ISIS Photoclinometry User Guide

Randolph L. Kirk

Version 0.1
7 March 2003

Purpose and Scope

This document is intended to provide users of the ISIS two-dimensional photoclinometry software (in particular, program pc2d) with essential information for preparing their data, using the programs, and obtaining reliable results. It contains a description of the end-to-end workflow with hints for solving common problems and using the software effectively in the majority of cases but does not attempt to describe every feature of the programs or document the underlying algorithms in any detail. Future updates of this document are planned and will include information about new tools and features, additional details about existing capabilities, additional references, and figures.

Additional Resources

- The web page http://astrogeology.usgs.gov/Teams/Geomatics/pc.html indicates the most recent updates to `pc2d` and other photometry/photoclinometry programs. Also found on this page are links to instructions for obtaining ISIS and downloading updated programs, and an installation guide that tells how to set up the environment variables needed by `pc2d`.
- The TAE help (*.pdf file) for `pc2d` contains a short description of how the program works, plus descriptions of each of the parameters that may be set at runtime.
- TAE help for `pc2d` and other photometry/photoclinometry programs contain definitions of various photometric function models used, and values of photometric parameters for Mars. A useful compilation of Hapke photometric parameters from the literature for other solar-system bodies is given by McEwen (1991) Photometric Functions for Photoclinometry and Other Applications, *Icarus*, 92, pp. 298–311.
- The full set of current and planned ISIS programs for photoclinometry and the typical workflow for using them are discussed by Kirk et al. (2003) Photoclinometry Made Simple…?, ISPRS Workshop "Advances in Planetary Mapping 2003," Houston, March 2003, online at on the web at http://wwwflag.wr.usgs.gov/USGSFlag/Space/Isprs/MEETINGS/Houston2003/abstracts/Kirk_isprs_mar03.pdf.
- A paper describing the algorithm used by the ISIS two-dimensional photoclinometry programs and accuracy analyses is in preparation. A preprint will be posted when available. The current reference for the algorithm is Kirk, R. L. (1987) "III. A Fast Finite-Element Algorithm for Two-dimensional Photoclinometry," Ph.D. Thesis unpubl.), Caltech, pp. 165-258.
- Questions, comments, and suggestions concerning the software can be addressed to Randy Kirk (rkirk@usgs.gov). Problems with obtaining or installing the software should be addressed to ISIS support (isis-support@flagmail.wr.usgs.gov)

Background

Photoclinometry (more descriptively shape-from-shading) is a cluster of related techniques for estimating topography that have been exploited by planetary scientists for more than 50 years. These methods can be classified by the dimensionality of the region over which height information is sought. One-dimensional methods determine slopes from image brightness (radiance) along a line or curve and integrate these into an elevation profile. In contrast, two-dimensional photoclinometry (2D PC) methods like the one implemented in ISIS work with an area of an image and build up a digital elevation model (DEM), i.e., a full three-dimensional model of the surface. 2D PC is naturally of greater interest for mapping applications than approaches that produce only isolated topographic profiles, but the increased usefulness comes at a cost of increased software complexity and increased effort by both computer and human.

A description of our approach to 2D PC, given "while standing on one leg" is as follows: An image is input. The program estimates the shape of the topographic surface that would come closest (in a least squares sense) to producing such an image under the viewing geometry described in the image label. The estimated topographic model can then be output as an ISIS cube. The process by which the solution is obtained is iterative, and in general doubly iterative. The nonlinear equations for the surface that best fits

the image are repeatedly linearized and solved for increments to the topography that improve the fit, a process known as "Newton-Raphson iteration". Several means are available to solve the matrix equation generated at each Newton step:

(a) Direct solution by matrix factorization. This is not iterative but its memory and CPU requirements grow very rapidly with image size so it is only practical for small images. The program `pcinfo` can be run to determine memory needs of each method for a given image.

(b) Incomplete Cholesky/Conjugate Gradient method (ICCG). This is an iterative method that is memory efficient so it can be used with large images. It converges rapidly when it works but unfortunately tends to diverge catastrophically for photoclinometry on all but the blandest images.

(c) Successive over-relaxation (SOR). This is another memory-efficient, iterative method. SOR is the method of choice for most images. Its main weakness is that long spatial wavelength components of the solution (horizontally "big" features) converge very slowly. Convergence can be greatly speeded up by "multigridding" (working on approximations of the same problem at different resolutions in order to arrive at the different spatial frequency components of the solution simultaneously). The multigrid algorithm for photoclinometry unfortunately requires human supervision to be effective.

The program `pcsi` will iterate at full resolution with any of the above methods until the solution has converged to a specified level of accuracy or for a maximum number of iterations or CPU minutes, whichever comes first. It can thus be run in batch mode but is only useful if the image is small enough that direct solution can be used or if another source such as stereo provides an initial approximation to the topography that is accurate for long spatial wavelengths. In the latter situation, photoclinometry serves as a kind of "smart interpolation" to add pixel-scale details to a lower resolution DEM.

The main challenges in using `pcsi` are finding and preparing a suitable image for processing, and determining appropriate parameters for the photometric model of the surface that will be used in solving for the topography.

For images that are too big for direct solution and don't have a good initial approximation to the DEM, the program `pc2d` allows the user to control the iteration process interactively, including evaluating how the solution is progressing, setting the parameters that control SOR, and changing between full and reduced resolutions in the multigrid algorithm when appropriate. Thus, `pc2d` presents the same challenges as `pcsi`, plus the larger challenges of understanding its user interface and learning to control the calculation to achieve a solution in the least amount of time. This document is intended to start users in the process of mastering these challenges.

Image Selection and Preparation

In order to solve for topography from a single image, the photoclinometry software must make the simplifying assumption that the photometric properties of the surface are uniform. This is, of course, never strictly true in the real world, but for useful results it is necessary to select images in which most of the contrast results from topographic shading rather than from variations in intrinsic reflectivity (albedo). In general, this means that images with relatively large incidence angles (certainly •30°) are preferred. Images with very large incidence angles should also be avoided because of the prevalence of shadows, which not only hide parts of the surface but can also make the photoclinometry iteration process unstable. On Mars, the atmosphere also causes the topographic contrast of images to decrease once a certain incidence angle is exceeded for a given amount of atmospheric dust. In general, the optimal incidence angle depends on the magnitude of albedo variations , the ruggedness of topography, and atmospheric conditions. The human eye is the best guide to selecting images that appear to contain mostly shading and little albedo variation. The photoclinometry process is very sensitive to albedo variations and will produce artifacts even for surfaces that look uniform to the eye. These artifacts can be removed by digital filtering (discussed below) if they are not too severe, but images that show albedo variations equal to or greater than topographic contrast should be avoided.

Images to be used for photoclinometry must be radiometrically calibrated (ISIS "Level 1" processing). Such processing is necessary to remove artifacts in the image due to pixel-to-pixel offset and gain variations, which would be misinterpreted as topographic shading in the photoclinometry calculation. Once

the image is calibrated, it is not necessary that it be kept in 32-bit floating-point format, as long as the conversion is done so as to preserve image contrast. Images that use most of the dynamic range of the 8-bit format will produce good results in most cases.

Programs `pc2d` and `pcsi` will work with images with different geometric properties and labels:

- Images from framing cameras defined in the original ISIS system in raw camera geometry, with the original image label format. For cameras with nonsquare pixels (e.g., NEAR MSI), the images must be resampled to square pixels before use.
- Images from framing cameras defined in the newer "plug-in" or "levels" ISIS software with the additional label information created by running the program `levinit`. Several important cameras (Viking Orbiter, Voyager, Galileo) are available in this format and others currently defined only in the earlier system may eventually be added. Images with nonsquare pixels must be resampled before use.
- Images from pusbhroom scanners defined in ISIS (e.g., MGS MOC). Scanners are handled only by the "plug-in" software and the images must be processed with `levinit` to have the appropriate labels. Images with nonsquare pixels are handled by the clinometry programs and do not need to be resampled before use (in fact, aspect-ratio corrected images cannot be used, because there is no record in the labels that they have been resampled).
- Images and image mosaics in map projection (ISIS "Level 2"), subject to certain limitations.

The label of an image projected with the plug-in program `lev1tolev2` will contain information about the illumination and viewing geometry of the original image that is needed in the photoclinometry calculation. When multiple such images are mosaicked together, this information is erased because (in the general case) different parts of the mosaic will have different illumination and viewing geometries. The photoclinometry programs therefore will not work with such mosaics. The program `lev1prop` has been provided in order to get around this limitation. It can be used to propagate or copy the illumination and viewing geometry information from a single plug-in system Level 1 image to a Level 2 mosaic, after which the mosaic can be used for photoclinometry. A mosaic made in the original ISIS system (e.g., an old Viking mosaic made before the ;lug-in system was introduced) can also have the required information added with `lev1prop`, provided one image can be initialized in the plug-in system with `levinit` so it can supply the information. This means that mosaics made solely from images not defined in the plug-in system cannot currently be used.

Using the information from one image to represent the entire mosaic is a good approximation for mosaics made from sequences of sequentially acquired images that have very similar subsolar and subspacecraft coordinates. In such cases, the ability to perform photoclinometry on mosaics addresses "Murphy's law of cartography" (that features of interest are always located at the boundaries between images). Note that `lev1prop` cannot and does not do any checking of the images in the mosaic. It is up to the user to ensure that these images have similar illumination and viewing geometry so that photoclinometry will yield sensible results.

The user is also responsible for ensuring that the map projection used is appropriate for photoclinometry. Projections that will give useful results are those that approximate the viewing geometry of an image. For example, a collection of framing camera images can be mosaicked in orthographic projection with a center near the mean subspacecraft point of the sequence. In this geometry, the entire mosaic (up to a full hemisphere) resembles an image formed from the mean spacecraft position and can be processed. For mosaics that cover a small portion of a hemisphere, most other projections can also be used. The software can deal with nonsquare pixels and with the effective geometric viewing direction of the reprojected images not being the same as the emission direction used in the photometric modeling. Projections to be avoided are those that have a significant "skew" relative to the line-sample grid, such as the sinusoidal projection away from its central meridian.

Parameters for pc2d and pcsi

The input parameters for the photoclinometry can be divided into three groups: input and output file information, photometric parameters, and parameters for controlling iteration.

Input and Output File Parameters

Parameter FROM is the name of the input image file and should be self-explanatory; the processing required to prepare an image for photoclinometry has been discussed above. A subarea of the image to be processed can be specified with SFROM, which works the same way as in other ISIS programs. The subregion specified must have the same sampling increment in the line and sample directions and must be a contiguous region.

The filename for the output DEM is TO. This file will have the same dimensions as the input image specified by FROM and SFROM and will contain elevations (in meters) at the centers of the image pixels. An optional second DEM output is given by ZOUT. This file will contain elevations at pixel corners (which is what is used in the internal calculation) and hence will have one more row and column than TO. The TO file is the real product of photoclinometry that would normally be used for subsequent analysis. The ZOUT file is offered so it can be re-input as the ZIN file in a future run of the program. Outputting ZOUT and then restarting with it as ZIN allows iteration to resume exactly where it left off, which can sometimes be useful, e.g, if `pcsi` is run and it is determined that more iterations are desired or that iteration should proceed with different parameter values. If the TO file were used for this purpose, it would be interpolated to pixel centers on output and then interpolated back to pixel corners on input, significantly degrading local details.

The ZIN parameter controls the starting approximation to the DEM that will be used in the calculation. One use, just described, is to restart the calculation with a ZOUT file from a previous run of the program. ZIN can also be the name of a DEM from another source such as stereo or altimetry that has been projected so it coregisters with the image area given by FROM and SFROM. When such reprojection is done with lev2tolev1 (if FROM is a Level 1 image) or lev2tolev2 (if it is a Level 2 map-projected file), ZIN will have the same number of lines and samples as the image area and will be interpolated inside the program to yield elevations at pixel corners.

Two special values for ZIN result in starting DEM values that are not based on input files. If ZIN=DATUM the elevations will be initialized to zero, i.e., the mean or datum surface will be used as the starting approximation to the topography. Convergence of SOR iteration can be slow for this choice, but the other iteration methods may work satisfactorily. If no value is given for ZIN (i.e., it is set to the TAE null value "--") then a first approximation to the topography will be generated by a fast method (called SSIPSF-PI). Use of this fast initial calculation usually works quite well because it supplies a reasonable approximation of the long-wavelength components of topography. If the image illumination is diagonal to the image (i.e.., not close to either the line or sample direction) and especially if the image is narrow in one direction and wide in the other, then the SSIPSF-PI approximation can be inaccurate at the corners of the DEM. This is a fundamental problem that arises because the topography in the corners would be determined from image information outside the image, which is not available. The solution, if possible, is to use a ZIN file with topography from an alternate source such as stereo or altimetry.

The LOGFILE parameter is optional. If the name of a file is provided, the file will be used to contain a record of the iteration statistics. The information provided is the same as that in the upper half of the log window, discussed below.

Finally, the MAXMEM parameter governs the amount of memory made available to the program. The amount of memory that will be required depends on the dimensions of the image, the algorithms used for the initial and subsequent iterations, and the number of levels of the multigrid resolution that will be used. The memory needed for the SSIPSF-PI initial approximation also depends on the illumination geometry and photometric parameters. The program `pcinfo`, which has many of the same inputs as `pc2d` and `pcsi`, will calculate the memory needs for a given image, but a rough rule of thumb is that MAXMEM must be greater than 10 times the number of image pixels for SOR iteration, which uses the least memory. If MAXMEM is set too small for the solution method selected in `pcsi`, the program will terminate. In `pc2d`, the solution method can be changed while running the program, and methods that require more memory than is available will be grayed out. For example, with realistic memory allocations, only SOR may be possible at full resolution, but the ICCG and direct solution methods may become available at reduced resolutions.

Photometric Model Parameters

These parameters can be further subdivided into the type of photometric model used for the surface, the parameters needed for the surface photometric function, the image normalization parameters, and parameters describing the datum (the mean or no-topography surface).

The photoclinometry programs offer several varieties of the physically motivated Hapke photometric function as well as four simple, empirical functions: Lambert, "lunar" (Lommel-Seeliger), lunar-Lambert, and Minnaert. Use of the Hapke functions is not recommended, simply because they are more complex and slower by a factor of as much as several hundred. McEwen (1991) has pointed out that, at any given phase angle, a Hapke model with given parameters can be approximated satisfactorily by either the lunar-Lambert or Minnaert function. The ISIS program `pho_emp_local` can be used to fit one of these simple functions to a Hapke model, and the best-fit parameters output by that program can the be used by `pc2d` or `pcsi`. McEwen also gives a useful compilation from the literature of Hapke photometric parameters for many solar-system bodies. Hapke parameter values for Mars are listed in the TAE documentation for the photoclinometry programs. All of this information will be collected in the user's manual now in preparation.

As a rule of thumb, small changes in the photometric parameters result in small changes in the scale of the inferred topography because they change the relation between slope and image contrast. Larger variations can actually change appearance of a given surface shape, so in photoclinometry they will change the shape of the topography inferred from the image. Software currently in development will allow estimation of the (lunar-Lambert or Minnaert) photometric parameter that gives the best fit between an image and a coregistered, low-resolution DEM.

There are two image normalization parameters, DNATM and DNDATUM, that affect how the image is scaled before being compared to the photometric function model to calculate topography. Values for these parameters must be determined from the image and are crucial to achieving meaningful results. Both are expressed in the same units as the image to be processed, i.e., as radiance factors for a floating-point calibrated image but as data numbers if the image has been converted to a lower bit type after calibration. DNATM represents the uniform additive contribution of atmospheric scattering to the image radiance. This value will be subtracted from all pixels before photoclinometry begins, thus approximately correcting the image for atmospheric haze. DNDATUM represents the pixel value of the datum (level surface) after the atmospheric haze correction has been subtracted.

Obviously, DNATM will be zero for bodies with no significant atmosphere. The value measured in a shadow can be used to estimate DNATM in the presence of an atmosphere, but results may be somewhat inaccurate because shadows vary in brightness, depending on the amount of skylight they receive. Mis-estimation of DNATM affects the overall scale of the output DEM, so if independent topographic data are available, they can be used to constrain DNATM as discussed below.

The sum DNATM+DNDATUM is the image value for a level area. It can be estimated as the mean or mode of the image histogram, or by taking spot measurements of level areas (avoiding areas that appear significantly darker or brighter than average). Mis-estimation of the sum of the two parameters results in an overall tilt of the output DEM toward or away from the sun. Specifically, if DNATM+DNDATM is set too low, the DEM will be tilted toward the sun and vice versa. The overall sunward tilt is reported in the `pc2d` user interface, so it is possible to make trial runs of `pc2d` with different DNATM+DNDATUM values and chose the value that results in a level DEM. This can be done with a subarea of the image to minimize the time spent adjusting the parameters.

If an *a priori* DEM is available, it can be used to constrain DNDATUM and DNATM. This can be done either by calculating an image from the DEM by using the photometric model and comparing it to the real image, or by performing trial photoclinometry and adjusting the parameters to give best agreement of the resulting DEM with the *a priori* version. These methods have been developed and tested, as described by Kirk et al. (2003). ISIS scripts `pc_fit_forward` and `pc_fit_inverse` are being developed to automate the respective processes within ISIS, and will be described in more detail once they are available.

Of the datum parameters, the most important is DATUMTYP, which determines the type of datum model that will be used. DATUMTYP=1 corresponds to a planar datum, while 2 corresponds to a sphere. The planar datum model is the normal choice; a sphere is only needed if the image or mosaic being processed is large enough to contain significant planetary curvature. DATUMTYP=2 is available only for Level 1 framing camera images and Level 2 products in Orthographic projection. Parameter DISTORTD is only relevant to images from vidicon cameras such as those on the Mariner, Viking, and Voyager missions. It determines whether a correction for vidicon distortions needs to be applied to image coordinates before calculating the ground coordinates and from them the datum orientation. The default is NO (do not do the distortion correction), which is strictly accurate if the image has already had distortions removed with `noproj`. The distortion correction has such a small effect that it is not crucial even if the image has not been through `noproj`.

The remaining parameters can normally be left at their default values, in which case the image labels will be used to determine the geometry of the datum. The planar datum will be chosen tangent to the planet at the center of the image, or at X, Y if these parameters are set. The center of the planetary sphere will also be taken from the labels by default but can be set with XSPHERE, YSPHERE if the camera pointing in the labels is inaccurate. The RADIUS of the planet can also be specified. RMASK1 and RMASK2 affect the handling of pixels near the limb when DATUMTYP=2 and should not need to be changed.

Iteration Parameters

Parameters ALPHA, ALPHAASYM, STEPASYM, WMAX, DIVTOL, ITMAX, and ETOL are common to `pc2d` and `pcsi`. TAUFAC, BIGTOL, OLDTOL, and DEPTHLIM are used only in `pc2d`, while CPULIMIT and MAXNR are used only in `pcsi`. Because `pc2d` is an interactive program, most of these parameters can be changed while it is running, as described below. In `pcsi` they keep the values set when the program starts.

The "penalty parameters" ALPHA, ALPHAASYM, and STEPASYM can normally be left at their default values. They govern the addition of a small penalty (proportional to 1/ALPHA) related to roughness of the topography to the least-squares equations being solved. This prevents a "checkerboard" pattern of alternating high and low DEM points (which would not have a visible effect on the image) from developing. ALPHA normally has little influence on the solution, but if the image contrast is high and iteration tends to diverge then lowering the value of this parameter is one potential remedy that can be tried. The other two parameters allow ALPHA to start small and grow to a bigger asymptotic value ALPHAASYM over a number of iteration steps proportional to STEPASYM. These parameters may be useful with the ICCG solution method but are not needed for SOR. As this is an advanced topic, it will not be discussed further here.

WMAX and ITMAX are the main parameters affecting SOR iteration, and adjusting them correctly can be crucial to getting `pc2d` or `pcsi` to converge to a solution. ITMAX is straightforward: it is the maximum number of SOR iterations performed before the equations are re-linearized and a new Newton-Raphson step is started (see the Background discussion above). ITMAX is also used to control the number of ICCG steps per Newton step in the ICCG algorithm. WMAX controls the "weight" used in the SOR iterations. For a discussion of SOR, see Press et al. (1992), *Numerical Recipes*, 2nd ed., Cambridge Univ. Press, Cambridge, pp. 857–860. In simple terms, WMAX governs the "aggressiveness" of the SOR method. Increasing WMAX tends to make the method converge faster, but only up to a point that depends on the individual dataset. Increasing WMAX too much will cause the iteration to diverge instead of converging. Such divergence can be detected in the plots of solution statistics provided as part of the user interface and described below.

We have not been successful in attempts to set WMAX and ITMAX automatically based on the behavior of the iteration statistics, so adjusting these parameters is one of the main duties of the user in running pc2d. For many images the optimal settings change as iteration progresses. The following rules of thumb can be stated, however:

- The defaults WMAX=1.5 and ITMAX=10 are appropriate for images with moderate to low contrast and surfaces with relatively low slopes. If the surface is very smooth it may be possible to increase ITMAX further and perform more SOR steps per Newton step, which saves some of the computation time needed to start each new Newton step.

- If the surface is a little rougher and the image is a little contrastier, these values will lead to decreasing solution residuals at first, but eventually the residuals will start to oscillate instead of declining. Decreasing WMAX toward 1.0 and/or decreasing ITMAX may prevent the oscillation or divergence.
- If the solution continues to oscillate or diverge, it may be useful to reduce WMAX to a value less than 1.0 (this is actually underrelaxation, not overrelaxation) and set ITMAX=1 so the equations are relinearized after every step. This usually causes the solution residuals to decline sharply at first, as the recent divergence is repaired. Unfortunately, the residuals then decline very slowly because underrelaxation is not as efficient as overrelaxation.
- It can be useful to alternate a few steps of overrelaxation with (e.g.) WMAX=1.2 to 1.5 and ITMAX=2 to 5, then a few steps of underrelaxation with WMAX=0.5 to 0.9 and ITMAX=1. The overrelaxation steps can improve the solution in most areas of the DEM but cause it to diverge locally, and the underrelaxation steps will repair that local divergence before it becomes severe. It may be possible to see the areas of local divergence in the displayed DEM. If this strategy is working, the errors will reach a lower level after each overrelaxation/underrelaxation cycle.
- If the solution continues to diverge after WMAX and ITMAX are reduced, try starting the program over and using lower values from the outset; the initial divergence may have been too severe to be repaired. In severe cases, try lowering ALPHA from 10000 to 1000 or even 100.
- Divergence can be triggered by artifacts (e.g., contouring) in an *a priori* DEM used as a starting point for iteration, especially if the incidence angle is large enough to result in shadowing. Try smoothing the DEM before using it as the ZIN file.

DIVTOL governs the detection of SOR steps that may be diverging severely: if any DEM points are changed by more than DIVTOL times the root-mean-square (RMS) change of all points, then the step is declared divergent and a new Newton step is started. If this happens frequently, it may be better to increase DIVTOL from the default value.

The best news about this admittedly delicate iteration process is that, if divergence can be avoided, the final result is not sensitive to the iteration parameters WMAX and ITMAX and the overall path of iteration. These factors mainly affect the amount of effort required to get to the solution.

TAUFAC, BIGTOL, OLDTOL, and DEPTHLIM relate to the multigrid algorithm used to speed up SOR convergence for long-wavelength topography in `pc2d` and should seldom need to be changed. The essence of the multigrid strategy is to switch to a coarser representation of the problem (from the full-resolution to 1/2 resolution, 1/2 to 1/4, etc.) when the rate of improvement with iteration becomes slow, and to switch back to the higher resolution when the residual errors fall to a small fraction of those at the finer resolution or to the level of truncation error inherent in the coarser grid. The parameters named govern the precise definitions of "slow" and "small" in these rules. They do not directly control iteration, however. Instead, they control when "hints" will be displayed, suggesting that the resolution be changed. Users who are annoyed by the hints can try changing these parameters.

CPULIMIT and MAXNR prevent `pcsi` from running indefinitely: the program will be stopped after CPULIMIT minutes of CPU usage, or MAXNR Newton-Raphson steps, whichever comes first.

ETOL is another stopping criterion. The solution will be considered satisfactorily converged when the RMS residual error is less than ETOL. This condition causes `pcsi` to terminate; in `pc2d` it causes a hint to be displayed that the solution is finished, provided that this error level is achieved at full resolution. In practice, the RMS error achievable by SOR iteration varies greatly from image to image so it is hard to set a value of ETOL that can always be reached. Therefore `pcsi` usually terminates because of CPULIMIT or MAXNR and the user of `pc2d` may have to decide when to terminate the program based on the reduction of the error relative to its starting magnitude or simply time and patience. For small images that can be handled with the direct solution method, however, ETOL=1.0E-6 is a reasonable and achievable target based on floating-point roundoff error.

Graphical User Interface

Once started, pc2d displays a graphical user interface (GUI) consisting of four windows: the main or display window, two plot windows for graphical display of statistics related to the progress of iteration, and

a log window in which the statistics are displayed as text. Figures illustrating each part of the GUI will be added in a future version of this document.

Main Window

The main window consists of three functional areas. At the top is the display area, which is similar to the image display window created by `qview` except that two images are displayed side-by-side. To the lower left are a group of "widgets" (buttons and sliders) for controlling iteration. The lower right portion of the window is used for text display of various summary statistics and hints for controlling iteration.

Displays

The image displays function very similarly to two `qview` windows that have been "linked": the pan and zoom tools change the region shown in both windows in a synchronized way, the world button resets both windows to the full dataset, and the pawprint button turns on and off reporting of the cursor coordinates in both images.

When `pc2d` starts, the image is displayed on the left and the starting approximation to the DEM is displayed on the right. The "Open Display…" item in the "File" menu opens a dialog box in which the datasets being displayed can be changed at any time. The following choices are available and can be selected at any time:

- The image (mainly useful for context)
- The elevations in the DEM
- The increment added to the elevations in the latest Newton-Raphson step (useful for locating divergent areas)
- The model image synthesized from the DEM by using the photometric model (emphasizes localized artifacts in the DEM)
- The difference between the actual image and the synthesized model

The image options are displayed with a normalization such that the datum value is 1.0 and shadows have a value of 0.0. Elevations are normalized to the image resolution (sample-direction width of the pixel). These scalings also apply to the statistics reported in the other windows.

Also in the "File" menu are two items "Left View" and "Right View" that bring up the "View" dialog box familiar from qview, allowing the user to change the display stretch of the left and right images respectively. The default stretch is automatic, based on the histogram of each dataset that is displayed. A user-defined stretch can be selected and will be used for successive iterations. This is useful for seeing if the overall magnitude of topography or of the image errors is changing significantly. When the dataset selected is changed, the stretch will be reset to automatic, because the manual stretch set for one datset is unlikely to be appropriate for another.

Similarly, the "Show" menu contains "Left Histogram" and "Right Histogram" items bring up a histogram window like that in `qview` for the left and right dataset respectively.

The "Photoclinometry Options…" item in the "Options" menu brings up a dialog box for changing the parameters ALPHA ("penalty number"), ALPHAASYM ("asymptotic penalty number"), STEPASYM ("time constant (steps) to approach asymptote"), DIVTOL ("(max inc/RMS inc) signaling divergence"), ETOL ("final RMS RHS" where "RHS" means "righthand side" of the equations), BIGTOL ("RMS RHS/ RMS fine RHS for +res"), OLODTOL ("RMS RHS/RMS old RHS for –res"), and TAUFAC. As discussed above, these parameters should not normally need to be changed.

Note that the display windows are interesting and may be helpful in assessing whether problems with iteration are due to local or more extensive divergence, but they are not essential. In the later stages of iteration, it is unlikely that the DEM will change visibly. Updating of the displays can therefore be suspended by clicking the "Display Refresh Off" button located above the text area in the lower right of the window. This will speed up the program, especially if a low bandwidth connection is being used. Display updating can be turned back on at any time, but the displays will not refresh until the next Newton step or change of resolution is completed.

Controls

The widgets for controlling iteration consist of seven buttons, two sliders, and a set of radio buttons. The radio buttons are used to select one of the solution methods (SOR, ICCG, or direct factorization) to be used in subsequent Newton-Raphson steps. Methods that would require more memory than is available will appear dimmed and cannot be selected, but may become active at reduced resolution. The sliders labeled "Relaxation" and "Max SOR Steps" govern the iteration parameters WMAX and ITMAX that were discussed in detail above.

The seven buttons provide a "VCR-like" interface for controlling iteration (but we've spared you the dreaded blinking "12:00" display!). From left to right, they are:

PLAY = Start iterating with the currently selected method and parameters. Iteration will continue until PAUSE is clicked. (Actually, clicking anywhere in the GUI will also pause iteration. This is an X-Windows bug.)

SINGLE FRAME = Perform one Newton-Raphson step with current method and parameters, then pause.

PAUSE = Halt iteration after the next Newton-Raphson step is completed.

DOWN ARROW = Decrease working resolution by a factor of two, in order to speed solution for long spatial wavelengths. Will be dimmed if the minimum working resolution is reached. Note that decreasing the resolution involves a complex calculation, not just averaging together pixels. This calculation usually proceeds quickly but can sometimes be very slow or even fail. In the latter case, return to the higher resolution and use the undo function as described below.

UP ARROW = Increase working resolution by a factor of two, to work on finer details of solution. Will be dimmed when working at the full resolution.

STOP = Exit the program. User will be prompted whether to save the DEM solution to the TO file (normal termination) or discard the solution (in case the program run was unsuccessful for some reason) and also has the option of canceling STOP and continuing the program.

BACK = Limited undo function. Clicking this button will restore the DEM solution to the most recent previous version at this working resolution. The button is therefore dimmed during the first iteration at a given resolution. It is also undone once it has been used and until another Newton step is performed, since there is only one level of undo provided.

The undo function is useful if a given Newton-Raphson step appears to be seriously divergent, as happens sometimes with SOR and more often with the ICCG method. The step can be undone and retried with different parameter settings, or with SOR in place of ICCG. The process of reducing the working resoluton can also diverge on rare occasions. If this happens (the DEM display and statistics will show garbage at the reduced resolution), re-increase the resolution then click BACK to restore the solution at the higher resolution just before the resolution was decreased. Reducing the resolution may be successful after additional iteration at the higher resolution.

Aspects of the overall strategy for iteration are discussed in more detail above. To summarize, the strategy is:

- When SOR iteration becomes "slow" (RMS residual does not decrease as much with each Newton-Raphson step as desired), reduce the working resolution. If iteration slows at this resolution also, reduce further, until the minimum resolution is reached, if necessary. NOTE that reducing the resolution is only needed for SOR, not ICCG or direct solution (should they be possible).
- When the residual at a reduced resolution is a small fraction of the residual at the next higher resolution, or when it approaches the truncation error, increase the working resolution.
- When the residual approaches ETOL or the truncation error at full resolution, the solution has converged and the program can be stopped.

- If iteration oscillates or diverges, try decreasing WMAX and ITMAX with the Relaxation and Max SOR Steps sliders as discussed above.

Text Area
The lower right of the main window is used to display a variety of useful pieces of information:

Azimuth of characteristics: This is the direction, measured clockwise from right, in which slopes have the maximum effect on the image brightness; it is roughly sunward but will not precisely equal the sun azimuth if the photometric function is not Lambert and the emission angle is nonzero. Albedo variations in the image lead to artifacts in the DEM in the form of "streaks" in this direction. This therefore also the angle by which the final DEM should be rotated to align the streaks with the sample axis so they can be suppressed by filtering. This quantity is also recorded in the print.prt file. Program `pcinfo` not only puts the azimuth of characteristics in the print file, it returns the value to TAE for use in processing scripts.

Resolution factor: Indicates the current working resolution of the multigrid algorithm, 1 for full resolution, 2 for half resolution (2x2 pixel averaging), 4 for quarter resolution, etc.

RMS Righthand Side, B error: The root-mean-square solution residual and difference between the real image and model (with the images normalized to 1 for the datum) after the most recent Newton-Raphson step. These quantities are also listed in the log window and plotted in the plot windows, and are shown here to minimize window shuffling.

Dip, az of dip: A least-squares fit of a plane to the most recent DEM solution is performed and the dip and azimuth of dip (clockwise from right) of this plane are reported.

Slope of plane toward sun: The component of slope of the best-fit plane in the direction of the Sun (more precisely, the characteristic direction) is reported. This value can be used to guide adjustment of the DNATM and DNDATUM parameters as discussed above. The goal is to pick parameters that cause the slope toward the Sun to vanish.

RMS residual to fit in m, pixels: The residuals to the best-fitting plane. These are given both in real units (meters) and in pixel units. Pixel units are used for the elevation statistics in the plot and log windows. If the solution is strongly tilted, these parameters will give a better idea of the amplitude of relief than the RMS elevation variation computed without subtracting the fitted plane.

Hints/Errors: When the solution statistics indicate that the resolution should be changed or the solution is finished (according to the strategy outlined above) hints to the user will be displayed here. Error messages will also be displayed, e.g., if the current SOR step appears to be diverging.

Plot Windows
Two plot windows are opened after the first Newton-Raphson step is complete, each displaying a semilog plot of several RMS statistics versus iteration number. The information in these plots is also available in text format in the log window.

Plot window 1 shows the RMS residual or righthand side in blue and the RMS increment to the elevations in yellow, as a function of SOR or ICCG step within the current Newton-Raphson step. As successive Newton steps are performed, the curves for previous steps are drawn in dimmer colors until they fade out after 5 steps. This plot thus allows the user to see how SOR or ICCG iteration is proceeding and also gives a limited idea of recent progress from one Newton step to the next. Note that it is entirely normal for the RMS increment (yellow) curve to rise slightly for the first few SOR steps, then reach a plateau. This is a result of the SOR weight being gradually increased from 1 to WMAX over the first few steps (so-called Chebyshev acceleration) to reduce the chance of divergence. The RMS residual should decline smoothly. In the early stages of iteration, the residual at the first SOR step of one Newton step will be nearly that after the last SOR step of the previous Newton step. As iteration proceeds, this may no longer be the case, and each curve will start closer to where the previous one began than to where it ended. If the successive curves for RMS residual start to rise, however, this indicates the iteration is diverging. The second plot will reveal the divergence more clearly.

Plot window 2 shows the RMS residual in blue, RMS solution increment in yellow, and RMS difference between the real and model images in green, as a function of Newton-Raphson step. Once the curves reach the right side of the plot (50 Newton steps), they begin to scroll to the left, with the most recent results appearing at the righthand edge of the plot from then on. Some of the important features of this plot to watch for are as follows:

- The RMS difference between the image and model is scaled so the image value on the datum is 1.0. This curve normally declines rapidly at first, then asymptotically approaches a level that is determined largely by noise in the image.
- The RMS residual also declines rapidly at first, then more slowly. When the decline slows sufficiently, it is time to decrease the working resolution. Usually the RMS residual is the same at the lower resolution as it was at the previous higher resolution, and initially declines rapidly until it slows at a lower value. Depending on whether progress again becomes slow (the curve levels out) or the residual decreases to a small fraction of its higher-resolution value, the resolution will be decreased further or increased.
- When the resolution is increased, the residual will temporarily increase to a value substantially larger than had previously been achieved at the higher resolution. This normal and results from errors in interpolating the changes to the solution made on the coarser grid onto the finer grid. Because these interpolation errors are highly localized, they will be eliminated in only a few iterations. The RMS residual should decrease to significantly less than its final value from the previous time at this resolution before progress again becomes slow.
- If WMAX and ITMAX are set too high for a given image, the RMS residuals may decrease for several steps, then suddenly increase, then decrease for several more steps, etc. If there is overall progress toward lower residuals this may be acceptable, but a few steps using underrelaxation (WMAX<1.0) at the end of iteration will "polish" the results and achieve even lower residuals. If the residuals are not, on the whole, decreasing, then WMAX and ITMAX should be reduced.
- Divergence of the iteration process is indicated when the RMS residual begins to increase steadily. The RMS solution increment will also start to increase. In this situation WMAX and ITMAX should definitely be decreased; it may be necessary to start the whole calculation over and use smaller values for these parameters from the beginning.

Log Window

The log window is divided into two sections that show, in text form, much of the same information that appears in the plot windows. This information may be of interest to some users and can be helpful in diagnosing problems, but the program can normally be run without reference to this window. It may be of some interest to note that the tables printed in the lower half of the log window were the main form of user feedback for controlling the photoclinometry algorithm from 1982 until pc2d was developed in 2002. The window is deliberately small in order to conserve display space, but the individual sections can be scrolled to review previous output.

The top half of the window shows almost the same data that appear in plot window 2. This information is also output to the optional LOGFILE if one is requested. For each Newton-Raphson step, the log shows the number of floating operations performed so far, the working resolution, and RMS values of the residual ("RMS E"), difference between the true and model image ("RMS DB") and the total topography in pixel units ("RMS Z"). Note that plot window 2 shows the RMS increment to the topography, not the total topography. An additional column in the top half of the log window, labeled "Contrast", prints a dummy variable that is always equal to 1. This is present for historical reasons and can be ignored.

In the bottom half of the log window, a small table is printed for each Newton-Raphson step, showing the progress of SOR or ICCG iteration within this step. For each SOR (or ICCG) step, the RMS magnitude of the SOR increment to the topography is shown along with the RMS residual and the evolving sum of SOR increments, which will be the solution increment for the Newton step as a whole. This information thus corresponds to plot window 1. When the Newton step is completed, the equations are re-linearized and additional statistics are printed at the bottom of the table for that step. These include not only the recalculated RMS residual, but the ratio of this to the residual at the previous Newton step, the ratio to the residual at higher resolution (if any) and the ratio to the estimated truncation error. As described above, it is these ratios that are used to determine when to "hint" that the resolution should be changed.

Post-Processing of the DEM

The many possible paths that scientific analysis and visualization of photoclinometric DEMs can take are beyond the scope of this document, except for a note about how the coordinate system used affects interpretation of the results. The ISIS photoclinometry software uses an image space coordinate system to solve for topography from framing camera images: the DEM grid is defined in a plane parallel to the image plane, and "height" is measured in the direction toward the spacecraft. For oblique images, the datum (mean ground plane) is thus inclined relative to the internal Z=0 surface. Ideally, once the shape of the surface is solved for in this oblique coordinate system (which is convenient because the DEM points do not move from pixel to pixel as their heights are adjusted), a full three-dimensional rotation to ground coordinates should be performed. Such a rotation would remove parallax distortions from the DEM, but it has not been implemented because of the difficulties that arise with uneven sampling, hidden surfaces, for an arbitrary rotation. Instead, we have implemented a much simpler but less accurate transformation in three steps:

1) The line-of-sight "height" coordinate of the datum is subtracted from the "height" of the DEM surface at each point. This gives the oblique distance from the surface point to the datum.
2) This oblique distance is multiplied by the cosine of the emission angle before the results of pc2d, pcsi are saved. This factor corrects the oblique distance-to-ground to vertical distance-to-ground, i.e., height.
3) The output is still sampled evenly in the image plane. If desired, the DEM may now be map-projected to obtain (approximately) even sampling in the ground plane. The ISIS map projection software can make use of a DEM to remove parallax distortions when projecting data, but doing so requires that the DEM data already be available in map coordinates. Therefore the photoclinomery results must be projected either onto an ellipsoid representing the datum or onto a coarse DEM from another source such as stereo or altimetry.

The map-projected DEM produced in this way will still register to the map-projected version of the image from which it was made, and the relative heights of features will be accurate, but local distortions due to parallax will not be corrected. The magnitude of these distortions is proportional to the product of two (hopefully small) angles in radians, the image emission angle and the typical surface slope angle.

Photoclinometric DEMs produced from map-projected images or mosaics will register to their input image or mosaic and will contain the unrectified parallax errors in those products. Scanner images are handled in a manner similar to map-projected images, in that the time-angle coordinates of the image are interpreted as approximately Cartesian coordinates on the ground, albeit with nonsquare pixels. Once again, the clinometric DEM will contain the same parallax errors as the input image as a result.

Destriping

Photoclinometric DEMs can nearly always be improved by specialized digital filtering to suppress the artifacts caused by albedo variations before they are used for other applications. These artifacts, which take the form of stripes, can be objectionable even in cases where the image does not reveal any obvious albedo variations. The artifacts arise as follows: consider an spot on the surface with a higher reflectivity than its surroundings. The photoclinometry algorithm will interpret the resulting bright spot in the image as a patch tilted toward the sun. This patch will be flanked by a trough running from its low edge toward the sun and a ridge running away from the sun, neither of which is visible because slopes at right angles to the sun direction (strictly, the azimuth of characteristics) have minimal effect on brightness. Conversely, a dark spot will be flanked by an upsun ridge and downsun trough, and a mottled surface will produce a DEM crossed by multiple ridges and troughs. The anomalous slope in each patch of different albedo cannot be removed from the DEM, but the stripes can be suppressed by filtering.

The first step in "destriping" a DEM is to rotate it so the azimuth of characteristics is aligned with the sample direction. If desired, the image can first be enlarged in order to reduce the amount of degradation caused by the required resampling. Programs `pc2d`, `pcsi`, and `pcinfo` each print the azimuth of characteristics in the print.prt file, and `pcinfo` also returns it as a parameter for use in a processing script. Program `random` yields a transformation file that is applied by `geom` to resample the file. Once the DEM is correctly oriented, the ISIS procedure `dstripe` can be used to perform a destriping operation, which consists of three steps

1) The DEM is lowpass filtered with a boxcar 1 line by SAMP samples. This eliminates features less than SAMP samples wide.

2) The lowpass filter result is highpass filtered with a boxcar LINE lines by 1 sample. This eliminates features more than LINE lines high. The result thus contains features that are both short and wide, i.e., stripes of a certain size.
3) The result of the highpass filter operation is subtracted from the original DEM, yielding a DEM from which stripes in a particular size range have been removed.

Several successive destriping operations are usually needed, each removing stripes of a particular minimum width and maximum height. In general, it works well to set SAMP = 3*LINE, i.e., to define stripes as features that are at least three times as wide as they are high. This ratio is a compromise between leaving traces of the stripes near the albedo feature that causes them, versus removing real topographic features. Successive destriping passes can then be run with both dimensions increased by a factor of 3 each time. A typical set of filters thus might be 5x3 pixels, 15x5, 51x15, and 151x51. Once the destriping process is complete, the DEM is rotated back to its original orientation with `random` and `geom` and, if necessary, rescaled to its original size.