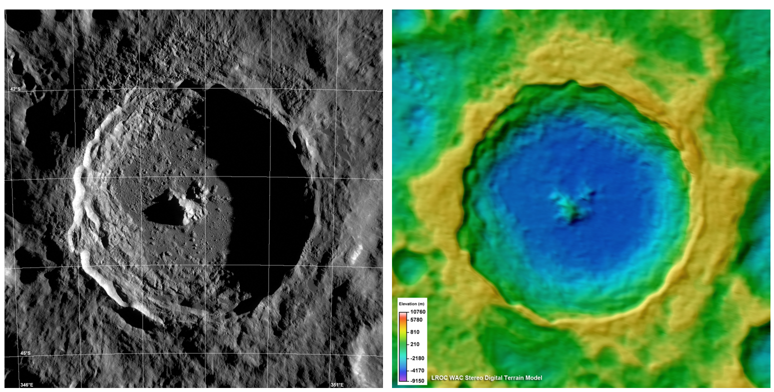
M3/4/5N9 – Project 1
Due 3 Nov 2017

Reminder: High level MATLAB functions, such as 'backslash' are not permitted to be used unless explicitly instructed to do so.

Representing lunar elevation data using moving least-squares

In computer graphics and medical imaging, one may wish to generate a continuous surface from position data defined on a set of points so that the surface to be accurately rendered and visualised. One approach for doing this is a technique known as *moving least-squares* (MLS) (see references provided with the project). In this project, we will use MLS to reconstruct the height of the Moon's surface along a line through the centre of its crater Tycho (left image). This data can be found in the file `tycho.mat` which provides the height, h_i, at the regularly spaced points, x_i. The data captures the steep drop in elevation along the sides of the crater, as well as the rise in elevation due to Tycho's characteristic central peak. The project uses topographical data (right image) collected by NASA's Lunar Reconnaissance Orbiter (LRO). For other images and data from LRO, see `https://www.youtube.com/watch?v=2iSZMv64wuU`.



(a) (b)

Figure 1: (a) An image from LRO of the lunar crater Tycho. (b) Elevation data for Tycho.

Given scatter data h_i at points x_i for $i = 1, \ldots, N$, MLS allows for the construction of a continuous function from the data through

$$h(x) = \min_{p \in \mathcal{P}_n} \sum_{i=1}^{N} \theta(|x - x_i|)|p(x_i) - h_i|^2, \tag{1}$$

where $x \in \mathbb{R}$, the minimisation is over polynomials of degree n or less, and θ is the weighting function that decays as $|\mathbf{x} - \mathbf{x}_i| \to \infty$ and $\theta(r) > 0$ for all r. A suitable choice for $\theta(r)$ is

$$\theta(r) = e^{-r^2/\eta} \tag{2}$$

where η is a parameter. The polynomial degree n and parameter η are set by the user and affect the resulting function.

1. Before going to MLS, generate polynomial fits of degrees $n = 5, 10$, and 20 of the elevation data using the standard least squares algorithm. You may use the function `house.m` as part of your implementation, but everything else you must write yourself.

   In your report, include plots the resulting functions, as well as the original data. Also, provide in a table the six monomial coefficients of lowest degree given by least squares. Provide another table containing $\|h(x_i) - h_i\|_2$ for each case. Discuss how well the fits reproduce features in the original data and how $\|h(x_i) - h_i\|_2$ changes with n. Does the fit improve as n increases?

2. In your report, provide a detailed version of the MLS algorithm. In particular, discuss and explain how the standard polynomial fitting least squares algorithm can be adjusted to perform MLS.

3. Modify your Matlab codes to implement MLS:

   (a) For some marks, implement MLS with the weight function $\theta(r)$ as described above.

   (b) For full marks, accelerate your MLS by using the modified weight function

   $$\tilde{\theta}(r) = \begin{cases} \theta(r), & \theta(r) \geq \epsilon \\ 0, & \theta(r) < \epsilon. \end{cases} \tag{3}$$

   Further, explain why this reduces the number of operations and describe the algorithmic changes you have made to realise this in your implementation.

4. Using your implementation, find $h(x_i)$ for

   (a) $\eta = 10^{-2}, 10^{-3}$, and 10^{-4} with $n = 3$

   (b) $n = 1, 3$, and 10 with $\eta = 10^{-3}$.

   For each case, plot $h(x_i)$ vs. x_i and provide in a table the values of $\|h(x_i) - h_i\|_2$. Additionally, if you are using the modified weight function, set $\epsilon = 10^{-6}$. Discuss how MLS performs as n and η are varied. Compare the performance of MLS with that of standard polynomial fitting, both in terms of fit quality and the number of operations needed to perform the computations. Discuss also how $\|h(x_i) - h_i\|_2$ changes with n and η, as well as with standard polynomial fitting.

5. Finally, the cost of MLS can be reduced by using fewer data points to find $h(x)$. Fixing $\eta = 10^{-3}$ and $n = 3$ and taking a subset of (x_i, h_i), systematically vary the number of data points used to generate the fits and investigate how this affects MLS fit quality. Be sure to reproduce the function at **all** x_i. You may wish to do this is by taking equispaced points, or by randomly selecting the subset. Discuss how the quality of the fit varies with both the number and distribution of data points and provide some insight into how the number of points required is influenced by η.