Appendix B - Discussion of Ellipse Fit Algorithm

Ellipse fitting in Cartesian coordinates with an arbitrary tilt is notoriously unstable and prone to rapidly approaching zero or infinity, as discussed in Section 2.2.2 of the text. In NLLS fits, the partial derivative of each parameter in the fit function is used. The equation of an ellipse in Cartesian coordinates with an arbitrary tilt (Eq. 8) is much more complex than a circle:

$$y = \frac{-2b^2\left(x_0 - x\cos(\theta)\right)\sin(\theta) + 2a^2\cos(\theta)\left(y_0 - x\sin(\theta)\right) \pm \sqrt{2a^2b^2\begin{pmatrix} a^2 + b^2 - x_0^2 - y_0^2 - 2x^2 + \left(a^2 - b^2 - x_0^2 + y_0^2\right)\cos(2\theta) \\ +4y_0\sin(\theta) + 4x_0\cos(\theta)\left(x - y_0\sin(\theta)\right) \end{pmatrix}}}{2\left(a^2\cos^2(\theta) + b^2\sin^2(\theta)\right)} \tag{8}$$

where a is the semi-major axis, b is the semi-minor axis, x_0 is the center longitude and y_0 the center latitude, and θ is the tilt of the ellipse's major axis. This fit is unstable because of the partial derivatives (Eq. 9):

$$\begin{aligned}
\frac{\partial x_0}{\partial y} &= \frac{-2\left(-x_0 + x\cos(\theta) - y\sin(\theta)\right)}{a^2} \\
\frac{\partial y_0}{\partial y} &= \frac{-2\left(-y_0 + y\cos(\theta) + x\sin(\theta)\right)}{b^2} \\
\frac{\partial a}{\partial y} &= \frac{-2\left(-x_0 + x\cos(\theta) - y\sin(\theta)\right)^2}{a^3} \\
\frac{\partial b}{\partial y} &= \frac{-2\left(-y_0 + y\cos(\theta) + x\sin(\theta)\right)^2}{b^3} \\
\frac{\partial \theta}{\partial y} &= \frac{2\left(-y_0 + y\cos(\theta) + x\sin(\theta)\right)\left(x\cos(\theta) - y\sin(\theta)\right)}{b^2} \\
&\quad + \frac{2\left(-x_0 + x\cos(\theta) - y\sin(\theta)\right)\left(-x\sin(\theta) - y\cos(\theta)\right)}{a^2}
\end{aligned} \tag{9}$$

The derivatives are sensitive to the inverse-square and -cube of the semi-major and -minor axes. If a or b become small, the fit will rapidly approach infinity; if they become large, the fit will approach 0. Consequently, multiple safeguards were in place to checks of each parameter were performed during every iteration of the fit in order to guard against bad results; these are illustrated in Figure **60**.

An intelligent guessing of the major and minor axes was employed at the start of the fit by taking the average of the maximum minus the minimum of the crater's longitudinal (x) and latitudinal (y) extents. These were multiplied by a random number drawn from a Gaussian distribution with $\mu = 1.0$ and $\sigma = 0.25$ and saved as the initial guesses for the major and minor axes. The tilt angle initial guess was drawn from a uniform distribution between $\pm\pi$. All five fit parameters were monitored throughout each iteration of the fit to increase the chances of the algorithm locking onto a realistic fit: The center latitude and longitude (from the center of mass of the points) were checked so if they became unreasonably large (1000 was the threshold) they would be re-set to 0.01. The major and minor axes were also monitored so if they became unreasonably large, they would be re-set to the diameter that the circle-fit found, multiplied by another random number as described above. The tilt angle was monitored so that if $|\theta| > \pi$, it was re-set to 0.001.

In addition to these safeguards, the ellipse fit was re-done for *every* crater a *minimum* of three times and a maximum of five. The algorithm required the NLLS fit to converge on the *same* fit values for all five parameters three times to a tolerance of 10^{-6}. If this did not happen the first three times the crater was fit, the algorithm would attempt to fit it up to twice more. If by the fifth fit, there were not three identical matches, then the algorithm would record that the fit did not converge for that crater, leaving the values blank. Finally, the algorithm required that the major and minor axes be within 35% of the circle-fit's diameter, otherwise it was rejected. This was chosen to, by definition, exclude fits with an ellipticity $\gtrsim$ 2.0 (of which there are less than a dozen on Mars that are $\varepsilon > 2.0$).

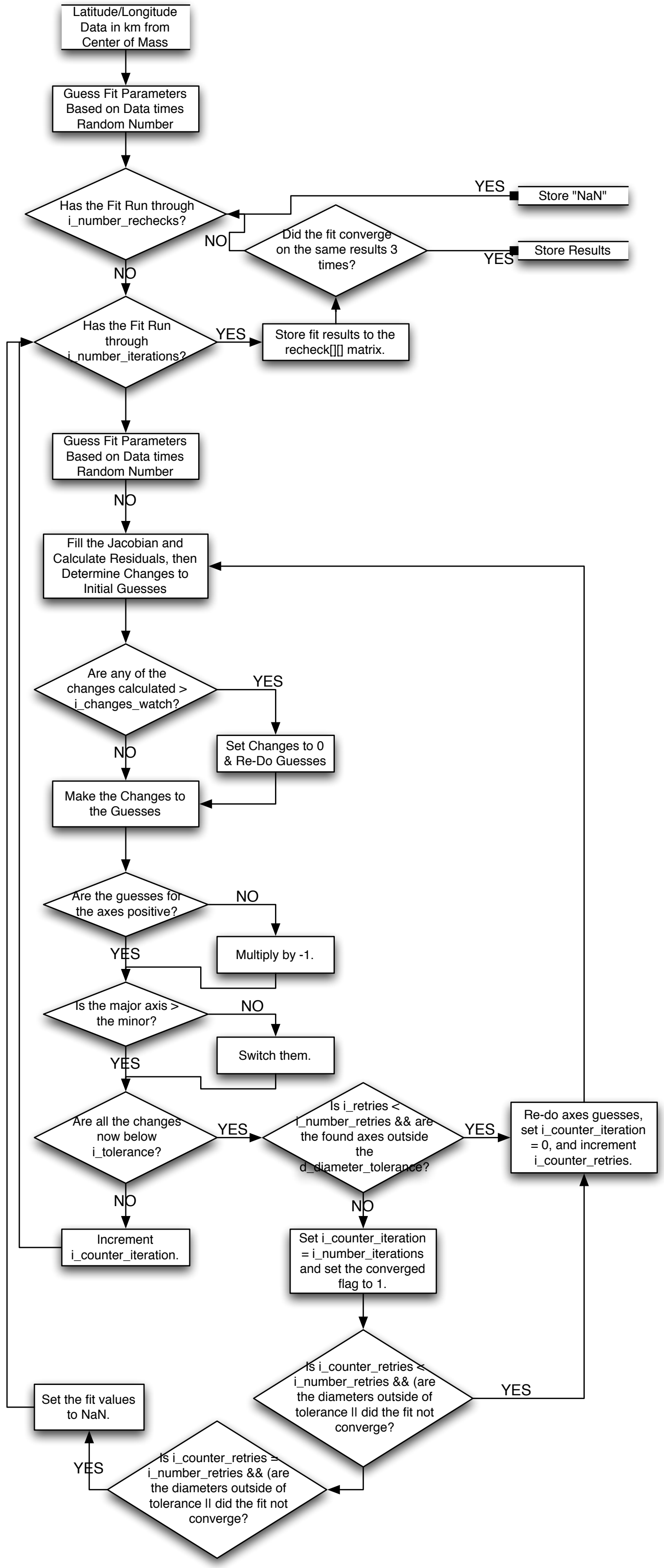

Figure 60: Flow chart illustrating the algorithm to monitor the ellipse fit and verify convergence.