**Version 1.0**

This is the original version, used in Oakley and Fisher (2015). Since publication, some errors have been found and corrected, mostly concerning the propagation of uncertainties. None substantially changes the results of that paper, but it is recommended to use the latest version of the program (currently v. 1.2) in all future work.

**Version 1.1**

Things fixed:

1. Probability was calculated as exp(p) = -sum(R2/(2σ2)), where R is the error in the distance from a restored point to the expected line. This ignores the coefficient in the normal distribution probability density function, since that will typically cancel out when comparing two probabilities in a Monte Carlo simulation. However, due to the propagation of uncertainties through restoration, this factor cannot be cancelled out. Thus, this was changed to exp(p) = -(sum(R2/(2σ2))+log(σ)). The factor of 1/sqrt(2π) continues to be canceled out.
2. In calculating the uncertainty in the distance from a point to a line, σx and σ­­­y were improperly used in place of σx2 and σ­­­y2. This was fixed.
3. For the fault with a bend in it model, in the case that the initial tip position is forced to be at the bend, the bend position from the previous model was incorrectly used instead of that from the current model. This was fixed.

**Version 1.2**

Things fixed:

1. In transforming the uncertainty in the position of a point from the trishear coordinate system to the cross section coordinate system, an incorrect rotation equation was used. This was fixed in the course of adding code that considers the covariance matrix for uncertainty in (x,y).

Things added:

1. When propagating uncertainty in (x,y) of a point through trishear deformation, the covariance in the uncertainty of the two coordinates is now considered.
2. Growth strata can now be used to help fit a model, with the slip necessary to restore each growth bed being an additional parameter.
3. Marine terraces can now be used as well. The slip necessary to restore each terrace is an additional parameter.
4. A model for fault parallel flow on a propagating, circular listric fault has been added.
5. A model for parallel fault propagation folding has been added.
6. Restored bed elevations and dips can now be fit for as additional model parameters.

**Version 1.2.1**

This version fixes a few bugs in the things that were newly added in version 1.2.

Things fixed:

1. I fixed an error in the parallel fault propagation fold model, in which data were not properly reflected during a transformation of coordinate systems.
2. I fixed an issue in which the extra parameters for terraces and growth strata were added before the number of parameters was calculated.
3. I fixed an error where only 1 was added to the number of parameters when fitting for bed elevations, regardless of the number of beds.
4. I fixed an error in which one too many parameter values were being read for growth strata, terrace beds, or restored bed elevations.
5. I fixed a case where if fitting for the pre-folding dip of beds, the dip was converted to radians where it shouldn’t have been. I also fixed an error in which the wrong value was sometimes read in for this dip, and a case in which (if fitting for each bed separately), did wasn’t converted to slope where it should have been.
6. I fixed an error in which covariance terms for uncertainty in marine terrace points were being improperly squared. I also fixed an error in which off-diagonal terms of the covariance matrix weren’t properly set to 0 when they should have been.
7. I fixed an error in which the uncertainty in bed points was used instead of that in terrace points in the fault with a bend in it model.
8. I fixed an error in which the program would sometimes crash on reading the parameters file.

**Version 1.2.2**

This version implements two quick bug fixes.

Things fixed:

1. I fixed a bug in the fault with a bend in it model that prevented points in the hanging wall from moving into the trishear zone if the fault tip is below the bend.
2. The previous code for calculating RMS error implicitly assumed that beds all have the same number of points in them. I changed it so that this is no longer the case and beds can be any number of points. The RMS is the RMS error of all points, irrespective of the number in each bed.

**Version 2**

This version is a major revision that implements a number of new features as well as fixing some bugs.

Things fixed:

1. I fixed an error in which the slopes of marine terraces were not read in properly when terraces are used as a data type.
2. For the APT algorithm, if the initial model is specified, I added a prompt to ask for the number of model parameters. This overcomes a problem that can occur with the program not knowing the correct total number of parameters at the time the initial model is asked for. Note that this problem can still occur in other Markov chain Monte Carlo methods, where it has not yet been fixed.
3. I fixed a bug in the treatment of errors for marine terrace points, which occurred if errors were not propagated through the cross section restoration and caused the entire array of uncertainties for all terraces to be used where just the uncertainty for a single terrace should be used.
4. In the parallel fault-propagation fold model, I added R0 (ratio of slip over the two fault segments) terms in calculating fault bend and fault tip position, which were previously missing.
5. In the parallel fault-propagation fold model, I increased the precision to which the angles γ, γ1, and γ\*are calculated, due to errors caused by insufficiently precise γ values.
6. In the parallel fault-propagation fold model, I fixed multiple errors that incorrectly prevented points from moving between the crest, forelimb, backlimb, and flat regions or caused movement into the wrong domain in some cases. I also instituted checks to prevent points from oscillating back and forth between two domains due to rounding errors.
7. In the fault with a bend model, I fixed an error in which the covariance matrix for uncertainties in the position of a point was not properly rotated when the fault tip moved from one segment into the other.
8. I fixed an error for growth strata in the detachment and listric fault models, in which the counter that keeps track of which growth bed is being restored was not properly initialized to 1 at the start of each run.

Things added:

1. A multi-bend trishear algorithm was added to allow faults consisting of an arbitrary number of segments and fault bends to be modeled. This also allows P/S and phi to change during fold growth if desired. It includes options for backlimb deformation by fault-parallel flow, fault-bend folding, and inclined simple shear.
2. I added an option to test that beds have been restored in the correct stratigraphic order.
3. For the s = 1 case, displacements in the trishear zone are now calculated by a semi-analytic method that is much faster than the original incremental method. This applies to all trishear fault models except the circular listric fault.
4. For the Adaptive Parallel Tempering (APT) algorithm, I added the option to save models only at some specified interval rather than save every model.
5. I added an option to restored beds to a multi-segment restored geometry rather than a straight line.
6. I added the option for different x and y uncertainties in bed data points.
7. I added an option to fit groups of beds all to the same restored-state dip, with multiple groups able to be defined.
8. I added an option to have a different uncertainty for each bed.

**Version 3**

This version is a major revision that implements a number of new features as well as fixing some bugs.

Things fixed:

1. For a multibend fault, when allowing the trishear zone to intersect the backlimb syncline axes above some elevation, I added code to deal with the specific case of vertical shear / a vertical fold axis, which would otherwise not work properly. I also fixed the code to allow cases in which the intersection point is below the bend, meaning that in fact the trishear zone dips harmlessly the other way, and it won't intersect within the region of interest.
2. For fault points as a data type, the program now checks whether the point is above the fault tip and uses the distance to the fault tip if that is greater than the error would be otherwise.
3. I fixed a bug in the multi-bend fault where a check for whether a point is above the lowest ramp segment didn’t work properly in the case that there is only one ramp segment.
4. I fixed a problem in trishear\_multibend where the question of whether to require the final fault tip to be in the upper segment would never exit during manual entry of options.
5. I changed the prompt asking for a file name to refer to "results" instead of "errors" as the quantity to be saved.
6. I modified the selection of a random initial model in the APT method to check if the model is an allowed set of parameters (in terms of tip position, stratigraphic order, etc. and to try a new set of parameters if it is not.
7. I fixed a problem that, when fitting for uncertainties and using dip data, caused the next parameter after the dip uncertainty to be used as the restored dip uncertainty even if there was not supposed to be a separate restored state dip uncertainty.
8. In the Setup program, I changed it to write the word "sigma\_xy\_diff" if the x and y uncertainties in bed data are different, since this is expected by the InvertTrishear program.
9. In the adaptive MCMC algorithms (AM, RAM, and APT), I changed the initiation of the variances in the covariance matrix from steps(i) to steps(i)\*\*2. This means that steps is used as the standard deviation of the initial jumping distance for that parameter, as intended, rather than the variance.
10. I fixed some errors in the analytic trishear function check for rounding errors when u0 is nearly equal to m, which was not always working properly.
11. In the data\_uncertainties module I changed all the subroutines to set the uncertainties to 0 if I am not using an uncertainty result type. This prevents a crash if trying to get RMS Error or Chi-square statistic.
12. In the trishear\_func\_dip\_multi\_bend function, for points in the trishear zone, I changed the line ptze(:,1) = (/x+PoverS(n)\*slip,y/) to ptze(:,1) = (/x+R(n)\*PoverS(n)\*slip,y/) (as it is in the non-dips version of the function), since the earlier version was in error.
13. If the trishear zone boundary is allowed to intersect the fault axis above some elevation, I modified the code to not check points where the ramp angle doesn't change, since these are P/S or phi change points, and there is no real fault axis there.
14. In the 'Objective Function for beds and/or dips:' option, I changed the somewhat unclear option '(3) Fit for dip.' to '(3) Fit to best fit line / mean dip.'
15. In trishear\_multi\_bend, in the options for fault tip staying above a specific bend or starting at a specific bend I made the maximum allowed number for the bend be nangles-1 rather than nbends, since this shouldn't include any "bends" introduced by extra P/S or phi values, which get counted in nbends.
16. For fault point data, I added the check for whether points are above the fault tip that was previously in trishear\_multibend to all the fault types.
17. In the err\_and\_prob module bed\_prob\_cov function, I changed the calculation of the determinant from multiplying terms followed by taking the log of its sqrt with a direct calculation of the log of the sqrt of the determinant by adding terms. This helps avoid the determinant going to infinity for large numbers of points.
18. In trishear\_straight, in calls to trishear\_func and trishear\_func\_dip, I changed PoverS to -slip\_sense\*PoverS to allow it to be used for normal faults. This won't change anything when restoring reverse faults (slip\_sense<0 for restoration), but it will change the sign of PoverS used in the trishear functions for restoring normal faults (slip\_sense>0 for restoration).
19. I made calls to the relevant trishear functions use -slip\_sense\*PoverS instead of PoverS in trishear\_bend, trishear\_decol, trishear\_listric, and trishear\_multi\_bend. I did not make the change for Parallel\_FPF, because that is exclusively for reverse faults.
20. In trishear straight, in trishear\_func and trishear\_func\_dip, when determining if points in the footwall can enter the trishear zone, I changed if (slip\_sign == 1) to if (slip\_sign == prop\_sign) where prop\_sign is the sign of P/S. This allows it to work properly with normal faults (which have negative P/S). I also made similar changes in trishear\_func\_bend, trishear\_decol, and trishear\_listric. In trishear\_multibend, due to the different way things are done there, it doesn't look like there's any equivalent change I need to make, so I haven't done so. I also did not make the change for Parallel\_FPF, because that is exclusively for reverse faults.

Things added:

1. I added the option to fit a circular or elliptical listric fault geometry, which is approximated by a specified number of straight fault segments.
2. I added the opportunity to place restrictions on the tip position (inital or final) that is not being fit for as a model parameter. Currently this only works when using the straight and multi-bend fault models.
3. I added an option in for multi-bend faults to require that the final fault tip position must be in the uppermost segment. This also applies to and is useful for the elliptical and circular fault models, in order to require that the fault tip be above the point where the maximum angle of the listric fault is reached.
4. I added an option to constrain the non-parameter fault tip position to include the option to require it to be right / left of or above / below a user-specified line.
5. I added propagation of uncertainty for points moving through fault bends.
6. I added the option to parameterize slip to restore growth strata in two different ways: Either as total slip or as slip since the previous layer was restored.
7. I added the possibility for synthetic inclined shear (so long as the axis dip is not less than the ramp angle) f or multi-bend faults.
8. I added an option to allow groups of beds with different uncertainties to be specified, rather than just all beds having the same uncertainty or all having different uncertainties.
9. I added an option to fit for correlated uncertainty correlation length for beds as a model parameter.
10. I added an option to constrain the maximum and minimum allowed x coordinates for any fault points for multi-bend faults.
11. I added an option to not require the fault to be concave upward for multi-bend faults.
12. I added options on how to check whether beds restored in the right order based on age. Instead of just checking their order at x=0, there are also options to prevent the beds from intersecting within the x domain of the data or to prevent them from intersecting within some specific x domain.
13. I added the option to use a spline fault geometry.
14. I added an option to put constraints on the x position of the backlimb syncline.
15. I added an option when fitting to best-fit restored bed slope (option 3 for FitType) to put minimum and maximum limits on the dip of the restored state beds.
16. I added the capability to have separate correlated and uncorrelated uncertainties for a bed. Note that the correlated errors are not propagated during restoration.
17. For circular / elliptic faults I added an option to have the fault tip start at the detachment (only available if fitting for initial fault tip position). Relatedly, in trishear\_multi\_bend, I made the option to have tipy be at a specified bend only available for pure multi-bend faults and not for elliptic or spline faults.