CALCULATION OF MICRO-GRAVITY USING A REFINED TETRAHEDRAL SCHEME FOR USE WITH RESERVOIR SIMULATORS

Peter Franz[1]

[1]Mighty River Power, 283 Vaughan Road, Rotorua, New Zealand

peter.franz@mightyriver.co.nz

Keywords: *micro-gravity, TOUGH2, TETRAD, VTK.*

ABSTRACT

Micro-Gravity is a valuable tool in reservoir engineering to monitor changes in fluid density during production and injection of geothermal fluids. Survey data from micro-gravity stations at regular intervals is therefore an important constraint to numerical reservoir simulations.

However, comparison between survey data and simulator output is not straightforward. While reservoir simulators like TOUGH2 and TETRAD can output relevant thermodynamic and porosity data, the accurate calculation of micro-gravity is difficult due to the highly geometric nature of the problem and the $1/r^2$ dependency of gravity.

The algorithm presented here decouples the time-dependent calculation of the fluid density in the model blocks from the static calculation of geometric factors between each block and each gravity station. The static volume integral over each model block is approximated by a series of tetrahedral refinements of the block. Comparisons between this method, the analytical solution of a sphere, cube, and the commonly used point-mass approximation are shown.

1. INTRODUCTION

Isaac Newton's studies into the effects of gravity signify the beginning of modern physics. Hence the mathematical treatment of gravitation is one of the most basic equations taught to every physicist and engineer.

Newton focused his studies on the movement of celestial bodies, characterized as very heavy objects separated over huge distances in space. However, the Newtonian treatment of gravity also holds true for small masses at short distances. To observe these effects in nature one needs to refer to specialized instruments which can resolve tiny changes in gravity.

Micro-gravity surveys are a valuable tool in geoscience. For the geothermal reservoir engineer they provide a window to observe changes in mixture density deep down below the ground. Typically these changes can describe shifts in the water table or change of fluid density – either by thermal expansion or by changes in gas saturations.

Surveys performed at regular intervals primarily document the change in spread of two-phase zones in a geothermal reservoir or the penetration of injection fluids into previously void rock structures. Surveys are typically run at intervals of several years. Care needs to be taken that measurements are made at the same locations (called stations) and the effects of shift in groundwater tables may need to be taken into consideration before the data can be used to describe changes in the geothermal reservoir.

Owing to the nature of the measurements, micro-gravity surveys generate 2D maps of gravitational changes relative to a primary survey. Quantitative interpretation of these maps can be difficult since each gravity measurement represents a volume integral of density weighted by the inverse of distance squared. Thus small changes in density close to a station can contribute similarly as large changes in density far away from a station.

Numerical reservoir simulators are widely used to model the thermodynamic changes in geothermal reservoirs in 3D. The calibration of any reservoir model requires a variety of physical measurements and observations, among them micro-gravity surveys. However calculation of micro-gravity from numerical models is not straightforward from standard reservoir simulator (e.g. TOUGH2, TETRAD) output. Many modelers therefore use approximations, like treating blocks as point masses and calculating gravity at regular locations (e.g. the center of a block surface) instead of mimicking the real world location of a station. Alternatively there exist some analytical solutions for common geometries, like the right-rectangular prism (Nagy 1966). However these usually involve multiple coordinate transformations and work only under certain geometric constraints.

The method presented here leads to an easy to use, accurate calculation of gravity from numerical reservoir simulator output and provides data which can be individually compared to station data.

2. MATHEMATICAL DESCRIPTION

2.1. General

At any given time, the gravity at a station (denoted by index j) can be calculated by summing the contributions from all N individual blocks (denoted by index i) in the model:

$$g(x_j) = G * \sum_{i=1}^{N} \rho_i * GF_{ij}$$

where G is the universal gravitational constant and ρ_i denotes the density of the block. GF_{ij} denote the static gravity factors between each block and station, and are calculated as:

$$GF_{ij} = \int_{V_i} \frac{x_j - x}{|x_j - x|^3} \, d^3x$$

where the volume integral needs to be evaluated over the block volume V_i. Note that GF_{ij}, g and x are vector quantities, though usually only the z-component of GF_{ij} and g need to be calculated.

The density of a block is calculated by summing contributions from all MINC layers:

$$\rho_t = \sum_m \left[\rho_{rock,m} * (1 - \varphi_m) + (SG_m * \rho_{G,m} + SL_m * \rho_{L,m}) * \varphi_m\right] * V_{fraction,m}$$

where m denotes the individual MINC layer with porosity φ and volume fraction $V_{fraction}$. SG and SL denote gas and liquid saturations and ρ_G, ρ_L the gas and liquid phase densities. The rock density part may be omitted if it is considered constant and only relative changes in gravity over time are of interest. Note that it is assumed that gas and liquid phase are well mixed over the block volume – tracking of a front or a water table is very complex and is beyond the scope of this paper.

Calculation of the density is straightforward, although the parameters need to be collated from different files. The approach taken here is to convert all the reservoir data into a visualization tool kit (VTK) file series before further processing. The added benefits are that both TOUGH2 and TETRAD simulator output can be used to generate standardized data, which can be plotted in 3D using tools like Paraview. Also the open source VTK libraries help with the task of calculating the gravity factors.

2.2. Gravity Factors

Calculation of the gravity factors is difficult since for proper evaluation of the volume integral not only the block shape needs to be considered but also the orientation towards the station. Only for a very limited amount of geometrical shapes the exact solution is known, for example the hollow or filled sphere, the infinite-length cylinder or the point mass. Nagy (1966) described an analytical method for integration of a right rectangular prism, which since has been extended by other authors for use with other polygons. However these methods usually still require certain criteria towards the shape and orientation of the objects.

The numerical algorithm presented here works on any given discretized block shape and any orientation, and is thus not limited to rectilinear blocks or symmetries.

Any given block shape is divided into smaller volumes which are treated as point masses, i.e. all the mass of the sub-volume is considered to be located at its center of gravity, x_c. The point mass approach for a gravity factor is:

$$GF_{ij} = \frac{x_i - x_c}{\|x_j - x_c\|^3} * V_i$$

The sum over all sub-volumes (cells) provides a better approximation to the volume integral than using the point-mass approach for the whole block. By controlling the local level of refinement it is possible to refine areas close to the gravity station to a higher degree than areas further away, thus taking care of the strong $1/r^2$ dependency.

At the 0th level of refinement, the block is treated as a point mass and the gravity factor $GF_{ij,0}$ is calculated.

Next, the block is divided into multiple tetrahedra by using its center of mass as a common vertex. The surface of the block is divided into triangles using a vtkTriangleFilter routine. Combining the triangles with the common vertex forms the tetrahedra of the 1st level. The number of tetrahedral formed in this level depends on the shape of the original block, for example there will be 4 tetrahedra formed if the original block was a tetrahedron, or 12 if the original block was a hexahedron. Treating these tetrahedra as point masses and summing over them gives the next refinement of the gravity factor, $GF_{ij,1}$.

The particular choice for using tetrahedrons as refinements was made due to practical reasons, since any 3D block shape (hexahedrons, wedges, tetrahedra) defined in a VTK unstructured grid can be easily broken down into simpler tetrahedral units by just following the rules given above.

Further, the volume and center of gravity of a tetrahedron can be easily calculated using simple vector algebra. If the 4 vertices of a tetrahedron are in 3D space given by vectors P_1 to P_4, the tetrahedron is spanned by vectors

$$r_{12} = P_2 - P_1$$

$$r_{13} = P_3 - P_1$$

$$r_{14} = P_4 - P_1$$

Thus its volume is calculated using a cross- and a dot-product:

$$V = \frac{1}{6} * (r_{12} \times r_{13}) * r_{14}$$

and its center of mass is:

$$x_c = \frac{1}{2} * (r_{12} + r_{13} + r_{14})$$

Further refinements to the block can now simply be made by successively refining tetrahedra into 8 smaller tetrahedra, using the mid-way points of the vectors r_{12}, r_{13} and r_{14} (see figure 1). Each successive refinement thus yields a better approximation $GF_{ij,L}$ where L denotes the level of refinement. The number of cells the original block is broken down to increases by factor 8 per refinement level, i.e.

$$N_V = N_1 * 8^{L-1}$$

If the original block was a hexahedron with $N_1=12$, at level 4 one would have 6144 individual sub-volumes.

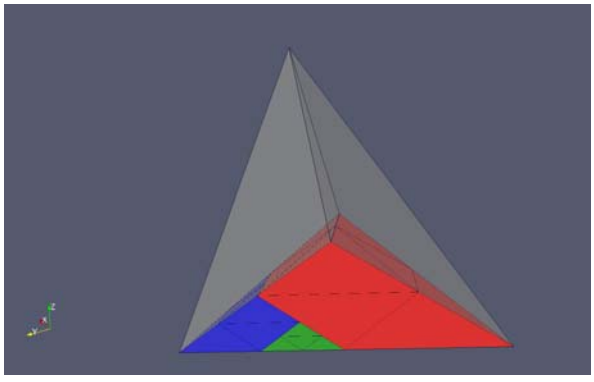

Figure 1: Tetrahedral refinement scheme. The original block (grey) is divided into cells using its center of mass as common point. After that tetrahedral refinements are applied, *L1* (red), *L2* (blue), *L3* (green).

At this stage one needs to make a choice about memory management. If a model with 50,000 blocks is used and is refined to level 4 and 4 vectors with 3 components and 4 byte of memory are required for storage, then the memory requirement is ~14GB, which is beyond what most modern desktop computers have installed. However, once a single block gravity factor is calculated, the memory for its calculation can be released and only the gravity factor saved for later usage. But if one wants to investigate the structure of the refinement to check for the performance of this algorithm all data points need to be kept, thus limiting the size of the model and the maximum possible refinement level L_{max}.

Another consideration is that not all regions of a single block need to be refined to the same level. For regions far away from the gravity station location, the point mass approximation will already be good, and refinements will only slightly improve the gravity factor. Hence it becomes useful to define a convergence criterion beyond which no further refinements are required.

The convergence criterion is specified via setting a global parameter *EPV* (error per volume). A block will be further refined if it is believed that by not refining it the difference to the actual convergence value is larger than *EPV* times the block volume. A discussion on how to best define the *EPV* will be given further down.

Once a refinement level of 2 or further has been created, the local rate of convergence can be calculated:

Consider a cell in level *L*. Let *Grav* denote the gravity factor calculated by the cell. This cell is part of a larger cell from level *L-1*, which we will denote as the "parent" cell. Let *ParGrav* denote the gravity factor calculated by the parent, weighted by the volume fraction factor between the cell and the parent volume. Also, the cell has been divided into smaller units, its "children" cells, which are from level *L+1*. Hence let *ChildGrav* be the sum of gravity factors calculated by its children cells.

The rate of convergence, c_{rate}, is then calculated using the differences between gravity factors:

$$c_{rate} = \left| \frac{ChildGrav - Grav}{Grav - ParGrav} \right|$$

If c_{rate} is larger or equal to 1 then the series has not converged yet. If $c_{rate} < 1$ then the truncation error is estimated using the geometric series:

$$\varepsilon = (Grav - ParGrav) * \frac{1}{1 - c_{rate}}$$

If ε is less than the *EPV* times the cell volume then convergence is assumed. Else the children cells will be called upon to refine to the next level.

2.3. Estimation of the EPV Parameter

Consider the total volume of the numerical reservoir model, *V*, and a typical distance *R* at which changes in the density of the fluid could be important. The typical total gravity factor is then in the order of $GF = V/R^2$.

Density changes occur over 2 orders of magnitude (10^2), i.e. from about 10kg/m^3 to 1000kg/m^3.

Changes in gravity can be typically determined to 5%.

The total error allowable is therefore $0.05 / 10^2 = 5 * 10^{-4}$ times the total gravity factor. Dividing by *V* yields the $EPV = 0.05 / 10^2 / R^2$.

This criterion may be a bit too strict though, and it needs to be determined if such a fine *EPV* will lead to too many refined cells – which could force the computer to run out of memory. A good strategy is to start the algorithm with the desired *EPV* but a low maximum level, followed by a run with a higher maximum level. If the gravity factors don't change significantly between these two runs higher refinements may not be needed.

3. COMPARISONS

3.1. Analytical Solution of a Homogenously Filled Sphere

The analytical solution for a homogeneously filled sphere shows linear increase with growing distance from the center to the surface. Outside the sphere, gravity falls with $1/R^2$.

For comparison with the algorithm shown here a discretized version of a sphere with 1.0m radius was created in VTK. Angles ψ and φ and the radial component were discretized in 20, 20 and 10 intervals, respectively. The discretization results in a representation of a sphere with slightly reduced volume, leading to an effective radius of 0.99m which was used in the analytical model. Note that the grid for the sphere already consist of 20*20*10=4000 single blocks, which itself is already very refined. We expect therefore quick convergence using the refinement scheme, even at low maximum refinement levels.

Total gravity factors for the sphere were calculated using *EPV*=0.1 and maximum levels 2 and 3 for stations located from 0.01m to 100m away from the center. Figure 2 shows the comparison with the analytical model. The agreement is excellent; the maximum residual encountered corresponds to an *EPV* of 0.015, showing that the above mentioned convergence criterion might indeed be too strict.

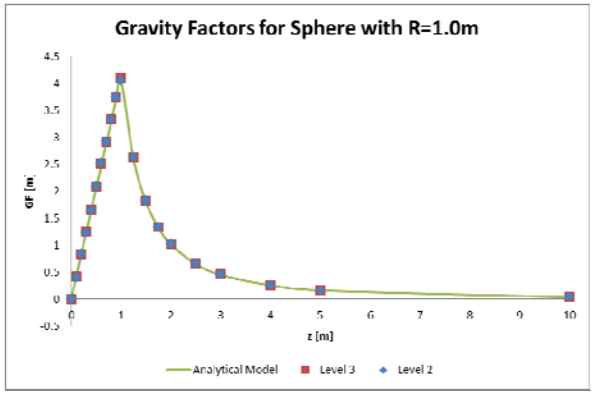

Figure 2: Comparison of the algorithm with the analytical solution of a sphere.

Note that the point mass approximation (not shown here) yields exactly the same solution for R>=1.0m, but diverges very rapidly for R<1.0m.

3.2. Homogeneously Filled Cube

Cubes or rectilinear blocks are a common feature in most geothermal reservoir simulations. Unfortunately, an

analytical model does not exist for cubes or rectilinear boxes of finite dimension for arbitrary orientations. However, analytical solutions exist for some special cases, like right-rectangular prisms (e.g. Nagy 1966). Standard practice by many modelers is to place artificial gravity stations at the center of the block's face on the surface and use the point mass approximation.

This practice makes direct station-by-station inter-comparison hard, since the data from the artificially created station needs to be interpolated to the actual station position, thus introducing another artifact. Further, not much thought has been given to the inaccuracies introduced by assuming the blocks to be point masses. Certainly, for blocks deep down in the reservoir this simplified treatment sounds reasonable, but it is not immediately clear how large the error is for blocks close to the surface.

For testing purposes, a cube of 1x1x1m dimensions was created with the z-axis perpendicular to one of its faces. The center of the cube was placed at (0,0,0). Gravity stations were investigated along the *z*-axis (face centered stations) and along an axis running from the center through one of the corner. Since the grid for the cube consists of only single block we expect a slower convergence than for the sphere example.

Gravity factors were calculated using *EPV*=0.1 and maximum levels refinement levels from 2 to 5.

Figure 3 shows the results for the face centered stations. At the surface, i.e. at 0.5m distance, the point mass model gives a gravity factor twice as large as the value calculated using the refinement scheme. While this difference at the surface is very large it vanishes quickly with increasing distance. At a distance of 1.5m the residual is only 1.3%.

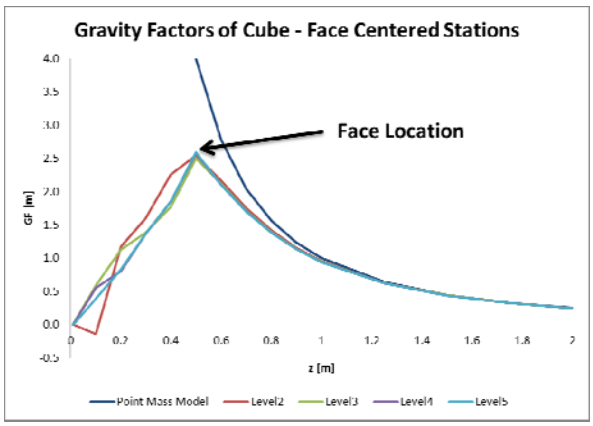

Figure 3: Gravity factors for a cube of 1x1x1m with gravity stations located along the *z*-axis.

Figure 4 shows the results for stations located along the axis through the corner of the block. At the surface, *z*=0.5m, the residual is about 20%, but at *z*=1.0m the residual is less than 1%. Convergence between the point mass model and the refined calculation is hence faster for corner centered stations than for face centered station.

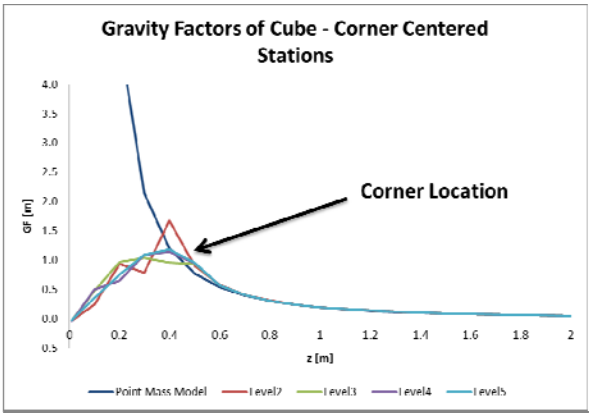

Figure 4: Gravity factors for the cube with stations located along an axis through the corner of the cube.

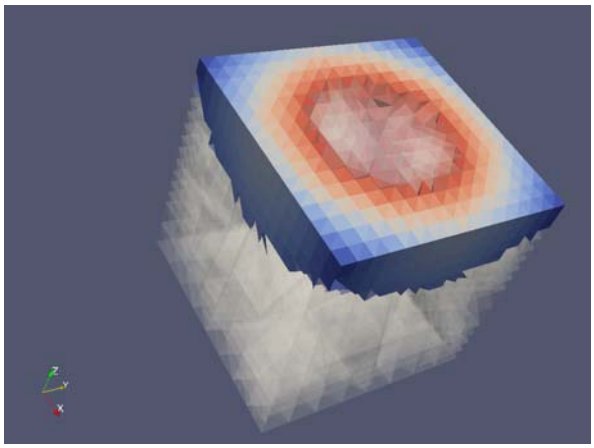

Figure 5: *z*-component of gravity for a station located at the face of the cube. Highlighted is a section bounded by two concentric rings.

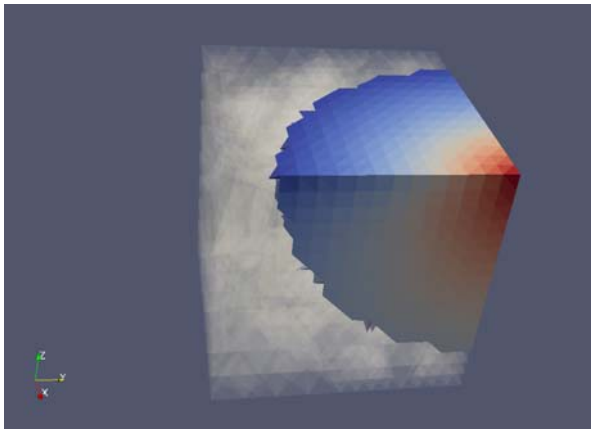

Figure 6: *z*-component of gravity for a station located at the corner of the cube. Note gravity falls off quicker radially than vertically due to the cosine factor.

To further verify the results for the cube a simple calculation was set up using the analytical method of Nagy (1966). Two gravity factors were calculated using the corner-centered points (which are easier to set up using the Nagy scheme) at *z*=1.0 and *z*=1.5m. The analytical solution differed from the numerical method by 0.9% and 0.2%, respectively, using the level 5 refinement scheme. Therefore, for any practical purposes, the numerical method described here can substitute the analytical method which

requires complex coordinate transformations and works only on rectilinear block shapes.

4. INTERACTION WITH TOUGH2

To test the functionality and correctness of the whole routine, a simple TOUGH2 test model was created that included just enough complexity to make the manual calculation of gravity changes possible for inter-comparison.

The model consists of a cube 1000x1000x1000m dimension and includes 1000 equal blocks. It is set up using a dual porosity model to ensure the calculation routine can handle multiple MINC layers. 3 gravity stations were placed on top of the model.

Two blocks, spatially far apart and deep down in the reservoir, were chosen to represent a source and a sink. In a first approach it was tried to use an ordinary TOUGH2 simulation to extract/inject fluid from/to the source and sink blocks. However, manually calculating the associated changes in gravity proved too hard. If permeabilities in the model were chosen too low, TOUGH2 would crash. If they were chosen too high, TOUGH2 would spread the changes over several blocks. The resulting gravity changes were certainly of the right magnitude but were not suitable for direct comparison with the manual calculations.

To overcome this problem the change in fluid mixture density was isolated from the TOUGH2 simulation. The model was run into steady state and two sets of identical thermodynamic variables were written to the output file.

The output file was then manually edited by changing liquid and gas saturations in the source/sink blocks in the second printout. In this manner one can exactly control the change of mass in particular blocks without having to deal with the TOUGH2 simulation complexity.

The complete algorithm – creation of a gravity matrix, VTK file series and subsequent gravity analysis – was run over this manipulated TOUGH2 output file and the changes in gravity were compared to a manual calculation. The maximum residuals encountered were 0.03%, which demonstrates that the algorithm and the associated workflow perform very well.

5. DISCUSSION

The refined integration scheme and the associated tools described here facilitate the calculation of changes in micro-gravity in geothermal reservoir modeling situations. Changes can be calculated directly at station locations. This makes direct inter-comparison of data easy; potentially the data could be used in inverse modeling for reservoir parameter estimation using iTOUGH or PEST.

The convergence criterion shown appears to be valid but too stringent, i.e. the errors calculated appear to be smaller than the error criterion given. More work could be done to find a better convergence criterion in order to improve the algorithm; however in practical terms the criterion suffices.

For cubic blocks the refined integration has shown that the difference to the point mass model vanishes for distances about 1.5 times the block height. This signifies that the commonly used point mass calculation is valid only if the dimensions of the blocks in the top layers are adequately chosen and if changes in mixture density in the surface layers are insignificant over the modeled period.

REFERENCES

Atkinson, P.G. & Pedersen, J.R.: *Using Precision Gravity Data in Geothermal Reservoir Engineering Modeling Studies,* Proceedings, Thirteenth Workshop on Geothermal Reservoir Engineering, Stanford, 1988, SGP-TR-113

Kitware Inc.: *The VTK User's Guide,* Kitware Inc, ISBN 978-1-930934-23-8

Nagy, D.: *The Gravitational Attraction of a Right Rectangular Prism,* Geophysics, Vol. 31, No.2 (April, 1966), pp. 362-371

Newton, I.: *Philosophiae Naturalis Principia Mathematica,* London 1686

Plouff, D.: *Gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections,* Geophysics, 41, pp. 727-741