mathbf{C} = \mathbf{B}$ and, as in this case, $\mathbf{B}$ does not depend on θ, iteration is not involved, $\mathbf{r}$ is replaced by $-\mathbf{y}$, and

$$\theta = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{y} \tag{B21}$$

with a covariance matrix of

$$\mathbf{V}_\theta = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1} \tag{B22}$$

These are the results for fitting data by simple weighted least squares.

Appendix C: SPINE Python code

The iteration in Appendix B is coded in the Python function, `huber`. The starting point for the iteration is preset to be the best preliminary fit of the data by $L1$ (function `lad`, Sadovski, 1974), Siegel (function `siegel`, Siegel, 1982), and ordinary least squares. The user may call the `huber` function with additional possible starting points. Best preliminary fit is defined as the one that gives the smallest $\sum \rho(r_k)$ as in eq. A1. At the ith iteration in the `huber` function, interval halving is used to get a steplength, `step`, to apply in $\theta_i = \theta_{i-1} + \mathrm{step}\,\Delta\theta$. This was sufficient for all the simulations run, but may need to be made more sophisticated for datasets for which the iteration does not converge. The calling function, `recipe`, is a placeholder for a more general function to be written by the user.

```
import sys
import datetime
import numpy as np

out = open("out.txt", "w") # opening output file
screen = sys.stdout
standard = [screen, out]   # default where print goes

defaulth = 1.4  # default h in huber

def nmad(e):
    return 1.4826 * np.median(np.absolute(e - np.median(e)))

def lsq(data): # ordinary least squares
    X = [[1, xk] for xk in data[:,0]]
    Y = data[:,2]
    inv = np.linalg.inv(np.dot(np.transpose(X), X))
    theta = np.dot(inv, np.dot(np.transpose(X), Y))
    e = np.dot(X, theta) - Y
    sigfit2 = np.dot(e, e)/(e.shape[0] - 2)
    return (theta, sigfit2 * inv)

def lad(data):  # L1 - sadovski (1974)
    n = data.shape[0]
    (x, sdx, y, sdy, cor) = np.transpose(data)
    rr = 1e-8 * np.random.random(n-1) # used for naive breaking of x ties
    bi = np.empty(n); bi.fill(False)
    k = 0; i = 0; i1 = 0; i2 = 0;
    while i != -1 and k < 12:
        i2 = i1; i1 = i; k += 1
        o = np.delete(np.arange(n), i);
        x1 = np.delete(x - x[i], i) + rr
        y1 = np.delete(y - y[i], i)
```

```
        oo = np.argsort(y1/x1)
        x2 = np.abs(x1[oo]);  mid = sum(x2)/2
        sx = 0; j = 0
        while sx < mid: sx += x2[j]; j += 1
        i = o[oo][j-1]
        if i == i2: i = -1
        elif bi[i]: i = -1
        else: bi[i] = True
    return np.array(((x[i1] * y[i2] - x[i2] * y[i1])/(x[i1] - x[i2]),
                     (y[i1] - y[i2])/(x[i1] - x[i2])))

def siegel(data):  # siegel (1982)
    n = data.shape[0];
    (x, sdx, y, sdy, cor) = np.transpose(data)
    x += 1e-8 * np.random.random(n) # naive breaking of x ties
    med = np.empty(n);
    for i in range(n):
        col = np.empty(n); # col.fill(0)
        for j in range(n):
            if i is not j: col[j] = (y[j] - y[i])/(x[j] - x[i])
         med[i] = np.median(np.delete(col, i))
    b = np.median(med)
    return [np.median(y - x * b), b]

def calcage(theta, covtheta = None):
    # add age and age uncertainty calculation code

def rhohub(r, h = 1.4):
    v = [rk**2 if abs(rk) < h else 2 * h * abs(rk) - h **2 for rk in r]
    return np.array(v)

def psihub(r, h = 1.4):
    v = [rk if abs(rk) < h else np.sign(rk) * h for rk in r]
    return np.array(v)

def dpsihub(r, h = 1.4):
    v = [1 if abs(rk) < h else 0 for rk in r]
    return np.array(v)

def sumrho(data, theta, h = 1.4):
    (a, b) = theta
    (x, sdx, y, sdy, cor) = np.transpose(data)
    e = a + b * x - y
    sde = np.sqrt(sdy**2 + b**2 * sdx**2 - 2*b*cor*sdx*sdy)
```

```
    r = e / sde
    return np.sum(rhohub(r, h))

def sumpsi2(data, theta, h = 1.4):
    (a, b) = theta
    (x, sdx, y, sdy, cor) = np.transpose(data)
    e = a + b * x - y
    sde = np.sqrt(sdy**2 + b**2 * sdx**2 - 2*b*cor*sdx*sdy)
    r = e / sde
    c = np.transpose([1/sde, (x - r * (b*sdx**2 - cor*sdx*sdy)/sde)/sde])
    pc = np.dot(np.transpose(c), psihub(r, h))
    return np.sqrt(np.dot(pc, pc))

def halving(data, theta, deltheta, m, h = 1.4):
# naive interval halving for a steplength, assumes < 1
    kmax = 16
    step1 = 0
    s1 = sumrho(data, theta, h)
    step2 = 1
    s2 = sumrho(data, theta + step2 * deltheta, h)
    k = 1
    while (k < m or step1 < 1e-10) and k < kmax:
        if s1 > s2:
            step1 = (step1 + step2)/2
            s1 = sumrho(data, theta + step1 * deltheta, h)
        else:
            step2 = (step1 + step2)/2;
            s2 = sumrho(data, theta + step2 * deltheta, h)
        (steps, ss) = (step2, s2) if s1 > s2 else (step1, s1);
        k += 1
    return (steps, ss)

def huber(data0, h = 1.4, th00 = []):  #   huber line-fitter
    n = data0.shape[0]
    itmax = 12; minsump = 1e-5; mindel = 1e-8; mincond = 1e-12
    code = 0;
    (x, sdx, y, sdy, cor) = np.transpose(data0)
    avx = np.dot(x, np.ones(n))/n;  avy = np.dot(y, np.ones(n))/n;
    div = np.array([1/avy, avx/avy])
    data = np.copy(data0)
    (x, sdx, y, sdy, cor) = np.transpose(data)
    x /= avx; sdx /= avx;  y /= avy; sdy /= avy
    th0 = (lad(data), siegel(data), lsq(data)[0])
    [th0.append(thetak * div) for thetak in th00]
```

```
    sr = np.array([sumrho(data, thetak) for thetak in th0])
    theta = th0[np.argsort(sr)[0]]
    sump = 1e10; step = 1; deltheta = (1e10, 1e10)
    k = 0; bb = 0
    while k < itmax and (sump > minsump or \
            np.sqrt(np.dot(deltheta,deltheta)) > mindel):
        k += 1
        (a, b) = theta
        e = a + b * x - y
        sde = np.sqrt(b**2*sdx**2 - 2*b*cor*sdx*sdy + sdy**2)
        r = e/sde
        sum = sumrho(data, theta, h)
        c = np.transpose([1/sde, (x - r * (b*sdx**2 - cor*sdx*sdy)/sde)/sde])
        rs = psihub(r, h)
        pc = np.dot(np.transpose(c), rs)
        sump = np.sqrt(np.dot(pc, pc)) # same as given by sumpsi2
        drs = dpsihub(r, h)
        d = np.transpose([drs/sde, drs * x/sde])
        cd = np.dot(np.transpose(c),d)
        (uu, sv, vv) = np.linalg.svd(cd)
        if mincond * sv[0] > sv[1]: code = 2; break
        inv = np.dot(np.dot(np.transpose(vv), np.diag(1/sv)), np.transpose(uu))
        bb = np.dot(inv, np.transpose(c))
        deltheta = np.dot(bb, -rs)
        sum1 = sumrho(data, theta + deltheta, h);
        (step, sum1) = (1, sum1) if sum1 < 1.01 * sum else \
                                    halving(data, theta, deltheta, 4, h)
        if step == 0: code = 3; break
        theta += step * deltheta
    sump = sumpsi2(data, theta, h)
    if step == 0 and sump < np.sqrt(minsump):
        code = 0 # not fully converged, but nearly good: ok?
    if sump > 10: code = 1
    rbb = np.dot(bb, np.diag(drs))
    theta /= div
    covtheta = np.dot(rbb, np.transpose(rbb)) / \
        np.array([[div[0]**2, div[0] * div[1]], [div[0] * div[1],div[1]**2]])
    return (code, theta, covtheta, sump, k, sv[1])

def recipe(title, data, where = [screen]):  # simple calculation driver
    h = defaulth
    (x, sdx, y, sdy, cor) = np.transpose(data)
    n = data.shape[0]
    today = datetime.datetime.now();
```

```
    pr("============================================================\n"+ \
       "running huber.py on "+today.ctime())
    res = huber(data)
    if res[0] != 0: return res(0) # exit if not converged
    (a, b) = theta = res[1]
    ucovtheta = res[2]
    (age, sdage) = calcage(theta, ucovtheta)
    e = a + b * x - y
    sde = np.sqrt(b**2 * sdx**2 - 2 * b * sdx * sdy * cor + sdy**2)
    s = nmad(e/sde); slim = 1.92 - 0.162 * np.log(10 + n)
    iso = ": isochron " if s < slim else ": errorchron "
    pr(("sample "+title+": s = %0.2f"+ iso + "age = %0.3f +/- %0.3f Ma") % \
       (s, age, 1.96 * sdage), printto = where)
    return [0, age, sdage, theta]

def pr(s, e="\n", printto=standard):  # prints a string
    for pr in printto: print(s, end=e, file=pr)

def pra(x, f, s="", e="\n", printto=standard):  # prints an array
    for pr in printto:
        print(s, end='', file=pr)
        for xk in np.array(x).flatten():
            print(f % xk, end='', file=pr)
        print(e, file=pr)

# data rows:  x  sdx  y  sdy  cor
data2 = np.loadtxt("data0708.txt", delimiter=",")
recipe("0708", data2, where=standard)
```

Datafile for sample 0708, data0708.txt, see Fig. 7

```
 73.2064, 1.12543, 0.753, 0.0075, -0.068224
 260.417, 4.06901, 0.435, 0.01, 0.18853
 169.205, 2.43357, 0.577, 0.01, 0.45644
 79.1766, 0.908995, 0.751, 0.01, 0.19585
 212.766, 2.94251, 0.473, 0.0085, 0.37494
 154.56, 1.79165, 0.615, 0.0105, 0.22567
 217.391, 2.83554, 0.474, 0.0095, 0.27879
 209.644, 4.83455, 0.484, 0.011, 0.10969
 144.092, 2.49151, 0.647, 0.0105, 0.10222
 174.216, 1.97283, 0.584, 0.009, 0.21416
 224.215, 3.26771, 0.477, 0.012, 0.26464
 236.407, 3.35329, 0.461, 0.0115, 0.14718
 161.551, 2.87086, 0.628, 0.011, -0.041164
 265.252, 5.62869, 0.385, 0.0115, -0.043072
```

```
685   152.439, 2.32377, 0.608, 0.0095, 0.019543
      151.745, 2.30266, 0.626, 0.0105, 0.0030752
      101.112, 1.17572, 0.727, 0.0085, 0.038467
      265.182, 3.41058, 0.427, 0.0095, 0.070384
      286.78, 3.16634, 0.391, 0.0085, 0.24059
690   287.604, 3.5568, 0.365, 0.008, -0.041298
      264.69, 2.83747, 0.419, 0.008, 0.15939
      274.574, 2.67638, 0.378, 0.008, 0.17686
      212.314, 4.28235, 0.501, 0.013, 0.34121
      161.29, 1.69095, 0.591, 0.009, 0.14398
695   140.647, 1.58253, 0.633, 0.0105, 0.18753
      183.15, 2.18036, 0.557, 0.0105, 0.49436
      218.818, 2.39407, 0.53, 0.012, 0.21725
      312.5, 5.3711, 0.334, 0.01, -0.080507
      227.79, 2.20526, 0.49, 0.0105, 0.073412
700   212.766, 2.71616, 0.507, 0.011, 0.25544
      139.276, 2.03676, 0.652, 0.01, 0.44912
      179.533, 2.25625, 0.556, 0.0105, 0.24303
      224.215, 2.51362, 0.48, 0.0085, 0.11165
      219.78, 3.62275, 0.519, 0.0115, 0.31258
705   165.017, 2.8592, 0.59, 0.0115, 0.03575
      145.773, 2.33746, 0.61, 0.0105, 0.098985
      239.808, 4.02556, 0.442, 0.0095, -0.10478
      187.266, 2.80548, 0.538, 0.0125, 0.11864
      255.232, 2.63831, 0.443, 0.0085, 0.025538
710   141.443, 4.10124, 0.616, 0.011, -0.15512
      115.34, 0.997753, 0.707, 0.01, 0.15087
      117.509, 1.17371, 0.68, 0.008, 0.33891
      73.6377, 0.623589, 0.757, 0.0065, 0.57883
      160.256, 2.31139, 0.605, 0.012, 0.16382
715   149.701, 1.56872, 0.619, 0.009, 0.24589
      245.278, 2.58694, 0.452, 0.008, 0.33748
      251.256, 4.10343, 0.435, 0.014, 0.3589
      130.208, 2.11928, 0.666, 0.011, 0.18376
      276.243, 4.19706, 0.419, 0.01, 0.16668
720   298.508, 4.90087, 0.359, 0.0105, -0.090909
      381.679, 6.40989, 0.241, 0.013, 0.0073965
```

Example output, running on the command line

```
running huber.py on Tue Sep  3 09:47:51 2019
sample 0708: s = 1.19: isochron age = 13.685 +/- 0.260 Ma
```

Appendix D: Simulation setup

This work was originally motivated by the dating of speleothems using the lower intercept with a U-Pb Concordia in Tera-Wasserburg style plots (Woodhead et al., 2012). This paper therefore discusses $\{x, y\}$ data with the expectation that $x = {}^{238}\mathrm{U}/{}^{206}\mathrm{Pb}$ and $y = {}^{207}\mathrm{Pb}/{}^{206}\mathrm{Pb}$, but the logic and the algorithm are in no way restricted to this system.

10,000 simulated datasets, each containing 5, 6, 8, 10 and 15 datapoints, respectively, were used to assess SPINE. Each dataset corresponds to an age of 4 Ma, with an underlying trend chosen to be $y = 0.811 - 0.000474737x$. For each dataset, the x-values were drawn from a uniform probability distribution with bounds, $\{400, 1100\}$ (so the x are not equi-spaced). Datapoints are assigned uncertainty with $\sigma_{x_k} = 0$ and a fixed $\sigma_{y_k} = 0.00125$, the latter representing the analytical uncertainty, propagated from both the x and y measurement into y. In a real $\{{}^{238}\mathrm{U}/{}^{206}\mathrm{Pb}, {}^{207}\mathrm{Pb}/{}^{206}\mathrm{Pb}\}$ dataset, σ_{x_k} and σ_{y_k} would be finite and correlated. However, this makes no difference to the calculations once data is processed into r_k form as in Fig. 2. For a given dataset, scatter is introduced into the data by drawing the y values from an uncertainty distribution, centred on the underlying trend, that may be either Gaussian (N) or one of three contaminated Gaussian distributions—$5\%3\mathrm{N}$, $25\%5\mathrm{N}$, or $10\%10\mathrm{N}$—as in Powell et al. (2002). For $n = 10$ and gaussian-distributed uncertainties, the age uncertainty obtained is approximately $\sigma_t = 0.01$ Ma.

Results are presented in terms of kernel density estimates using an Epanechnikov kernel (Wand and Jones, 1995). Kernel density estimates (`kde`) are a way of presenting data that could otherwise be plotted as a histogram, normally normalised so that—like a probability distribution—the area under the `kde` curve is 1. The smoothness of the `kde` is controlled by a smoothing constant whose value was chosen to be just large enough for the `kde` to appear smooth, given that 10,000 datasets are used in each `kde`.

Author contributions. Roger Powell created the approach and coded the Python script; Eleanor Green helped validate the maths/statistics and write the paper; Tephy Marillo Sialer helped with the simulations; and Jon Woodhead oversaw the applicability of the approach.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We would like to thank the anonymous reviewers for their work. Tim Pollard has materially helped in our understanding of the ISOPLOT code functionality. JW is funded by Australian Research Council grant FL160100028.

References

Brooks, C., Hart, S.R., and Wendt, I.: Realistic use of two-error regression treatments as applied to Rubidium-Strontium data. Reviews of Geophysics and Space Physics, 10, 551-577, 1972.

Dickin, A.P.: Radiogenic isotope geology. Cambridge University Press, 492pp, 2005.

Fuller, W.A.: Measurement error models. John Wiley and Sons, 440pp: 1987.

Hampel, F.R., Rousseeuw, P.J., Ronchetti, E.M., and Stahel, W.A.: Robust statistics. Wiley and Sons, New York, 502pp, 1986.

Huber, P.J.: Robust Statistics, John Wiley and Sons, Inc., New York: 305pp, 1981.

Huber, P.J., and Ronchetti, E.M.: Robust Statistics, John Wiley and Sons, Inc., New York: 354pp, 2009.

Ludwig, K. R.: Isoplot/Ex Version 3.75: A Geochronological Toolkit for Microsoft Excel. Special Publication 4, Berkeley Geochronology Center, 75pp, 2012.

Maronna, R.A., Martin, D., and Yohai, V.J.: Robust statistics. John Wiley and Sons, Chichester. 403pp, 2006; 430pp, 2019.

McLean, N.M.: Straight line regression through data with correlated uncertainties in two or more dimensions. Geochimica et Cosmochimica Acta, 124, 237–249, 2014.

Powell, R., Woodhead, J., and Hergt, J.: Improving isochron calculations with robust statistics and the bootstrap. Chemical Geology 185, 191–204, 2002.

Reiners, P.W., Carlson, R.W., Renne, P.R., Cooper, K.M., Granger, D.E., McLean, N.M., Schoene, B.: Geochronology and Thermochronology. John Wiley & Sons, 464pp, 2018.

Sadovski, A.N.: Algorithm AS74: $L1$-norm fit of a straight line. Journal of the Royal Statistical Society, 23, 244-248, 1974.

Siegel, A.F.: Robust regression Using repeated medians. Biometrika, 69, 242–244, 1982.

Titterington, D.M., and Halliday, A.N.: On the fitting of parallel isochrons and the method of least squares. Chemical Geology, 26, 183–195, 1979.

Vermeesch, P.: IsoplotR: a free and open toolbox for geochronology. Geoscience Frontiers, 9, 1479–1493, 2018.

Wand, M.P. and Jones, M.C.: Kernel smoothing. Chapman and Hall, London. 212pp, 1995.

Wendt, I., and Carl, C.: The statistical distribution of the mean squared weighted deviation. Chemical Geology (Isotope Geosciences Section) 86, 275–285, 1991.

Woodhead, J., Hand, S.J., Archer, M., Graham, I., Sniderman, K., Arena, D.A., Black, K.H., Godhelp, H., and Price, E.: Developing a radiometrically-dated chronologic sequence for Neogene biotic change in Australia, from the Riversleigh World Heritage Area of Queensland. Gondwana Research 29, 153–167, 2016.

Woodhead, J., Hellstrom, J., Pickering, R., Drysdale, R., Paul, B., and Bajo, P.: U and Pb variability in older speleothems and strategies for their chronology. Quaternary Geochronology 14, 105–113, 2012.

Woodhead, J. and Petrus, J., Exploring the advantages and limitations of in situ U-Pb carbonate geochronology using speleothems, Geochronology 1, 67–84, https://doi.org/10.5194/gchron-1-69-2019.

York, D.: Least squares fitting of a straight line. Canadian Journal of Physics, 44, 1079–1086: 1966

York, D.: Least squares fitting of a straight line with correlated errors. Earth and Planetary Science Letters 5, 320–324, 1969.

York, D., Evensen, N.M., Martinez, M.L., and Delgado, J.D.: Unified equations for the slope, intercept, and standard errors of the best straight line. American Journal of Physics 72, 367–375, 2004.