



UNIVERSITY OF THE WITWATERSRAND

MSc Dissertation

Applications of chaos and fractals to geophysical inversion problems

Author:

Brandon Dias
676053

Supervisors:

Prof. Gordon. R. J. Cooper
Prof. Raymond. J. Durrheim

A dissertation submitted in fulfillment of the requirements
for the degree of Master of Science at the
University of the Witwatersrand
School of Geosciences

October 25, 2019

DECLARATION

I declare that this dissertation is my own work. It is being submitted for the Degree of Master of Sciences (Geophysics) at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.



(Signature of candidate)

22nd day of October 2019 at the University of the Witwatersrand

ABSTRACT

In order to gain clear insight into the structure and composition of the Earth and its subsurface, geophysicists and geologists take readings of geophysics responses. Gravitational responses are among the most often recorded datasets and among the most used means of model analysis is the least-squares inversion process. Here; this is demonstrated using synthetic gravitational responses from buried sphere and cylinder models of different density contrasts to the background. The least-squares inversion attempts to utilize initial user chosen parameters to create models which correlate strongly with observed data and thus create potential geological models of the Earth's subsurface or submerged geological structures. The inversion processes, misfit hyper-functions, basins of attraction and fractal dimensions are studied as functions of initial model parameters. We observe that the fractal dimension and basin of attraction vary with respect to observed model depths and positions. Additionally the fractal dimension is inversely proportional to the degree of damping of the least-squares inversion process.

A potential problem with the least-squares inversion method is the possibility for solutions to tend to local minima. These may better fit an observed model response but may not provide a geologically viable option. The equation for the least-squares inversion, as applied to observed models, is altered to induce bifurcations and chaos within solutions. Chaos can be used to move the inversion process out of local minima on the misfit surface, potentially improving the fit of the model response to the data. Bifurcation diagrams are established and the periodicity analysed.

I would like to thank my supervisors for their constant support over the last three years; both academically and financially. Thanks to my supervisors and the School of Geosciences; I have been able to present my research at conferences in Cape Town, South Africa and Anaheim, United States of America.

Special thanks are given to Dr Steven Zhang and Dr Stephanie Enslin for their assistance in this dissertation and my past conference presentations. Their help in reading and providing comments has proven invaluable.

Most of all I would like to thank my friends, family and parents for their endless support, guidance and love. Without them, none of this would have been possible.

TABLE OF CONTENT	PAGE
COVER PAGE	i
DECLARATION	ii
ABSTRACT	iii
TABLE OF CONTENT	v
TABLE OF FIGURES	viii
TABLE OF TABLES	xxxiv
CHAPTER ONE - INTRODUCTION	1
1.1 General introduction	1
1.2 Aims and objectives	3
1.2.1 Aims	3
1.2.2 Objectives	3
1.3 Literature Review	4
1.3.1 Fractals	4
1.3.2 Fractal dimensions	6
1.3.3 Fractals in Euclidean space and the box-counting method	11
1.3.4 Chaos theory	12
1.3.5 The logistic map and bifurcation diagram	14
1.3.6 Inverse theory and geophysical forward modelling	20
1.3.7 Similar methods used to solve non-linear systems	23
CHAPTER TWO – METHODS AND MATERIALS	27
2.1 Inversion methodology	27
2.2 Inversion operation and the basin of attraction	30
2.3 Analysing the basin of attraction	30
CHAPTER THREE - RESULTS	33
3.1 The inversion process	33
3.2 The basin of attraction	37
3.3 Analysis of the basin of attraction and fractal dimensions	43
3.3.1 Varied depths of a single cylinder source using the Moore-Penrose inversion process	43
3.3.2 Varied depths of a single cylinder source using the Ridge-regression inversion process	44
3.3.3 Damping coefficient's effect on the basin of attraction and convergence conditions	45

3.4 The scaling factor's effect on the inversion process	50
3.4.1 The scaling factor's role in the rate of convergence	54
3.4.2 The simultaneous effect of k and C on the amount of iterations to convergence	60
3.4.3 Quantifying the separation of inversion trajectories due to different scaling factors	63
3.5 The bifurcation diagram for the geophysical inversion problem	66
3.6 The Feigenbaum constants	67
3.7 Analysis of basins of attraction for models containing two cylindrical sources	69
3.7.1 Moore-Penrose inversion process for models at varied source depths	70
3.7.2 Ridge-regression inversion process for models at varied source depths	74
3.7.3 Moore-Penrose inversion process for models at varied source separations	77
3.7.4 Ridge-regression inversion process for models at varied source separations	81
3.7.5 Varied resolutions of the basin of attraction using the Ridge-regression inversion process	83
3.7.6 Damping coefficient's effect on the fractal dimension	90
3.7.7 The scaling factors' effect on the basin of attraction and the fractal dimension	95
3.8 Avoiding local minima	97
3.9 Three dimensional parameter basins of attraction	100
CHAPTER FOUR – APPLICATION TO GRAVITY DATA FROM SOUTH AFRICA	114
4.1 Basin of attraction	115
4.2 Avoiding local minima	117
CHAPTER FIVE – DISCUSSION	122
5.1 The Moore-Penrose inversion and the Ridge-regression method	122
5.2 Analysing the basin of attraction as a result of varying observed model parameters	125
5.2.1 Varying model depths	125
5.2.2 Varying model separations	125
5.2.3 Varying the resolution of the solution space	126
5.2.4 Analysis of time taken to construct basins of attraction	127
5.3 The damping factor's affect on the fractal dimension	128
5.4 The scaling factor's effect on the inversion process	129
5.5 Bifurcation Diagram and the Feigenbaum constants	130
5.6 Determining the scaling factor C	131
5.7 Avoiding local minima	132
5.8 The Lyapunov Exponent	132
5.9 The affect of k and C collaboratively on the inversion process	134

5.10 Applications to gravity data from South Africa	136
CHAPTER SIX – CONCLUSIONS AND RECCOMENDATIONS	138
REFERENCES	140
Image references	142
APPENDIX	145
1 - LogisticMapGenerator	145
2 – InversionStepsPlusModel	146
3 - MisfitJumNew	147
4 – MakeBMap	151
5 – axazPlots	153
6 - BasinsOfAttractionsAndDimensionsThreeImagesK	154
7 – BasinOfAttractionKIterationsTracked	157
8 – FracZoom	159
9 – MisfitFunctionPlusModel	162
10 – BasinOfAttractionKIterationsTrackedTwoBodies	163
11 – VariedXorZkTwoBodies	164
12 – ResponsePlusModel	168
13 – InversionsWithScalingFactorModelIncluded	168
14 – MisfitVsC	170
15 – CVSILoop	173
16 – CsLoop	174
17 – CvsKvsI2	176
18 - LapyunovC	177
19 – LyExp	180
20 – ThreeParametersFRAC	182
21 – Pseudo3D	186
22 – mapREAL	189
23 – RealDataMisfitJumpFRAC	190
24 – RealDataMisfitJump	194
25 – MultiBasinKorC	201
26 – CompareK	203

TABLE OF FIGURES

Figure 1: Koch snowflake; as constructed through the protrusion of triangles along the body's surface (Koch snowflake, 2016).	4
Figure 2: Image of a lightning strike and a snowflake as shown on the left and right, respectively (Lightning strike, n d) (Snowflake, n d).	5
Figure 3: Completed Sierpinski triangle (Sierpinski triangle, n d).	6
Figure 4: The Mandelbrot set (Sprott, 1997). The pixels of colour represent starting c (complex numbers) which do not diverge to infinity using the Mandelbrot expression.	7
Figure 5: Establishing self-similarity in the Mandelbrot set. A basin of convergence is identified in a , and is enlarged to form the image in b . This process is repeated to form the images in c and d , respectively. At each point the same basin is observed and it becomes evident that the series of patterns repeat themselves at different scales, thus exhibiting self-similarity. (Mandelbrot, n d)	8
Figure 6: Illustration of progressing dimensions. N , r and D represents the amount of originals, scalar multiple and the dimension, respectively. (Progressing dimensions, n d).	9
Figure 7: An iterative diagram showing the construction of the Sierpinski triangle (Sierpinski triangle constructions, n d).	10
Figure 8: Visualisation of the box-counting method. A theoretical fractal image (in white) is placed on a grid and the box counting method is applied to it. Blue boxes indicate the original smallest box size and the yellow boxes show a larger box sizes. As the box-counting method is employed, the smaller boxes (blue) move through the image (as shown by the arrow) until the entire image has been sampled. The box size increases (yellow) and the process repeats until the boxes are as large as they can be.	11
Figure 9: The bifurcation of a river as it cuts through the landscape (Bifurcating river, n d).	13
Figure 10: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). This logistic map plots potential population magnitudes as a result of growth rates. (See appendix 1 – LogisticMapGenerator)	15

Figure 11: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The growth rate is only varied from 3.5 to 4. (See appendix 1 – LogisticMapGenerator) 16

Figure 12: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The logistic map here records the growth rates, A , as a function of the fixed stable points needed for calculating the first Feigenbaum constant (see equations (5)). The growth rate is only varied from 2.9 to 3.7. The A values for the red, yellow, black and pick line are 2.98, 3.45, 3.54 and 3.56, respectively. (See appendix 1 – LogisticMapGenerator) 17

Figure 13: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The logistic map here records the growth rates, A , as a function of the fixed stable points needed for calculating the second Feigenbaum constant (see equation (6)). The growth rate is only varied from 2.9 to 3.7. The widths of the yellow, black and pick line are 0.4081, 0.1588 and 0.0.0273, respectively. (See appendix 1 – LogisticMapGenerator) 19

Figure 14: Representations of Misfit surfaces for parameters linear under inversion (a) and non-linear under inversion (b). 21

Figure 15: An illustration of how the same gravitational profile can be generated from an infinite amount of models. Shallow, large and low density bodies will generate the same gravitational response as small, deep and very dense bodies, respectively. 22

Figure 16: An example of the Newton's fractal. A complex x -plane with real and imaginary components in the range of -2 to 2 is established. Black points in the solution space indicate the set of points from which Newton's method converges to the root $x = 1$ for the equation $x^3 - 1 = 0$ (Press et al, 2007, pg 462). 25

Figure 17: Gravitational responses for an associated synthetic model. Two responses, in a , are shown as a result of two separate buried cylinder models, in b , using the expression for the gravitational response due to a buried cylinder (see equation (9)). The observed model (in red) is set at 45 m along the range and buried at a depth of 15 m. The initial calculated model (in green) is set at 0 m along the range and buried at a depth of 25 m. (See appendix 12 – ResponsePlusModel) 28

Figure 18: A misfit function, from equation (10), for the geological model established in Figure 17. (See appendix 9 – MisfitFunctionPlusModel) 29

Figure 19: Plots of the x and z derivative are shown in blue and green, respectively. The x and z derivative arrays are the result of equation (11) and equation (12), respectively. (See appendix 8 – axazPlots) 30

Figure 20: Gravitational responses for associated synthetic models. Observed model responses for buried cylinder models at varied depths of 5 m, 25 m and 45 m in a , b and c , respectively. Each response is the result of a buried cylinder model, as established in equation (9). (See appendix 12 - ResponsePlusModel) 31

Figure 21: Gravitational responses for associated synthetic models. Observed model responses of buried cylinder models at varied separations. The separations between the centre of the two buried bodies in the model is equal to 50 m, 30 m and 10 m in a , b and c , respectively. Each response is the result of a buried cylinder model, as established in equation (9). (See appendix 12 - ResponsePlusModel) 32

Figure 22: Observation of the calculated and observed models through the Moore-Penrose inversion process as defined by equation (7). The observed cylindrical model is at 0 m along the range and is buried at 15 m. The calculated cylindrical model starts at 15 m along the range and is buried at 32.5 m. The models are tracked through the inversion iterations; the blue model is the observed model and the yellow model is the calculated model. (See appendix 2 – InversionStepsPlusModel) 34

Figure 23: Observation of the calculated and observed responses during the inversion process at each iteration. The observed cylindrical model is set at 0 m along the range and buried at a depth of 15 m. The calculated cylindrical model is set at 15 m along the range and buried at a depth of 32.5 m. Each response is the result of a buried cylinder model, as established in equation (9), with different parameter values. (See appendix 2 – InversionStepsPlusModel) 35

Figure 24: The misfit, as calculated from equation (10), as a function of iteration for the models defined in Figure 22 and Figure 23. (See appendix 2 – InversionStepsPlusModel) 36

Figure 25: Observed model response, seen in a as a result of the model set up in b . The observed cylindrical model is set at 0 m along the range and buried at a depth of 15 m. (See appendix 12 - ResponsePlusModel) 37

Figure 26: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Moore-Penrose inversion, equation (7), is used on each initial parameter value set in the solution space. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. (See appendix 8 – FracZoom) 38

Figure 27: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Ridge-regression inversion, equation (8), is used on each initial parameter value set in the solution space. Yellow indicates initial parameters which converged under inversion and blue indicates divergence from observed model. (See appendix 8 – FracZoom) 39

Figure 28: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. Pattern colourized by the amount of iterations to converge. The dark blue background are initial models that did not converge to the observed model. The colorbar indicates the amount of iterations to converge. (See appendix 7 – BasinOfAttractionKIterationsTracked) 40

Figure 29: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. Yellow indicates initial models which converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. The white box highlights the area which will be investigated and enlarged to investigate self-similarity. (See appendix 8 – FracZoom) 41

Figure 30: Basin of attraction using an observed model set at 0 m along the range and buried at 15 m. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7), is used on each initial model in the solution space. The white box highlights the area which will be enlarged. a is the result of enlarging the area identified in Figure 29. Progressing further; b and c are the result of enlarging the areas identified in a and b , respectively. At each enlargement the same pattern is continuously observed. This is a result of “self-similarity” as defined in section 1.3.2. (See appendix 8 – FracZoom) 42

Figure 31: An example of how the box-counting can be applied to the boundary of the basin of attraction. Basin of attraction using an observed model set at 0 m along the range and buried at 15 m. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7), is used on each initial model in the solution space. The box size is decreased from a to c and at each instance the amount of boxes is compared to the size of the respective box to investigate the dimension of the basin. (See appendix 8 – FracZoom) 43

Figure 32: Basins of attraction and dimension calculations for a single cylindrical body centred at 0 m and buried at various depths. a , b and c represents depths of 5 m, 10 m and 15 m, respectively. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. (See appendix 6 – BasinsOfAttractionsAndDimensionsThreeImagesK) 44

Figure 33: Basins of attraction and dimension calculations for a single cylindrical body buried at various depths. a , b and c represent depths of 5 m, 10 m and 15 m, respectively. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Ridge-regression inversion, equation (7) is used on each initial model in the solution space. (See appendix 6 – BasinsOfAttractionsAndDimensionsThreeImagesK) 45

Figure 34: Observed model response, seen in a as a result of the system set up in b . The observed cylindrical model is set at 0 m along the range and buried at a depth of 10 m. (See appendix 12 - ResponsePlusModel) 46

Figure 35: Basin of attraction using a constant damping coefficient of 0 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 47

Figure 36: Basin of attraction using a constant damping coefficient of 2 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar

indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 47

Figure 37: Basin of attraction using a constant damping coefficient of 8 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 48

Figure 38: Basin of attraction using a constant damping coefficient of 10 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 48

Figure 39: Basin of attraction using a constant damping coefficient of 12 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 49

Figure 40: Basin of attraction using a constant damping coefficient of 18 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 49

Figure 41: Basin of attraction using a constant damping coefficient of 20 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates

parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked) 50

Figure 42: Gravitational responses, in a , for buried cylinder models, in b , as defined in equation (9). The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at -5 m along the range and is buried at 10 m. (See appendix 12 – ResponsePlusModel) 51

Figure 43: Plots of the x position as a function of iterations under different scaling factors in the modified Moore-Penrose inversion; equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. As the scaling factor increases, the inversion system succumbs to eventual bifurcations at $C = 2$. In b the the system is unperturbed and the inversion is carried out as normal. From c to d , as the magnitude of the scaling constant increases and the system grows more perturbed and results in periodicity and a bifurcation. (See appendix 13 – InversionsWithScalingFactorModelIncluded) 52

Figure 44: Plots of the x position as a function of iterations under different scaling factors in the modified Moore-Penrose inversion; equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. As the scaling factor increases, the inversion system creates additional bifurcations which can masquerade as apparent chaos. (See appendix 13 – InversionsWithScalingFactorModelIncluded) 53

Figure 45: Plot of the x parameter position as a function of iterations using a scaling factor value of 9 in the modified Moore-Penrose inversion; equation (13). Plots of the x position as a function of iterations under different scaling factors in equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. The x position appears to behave chaotically. (See appendix 13 – InversionsWithScalingFactorModelIncluded) 54

Figure 46: Plots of the original misfit, over a range of different initial parameter values, as a function of the respective scaling factor which would result in the fewest amounts of iterations to converge. The modified Moore-Penrose inversion process, using equation (13), is set up over an array of different initial model positions and depths and for each set; the scaling factor which would result in the fewest amounts of iterations to converge is

recorded. The observed model is at 10 m along the range and is buried at 10 m. Convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. (See appendix 14 – MisfitVsC) 55

Figure 47: Investigation into the affect the scaling factor has on the amount of iterations to converge. The same observed model is used in all cases but in each case the initial model parameters are changed. The observed model is at 10 m along the range and is buried at 10 m. The position and depth of the initial model is set as 15 m, 25 m and 35 m in a , b and c , respectively. The amount of iterations is capped at 25. Convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. All inversion processes run using the modified Moore-Penrose inversion in equation (13). (See appendix 15 – CVSILoop) 56

Figure 48: Gravitational responses for associated models. Calculated and observed models in b and associated responses in a . The observed model is at 0 m along the range and is buried at 20 m. The initial model is at 10 m along the range and is buried at 20 m. The blue model is the observed model and the yellow model is the calculated model. (See appendix 12 – ResponsePlusModel) 58

Figure 49: Plot of the calculated scaling factor as a function of the initial input scaling factors. Equation (17) is run after the third iteration to solve for the calculated C value. (See appendix 16 – CsLoop) 59

Figure 50: Plot of the calculated scaling factor as a function of the initial input scaling factors. Equation (17) is run after the two hundredth iteration to solve for the calculated C value (See appendix 16 – CsLoop) 59

Figure 51: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 20 m. (See appendix 17 – CvsKvsI2) 61

Figure 52: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of

iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 40 m. (See appendix 17 – CvsKvsI2) 61

Figure 53: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 60 m. (See appendix 17 – CvsKvsI2) 62

Figure 54: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 80 m. (See appendix 17 – CvsKvsI2) 62

Figure 55: Plots of the x position as a function of the iteration for two separate inversion trajectories. The two inversion trajectories are indistinguishable from one another. The two points are originally separated by 0.001 m and are separated by 0 m after 10 iterations. $C = 2$ and the inversion is carried out by the modified Moore-Penrose inversion process in equation (13). The distance between the two inversion trajectories decrease as the iterations progress. (See appendix 18 – LapyunovC) 64

Figure 56: Plot of the log of the separation between two inversion trajectories as a function of the iterations. The gradient of the line of best fit is used to solve for the Lyapunov exponent. Here the Lyapunov exponent is negative as the two bodies drift closer toward one another instead of diverging from one another. (See appendix 18 – LapyunovC) 65

Figure 57: Plots of the x position as a function of the iteration for separate inversion trajectories. The red and blue trajectories show how the distance between the two points is increased as the iterations progress through the inversion. The two bodies are originally separated by 0.001 m and $C = 3, 4$ and 5 in a, b and c , respectively. The modified Moore-Penrose inversion processes, as set up in equation (13), are run on all initial models. (See appendix 18 – LapyunovC) 66

Figure 58: A bifurcation diagram created by recording the fixed points observed after parameter inversion at different scaling factors. The bifurcation diagram is vertically symmetric about the observed x parameter value. The initial model location is at -10 m along the range and the observed model location along the range is at 10 m. The scaling factor is varied from 0.5 to 7. All scaling factors are kept constant in the modified Moore-Penrose inversion, see equation (13). (See appendix 4 – MakeBMap) 67

Figure 59: A bifurcation diagram created by recording the fixed points observed after parameter inversion at different scaling factors. The bifurcation diagram is vertically symmetric about the observed x parameter value. The initial model location is at -10 m along the range and the observed model location along the range is at 10 m. The scaling factor is varied from 0 to 7. All scaling factors are kept constant in the modified Moore-Penrose inversion, see equation (13). The values used to calculate the first (equation (5)) and second (equation (6)) Feigenbaum constant are indicated by the black lines. A similar methodology was applied in section 1.3.5. (See appendix 4 – MakeBMap) 68

Figure 60: Two anomalies are shown as a result of two separate bodies and generated using the cylinder models. The two bodies in the observed model are separated by 90 m about the centre of the profile and are both buried at 15 m. The initial model is at 0 m along the range and is buried at a depth of 25 m. (See appendix 12 - ResponsePlusModel) 69

Figure 61: Responses due to a model containing two cylindrical sources both buried at different depths. Gravitational responses are generated by the buried cylinder models, as expressed in equation (9). (See appendix 12 – ResponsePlusModel) 70

Figure 62: a is the basin of attraction for a model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 61). b is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 71

Figure 63: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 7.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 72

Figure 64 *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 10 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 72

Figure 65: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 12.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 73

Figure 66: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 15 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 73

Figure 67: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 74

Figure 68: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 7.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 75

Figure 69: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 10 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 75

Figure 70: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 12.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 76

Figure 71: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 15 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 76

Figure 72: Responses due to a model containing two cylindrical sources both buried at different separations. Gravitational responses are generated by the buried cylinder models, as expressed in equation (9). (See appendix 12 – ResponsePlusModel) 78

Figure 73: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 30 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 79

Figure 74: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 60 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 79

Figure 75: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the

observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 80

Figure 76: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 120 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 80

Figure 77: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 30 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 81

Figure 78: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 60 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the

observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 82

Figure 79: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 82

Figure 80: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 120 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies) 83

Figure 81: Observed response in *a* as a result of the model in *b*. Model is created with two cylindrical bodies, buried at 5 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel) 84

Figure 82 *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A standard solution space is used where the parameters change by 1 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue

background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension takes 9.2 seconds. (See appendix 11 – VariedXorZkTwoBodies)

85

Figure 83: *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A solution space five times as dense is used where the parameters change by 0.2 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension takes 3 minutes 6.3 seconds. (See appendix 11 – VariedXorZkTwoBodies)

86

Figure 84: *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A solution space ten times as dense is used where the parameters change by 0.1 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension took 8 minutes 2.1 seconds. (See appendix 11 – VariedXorZkTwoBodies)

87

Figure 85: Observed response in a as a result of the model in b . Model is created with two cylindrical bodies, buried at 10 m and separated by 100 m about the centre of the profile. (See appendix12 – ResponsePlusModel) 88

Figure 86: Basin of attraction for a model of two buried cylindrical bodies. The observed model is defined in Figure 85. Standard resolution used in the solution space where every parameter value increases by 1 m. Colorbar indicates the amount of iterations to converge to either body. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image takes 8.9 seconds. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 89

Figure 87: Basin of attraction for a model of two buried cylindrical bodies. The observed model is defined in Figure 85. A resolution one hundred times greater is used. Colorbar indicates the amount of iterations to converge to either body. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image took 15 hours 59 minutes and 35.52 seconds. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 90

Figure 88: Observed response in a as a result of the model in b . Model is created with two cylindrical bodies, buried at 10 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel) 91

Figure 89: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 0 in, equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 92

Figure 90: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 2 in, equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial

model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 92

Figure 91: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 8, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 93

Figure 92: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 10, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 93

Figure 93: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 12, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 94

Figure 94: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 18, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 94

Figure 95: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 20, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies) 95

Figure 96: The basin of attraction, as created using the modified Moore-Penrose inversion in equation (13) and various scaling constants C . The scaling constant is varied from 0.00 to 0.75. All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations to converge to either the left or right observed body. The black background indicates initial models which did not meet convergence conditions. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 25 – MultiBasinKorC) 96

Figure 97: The basin of attraction, as created using the modified Moore-Penrose inversion in equation (13) and various scaling constants C . The scaling constant is varied from 0.90 to 1.65. All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations to converge to either the left or right observed body. The black background indicates initial models which did not meet convergence conditions. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 25 – MultiBasinKorC) 97

Figure 98: *a* shows the observed gravitational response due to the associated model in *b*. In *b*, the red model represents the left response and the blue model represents the right response. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. (See appendix 12 – ResponsePlusModel) 98

Figure 99: *a* is the misfit function as a result of the initial model parameters and *b* is a plot of the iterations versus the x position. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. The initial model is acted upon by the modified Moore-Penrose inversion, in equation (13). The initial model starts at 40 m along the range and is buried at 10 m. In *b*, the x position initially tends toward the local minimum and oscillates chaotically about the minimum, which

behaves like a strange attractor. Eventually, as the scaling factor is continuously increased, the x position tends toward the global minimum and oscillates about the minimum again. (See appendix 3 – MisfitJumpNew) 99

Figure 100: Relationship between the misfit, x coordinate position and the scaling factor. *a* shows the misfit as a function of the iterations within the inversion process. *b* shows the x position as a function of the amount of iterations. *c* is the scaling factor as a function of the iterations. The black and red boxes indicate the parameter values associated with the corresponding local and global minima, respectively. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. The initial model is acted upon by the modified Moore-Penrose inversion in equation (13). The initial model starts at 40 m along the range and is buried at 10 m. (See appendix 3 – MisfitJumpNew) 100

Figure 101: A synthetic observed gravitational surface response as generated using equation (23). The response is the result of a buried sphere centred at 0 along the x and y range and buried at 25 m. (See appendix 20 – ThreeParametersFRAC) 102

Figure 102: Model for two buried spherical sources. The observed and calculated models are represented by the blue and red plots, respectively. The observed model is placed at 0 m along the x range, 0 m along the y range and is buried at 25 m. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at 20 m. (See appendix 20 – ThreeParametersFRAC) 103

Figure 103: Observed and calculated gravitational responses of a buried sphere system. *B* is the calculated forward model and *A* is the observed response. The observed model is placed at 0 m along the x range, 0 m along the y range and is buried at 25 m. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at 20 m. (See appendix 20 – ThreeParametersFRAC) 104

Figure 104: The red, blue and green trajectories track the parameter values of the x , y and z position, respectively as they are acted upon by the Moore-Penrose inversion, see equation (7). The observed model is placed at 0 m along the x range, 0 m along the y range and is buried at 25 m. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at 20 m. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. As the trajectories never stray

further than 60 m away from 0 m, only that range is shown to highlight trajectories. (See appendix 20 – ThreeParametersFRAC) 105

Figure 105: Three-dimensional basin of convergence; with the Moore-Penrose inversion process, in equation (7), run over a range of x , y and z positions. The observed model is as established in Figure 103. This solution space is set up in increments of 5 m as a proof of concept. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 20 – ThreeParametersFRAC) 106

Figure 106: Three-dimensional basin of attraction with the Moore-Penrose inversion process, as set out in equation (7) run over a range of x , y and z parameter values. The observed model is as established in Figure 103. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 20 – ThreeParametersFRAC) 107

Figure 107: Basin of attraction in the solution space. The observed model is as established in Figure 103. Blue and yellow represent parameter values which met divergent and convergent conditions, respectively. At each initial model, the Moore-Penrose inversion process, in equation (7), is run to test convergence conditions. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D) 108

Figure 108: Observed gravitational response due to a buried sphere model. The observed model is placed at 125 m along the x range, 125 m along the y range and is buried at 25 m. (See appendix 20 – ThreeParametersFRAC) 109

Figure 109: Basin of attraction in solution space with $z = 56$ m. The observed model is as established in Figure 108. Blue and yellow represent parameter sets which met divergent and convergent conditions, respectively. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D) 110

Figure 110: A three-dimensional basin of convergence model created by stacking three two-dimensional basins of attraction. The observed model is as established in Figure. The three two-dimensional convergence patterns are created from three processes where the x and y parameters are varied, but the depth is fixed at 54 m, 55 m and 56 m for the slices.

Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D) 111

Figure 111: Top view of pseudo three-dimensional basin of attraction model. The observed model is as established in Figure. The ring structure around the centre is now more visible. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D) 112

Figure 112: Pseudo three-dimensional basin of attraction model. The observed model is as established in Figure. The ring structure around the centre is again visible. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D) 113

Figure 113: Trompsburg as situated in South Africa at 30.032 S and 25.779 E. (Google Earth, 2017) 114

Figure 114: The relative gravitational anomaly over the Trompsburg Complex. The attached colorbar is used to gauge the relative gravitational intensity in mGal. The red and black lines show sample lines run at x ranges of 163 km and 183 km, respectively. A indicates a potential central spherical response and B indicates a potential linear response running east to west. (See appendix 22 – mapREAL) 115

Figure 115: The relative gravity anomaly extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. A and B correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump) 116

Figure 116: a illustrates the extracted profile in red and the forward models for the east-west feature and the central peak in blue and green, respectively. b is the associated basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the northern body and convergence to the southern body, respectively. Convergence conditions are met when the final model's location and depth lie within 5 m of the observed model. The southern forward, B , model is at 38 km along the range and is buried at 24km. The northern sphere forward model, A , is at 116 km along the range, is buried at 24 km and is 2.3 times as dense as the cylinder model. (See appendix 23 – RealDataMisfitJumpFRAC) 117

Figure 117: The relative gravity anomaly extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. *A* and *B* correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump) 118

Figure 118: *a* illustrates the extracted profile in red and the forward models for the east-west feature and the central peak in blue and green, respectively. *b* is the associated basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the northern body and convergence to the southern body, respectively. Convergence conditions are met when the final model's location and depth lie within 5 m of the observed model. The southern forward model, *B*, is at 35 km along the range and is buried at 15km. The sphere forward model, *A*, is at 126 km along the range, is buried at 15 km and is 2.3 times as dense as the cylinder model. (See appendix 23 – RealDataMisfitJumpFRAC) 119

Figure 119: The forward model parameters are set with *x* and *z* at 35 km and 15 km, and 125 km and 15 km for the southern and northern forward model, respectively. *a* shows the misfit function as a result of the initial model parameters. *b* is a plot of the *x* position as a function of the iteration. The initial model is at 60 km along the range and is buried at 15 km. The plot shows that initially, the position tends toward the local minimum and that the increasing scaling factor causes it to oscillate chaotically, which eventually causes the *x* position to tend towards the global minimum. (See appendix 24 – RealDataMisfitJump) 120

Figure 120: With the same model parameters and values as in Figure 119 a correlation is established within *a* and *b*. *a* shows the *x* position as a function of the amount of iterations. *b* is a plot of the misfit as a function of the number of iterations within the inversion process. The black boxes indicate the parameter values associated with the corresponding global minimum. (See appendix 24 – RealDataMisfitJump) 121

Figure 121: Gravitational responses, in *a*, for buried cylinder models, in *b*, as defined in equation (9). The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at 10 m along the range and is buried at 20 m. (See appendix 12 – ResponsePlusModel) 123

Figure 122: Plots of the depth of the initial model at varied damping constants (*k*), when acted upon by the Ridge-regression inversion process in equation (8). The observed and initial

models are defined in Figure 121. The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at 10 m along the range and is buried at 20 m. (See appendix 26 – CompareK) 124

Figure 123: The basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively. Yellow and green represent convergence to the left and right model's respectively. The blue background colour indicates instances where initial parameters did not converge to observed model parameter values. Basins of attraction which converge to the left model, but situated near the right model, are highlighted by white squares. (See appendix 11 – VariedXorZkTwoBodies) 126

Figure 124: Calculated fractal dimension as a function of the damping factor used. The same model is used for every damping constant. As is in Figure 35 - Figure 39, the observed model is at 0 m along the range and buried at a depth of 10 m. (See appendix 7 – BasinOfAttractionKIterationsTracked) 128

Figure 125: Plot of the x position as a function of iteration; the modified Moore-Penrose inversion, in equation (13), is used with a C value of 2. The fixed points are highlighted in red. (See appendix 18 – LapyunovC) 129

Figure 126: Plot of the x position as a function of iteration; the modified Moore-Penrose inversion, in equation (13), is used with a C value of 5.5. The fixed-points are highlighted in dark red and linked to each other by pink. (See appendix 18 – LapyunovC) 130

Figure 127: Plot of the x position as a function of the iterations, the modified Moore-Penrose inversion process, as described in equation (13), is used with $C = 7$. (See appendix 18 – LapyunovC) 133

Figure 128: Observed response in a as a result of the model in b . Model is created with two cylindrical bodies, buried at 10 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel) 134

Figure 129: The basin of attraction, as created using the Ridge-regression inversion, in equation (8), and various damping constants k . All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations for parameter sets to converge to either the left or right observed body. The black

background indicates parameter sets which did not meet convergence conditions. (See appendix 7 – BasinOfAttractionKIterationsTracked) 135

Figure 130: The relative gravity profile extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. *A* and *B* correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump) 136

Figure 131: Basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the buried spherical forward model and convergence to the buried cylindrical forward model, respectively. Localised basins of convergence to the cylinder model highlighted by the white boxes. (See appendix 23 – RealDataMisfitJumpFRAC) 137

TABLE OF TABLES

Table 1: Parametric summary of the inversion process as observed in Figure 22 and Figure 23.

The table tracks the iteration, x parameter value (location), z parameter value (depth), x parameter change value and z parameter change value, respectively. 36

Table 2: Comparison of computational times required to construct basins of attraction with different solution spaces. 127

CHAPTER ONE - INTRODUCTION

1.1 General introduction

The analysis of the structure and composition of the Earth and its subsurface is the goal of many geoscientists. Geoscientists use equipment to measure geophysical responses, which are modeled to create a picture of the subsurface. Gravity method is one of the well-established and commonly-used geophysical methods. Gravitational responses are measurements of the vertical component of the Earth's gravitational field at a point. The gravity field at a point will be affected by rocks or materials of different densities in its vicinity. Gravity surveys are conducted for many reasons, such as for mineral exploration or void detection by exploiting density contrasts wherever the surveys are conducted. Interpretations of the response in order to create models of the subsurface are not unique processes and the interpretations should always be constrained by geology and other geophysical techniques. The most common means of data analysis and model generation from geophysical responses are the forward modeling and inversion processes. (Cooper, 2000)

Forward modelling involves manually altering model parameters to create the most accurate subsurface model. Geophysicists start by estimating initial geological model parameters, such as geometric data, to generate geophysical responses. These responses are then contrasted with the observed data. The parameters of the model are then manually altered, if needed, in an attempt to better match the observed data, while preserving the model's geological feasibility.

Inversion is a semi-automated process that seeks to minimise the difference between a model's generated geophysical response and the observed data. The starting model created by the user is characterised by its parameters. This model generates a geophysical response, which is compared with the observed data, and the resulting misfit is calculated as the difference between the geophysical response generated by the model and the observed data. This difference, when plotted over the parameters of the model defines the misfit function. Thus, inversion is a

progressive and often iterative process that modifies the existing model parameters in order to minimize the total misfit.

The main goal of inversion is to find a set of parameters that minimise the misfit function. If the misfit function is a flat hyper-plane, then the parameters being inverted are said to be linear under inversion and inversion applied to the starting parameters will cause the point to immediately converge to the global minimum solution, which is the solution with the smallest misfit. However, the non-linearity of the misfit surface complicates the inversion process. This is unfortunately the case for most geophysical inversions. If a parameter is not linear under inversion, the misfit surface is curved and may require multiple inversion iterations to produce a suitable solution that is characterised by a low but not the lowest possible misfit. Such solutions are local minima, of which there may be many. Thus the inversion solutions are dependent on the starting model parameters. Local minima are problematic not only in geophysical inversions but also nearly all types of non-linear optimization problems. (Cooper, 2000)

One potential method of avoiding local minima is to exploit the effects of chaos on non-linear systems. Chaos is a state of a system that appears to be unpredictable or random. Inversion is a form of non-linear dynamics and subsequently can exhibit chaos. If the path an inversion would follow is observed to tend toward a local minimum, or undesired results, then the manipulation of the system's intrinsic chaos might steer the inversion out of local minimum. In geophysical inversions, this would result in better matches between the model response and the observed data. This area of research is important considering almost all known real world systems can exhibit chaos at various observational scales. The inversion process can be modified to introduce bifurcations and chaos into a system's solutions. Studying the iterative inversion process as it is subjected to chaos, and the associated bifurcations can provide insight into the inversion process and some fundamentals of non-linear dynamics. (Cooper, 2001)

Many other methods can be used to invert gravity data, such as; Euler inversion or the equivalent layer method. The least-squares inversion method is investigated here, due to its existence in the field of non-linear dynamics and the amount of references and support available for the method. The inversion process has an easy to track iterative nature and that translates easily into a

geophysical space. Monitoring the inversion process can help users investigate its chaotic and fractal nature.

Aspects of this work have been presented at conferences or annual meetings. In 2017 a poster was presented at the South African Geophysical Association's (SAGA) conference held in Cape Town, South Africa. The poster was titled "Analysis of the fractal convergence properties of geophysical inverse problems". A poster with the same title, but improved analysis, was additionally presented at the 2018 Society of Exploration Geophysicist's (SEG) annual meeting in California, United States of America.

The abstract summary for the SAGA 2017 conference is available at <https://drive.google.com/open?id=0B6Ho26AP3EubeDhYQU5qUUp6OUE>.

The abstract summary for the SEG 2018 annual meeting is available at <https://library.seg.org/doi/10.1190/segam2018-2997776.1>.

1.2 Aims and objectives

1.2.1 Aims

- Investigate the geophysical inversion process and understand the role chaos plays therein.
- Investigate inversion convergence properties and attempt to improve them, including avoiding local minima and descending towards global minima.

1.2.2 Objectives

- Understand the inversion process:
 - Understand its convergence properties:
 - Determining if the inversion process is fractal in nature.
 - Measuring the basin of attraction as a function of model parameters, such as damping.
 - Study particular parameter trajectories as the inversion iterates.
 - Describe quantitatively the fixed-point and chaotic behavior of geophysical inversion processes.

- Apply research results to predict the behavior of the system (convergence, periodic, or chaotic) as a function of model parameters.
- Optimize non-linear geophysical inverse problems to improve their convergence properties, in the sense that it reaches the global minimum rather than a local minimum.
- Apply the above methods to real geophysical data sampled from the Trompsburg area in the Free State province of South Africa.

1.3 Literature Review

1.3.1 Fractals

Fractals are an integral contribution to the field of mathematics. A fractal is a pattern which exhibits scaling-invariance and repeats itself at varied scales. Fractals were only computationally-feasible roughly 35 years ago and were first attempted by *Benoit Mandelbrot* (Frame, 2014, pp 5-10). The fundamental mathematics behind fractals was however originally formulated in the 17th century. The idea of fractals was a contribution of the mathematician *Gottfried Leibniz*; who postulated recursive self similarity and fractional exponents (Feng and Chaun-Feng, 2013).

One of the first fractals on paper printed was by *Helge von Koch* in 1904 in the form of, what is now known as, the “Koch snowflake”. This pattern results from modifying all flat sides of an initially-triangular object, to triangles and repeating the procedure as much as desired, mathematically but not physically *ad infinitum* as in Figure 1.

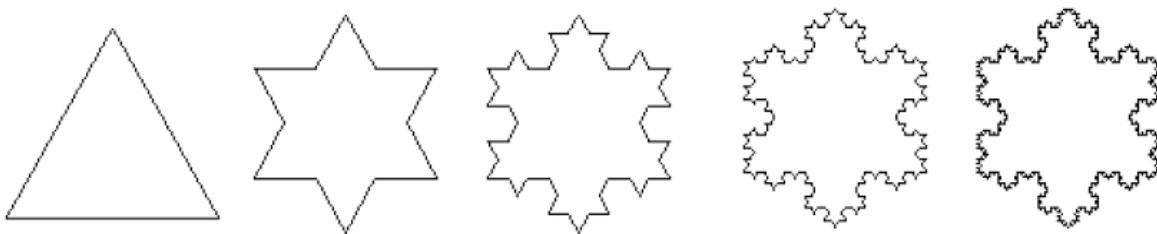


Figure 1: Koch snowflake; as constructed through the protrusion of triangles along the body's surface (Koch snowflake, 2016).

Fractals are often observed as complicated patterns, but their origins are just the repetition of a simple geometric or arithmetic process repeatedly performed. Fractals have been used to interpret non-linear dynamics systems and aspects of the universe for a few decades; the first attempt at modelling galaxies and tier distribution using fractals was made as early 1987 (Coleman and Pietronero, 1992).

The fractal nature of chaotic systems is often studied in the domain of fractal mathematics. Fractal patterns often account for the complex nature of the observed chaos; much like the shape lightning takes upon its descent. Fractals can be computationally constructed but are also observable in nature; these fractals start from a basic repeating pattern and evolve to form complex structures such as lightning strikes or snowflakes, seen respectively in Figure 2.



Figure 2: Image of a lightning strike and a snowflake as shown on the left and right, respectively (Lightning strike, n d) (Snowflake, n d).

The *Sierpinski Triangle*, see Figure 3, is a synthetic fractal generated from the continuous appending of equilateral triangles with the same geometric orientation, on all existing triangles' edges (Conway and Guy, 1996).

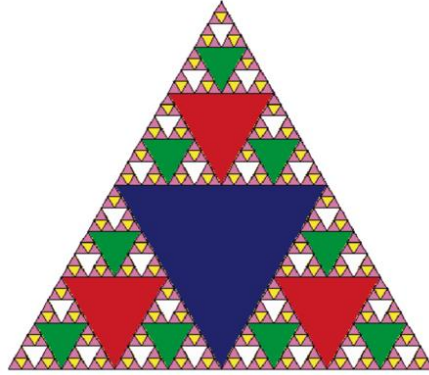


Figure 3: Completed Sierpinski triangle (Sierpinski triangle, n d).

Fractals do not lie within the same dimensional space as Euclidean geometry and instead possess fractional dimensions.

1.3.2 Fractal dimensions

The property in which a pattern repeats itself exactly but at different scales is called “self-similarity” (Hutchinson, 1981). The Mandelbrot set and subsequent fractal pattern are often used to illustrate self-similarity. The Mandelbrot set is constructed by tracking values in a complex plane and testing if iterations tend to infinity (or out of defined bounds). Arrays of complex numbers c , within the complex plane, are iteratively tested in the function:

$$f(z) = z^2 + c$$

Where each $f(z)$ value operates on the precursor, to test for c values which do not converge when the iterations start from a z value of 0 (Hillborn, 1994, pp 406). Figure 4 is the Mandelbrot set as constructed using this method.

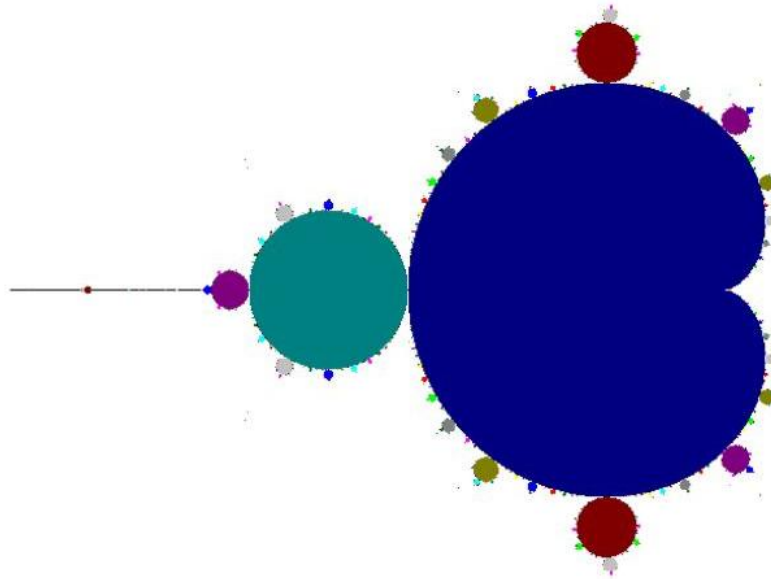


Figure 4: The Mandelbrot set (Sprott, 1997). The pixels of colour represent starting c (complex numbers) which do not diverge to infinity using the Mandelbrot expression.

The self-similarity present in the Mandelbrot set can be seen by enlarging a section within the set, and observing if the pattern repeats itself. Figure 5 shows the Mandelbrot set as a basin is enlarged in order to define self-similarity within the set.

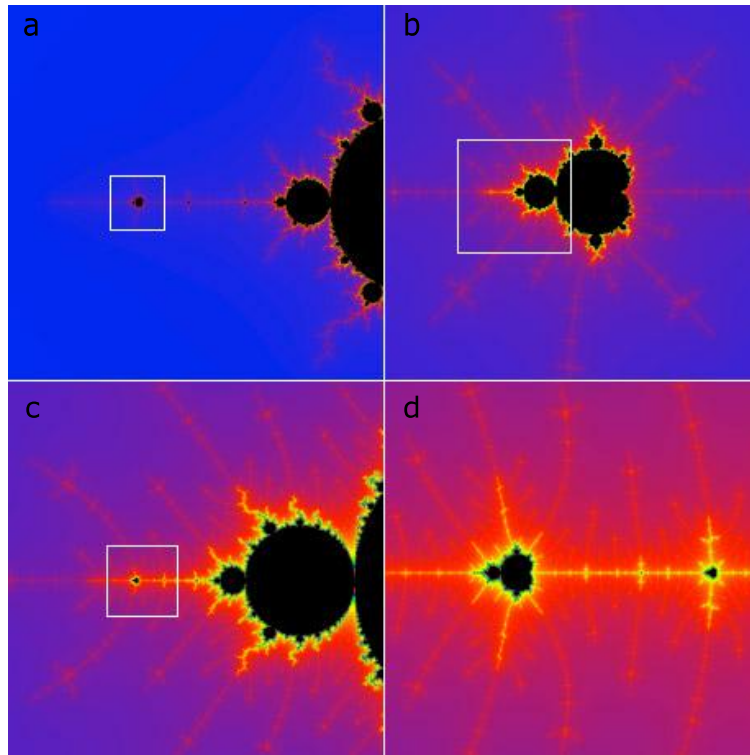


Figure 5: Establishing self-similarity in the Mandelbrot set. A basin of convergence is identified in *a*, and is enlarged to form the image in *b*. This process is repeated to form the images in *c* and *d*, respectively. At each point the same basin is observed and it becomes evident that the series of patterns repeat themselves at different scales, thus exhibiting self-similarity. (Mandelbrot, n d)

The Mandelbrot set exhibits self-similarity and therefore possesses a fractional dimension. Fractional dimensions result in fractals not following linear multiplication rules such as scaling by a scalar constant. The idea of fractional dimensions is a difficult one to visualise but it is one which can be mathematically evaluated just by using progressing dimensions (Kaye, 1994).

One dimensional constructs will follow simple linear multiplication. Multiplying the length of the line segment by a constant outputs the same amount of segments, each as long as the original segment. For example, if a 1 m long line were doubled, the end result would be a 2 m line comprised of two 1 m long portions. A two dimensional construct will follow a square law in multiplication. Enlarging a square by a set constant, enlarging both sides by the same constant, results in a powered amount of copies generated. Doubling the size of a square produces a square with four times the area of the original square. For example, if a 1 m by 1 m square were doubled in size, with both sides being doubled, the resultant square would be a 2 m by 2 m square comprised of four 1 m by 1 m squares. By following the same principles; doubling the size of a

cube leads to 8 original sized cubes being represented in the final volume. Algebraically; the number of original copies which can be represented again in the final construct is represented by the linear multiple raised to the dimension of the object. This can be expressed as in either equations (1) or (2):

$$L^D = N \quad (1)$$

$$a = \frac{1}{s^D} \quad (2)$$

L is the linear multiple, D is the dimension of the object, N is the amount of original copies that can be represented within the end construct, a is the number of segments, s is the factor of reduction or enlargement and D is the self-similar dimension. This idea can also be visualized geometrically, as illustrated in Figure 6, where the r constant is equivalent to the L or s factor in equations (1) or (2), respectively.

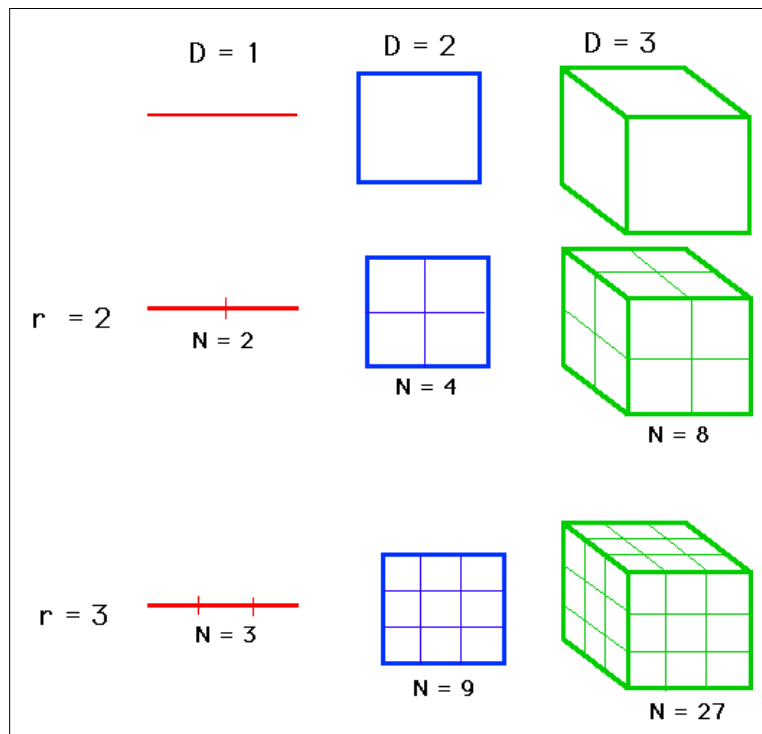


Figure 6: Illustration of progressing dimensions. N , r and D represents the amount of originals, scalar multiple and the dimension, respectively. (Progressing dimensions, n d).

Since fractals do not produce the same trend as with the line, square and cube it becomes understandable how solving for the dimension, D , in equations (1) or (2) above might yield fractional numbers for fractal images. A leading example is the Sierpinski triangle, as introduced in Figure 3, and an alternate construction method is expanded upon in Figure 7. The Sierpinski triangle is constructed with its basis in an equilateral triangle. Construct a new triangle with vertices at the midpoints of each side of the original triangle, the new triangle is removed from the existing triangle. This leaves three triangles with half the dimensions of the previous triangle. The new structure also has one fourth of the previous area.



Figure 7: An iterative diagram showing the construction of the Sierpinski triangle (Sierpinski triangle constructions, n d).

In Figure 7; a is the 0^{th} order, b is the 1^{st} order and c is the 2^{nd} order. As the order increases, and the pattern is iterated, the amount of triangles is tripled. The scaling factor is the factor by which each triangle in the subsequent order is shrunk. The construction of interior triangle is accomplished from midpoints along the original triangle, thus the scaling factor is 2. Equation (1) can be used to discern the dimension D if; L is the linear multiple (scaling factor) of 2 and N is equal to 3:

$$L^D = N$$

$$\text{Log}(L^D) = \text{Log}(N)$$

$$D \text{ Log}(L) = \text{Log}(N)$$

$$D = \frac{\text{Log}(N)}{\text{Log}(L)} = \frac{\text{Log}(3)}{\text{Log}(2)}$$

$$D = 1.59$$

A dimensions of 1.59 for the Sierpinski triangle, is a real, non-integer number, resulting in a fractal pattern and dimension. The fractal dimension, as calculated by equation (1), is easy for

progressing patterns but is more difficult to execute on existing static patterns in Euclidean space, such as those in Figure 2 or Figure 8. Two methods of calculating the fractal dimension in Euclidean space are the self-similarity dimension approach, as described in equation (2) above, and the box-counting dimension approach. The box-counting dimension, also known as the Minkowski-Bouligand dimension, is a collaborative effort of the mathematicians Hermann Minkowski and Georges Bouligand (Ge and Lin, 2009).

1.3.3 Fractals in Euclidean space and the box-counting method

Calculating the box-dimension (fractal dimension) is achieved by first placing the image or pattern flat on a grid. The overall grid is separated into a mesh of equal sized boxes, as small as desired or as large as possible within the grid. The boxes which cover the boundary of the object are then tallied. Variations (in colour or index) often form around the edges of the fractal; this results in the box-counting method running along the boundary of the image. The boxes' sizes are then equilaterally increased or decreased incrementally and the tallies again recorded each time. A visual representation of this is shown in Figure 8.

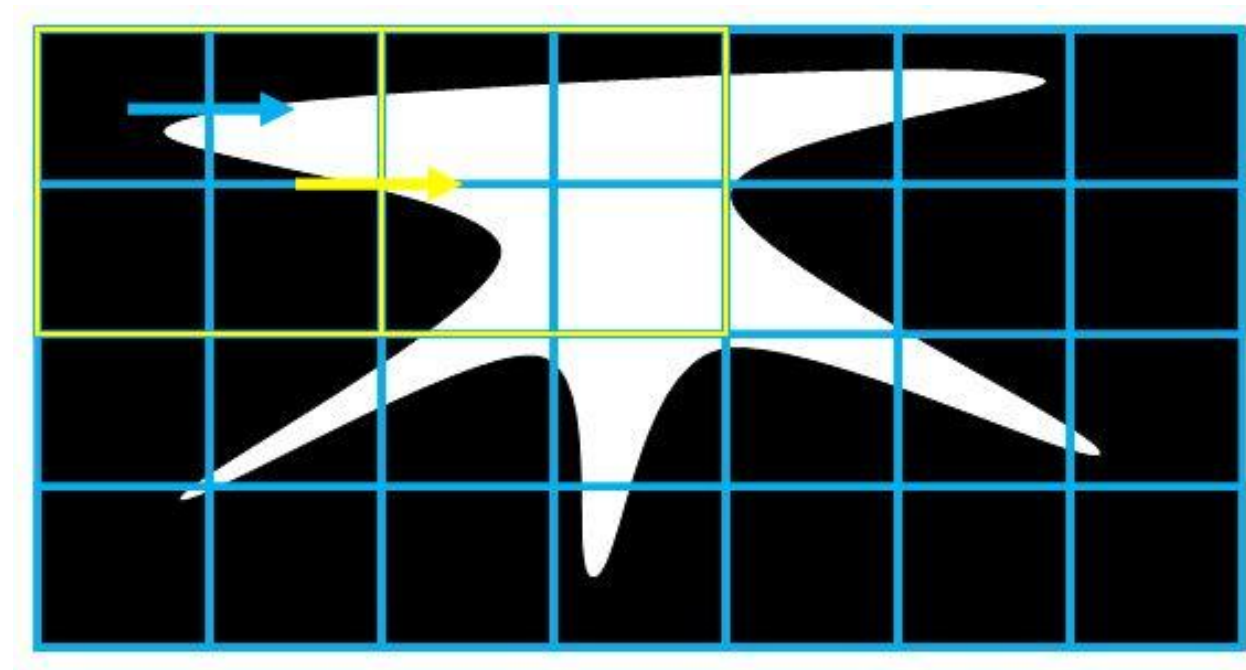


Figure 8: Visualisation of the box-counting method. A theoretical fractal image (in white) is placed on a grid and the box counting method is applied to it. Blue boxes indicate the original smallest box size and the yellow boxes show a larger box sizes. As the box-counting method is employed, the smaller boxes (blue) move through the image (as shown by the arrow) until the entire image has been sampled. The box size increases (yellow) and the process repeats until the boxes are as large as they can be.

When a box is sampled, a check is done for whether or not there is variance in colour or shape within the box. A record of how many times there is variance within a box relative to its given size is used to aid in discerning the dimension. The gradient of the line of best fit, for the log of these increments against the log of the respective box sizes is equal to the box-dimension and subsequent fractal dimension (Cooper, 2000).

Equation (3) describes the box-counting dimension D as a function of N_S , the box sizes and N_C , the corresponding count of variances. $\ln$ is the natural logarithm of the array and Δ is the delta symbol for a finite difference (Hillborn, 1994, pp 394).

$$D = \frac{\Delta \ln(N_C)}{\Delta \ln(N_S)} \quad (3)$$

1.3.4 Chaos theory

The principles of chaos theory owe, in part, their existence to the meteorologist Edward Lorenz. In the 1960s Edward Lorenz used a set of equations to model a theoretical weather system. Lorenz noticed that every model was identical as it progressed through time so long as the initial parameters (conditions such as pressure, temperature, time, humidity, etc.) were kept the same. One day, Lorenz started the system at parameters further along in time, it was expected that all other parameters would be equal when the two time parameters were again equal. Essentially one of the systems was started later, late morning, and it was assumed by Lorenz that in the afternoon all other parameters would be equal to the same parameters of a system which was started earlier in the day. He noted that although the original parameters were only slightly altered the final model had deviated greatly. When Lorenz ran his system even longer he noticed these variances increased. Lorenz originally believed the mechanical systems were at fault, but he eventually discovered his varied model results were exhibiting what was at the time defined as “sensitive dependence on initial conditions”, more commonly now known as the Butterfly effect. Edward Lorenz would go on to describe chaos as “when the present determines future, but the approximate present does not approximately determine the future” (Gleick, 1987).

The field of chaos theory has grown substantially from its inception. Chaos theory is now defined as the study of the behaviours of dynamic systems. Dynamic systems are highly sensitive to initial conditions, such as was shown by Lorenz and his weather predicting system. These systems are sensitive to measurement and numerical precision and as such; are often deemed unpredictable and hence seemingly random in behaviour. Despite the apparent randomness, there are underlying patterns in the system solutions, including self-similarity, fractals and bifurcations.

A bifurcation is the division of something into two similar parts and bifurcations are evident in mathematics as well as in nature. In nature one can observe bifurcations in the forks of tree branches, fractures in rocks or even rivers; an example of such can be seen in Figure 9. Fractals have been described as one representation of chaos and fractals have also been seen to exhibit bifurcations (Devaney and Keen, 1988).



Figure 9: The bifurcation of a river as it cuts through the landscape (Bifurcating river, n d).

Many physical systems, such as the stock market or the weather, exhibit chaos. The chaotic dynamics exhibited in physical systems contain features such as: unpredictability, feedback, fractals, the butterfly effect, order and disorder, and mixing.

- Unpredictability is a result of the inability to know, with absolute certainty, the initial conditions of a system, and thus the ultimate outcome of these systems is impossible to determine.
- Systems are able to succumb to chaos when subjected to the effects of feedback, which is a method of over-correcting system variables to a state where the variables themselves need to be re-corrected repeatedly introducing chaos.
- Fractals are the recursive dynamic displays of chaos.
- Chaos is not simply a distinction between order and disorder, and instead features both.
- Mixing occurs when two adjacent trajectories (solutions of dynamical systems) diverge exponentially over time and resemble the trajectories of distant initial conditions. A good example of mixing is the case of two water molecules, which may have once been close to one another in the ocean, but after time, may end up incredibly far apart.
- The butterfly effect is amongst the most well known principles of chaos; this principle states that small changes in an initial condition – no matter how small - can cause unexpected outcomes which deviate far from expected results.

If a system is defined as a set of various interacting components which contribute to a larger whole than before, then a non-linear system is the result of the aforementioned feedback effect. A non-linear system is the result of feedback between the components that exhibits behaviour at a larger scale than that of the individual constituents. Chaotic systems may contain minimal or many interacting parts and may follow simple rules but are still very sensitive to initial conditions (Abarbanel, 1995).

1.3.5 The logistic map and bifurcation diagram

One example of visualising chaos in a non-linear dynamical system is with a logistic map, or bifurcation diagram. The bifurcation diagram is a type of a recurrence relation where each additional term is defined as a function of previous terms. It can be thought of as a discrete version of differential equations and hence is a type of discrete dynamical system. It is useful to illustrate chaos that arises from relatively simple non-linear dynamical systems. The most notable bifurcation diagram is the one based on the s-curve logistic function, which attempts to map population growth, called the logistic map. The population growth is represented by the

logistic map's non-linear difference equation that states that the population at a time $t+1$ is dependent on the population at time t . Such a relation can be written generally as equation (4).

$$x_{t+1} = rx_t(1 - x_t). \quad (4)$$

Where x represents the current population at a given time normalized to a maximum carrying capacity and hence ranges from 0 to 1. t and r are the time and the rate of growth of the population, respectively. This simple model captures two competing phenomena – reproduction via the rx term and starvation via the $(1-x)$ term. This process can be iteratively solved for various initial conditions. If the initial conditions which result in stable populations (such that the population over time does not change) are recorded, the resultant logistic map is constructed, see Figure 10.

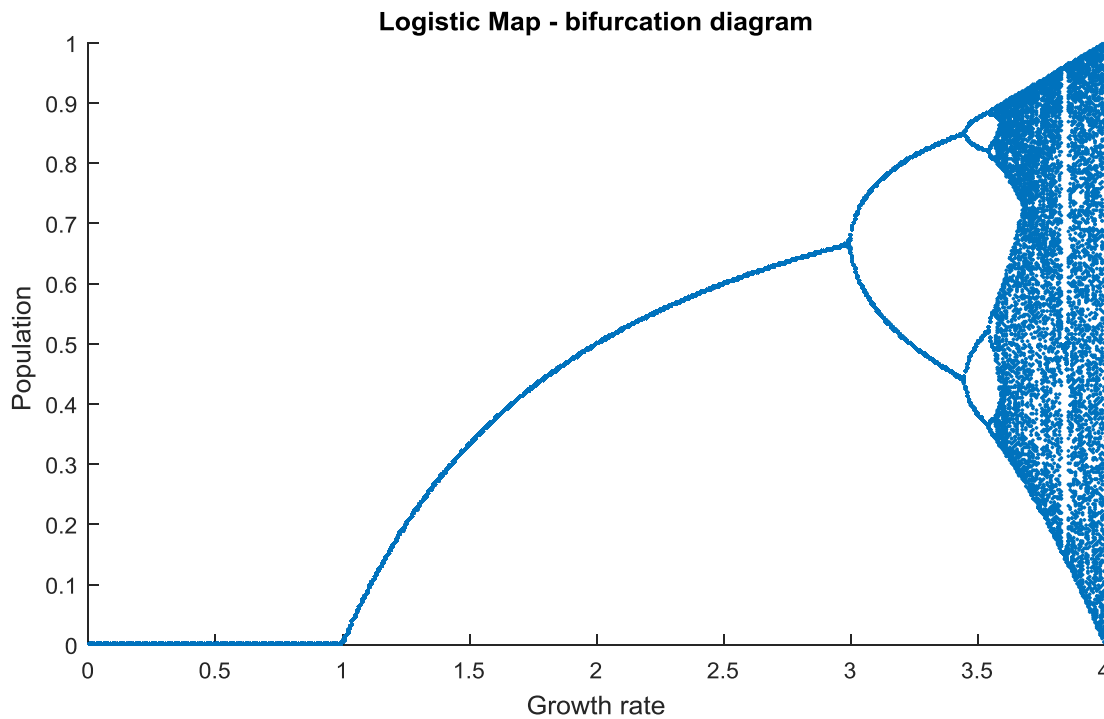


Figure 10: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). This logistic map plots potential population magnitudes as a result of growth rates. (See appendix 1 – LogisticMapGenerator)

In Figure 10 it is observed that the stable solutions (fixed points) are dependent on the initial population and the growth rate. All the system runs are set up where the initial populations are

0.5 (half of the maximum). Trends upward indicate instances of stability in higher populations (reproduction) and trends downward indicate the opposite (starvation). When the growth rate is too small the population eventually becomes extinct (0 in Figure 10). Increasing the growth rate results in the doubling of fixed points (bifurcations); called *period doubling*. This behaviour is seen to repeat several times as the growth rate is gradually incremented, such that the number of stable populations double at around $r = 3$ and again around $r = 3.5$. After about $r = 3.5$, the number of stable populations rapidly double as r is further incremented. These stable values toward which the population tends to settle belong to a set of fixed points known as attractors. For example, at growth rates smaller than 1, the system's fixed point attractor is $x=0$. At a growth rate of 3.5 the system's attractors form a *limit cycle*, beyond which periodicity is not discernible and chaos appears to set in. These attractors in the chaotic behaviour parameter space are a type of a *strange attractor*. A strange attractor exhibits fractal structure and does not result in randomness but rather apparent randomness (Hillborn, 1994, pp 27). By analysing a subsection of the logistic map near chaotic regions, see Figure 11, sections of order occur where the system behaviour appeared to be non-chaotic.

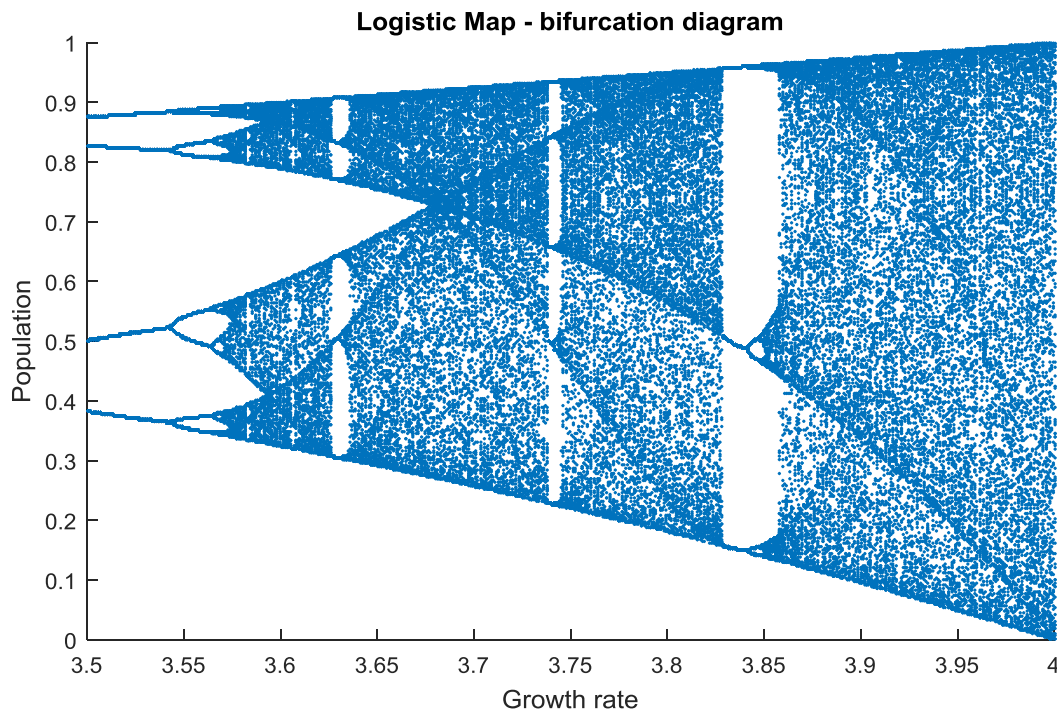


Figure 11: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The growth rate is only varied from 3.5 to 4. (See appendix 1 – LogisticMapGenerator)

The logistic map demonstrates beauty in chaos; where parameter thresholds create limits in which the same periodicity is again observed and ranges of once declared chaos allow fleeting periodicity to exist.

Exploring the periodicity in bifurcation diagrams can be highlighted with the Feigenbaum constants. The Feigenbaum constants are mathematical constants which express ratios within the bifurcation diagrams. The first Feigenbaum constant δ_n is the limiting ratio of each bifurcation interval to the successive bifurcation interval within every bifurcation. The first Feigenbaum constant is calculable from equation (5), where A_n is the n th bifurcation interval for that period doubling (Hillborn, 1994, pp 48):

$$\delta_n = \lim_{n \rightarrow \infty} \frac{A_n - A_{n-1}}{A_{n+1} - A_n} \quad (5)$$

A is the bifurcation parameter; here representing growth rates. The growth rates used in solving for the first Feigenbaum constant can be seen in Figure 12, where the different A values are identified by colour.

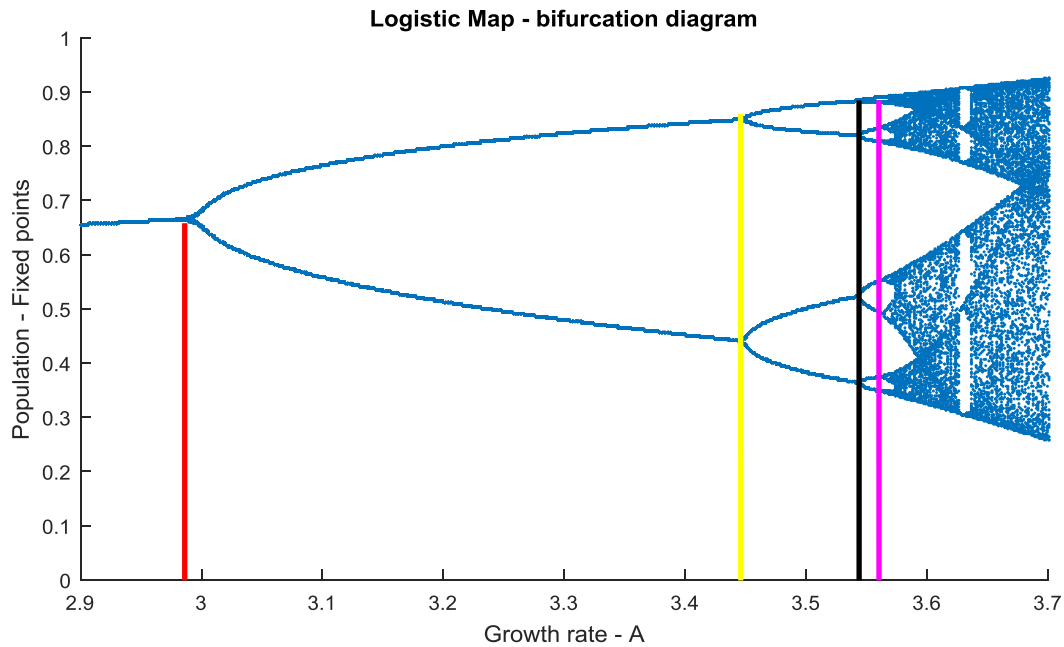


Figure 12: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The logistic map here records the growth rates, A , as a function of the fixed stable points needed for calculating the first Feigenbaum constant (see equations (5)). The growth rate is only varied from 2.9 to 3.7. The A values for the red, yellow, black and pink line are 2.98, 3.45, 3.54 and 3.56, respectively. (See appendix 1 – LogisticMapGenerator)

Calculating the first Feigenbaum constant; using equation (5) and the values identified in Figure 12:

$$\delta_2 = \frac{A_n - A_{n-1}}{A_{n+1} - A_n}$$

$$\delta_2 = \frac{3.45 - 2.98}{3.54 - 3.45}$$

$$\delta_2 = 5.22$$

The second Feigenbaum constant α_n is a ratio between the width of a tine (branch of a bifurcation diagram) and the width of one of the two successive sub-tines, the ratio is multiplied by a negative 1 when the ratios are read backwards toward a lower n value. The second Feigenbaum constant can be calculated using equation (6) (Hillborn, 1994, pp 54).

$$\alpha_n = \frac{D_n}{D_{n-1}} \quad (6)$$

The widths of tines as used for solving the second Feigenbaum constant can be seen in Figure 13, where the widths are identified by colour.

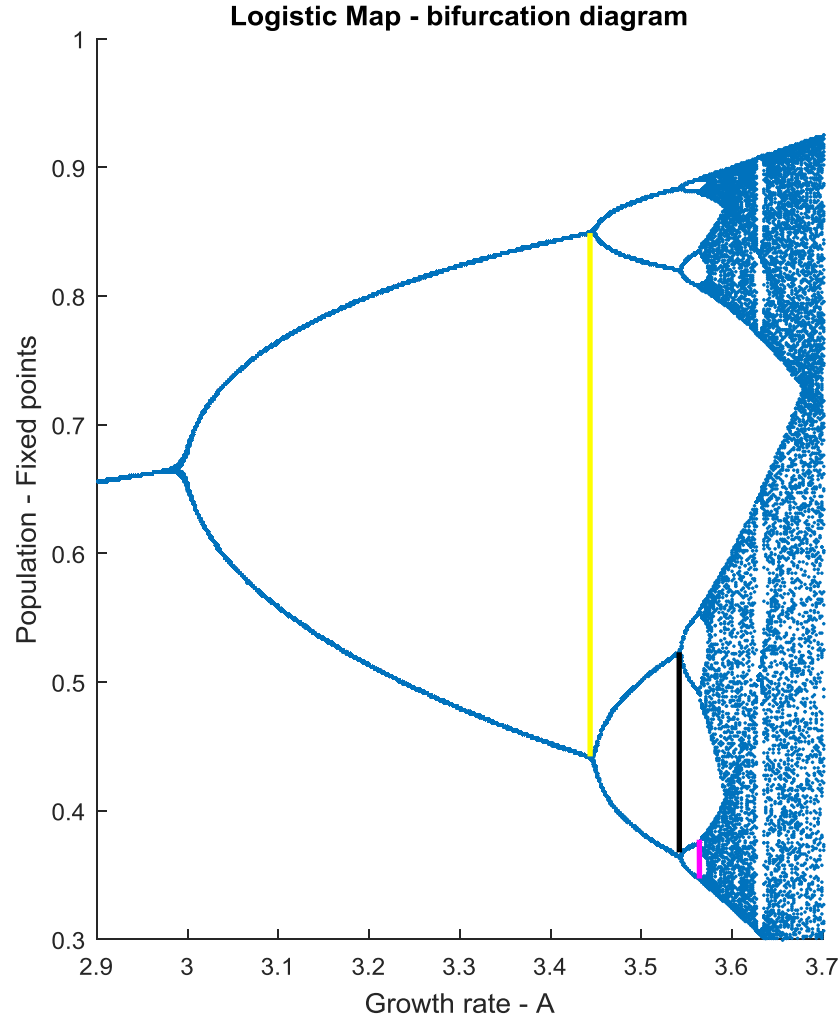


Figure 13: A logistic map is established through the recording the stable points in iterations of the quadratic equation in equation (4). The logistic map here records the growth rates, A, as a function of the fixed stable points needed for calculating the second Feigenbaum constant (see equation (6)). The growth rate is only varied from 2.9 to 3.7. The widths of the yellow, black and pink line are 0.4081, 0.1588 and 0.0273, respectively. (See appendix 1 – LogisticMapGenerator)

Calculating the second Feigenbaum constant; using equation (6) and the values identified in Figure 13:

$$\alpha_n = \frac{D_n}{D_{n-1}}$$

$$\alpha_2 = \frac{0.41}{0.16}$$

$$\alpha_2 = 2.56$$

My calculated values do not match published results exactly; where δ_n and α_n are 4.67 and 2.50, respectively (Feigenbaum, 1979). A potential reason as to the deviation between the published results and my calculated results, are that published results have tended toward an infinite scale while I have used the first few bifurcations exclusively.

1.3.6 Inverse theory and geophysical forward modelling

Two useful methods of investigating the Earth's interior are the analysis of the magnetic and gravitational fields. The Earth possesses an electromagnetic field. The field is the result of the Earth's solid and liquid iron core, the influences of convection and the associated spin of the Earth. The Earth also possesses a gravitational field due to its mass. The field, however, is not constant at all places on the Earth and is influenced by the distance of a location from the core's centre and the density of the subsurface. The most common methods of evaluating the geophysical data and interpreting the results are inverse theory and forward modelling.

Forward modelling is the process of creating a subsurface model and manually tuning its parameters to generate a geophysical response that best matches the observed response. This process is iterated until a viable geological model is established. The discrepancy between the observed response and the generated response is the misfit of the forward model. Thus a principle quality control of forward modelling is to minimise the misfit to within geological plausibility.

Inversion operates by using mathematical processes to minimise the misfit with little user interaction. A starting model is created by the user and its geophysical response is calculated. Inversion manipulates the parameters of the starting model to create responses that better fit the observed response and thus minimise the misfit. Inversion is a slightly faster method of model generation compared to forward modelling, as it uses the observed data to generate calculated responses directly. A record of the misfit versus the model parameters grants insight into the nature of the system and is called the misfit function. If only one parameter is inverted for, the record, of the misfit function is a lineament, and the parameter is said to be linear under inversion. If the parameter acts linearly under inversion then the inversion acting on the initial parameter will cause it to immediately invert to the most favourable parameter solution, that being the one with the smallest misfit. If the inversion acts on two or more parameters then the

misfit function is non-linear (a curve). If the misfit function is non-linear then the parameters are said to act non-linear under inversion and can require multiple iterations to come to a suitable solution; graphically represented as a minimum on the misfit function. A misfit surface may contain many local minima. As such, the inversion solutions are dependent on the starting positions of the inversion process. The minimum possessing the smallest misfit is the global minimum. In geophysical inversions a local minima is a set of model parameters that produce a geophysical response which fits the observed data better than the starting model, but is not the set of parameters that produces the best possible fit. Examples of the misfit surfaces for parameters that linear and non-linear under inversion are shown in Figure 14.

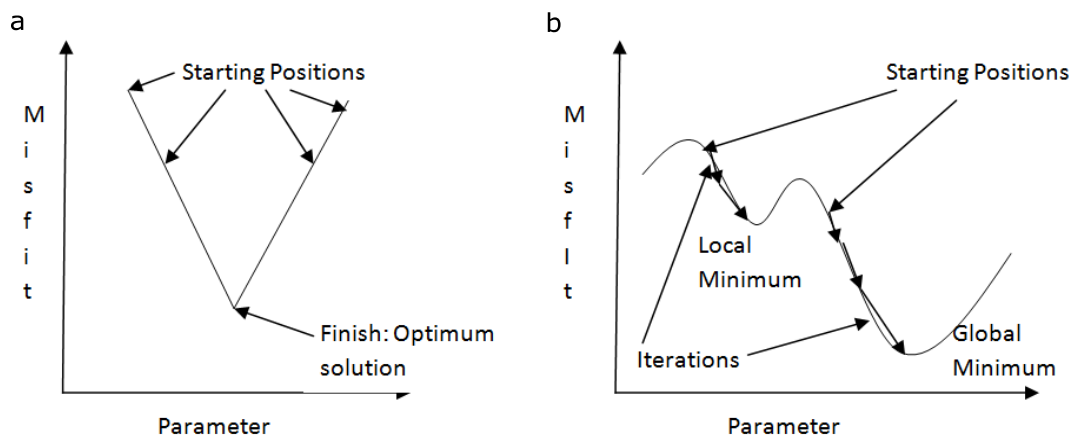


Figure 14: Representations of Misfit surfaces for parameters linear under inversion (a) and non-linear under inversion (b).

Due to the potential for many local minima along a misfit surface, inversion may not provide a perfect solution to a problem. Many inversion schemes may converge to local minima and provide reasonable fits for the observed responses but do not provide valid models to match the subsurface geology. For example, a body can be buried at different depths with different density contrasts and create identical gravity responses; this problem is illustrated in Figure 15.

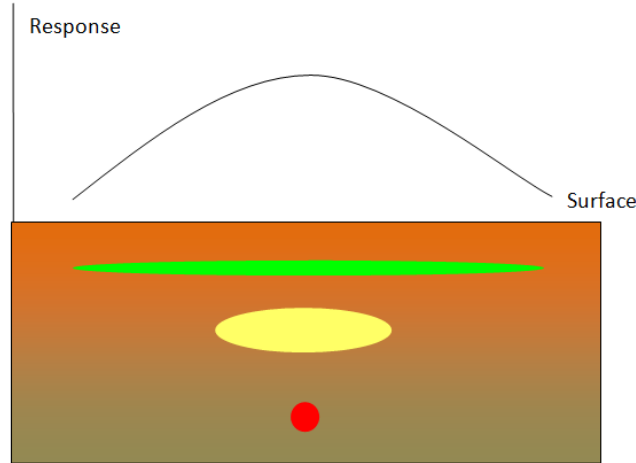


Figure 15: An illustration of how the same gravitational profile can be generated from an infinite amount of models. Shallow, large and low density bodies will generate the same gravitational response as small, deep and very dense bodies, respectively.

Moore-Penrose inversion

A type of inversion commonly used is the least-squares inversion. In this scheme, the sum of squares of residuals is used to measure the misfit between the observed and calculated responses. If a model's responses are non-linear under inversion, then the non-linear least squares inversion is useful. One popular formulation of the least-squares inversion is commonly called the Moore-Penrose inversion.

$$dP = (A^T A)^{-1} A^T D \quad (7)$$

Equation (7) solves for dP , which is the parameter change vector and is the calculated perturbation to the starting parameters. D is the difference matrix; the calculable misfit between the observable and calculated response. A is a matrix of partial derivatives with respect to each parameter, such as modelling the depth or displacement of an object in a forward model. The superscript T denotes the matrix transpose of the matrix and the superscript -1 denotes the inverse. The generation of the responses and the calculation of the associated misfits and partial derivatives are calculable from synthetically generated data or can be extracted from real data.

Ridge-regression

The Moore-Penrose inversion has unfortunate shortcomings in its operation. Problems such as divergence of starting conditions, excessive processing due to the necessity of multiple iterations to converge, and the possibility of $A^T A$ not existing can be addressed through the use of Ridge-regression. This improved scheme can be written:

$$dP = (A^T A + kI)^{-1} A^T D \quad (8)$$

Where k is a damping factor and I is an identity matrix sharing its size with the size of $A^T A$. The damping factor also allows the $A^T A$ term to refrain from becoming arbitrarily large under inversion. The damping factor can be kept constant through all iterations or a more common practice is to have the damping factor change as a result of the misfit after each iteration of the inversion. This is put into effect by increasing k should the misfit increase and decreasing k should the misfit decrease. This also reduces the computational workload, as the solutions will converge in less iteration (Cooper, 2000).

1.3.7 Similar methods used to solve non-linear systems

This dissertation focuses on the operations of the least-squares and Moore-Penrose inversion processes and their influence by chaos- and fractal-based manipulations. There are different numerical solvers for non-linear dynamics equations, such as the Secant, Newton and Bisection methods. The Newton's method is often favoured due to its easy and repetitive nature. It is often used to solve for the roots of a real-valued function based on an initial approximation of the result, and subsequent sequential iterations to correct this approximation. The Newton's method tries to solve for x in a function $f(x)$ such that:

$$x: f(x) = 0,$$

A better root is then proposed by the function's digression line. Numerically this is expressed as the iterative stance:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Here $f(x_n)$ is traditionally the real-values function and $f'(x_n)$ is the derivative with respect to x at the point of x_n , the current approximation. This process solves for x_{n+1} , which approximates the

real root better than x_n . Iteratively repeating this process would result in approximating an accurate root to the problem. Solving for higher order polynomial equations can however result in complex roots and the Newton's method can additionally be applied to complex numbers. A famous application of the Newton's method is the creation of what is commonly called a "Newton's fractal". Start with the equation:

$$x^3 - 1 = 0$$

Analytically, the solution has three roots; one real root at $x = 1$ and two equal imaginary roots at $x = e^{\pm 2\pi i/3}$. Newton's method, as applied to this equation, would result in the iterative equation:

$$x_{n+1} = x_n - \frac{x_n^3 - 1}{3x_n^2}$$

A solution space is set up in the complex plane over a range of input complex approximations. On each starting complex number, in the solution space, the iterative process (Newton's method) is run. If the equation converges to the real root at $x = 1$, the initial approximation, in the solution space, is set black (Press et al, 2007, pg 461). The resulting Newton fractal can be seen in Figure 16.

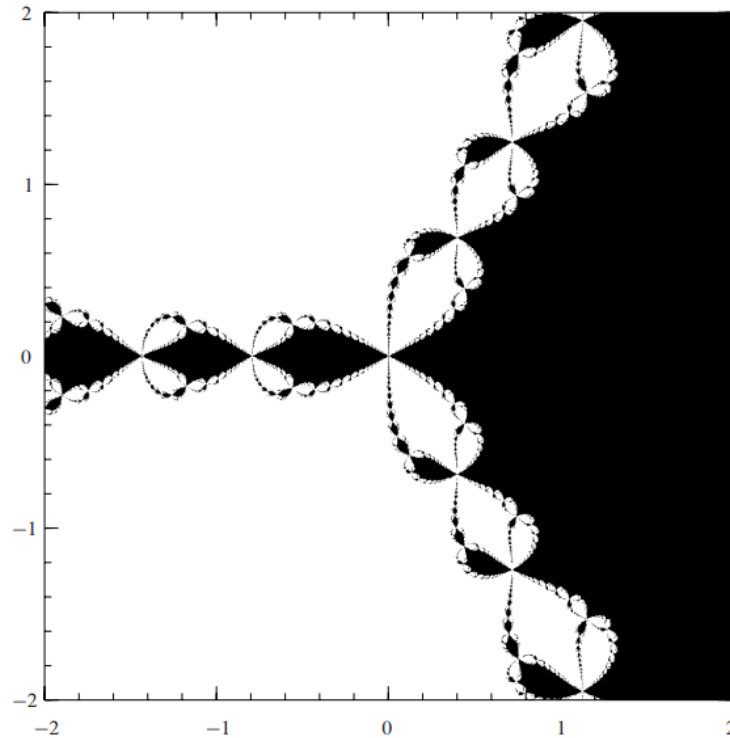


Figure 16: An example of the Newton's fractal. A complex x -plane with real and imaginary components in the range of -2 to 2 is established. Black points in the solution space indicate the set of points from which Newton's method converges to the root $x = 1$ for the equation $x^3 - 1 = 0$ (Press et al, 2007, pg 462).

The Newton method can fail to create accurate solutions to the problem. Should the initial approximation be far from the actual solution, or at a point which results in a zero functional derivative, then the iterations will not create better approximations. Should the problem exhibit many roots, the initial approximation is additionally likely to approximate the nearest root and not necessarily a desired root. It is also possible for the Newton method to cause approximations to oscillate about the root and never converge. This shows sensitivity to the initial conditions of the problem (Akram and Qurrat, 2015)

For this report the Moore-Penrose and Ridge-regression inversion processes are used because of their popularity in the field of geophysical systems. The inversion processes are also easy to program and have ample existing support. These inversion processes are also well-tested on geophysical problems and datasets similar to ours. Sensitivity to initial conditions is a shortcoming of both methods. However, while Newton's method can exhibit limitations since it may succumb to oscillations about an attractor without prior manipulation. This is one of the

aspects tested here and having pre-existing tendencies to oscillations and bifurcations may introduce additional complexities. An example of this oscillatory nature can be seen when the Newton's method is applied to the function:

$$x^3 - 2x + 2 = 0$$

Using an initial approximation of $x = 1$ and applying the Newton's method:

$$x_1 = x_0 - \frac{x_0^3 - 2x_0 + 2}{3x_0^2 - 2} = 1 - \frac{(1)^3 - 2(1) + 2}{3(1)^2 - 2} = 0$$

Following this with $x = 0$ as the new approximation:

$$x_2 = x_1 - \frac{x_1^3 - 2x_1 + 2}{3x_1^2 - 2} = 0 - \frac{(0)^3 - 2(0) + 2}{3(0)^2 - 2} = 1$$

Placing this solution back into the iterative equation repeats the process and creates a period-2 oscillation.

CHAPTER TWO – METHODS AND MATERIALS

2.1 Inversion methodology

One of the most frequently used methods to study the Earth's subsurface is the gravitational method. Gravitational fields are easy to simulate synthetically. Programs and algorithms can generate synthetic gravitational profiles and can be used to calculate misfits between the observed and calculated data sets. The analytical solution to the gravitational response of a buried cylinder follows:

$$g_{cylinder} = \frac{Gmz}{x^2+z^2} \quad (9)$$

$G = 6.67 \cdot 10^{-11} \text{N.m}^2/\text{kg}^2$ and is the gravitational constant. g is the vertical gravitational component for a buried cylinder model at a point along a horizontal range x and a depth z in meters of the centre of the buried cylinder. Note that m the mass of the cylinder, is infinite as it extends infinitely. Analytically; the equation uses the density of the cylinder and the parameter m is computational (Telford et al, 1990).

The cylinder models are set up to have a unit radius of 1 m and are modelled in theoretical synthetic geological model half-spaces thusly. The ability to visually discern changes in the model responses as a result of the inversion process is imperative to understanding it. Figure 17 illustrates a system with one observed cylinder model response (red), and an initial calculated cylinder response (green). Cylinder models are identical in density but offset by a horizontal displacement and depth.

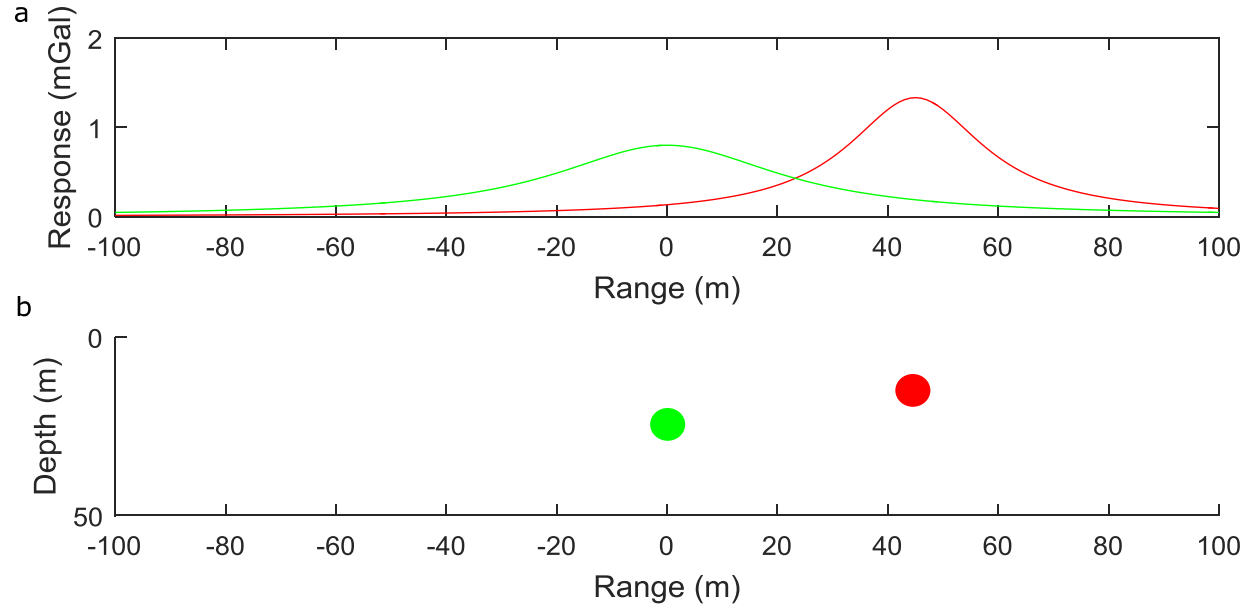


Figure 17: Gravitational responses for an associated synthetic model. Two responses, in *a*, are shown as a result of two separate buried cylinder models, in *b*, using the expression for the gravitational response due to a buried cylinder (see equation (9)). The observed model (in red) is set at 45 m along the range and buried at a depth of 15 m. The initial calculated model (in green) is set at 0 m along the range and buried at a depth of 25 m. (See appendix 12 – ResponsePlusModel)

For the system as defined in Figure 17, the discrepancy used in the Moore-penrose and Ridge-regression inversion processes, listed in section 1.3.6, is the lateral difference between the two responses. The inversion process works to alter the initial model parameter values in an attempt to minimise the discrepancy between the observed and initial model responses, and create a set of model parameters which would result in a minimum on the misfit function. For a geophysical system, the misfit function is defined as the mean squared deviation between the observed (*OBS*) and calculated (*CALC*) responses over *N* measurements along the profile:

$$M = \frac{1}{N} \sum_{j=1}^N (OBS_j - CALC_j)^2. \quad (10)$$

Figure 18 defines a misfit function for the geological model as set up before in Figure 17 and using equation (10) over each point along the range.

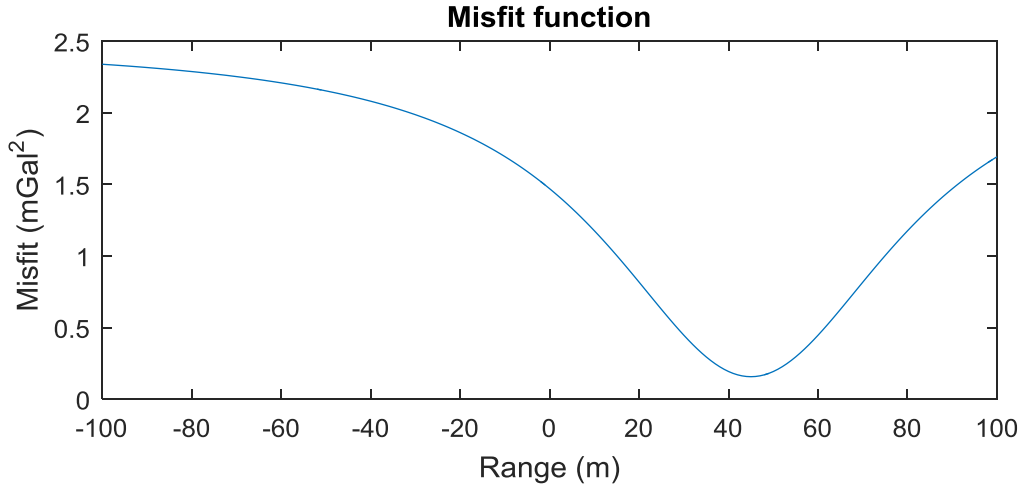


Figure 18: A misfit function, from equation (10), for the geological model established in Figure 17. (See appendix 9 – MisfitFunctionPlusModel)

Minimising the discrepancy matrix should result in the observed and final model's responses being equal to one another. An inversion can be run to correct multiple parameters in several dimensions.

The partial derivatives in matrix A , introduced in the Moore-Penrose inversion process (equation (7)), with respect to each respective parameter can be analytically derived or numerically calculated. Equations (11) and (12) are partial derivatives of g with respect to x and z for a cylinder model.

$$a_x = \frac{\delta g}{\delta x} = \frac{-2xGm}{(x^2+z^2)^2} \quad (11)$$

$$a_z = \frac{\delta g}{\delta z} = \frac{Gm(x^2-z^2)}{(x^2+z^2)^2} \quad (12)$$

Using the observed model response from Figure 17, Figure 19 is created to display the partial derivative plots with respect to each parameter. The derivative functions with respect to x and z are calculated using equations (11) and (12), respectively. The partial derivative functions concatenate to form the A matrix.

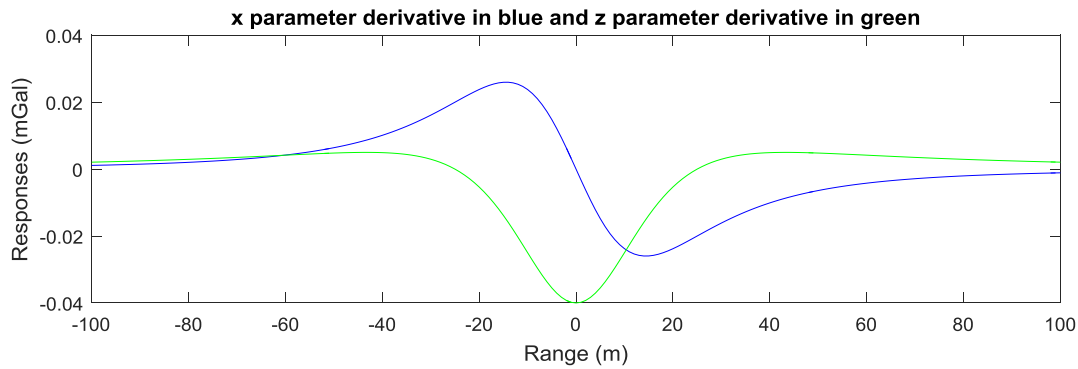


Figure 19: Plots of the x and z derivative are shown in blue and green, respectively. The x and z derivative arrays are the result of equation (11) and equation (12), respectively. (See appendix 8 – axazPlots)

2.2 Inversion operation and the basin of attraction

After each iteration, input parameters are affected by the inversion process and are appended by constants from the parameter change matrix, in an attempt to minimize the misfit. With regards to geophysical inversions; this means creating an initial model, generating a response that better fits the observed response, then potentially creating a more viable geological solutions. This however is not the case all the time; the inversion process can succumb to chaos, local minima or diverge to unrealistic limits or impractical parameter values. A matrix grid, or solution space, can be used to keep track of an array of initial models or parameter sets. This aids in identifying initial models that have descended to local minima or exited from accepted bounds. The solution space can be visualized as an image in parameter-coordinates to investigate emergent patterns or trends. This approach can also be applied to geological models with one or more buried bodies.

2.3 Analysing the basin of attraction

The nature of the inversion process was introduced in section 1.3.6 and illustrated in Figure 14. If the inversion process is linear in nature, the solutions can be considered attractors and result in infinite basins of attraction. This is due to a single scalar difference between the observed and the calculated models. This results in all initial models in a sample space converging to the observed model. If the inversion processes is non-linear in nature then it would exhibit strange attractors and finite basins of attraction. With respect to the geophysical inversion process; the observed

model parameter values are the attractors around which basins of attractions should form. The basin of attraction is then defined as the sets of initial model parameter values within the solution space whose inversion trajectories tend toward the attractors (Sprott and Xiong, 2015). With an established method of calculating the dimension of an image, see section 1.3.3, the dimension of the basin of attraction can be studied as a function of the initial parameters of the observed models. The shape, amplitude and period of observed gravitational profiles are dependent on the parameters of the observed model. Changing the parameter values changes the observed response and would therefore change the basin of attraction in the solution space.

The gravitational response due to a buried cylinder is the result of the initial parameter values, as defined in equation (9). There are trends which can be observed as changes are made with respect to position (x parameter value) or depth (z parameter value). In Figure 20, all observed model parameters are kept constant with the exclusion of the depth parameter. The depth of a buried cylindrical model is increased and at each point the associated response is generated.

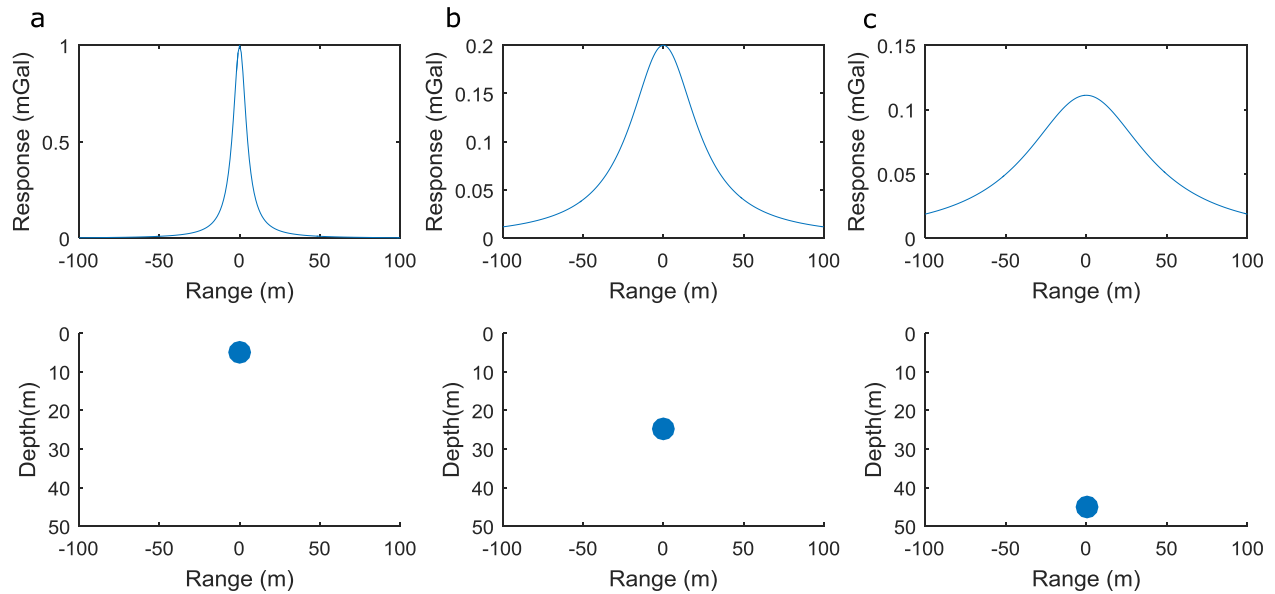


Figure 20: Gravitational responses for associated synthetic models. Observed model responses for buried cylinder models at varied depths of 5 m, 25 m and 45 m in a, b and c, respectively. Each response is the result of a buried cylinder model, as established in equation (9). (See appendix 12 - ResponsePlusModel)

As the depth of the cylinder model is increased the amplitude, absolute magnitude, of the response is decreased and the anomaly is made wider. In measured gravity data, with noise

present, a deeper buried model can result in the lower amplitude being harder to distinguish. The position of a buried model affects the peak of an observed response. This is important if other models are visible in the range. In Figure 21, all observed model parameters are kept constant except for of the location along the range. The separation of two buried cylindrical models is varied and at each point the associated responses are generated.

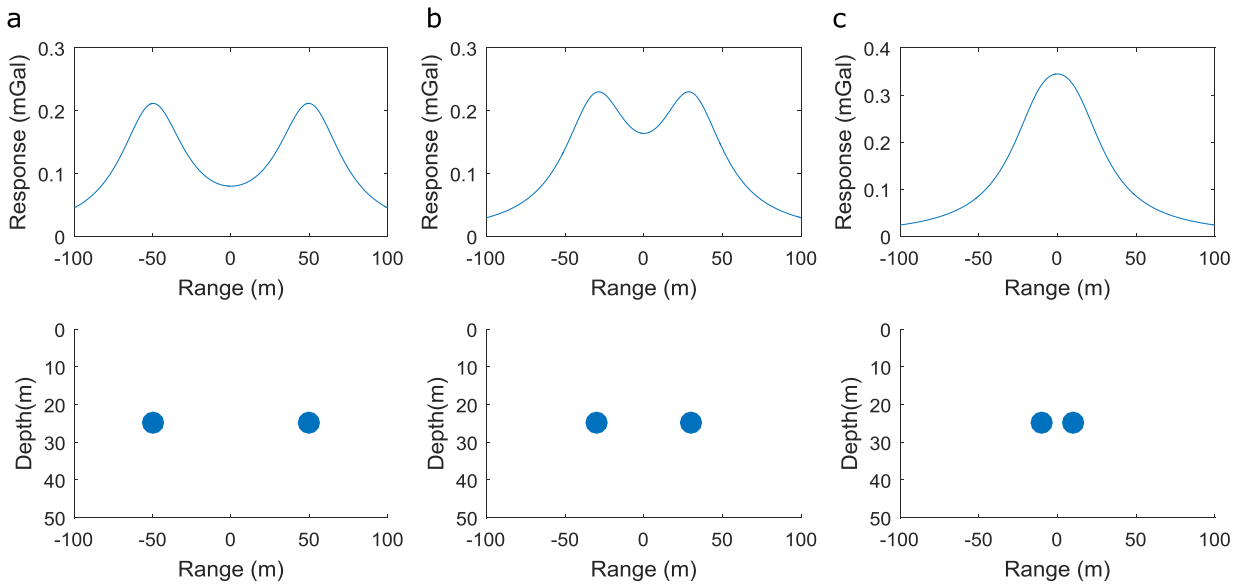


Figure 21: Gravitational responses for associated synthetic models. Observed model responses of buried cylinder models at varied separations. The separations between the centre of the two buried bodies in the model is equal to 50 m, 30 m and 10 m in *a*, *b* and *c*, respectively. Each response is the result of a buried cylinder model, as established in equation (9). (See appendix 12 - ResponsePlusModel)

As two bodies are brought together it gets more difficult to discern each respective anomaly in its entirety. In systems with many buried bodies it can be difficult to distinguish individual bodies and responses in the inversion process.

The computing ability of the software or hardware have on these results is also to be considered; the system used for all results is a 32bit computing system. Using a 64bit computing system would increase the register size and result in shorter computation lengths when running the algorithms and constructing models.

CHAPTER THREE - RESULTS

3.1 The inversion process

The Moore-Penrose inversion process, as set out in equation (7), is applied to data sets calculated using buried cylinder models. The gravitational response due to a buried cylinder, equation (9), is used to create an observed and initial response with different model positions and depths. The inversion process then perturbs the calculated model parameter values by an offset as influenced by the misfit calculated in equation (10). Figure 22 shows the observed and calculated models throughout an inversion process and Figure 23 shows the generated associative responses. The observed gravitational response is generated by a body centred in the middle of the range at a depth of 15 m. The initial calculated gravitational response is generated by a body at 15 m along the range and buried at a depth of 32.5 m.

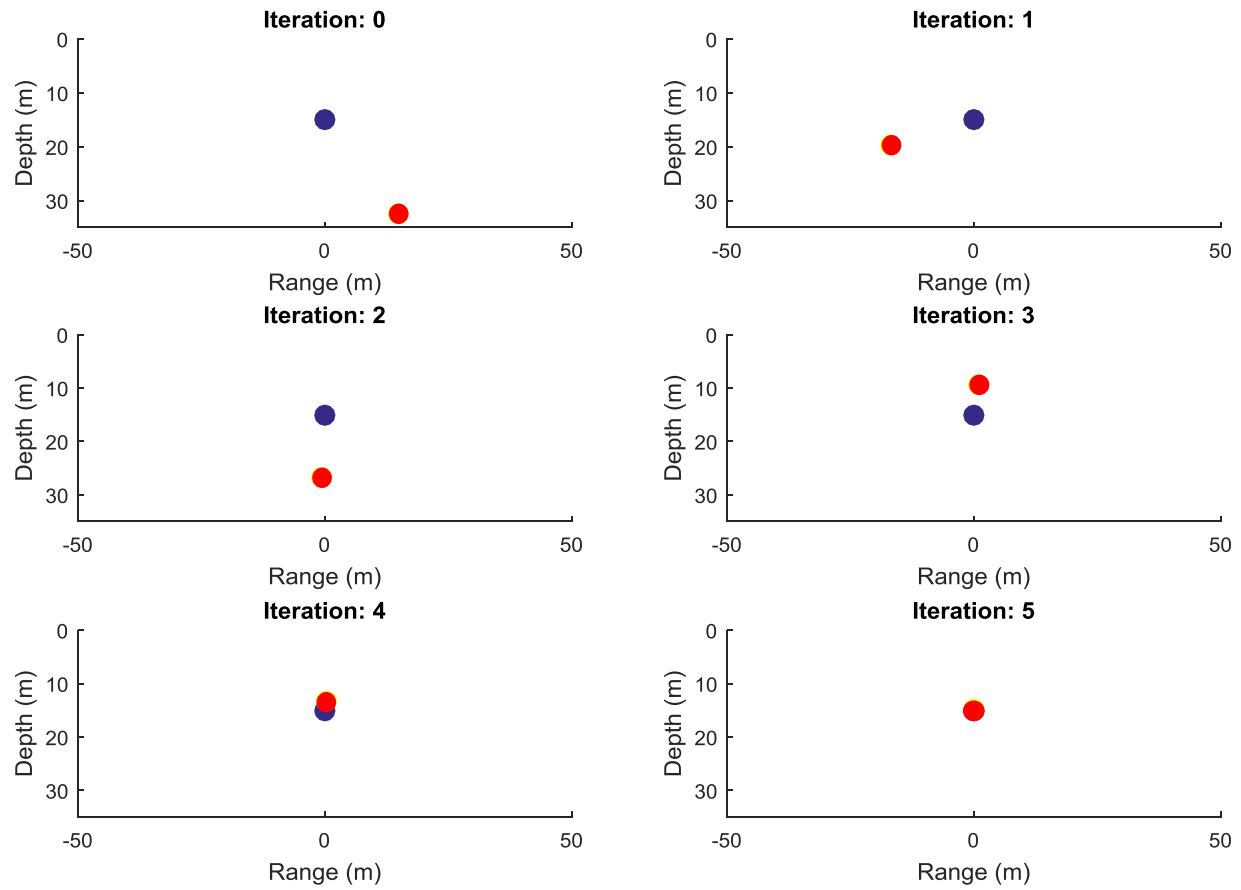


Figure 22: Observation of the calculated and observed models through the Moore-Penrose inversion process as defined by equation (7). The observed cylindrical model is at 0 m along the range and is buried at 15 m. The calculated cylindrical model starts at 15 m along the range and is buried at 32.5 m. The models are tracked through the inversion iterations; the blue model is the observed model and the yellow model is the calculated model. (See appendix 2 – InversionStepsPlusModel)

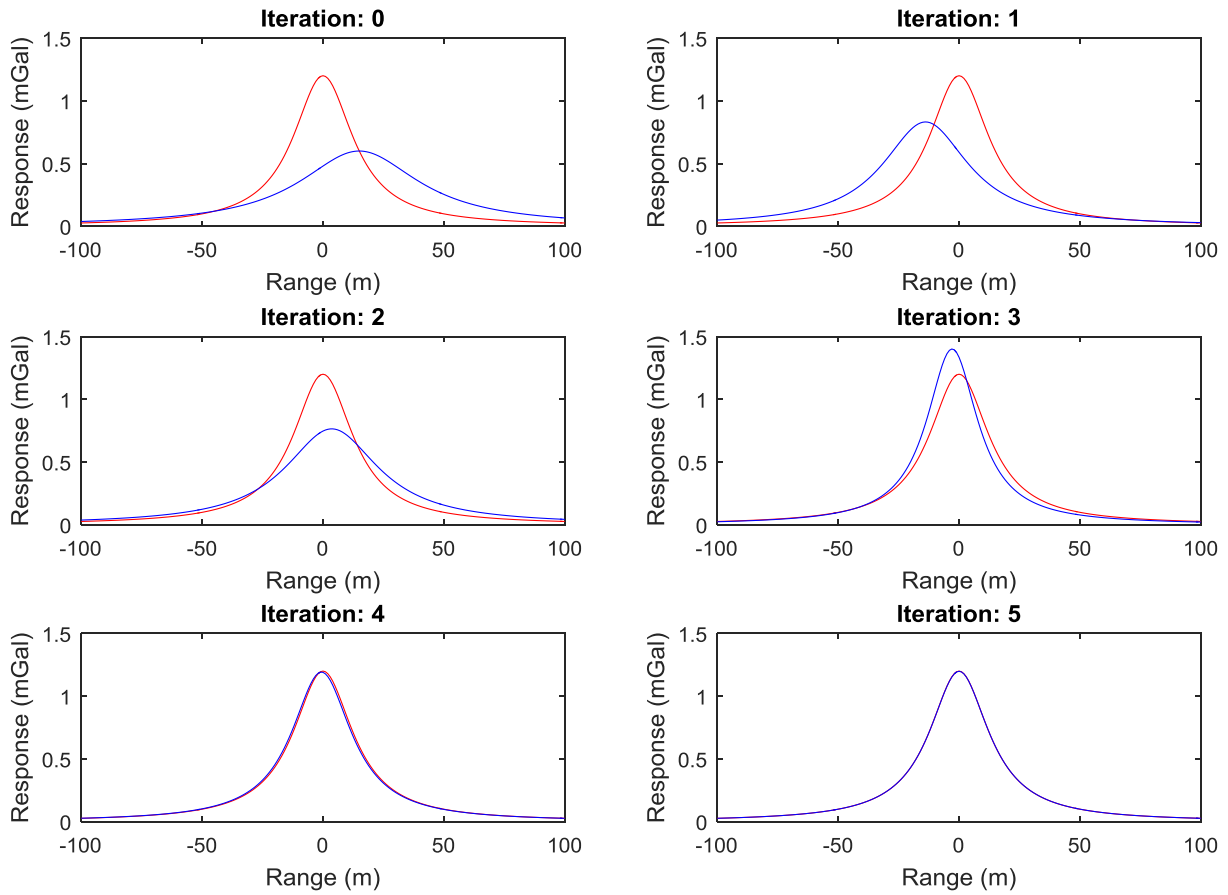


Figure 23: Observation of the calculated and observed responses during the inversion process at each iteration. The observed cylindrical model is set at 0 m along the range and buried at a depth of 15 m. The calculated cylindrical model is set at 15 m along the range and buried at a depth of 32.5 m. Each response is the result of a buried cylinder model, as established in equation (9), with different parameter values. (See appendix 2 – InversionStepsPlusModel)

Numerically, the inversion process can be tracked as in Table 1. At each iteration the current parameter values and associated parameter change values are logged. At the last iteration the two calculated parameter values are the same as the observed parameter values. This is shown as the two profiles being indistinguishable from one another as in Figure 23.

Iteration	x	Z	$\Delta p / \Delta x$	$\Delta p / \Delta z$
0	15.00	32.50	-31.91	-12.80
1	-16.91	19.69	16.14	7.13
2	-0.77	26.82	1.63	-17.32
3	0.86	9.50	-0.40	3.80
4	0.46	13.30	-0.46	1.70
5	0.00	15.00	0.00	0.00

Table 1: Parametric summary of the inversion process as observed in Figure 22 and Figure 23. The table tracks the iteration, x parameter value (location), z parameter value (depth), x parameter change value and z parameter change value, respectively.

The misfit between the observed and calculated model responses, as defined in section 2.1, can also be tracked as the calculated model is perturbed. The misfit as a function of iterations is shown in Figure 24.

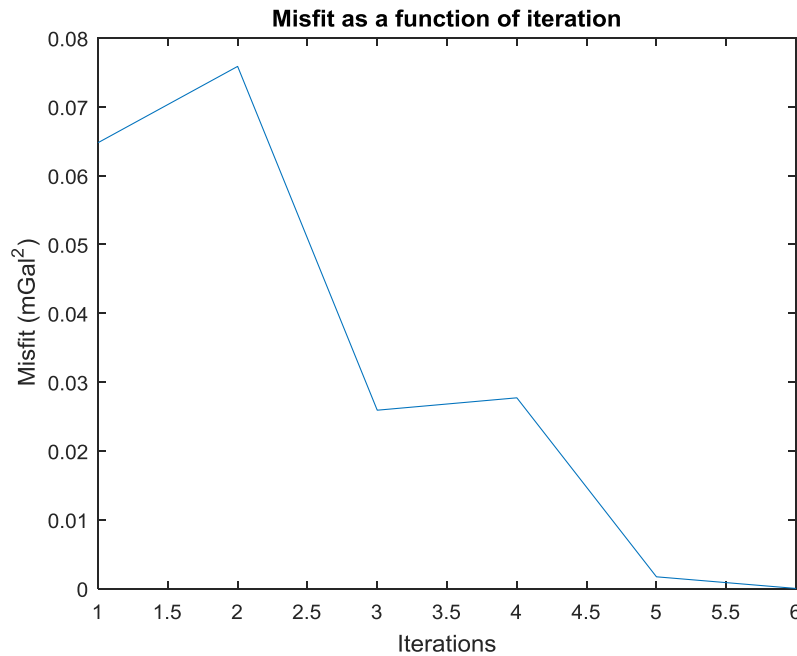


Figure 24: The misfit, as calculated from equation (10), as a function of iteration for the models defined in Figure 22 and Figure 23. (See appendix 2 – InversionStepsPlusModel)

3.2 The basin of attraction

One method to visualize successful convergence for the inversion process is to plot a basin of attraction. Points within the parameter space form a two-dimensional plane and individual points can be colour-mapped by their behaviour under inversion. Convergence is defined as when the initial model parameters tend to parameter values very close, in magnitude, to the observed parameter values. In this section convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. Divergence is then a result of initial models tending toward unrealistic values, infinity or unstable orbits about the attractors. With these definitions a system is defined using the gravitational response due to a buried cylinder, equation (10), for a model centred at 0 m along the range and buried at 15 m. This model and the associated response are shown in Figure 25.

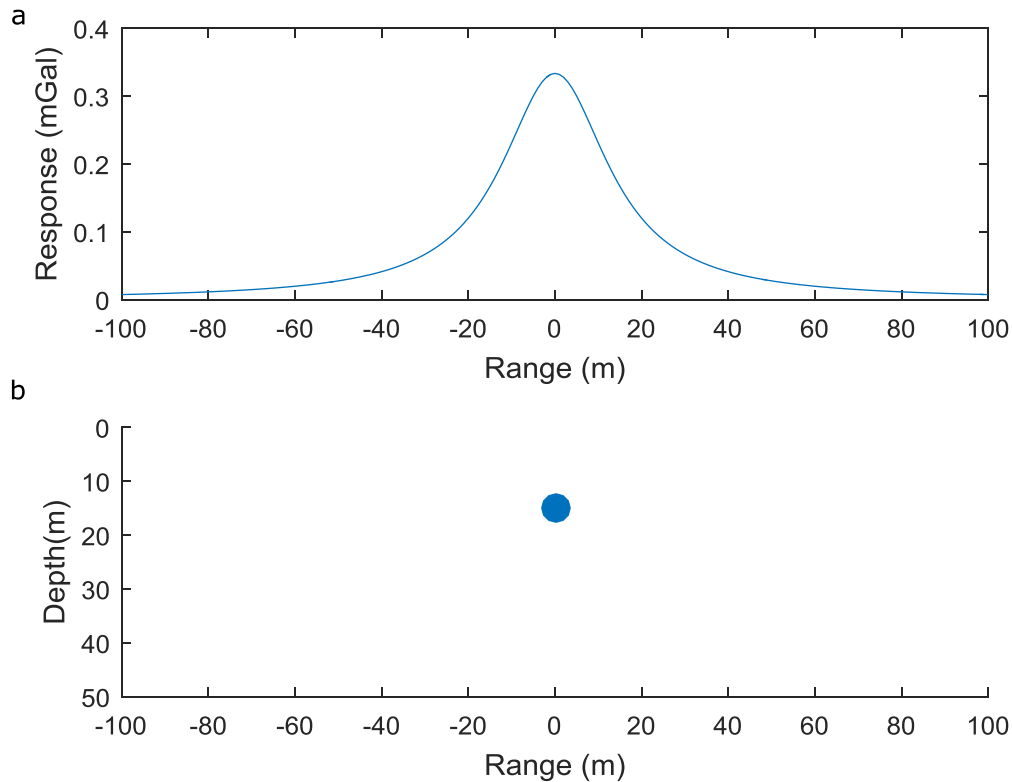


Figure 25: Observed model response, seen in *a* as a result of the model set up in *b*. The observed cylindrical model is set at 0 m along the range and buried at a depth of 15 m. (See appendix 12 - ResponsePlusModel)

Using this system, a basin of attraction can be established in a solution space by reiteratively performing an inversion process over an array of input models. In Figure 26 the Moore-Penrose inversion process, equation (7), is run on all initial models in the solution space.

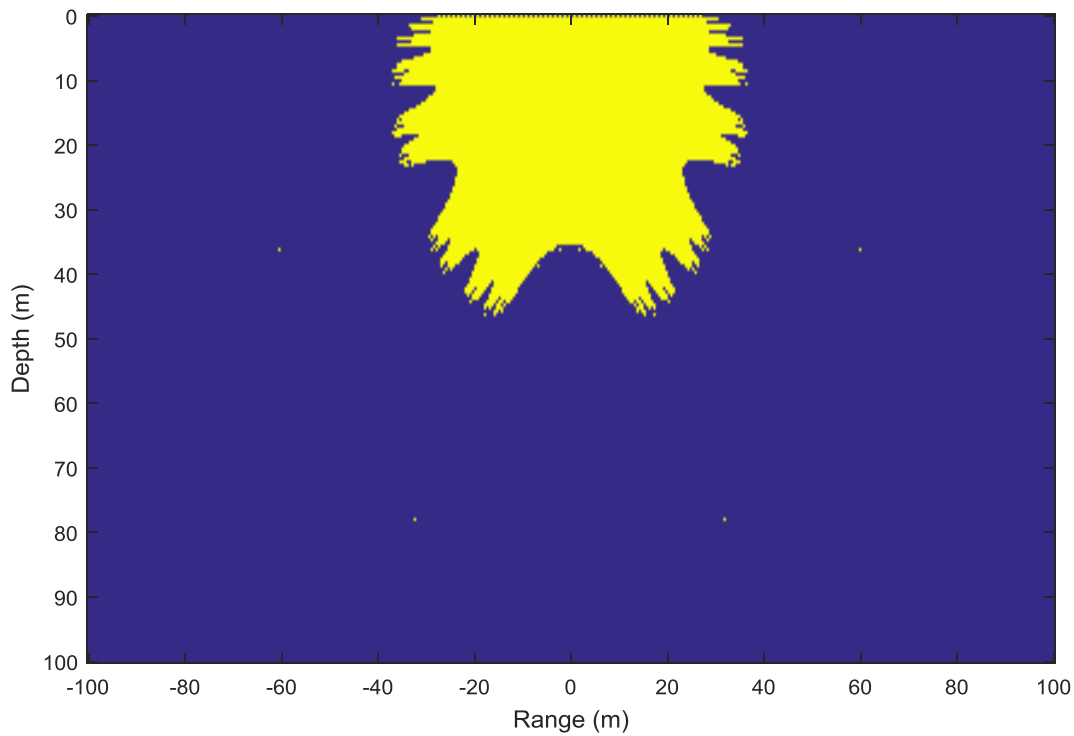


Figure 26: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Moore-Penrose inversion, equation (7), is used on each initial parameter value set in the solution space. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. (See appendix 8 – FracZoom)

In Figure 27 the Ridge-regression inversion, equation (8), is run on all initial models in the solution space.

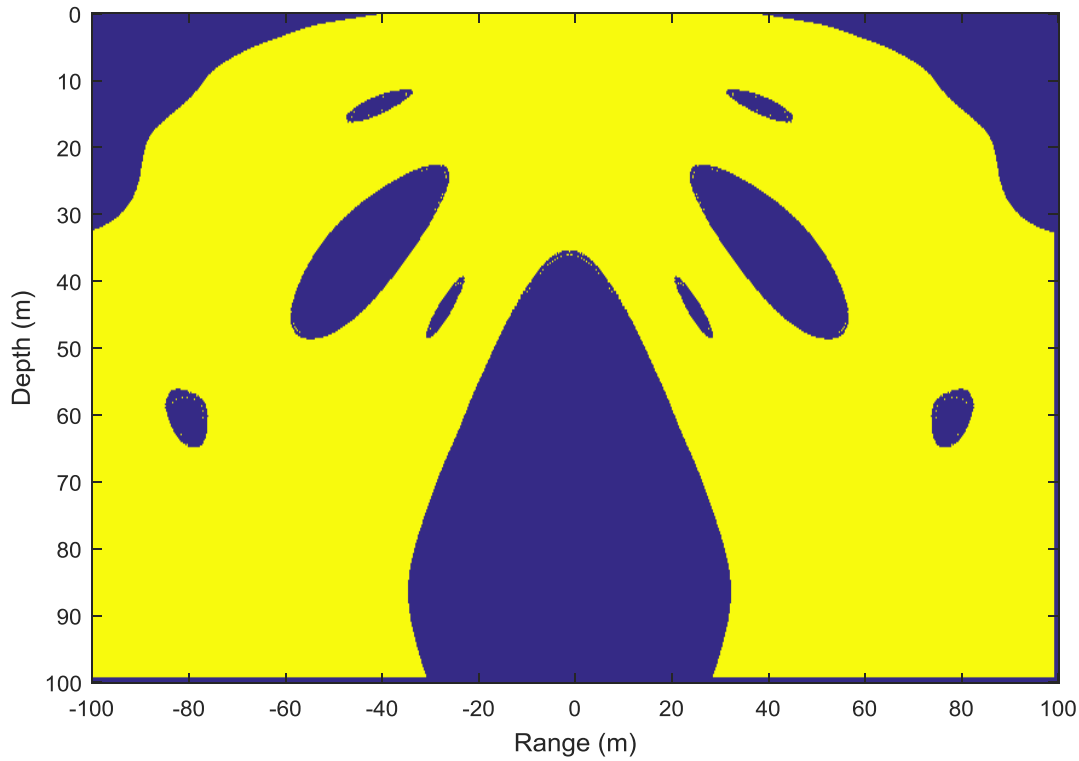


Figure 27: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Ridge-regression inversion, equation (8), is used on each initial parameter value set in the solution space. Yellow indicates initial parameters which converged under inversion and blue indicates divergence from observed model. (See appendix 8 – FracZoom)

In Figure 27 and the Ridge-regression method, equation (8), the damping factor k is varied throughout the inversion process. At each iteration the misfit, as defined in equation (10), is compared to the previous iteration's misfit. If the misfit has grown, the k parameter value is doubled, and if the misfit has decreased, the k parameter value is halved. The k factor starts at a user defined constant but generally the constant has an initial value of 2.

The basins of attraction can also be created when the initial models in the solution space instead track the amount of iterations to meet convergence conditions. The basins of attraction created this way don't change their shape and maintain their fractal properties. Figure 28 is the result of tracking the amount of iterations to reach convergence using the Moore-Penrose inversion method, see equation (7).

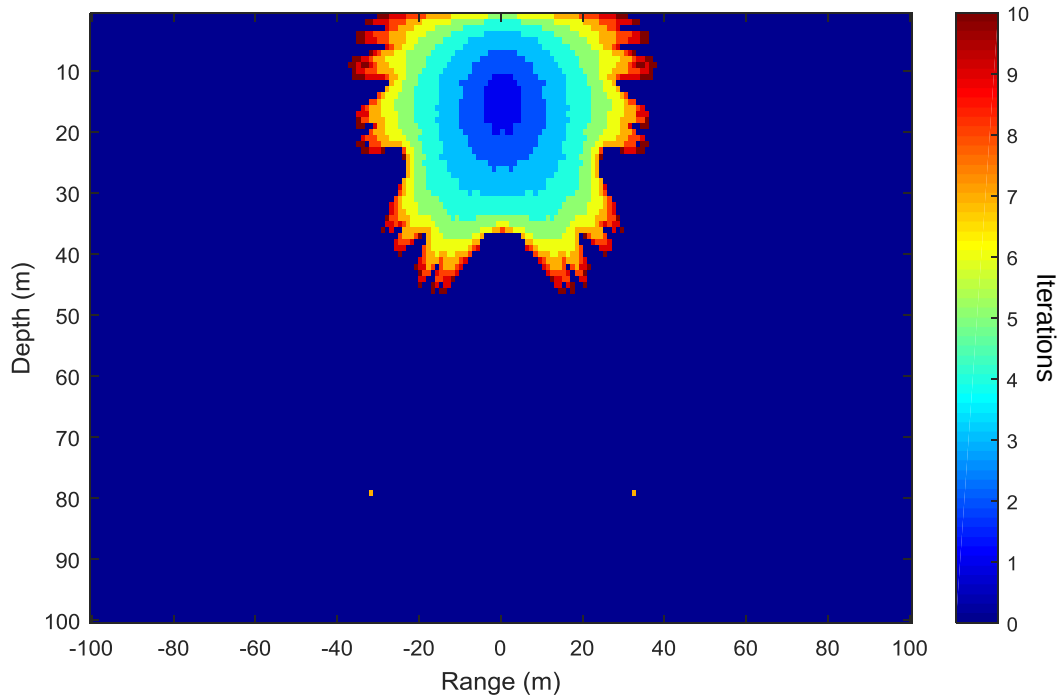


Figure 28: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. Pattern coloured by the amount of iterations to converge. The dark blue background are initial models that did not converge to the observed model. The colorbar indicates the amount of iterations to converge. (See appendix 7 – BasinOfAttractionKIterationsTracked)

When the boundary of the basin of attraction is enlarged in Figure 26, the boundary shows evidence of self-similarity as introduced in section 1.3.2. Figure 29 shows the original basin of attraction and indicates the area of interest which is focused upon.

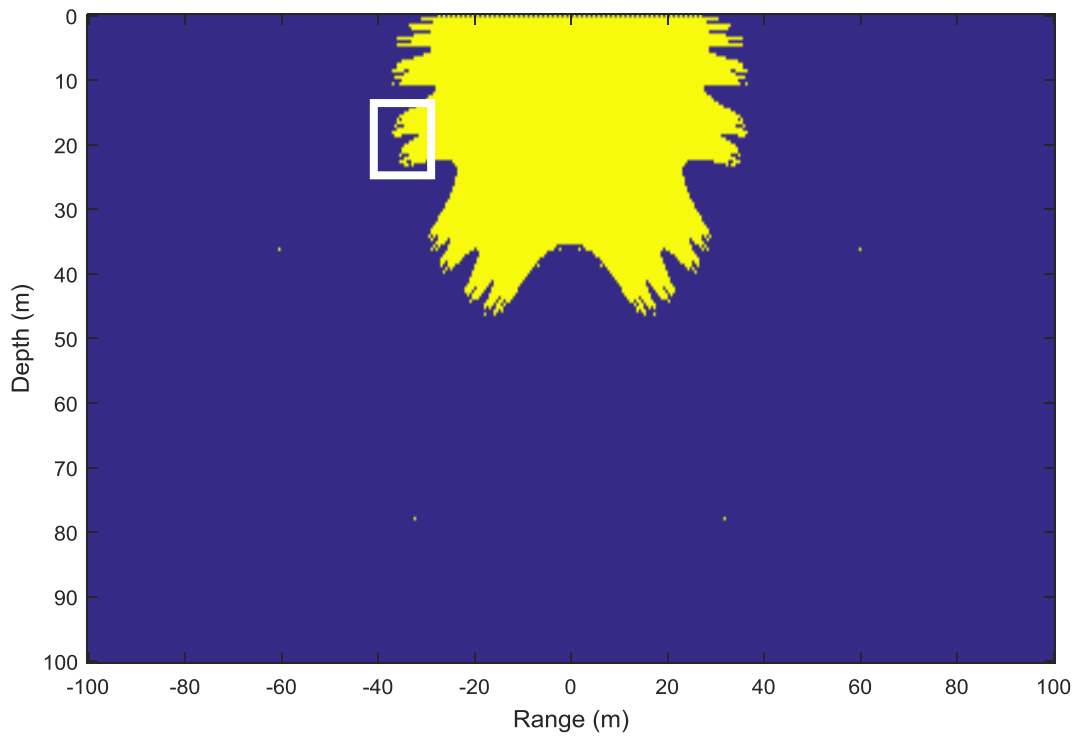


Figure 29: Basin of attraction using the observed cylindrical model as set at 0 m along the range and buried at a depth of 15 m. Yellow indicates initial models which converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. The white box highlights the area which will be investigated and enlarged to investigate self-similarity. (See appendix 8 – FracZoom)

The area identified in Figure 29 is expanded upon repeatedly in Figure 30 and the self-similarity becomes evident. This shows that the basin of attraction is fractal in nature and will have a fractal dimension.

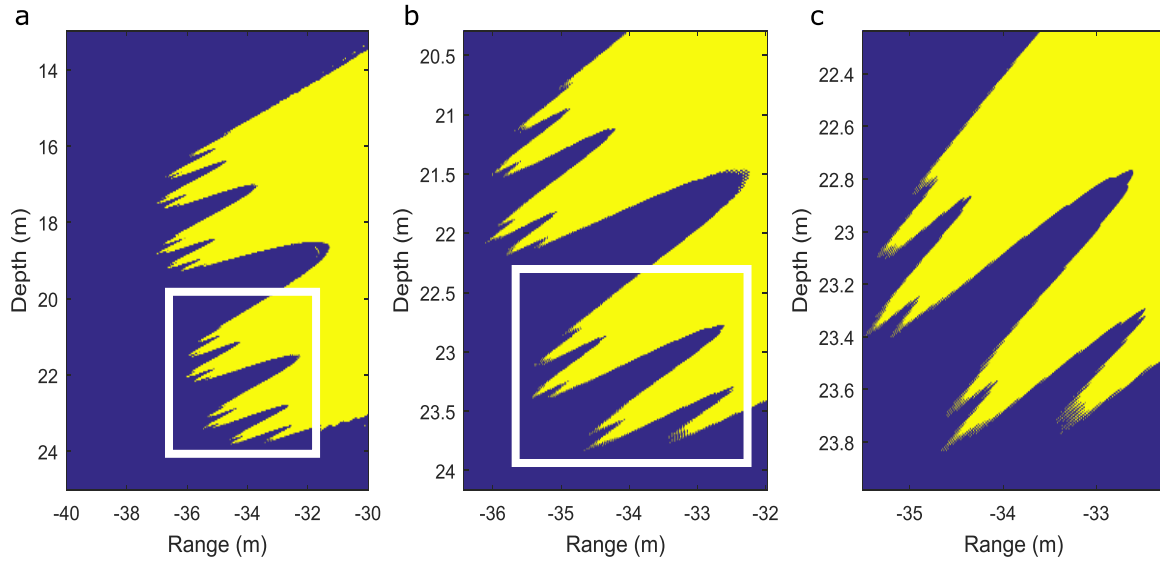


Figure 30: Basin of attraction using an observed model set at 0 m along the range and buried at 15 m. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7), is used on each initial model in the solution space. The white box highlights the area which will be enlarged. *a* is the result of enlarging the area identified in Figure 29. Progressing further; *b* and *c* are the result of enlarging the areas identified in *a* and *b*, respectively. At each enlargement the same pattern is continuously observed. This is a result of “self-similarity” as defined in section 1.3.2. (See appendix 8 – FracZoom)

Since the basin of attraction is fractal in nature, the fractal dimension can be calculated, using the box-counting dimension as set up in equation (3), introduced in section 1.3.3. Using the same process as illustrated in Figure 8, the box-counting method can be applied to the basins of attraction, created from the inversion tracking introduced in this section. Since the box-counting method only tracks variance within a box, the boxes are run over the boundary of a basin. Figure 31 shows an example of how the box-counting method might be applied to the boundary of the basin of attraction, as created in Figure 26.

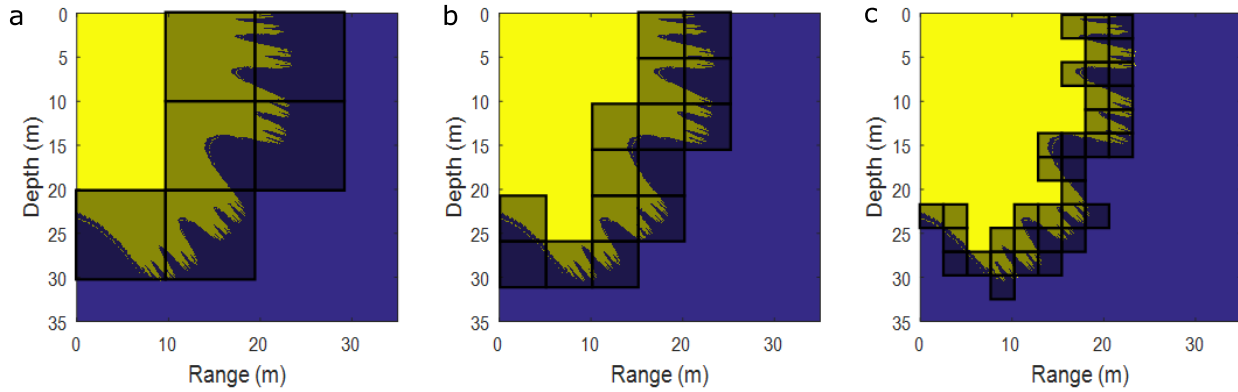


Figure 31: An example of how the box-counting can be applied to the boundary of the basin of attraction. Basin of attraction using an observed model set at 0 m along the range and buried at 15 m. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7), is used on each initial model in the solution space. The box size is decreased from *a* to *c* and at each instance the amount of boxes is compared to the size of the respective box to investigate the dimension of the basin. (See appendix 8 – FracZoom)

3.3 Analysis of the basin of attraction and fractal dimensions

Following the methodology introduced in section 2.3, the original parameters of the observed model can be altered, to change the attractors, and basins of attraction can be constructed. In this section the parameter values of an observed single buried cylinder model are varied, the basins of convergence plotted and the dimensions calculated from the associated response as given in equation (9). Convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. The fractal dimensions are established with the box-counting method defined in equation (3).

3.3.1 Varied depths of a single cylinder source using the Moore-Penrose inversion process

Figure 32 shows the basins of attraction, as a result of varying the observed model's depth, using the Moore-Penrose inversion process as defined in equation (7). The systems used are for a single buried cylindrical model.

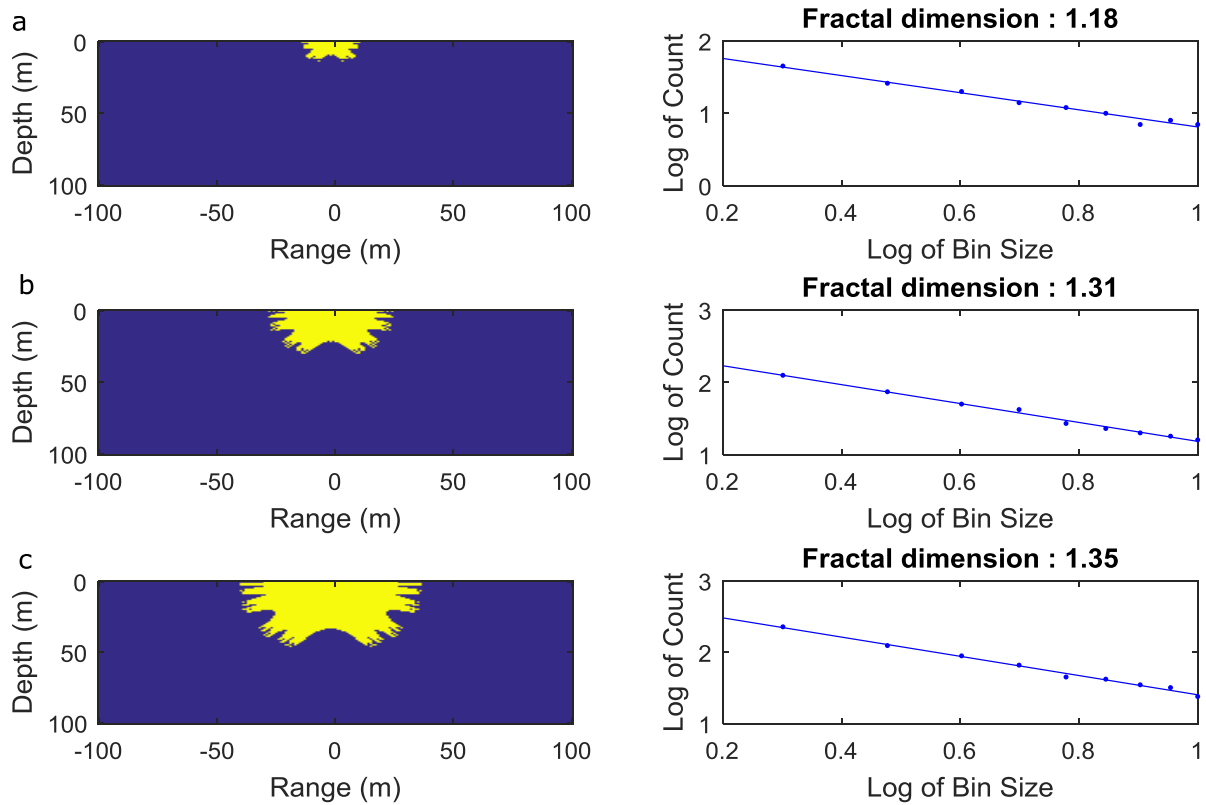


Figure 32: Basins of attraction and dimension calculations for a single cylindrical body centred at 0 m and buried at various depths. *a*, *b* and *c* represents depths of 5 m, 10 m and 15 m, respectively. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Moore-Penrose inversion, equation (7) is used on each initial model in the solution space. (See appendix 6 – BasinsOfAttractionsAndDimensionsThreeImagesK)

Figure 32 shows that, initially at least, as the depth of the cylinder model is increased, the basin of attraction grows larger and the fractal dimension changes. This is a result of the response changes illustrated in Figure 20.

3.3.2 Varied depths of a single cylinder source using the Ridge-regression inversion process

Figure 33 shows the basins of attraction, as a result of varying the observed model's depth, using the Ridge-regression inversion process as defined in equation (8). The systems used are for a single buried cylindrical model.

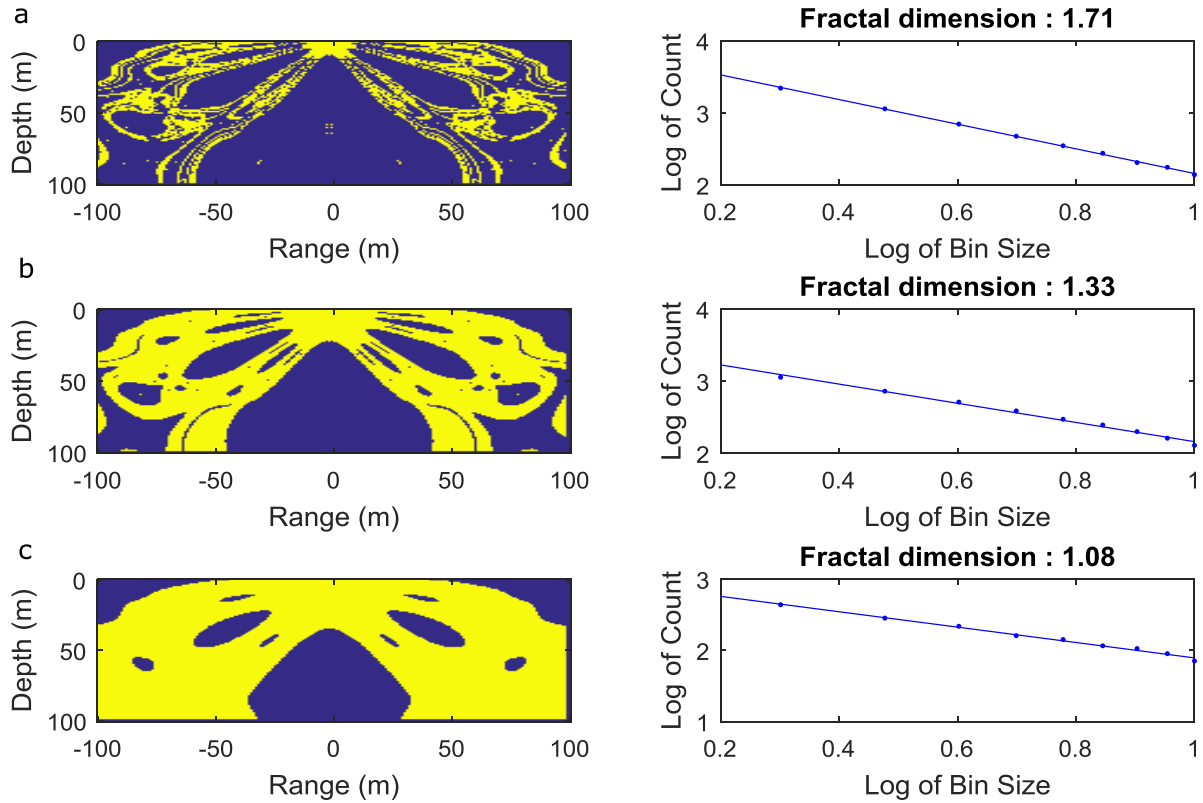


Figure 33: Basins of attraction and dimension calculations for a single cylindrical body buried at various depths. *a*, *b* and *c* represent depths of 5 m, 10 m and 15 m, respectively. Yellow indicates initial parameters that converged under inversion and blue indicates divergence from the observed model. The Ridge-regression inversion, equation (7) is used on each initial model in the solution space. (See appendix 6 – BasinsOfAttractionsAndDimensionsThreeImagesK)

Figure 33 reinforces that, initially at least, as the depth of the cylinder model is increased, the basin of attraction grows larger and the fractal dimension changes. This is a result of the response changes as illustrated in Figure 20.

3.3.3 Damping coefficient's effect on the basin of attraction and convergence conditions

The damping constant k can be kept constant in the ridge-regression inversion as defined in equation (8). Figure 35 – 41 show the basins of attraction at varied damping constants. All systems maintain the same observed parameter values. All models are for a single buried cylindrical model, buried at 10 m and centred about the range at 0 m as seen in Figure 34.

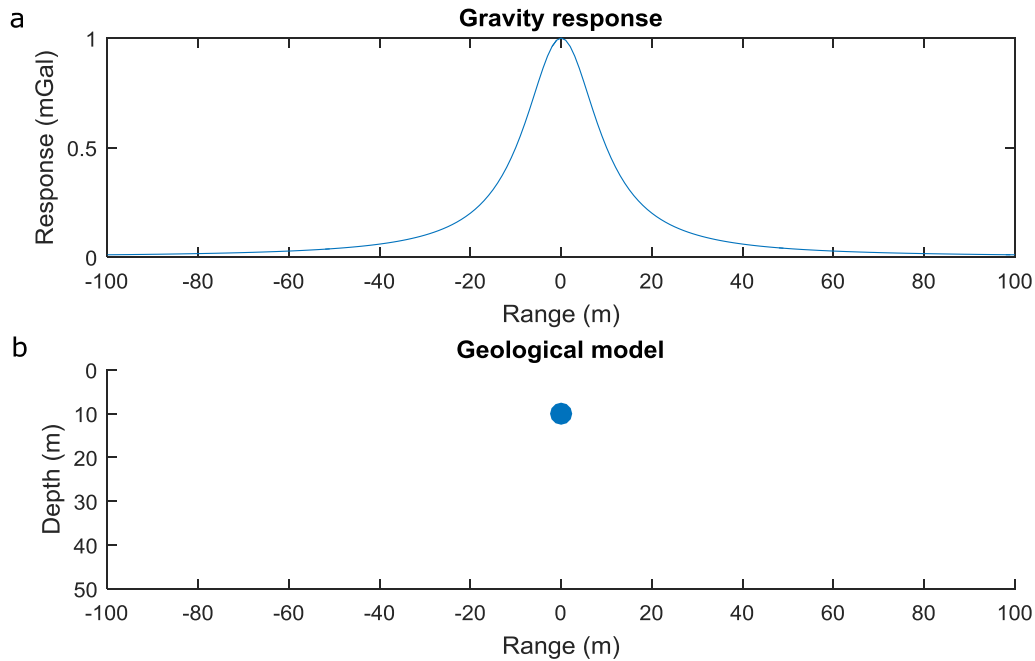


Figure 34: Observed model response, seen in *a* as a result of the system set up in *b*. The observed cylindrical model is set at 0 m along the range and buried at a depth of 10 m. (See appendix 12 - ResponsePlusModel)

The figures chosen here are the figures which illustrated the most easily discernible changes in the shape of the basin of attractions. This is done to demonstrate the affect the damping factor has on the inversion process and subsequent basin of attraction.

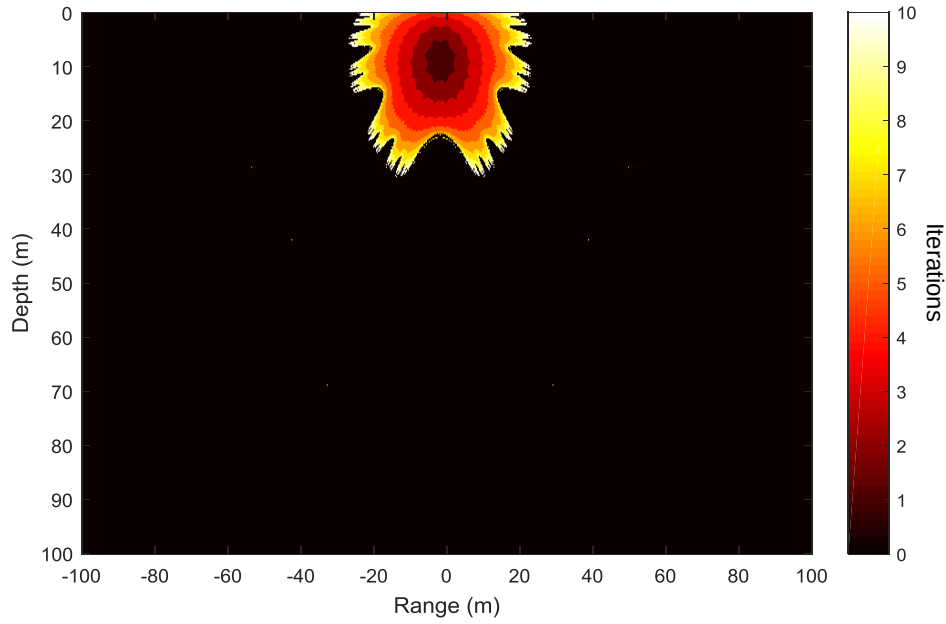


Figure 35: Basin of attraction using a constant damping coefficient of 0 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

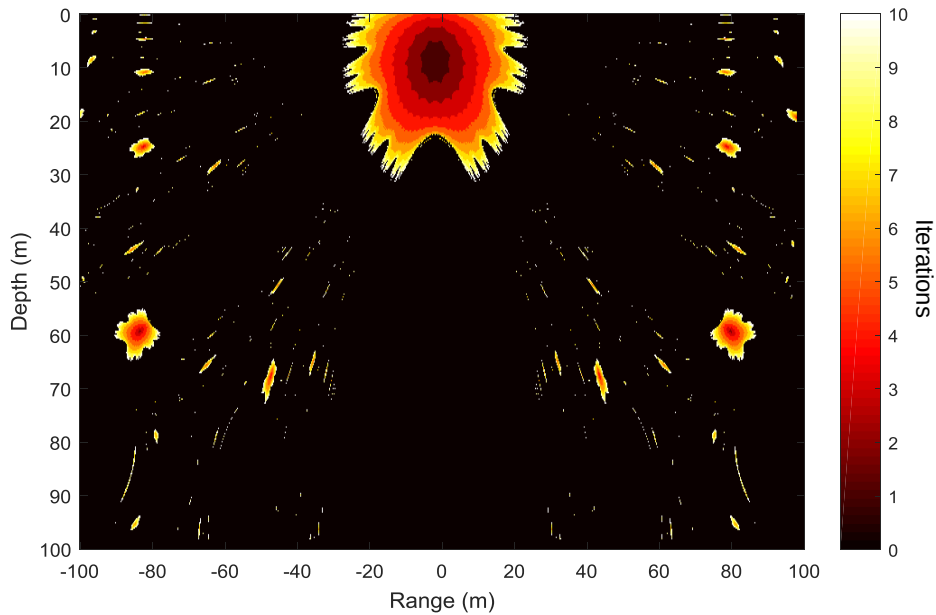


Figure 36: Basin of attraction using a constant damping coefficient of 2 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

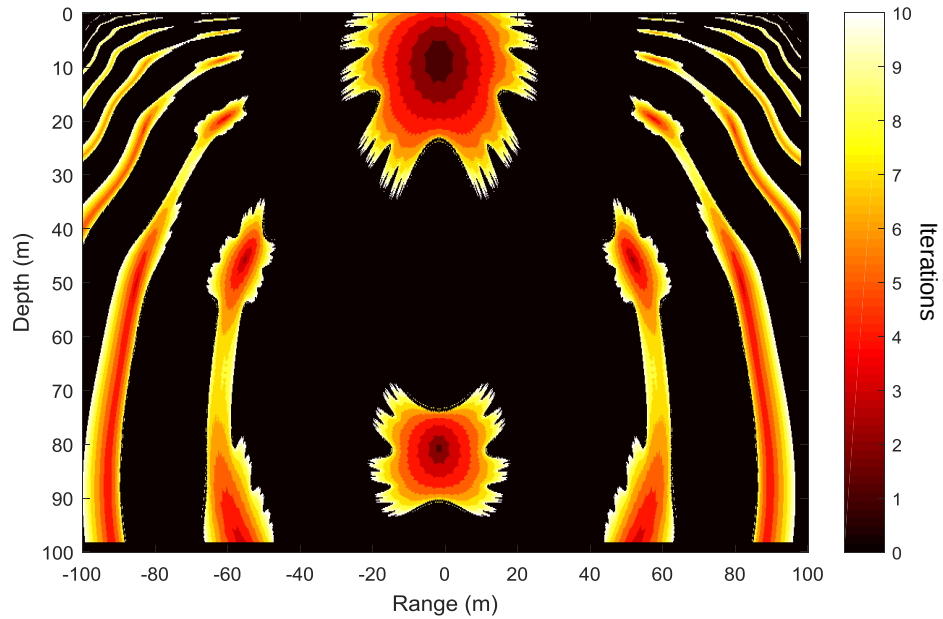


Figure 37: Basin of attraction using a constant damping coefficient of 8 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

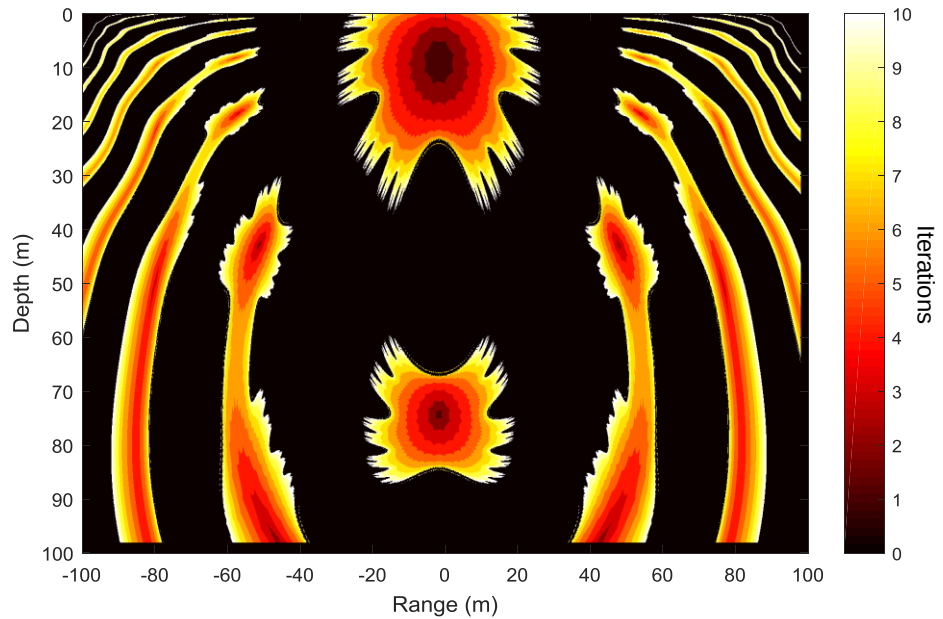


Figure 38: Basin of attraction using a constant damping coefficient of 10 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

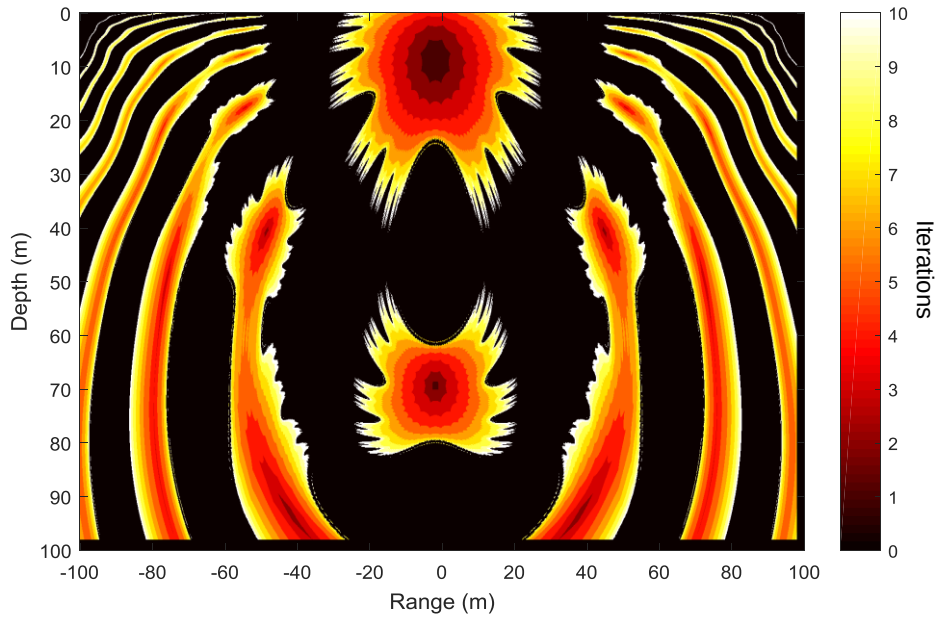


Figure 39: Basin of attraction using a constant damping coefficient of 12 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

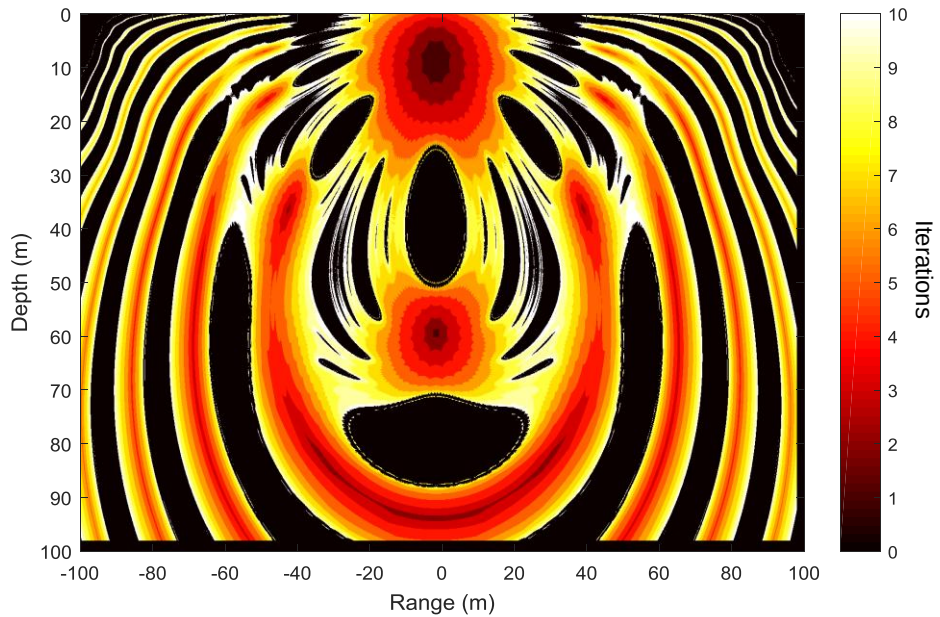


Figure 40: Basin of attraction using a constant damping coefficient of 18 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

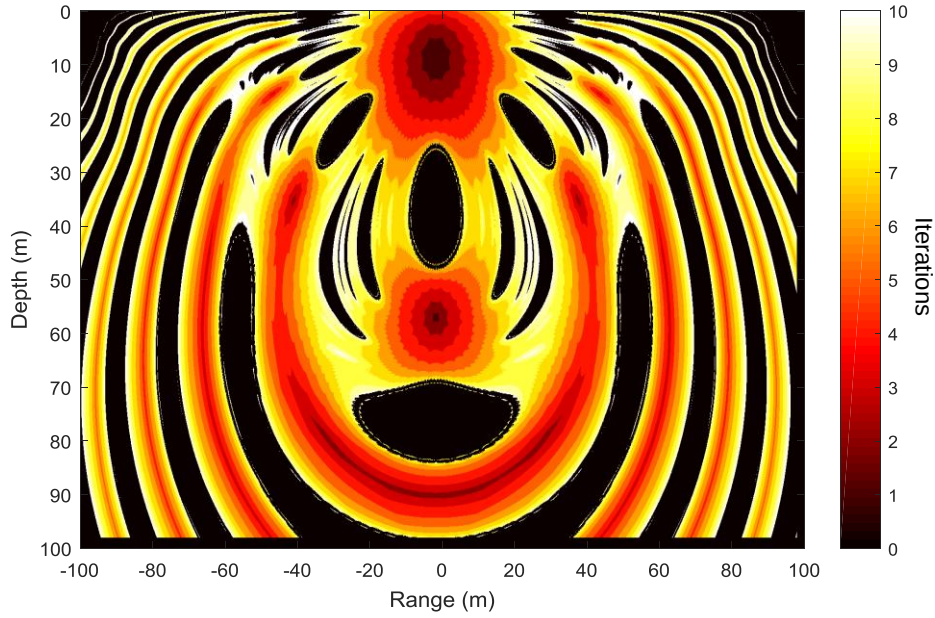


Figure 41: Basin of attraction using a constant damping coefficient of 20 in the Ridge-regression inversion process as defined in equation (8). The observed model is defined in Figure 34: a cylindrical model set at 0 m along the range and buried at a depth of 10 m. the colorbar indicates the amount of iterations to converge and the black background colour indicates parameter sets where initial models did not converge to the observed model. (See appendix 7 – BasinOfAttractionKIterationsTracked)

As the damping factor is increased the basin of attraction grows larger. This is not a result of altering the model response but the parameters of the Ridge-regression inversion process in equation (8). The basin of attraction does not grow from a singular point solely, but instead, initial models which originally might have diverged now instead meet convergence conditions.

3.4 The scaling factor's effect on the inversion process

Another modification on the existing Moore-Penrose inversion is the introduction of a scaling factor. A scaling constant C is defined as a scalar multiple acting on the difference matrix, defined before as D . The modified equation can be written as:

$$dP = (A^T A)^{-1} A^T (CD) \quad (13)$$

To investigate the function this scaling factor plays in the inversion process, the inversion can be tested when subjected to varied scaling factors (Cooper, 2001). The scaling factor acts on the discrepancy matrix, amplifying or diminishing its effect. The scaling factor perturbs the system

and allows for the chaos in the system to be more easily observed. Observing the inversion trajectories as a result of altering the scaling factor can aid in understanding chaos' role in the inversion process. The models for a buried cylindrical body and its associated responses used to illustrate this process are shown in Figure 42.

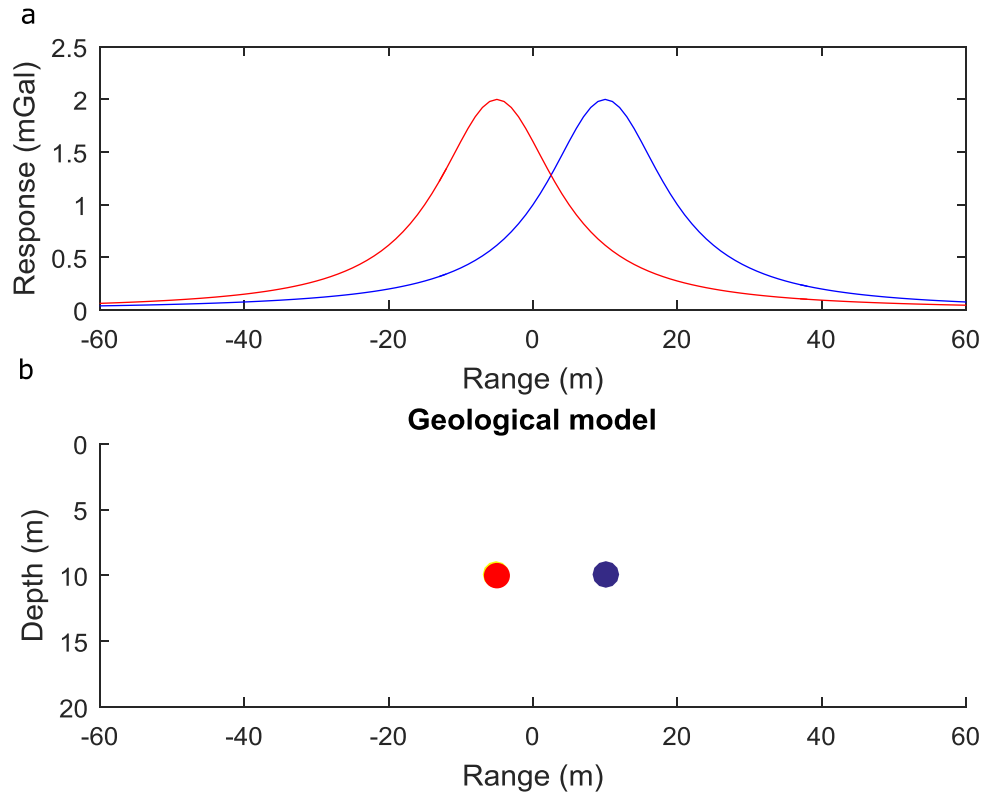


Figure 42: Gravitational responses, in *a*, for buried cylinder models, in *b*, as defined in equation (9). The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at -5 m along the range and is buried at 10 m. (See appendix 12 – ResponsePlusModel)

Using this model; Figure 43 and Figure 44 show how different scaling factors affect a parameter, as a function of the iterations throughout the modified Moore-Penrose inversion process, see equation (13).

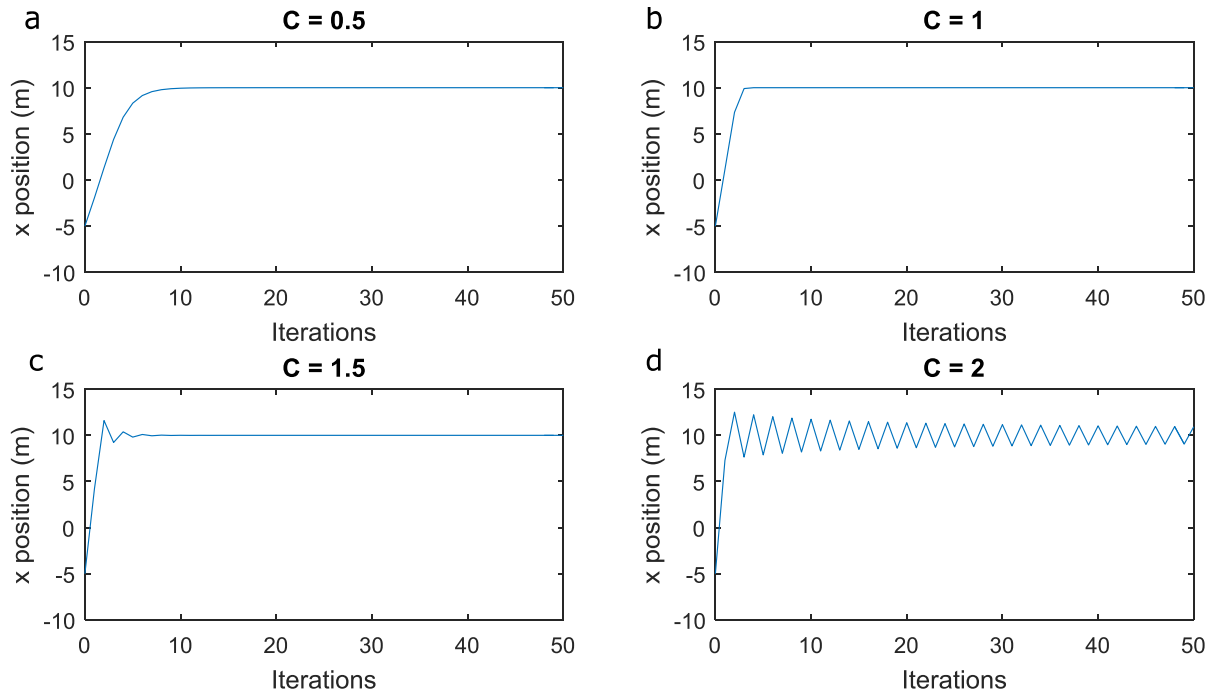


Figure 43: Plots of the x position as a function of iterations under different scaling factors in the modified Moore-Penrose inversion; equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. As the scaling factor increases, the inversion system succumbs to eventual bifurcations at $C = 2$. In *b* the system is unperturbed and the inversion is carried out as normal. From *c* to *d*, as the magnitude of the scaling constant increases and the system grows more perturbed and results in periodicity and a bifurcation. (See appendix 13 – InversionsWithScalingFactorModelIncluded)

In Figure 43 while C was less than 2, the initial model parameter values converged to observed model parameter values. When the scaling factor reaches 2, the model position oscillates periodically about the observed parameter value, creating fixed points, instead of converging as before. In Figure 44 additional bifurcations emerge as the scaling factor is increased further. Bifurcations here are observed as instances of period doubling (the amount of fixed points observed are doubled).

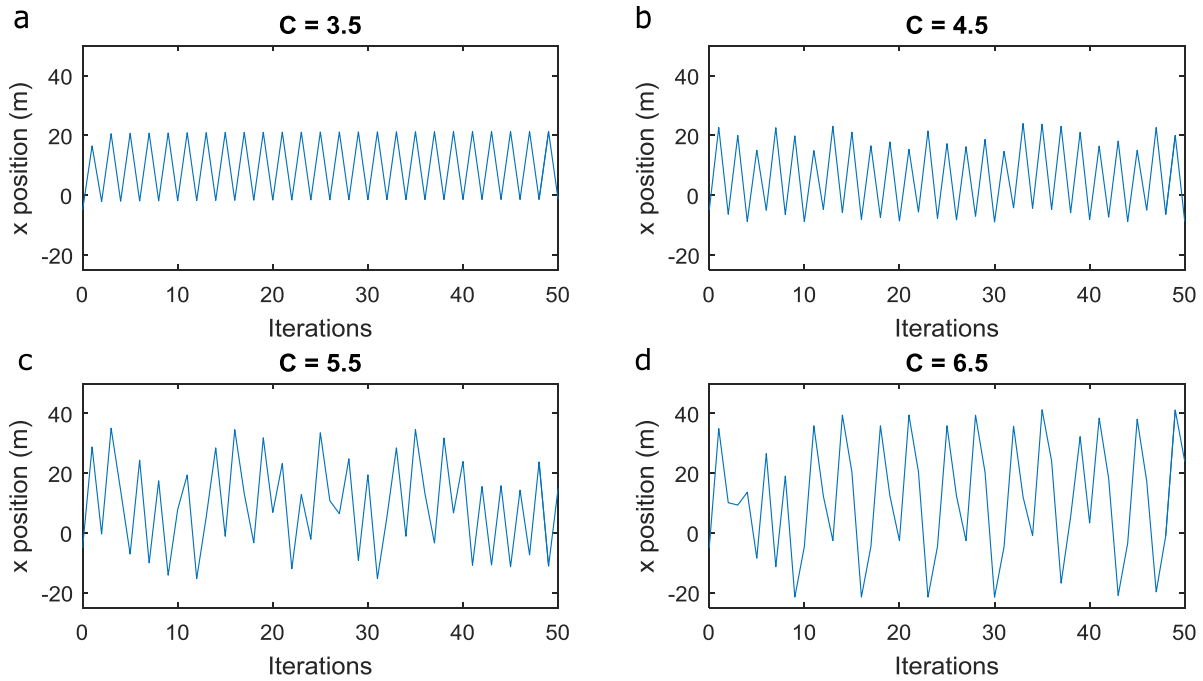


Figure 44: Plots of the x position as a function of iterations under different scaling factors in the modified Moore-Penrose inversion; equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. As the scaling factor increases, the inversion system creates additional bifurcations which can masquerade as apparent chaos. (See appendix 13 – InversionsWithScalingFactorModelIncluded)

Initially it can appear as though the inversion process is behaving purely chaotically, however, by isolating fixed points and associated bifurcations, the bifurcation map can be constructed and the trends better observed. At even larger values of the scaling factor the inversion process can appear to behave chaotically and it becomes challenging to discern possible trends or bifurcations. Figure 45 is the inversion system with a scaling factor of 9. The system appears to behave chaotically and even when patterns appear repetitive, the actual values appear different.

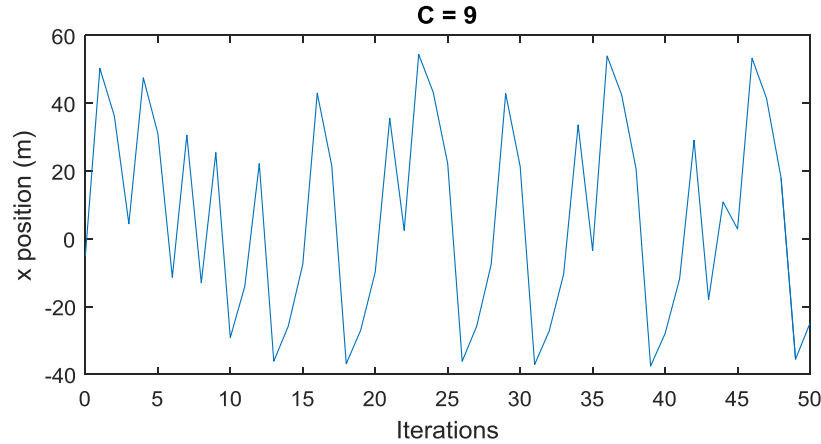


Figure 45: Plot of the x parameter position as a function of iterations using a scaling factor value of 9 in the modified Moore-Penrose inversion; equation (13). Plots of the x position as a function of iterations under different scaling factors in equation (13). The observed model is at 10 m along the range and is buried at 10 m. The initial model is at -5 m along the range and is buried at 10 m. The x position appears to behave chaotically. (See appendix 13 – InversionsWithScalingFactorModelIncluded)

3.4.1 The scaling factor's role in the rate of convergence

Since the scaling factor is able to alter inversion trajectories, see section 3.4, it stands to reason that it might be possible to hasten the rate of convergence. It is probable that this constant would depend on the existing configuration of the inversion and all initial parameters. One might conclude that for every system, there is a particular scaling factor which would result in the fewest amounts of iterations to reach convergence conditions. To observe if there is a correlation, a system is first established, the observed model is set up with the same parameters as in Figure 42. In a similar method to how the basins of attraction are created, the modified Moore-Penrose inversion process in equation (13) is run over an array of initial calculated model parameter values. For each set of initial parameter values, the inversion process is run with an array of scaling factors. The scaling factors which result in the fewest amounts of iterations are recorded as a function of the misfit due to the initial parameter values. Values of the scaling factor are only varied between 0 and 2; since at $C > 2$, bifurcations occur and convergence is more difficult to achieve. A plot of the misfits between the initial model and the observed model as a function of the scaling factors which result in the least amount of iterations to converge is shown in Figure 46.

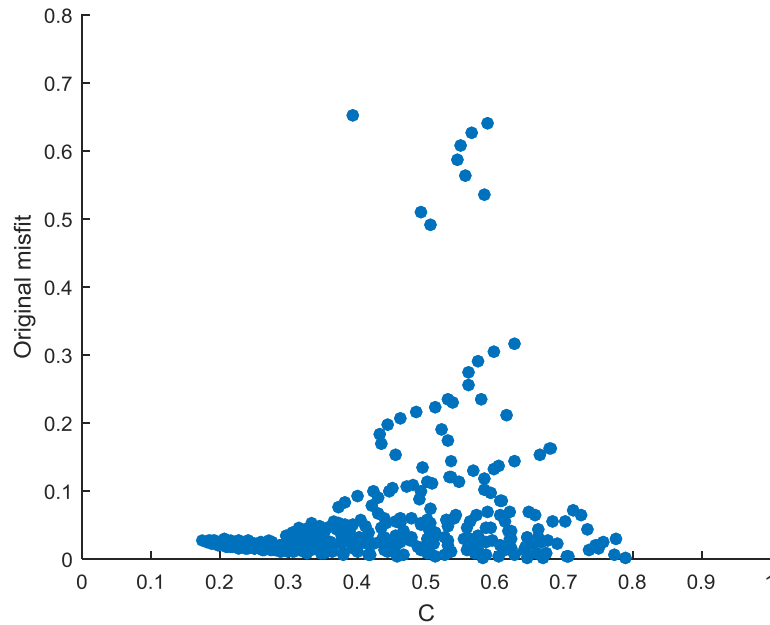


Figure 46: Plots of the original misfit, over a range of different initial parameter values, as a function of the respective scaling factor which would result in the fewest amounts of iterations to converge. The modified Moore-Penrose inversion process, using equation (13), is set up over an array of different initial model positions and depths and for each set; the scaling factor which would result in the fewest amounts of iterations to converge is recorded. The observed model is at 10 m along the range and is buried at 10 m. Convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. (See appendix 14 – MisfitVsC)

Analysis of the results in Figure 46, suggests that no direct relationship exists between the misfit, the scaling factor and the fewest amounts of iterations to convergence. Instead, it is plausible that the initial parameters uniquely define the amount of iterations to converge. It is apt to then investigate how varying the scaling factor, for the same set of initial and observed models, affects the amounts of iterations to converge. In Figure 47, the same observed model as in Figure 42 is used against different calculated models. In each system the inversion is run over a range of scaling factors, and for each one the amount of iterations to converge is recorded.

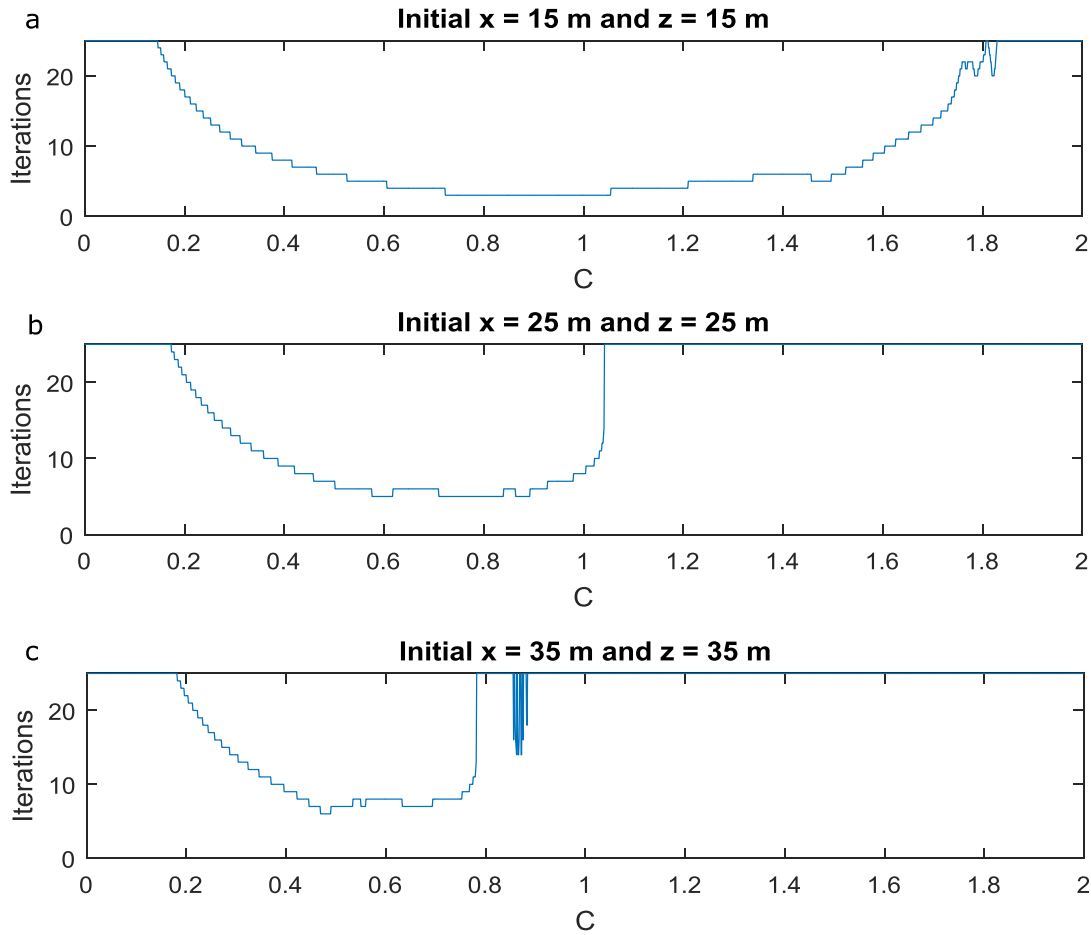


Figure 47: Investigation into the affect the scaling factor has on the amount of iterations to converge. The same observed model is used in all cases but in each case the initial model parameters are changed. The observed model is at 10 m along the range and is buried at 10 m. The position and depth of the initial model is set as 15 m, 25 m and 35 m in *a*, *b* and *c*, respectively. The amount of iterations is capped at 25. Convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model. All inversion processes run using the modified Moore-Penrose inversion in equation (13). (See appendix 15 – CVSILoop)

Figure 47 suggests a trend where the iterations to convergence decrease to a point and then increase as the scaling factor progresses further. The actual scaling factor which reduces the iterations to a minimum is however different depending on the initial model parameters. Subsequently this suggests that the ideal scaling factor per inversion configuration is dependant uniquely on the initial parameters.

Determining the scaling factor C

Since at higher scaling factors, the inversion solutions feature bifurcations, such as in Figure 43, it might be possible to calculate the scaling factor used to produce a given solution trajectory. Using the modified Moore-Penrose inversion, equation (13), to solve for C outright can be difficult. Instead, attempting to solve for C by looking initially at the over-determined situation might prove easier. In an over-determined situation the dataset provides more observed data points than model inversion parameters. In these situations the contributions from each of the individual data points are averaged, as they affect the parameter change matrix. For parameter change matrix that solely operates on the depth of a theoretical cylinder, expressions for the parameter change dP , discrepancy matrix D and the component derivative dg/dz for an over-determined situation can be redefined with equations (14), (15) and (16) (Menke, 1989):

$$dP = \frac{D}{\left(\frac{dg}{dz}\right)}, \quad (14)$$

$$D = \frac{gmz_1}{x_1^2 + z_1^2} - \frac{gmz_2}{x_1^2 + z_2^2}, \quad (15)$$

$$\frac{dg}{dz} = \frac{gm(-z_1^2 + x_1^2)}{(x_1^2 + z_1^2)^2}. \quad (16)$$

If the scaling factor is once more attached to the discrepancy matrix, the parameter change matrix expressed in equation (14) allows for C to be solved using equation (17):

$$C = dP \times \frac{gm \times (x_1^2 - z_1^2)}{(x_1^2 + z_1^2)^2} \times \left(\frac{gm \times z_1}{(x_1^2 + z_1^2)} - \frac{gm \times z_2}{(x_1^2 + z_2^2)} \right) \quad (17)$$

This equation makes it possible to numerically solve for the scaling factor C . Figure 49 and Figure 50 show the operation of using equation (17) over a range of input C values all acting on the same model. The observed and initial models are defined in Figure 48.

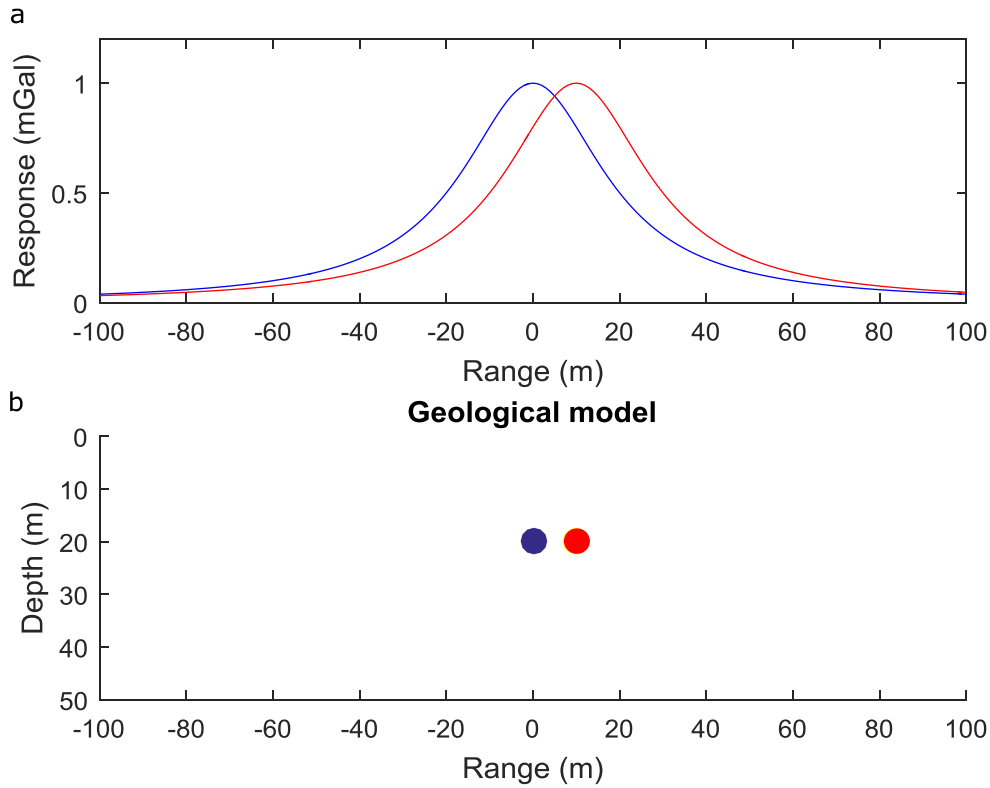


Figure 48: Gravitational responses for associated models. Calculated and observed models in *b* and associated responses in *a*. The observed model is at 0 m along the range and is buried at 20 m. The initial model is at 10 m along the range and is buried at 20 m. The blue model is the observed model and the yellow model is the calculated model. (See appendix 12 – ResponsePlusModel)

The simplified inversion process in equation (14) acts on the system in Figure 48 at different input scaling factor constants. At each iteration the initial model's x location and z depth are altered by the parameter change matrix in equation (14). After a set amount of iterations; equation (17) is used to calculate the scaling constant C from the calculated model's x location and z depth in the last two iterations in the inversion process. In Figure 49, equation (17) is applied after the third iteration of the inversion process. In Figure 50, equation (17) is applied after two hundred iterations of the inversion process; this is so that bifurcations can be observed.

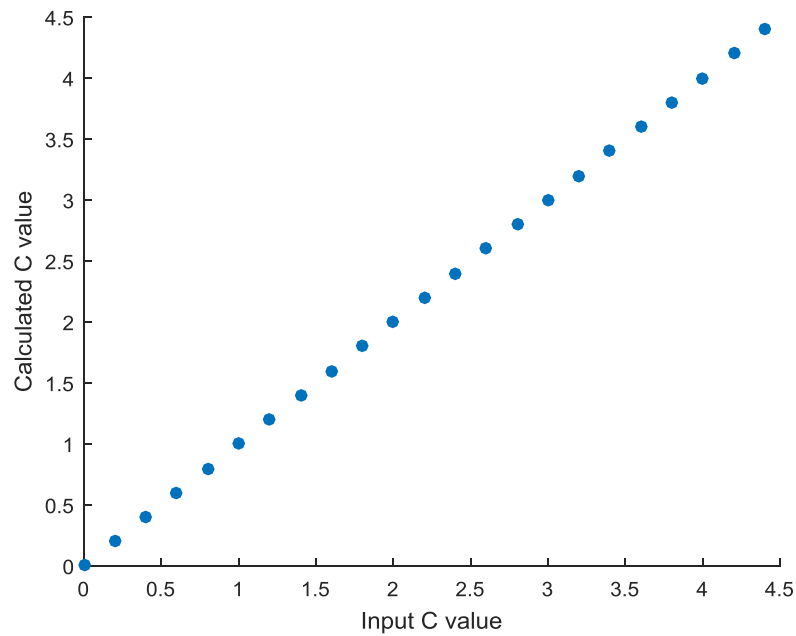


Figure 49: Plot of the calculated scaling factor as a function of the initial input scaling factors. Equation (17) is run after the third iteration to solve for the calculated C value. (See appendix 16 – CsLoop)

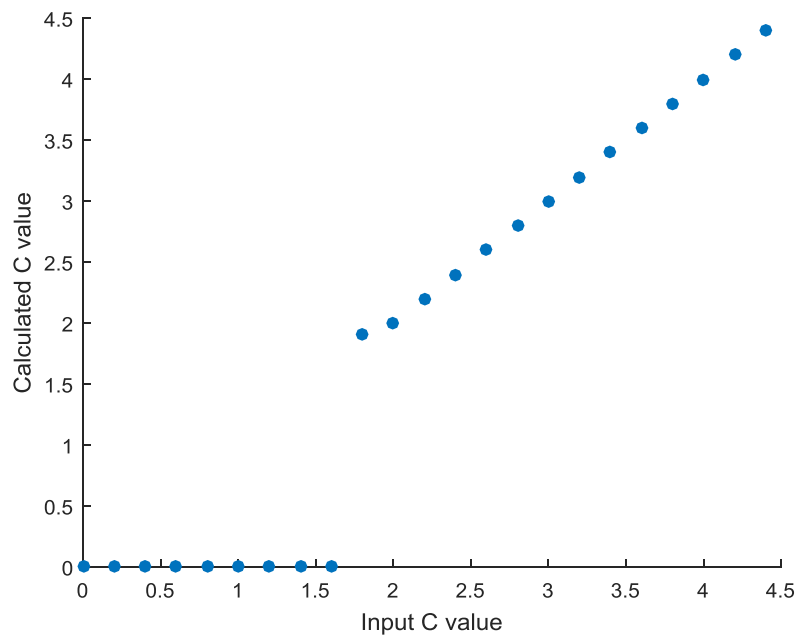


Figure 50: Plot of the calculated scaling factor as a function of the initial input scaling factors. Equation (17) is run after the two hundredth iteration to solve for the calculated C value (See appendix 16 – CsLoop)

3.4.2 The simultaneous effect of k and C on the amount of iterations to convergence

If both the scaling factor C and damping factor k contribute to the amount of iterations to convergence, it is possible to observe if there is a correlation between the two. An observed model is again set up with the same parameter values as in Figure 42; a buried cylinder model at a depth and position of 10 m. The observed model is kept constant and systems with three different initial calculated models are used. For each system, the initial calculated model is acted upon by an inversion process with the damping and scaling factors as initial parameters. This inversion process is set up in equation (18) as a modified Ridge-regression inversion:

$$dP = (A^T A + kI)^{-1} A^T (CD) \quad (18)$$

This process is carried out over an array of C and k values and the amount of iterations to converge is recorded for each. In each progressive system, the initial parameter values are set further away from the observed model's to make convergence conditions more difficult to achieve. The observed model is defined as set up in Figure 42 and all inversion processes used are run to not exceed 20 iterations. Figure 51 - 54 define new basins of attraction, as defined in section 2.2, created using k and C as parameters. Once more as before, in this section convergence conditions are met when the final model's location and depth lie within 0.5 m of the observed model.

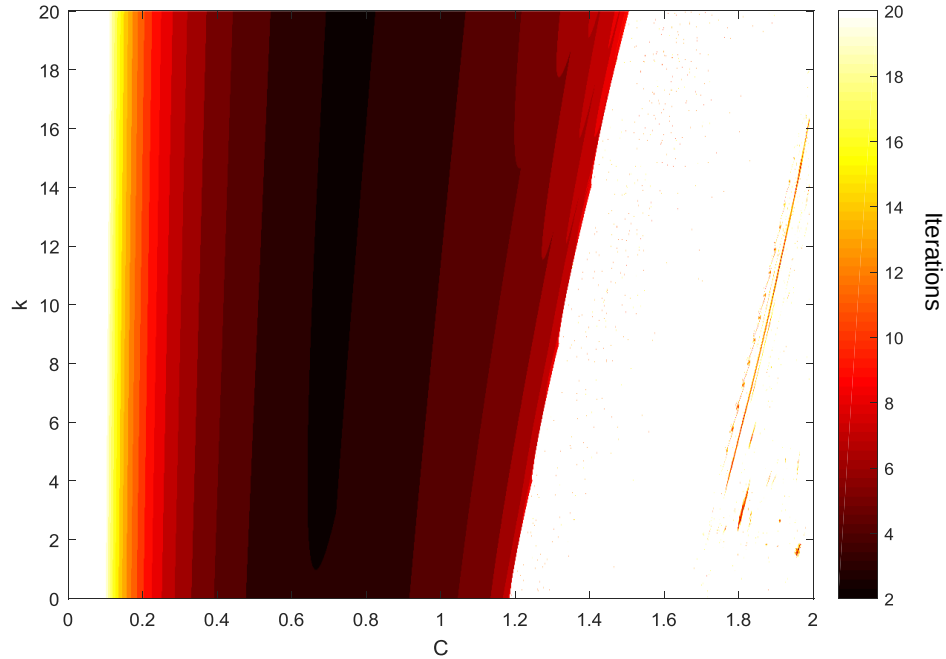


Figure 51: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 20 m. (See appendix 17 – CvsKvsI2)

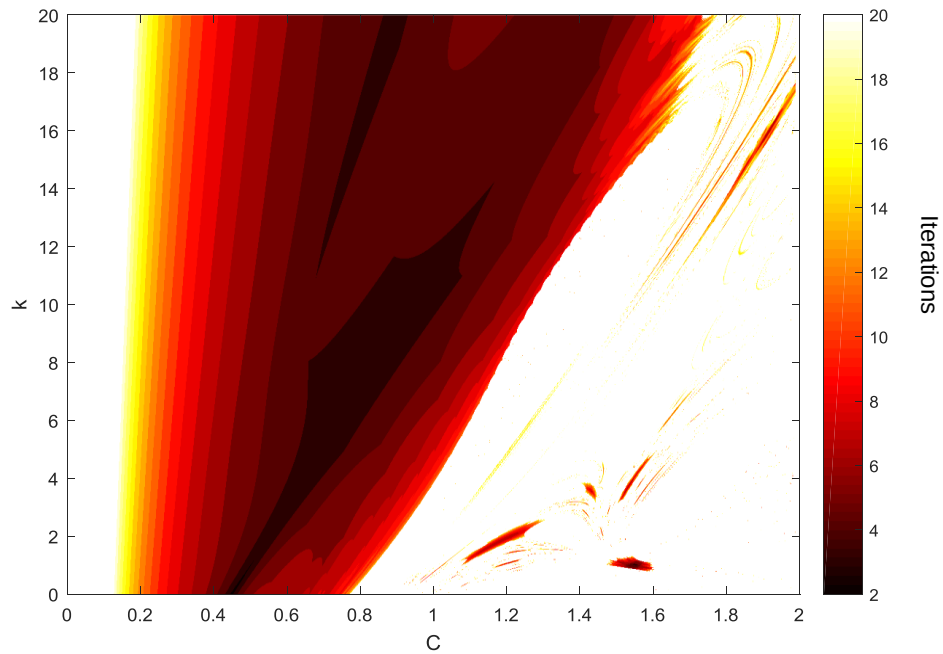


Figure 52: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 40 m. (See appendix 17 – CvsKvsI2)

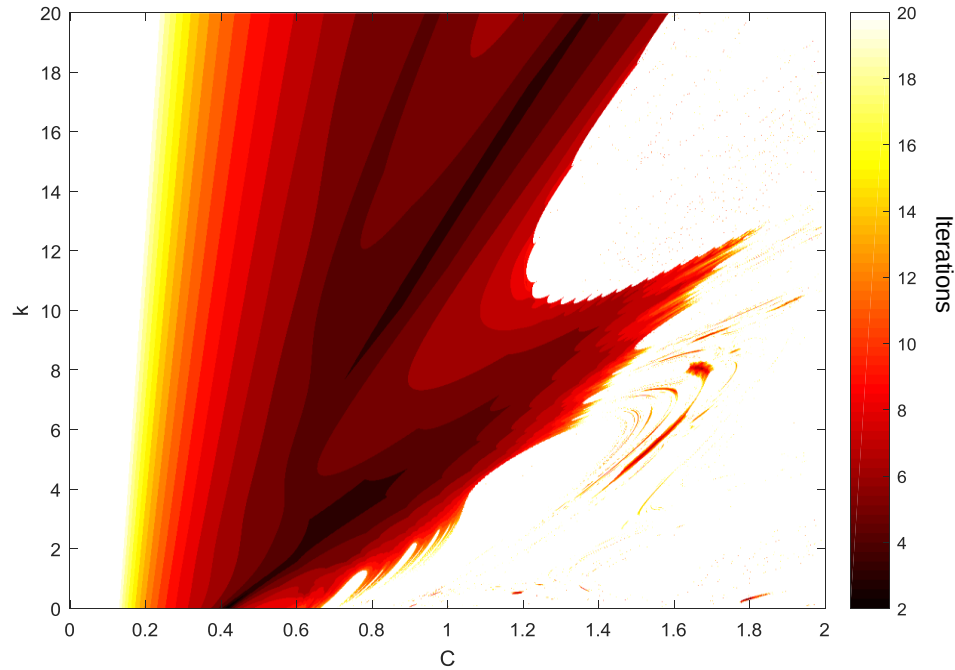


Figure 53: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 60 m. (See appendix 17 – CvsKvsI2)

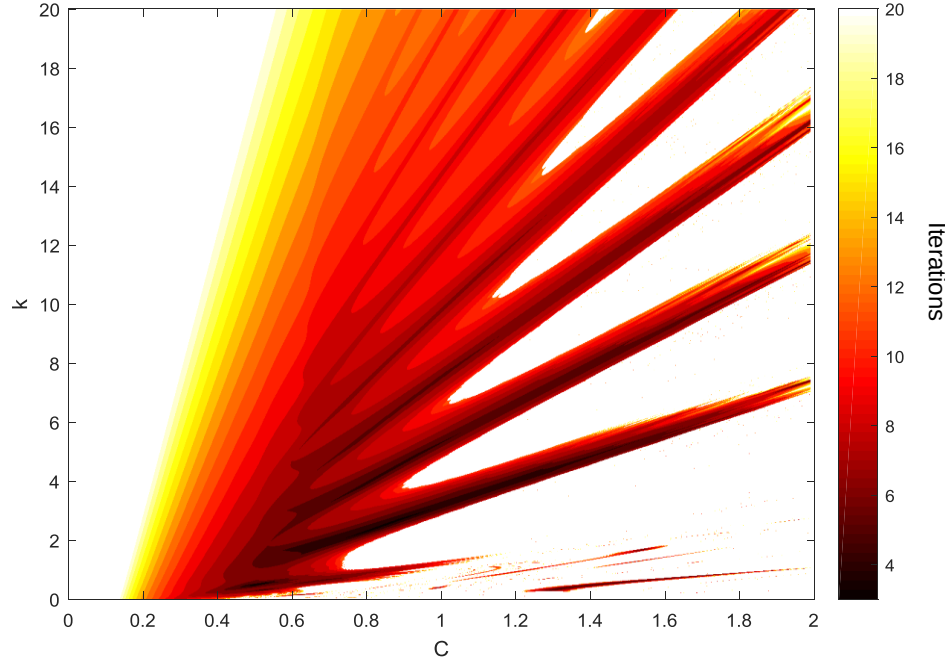


Figure 54: A plot of the amount of iterations to convergence as a function of the damping and scaling factors, the modified Ridge-regression, equation (18), is used on all initial k and C parameter values in the solution space. The adjoining colour bar indicates the amount of iterations. The inversion process is allowed a maximum of 20 iterations. Observed model defined in Figure 42. The initial model is set at a position and depth of 80 m. (See appendix 17 – CvsKvsI2)

As the convergence conditions are made more difficult, the basins of attraction diminish and the amount of iterations to converge increases.

3.4.3 Quantifying the separation of inversion trajectories due to different scaling factors

Inversion trajectories as a function of parameters values can be mapped to observe the inversion process. These trajectories are useful in showing the perturbations of the starting parameter values throughout the inversion process. Tracking divergence of initially close-by trajectories over iterations, can define a rate of separation. The Lyapunov exponent L , is a way of quantifying rates of trajectory separation and is given by equation (19).

$$\delta d(t) = e^{Lt} |\delta d(t_0)| \quad (19)$$

$\delta d(t)$ is the separation of the two trajectories at a time t , where t_0 is the initial time. Therefore, the Lyapunov exponent is a quantitative representation of the rate of separation of two points which were initially close to one another (Dingwell, 2006). In order to calculate the Lyapunov exponent with regards to geophysical inverse problems, the trajectories are monitored with respect to inversion iterations I , which is similar to discrete time. The process then tracks I and I_0 , in the place of t and t_0 , as the current iteration and the initial iteration numbers, respectively.

The scaling factor has been shown to be able to alter the outcomes of the inversion process, see Figure 43 - 45. Another way to investigate the effect of the scaling factor on the iterative inversion process is to select two parameter set values which are close to one another, subject them to the same constant scaling factor, and analyze any separation between the two inversion trajectories. The Lyapunov exponent is used to quantify the separation of the two trajectories once more. A system is established with only one minima of convergence with an observed model as defined in Figure 42 again. Two initial model x positions are set as separated by 0.001 m. In Figure 55 the scaling factor is set at 2 and the inversion is run on the two different initial models, the modified Moore-Penrose inversion process is run with equation (13).

Two points originally separated by 0.001 m and separated by 0 m after 10th iteration | $C = 2$ and Lyapunov exponent = -0.12

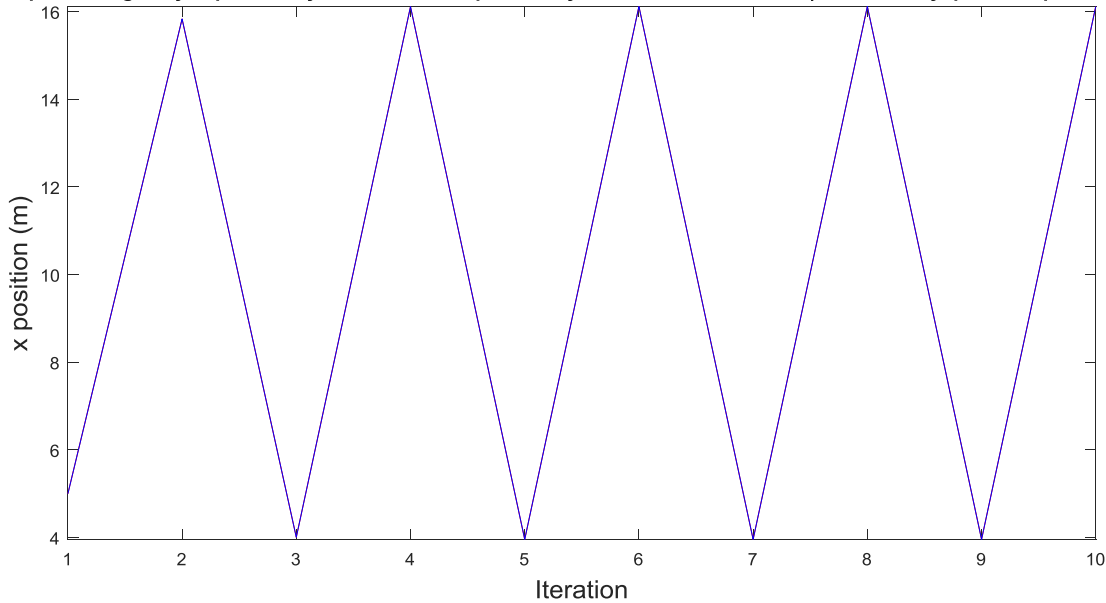


Figure 55: Plots of the x position as a function of the iteration for two separate inversion trajectories. The two inversion trajectories are indistinguishable from one another. The two points are originally separated by 0.001 m and are separated by 0 m after 10 iterations. $C = 2$ and the inversion is carried out by the modified Moore-Penrose inversion process in equation (13). The distance between the two inversion trajectories decrease as the iterations progress. (See appendix 18 – LapyunovC)

As the two parameter sets tend toward the same minima, the inversion trajectories are indistinguishable. This is better shown in Figure 56 where the x parameter values tend closer to one another resulting in a negative Lyapunov exponent value. The Lyapunov exponent is calculated from the relationship between the logs of the separation of the trajectories as function of the iterations in the inversion process.

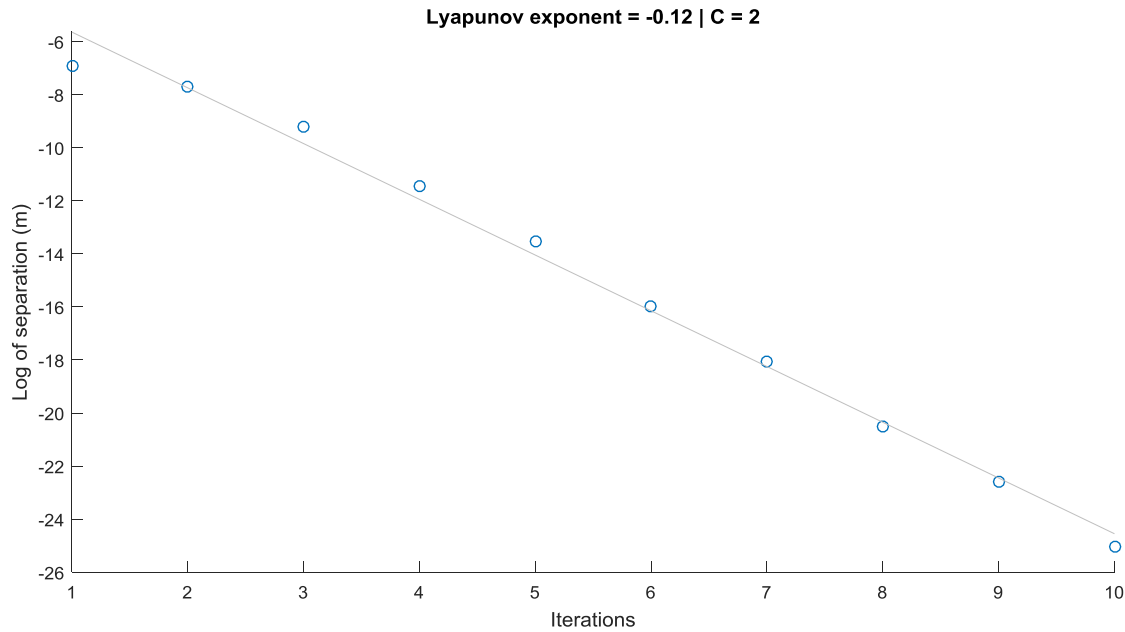


Figure 56: Plot of the log of the separation between two inversion trajectories as a function of the iterations. The gradient of the line of best fit is used to solve for the Lyapunov exponent. Here the Lyapunov exponent is negative as the two bodies drift closer toward one another instead of diverging from one another. (See appendix 18 – LapyunovC)

Next, using the same system and model parameters, higher scaling factor values are used in the inversion processes to again observe inversion trajectories. For each scaling factor value and associated inversion trajectories the Lyapunov exponent is extracted as was done in Figure 56 and using the Lyapunov expression in equation (19); see Figure 57. In all systems, as before, the initial x positions are separated by 0.001 m and the separations at the end of the processed is displayed for each scaling factor, respectively.

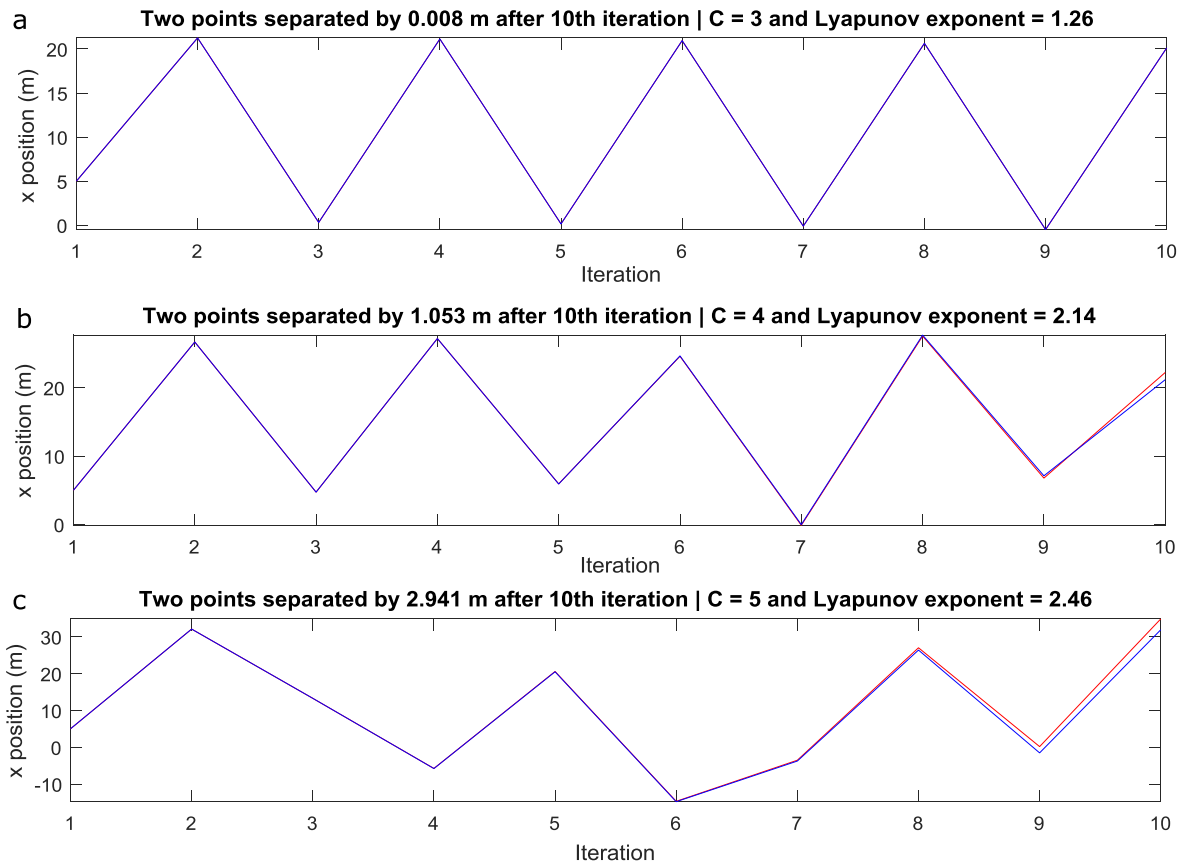


Figure 57: Plots of the x position as a function of the iteration for separate inversion trajectories. The red and blue trajectories show how the distance between the two points is increased as the iterations progress through the inversion. The two bodies are originally separated by 0.001 m and $C = 3, 4$ and 5 in *a*, *b* and *c*, respectively. The modified Moore-Penrose inversion processes, as set up in equation (13), are run on all initial models. (See appendix 18 – LapyunovC)

3.5 The bifurcation diagram for the geophysical inversion problem

A bifurcation diagram can be constructed from the relationship between the observed periodicity (bifurcations) and the scaling factors used in modified Moore-Penrose inversion, see equation (13). These relationships are seen in Figure 43 and Figure 44. The observed model is set up as in Figure 42 again. Figure 58 is the result of tracking the stable fixed x positions, as a function of the scaling factor.

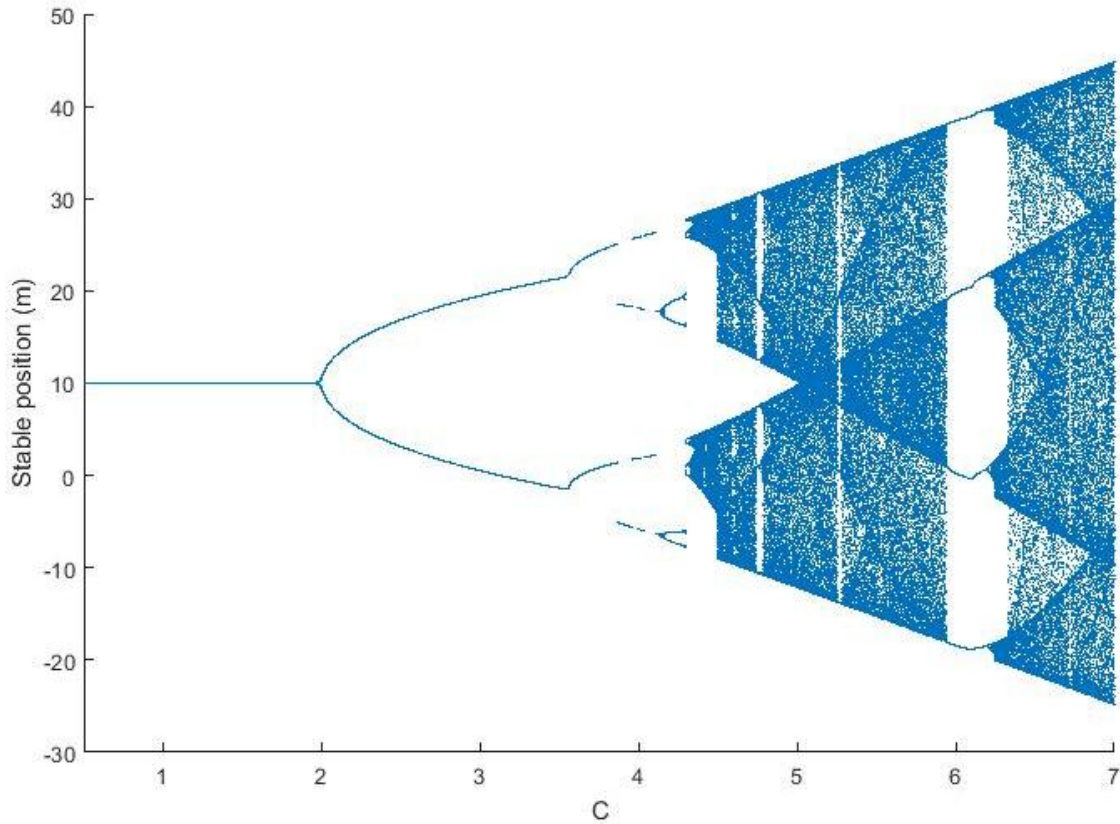


Figure 58: A bifurcation diagram created by recording the fixed points observed after parameter inversion at different scaling factors. The bifurcation diagram is vertically symmetric about the observed x parameter value. The initial model location is at -10 m along the range and the observed model location along the range is at 10 m. The scaling factor is varied from 0.5 to 7. All scaling factors are kept constant in the modified Moore-Penrose inversion, see equation (13). (See appendix 4 – MakeBMap)

3.6 The Feigenbaum constants

A bifurcation diagram is created by running the same inversion process with the same initial model parameter values; observed x parameter value = 10 m and initial x parameter value = -10 m. For each value of the scaling factor, the asymptotic parameter values are recorded, see Figure 59. The nature of the bifurcation in the inversion system is a period doubling, which is similar to that of the logistic map seen in Figure 10. The bifurcation diagram can be used to calculate the Feigenbaum constants introduced in 1.3.4.

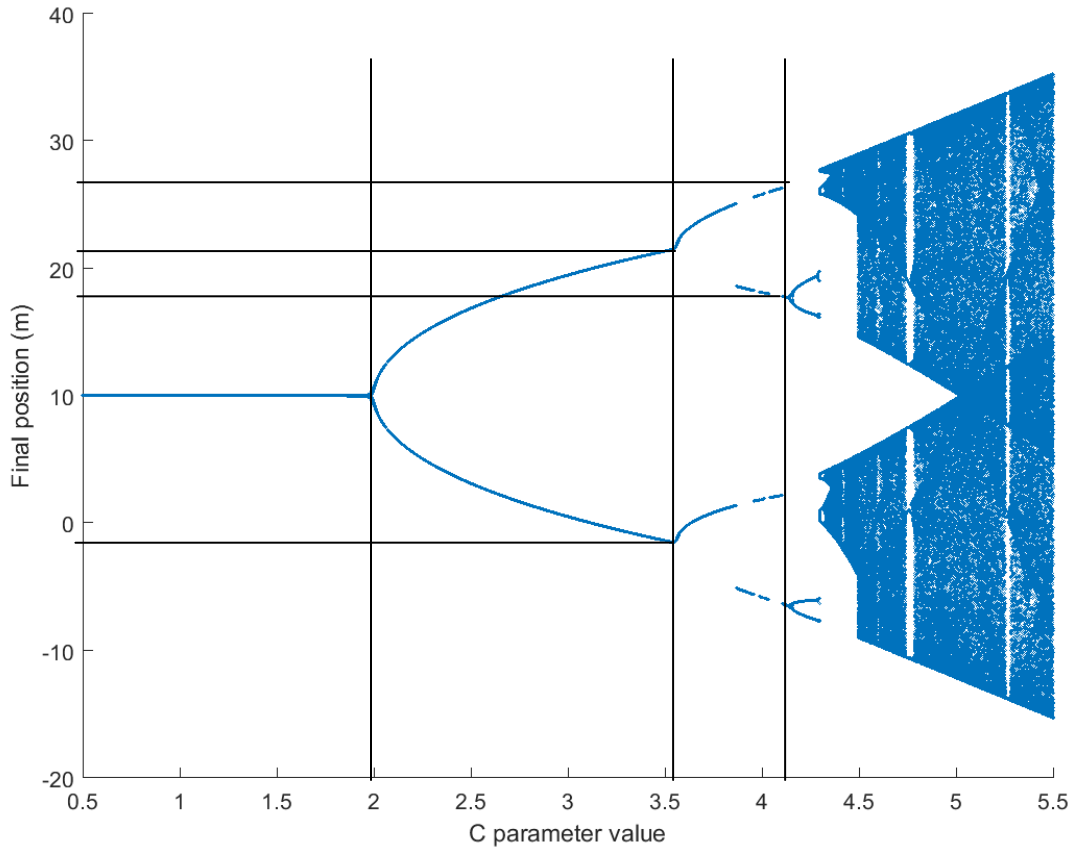


Figure 59: A bifurcation diagram created by recording the fixed points observed after parameter inversion at different scaling factors. The bifurcation diagram is vertically symmetric about the observed x parameter value. The initial model location is at -10 m along the range and the observed model location along the range is at 10 m. The scaling factor is varied from 0 to 7. All scaling factors are kept constant in the modified Moore-Penrose inversion, see equation (13). The values used to calculate the first (equation (5)) and second (equation (6)) Feigenbaum constant are indicated by the black lines. A similar methodology was applied in section 1.3.5. (See appendix 4 – MakeBMap)

From equation (5) the first Feigenbaum constant can be calculated from:

$$\delta_n = \frac{A_n - A_{n-1}}{A_{n+1} - A_n}$$

$$\delta_2 = \frac{3.56 - 2.00}{4.14 - 3.56}$$

$$\delta_2 = 2.69$$

From equation (6) the second Feigenbaum constant can be calculated from:

$$\alpha_n = \frac{D_n}{D_{n-1}}$$

$$\alpha_2 = \frac{23.17}{8.46}$$

$$\alpha_2 = 2.74$$

Once more, as in section 1.3.5, the calculated results do not initially match the published results exactly. Assuming the constants hold true at later bifurcations however, it should be possible to make predictions for at which scaling factor values, bifurcations or chaos might emerge.

3.7 Analysis of basins of attraction for models containing two cylindrical sources

The gravitational response as a result of two buried cylinders is introduced here. In Figure 60, the observed response is generated from two buried cylinders in red, and the modelled response in green is generated from a single buried cylinder. There are now two distinct responses within the observed system and the inversion process now poses new challenges which can be investigated.

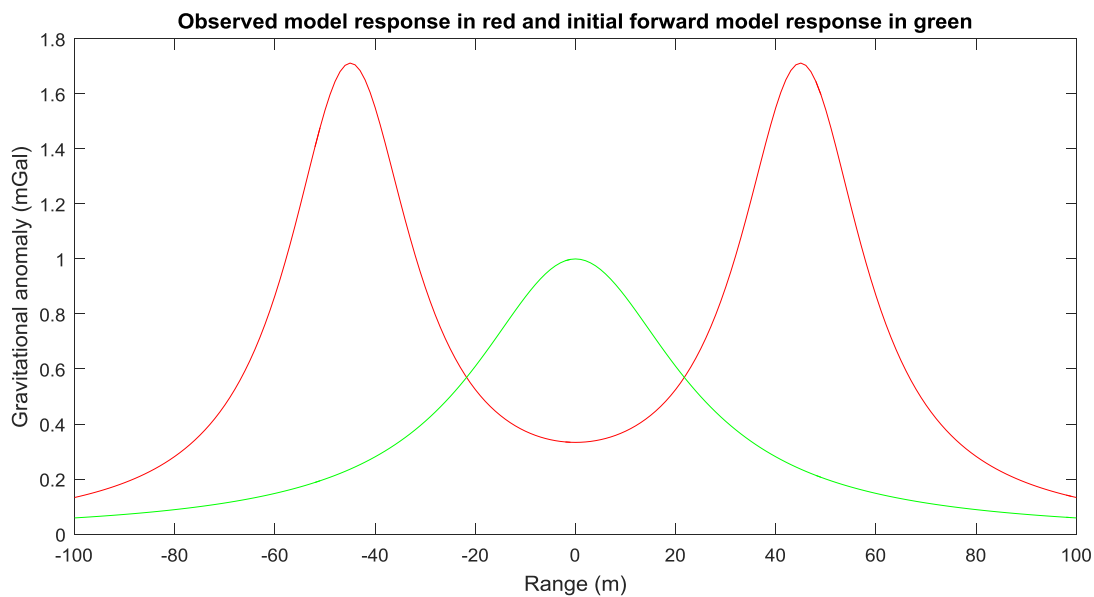


Figure 60: Two anomalies are shown as a result of two separate bodies and generated using the cylinder models. The two bodies in the observed model are separated by 90 m about the centre of the profile and are both buried at 15 m. The initial model is at 0 m along the range and is buried at a depth of 25 m. (See appendix 12 - ResponsePlusModel)

Using the work already done in section 3.2, the original parameters of a model with two buried cylindrical sources can be altered, to change the attractors, and basins of convergence.

3.7.1 Moore-Penrose inversion process for models at varied source depths

First; models which contain two buried cylindrical bodies, equally separated about the centre of the profile and buried at different depths, are established. The responses due to these models buried at different depths are seen in Figure 61.

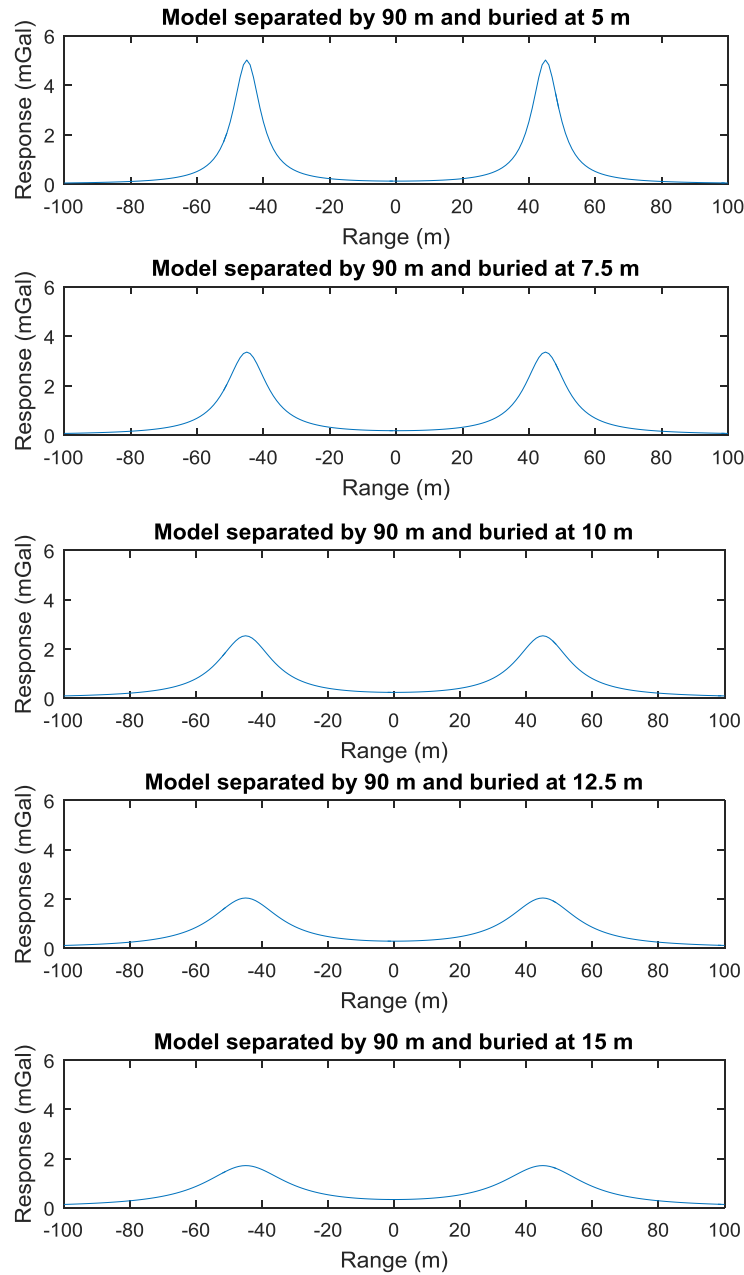


Figure 61: Responses due to a model containing two cylindrical sources both buried at different depths. Gravitational responses are generated by the buried cylinder models, as expressed in equation (9). (See appendix 12 – ResponsePlusModel)

In this section; Figure 62 - 66 show the basins of attraction, as a result of varying the observed model's depths.

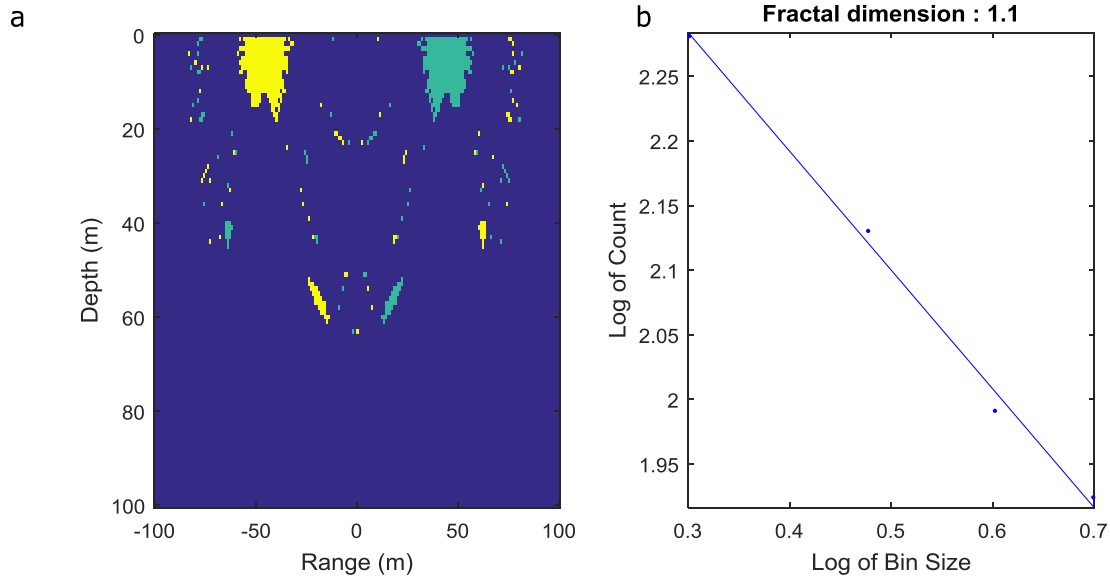


Figure 62: *a* is the basin of attraction for a model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

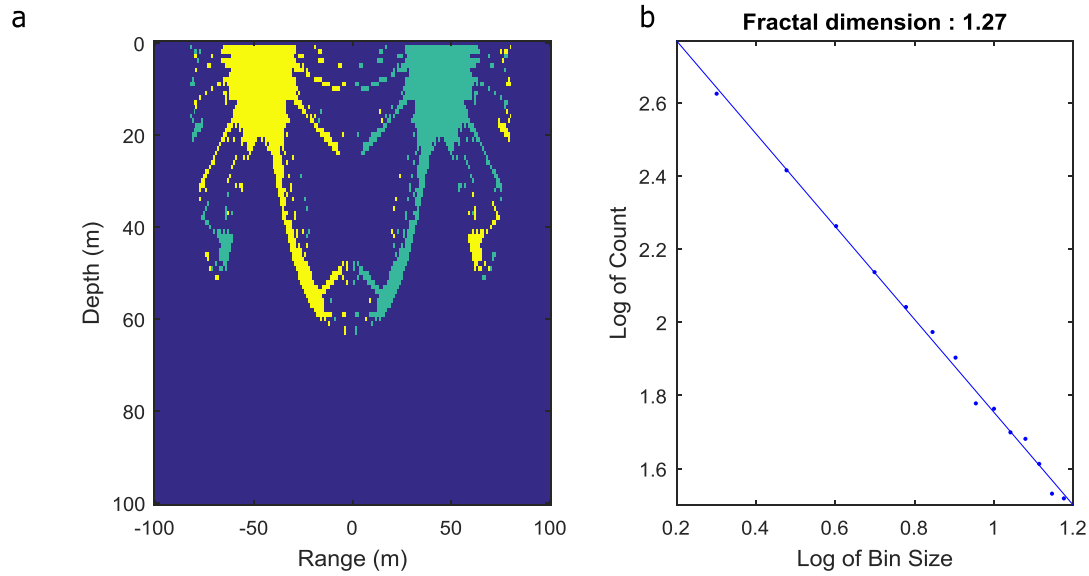


Figure 63: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 7.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

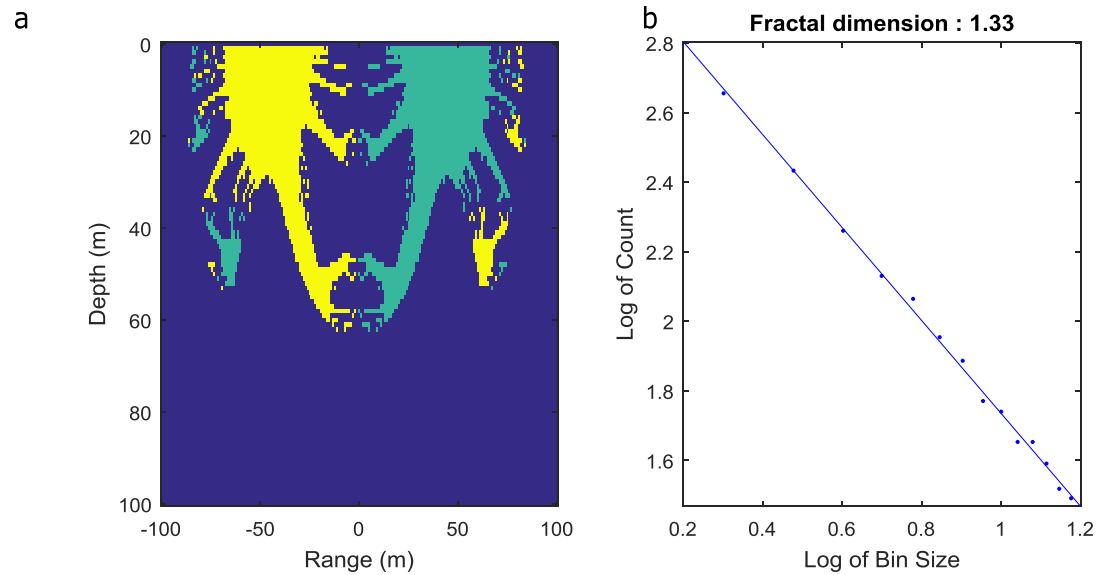


Figure 64 *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 10 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

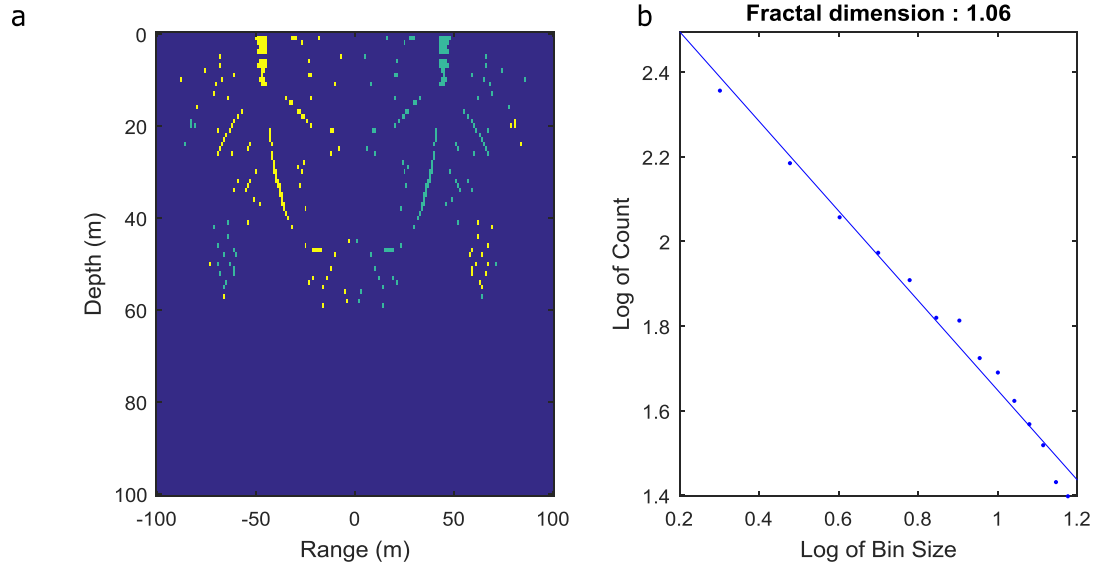


Figure 65: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 12.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

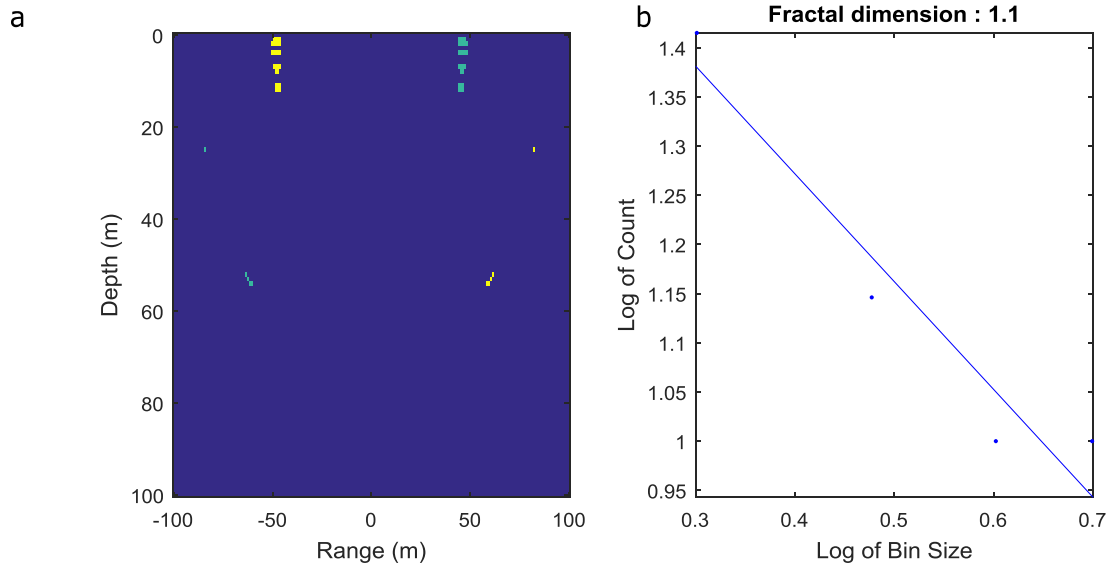


Figure 66: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 15 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, as defined in equation (7), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

As the depth is increased, the two responses interact with one another and, initially at least, the basins of attraction grow larger. At a certain point however, roughly three times the original depth, the responses grow too shallow and it becomes harder to converge to either the left or right body; as a result the basins of attraction reduce in size.

3.7.2 Ridge-regression inversion process for models at varied source depths

In this section; Figure 67 - 71 show the basins of attraction, as a result of varying the observed model's depths, using the Ridge-regression inversion process, as defined in equation (8), on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The fractal dimensions are established with the box-counting method defined in equation (3). Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model.

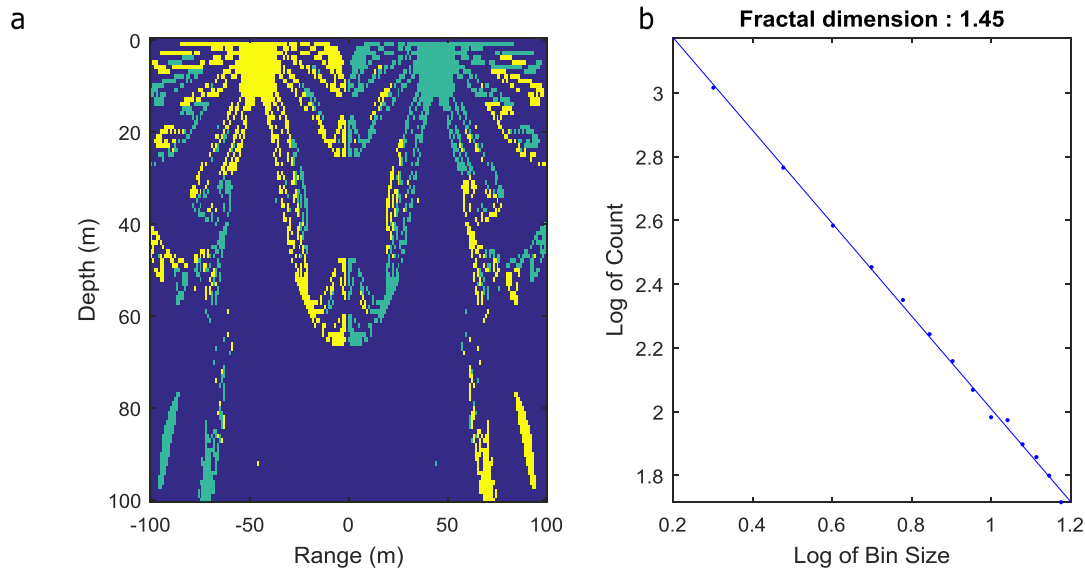


Figure 67: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

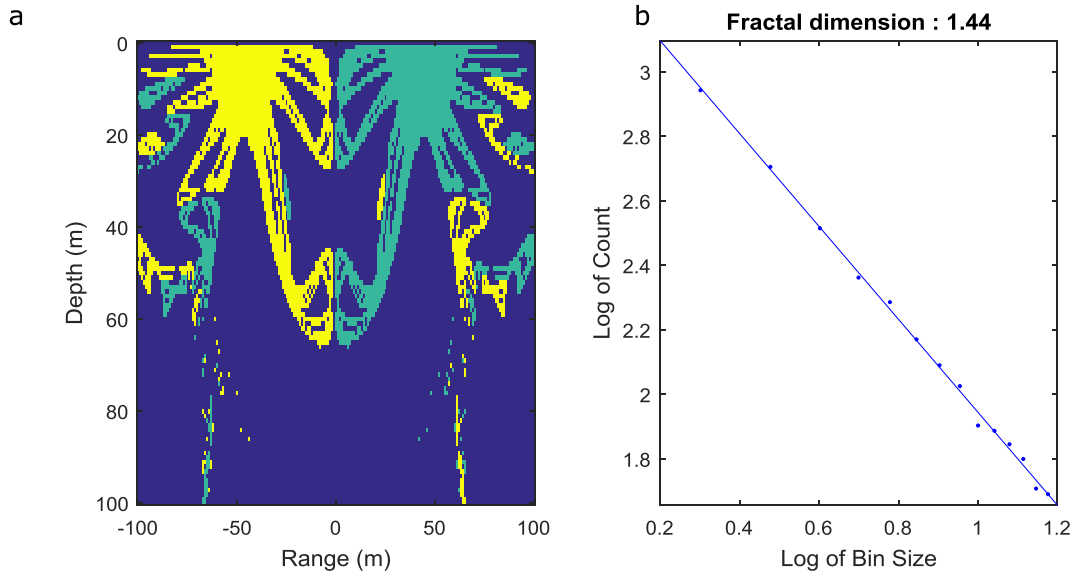


Figure 68: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 7.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

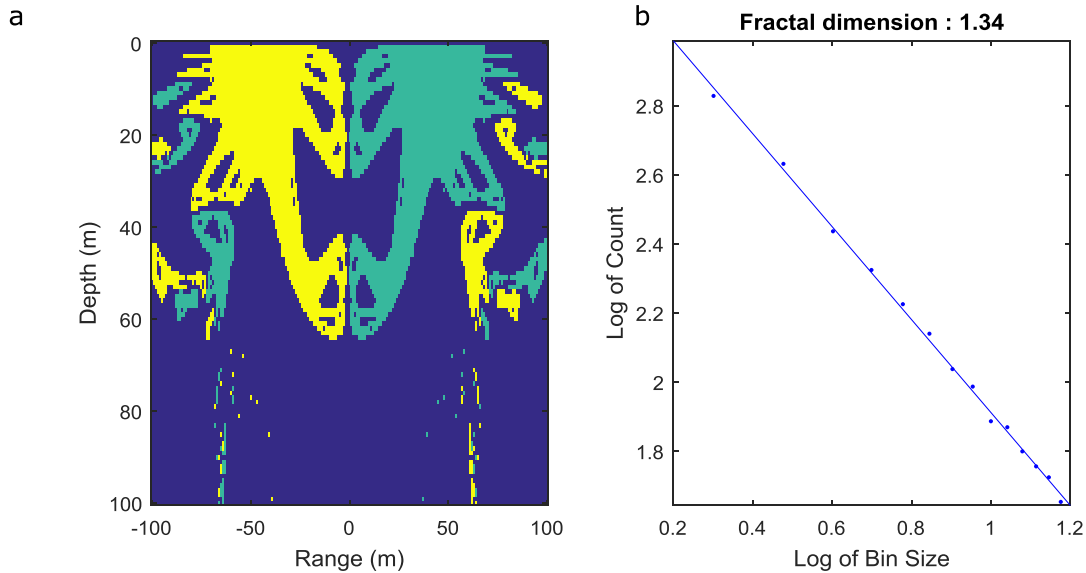


Figure 69: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 10 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

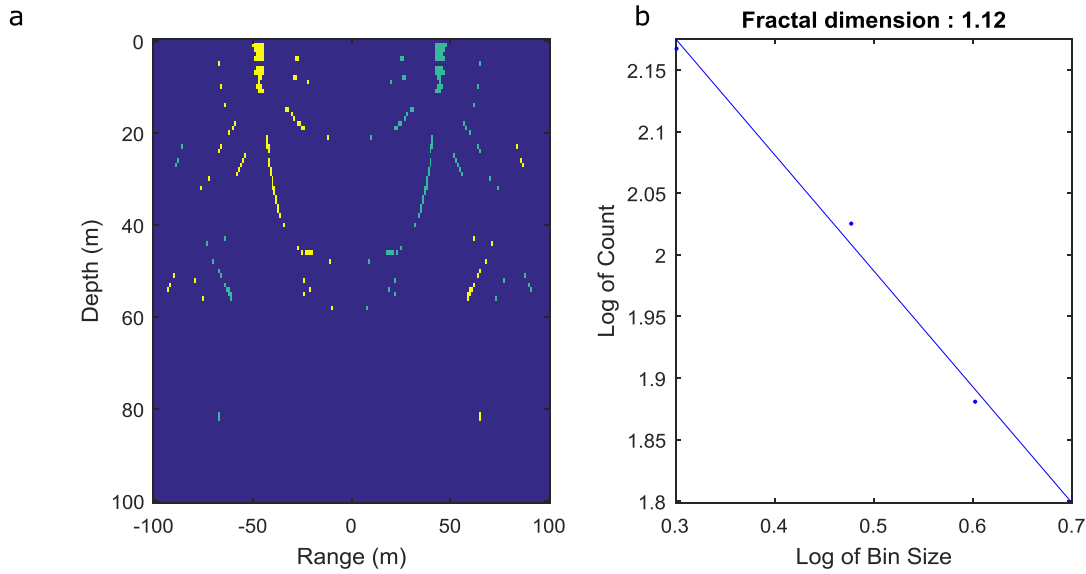


Figure 70: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 12.5 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

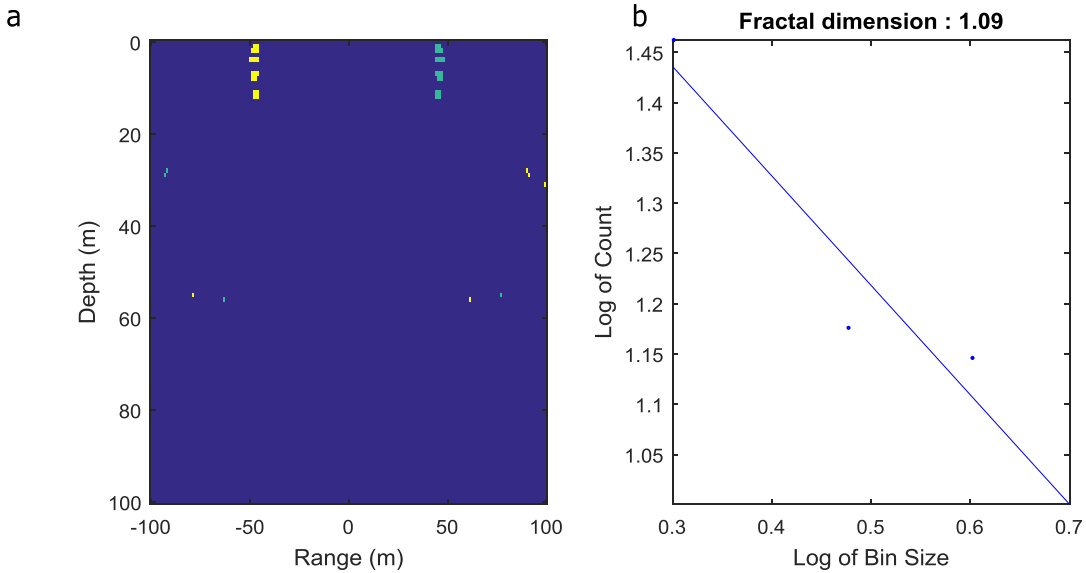


Figure 71: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 15 m, respectively (refer to Figure 61). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, as defined in equation (8), acts on all initial models in the solution space. Different basins of attraction are defined for the left and right body within the model, respectively. Yellow and green represent convergence to the left and right bodies, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

It is again noted that as the depth is increased, initially at least, the basins of attraction grow larger. With the aid of damping the system however; the basins of attraction are larger as more initial models are able to converge to either of the observed model bodies. Once more however, as the responses grow too shallow and it becomes harder to converge to either the left or right body.

3.7.3 Moore-Penrose inversion process for models at varied source separations

Models which contain two buried cylindrical bodies equally separated about the centre of the profile at different separation and which are buried at the same depths are established. The responses due to these models buried at different depths are seen in Figure 72.

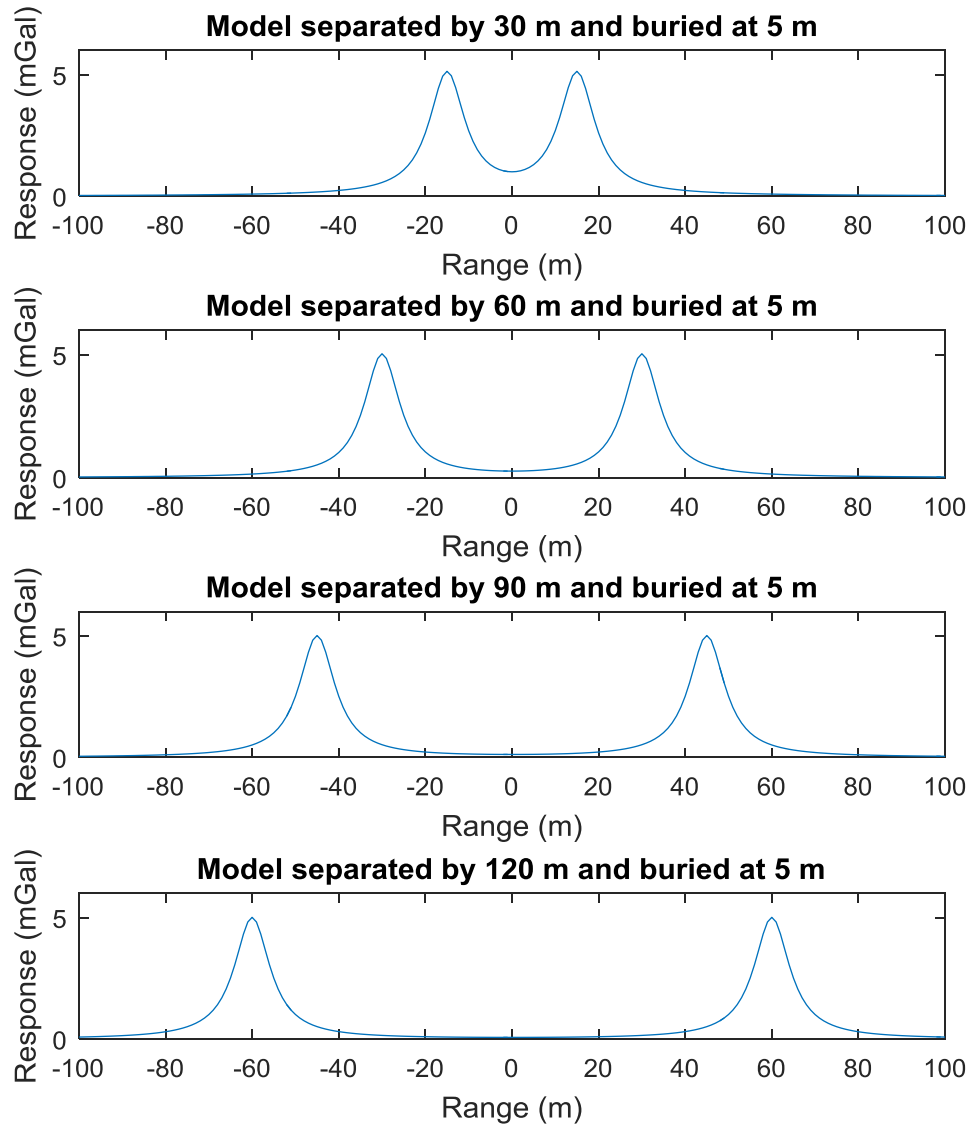


Figure 72: Responses due to a model containing two cylindrical sources both buried at different separations. Gravitational responses are generated by the buried cylinder models, as expressed in equation (9). (See appendix 12 – ResponsePlusModel)

In this section; Figure 73 - 76 show the basins of attraction and associated basin of attraction's dimension as a result of varying the source separations between two buried cylindrical bodies. Convergence to the centre of the profile, between the two bodies, is here also tracked to observe any new basins of attraction.

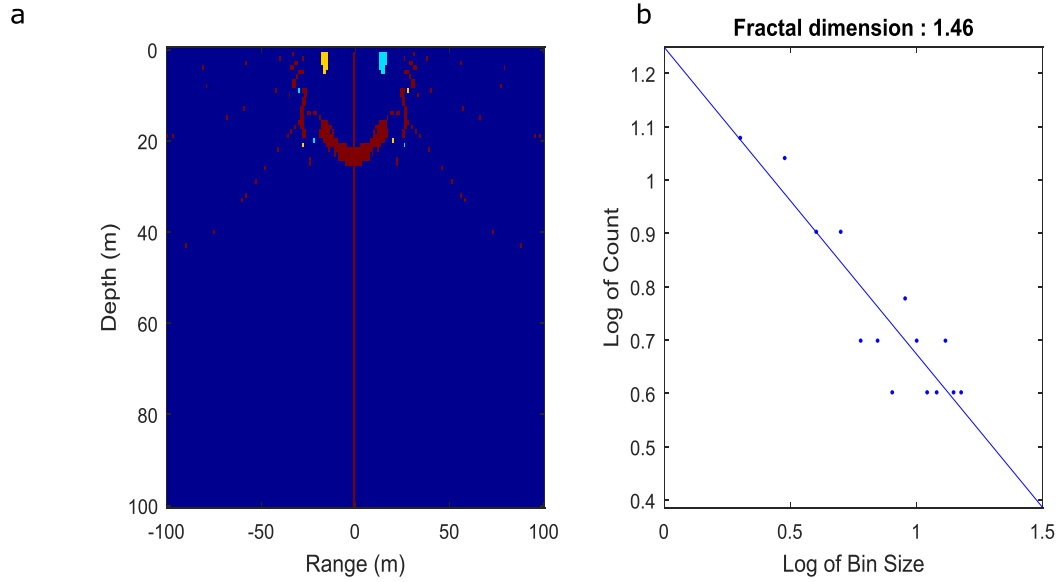


Figure 73: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 30 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

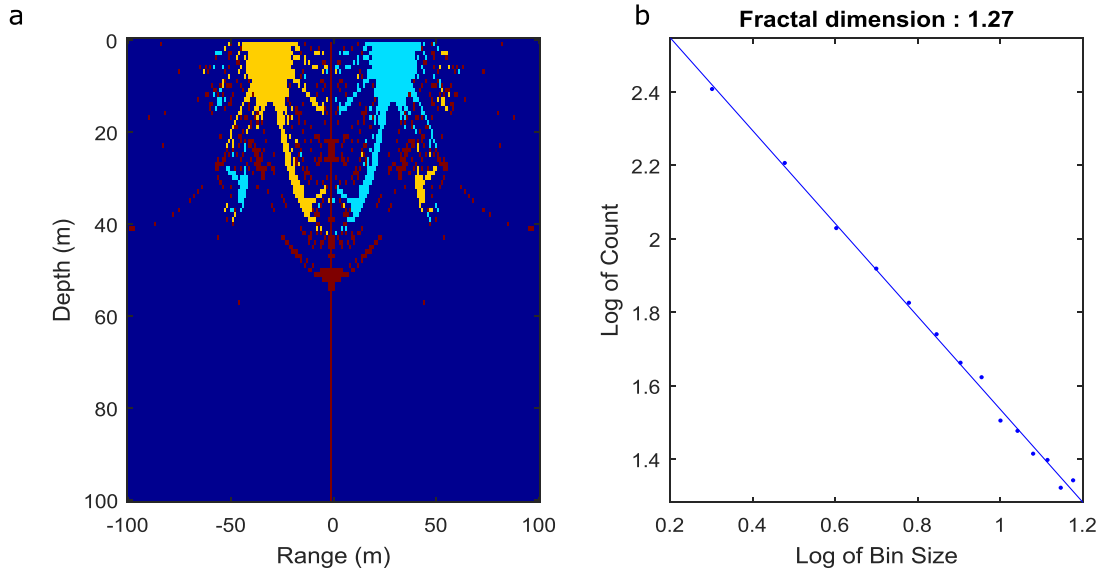


Figure 74: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 60 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

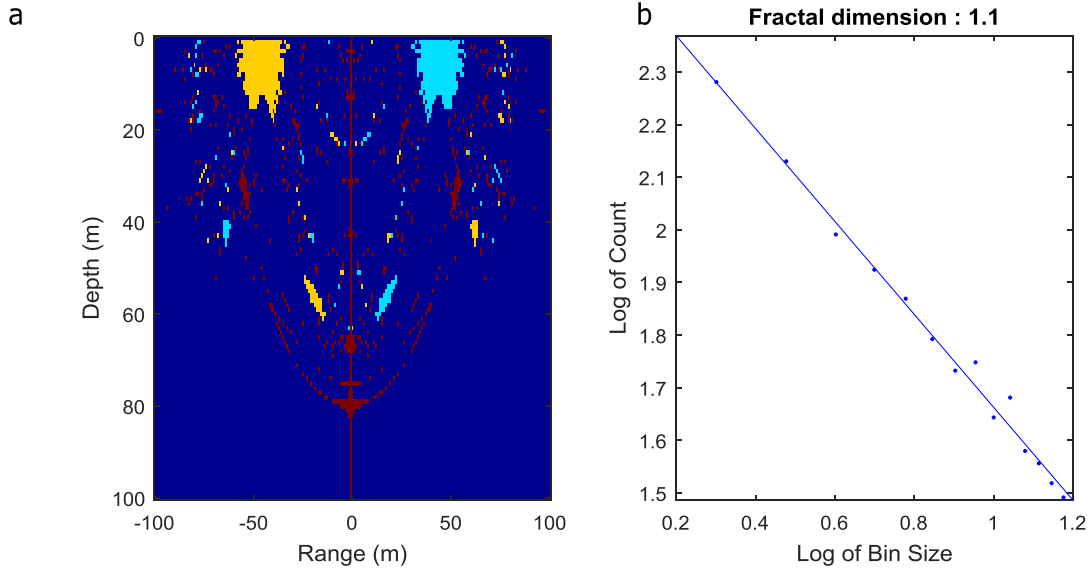


Figure 75: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

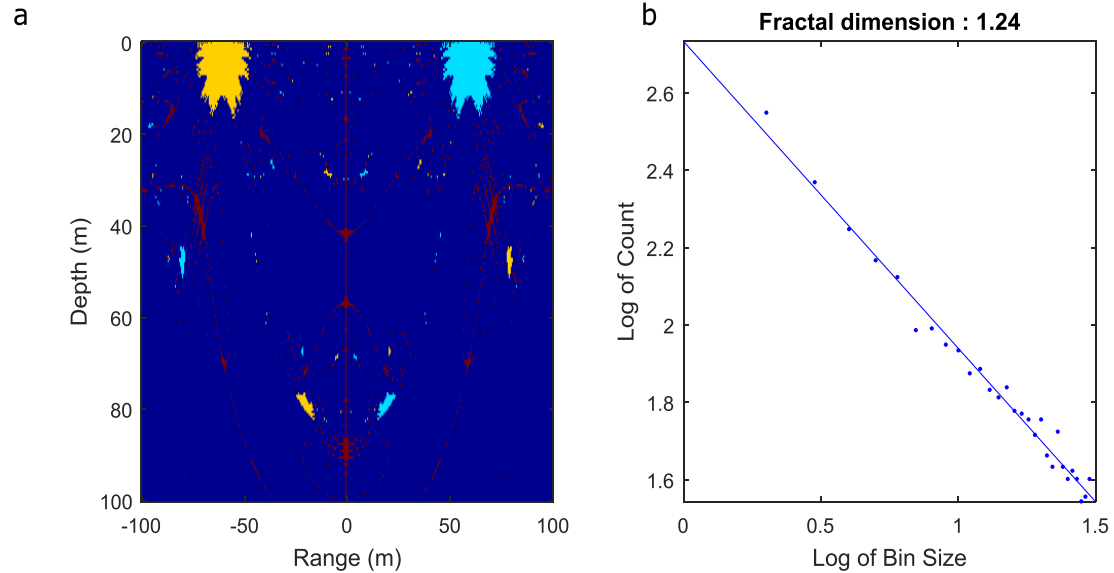


Figure 76: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 120 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Moore-Penrose inversion process, see equation (7), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

Initially the two responses are too close together and it becomes difficult to converge to solely the left or right body; this is observed by the diminished basins of attraction. As the bodies drift apart, the basins of attraction grow larger and more instances of converge occur. At a certain point the two responses are far enough apart that the basins of attraction begin to take on the appearance of the basins of attraction as created by a single buried source, see Figure 26.

3.7.4 Ridge-regression inversion process for models at varied source separations

In this section; Figure 77 - 80 show the basins of attraction and associated basin of attraction's dimension as a result of varying the source separations between two buried cylindrical bodies. Convergence to the centre of the profile is here again tracked to observe any new basins of attraction.

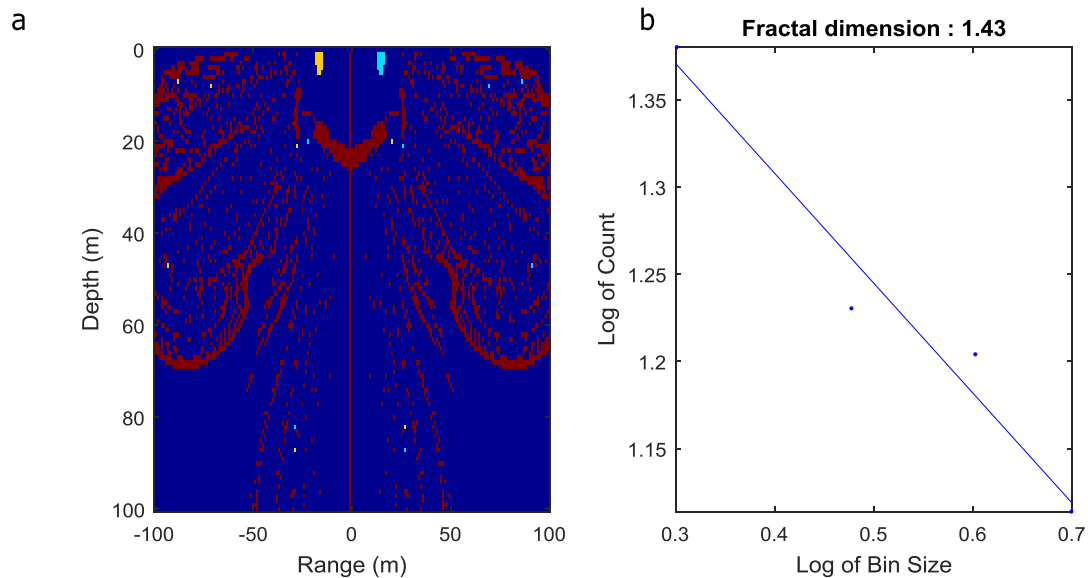


Figure 77: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 30 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

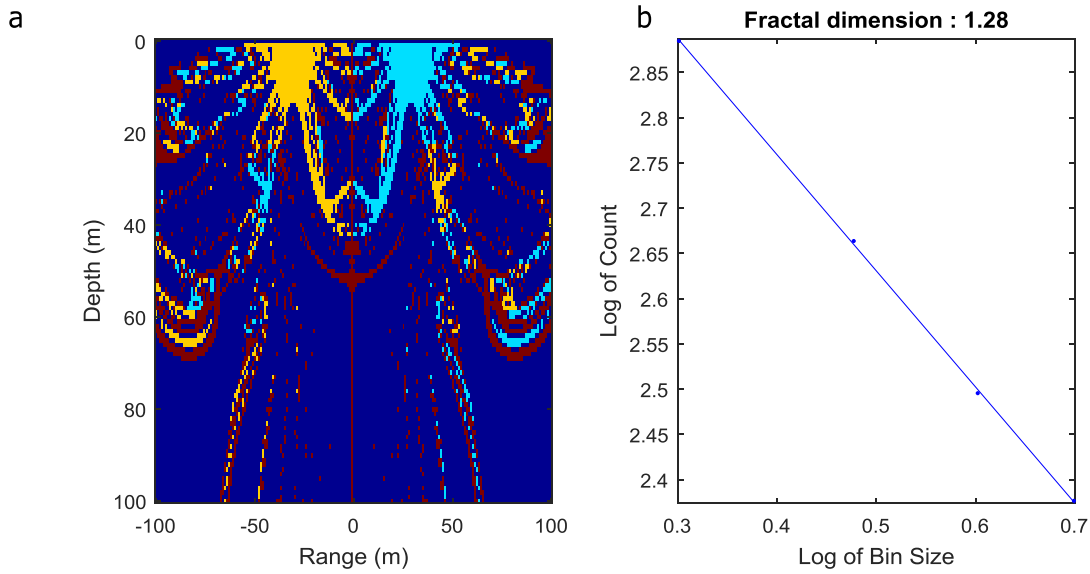


Figure 78: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 60 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

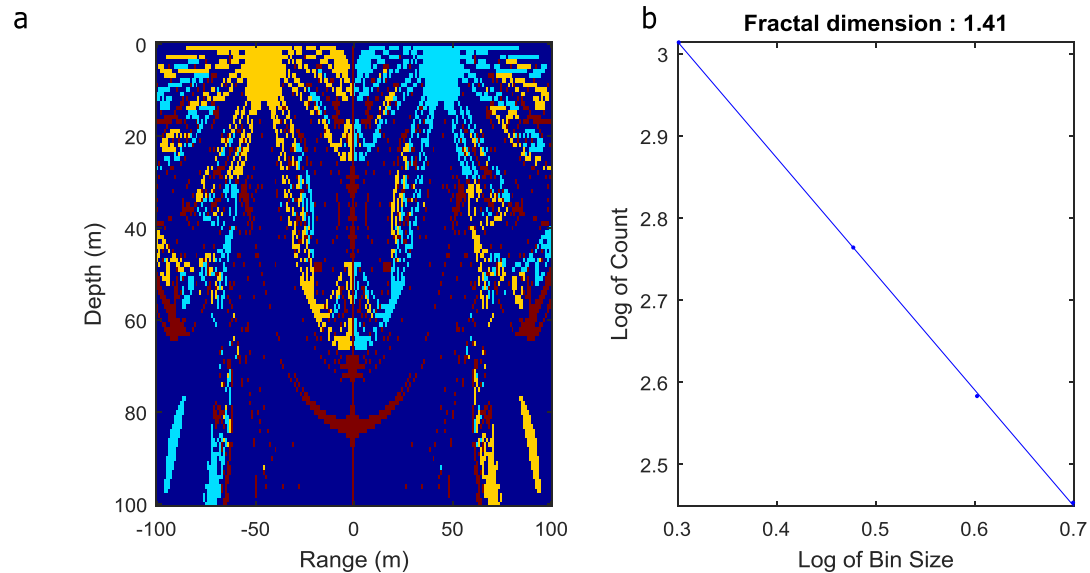


Figure 79: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

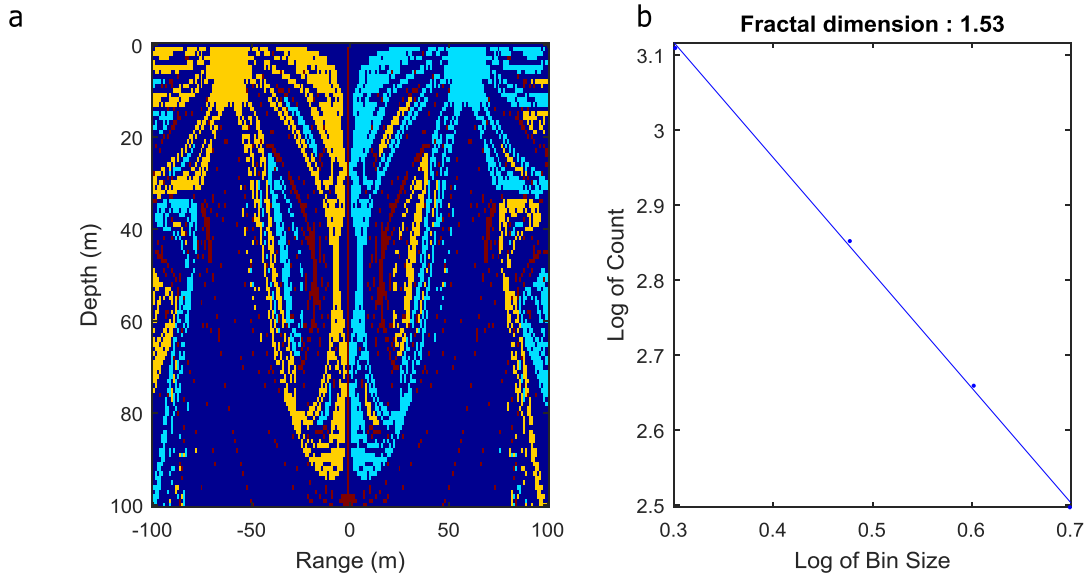


Figure 80: *a* is the basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 120 m and 5 m, respectively (refer to Figure 72). *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). The Ridge-regression inversion process, see equation (8), acts on all initial models. Different basins of attraction are defined for the left body, right body and centre of the profile, respectively. Yellow, red and light blue represent convergence to the left body, centre of the profile and the right body, respectively. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 11 – VariedXorZkTwoBodies)

Once again; initially the two responses are too close together and it is difficult to converge to solely the left or right body. As the bodies drift apart, the basins of attraction grow larger and more instances of converge occur. At a certain point the two responses are far enough apart that the basins of attraction begin to resemble the basins of attraction as created by a single buried source, see Figure 27. For this model, this is seen in Figure 78 at a source separation of approximately 90 m. This separation limit is dependent upon the initial depth of the models and is approximately eight to ten times the depth of the model. With the aid of damping the system however; the basins of attraction are larger as more initial models are able to converge to either of the observed model bodies. Interactions between the two responses near the middle of the profile are now also observable.

3.7.5 Varied resolutions of the basin of attraction using the Ridge-regression inversion process

Figure 82 - 84 show the basins of attraction and associated fractal dimension as a result of varying the density of the solution space. The bounds of the solution space are kept constant, but

the solution space is sampled more densely. In each set-up, the same observed model is used and consists of two buried cylindrical sources, separated by 76 m and buried at 5 m, as seen in Figure 81.

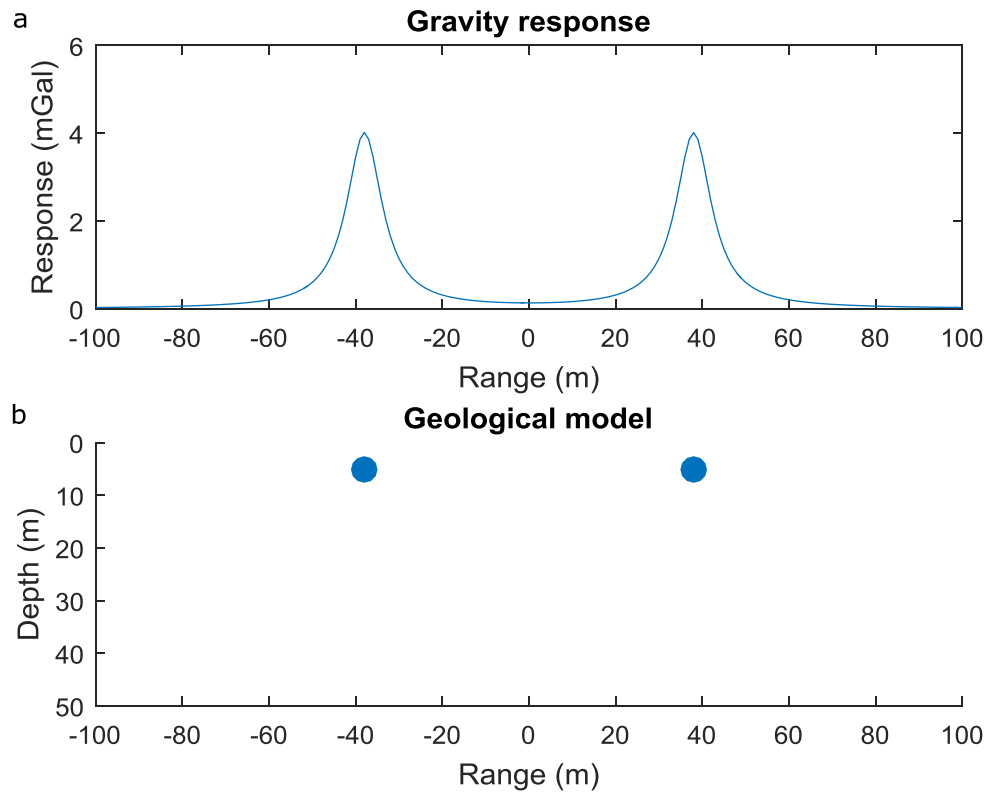


Figure 81: Observed response in *a* as a result of the model in *b*. Model is created with two cylindrical bodies, buried at 5 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel)

In this section convergence to the centre of the profile is again tracked and the time it takes to construct the basins of attraction is also recorded, to gauge the computational intensity.

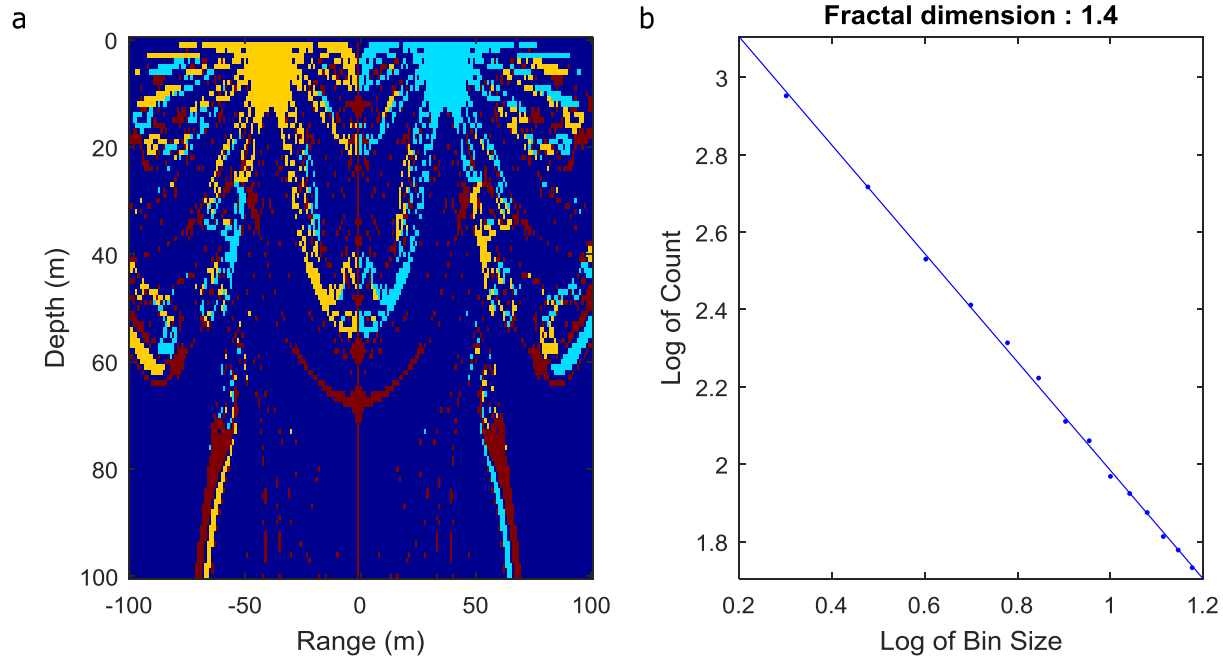


Figure 82 *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A standard solution space is used where the parameters change by 1 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension takes 9.2 seconds. (See appendix 11 – VariedXorZkTwoBodies)

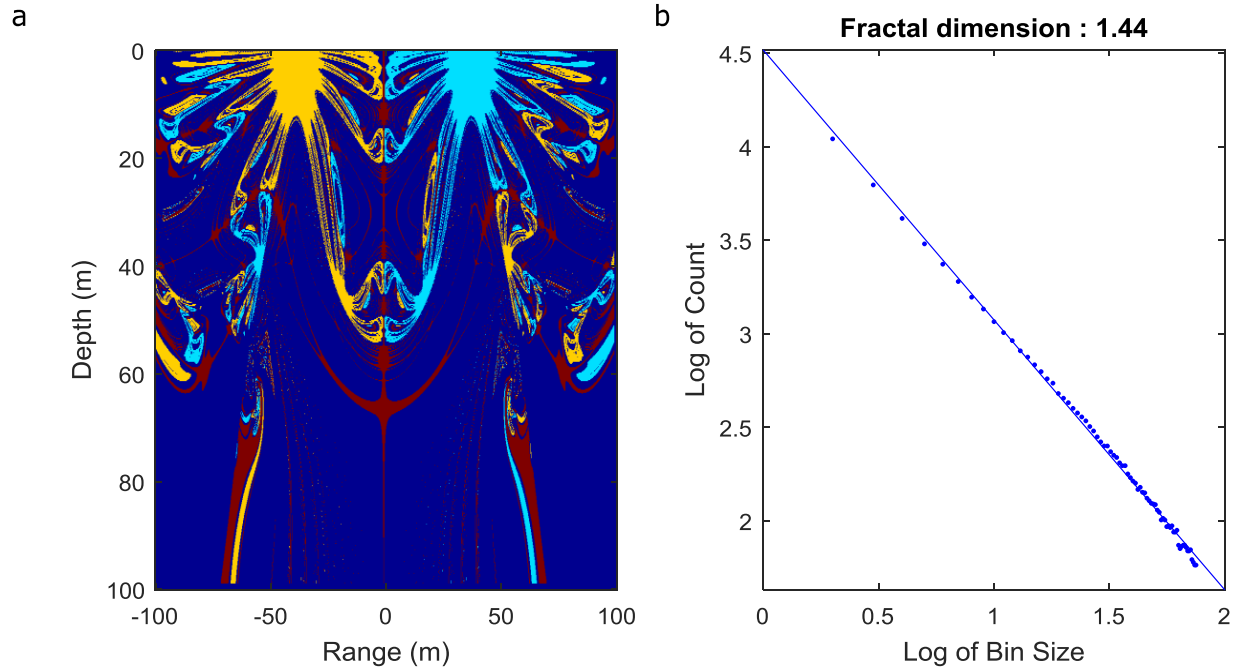


Figure 83: *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A solution space five times as dense is used where the parameters change by 0.2 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension takes 3 minutes 6.3 seconds. (See appendix 11 – VariedXorZkTwoBodies)

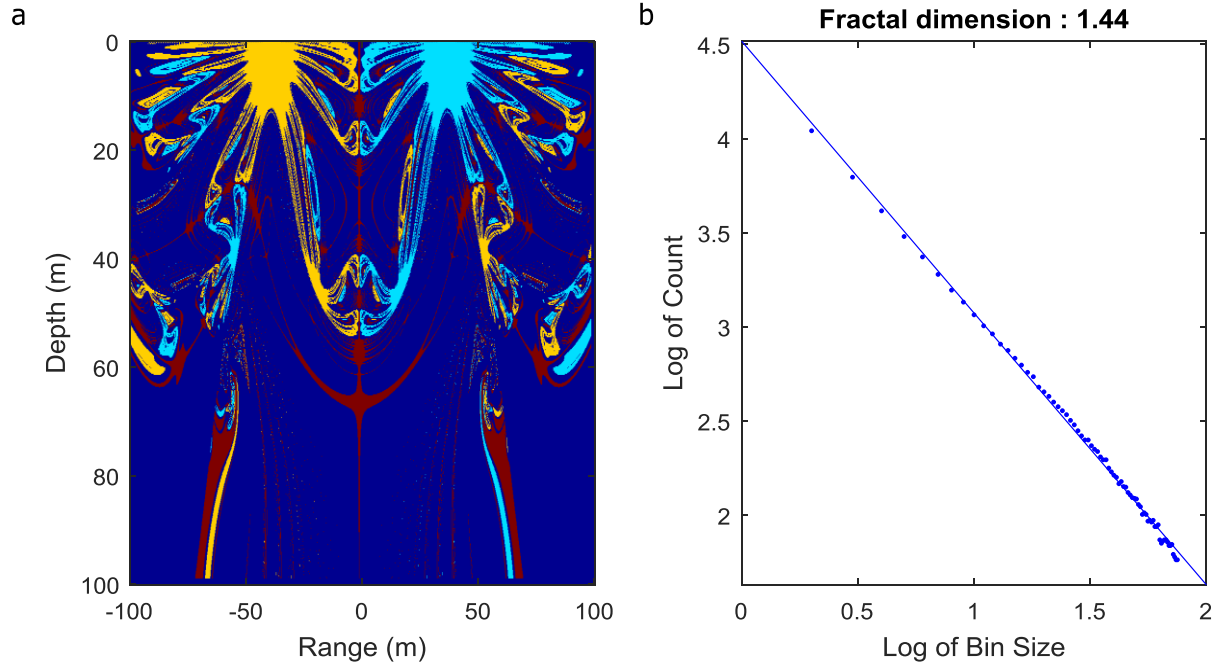


Figure 84: *a* is the basin of attraction for model containing two cylindrical sources. The observed model is defined in Figure 81. *b* is the calculation of the dimension of the basin of attraction, using the box-counting method, see equation (3). A solution space ten times as dense is used where the parameters change by 0.1 m along each cell in the solution space. Yellow and light blue represent convergence to the left and right bodies, respectively. Red indicates initial models which converged to the centre of the profile, between the two bodies. The dark blue background colour indicates instances where initial model did not converge to either body in the observed model. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image and calculation of the dimension took 8 minutes 2.1 seconds. (See appendix 11 – VariedXorZkTwoBodies)

The dimension of the basin of attraction is seen to remain mainly constant, even as the solution space is more densely sampled. Following this, Figure 86 and Figure 87 show the difference in basins of attraction as a result of greatly increasing the density of the solution space and using Ridge-regression process, as defined in equation (8), on all initial models in the solution space. Here the amount of iterations to reach convergence conditions is also tracked. In each set-up, the same observed model is used and consists of two buried cylindrical sources, separated by 100 m and buried at 10 m, as seen in Figure 85. The bodies are set further apart to create more distinguishable profiles and subsequent larger basins of attraction.

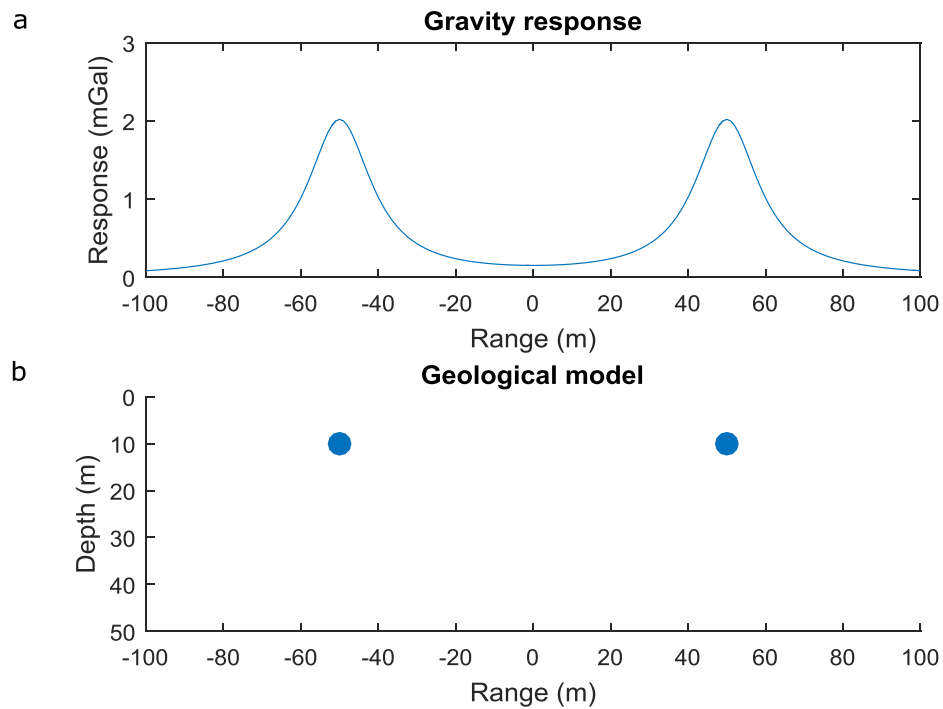


Figure 85: Observed response in *a* as a result of the model in *b*. Model is created with two cylindrical bodies, buried at 10 m and separated by 100 m about the centre of the profile. (See appendix12 – ResponsePlusModel)

Increasing the density of the solution space greatly increases the computational intensity of the problem. The benefits however can often be considered as mainly aesthetic but the basins of attraction can now reveal additional smaller basins of attraction and is essential is observing self-similarity, as was done in Figure 30.

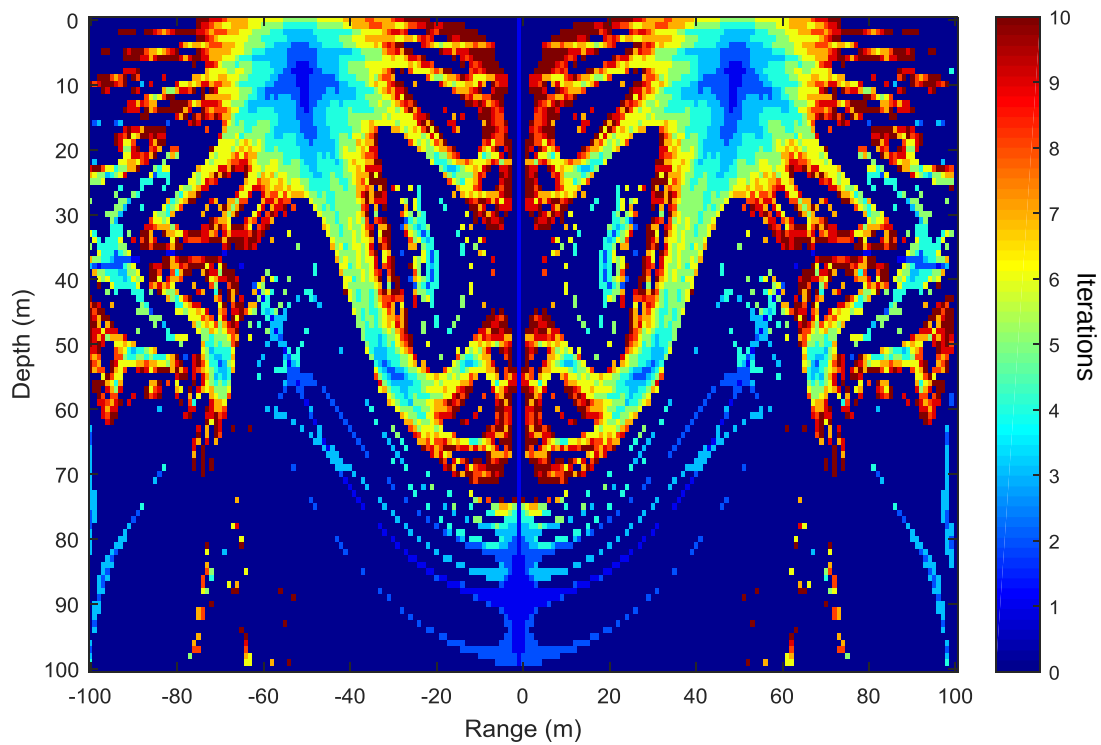


Figure 86: Basin of attraction for a model of two buried cylindrical bodies. The observed model is defined in Figure 85. Standard resolution used in the solution space where every parameter value increases by 1 m. Colorbar indicates the amount of iterations to converge to either body. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image takes 8.9 seconds. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

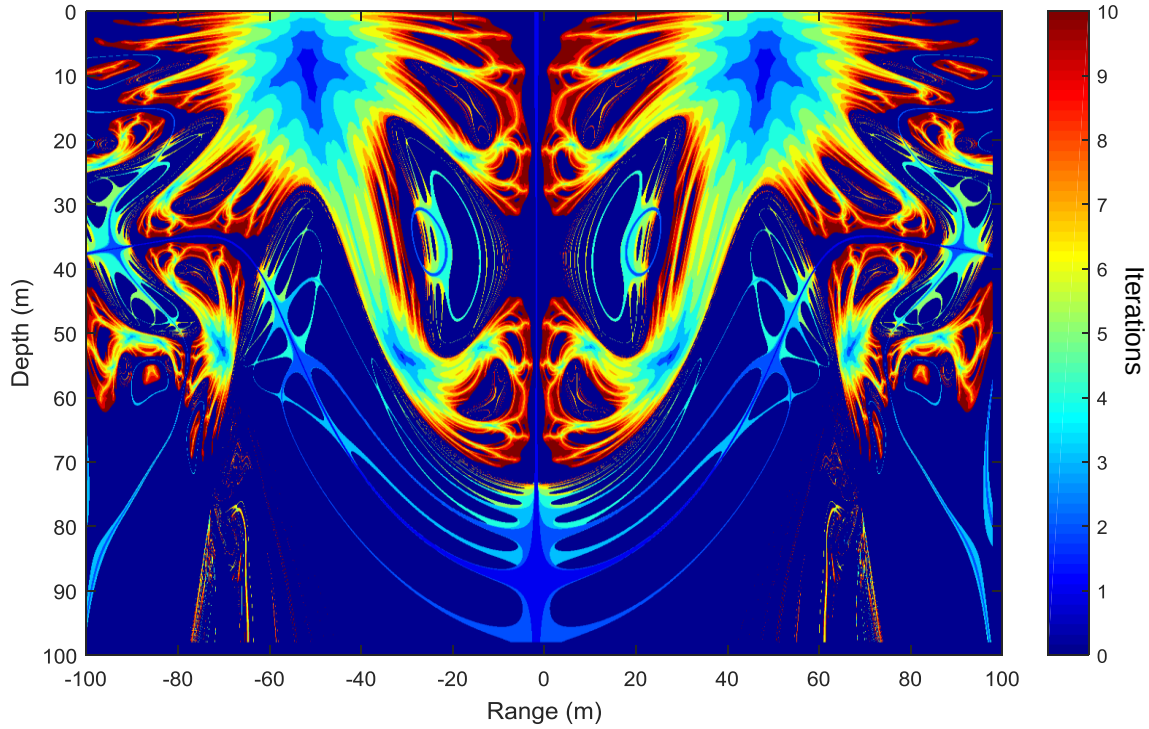


Figure 87: Basin of attraction for a model of two buried cylindrical bodies. The observed model is defined in Figure 85. A resolution one hundred times greater is used. Colorbar indicates the amount of iterations to converge to either body. The Ridge-regression inversion processes in equation (8) are run on all initial models. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. Generation of the basin of attraction image took 15 hours 59 minutes and 35.52 seconds. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

3.7.6 Damping coefficient's effect on the fractal dimension

The damping constant k can be kept constant, and not be allowed to vary throughout the inversion process. Figure 89 - 95 demonstrate different fractal convergence pattern as a result of different damping constants in the ridge-regression inversion process, see equation (8). All systems maintain the same observed model as in Figure 88.

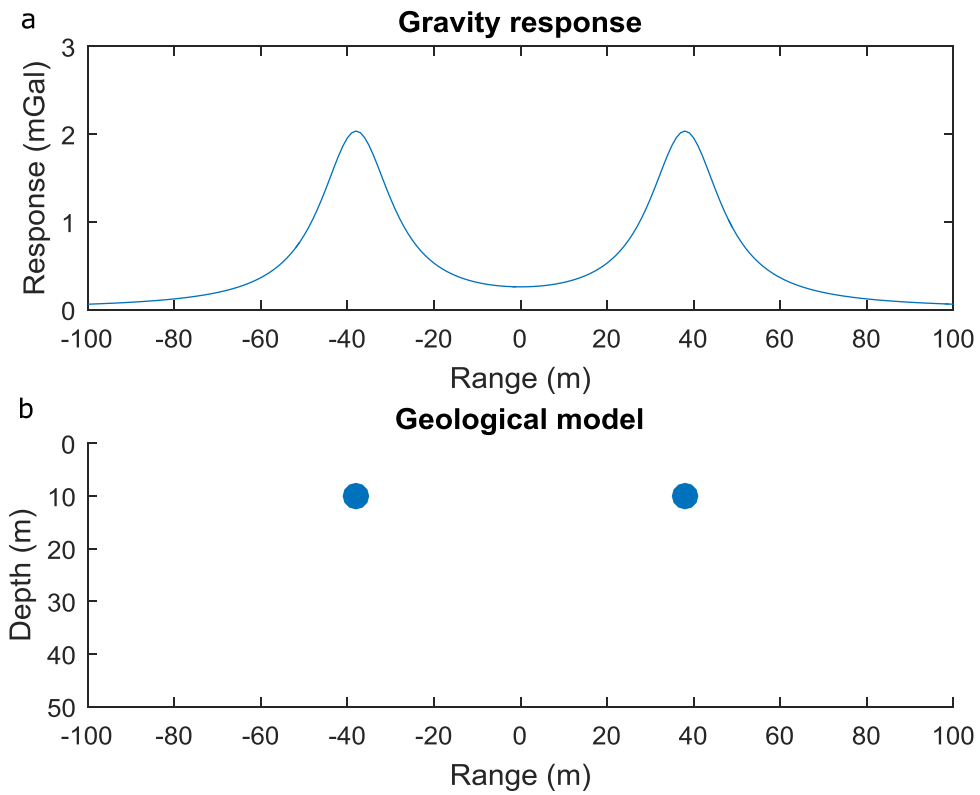


Figure 88: Observed response in *a* as a result of the model in *b*. Model is created with two cylindrical bodies, buried at 10 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel)

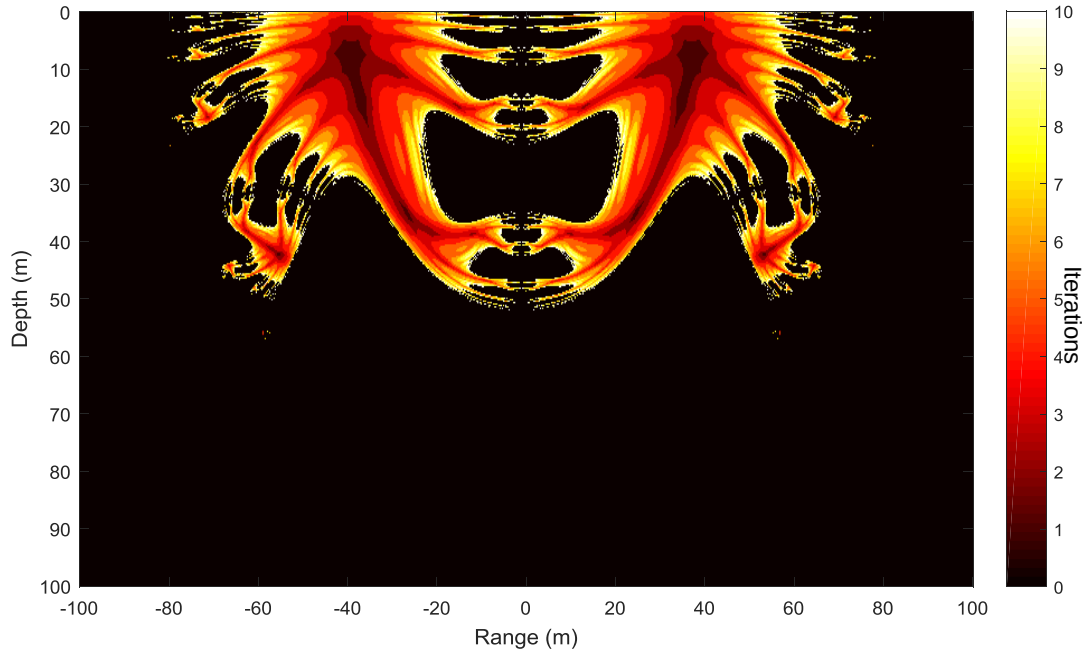


Figure 89: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 0 in, equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

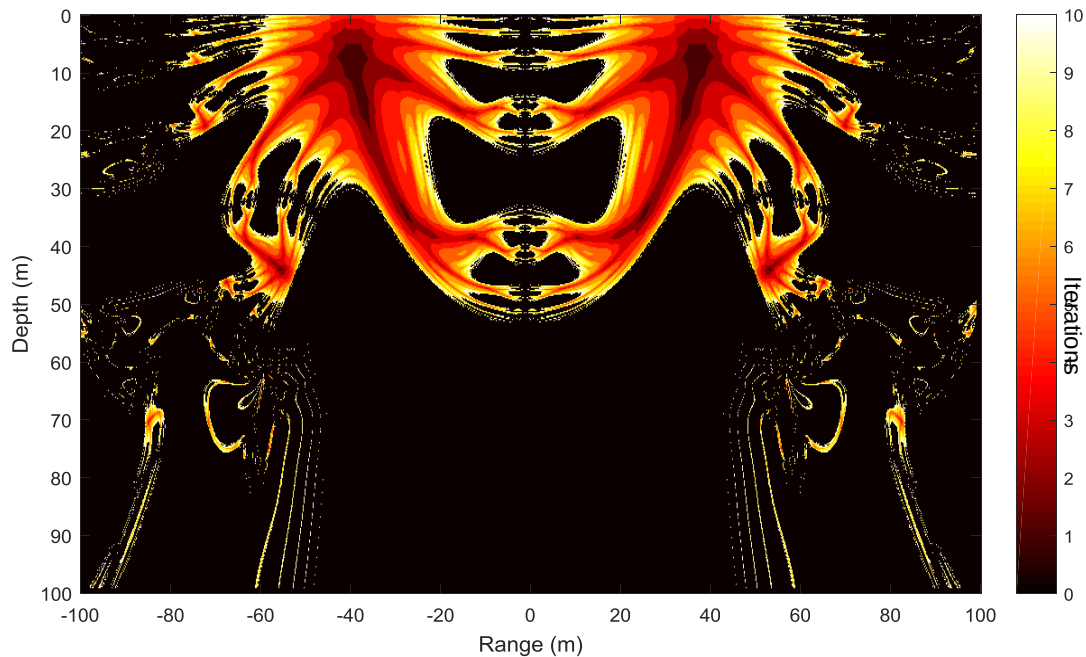


Figure 90: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 2 in, equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

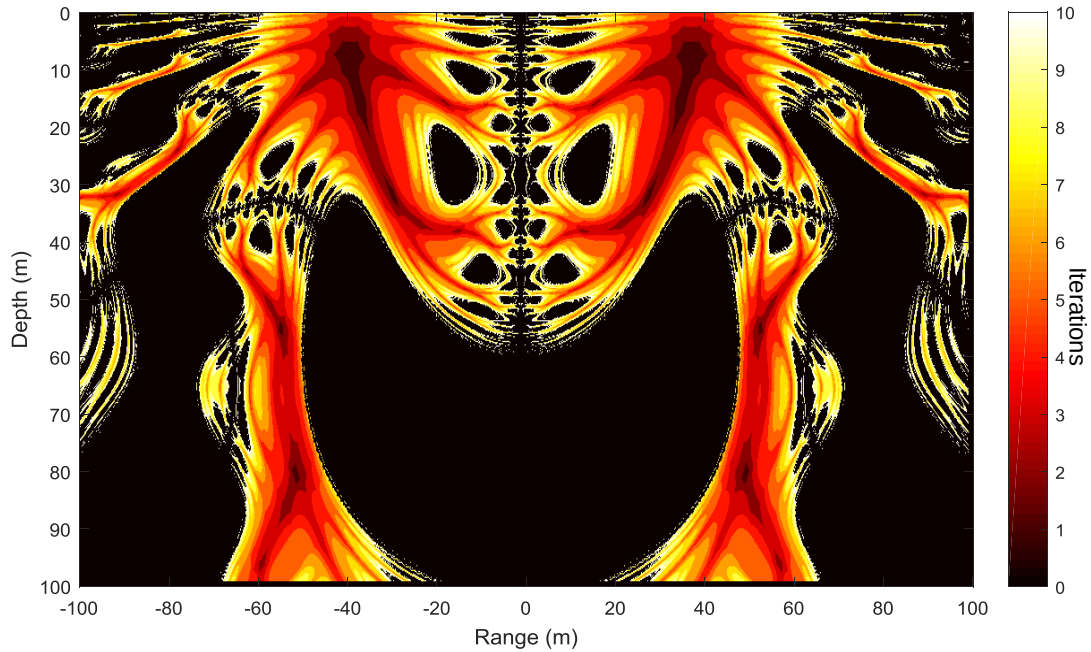


Figure 91: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 8, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

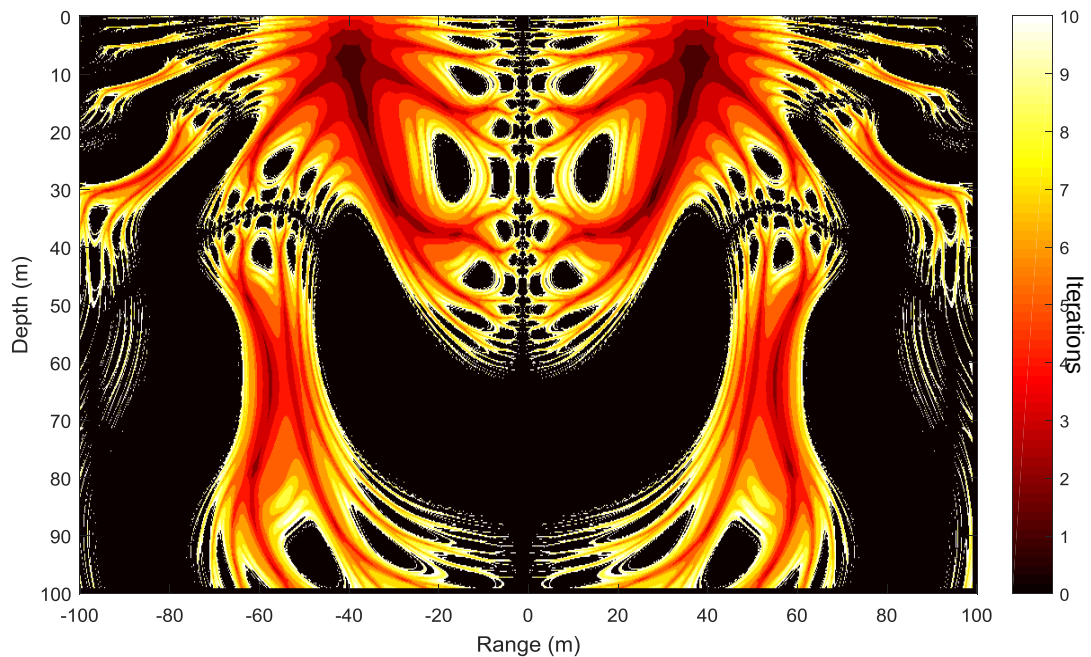


Figure 92: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 10, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

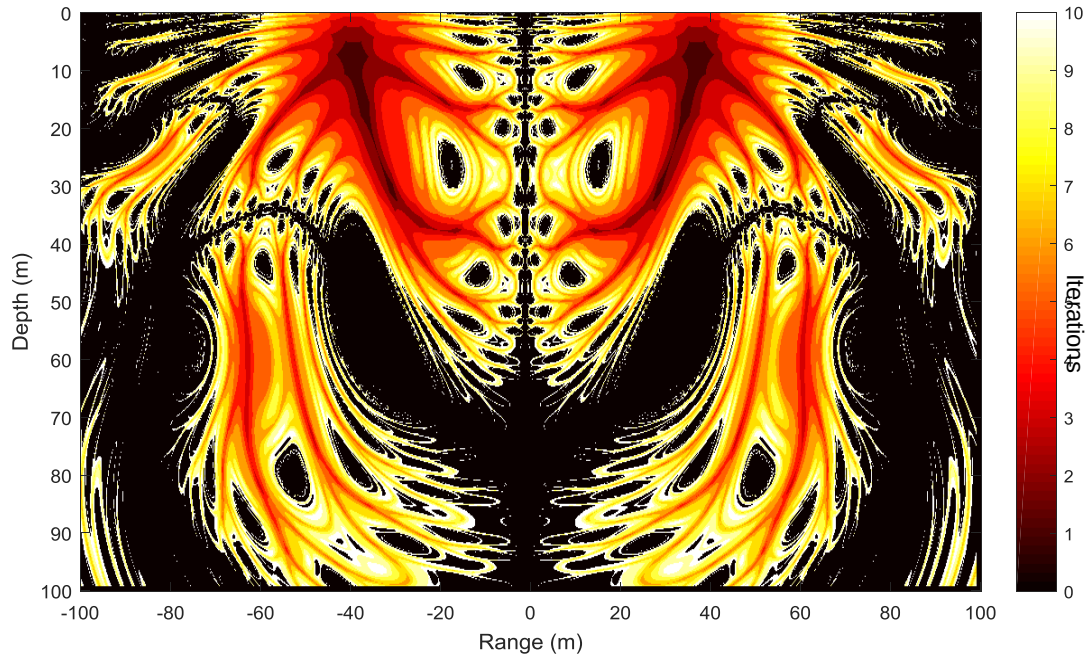


Figure 93: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 12, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

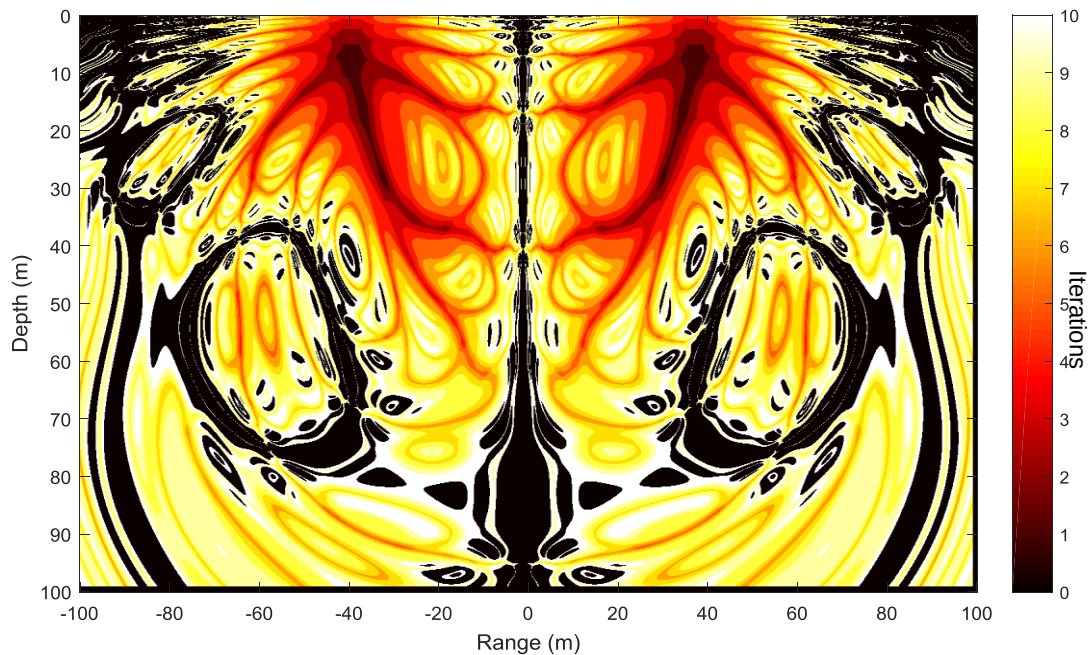


Figure 94: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 18, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

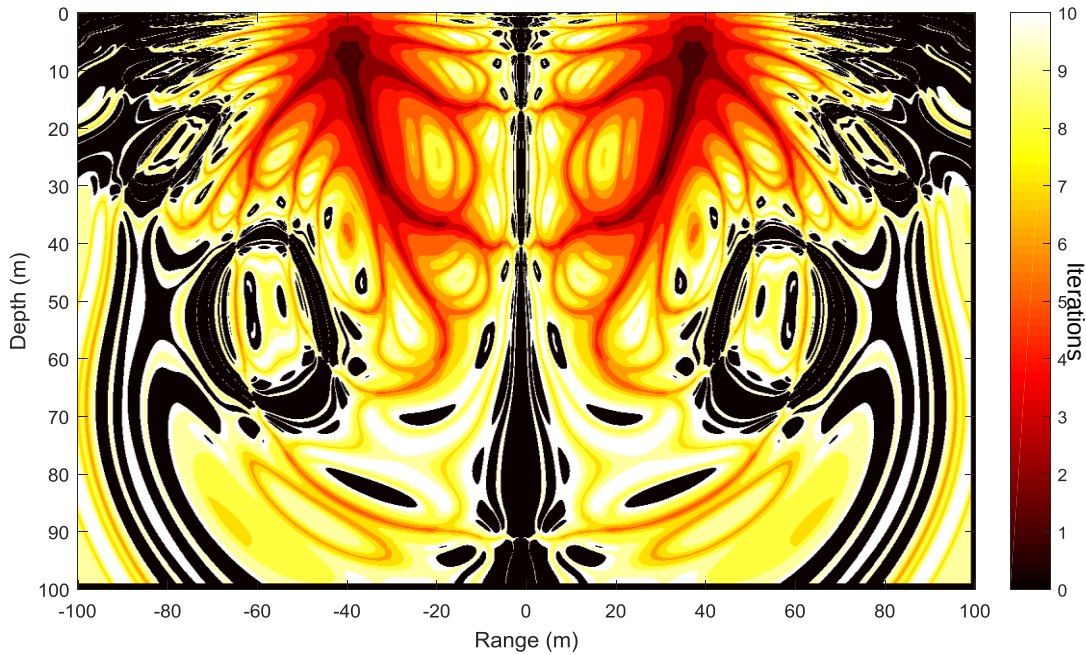


Figure 95: Basin of attraction using the observed model established in Figure 88. The Ridge-regression inversions are run with a constant damping coefficient of 20, in equation (8), on all initial models. The attached colorbar indicate the amount of iterations to converge to either body in the observed model. The black background colour indicates instances where initial model did not converge to observed model. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 10 – BasinOfAttractionKIterationsTrackedTwoBodies)

As was observed in section 3.3.3; as the damping factor is increased the basin of attraction grows larger. This is a result of changing the parameters of the Ridge-regression inversion process in equation (8). The basins of attraction do not grow from the two points solely, but instead, initial models which originally might have diverged now instead meet convergence conditions.

3.7.7 The scaling factors' effect on the basin of attraction and the fractal dimension

It was observed that the dimension and basin of attraction change as the damping factor is changed. Computationally, this is the result of changing the initial conditions of the inversion process. Since altering the scaling factor also affects convergence conditions and initial parameters, its effects on the fractal pattern can also be observed. To illustrate this effect, the observed model seen in Figure 88 is used with the inversion process outlined in equation (13). The basin of attraction is created over a range of scaling factors and the results recorded in Figure 96 and Figure 97.

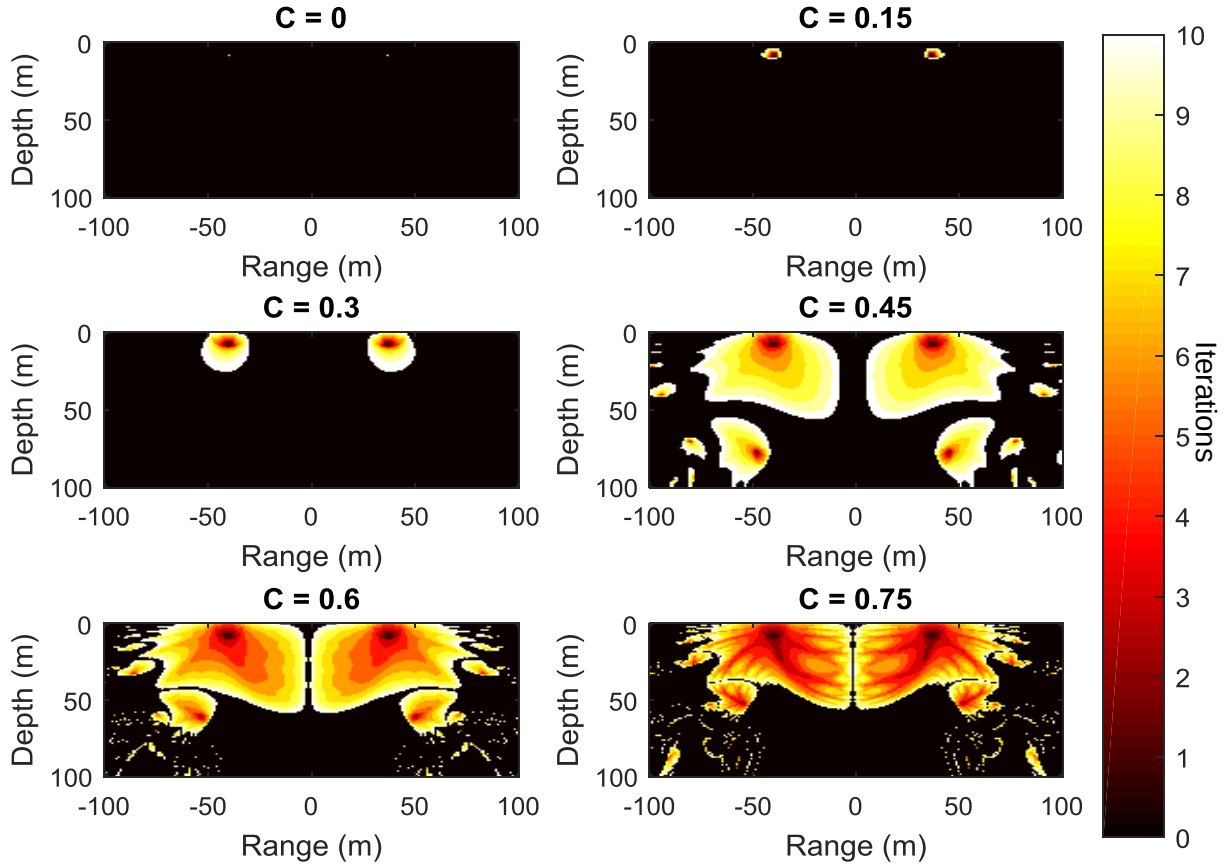


Figure 96: The basin of attraction, as created using the modified Moore-Penrose inversion in equation (13) and various scaling constants C . The scaling constant is varied from 0.00 to 0.75. All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations to converge to either the left or right observed body. The black background indicates initial models which did not meet convergence conditions. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 25 – MultiBasinKorC)

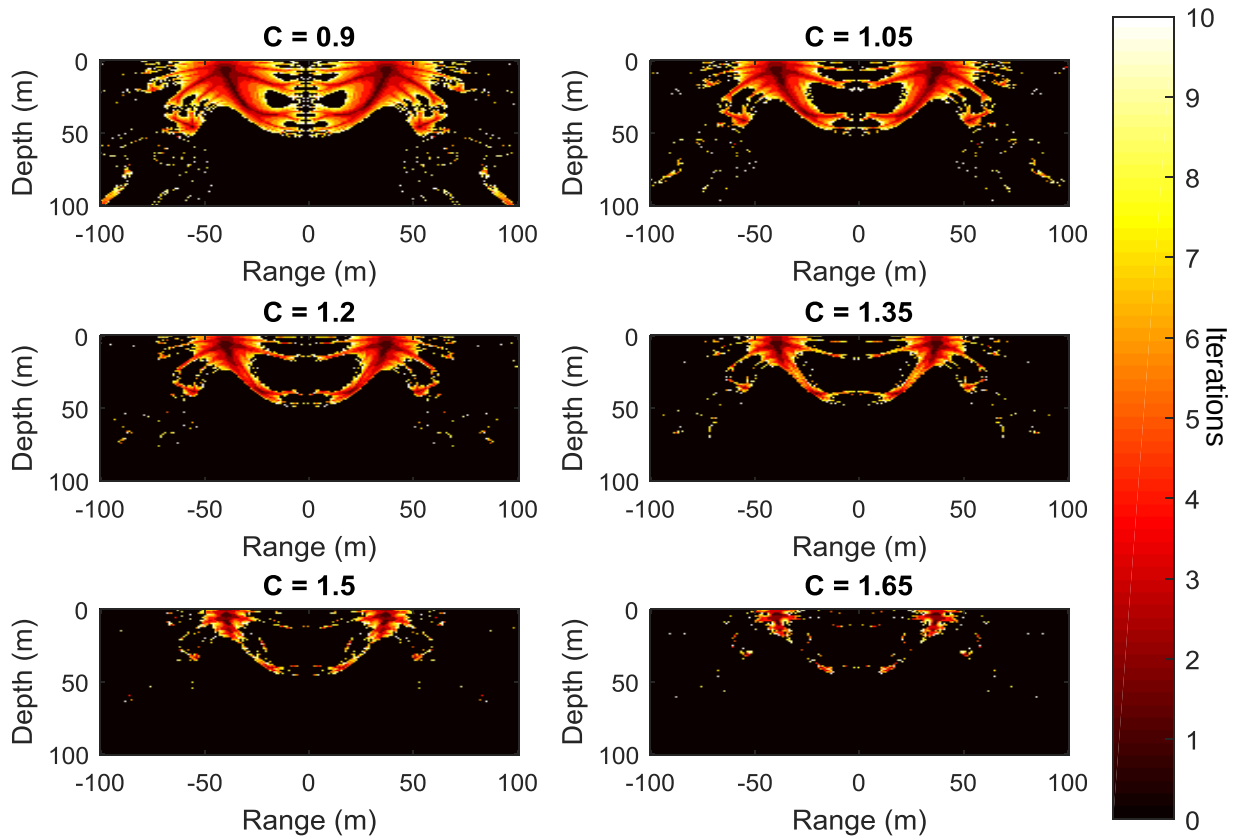


Figure 97: The basin of attraction, as created using the modified Moore-Penrose inversion in equation (13) and various scaling constants C . The scaling constant is varied from 0.90 to 1.65. All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations to converge to either the left or right observed body. The black background indicates initial models which did not meet convergence conditions.

Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 25 – MultiBasinKorC)

3.8 Avoiding local minima

Section 3.7 showed that the inversion process will tend to different minima as influenced by the initial model's location and depth. The scaling factor has been shown to alter the outcomes of the inversion process, see section 3.4. It should be possible to use the scaling factor to alter the parameter values within the inversion process, to steer inversion trajectories out of local minima and toward a global minimum. A new system is defined in Figure 98 with two buried cylinders. Both cylinders have the same mass and radius but are spaced apart. One cylinder is buried shallower than the other and thus would have a smaller minimum on the misfit function.

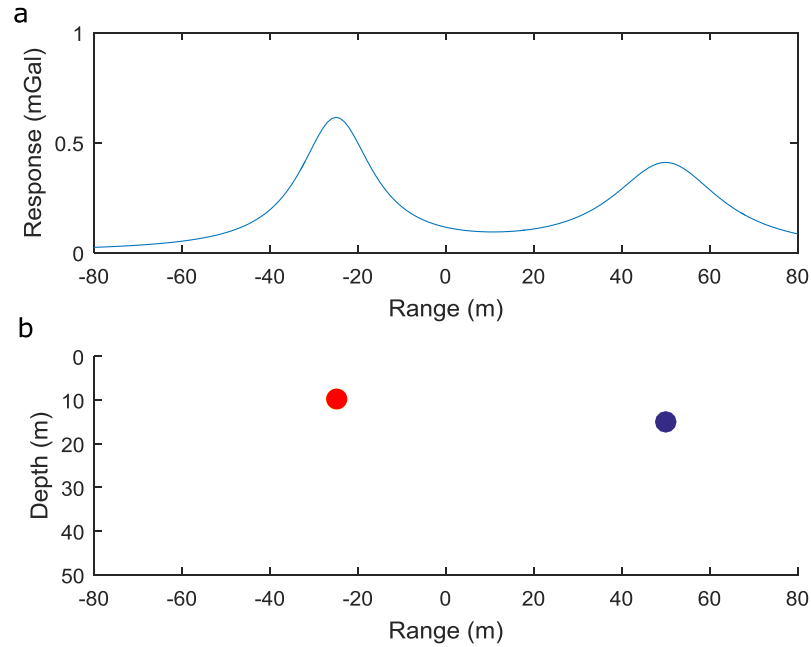


Figure 98: *a* shows the observed gravitational response due to the associated model in *b*. In *b*, the red model represents the left response and the blue model represents the right response. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. (See appendix 12 – ResponsePlusModel)

An initial calculated model is set near the observed right model, at 40 m along the range at a depth of 10 m. The modified Moore-Penrose inversion process, equation (13), acts on the initial model and in Figure 99 it is observed that the initial model's parameter values tend toward the local minimum. Initially the scaling factor is set as 1 and the system is unperturbed. The scaling factor is increased at each iteration for as long as the misfit does not decrease further than the value of the trough of the local minimum. As the scaling factor increases, the trajectories stray further from the local minimum and the scaling factor is only removed once the trajectory enters a new minimum. If the new minimum is smaller in magnitude than the previous minimum, then the trajectory has entered a global minimum and the associated parameter values are accepted as the global minimum's parameter values and the ideal inversion process model, see Figure 99.

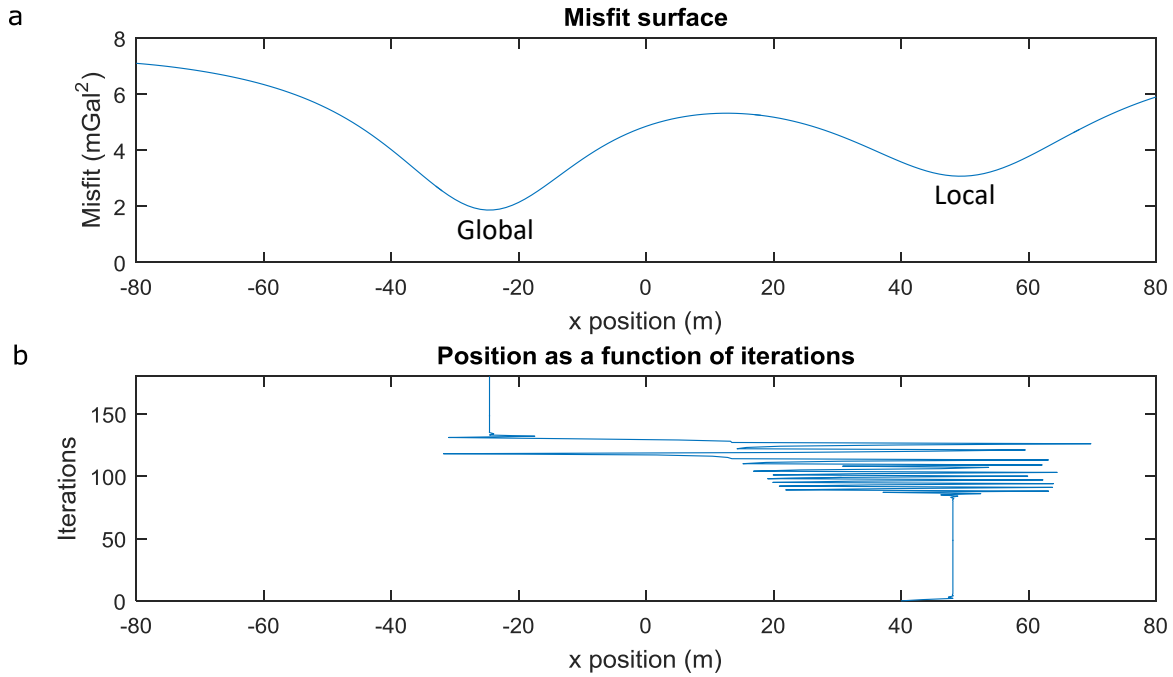


Figure 99: *a* is the misfit function as a result of the initial model parameters and *b* is a plot of the iterations versus the *x* position. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. The initial model is acted upon by the modified Moore-Penrose inversion, in equation (13). The initial model starts at 40 m along the range and is buried at 10 m. In *b*, the *x* position initially tends toward the local minimum and oscillates chaotically about the minimum, which behaves like a strange attractor. Eventually, as the scaling factor is continuously increased, the *x* position tends toward the global minimum and oscillates about the minimum again. (See appendix 3 – MisfitJumpNew)

The increasing scaling factor causes the parameter value to fluctuate and eventually tend toward the global minimum, around which the trajectory also oscillates again, seen in Figure 99. The oscillatory behaviour of the trajectory around the minima indicates that the minima are strange attractors. The parameter values associated with the smallest minimum is then recorded as the global minimum and the ideal solution; this is further demonstrated in Figure 100.

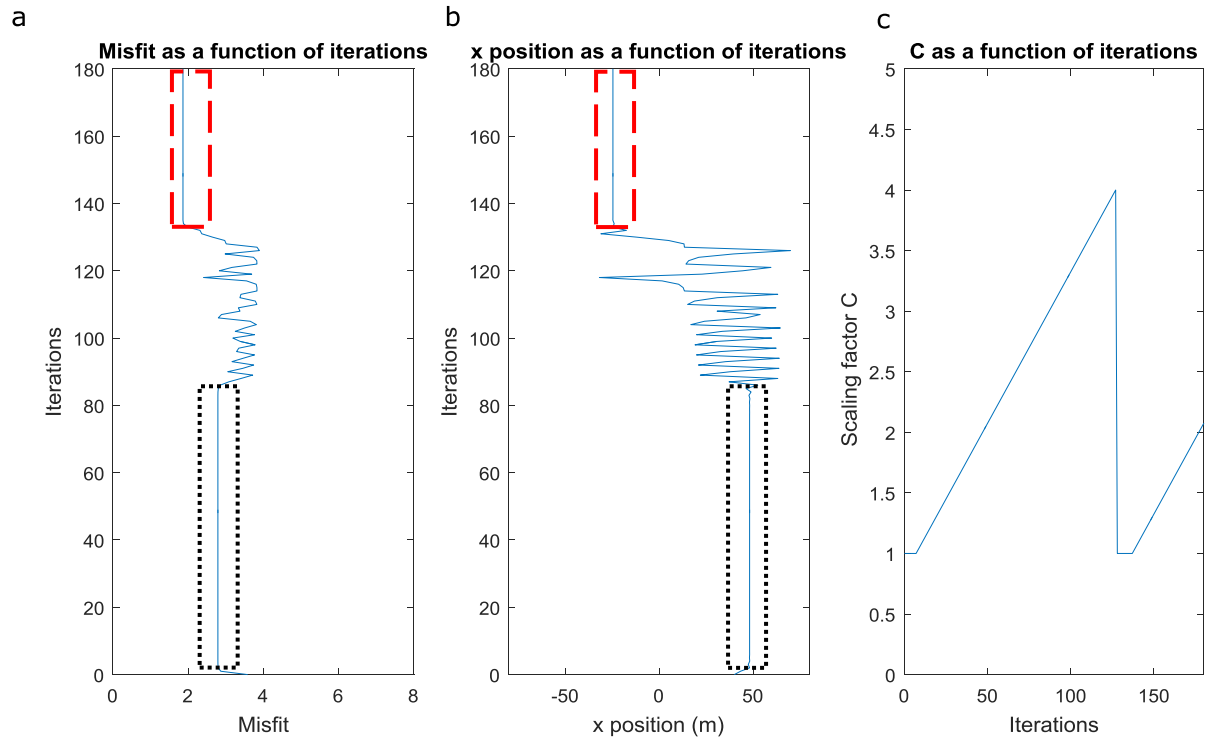


Figure 100: Relationship between the misfit, x coordinate position and the scaling factor. *a* shows the misfit as a function of the iterations within the inversion process. *b* shows the x position as a function of the amount of iterations. *c* is the scaling factor as a function of the iterations. The black and red boxes indicate the parameter values associated with the corresponding local and global minima, respectively. The left model is at -25 m on the range and is buried at a depth of 10 m. The right model is at 50 m on the range and is buried at a depth of 15 m. The initial model is acted upon by the modified Moore-Penrose inversion in equation (13). The initial model starts at 40 m along the range and is buried at 10 m. (See appendix 3 – MisfitJumpNew)

By using the scaling factor, the inversion process yields an x position of -24.6 m, which is within acceptable bounds of the true x location value of -25 m.

3.9 Three dimensional parameter basins of attraction

The inversion processes can represent the simultaneous inversion of many parameters. If the inversion process is set up to invert for three parameters, then a solution space of three dimensions can be used to create a three-dimensional basin of attraction. The geophysical model has to then be established in three dimensions. A synthetic observed gravity dataset can be created by modelling the gravitational response of a buried sphere. The model is set-up with respect to only a horizontal position (x parameter value) and a depth (z parameter value) to the centre of the sphere model. The analytical solution to the gravitational response due to a buried

sphere is given in equation (20), where all parameters have been defined in section 1.3.6 (Telford et al, 1990).

$$g_{sphere} = \frac{Gmz}{(x^2+z^2)^{\frac{3}{2}}} \quad (20)$$

In equations (20) – (26) the m parameter is again only used for computational purposes, the responses use the density of a buried sphere model with a unit radius of 1 m. As in section 1.3.6, the partial derivatives in matrix A , introduced in the Moore-Penrose equation (7), with respect to each respective parameter can be analytically derived or numerically calculated. Equations (21) and (22) are partial derivatives of g with respect to x and z for a sphere model.

$$a_x = \frac{\delta g}{\delta x} = \frac{-3Gmxz}{(x^2+z^2)^{\frac{5}{2}}} \quad (21)$$

$$a_z = \frac{\delta g}{\delta z} = \frac{Gm(x^2-2z^2)}{(x^2+z^2)^{\frac{5}{2}}} \quad (22)$$

In a three parameter solution space, the addition of a y parameter is introduced to track the inversion in three dimensions. The gravitational response of a buried sphere is then redefined in equation (23).

$$g_{sphere} = \frac{Gmz}{(x^2+y^2+z^2)^{\frac{3}{2}}} \quad (23)$$

Figure 101 is the result of using equation (23) to construct a gravitational response with an x position of 0 m, a y position of 0 m, and a depth of 25 m.

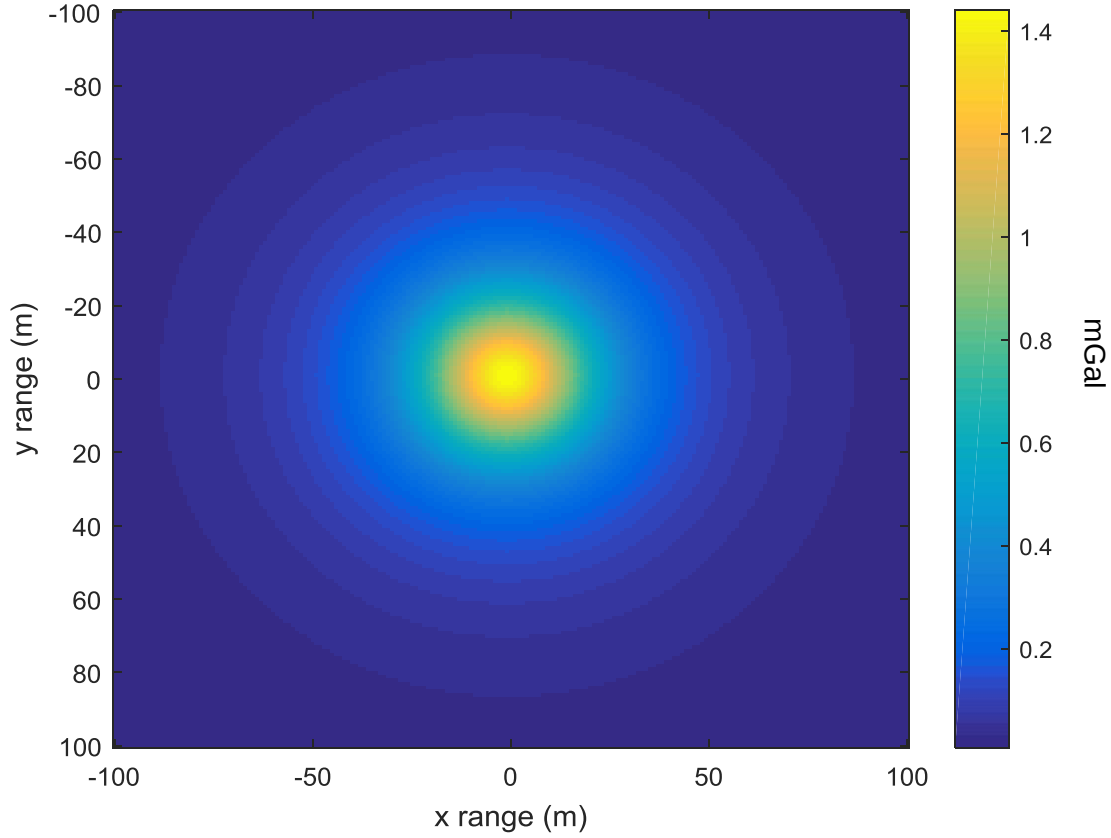


Figure 101: A synthetic observed gravitational surface response as generated using equation (23). The response is the result of a buried sphere centred at 0 along the x and y range and buried at 25 m. (See appendix 20 – ThreeParametersFRAC)

The inversion process, as set up in the Moore-Penrose inversion, equation (7), can be set up to act on three parameters. The matrices of partial derivatives, A , is adjusted from equations (21) and (22) to include the y parameter. The A matrix for the sphere models are concatenated from equations (24) - (26) which are derivatives of the sphere response with respect to x , y and z , respectively.

$$a_x = \frac{\delta g}{\delta x} = \frac{-3Gmxz}{(x^2+y^2+z^2)^{\frac{5}{2}}} \quad (24)$$

$$a_z = \frac{\delta g}{\delta z} = \frac{Gm(x^2+y^2-2z^2)}{(x^2+y^2+z^2)^{\frac{5}{2}}} \quad (25)$$

$$a_y = \frac{\delta g}{\delta y} = \frac{-3Gmyz}{(x^2+y^2+z^2)^{\frac{5}{2}}} \quad (26)$$

A model of two identical buried spherical bodies is established using the observed response established in Figure 101 and a calculated response created in the same way using equation (23) again with offset parameter values. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at a depth of 20 m. The corresponding model is shown in Figure 102.

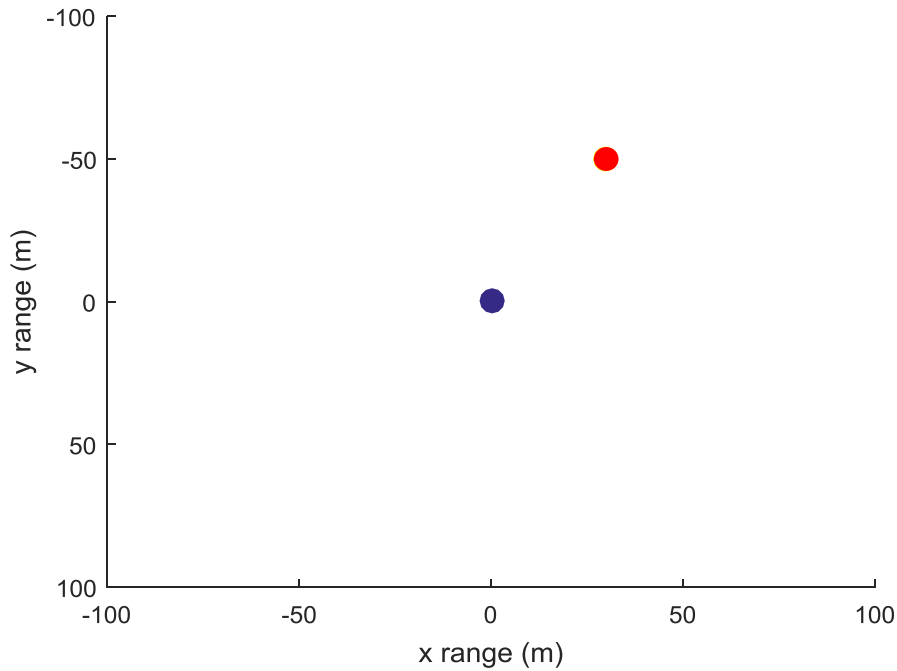


Figure 102: Model for two buried spherical sources. The observed and calculated models are represented by the blue and red plots, respectively. The observed model is placed at 0 m along the x range, 0 m along the y range and is buried at 25 m. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at 20 m. (See appendix 20 – ThreeParametersFRAC)

The overlaid observed and calculated responses, as a result of the model defined in Figure 102, are seen in Figure 103.

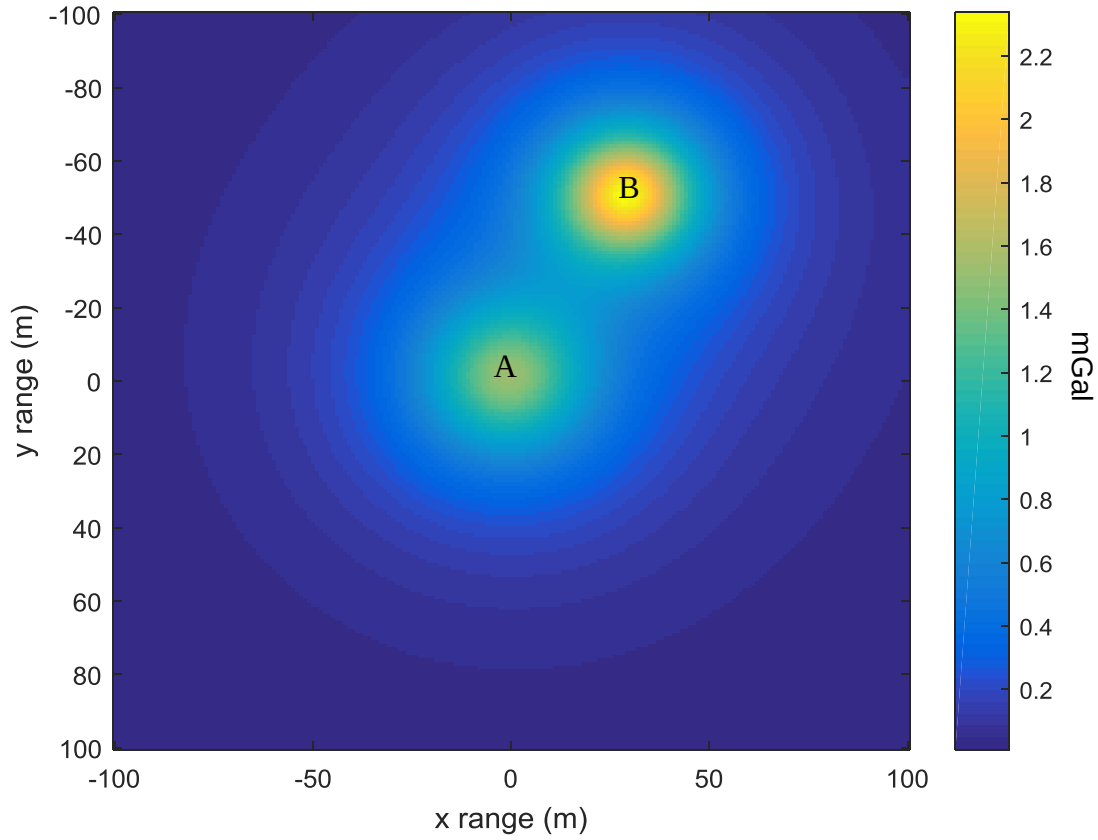


Figure 103: Observed and calculated gravitational responses of a buried sphere system. *B* is the calculated forward model and *A* is the observed response. The observed model is placed at 0 m along the *x* range, 0 m along the *y* range and is buried at 25 m. The initial calculated model is placed at 30 m along the *x* range, -50 m along the *y* range and is buried at 20 m. (See appendix 20 – ThreeParametersFRAC)

Using the established inversion process and accounting for a third parameter, the Moore-Penrose inversion process in equation (7) is run. During inversion, the parameters are tracked to trace the process, which is shown in Figure 104.

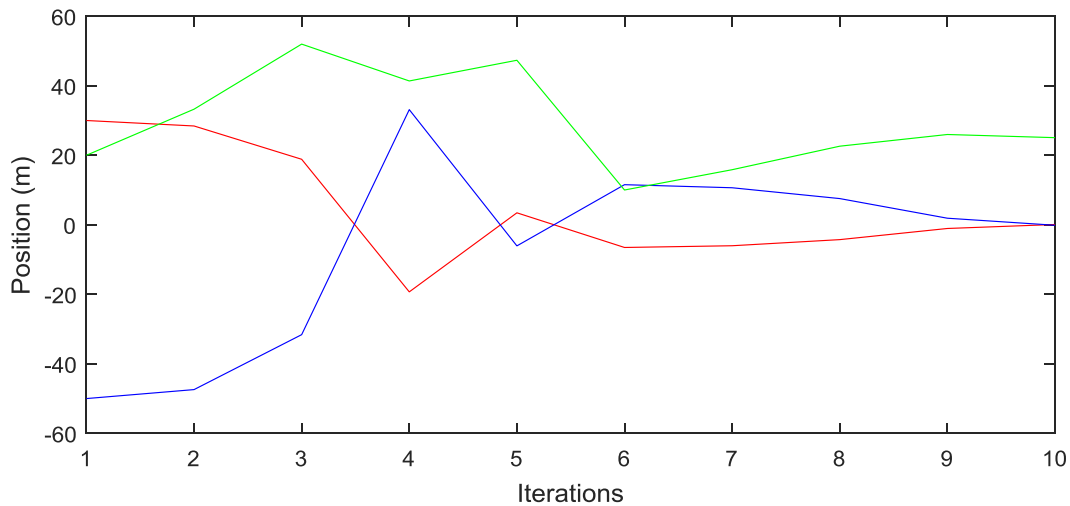


Figure 104: The red, blue and green trajectories track the parameter values of the x , y and z position, respectively as they are acted upon by the Moore-Penrose inversion, see equation (7). The observed model is placed at 0 m along the x range, 0 m along the y range and is buried at 25 m. The initial calculated model is placed at 30 m along the x range, -50 m along the y range and is buried at 20 m. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. As the trajectories never stray further than 60 m away from 0 m, only that range is shown to highlight trajectories. (See appendix 20 – ThreeParametersFRAC)

The basin of attraction is created again, as in section 2.3, within a three-dimensional solution space. The inversion is again run from each parameter value set and a record is taken of whether or not convergence is reached for each point in the solution space. Figure 105 shows a three dimension solution space and an iso-surface of the basin of attraction. The inversion process for three parameters is less linear; therefore the convergence conditions are more difficult to achieve.

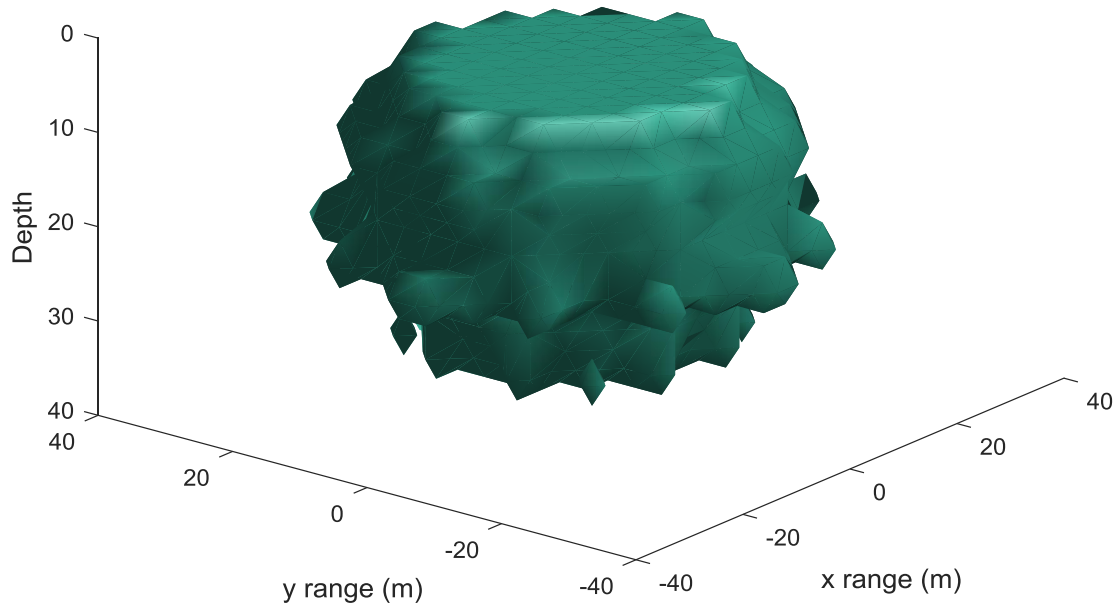


Figure 105: Three-dimensional basin of convergence; with the Moore-Penrose inversion process, in equation (7), run over a range of x , y and z positions. The observed model is as established in Figure 103. This solution space is set up in increments of 5 m as a proof of concept. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 20 – ThreeParametersFRAC)

The fractal-nature of the iso-surface is difficult to see in Figure 105 due to the resolution and size of the solution space. At larger solution spaces and finer intervals, more isolated iso-surfaces are seen, see Figure 106.

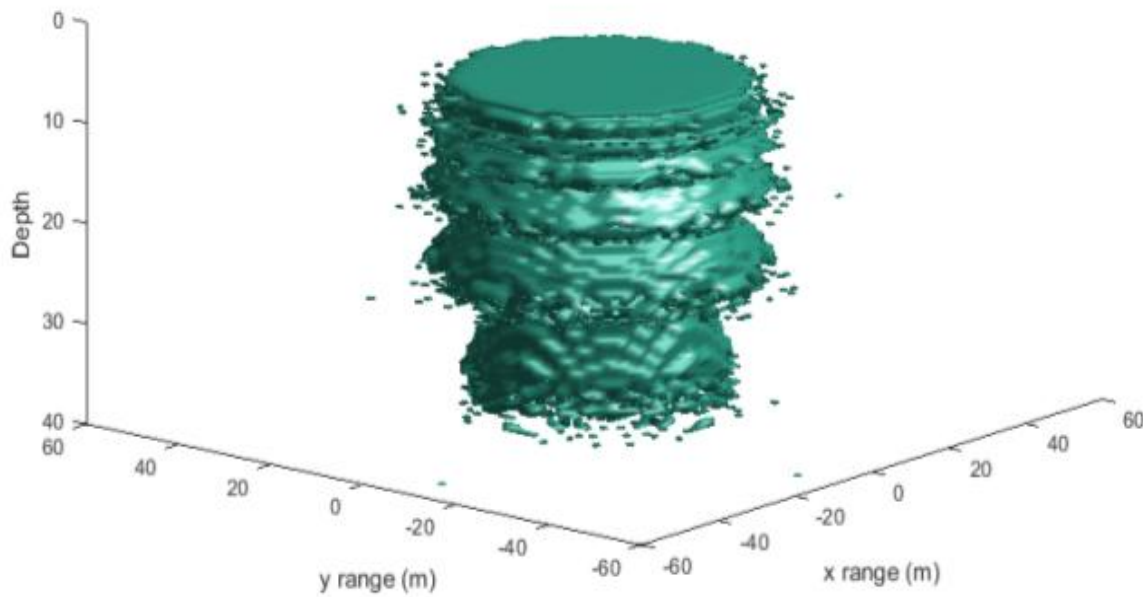


Figure 106: Three-dimensional basin of attraction with the Moore-Penrose inversion process, as set out in equation (7) run over a range of x , y and z parameter values. The observed model is as established in Figure 103. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 20 – ThreeParametersFRAC)

An alternate approach to constructing a three-dimensional basin of attraction is to run inversion processes over the x and y range while keeping the depth constant. This creates a two-dimensional fractal convergence pattern within the solution space. By keeping the depth component constant, the inversion process is made simpler and more values in the parameter space are convergent. An example of a horizontal fractal pattern at a fixed depth is seen in Figure 107, where the system is reduced to a horizontal slice.

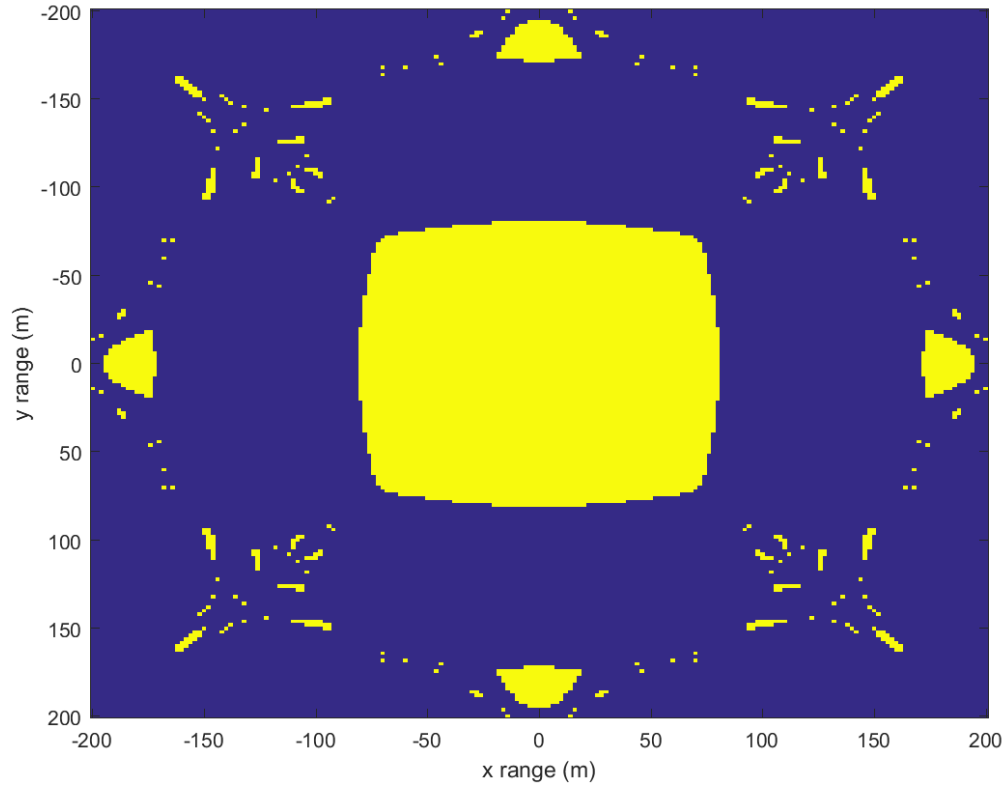


Figure 107: Basin of attraction in the solution space. The observed model is as established in Figure 103. Blue and yellow represent parameter values which met divergent and convergent conditions, respectively. At each initial model, the Moore-Penrose inversion process, in equation (7), is run to test convergence conditions. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D)

Repeating this process and incrementally sweeping the z parameter creates additional two-dimensional fractal convergence patterns. Stacking these slices on top of each other constructs a pseudo-three-dimensional model. A new system is defined, to alleviate computational intensity, with a smaller range and with the centre of a buried sphere model placed at 125 m along the x range, 125 m along the y range and buried at a depth of 30 m, see Figure 108.

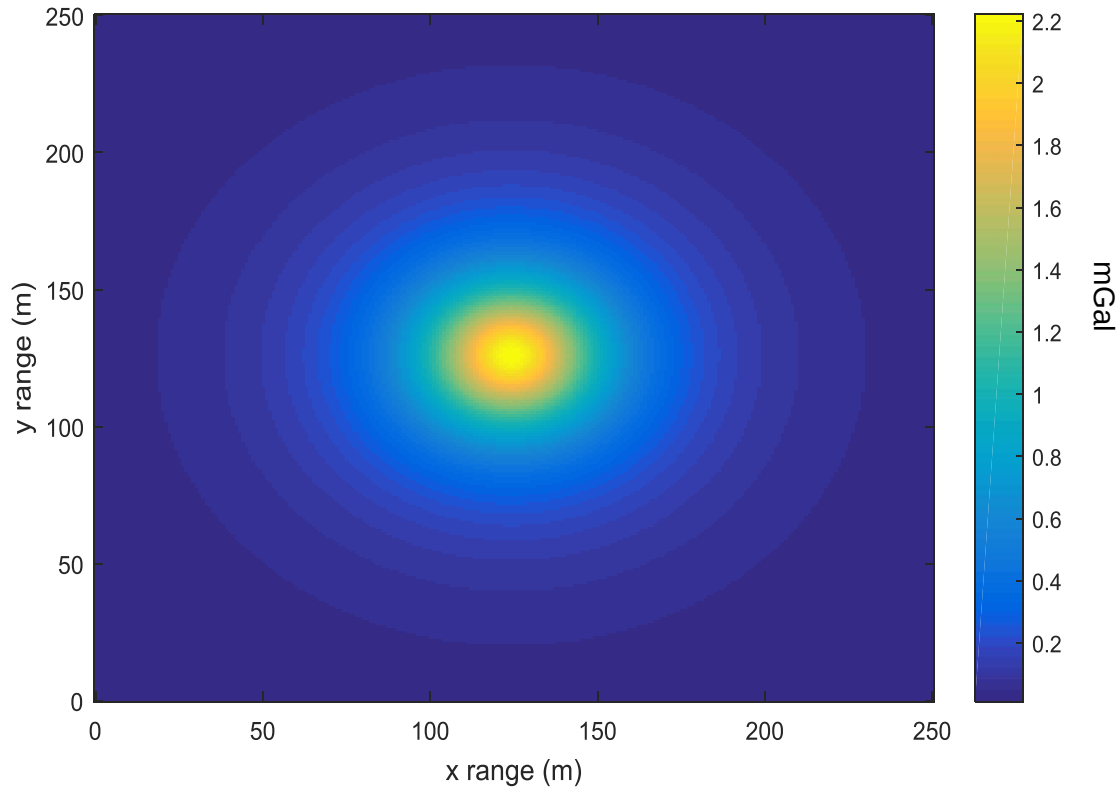


Figure 108: Observed gravitational response due to a buried sphere model. The observed model is placed at 125 m along the x range, 125 m along the y range and is buried at 25 m. (See appendix 20 – ThreeParametersFRAC)

Figure 109 is the result of the Moore-Penrose inversion processes, in equation (7), run at an offset depth ($z = 56$ m) within this system, monitoring only a slice in the x-y plane.

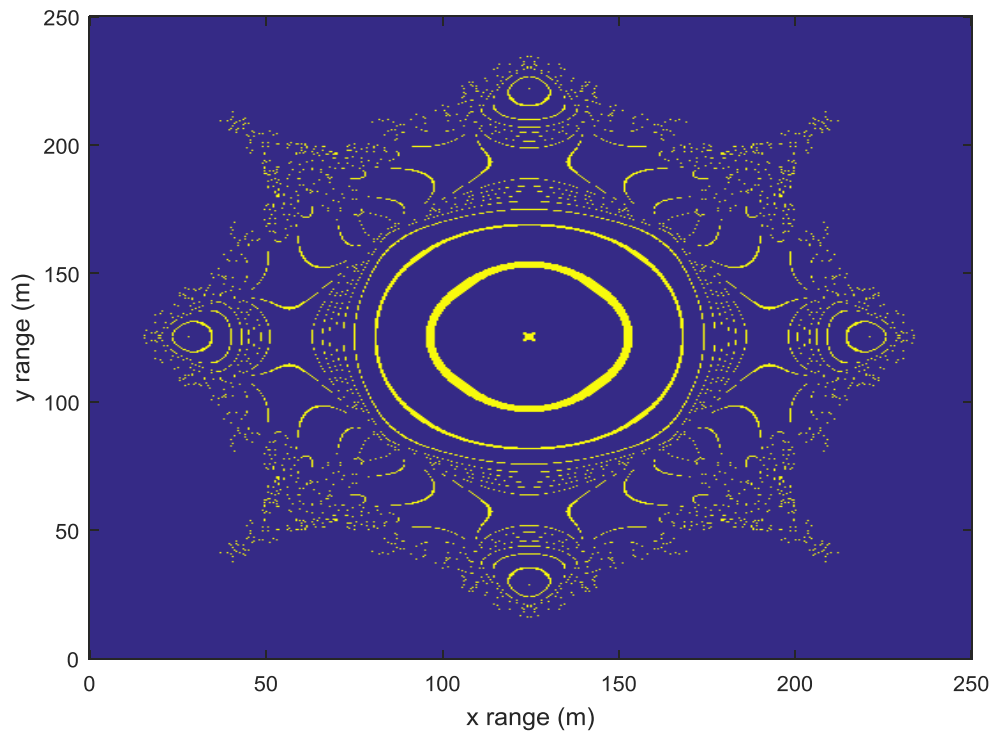


Figure 109: Basin of attraction in solution space with $z = 56$ m. The observed model is as established in Figure 108. Blue and yellow represent parameter sets which met divergent and convergent conditions, respectively. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D)

The slice shown in Figure 109 shows an intricate pattern and suggests it is still possible to have successful initial models even when the displacement in depth is as great as it is. This is a result of simplifying the problem and complying three-dimensional basin of attraction with an array of two-dimensional basins of attraction. Figure 109 is used as the base within the three-dimensional system. Incrementing the value of z repeatedly and stacking the slices results in a pseudo-three-dimensional pattern, shown in Figure 110.

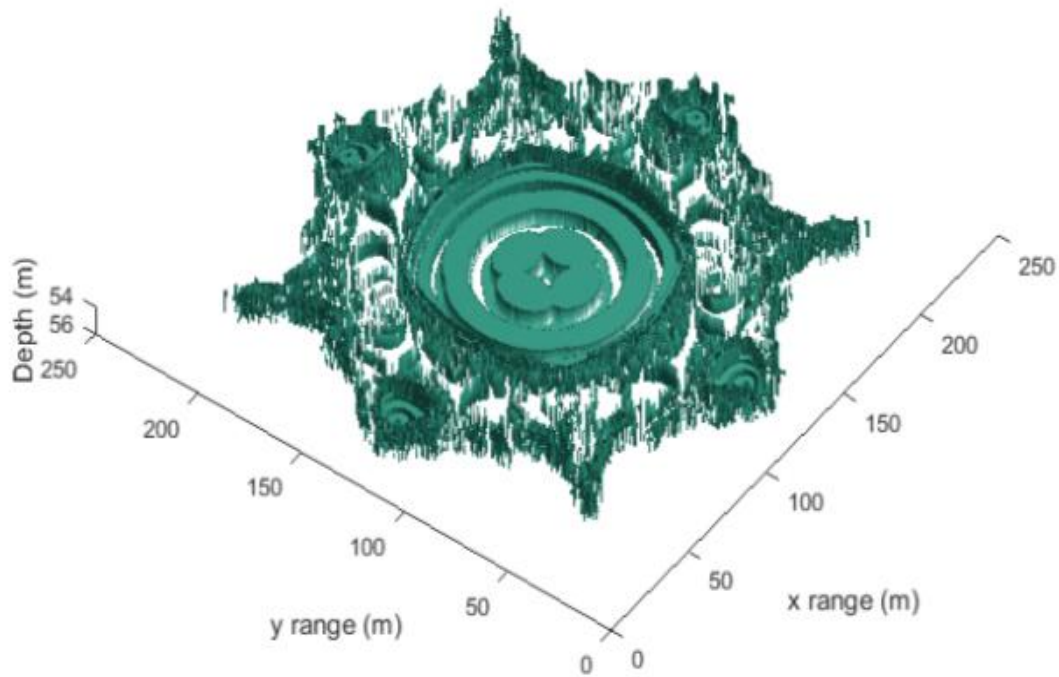


Figure 110: A three-dimensional basin of convergence model created by stacking three two-dimensional basins of attraction. The observed model is as established in Figure. The three two-dimensional convergence patterns are created from three processes where the x and y parameters are varied, but the depth is fixed at 54 m, 55 m and 56 m for the slices. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D)

To create a full three-dimensional model, the depth is varied from 56 m to 0 m and all the individual slices are stacked together, see Figure 111 and Figure 112.

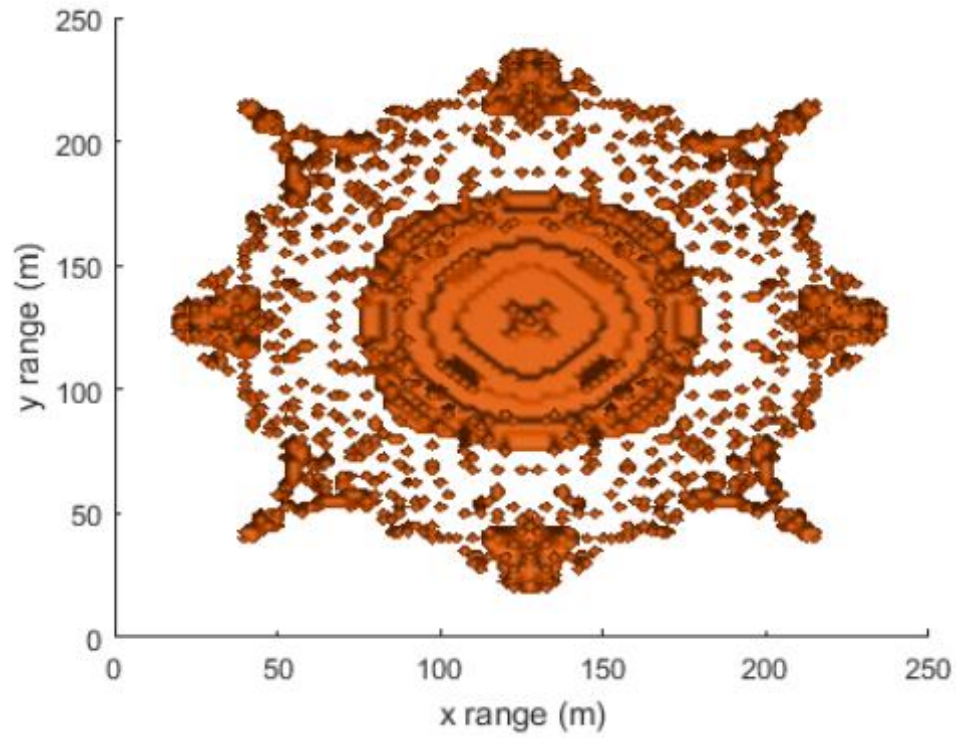


Figure 111: Top view of pseudo three-dimensional basin of attraction model. The observed model is as established in Figure. The ring structure around the centre is now more visible. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D)

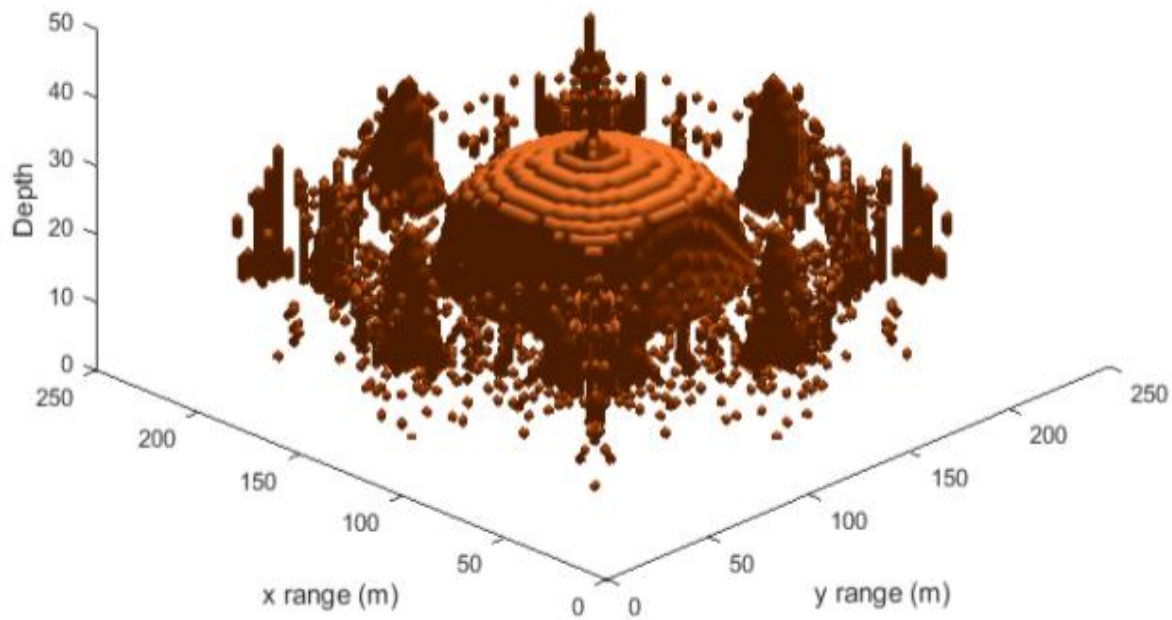


Figure 112: Pseudo three-dimensional basin of attraction model. The observed model is as established in Figure. The ring structure around the centre is again visible. Convergence conditions are met when the final model's location and depth lie within 1 m of the observed model. (See appendix 21 – Pseudo3D)

A pseudo three-dimensional model can be constructed more for visual appeal, as increased convergence parameters create more elaborate fractal patterns. It is often also done to reduce computational strain; parameters reach convergence faster in a two dimensional solution space. Figure 106 and Figure 112 are both constructed in approximately the same amount of time (roughly 48 hours) despite Figure 112 taking readings over a larger solution space.

CHAPTER FOUR – APPLICATION TO GRAVITY DATA FROM SOUTH AFRICA

It is not worthwhile to only test the applications of chaos and fractals on synthetic data. Real data introduces noise and non-Euclidean geometries. A sample of gravity data in the area of Trompsburg is used. The data is the culmination of approximately 98000 data points gridded to a 1km interval. No data is interpreted here but instead only applications of chaos and fractals are tested against the data sets. A lot of work has already been done in modelling the area, such as the work done by Rezaie et al. (2017) or Mare, L and Cole, J (2006). The Trompsburg Complex is a layered mafic intrusion and is named after the nearby town. Trompsburg is a town situated within the Free State province in South Africa and the location is indicated by the yellow pin in Figure 113.



Figure 113: Trompsburg as situated in South Africa at 30.032 S and 25.779 E. (Google Earth, 2017)

The Trompsburg Complex data is chosen due to its high isolated gravity anomaly, which resembles the gravity response as generated by a buried sphere; see Figure 101 and section 3.9. Figure 114 shows a relative gravity map of the Trompsburg Complex, created by subtracting the smallest value from the entire set.

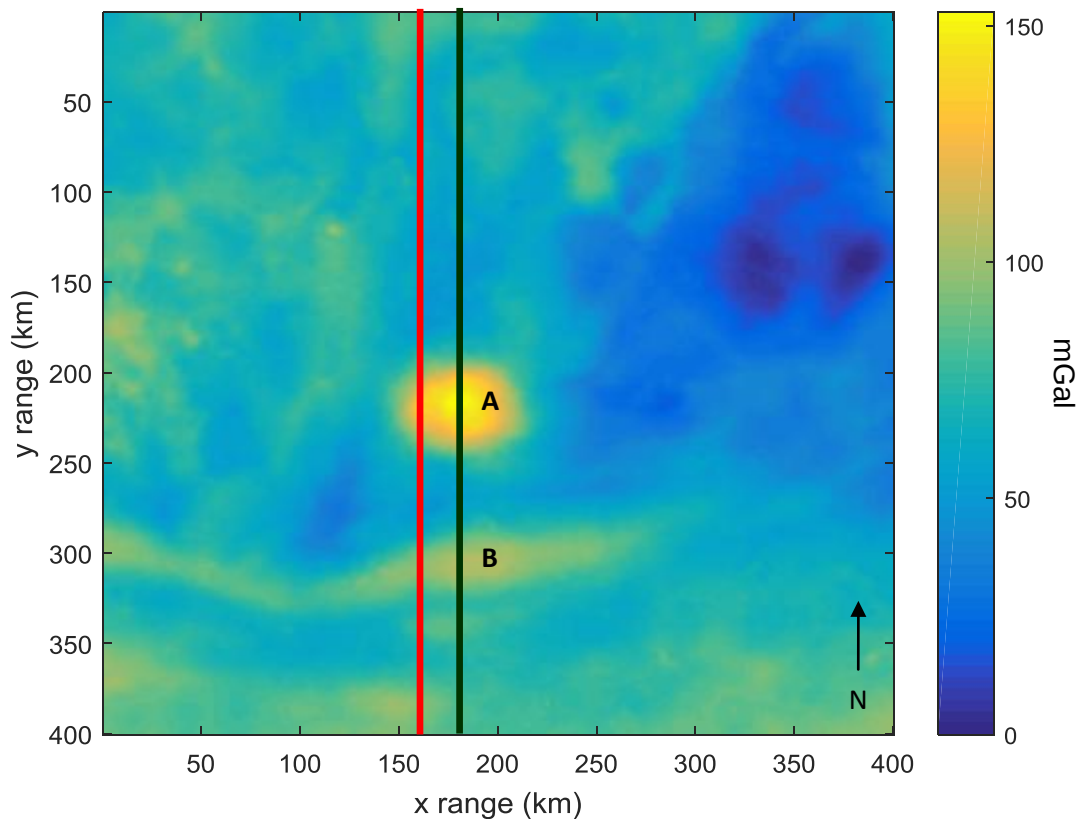


Figure 114: The relative gravitational anomaly over the Trompsburg Complex. The attached colorbar is used to gauge the relative gravitational intensity in mGal. The red and black lines show sample lines run at x ranges of 163 km and 183 km, respectively. *A* indicates a potential central spherical response and *B* indicates a potential linear response running east to west. (See appendix 22 – mapREAL)

4.1 Basin of attraction

In order to construct the fractal convergence pattern, first a profile is extracted from the Trompsburg anomaly map. Figure 115 is the relative response from a longitudinal line taken at longitude 183 km running south to north as shown with the black line in Figure 114.

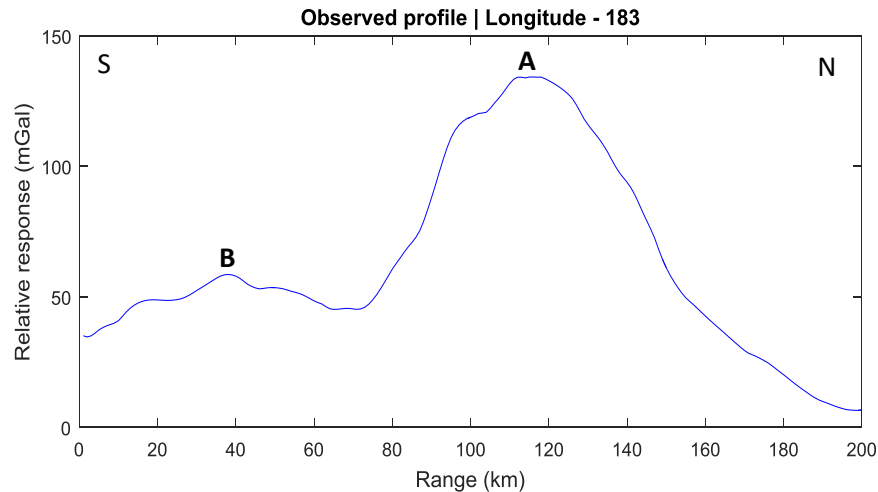


Figure 115: The relative gravity anomaly extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. *A* and *B* correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump)

The two peaks along the profile can be represented by different geometric models depending on the desired objective. The greater northern peak, *A*, exhibits characteristics of a buried spherical body, while the anomaly to the south, *B*, can be represented by a buried cylindrical body since it extends further to the east and west. The northern peak is larger than the southern peak and would result in a smaller misfit along the misfit function. In order to establish convergence conditions for the global minimum; forward models for buried spherical bodies are constructed for the two peaks. Existing research has been done into the shape and depth of the Trompsburg complex and this information can be used to constrain the inversion process. These constraints are not employed in this section however. This is done to allow the inversion trajectories to more closely mimic the conditions established in CHAPTER THREE, namely only a relative observed response and a forward model response.

To create a forward model of a buried sphere system, equation (20) is used, which is the response due to a buried sphere in two dimensions. In order to create the basins of attraction, the Moore-Penrose inversion processes, in equation (7), use the partial derivative matrices with respect to position and depth, as set up in equations (21) and (22) for a sphere model. The extracted profile, both forward model responses and the accompanying basins of attraction are shown in Figure 116.

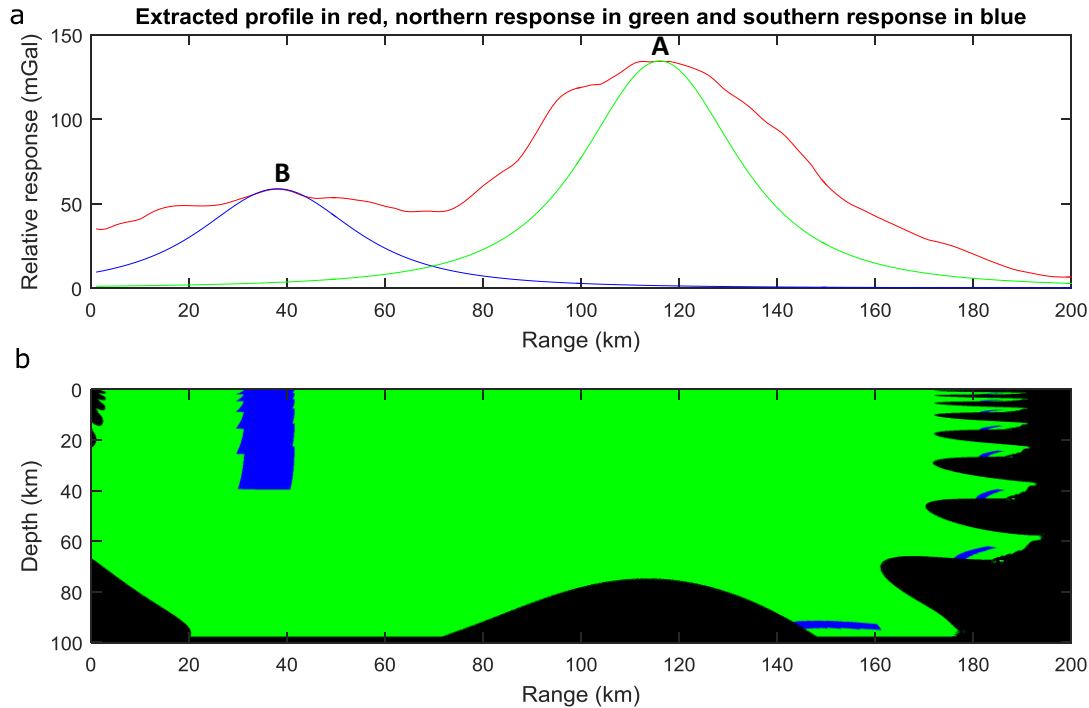


Figure 116: *a* illustrates the extracted profile in red and the forward models for the east-west feature and the central peak in blue and green, respectively. *b* is the associated basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the northern body and convergence to the southern body, respectively. Convergence conditions are met when the final model's location and depth lie within 5 m of the observed model. The southern forward, *B*, model is at 38 km along the range and is buried at 24 km. The northern sphere forward model, *A*, is at 116 km along the range, is buried at 24 km and is 2.3 times as dense as the cylinder model. (See appendix 23 – RealDataMisfitJumpFRAC)

4.2 Avoiding local minima

The basin of attraction in Figure 118 shows that the smaller anomaly created from the southern buried body can still create instances of convergence, even near the northern buried body. These instances of convergence can lead to inversions with certain initial parameters tending toward a local minimum instead of the global minimum. Increasing the scaling factor in these cases can steer the solution towards the global minimum. To make convergent conditions more difficult to meet an additional profile is run at 163 km as indicated by the red line in Figure 114. Figure 117 is the relative response from this longitudinal line.

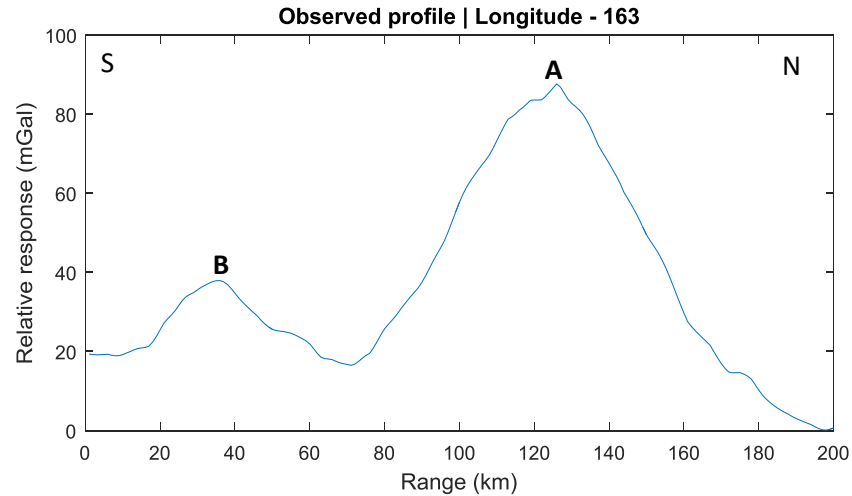


Figure 117: The relative gravity anomaly extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. A and B correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump)

The east-west linear feature is more clearly defined in this profile which would create a larger misfit than before. Once more, the greater northern peak, A, is characterized by a buried spherical body, while the anomaly to the south, B, is categorized by a buried east-west cylindrical body. The extracted profile, both forward model responses and the accompanying basins of attraction, for the longitudinal line at 163 km, are shown in Figure 118.

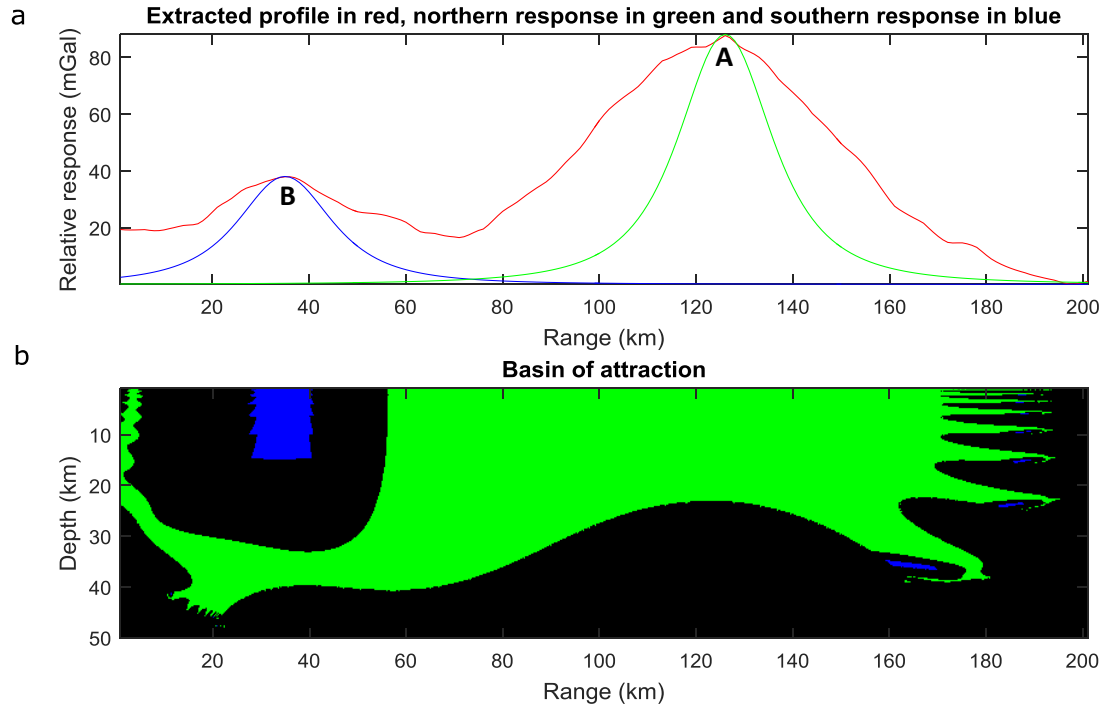


Figure 118: *a* illustrates the extracted profile in red and the forward models for the east-west feature and the central peak in blue and green, respectively. *b* is the associated basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the northern body and convergence to the southern body, respectively. Convergence conditions are met when the final model's location and depth lie within 5 m of the observed model. The southern forward model, *B*, is at 35 km along the range and is buried at 15 km. The sphere forward model, *A*, is at 126 km along the range, is buried at 15 km and is 2.3 times as dense as the cylinder model. (See appendix 23 – RealDataMisfitJumpFRAC)

Using the profiles as shown in Figure 118, a misfit function can be created and the scaling factor can be used to avoid local minima. Figure 119 shows the plotted misfit function and the x position as a function of the number of iterations within the inversion process.

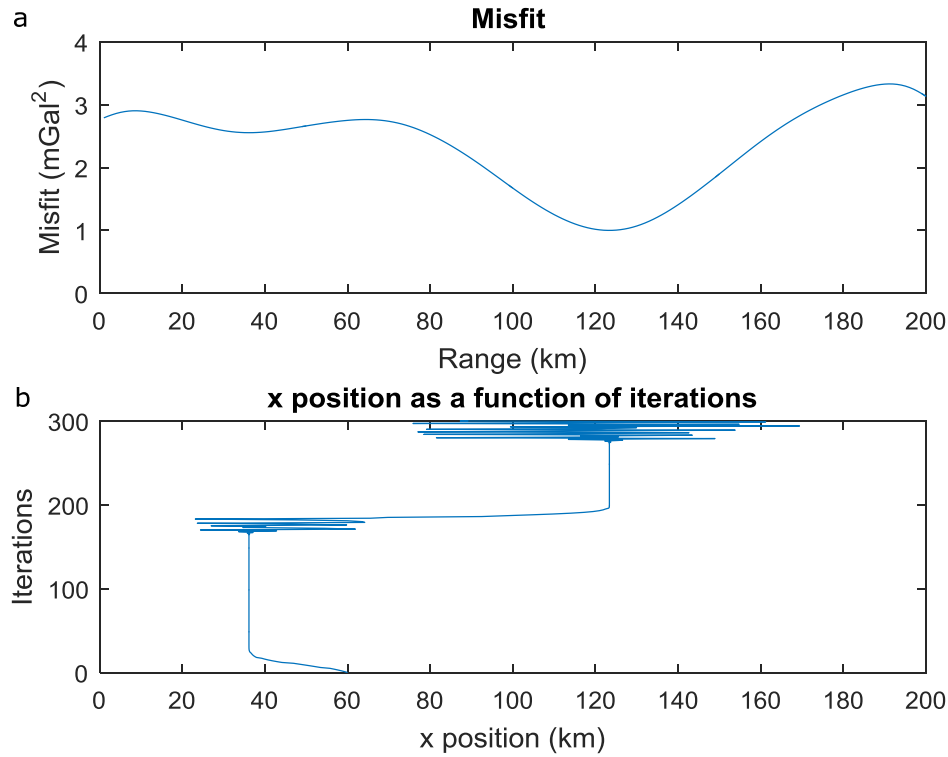


Figure 119: The forward model parameters are set with x and z at 35 km and 15 km, and 125 km and 15 km for the southern and northern forward model, respectively. *a* shows the misfit function as a result of the initial model parameters. *b* is a plot of the x position as a function of the iteration. The initial model is at 60 km along the range and is buried at 15 km. The plot shows that initially, the position tends toward the local minimum and that the increasing scaling factor causes it to oscillate chaotically, which eventually causes the x position to tend towards the global minimum. (See appendix 24 – RealDataMisfitJump)

Figure 119 shows how the initial model parameter tends toward the local minimum during the inversion process. Increasing the scaling factor when the parameter values no longer vary; causes the parameter to fluctuate and eventually tends toward the global minimum. The parameter values associated with the global minimum are accepted as the ideal solution. This is also shown in Figure 120.

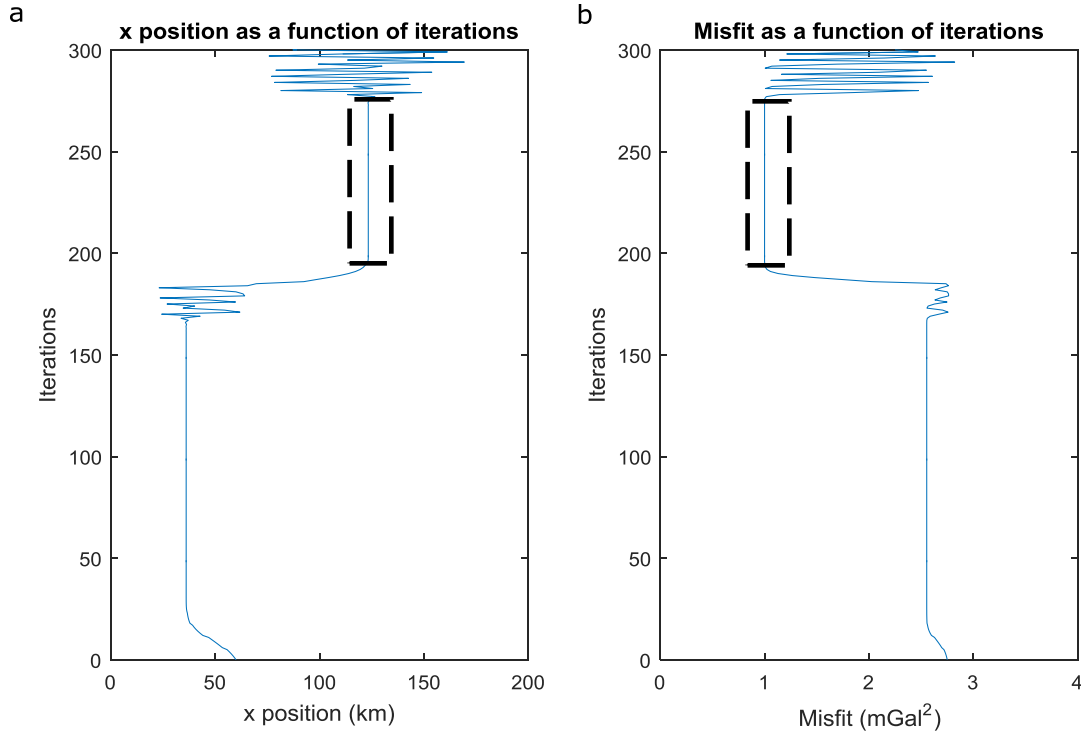


Figure 120: With the same model parameters and values as in Figure 119 a correlation is established within *a* and *b*. *a* shows the *x* position as a function of the amount of iterations. *b* is a plot of the misfit as a function of the number of iterations within the inversion process. The black boxes indicate the parameter values associated with the corresponding global minimum. (See appendix 24 – RealDataMisfitJump)

The parameter values that correspond to the lowest minimum are accepted as the ideal model parameter values. Here the inversion process inverts to an *x* position of 122.4 km, which is within acceptable bounds of the modelled *x* position of 126 km.

CHAPTER FIVE – DISCUSSION

5.1 The Moore-Penrose inversion and the Ridge-regression method

Basins of attraction, as created by the Ridge-regression inversion processes (see equation (8)), are larger due to additional instances of convergence, as when compared to the Moore-Penrose inversion (see equation (7)). The effect of the damping factor k , in the Ridge-regression inversion process, in equation (8), can be visually seen in the distinction between Figure 26 and Figure 27. As the damping factor is increased, the amount of initial models (at various locations and depths) which would have not converged to the observed model is decreased, refer to Chapter 3.7.6. This is a result of the damping factor k acting on the $A^T A$ term, limiting the term's rate of growth.

Chapter 3.3 highlighted how the size and shape of the basins of attraction are dependant upon the observed model's parameter values. When using the Moore-Penrose inversion method, equation (7), successful convergence is largely dependent upon the distance between the observed model and the initial model. This is largely mitigated when using the Ridge-regression inversion process, see equation (8). It is however possible for the Ridge-regression method, when using higher values of the damping constant k , to require additional iterations to reach convergence. To observe this we again define an observed and initial model, Figure 121.

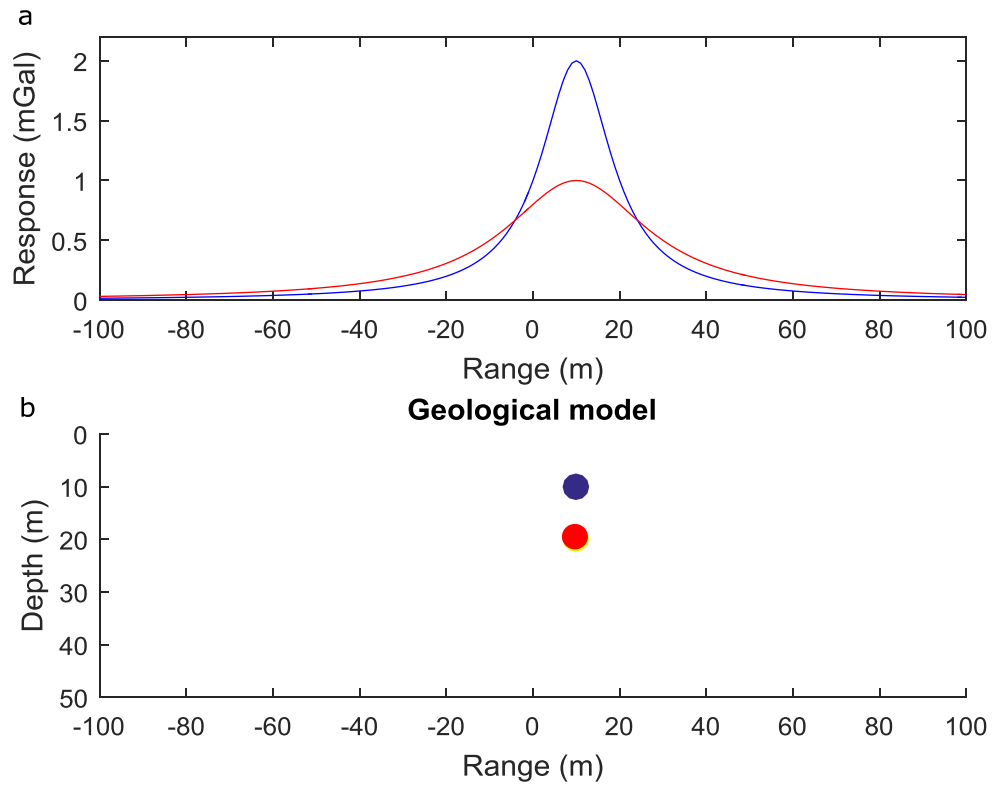


Figure 121: Gravitational responses, in *a*, for buried cylinder models, in *b*, as defined in equation (9). The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at 10 m along the range and is buried at 20 m. (See appendix 12 – ResponsePlusModel)

Using this observed model, the Ridge-regression inversion method, equation (8), acts on the initial model. In Figure 122, the damping constant k is varied and the depth of the model is tracked throughout the inversion process.

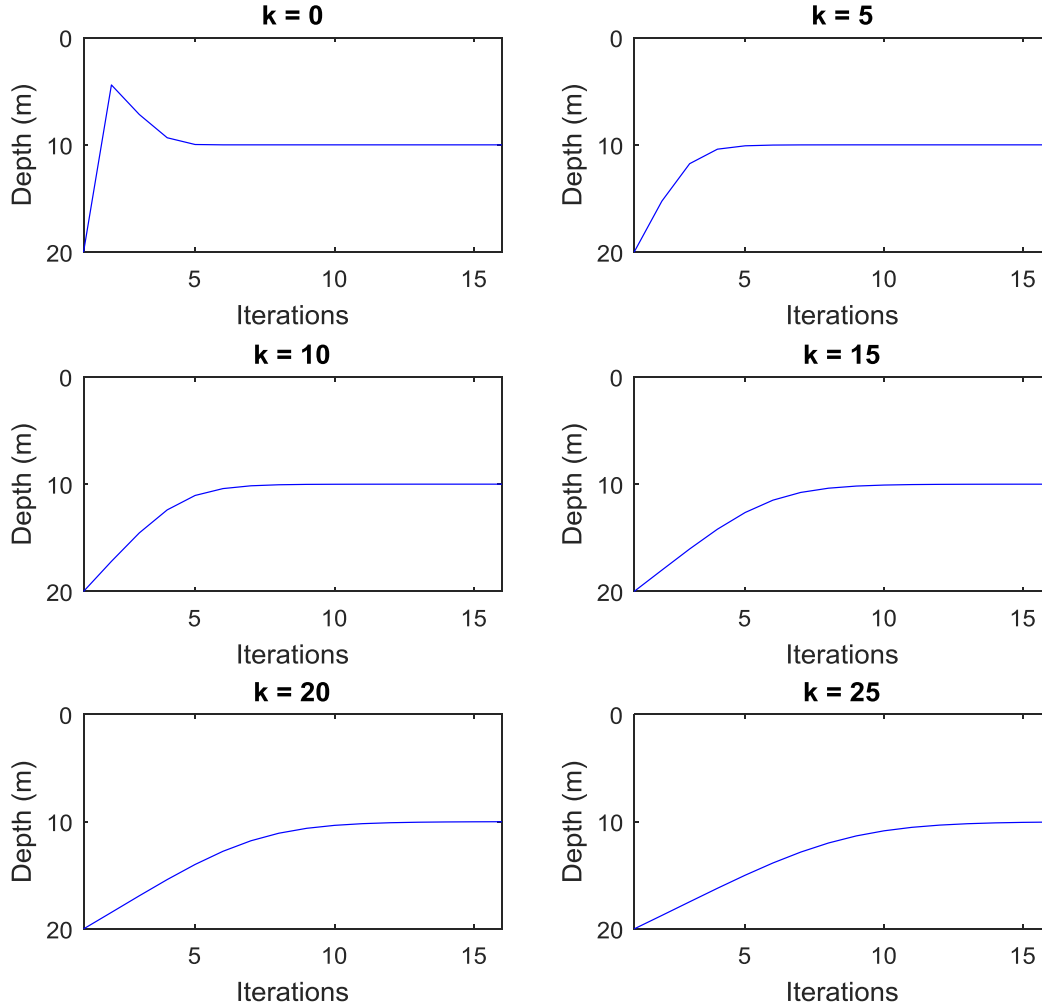


Figure 122: Plots of the depth of the initial model at varied damping constants (k), when acted upon by the Ridge-regression inversion process in equation (8). The observed and initial models are defined in Figure 121. The observed model (blue) is at 10 m along the range and is buried at 10 m. The initial model (red) is at 10 m along the range and is buried at 20 m. (See appendix 26 – CompareK)

It is evident that increasing the damping constant can result in the inversion process requiring additional iterations to reach convergence conditions. It is however additionally noted, that these effects are negligible except at relatively large values of the damping constant k . These problems are largely avoided by the damping constant rarely becoming that large and the increased iterations negligibly affecting the computational requirements.

Throughout the dissertation, the inversion processes act on spherical and cylindrical models. Since the inversion operates using the generated responses of these different geometric bodies, it

stands to reason, that the outcomes could be accomplished using different theoretical geological bodies (dykes, slabs, etc.).

5.2 Analysing the basin of attraction as a result of varying observed model parameters

5.2.1 Varying model depths

Initially, the observed model depth is increased while the positions along the range are kept constant. The basins of attraction grow larger as more initial parameter sets are able to meet convergence conditions; see Figure 32. With regards to the generated responses, this is the result of the observed response and the accompanying misfit minimum growing wider and more easily defined; see Figure 20. The basin of attraction, as a result of increased convergences, grows more complicated and fractal in nature. The results show how the fractal dimension increases as the cylinder's depth is increased within the range.

The same trend is initially observed for a model with two buried bodies as well. However, as the bodies get deeper the responses created, with respect to each buried body, grow wider and it becomes progressively more difficult to converge to solely the left or right bodies, see Figure 64 - 66. Beyond a depth threshold, unique to the original parameters of the model, the observed responses are not able to provide enough information to create larger basins of attraction. Small isolated basins of attraction, as seen in Figure 65 or Figure 66, show the inversion process has become more difficult due to the nature of the observed responses. As the depth of the model is increased the edge effects become more important. The decrease in initial models which converge, as the model's depth is increased, is a result of the edges of respective bodies growing harder to define. As a result, in these systems, convergence is accomplished where the initial model parameter values are close to the observed model parameter values.

5.2.2 Varying model separations

If two buried models are distinguishable from one another, two separate basins of attraction are created, see Figure 62 or Figure 123. However, there are small basins of attraction, near the position of one of the buried bodies, which represent convergence to the opposite buried body. In Figure 123 basins of attraction are highlighted near the right body which indicates convergence

to the left body. Each of these isolated basins of attraction is the result of initial large discrepancy matrices causing overshoots of the trajectories.

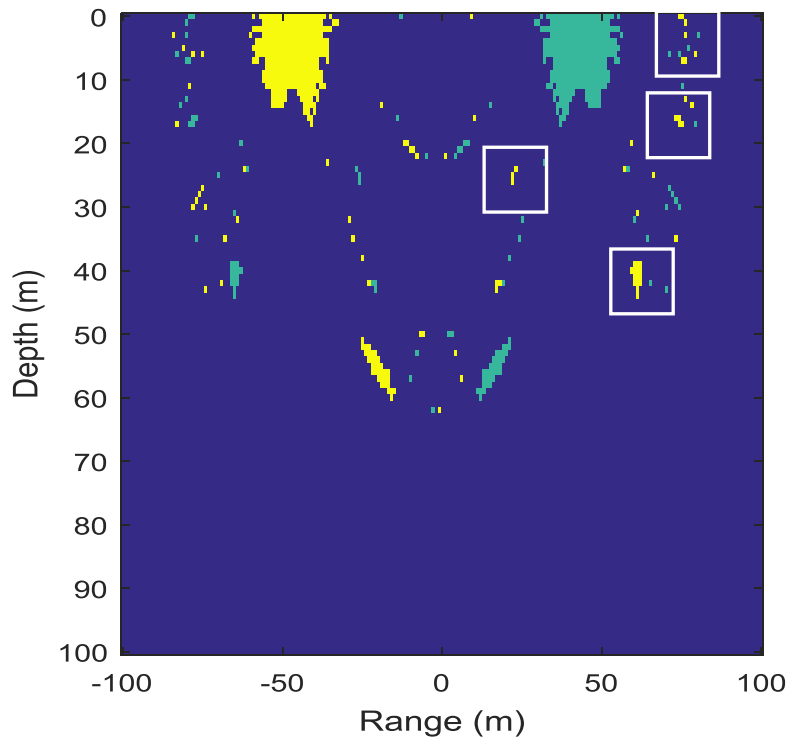


Figure 123: The basin of attraction for model containing two cylindrical sources. The observed model has a source separation and depth of 90 m and 5 m, respectively. Yellow and green represent convergence to the left and right model's respectively. The blue background colour indicates instances where initial parameters did not converge to observed model parameter values. Basins of attraction which converge to the left model, but situated near the right model, are highlighted by white squares. (See appendix 11 – VariedXorZkTwoBodies)

As the two buried bodies in the model separate the fractal dimension again becomes more fractional and deviates from traditional geometric standards. When the observed model's positions drift far apart from one another, the basins of attraction split into independent basins of attraction and resemble those that result from a single buried cylindrical source, see Figure 75 or Figure 76.

5.2.3 Varying the resolution of the solution space

If the scalar change in the area of the solution space is the same as the scalar change in the area of boxes, as used in the box counting method, then it is expected the fractal dimension of the basin of attraction to remain the same. Figure 82 - 84 do show minor changes in the fractal

dimension but they are not large enough to suggest a relationship. Changes in resolution are predominantly used for aesthetic appeal or to observe addition basins of attraction. One major disadvantage to increasing the size of the solution space is the increase in computational time to create the basins of attraction in the solution space. This can be seen in Figure 86 and Figure 87, where the solution space is increased by a factor of 100 but the computation time is increased by a factor of 6470. This is a result of the dimension rules as introduced in section 1.3.2. If the solution space is increased by a factor of 100, then the range and depth parameter indices are each increased by a factor of 100, these results in a solution space 10000 times as large.

5.2.4 Analysis of time taken to construct basins of attraction

An analysis of the time needed to construct some of the basins of attractions for different models is tallied in Table 2.

Page Nr.	Figure Nr.	Solution space size (Width X Height) for 2D basins of attraction (Width X Length X Height) for 3D basins of attraction	Time taken to construct basins of attraction (s)
85	82	200 X 100	9.2 seconds
86	83	1000 X 500	3 minutes 6.3 seconds
87	84	2000 X 1000	8 minutes 2.1 seconds
89	86	200 X 100	8.9 seconds
90	87	20000 X 10000	15 hours 59 minutes 35.52 seconds
107	106	240 X 240 X 80	47 hours 50 minutes 6.1 seconds
113	112	250 X 250 X 50	48 hours 6 minutes 4.7 seconds

Table 2: Comparison of computational times required to construct basins of attraction with different solution spaces.

The time taken to construct the basins of attraction is directly proportional to the size, density, of the solution space. This is to be expected but the rate of increase in time is often surprising. This is due to the fact that doubling the size of the solution space actually results in a squared power of new indices within the solution space. Doubling the size of the solution space means the new solution space is now four times greater than before. Following this; three-dimensional solution spaces result in even greater computational strain. Three-dimensional basins of attraction additionally take longer to create due to the inversion accounting for an additional dimension, parameter, about which to invert for.

This computational strain is greater amplified during the testing and calibrating of the algorithms involved. Many basins of attractions were created for an array of models until the most visually interesting were acquired and then the solution spaces made denser to highlight any unseen details.

5.3 The damping factor's affect on the fractal dimension

As the damping increases, the problem being investigated becomes more non-linear in nature and the fractal dimension of the basin of attraction generated in the solution space is seen to decrease as shown in Figure 124.

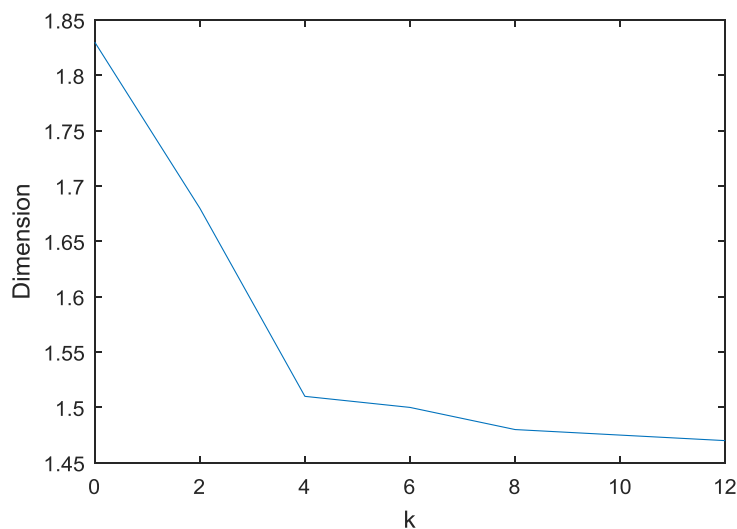


Figure 124: Calculated fractal dimension as a function of the damping factor used. The same model is used for every damping constant. As is in Figure 35 - 39, the observed model is at 0 m along the range and buried at a depth of 10 m. (See appendix 7 – BasinOfAttractionKIterationsTracked)

5.4 The scaling factor's effect on the inversion process

The plots of position as a function of iteration illustrate how the inversion process is affected by the scaling factor; this is seen in Figure 43 - 45. The over-correcting of parameters as a result of manipulations is a function of “feedback” as discussed in section 1.3.4. For values of C lower than 2, the scaling factor acts much like a damping factor. As the scaling factor surpasses 2 and ascends to higher values, periodic oscillations and bifurcations occur. A bifurcation occurs when the number of fixed points in a system doubles. Noting the fixed points as a function of scaling factor can be used to note bifurcations and create the bifurcation diagrams. Identifying the fixed points can be easily done at lower values of the scaling factor, such as with Figure 125, extracted from Figure 43.

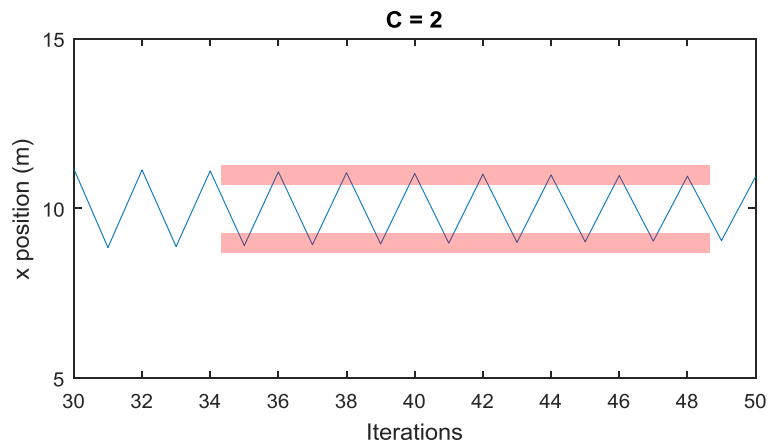


Figure 125: Plot of the x position as a function of iteration; the modified Moore-Penrose inversion, in equation (13), is used with a C value of 2. The fixed points are highlighted in red. (See appendix 18 – LapyunovC)

The scaling factor does eventually cause the inversion to exhibit chaos, see where $C > 4$ in Figure 44. The observable patterns are not clearly recognizable. The fixed points are still present but are now hidden by the apparent randomness. Figure 126 as extracted from Figure 44, is present within an area of apparent randomness. The figure however highlights instances where fixed points still occur.

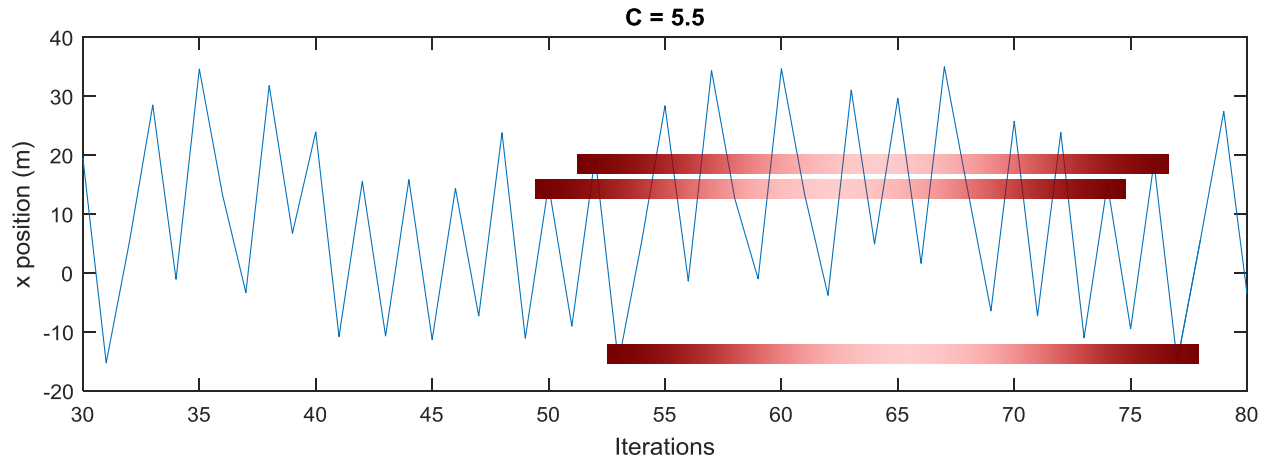


Figure 126: Plot of the x position as a function of iteration; the modified Moore-Penrose inversion, in equation (13), is used with a C value of 5.5. The fixed-points are highlighted in dark red and linked to each other by pink. (See appendix 18 – LapyunovC)

These fixed points illustrate where the higher-order bifurcations are recorded from in the bifurcation map as seen in Figure 58.

5.5 Bifurcation Diagram and the Feigenbaum constants

The bifurcation diagram in Figure 58 does show a general trend; whereas the C parameter increases the period increases and more fixed points become available, this is identified as period doubling. The expressions for the inversion however suggests one might be able to express the scaling factor C as a function of fixed points in the inversion iterations and thus further investigation is possible.

The values calculated in section 3.6 do not match published results exactly; where δ and β are 4.67 and 2.50, respectively (Feigenbaum, 1979). The second calculated Feigenbaum constant β , calculated to be 2.74, has a discrepancy of +0.24 to the published result, which indicates that the result is a good approximation. If one assumes the first Feigenbaum constant δ to be correctly calculated then it would be possible to make predictions. Equation (5), as used to calculate the first Feigenbaum constant, could be redefined as:

$$A_{n+1} = \frac{A_n - A_{n-1}}{\delta_n} + A_n$$

If the calculated value of 2.69 is used for $\square$, then the 4th bifurcation is calculated to be at $C = 4.36$. Using the published value for $\square$, 4.66, the 4th bifurcation is calculated to be at $C = 4.26$. The latter result is closer to the actual scaling factor which results in the 4th bifurcation, which occurs at $C = 4.27$, see Figure 59. To test if my calculated Feigenbaum constant was different than the published results, due to not tending to an infinite scale, equation (5) can be applied to a proceeding period-doubling (the third bifurcation). Following this, the first Feigenbaum constant, in equation (5), is again calculated:

$$\delta_n = \frac{A_n - A_{n-1}}{A_{n+1} - A_n}$$

$$\delta_3 = \frac{4.14 - 3.56}{4.27 - 4.14}$$

$$\delta_3 = 4.46$$

This value is much closer to the published result, now only having a discrepancy of +0.20. It is reasonable to assume that if the process is repeated at higher orders, the $\square$ value would eventually tend to a value close to the published result.

5.6 Determining the scaling factor C

In Figure 50 it appears as though solving for the scaling constant, see equation (24), fails to function properly at $C < 2$. The cause for this is a magnitude of 0 of the parameter change matrix. At lower scaling factor values convergence is reached after only a few iterations, therefore each x location inversion result is the same as its precursor. This is made more evident by Figure 49 where the equation acts after only 3 iterations of the inversion. Determining the scaling factor without information on the model parameters and nature of the response can be problematic. In this section the system was made substantially simpler by assuming an over-determined case and knowing the synthetic model parameters and response type. The complexity in this problem is a result of the “unpredictability” as discussed in section 1.3.4. Further work is necessary if one hopes to determine an optimum scaling factor, for a particular model from the observed responses alone.

5.7 Avoiding local minima

A trained geophysicist may be able to instinctually generate responses similar to observed responses by estimating the initial model parameter values based on prior experience. Even those estimates, under inopportune conditions (noise or unique geology) can result in the inversion process running into local minima instead of a global minimum. Both inversion methodologies investigated (Moore-Penrose and Ridge-regression) show a tendency to tend toward local minima. Automated inversion procedures can thus run inversion processes ad infinitum without tending toward global minima. Someone not well trained in Geoscience might then not be able to correct this problem. Inversion is not a guaranteed solution to all modelling problems.

Having observed how the inversion trajectories can be greatly influenced by altering the scaling factor, it stands to reason that increasing the scaling factor during the inversion can result in the inversion trajectories leaving local minima. Figure 98 and Figure 99 showed a synthetic dataset with two minima. Applying the method as discussed in section 3.8 does result in the inversion trajectory converging to the global minimum. Figure 100 demonstrates convergence to local minima. To avoid local minima, the algorithm notes how the misfit does not change for a few iterations and starts incrementally increasing the scaling factor. As the scaling factor increases the inversion trajectory grows more erratic, and eventually leaves the local minima. Where the inversion finds a new minimum the process repeats itself; the parameter values associated with the smallest misfit are accepted as the global minimum. The tendency for inversion trajectories to tend toward local minima is the effect of “sensitive dependence on initial conditions”, or Butterfly Effect, as discussed in section 1.3.4.

5.8 The Lyapunov Exponent

The scaling factor acts on the inversion process in the same way chaos acts on turbulent systems. The exponential separation of nearby parameters is an effect of “mixing” as discussed in section 1.3.4. In our investigation this is seen when quantifying the rate of separation of model parameters during inversion and calculating the Lyapunov exponent. This is a good way to understand the intricacies of the inversion process.

In section 3.4.3 the Lyapunov exponent is used to quantify the separation of inversion trajectories due to increased scaling factors. Two models are chosen which are initially separated by 0.001 m in the x parameter space, in the presence of one observed model response and one minimum. The two location parameter sets are inverted for with the modified Moore-Penrose inversion, in equation (13), at increasing magnitudes of the scaling factor and the separation recorded. Figure 57 shows that as the scaling factor is increased, the rate of separation increased and the subsequent Lyapunov exponent increased. It might be tempting to surmise that the scaling factor and the Lyapunov exponent were directly proportional but that is insufficiently supported here. Figure 55 showed that in the presence of a large enough minimum on the misfit surface, and with a small enough scaling factor, two inversion trajectories can actually tend closer toward one another. Chaos, by its definition is unpredictable in a sense.

Figure 127 is set up to track the location of the model along a range as inverted for using the modified Moore-Penrose inversion, in equation (13), and a $C = 7$. The process acts on the same model as defined in Figure 42; a single cylinder model buried at 10 m and placed at 10 m along the x -axis. Figure 127 exhibits a Lyapunov exponent of 2.15 as opposed to 2.46 when the modified Moore-Penrose inversion acts on the same system with $C = 5$, see Figure 57.

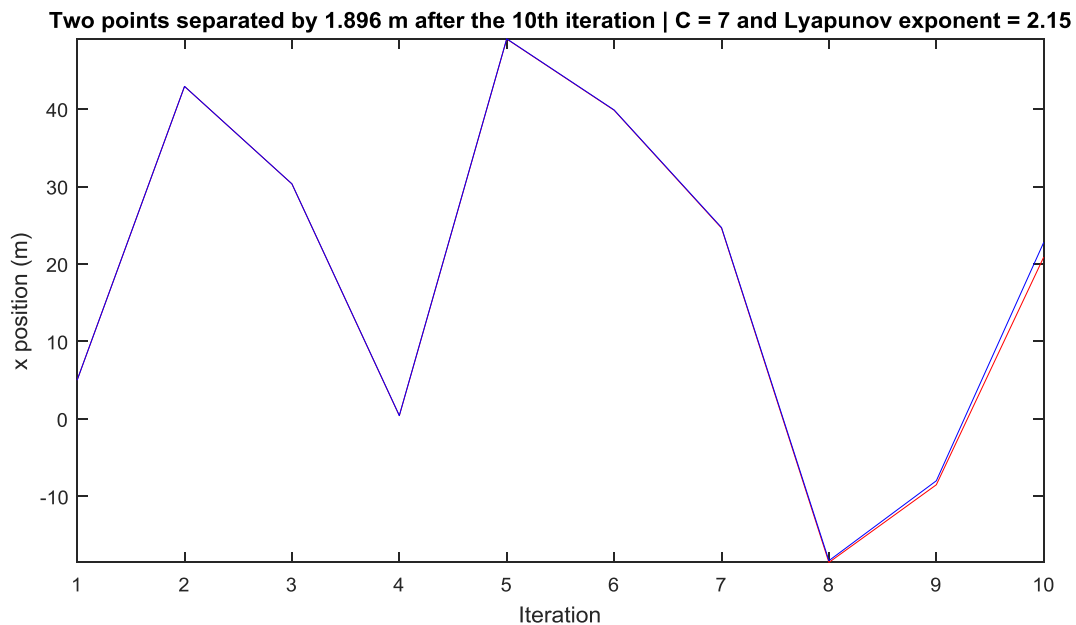


Figure 127: Plot of the x position as a function of the iterations, the modified Moore-Penrose inversion process, as described in equation (13), is used with $C = 7$. (See appendix 18 – LapyunovC)

When in the presence of one attractor, two nearby points should tend to the attractor and exhibit a negative Lyapunov exponent, see Figure 56. In Figure 57, only one attractor exists but chaos is augmented within the system; this results in chaotic oscillations about the attractor and a positive Lyapunov exponent.

5.9 The affect of k and C collaboratively on the inversion process

Figure 35 - 41 show how the damping factor works to aid in reducing the amount of models which meet divergence conditions in the solution space. A comparison between these figures and Figure 96 and Figure 97 illustrates that the damping factor can affect the amount of iterations to converge to observed models less than the scaling factor does. This is better understood by observing Figure 129, where Figure 35 - 41 are summarised and show basins of attraction, created using the Ridge-regression inversion process, in equation (8), and the observed response in Figure 128, with various damping factor constants.

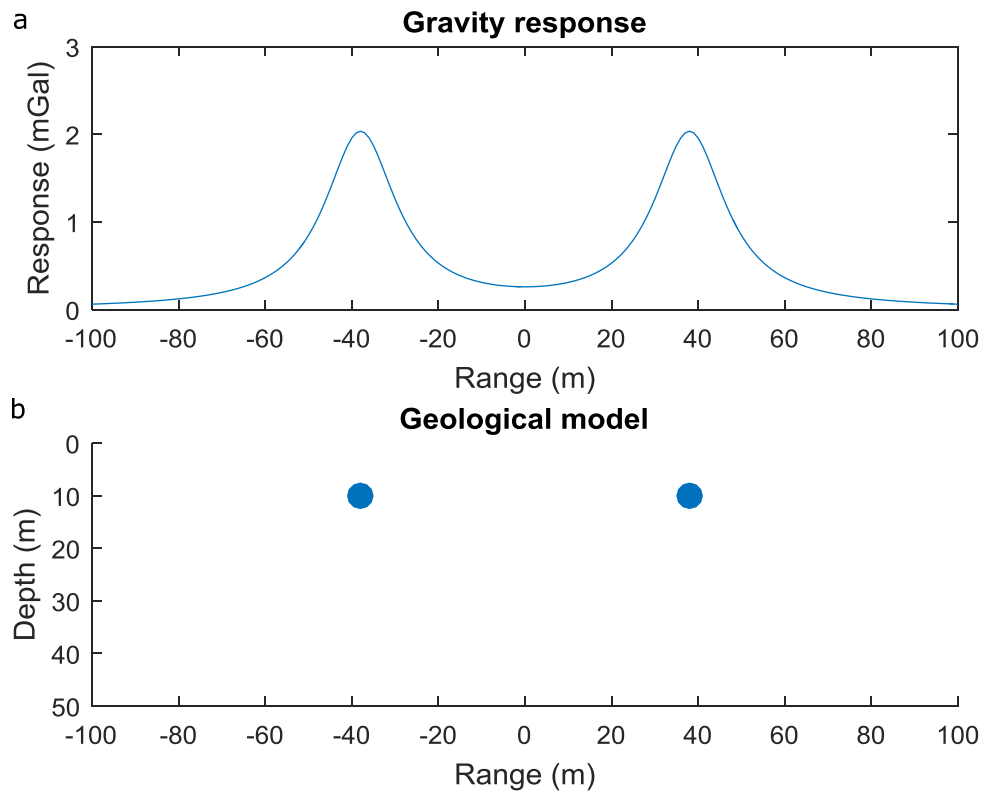


Figure 128: Observed response in *a* as a result of the model in *b*. Model is created with two cylindrical bodies, buried at 10 m and separated by 76 m about the centre of the profile. (See appendix12 – ResponsePlusModel)

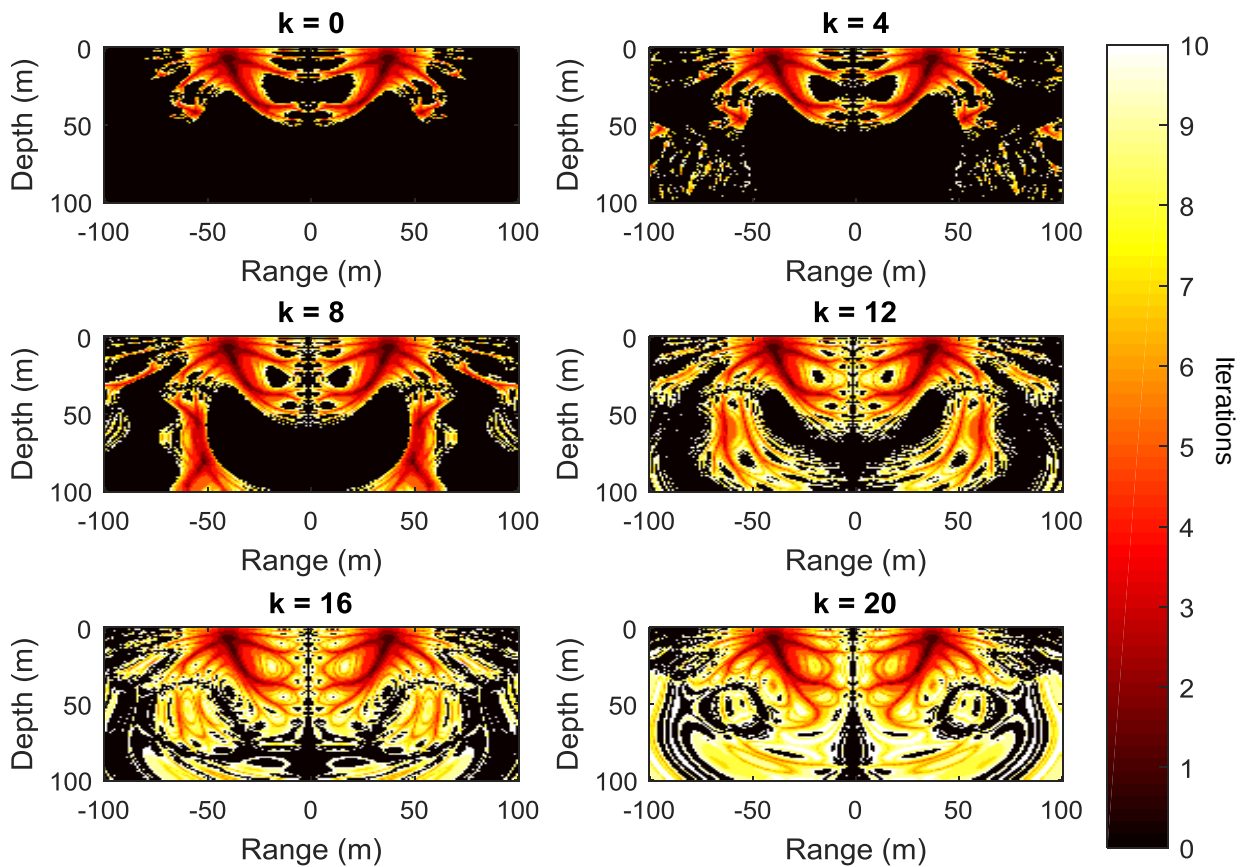


Figure 129: The basin of attraction, as created using the Ridge-regression inversion, in equation (8), and various damping constants k . All observed models are for two buried cylindrical bodies separated by 90 m and at a depth of 10 m. Attached colorbar indicates amount of iterations for parameter sets to converge to either the left or right observed body. The black background indicates parameter sets which did not meet convergence conditions. (See appendix 7 – BasinOfAttractionKIterationsTracked)

Increasing the damping factor allows more initial parameter models to converge within the solution space. The damping factor does increase the amount of iterations to converge; this is a result of the inversion taking smaller steps with respect to iterations. Figure 96 and Figure 97 however show that even small changes in the scaling factor can greatly influence the amount of iterations to converge to an observed model. In section 3.4.2 new basins of attraction are created in a solution space using k and C as parameters. These basins of attraction demonstrate that where the calculated and observed models were nearby one another, the scaling factor almost solely determines the amount of iterations to converge. As the observed and calculated models

grew further apart, the effect of the damping factor becomes more evident, as it now influences parameter convergence; this is illustrated in Figure 51 - 54.

5.10 Applications to gravity data from South Africa

The relative gravity profile from Figure 117 is highlighted again in Figure 130.

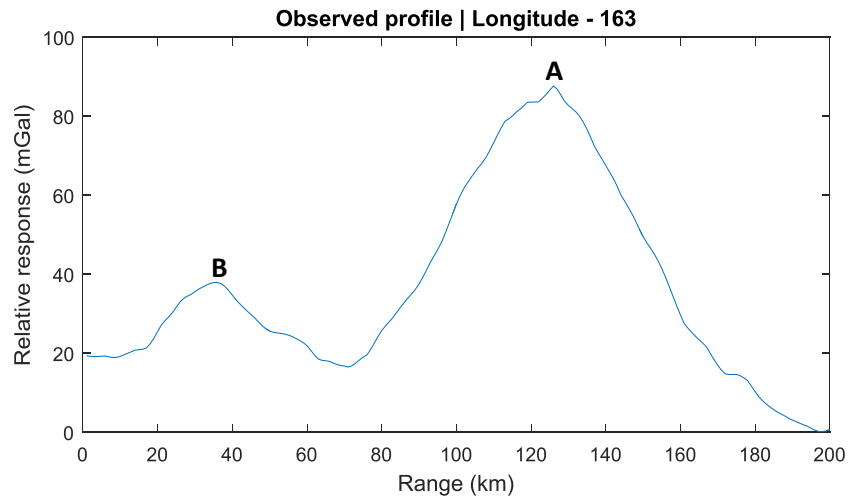


Figure 130: The relative gravity profile extracted along the longitudinal line indicated in Figure 114 at longitude marker 163. The profile is run south to north and is cropped about the two peaks. A and B correspond with the locations of the responses observable in Figure 114. (See appendix 24 – RealDataMisfitJump)

Using these gravity data, and user defined forward models, basins of attractions can be created through inversions in a colour-coding scheme similar to that of Figure 118 and again in Figure 131. Even though the instances of converging to the local minima, as shown by blue elements within the solution space, are small they are still possible. This means it is possible to converge to the cylinder model through initial body parameter values near the location of the buried sphere forward model, B and A, respectively. Using profiles to create basins of attraction highlights this. Figure 131 highlights some basins of attraction for the local minima, as a result of the buried cylindrical model B, which are near the location of the buried spherical forward model A.

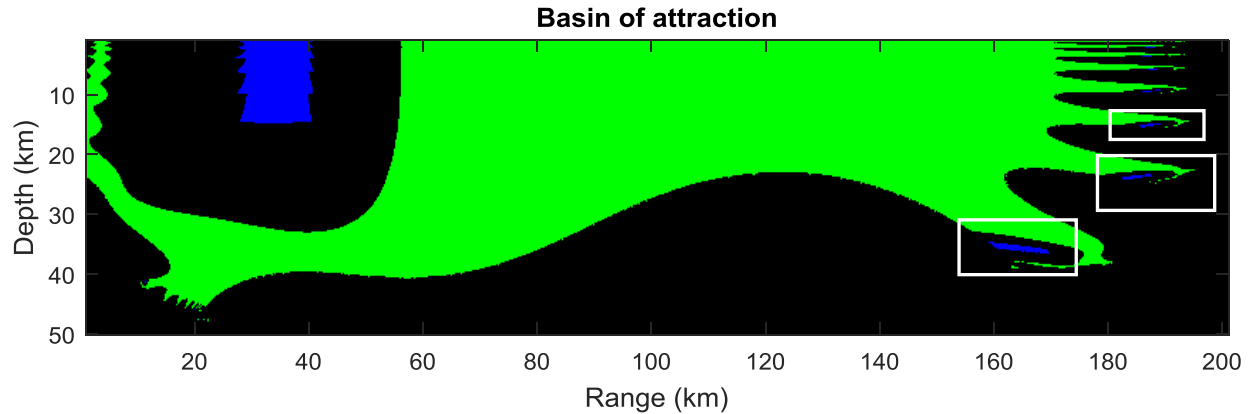


Figure 131: Basin of attraction as a result of the two established minima. Black, green and blue represent divergence from either model parameter values, convergence to the buried spherical forward model and convergence to the buried cylindrical forward model, respectively. Localised basins of convergence to the cylinder model highlighted by the white boxes. (See appendix 23 – RealDataMisfitJumpFRAC)

Through synthetic data systems it has been shown that it is possible to avoid local minima by using increased scaling factor values and this is effective even in systems with noise. In Figure 119, the inversion acts on a model close to the forward model and the inversion still tends toward the local minima. As the inversion trajectory rests within the local minima, the modified Moore-Penrose inversion process (see equation (13)) iteratively increases the constant C , to attempt to exit the local minima. Upon successful departure from the local minima, the parameter trajectory tends toward the global minimum and those associated parameter values are then accepted as the final results, this is highlighted in Figure 120. The results closely match the parameter values used in the forward model.

CHAPTER SIX – CONCLUSIONS AND RECCOMENDATIONS

The initial parameters of non-linear dynamic systems greatly influence their outcome, see section 1.3.4. The inversion process, as it applies to geophysical models, follows the same trends and is greatly influenced by the initial parameters present in the inversion process. With the importance inversion plays in analysing geophysical responses, analysing these influences is integral in understanding inversion problems.

The initial models which converge to local minima under the effect of Moore-Penrose least-squares inversion, equation (7), and Ridge-regression inversion, equation (8), do so as a fractal (as do the modified forms in equations (13) and (18), respectively). Tracking convergence conditions of initial models in solution spaces create basins of attraction, and the associated fractal dimension is dependent upon the initial model's parameters. The fractal nature of the basins of attraction is due to models undergoing inversions for two or more parameters as this involves the parameter having a non-linear response under inversion. The importance of the initial estimation of the initial model's parameter values can make it very difficult to create accurate models using inversion. This is normally corrected for with knowledge of the existing geology or supporting geophysical survey methods. In this dissertation however, it is observed that damping the inversion process works to reduce the risk of selecting initial model parameters which would not successfully converge to observed models. Subsequently, this creates more occurrences of convergence to the theoretically correct geological model and the basin of attraction increases in size. Additionally; increasing the damping factor k decreases the observed fractal dimension.

The effect which the damping factor has on the inversion process is easy to understand if k is thought of as a model parameter, affecting the inversion process in a similar way that changes in the models' locations or depths might. In the modified Moore-Penrose inversion, in equation (13), an additional parameter, the scaling factor C , is added to equation (7), the Moore-Penrose least-squares inversion equation. This new equation allows for chaos to be introduced into the inversion process, for model parameters which show a non-linear response under inversion. The chaos introduces bifurcations and can alter the expected inversion trajectory. It has been

demonstrated that these manipulations can be utilized to move inversion processes out of local minima in attempts to steer the inversion toward a global minimum. This was shown on synthetic and real models of gravity data.

The scaling factor and the damping factor each effect the inversion process by altering the initial model parameters. Increasing the amount of parameters, up to three in this dissertation, which the inversion process acts on can make the process increasingly non-linear. This can result in smaller basins of attraction due to the amount of iterations to converge increasing, as the process is made more difficult. The associated computational times needed to create the three dimensional basins of attraction are also greatly increased and can take days to create. These alterations can create convergence from divergent parameter sets, change the positions of strange attractors and alter trajectories of the inversion process. It should also possible to define initial parameter values which would result in the fewest amounts of iterations to converge.

With regards to recommendations into future work; in this dissertation, a three dimensional basin of attraction is created to explore the effect on the computational time, size of the basins of attraction and the shape. The effects that chaos and fractals have on geophysical inversion processes, applied in this dissertation, can also be applied to the three dimensional systems to investigate if the observed trends are maintained. The scaling factor which results in the fewest amounts of iterations to converge is unique to the initial parameters of the model. Defining a method to calculate the scaling factor, purely from the observed and calculated responses, is a possible point of further research.

REFERENCES

- Abarbanel, H. D. I. (1995) Analysis of Observed Chaotic Data, Institute for Nonlinear Science, Springer Science & Business Media, pp. 12 - 95.
- Akram, S and Qurrat ul, U, A. (2015) Newton Raphson Method, International Journal of Scientific & Engineering Research, Volume 6, Issue 7. pp. 1 - 4.
- Coleman, P and Pietronero, L. (1992) The fractal structure of the universe, Physics reports, North-Holland, p. 30.
- Conway, J. H. and Guy, R. K. (1996) The Book of Numbers, Springer Science & Business Media, p. 23.
- Cooper, G.R.J . (2000) Fractal convergence properties of geophysical inversion, Computers & Graphics 24, pp. 603 - 609.
- Cooper, G.R.J . (2001) Aspects of chaotic dynamics in the least-squares inversion of gravity data, Computers & Graphics 25, pp. 691 - 697.
- Devaney, R L and Keen, L. (1988) Chaos and Fractals: The Mathematics Behind the Computer Graphics, Proceedings Of Symposia In Applied Mathematics, Vol 39, pp. 1 - 14.
- Dingwell, J. B. (2006) Lyapunov exponents, Wiley Encyclopedia of Biomedical Engineering, John Wiley & Sons: Hoboken, NJ, USA, p. 2.
- Feigenbaum, M. J. (1979) The universal metric properties of nonlinear transformations, Journal of Statistical Physics, Vol. 21, No. 6, pp. 703 - 706.
- Feng, B and Chaun-Feng, L. (2013) Fractals in Quantum Information Process, Chinese Physics Letters, Vol 30, No 1, pp. 1 - 4.
- Frame, M. (2014) Biographical Memoirs: Benoit B. Mandelbrot. National Academy of Sciences, pp. 5 - 10.

- Ge, M and Lin, Q. (2009) Realizing the box-counting method for calculating fractal dimension of urban form based on remote sensing image. *Geo-Spatial Information Science*, pp. 1 - 24.
- Gleick J. (1987) *Chaos: Making a new science*, A Penguin Book: Science, pp. 1 - 25.
- Hillborn, R, C. (1994) *Chaos and non-linear dynamics, An introduction for scientists and engineers*, New York Oxford University Press, pp. 16 - 56, 390 - 406.
- Hutchinson, J. E. (1981). *Fractals and Self Similarity*, *Indiana University Mathematics Journal*, vol 30, pp. 713 - 747.
- Kaye, B. H. (1994). *A Random Walk Through Fractal Dimensions*, John Wiley & Sons, p. 3.
- Mare, L.P. and Cole, J. (2006). The Trompsburg Complex, South Africa: A preliminary three dimensional model. *Journal of African Earth Sciences*, vol 44, pp. 314 - 330.
- Menke, W. (1989). *Geophysical Data Analysis: Discrete Inverse Theory (revised edition)*. Academic Press, New York, vol 289, pp. 38 - 43
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. And Flannery, B. P, (2007). *Numerical recipes, The Art of Scientific Computing, Third Edition*, Cambridge University Press, pp. 461 - 462.
- Rezaie, M., Moradzadeh, A., Nejati, A., Aghajani, H., Roshandel Kahoo, A. and Moazam, S. (2017). 3D modelling of Trompsburg Complex (in South Africa) using 3D focusing inversion of gravity data. *Journal of African Earth Sciences*, vol 130, pp. 1 - 7.
- Snieder R. and Trampert J. (1999). *Inverse Problems in Geophysics. International Centre for Mechanical Sciences (Courses and Lectures)*, vol 398, pp. 1 - 4.
- Telford, W., Geldart, L., and Sheriff, R. (1990). *Applied Geophysics*, Cambridge: Cambridge University Press, 2nd Edition, pp. 34 - 46.
- Sprott, J. C. and Xiong, A. (2015). *Classifying and quantifying basins of attraction*. Department of Physics, University of Wisconsin, Madison, *Chaos* 25, pp. 1 – 6.

Image references

Bifurcating river, n d. Photograph, viewed 13 October 2019, <
<https://www.yourdictionary.com/bifurcate> >

Google Earth 7.3.2. 2017. *Trompsburg, South Africa* 30.032 S and 25.779 E, elevation 1424 m, viewed 2 January 2019. <
https://earth.app.goo.gl/?apn=com.google.earth&ibi=com.google.b612&isi=293622097&ius=googleearth&link=https%3a%2f%2fearth.google.com%2fweb%2f%4030.04601095,25.79638784,1420.02568561a,20127.5939651d,35y,0h,0t,0r%2fdata%3dCk8aTRJFCiUweDFIOGY2NzZhNDAyMjdkMDM6MHhkMDAzNDUzMzMmYmJmZjY2GXQtEPObCD7AIRu0_GWcxzlAKgpUcm9tcHNidXJnGAIgASgC>

Koch snowflake, 2016. RS1 Level G, Lesson 112, Kathleen Lawler. Image viewed 24 December 2018, < <https://rightstartmath.com/level-g-lesson-112/> >

Lighting strike, n d. Photograph, viewed 6 January 2019, <<https://www.picsnaper.com/p/One-Big-Lightning-Bolt-Stock-Photos-One-Big-Lightning-Bolt-Stock-Images-Alamy>>

Mandelbrot, n d. Image, viewed 12 January 2019, <<http://www.pomegranateapps.com/fractals/>>

Progressing dimensions, n d. Image, viewed 6 January 2019,
<<https://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html>>.

Sierpinski triangle, n d. Image, viewed 6 January 2019,
<<http://interactivepython.org/courselib/static/pythonds/Recursion/pythondsSierpinskiTriangle.html>>

Sierpinski triangle constructions, n d. Image, Viewed 6 January 2019,
<http://math.bu.edu/DYSYS/chaos-game/node2.html>.

Sprott, J. C (1997). *Mandelbrot Set Chaos*, Department of Physics, University of Wisconsin, Madison, Technical notes.

Snowflake, n d. Photograph, viewed 6 January 2019,

<<https://medium.com/@GluedToTheScreen/afterpod-skepticism-in-podcasting-3deba666431c>>

APPENDIX

1 - LogisticMapGenerator

`function` LogisticMapGenerator

`%Brandon Dias`

`%Revised on 25/01/2018`

`%676053`

`%The aim of this function is to create a logistic map by recording how
%bifurcations occur at varied parameter changes. This bifurcation map will
%be created from the quadratic function $x_{n+1}=C*x_n(1-x_n)$. Bifurcations are
%the collective term for qualitative changes in a system's dynamics, the
%points at which the bifurcations land are then subsequently called the
%bifurcation points.`

`%The behavior of a system can change drastically with even small changes in
%system parameters, in the case of this code it is the variable "C". This
%change is not purely quantitative but is also qualitative. What this means
%is not only a resulting change of fixed position but that new fixed points can
%also be created or destroyed and succumb to changes in stability. By
%changing the parameter "C" the hope is to observe how the system's
%stability can change from regular (stable points or stable periodicity)
%to chaos. The qualitative changes are a great area of investigation and
%the aim of the theory of nonlinear dynamics.`

`Cs=[]; %Establish a array in which to store the C values
Cstart=0; %C starts at
Cend =4.5; %C ends at
Crange=100; %Points in-between start and ending C
CInterval = (Cend-Cstart)/Crange; %How much between the C values
DiscardAmount=200; %The iterations need time to reach a stable amount and thus some iterations
are thrown away
HoldAmount=100; %The points used to show the bifurcation points
Data = zeros(HoldAmount,1); %Holds the bifurcation points
DataD=[];`

`for C = Cstart:CInterval:Cend
Data=0.1; %Starting location
for i = 1:DiscardAmount
Data = C * Data * (1-Data); %Iteration
end
for j = 1:HoldAmount
Data = C * Data * (1-Data);
Cs=[Cs C];
DataD=[DataD Data];
end
end
scatter(Cs,DataD); xlabel('C');ylabel('Final Position'); title('Logistic Map');
end`

2 – InversionStepsPlusModel

```
function InversionStepsPlusModel
```

```
%Brandon Dias
```

```
%Revised on 25/01/2019
```

```
%676053
```

```
%This algorithm tracks the calculated model through the inversion process
%and at each iteration; the model and associated responses are generated.
%The misfit is recorded at each iteration and plotted as a function of the
%iteration.
```

```
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
```

```
X = -100:100; % Range
```

```
Shift = 0; % position of the observed centre of the cylinder
```

```
Depth = 15; % The depth at which the observed cylinder is buried
```

```
Mass = 20;
```

```
gm = G*Mass;
```

```
%for Shift = 35:10:45 %A for loop can be run here should you want to generate many images, this loop can be run
for depth, separation or even scale
```

```
gmz = G*Mass*Depth;
```

```
Gobs = gmz./((X-Shift).^2+Depth^2);%Synthetic gravitational response for a buried cylinder model
```

```
xp = 15; % position of the centre of the initial model cylinder
```

```
zp = 32.5; % depth of the centre of the initial model cylinder
```

```
gmz = G*Mass*zp;
```

```
M=[];
```

```
Isteps=1:6;
```

```
for i = 1:6
```

```
    %Parameter values for cylinders in scatter
```

```
    Xmodel=[Shift xp];
```

```
    Zmodel=[Depth zp];
```

```
    StepNum=i-1; %Index for subplot
```

```
    Gcalc=gm*zp./((X-xp).^2+zp^2);
```

```
    Error=Gcalc-Gobs; %Discrepancy matrix
```

```
    %Here the misfit is calculated as in equation (9)
```

```
    N=length(Gcalc);
```

```
    Mis=sum(Error.^2);
```

```
    MisfitStep=Mis/N;
```

```
    M=[M MisfitStep];
```

```
    % Responses are calculated and plotted throughout the iteration steps
```

```
    figure(1)
```

```
    subplot(3,2,i);plot(X,Gobs,'r',X,Gcalc,'b');xlabel('Range (m)');ylabel('Response (mGal)');axis([-100 100 0
2]);title(['Iteration: ',num2str(StepNum)]);
```

```
    %New calculated profile
```

```
    C=linspace(1,2,length(Xmodel)); %Linspace creates a linear array of colours as spread out over the length of the
amount of indices in the array
```

```

% models are plotted throughout the iteration steps
figure(2)
subplot(3,2,i);scatter(Xmodel,Zmodel,[100],C,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-50 50
0 35]);title(['Iteration: ',num2str(StepNum)]);%title(['Inversion validation | Iteration: ',num2str(StepNum)]);

Error=Error';

gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az
A = [gx;gz];    %A
A = A';
At=A';
AtA=At*A;
iAtA = inv(AtA);
dp = real((iAtA*At)*Error);    %The parameter change matrix is created with regards to the x and z parameter
held in dp(1,1) and dp(2,1), respectively
xp = xp+dp(1,1);
zp = zp-dp(2,1);

if zp<0
    zp=0;
end
end
%The misfit is plotted as function of the iterations through the inversion
%process. By the last iteration the misfit should be 0 as the calculated
%and observed models should have the same parameter values.
figure(3)
plot(Isteps,M);xlabel('Iterations');ylabel('Misfit');title('Misfit as a function of iteration');
end

```

3 - MisfitJumNew

```
function MisfitJumpNew
```

```

%Brandon Dias
%Revised on 21/12/2018
%676053

```

```

%The algorithm runs the inversion process as defined in equation (13). The
%misfit function over which the inversion runs is modelled to have two
%minima; the model has two buried cyliders. Initial calculated parameter
%values are chosen near the shallower misfit trough and allowed to tend to
%it. Once the parameter values do not vary more than a specified limit,
%the scaling factors are increased incrementally.

```

```

TempCount=0;
%Arrays for plotting
Misfits=[];
Scalings=[];

```

```

C=1;
%Initial parameter values for the observed and calculated models
G = 6.67E-11;

```

```

X = -100:100;
Mass = 90000;
gm=G*Mass;
%Observed model's parameter values
x1=-25;
z1=10;
x2=50;
z2=15;
Mass2 = 90000;
gm2=G*Mass2;
%Shape of observed model is two horizontal cylinder
g1= (gm*z1./((X-x1).^2+z1^2));
g2= (gm2*z2./((X-x2).^2+z2^2));
Gobs = g1 + g2;

%An initial calculated model is set up at a minima to
%observe a minimum misfit value
GcalcM1=gm*z1./((X-x1).^2+z1^2);
Error = GcalcM1-Gobs;
%Misfit equation (9)
N=length(GcalcM1);
Mis=sum(Error.^2);
MisfitM1=Mis/N;
%An initial calculated model is set up at a minima to
%observe a minimum misfit value
GcalcM2=gm*z2./((X-x2).^2+z2^2);
Error = GcalcM2-Gobs;
%Misfit equation (9)
N=length(GcalcM2);
Mis=sum(Error.^2);
MisfitM2=Mis/N;
%This display shows the value of the two minima
MisfitString=['Misfit 1: ',num2str(MisfitM1),' | Misfit 2: ',num2str(MisfitM2)];
display(MisfitString);

figure(2);
plot(X,Gobs);title('Profiles');

zp=10;
Ma=[];
for xp = -100:100
    Gcalc = gm*zp./((X-xp).^2+zp^2);
    Error = Gcalc-Gobs;

    N=length(Gcalc);
    Mis=sum(Error.^2);
    MisfitT=Mis/N;
    Ma=[Ma MisfitT];
end
X=-100:100;
figure(3)
plot(X,Ma);title('Misfit surface');xlabel('Position along profile (m)');ylabel('Misfit');%axis([-50 150 3 15]);
%pause;

%In figure(3) the misfit function is plotted, with that information select

```

```

%an initial parameter value nearby the local minimum
%Parameters can be input such as here
xp=input('Guessing x: ');
%zp=input('Guessing z: ');

    %Guess profile
Xps=[];
%Zps=[];
Xps=[Xps xp];
%Zps=[Zps zp];
%Parameters can be hard coded such as here
zp=10;
Gcalc=gm*zp./((X-xp).^2+zp^2);
Error = Gcalc-Gobs;
N=length(Gcalc);
Mis=sum(Error.^2);
Mis=Mis/N;
Misfits=[Misfits Mis];
Scalings=[Scalings C];
%The inversion is run long enough to enter two minima. The amount of
%iterations will be much greater due to reiterative iterations in the
%minima.
for i = 1:50

    %equation (13) set up
    Error=Error'*C;

    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az
    A = [gx;gz];    %A
    A = A';
    At=A';
    AtA=At*A;
    iAtA = inv(AtA);
    dp = real((iAtA*At)*Error);
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
    if zp<0
        zp=1;
    end
    %Record the position
    Xps=[Xps xp];

    Gcalc=gm*zp./((X-xp).^2+zp^2);
    Error = Gcalc-Gobs;
    %Misfit
    N=length(Gcalc);
    Mis2=sum(Error.^2);
    AfterMis=Mis2/N;
    Misfits=[Misfits AfterMis];
    Scalings=[Scalings C];
    %Test if the position has left the range
    if isnan(AfterMis) == 1
        display('Divergence');
        break;
    end
end

```

```

end

%This check observes if whether or not the position of the model
%deviates with the minima
Diff=abs(Xps(end)-Xps(end-1));
%Check
if Diff < 1;
    %pause;
    TempCount = TempCount +1;
end
%If the point doesnt move over two iterations it is deemed locked in a minima. It can defined to more than 2 but the
C constant scales slowly so its okay
if TempCount == 5
    display('Starting C up');
    %The "potential" variables record every instance into a minima and
    %the smallest is the global
    Potentialxp=xp;
    PotentialMisfit=AfterMis;
    Potentialzp=zp;
    TempCount=0;
    C=1;
    for CCount = 1:120 %How high can C get

        C=C+0.025;%Due to the small displacement between the two minima, the scaling climbs slower
        Gcalc=gm*zp./((X-xp).^2+zp^2);
        Error = Gcalc-Gobs;
        Error=Error*C;

%
        gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
        gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az
        A = [gx;gz];    %A
        A = A';
        At=A';
        AtA=At*A;
        iAtA = inv(AtAA);
        dp = real((iAtA*At)*Error);
        xp = xp+dp(1,1);

        Xps=[Xps xp];

        Gcalc=gm*zp./((X-xp).^2+zp^2);
        Error = Gcalc-Gobs;

%
        N=length(Gcalc);
        Mis=sum(Error.^2);
        TMis=Mis/N;
        Misfits=[Misfits TMis];
        Scalings=[Scalings C];

        if TMis < (PotentialMisfit-1E-14) %User defined threshold. Can observe from misfit plots
            C=1;
            break;
        end
    if isnan(TMis)==1
        break;
    end
end

```

```

        end

    end
    C=1;

%
end

end

L1=length(Misfits);
L1=0:L1-1;

Misfits=Misfits*1E14; %Can eliminate the exponents for ease of readability
%Plots of the locations,csaling factors, profiles or depths can be copied
%as required
%Axis can be used after the first run to enable zooming
figure(1);
subplot(1,3,2);plot(Xps,L1);title('x position as a function of iterations');axis([-80 80 0 180]);xlabel('x position
(m)');ylabel('Iterations');
subplot(1,3,1);plot(Misfits,L1);title('Misfit as a function of iterations');axis([0 8 0
180]);xlabel('Misfit');ylabel('Iterations');
subplot(1,3,3);plot(L1,Scalings);title('C as a function of iterations');axis([0 180 0 5]); xlabel('Iterations');ylabel('
Scaling factor C');
figure(4);
plot(L1,Scalings);title('Scaling factors');
X=-100:100;
Ma=Ma*1E14;
Gobs=Gobs*1E6;
figure(5)
subplot(3,1,1);plot(X,Gobs);title('Profiles');xlabel('x position (m)');ylabel('mGal');axis([-80 80 0 1]);
subplot(3,1,2);plot(X,Ma);title('Misfit surface');xlabel('x position (m)');ylabel('Misfit');axis([-80 80 0 8]);%axis([-50
150 3 15]);
subplot(3,1,3);plot(Xps,L1);title('Position as a function of iterations');axis([-80 80 0 180]);xlabel('x position
(m)');ylabel('Iterations');

figure(6)
subplot(2,1,1);plot(X,Ma);title('Misfit surface');xlabel('x position (m)');ylabel('Misfit (mGal^2)');axis([-80 80 0
8]);%axis([-50 150 3 15]);
subplot(2,1,2);plot(Xps,L1);title('Position as a function of iterations');axis([-80 80 0 180]);xlabel('x position
(m)');ylabel('Iterations');

%In the inversions above the "potential" variables were reassigned as the
%inversion run. Here the answers are displayed
Potentials=['xp: ',num2str(Potentialxp),' zp: ',num2str(Potentialzp),' Misfit ',num2str(PotentialMisfit)];
display(Potentials);
end

```

4 – MakeBMap

```

function MakeBMap
%Brandon Dias

```

```

%676053
%23/03/2018
%This algorithm maintains the same observed and initial calculated model
%over an array of C values to be used in the modified Moore-Penrose
%inversion. At each C value the inversion is run and bifurcations recorded.
%A bifurcation diagram is plotted.

%Initial parameters set up
G = 1;% 6.67E-11;
X = -100:100;           %Range
k=0;%Moore-Penrose
%Observed model parameter values
Depth = 10;
Shift = 0;
Mass = 2000;
gmz = G*Mass*Depth;
gm = G*Mass;
Gobs = gmz./(((X-Shift).^2+Depth^2).^1.5);
%Display to have user confirm
display('Initial model parameters');
display(['Actual depth : ', num2str(Depth)]);
display(['Actual shift : ', num2str(Shift)]);
%Initial calculate model parameter values
xp = 5;
OrigX = xp;
zp = 10;
%Display to have user confirm
display(['Starting inversion depth : ', num2str(zp)]);
display(['Starting inversion displacement : ', num2str(xp)]);
%How many iterations to perform of the inversion process. At what point
%should bifurcations be met is determined by the initial set up partly
j=50;

FinalPositions = [];
Iterations = [0];
CArray = [];
for C = 0.1:0.01:4.5 %Array of C values. Hard-coded as desired
    display((C/4.5)*100);%Monitor progression
    Pos=[];
    xp=OrigX;
    zp=10;
    for i = 1:j

        Gcalc=gmz./(((X-xp).^2+zp^2).^1.5);
        Error = Gcalc-Gobs; %difference "D"
        Error=Error'*C; %Modified Moore-Penrose by adding the C factor
        gx = ((X-xp)*(-3*gm*zp))./(((X-xp).^2+zp^2).^2.5); %ax
        gz = (-1*gm*(2*zp^2-((X-xp).^2))./(zp^2+((X-xp).^2.5));
        A = [gx;gz]; %A
        A = A';
        At=A';
        AtA=At*A;
        iAtAnew = inv(AtA+k*eye(2)); %As in equation in the method section above
        dp = real((iAtAnew*At)*Error); %Equation for parameter change, essentially how much to move over a
        point to the calculated correct location
    end
end

```

```

    xp = xp+real(dp(1,1));
    zp = zp-real(dp(2,1));
    %Array to hold all positions for the inversion
    Pos = [Pos xp];
end
%40 is the "Hold" variable as used in the Logistic map construction
Pos2 = Pos(j-40:j);%Throw away some positions as only the bifurcations (reached at the end) are of interest
Pos2 = round(Pos2);%Can round to have more easily read values or left in decimal form
%LocalisedFinalPos = unique(Pos2);%Can extracte unique entries to have more easily read values or left in
decimal form
AmountOfUniques=length(Pos2);%length(LocalisedFinalPos);
for n = 1:AmountOfUniques
    CArray = [CArray C]; %For plotting
end
FinalPositions = [FinalPositions Pos2];%For plotting

CL = length(CArray);
PL = length(FinalPositions);

end

%Plotting
%Titles and labels can be altered as desired by the user or an input
%function can be used
figure(1);
Title1 = ['actual X : 10 -- starting position ', num2str(OrigX), ' Iterations: ',num2str(j)];
scatter(CArray,FinalPositions,0.7); xlabel('C Parameter');ylabel('Final position (m)'); title(Title1);axis([1 4.5 -30
30]);

%Alter the density and range of the bifurcation diagram by changing the
%"Hold", rounding and uniques
end

```

5 – axazPlots

```
function axazPlots
```

```

%Brandon Dias
%Revised on 08/10/2018
%676053
%This algorithm uses established equations for the generation of gravitational responses
%due to a buried cylinder model and its associated derivatives with respect to x and z.

%This algorithm can be used to generate the responses due to single or
%multi body systems and the user can add or remove bodies and display
%accordingly

CorS=input('1 for cylinder or 2 for sphere: '); %Plot the derivative functions for the cylinder or sphere model

G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
Mass = 25;
X = -100:100; %Range
x1 = 0; % Position along the range - can be hard coded as desired

```

```

z1 = 25;          % The depth at which the cylinder is buried - can be hard coded as desired
gmz = G*Mass*z1;
gm = G*Mass;

if CorS == 1
    Gobs2 = gmz./((X-x2).^2+z1^2);%Gravitational response due to a buried cylinder model
    gx = (X-x1)*(-2*gm*z1)/(((X-x1).^2+z1^2).^2);          %ax | Derivative of the profile Gobs2 with respect to x
    (position)
    gz = (gm*(-z1^2+(X-x1).^2))./(z1^2+(X-x1).^2).^2;      %az | Derivative of the profile Gobs2 with respect to z
    (depth)
    %Display
    figure(1)
    subplot(2,1,2);plot(X,gx,'b',X,gz,'g');title('x parameter derivative in blue and z parameter derivative in
green');xlabel('Range (m)');ylabel('Responses (mGal)');
    subplot(2,1,1);plot(X,Gobs2,'r');title('Observed model response');xlabel('Range (m)');ylabel('Responses (mGal)');
    figure(2)
    plot(X,gx,'b',X,gz,'g');title('x parameter derivative in blue and z parameter derivative in green');xlabel('Range
(m)');ylabel('Responses (mGal)');
    % 'b' and 'g' refer to a blue or green coloured response, respectively
end

if CorS == 2
    Gobs2=gm*z1./(((X-x1).^2+z1^2).^1.5);%Gravitational response due to a buried sphere model
    gx = (X-x1)*(-3*gm*z1)/(((X-x1).^2+z1^2).^2.5);          %ax | Derivative of the profile Gobs2 with respect to x
    (position)
    gz = (gm*(-2*z1^2+(X-x1).^2))./(z1^2+(X-x1).^2).^2.5;    %az | Derivative of the profile Gobs2 with respect to
z (depth)
    %Display
    figure(1)
    subplot(2,1,2);plot(X,gx,'b',X,gz,'g');title('x parameter derivative in blue and z parameter derivative in
green');xlabel('Range (m)');ylabel('Responses (mGal)');
    subplot(2,1,1);plot(X,Gobs2,'r');title('Observed model response');xlabel('Range (m)');ylabel('Responses (mGal)');
    figure(2)
    plot(X,gx,'b',X,gz,'g');title('x parameter derivative in blue and z parameter derivative in green');xlabel('Range
(m)');ylabel('Responses (mGal)');
    % 'b' and 'g' refer to a blue or green coloured response, respectively
end

end
end

```

6 - BasinsOfAttractionsAndDimensionsThreeImagesK

```

function BasinsOfAttractionsAndDimensionsThreeImagesK
%Brandon Dias
%Revised on 20/01/2019
%676053

```

```

%This algorithm runs over an x and z range and at each point runs the
%Ridge-regression inversion process. Whether or not convergence to the
%defined local minima is met is then marked by the basin of attraction in
%yellow. The inversion is run with damping and the damping factor is
%increased should the discrepancy matrix decrease and vice-versa. Once the

```

%basin of attraction is created the box-counting method is run over it and
 %the dimension calculated. The models present are for single buried
 %cylinders.

```

XR=-100:100;%Range over the geological model (for plotting images)
ZR=0:100; %Range of depths for the geological model
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X = -100:100; %Range (for calculations)
Shift = 0; %Position of the buried cylinder
Mass = 2000;
gm = G*Mass;
MaxBin=10; %Defined largest size which the box can grow to in the box-counting method
INDEXi=1; %Place holder index for placing the convergence patterns
INDEXj=2; %Place holder index for placing the dimension calculations

for AmountOfImages = 1:3 %Amount of images at varied depths to sample
    Depth = input('Depth: ');
    gmz = G*Mass*Depth;
    GravityResponse1 = gmz./((X-Shift).^2+Depth^2); %Synthetic gravitational response for a buried cylinder model
    Gobs= GravityResponse1;

    I=0; %Index for basin of convergence

    XI = (1-100); %Starting initial parameter value for x
    ZI = 1; %Starting initial parameter value for z
    Gcalc=gm*ZI./((X-XI).^2+ZI^2);
    previousD = Gcalc-Gobs; %previousD is used to establish a difference at each iteration between the current
    parameter response and the previous parameter response. This difference is DampeningDetermination below

    FFF=zeros(101,201); %Empty array used to create the basin of attraction pattern
    for ICount = 1:200 %Range along x
        P=ICount/201; %P is displayed on the command window so the user can observe if the program is still
        running, stuck and how far along it is
        display([num2str(P*100/3), '%']); %Display P
        I=I+1;
        J=0;
        for JCount = 1:100

            J=J+1;
            k=2; %Starting constant for the damping factor. k=0 for Moore-Penrose
            xp = ICount-100; %Start at first index in the ranges
            zp = JCount;
            for InversionCount = 1:15 %Arbitrary maximum amount of iterations
                %The inversion runs as described in the write up
                %First the discrepancy matrix, D, is created
                Gcalc=gm*zp./((X-xp).^2+zp^2);

                Error = Gcalc-Gobs; %difference "D"
                Error=Error';

                %Here the A matrix is created by concatenating ax and az
                gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2); %ax
                gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

                A = [gx;gz]; %A
            end
        end
    end
end

```

```

    A = A';
    At=A';
    AtA=At*A;
%The equation stands with the addition of the k constant
    iAtAnew = inv(AtA+k*eye(2));
%The parameter change matrix is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
respectively
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
%Here a check is performed to ensure the depth doesn't turn
%negative, indicating the cylinder existing above the
%ground. The maths tends to solve for numbers and not from
%real world geophysical systems
    if zp < 0
        zp=0;
    end

%Here the check is performed to see if the discrepancy matrix has grown or
%not. Following this; should the discrepancy or misfit increase, the k
%factor is halved and vice versa.
    DampeningDetermination = sum(Error'>previousD);
    if DampeningDetermination > 201/2
        k=k*2;
    end
    if DampeningDetermination < 201/2
        k=k/2;
    end

    PreviousD=Error';
%Should convergence have been met, the parameter value indices are marked
%and the basin of attraction is created.
    if abs(xp-Shift)<0.1 && abs(zp-Depth)<0.1 %Left
        FFF(J,I)=1;
        break;
    end

end
end
end

figure(1)
subplot(3,2,INDEXi);image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');%The basin of attraction image
INDEXi=INDEXi+2; %Next image
A=FFF;
[Yr,Xr]=size(A); %Bounds within the image (x or y limits) are defined
%In the box-counting method a box is moved over the image and whenever the
%box encounters a boundary the a count is incremented. This is done over a
%range of box sizes. What is created is an array of box sizes as a function
%of the amount of times said box encountered a boundary.
SizeOfBox = [];
CountOfBox = [];

%Bin Code starts here
i=1; %Box growth rate

```

```

N=1;

while N<MaxBin %Maintain box doesnt grow larger than you would like
    N=N+i; %Box size
    %display(N);
    SizeOfBox = [SizeOfBox N];
    BinSize = N;
    BOXcount1 = 0;
    for I = 1:BinSize:(Xr-BinSize-1)
        for J = 1:BinSize:(Yr-BinSize-1)
            Count0 = 0;
            Count1 = 0;

            Bin = A(J:J+BinSize,I:I+BinSize);
            if sum(any(Bin==0)) > 0
                Count0 = 1; %0s are indicative of failure to converge
            end
            if sum(any(Bin==1)) > 0
                Count1 = 1; %1s are indicative of convergence
            end
            if Count0 == 1 && Count1 == 1 %A combination of 0s and 1s within a cell would represent a boundary
of the basin of attraction and as a result the count is incremented
                BOXcount1 = BOXcount1+1;
            end
        end
    end
    CountOfBox = [CountOfBox BOXcount1];
end

LSize = log10(SizeOfBox);
LCount =log10(CountOfBox);
%The polyfit function computes the gradient from the line of best fit from
%provided data. It does this by creating a y=mx+b relationship from the
%supplied data and calculates m, the gradient. polyfit(1)=m and
%polyfit(2)=b
p1=polyfit(LSize,LCount,1);
m1 = p1(1);
Fdim1=m1*(-1); %The slope of the log of the bin size as a function of the log of the count is negative. Therefore
to obtain the dimension, the value is multiplied by -1
Fdim1=round(Fdim1,2);

%.b' and lsline graph a line of best fit over the data
figure(1)
subplot(3,2,INDEXj);plot(LSize,LCount,'.b');lsline;xlabel('Log of Bin Size');ylabel('Log of Count');Title1 =
['Fractal dimension : ',num2str(Fdim1)];title(Title1);
INDEXj=INDEXj+2; %The plot index is increased to plot the next plot

end

end

```

7 – BasinOfAttractionKIterationsTracked

```
function BasinOfAttractionKIterationsTracked
```

```
%Brandon Dias
```

```
%Revised on 30/01/2019
```

```
%676053
```

```
%This algorithm runs over an x and z range and at each point runs the
```

```
%Ridge-regression inversion process. Whether or not convergence to the
```

```
%defined local minima is met is then marked by the basin of attraction in
```

```
%yellow. The inversion is run with damping and the damping factor is
```

```
%increased should the discrepancy matrix decrease and vice-versa.
```

```
%This can be run as a Moore-Penrose inversion with no dampin if you set k=0
```

```
XR=-100:100;%Range over the geological model (for plotting images)
```

```
ZR=0:100; %Range of depths for the geological model
```

```
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
```

```
X = -100:100; %Range (for calculations)
```

```
Shift = 0;%Separation between sources or location along the range Input or hard coded. 0 is used since the model is  
a single buried cylinder
```

```
Mass = 2000;
```

```
gm = G*Mass;
```

```
%for Shift = 35:10:45 %A for loop can be run here should you want to generate many images, this loop can be run  
for depth, separation or even scale
```

```
%-----
```

```
Depth=15; %Depth at which cylinder is buried
```

```
gmz = G*Mass*Depth;
```

```
Gobs = gmz./((X-Shift).^2+Depth^2); %Synthetic gravitational response for a buried cylinder model
```

```
I=0; %Index for basin of convergence
```

```
XI = (1-100); %Starting initial parameter value for x
```

```
ZI = 1; %Starting initial parameter value for z
```

```
Gcalc=gm*ZI./((X-XI).^2+ZI^2);
```

```
previousD = Gcalc-Gobs; %previousD is used to establish a difference at each iteration between the current  
parameter response and the previous parameter response. This difference is DampeningDetermination below  
SCALE=input('Scale : '); %The constant by which the solution space will be enlarged. Larger numbers mean more  
densely spaced parameter values and therefore an increase in resolution
```

```
FFF=zeros(SCALE*100,SCALE*200); %Empty array used to create the basin of attraction pattern
```

```
for ICount = 1:1/SCALE:200 %Range along x | 1/SCALE accounts for the increased solution space exactly
```

```
    P=ICount/201; %P is displayed on the command window so the user can observe if the program is still  
running, stuck and how far along it is
```

```
    display([num2str(P*100),'%']); %Display P
```

```
    I=I+1;
```

```
    J=0;
```

```
    for JCount = 1:1/SCALE:100
```

```
        J=J+1;
```

```
        k=6; %Starting constant for the damping factor, can be set to zero to remove damping
```

```
        xp = ICount-100; %Start at first index in the ranges
```

```
        zp = JCount;
```

```
        for InversionCount = 1:10 %Arbitrary maximum amount of iterations
```

```
%The inversion runs as described in the write up
```

```
%First the discrepancy matrix, D, is created
```

```
        Gcalc=gm*zp./((X-xp).^2+zp^2);
```

```
        Error = Gcalc-Gobs;
```

```
%difference "D"
```

```

    Error=Error';
%Here the A matrix is created by concatenating ax and az
    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az

    A = [gx;gz];    %A
    A = A';
    At=A';
    AtA=At*A;
%The equation stands with the addition of the k constant
    iAtAnew = inv(AtA+k*eye(2));
%The parameter change matrix is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
respectively
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);

%Here the check is performed to see if the discrepancy matrix has grown or
%not. Following this; should the discrepancy or misfit increase, the k
%factor is halved and vice versa.
%Comment out this section in order to remove the effects of damping or keep
%the damping constant set at a value
%-----
%    DampeningDetermination = sum(Error'>previousD);
%    if DampeningDetermination > 201/2
%        k=k*2;
%    end
%    if DampeningDetermination < 201/2
%        k=k/2;
%    end
%    PreviousD=Error';
%-----
%Should convergence have been met, the parameter value indices are marked
%and the basin of attraction is created.
%Here the basin is tracked as a function of the amount of iterations to
%converge. "break" exits the iterations ensuring the value is not over
%written
    if abs(xp-Shift)<0.5 && abs(zp-Depth)<0.5    %Left
        FFF(J,I)=InversionCount;
        %FFF(J,I)=1;
        break;
    end
end
end
end
end
%Image display of basin of attraction in the solution space
figure(1)
image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap hot;colorbar;
%Colormap is set by preference
end

```

8 – FracZoom

```

function FracZoom
%Brandon Dias
%Revised on 20/01/2019
%676053

%This algorithm runs over an x and z range and at each point runs the
%Moore-Penrose inversion process. Whether or not convergence to the
%defined local minima is met is then marked by the basin of attraction in
%yellow. The input x and y ranges are defined by the user here (in or out
%of a loop) to zoom into areas of interest in the basins of attraction

%for W = 1:3
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X = -100:100; %Range (for calculations)
x1 = 0;%Separation between sources or location along the range Input or hard coded

gm = G*Mass;
z1=15; %Depth of buried source
Mass = 2000;
gmz = G*Mass*z1;
gm = G*Mass;
Gobs = gmz./((X-x1).^2+z1^2);%Synthetic gravitational response for a buried cylinder model

SCALE=1;
ConvergeMatrix = zeros(SCALE*100,SCALE*200);%Empty array used to create the basin of attraction pattern.
Solution space
I=0; %Index for basin of convergence
SX=input('Starting X: '); %Starting x position along which to create the basin of attraction. The range can be run
normally from -100:100 as set up in X, then defined again once a boundary is observed.
SX=SX+100; %100 is added since X starts at -100
EX=input('Ending X: '); %Ending x position along which to create the basin of attraction. The range can be run
normally from -100:100 as set up in X, then defined again once a boundary is observed.
EX=EX+100; %100 is added since X starts at -100
SD=input('Starting Z: '); %Starting z position along which to create the basin of attraction. The range can be run
traditionally as from 0:100, then defined again once a boundary is observed.
ED=input('Ending Z: '); %Ending z position along which to create the basin of attraction. The range can be run
traditionally as from 0:100, then defined again once a boundary is observed.
XR=(EX-SX)/200; %The interval is divided by 200 to match the size of the solution space
ZR=(ED-SD)/100; %The interval is divided by 100 to match the size of the solution space

for ICount = SX:XR/SCALE:EX %Range along x | 1/SCALE accounts for the increased solution space exactly
    PCheck = ICount/EX; %P is displayed on the command window so the user can observe if the program is
still running, stuck and how far along it is
    display(num2str(PCheck*100));
    I=I+1;
    J=0;
    for JCount = SD:ZR/SCALE:ED

        J=J+1;
        k=0; %Starting constant for the damping factor, can be set to zero to remove damping
        xp = ICount-100; %Start at first index in the ranges
        zp = JCount;
        for i = 1:10
%Arbitrary maximum amount of iterations
%The inversion runs as described in the write up

```

```

%First the discrepancy matrix, D, is created
Gcalc=gm*zp./((X-xp).^2+zp^2);

Error = Gcalc-Gobs;           %difference "D"
Error=Error';

%Here the A matrix is created by concatenating ax and az
gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

A = [gx;gz];                %A
A = A';
At=A';
AtA=At*A;
iAtAnew = inv(AtA+k*eye(2)); %The equation stands with the addition of the k constant
dp = (iAtAnew*At)*Error;    %The parameter change matrix is created with regards to the x and z
parameter held in dp(1,1) and dp(2,1), respectively
xp = xp+dp(1,1);
zp = zp-dp(2,1);

%Here the check is performed to see if the discrepancy matrix has grown or
%not. Following this; should the discrepancy or misfit increase, the k
%factor is halved and vice versa.
%Comment out this section in order to remove the effects of damping or keep
%the damping constant set at a value
%-----
%    DampeningDetermination = sum(Error'>previousD);
%    if DampeningDetermination > 201/2
%        k=k*2;
%    end
%    if DampeningDetermination < 201/2
%        k=k/2;
%    end
%
%    PreviousD=Error';
%-----
%Should convergence have been met, the parameter value indices are marked
%and the basin of attraction is created.
%Here the basin is tracked as a function of the amount of iterations to
%converge. "break" exits the iterations ensuring the value is not over
%written

    if abs(xp-x1)<1 && abs(zp-z1)<1 %Left
        ConvergeMatrix(J,I)=i;
        break;
        %whos;pause;
    end
end
end
end
end
%Display
xRange=SX-100:XR:EX-100;%Axis indeces

yDepth=SD:ZR:ED;%Axis indeces
display('Convergence Matrix created');
figure(1)

```

```
image(xRange,yDepth,ConvergeMatrix);xlabel('Range (m)');ylabel('Depth (m)');%colormap
hot;colorbar;caxis([0,3]); %title('Fractal convergence pattern showing intensity as amount of iterations - separation:
76 m and depth: 10 m');
```

```
end
```

9 – MisfitFunctionPlusModel

```
function MisfitFunctionPlusModel
```

```
%Brandon Dias
```

```
%Revised on 01/11/2018
```

```
%676053
```

```
%This algorithm calculates the misfit function as a result of two buried
%cylinders (minimums) and an input cylinder of identical masses and radii
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X=-200:200; %Range of profiles created
xm1=-30; %Model x location
xm2=50; %Mode2 x location
zm1=5; %Model z location
zm2=15; %Mode2 z location
XMOD=[xm1 xm2]; %Parameter values for plotting models in scatter
ZMOD=[zm1 zm2]; %Parameter values for plotting models in scatter
Mass = 20;
gm = G*Mass;
Gobs = gm*zm1./((X-xm1).^2+zm1^2)+gm*zm2./((X-xm2).^2+zm2^2); %Synthetic observed gravitational
response for two buried cylinder models
z1=10; %Depth of the test cylinder is placed at the lowest depth to get the smallest minimum
M=[];%Empty misfit array which will be used to create misfit function
```

```
for xp=-100:100 %Range of misfit function
    Gcalc=gm*z1./((X-xp).^2+z1^2); %Synthetic calculated gravitational response for a buried cylinder model
    %Misfit calculated as in equation (9)
    Error = Gcalc-Gobs;
    N=length(Gcalc);
    Mis=sum(Error.^2);
    MisfitM1=Mis/N;
    M=[M MisfitM1];
end
```

```
M=M*10;
figure(2)
scatter(XMOD,ZMOD,100,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-100 100 0 50]);

XR=-100:100;
figure(1)
plot(XR,M);
figure(3)
subplot(2,1,1);plot(XR,M);xlabel('Range (m)');ylabel('Misfit');title('Misfit surface');
subplot(2,1,2);scatter(XMOD,ZMOD,100,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-100 100 0
50]);title('Geological model');
```

```
end
```

10 – BasinOfAttractionKIterationsTrackedTwoBodies

```
function BasinOfAttractionKIterationsTrackedTwoBodies
```

```
%Brandon Dias
%Revised on 07/01/2018
%676053
```

```
%This algorithm runs over an x and z range and at each point runs the
%Ridge-regression inversion process. Whether or not convergence to the
%defined local minima is met is then marked by the basin of attraction in
%yellow. The inversion is run with damping and the damping factor is
%increased should the discrepancy matrix decrease and vice-versa.
```

```
XR=-100:100;%Range over the geological model (for plotting images)
ZR=0:100; %Range of depths for the geological model
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X = -100:100; %Range (for calculations)
Shift = 76/2;%Separation between sources or location along the range | Can be input or hard coded as desired
```

```
Mass = 20000;
gm = G*Mass;
%for Shift = 35:10:45 %A for loop can be run here should you want to generate many images, this loop can be run
for depth, separation or even scale
%-----
Depth=10; %Depth at which cylinder is buried | Can be input or hard coded as desired
```

```
gmz = G*Mass*Depth;
Gobs = gmz./((X-Shift).^2+Depth^2)+gmz./((X+Shift).^2+Depth^2); %Synthetic gravitational response for two
buried cylinder models
```

```
I=0; %Index for basin of convergence
kConstant=input('k: '); %In this algorithm the k value is kept constant throughout the inversion process. It is defined
by the user
```

```
SCALE=5;%The constant by which the solution space will be enlarged. Larger numbers mean more densely spaced
parameter values and therefore an increase in resolution
```

```
FFF=zeros(SCALE*100,SCALE*200); %Empty array used to create the basin of attraction pattern
```

```
for ICount = 1:1/SCALE:200 %Range along x | 1/SCALE accounts for the increased solution space exactly
P=ICount/201; %P is displayed on the command window so the user can observe if the program is still
running, stuck and how far along it is
```

```
display([num2str(P*100),'%']); %Display P
```

```
I=I+1;
```

```
J=0;
```

```
for JCount = 1:1/SCALE:100
```

```
J=J+1;
```

```
k=kConstant; %Starting constant for the damping factor
```

```
xp = ICount-100; %Start at first index in the ranges
```

```
zp = JCount;
```

```
for InversionCount = 1:10 %Arbitrary maximum amount of iterations
```

```
%The inversion runs as described in the write up
```

```

%First the discrepancy matrix, D, is created
Gcalc=gm*zp./((X-xp).^2+zp^2);

Error = Gcalc-Gobs;           %difference "D"
Error=Error';

%Here the A matrix is created by concatenating ax and az
gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

A = [gx;gz];                %A
A = A';
At=A';
AtA=At*A;

%The equation stands with the addition of the k constant
iAtAnew = inv(AtA+k*eye(2));
%The parameter change matric is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
respectively
dp = (iAtAnew*At)*Error;
xp = xp+dp(1,1);
zp = zp-dp(2,1);

%Should convergence have been met, the parameter value indices are marked
%and the basin of attraction is created.
%Here the basin is tracked as a function of the amount of iterations to
%converge. "break" exits the iterations ensuring the value is not over
%written
    if abs(xp-Shift)<1 && abs(zp-Depth)<1 %Left model
        FFF(J,I)=InversionCount; %By assigning a value equal to the amount of iterations in the solution, the
basin of attraction now tracks the amount of iterations to converge, as intensity in colour
        %FFF(J,I)=1;
        break;
    end
    if abs(xp+Shift)<1 && abs(zp-Depth)<1 %Right model
        FFF(J,I)=InversionCount; %By assigning a value equal to the amount of iterations in the solution, the
basin of attraction now tracks the amount of iterations to converge, as intensity in colour
        %FFF(J,I)=1;
        break;
    end
end
end
end
end
%Image display of basin of attraction in the solution space
figure(1)
image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap hot;
%Colormap is set by preference
end

```

11 – VariedXorZkTwoBodies

```
function VariedXorZkTwoBodies
```

```
%Brandon Dias
```

%Revised on 08/04/2018

%676053

```
%This algorithm runs over an x and z range and at each point runs the
%Ridge-regression inversion process. Whether or not convergence to the
%defined local minima is met is then marked by the basin of attraction in
%yellow. The inversion is run with damping and the damping factor is
%increased should the discrepancy matrix decrease and vice-versa. Once the
%basin of attraction is created the box-counting method is run over it and
%the dimension calculated. The models present are for two buried
%cylinders.
%This can be run as a Moore-Penrose inversion with no dampin if you set k=0
%for W=1:3
XR=-100:100;%Range over the geological model (for plotting images)
ZR=0:100; %Range of depths for the geological model
G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X = -100:100; %Range (for calculations)
Shift = 76;%Separation between the two buried cylinders is twice Shift | Input or hard coded
Shift = Shift/2;
Mass = 2000;
gm = G*Mass;
%for Shift = 35:10:45 %A for loop can be run here should you want to generate many images, this loop can be run
for depth, separation or even scale
%-----
Depth=5; %Depth at which cylinder is buried | Input or hardcoded
gmz = G*Mass*Depth;
GravityResponse1 = gmz./((X-Shift).^2+Depth^2) + gmz./((X+Shift).^2+Depth^2); %Synthetic gravitational
response for a buried cylinder model
Gobs= GravityResponse1;

I=0; %Index for basin of convergence

XI = (1-100); %Starting initial parameter value for x
ZI = 1; %Starting initial parameter value for z
Gcalc=gm*ZI./((X-XI).^2+ZI^2);
previousD = Gcalc-Gobs; %previousD is used to establish a difference at each iteration between the current
parameter response and the previous parameter response. This difference is DampeningDetermination below
SCALE=input('Scale: ');
MaxBin=15*SCALE; %Defined largest size which the box can grow to in the box-counting method
FFF=zeros(SCALE*101,SCALE*201); %Empty array used to create the basin of attraction pattern
for ICount = 0:1/SCALE:200 %Range along x
    P=ICount/201; %P is displayed on the command window so the user can observe if the program is still
running, stuck and how far along it is
    display([num2str(P*100),' %']); %Display P
    I=I+1;
    J=0;
    for JCount = 0:1/SCALE:100

        J=J+1;
        k=2; %Starting constant for the damping factor, can be set to zero to remove damping
        xp = ICount-100; %Start at first index in the ranges
        zp = JCount;
        for InversionCount = 1:15 %Arbitrary maximum amount of iterations
            %The inversion runs as described in the write up
            %First the discrepancy matrix, D, is created
```

```

Gcalc=gm*zp./((X-xp).^2+zp^2);

Error = Gcalc-Gobs;           %difference "D"
Error=Error';

%Here the A matrix is created by concatenating ax and az
gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;  %az

A = [gx;gz];                %A
A = A';
At=A';
AtA=At*A;

%The equation stands with the addition of the k constant
iAtAnew = inv(AtA+k*eye(2));
%The parameter change matrix is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
%respectively
dp = (iAtAnew*At)*Error;
xp = xp+dp(1,1);
zp = zp-dp(2,1);

%Here the check is performed to see if the discrepancy matrix has grown or
%not. Following this; should the discrepancy or misfit increase, the k
%factor is halved and vice versa.
%Comment out this section in order to remove the effects of damping
%-----
DampeningDetermination = sum(Error'>previousD);
if DampeningDetermination > 201/2
    k=k*2;
end
if DampeningDetermination < 201/2
    k=k/2;
end

PreviousD=Error';
%-----
%Should convergence have been met, the parameter value indices are marked
%and the basin of attraction is created.
if abs(xp-Shift)<0.5 && abs(zp-Depth)<0.5 %Left model
    FFF(J,I)=1;
    break;
end
if abs(xp+Shift)<0.5 && abs(zp-Depth)<0.5 %Right model
    FFF(J,I)=2;
    break;
end
if abs(xp)<0.5 %middle of the profile | Can be commented out to solely track left and right model if desired
    FFF(J,I)=3;
    break;
end

end
end
end
figure(2) %Ordinarily images as generated by the "image" function are scaled to the length of the colormaps (60),
using imagesc scales the image to the indices in the solution matrix and makes a good first pass check

```

```

imagesc(FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap jet;

%Basin
figure(1)
subplot(1,2,1);image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap jet;

A=FFF;
[Yr,Xr]=size(A); %Bounds within the image (x or y limits) are defined
%In the box-counting method a box is moved over the image and whenever the
%box encounters a boundary the a count is incremented. This is done over a
%range of box sizes. What is created is an array of box sizes as a function
%of the amount of times said box encountered a boundary.
SizeOfBox = [];
CountOfBox = [];

%Bin Code starts here
i=1; %Box growth rate
N=1;

while N<MaxBin %Maintain box doesnt grow larger than you would like
    N=N+i; %Box size
    %display(N);
    SizeOfBox = [SizeOfBox N];
    BinSize = N;
    BOXcount1 = 0;
    for I = 1:BinSize:(Xr-BinSize-1)
        for J = 1:BinSize:(Yr-BinSize-1)
            Count0 = 0;
            Count1 = 0;

            Bin = A(J:J+BinSize,I:I+BinSize);
            if sum(any(Bin==0)) > 0
                Count0 = 1; %0s are indicative of failure to converge
            end
            if sum(any(Bin==1)) > 0
                Count1 = 1; %1s are indicative of convergence
            end
            if Count0 == 1 && Count1 == 1 %A combination of 0s and 1s within a cell would represent a boundary of
the basin of attraction and as a result the count is incremented
                BOXcount1 = BOXcount1+1;
            end
        end
    end
    CountOfBox = [CountOfBox BOXcount1];
end
%The arrays have their values logged and negative slopes inverted so as to
%maintain positive dimension value (m=m*-1)

LSize = log10(SizeOfBox);
LCount =log10(CountOfBox);

%The polyfit function computes the gradient from the line of best fit from
%provided data. It does this by creating a y=mx+b relationship from the
%supplied data and calculates m, the gradient. polyfit(1)=m and

```

```

%polyfit(2)=b
p1=polyfit(LSize,LCount,1);
m1 = p1(1);
Fdim1=m1*(-1);

Fdim1=round(Fdim1,2);
figure(1)
subplot(1,2,2);plot(LSize,LCount,'b');lsline; Title1 = ['Fractal dimension : ',num2str(Fdim1)]; xlabel('Log of Bin
Size');ylabel('Log of Count');title(Title1); axis tight; colormap parula;

end

```

12 – ResponsePlusModel

```

function ResponsePlusModel
%Brandon Dias
%Revised on 05/06/2017
%676053

%This algorithm takes inputs from the user for two buried cylinder's
%locations and depths and generates two responses, respectively.

G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X=-100:100; %Range of profiles created
xm1=input('x1: '); %Model x location
xm2=input('x2: '); %Mode2 x location
zm1=input('z1: '); %Model z location
zm2=input('z2: '); %Mode2 z location
XMOD=[xm1 xm2]; %Parameter values for plotting models in scatter
ZMOD=[zm1 zm2]; %Parameter values for plotting models in scatter
Mass = 20;
gm = G*Mass;
G1 = gm*zm1./((X-xm1).^2+zm1^2); %Synthetic observed gravitational response for first buried cylinder model
G2 = gm*zm2./((X-xm2).^2+zm2^2); %Synthetic observed gravitational response for first buried cylinder model
figure(3)
subplot(2,1,1);plot(X,G1,'y',X,G2,'b');xlabel('Range (m)');ylabel('Response (mGal)');
%Or can plot observed model for a two body system with G3 = G1+G2
subplot(2,1,1);plot(X,G3);xlabel('Range (m)');ylabel('Response (mGal)');
subplot(2,1,2);scatter(XMOD,ZMOD,100,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-100 100 0
50]);title('Geological model');

%Spheres in scatter can also be plotted in different colours
%C=linspace(1,2,length(XMOD)); %Linspace creates a linear array of colours as spread out over the length of the
amount of indices in the array
%scatter(XMOD,ZMOD,[100],C,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-50 50 0 50]);

end

```

13 – InversionsWithScalingFactorModelIncluded

```

function InversionsWithScalingFactorModelIncluded
%Brandon Dias
%Revised on 18/01/2019
%676053

%This algorithm runs an inversion four times at varied (input) scaling
%factors. The positions as function of the iterations are tracked to
%observe the effect of the scaling factor.
%A model is also created of the observed and calculated models

Xmodel=[];%Array used for model plots (x location)
Zmodel=[];%Array used for model plots (z depth)

for i = 1:4
    k=0;%Moore-Penrose
    C = input('C value: ');%Scaling factor is supplied by the user

    G = 1; % 6.67E-11; % G and Mass is almost entirely analytical so can be any value
    X = -100:100;%Range over the geological model (for plotting images)

    Depth = 10; %Observed model depth
    Shift = 10; %Observed model location along the range
    Xmodel=[Xmodel Shift];
    Zmodel=[Zmodel Depth];
    Mass = 20;
    gmz = G*Mass*Depth;
    gm = G*Mass;
    %Gobs = gmz./(((X-Shift).^2+Depth^2).^1.5); %Sphere
    Gobs = gmz./((X-Shift).^2+Depth^2); %Observed response

    xp = -5; %Initial calculated model location
    OrigX = xp;
        %Hard code
    zp = 10; %Initial calculated model depth
    Xmodel=[Xmodel xp];
    Zmodel=[Zmodel zp];
    GcalcPlot=gm*zp./((X-xp).^2+zp^2);%Calculated response
    D=linspace(2,7,length(Xmodel));%Linspace creates a linear array of colours as spread out over the length of the
    amount of indices in the array
    %The model and responses
    figure(1)
    subplot(2,1,1);plot(X,Gobs,'b',X,GcalcPlot,'y');xlabel('Range (m)');ylabel('Response (mGal)');axis([-60 60 0 2.5]);
    subplot(2,1,2);scatter(Xmodel,Zmodel,[100],D,'filled');xlabel('Range (m)');ylabel('Depth (m)'); axis ij; axis([-60
    60 0 20]);
    %Position array tracks the parameters as they invert through the
    %iterations
    Positions = [xp];
    P2=[zp];
    Iterations = [0];

    ENDPOINT = 80;%How many iterations are run

    for InversionCount = 1:ENDPOINT

```

```

Iterations = [Iterations InversionCount];
Gcalc=gm*zp./((X-xp).^2+zp^2); %Calculated response
Error=Gcalc-Gobs;

Error=Error*C; %As per equation (13)

gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2); %ax
gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az
A = [gx;gz]; %A
A = A';
At=A';
AtA=At*A;
iAtA = inv(AtA);
dp = real((iAtA*At)*Error);
xp = xp+dp(1,1);
zp = zp-dp(2,1);
%This prevents the model from having a negative depth, and thus existing
%above the ground
if zp<0
    zp=0;
end

Positions = [Positions xp];
P2=[P2 zp];

end
%Display
figure(2);
Title1 = ['C = ',num2str(C)];
subplot(2,2,i); plot(Iterations,Positions); xlabel('Iterations');ylabel('x position (m)'); title(Title1);% The axis
function can be used if after the inversion is run bounds of interest are identified | axis([0 50 -25 50]);

end

end

```

14 – MisfitVsC

function MisfitVsC

```

%Brandon Dias
%Revised on 28/12/2017
%676053

```

```

%In order to investigate a potential relationship between the misfit at the
%start of an inversion process and the C value which would result in fewest
%amount of iterations to converge, the following algorithm is run.
%The inversion is run over an array of starting parameter values and at
%each the misfit is recorded, the inversion is run amd a record is taken of
%the amount of iterations to converge for the respective misfit.

```

```

Mscatter=[];
Cscatter=[];
Clist=0:0.001:1;

%Requirements to converge
ConCon=0.1;
k=0;
%Initial observed parameters
G = 1;% 6.67E-11;
X = -100:100;
Mass = 9;
x1=10;
z1=10;
gm = G*Mass;
Gobs = (gm*z1./((X-x1).^2+z1^2));
r=0;
for xpi=-40:2:40 %Initial calculated parameters
    r=r+1;
    PERC=100*(r/41);
    display(PERC);
    c=0;
    for zpi=0:40%Initial calculated parameters
        c=c+1;
        PERC2=100*(c/41);
        display([num2str(PERC2),' % in ',num2str(PERC),' %']);
        iarray=[];
        Carray=[];
        xp=xpi;
        zp=zpi;
        %Here the misfit is calculated as per equation (9)
        Gcalc=gm*zp./((X-xp).^2+zp^2);
        N=length(Gcalc);
        Error = Gcalc-Gobs;
        Mis=sum(Error.^2);
        Mis=Mis/N;
        %flag=1;

    for C= 0:0.001:2; %The array of scaling factors to test against the initial parameter values

        xp=xpi;
        zp=zpi;
        CHECKCHECK=1;
        for Iteration = 1:15

            Gcalc=gm*zp./((X-xp).^2+zp^2);
            Error = Gcalc-Gobs; %difference "D"
            Error=Error'*C;

            gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2); %ax
            gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

            A = [gx;gz]; %A
            A = A';

```

```

    At=A';
    AtA=At*A;
    iAtAnew = inv(AtA+k*eye(2));
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
    if abs(xp-x1)<ConCon
        iarray=[iarray Iteration];
        %flag=0;
        %These ensure the inversions dont run beyond the ideal
        %scaling factor
        SecondCheck=2;
        CHECKCHECK=2;
        break
    end
    %This forces the inversion to run if an ideal C factor was
    %not found
    if CHECKCHECK==1;
        SecondCheck=1;
    end

end

end
%    if flag == 1
%        break
%    end
%If an ideal scaling factor is found, append it and the corresponding
%misfit into arrays for plotting
    if SecondCheck ~= 1
        [A,B]=min(iarray);
        Cscatter=[Cscatter Clist(B)];
        Mscatter=[Mscatter Mis];
        display('YES');
    end
end
end
end

%This boolean check (BOO) eliminates ideal scaling factors of C, those are
%for respective Initial misfits of 0 where the initial parameters are equal
%to the initial observed parameters
A=Cscatter;
B=Mscatter;
n=0;
BOO=1;
while BOO==1
    n=n+1;
    if A(n)==0
        display('Yes');
        A(n)=[];
        B(n)=[];
        n=n-1;
    end
    if n>= length(A)
        BOO=2;

```

```

    end
end

Cscatter=A;
Mscatter=B;

%These plots are to observe the relationship and once observed the "axis" function can be applied to zoom in should
you so wish
figure(2)
plot(Cscatter,Mscatter);xlabel('C');ylabel('Original misfit');

figure(1)
scatter(Cscatter,Mscatter,'filled');xlabel('C');ylabel('Original misfit');axis([0 1 0 0.8]);

end

```

15 – CVSILoop

```

function CVSILoop

%Brandon Dias
%Revised on 28/12/2017
%676053

%This code will attempt to measure the attempted iterations as a function
%the scaling factor C used in an attempt to, atleast for this scenario,
%determine the optimum scaling factor.
%Initial conditions

for INDEXnum=1:3
    display(INDEXnum);
    %Convergence conditions. The initial parameters must fall within this
    %value of the observed parameter values
    ConCon=0.1;
    k=0;
    %Initial parameters
    G = 6.67E-11;    %G
    X = -100:100;    %Range
    Mass = 9000;    %Mass
    gm=G*Mass;
    x1=10;          %Observed location along the range
    z1=10;          %Observed depth of the observed model
    Gobs = (gm*z1./((X-x1).^2+z1^2)); %Observed response
    Csteps=0.001; %The interval with which the C value is incremented
    CLIM=2;
    Vs=CLIM/Csteps + 1; %There is a plus 1 due to the initial C value of 0
    %Arrays
    Carray=[];
    I=zeros(1,Vs)+25; %The inversions are capped at 25 (user defined)
    xG=input('Guess x: '); %Initial model location
    zG=input('Guess z: '); %Initial model depth
    %xG=40;

```

```

a=0;
%Loops
for C = 0:Csteps:CLIM
    Carray=[Carray C];
    xp=xG;
    zp=zG;
    a=a+1;
    for i = 1:25
        %Inversion as has been set up before
        Gcalc=gm*zp./((X-xp).^2+zp^2);

        Error = Gcalc-Gobs;          %difference "D"
        Error=Error*C;

        gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
        gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

        A = [gx,gz];                %A
        A = A';
        At=A';
        AtA=At*A;
        iAtAnew = inv(AtA+k*eye(2));
        dp = (iAtAnew*At)*Error;
        xp = xp+dp(1,1);
        zp = zp-dp(2,1);
        if abs(xp-x1)<ConCon && abs(zp-z1)<ConCon
            I(a)=i;
            %This assigns the amount of iterations, for a respective C
            %value, to the I array.
            break
        end
    end

end

end
%whos;pause;
%Display plots and titles. Titles and types of plots can be altered as
%desired.
figure(1)
T=['Iterations till convergence as a function of C - initial x = ',num2str(xG),' m and z = ',num2str(zG),' m -
observed x = 10 m and z = 10 m'];
scatter(Carray,I); title(T);xlabel('C');ylabel('Iterations to converge');
T=['Initial x = ',num2str(xG),' m and z = ',num2str(zG),' m'];
figure(2)
suptitle('Observed x = 10 m and z = 10 m');
subplot(3,1,INDEXnum);plot(Carray,I); title(T);xlabel('C');ylabel('Iterations');axis([0 2 0 25]);
end

end

```

16 – CsLoop

```
function CsLoop
```

```
%Revised on 29/12/2017
```

```
%Brandon Dias
```

```
%676053
```

```
%The algorithm runs over an array of input C values and attempts to solve
```

```
%for the C value in the hopes of landing at the same value.
```

```
%In order to use MATLAB's solve function the variables need to be defined
```

```
%as algebraic variables, with real values, and not as variable place
```

```
%holders in MATLAB
```

```
syms x C K D d y z1 z2 Zf Zi dp
```

```
assume([x,y,z1,z2,C,K,d,D,Zf,Zi,dp], 'real');
```

```
Zs=[];
```

```
CCalc=[];
```

```
CObs=[];
```

```
%Input parameters for observed and initial models
```

```
G = 6.67E-11;
```

```
X = -100:100;
```

```
%C=input('C: ');
```

```
for C = 0:0.2:4.5 %Array of input C values
```

```
    P=(C/5)*100;
```

```
    display([num2str(P), '%']);
```

```
    CObs=[CObs C];
```

```
    x1=0; %Observed model location
```

```
    x2=x1; %Initial model location
```

```
    z1=20; %Observed model depth
```

```
    Mass = 2000;
```

```
    gm = G*Mass;
```

```
    Data1 = gm*z1./((x1).^2+z1^2); %Observed response in over determined case
```

```
    Gradientz = (gm*(-z1^2+(x1).^2))./(z1^2+x1^2).^2;
```

```
    d=Gradientz;
```

```
    z2=10; %Initial model depth
```

```
    for i = 1:1; %Iterations can be run to first meet bifurcations, or the end of the i variable can be kept low to run equation (17) upon the inversion process's initiation
```

```
        %Equations (14) - (16) in practice
```

```
        Data2 = gm*z2./((x2).^2+z2^2);
```

```
        Data=Data1-Data2;
```

```
        Data=Data*C;
```

```
        dp=Data./Gradientz;
```

```
        z2=z2+dp;
```

```
    end
```

```
for i = 1:2;
```

```
    %Equations (14) - (16) in practice
```

```
    Data2 = gm*z2./((x2).^2+z2^2);
```

```
    Data=Data1-Data2;
```

```
    Data=Data*C;
```

```
    dp=Data./Gradientz;
```

```
    z2=z2+dp;
```

```
    Zs=[Zs z2];
```

```
end
```

```
%display(Zs);
```

```
L=length(Zs);
```

```

Zf=Zs(L);
Zi=Zs(L-1);
%Zf= input('Zf: ');
%Zi= input('Zi: ');
%Equation (17) in practice
Data2 = gm*Zi./((x2).^2+Zi^2);
Data=Data1-Data2;
dp=Data./Gradientz;
NumericC = solve(Zf-Zi-K*dp, K);
NumericC=double(NumericC);
CCalc=[CCalc NumericC];

end
%Display
figure(1)
scatter(CObs,CCalc,'filled'); xlabel('Input C value');ylabel('Calculated C value');% title('Calculated C vs input C');
end

```

17 – CvsKvsI2

`function CvsKvsI2`

```

%Brandon Dias
%676053
%Revised on 29/07/2018

```

```

%This algorithm runs an inversion on the same system (same initial and
%observed parameter values) over a range of C and k values. For each
%respective k and C value, the amount of iterations to converge, for that
%respective set-up, is recorded. This defines a new basin of attraction
%Initial parameters as defined in appendix 6

```

```

G = 1;%6.67E-11;
X = -100:100;
Mass = 900;
gm=G*Mass;
x1=10;
z1=10;
Gobs = (gm*z1./((X-x1).^2+z1^2));

```

```

iCount=[];

```

```

a=0;
%Same frac array and Scale as in appendix 6
SCALE=1;
iArray = zeros(SCALE*201,SCALE*201)+20;

```

```

for Chaos = 0:0.01/SCALE:2 %C array
    P=(Chaos/2)*100;
    display([num2str(P), ' %']);
    a=a+1;
    b=0;
    for k = 0:0.1/SCALE:20 %k array

```

```

b=b+1;
%Initial model depth and location
zp=input('Initial model depth: ');
xp=input('Initial model location: ');
for InversionCount = 1:20

    %display(InversionCount);
    %    display(k);

    Gcalc=gm*zp./((X-xp).^2+zp^2);

    Error = Gcalc-Gobs;                %difference "D"
    Error=Error*Chaos;

    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az

    A = [gx;gz];                        %A
    A = A';
    At=A';
    AtA=At*A;
    iAtAnew = inv(AtA+k*eye(2));
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);

    if zp<0
        zp=0;
    end
%Convergence conditions
    if abs(xp-x1)<1 && abs(zp-z1)<1    %Left
        iArray(b,a)=InversionCount;
        break;
    end
    %whos;pause;
end

end
end
kaxis = 0:20;
Caxis = 0:2;
%Flipdim is used beacuse the array was filled column by row but image plots
%rpws by columns
iArray=flipdim(iArray,1);
figure(4)
image(Caxis,kaxis,iArray);ylabel('k');xlabel('C');colorbar;

end

```

18 - LapyunovC

```
function LapuynovC
```

```
%Brandon Dias
%Revised on 19/08/2017
%676053
```

```
%This algorithm runs the inversion process, as laid out in equation (13) on
%two models with the same observed model but with initial models who's
%location is varied by a small value at the initial iteration. The Lyapunov
%exponent is calculated from the exponent of the gradient of the line
%of best fit between the log of the separation between the two trajectories
%as a function of the inversion iterations. The Lyapunov exponent is laid
%out in equation (19).
```

```
C=input('C: '); %C as defined by user
```

```
D=0.001; %The distance between the two initial locations can be hardcoded or made with an input
```

```
%Initial parameter values as defined in appendix 6
```

```
G = 6.67E-11;
```

```
X = -100:100;
```

```
k=0;
```

```
TotI=10;
```

```
%Observed model parameter values
```

```
Depth = 10;
```

```
Shift = 10;
```

```
Mass = 2000;
```

```
gmz = G*Mass*Depth;
```

```
gm = G*Mass;
```

```
Gobs = gmz./(((X-Shift).^2+Depth^2).^1.5);
```

```
FirstX=5; %First initial calculated model's location
```

```
SecondX=FirstX-D; %offset to create second initial model
```

```
xp = FirstX;
```

```
OrigX = xp;
```

```
zp=Depth+1;
```

```
X1=[];
```

```
X2=[];
```

```
%Inversion as laid out in appendix 6
```

```
%The inversion is first run on the the first model
```

```
for InversionCount1 = 1:TotI
```

```
    X1=[X1 xp];
```

```
    Gcalc=(gm*zp)./(((X-xp).^2+zp^2).^1.5);
```

```
    Error = Gcalc-Gobs;
```

```
    Error=Error'*C;
```

```
    gx = ((X-xp)*(-3*gm*zp))./(((X-xp).^2+zp^2).^(2.5)); %ax
```

```
    gz = (-1*gm*(2*zp^2-((X-xp).^2))./(zp^2+((X-xp).^2.5)));
```

```
    A = [gx;gz]; %A
```

```
    A = A';
```

```
    At=A';
```

```
    AtA=At*A;
```

```
    iAtAnew = inv(AtA+k*eye(2)); %As in equation in the method section above
```

```

    dp = real((iAtAnew*At)*Error);    %Equation for parameter change, essentially how much to move over a
    point to the calculated correct location
    xp = xp+real(dp(1,1));
    %zp = zp-real(dp(2,1));

```

```

end

```

```

xp = SecondX;
%The inversion, with the same observed model is run on the second initial
%model

```

```

for InversionCount2 = 1:TotI

```

```

    X2=[X2 xp];
    Gcalc=(gm*zp)./(((X-xp).^2+zp^2).^1.5);
    Error = Gcalc-Gobs;
    Error=Error'*C;
    gx = ((X-xp)*(-3*gm*zp))./(((X-xp).^2+zp^2).^2.5);    %ax
    gz = (-1*gm*(2*zp^2-((X-xp).^2)))./(zp^2+((X-xp).^2.5));
    A = [gx;gz];    %A
    A = A';
    At=A';
    AtA=At*A;
    iAtAnew = inv(AtA+k*eye(2));    %As in equation in the method section above
    dp = real((iAtAnew*At)*Error);    %Equation for parameter change, essentially how much to move over a
    point to the calculated correct location
    xp = xp+real(dp(1,1));
    %zp = zp-real(dp(2,1));

```

```

end

```

```

%The distances are computed
%The original distance
OriginalXDisplacement = abs(FirstX-SecondX);
%The distance over the iterations
AllAbsDist=abs(X2-X1);
%Final displacement
FinalXDisplacement = AllAbsDist(TotI);
Iterations = 1:TotI;
%How far the points have strayed if the user is interested
Strayed = FinalXDisplacement/OriginalXDisplacement;

```

```

%Values are logged as per equation (19)
DArray=log(AllAbsDist);

```

```

p=polyfit(Iterations,DArray,1);
%Here equation (19) is put into practice with polyfit, which calculates the
%gradient of the line of best fit
m = p(1);
Fdim=m*(1);
%equation (19)
L=exp(Fdim);
L=round(L,2);

```

```

%Displays and titles
figure(2);
Title1 = ['Lyapunov exponent = ', num2str(L), ' | C = ', num2str(C)];
scatter(Iterations,DArray); xlabel('Iterations');ylabel('Log of separation (m)'); title(Title1); lsline;
FinalXDisplacement=round(FinalXDisplacement,3);
figure(1);
Title3 = ['Two points separated by ', num2str(FinalXDisplacement), ' m after the 10th iteration | ', 'C = ', num2str(C), '
and Lyapunov exponent = ', num2str(L)];
plot(Iterations,X1,'r',Iterations,X2,'b'); xlabel('Iteration');ylabel('x position (m)'); title(Title3);axis
tight;%title({Title1;Title2});

end

```

19 – LyExp

```

function LyExp
%Brandon Dias
%676053
%Revised on 06/05/2017

```

```

%This algorithm runs the inversion process (equation (7)) on two initial
%parameter sets which start off close to one another but tend to different
%minima. The separation at each point is logged and using this the Lyapunov
%exponent, as described in equation (19), is calculated.

```

```

%Initial parameter set up
G = 1; %6.67E-11;
X = -100:100;
%Observed model parameters
Depth=10;
Shift=90/2;
Mass = 2000;
gmz = G*Mass*Depth;
gm = G*Mass;
k=0;
%Observed response
GravityResponse1 = gmz./((X-Shift).^2+Depth^2);
GravityResponse2 = gmz./((X+Shift).^2+Depth^2);
Gobs= GravityResponse1 + GravityResponse2 ;

```

```

%To discern the two initial model parameter sets needed run appendix 6 to
%first create basins of attraction and read off initial parameter values

```

```

%Initial model 1 parameter values
xp=-1;
zp=47;
Mod1x=xp;
Mod1z=zp;
%Track position through iterations
x1=[];
z1=[];
%equation (7)

```

```

for i1=1:15

    x1=[x1 xp];
    z1=[z1 zp];

    Gcalc=gm*zp./((X-xp).^2+zp^2);

    Error = Gcalc-Gobs;          %difference "D"
    Error=Error';

    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

    A = [gx;gz];                %A
    A = A';
    At=A';
    AtA=At*A;
    iAtAnew = inv(AtA+k*eye(2));
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
end
%Initial model parameter values
xp=1;
zp=47;
Mod2x=xp;
Mod2z=zp;
x2=[];
z2=[];
%equation (7)
for i1=1:15

    x2=[x2 xp];
    z2=[z2 zp];

    Gcalc=gm*zp./((X-xp).^2+zp^2);

    Error = Gcalc-Gobs;          %difference "D"
    Error=Error';

    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2; %az

    A = [gx;gz];                %A
    A = A';
    At=A';
    AtA=At*A;
    iAtAnew = inv(AtA+k*eye(2));
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
end

DistanceArray=[];

```

```

%The distance is calculated at each point along the inversion process. This
%is done with the pythagorus equation since the model moves in two
%dimensions
for D = 1:15
    xt1=x1(D);
    xt2=x2(D);
    zt1=z1(D);
    zt2=z2(D);
    Distance = ((xt2-xt1)^2+(zt2-zt1)^2)^0.5 ;
    DistanceArray = [DistanceArray Distance];
end

Iterations=1:15;
DArray=log(DistanceArray);
%In this case, after 6 iterations the points have reached their respective
%minima. This value will have to be change as per the set up. The value can
%be discredned from figure(1)
IShort=Iterations(1:6);
DShort=DArray(1:6);
%Polyfit extracts the gradient of the line of best fit
p=polyfit(DShort,IShort,1);
m = p(1);
Fdim=m*(1);
L=exp(Fdim);
L=round(L,2);
%The model can also be plotted as a scatter plot. Here the x and z
%parameters are concatenated into an array for plotting
XMODELguess=[Mod1x Mod2x];
YMODELguess=[Mod1z Mod2z];
XMODEL=[-1*Shift Shift];
YMODEL=[Depth Depth];
%Displays
figure(1)
subplot(2,1,1);scatter(Iterations,DArray);title('Divergence distance vs. iterations');xlabel('Iterations');ylabel('Log of
separation (m)');
subplot(2,1,2);scatter(IShort,DShort);lsline;title(['Divergence distance vs. iterations. Lyapunov exponent for initial
convergence to local minima = ',num2str(L)]);xlabel('Iterations');ylabel('Log of separation (m)');
display(L);
figure(2)
hold on
plot(x1,z1,'r',x2,z2,'b');axis ij;title(['Initial models parameters - model 1: x = ',num2str(Mod1x),' z =
',num2str(Mod1z),' - model 2: x = ',num2str(Mod2x),' z = ',num2str(Mod2z)]);axis([-100 100 0 60]);xlabel('Range
(m)');ylabel('Depth (m)');
scatter(XMODELguess,YMODELguess,'g','filled');
scatter(XMODEL,YMODEL,'f','filled');
hold off

end

```

20 – ThreeParametersFRAC

```

function ThreeParametersFRAC
tic

```

```
%Revised on 06/01/2019
%Brandon Dias
%676053
```

```
%The inversion process (equations (7),(8) or (13)) can be run to invert for
%three parameters (x,y and z). Here equation (7) acts on a model with one
%observed buried sphere and one initial calculated sphere. The inversion
%runs over a range of x,y and z parameter arrays to create a
%three-dimensional basin of attraction. At each parameter set, the
%inversion is run and a basin is created by indexing a three-dimensional
%matrix with either convergence (1) or remains divergence (0).
```

```
SCALE=1; %Scale for the size of the area of interest
```

```
Gobs=zeros(SCALE*200+1,SCALE*200+1);
```

```
Gcalc=zeros(SCALE*200+1,SCALE*200+1);
```

```
X=-100*SCALE:100*SCALE; %Range
```

```
Y=X; %Maintains a square area of interest
```

```
%Initial parameter values
```

```
G = 1;%6.67E-11;
```

```
Mass = 45;
```

```
gm=G*Mass;
```

```
%Observed model parameter values
```

```
z1=25; %Depth
```

```
x1=0; %x location
```

```
y1=0; %y location
```

```
%Creating a 2 dimensional gravity map matrix.
```

```
i=1;
```

```
for x= -100*SCALE:100*SCALE
```

```
    j=1;
```

```
    for y = -100*SCALE:100*SCALE
```

```
        Gobs(i,j) =gm*z1./((x-x1)^2+(y-y1)^2+z1^2)^1.5;
```

```
        j=j+1;
```

```
    end
```

```
    i=i+1;
```

```
end
```

```
G1=Gobs;
```

```
%
```

```
%
```

```
%This area of the algorithm is uncommented should you wish to first observe
%the observed response and a calculated response. The observed response is
%already calculated and this section adds in an example forward model
%This can be commented out if the user wants to just head straight to the
%fractal basin creation
```

```
Gobs=zeros(SCALE*200+1,SCALE*200+1);
```

```
Gcalc=zeros(SCALE*200+1,SCALE*200+1);
```

```
X=-100*SCALE:100*SCALE; %Range
```

```
Y=X; %Maintains a square area of interest
```

```
%Initial parameter values
```

```
G = 1;%6.67E-11;
```

```
Mass = 45;
```

```
gm=G*Mass;
```

```
z1=input('Depth: '); %Depth
```

```
zt=z1;
```

```

x1=input('x location: '); %x location
xt=x1:
y1=input('y location: '); %y location
yt=y1:
i=1;
for x= -100*SCALE:100*SCALE
    j=1;
    for y = -100*SCALE:100*SCALE
        Gcalc(i,j) =gm*z1./((x-x1)^2+(y-y1)^2+z1^2)^1.5;
        j=j+1;
    end
    i=i+1;
end
G2=Gcalc;
%This plot shows a 3d surface of the observed and calculated responses
figure(1)
subplot(2,1,1);mesh(X,Y,G1);xlabel('x-axis range (m)');ylabel('y-axis range (m)');zlabel('G response (mGal)');
title('Observed gravitational response for a buried sphere');
subplot(2,1,2);mesh(X,Y,G2);xlabel('x-axis range (m)');ylabel('y-axis range (m)');zlabel('G response (mGal)');
title('Observed gravitational response for a buried sphere');

G3=G1+G2; %A response is also plotted of the two responses on a 2D surface
G3=flipdim(G3,1);
figure(3)
image(X,Y,G3);xlabel('x range (m)');ylabel('y range (m)');% title('Gravitational and calculated response for buried
spheres of equal size and density');
% XModel =[x1 xt]
% YModel =[y1 yt]
% The model can also be plotted
% C=linspace(1,2,length(XModel));
% figure(2)
% scatter(XModel,YModel,[100],C,'filled');xlabel('x range (m)');ylabel('y range (m)');axis([-100 100 -100 100]);axis
ij;%title(['Iteration: ',num2str(StepNum)]); grid minor;%title(['Inversion validation | Iteration:
',num2str(StepNum)]);

whos; pause;
%
%
%Fractal basin begins here
%
%
%Working with a three-dimensional solution space and inversion can be
%difficult, therefore, the three-dimensional algebra is made easier by
%concatenating the rows of the three dimensional data set into one long
%column of data. This simplifies the multiplication of arrays in three
%dimensions, and simplifies them to base two dimensional arrays
GoT=[];
LengthSize=size(Gobs);
LengthSize=LengthSize(1);
for iStack = 1:LengthSize
    GoT=[GoT Gobs(iStack,:)];
end
%The empty matrix used to create the basin of attraction
%FRAC=zeros(x,y,z) x,y and z are the result of how far the x,y and z arrays

```

```

%are run. It is traditionally the end of the range multiplied by 2 and +1
%to account for the 0 index
FRAC=zeros(35,35,31);
r=0;%Used to fill matrix
for xpi = -85:5:85 %The range over which the x location is run
    r=r+1;
    c=0;
    disp = (100*(r/35));
    %display([num2str(disp),'%']);
    for ypi = -85:5:85 %The range over which the y location is run
        c=c+1;
        InnerP=(100*(c/35));
        display([num2str(disp),'% in x ||',num2str(InnerP),'% in y']);
        d=0;
        for zp = 0:30 %The range over which the depth is run
            d=d+1;
            xp=xpi;
            yp=ypi;
            for i = 1:15

                Gcalc=zeros(SCALE*200+1,SCALE*200+1);
                gx=zeros(SCALE*200+1,SCALE*200+1);
                gz=zeros(SCALE*200+1,SCALE*200+1);
                gy=zeros(SCALE*200+1,SCALE*200+1);

                i=1;
                for X = -100*SCALE:100*SCALE
                    j=1;
                    for Y = -100*SCALE:100*SCALE
                        Gcalc(i,j) = gm*zp./((X-xp)^2+(Y-yp)^2+zp^2)^1.5;
                        gx(i,j) = (X-xp)*(-3*gm*zp)./(((X-xp)^2+(Y-yp)^2+zp^2)^2.5); %ax
                        gy(i,j) = (Y-yp)*(-3*gm*zp)./(((X-xp)^2+(Y-yp)^2+zp^2)^2.5); %ay
                        gz(i,j) = (-1*gm*(2*zp^2-(Y-yp)^2-(X-xp)^2))./(zp^2+(X-xp)^2+(Y-yp)^2)^2.5; %az
                        j=j+1;
                    end
                    i=i+1;
                end

                GcT=[];
                gzT=[];
                gxT=[];
                gyT=[];

                %The three-dimensional algebra is made easier by
                %concatenating the rows of the three dimensional data set into one long
                %column of data.
                LengthSize=size(Gcalc);
                LengthSize=LengthSize(1);
                for iStack = 1:LengthSize
                    GcT=[GcT Gcalc(iStack,:)];
                    gxT=[gxT gx(iStack,:)];
                    gyT=[gyT gy(iStack,:)];
                    gzT=[gzT gz(iStack,:)];
                end
                %whos;pause;

```

```

Gcalc=GcT;
Gobs=GoT;
gx=gxT;
gy=gyT;
gz=gzT;
%Inversion equation (7)
Error = Gcalc-Gobs;
Error=Error';

A = [gx;gy;gz]; %A
A = A';
At=A';
AtA=At*A;
iAtA = inv(AtA);
dp = real((iAtA*At)*Error);
%whos;pause;
xp = xp+dp(1,1);
yp = yp+dp(2,1);
zp = zp-dp(3,1);
if zp<0
    zp=0;
end
%Higher convergence conditions are used here due to the increased
%non-linearity of the problem
if abs(xp-x1)<3 && abs(yp-y1)<3 && abs(zp-z1)<3
    FRAC(r,c,d)=1;
    %display('WOOHOO!');
    break;
end
end
end
end
end
end
end
%The 3D models are displayed from different angles to fully investigate.
%Flip inverts an axis in a desired direction
figure(2)
isosurface(FRAC,0.5);xlabel('x range (m)');ylabel('y range (m)');zlabel('Depth');colormap hot;
FRAC3=flip(FRAC,3);
figure(5)
isosurface(FRAC3,0.5);xlabel('x range (m)');ylabel('y range (m)');zlabel('Depth');
toc
end

```

21 – Pseudo3D

function Pseudo3D

```

%Revised on 09/09/2018
%Brandon Dias
%676053

```

```

%The inversion processes defined above usually run over a location and
%depth range. Here the inversion process is run over a x and y location
%range. The basin of attraction is again made by tracking which parameter

```

```

%sets in the x and y range converge to observed model parameter values
k=0;
%2D maps for observed and calculated responses
Gobs=zeros(201,201);
Gcalc=zeros(201,201);
%Initial parameters
X=-100:100;
Y=X;
G = 1;%6.67E-11;
Mass = 900;
gm=G*Mass;
%Observed model parameters
z1=40;
x1=0;
y1=0;
%Make 2D data
i=1;
for x=-100:100
    j=1;
    for y = -100:100
        Gobs(i,j) =gm*z1./((x-x1)^2+(y-y1)^2+z1^2)^1.5;
        j=j+1;
    end
    i=i+1;
end
%Working with a three-dimensional solution space and inversion can be
%difficult, therefore, the three-dimensional algebra is made easier by
%concatenating the rows of the three dimensional data set into one long
%column of data. This simplifies the multiplication of arrays in three
%dimensions, and simplifies them to base two dimensional arrays
%Working with a three-dimensional solution space and inversion can be
%difficult, therefore, the three-dimensional algebra is made easier by
%concatenating the rows of the three dimensional data set into one long
%column of data. This simplifies the multiplication of arrays in three
%dimensions, and simplifies them to base two dimensional arrays

GoT=[];
LengthSize=size(Gobs);
LengthSize=LengthSize(1);
for iStack = 1:LengthSize
    GoT=[GoT Gobs(iStack,:)];
end

STEP=1; %The original increment over which the inversion is run
Range=250; %The range over which the inversion runs | Can be input or hard coded
LINE=-1*Range:STEP:Range; %The profile defined
LL=length(LINE);%From the range,*2 for the other side, then +1 for the 0 value
ArraySize=Range*2+1;
L=ArraySize/STEP;
FRAC1=zeros(LL,LL); %Size of solution space
FFinal=[];
c=0;
for Zindex = 57:1:57; %Depths

    FRAC=FRAC1;

```

```

a=0;
c=c+1;
InnerP=(100*(c/2));

for xpi=-1*Range:STEP:Range
    a=a+1;
    disp = (100*(a/L));
    display([num2str(disp),'% in x ||',num2str(InnerP),'% in z']);
    b=0;
    for yp=-1*Range:STEP:Range
        xp=xpi;
        b=b+1;
        zp=Zindex;
        for i = 1:15;

            Gcalc=zeros(201,201);
            gx=zeros(201,201);
            %gz=zeros(201,201);
            gy=zeros(201,201);
            %Calculated response in 2D
            i=1;
            for X=-100:100
                j=1;
                for Y = -100:100
                    Gcalc(i,j) = gm*zp./((X-xp)^2+(Y-yp)^2+zp^2)^1.5;
                    gx(i,j) = (X-xp)*(-3*gm*zp)./(((X-xp)^2+(Y-yp)^2+zp^2)^2.5);    %ax
                    gy(i,j) = (Y-yp)*(-3*gm*zp)./(((X-xp)^2+(Y-yp)^2+zp^2)^2.5);    %ay
                    % gz(i,j) = (-1*gm*(2*zp^2-(Y-yp)^2-(X-xp)^2))/(zp^2+(X-xp)^2+(Y-yp)^2)^2.5;    %az
                    j=j+1;
                end
                i=i+1;
            end

            GcT=[];
            gxT=[];
            gyT=[];
            %3D -> 2D
            LengthSize=size(Gcalc);
            LengthSize=LengthSize(1);
            for iStack = 1:LengthSize
                GcT=[GcT Gcalc(iStack,:)];
                gxT=[gxT gx(iStack,:)];
                gyT=[gyT gy(iStack,:)];
            end
            %whos;pause;
            Gcalc=GcT;
            Gobs=GoT;
            gx=gxT;
            gy=gyT;

            %Inversion equation (7)
            Error = Gcalc-Gobs;
            Error=Error';

            A = [gx;gy];    %A

```

```

A = A';
At=A';
AtA=At*A;
%iAtA = inv(AtA);
iAtA = inv(AtA+k*eye(2));
dp = real((iAtA*At)*Error);

xp = xp+dp(1,1);
yp = yp+dp(2,1);
%Conditions for convergence done have to be as strict as in
%the 3D type
if abs(xp-x1)<1 && abs(yp-y1)<1
    FRAC(b,a)=1;
    break;
end
end
end
end

A = FRAC;
x=-1*Range:Range;
y=x;
FFinal=cat(3,FFinal,FRAC); %The slices can be concatenated. Alternatively the depth range on the inversion can
be keot to one depth and then only one slice would be produced

figure(1)
imagesc(x,y,FRAC);xlabel('x range (m)');ylabel('y range (m)');title(T);
end
%The concatenated image
figure(2)
isosurface(FFinal,0.5);xlabel('x range (m)');ylabel('y range (m)');zlabel('Depth (m)');colormap hot;
FRAC3=flip(FFinal,3);
%The concatenated image as viewed from underneath; from a giher z values to
%lower z values
figure(3)
isosurface(FRAC3,0.5);xlabel('x range (m)');ylabel('y range (m)');zlabel('Depth (m)');
end

```

22 – mapREAL

```

function mapREAL
%Brandon Dias
%Revised on 17/10/2018
%676053

%Extract in real data. Data is flipped such that North is at the upper edge
%of the image. The data has its minimum value subtracted to create a
%relative map.
FileName1=input('Filename: ');
Data1=xlsread(FileName1);%Read in data
D2 = flipdim(Data1 ,1);%Flip data such that North points up
MINV=min(min(D2));%Subtract minimum to create a relative map

```

```

D2=D2-MINV;
%Create image
imagesc(D2); xlabel('x range (km)');ylabel('y range (km)'); colorbar;
end

```

23 – RealDataMisfitJumpFRAC

```
function RealDataMisfitJumpFRAC
```

```
%Brandon Dias
```

```
%676053
```

```
%Revised on 18/10/2018
```

```

%This algorithm serves to first extract a 2D grav map and from that select
%a latitudinal or longitudinal profile. A loop is entered which repeats
%forward modelling checks until one is satisfied. Once a forward model is
%established the code enters a series of inversion checks at a range of
%displacements and depths. The conditions for convergence are checked at eah
%iteration in the two ranges and marked as divergence or convergence. This
%allows the algorithm to again construct a fractal image. A misfit surface
%is showed, following this the user inputs a guess x value and inversion is
%run for that value. The value can also be taken from the basin of
%attraction. If the x value is near the local minima it will succumb to
%that and the scaling factor will begin increasing to exit the local
%minimum

```

```
TempCount=0;
```

```
%Extract the data profile
```

```
%Extract the 2D file
```

```
%FileName1=input('Filename: '); %Can be input or hardcode if the name of
```

```
%the data file is not going to change
```

```
FileName1='R.xlsx';
```

```
Data1=xlsread(FileName1);
```

```
MinV=min(min(Data1));
```

```
Data1=Data1-MinV;%Minimum value subtracted to create relative map
```

```
%Surface plot is made for user information
```

```
figure(2)
```

```
mesh(Data1);xlabel('Range (m)');ylabel('Range (m)');zlabel('Relative gravitational intensity (mGal)');title('Trompsburg relative gravity map'); colormap;
```

```
%Select latitude or longitude
```

```
%LatLon=input('Latitude or longitude? Press either 1 or 2 respectively - ');
```

```
LatLon=1; %Can be selected by the user or hard coded if a certain profile/area is of interest
```

```
if LatLon == 1
```

```
    %LAT = input('Latitude: ');
```

```
    LAT=163;%Can be selected by the user or hard coded if a certain profile/area is of interest
```

```
    Line=Data1(LAT,:);
```

```
    LineTitle=['Longitude - ',num2str(LAT)];
```

```
elseif LatLon == 2
```

```
    %Lon = input('Longitude: ');
```

```
    Lon=182;%Can be selected by the user or hard coded if a certain profile/area is of interest
```

```

    Line=Data1(:,Lon);
    Line=Line';
    LineTitle=['Latitude - ',num2str(Lon)];
else
    display('Invalid Selection');
end
%By subtracting the smallest value in the profile we bring the data down to
%a datum and allow ease of inversion with easier values to use for the
%forward modelling

MinL=min(Line);
Line=Line-MinL;
LengthLine=length(Line);
X=1:LengthLine;
figure(1);
plot(X,Line); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile | ',LineTitle]);

%Profile redefining
%The profile can first be seen and if a peak is of interest the line can be
%reduced. First we provide the length of the line though.
display(['Length of original profile runs from 1 to ',num2str(LengthLine)]);
%Restricting the profile to an area of interest can be read by the user off
%the plot or hard coded
%SL=input('Define start of profile: ');
%EL=input('Define end of profile: ');
SL=60;
EL=260;
Line=Line(SL:EL);

%New profile created
LengthLine=length(Line);
X=1:LengthLine;
DepthLine=floor(LengthLine/4);
Z=1:DepthLine;
figure(1);
plot(X,Line); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in red and forward model profile
in blue | ',LineTitle,' | Profile sampled from ',num2str(SL),' to ',num2str(EL)]);

%Forward Modelling begins
%Constants defined
Gobs=Line;
G = 1;%6.67E-11; %G is defined as 1 to ease computational effort and makes input masses smaller

%Satisfied=2;
%Forward modelling
%Can be for cylinder or sphere, comment out undesired model
%This loop runs until the user is happy with the chosen parameters and a
%valid match is created. The last used variables are then used in the
%inversion
Satisfied=1; %Satisfied can be assigned 1 if the calculated parameters have already been decided upon
while Satisfied == 2
    xp=input('x value: ');
    zp=input('z value: ');
    Mass=input('Mass value: ');
    gm=G*Mass;

```

```

%Comment out what is not needed
%GForward=gm*zp./((X-xp).^2+zp^2);%Cylinder
GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
figure(1)
plot(X,Gobs,'r',X,GForward,'b'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in red and
forward model profile in blue | ',LineTitle]);
Satisfied=input('If satisfied enter 1, otherwise enter 2 - ');
end
%In order to display this alongside new forward models for second bodies we
%save the already established profile
%Defined by forward modelling loop or hard coded
xp=126;
zp=15;
Mass=19800;
gm=G*Mass;
GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
GBody1=GForward;
gm=G*Mass;
figure(1)
plot(X,Gobs,'r',X,GForward,'b'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in red and
forward model profile in blue | ',LineTitle]);

%The forward modelling variables are declared as the optimum parameters
x1=xp;
z1=zp;

%Satisfied=2;
Satisfied=1;
%Second Body
%Can be for cylinder or sphere, comment out undesired model
%This loop runs until the user is happy with the chosen parameters and a
%valid match is created. The last used variables are then used in the
%inversion
while Satisfied == 2%Satisfied can be assigned 1 if the calculated parameters have already been decided upon
    xp=input('x value: ');
    zp=input('z value: ');
    Mass=input('Mass value: ');
    %GForward=gm*zp./((X-xp).^2+zp^2);%Cylinder
    gm=G*Mass;
    GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
    figure(1)
    plot(X,Gobs,'r',X,GForward,'b',X,GBody1,'g'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed
profile in red, first body in green and current forward model profile in blue | ',LineTitle]);
    Satisfied=input('If satisfied enter 1, otherwise enter 2 - ');
end
%Defined by forward modelling loop or hard coded
xp=35;
Mass=8550;
gm2=G*Mass;
GForward=gm2*zp./(((X-xp).^2+zp^2).^1.5);
GBody2=GForward;
%The forward modelling variables are declared as the optimum parameters
x2=xp;
z2=zp;

```

```

figure(4)
plot(X,Gobs,'r',X,GBody2,'b',X,GBody1,'g'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in
red, first body in green and second model profile in blue | ',LineTitle]);
%pause;
zp=z1;
Ma=[];
%Misfit surface is created
for xp = 1:LengthLine
    Gcalc = gm*zp./(((X-xp).^2+zp^2).^1.5);
    Error = Gcalc-Gobs;

    N=length(Gcalc);
    Mis=sum(Error.^2);
    MisfitT=Mis/N;
    Ma=[Ma MisfitT];
end
%Misfit surface can be normalized to ease misfit jump process should
%validation be needed
MinV2=min(Ma);
Ma=Ma/MinV2;
Xt=1:LengthLine;
figure(3)
plot(Xt,Ma);title('Misfit surface');xlabel('Position along profile (m)');ylabel('Misfit');axis([0 200 0 4]);
%pause;
%Inversion begins
%The inversion process is run over intervals of displacement and depth and
%at each instance the inversion is checked to see if the parameters invert
%to those decided upon in the forward modelling section.

%FRACarray is initially empty and is only altered should convergence occur
SCALER=1;%Size of solution space
Scaler2=inv(SCALER);%Incriment over which inversion is run
FRACarray=zeros(SCALER*DepthLine,SCALER*LengthLine);
a=0;
for xpi=1:Scaler2:LengthLine
    a=a+1;
    P=xpi/LengthLine;
    P=P*100;
    display(P);
    b=0;
    for zp=1:Scaler2:DepthLine
        xp=xpi;
        b=b+1;
        for i = 1:20
            Gcalc=gm*zp./(((X-xp).^2+zp^2).^1.5);
            %whos;pause;
            Error = Gcalc-Gobs;
            Error=Error';

            %gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
            gx = (X-xp)*(-3*gm*zp)./(((X-xp).^2+zp^2).^2.5);    %ax
            %gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;    %az
            gz = (gm*(-2*zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2.5;    %az
            A = [gx,gz];    %A
            A = A';
        end
    end
end

```

```

    At=A';
    AtA=At*A;
    iAtA = inv(AtA);
    dp = real((iAtA*At)*Error);
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
    if abs(xp-x1)<5 && abs(zp-z1)<5
        FRACarray(b,a)=1;
        break;
    end
    if abs(xp-x2)<5 && abs(zp-z2)<5
        FRACarray(b,a)=2;
        break;
    end
end
end
end

xRange=0:200;
yDepth=0:100;
figure(1)
image(xRange,yDepth,FRACarray);xlabel('Range (m)');ylabel('Depth (m)');
%Image will need to be scaled or use "imagesc"

end

```

24 – RealDataMisfitJump

```
function RealDataMisfitJump
```

```
%Brandon Dias
```

```
%676053
```

```
%Revised on 17/10/2018
```

```

%This algorithm serves to first extract a 2D grav map and from that select
%a latitudinal or longitudinal profile. A loop is entered which repeats
%forward modelling checks until one is satisfied. Once a forward model is
%established the code enters a series of inversion checks at a range of
%displacements and depths. The conditons for convergence are checked at eah
%iteration in the two ranges and marked as divergence or convergence. This
%allows the algorithm to again construct a fractal image. A misfit surface
%is showed, following this the user inputs a guess x value and inversion is
%run for that value. The value can also be taken from the basin of
%attraction. If the x value is near the local minima it will succumb to
%that and the scaling factor will begin increasing to exit the local
%minimum

```

```
TempCount=0;
```

```
%Extract the data profile
```

```
%Extract the 2D file
```

```
%FileName1=input('Filename: '); %Can be input or hardcode if the name of
```

```
%the data file is not going to change
```

```

FileName1='R.xlsx';
Data1=xlsread(FileName1);
MinV=min(min(Data1));
Data1=Data1-MinV;%Minimum value subtracted to create relative map
%Surface plot is made for user information
figure(2)
mesh(Data1);xlabel('Range (m)');ylabel('Range (m)');zlabel('Relative gravitational intensity
(mGal)');title('Trompsburg relative gravity map'); colormap;
%Select latitude or longitude

%LatLon=input('Latitude or longitude? Press either 1 or 2 respectively - ');

LatLon=1; %Can be selected by the user or hard coded if a certain profile/area is of interest
if LatLon == 1
    %LAT = input('Latitude: ');
    LAT=163;%Can be selected by the user or hard coded if a certain profile/area is of interest
    Line=Data1(LAT,:);
    LineTitle=['Longitude - ',num2str(LAT)];
elseif LatLon == 2
    %Lon = input('Longitude: ');
    Lon=182;%Can be selected by the user or hard coded if a certain profile/area is of interest
    Line=Data1(:,Lon);
    Line=Line';
    LineTitle=['Latitude - ',num2str(Lon)];
else
    display('Invalid Selection');
end
%By subtracting the smallest value in the profile we bring the data down to
%a datum and allow ease of inversion with easier values to use for the
%forward modelling

MinL=min(Line);
Line=Line-MinL;
LengthLine=length(Line);
X=1:LengthLine;
figure(1);
plot(X,Line); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile | ',LineTitle]);
%pause;
%Profile redefining
%The profile can first be seen and if a peak is of interest the line can be
%reduced. First we provide the length of the line though.
display(['Length of original profile runs from 1 to ',num2str(LengthLine)]);
%Restricting the profile to an area of interest can be read by the user off
%the plot or hard coded
%SL=input('Define start of profile: ');
%EL=input('Define end of profile: ');
SL=60;
EL=260;
Line=Line(SL:EL);

%New profile created
LengthLine=length(Line);
X=1:LengthLine;
DepthLine=floor(LengthLine/4);
Z=1:DepthLine;

```

```

figure(1);
plot(X,Line); xlabel('Range (km)');ylabel('Relative response (mGal)');title(['Observed profile | ',LineTitle]);axis([0
200 0 100]);
pause;
%Forward Modelling begins
%Constants defined
Gobs=Line;
G = 1;%6.67E-11; %G is defined as 1 to ease computational effort and makes input masses smaller

%Satisfied=2;
%Forward modelling
%Can be for cylinder or sphere, comment out undesired model
%This loop runs until the user is happy with the chosen parameters and a
%valid match is created. The last used variables are then used in the
%inversion
Satisfied=1; %Satisfied can be assigned 1 if the calculated parameters have already been decided upon
while Satisfied == 2
    xp=input('x value: ');
    zp=input('z value: ');
    Mass=input('Mass value: ');
    gm=G*Mass;
    %Comment out what is not needed
    %GForward=gm*zp./((X-xp).^2+zp^2);%Cylinder
    GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
    figure(1)
    plot(X,Gobs,'r',X,GForward,'b'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in red and
forward model profile in blue | ',LineTitle]);
    Satisfied=input('If satisfied enter 1, otherwise enter 2 - ');
end
%In order to display this alongside new forward models for second bodies we
%save the already established profile
%Defined by forward modelling loop or hard coded
xp=126;
zp=15;
Mass=19800;
gm=G*Mass;
GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
GBody1=GForward;
gm=G*Mass;
figure(1)
plot(X,Gobs,'r',X,GForward,'b'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in red and
forward model profile in blue | ',LineTitle]);

%The forward modelling variables are declared as the optimum parameters
x1=xp;
z1=zp;

%Satisfied=2;
Satisfied=1;
%Second Body
%Can be for cylinder or sphere, comment out undesired model
%This loop runs until the user is happy with the chosen parameters and a
%valid match is created. The last used variables are then used in the
%inversion
while Satisfied == 2%Satisfied can be assigned 1 if the calculated parameters have already been decided upon

```

```

xp=input('x value: ');
%zp=input('z value: ');
Mass=input('Mass value: ');
%GForward=gm*zp./((X-xp).^2+zp^2);%Cylinder
gm=G*Mass;
GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
figure(1)
plot(X,Gobs,'r',X,GForward,'b',X,GBody1,'g'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed
profile in red, first body in green and current forward model profile in blue | ',LineTitle]);
Satisfied=input('If satisfied enter 1, otherwise enter 2 - ');
end
%Defined by forward modelling loop or hard coded
xp=35;
Mass=8550;
gm=G*Mass;
GForward=gm*zp./(((X-xp).^2+zp^2).^1.5);
GBody2=GForward;
%The forward modelling variables are declared as the optimum parameters
x2=xp;
z2=zp;

figure(4)
plot(X,Gobs,'r',X,GBody2,'b',X,GBody1,'g'); xlabel('Range (m)');ylabel('Profile strength');title(['Observed profile in
red, first body in green and second model profile in blue | ',LineTitle]);
%pause;
zp=z1;
Ma=[];
%Misfit surface is created
for xp = 1:LengthLine
    Gcalc = gm*zp./(((X-xp).^2+zp^2).^1.5);
    Error = Gcalc-Gobs;

    N=length(Gcalc);
    Mis=sum(Error.^2);
    MisfitT=Mis/N;
    Ma=[Ma MisfitT];
end
%Misfit surface can be normalized to ease misfit jump process should
%validation be needed
MinV2=min(Ma);
Ma=Ma/MinV2;
Xt=1:LengthLine;
figure(3)
plot(Xt,Ma);title('Misfit surface');xlabel('Position along profile (m)');ylabel('Misfit');axis([0 200 0 4]);

xRange=0:200;
yDepth=0:100;

%%MISFIT JUMP

C=1;
%User defines initial calucated model parameter values
xp=input('Input x: ');
XPsave=xp; %This value is saved so that creating the basin of attraction doesnt alter these values
%zp=input('Guessing z: ');

```

```

                                %Guess profile
Misfits=[];
Xps=[];
%Zps=[];
Xps=[Xps xp];
%Zps=[Zps zp];
X=1:LengthLine;
%Initial misfit calculated
Gcalc=gm*zp./(((X-xp).^2+zp^2).^1.5);
Error = Gcalc-Gobs;
N=length(Gcalc);
Mis=sum(Error.^2);
Mis=Mis/N;
Misfits=[Misfits Mis];
Scalings=[];
Scalings=[Scalings C];
%Initial depth defined or can be defined by input should user prefer
zp=15;%input('z: ');
ZPsave=zp;
for i = 1:50

    Error=Error*C;
    gx = (X-xp)*(-3*gm*zp)/(((X-xp).^2+zp^2).^2.5);    %ax

    gz = (gm*(-2*zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2.5;    %az
    A = [gx;gz];    %A
    A = A';
    At=A';
    AtA=At*A;
    iAtA = inv(AtA);
    dp = real((iAtA*At)*Error);
    xp = xp+dp(1,1);
    %This area can be commented out if only the x parameter value is
    %investigated
    %    zp = zp-dp(2,1);
    %    if zp<0
    %        zp=1;
    %    end
    Xps=[Xps xp];
    %    Zps=[Zps zp];

    % C=1;

    %Gcalc=gm*zp./((X-xp).^2+zp^2);
    Gcalc=gm*zp./(((X-xp).^2+zp^2).^1.5);
    Error = Gcalc-Gobs;
    %Defined new misfit
    N=length(Gcalc);
    Mis2=sum(Error.^2);
    AfterMis=Mis2/N;
    Misfits=[Misfits AfterMis];
    Scalings=[Scalings C];
    %Defined a case of divergence

```

```

if isnan(AfterMis) == 1
    display('Divergence');
    break;
end

%Observe if the x parameter is stuck within a minimum
Diff=abs(Xps(end)-Xps(end-1));

if Diff < 4; %Threshold minimum
    %pause;
    TempCount = TempCount +1;
end
%If the parameter is within the minimum for 5 iterations the scaling factor portion begins
if TempCount == 5
    display('Starting C up');
    % pause;
    Potentialxp=xp;
    PotentialMisfit=AfterMis;
    Potentialzp=zp;
    TempCount=0;
    C=2;
    for CCount = 1:180 %Magnitude of C which the parameter can grow until

        C=C+0.1;
        %display(C);
        %Gcalc=gm*zp./((X-xp).^2+zp^2);
        Gcalc=gm*zp./(((X-xp).^2+zp^2).^1.5);
        Error = Gcalc-Gobs;
        Error=Error*C;

        gx = (X-xp)*(-3*gm*zp)./(((X-xp).^2+zp^2).^2.5);    %ax
        gz = (gm*(-2*zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2.5;    %az
        A = [gx;gz];    %A
        A = A';
        At=A';
        AtA=At*A;
        iAtA = inv(AtA);
        dp = real((iAtA*At)*Error);
        xp = xp+dp(1,1);
    %    zp = zp-dp(2,1);
    %    if zp<0
    %        zp=1;
    %    end

    Xps=[Xps xp];
    %    Zps=[Zps zp];
    %
    %    Xps=[Xps xp];
    %
    Gcalc=gm*zp./(((X-xp).^2+zp^2).^1.5);
    Error = Gcalc-Gobs;
    %
    N=length(Gcalc);
    Mis=sum(Error.^2);
    TMis=Mis/N;

```

```

Misfits=[Misfits TMis];
Scalings=[Scalings C];
if TMis < (PotentialMisfit-1)%User defined limit which the misfit is allowed to vary within
    C=1;
    %When the misfit is lower than any misfit identified before,
    %the corresponding parameter values are accepted as the
    %parameter values for the global minimum
    Potentialxp=xp;
    PotentialMisfit=TMis;
    Potentialzp=zp;
    break;
end
if isnan(TMis)==1
    break;
end

end
C=1;

%
end

end

%The scaling factors, x location, responses, misfits and final parameter
%values are presented

PotentialMisfit=PotentialMisfit/MinV2;

Min2=min(Misfits);
Misfits=Misfits/MinV2; %This reduces the misfit to be normalized about the global minimum
L1=length(Misfits);
L1=0:L1-1;

figure(5);
subplot(1,2,1);plot(Xps,L1);title('x position as a function of iterations');xlabel('x position
(m)');ylabel('Iterations');axis([0 200 0 300]);
subplot(1,2,2);plot(Misfits,L1);title('Misfit as a function of iterations');xlabel('Misfit
(mGal^2)');ylabel('Iterations');axis([0 4 0 300]);
figure(6);
plot(L1,Scalings);title('Scaling factors');
X=1:LengthLine;

figure(7)
subplot(3,1,1);plot(X,Gobs);title('Profiles');xlabel('Position along profile (km)');ylabel('Gravitational intensity
(mGal)');axis tight;%axis([-80 80 0 1]);
subplot(3,1,2);plot(X,Ma);title('Misfit surface');xlabel('Position along profile (km)');ylabel('Misfit
(mGal^2)');axis([0 200 0 4]);%axis tight;%axis([-80 80 0 8]);%axis([-50 150 3 15]);
subplot(3,1,3);plot(Xps,L1);title('Displacements as a function of iterations');xlabel('Displacement
(km)');ylabel('Iterations');axis([0 200 0 300]);
%Display the accepted parameter values for the global minimum
Potentials=['xp: ',num2str(Potentialxp),' zp: ',num2str(Potentialzp),' Misfit ',num2str(PotentialMisfit)];
display(Potentials);

```

```

figure(8)
subplot(2,1,1);plot(X,Ma);title('Misfit ');xlabel('Range (km)');ylabel('Misfit (mGal^2)');axis([0 200 0 4]);%axis
tight;%axis([-80 80 0 8]);%axis([-50 150 3 15]);
subplot(2,1,2);plot(Xps,L1);title('x position as a function of iterations');xlabel('x position
(km)');ylabel('Iterations');axis([0 200 0 300]);

end

```

25 – MultiBasinKorC

```

function MultiBasinKorC
%Brandon Dias
%676053
%Revised on 17/12/2018

%This algorithm runs over an x and z range and at each point runs the
%inversion process as described in equation (18). Whether or not convergence to the
%defined local minima is met is then marked by the basin of attraction
%. The inversion is run with or without damping or scaling
%6 Basin of attraction plots are created in one image
%Define initial scaling or damping constants
kV=1;
%kV=0 means no damping influence
CV=0.90;
%CV=1 means no influence of the scaling factor
for IMAGEi = 1:6 %Image index

    XR=-100:100;%Range over the geological model (for plotting images)
    ZR=0:100; %Range of depths for the geological model
    G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
    X = -100:100; %Range (for calculations)
    Shift = 76/2;%Separation between sources or location along the rangeInput or hard coded as desired
    Mass = 20;
    gm = G*Mass;
    %for Shift = 35:10:45 %A for loop can be run here should you want to generate many images, this loop can be
run for depth, separation or even scale
    %-----
    Depth=10; %Depth at which cylinder is buried | Can be input or hard coded as desired
    gmz = G*Mass*Depth;
    Gobs = gmz./((X-Shift).^2+Depth^2)+gmz./((X+Shift).^2+Depth^2); %Synthetic gravitational response for two
buried cylinder models
    figure(11)
    plot(X,Gobs);xlabel('Range (m)');ylabel('Response (mGal)');
    pause;
    I=0; %Index for basin of convergence
    SCALE=1;%The constant by which the solution space will be enlarged. Larger numbers mean more densely
spaced parameter values and therefore an increase in resolution
    FFF=zeros(SCALE*100,SCALE*200); %Empty array used to create the basin of attraction pattern
    for ICount = 1:1/SCALE:200 %Range along x | 1/SCALE accounts for the increased solution space exactly
        P=ICount/201; %P is displayed on the command window so the user can observe if the program is still
running, stuck and how far along it is
        display([num2str(P*100),' %']); %Display P
    end
end

```

```

I=I+1;
J=0;
for JCount = 1:1/SCALE:100

    J=J+1;
    k=kV;      %Starting constant for the damping factor
    xp = ICount-100; %Start at first index in the ranges
    zp = JCount;
    for InversionCount = 1:10 %Arbitrary maximum amount of iterations
    %The inversion runs as described in the write up
    %First the discrepancy matrix, D, is created
        Gcalc=gm*zp./((X-xp).^2+zp^2);

        Error = Gcalc-Gobs;          %difference "D"
        Error=Error*CV;

    %Here the A matrix is created by concatenating ax and az
        gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
        gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2).^2;  %az

        A = [gx;gz];          %A
        A = A';
        At=A';
        AtA=At*A;

    %The equation stands with the addition of the k constant
        iAtAnew = inv(AtA+k*eye(2));
    %The parameter change matrix is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
    respectively
        dp = (iAtAnew*At)*Error;
        xp = xp+dp(1,1);
        zp = zp-dp(2,1);

    %Should convergence have been met, the parameter value indices are marked
    %and the basin of attraction is created.
    %Here the basin is tracked as a function of the amount of iterations to
    %converge. "break" exits the iterations ensuring the value is not over
    %written
        if abs(xp-Shift)<1 && abs(zp-Depth)<1 %Left model
            FFF(J,I)=InversionCount; %By assigning a value equal to the amount of iterations in the solution, the
            basin of attraction now tracks the amount of iterations to converge, as intensity in colour
            %FFF(J,I)=1;
            break;
        end
        if abs(xp+Shift)<1 && abs(zp-Depth)<1 %Right model
            FFF(J,I)=InversionCount; %By assigning a value equal to the amount of iterations in the solution, the
            basin of attraction now tracks the amount of iterations to converge, as intensity in colour
            %FFF(J,I)=1;
            break;
        end
    end
end
end
end
%Image display of basin of attraction in the solution space
figure(1)
image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap hot;
%Colormap is set by preference

```

```

figure(2)
subplot(3,2,IMAGEi);image(XR,ZR,FFF);xlabel('Range (m)');ylabel('Depth (m)');colormap hot;title(['C = ',num2str(CV)]);
%
%Vary the scaling factor or damping factor for the subsequent image
%kV=kV+4;
CV=CV+0.15;
end

end

```

26 – CompareK

```
function CompareK
```

```

%Brandon Dias
%676053

```

```

%This algorithm runs the Ridge-regression inversion process over the same
%observed and initial model at varied damping constant. The depth (or
%position on the x-range) is then tracked through the process and plotted
%as a function of the iterations. All these plots are subsequently plotted
%on the same plot.

```

```

G = 1; %6.67E-11; % G and Mass is almost entirely analytical so can be any value
X = -100:100; %Range (for calculations)
Shift = 10;%Location along the rangeInput or hard coded
Mass = 150;
gm = G*Mass;
%-----
Depth=10; %Depth at which cylinder is buried
gmz = G*Mass*Depth;
Gobs = gmz./((X-Shift).^2+Depth^2); %Synthetic gravitational response for a buried cylinder model

```

```

JJ=0;%For where to plot the plot in the subplot
for k = 0:5:25
    xp=10;%Input model location
    zp=20;%Input model depth
    JJ=JJ+1;
    XPOS=[];%Recording the location along the range
    ZPOS=[];%Recording the depth
    III=[];%Recording the corresponding iteration
    for InversionCount = 1:16 %Arbitrary maximum amount of iterations
        %The inversion runs as described in the write up

        III=[III InversionCount];
        XPOS=[XPOS xp];
        ZPOS=[ZPOS zp];
        %Input model response
        Gcalc=gm*zp./((X-xp).^2+zp^2);
        %First the discrepancy matrix, D, is created
        Error = Gcalc-Gobs; %difference "D"
    end
end

```

```

    Error=Error';
%Here the A matrix is created by concatenating ax and az
    gx = (X-xp)*(-2*gm*zp)./(((X-xp).^2+zp^2).^2);    %ax
    gz = (gm*(-zp^2+(X-xp).^2))./(zp^2+(X-xp).^2);    %az

    A = [gx;gz];    %A
    A = A';
    At=A';
    AtA=At*A;
%The equation stands with the addition of the k constant
    iAtAnew = inv(AtA+k*eye(2));
%The parameter change matrix is created with regards to the x and z parameter held in dp(1,1) and dp(2,1),
respectively
    dp = (iAtAnew*At)*Error;
    xp = xp+dp(1,1);
    zp = zp-dp(2,1);
end
%Creating the plots
figure(2)
subplot(3,2,JJ);plot(III,ZPOS,'b');title(['k = ',num2str(k)]);axis([1 16 0 20]);xlabel('Iterations');axis
ij;ylabel('Depth (m)');
end

end

```