Leveraging modern desktop computing power to enable full-fledged gravity terrain correction processing

P. Linzer

SRK Consulting, South Africa

Gravity surveying is a geophysical method for detecting subsurface density variations in mining and mining-related endeavours, such as mineral exploration, mine planning, and infrastructure siting. Such density variations can be produced by a variety of subsurface features, including orebodies, cavities, large water-bearing fissures, karst profiles, etc. Gravity surveys can also augment mining safety and risk management, e.g., as a tool for the early detection of incipient sinkholes in opencast operations. To be useful, the raw gravity field data require several reduction and processing steps, one of which is gravity terrain correction (GTC) processing. The latter step is often omitted owing mainly to its computational demand, thereby diminishing the value of the final survey data. This paper describes how recent advances in modern desktop computing technology with features such as hyper- and multithreading, multicore central processing units, instruction set extensions, and more, have made exhaustive GTC processing hugely more practicable. A resultant GTC processing code derived, prototyped, and developed in-house is briefly described. Finally, a synthetic example of a gravity survey conducted in a hypothetical open pit mine to detect subsurface cavities is presented, which illustrates the difference that proper GTC processing can make to a gravity survey's target delineation, definition, and contrast.

INTRODUCTION AND BACKGROUND

Problem outline

One among several geophysical techniques, the gravity survey is used for the detection of subsurface density anomalies, both high and low. It involves measuring the gravitational potential with a sensitive instrument called a *gravimeter* at a collection of points ('stations') on or near the Earth's surface, usually lying on a predefined grid, which forms part of a detailed gravity survey design that considers the survey's purpose and location.

The gravity method is useful in several mining and mining-related activities including mineral exploration, orebody and geological structure mapping, mine design and planning, infrastructure siting, mapping subsurface karst profiles, groundwater exploration, hydrogeological surveys, oil and gas exploration, and the detection of subsurface features such as cavities, large water-bearing fissures and aquifers. Some more obscure applications of the method include the mapping of old mine workings, and finding old bomb shelters, hidden tunnels, and pipes and sewers. These various applications identify the many potential causes that can produce subsurface density anomalies.

Gravity surveys can also assist with mining safety and risk management initiatives, for example as a method for the early detection of cavities that could become sinkholes in open pit ventures, since such events will endanger personnel and equipment, besides disrupting normal production operations.

However, the success and usefulness of a gravity survey's results depends critically on the correct application of the appropriate field discipline and protocols that will maximise the ultimate value of gravity survey data.

Gravity data corrections

Equally importantly, the raw gravity readings require processing to remove the effects of known phenomena that influence the data values before they can become fully meaningful. These corrections include:

- Gravimeter drift – temperature and instrument characteristics change slowly during use
- Latitude – Earth is spinning and is not perfectly spherical
- Elevation – a.k.a. free-air; distance to Earth's centre, per Newton's $g \propto 1/r^2$
- Bouguer – the mass between the reference and measurement elevations
- Earth tides – Moon's effect
- Regional – deep, large-scale, gradual variations that aren't of direct interest
- Eötvös – only for moving platforms; Earth's rotation and any east-west motion of the gravimeter
- Terrain – uneven topography, especially in the nearfield.

This paper focusses on the last of these, the terrain correction, and its importance.

Terrain effect on gravity measurements

The importance of the terrain correction cannot be overstated, especially in gravity surveys where the surrounding topography is highly variable (Blais and Ferland, 1984; LaFehr, 1991), and/or where anticipated density anomalies are small. In extreme cases, the nearby terrain's gravitational effect can cause an anomaly to be missed or written off as insignificant, or even to be completely masked. An example of an extreme case is a survey conducted in an open pit where the benches and highwall can mask gravity lows related to subsurface cavities.

Briefly, the terrain effect arises from two related mechanisms. The first of these is the case of a nearby hill or other elevated topographical feature, including manmade features such as a rock dump, which introduces excess mass that exerts a gravitational attraction on the measurement station. This attraction has a vertical component that acts upwards, thereby reducing the gravity reading at the station. The magnitude of this reduction depends on several factors, mainly the elevation difference between the station and the elevated terrain, the distance between them, and the average density of the elevated terrain portion.

The second mechanism is the case of a nearby valley or other topographical depression, including manmade features such as an open pit, which introduces a mass deficit that lies below the measurement station. The effect of this mass deficit is to reduce the downward gravitational force component exerted at the station, thereby reducing its gravity reading. The magnitude of the reduction depends mainly on the elevation difference between the station and the depressed terrain, and the distance between them.

Thus, the presence of both nearby hills and valleys tends to reduce gravity readings, and therefore the aggregated terrain correction for a measurement station will always be zero or greater and needs to be added to the uncorrected gravity readings. Moreover, it is worth noting that the terrain correction is entirely independent of all the other gravity data corrections mentioned earlier. (To be clear, while the terrain correction is apparently similar to the Bouguer correction, the latter considers only the elevation difference between the measurement station and the reference plane, not the effect of surrounding elevation variations).

Determining the terrain effect

The calculation of the terrain correction requires the positions and elevations of the measurement stations as well as a sampling of data for known points of the area containing and surrounding the gravity survey area. The latter component constitutes a terrain model that approximates the topographic features of the survey area and its surroundings. The terrain model's required point data

consist of positions, elevations, and local rock column densities if they are available (which typically they are not, and so a uniform density is usually assumed). The terrain model should extend a few tens of kilometres in all directions beyond the survey area. The preferred extent is largely dependent on the terrain's vertical variability—the more variable, the larger the terrain model extent should be—but rarely, if ever, does it exceed 160 km.

Early approaches to calculating terrain corrections, starting as far back as 1912, were developed by workers such as Bullard and Hayford (Connor and Connor, 2017), but a properly formalised method was only presented in 1939 by Hammer (Hammer, 1939). This method is still in occasional use today owing to its relative simplicity. The method uses a circular template that divides the area surrounding a measurement station into equal sectors and concentric rings to form a system of compartments called a 'Hammer net' (see Figure 1).

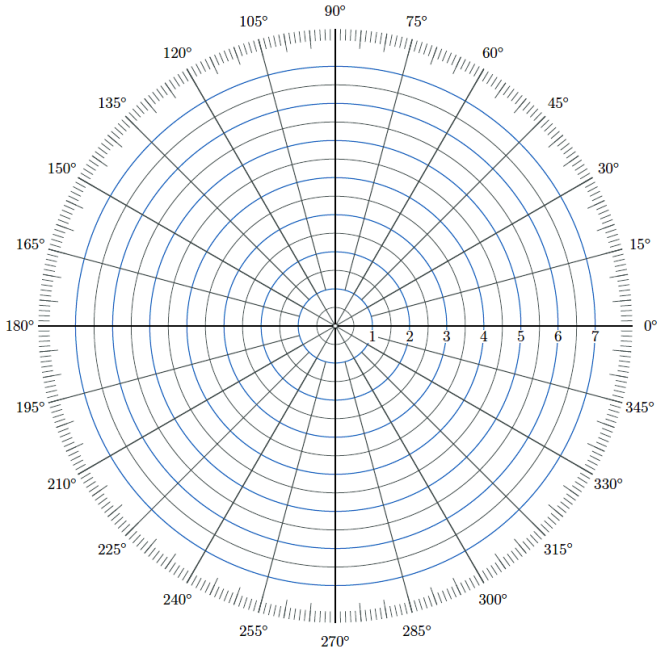

Figure 1. The Hammer net method template (adapted from Connor and Connor, 2017).

The Hammer net's centre is placed at the measurement station currently being considered on a topographic map, whereafter the average elevation in each compartment is calculated and used to determine the average elevation difference between the station and the compartment in question. Together with the inner and outer radii of the compartment, its angular extent, and a bulk density value of the terrain, a terrain correction value is calculated for the compartment. The total terrain correction for the station is then the sum of the corrections over all the compartments. Mathematically, it is a large numerical integral over the entire surrounding area for each station.

Hammer's method is obviously repetitive and laborious, and therefore prone to calculation errors if done manually – "Terrain corrections can be extremely tedious" (Milsom, 2003). For the same reason, the method lends itself well to computerisation. Some later methodological developments include fast Fourier transform computation of the terrain effect, the use of digital elevation models (DEMs), and model approaches that variously involve conic prisms, multiquadric equations, assumptions of uniformly sloping surfaces, and statistical assumptions regarding the areal distributions of elevations (Banerjee, 1998). These efforts have all sought to reduce the computational effort required to calculate the terrain effect reliably, and some of the later approaches can be classified as proxy methods because they avoid direct point-by-point calculation. The main shortcoming of all these proposed enhancements is that they tend to lose relevance precisely where accuracy is most critical, i.e., where the terrain varies wildly in the near vicinity of a gravity survey area.

The example Hammer net shown in Figure 1 has a total of 336 compartments. When using a DEM rather than a topographic map, each compartment can contain several elevation points, all of which need to be included in the calculations. Since a DEM is typically on a regular (near-)rectangular grid, the number of points in the compartments increases as their distance from the station at the centre of the net increases. Moreover, the calculations need to be done at every station, and a survey can consist of a few thousand stations, or even several hundred thousand for a large airborne survey. The number of point calculations is given by the product of the number of DEM points and the number of stations, which helps illustrate the formidable computational magnitude of the problem.

Modern desktop computing developments and their role

Several advances in desktop computing technology, principally hardware but also including software, can contribute significantly to alleviating the computation of terrain corrections. Among a few others, the hardware advances related to a modern personal computer's central processing unit (CPU) involve:

- Full 64-bit processing (providing faster primary operations, and the use and management of very large memory blocks)
- Instruction set extensions such as SSE*n* and AVX*n* that implement the SIMD concept (SIMD = Single Instruction, Multiple Data, allowing the same operation to be executed simultaneously on more than a single set of operands, e.g., pairwise multiplying together two equal-length lists of numbers; SSE*n* = Streaming SIMD Extensions version *n*, which provides some additional large CPU registers; and AVX*n* = Advanced Vector Extensions version *n*, which provides several further complex SIMD operations and up to 32 additional *very* large registers – up to 512 bits – for executing SIMD-type operations even more rapidly by avoiding external memory access penalties)
- Hyperthreading (within certain constraints, the CPU can execute two sequential instructions simultaneously in the same clock cycle), and
- Multicore processors for true parallel processing (a single CPU die can contain more than one physical processor, currently as many as 32).
  (Liang, 2019)

Other hardware developments include larger and faster random-access memory (RAM) chips and more powerful graphics processing units (GPUs), which handle operations related to drawing and rendering of images on the computer screen. Modern GPUs are not passive in the sense that older-generation graphics cards were. The latter were fed final drawing instructions by the CPU after the CPU had processed data appropriately for the intended image. GPUs can accept unprocessed data and have their own data manipulation firmware for performing a range of raster and vector operations, thereby relieving the CPU of what could be significant computational burdens.

With regard to software improvements, pre-emptive multitasking operating systems (OSes) such as the Microsoft Windows NT family have benefitted from some of the aforementioned hardware developments, particularly in respect of using a CPU's multicore capabilities to implement true multitasking (in contrast to single-processor time-slice allocation and task-switching, both of which still occur, only on all available processors simultaneously) and providing the application programming interfaces required by user code for managing these facilities. Equally importantly, these OSes have seen some notable stability improvements over the recent past, increasing their robustness and native capabilities.

In addition, modern computers come equipped with unified extensible firmware interface (UEFI)-compliant basic hardware input-output firmware system via which the OS uses and manages connected or integrated hardware devices. A common feature in UEFI systems is the ability to virtualise each single *physical* processor into two (occasionally four) *logical* processors. That is, a 16-core processor can be made to look as though it had 32 processors available. This feature can be useful for running more than one virtual machine on a single system but introduces additional task-switching penalties. For high-intensity computing, CPU virtualisation should therefore be disabled.

Regarding GTC processing, these modern computing system features can be exploited by adapting the

problem to suit the available computing environment. Perhaps most obviously, the problem is eminently parallelisable because not only, as mentioned earlier, is each station's terrain correction calculation entirely independent of any of the other corrections (drift, latitude, etc.), each is also completely independent of any other station's terrain correction. This independence means that the calculations can be apportioned uniformly[1] to as many processors as are available, and even across multiple separate computer systems.

The gravity terrain correction code

Initial development and testing

During 2022 and 2023, a project commenced aimed at identifying subsurface cavities at the floor of an open pit mining excavation. A gravity survey was recommended and performed. The gravity data were properly reduced and processed, except for GTC processing. The final non-GTC survey data displayed clear artefacts produced by the nearby topography, specifically the benches, highwalls, and a large man-made structure. (Client confidentiality restrictions prevent the inclusion of further details and illustrative material concerning this case.)

The reason for omitting GTC processing initially was the unavailability of suitable software to perform this processing step. Although several codes are mentioned in the subject matter literature, in many cases they are no longer available, or they are outdated in terms of the platforms on which they will run, or they are too limited in terms of their capabilities and input data specifications, or all of these limitations together. And while some commercial packages and package extensions are currently available, their effectiveness is uncertain (apparently relying on simplifications and proxy approaches to improve speed, as described earlier), and their cost tends to be high.

These defects prompted considering the possibility of preparing an in-house GTC code. The principles of the Hammer net approach guided setting up a model of a survey station surrounded by a collection of vertical slabs where each DEM point represents a single slab. The required calculations were derived from first principles before being thoroughly examined for the purpose of eliminating any mathematical constructs that could result in computational inefficiencies if naïvely implemented. These optimisations included pre-calculating terms that would be repeated at a station for different terrain points being considered, the removal wherever possible of trigonometric functions in favour of Pythagorean ratios (the computation of trigonometric functions is *much* more time-consuming than square roots or straight ratios, typically by a whole order of magnitude or more), and replacing repeated divisions by constant factors or terms with multiplications by their inverses (division is computationally more expensive than multiplication).

These derivations and analyses revealed two tweaks that could be made to the adopted method to improve its accuracy, and they were duly included.

The refined algorithms were then implemented for rigorous testing in SAS[2], a powerful scripting environment for statistical and mathematical analyses. In addition to the measurement station positional data, a large DEM drawn from public-domain Shuttle Radar Topography Mission (SRTM) data, was used for testing. The DEM had a 1 arcsecond horizontal resolution (±31 metres on the ground at the equator), while elevations were accurate to the nearest full metre. The SAS implementation was verified for correctness using an independent Microsoft Excel implementation (i.e., essentially a manual validation exercise) for several portions across a number of different measurement stations. Based on this testing, the SAS version was subsequently accepted as being correct and the standard for assessing any subsequent implementations.

[1] Uniform portion allotment comes with some challenges that will be discussed in more detail shortly.
[2] SAS = Statistical Analysis Software, see https://www.sas.com

The gravity survey data of the aforementioned project were then treated with the SAS GTC implementation and the terrain artefacts became noticeably less pronounced, allowing more reliable identification of subsurface anomalies. This first job also revealed the potential for a rare pathological case that was subsequently catered for in the code.

Standalone GTC code development

A later, much larger GTC task for a mineral exploration project arose, which showed the limitations of a scripting environment such as SAS in which code requires a purpose-built runtime environment that interprets and executes the scripted code. The runtime engine overhead slows down the execution rate significantly, requiring many CPU instructions to perform what appear to be even the simplest of scripted commands, even in runtime environments that execute precompiled intermediate code such as Python or Java.

This bottleneck necessitated the porting of the SAS code to a programming environment that would produce compiled native code that the CPU can execute directly without needing any intermediate support. Embarcadero Delphi RAD Studio v11.0 was used for this translation, which also enabled several other code enhancements and tweaks:

- No object-oriented programming constructs were used to avoid the additional processing overhead that these introduce.
- Multithreaded parallel processing was implemented in such a way that the code automatically deploys as many processing threads as there are logical CPUs (henceforth 'cores') available on the machine on which it is running, and each thread is assigned to a specific core to process a unique portion of the overall job, so as to make use of all available computing power. Each thread is in effect completely autonomous and produces its own output data file.
- The use of standard Delphi library code was extensively avoided in favour of direct Windows system calls (e.g., for file IO) and in several cases with homegrown replacement routines (e.g., trigonometric functions). The reason for doing this is that library code adds processing overhead, usually because it is generally applicable and caters for cases that simply do not occur in the code. For example, the standard trigonometric sine function, sin(x), makes provision for values of x that lie well outside of the normal $0 \leq x < 2\pi$ (radians) range, but the GTC code expressly prevents such out-of-range angles ever arising, so that the additional processing overhead is pointless and wasteful.
- A few frequently called primary functions were coded in x64 Assembly language to make greater use of the SIMD extensions after it became clear that the code compiler did automatic vectorisation only sparingly.
- A console program setup was used rather than a Windows GUI arrangement because writing user feedback in a console uses asynchronous buffered lazy writing instead of putting the writing program into a wait state until a GUI write operation has been completed, which can give a console process a slight performance advantage.
- The code allows a job to be parcelled out to multiple independent computers.
- The job configuration is defined in a simple text-based initialisation file.
- A running job can be halted at any time by a user and resumed later where it left off.
- The program's own and all its threads' execution priorities are set to the lowest rank to avoid. the computer becoming unresponsive owing to high CPU load.

The approach of replacing library code with appropriate homegrown substitutions improved the GTC code's overall processing speed by 15 to 20%, compared to a benchmarking implementation that used only library code. The results were identical between the two versions.

It is therefore clear that the GTC program has a significant CPU demand, but it can also have a high RAM requirement. This aspect is determined by a combination of the number of stations, the number of DEM points, and the number of processing cores. The RAM is required for storing precalculated data. On startup, the GTC program will check for available physical RAM and emit a warning if it is insufficient but will commence processing regardless. The purpose of the warning is to alert the user because Windows will provide virtual RAM if physical RAM is insufficient, but this mechanism will

significantly impede processing rate.

GTC code input data requirements
Apart from the aforementioned job configuration file, the GTC code requires two separate input data files. Both of these files are expected in text-based CSV format. The first of these input files gives the survey stations' positional information, namely station locations in GPS coordinates and station elevations. The second input file similarly defines the DEM in terms of DEM point positions, but it also includes the rock column relative density for each DEM point. (Typically, these rock column densities are unknown, in which case a uniform representative value such as 2.5 is assumed.)

It should be clear that the DEM input data file is invariably much larger than the survey station definition file, containing many more points, including ones within the survey area itself. Indeed, the station positions themselves are included in the DEM data.

The DEM itself can be constituted from several different DEM or DTM sources (e.g., SRTM and any in-house client terrain elevation data) but it is critically important that they are all corrected to the same elevation datum before they are amalgamated.

In all cases where the GTC code was applied up to the present, it was found that the elevation datums differed between the DEM and the survey area. The reasons for this difference in datums include different surveying techniques, topographical changes over time, and different reference datums. Obviously, an appropriate reconciliation between the different datums was needed to avoid erroneous GTC results, and this problem was at first addressed via a separate SAS script but later coded into a separate GTC support program owing to the problem's ubiquity and the relative slowness of the SAS script for performing the datum reconciliation.

Additional GTC support codes
The datum problem was addressed using an inverse-distance weighted scheme that considers the three horizontally nearest DEM points to each station to calculate a datum correction. A single overall datum shift is then calculated from the individual stations' corrections using the inverse of the distance to the nearest DEM point as a weighting factor. This final datum correction is then applied to each station elevation. As mentioned, this scheme was implemented as a separate GTC support code, in this case a required preprocessing step.

Four other support codes were found to be necessary to address tangential GTC processing issues:

- A SAS script for importing raw gravity survey and DEM datasets, incorporating station data in the DEM, trimming the DEM to a circular area of sufficient size (default is 100 km in all directions beyond the survey area), and exporting them in the correct CSV format, as required by the GTC code. The script includes simple plotting features that allow the user to verify geographical correctness of the survey and DEM areas, and to examine the relationship between their datums
- A separate standalone code that calculates for each survey station a series of geometrical statistics for all DEM points that lie within a user-defined distance of the station, say 100 m. This utility is useful for identifying stations at which GTC calculations are potentially problematic, for example where a DEM point lies very close to a station but the two have significantly different elevations
- A crash recovery program. It has happened occasionally that high-spec computers shut down suddenly because of CPU overheating (a.k.a., thermal events) from sustained near-100% load. The recovery program trims trailing partial writes from the output data files and corrects / updates the job configuration file so that GTC processing can later be resumed with each thread starting from the last properly completed station
- An assistant application for deploying a job on more than one computer. This program accepts a list of target computers together with certain technical specifications of each one (CPU type, clock speed, cores, installed RAM) and assigns the processing in such a way that the expected completion times for all machines are all equal.

There was also an earlier additional support code the functionality of which was subsequently added to the GTC code itself. This functionality automatically reassigns the processing threads to different cores, depending on their processing rates, with the aim of all threads completing more or less simultaneously to minimise total processing time. The main reasons for including this feature are (i) the major CPU manufacturers (chiefly, Intel and AMD) have a mix of so-called 'performance' and 'efficiency' cores on their CPUs, which core types intrinsically process at different rates; (ii) other concurrent processes often place intermittently high loads on a few selected cores, and (iii) Microsoft Windows CPU load balancing is often suboptimal with different cores being assigned disparate loads.

The GTC program in action

Program appearance and status reporting

Figure 2 shows the GTC program while running. As described earlier, the user interface is in the form of a console. While rudimentary, it provides information in the form of an optional job name, the statuses of input data file reads, the size and station count of the survey, and the size and point count of the DEM. The term 'radius' is used to describe the smallest circle that contains all of the survey stations and should not be read as implying that the survey itself covers a circular area. In contrast, the DEM area is indeed circular, having been trimmed thus by the preprocessing SAS script.

In addition to the job parameters, the console also displays per-thread progress, the time it has been active, and its overall progress. In the example shown, five of the 12 threads have completed their respective portions of the job. Some system menu, title bar, and keyboard shortcut controls of the console have been disabled to avoid inadvertent closing of the program, and the user is required to use the standard console **<Ctrl>+<C>** keystroke combination to terminate a processing session gracefully. On termination, the program reassigns the threads according to their individual rates (slowest to fastest and vice versa) and displays the overall processing rate for the session in terms of the number of stations completed per minute.

Finally, any errors that may occur – for example, an invalid job specification file or faulty input data files – are reported on the console as they are encountered. Any error terminates the job, but the console will remain open until the user acknowledges it.

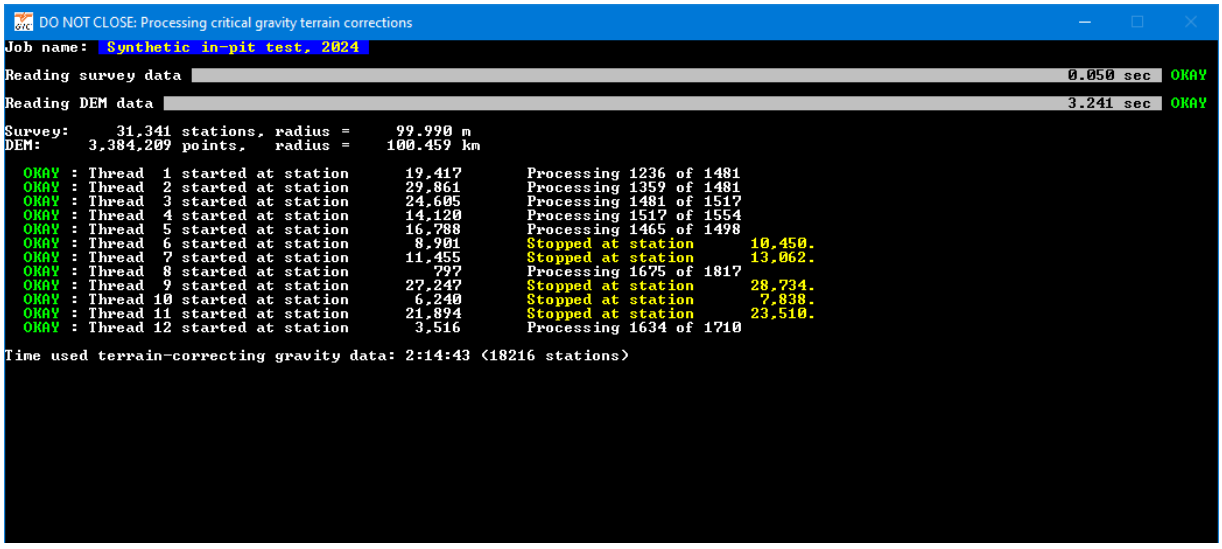

Figure 2. The GTC code running on a 12-core laptop (Intel 2.2 GHz i7 CPU + 48 GB RAM).

A real-world example of GTC processing

Figure 3 shows an excerpt of the same area from two final gravity data plots of a large airborne gravity survey done on a regular grid point spacing of about 25 m. All the required gravity data corrections

were performed in both cases, except for GTC processing, which was omitted from the plot on the left. It is worth noting that in many cases, the plot on the left is considered the final output of a gravity survey because the terrain correction, being problematic, is often omitted for reasons stated earlier.

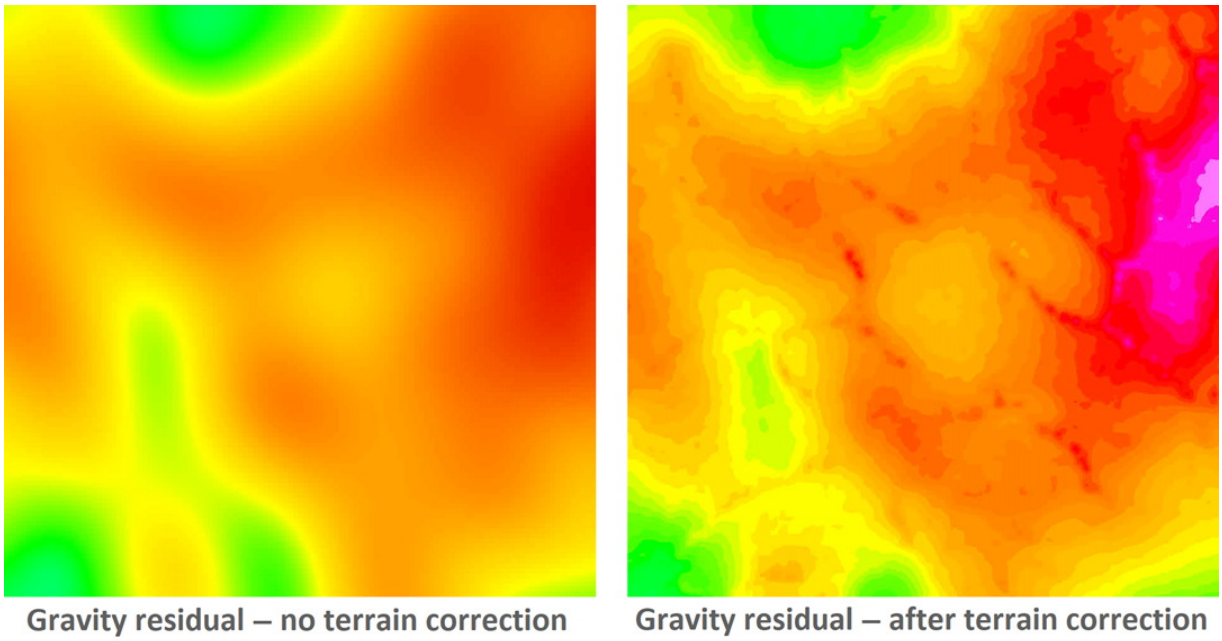

Figure 3. Areal gravity residual plots of the same area. GTC processing was omitted from the plot on the left.

The differences between the plots are immediately clear. In this case, GTC processing has significantly enhanced gravity residual definition, contrast, and resolution with intensity contours being more unambiguously demarcated. The terrain surrounding this survey is mountainous with nearby elevation variations of a few hundred metres, and their combined effect appears to have smeared out the areal gravity variations.

It was noted that some of the GTC-enhanced gravity features seem to echo the terrain's local topography, and many features identified by this survey correlated well with the results of a simultaneous aeromagnetic survey. However, a detailed analysis and interpretation of the results lies outside of this author's area of expertise and is not this paper's primary focus.

A synthetic example of GTC processing

As a further illustration of the value that proper GTC processing can add to a gravity survey, a synthetic example was created of a hypothetical open pit mine at the floor of which subsurface cavities are sought using a gravity survey.

The pit is assumed to have a circular footprint with five benches and a spherical subsurface cavity located halfway between the pit floor centre and its periphery. Figure 4 shows a vertical section through the pit's centre with its salient dimensions.

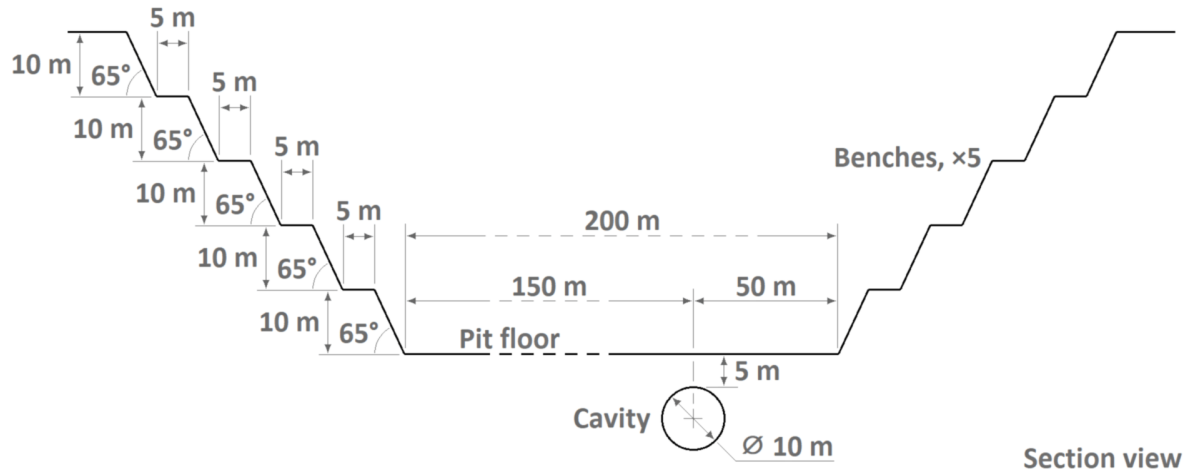

Figure 4. Section view showing the geometry and dimensions of a hypothetical circular opencast pit with a shallow spherical subsurface cavity off to one side.

The surface terrain surrounding the pit is assumed to be sufficiently flat and even in all directions, and its effect is therefore omitted from the GTC processing. Thus, only the effect of the benches on gravity at the pit's floor is considered in this example.

Figure 5 shows the difference between the non-GTC and the GTC processed data when plotted. The non-GTC processed data (plot on the left in the figure) shows a systematic increase in gravity values from the pit floor's edge towards its centre. This effect is wholly attributable to the presence of the benches which lie above the measurement elevation. The benches exert a net upward gravitational force at the measurement stations which force decreases as the distance from the benches increases, explaining lower gravity values at the pit floor periphery compared to its centre.

As can be seen, once the effect of the benches has been removed by GTC processing, the presence of the cavity is brought into much starker relief (plot on the right in the figure).

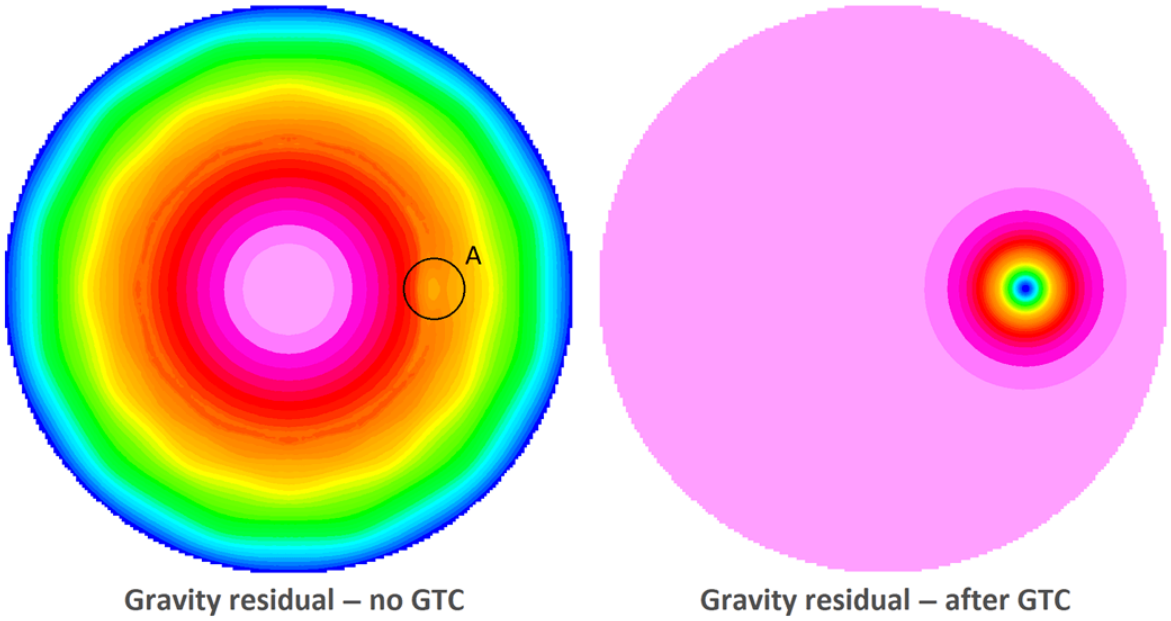

Figure 5. Gravity residual plots at the pit floor for the synthetic example with and without GTC processing.

The non-GTC plot also identifies a small anomaly labelled 'A' that resulted from the cavity. The anomaly is very small in magnitude relative to its immediate surroundings, and it could therefore easily be written off as insignificant or even missed entirely. It should be noted that the terrain's effect can mask faint signals that may be of critical importance. If, in a similar real-world scenario the presence of the cavity was missed due to it being masked by the benches and it were subsequently to collapse, the mine operator might rightly question the usefulness of doing gravity surveys as a safety precaution against such an eventuality.

CONCLUSIONS AND RECOMMENDATIONS

By way of both a real-world and a synthetic example, the present work has demonstrated the value that full-fledged GTC processing can add to gravity surveys. The in-house code development for this purpose was described in some detail and with reference to several modern desktop computing system features that were exploited to enhance the code's efficiency, given that it is a computationally highly demanding task.

The benefits provided by GTC processing include improved target detection and isolation, and clearer resolution of subsurface density variations. GTC processing becomes increasingly important where anticipated targets are faint and/or where the surrounding terrain is highly variable, especially in the nearfield region. These benefits can assist with or improve project execution in a variety of ways across several disciplines, including mining, civil engineering, geological, exploration, and geotechnical work. Wherever a gravity survey is desired or indicated, GTC processing can be advantageous.

However, a useful rule/guideline/formula/heuristic is still needed to assist with deciding whether proper GTC processing is justified in a given gravity survey scenario. For now, it is recommended that it be done for every survey since doing so will provide the necessary empirical data to establish the missing decision rule(s).

Besides the need for these decision principles, techniques for progressively coarser-grained re-gridding of the DEM with distance from its centre should be examined, owing to the significant reduction in computational effort it could entail.

Finally, where reprocessing of historical gravity data to include GTC is being contemplated, it should be borne in mind that the surrounding terrain may have changed, perhaps significantly so in the immediate vicinity (such as in the case of an open pit mine), from the time when the survey was done, and therefore current DEMs may no longer be sufficiently accurate to facilitate meaningful improvements. In this regard, gravity practitioners should not neglect the terrain's time dependency.

REFERENCES

BANERJEE, P. (1998). Gravity measurements and terrain corrections using a digital terrain model in the NW Himalaya. Computers & Geosciences 24(10), 1009–1020.

BLAIS, J.A.R. AND FERLAND, R. (1984). Optimization in gravimetric terrain corrections. Canadian Journal of Earth Sciences 21, 505–515.

CONNOR, C. AND CONNOR, L. (2017). Gravity Lecture 7 – The Terrain Correction. University of South Florida. http://www.cas.usf.edu/~cconnor/pot_fields_lectures/Lecture7_graity_terrain.pdf (retrieved 18 October 2023).

HAMMER, S. (1939). Terrain corrections for gravity stations. Geophysics 4(3), 184–194.

LAFEHR, T.R. (1991). Standardization in gravity reduction. Geophysics 56(8), 1170–1178.

LIANG, S.L. (2019). Understanding Windows x64 Assembly. https://sonictk.github.io/asm_tutorial via https://www.sonictk.com/post/understanding-windows-x64-assembly (retrieved 08 April 2024).

MILSOM, J. (2003). Field Geophysics Third Edition. John Wiley & Sons Ltd, ISBN 0-470-84347-0.

Patrick Linzer

Principal Data Analyst
SRK Consulting (Pty) Ltd

2019-Present	SRK Consulting South Africa (Pty) Ltd, Principal Data Analyst, Johannesburg
2013-2019	MBD Credit Solutions/Transaction Capital Recoveries, Senior Analyst – Scoring, Johannesburg
2005-2013	African Bank (Pty) Ltd, Senior Statistical/Decision Analyst, Johannesburg
2004-2005	TracIT Solutions, Senior Developer, Pretoria
2000-2004	Diamond Network Technologies / PreWorx, Senior Developer, Pretoria
1992-2000	COMRO/CSIR Division of Mining Technology, Senior Research Engineer, Johannesburg
1983-1992	Harmony Gold Mining Co. Ltd, Learner Official, Shiftboss, Planning Officer, Virginia
1988-1990	SA Defence Force, National Serviceman, Junior Leader, Corporal, Kroonstad & Pretoria