



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIAS DE LA TIERRA
INSTITUTO DE GEOFÍSICA

**APLICACIÓN DEL MODELADO GRAVIMÉTRICO 3D AL
ESTUDIO DE PLAYS SUBSALINOS**

TESIS

QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN CIENCIAS DE LA TIERRA

PRESENTA:
MAURICIO NAVA FLORES

DIRECTOR DE TESIS:
DR. JOSÉ CARLOS ORTIZ ALEMÁN

INSTITUTO MEXICANO DEL PETRÓLEO

Ciudad Universitaria, Cd. Mx., Agosto de 2018



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

“Yo no sé muchas cosas, es verdad. Digo sólo lo que he visto ...”

León Felipe

Dedicada a mis padres y hermanos

Agradecimientos

A la Universidad Nacional Autónoma de México, mi alma mater.

Al departamento de Geofísica de la Facultad de Ingeniería de la UNAM, especialmente a David Escobedo y Martín Cárdenas por su apoyo y amistad durante el tiempo que coincidimos en ese inusual sitio.

Al Instituto de Geofísica y al Programa de Posgrado en Ciencias de la Tierra, que me permitieron iniciar y concluir mis estudios de doctorado, con especial énfasis en su personal administrativo.

Al Dr. Carlos Ortiz Alemán por ser el director de este trabajo, además de un gran amigo y colega. A mis sinodales: Dr. Oscar Campos Enríquez, Dr. William Bandy, Dr. Carlos Mortera y Dr. Juan García Abdeslem por sus valiosos comentarios y críticas que ayudaron a mejorar la calidad de la misma.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

*Resumen***Aplicación del Modelado Gravimétrico 3D al Estudio de Plays Subsalinos**por **Mauricio Nava Flores**

Se presentan las bases teóricas y aplicaciones del procesamiento e interpretación de mallas de anomalía y gradiometría gravimétrica, así como del modelado directo e inverso 3D basado en ensambles de prismas, desde una perspectiva integral que ha sido exitosa en la exploración de hidrocarburos en escenarios geológicos complejos, caracterizados por la presencia de cuerpos salinos alóctonos emplazados en sedimentos terrígenos.

La metodología propuesta contempla el uso de resultados de diferentes métodos de realce, procesamiento e interpretación de datos gravimétricos, para crear un modelo de distribución de densidades del subsuelo que es utilizado como modelo inicial en un algoritmo de inversión gravimétrica 3D.

Para realizar la inversión gravimétrica 3D se aplicó un algoritmo tipo Simulated Annealing, numéricamente optimizado que incluye un factor de amplificación de sensibilidad con la profundidad, lo que contribuye a contrarrestar el cálculo excesivo de soluciones cercanas a los puntos de observación. Este algoritmo fue analizado con un barrido paramétrico, a partir del cual se generaron criterios que permitieron seleccionar valores iniciales adecuados de los parámetros que controlan su rendimiento y ejecución.

Todos los métodos de procesamiento e interpretación de mallas de anomalía y gradiometría gravimétrica utilizados en este trabajo, así como el método de inversión, fueron probados y validados con datos libres de ruido y con ruido pseudoaleatorio añadido, generados a partir de modelos sintéticos.

Finalmente, se presentan dos casos de aplicación: **Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México**, aplicación que ya ha sido publicada y **Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.**, cuyos resultados resaltan la importancia de la integración de métodos de procesamiento, interpretación semiautomática e inversión de datos gravimétricos en la generación de modelos del subsuelo de alta resolución de estructuras geológicas complejas.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Abstract

3D Gravity Modeling Applied to Subsalt Imaging

by **Mauricio Nava Flores**

Theoretical basis and applications of processing and interpretation of gravity anomaly and gravity gradient grids, and direct and inverse 3D modeling based on ensembles of prisms are presented from an integral perspective that has been successfully applied in oil exploration carried out in geologic complex scenarios characterized by the presence of allochthonous salt bodies embedded in terrigenous sediments.

The proposed methodology consists in using the results from different methods of enhancement, processing and interpretation of gravity data, to create a model of distribution of subsurface densities that is used as initial model in a 3D gravity inversion algorithm.

To perform the 3D gravity inversion, a numerically optimized Simulated Annealing algorithm was applied, which includes a depth sensitivity amplification factor that serves to counteract the excessive obtaining of solutions near the observation points. This algorithm was analyzed by a parametric scan, resulting in criteria suggested to select appropriate initial values of the parameters that control its performance and execution.

All the methods of processing and interpretation of gravity anomaly and gravity gradient grids used in this work, as well as the inversion method, were tested and validated with noise free synthetic data as well as data contaminated by pseudo-random gaussian noise.

Finally, are presented two application cases: **3D Gravity Modeling of Complex Salt Features in the Southern Gulf of Mexico**, that has already been published and **3D Modeling of Gravity Gradient Data applied to the Vinton Dome, LA, USA**, whose results highlight the significance of integrating methods of processing, semiautomatic interpretation and inversion of gravity data in the construction of high resolution models for complex geological structures.

ÍNDICE GENERAL

Agradecimientos	v
Resumen	vii
Abstract	ix
Índice General	xi
Lista de Figuras	xiii
Lista de Tablas	xxiii
1 Introducción	1
1.1 Objetivos	3
1.2 Metodología	4
1.3 Información Geofísica	5
1.4 Contribuciones	5
2 Fundamentos Teóricos	7
2.1 Modelado Gravimétrico Directo 3D	7
2.1.1 Potencial, vector y tensor gravitacional	7
2.1.2 Modelado de cuerpos simples	13
2.1.3 Modelado de estructuras complejas	23
Ensamblajes de bloques elementales	23
2.2 Procesamiento e Interpretación de Datos en el Modelado	26
2.2.1 Métodos de realce de anomalías y bordes	26
Realce de anomalías y bordes en mallas de anomalía gravimétrica	27
Realce de anomalías y bordes en mallas de gradiometría gravimétrica	32
2.2.2 Interpretación semiautomática de anomalías	34
Deconvolución de Euler 3D	34
Deconvolución de Euler del Tensor Gravimétrico	38
2.3 Modelado Gravimétrico Inverso 3D	39
2.3.1 Formulación del problema gravimétrico inverso	39
2.3.2 Inversión por el método Simulated Annealing (SA)	41
Simulated Annealing con sensibilidad aumentada (SA-SA)	46

3 Modelos Sintéticos y Pruebas de Métodos	49
3.1 Modelos sintéticos	49
3.2 Pruebas de métodos de realce de bordes y estimación de profundidades . . .	56
3.2.1 Pruebas en mallas de anomalía gravimétrica	56
3.2.2 Pruebas en mallas de gradiometría gravimétrica	61
3.3 Pruebas del algoritmo de inversión SA en datos sintéticos	65
3.3.1 Algoritmo SA implementado en mallas de anomalía gravimétrica . . .	65
3.3.2 Algoritmo SA implementado en mallas de tensor gravimétrico	73
3.3.3 Análisis del SA y criterios de selección de parámetros iniciales	85
4 Aplicaciones del Modelado Gravimétrico	91
4.1 Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México	91
4.1.1 Anomalía gravimétrica de estructuras salinas	91
4.1.2 Estimación de la forma y profundidad de las estructuras salinas	94
4.1.3 Inversión de datos gravimétricos	97
4.2 Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.	102
4.2.1 Mallas de las componentes del tensor gravimétrico de la capa de roca del domo Vinton	102
4.2.2 Interpretación de las mallas del tensor gravimétrico de la capa de roca del domo Vinton	105
4.2.3 Inversión de las mallas del tensor gravimétrico de la capa de roca del domo Vinton	112
5 Conclusiones y Recomendaciones	119
5.1 Conclusiones	119
5.2 Recomendaciones para trabajos futuros	120
Referencias	121
Anexos	127
Anexo A Publicación Derivada de esta Tesis y Artículos Relacionados	129
Anexo B Implementación Computacional	175
Anexo C Complemento de Procesamiento de Datos Gravimétricos	191
C.1 Procesos aplicados a mallas de anomalía gravimétrica	191
C.2 Procesos aplicados a mallas de gradiometría gravimétrica	199

Lista de Figuras

FIGURA 2.1:	Fuerza de atracción gravitacional entre las masas puntuales m y m_0 , ubicadas en $Q(x', y', z')$ y $P(x, y, z)$, respectivamente. Por convención, el vector unitario \hat{r} se dirige de la fuente de campo, m , hacia el punto de observación, en el que se encuentra la masa de prueba m_0	8
FIGURA 2.2:	Potencial gravitacional, U , de un cuerpo con geometría compleja y distribución de densidad $\rho(x', y', z')$, observado en el punto $P(x, y, z)$	9
FIGURA 2.3:	Representación gráfica simplificada de las componentes del vector y tensor gravitacional \vec{g} y T_{ij} . Por cada componente vectorial, existen tres componentes tensoriales que son sus gradientes	12
FIGURA 2.4:	Fuente de campo gravitacional esférica con densidad uniforme ρ , radio a y centro situado en el origen, cuyo potencial es observado en el punto $P(x, y, R)$	14
FIGURA 2.5:	Superficie de potencial gravitacional generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos	16
FIGURA 2.6:	Superficies de las componentes g_x, g_y y g_z del vector de aceleración gravitacional, generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos	16
FIGURA 2.7:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravitacional, generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos	17
FIGURA 2.8:	Fuente de campo gravitacional de forma prismática rectangular con densidad constante, definida a partir de cuatro vértices: v_1, v_2, v_3 y v_4	18
FIGURA 2.9:	Superficie de potencial gravitacional generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X, Y y Z , respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X : -150 a 150 [m] , Y : -225 a 225 [m] y Z : 25 a 525 [m]	21

FIGURA 2.10: Superficies de las componentes g_x , g_y y g_z del vector de aceleración gravitacional, generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X, Y y Z, respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X: -150 a 150 [m] , Y: -225 a 225 [m] y Z: 25 a 525 [m]	21
FIGURA 2.11: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X, Y y Z, respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X: -150 a 150 [m] , Y: -225 a 225 [m] y Z: 25 a 525 [m]	22
FIGURA 2.12: Representación de la discretización de un medio en M prismas rectangulares y el cálculo de la respuesta gravimétrica g_i en el punto de observación P	24
FIGURA 2.13: Fuente puntual ubicada en x_0, y_0, z_0 , cuya anomalía es observada en el punto $P(x, y)$, de la malla situada sobre el plano $z = 0$. El vector \vec{r} define la posición de la fuente con respecto a $P(x, y)$	35
FIGURA 2.14: Diagrama de flujo representativo del algoritmo de inversión SA implementado en esta tesis (modificado de Ortiz-Alemán y Martín, 2005a)	45
FIGURA 3.1: Modelo sintético G1	50
FIGURA 3.2: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo sintético G1 , observado en una malla regular de 101×101 puntos. A la izquierda se muestra la malla libre de ruido y a la derecha, la malla con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$	50
FIGURA 3.3: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G1 , observado en una malla regular de 101×101 puntos	51
FIGURA 3.4: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G1 , observado en una malla regular de 101×101 puntos con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$	52
FIGURA 3.5: Modelo sintético G2	53
FIGURA 3.6: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo sintético G2 , observado en una malla regular de 101×101 puntos. A la izquierda se muestra la malla libre de ruido y a la derecha, la malla con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$	53

FIGURA 3.7:	Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G2 , observado en una malla regular de 101×101 puntos	54
FIGURA 3.8:	Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G2 , observado en una malla regular de 101×101 puntos con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$	55
FIGURA 3.9:	Izquierda: Malla de la componente G_z del modelo sintético G1 con ruido aleatorio añadido. Derecha: Malla CA_Gz_G1-RA , resultante del proceso de continuación ascendente de la malla $G_z(x, y, z)$, a $2UM$.	56
FIGURA 3.10:	Bordes interpretados de los cuerpos del modelo sintético G1 , a partir de procesamiento de la malla G_z de datos libres de ruido (arriba) y con ruido aleatorio añadido (abajo). Se muestran los contornos de las fuentes P (prisma), C (cilindro) y E (esfera) y se resaltan en tono gris	58
FIGURA 3.11:	Resultados de la deconvolución de Euler 3D aplicada a la malla G_z del modelo sintético G1 libre de ruido (izquierda) y con ruido aleatorio añadido (derecha). Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	59
FIGURA 3.12:	Bordes interpretados de los cuerpos del modelo sintético G1 , a partir de procesamiento de las mallas FTGG con datos libres de ruido (arriba) y con ruido aleatorio añadido (abajo). Se muestran los contornos de las fuentes P (prisma), C (cilindro) y E (esfera) y se resaltan en tono gris . .	62
FIGURA 3.13:	Resultados de la deconvolución de Euler 3D del tensor gravimétrico del modelo sintético G1 libre de ruido (izquierda) y con ruido aleatorio añadido (derecha). Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	63
FIGURA 3.14:	Modelo inicial aleatorio para la inversión de la componente g_z del vector de aceleración gravitacional y las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica	67
FIGURA 3.15:	Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo inicial (Figura 3.14), observado en una malla regular de 101×101 puntos	67
FIGURA 3.16:	Curvas de convergencia generadas por la inversión de datos gravimétricos del modelo G2 con los algoritmos SA y SA-SA	68
FIGURA 3.17:	Modelo final generado por la inversión de la componente g_z del vector de aceleración gravitacional, a través del algoritmo SA	69
FIGURA 3.18:	Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.17 , observado en una malla regular de 101×101 puntos	69
FIGURA 3.19:	Modelo final generado por la inversión de la componente g_z con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del vector de aceleración gravitacional, a través del algoritmo SA	70

FIGURA 3.20:	Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.19 , observado en una malla regular de 101×101 puntos	70
FIGURA 3.21:	Modelo final generado por la inversión de la componente g_z del vector de aceleración gravitacional, a través del algoritmo SA-SA	71
FIGURA 3.22:	Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.21 , observado en una malla regular de 101×101 puntos	71
FIGURA 3.23:	Modelo final generado por la inversión de la componente g_z con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del vector de aceleración gravitacional, a través del algoritmo SA-SA	72
FIGURA 3.24:	Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.23 , observado en una malla regular de 101×101 puntos	72
FIGURA 3.25:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo inicial (Figura 3.14), observado en una malla regular de 101×101 puntos	75
FIGURA 3.26:	Curvas de convergencia generadas por la inversión del tensor gravimétrico del modelo G2 con los algoritmos SA y SA-SA	76
FIGURA 3.27:	Modelo final generado por la inversión de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, a través del algoritmo SA	77
FIGURA 3.28:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.27 , observado en una malla regular de 101×101 puntos	78
FIGURA 3.29:	Modelo final generado por la inversión de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del tensor de gradiometría gravimétrica, a través del algoritmo SA	79
FIGURA 3.30:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.29 , observado en una malla regular de 101×101 puntos	80
FIGURA 3.31:	Modelo final generado por la inversión de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, a través del algoritmo SA-SA	81
FIGURA 3.32:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.31 , observado en una malla regular de 101×101 puntos	82
FIGURA 3.33:	Modelo final generado por la inversión de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del tensor de gradiometría gravimétrica, a través del algoritmo SA-SA	83
FIGURA 3.34:	Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.33 , observado en una malla regular de 101×101 puntos	84

FIGURA 3.35:	Superficie de aceptación de modelos por etapa de reducción de temperatura, para el método SA aplicado a la inversión de datos gravimétricos del modelo sintético G2	86
FIGURA 3.36:	Representación en perspectiva de las curvas de aceptación de modelos por etapa de reducción de temperatura, para $T_0 = 0.1$	87
FIGURA 3.37:	Curvas de convergencia del algoritmo SA aplicado a la inversión de datos gravimétricos del modelo sintético G2 con los parámetros $T_0 = 0.01$ y $RT \geq 0.9$	89
FIGURA 4.1:	Cuerpos de sal interpretados a partir de un modelo de velocidades 3D de sísmica PSDM. Cada cuerpo se compone por múltiples prismas de idénticas dimensiones	92
FIGURA 4.2:	Curva de densidad contra profundidad representativa de sedimentos del Golfo de México y densidad de cuerpos salinos (basada en Nelson y Fairchild, 1989)	93
FIGURA 4.3:	Anomalía gravimétrica generada por los cuerpos salinos mostrados en la Figura 4.1	93
FIGURA 4.4:	Mallas resultantes de métodos de realce de anomalías para definir la extensión lateral de las fuentes. a) Gradiente Horizontal, b) Amplitud de la Señal Analítica, c) Señal Analítica Mejorada de 1er Orden y d) Señal Analítica Mejorada de 2o Orden	94
FIGURA 4.5:	Izquierda: Ubicación de los máximos de los métodos mostrados en la Figura 4.4. Derecha: Interpretación de las fuentes de campo en planta, a partir de los máximos de dichos métodos	95
FIGURA 4.6:	Soluciones del método DE3D aplicado a la malla de anomalía gravimétrica (Figura 4.3)	96
FIGURA 4.7:	Modelo inicial de densidades 3D, construido a partir de información derivada de los métodos de realce de anomalía y estimación de profundidades aplicados a la malla de anomalía gravimétrica	96
FIGURA 4.8:	Anomalía gravimétrica generada por el modelo inicial mostrado en la Figura 4.7	97
FIGURA 4.9:	Modelo final de densidades 3D, construido a partir de la inversión de la anomalía gravimétrica de cuerpos salinos mostrada en la Figura 4.3	99
FIGURA 4.10:	Anomalía gravimétrica generada por el modelo final de densidades mostrado en la Figura 4.9	99
FIGURA 4.11:	Distribución espacial de los valores absolutos de los residuales en la malla de observaciones	100
FIGURA 4.12:	Distribución estadística de residuales entre las mallas de anomalía observada (Figura 4.3) y calculada (Figura 4.10)	100
FIGURA 4.13:	Curva de convergencia del proceso de inversión de datos gravimétricos	101
FIGURA 4.14:	Ubicación geográfica del área del levantamiento gradiométrico realizado por la empresa Bell Geospace	103

FIGURA 4.15: Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravitacional de la capa de roca del domo Vinton . . . 104

FIGURA 4.16: Mallas resultantes de métodos de realce de bordes para definir la extensión lateral de la capa de roca del domo Vinton. **a)** Valor absoluto del invariante I_1 , **b)** Invariante I_2 , **c)** Función detectora de bordes y **d)** Función detectora de bordes mejorada 105

FIGURA 4.17: **Izquierda:** Límites laterales de la capa de roca del domo Vinton inferidos por los métodos de realce de bordes mostrados en la Figura 4.16. **Derecha:** Interpretación unificada de la extensión lateral de la capa de roca del domo Vinton, a partir de estos límites 106

FIGURA 4.18: Soluciones de la deconvolución de Euler 3D del tensor gravimétrico de la capa de roca del domo Vinton (Figura 4.15). La línea negra representa la extensión lateral interpretada para la capa de roca 107

FIGURA 4.19: Geometría de la cima de la capa de roca del domo Vinton, estimada a partir de las profundidades del algoritmo de deconvolución de Euler 3D del tensor gravimétrico, la superficie del invariante I_2 y la extensión lateral interpretada y mostrada en la Figura 4.17 108

FIGURA 4.20: Visualización 3D del modelo de la capa de roca del domo Vinton, formado a partir de la geometría estimada de la cima (Figura 4.19) y las profundidades calculadas con la deconvolución de Euler (Figura 4.18) . 109

FIGURA 4.21: Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravitacional del modelo inicial de la capa de roca del domo Vinton (Figura 4.20) 111

FIGURA 4.22: Visualización 3D del modelo final de la capa de roca del domo Vinton, generado a partir del método de inversión aplicado. Los prismas azules traslúcidos representan elementos de la capa de roca cuya densidad fue reasignada a densidad de sedimentos en el proceso de inversión 113

FIGURA 4.23: Superficies de las componentes $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz} del tensor de gradiometría gravitacional del modelo final de la capa de roca del domo Vinton (Figura 4.22), generado a partir del método de inversión aplicado 114

FIGURA 4.24: Distribución estadística de residuales entre las mallas observadas (Figura 4.15) y calculadas (Figura 4.23) del tensor de gradiometría gravimétrica de la capa de roca del domo Vinton 115

FIGURA 4.25: Curva de convergencia del proceso de inversión de mallas de gradiometría gravimétrica de la capa de roca del domo Vinton 116

FIGURA 4.26: Mallas de residuales absolutos de las componentes del tensor de gradiometría gravimétrica del modelo final de la capa de roca del domo Vinton. Los límites laterales de la capa de roca interpretada se indican en línea discontinua 118

FIGURA B.1:	Curvas de costo computacional empleado para el cálculo de las componentes vectoriales G_x , G_y y G_z , de ensamblajes de hasta 100 millones de prismas, en un punto de observación. Procesador Utilizado: Intel [®] Xeon [®] CPU E5-2687W v2 @ 3.40GHz × 16	176
FIGURA B.2:	Curvas de costo computacional empleado para el cálculo de las componentes tensoriales T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} , de ensamblajes de hasta 100 millones de prismas, en un punto de observación. Procesador Utilizado: Intel [®] Xeon [®] CPU E5-2687W v2 @ 3.40GHz × 16	177
FIGURA C.1:	Izquierda: Malla GH_G1 , de gradiente horizontal de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla GH_CA_G1-RA , de gradiente horizontal de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	193
FIGURA C.2:	Izquierda: Malla Dz_G1 , de gradiente vertical de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla Dz_CA_G1-RA , de gradiente vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	194
FIGURA C.3:	Izquierda: Malla GH_Dz_G1 , de gradiente horizontal de la derivada vertical de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla GH_Dz_CA_G1-RA , de gradiente horizontal de la derivada vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	194
FIGURA C.4:	Izquierda: Malla SA_G1 , de señal analítica de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla SA_CA_G1-RA , de señal analítica de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	196
FIGURA C.5:	Izquierda: Malla SAM1_G1 , de señal analítica mejorada de primer orden de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla SAM1_CA_G1-RA , de señal analítica mejorada de primer orden de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E)	196

- FIGURA C.6:** **Izquierda:** Malla AIG_Dz_G1 , de ángulo de inclinación de gradientes de la derivada vertical de la componente G_z del modelo sintético $G1$ libre de ruido. **Derecha:** Malla $AIG_Dz_CA_G1-RA$, de ángulo de inclinación de gradientes de la derivada vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro) en línea continua, así como la proyección de la esfera (E) en línea discontinua y los contornos cero de las mallas procesadas en línea punteada 198
- FIGURA C.7:** **Izquierda:** Malla $|I1|_G1$, de valor absoluto del invariante I_1 del $FTGG$ del modelo sintético $G1$ libre de ruido. **Derecha:** Malla $|I1|_G1-RA$, de valor absoluto del invariante I_1 del $FTGG$ con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 199
- FIGURA C.8:** **Izquierda:** Malla $I2_G1$, del invariante I_2 del $FTGG$ del modelo sintético $G1$ libre de ruido. **Derecha:** Malla $I2_G1-RA$, del invariante I_2 del $FTGG$ con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 200
- FIGURA C.9:** Mallas de señal analítica direccional en X , Y y Z del $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla SAX_G1 , de señal analítica direccional en X . **Derecha:** Malla SAy_G1 , de señal analítica direccional en Y . **Abajo:** Malla SAz_G1 , de señal analítica direccional en Z . Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 202
- FIGURA C.10:** Mallas de señal analítica direccional en X , Y y Z de la continuación ascendente del $FTGG$ del modelo sintético $G1$ con ruido aleatorio añadido. **Izquierda:** Malla SAX_CA_G1-RA , de señal analítica direccional en X . **Derecha:** Malla SAy_CA_G1-RA , de señal analítica direccional en Y . **Abajo:** Malla SAz_CA_G1-RA , de señal analítica direccional en Z . Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 203
- FIGURA C.11:** Mallas de gradiente vertical de la señal analítica direccional en X y Y , así como de la función detectora de bordes del $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla Dz_SAX_G1 , de gradiente vertical de la señal analítica direccional en X . **Derecha:** Malla Dz_SAy_G1 , de gradiente vertical de la señal analítica direccional en Y . **Abajo:** Malla DB_G1 , de la función detectora de bordes. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 204

- FIGURA C.12:** Mallas del gradiente vertical de la señal analítica direccional en X y Y , así como de la función detectora de bordes de la continuación ascendente $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla $Dz_SAx_CA_G1-RA$, de gradiente vertical de la señal analítica direccional en X . **Derecha:** Malla $Dz_SAy_CA_G1-RA$, de gradiente vertical de la señal analítica direccional en Y . **Abajo:** Malla DB_G1-RA , de la función detectora de bordes. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E) 205
- FIGURA C.13:** **Izquierda:** Malla DBM_G1 , de la función detectora de bordes mejorada del $FTGG$ del modelo sintético $G1$ libre de ruido. **Derecha:** Malla DBM_G1-RA , de la función detectora de bordes mejorada del $FTGG$ con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro) en línea continua, así como la proyección de la esfera (E) en línea discontinua y los contornos cero de las mallas procesadas en línea punteada 206

Lista de Tablas

TABLA 1.1:	Velocidad, densidad y permeabilidad de cuerpos salinos emplazados en zonas de exploración petrolera	1
TABLA 2.1:	Índices estructurales del potencial gravimétrico y sus derivadas espaciales hasta de segundo orden, para diferentes modelos utilizados como base en la interpretación a través de la deconvolución de Euler	37
TABLA 2.2:	Analogías entre términos utilizados en la mecánica estadística y los problemas de optimización por el método Simulated Annealing	41
TABLA 3.1:	Modelos sintéticos generados para probar el rendimiento de los algoritmos de delimitación de fuentes, estimación de profundidades e inversión de mallas de anomalías	49
TABLA 3.2:	Interpretación de cuerpos del modelo G1 , a partir del procesamiento de mallas de anomalía gravimétrica libre de ruido y con ruido aleatorio añadido	60
TABLA 3.3:	Interpretación de cuerpos del modelo G1 , a partir del procesamiento de mallas del tensor de gradiometría gravimétrica libres de ruido y con ruido aleatorio añadido	64
TABLA 3.4:	Parámetros iniciales de los algoritmos SA y SA-SA utilizados en la inversión de la malla de anomalía gravimétrica, G_z , del modelo G2	66
TABLA 3.5:	Resultados de la inversión de la malla de anomalía gravimétrica, G_z , del modelo G2 , a través de los algoritmos SA y SA-SA	68
TABLA 3.6:	Parámetros iniciales de los algoritmos SA y SA-SA utilizados en la inversión de mallas de tensor gravimétrico del modelo G2	74
TABLA 3.7:	Resultados de la inversión de las mallas de tensor gravimétrico del modelo G2 , a través de los algoritmos SA y SA-SA	76
TABLA 3.8:	Valores asignados a T_0 , RT y VM en el barrido paramétrico	85
TABLA 4.1:	Características principales del modelo 3D inicial del algoritmo de inversión gravimétrica utilizado	98
TABLA 4.2:	Parámetros relacionados con el algoritmo de inversión 3D de datos gravimétricos aplicados a la anomalía de cuerpos salinos del Golfo de México	98
TABLA 4.3:	Características del modelo 3D inicial del algoritmo de inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton	110

TABLA 4.4:	Parámetros iniciales del algoritmo SA-SA utilizados en la inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton . . .	112
TABLA 4.5:	Resultados del algoritmo de inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton	112
TABLA 4.6:	Valores máximos y mínimos de las diferencias absolutas entre mallas de gradiometría gravimétrica observadas y calculadas para el modelo final de la capa de roca del domo Vinton	116

1

INTRODUCCIÓN

El proceso exploratorio en busca de hidrocarburos se basa en gran medida en métodos geofísicos, siendo el método sísmico de reflexión, el más intensamente utilizado en la actualidad. El creciente interés en los plays subsalinos ubicados en el Golfo de México y en otras cuencas sedimentarias alrededor del mundo, han convertido en un reto mayor a la exploración de aceite y gas en esas regiones (Dribus et al., 2008). Las propiedades físicas contrastantes de los cuerpos salinos y estructuras asociadas, tales como el alto contraste de velocidades y densidades con respecto a los sedimentos del medio encajonante, generan patrones complejos de difracción de ondas, impidiendo una correcta iluminación de las zonas cercanas y bajo ellas.

En este contexto, el método gravimétrico es considerado como una herramienta auxiliar del método sísmico y se aplica para mejorar la calidad de las imágenes subsalinas, dada su capacidad de respuesta ante los marcados contrastes de densidad en zonas de emplazamiento de cuerpos salinos (e.g. Colombo y De Stefano, 2007). En la Tabla 1.1 se muestran rangos típicos de valores de velocidad de onda sísmica, densidad y permeabilidad de cuerpos de sal emplazados en zonas de exploración de hidrocarburos.

TABLA 1.1: Velocidad, densidad y permeabilidad de cuerpos salinos emplazados en zonas de exploración petrolera (unidades del Sistema Internacional).

Propiedad	Valor (rango)	Fuente
Velocidad Sísmica (onda P)	4,270 a 5,190 [m · s ⁻¹]	Grant y West (1965)
Densidad	2,100 a 2,200 [kg · m ⁻³]	Gardner et al. (1974)
Permeabilidad	<10 ⁻²⁰ [m ²]	Carter et al. (1993)

El alto contraste de velocidad de onda sísmica en la interfase entre los cuerpos salinos y los sedimentos genera reflexiones y refracciones de gran amplitud, haciendo muy difícil obtener imágenes claras de la base de la sal, así como de estructuras que la subyacen (Montgomery y Moore, 1997; Dribus et al., 2008). Aunque se han logrado grandes avances en la mejora de imágenes de plays subsalinos a través de la aplicación de técnicas de Migración

en Profundidad Pre-Apilamiento (PSDM por sus siglas en inglés), la complejidad de las estructuras subsalinas sigue siendo un reto importante en la exploración, lo que requiere de nuevas metodologías de modelado y análisis de datos provenientes de los métodos geofísicos empleados.

En estudios recientes se ha abordado el problema de la obtención de imágenes representativas del subsuelo en zonas de geología compleja involucrando el modelado directo e inverso de anomalías de campos potenciales. Ortiz-Alemán y Urrutia-Fucugauchi (2010) aplicaron el modelado magnético en la zona central de la estructura de impacto del Chicxulub, donde ha sido difícil obtener imágenes claras a partir de datos sísmicos de reflexión. René (1986) desarrolló un método de inversión de datos gravimétricos mediante la aplicación iterativa de criterios de “apertura, rechazo y relleno” dentro de un procedimiento de modelación basado en el uso de ensambles de prismas con contrastes de densidad previamente asignados y el criterio de “forma de la anomalía” para el llenado de los prismas. (Nagihara y Hall, 2001) aplicaron el método de optimización global denominado Simulated Annealing (SA) para invertir datos sintéticos provenientes de un diapiro salino simplificado, restringiendo la profundidad del diapiro a través de modelado inverso 3D. Zhang et al. (2004) determinaron la estructura cortical central de Taiwan a través de inversión gravimétrica con un algoritmo genético paralelo, utilizando un modelo inicial derivado de tomografía sísmica de onda P en 3D. Roy et al. (2005) invirtieron datos gravimétricos usando SA en el lago Vostok en el Antártico Este con el objetivo de estimar las interfaces agua-sedimento y sedimento-basamento y Krahenbuhl (2005) introdujo un método de inversión denominado inversión binaria, el cual utiliza un ensamble de prismas como espacio de modelos y contrastes de densidad como parámetros.

Por otro lado, debido al avance tecnológico en adquisición de datos, se han vuelto más comunes los estudios de exploración utilizando observaciones de las componentes del Tensor Completo de Gradiometría Gravimétrica (FTGG por sus siglas en inglés), en conjunto con las técnicas desarrolladas para su modelado directo e inverso. Couder-Castañeda et al. (2015) y Couder-Castañeda et al. (2013) introdujeron técnicas de cómputo de alto rendimiento en clusters de arquitecturas híbridas, empleando los paradigmas de Procesamiento Múltiple en memoria compartida, Interfaz de Paso de Mensajes y Arquitectura Unificada de Dispositivos de Cómputo (OpenMP, MPI y CUDA, respectivamente) para calcular las componentes vectoriales y tensoriales gravitacionales de ensambles de grandes dimensiones. Hou et al. (2015) efectuaron un análisis de rendimiento de algoritmos de inversión “pesada y enfocada” de datos de FTGG implementados en arquitecturas paralelas, con los paradigmas MPI y CUDA, combinándolos para formular un algoritmo híbrido, con el que obtuvieron resultados congruentes con otros estudios, al aplicarlos en datos del domo Vinton, LA, E.U.A. Uieda y Barbosa (2012) presentaron un método de inversión de datos del FTGG plantando densidades anómalas en el que utilizaron el criterio de “forma de la anomalía” para el desajuste de datos en conjunción con la norma de desajuste L_2 , logrando una correcta recuperación de la geometría de las fuentes de interés aún en presencia de interferencia causada por fuentes cercanas no consideradas y Oliveira y Barbosa (2013) implementaron un método no lineal de inversión conjunta de las componentes del FTGG, definiendo la geometría de la fuente

a través de capas horizontales de sección polinomial con espesor y densidad constante y lo aplicaron a datos del domo Vinton, obteniendo un modelo para la capa de roca (caprock) del mismo.

En esta tesis se presenta un método para la construcción de modelos gravimétricos 3D de estructuras geológicas relacionadas con el emplazamiento de cuerpos salinos en las que se podrían presentar acumulaciones de hidrocarburos. Este método consiste en la integración de resultados de algoritmos de detección de bordes e interpretación semiautomática de anomalías, con inversión de datos en 3D. El modelado propuesto es aplicable tanto a mallas de anomalía, como de gradiometría gravimétrica, ejemplificando su aplicación en dos casos: **Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México**, publicación derivada de este trabajo (Nava-Flores et al., 2016) y **Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.**

1.1 Objetivos

En el trabajo de tesis se contempla, como objetivo principal, el modelado 3D de estructuras en el subsuelo, a partir de datos de gravimetría o gradiometría gravimétrica en zonas bajo influencia de tectónica salina, a través de la integración de resultados de diferentes técnicas auxiliares en la interpretación de datos de métodos potenciales: realce y detección de bordes, interpretación semiautomática de anomalías e inversión 3D.

Como objetivos complementarios, requeridos para el cumplimiento del objetivo principal, se establecieron la implementación computacional de los métodos utilizados y la prueba y validación de dichos métodos, a través del análisis de resultados y pruebas de rendimiento en modelos sintéticos libres de ruido y con ruido pseudoaleatorio añadido.

1.2 Metodología

La metodología propuesta para cumplir con los objetivos planteados, se basa en una serie de procedimientos ordenados que inicia con la generación de modelos sintéticos en los que se prueban y validan los métodos requeridos y termina con la obtención de un modelo representativo del subsuelo en la zona de estudio, incluyendo un análisis de calidad del mismo.

Describiendo esta metodología en general, comienza con la generación de modelos sintéticos para la prueba y validación de métodos de realce de bordes, interpretación semiautomática de anomalías e inversión de datos 3D.

Posteriormente se implementan computacionalmente todos los métodos por aplicar y se realizan las pruebas en datos sintéticos libres de ruido, así como con ruido pseudoaleatorio añadido. Para la prueba de los métodos de realce de bordes e interpretación semiautomática de anomalías, se hace una comparación de bordes y profundidades estimadas a partir de los métodos aplicados, con respecto a los bordes y profundidades de los cuerpos que conforman el modelo sintético, lo que permitirá seleccionar los métodos de realce de bordes más eficientes, así como los parámetros que controlan el algoritmo de interpretación semiautomática utilizado. Por otro lado, para la prueba del algoritmo de inversión 3D, primero se realiza un barrido paramétrico cuyos resultados se analizan para seleccionar los parámetros iniciales del algoritmo y posteriormente se compara el modelo resultante con respecto al modelo sintético.

Una vez concluidas las pruebas de métodos en datos sintéticos, se aplican a los datos de las zonas propuestas influenciadas por tectónica salina, que en este caso fueron: Sur del Golfo de México (ubicación confidencial) y Domo Vinton, LA, E.U.A. Los resultados obtenidos por los métodos de realce de bordes e interpretación semiautomática de anomalías, se utilizan para crear modelos de las zonas de estudio, que posteriormente serán depurados al utilizarlos como modelos iniciales en el algoritmo de inversión.

Finalmente, se cuantifica la calidad del modelo final, a partir de un análisis del desajuste de datos invertidos con respecto a datos observados y estadísticas de residuales.

1.3 Información Geofísica

La información geofísica con la que se realizó el modelado que se presenta en esta tesis, de acuerdo a los dos casos de aplicación, se muestra a continuación:

Para el **Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México**, se utilizaron las coordenadas de cuerpos salinos interpretados de un modelo de velocidades de migración preapilamiento 3D (3D PSDM por sus siglas en inglés), de un cubo de datos sísmicos pertenecientes a una zona de exploración en la porción sureste del Golfo de México, cuya ubicación exacta no se presenta por cuestiones confidenciales.

En el caso del **Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.**, se usaron las mallas de datos de gradiometría gravimétrica aérea del domo Vinton, adquiridos y proporcionados por la empresa Bell Geospace Inc.

En ambos casos se utilizó la curva de densidades del subsuelo representativa del Golfo de México (densidad contra profundidad), estimada a partir de datos de pozos (Hudec y Jackson, 2007; Nelson y Fairchild, 1989).

1.4 Contribuciones

Como principal contribución derivada de este trabajo de tesis, se presenta la metodología de integración de resultados y construcción de modelos 3D basados en ensambles de prismas rectangulares, a partir de diferentes métodos de estimación de bordes, interpretación semiautomática de anomalías e inversión de datos 3D.

Otra contribución importante en el campo del modelado de datos gravimétricos y de gradiometría gravimétrica, es el algoritmo de inversión tipo Simulated Annealing con la inclusión de un factor de incremento de sensibilidad con la profundidad (Simulated Annealing con Sensitividad Aumentada).

Finalmente, una contribución importante más, es el análisis paramétrico del algoritmo Simulated Annealing y los criterios de selección de sus parámetros iniciales.

2

FUNDAMENTOS TEÓRICOS

2.1 Modelado Gravimétrico Directo 3D

El modelado gravimétrico tiene sus bases en la teoría del potencial, a la cual se le podría considerar como encargada del estudio de fenómenos caracterizados por la ecuación de Laplace (Kellogg, 1967). Entre los diferentes objetos de estudio de esta disciplina, se encuentra la fuerza de atracción entre masas (gravitación), que es de interés en esta tesis, debido a los conceptos que se utilizan para su estudio, entre los que destacan: *potencial, vector y tensor gravitacionales*, definidos a continuación.

2.1.1 Potencial, vector y tensor gravitacional

De acuerdo a la ley de Gravitación Universal de Newton (Newton, 1687), si se consideran dos masas puntuales en un sistema de referencia cartesiano: m , ubicada en $Q(x', y', z')$ y m_0 , ubicada en $P(x, y, z)$ (Figura 2.1), la magnitud de la fuerza de atracción entre ellas está dada por:

$$F = \gamma \frac{m m_0}{r^2} \quad (2.1)$$

donde γ es la constante de gravitación universal, que en las unidades del sistema internacional (S.I.) toma el valor de $6.67 \times 10^{-11} \left[\frac{m^3}{kg s^2} \right]$, m es la masa puntual que genera el campo gravitacional, m_0 es una masa de prueba unitaria y $r = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2}$ es la separación entre ellas.

A partir de la ecuación 2.1, es posible determinar la aceleración gravitacional producida por m , en la ubicación de la masa de prueba, dividiendo la fuerza de atracción entre la masa m_0 :

$$\vec{g}(P) = -\gamma \frac{m}{r^2} \hat{r} \quad (2.2)$$

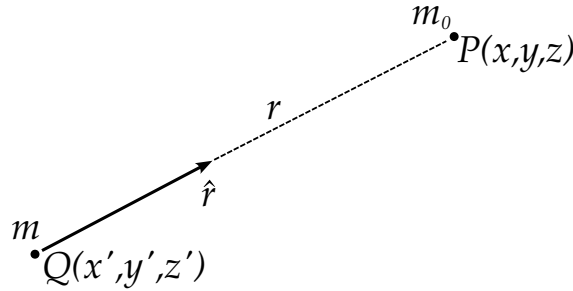


FIGURA 2.1: Fuerza de atracción gravitacional entre las masas puntuales m y m_0 , ubicadas en $Q(x', y', z')$ y $P(x, y, z)$, respectivamente. Por convención, el vector unitario \hat{r} se dirige de la fuente de campo, m , hacia el punto de observación, en el que se encuentra la masa de prueba m_0 .

En la ecuación 2.2, \hat{r} es un vector unitario, que por convención se dirige de la masa causante del campo, m , al punto de observación $P(x, y, z)$ (de ahí el signo negativo).

Por otro lado, el teorema de Helmholtz establece que: “Cualquier campo vectorial continuo, \vec{F} , que tienda a desvanecerse en el infinito, se puede representar como la suma del gradiente de un campo escalar y el rotacional de un campo vectorial”:

$$\vec{F} = \nabla\phi + \nabla \times \vec{A} \quad (2.3)$$

donde ϕ y \vec{A} son los potenciales escalar y vectorial de \vec{F} , respectivamente.

Ahora bien, considerando la ecuación 2.2 y debido a que \vec{g} es un campo irrotacional o conservativo ($\nabla \times \vec{g} = 0$), de acuerdo al teorema de Helmholtz (Ec. 2.3), \vec{g} se obtiene a través del gradiente de un campo potencial escalar U :

$$\vec{g} = \nabla U(P)$$

donde $U(P) = \gamma \frac{m}{r}$ es el potencial gravitacional de la masa puntual m .

Para obtener la expresión del potencial gravitacional generalizado a cuerpos con geometrías irregulares y densidades heterogéneas, se considera el caso de un cuerpo cuya masa ocupa un volumen v y su densidad presenta una distribución que puede variar de punto a punto en su interior, $\rho(x', y', z')$, tal como se esquematiza en la Figura 2.2:

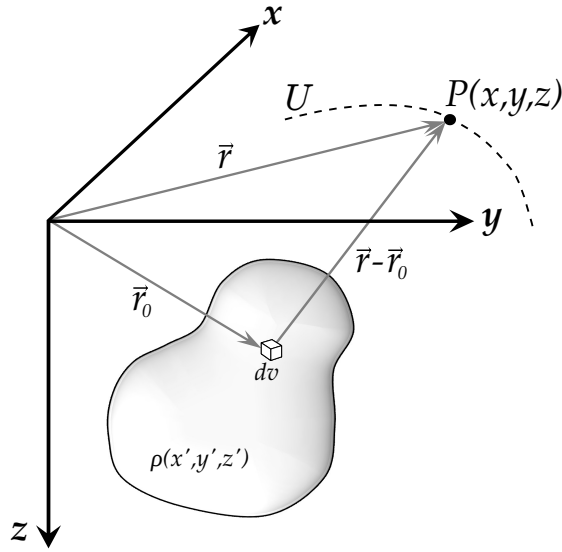


FIGURA 2.2: Potencial gravitacional, U , de un cuerpo con geometría compleja y distribución de densidad $\rho(x', y', z')$, observado en el punto $P(x, y, z)$.

En este caso, debido a que el potencial gravitacional cumple con la propiedad de superposición, $U(P)$ será la suma del potencial de todos y cada uno los elementos que conforman el cuerpo en cuestión, que llevada al límite cuando cada elemento tiene dimensiones infinitesimales, estará dado por la integral de volumen:

$$\begin{aligned}
 U(P) &= \gamma \int_v \frac{dm}{|\vec{r} - \vec{r}_0|} \\
 &= \gamma \int_v \frac{\rho}{|\vec{r} - \vec{r}_0|} dv \\
 &= \gamma \int_{z'} \int_{y'} \int_{x'} \frac{\rho(x', y', z')}{r} dx' dy' dz' \quad (2.4)
 \end{aligned}$$

La ecuación 2.4 representa el **potencial gravitacional** de un cuerpo con geometría arbitraria y densidad heterogénea, observado en un punto, considerando un sistema de referencia cartesiano.

La importancia del potencial gravitacional radica en que a través de él se obtienen tanto el vector como el tensor gravitacional, por medio de la aplicación del operador diferencial nabla (∇) en forma de gradiente:

Vector de atracción gravitacional

Se obtiene a través del gradiente del potencial:

$$\begin{aligned}
 \vec{g}(P) &= \nabla U(P) \\
 &= \frac{\partial U}{\partial x_i} \hat{e}_i = U_{,i} \hat{e}_i \\
 &= g_i \hat{e}_i
 \end{aligned} \tag{2.5}$$

En la ecuación 2.5 se ha utilizado notación indicial, que al desarrollarla en un sistema de referencia cartesiano 3D, permite obtener el vector \vec{g} , con sus tres componentes ortogonales:

$$\vec{g}(x, y, z) = g_x \hat{i} + g_y \hat{j} + g_z \hat{k}$$

donde:

$$g_x = \frac{\partial U(P)}{\partial x} = -\gamma \int_{z'} \int_{y'} \int_{x'} \frac{(x - x') \rho(x', y', z')}{r^3} dx' dy' dz' \tag{2.6}$$

$$g_y = \frac{\partial U(P)}{\partial y} = -\gamma \int_{z'} \int_{y'} \int_{x'} \frac{(y - y') \rho(x', y', z')}{r^3} dx' dy' dz' \tag{2.7}$$

$$g_z = \frac{\partial U(P)}{\partial z} = -\gamma \int_{z'} \int_{y'} \int_{x'} \frac{(z - z') \rho(x', y', z')}{r^3} dx' dy' dz' \tag{2.8}$$

Tensor gravitacional

Se obtiene a través del gradiente del vector de atracción gravitacional:

$$\begin{aligned}
 T_{ij} &= \nabla \vec{g}(P) \\
 &= \frac{\partial g_i \hat{e}_i}{\partial x_j} \otimes \hat{e}_j = (g_i \hat{e}_i)_{,j} \otimes \hat{e}_j \\
 &= (g_i)_{,j} \hat{e}_i \otimes \hat{e}_j
 \end{aligned} \tag{2.9}$$

En la ecuación 2.9, el operador \otimes es el producto tensorial y $\hat{e}_i \otimes \hat{e}_j$ son las bases (diadas) del tensor T_{ij} , cuyas componentes son las derivadas de las componentes de \vec{g} (base \hat{e}_i), con respecto a la base \hat{e}_j . En un sistema de referencia cartesiano 3D, las componentes de T_{ij} son:

$$T_{xx} = \frac{\partial g_x}{\partial x} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(x - x')^2 - r^2}{r^5} \right] \rho(x', y', z') dx' dy' dz' \tag{2.10}$$

$$T_{xy} = \frac{\partial g_x}{\partial y} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(x-x')(y-y')}{r^5} \right] \rho(x', y', z') dx' dy' dz' \quad (2.11)$$

$$T_{xz} = \frac{\partial g_x}{\partial z} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(x-x')(z-z')}{r^5} \right] \rho(x', y', z') dx' dy' dz' \quad (2.12)$$

$$T_{yy} = \frac{\partial g_y}{\partial y} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(y-y')^2 - r^2}{r^5} \right] \rho(x', y', z') dx' dy' dz' \quad (2.13)$$

$$T_{yz} = \frac{\partial g_y}{\partial z} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(y-y')(z-z')}{r^5} \right] \rho(x', y', z') dx' dy' dz' \quad (2.14)$$

$$T_{zz} = \frac{\partial g_z}{\partial z} = \gamma \int_{z'} \int_{y'} \int_{x'} \left[\frac{3(z-z')^2 - r^2}{r^5} \right] \rho(x', y', z') dx' dy' dz' \quad (2.15)$$

Vale la pena enfatizar que, aunque el tensor T_{ij} tiene nueve componentes en el sistema cartesiano 3D en el que está referenciado, sólo seis de ellas son diferentes debido a que es simétrico ($T_{ij} = T_{ji}$) y son estas seis componentes las que se muestran en las ecuaciones 2.10 a 2.15.

Las componentes de \vec{g} y T_{ij} se pueden agrupar en arreglos 1D y 2D, respectivamente (ecuaciones 2.16 y 2.17) y se pueden representar geoméricamente con ayuda de un gráfico simplificado (Figura 2.3):

$$\vec{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \quad (2.16)$$

$$T_{ij} = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix} \quad (2.17)$$

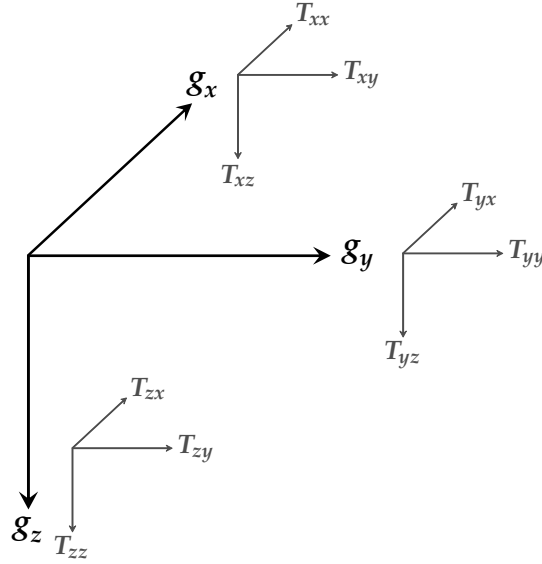


FIGURA 2.3: Representación gráfica simplificada de las componentes del vector y tensor gravitacional \vec{g} y T_{ij} . Por cada componente vectorial, existen tres componentes tensoriales que son sus gradientes.

Una propiedad importante que surge al efectuar la operación $T_{xx} + T_{yy} + T_{zz}$, utilizando las ecuaciones 2.10, 2.13 y 2.15, que es equivalente a obtener la traza del tensor T_{ij} , es que: "El potencial gravitacional es armónico en todos los puntos libres de fuentes de campo y en estos sitios está caracterizado por la ecuación de Laplace" (Ec. 2.18):

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = 0$$

$$\Rightarrow \gamma \int \int \int_{z' y' x'} \left[\frac{3\{(x-x')^2 + (y-y')^2 + (z-z')^2\} - 3r^2}{r^5} \right] \rho(x', y', z') dx' dy' dz' = 0$$

$$\Rightarrow \nabla^2 U(P) = 0 \quad (2.18)$$

Finalmente, el hecho de que el potencial gravimétrico sea armónico en sitios libres de fuentes de campo, tiene entre otras implicaciones, que el número de componentes independientes del tensor gravitacional, T_{ij} , se reduzca a cinco, debido a la dependencia que existe de cualquiera de los tres elementos de su diagonal principal hacia los otros dos:

$$\begin{aligned} \nabla^2 U(P) &= 0 \\ \Rightarrow T_{xx} + T_{yy} + T_{zz} &= 0 \\ \therefore \begin{cases} T_{xx} = -[T_{yy} + T_{zz}] \\ T_{yy} = -[T_{xx} + T_{zz}] \\ T_{zz} = -[T_{xx} + T_{yy}] \end{cases} \end{aligned}$$

2.1.2 Modelado de cuerpos simples

Una vez que se tienen las expresiones matemáticas necesarias para calcular el potencial (Ec. 2.4), así como las componentes del vector (Ecs. 2.6 a 2.8) y tensor gravitacional (Ecs. 2.10 a 2.15) de un cuerpo 3D en un punto de observación, el modelado gravimétrico consistirá en elegir el tipo de fuente de campo y los puntos de observación en que se calcularán algunas o todas estas cantidades gravitacionales.

Se muestran a continuación las derivaciones de estas cantidades gravitacionales para dos tipos de fuentes con geometría simple que se utilizaron para generar los modelos de prueba de algoritmos de procesamiento e inversión de datos en esta tesis: esfera y prisma rectangular con densidad uniforme.¹

Esfera

El potencial de una esfera con densidad uniforme y centro ubicado en el origen de coordenadas en un sistema de referencia 3D, observado en un punto externo, que para simplificar los cálculos se ha situado sobre uno de los ejes (Figura 2.4), es:

$$\begin{aligned}
 U(P) &= \gamma\rho \int_0^\pi \int_0^{2\pi} \int_0^a \frac{r'^2 \sin\theta}{r'} d\theta d\phi dr' & \text{con: } r' &= \sqrt{a^2 + R^2 - 2aR\cos\theta} \\
 &= \frac{2\pi\gamma\rho a^3}{3} \int_0^\pi \left[\frac{\sin\theta}{r'} \right] d\theta & & (2.19)
 \end{aligned}$$

- Nótese que debido a la geometría del cuerpo, es más conveniente expresar la integral en coordenadas esféricas. -

Ahora, tomando en cuenta que $dr' = \frac{aR\sin\theta}{r'} d\theta$, sustituyendo y cambiando la variable de integración θ por r' en la ecuación 2.19:

$$U(P) = \frac{2\pi\gamma\rho a^2}{3R} \int_{R-a}^{R+a} dr' = \frac{4\pi\gamma\rho a^3}{3R} \quad (2.20)$$

¹Debido al tipo de modelado directo e inverso abordado en este trabajo de tesis, sólo se muestran las expresiones para fuentes esféricas y prismáticas rectangulares con densidad uniforme. Para mayores detalles respecto al modelado gravimétrico de fuentes en general, se recomienda consultar bibliografía especializada relacionada con el tema (e.g. Kellogg, 1967; Blakely, 1996; Hinze et al., 2013).

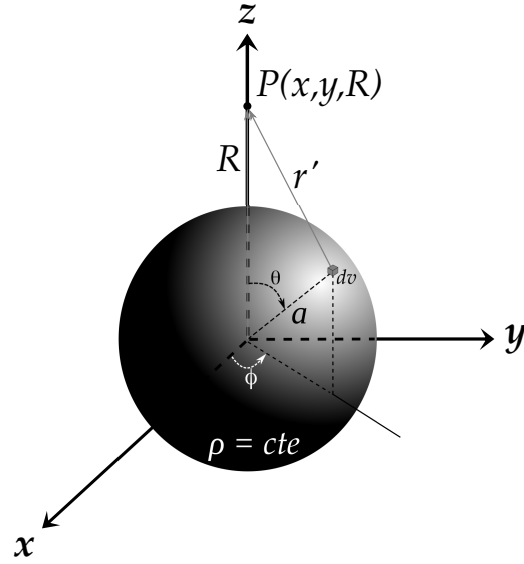


FIGURA 2.4: Fuente de campo gravitacional esférica con densidad uniforme ρ , radio a y centro situado en el origen, cuyo potencial es observado en el punto $P(x, y, R)$.

Finalmente, si se considera que la posición del punto de observación puede ser arbitraria en la ecuación 2.20, ($P(x, y, z) \Rightarrow R = \sqrt{x^2 + y^2 + z^2}$), el potencial quedará definido en coordenadas cartesianas como:

$$U(P) = \frac{4\pi\gamma\rho a^3}{3\sqrt{x^2 + y^2 + z^2}} \quad (2.21)$$

En analogía a la formulación de las ecuaciones 2.6 a 2.8, las componentes del vector de aceleración gravitacional para el cuerpo esférico serán:

$$\begin{aligned} g_x(x, y, z) &= \frac{\partial}{\partial x} \left[\frac{4\pi\gamma\rho a^3}{3\sqrt{x^2 + y^2 + z^2}} \right] \\ &= -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{x}{\sqrt{x^2 + y^2 + z^2}^3} \right] \end{aligned} \quad (2.22)$$

$$\begin{aligned} g_y(x, y, z) &= \frac{\partial}{\partial y} \left[\frac{4\pi\gamma\rho a^3}{3\sqrt{x^2 + y^2 + z^2}} \right] \\ &= -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{y}{\sqrt{x^2 + y^2 + z^2}^3} \right] \end{aligned} \quad (2.23)$$

$$\begin{aligned} g_z(x, y, z) &= \frac{\partial}{\partial z} \left[\frac{4\pi\gamma\rho a^3}{3\sqrt{x^2 + y^2 + z^2}} \right] \\ &= -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{z}{\sqrt{x^2 + y^2 + z^2}^3} \right] \end{aligned} \quad (2.24)$$

y haciendo una nueva analogía, ahora con respecto a la formulación de las ecuaciones 2.10 a 2.15, las componentes del tensor gravitacional para el cuerpo esférico serán:

$$\begin{aligned} T_{xx}(x, y, z) &= \frac{\partial}{\partial x} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{x}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= \frac{4}{3}\pi\gamma\rho a^3 \left[\frac{2x^2 - (y^2 + z^2)}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.25)$$

$$\begin{aligned} T_{xy}(x, y, z) &= \frac{\partial}{\partial y} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{x}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= 4\pi\gamma\rho a^3 \left[\frac{xy}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.26)$$

$$\begin{aligned} T_{xz}(x, y, z) &= \frac{\partial}{\partial z} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{x}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= 4\pi\gamma\rho a^3 \left[\frac{xz}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.27)$$

$$\begin{aligned} T_{yy}(x, y, z) &= \frac{\partial}{\partial y} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{y}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= \frac{4}{3}\pi\gamma\rho a^3 \left[\frac{2y^2 - (x^2 + z^2)}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.28)$$

$$\begin{aligned} T_{yz}(x, y, z) &= \frac{\partial}{\partial Z} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{y}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= 4\pi\gamma\rho a^3 \left[\frac{yz}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.29)$$

$$\begin{aligned} T_{zz}(x, y, z) &= \frac{\partial}{\partial z} \left\{ -\frac{4}{3}\pi\gamma\rho a^3 \left[\frac{z}{\sqrt{x^2 + y^2 + z^2}^3} \right] \right\} \\ &= \frac{4}{3}\pi\gamma\rho a^3 \left[\frac{2z^2 - (x^2 + y^2)}{\sqrt{x^2 + y^2 + z^2}^5} \right] \end{aligned} \quad (2.30)$$

La forma que tienen las superficies de potencial, componentes del vector y del tensor gravitacional del cuerpo esférico en mallas de observación equiespaciadas, se puede ver en las Figuras 2.5, 2.6 y 2.7:

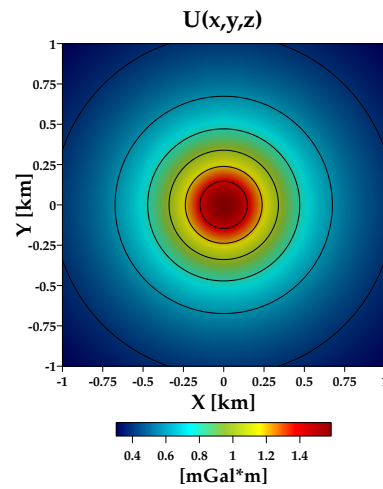


FIGURA 2.5: Superficie de potencial gravitacional generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos.

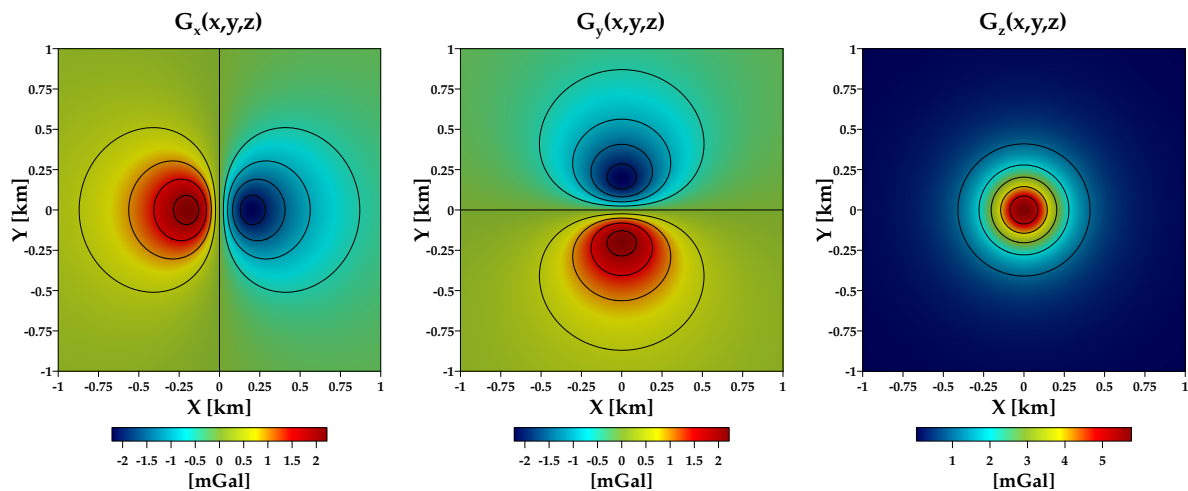


FIGURA 2.6: Superficies de las componentes g_x , g_y y g_z del vector de aceleración gravitacional, generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos.

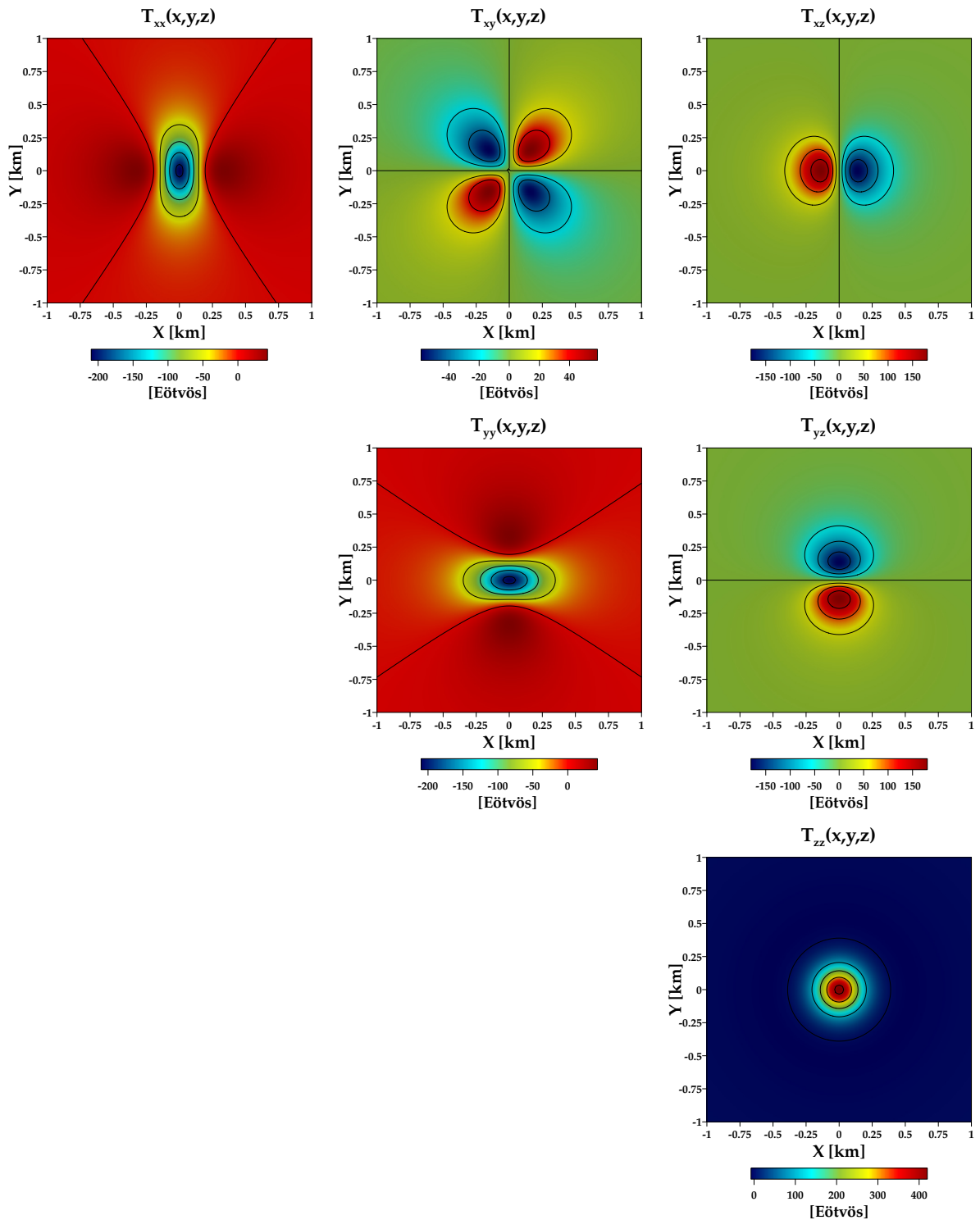


FIGURA 2.7: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por un cuerpo esférico con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$, radio $a = 250 \text{ [m]}$ y centro ubicado a 275 [m] de profundidad, observado en una malla regular de 101×101 puntos.

Prisma rectangular

Para el caso del prisma rectangular, se tiene el cuerpo esbozado en la Figura 2.8, con el punto de observación, P , situado en el origen para simplificar los cálculos:

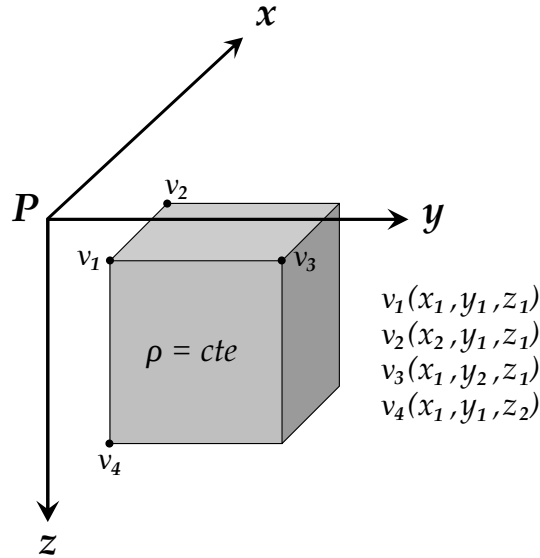


FIGURA 2.8: Fuente de campo gravitacional de forma prismática rectangular con densidad constante, definida a partir de cuatro vértices: v_1 , v_2 , v_3 y v_4 .

El potencial gravitacional debido a esta fuente de campo se expresa como:

$$U(P) = \gamma\rho \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{1}{r} dx dy dz \quad (2.31)$$

donde: $r = \sqrt{x^2 + y^2 + z^2}$, debido a que $P(x', y', z') = (0, 0, 0)$.

De acuerdo a Nagy et al. (2000), la solución analítica de la ecuación 2.31 está dada por:

$$U(P) = \left| \left| \left| \begin{aligned} &xy \ln(z+r) + yz \ln(x+r) + zx \ln(y+r) \\ & - \frac{x^2}{2} \tan^{-1} \left[\frac{yz}{xr} \right] - \frac{y^2}{2} \tan^{-1} \left[\frac{zx}{yr} \right] - \frac{z^2}{2} \tan^{-1} \left[\frac{xy}{zr} \right] \right| \right|_{x_1, y_1, z_1}^{x_2, y_2, z_2} \quad (2.32)$$

En cuanto a las componentes del vector de aceleración gravitacional, Plouff (1976) presentó una solución analítica para la componente vertical, g_z , dada por:

$$\begin{aligned}
g_z &= \frac{\partial}{\partial z} \left[\gamma \rho \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{1}{\sqrt{x^2 + y^2 + z^2}} dx dy dz \right] \\
&= \gamma \rho \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{z}{\sqrt{x^2 + y^2 + z^2}^3} dx dy dz \\
&= \gamma \rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[z_k \tan^{-1} \left(\frac{x_i y_j}{z_k R_{ijk}} \right) - x_i \ln(R_{ijk} + y_j) - y_j \ln(R_{ijk} + x_i) \right] \quad (2.33)
\end{aligned}$$

$$\text{donde: } \begin{cases} R_{ijk} = \sqrt{x_i^2 + y_j^2 + z_k^2} \\ S = S_i S_j S_k \\ S_1 = -1 \\ S_2 = 1 \end{cases}$$

Con base en esta solución (Ec. 2.33), la formulación para las componentes g_x y g_y se hace directamente, a través de permutación cíclica, obteniéndose:

$$g_x = \gamma \rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[x_i \tan^{-1} \left(\frac{y_j z_k}{x_i R_{ijk}} \right) - y_j \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + y_j) \right] \quad (2.34)$$

$$g_y = \gamma \rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[y_j \tan^{-1} \left(\frac{x_i z_k}{y_j R_{ijk}} \right) - x_i \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + x_i) \right] \quad (2.35)$$

Finalmente, derivando espacialmente las ecuaciones 2.33, 2.34 y 2.35, de acuerdo a la definición del tensor gravitacional (Ec. 2.9), se obtienen las expresiones para determinar las componentes tensoriales debidas al prisma rectangular:

$$\begin{aligned}
T_{xx} &= \frac{\partial g_x}{\partial x} \\
&= \gamma \rho \frac{\partial}{\partial x_i} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[x_i \tan^{-1} \left(\frac{y_j z_k}{x_i R_{ijk}} \right) - y_j \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + y_j) \right] \\
\therefore T_{xx} &= \gamma \rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[\tan^{-1} \left(\frac{y_j z_k}{x_i R_{ijk}} \right) \right] \quad (2.36)
\end{aligned}$$

$$\begin{aligned}
T_{xy} &= \frac{\partial g_x}{\partial y} \\
&= \gamma\rho \frac{\partial}{\partial y_j} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[x_i \tan^{-1} \left(\frac{y_j z_k}{x_i R_{ijk}} \right) - y_j \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + y_j) \right] \\
\therefore T_{xy} &= -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S [\ln(R_{ijk} + z_k)] \tag{2.37}
\end{aligned}$$

$$\begin{aligned}
T_{xz} &= \frac{\partial g_x}{\partial z} \\
&= \gamma\rho \frac{\partial}{\partial z_k} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[x_i \tan^{-1} \left(\frac{y_j z_k}{x_i R_{ijk}} \right) - y_j \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + y_j) \right] \\
\therefore T_{xz} &= -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S [\ln(R_{ijk} + y_j)] \tag{2.38}
\end{aligned}$$

$$\begin{aligned}
T_{yy} &= \frac{\partial g_y}{\partial y} \\
&= \gamma\rho \frac{\partial}{\partial y_j} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[y_j \tan^{-1} \left(\frac{x_i z_k}{y_j R_{ijk}} \right) - x_i \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + x_i) \right] \\
\therefore T_{yy} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[\tan^{-1} \left(\frac{x_i z_k}{y_j R_{ijk}} \right) \right] \tag{2.39}
\end{aligned}$$

$$\begin{aligned}
T_{yz} &= \frac{\partial g_y}{\partial z} \\
&= \gamma\rho \frac{\partial}{\partial z_k} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[y_j \tan^{-1} \left(\frac{x_i z_k}{y_j R_{ijk}} \right) - x_i \ln(R_{ijk} + z_k) - z_k \ln(R_{ijk} + x_i) \right] \\
\therefore T_{yz} &= -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S [\ln(R_{ijk} + x_i)] \tag{2.40}
\end{aligned}$$

$$\begin{aligned}
T_{zz} &= \frac{\partial g_z}{\partial z} \\
&= \gamma\rho \frac{\partial}{\partial z_k} \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[z_k \tan^{-1} \left(\frac{x_i y_j}{z_k R_{ijk}} \right) - x_i \ln(R_{ijk} + y_j) - y_j \ln(R_{ijk} + x_i) \right] \\
\therefore T_{zz} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S \left[\tan^{-1} \left(\frac{x_i y_j}{z_k R_{ijk}} \right) \right] \tag{2.41}
\end{aligned}$$

Del mismo modo que se hizo para el cuerpo esférico, las formas que tienen las superficies de potencial, componentes del vector y del tensor gravitacional de un cuerpo prismático rectangular en mallas de observación equiespaciadas, se pueden ver en las Figuras 2.9, 2.10 y 2.11:

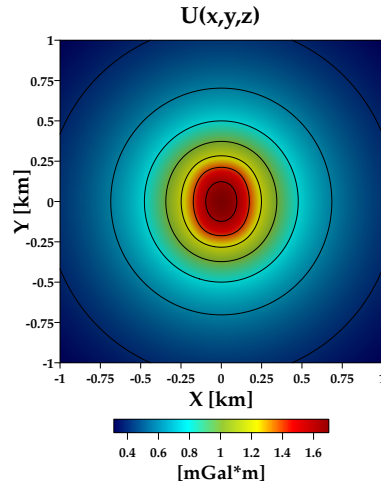


FIGURA 2.9: Superficie de potencial gravitacional generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X, Y y Z, respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X: -150 a 150 [m] , Y: -225 a 225 [m] y Z: 25 a 525 [m] .

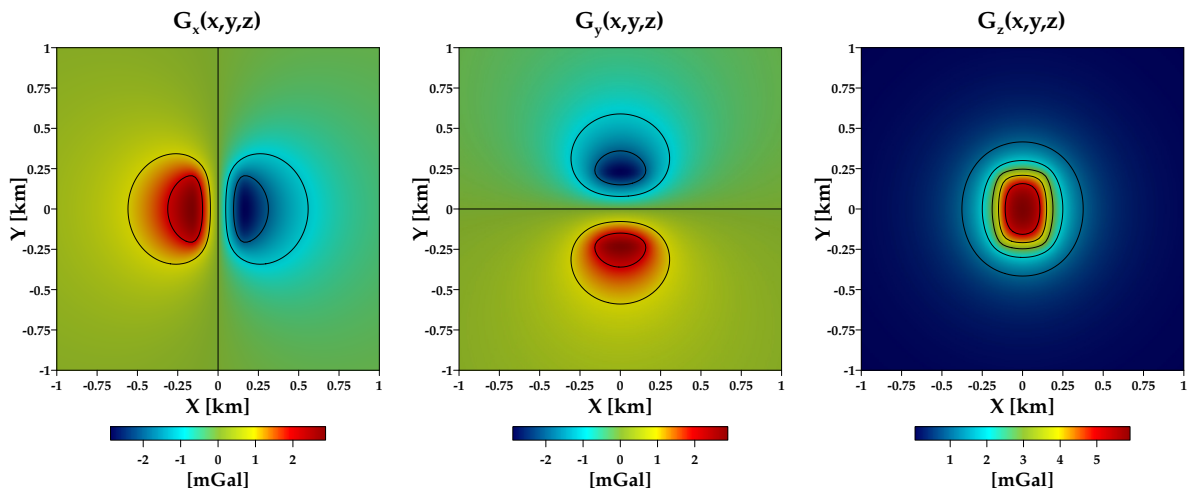


FIGURA 2.10: Superficies de las componentes g_x , g_y y g_z del vector de aceleración gravitacional, generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X, Y y Z, respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X: -150 a 150 [m] , Y: -225 a 225 [m] y Z: 25 a 525 [m] .

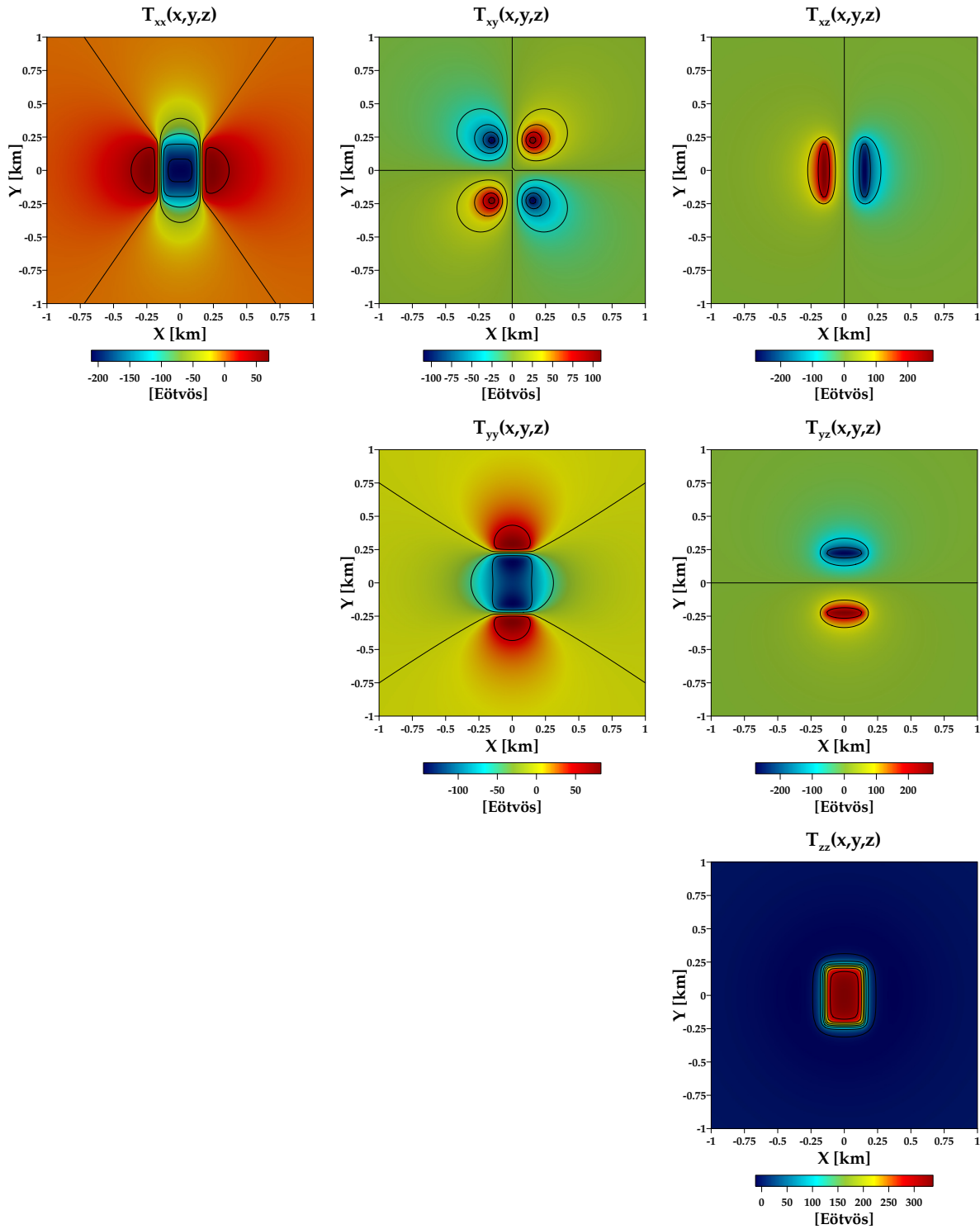


FIGURA 2.11: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por un cuerpo prismático rectangular con densidad $\rho = 1000.0 \text{ [kg/m}^3\text{]}$ y dimensiones de $300 \text{ [m]} \times 450 \text{ [m]} \times 500 \text{ [m]}$ en X , Y y Z , respectivamente, observado en una malla regular de 101×101 puntos. La ubicación del prisma está dada por las coordenadas de sus extremos: X : -150 a 150 [m] , Y : -225 a 225 [m] y Z : 25 a 525 [m] .

2.1.3 Modelado de estructuras complejas

El modelado gravimétrico de cualquier cuerpo depende de su forma y densidad, dejando el problema en función de una integral volumétrica (Ec. 2.4). La integral existe para fuentes con geometría simple y densidad uniforme, como la esfera y prisma rectangular presentados. Sin embargo, cuando el cuerpo tiene una forma irregular o una distribución de densidad que no se pueda describir analíticamente², la integral en cuestión podría no tener una solución analítica, haciendo imposible el modelado de dicho cuerpo por este medio.

Una solución al problema de inexistencia de la solución analítica para una fuente de campo de interés, es su discretización a través de un conjunto de bloques individuales cuyas soluciones sean conocidas y se puedan utilizar para realizar el cálculo de los efectos gravitacionales de cada bloque en cada punto de observación, sumando posteriormente todos sus efectos.

Ensamblajes de bloques elementales

El modelado gravimétrico de estructuras complejas a través de ensamblajes de cuerpos simples no es nuevo. Entre los trabajos más destacados se pueden mencionar los siguientes: Talwani e Ewing (1960) utilizaron un ensamblaje de discos poligonales horizontales con densidad uniforme para aproximar la forma del cuerpo en cuestión. Plouff (1976) propuso la utilización de láminas en lugar de discos en este tipo de ensamblajes, llegando a una solución general para un prisma poligonal, cuyo caso particular es el prisma mencionado en el subcapítulo 2.1.2. Barnett (1976) desarrolló un método para el cálculo de la anomalía gravimétrica de cuerpos con densidad homogénea y formas arbitrarias, estimadas a través de poliedros con facetas triangulares que sentó las bases para el modelado gravimétrico de cuerpos poliédricos con facetas poligonales (e.g. Okabe, 1979; Singh y Guptasarma, 2001; D'Urso, 2014). Bhattacharyya (1978) presentó una solución similar a la de Plouff (1976) para calcular la anomalía gravimétrica debida a un cuerpo prismático rectangular con densidad uniforme, sugiriendo que un cuerpo con densidad heterogénea podría descomponerse en pequeños bloques rectangulares de ese tipo. Garcia-Abdeslem (1992) propuso un método para calcular el efecto gravitacional causado por un prisma rectangular con variación de densidad en función de la profundidad y Nagy et al. (2000) publicaron las expresiones correspondientes al potencial gravitacional y sus derivadas hasta de tercer orden, para un cuerpo prismático rectangular con densidad uniforme.

En este trabajo de tesis, el problema del modelado de estructuras complejas se abordó a través de ensamblajes de prismas rectangulares, utilizando las soluciones de las cantidades gravitacionales mostradas en el subcapítulo 2.1.2, dadas por las ecuaciones 2.33 a 2.35 para las componentes del vector de atracción gravitacional y las ecuaciones 2.36 a 2.41 para las componentes del tensor de gradientes gravitacionales.

²En adelante se denominará bajo el término genérico *estructuras complejas*, a este tipo de cuerpos o fuentes de campo gravitacional.

La idea detrás de la utilización de ensambles de prismas para modelar una estructura compleja, es considerar que esta se puede aproximar como un agregado de prismas con densidad uniforme, que son lo suficientemente pequeños como para representar de manera adecuada, tanto su forma como su densidad. La estructura compleja se discretiza en M bloques elementales cuya respuesta gravimétrica es observada en uno o más puntos y está dada por la suma de todos y cada uno de los efectos causados por los M prismas (Ec. 2.42):

$$g_i(P) = \sum_{j=1}^M g_i^j(P) \quad (2.42)$$

En la ecuación 2.42, $g_i^j(P)$ representa la respuesta gravimétrica (cualquier componente del vector de atracción gravitacional o del tensor de gradientes) observada en el punto P , debida al j -ésimo prisma del ensamble (Figura 2.12):

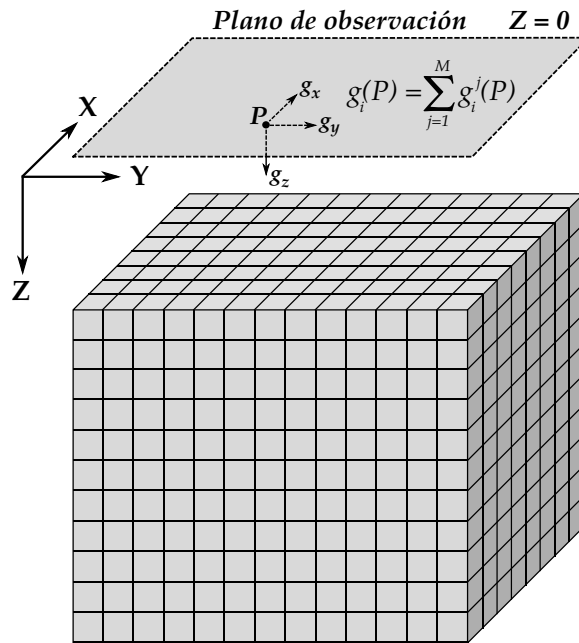


FIGURA 2.12: Representación de la discretización de un medio en M prismas rectangulares y el cálculo de la respuesta gravimétrica g_i en el punto de observación P .

El uso de prismas rectangulares como bloques de construcción de estructuras complejas en un sistema de referencia cartesiano 3D es relativamente fácil de implementar, debido a los elementos requeridos por la solución analítica de las cantidades gravitacionales por calcular (Figura 2.8 y Ecs. 2.33 a 2.41), que son esencialmente cuatro de sus ocho vértices, punto de observación y densidad.

Por otro lado, la cantidad de información requerida para realizar la discretización del medio, es relativamente poca, ya que un ensamble regular como el mostrado en la Figura 2.12 se logrará discretizando en las direcciones X , Y y Z , de la siguiente manera:

Dirección X :

$$x_i = x_0 + (i - 1) \Delta x; \quad i = 1, 2, \dots, N_x + 1$$

Dirección Y :

$$y_j = y_0 + (j - 1) \Delta y; \quad j = 1, 2, \dots, N_y + 1$$

Dirección Z :

$$z_k = z_0 + (k - 1) \Delta z; \quad k = 1, 2, \dots, N_z + 1$$

donde:

- x_0, y_0 y z_0 : Son las coordenadas iniciales del ensamble en las direcciones X , Y y Z .
- $\Delta x, \Delta y$ y Δz : Son las distancias entre vértices contiguos de prismas del ensamble en estas mismas direcciones (longitud de arista de cada prisma o intervalos de muestreo del ensamble).
- N_x, N_y, N_z : Número de prismas del ensamble en cada dirección.

2.2 Procesamiento e Interpretación de Datos en el Modelado

Se sabe que las anomalías gravimétricas (componentes vectoriales y/o tensoriales) son generadas por la presencia de cuerpos o estructuras en el subsuelo con densidad diferente a su entorno. Debido a esto, se puede intuir que las mallas de anomalías pueden ser utilizadas para ubicar espacialmente las estructuras que las generan y obtener información suficiente para construir modelos del subsuelo que expliquen su presencia. Esto representa un reto, dado que las anomalías observadas son resultado de la superposición de los campos generados por diferentes fuentes, los datos se encuentran contaminados por ruido y el problema gravimétrico no tiene soluciones únicas. Por lo tanto, es indispensable aplicar procesos a estas mallas con la finalidad de atenuar el ruido presente, realzar efectos de las fuentes y generar información útil que sirva para inferir la forma y profundidad de las estructuras causantes de las anomalías.

Los métodos de procesamiento de anomalías que se han desarrollado para ubicar espacialmente las fuentes, se pueden clasificar en diferentes familias, siendo especialmente útiles en la interpretación y el modelado, los métodos de realce de anomalías y bordes e interpretación semiautomática. A continuación se describen brevemente los métodos que se eligieron en este trabajo de tesis.

2.2.1 Métodos de realce de anomalías y bordes

Los métodos de realce de anomalías y bordes seleccionados para su aplicación en mallas de anomalía y gradiometría gravimétrica, fueron los siguientes:

Métodos aplicados a mallas de anomalía gravimétrica (G_z):

1. *Continuación Analítica.*
2. *Gradiente Horizontal.*
3. *Derivada Vertical.*
4. *Amplitud de la Señal Analítica.*
5. *Ángulo de Inclinación de Gradientes.*

Métodos aplicados a mallas de componentes del tensor gravitacional ($FTGG$):

1. *Invariantes.*
2. *Señales Analíticas Direccionales y Función Detectora de Bordes.*
3. *Función Detectora de Bordes Mejorada.*

Realce de anomalías y bordes en mallas de anomalía gravimétrica

Continuación Analítica

La continuación analítica es un operador que permite transformar el campo potencial observado en una superficie, al campo que sería medido en otra que podría estar más cercana o lejana a las masas o fuentes de campo, siempre y cuando, estas no se encuentren entre ambas superficies (Telford et al., 1990).

La continuación analítica se basa en la tercera identidad de Green para funciones potenciales armónicas (Blakely, 1996), transformando la función potencial de una superficie de observación plana a otra:

$$U(x, y, z_0 - \Delta z) = \frac{\Delta z}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{U(x', y', z_0)}{\left[(x - x')^2 + (y - y')^2 + \Delta z^2\right]^{\frac{3}{2}}} dx' dy' \quad (2.43)$$

donde:

z_0 : Es la superficie plana en la que fue observado el campo potencial U .

Δz : Es la separación entre la superficie z_0 y la superficie de continuación.

x, y : Coordenadas X y Y del campo U en el plano de observación (z_0).

x', y' : Coordenadas X y Y del campo U en el plano de continuación ($z_0 - \Delta z$).

(Se considera un sistema de coordenadas derecho, con eje Z dirigido hacia abajo)

La ecuación 2.43 representa una integral de convolución en dos dimensiones (2D), del campo potencial $U(x, y, z_0)$ con la función $\psi_{CA}(x, y, \Delta z)$:

$$\psi_{CA}(x, y, \Delta z) = \frac{\Delta z}{2\pi} \frac{1}{(x^2 + y^2 + \Delta z^2)^{\frac{3}{2}}}$$

por lo tanto, puede expresarse en el dominio del número de onda como:

$$\mathcal{F}\{U_{CA}\} = \mathcal{F}\{U\} \mathcal{F}\{\psi_{CA}\}$$

en la que $\mathcal{F}\{U_{CA}\}$ es la transformada de Fourier del campo continuado analíticamente al plano $z = z_0 - \Delta z$ y $\mathcal{F}\{\psi_{CA}\}$ es la función de transferencia de la continuación analítica:

$$\mathcal{F}\{\psi_{CA}\} = e^{-\Delta z|k|}$$

donde:

$$\begin{aligned} |k| &= \sqrt{p^2 + q^2} \\ p &= 2\pi k_x \text{ (números de onda en dirección X)} \\ q &= 2\pi k_y \text{ (números de onda en dirección Y)} \end{aligned}$$

El signo del parámetro Δz determina si la continuación analítica se hace sobre un plano más alejado (ascendente) o cercano a las fuentes (descendente):

$$U(x, y, z \pm \Delta z) = \begin{cases} \mathcal{F}^{-1} \{ \mathcal{F} [U] e^{-\Delta z |k|} \} & \text{Continuación Ascendente} \\ \mathcal{F}^{-1} \{ \mathcal{F} [U] e^{\Delta z |k|} \} & \text{Continuación Descendente} \end{cases} \quad (2.44)$$

La continuación analítica ascendente es un importante proceso de realce de anomalías de las componentes vectoriales y tensoriales gravitacionales, que actúa como un filtro pasabajas, enfatizando las anomalías de grandes longitudes de onda generadas por fuentes profundas, a expensas de anomalías de fuentes superficiales y ruido de longitud de onda corta (Hinze et al., 2013), mientras que la continuación analítica descendente es un filtro inestable que no suele utilizarse por los efectos indeseados que introduce en los datos.

Gradiente Horizontal

El gradiente horizontal, definido por Cordell (1979), es un proceso utilizado para realzar valores en datos de anomalía relacionados con bordes de fuentes de campo gravitacional anómalo en el subsuelo en las que se involucren cambios laterales de densidad considerables como fallas o intrusiones.

Este proceso tiende a exhibir sus máximos directamente sobre los bordes de las fuentes causantes de las anomalías, a través de la formación de crestas angostas sobre los contrastes abruptos de densidad, por lo que puede ser utilizado para aproximar bordes en mallas de datos (Blakely y Simpson, 1986). Sin embargo, sus resultados deben interpretarse con precaución, ya que estos máximos no se ubicarán directamente sobre los límites de las estructuras cuando los contactos no sean verticales o semiverticales, cuando se tengan contactos muy cercanos uno del otro (cuerpos tabulares delgados) y cuando la superficie de observación no sea plana.

El gradiente horizontal, $GH(x, y)$, aplicado a una malla regular de datos de anomalía gravimétrica $g(x, y)$, se define como:

$$GH(x, y) = \sqrt{\left[\frac{\partial}{\partial x} g(x, y) \right]^2 + \left[\frac{\partial}{\partial y} g(x, y) \right]^2} \quad (2.45)$$

donde $\frac{\partial}{\partial x}g(x, y)$ y $\frac{\partial}{\partial y}g(x, y)$ son mallas de derivadas o gradientes de la anomalía en las direcciones X y Y , respectivamente.

Las mallas de derivadas direccionales se pueden aproximar en el dominio espacial a través de operadores de diferencias, o en el dominio del número de onda, a través de la propiedad de derivación de la Transformada de Fourier.

Derivada Vertical

La derivada vertical (también denominada gradiente vertical) es uno de los procesos más utilizados en la exploración por el método gravimétrico, debido a que uno de sus efectos es la mejora en definición de anomalías generadas por fuentes que se interfieren mutuamente (Evjen, 1936; Marson y Klingele, 1993), lo cual se refleja en un incremento de resolución en métodos de realce de bordes en la interpretación estructural.

A diferencia de las derivadas direccionales en X y Y , la derivada vertical únicamente se puede calcular en el dominio del número de onda, a través de diferencias entre la anomalía observada en un plano, $g(x, y, z)$, y la anomalía continuada analíticamente a un plano diferente, $g(x, y, z - \Delta z)$ (Blakely, 1996), resultando en:

$$\begin{aligned} \frac{\partial}{\partial z}g(x, y) &= \lim_{\Delta z \rightarrow 0} \frac{g(x, y, z) - g(x, y, z - \Delta z)}{\Delta z} \\ \Rightarrow \mathcal{F} \left\{ \frac{\partial}{\partial z}g(x, y) \right\} &= \lim_{\Delta z \rightarrow 0} \frac{1 - e^{-|k|\Delta z}}{\Delta z} G(p, q) \\ &= |k|G(p, q) \\ \therefore \frac{\partial}{\partial z}g(x, y) &= \mathcal{F}^{-1} \{ |k|G(p, q) \} \end{aligned} \quad (2.46)$$

donde:

$$\begin{aligned} G(p, q) &= \mathcal{F} \{ g(x, y) \} \\ |k| &= \sqrt{p^2 + q^2} \end{aligned}$$

Cabe mencionar que este método se puede generalizar para estimar las derivadas verticales hasta un orden n :

$$\begin{aligned} \mathcal{F} \left\{ \frac{\partial^n}{\partial z^n}g(x, y) \right\} &= |k|^n G(p, q) \\ \therefore \frac{\partial^n}{\partial z^n}g(x, y) &= \mathcal{F}^{-1} \{ |k|^n G(p, q) \} \end{aligned} \quad (2.47)$$

Amplitud de la Señal Analítica

La Señal Analítica surgió como un concepto originalmente empleado en el procesamiento y análisis de señales, relacionado con el cálculo de la frecuencia instantánea (Boas-hash, 2015). En geofísica, a principios de la década de 1970, Nabighian fue el primero en aplicar este concepto a perfiles de datos magnéticos (Nabighian, 1972), demostrando que para un campo potencial $\phi(x)$ observado a lo largo de un perfil en dirección X, sin variaciones en altura ($z = cte$) y considerando una fuente de campo 2D (extendida infinitamente en dirección perpendicular al plano XZ), las derivadas direccionales $\phi_x = \frac{\partial\phi}{\partial x}$ y $\phi_z = \frac{\partial\phi}{\partial z}$, forman un par de transformadas de Hilbert, por lo que la señal analítica del campo potencial $\phi(x)$ se puede escribir como:

$$A(x) = \phi_x + i\phi_z$$

y su amplitud será:

$$|A(x)| = \sqrt{\phi_x^2 + \phi_z^2}$$

Para el caso 3D, Nabighian (1984) extendió las relaciones que guardan las transformadas de Hilbert en 2D entre derivadas direccionales de los campos potenciales, llegando a la generalización de transformadas de Hilbert en 3D y Roest et al. (1992) demostraron que la señal analítica en 3D de un campo potencial $\phi(x, y)$ puede escribirse como:

$$A(x, y) = \frac{\partial\phi}{\partial x}\hat{x} + \frac{\partial\phi}{\partial y}\hat{y} + i\frac{\partial\phi}{\partial z}\hat{z} \quad (2.48)$$

donde \hat{x} , \hat{y} y \hat{z} son vectores unitarios en las direcciones X, Y y Z, respectivamente.

A partir de la expresión 2.48, se define la *Amplitud de la Señal Analítica 3D* como:

$$|A(x, y)| = \sqrt{\left[\frac{\partial\phi}{\partial x}\right]^2 + \left[\frac{\partial\phi}{\partial y}\right]^2 + \left[\frac{\partial\phi}{\partial z}\right]^2} \quad (2.49)$$

La importancia de la señal analítica en 3D radica en que la ubicación de sus máximos depende, en general, de la profundidad de sepultamiento, tamaño e inclinación de los contactos o bordes de los cuerpos causantes de anomalías, tendiendo a ubicarse directamente sobre sus límites, en la medida en que estos se aproximen a una forma vertical y su profundidad de sepultamiento sea pequeña, con respecto a su extensión vertical (Li, 2006), independientemente de los efectos de interferencia debidos a fuentes cercanas.

En ocasiones, la señal analítica no realza suficientemente bien los bordes de fuentes de campo. Cuando esto sucede, un método que ayuda a situar con mayor precisión los máximos de amplitud de la señal analítica, es la Amplitud de la Señal Analítica Mejorada, propuesta por Hsu et al. (1996), definida como:

$$|A_n(x, y)| = \sqrt{\left[\frac{\partial^n G_x}{\partial z^n}\right]^2 + \left[\frac{\partial^n G_y}{\partial z^n}\right]^2 + \left[\frac{\partial^n G_z}{\partial z^n}\right]^2} \quad (2.50)$$

donde G es la anomalía del campo potencial observado y:

$$G_x = \frac{\partial G}{\partial x}, G_y = \frac{\partial G}{\partial y} \text{ y } G_z = \frac{\partial G}{\partial z}$$

Ángulo de Inclinación de Gradientes

El ángulo de inclinación de gradientes es un método propuesto por Miller y Singh (1994) que consiste en calcular el ángulo que se forma entre la componente vertical y la componente horizontal del gradiente del campo potencial, f , observado:

$$AIG(x, y) = \tan^{-1} \left[\frac{\frac{\partial f}{\partial z}}{\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}} \right] \quad (2.51)$$

Este proceso funciona como un “ecualizador” de anomalías presentes en los datos, normalizando sus amplitudes al rango $\pm \frac{\pi}{2}$, realizando los contrastes entre anomalías poco perceptibles por sus diferencias en amplitud (Verduzco et al., 2004).

La importancia y utilidad de este proceso radica en que, al aplicarse en datos del gradiente vertical de anomalía gravimétrica, exhibe un valor positivo sobre las fuentes de campo (cuerpos o estructuras con un contraste de densidad positivo), cero sobre los límites de dichas fuentes y negativo en cualquier otro sitio, independientemente de la profundidad de dichas fuentes, por lo cual se podría utilizar como herramienta complementaria en el mapeo estructural, proporcionando información de la extensión lateral de las fuentes, como un método auxiliar en el realce de bordes (Ferreira et al., 2013), e incluso como un método de estimación de profundidad (Cooper, 2011; Oruç, 2011; Salem et al., 2013).

Realce de anomalías y bordes en mallas de gradiometría gravimétrica

Invariantes

Sabiendo que las componentes del tensor gravimétrico se pueden agrupar dentro de una matriz real simétrica T_{ij} (Ec. 2.17), ésta se puede diagonalizar con eigenvalores reales y eigenvectores mutuamente ortogonales, a través de la transformación de coordenadas:

$$V^T T V = \Lambda$$

Donde: $V = [v_1 \ v_2 \ v_3]$ es un arreglo que contiene los eigenvectores de la transformación (vectores columna) y λ_1, λ_2 y λ_3 , denominados eigenvalores, son los elementos de la matriz diagonalizada:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Pedersen y Rasmussen (1990) propusieron que bajo cualquier transformación de coordenadas, T_{ij} tiene tres invariantes, de los cuales, dos reflejan la forma de las fuentes que causan las anomalías del $FTGG$:

$$\begin{aligned} I_0 &= Tr(T_{ij}) \\ &= T_{xx} + T_{yy} + T_{zz} \end{aligned} \quad (2.52)$$

$$I_1 = T_{xx}T_{yy} + T_{yy}T_{zz} + T_{xx}T_{zz} - (T_{xy}^2 + T_{yz}^2 + T_{xz}^2) \quad (2.53)$$

$$\begin{aligned} I_2 &= det(T_{ij}) \\ &= T_{xx}(T_{yy}T_{zz} - T_{yz}^2) + T_{xy}(T_{yz}T_{xz} - T_{xy}T_{zz}) + T_{xz}(T_{xy}T_{yz} - T_{xz}T_{yy}) \end{aligned} \quad (2.54)$$

El polinomio característico, cuyas raíces son los eigenvalores de la transformación, es:

$$\begin{aligned} \lambda^3 - I_0\lambda^2 + I_1\lambda - I_2 &= 0 \\ \Rightarrow \lambda^3 + I_1\lambda - I_2 &= 0 \end{aligned} \quad (2.55)$$

Por otro lado, es importante mencionar que los invariantes I_0, I_1 e I_2 son arreglos con las mismas dimensiones que el conjunto de observaciones de las componentes tensoriales: 1D para levantamientos en forma de perfiles y 2D para levantamientos en mallas.

Regresando al polinomio característico (Ec. 2.55), para que todos sus eigenvalores sean reales, se debe cumplir la condición: $I_1 \leq 0$, lo que generará una malla de valores negativos para dicho invariante, que tendrá relativamente poca utilidad práctica en la interpretación. Debido a esto, usualmente se visualizan las mallas de valor absoluto de este invariante, $|I_1|$, mientras que el invariante I_2 suele visualizarse sin modificación alguna.

Señales Analíticas Direccionales y Función Detectora de Bordes

Presentadas por Beiki (2010), las señales analíticas direccionales son la extensión del concepto de señal analítica al *FTGG*, considerando que las derivadas horizontales y verticales de las componentes del vector de aceleración gravitacional son pares transformados de Hilbert y se pueden formar tres señales analíticas con ellas: A_x , A_y y A_z , asociadas a las direcciones X, Y y Z, cuyas amplitudes tienden a realzar bordes en dichas direcciones y por lo tanto, pueden servir como procesos de realce de bordes laterales y en planta, de fuentes de anomalía gravimétrica:

$$\begin{aligned} A_x(x, y, z) &= (T_{xx}) \hat{x} + (T_{xy}) \hat{y} + i (T_{xz}) \hat{z} \\ \Rightarrow |A_x(x, y, z)| &= \sqrt{(T_{xx})^2 + (T_{xy})^2 + (T_{xz})^2} \end{aligned} \quad (2.56)$$

$$\begin{aligned} A_y(x, y, z) &= (T_{xy}) \hat{x} + (T_{yy}) \hat{y} + i (T_{yz}) \hat{z} \\ \Rightarrow |A_y(x, y, z)| &= \sqrt{(T_{xy})^2 + (T_{yy})^2 + (T_{yz})^2} \end{aligned} \quad (2.57)$$

$$\begin{aligned} A_z(x, y, z) &= (T_{xz}) \hat{x} + (T_{yz}) \hat{y} + i (T_{zz}) \hat{z} \\ \Rightarrow |A_z(x, y, z)| &= \sqrt{(T_{xz})^2 + (T_{yz})^2 + (T_{zz})^2} \end{aligned} \quad (2.58)$$

Aunado a estas señales analíticas, Beiki (2010) también propuso la utilización de las derivadas direccionales de estas, basándose en el trabajo de Debeglia y Corpel (1997), combinando las derivadas verticales de A_x y A_y para formar un proceso denominado *Función Detectora de Bordes*:

A partir de $|A_{x,z}| = \frac{\partial |A_x(x, y, z)|}{\partial z}$ y $|A_{y,z}| = \frac{\partial |A_y(x, y, z)|}{\partial z}$, la función detectora de bordes, $|DB|$, se define como:

$$|DB| = \sqrt{|A_{x,z}|^2 + |A_{y,z}|^2} \quad (2.59)$$

Función Detectora de Bordes Mejorada

Este proceso está basado en la utilización de los eigenvalores del Tensor de Curvatura de Gradientes Gravimétricos (CGGT) (Oruç et al., 2013) como detectores de bordes, realizando una modificación, de acuerdo a Zhou et al. (2013), que incluye el uso de la malla de anomalía gravimétrica en la expresión que permite determinar el mayor de los eigenvalores del CGGT:

La matriz que contiene los elementos del tensor CGGT se define como:

$$CGGT = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix}$$

el mayor de sus eigenvalores (λ_1) es:

$$\lambda_1 = \frac{1}{2} \left[T_{xx} + T_{yy} + \sqrt{(T_{xx} - T_{yy})^2 + 4(T_{xy})^2} \right]$$

y finalmente, la **Función Detectora de Bordes Mejorada (DBM)**, que es obtenida a partir de una modificación a la expresión para obtener el eigenvalor λ_1 y cuyo valor cero tiende a ubicarse directamente sobre los contornos de las fuentes de campo, es:

$$DBM = \frac{1}{2} \left[T_{xx}g_z + T_{yy}g_z + \sqrt{(T_{xx}g_z - T_{yy}g_z)^2 + 4(T_{xy}g_z)^2} \right] \quad (2.60)$$

2.2.2 Interpretación semiautomática de anomalías

El método de interpretación semiautomática considerado en este trabajo de tesis es la Deconvolución de Euler 3D, aplicada a mallas de anomalía gravimétrica (G_z) y gradiometría gravimétrica ($FTGG$).

Deconvolución de Euler 3D

La deconvolución de Euler fue propuesta inicialmente por Thompson (1982) como una técnica para determinar la posición espacial de fuentes de campo magnético a partir de mediciones adquiridas sobre un perfil. El método fue extendido a 3D por Reid et al. (1990) y desde entonces se ha utilizado como una herramienta auxiliar en la interpretación de datos gravimétricos y magnéticos.

Este método está basado en la ecuación de homogeneidad de Euler:

$$\vec{r} \cdot \nabla f = n f \quad (2.61)$$

Toda función f que cumpla con la ecuación 2.61 será homogénea, con grado de homogeneidad n . De acuerdo a Blakely (1996) y Stavrev y Reid (2006), los potenciales gravitacional y magnético, así como sus derivadas espaciales, son funciones homogéneas, razón por la cual, este método se puede extender a datos vectoriales y tensoriales de estos dos campos potenciales.

A continuación se describe el método de deconvolución de Euler 3D “convencional” (aplicado a la malla G_z):

Considerando que se tiene un dato de anomalía de un campo potencial, Δf , en el punto $P(x, y)$, situado en una malla de observación en el plano $z = 0$ y que la anomalía es debida a

una fuente puntual ubicada en el punto de coordenadas (x_0, y_0, z_0) (Figura 2.13), la ecuación de Euler (2.61) para este caso será:

$$\begin{aligned} \left[(x - x_0) \hat{i} + (y - y_0) \hat{j} + (-z_0) \hat{k} \right] \cdot \left[\frac{\partial \Delta f}{\partial x} \hat{i} + \frac{\partial \Delta f}{\partial y} \hat{j} + \frac{\partial \Delta f}{\partial z} \hat{k} \right] &= -N \Delta f \\ \Rightarrow (x - x_0) \frac{\partial \Delta f}{\partial x} + (y - y_0) \frac{\partial \Delta f}{\partial y} + (-z_0) \frac{\partial \Delta f}{\partial z} &= -N \Delta f \end{aligned} \quad (2.62)$$

donde: x_0, y_0, z_0 y N son incógnitas.

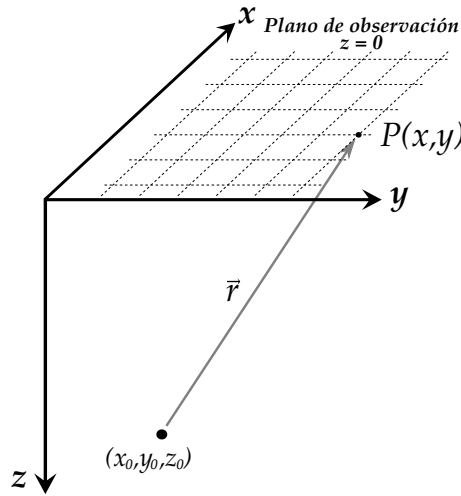


FIGURA 2.13: Fuente puntual ubicada en x_0, y_0, z_0 , cuya anomalía es observada en el punto $P(x, y)$, de la malla situada sobre el plano $z = 0$. El vector \vec{r} define la posición de la fuente con respecto a $P(x, y)$.

La expresión 2.62 se puede resolver a través de un sistema de ecuaciones, si además de la malla de anomalía, se conocen por observación o mediante cálculo, mallas de los gradientes en direcciones X, Y y Z de la anomalía y se seleccionan al menos cuatro puntos de estas, pero existen dos limitaciones claras:

1. Las anomalías raramente se deben a fuentes puntuales.
2. Las anomalías difícilmente se presentan en forma aislada, existiendo interferencia de efectos debida a fuentes cercanas.

El problema de interferencia se minimiza considerando que los datos de anomalía observada, Δf , se encuentran bajo la influencia de un campo B , denominado *Campo Ambiental*, cuya amplitud es desconocida, pero constante en una región relativamente pequeña dentro del área estudiada (ventana) y sus efectos se suman a los de la anomalía, de tal modo que en cada uno de los v puntos de esa ventana, se tendrá el campo ambiental añadido a la anomalía:

$$f(x, y) = \Delta f(x, y) + B$$

$$\text{donde: } \sigma_z = \sqrt{[Cov\ m^{est}]_{3,3}}$$

$$\text{y: } [Cov\ m^{est}] = \left[\frac{1}{v} \sum_{i=1}^v (d_i - [Gm^{est}]_i)^2 \right] [G^T G]^{-1}$$

$[Cov\ m^{est}]$ es la matriz de covarianza de la estimación del vector m

Se repite este procedimiento en tantas ventanas como sea posible, desplazándolas a través de las mallas de anomalía y sus gradientes, de forma similar a una ventana convolutiva y las soluciones aceptadas se pueden graficar en mapas con símbolos de tamaño proporcional a las profundidades o de colores distintivos, lo que es de gran ayuda en la interpretación estructural del subsuelo.

Por otro lado, la limitación del método relacionada con la forma de las fuentes se puede atacar asignando valores al índice estructural, N , sabiendo que es un indicador de la forma en la que el campo potencial observado varía con la distancia³, utilizando como referencia, tablas con dichos índices, como la Tabla 2.1, basada en el trabajo de Stavrev y Reid (2006):

TABLA 2.1: Índices estructurales del potencial gravimétrico y sus derivadas espaciales hasta de segundo orden, para diferentes modelos utilizados como base en la interpretación a través de la deconvolución de Euler (Modificada de Stavrev y Reid, 2006).

Fuente/Modelo	Potencial U	Componentes vectoriales $g_i = \frac{\partial U}{\partial x_i} = U'_{i_i}$ ($x_i = x, y, z$)	Componentes tensoriales $T_{ij} = \frac{\partial^2 U}{\partial x_j \partial x_i} = (U'_{i_i})'_{j_j}$ ($x_i, x_j = x, y, z$)
Masa puntual / Esfera	1	2	3
Línea de masas / Cilindro	0	1	2
Lámina delgada semi-infinita / Dique, sill, contacto, falla	-1	0	1
Contacto con extensión infinita en profundidad	-2	-1	0

³Ejemplo: Si el campo varía con el cuadrado de la distancia, como sucede con la aceleración gravitacional en el caso de una fuente puntual, $N = 2$

Deconvolución de Euler del Tensor Gravimétrico

La deconvolución de Euler del tensor gravimétrico (Zhang et al., 2000) es la extensión del método de la deconvolución de Euler 3D, a todas las componentes del vector gravimétrico y sus gradientes. Su procedimiento es el mismo, basado en la solución de un sistema de ecuaciones con datos de una ventana convolutiva, aceptando o rechazando soluciones con base en la calidad de las estimaciones:

A partir de la ecuación que define la deconvolución de Euler 3D (Ec. 2.63), se pueden plantear tres ecuaciones de homogeneidad similares, en las que se requieren las tres componentes vectoriales (G_x, G_y, G_z) y las seis tensoriales ($T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz}):

$$\begin{aligned} (x - x_0)T_{xx} + (y - y_0)T_{xy} + (-z_0)T_{xz} &= N[B_x - G_x] \\ (x - x_0)T_{xy} + (y - y_0)T_{yy} + (-z_0)T_{yz} &= N[B_y - G_y] \\ (x - x_0)T_{xz} + (y - y_0)T_{yz} + (-z_0)T_{zz} &= N[B_z - G_z] \end{aligned} \quad (2.66)$$

Desarrollando el sistema de ecuaciones 2.66 y sustituyendo datos pertenecientes a la ventana de v puntos:

$$\begin{aligned} x_0T_{xx} + y_0T_{xy} + z_0T_{xz} + NB_x &= xT_{xx} + yT_{xy} + NG_x \\ x_0T_{xy} + y_0T_{yy} + z_0T_{yz} + NB_y &= xT_{xy} + yT_{yy} + NG_y \\ x_0T_{xz} + y_0T_{yz} + z_0T_{zz} + NB_z &= xT_{xz} + yT_{yz} + NG_z \end{aligned}$$

$$\Rightarrow \begin{bmatrix} T_{xx_1} & T_{xy_1} & T_{xz_1} & N & 0 & 0 \\ T_{xx_2} & T_{xy_2} & T_{xz_2} & N & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_{xx_v} & T_{xy_v} & T_{xz_v} & N & 0 & 0 \\ \\ T_{xy_1} & T_{yy_1} & T_{yz_1} & 0 & N & 0 \\ T_{xy_2} & T_{yy_2} & T_{yz_2} & 0 & N & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_{xy_v} & T_{yy_v} & T_{yz_v} & 0 & N & 0 \\ \\ T_{xz_1} & T_{yz_1} & T_{zz_1} & 0 & 0 & N \\ T_{xz_2} & T_{yz_2} & T_{zz_2} & 0 & 0 & N \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_{xz_v} & T_{yz_v} & T_{zz_v} & 0 & 0 & N \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ B_x \\ B_y \\ B_z \end{bmatrix} = \begin{bmatrix} x_1T_{xx_1} + y_1T_{xy_1} + NG_{x_1} \\ x_2T_{xx_2} + y_2T_{xy_2} + NG_{x_2} \\ \vdots \\ x_vT_{xx_v} + y_vT_{xy_v} + NG_{x_v} \\ \\ x_1T_{xy_1} + y_1T_{yy_1} + NG_{y_1} \\ x_2T_{xy_2} + y_2T_{yy_2} + NG_{y_2} \\ \vdots \\ x_vT_{xy_v} + y_vT_{yy_v} + NG_{y_v} \\ \\ x_1T_{xz_1} + y_1T_{yz_1} + NG_{z_1} \\ x_2T_{xz_2} + y_2T_{yz_2} + NG_{z_2} \\ \vdots \\ x_vT_{xz_v} + y_vT_{yz_v} + NG_{z_v} \end{bmatrix} \quad (2.67)$$

Una vez que se resuelve el sistema de ecuaciones resultante (Ec. 2.67), se aplica el criterio de aceptación y se repite el procedimiento en cada posición de la ventana móvil.

2.3 Modelado Gravimétrico Inverso 3D

De acuerdo a Tarantola (2005), el procedimiento científico para estudiar un sistema físico puede ser dividido en tres grandes pasos: parametrización del sistema, modelado directo y modelado inverso, entendiéndose por este último, el uso de resultados de mediciones (datos) para inferir los valores de parámetros que caracterizan el sistema. En este sentido, el modelado gravimétrico inverso 3D es el uso de observaciones del vector o tensor gravimétrico para determinar la forma y densidad de las estructuras del subsuelo bajo los puntos de observación.

En este trabajo de tesis se propone la utilización de un método de inversión heurístico tipo Simulated Annealing (SA), modificado para lograr un mejor desempeño en tiempo de cómputo y reducción del problema de no unicidad.

2.3.1 Formulación del problema gravimétrico inverso

La formulación propuesta para resolver el problema gravimétrico inverso está estrechamente relacionada con el problema directo: del mismo modo en que se discretiza una estructura compleja en un ensamble de prismas para realizar el modelado directo (subcapítulo 2.1.3), es necesario contar con un medio discreto en el que tomarán forma las estructuras del subsuelo al realizarse el modelado inverso.

Si se considera que el medio se discretiza con prismas de idénticas dimensiones, el problema directo dependerá únicamente de sus densidades, que serán a su vez, los únicos parámetros por invertir:

$$\begin{aligned} f_i &= g(\rho) = g(m) \\ \Rightarrow f_i &= \sum_{j=1}^M g(m_j) \end{aligned} \quad (2.68)$$

donde f_i es la respuesta gravimétrica del ensamble (cualquier componente vectorial o tensorial) en cada punto de observación i .

Expandiendo $g(m)$ en series de Taylor hasta sus primeros dos términos:

$$g(m) = g(m^0) + \frac{\partial}{\partial m} g(m^0) \cdot (m - m^0) \quad (2.69)$$

Sustituyendo 2.69 en la ecuación 2.68:

$$f_i = f^0 + \sum_{j=1}^M \left. \frac{\partial g(m)}{\partial m_j} \right|_{m=m^0} \cdot (m_j - m^0) \quad (2.70)$$

La ecuación 2.70 se puede reescribir como:

$$f_i = f^0 + G_{ij} \cdot (m_j - m^0) \quad (2.71)$$

donde G_{ij} es una matriz jacobiana denominada *kernel* o *matriz de sensibilidades*, formada por las derivadas parciales del problema directo en los puntos de observación, con respecto a los parámetros del ensamble:

$$G_{ij} = \begin{bmatrix} \left. \frac{\partial f}{\partial m_1} \right|_{i_1} & \left. \frac{\partial f}{\partial m_2} \right|_{i_1} & \cdots & \left. \frac{\partial f}{\partial m_M} \right|_{i_1} \\ \left. \frac{\partial f}{\partial m_1} \right|_{i_2} & \left. \frac{\partial f}{\partial m_2} \right|_{i_2} & \cdots & \left. \frac{\partial f}{\partial m_M} \right|_{i_2} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial f}{\partial m_1} \right|_{i_N} & \left. \frac{\partial f}{\partial m_2} \right|_{i_N} & \cdots & \left. \frac{\partial f}{\partial m_M} \right|_{i_N} \end{bmatrix} \quad (2.72)$$

Cada elemento de esta matriz representa la contribución del prisma j a la respuesta gravimétrica del ensamble, f , en el punto de observación i , es decir, la sensibilidad del problema directo con respecto a cada prisma del ensamble, en cada punto de observación.

Ahora, si m^0 es un vector nulo (modelo con densidades nulas), f^0 también lo será (no habrá anomalía) y al tomar estas consideraciones en la expresión 2.71, el problema directo asumirá la siguiente forma:

$$f_i = G_{ij} \cdot m_j \quad (2.73)$$

Una vez que se tiene planteado el problema gravimétrico directo para el ensamble de M prismas como el producto de una matriz por un vector (Ec. 2.73), se puede formular el problema inverso.

Antes de describir el método heurístico de inversión empleado, vale la pena mencionar que aunque los métodos de inversión basados en algoritmos de optimización global requieren del cálculo del problema directo en múltiples ocasiones de forma excesiva, estos han sido aplicados exitosamente en diversos problemas de exploración geofísica, donde las dimensiones del problema no han supuesto un problema mayor (e.g. Rodríguez-Zúñiga et al., 1997; Orozco-del-Castillo et al., 2014).

2.3.2 Inversión por el método Simulated Annealing (SA)

El método de inversión SA está basado en el algoritmo de optimización global del mismo nombre, considerando que el óptimo buscado es el mínimo de la función de error entre datos observados y datos sintéticos, generados por modelado directo.

El algoritmo SA fue inspirado en el proceso de optimización natural de la formación de cristales a partir de un fluido mineral a altas temperaturas y sus conceptos básicos se tomaron de la mecánica estadística, estableciéndose las siguientes analogías entre parámetros del modelo y partículas en un sistema físico idealizado:

TABLA 2.2: Analogías entre términos utilizados en la mecánica estadística y los problemas de optimización por el método Simulated Annealing.

Mecánica Estadística	Problema de Optimización
Nivel energético del sistema	Función de costo del problema
Configuración de partículas	Solución o modelo particular
Moléculas	Parámetros del modelo

Este proceso de optimización simula la evolución de un sistema físico que inicialmente es calentado incrementando su temperatura, T , hasta que todas sus partículas se distribuyen aleatoriamente en una fase líquida y posteriormente se enfría gradualmente, permitiéndole alcanzar el equilibrio térmico en cada valor transicional de temperatura, de tal modo que en el límite $T \rightarrow 0$, el estado de mínima energía se vuelve predominantemente probable, así como la formación de cristales y por lo tanto, la configuración de parámetros análoga podría considerarse como un modelo óptimo (Sen y Stoffa, 2013).

Cabe mencionar que en cada temperatura, la probabilidad que tienen las partículas en un estado “ i ”, con energía “ E_i ”, de alcanzar el equilibrio térmico, está dada por la función de densidad de probabilidades de Boltzmann:

$$P(E_i) = \frac{\exp\left(-\frac{E_i}{KT}\right)}{\sum_{j \in S} \exp\left(-\frac{E_j}{KT}\right)} = \frac{1}{Z(T)} \exp\left(-\frac{E_i}{KT}\right) \quad (2.74)$$

donde S es el conjunto de todas las posibles configuraciones moleculares, K es la constante de Boltzmann, T es la temperatura y $Z(T)$ la función de partición.

De acuerdo a esta función de densidad de probabilidades, algunas configuraciones de moléculas (modelos) tendrán mayor probabilidad de alcanzar el equilibrio térmico que otras, lo cual, en la práctica, podría servir como control de calidad de soluciones en el algoritmo de inversión. Desafortunadamente no se puede aplicar directamente, debido a que se requeriría conocer el contenido energético de todas las configuraciones moleculares posibles. En su lugar, de acuerdo a Kirkpatrick et al. (1983), se puede utilizar el *algoritmo de Metrópolis*, que

se aproxima asintóticamente a esta función y dota al algoritmo de inversión con la habilidad de escapar de óptimos locales, incrementando la probabilidad de alcanzar el óptimo global.

El algoritmo de Metrópolis consiste en perturbar un modelo inicial, m_i , cuya energía es $E(m_i)$, para obtener un nuevo modelo m_j , con energía $E(m_j)$, calcular el incremento de energía debido a la perturbación:

$$\Delta E_{ij} = E(m_j) - E(m_i)$$

y aceptar o rechazar el modelo m_j , de acuerdo al incremento energético ΔE_{ij} calculado: Si $\Delta E_{ij} \leq 0$, el modelo m_j se aceptará incondicionalmente, pero si $\Delta E_{ij} > 0$, m_j será aceptado con la probabilidad:

$$P(m_j) = \exp\left(-\frac{\Delta E_{ij}}{T}\right) \quad (2.75)$$

para lo cual, se genera un número aleatorio A perteneciente al intervalo $[0, 1]$ y se compara con la probabilidad calculada con la ecuación 2.75: Si $A \leq P(m_j)$, el modelo m_j será aceptado, de lo contrario se rechazará.

Este proceso de aceptación y rechazo de modelos se repite varias veces a la misma temperatura T_i , hasta que se alcanza el equilibrio térmico, caracterizado por no exhibir cambios importantes en el nivel energético y se continúa con la siguiente etapa del algoritmo, que es la reducción de temperatura de acuerdo a un *esquema de enfriamiento* definido. En este trabajo se consideró un decaimiento térmico exponencial, que de acuerdo a Nagihara y Hall (2001), se caracteriza por garantizar la convergencia hacia el óptimo global:

$$T_k = T_i (RT)^k \quad (2.76)$$

donde: T_i es la temperatura inicial del sistema

T_k es la temperatura en el k -ésimo estado

RT es un factor de reducción de temperatura ($0 < RT < 1$)

Finalmente, se repite este proceso hasta el límite $T \rightarrow 0$, controlado por un criterio de detención, dado en forma de tolerancia con respecto al nivel energético del modelo perturbado, Tol , combinado con un número máximo de disminuciones de temperatura predeterminado, k_{max} : cuando $E(m_j) \leq Tol$, o cuando $k = k_{max}$, el algoritmo se detiene y como resultado, se tendrá un modelo potencialmente útil para la interpretación.

Para cuantificar el nivel energético del sistema (función de costo o error) en cada etapa de perturbación del algoritmo, se utilizó la norma L_2 "normalizada" (Ortiz-Alemán et al., 2004; Ortiz-Alemán y Martin, 2005b), definida como:

$$E = L_2 = \frac{\sum_{i=1}^N (d_i^{obs} - d_i^{calc})^2}{\sum_{i=1}^N (d_i^{obs})^2} \quad (2.77)$$

donde d_i^{obs} es el conjunto de datos observados y d_i^{calc} es el conjunto de datos calculados para el k -ésimo modelo perturbado, m_k .

Cabe mencionar que para la inversión de datos de anomalía gravimétrica, g_z , la energía se calculó directamente con la función de error L_2 normalizada (2.77):

$$E = L_2(g_z) \quad (2.78)$$

mientras que para la inversión de mallas de gradiometría gravimétrica, la energía se calculó como una suma pesada de las funciones de error de cada componente del tensor:

$$E = \alpha L_2(T_{xx}) + \beta L_2(T_{xy}) + \gamma L_2(T_{xz}) + \delta L_2(T_{yy}) + \varepsilon L_2(T_{yz}) + \eta L_2(T_{zz}) \quad (2.79)$$

donde la suma de los factores de peso es igual a la unidad:

$$\alpha + \beta + \gamma + \delta + \varepsilon + \eta = 1$$

y cada uno de ellos es inversamente proporcional al máximo de la matriz de sensibilidades de la componente tensorial asociada, lo cual otorga mayor peso a la componente de menor sensibilidad hacia los parámetros estimados, causando que las componentes poco sensibles, pero importantes, se reflejen en el error calculado:

$$\begin{aligned} \alpha &= \frac{1}{5} \left[1 - \frac{\max(G_{xx})}{SG} \right] & \beta &= \frac{1}{5} \left[1 - \frac{\max(G_{xy})}{SG} \right] & \gamma &= \frac{1}{5} \left[1 - \frac{\max(G_{xz})}{SG} \right] \\ \delta &= \frac{1}{5} \left[1 - \frac{\max(G_{yy})}{SG} \right] & \varepsilon &= \frac{1}{5} \left[1 - \frac{\max(G_{yz})}{SG} \right] & \eta &= \frac{1}{5} \left[1 - \frac{\max(G_{zz})}{SG} \right] \end{aligned}$$

con: $SG = \max(G_{xx}) + \max(G_{xy}) + \max(G_{xz}) + \max(G_{yy}) + \max(G_{yz}) + \max(G_{zz})$

Dado que el algoritmo SA requiere de un gran número de evaluaciones del problema directo, se aplicó una mejora al algoritmo básico para acelerar el producto $G_{ij} \cdot m_j$, tal como lo propusieron Ortiz-Alemán y Martín (2005a), utilizando un problema directo previamente calculado y actualizándolo con la suma del producto $G_{i,j=q} \cdot \Delta m_j^q$ (Δm_j^q es la perturbación aplicada al q -ésimo parámetro del modelo m_j y $G_{i,j=q}$ es la q -ésima columna del kernel de sensibilidades):

Sea d_i un problema directo previamente calculado:

$$d_i = G_{ij} \cdot m_j \quad (2.80)$$

Al aplicar una perturbación, Δm_j^q , al q -ésimo parámetro del modelo m_j , el problema directo, d_i^{calc} , de este nuevo modelo será:

$$\begin{aligned} d_i^{calc} &= G_{ij} \cdot (m_j + \Delta m_j^q) \\ &= \begin{bmatrix} G_{1,1} & G_{1,2} & \cdots & G_{1,M} \\ G_{2,1} & G_{2,2} & \cdots & G_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N,1} & G_{N,2} & \cdots & G_{N,M} \end{bmatrix} \begin{bmatrix} m_1 \\ \vdots \\ m_q + \Delta m_q \\ \vdots \\ m_M \end{bmatrix} \end{aligned} \quad (2.81)$$

Desarrollando el producto matriz-vector de la ecuación 2.81 (miembro derecho):

$$\begin{aligned} G_{ij} \cdot (m_j + \Delta m_j^q) &= \begin{bmatrix} \sum_{j=1}^M (G_{1,j} \cdot m_j) + G_{1,q} \cdot \Delta m_q \\ \sum_{j=1}^M (G_{2,j} \cdot m_j) + G_{2,q} \cdot \Delta m_q \\ \vdots \\ \sum_{j=1}^M (G_{N,j} \cdot m_j) + G_{N,q} \cdot \Delta m_q \end{bmatrix} \\ &= G_{ij} \cdot m_j + \begin{bmatrix} G_{1,q} \\ G_{2,q} \\ \vdots \\ G_{N,q} \end{bmatrix} \cdot \Delta m_q \end{aligned} \quad (2.82)$$

Ahora, dado que $d_i = G_{ij} \cdot m_j$, la expresión 2.82 se puede reescribir como:

$$\begin{aligned} G_{ij} \cdot (m_j + \Delta m_j^q) &= d_i + \begin{bmatrix} G_{1,q} \\ G_{2,q} \\ \vdots \\ G_{N,q} \end{bmatrix} \cdot \Delta m_q \\ \therefore d_i^{calc} &= d_i + G_{i,j=q} \cdot \Delta m_j^q \end{aligned} \quad (2.83)$$

Esta forma de calcular el problema directo del modelo después de la aplicación de una perturbación en un parámetro, justifica el empleo del SA en la inversión de un sistema lineal

de grandes dimensiones, ya que después de la primera iteración no es necesario repetir el cálculo para todo el ensamble y por lo tanto, la carga computacional se reducirá dramáticamente.

La estructura del algoritmo de inversión se puede resumir en el diagrama mostrado en la Figura 2.14, el cual consiste en un ciclo anidado de tres niveles, basado en el algoritmo presentado por Goffe et al. (1994):

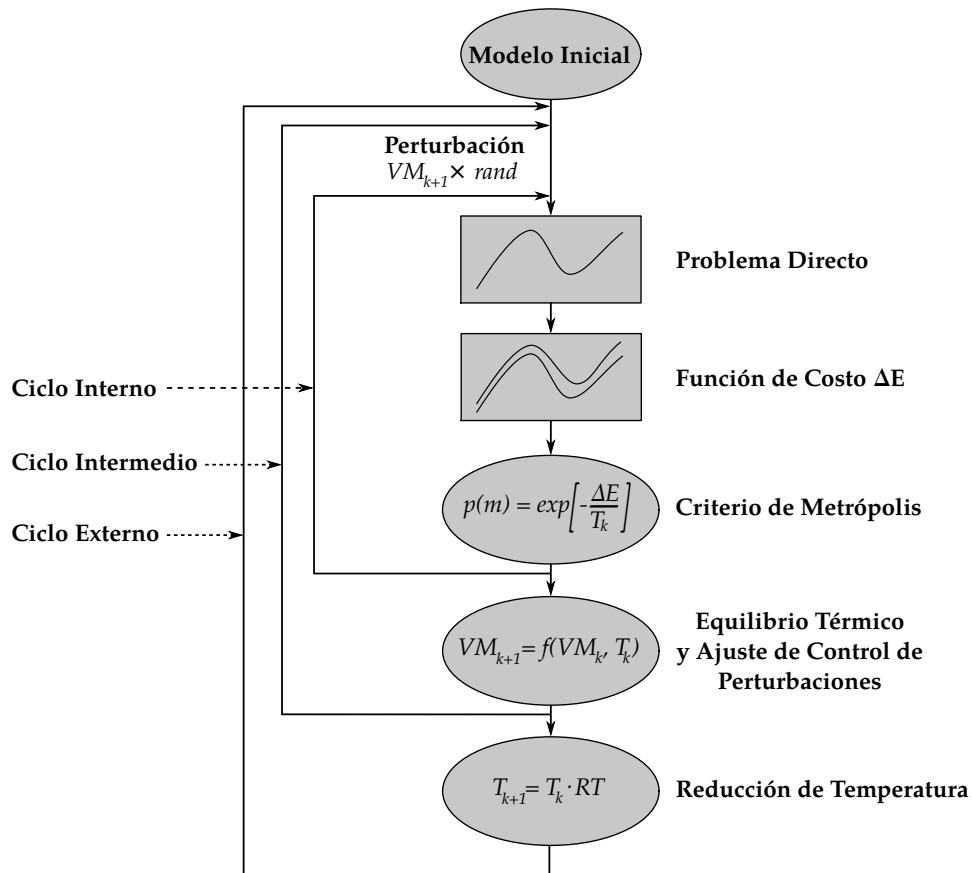


FIGURA 2.14: Diagrama de flujo representativo del algoritmo de inversión SA implementado en esta tesis (modificado de Ortiz-Alemán y Martín, 2005a).

Cabe mencionar que para la inversión gravimétrica de estructuras salinas del Golfo de México, se hizo una última modificación que consistió en aplicar un control de amplitud de paso auto-ajutable, VM , por parámetro, entre cada etapa de reducción de temperatura, tal como fue propuesto por Corana et al. (1987):

Sea r el cociente entre el número de modelos aceptados, N_A y totales probados, N_T por el criterio de Metrópolis:

$$r = \frac{N_A}{N_T}$$

Se define un escalar VM en función de r , cuya función es tratar de igualar el número de modelos rechazados y aceptados en el ciclo de equilibrio térmico:

$$\begin{aligned} \text{Si } r > 0.6: VM_{k+1} &= VM_k \left[1 + \frac{r - 0.6}{0.2} \right] \\ \text{y si } r < 0.4: VM_{k+1} &= \frac{VM_k}{\left[1 + \frac{0.4 - r}{0.2} \right]} \end{aligned}$$

Una vez calculado VM , se multiplicará por un número aleatorio perteneciente al intervalo $[-1, 1]$ para dar lugar a la perturbación por aplicar a cada parámetro, Δm , previa a la reducción de temperatura:

$$\Delta m = VM \cdot rand(-1, 1)$$

Simulated Annealing con sensibilidad aumentada (SA-SA)

Debido a que el campo gravitacional decae con la distancia, las matrices de sensibilidades que representan el problema directo contendrán elementos cuyos valores pueden llegar a diferir entre sí algunos órdenes de magnitud. Los mayores valores corresponden con los prismas ubicados cerca de los puntos de observación y los menores con los prismas situados lejos de ellos, causando que los modelos obtenidos por la inversión contengan soluciones sesgadas hacia los elementos de mayor sensibilidad.

Para compensar este comportamiento, se propone aplicar un factor de peso a la matriz de sensibilidades denominado *amplificador de sensibilidad* (w), cuyos valores son inversamente proporcionales a la sensibilidad del problema.

Considerando que la inversión se realiza en un dominio definido por un ensamble de prismas similar al que se ilustra en la Figura 2.12, el factor w tendrá M elementos y dependerá de la profundidad:

$$w(z_i) = \frac{z_i}{\min(z_p)} \quad (2.84)$$

donde z_i es la profundidad de la i -ésima capa del ensamble y $\min(z_p)$ es la profundidad mínima de todo el ensamble.

La aplicación de un factor de peso en los kernels de inversión para contrarrestar el efecto de la obtención de soluciones cercanas a la superficie (puntos de observación) fue propuesta por Li y Oldenburg (1998), sin embargo la forma en que lo implementaron fue en la función de costo, como parte de una estrategia de inversión lineal, a diferencia de la forma en que se plantea en este trabajo de tesis, donde el factor de peso (amplificador) es aplicado al kernel de sensibilidades de forma análoga al uso temporal del lente de una lupa para amplificar un objeto pequeño bajo observación: a simple vista el objeto puede lucir poco definido, lo que causará que sea mal interpretado o incluso no ser registrado visualmente. En

el momento en que se utiliza el lente, el objeto se amplificará permitiéndole ser interpretado de forma correcta y una vez hecho esto, el lente se puede retirar.

El factor w se implementa en el algoritmo de inversión SA después del cálculo del primer problema directo (respuesta del modelo inicial), como parte de la perturbación aplicada al modelo:

Sea: $d_i = G_{ij} \cdot m_j$ la respuesta del modelo inicial.

Al aplicar una perturbación al modelo en el q -ésimo parámetro, la respuesta para el problema con sensibilidad aumentada -*colocación del lente*- será:

$$d_i^{calc^w} = d_i + \begin{bmatrix} G_{1,q} \\ G_{2,q} \\ \vdots \\ G_{N,q} \end{bmatrix} \cdot w_q \cdot \Delta m_q$$

Una vez obtenida esta respuesta, se calcula el incremento de energía causado por la perturbación en el problema con sensibilidad aumentada -*utilización del lente*- y se aplica el criterio de Metrópolis:

$$\Delta E = E(m_j + w_q \cdot \Delta m_j^q) - E(m_j)$$

Si $\Delta E < 0$: $m_j + w_q \cdot \Delta m_j^q$ se acepta incondicionalmente

y si $\Delta E \geq 0$: $m_j + w_q \cdot \Delta m_j^q$ se aceptará con la probabilidad $Pr = \exp\left(-\frac{\Delta E}{T}\right)$

Finalmente se recalcula la respuesta del modelo perturbado para el problema sin aumento de sensibilidad -*remoción del lente*- y el algoritmo SA continúa con su ejecución:

$$d_i^{calc} = d_i + \begin{bmatrix} G_{1,q} \\ G_{2,q} \\ \vdots \\ G_{N,q} \end{bmatrix} \cdot \Delta m_q$$

MODELOS SINTÉTICOS Y PRUEBAS DE MÉTODOS

Los modelos sintéticos son esenciales en la interpretación de datos de anomalía o gradiente gravitacional, debido a que permiten evaluar la eficiencia, rendimiento y alcances de los métodos empleados, al comparar la información derivada de sus resultados con respecto a los cuerpos conocidos que forman los modelos en los que fueron aplicados.

3.1 Modelos sintéticos

A partir de las soluciones analíticas para el vector y tensor gravimétricos causados por una esfera y un prisma rectangular, se diseñaron dos modelos sintéticos denominados **G1** y **G2**. El modelo **G1** se diseñó para probar métodos de realce de anomalías y bordes, así como estimación de profundidades, mientras que el modelo **G2** fue diseñado para probar los algoritmos de inversión. Los datos generados para ambos modelos consistieron en mallas regulares libres de ruido y con ruido gaussiano pseudoaleatorio añadido, con media cero y desviación estándar igual al 5 % de la desviación estándar de los datos libres de ruido, denotado como $N(0, 0.05\sigma)$.

TABLA 3.1: Modelos sintéticos generados para probar el rendimiento de los algoritmos de delimitación de fuentes, estimación de profundidades e inversión de mallas de anomalías.

Modelo	Cuerpos	$\Delta\rho \left[\frac{kg}{m^3} \right]$	Dimensiones [m]	Profundidad [m]
G1	E (Esfera)	1000.0	Radio: 100.0	$z_c = 120.0$
	C (Cilindro)	900.0	Radio: 125.0 Altura: 650.0	$z = 30.0$
	P (Prisma)	1250.0	400.0 × 150.0 × 600.0	$z = 50.0$
G2	Ensamble (1, 000, 000 prismas)	0.0; 1000.0	1000.0 × 1000.0 × 1000.0	$z = 1.0$

En la Tabla 3.1 se muestran las características de los modelos sintéticos generados, mientras que en las Figuras 3.1 a 3.8 se ilustran los modelos y las mallas de anomalías causadas por ellos.

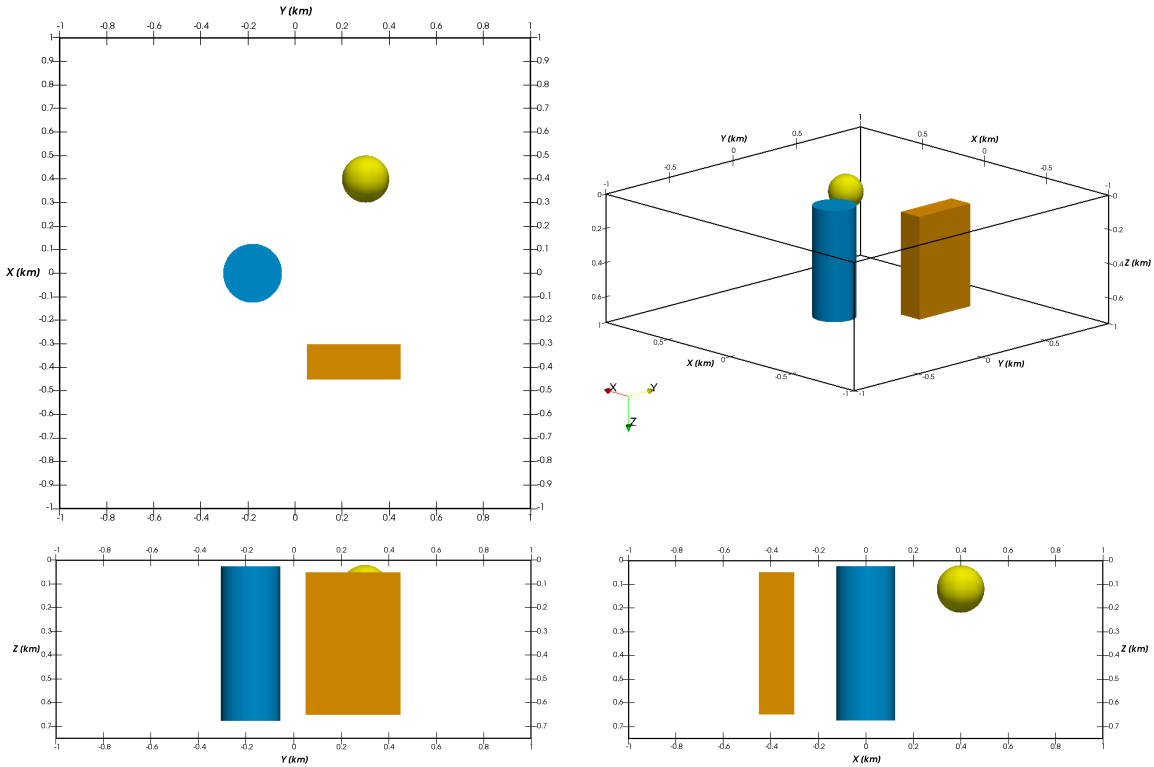


FIGURA 3.1: Modelo sintético G1.

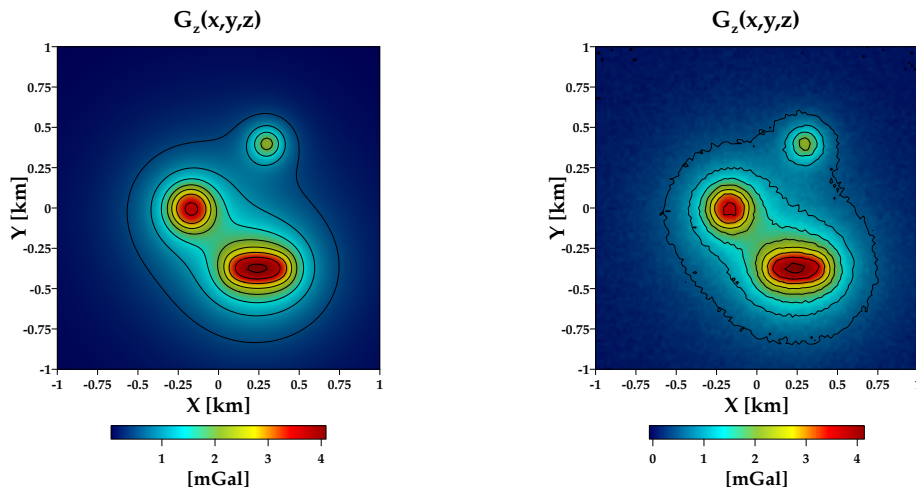


FIGURA 3.2: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo sintético G1, observado en una malla regular de 101×101 puntos. A la izquierda se muestra la malla libre de ruido y a la derecha, la malla con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$.

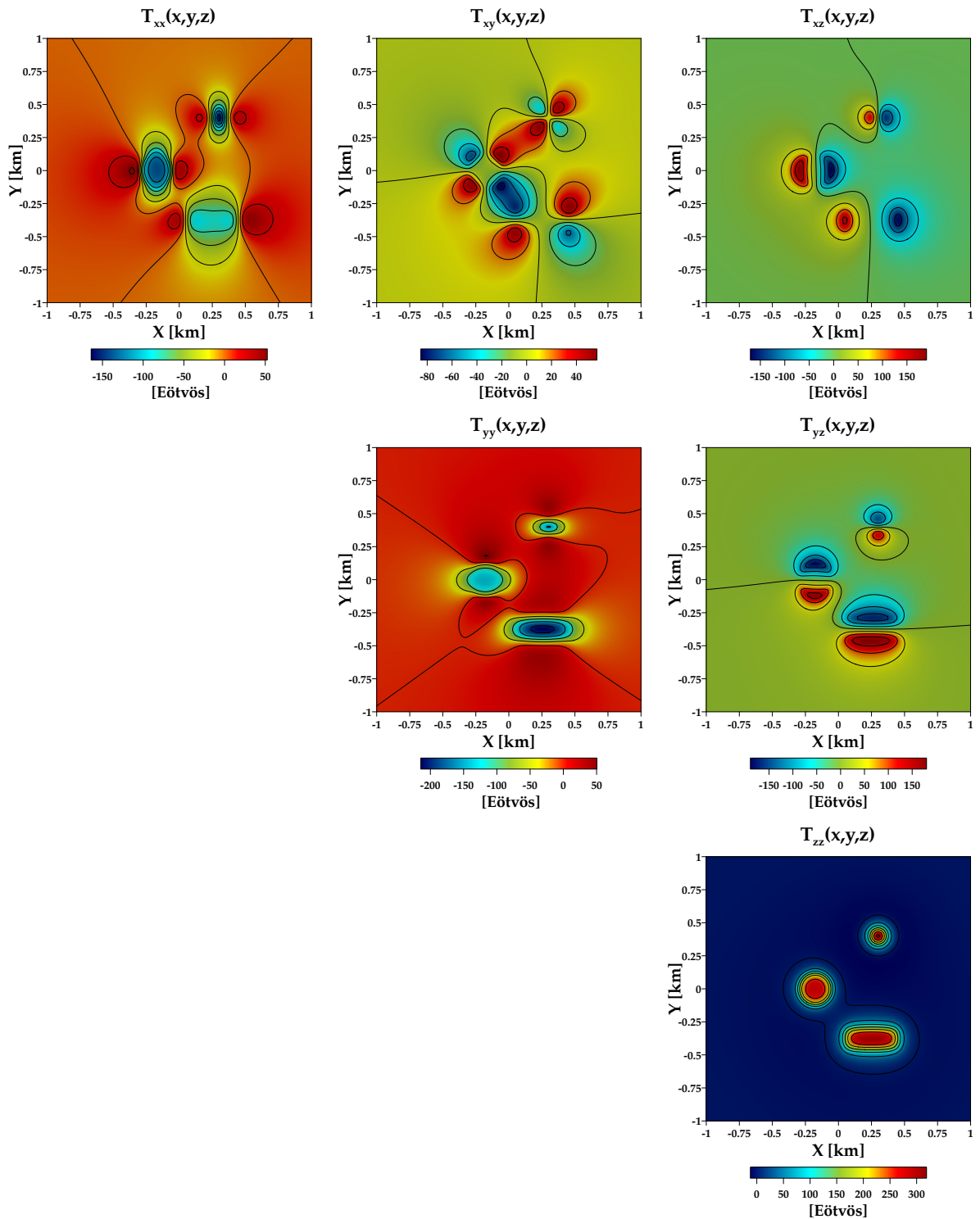


FIGURA 3.3: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G1, observado en una malla regular de 101×101 puntos.

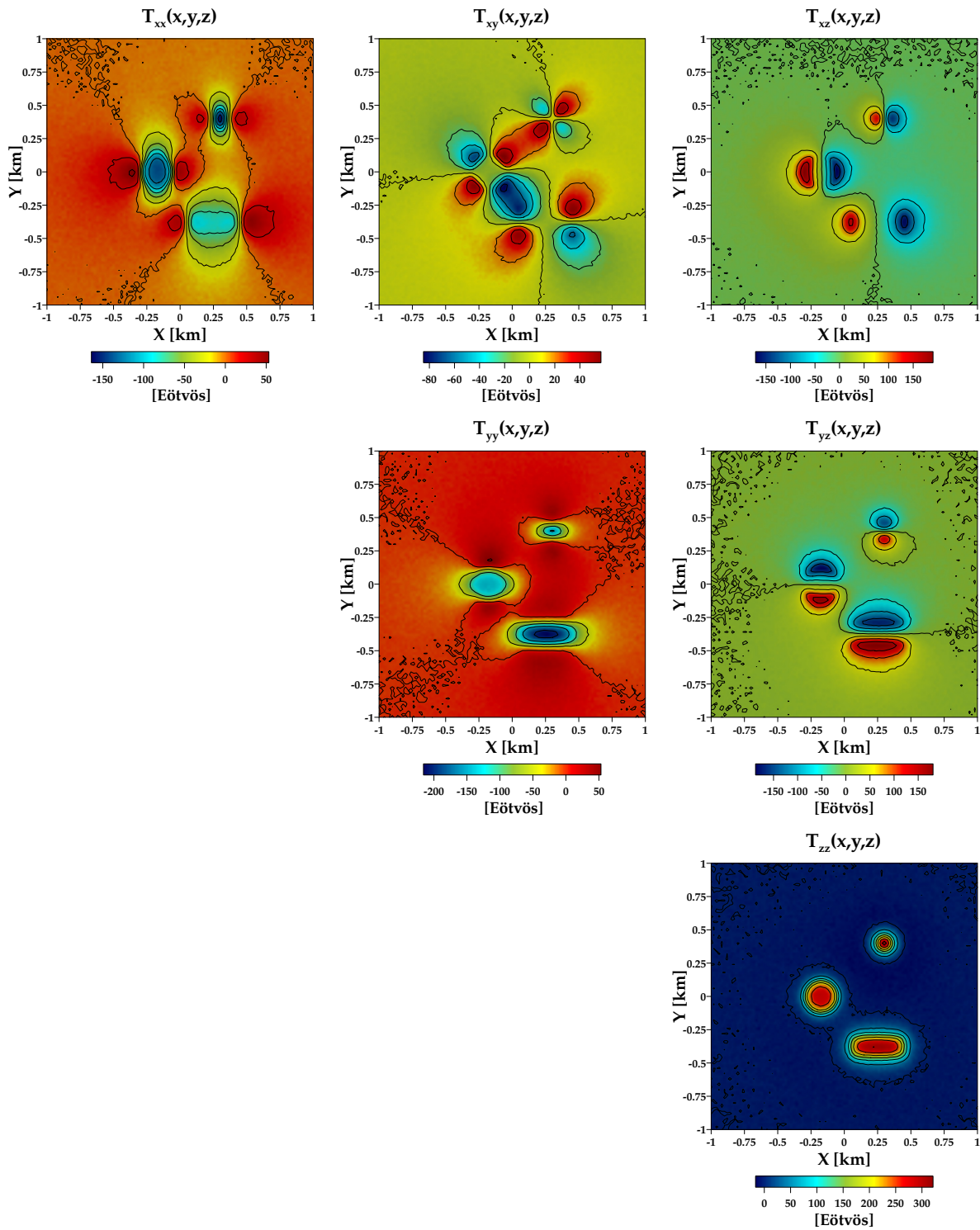


FIGURA 3.4: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G1, observado en una malla regular de 101×101 puntos con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$.

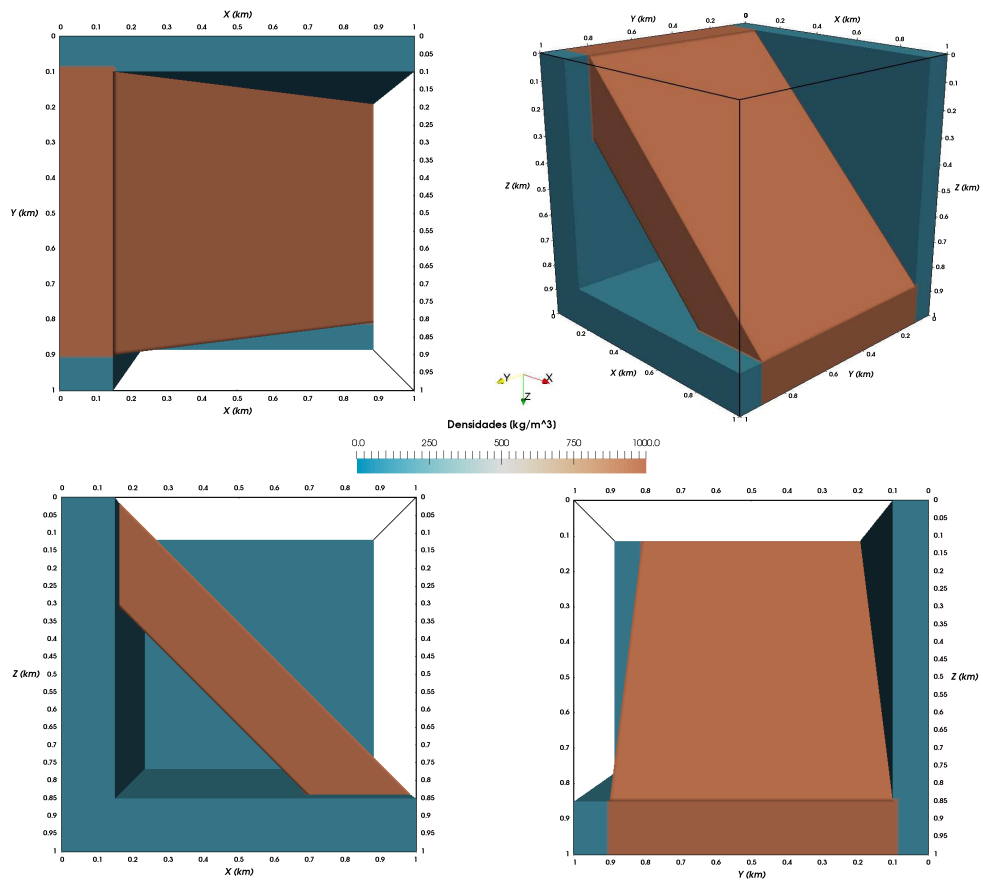


FIGURA 3.5: Modelo sintético G2.

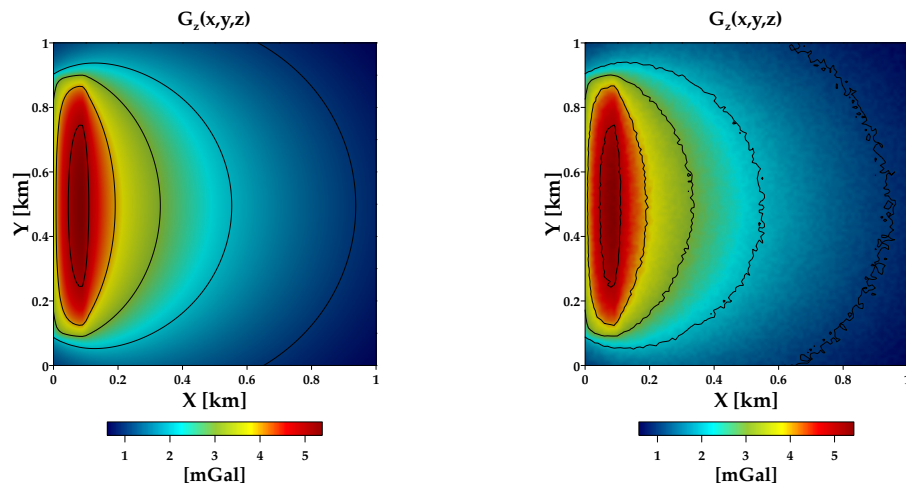


FIGURA 3.6: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo sintético G2, observado en una malla regular de 101×101 puntos. A la izquierda se muestra la malla libre de ruido y a la derecha, la malla con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$.

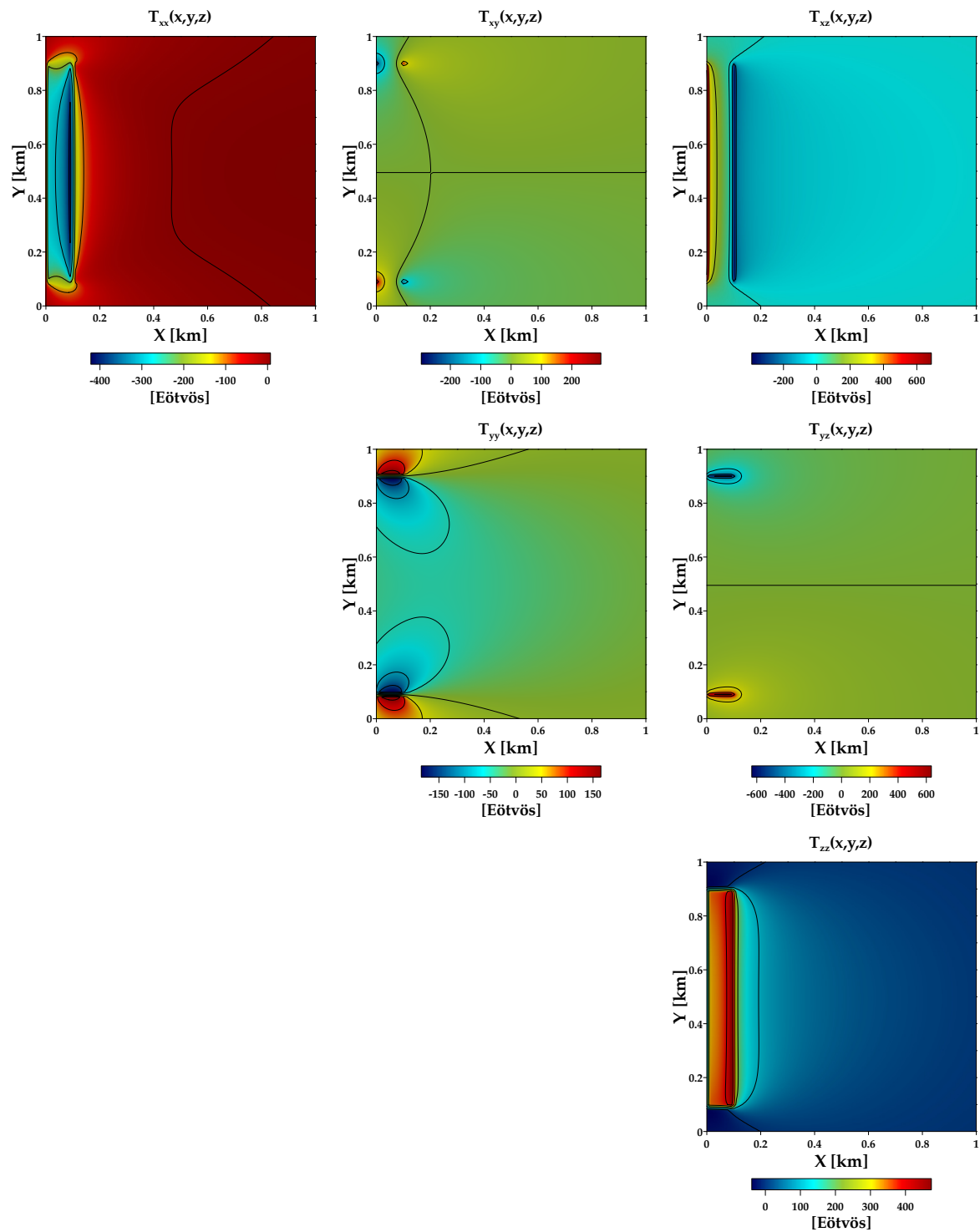


FIGURA 3.7: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G2, observado en una malla regular de 101×101 puntos.

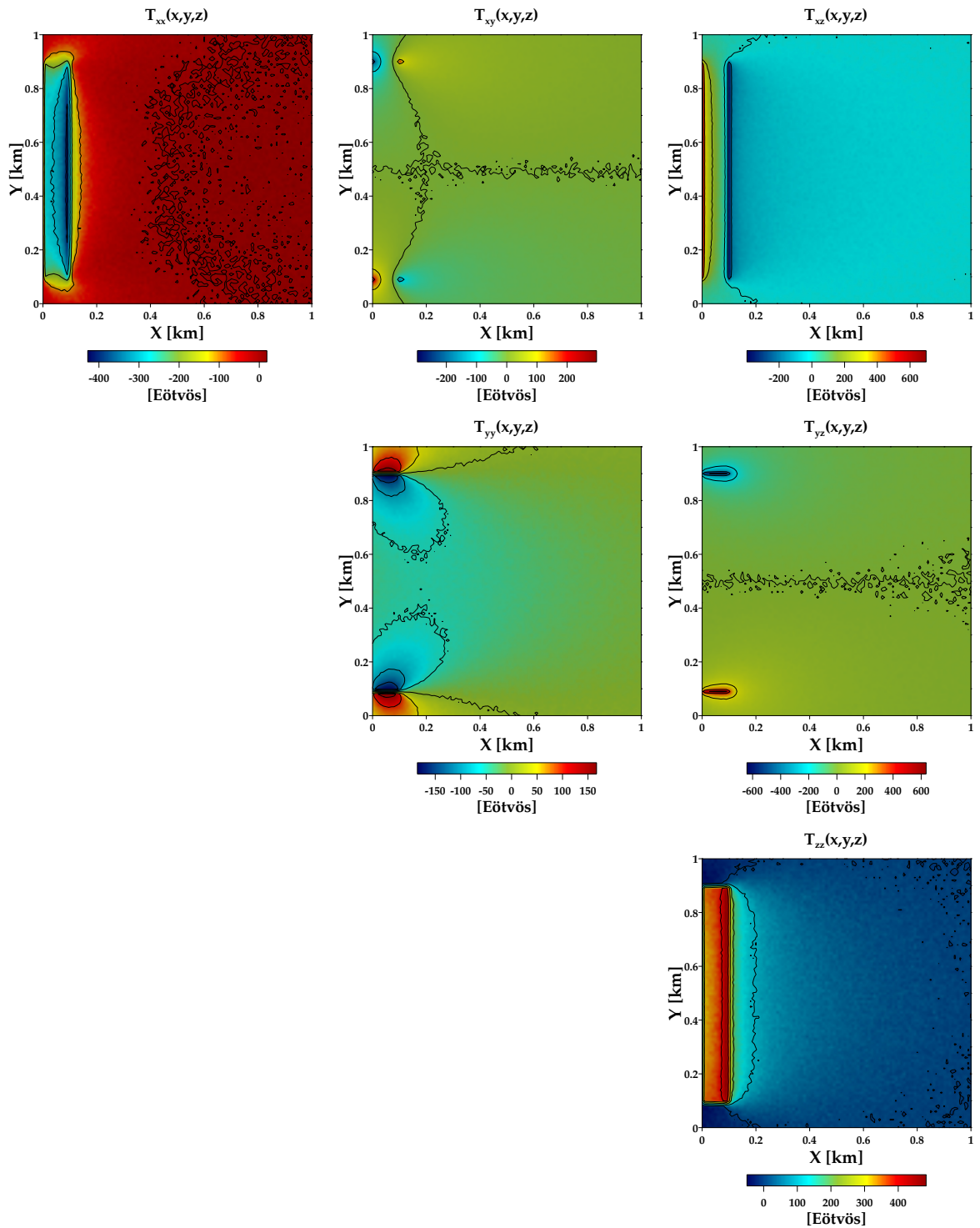


FIGURA 3.8: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional, generado por el modelo sintético G2, observado en una malla regular de 101×101 puntos con ruido gaussiano pseudoaleatorio añadido $N(0, 0.05\sigma)$.

3.2 Pruebas de métodos de realce de bordes y estimación de profundidades

En este subcapítulo se muestran las pruebas de aplicación de los métodos de realce de anomalías, bordes e interpretación semiautomática, descritos en el subcapítulo 2.2, utilizando las mallas de datos del modelo **G1** (Figuras 3.2, 3.3 y 3.4). En cuanto a los procesos de realce de bordes, se muestra únicamente la interpretación de aquellos que resultaron ser más eficientes. Todas las mallas procesadas se pueden consultar en el Anexo C.

3.2.1 Pruebas en mallas de anomalía gravimétrica

Continuación Analítica

En la Figura 3.9 se muestra el efecto de la aplicación del proceso de continuación ascendente a una altura de 40 [m], equivalente a 2 unidades de malla (UM), a la malla de componente G_z del modelo sintético **G1**, contaminada con ruido:

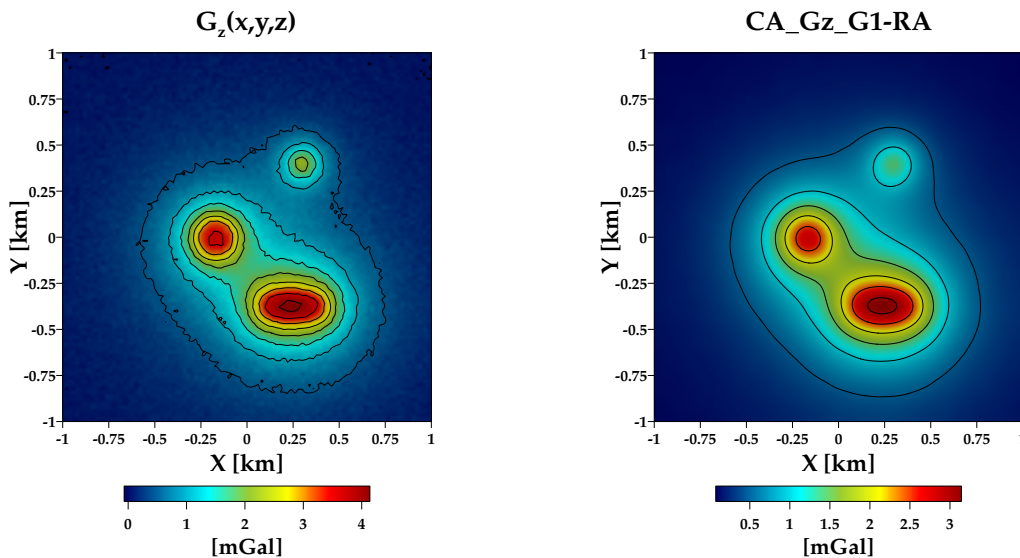


FIGURA 3.9: Izquierda: Malla de la componente G_z del modelo sintético **G1** con ruido aleatorio añadido. Derecha: Malla **CA_Gz_G1-RA**, resultante del proceso de continuación ascendente de la malla $G_z(x, y, z)$, a 2 UM.

Al inspeccionar la Figura 3.9, es notable la atenuación de ruido a través del proceso de continuación ascendente y la conservación de los rasgos principales de las anomalías, lo que muestra su utilidad práctica, permitiendo tomar la malla resultante de este proceso, para la aplicación de posteriores procesos y análisis cuantitativos.

Interpretación de bordes

Para probar los métodos de realce de bordes, se interpretaron los contornos de los cuerpos del modelo sintético **G1** en planta, a partir de las mallas procesadas y se graficaron en un mapa. Los métodos se probaron en datos libres de ruido y con ruido aleatorio añadido, continuados ascendentemente 2 UM.

En la Figura 3.10 se muestran los bordes estimados por los métodos que arrojaron los mejores resultados: señal analítica mejorada de primer orden, gradiente horizontal aplicado a la anomalía gravimétrica y a su primera derivada vertical, así como el ángulo de inclinación de gradientes aplicado a la malla de primera derivada vertical. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas encerrando los cuerpos resaltados en tono gris y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro).

La notación utilizada para identificar los contornos interpretados a partir de cada método es la siguiente:

- SAM1** : Señal analítica mejorada de primer orden
- GH** : Gradiente horizontal
- GH_Dz** : Gradiente horizontal, aplicado a la derivada vertical
- AIG_Dz** : Ángulo de inclinación de gradientes, aplicado a la derivada vertical

El sufijo **G1** añadido a cada indicador, significa que el método fue aplicado a la malla de anomalía libre de ruido (Figura 3.10 superior), mientras que el sufijo **G1-RA** representa la aplicación del método a la malla de anomalía con ruido añadido (Figura 3.10 inferior).

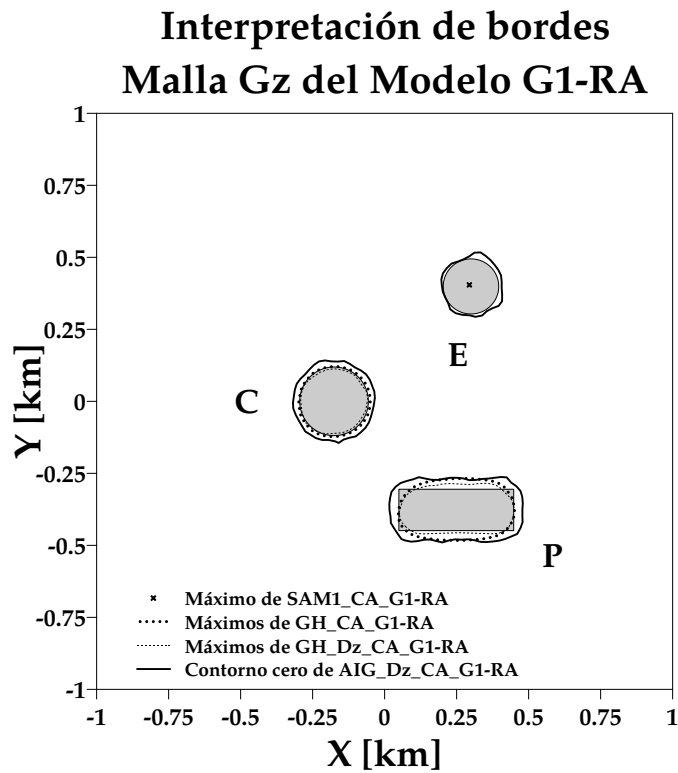
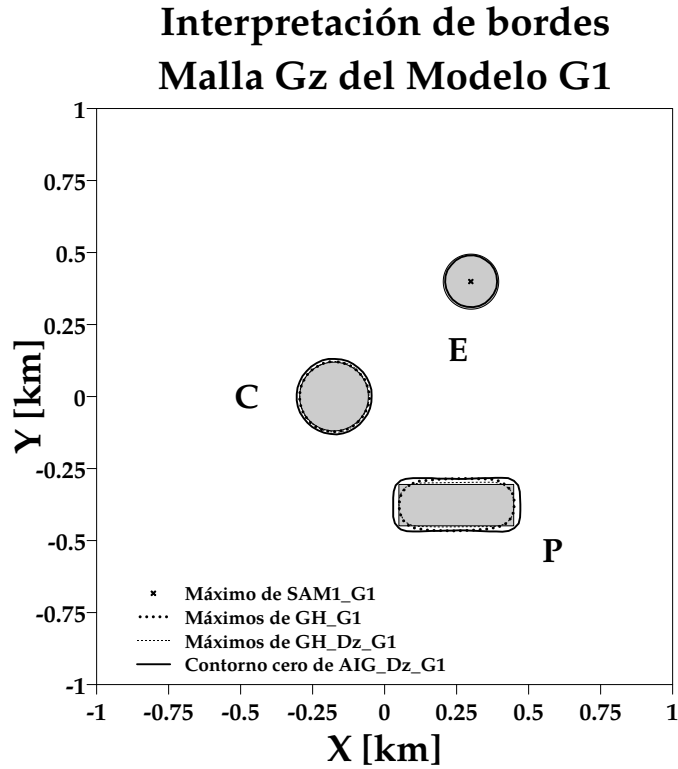


FIGURA 3.10: Bordes interpretados de los cuerpos del modelo sintético G1, a partir de procesamiento de la malla G_z de datos libres de ruido (arriba) y con ruido aleatorio añadido (abajo). Se muestran los contornos de las fuentes P (prisma), C (cilindro) y E (esfera) y se resaltan en tono gris.

Al analizar la Figura 3.10, se infiere que el método que define de una mejor forma los bordes de los cuerpos **C** y **P**, tanto para datos libres de ruido, **G1**, como para datos con ruido aleatorio añadido, **G1-RA**, es el gradiente horizontal aplicado a la derivada vertical de la anomalía gravimétrica, **GH_Dz**, mientras que la señal analítica mejorada de primer orden, **SAM1**, arroja un máximo que determina con gran exactitud la posición horizontal del centro del cuerpo esférico (**E**). El contorno cero del ángulo de inclinación de gradientes aplicado a la derivada vertical de la anomalía, **AIG_Dz**, define muy bien la proyección lateral de la extensión del cuerpo **E** en el plano XY, generando un contorno cerrado de forma circular cuyo radio es ligeramente menor al de la esfera en la malla **G1**, mientras que en la malla **G1-RA**, este contorno no presenta una forma circular, pero al no alejarse demasiado del círculo proyectado por el cuerpo **E**, puede interpretarse como la proyección de un cuerpo esférico, o un cuerpo con sección XY circular.

Interpretación semiautomática: Deconvolución de Euler 3D

La deconvolución de Euler 3D fue probada comparando las soluciones obtenidas, con la ubicación espacial de los cuerpos que forman dicho modelo. De forma similar a los métodos de realce de bordes, se utilizaron datos libres de ruido y con ruido aleatorio añadido, continuados ascendentemente 2 UM.

En la Figura 3.11 se muestran los resultados de la aplicación de la deconvolución de Euler 3D a la malla G_z del modelo **G1**. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

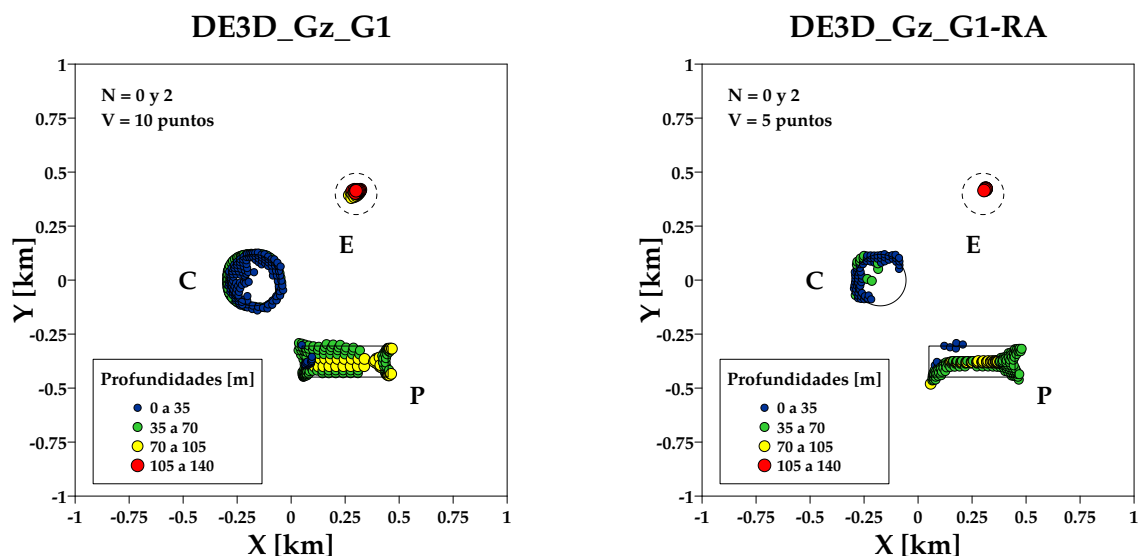


FIGURA 3.11: Resultados de la deconvolución de Euler 3D aplicada a la malla G_z del modelo sintético **G1** libre de ruido (izquierda) y con ruido aleatorio añadido (derecha). Se muestran los contornos de las fuentes **P** (prisma) y **C** (cilindro), así como la proyección de la esfera (**E**).

Las soluciones mostradas en la Figura 3.11 fueron calculadas para los índices estructurales 0 y 2, y se muestran simultáneamente, ya que no se interfieren. Las soluciones obtenidas para $N = 0$ tienden a agruparse en los bordes de los cuerpos **C** y **P**, mientras que las soluciones obtenidas con $N = 2$, lo hacen en el centro del cuerpo **E**. El tamaño de ventana que arrojó los mejores resultados fue de 10 puntos (200 [m]) para los datos libres de ruido y de 5 puntos (100 [m]) para los datos con ruido aleatorio añadido.

Las soluciones en datos sin ruido que se agrupan en el cuerpo **C** definen muy bien su contorno, a diferencia de las soluciones agrupadas en el cuerpo **P**, donde la mayoría se ubican en los vértices y algunas lo hacen al interior del cuerpo. Las profundidades para ambos cuerpos son ligeramente sobreestimadas, mientras que para el cuerpo **E**, las soluciones se agrupan en el centro y su profundidad estimada es correcta.

En el caso de los datos con ruido aleatorio añadido, las soluciones no se agrupan en todo el contorno del cuerpo **C**. Para el cuerpo **P**, al igual que en los datos libres de ruido, la mayoría de las soluciones se agrupan en los vértices, pero también lo hacen en una línea interior paralela a las aristas más largas, centrada con respecto a estas. La posición en X y Y de las soluciones estimadas para el cuerpo **E** está ligeramente desplazada hacia arriba y a la derecha, aunque la profundidad es correcta.

Considerando ahora en forma conjunta los resultados de los métodos de realce de bordes e interpretación semiautomática (Figuras 3.11 y 3.10), se pueden interpretar los bordes de los cuerpos **C** y **P** y las profundidades de sus cimas, así como la forma esférica o semiesférica del cuerpo **E**, la posición de su centro y radio, lo cual se muestra en la Tabla 3.2:

TABLA 3.2: Interpretación de cuerpos del modelo **G1**, a partir del procesamiento de mallas de anomalía gravimétrica libre de ruido (**Gz_G1**) y con ruido aleatorio añadido (**Gz_G1-RA**). Se muestran las dimensiones y profundidades interpretadas de los cuerpos, así como los errores de interpretación.

Cuerpos	Dimensiones [m]	Error _{dim} (%)	Profundidad [m]	Error _{prof} (%)
E (malla Gz_G1)	Radio: 90.00	10.00	$z_c = 122.50$	2.08
C (malla Gz_G1)	Radio: 122.00	2.40	$z = 35.00$	16.66
P (malla Gz_G1)	$L_x = 400.00$ $L_y = 155.00$	0.00 3.33	$z = 52.50$	5.00
E (malla Gz_G1-RA)	Radio: 110.00	10.00	$z_c = 122.50$	2.08
C (malla Gz_G1-RA)	Radio: 116.50	6.80	$z = 35.00$	16.66
P (malla Gz_G1-RA)	$L_x = 400.00$ $L_y = 170.00$	0.00 13.33	$z = 52.50$	5.00

3.2.2 Pruebas en mallas de gradiometría gravimétrica

Interpretación de bordes

Del mismo modo en que se integraron resultados de procesamiento para la interpretación de bordes de las mallas de anomalía gravimétrica, se interpretaron los bordes de los cuerpos del modelo **G1**, a partir de los métodos de realce de bordes de las mallas de gradiometría gravimétrica que resultaron ser más eficientes.

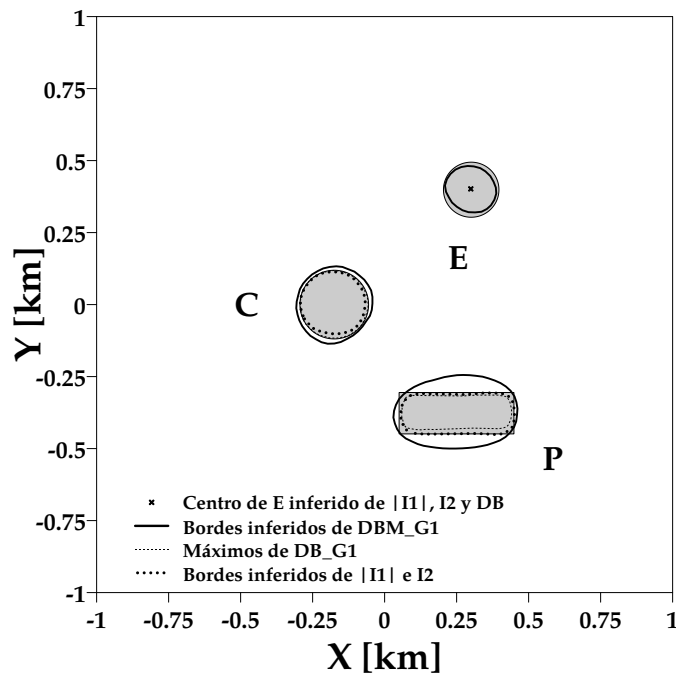
En la Figura 3.12 se muestran los bordes estimados por los métodos: análisis de invariantes I_1 e I_2 , función detectora de bordes y función detectora de bordes mejorada. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas encerrando los cuerpos resaltados en tono gris y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro).

La notación utilizada para identificar los contornos interpretados a partir de cada método es la siguiente:

- |I1|** : Valor absoluto del invariante I_1
- I2** : Invariante I_2
- DB** : Función detectora de bordes
- DBM** : Función detectora de bordes mejorada

El sufijo **G1** añadido al indicador **DB** y **DBM**, significa que el método fue aplicado a la malla de anomalía libre de ruido (Figura 3.12 superior), mientras que el sufijo **G1-RA** representa la aplicación del método a la malla de anomalía con ruido añadido (Figura 3.12 inferior).

Interpretación de bordes Mallas FTGG del modelo G1



Interpretación de bordes Mallas FTGG del modelo G1-RA

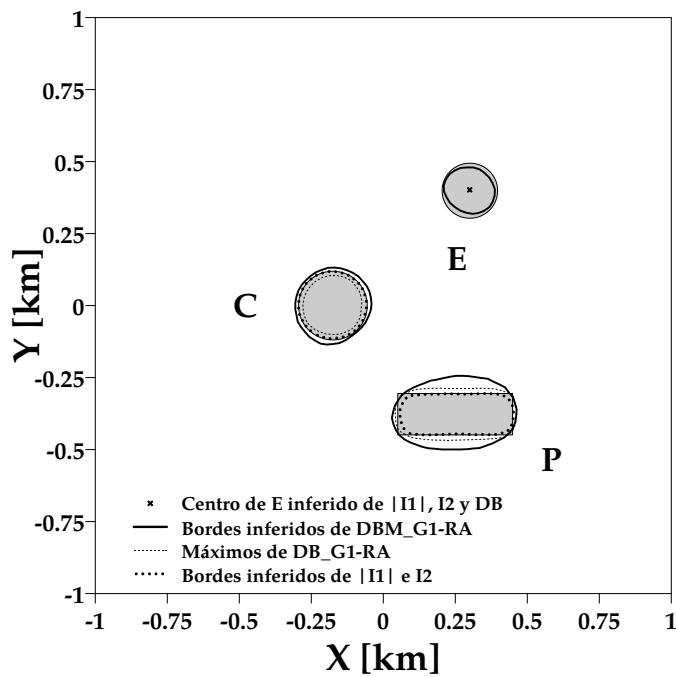


FIGURA 3.12: Bordes interpretados de los cuerpos del modelo sintético G1, a partir de procesamiento de las mallas *FTGG* con datos libres de ruido (arriba) y con ruido aleatorio añadido (abajo). Se muestran los contornos de las fuentes P (prisma), C (cilindro) y E (esfera) y se resaltan en tono gris.

Los resultados mostrados en la Figura 3.12, permite inferir que la mejor herramienta para definir los bordes de los cuerpos **C** y **P**, tanto en datos libres de ruido, **G1**, como en datos con ruido aleatorio añadido, **G1-RA**, es el cálculo de invariantes $|I1|$ e $I2$ y su análisis simultáneo. Por otro lado, la posición del centro de la esfera (**E**) queda determinada con gran exactitud a partir de los máximos de estos invariantes y del máximo de la función detectora de bordes, **DB**. Finalmente, el contorno cero de la función detectora de bordes mejorada, **DBM**, aproxima relativamente bien el contorno del cuerpo **C**, así como la proyección de **E** en el plano **XY**, aunque no define muy bien los bordes del cuerpo **P**, a pesar de ser poco sensible al ruido aleatorio añadido.

Interpretación semiautomática: Deconvolución de Euler 3D del Tensor Gravimétrico

En la Figura 3.13 se muestran los resultados de la deconvolución de Euler 3D del tensor gravimétrico del modelo sintético **G1**. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua.

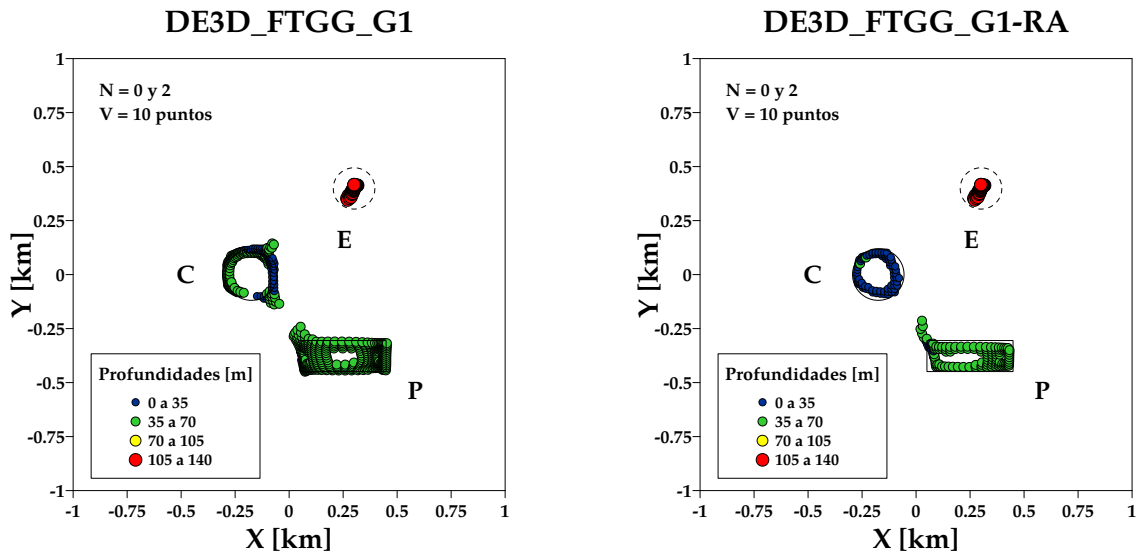


FIGURA 3.13: Resultados de la deconvolución de Euler 3D del tensor gravimétrico del modelo sintético **G1** libre de ruido (izquierda) y con ruido aleatorio añadido (derecha). Se muestran los contornos de las fuentes **P** (prisma) y **C** (cilindro), así como la proyección de la esfera (**E**).

Las soluciones mostradas en la Figura 3.13 fueron calculadas para los índices estructurales 0 y 2, y del mismo modo que para la Figura 3.11, se muestran simultáneamente. Las soluciones obtenidas para $N = 0$ tienden a agruparse en los bordes de los cuerpos **C** y **P**, mientras que las soluciones obtenidas con $N = 2$, lo hacen en el centro del cuerpo **E**. El

tamaño de ventana que arrojó los mejores resultados fue de 10 puntos ($200 [m]$) para datos libres de ruido y con ruido aleatorio añadido.

Las soluciones en datos sin ruido y con ruido aleatorio añadido que se agrupan en el cuerpo **C**, definen muy bien su contorno, con excepción de algunas soluciones que se salen ligeramente del contorno en dos direcciones: hacia abajo y a la derecha y hacia arriba y a la derecha sólo en datos libres de ruido. Las soluciones agrupadas en el cuerpo **P**, para ambos tipos de datos, se ubican ligeramente en el interior del cuerpo y algunas de ellas se salen ligeramente en el vértice superior izquierdo, lo cual puede ser reflejo de la influencia del cuerpo **C**. Para el cuerpo **E**, las soluciones muestran el mismo comportamiento en datos libres de ruido y con ruido añadido, agrupándose la mayoría en el centro, mientras algunas se desvían ligeramente a la izquierda y hacia abajo, aunque la profundidad estimada para todas ellas es correcta.

Ahora, en la Tabla 3.3 se muestra la interpretación de cuerpos a partir del procesamiento de las mallas *FTGG*, analizando simultáneamente los resultados de procesos de realce de bordes y la deconvolución de Euler 3D del Tensor Gravimétrico (Figura 3.13):

TABLA 3.3: Interpretación de cuerpos del modelo **G1**, a partir del procesamiento de mallas del tensor de gradiometría gravimétrica libres de ruido (**FTGG_G1**) y con ruido aleatorio añadido (**FTGG_G1-RA**). Se muestran las dimensiones y profundidades interpretadas de los cuerpos, así como los errores de interpretación.

Cuerpos	Dimensiones [m]	Error _{dim} (%)	Profundidad [m]	Error _{prof} (%)
E (malla FTGG_G1)	Radio: 89.00	11.00	$z_c = 122.50$	2.08
C (malla FTGG_G1)	Radio: 122.50	2.00	$z = 35.00$	16.66
P (malla FTGG_G1)	$L_x = 400.00$	0.00	$z = 52.50$	5.00
	$L_y = 145.00$	3.33		
E (malla FTGG_G1-RA)	Radio: 86.00	14.00	$z_c = 122.50$	2.08
C (malla FTGG_G1-RA)	Radio: 122.5	2.00	$z = 35.00$	16.66
P (malla FTGG_G1-RA)	$L_x = 397.00$	0.75	$z = 52.50$	5.00
	$L_y = 146.00$	2.66		

3.3 Pruebas del algoritmo de inversión SA en datos sintéticos

Para probar la efectividad y eficiencia del algoritmo SA diseñado, se aplicó al modelo sintético **G2**. El propósito fundamental de esta prueba fue invertir un modelo sintético relativamente simple para detectar sin dificultad las diferencias entre el modelo de referencia y el modelo generado por la inversión y de este modo, reconocer los alcances y limitaciones del algoritmo de inversión en un escenario ideal (datos sin ruido) y uno cercano a la realidad (datos con ruido pseudoaleatorio añadido).

La implementación del método de inversión en datos sintéticos también sirvió para realizar un análisis y selección de parámetros requeridos por el SA que influyen en su rendimiento y cuyos resultados se muestran en la sección 3.3.3.

3.3.1 Algoritmo SA implementado en mallas de anomalía gravimétrica

La implementación del algoritmo SA, tal como se planteó en este trabajo (subcapítulo 2.3.1), requiere de un kernel de sensibilidades, G_{ij} , cuyos elementos se calculan con base en la solución analítica para el prisma (Plouff, 1976), presentada en el subcapítulo 2.1.2:

Reescribiendo la solución para la componente g_z (Ec. 2.33):

$$g_z = \gamma \rho \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{z}{\sqrt{x^2 + y^2 + z^2}^3} dx dy dz$$

Ahora, reescribiendo el problema directo como el producto del kernel por el vector de parámetros (Ec. 2.73), donde los parámetros son las densidades de los prismas del ensamble:

$$g_z = G_{ij} \cdot \rho$$

Por comparación directa entre estas dos ecuaciones, el kernel de sensibilidades se calculará resolviendo la siguiente integral:

$$G = \gamma \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{z}{\sqrt{x^2 + y^2 + z^2}^3} dx dy dz$$

lo cual, en la práctica, se puede realizar con ayuda del subprograma **VectorGBox.f08** (Anexo B).

El kernel de sensibilidades generado para la inversión de datos de anomalía gravimétrica del modelo sintético **G2** tiene las siguientes características:

Número de renglones : 10201 (Número de datos observados)
Número de columnas : 1000 (Número de parámetros por invertir)
Espacio en disco : 77.827 MB
Máximo del kernel : 0.169592×10^{-2}
Mínimo del kernel : 0.139971×10^{-6}

Una vez calculado el kernel, se necesita un modelo inicial que puede contener información a priori de la zona estudiada (si se tiene), o carecer de ella por completo, proponiéndose un modelo con densidades nulas o aleatorias dentro de un rango conocido. En este caso, al tratarse de la inversión de datos de un modelo sintético, se propuso un modelo inicial aleatorio (Figuras 3.14 y 3.15), que además es ideal para probar la efectividad del algoritmo de inversión, al no alimentarlo con ningún tipo de información.

Los parámetros utilizados en los algoritmos SA y SA-SA, implementados para invertir la malla de datos de la componente g_z del vector de aceleración gravitacional, se muestran en la Tabla 3.4:

TABLA 3.4: Parámetros iniciales de los algoritmos SA y SA-SA utilizados en la inversión de la malla de anomalía gravimétrica, G_z , del modelo G2.

Parámetros Iniciales	Algoritmo SA	Algoritmo SA-SA
Temperatura inicial:	0.01	0.1
Descensos totales de temperatura:	1000	1000
Factor RT:	0.95	0.95
Ciclos de estabilidad termodinámica:	25	25

Cabe mencionar que los parámetros iniciales para los algoritmos de inversión SA y SA-SA, aplicados en datos con ruido, fueron exactamente los mismos que para los datos sin ruido y se determinaron a partir del análisis paramétrico que se mostrará en el subcapítulo 3.3.3.

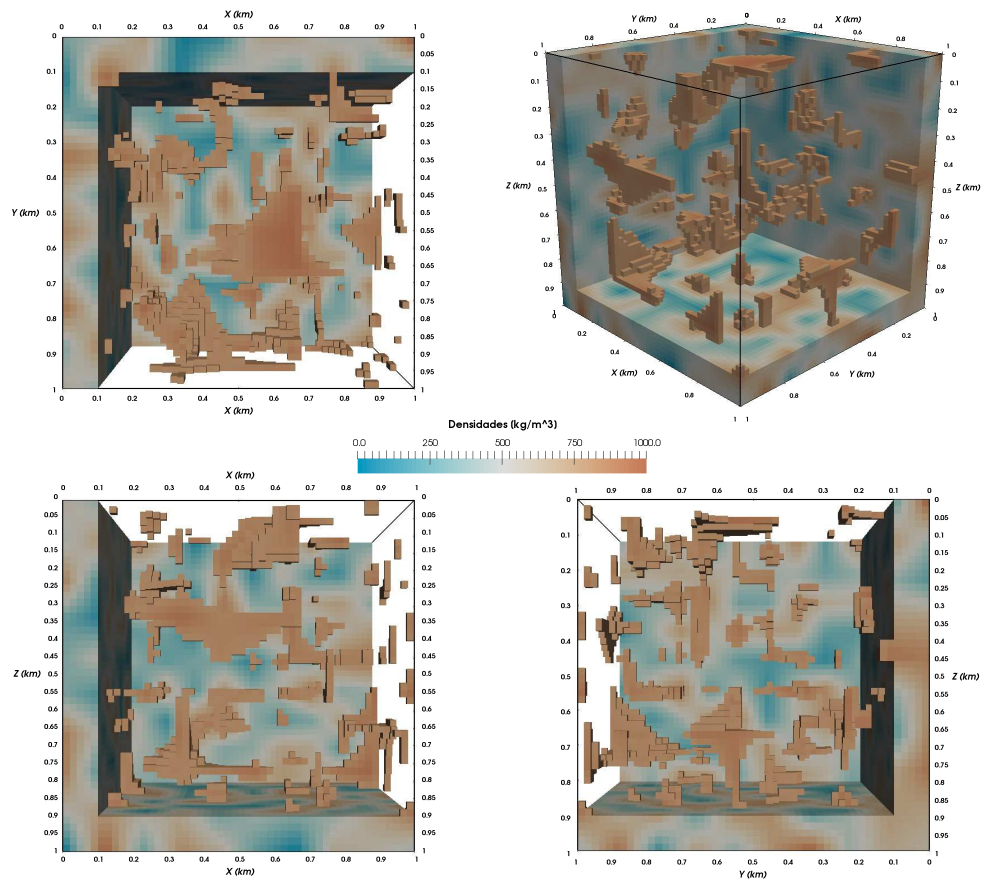


FIGURA 3.14: Modelo inicial aleatorio para la inversión de la componente g_z del vector de aceleración gravitacional y las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica.

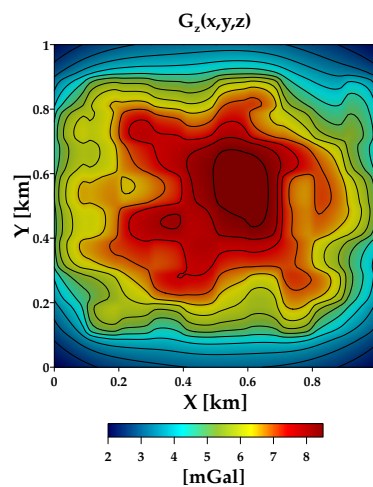


FIGURA 3.15: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo inicial (Figura 3.14), observado en una malla regular de 101×101 puntos.

Los parámetros relacionados con el desempeño de los algoritmos empleados en la inversión se definieron al finalizar su ejecución y se muestran en la Tabla 3.5:

TABLA 3.5: Resultados de la inversión de la malla de anomalía gravimétrica, G_z , del modelo G2, a través de los algoritmos SA y SA-SA.

	Datos sin ruido		Datos con ruido	
	Algoritmo SA	Algoritmo SA-SA	Algoritmo SA	Algoritmo SA-SA
Temperatura final:	0.529×10^{-24}	0.529×10^{-23}	0.529×10^{-24}	0.529×10^{-23}
Energía del modelo inicial:	2.834	2.834	2.833	2.833
Energía del modelo final:	4.503×10^{-4}	4.560×10^{-4}	6.657×10^{-4}	6.701×10^{-4}
Modelos evaluados:	20,000,000	20,000,000	20,000,000	20,000,000
Modelos aceptados (%):	21.919 %	21.492 %	21.913 %	21.452 %
Tiempo de cómputo [seg]:	662.764	848.812	647.472	859.796

Ahora, en la Figura 3.16 se muestran las curvas de convergencia generadas por los algoritmos de inversión de datos gravimétricos del modelo G2, cuyos resultados fueron capturados en la Tabla 3.5:

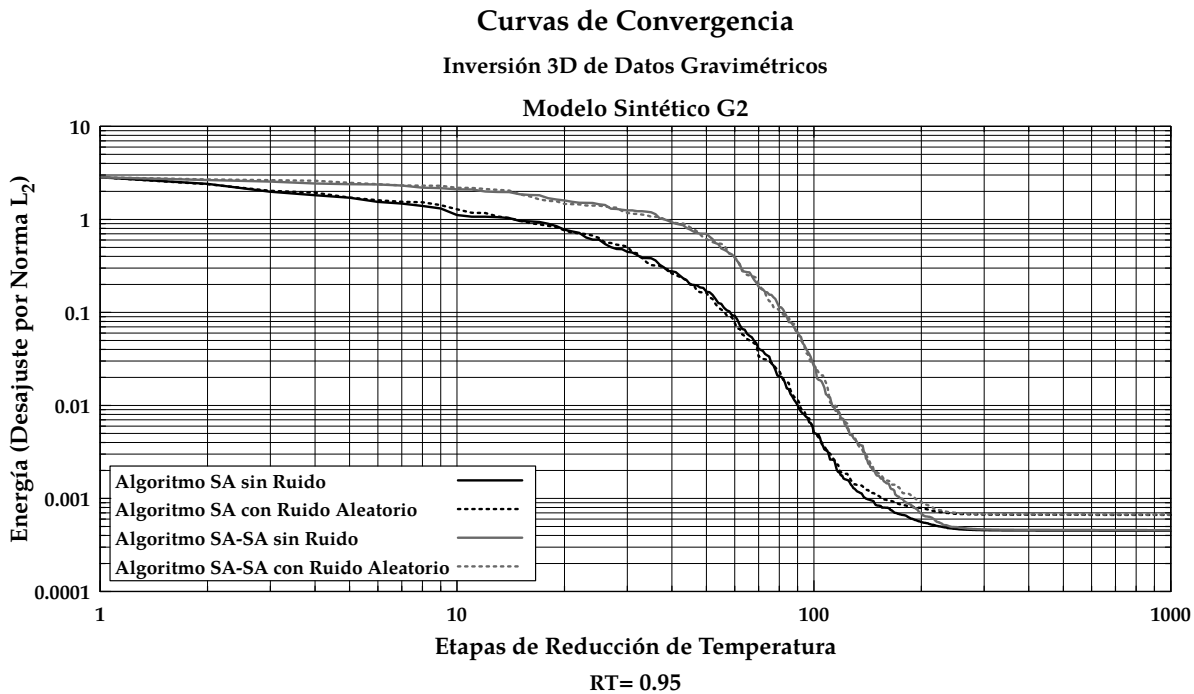


FIGURA 3.16: Curvas de convergencia generadas por la inversión de datos gravimétricos del modelo G2 con los algoritmos SA y SA-SA.

Las configuraciones 3D de los modelos generados al invertir datos sin ruido y con ruido pseudoaleatorio añadido, por medio de los algoritmos SA y SA-SA, así como sus anomalías gravimétricas generadas, se despliegan en las Figuras 3.17 a 3.24:

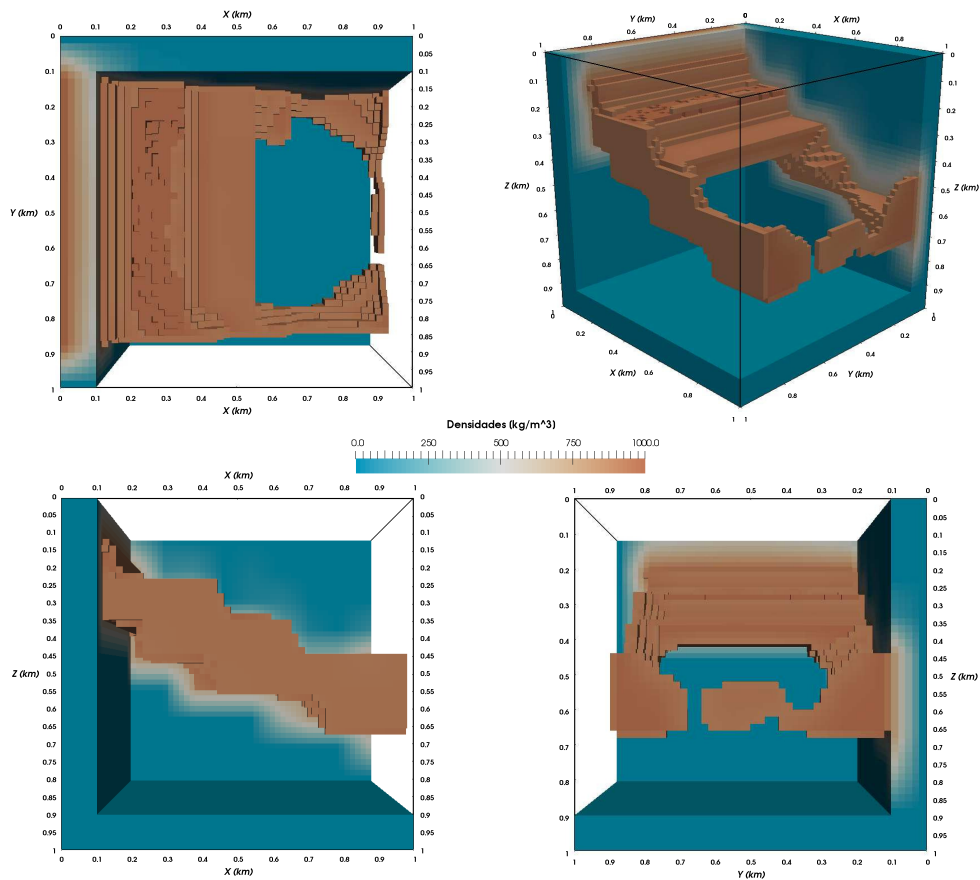


FIGURA 3.17: Modelo final generado por la inversión de la componente g_z del vector de aceleración gravitacional, a través del algoritmo SA.

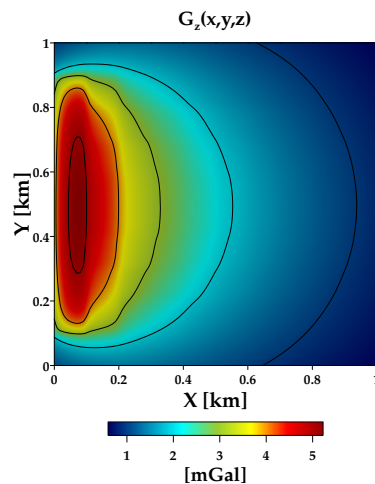


FIGURA 3.18: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.17, observado en una malla regular de 101×101 puntos.

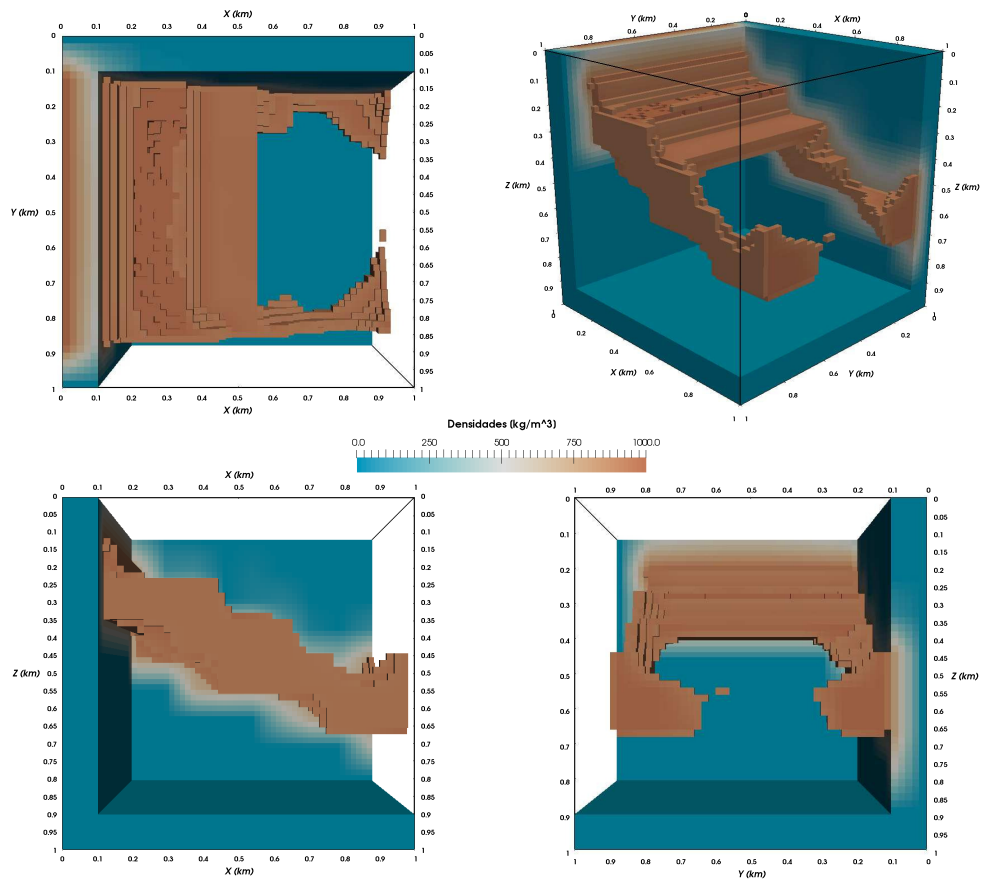


FIGURA 3.19: Modelo final generado por la inversión de la componente g_z con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del vector de aceleración gravitacional, a través del algoritmo SA.

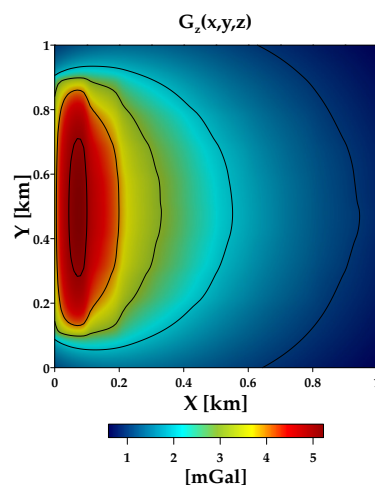


FIGURA 3.20: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.19, observado en una malla regular de 101×101 puntos.

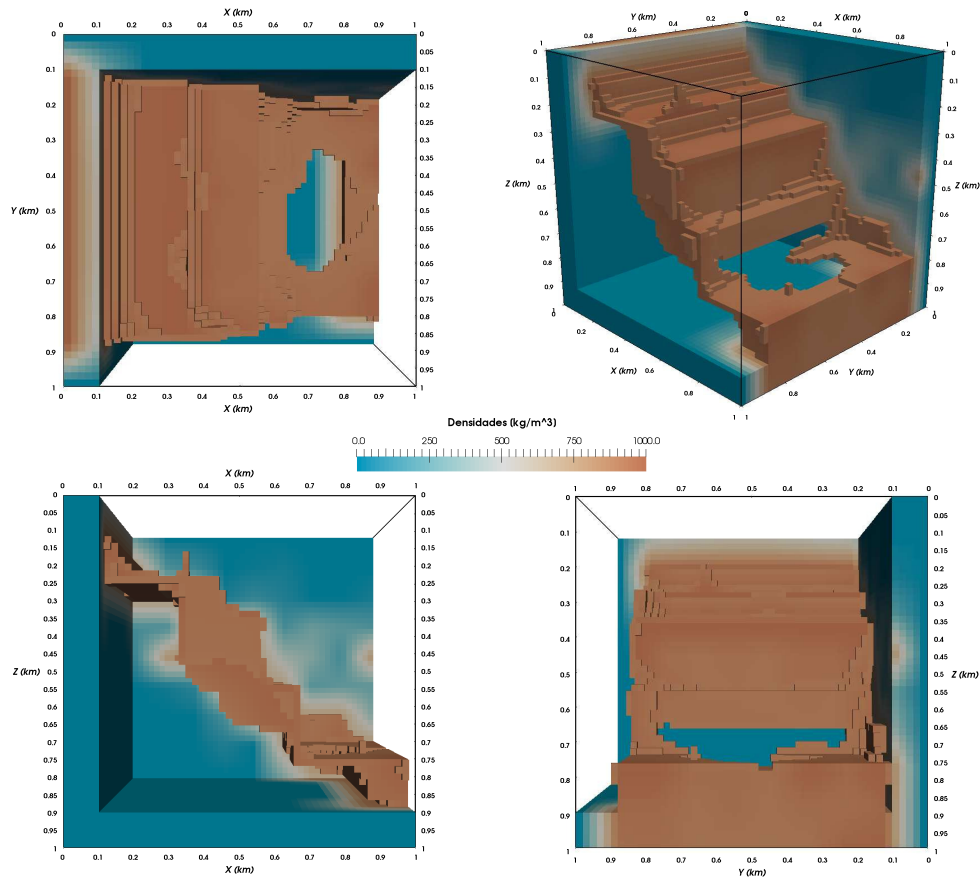


FIGURA 3.21: Modelo final generado por la inversión de la componente g_z del vector de aceleración gravitacional, a través del algoritmo SA-SA.

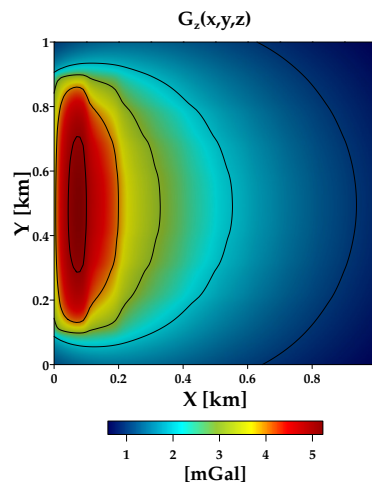


FIGURA 3.22: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.21, observado en una malla regular de 101×101 puntos.

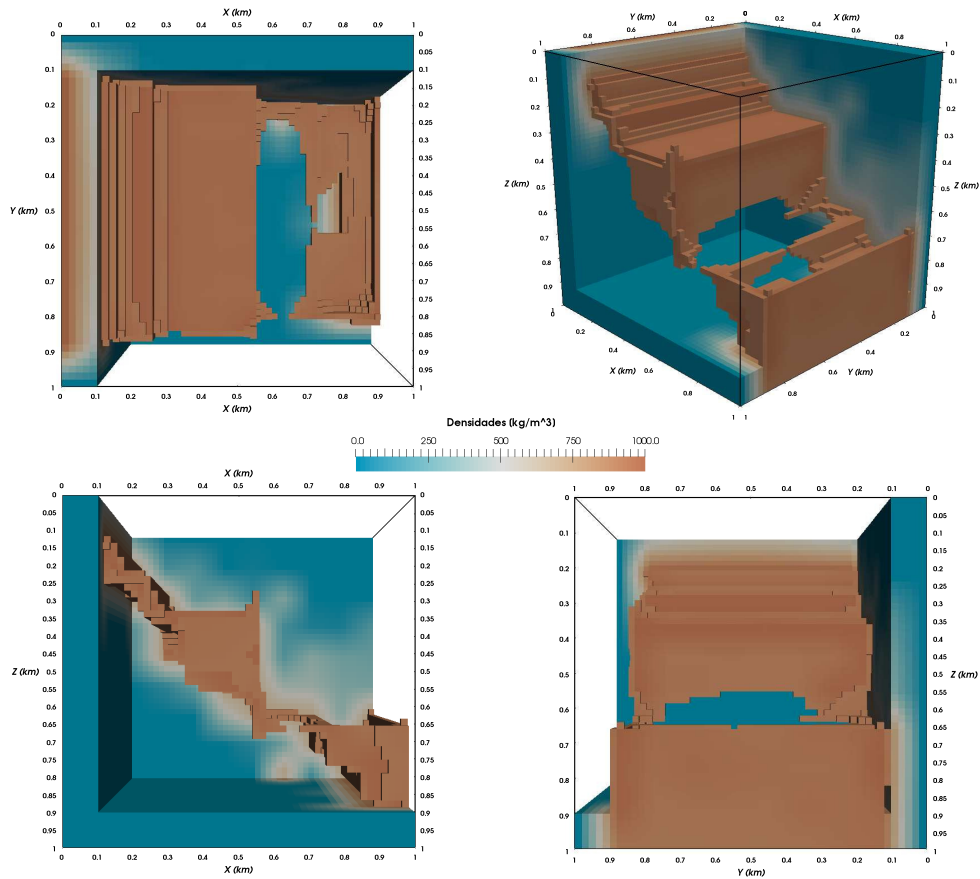


FIGURA 3.23: Modelo final generado por la inversión de la componente g_z con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del vector de aceleración gravitacional, a través del algoritmo SA-SA.

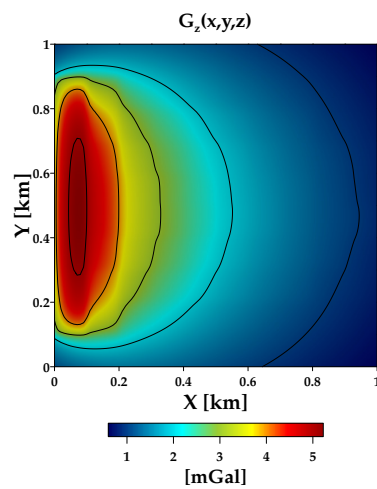


FIGURA 3.24: Superficie de la componente g_z del vector de aceleración gravitacional, generado por el modelo invertido de la Figura 3.23, observado en una malla regular de 101×101 puntos.

A partir de los resultados mostrados en la Tabla 3.5, así como de las curvas de convergencia generadas por los métodos de inversión (Figura 3.16) y la configuración 3D de los modelos óptimos y sus anomalías (Figuras 3.17 a 3.24), se infiere que los algoritmos SA y SA-SA generan modelos con niveles de error muy similares al aplicarse ambos al mismo tipo de datos (sin ruido, o con ruido añadido), siendo el algoritmo SA-SA el que produce modelos cuya geometría se aproxima mejor a la del modelo sintético **G2**, lo cual refleja el efecto del factor amplificador de sensibilidad incluido.

Por otro lado, las similitudes entre las configuraciones 3D de los modelos obtenidos a partir de la inversión de datos con ruido añadido, con respecto a la inversión de datos libres de ruido, así como sus anomalías, indican que los algoritmos SA y SA-SA son poco sensibles a dicho ruido, lo que corrobora su aplicabilidad en datos reales.

3.3.2 Algoritmo SA implementado en mallas de tensor gravimétrico

La implementación del algoritmo SA en mallas de gradiometría gravimétrica se realizó de forma análoga a la implementación en mallas de la componente g_z del vector de aceleración gravitacional (subcapítulo 3.3.1), pero a diferencia de esta, donde los datos por invertir consisten en una malla de observaciones, ahora se tienen seis mallas de datos, correspondientes con las seis diferentes componentes del tensor de gradiometría gravimétrica, por lo que fue necesario calcular un kernel de sensibilidades para cada componente a partir de la solución analítica para el prisma (Nagy et al., 2000), presentada en el subcapítulo 2.1.2:

La expresión generalizada para el cálculo de las componentes del tensor de gradiometría gravimétrica, T_{ij} , basada en el potencial de una fuente prismática rectangular (Ec. 2.31) y la definición del tensor (Ec. 2.9) es:

$$T_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} \left[\gamma \rho \int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{1}{\sqrt{x^2 + y^2 + z^2}} dx dy dz \right]$$

con: $x_i, x_j = x, y, z$

Reescribiendo el problema directo como el producto del kernel correspondiente a la componente observada, por el vector de parámetros (densidades) del ensamble (Ec. 2.73):

$$T_{ij} = G_{ij} \cdot \rho$$

Entonces, el kernel de sensibilidades se obtendrá con la siguiente expresión:

$$G = \gamma \frac{\partial^2}{\partial x_i \partial x_j} \left[\int_{z_1}^{z_2} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{1}{\sqrt{x^2 + y^2 + z^2}} dx dy dz \right]$$

donde los elementos de G_{ij} se pueden calcular numéricamente con ayuda del subprograma **TensorGBox.f08** (Anexo B).

Los kernels de sensibilidades generados para la inversión de mallas de tensor gravimétrico del modelo sintético **G2** tienen las siguientes características:

Número de renglones:	10201 (Número de datos observados)
Número de columnas:	1000 (Número de parámetros por invertir)
Espacio en disco:	77.827 MB
Valores máximos:	$\max(G_{xx})= 0.129768$; $\max(G_{xy})= 0.234480$; $\max(G_{xz})= 0.532561$ $\max(G_{yy})= 0.129768$; $\max(G_{yz})= 0.532561$; $\max(G_{zz})= 0.361284$
Valores mínimos:	$\min(G_{xx})= -0.215882$; $\min(G_{xy})= -0.234479$; $\min(G_{xz})= -0.532560$ $\min(G_{yy})= -0.215882$; $\min(G_{yz})= -0.532560$; $\min(G_{zz})= -0.030399$

Los parámetros utilizados en los algoritmos SA y SA-SA, implementados para invertir las mallas de datos de tensor gravimétrico se muestran en la Tabla 3.6:

TABLA 3.6: Parámetros iniciales de los algoritmos SA y SA-SA utilizados en la inversión de mallas de tensor gravimétrico del modelo **G2**.

Parámetros Iniciales	Algoritmo SA	Algoritmo SA-SA
Temperatura inicial:	0.01	0.1
Descensos totales de temperatura:	1000	1000
Factor RT:	0.95	0.95
Ciclos de estabilidad termodinámica:	25	25

Del mismo modo que en el caso de la inversión gravimétrica, para los datos del tensor con ruido pseudoaleatorio añadido, los parámetros iniciales para los algoritmos de inversión SA y SA-SA, fueron exactamente los mismos que para las mallas de datos sin ruido y se determinaron a partir del análisis paramétrico que se mostrará en el subcapítulo 3.3.3.

En la figura 3.25 se despliegan las superficies de las componentes tensoriales debidas al modelo inicial generado de forma aleatoria (Figura 3.14) en el que se probaron los algoritmos de inversión SA y SA-SA.

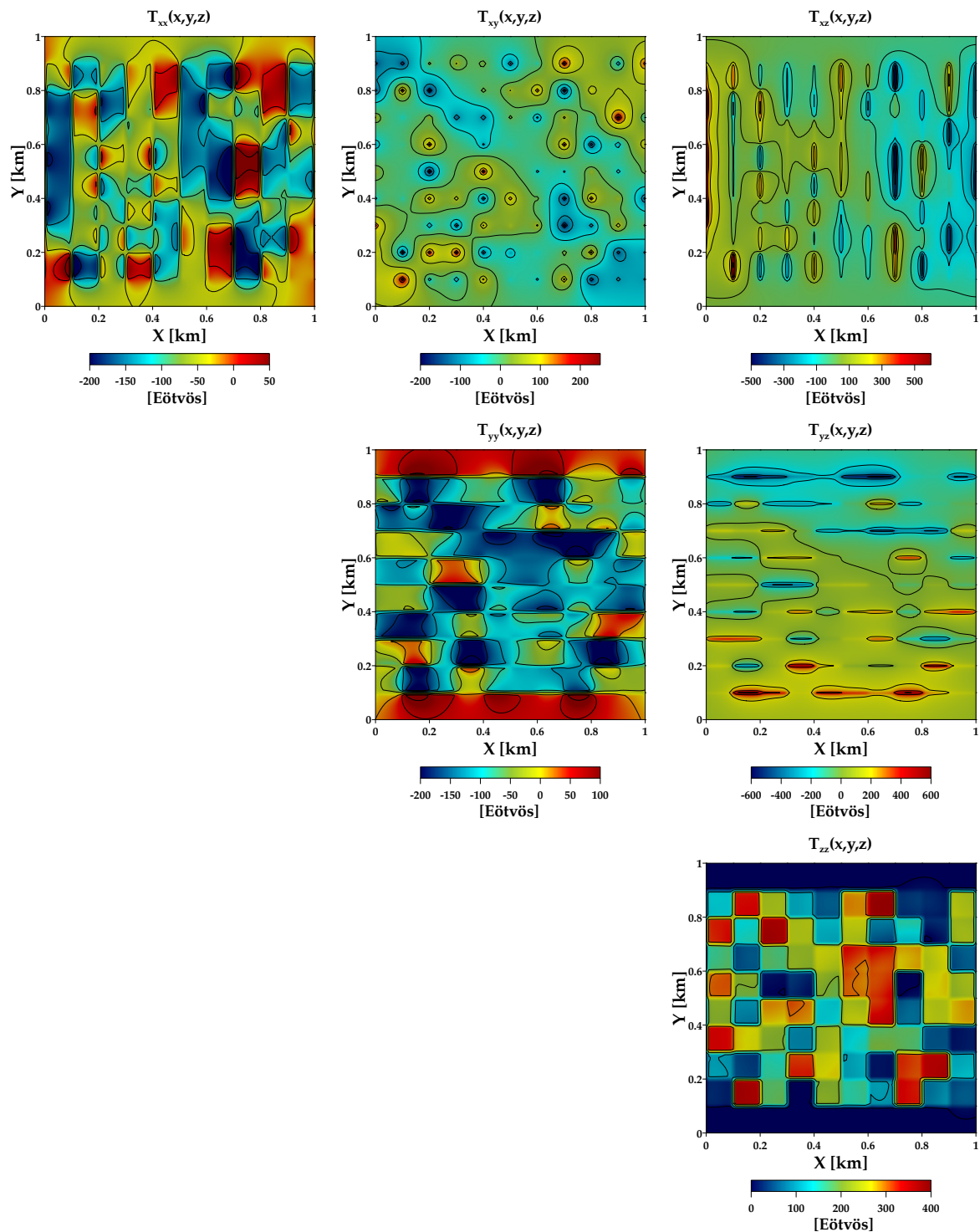


FIGURA 3.25: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo inicial (Figura 3.14), observado en una malla regular de 101×101 puntos.

Los parámetros relacionados con el desempeño de los algoritmos empleados en la inversión de las mallas del tensor gravimétrico del modelo **G2** sin ruido y con ruido pseudoaleatorio añadido, definidos al terminar la ejecución, se muestran en la Tabla 3.7:

TABLA 3.7: Resultados de la inversión de las mallas de tensor gravimétrico del modelo **G2**, a través de los algoritmos SA y SA-SA.

	Datos sin ruido		Datos con ruido	
	Algoritmo SA	Algoritmo SA-SA	Algoritmo SA	Algoritmo SA-SA
Temperatura final:	0.529×10^{-24}	0.529×10^{-23}	0.529×10^{-24}	0.529×10^{-23}
Energía del modelo inicial:	3.267	3.267	3.267	3.267
Energía del modelo final:	2.704×10^{-2}	2.704×10^{-2}	3.874×10^{-2}	3.874×10^{-2}
Modelos evaluados:	20,000,000	20,000,000	20,000,000	20,000,000
Modelos aceptados (%):	20.018 %	21.069 %	20.044 %	21.111 %
Tiempo de cómputo [seg]:	3,688.468	4,321.872	3,696.052	4,313.404

En la Figura 3.26 se muestran las curvas de convergencia generadas por los algoritmos de inversión de mallas de tensor gravimétrico del modelo **G2**, cuyos resultados fueron capturados en la Tabla 3.7:

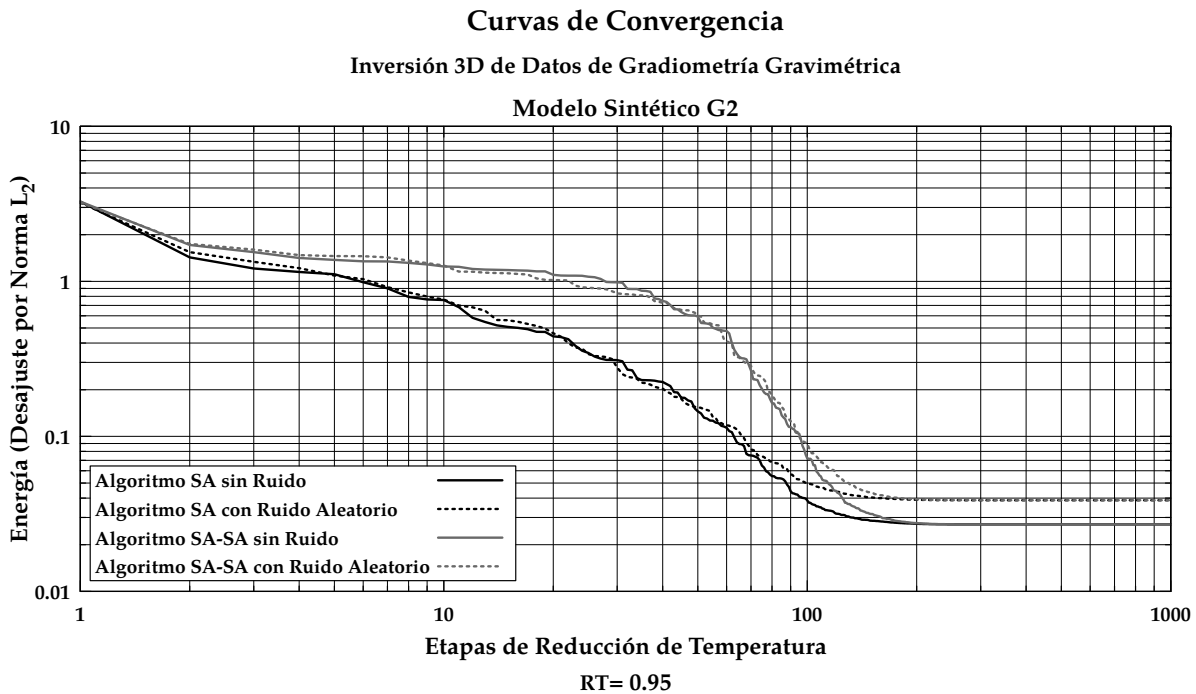


FIGURA 3.26: Curvas de convergencia generadas por la inversión del tensor gravimétrico del modelo **G2** con los algoritmos SA y SA-SA.

Las configuraciones 3D de los modelos generados al invertir datos sin ruido y con ruido pseudoaleatorio añadido, por medio de los algoritmos SA y SA-SA, así como las mallas de componentes tensoriales generadas, se despliegan en las Figuras 3.27 a 3.34:

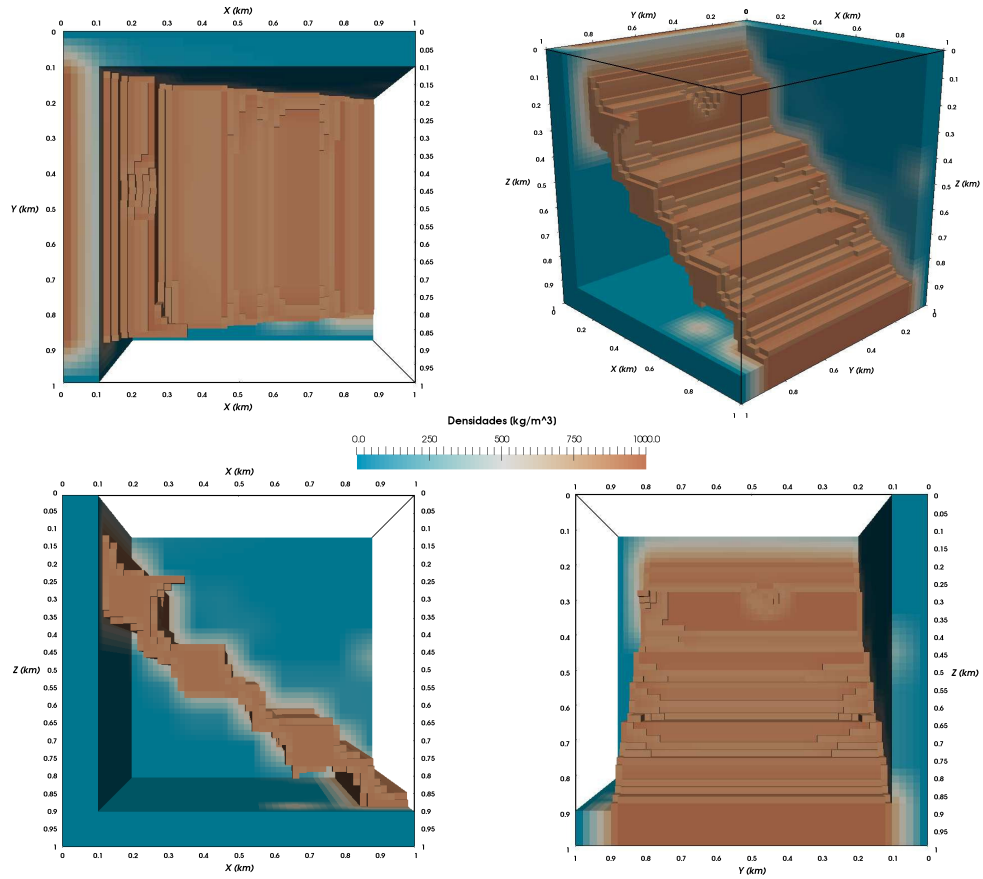


FIGURA 3.27: Modelo final generado por la inversión de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, a través del algoritmo SA.

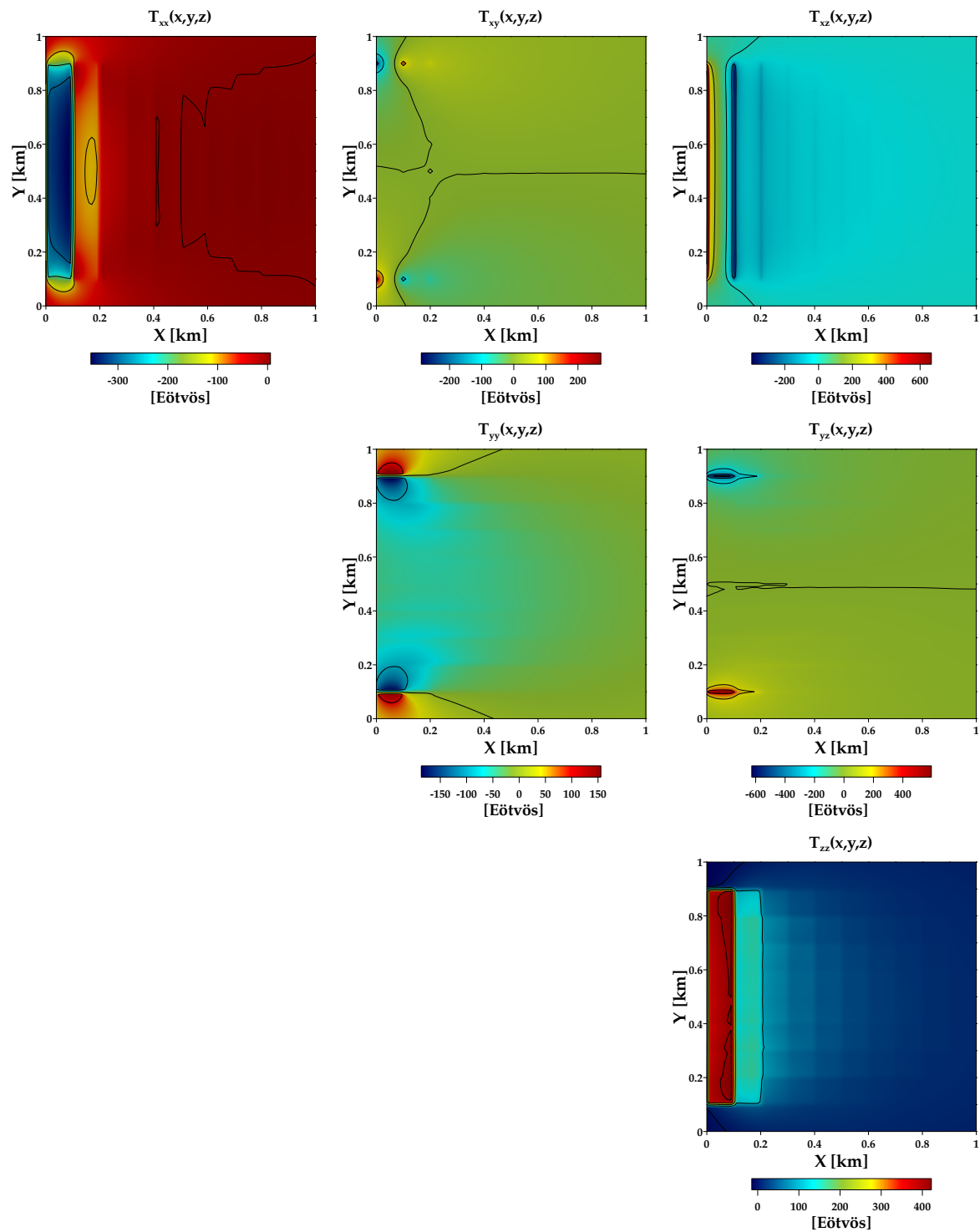


FIGURA 3.28: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.27, observado en una malla regular de 101×101 puntos.

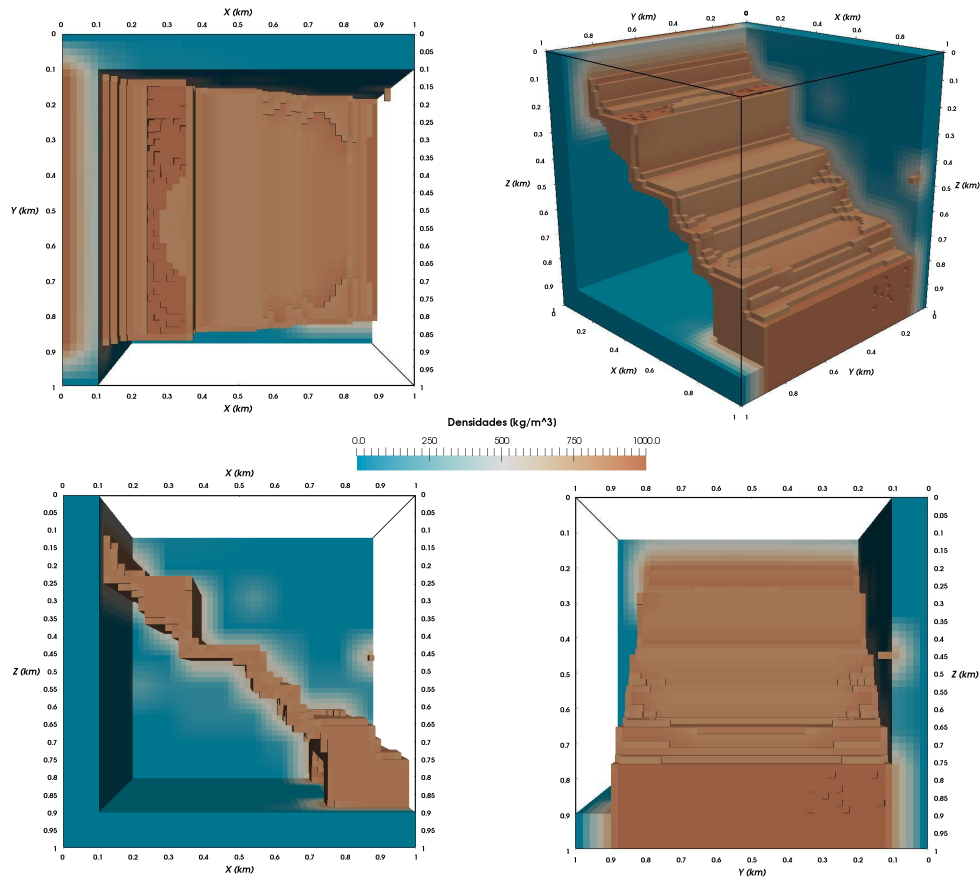


FIGURA 3.29: Modelo final generado por la inversión de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del tensor de gradiometría gravimétrica, a través del algoritmo SA.

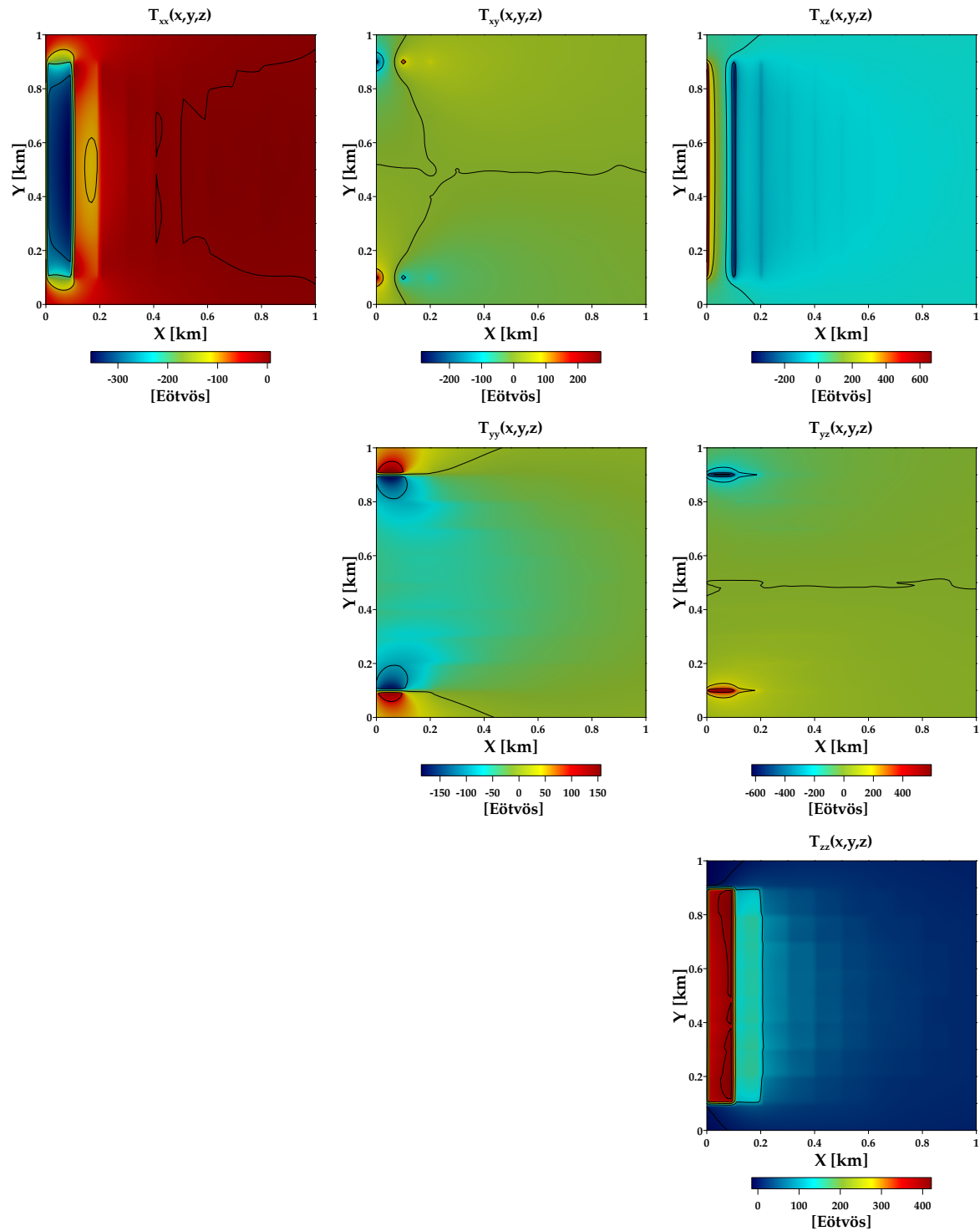


FIGURA 3.30: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.29, observado en una malla regular de 101×101 puntos.

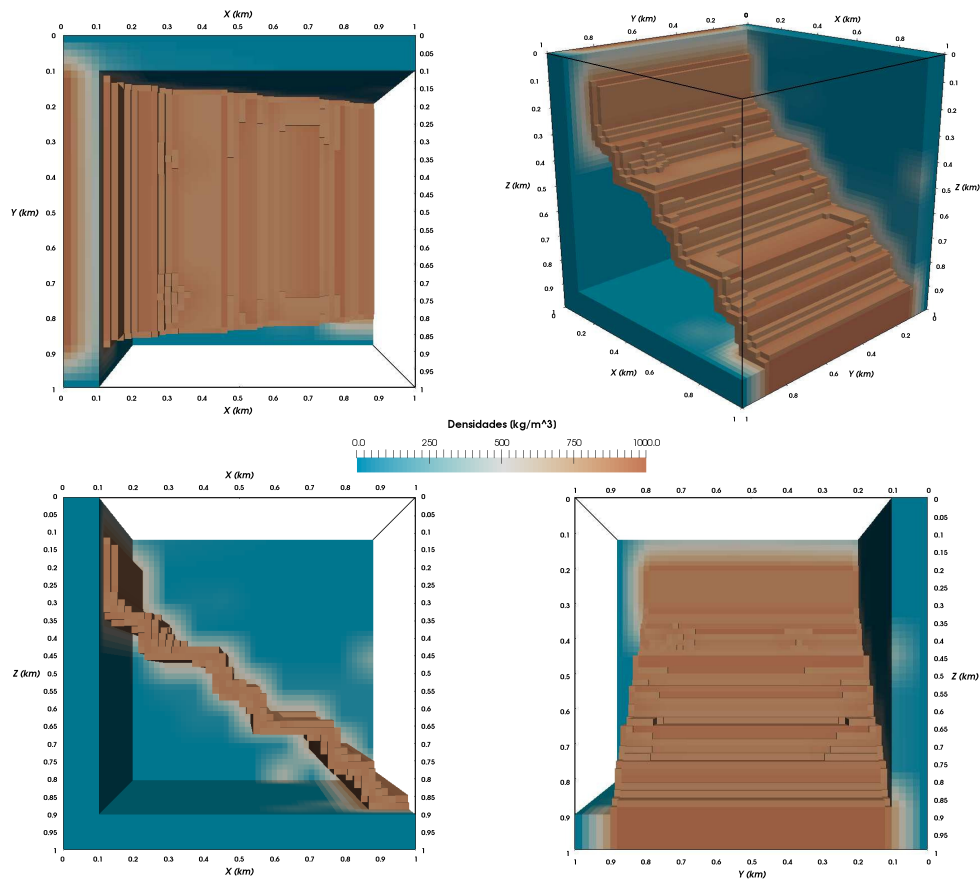


FIGURA 3.31: Modelo final generado por la inversión de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, a través del algoritmo SA-SA.

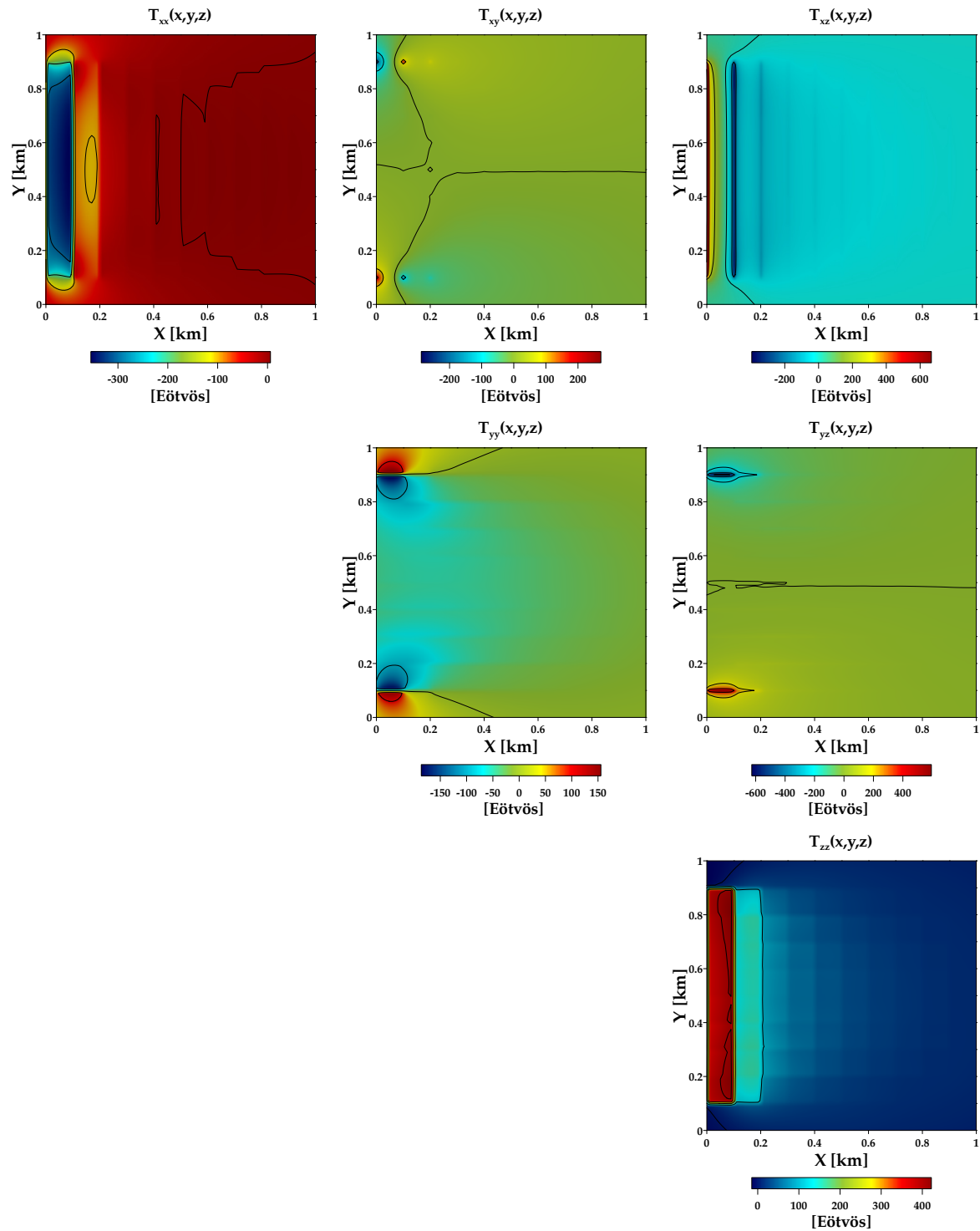


FIGURA 3.32: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.31, observado en una malla regular de 101×101 puntos.

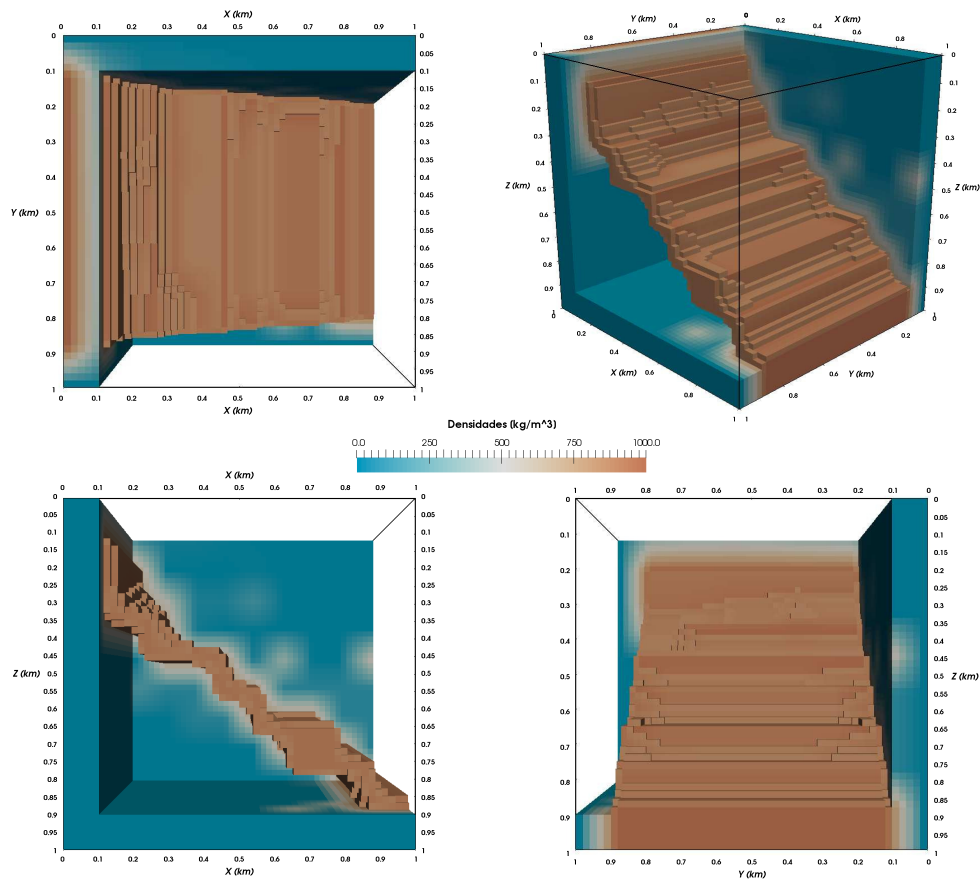


FIGURA 3.33: Modelo final generado por la inversión de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} con ruido pseudoaleatorio añadido, $N(0, 0.05\sigma)$, del tensor de gradiometría gravimétrica, a través del algoritmo SA-SA.

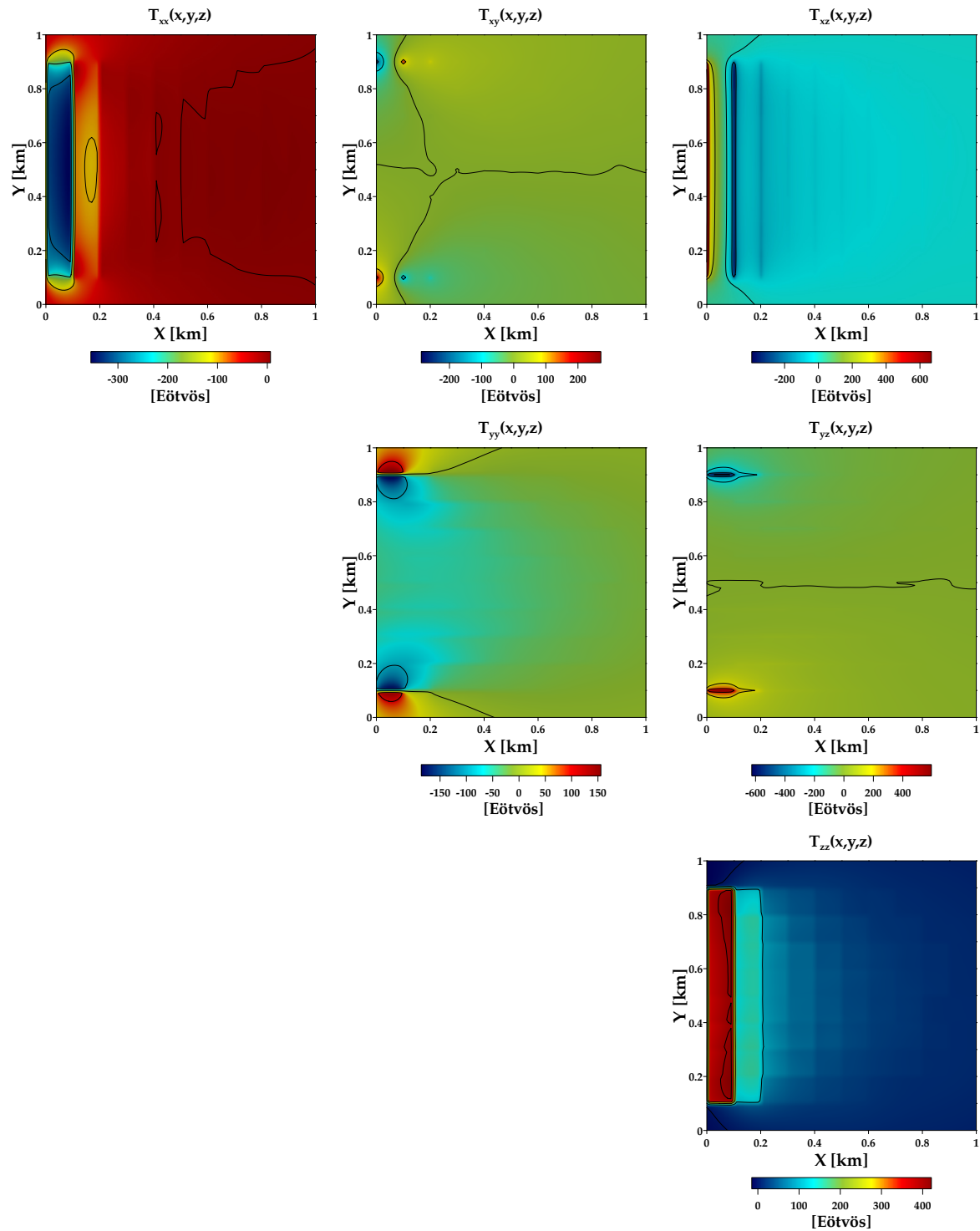


FIGURA 3.34: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravimétrica, generado por el modelo invertido de la Figura 3.33, observado en una malla regular de 101×101 puntos.

De manera similar a la inversión de datos de anomalía gravimétrica del modelo **G2**, los resultados de la inversión de las mallas de componentes del tensor gravimétrico mostrados en la Tabla 3.7 y las curvas de convergencia de los procesos de inversión ejecutados (Figura 3.26), así como de las con figuraciones 3D de los modelos óptimos obtenidos y sus correspondientes mallas de componentes tensoriales (Figuras 3.27 a 3.34), se observa que los algoritmos SA y SA-SA aplicados al mismo tipo de datos (sin ruido o con ruido añadido), generan modelos con niveles de error muy parecidos, y a diferencia de la inversión de datos de anomalía gravimétrica, la geometría de los modelos generados por ambos métodos de inversión es cercana a la del modelo sintético.

El ruido aleatorio añadido a las mallas de componentes tensoriales tampoco influye en los modelos generados por los algoritmos de inversión SA y SA-SA, lo que corrobora su utilidad en datos reales.

3.3.3 Análisis del SA y criterios de selección de parámetros iniciales

El rendimiento de los algoritmos SA y SA-SA depende de varios parámetros, cuyos valores deben ser introducidos previamente a su ejecución. Estos parámetros son: Número máximo de reducciones de temperatura, $nIter$, temperatura inicial, T_0 , factor de enfriamiento, RT , amplitud máxima de perturbación, VM , número de perturbaciones por ciclo de temperatura, N_T y tolerancia o energía máxima aceptable para la detención del algoritmo, Tol .

Para determinar los valores de T_0 , RT y VM se realizó un barrido paramétrico ejecutando los algoritmos de inversión para diferentes combinaciones de parámetros, obteniendo para cada combinación, curvas de aceptación de modelos probados en cada reducción de temperatura, así como sus curvas de convergencia, planteando criterios para seleccionar la combinación paramétrica que arrojará los mejores resultados.

En la Tabla 3.8 se muestran los valores asignados para estos tres parámetros en el barrido propuesto:

TABLA 3.8: Valores asignados a T_0 , RT y VM en el barrido paramétrico.

Parámetro	Valores
T_0	10, 1, 0.1, 0.01, 0.001, 0.0001, 0.00001 y 0.000001
RT	0.6, 0.7, 0.8, 0.9, 0.95 y 0.98
VM	100 %, 50 %, 25 %, 10 %, 5 % y 1 % (% de la amplitud máxima de la variación de parámetros)

El barrido paramétrico, que consistió en un total de 288 ejecuciones de cada algoritmo de inversión, con $nIter = 2000$, se realizó únicamente para $N_T = 1$, considerando que la combinación de parámetros obtenida representa el mínimo rendimiento aceptable, de tal modo que la combinación de parámetros que cumple con los criterios establecidos para $N_T = 1$, también cumplirá dichos criterios para $N_T > 1$.

Los criterios para la selección de parámetros iniciales de los algoritmos SA y SA-SA se establecieron a partir del cumplimiento de algunas condiciones relacionadas con el desempeño de dichos algoritmos y se muestran a continuación:

Condición para establecer el criterio de selección de T_0 :

Al inicio del proceso de inversión es deseable que cerca del 100 % de los parámetros del modelo sean aceptados, lo cual refleja una temperatura inicial suficientemente alta para permitir una exploración adecuada del espacio de soluciones. También es deseable que no pasen muchas iteraciones antes de que se comiencen a rechazar modelos de forma importante para ahorrar tiempo de cómputo.

El cumplimiento de esta primera condición se verificó al analizar simultáneamente todas las curvas de aceptación de modelos con respecto a las etapas de reducción de temperatura, agrupándolas en un arreglo y desplegándolas en forma de superficie, tal como se ilustra en la Figura 3.35:

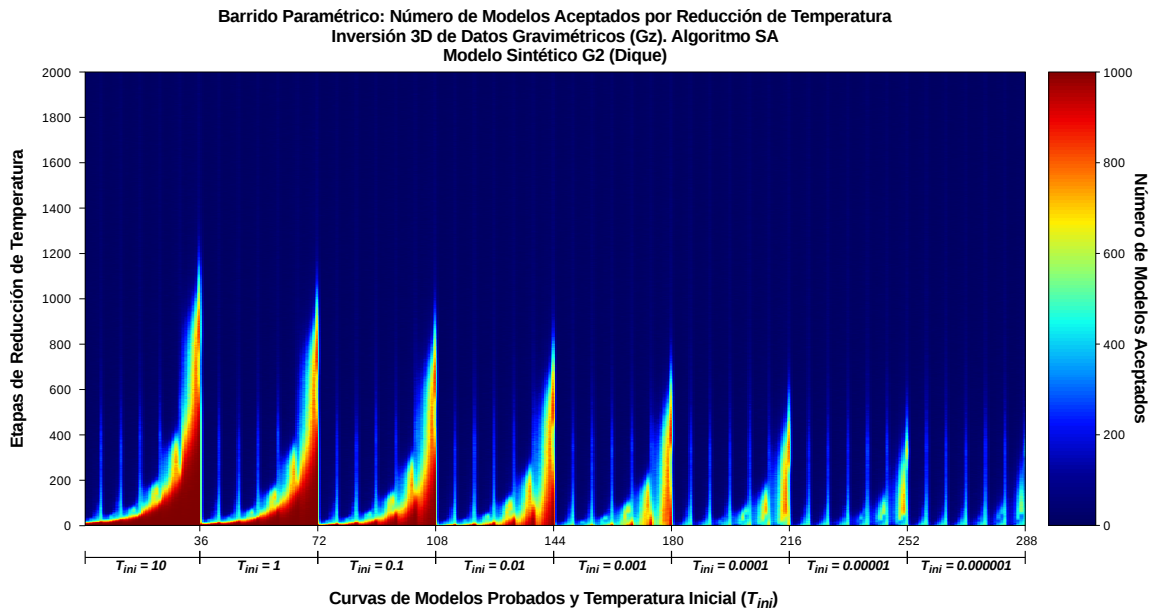


FIGURA 3.35: Superficie de aceptación de modelos por etapa de reducción de temperatura, para el método SA aplicado a la inversión de datos gravimétricos del modelo sintético G2.

Las curvas que forman la superficie de la Figura 3.35 están agrupadas en 8 bloques, correspondientes a los valores de prueba para T_0 , con 6 sub-bloques cada uno de ellos, respectivos a los valores de prueba de RT y a su vez, cada sub-bloque contiene 6 curvas calculadas para los diferentes valores de prueba de VM , tal como se ilustra en la Figura 3.36:

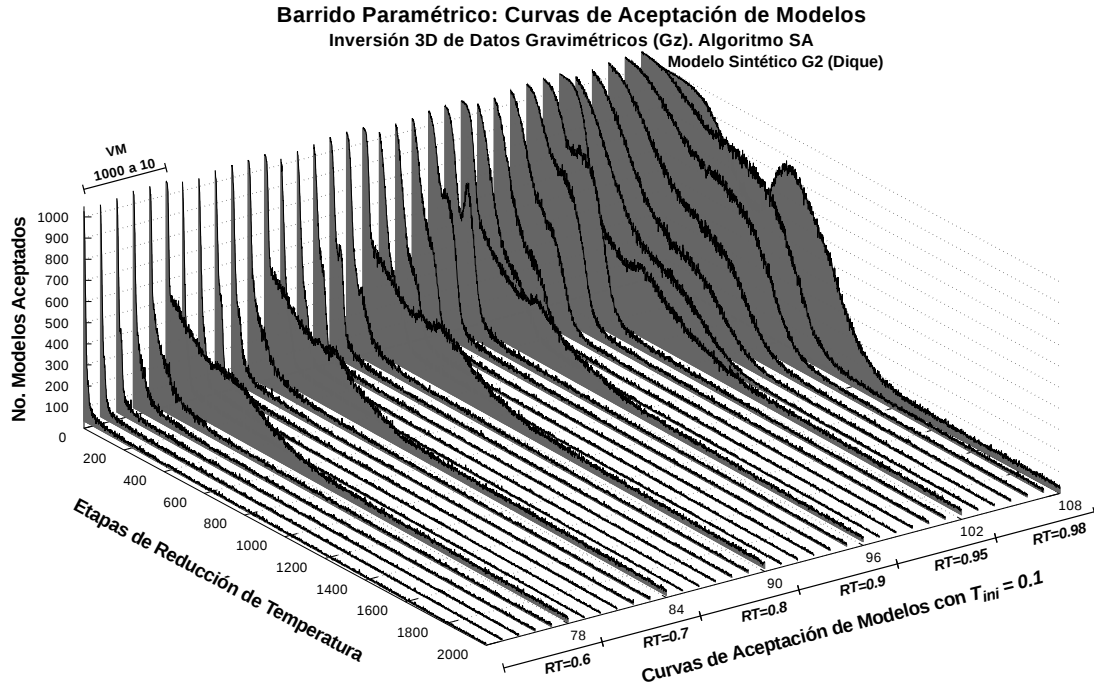


FIGURA 3.36: Representación en perspectiva de las curvas de aceptación de modelos por etapa de reducción de temperatura, para $T_0 = 0.1$.

Después de inspeccionar las superficies de aceptación de modelos por etapa de reducción de temperatura generadas para cada método de inversión, se determinó el siguiente criterio para seleccionar el valor de T_0 :

Criterio para la selección de T_0 : Más del 90 % de los modelos probados deben ser aceptados en las primeras 5 reducciones de temperatura para todos los valores del parámetro VM y sólo para $RT \geq 0.9$

Valores de T_0 seleccionados para la inversión de datos del modelo **G2**:

Inversión de G_z	Algoritmo SA:	$T_0 = 0.01$
	Algoritmo SA-SA:	$T_0 = 0.1$
Inversión de T_{ij}	Algoritmo SA:	$T_0 = 0.01$
	Algoritmo SA-SA:	$T_0 = 0.1$

Condición para establecer el criterio de selección de RT :

Es necesario que el descenso de temperatura sea lento, lo cual favorece la correcta exploración del espacio de modelos, al mismo tiempo que permite escapar de mínimos locales y se refleja en un descenso paulatino en la energía del sistema con las etapas de reducción de temperatura.

Para verificar el cumplimiento de esta segunda condición, se inspeccionaron las curvas de convergencia generadas para los T_0 que cumplen con su criterio de selección correspondiente, considerando todos los valores de prueba del VM . Esta inspección permitió descartar las curvas de convergencia generadas para $RT < 0.9$, debido al rápido descenso en temperatura causado. En la Figura 3.37 se muestra un ejemplo de curvas de convergencia analizadas, para $T_0 = 0.01$ y $RT \geq 0.9$, correspondientes con la inversión de datos gravimétricos con el algoritmo SA, del modelo sintético **G2**.

Una vez que se revisaron las curvas de convergencia, se definió el siguiente criterio para seleccionar el valor de RT :

Criterio para la selección de RT : El número de etapas de reducción de temperatura requeridas para la convergencia hacia la menor energía, considerando todos los valores propuestos del parámetro VM y el valor de T_0 elegido del criterio anterior, debe situarse entre el 25 % y el 35 % del máximo establecido ($nIter = 2000$).

RT seleccionado para la inversión de datos del modelo **G2**:

Inversión de G_z	Algoritmo SA:	$RT = 0.95$
	Algoritmo SA-SA:	$RT = 0.95$
Inversión de T_{ij}	Algoritmo SA:	$T_0 = 0.95$
	Algoritmo SA-SA:	$T_0 = 0.95$

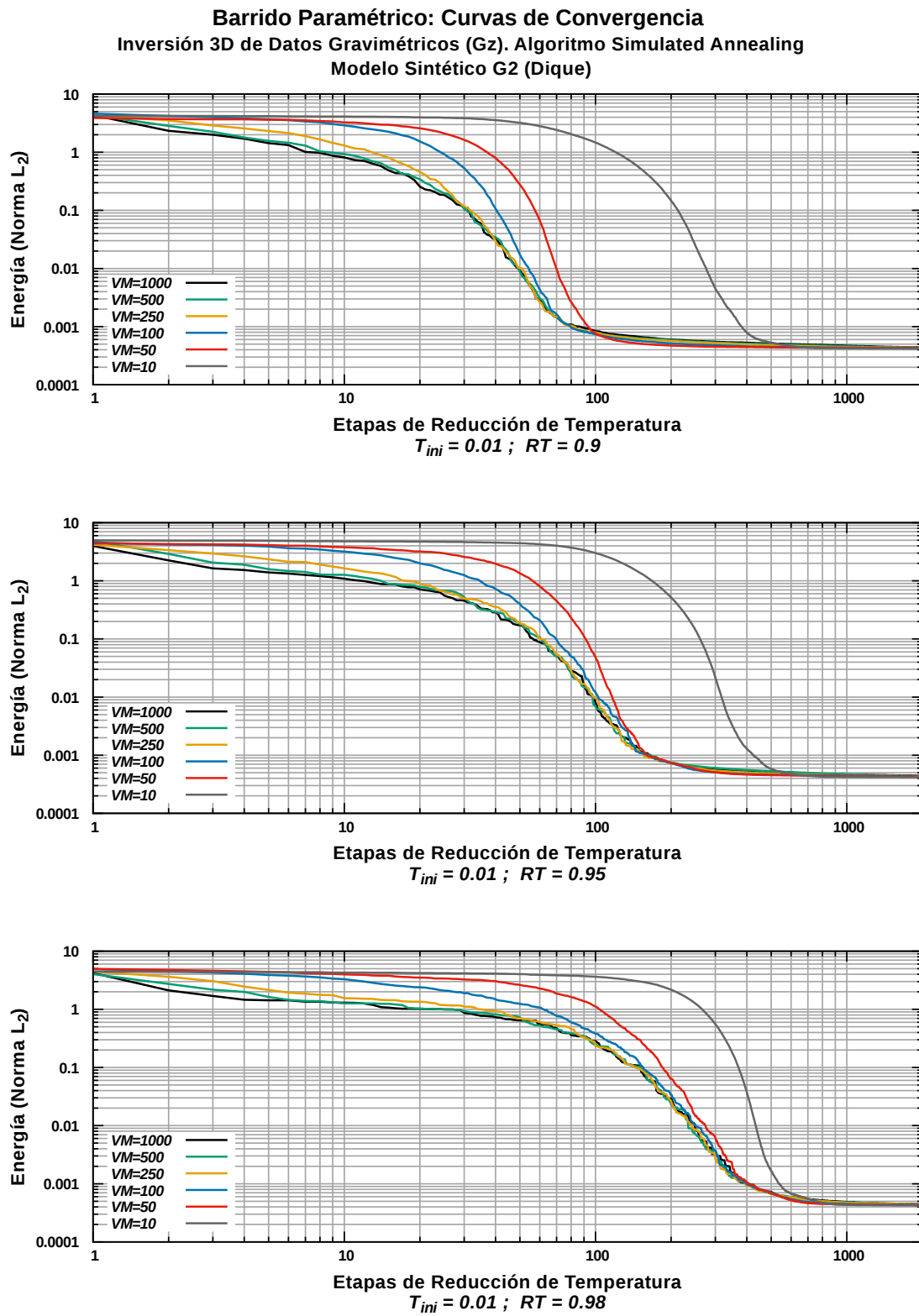


FIGURA 3.37: Curvas de convergencia del algoritmo SA aplicado a la inversión de datos gravimétricos del modelo sintético G2 con los parámetros $T_0 = 0.01$ y $RT \geq 0.9$.

Condición para establecer el criterio de selección de VM :

Se requiere un VM constante y relativamente pequeño para que los parámetros no salgan de los límites de interés una vez que han sido perturbados. También deben evitarse valores excesivamente pequeños, ya que esto puede provocar una convergencia demasiado lenta.

El VM es la amplitud de paso máxima que pueden alcanzar las perturbaciones por aplicarse a los parámetros en cada iteración y de acuerdo a las curvas de convergencia generadas por el barrido paramétrico (Ej. Figura 3.37), influye en la rapidez de convergencia del algoritmo.

Criterio para la selección de VM : Se elegirá el valor del VM que genere curvas de convergencia de aspecto suave y que contribuyan en alcanzar el mínimo de energía, inspeccionando las curvas de convergencia correspondientes a los parámetros T_0 y RT previamente elegidos.

VM seleccionado para la inversión de datos del modelo **G2**:

Inversión de G_z	Algoritmo SA:	$VM = 5\%^*$
	Algoritmo SA-SA:	$VM = 5\%^*$
Inversión de T_{ij}	Algoritmo SA:	$VM = 5\%^*$
	Algoritmo SA-SA:	$VM = 5\%^*$

* % de la amplitud máxima de la variación de parámetros

Finalmente, cabe mencionar que, debido a que la influencia que tienen los valores iniciales de estos parámetros en el desempeño de los algoritmos de inversión es dependiente de cada problema específico, los resultados del barrido y análisis paramétrico para la inversión de datos del modelo sintético **G2**, sólo deben tomarse como referencia. Se recomienda realizar análisis similares para cada caso particular.

4

APLICACIONES DEL MODELADO GRAVIMÉTRICO

4.1 Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México

Esta aplicación corresponde a la publicación directamente derivada de este trabajo de tesis (Nava-Flores et al., 2016), cuyo texto completo se puede consultar en el Anexo A, por lo cual, en este apartado únicamente se enfatizarán los aspectos más relevantes de la misma.

La aplicación en cuestión consistió en implementar el modelado gravimétrico 3D en un escenario geológico complejo caracterizado por la presencia de cuerpos salinos alóctonos emplazados en sedimentos terrígenos, a través de la integración de resultados de distintos métodos de delimitación de fuentes de anomalía en planta, estimación de profundidades e inversión de la anomalía gravimétrica generada por dichos cuerpos.

4.1.1 Anomalía gravimétrica de estructuras salinas

El modelado gravimétrico directo fue implementado para calcular la anomalía de cuerpos de sal interpretados a partir de imágenes sísmicas migradas en profundidad antes de apilar (PSDM por sus siglas en inglés), correspondientes a una zona de estudio ubicada al sur del Golfo de México (Figura 4.1).

Para realizar el cálculo, todo el dominio computacional se discretizó en un ensamble de elementos prismáticos rectangulares idénticos con densidad constante, cuya respuesta gravimétrica está dada por la ecuación 2.33 (Plouff, 1976; Blakely, 1996; Nagy et al., 2000). Posteriormente, se obtuvo la respuesta gravimétrica del ensamble, g_z , sumando las respuestas individuales de todos y cada uno de sus elementos, en los puntos de la malla de observaciones, ubicada sobre un plano, tal como se mencionó en el subcapítulo 2.1.3 (Figura 2.12; Ec. 2.42).

El ensamble consistió en 201, 540 prismas, cada uno con un tamaño de $50 [m] \times 50 [m] \times 25 [m]$ en direcciones X , Y y Z , respectivamente, mientras que la malla de observaciones se formó por 51×51 puntos equiespaciados en direcciones X y Y , con un intervalo de muestreo de $400 [m]$.

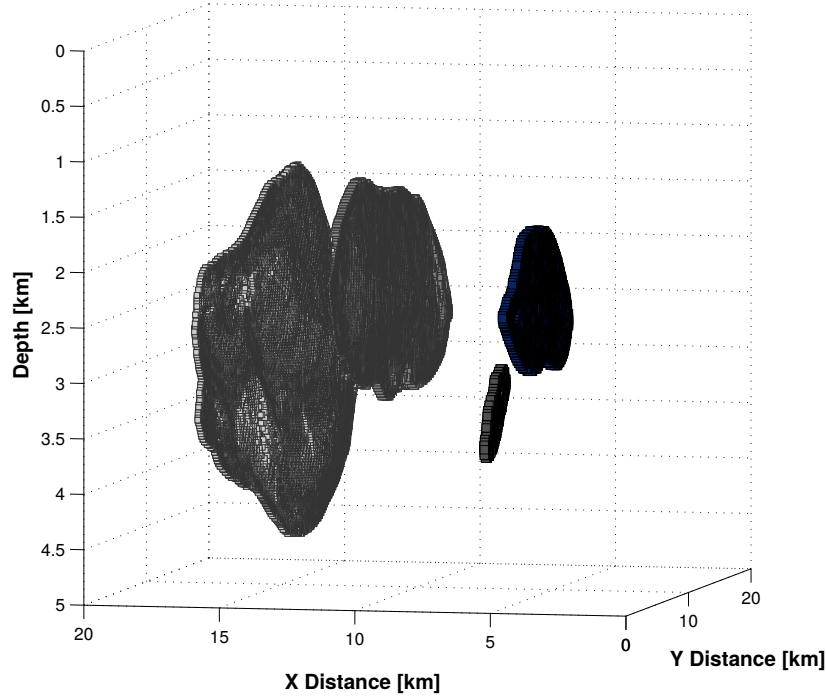


FIGURA 4.1: Cuerpos de sal interpretados a partir de un modelo de velocidades 3D de sísmica PSDM. Cada cuerpo se compone por múltiples prismas de idénticas dimensiones.

Los contrastes de densidad de cada prisma, requeridos para el cálculo de la respuesta gravimétrica de los cuerpos salinos, fueron calculados considerando un valor constante para los elementos constituidos por sal, $\rho = 2,180.0 \left[\frac{kg}{m^3} \right]$ (Gardner et al., 1974), mientras que la densidad de los sedimentos circundantes se estimó con base a curvas de compactación de sedimentos correspondientes al Golfo de México, publicadas por Hudcok y Jackson (2007), para el Laboratorio de Geodinámica Aplicada (AGL por sus siglas en inglés), basándose en el trabajo de Nelson y Fairchild (1989), quienes propusieron que la densidad de las unidades sedimentarias pueden modelarse como una función exponencial dependiente de la profundidad (Figura 4.2), a través de la curva:

$$\rho(z) = 1400.0 + 172.0z^{0.21} \quad (4.1)$$

Donde: ρ es la densidad de los sedimentos $\left[\frac{kg}{m^3} \right]$ y z es la profundidad $[m]$.

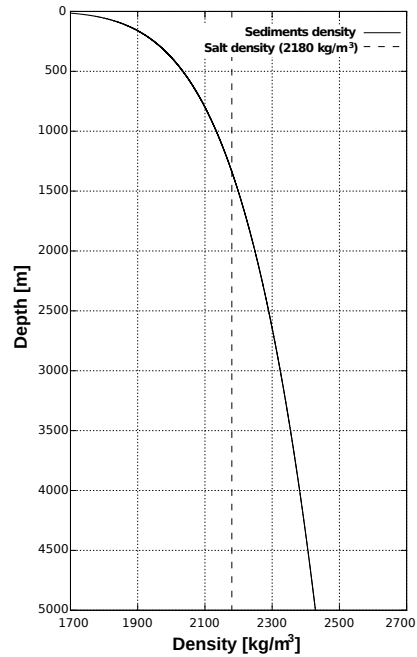


FIGURA 4.2: Curva de densidad contra profundidad representativa de sedimentos del Golfo de México y densidad de cuerpos salinos (basada en Nelson y Fairchild, 1989).

En la Figura 4.3 se muestra la malla de anomalía gravimétrica causada por los cuerpos de sal interpretados.

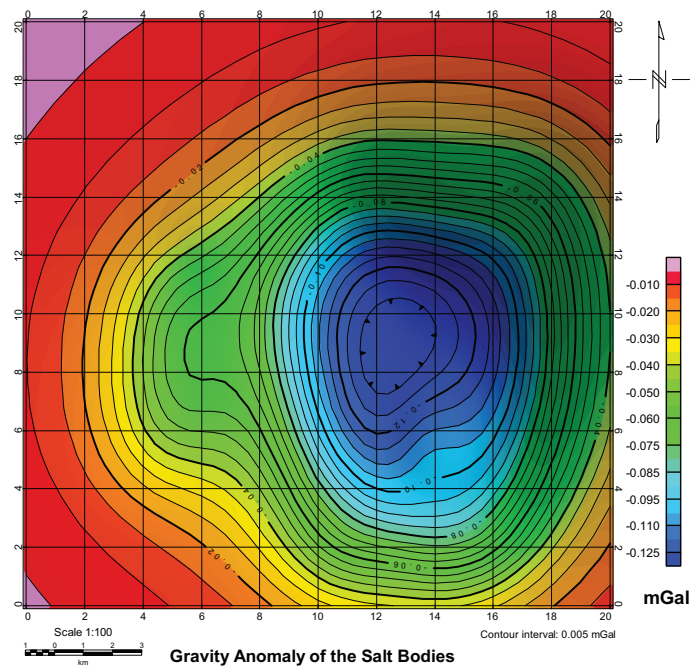


FIGURA 4.3: Anomalía gravimétrica generada por los cuerpos salinos mostrados en la Figura 4.1.

4.1.2 Estimación de la forma y profundidad de las estructuras salinas

Una vez calculada la malla de anomalía gravimétrica causada por los cuerpos salinos, se le aplicaron métodos de realce de anomalía y estimación de profundidades para definir un modelo de distribución de densidades 3D, que posteriormente se utilizó como modelo inicial en el algoritmo de inversión de datos.

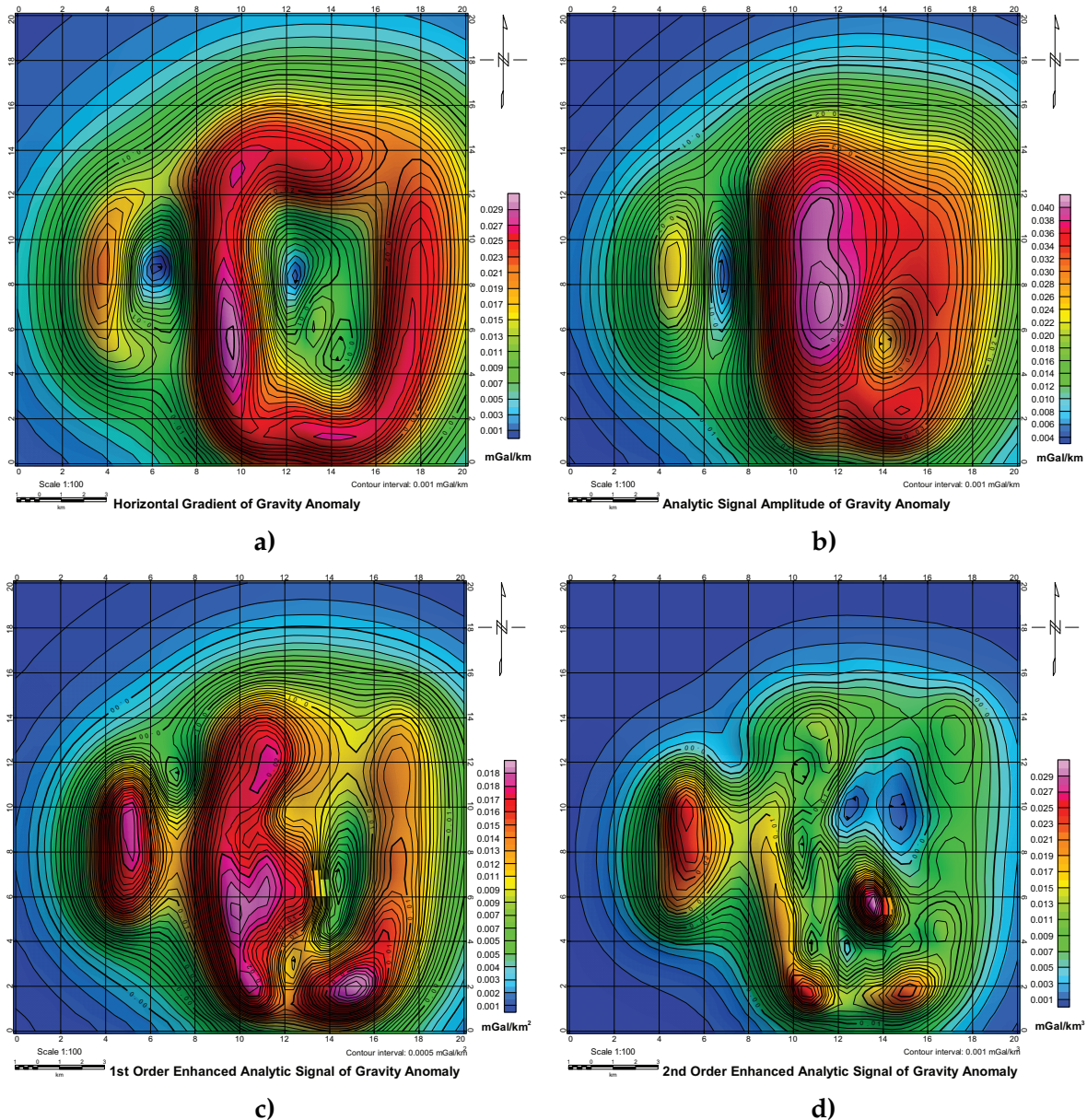


FIGURA 4.4: Mallas resultantes de métodos de realce de anomalías para definir la extensión lateral de las fuentes. **a)** Gradiente Horizontal, **b)** Amplitud de la Señal Analítica, **c)** Señal Analítica Mejorada de 1er Orden y **d)** Señal Analítica Mejorada de 2o Orden.

Los métodos de realce de anomalía aplicados fueron: gradiente horizontal, propuesto por Cordell (1979), amplitud de la señal analítica en 3D, desarrollada por Nabighian (1984) y señal analítica mejorada, introducida por Hsu et al. (1996). En la Figura 4.4 se muestran las mallas resultantes de la aplicación de estos métodos.

La ubicación de los máximos de las mallas generadas al aplicar los métodos de realce de anomalía son útiles para definir la extensión lateral de las fuentes. En este trabajo, la ubicación de los máximos fue determinada por el método de Blakely y Simpson (1986) y a partir de esa información, se interpretaron los cuerpos salinos vistos en planta, lo cual se muestra en la Figura 4.5.

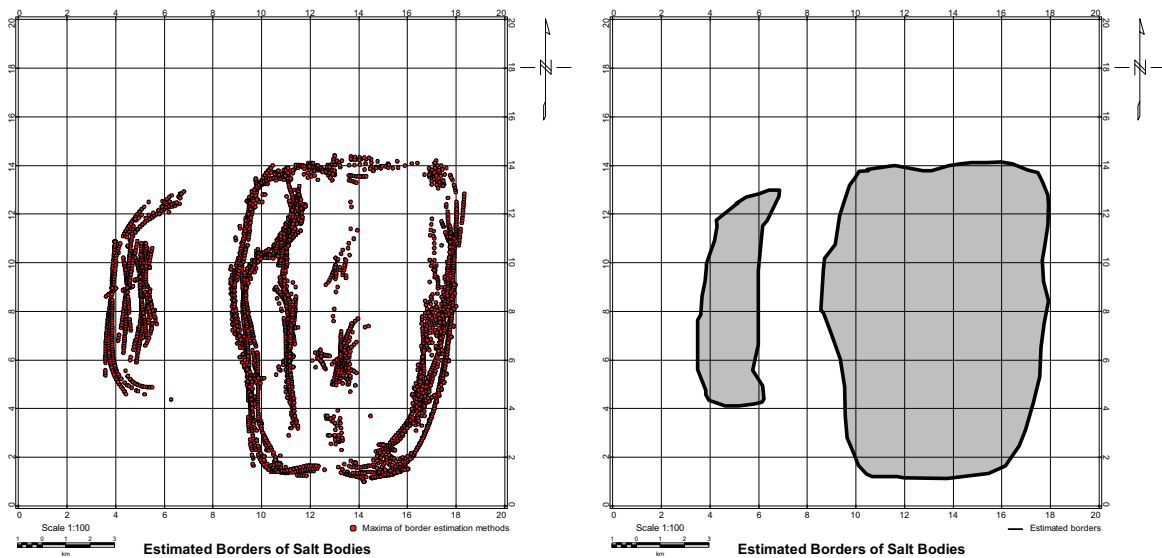


FIGURA 4.5: Izquierda: Ubicación de los máximos de los métodos mostrados en la Figura 4.4. Derecha: Interpretación de las fuentes de campo en planta, a partir de los máximos de dichos métodos.

Después de haber interpretado la extensión lateral de las fuentes de campo (Figura 4.5), se estimó su profundidad, aplicando el algoritmo de deconvolución de Euler 3D (Reid et al., 1990) a la malla de anomalía gravimétrica con un índice estructural $N = 0$, correspondiente con fuentes tipo contacto (Stavrev y Reid, 2006) y una ventana cuadrada de 4 [km] por lado. Las soluciones calculadas se muestran en la Figura 4.6.

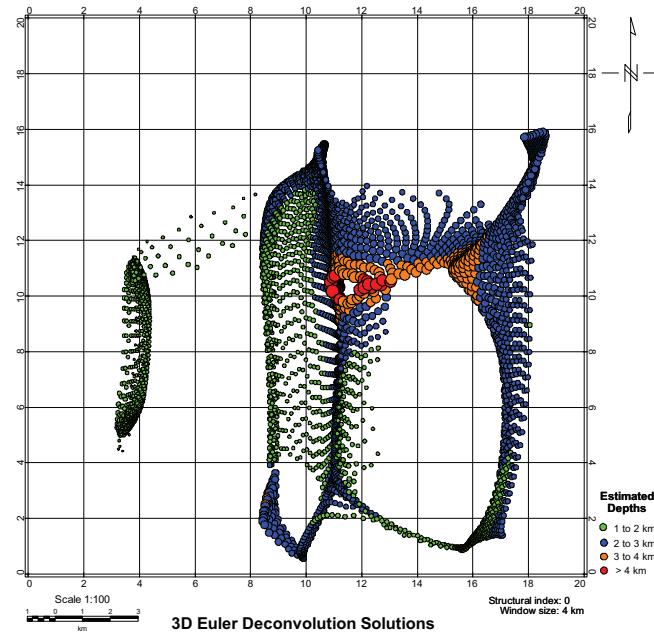


FIGURA 4.6: Soluciones del método DE3D aplicado a la malla de anomalía gravimétrica (Figura 4.3).

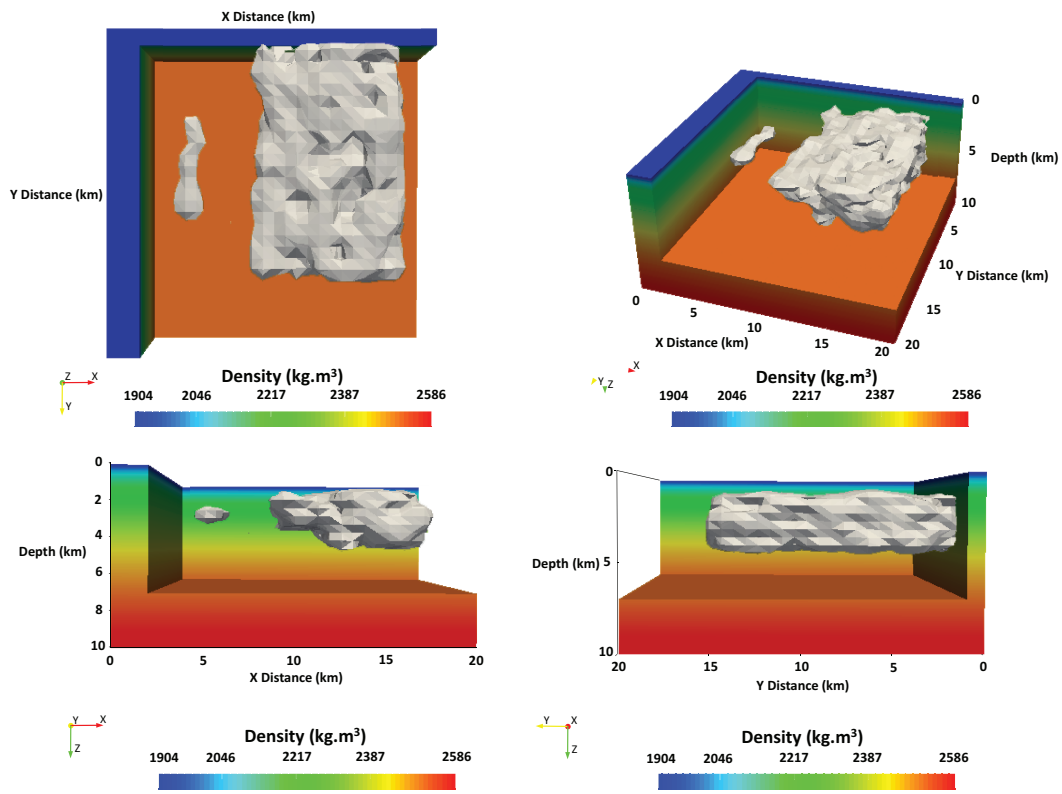


FIGURA 4.7: Modelo inicial de densidades 3D, construido a partir de información derivada de los métodos de realce de anomalía y estimación de profundidades aplicados a la malla de anomalía gravimétrica.

Con la información obtenida por los métodos de realce de anomalía y estimación de profundidades, se configuró el modelo de densidades 3D desplegado en la Figura 4.7 y se calculó su respuesta gravimétrica, mostrada en la Figura 4.8:

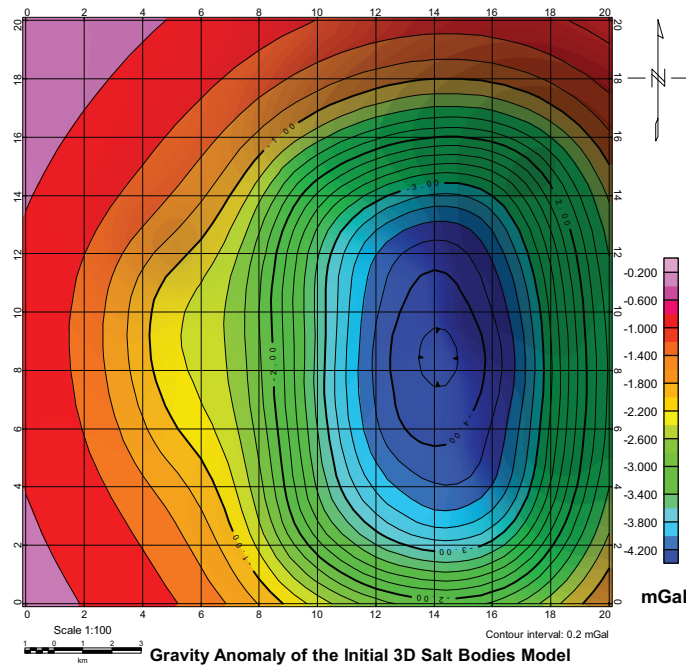


FIGURA 4.8: Anomalía gravimétrica generada por el modelo inicial mostrado en la Figura 4.7.

La anomalía gravimétrica calculada (Figura 4.8), debida al modelo inicial de densidades (Figura 4.7) se asemeja relativamente bien en forma a la anomalía gravimétrica de los cuerpos salinos interpretados (Figura 4.3), sin embargo, sus amplitudes son diferentes. Para reducir el error, se invirtió la anomalía gravimétrica a través de un método tipo Simulated Annealing (SA), considerando como modelo inicial el modelo de la Figura 4.7.

4.1.3 Inversión de datos gravimétricos

La inversión de datos gravimétricos en 3D se llevó a cabo con un algoritmo SA similar al descrito en el subcapítulo 2.3.2, numericamente optimizado e incluye un control de amplitud de paso autoajutable por parámetro entre cada etapa de reducción de temperatura (Corana et al., 1987), cuyo diagrama de flujo se ilustra en la Figura 2.14.

En la Tabla 4.1 se muestran las características principales del modelo inicial, que se incluyó en la inversión de datos gravimétricos:

TABLA 4.1: Características principales del modelo 3D inicial del algoritmo de inversión gravimétrica utilizado.

Dimensiones del modelo	Discretización del ensamble	Rango de densidades
Dirección X: 20,000 [m]	Dirección X: 30 prismas	1,900.0 a 2,590.0 $\left[\frac{kg}{m^3}\right]$
Dirección Y: 20,000 [m]	Dirección Y: 30 prismas	
Dirección Z: 10,000 [m]	Dirección Z: 30 prismas	
<p>Número total de prismas: 27,000 Tamaño de cada prisma: 666.66 [m] × 666.66 [m] × 333.33 [m] (Direcciones X, Y y Z)</p>		

Cabe mencionar que se impusieron las siguientes restricciones en la implementación del algoritmo de inversión:

1. La extensión lateral de los modelos generados durante la inversión se limitó al interior de los bordes y profundidades interpretados (Figuras 4.5 y 4.6).
2. El espacio de modelos se generó con los contrastes de densidad generados por la curva representativa de los sedimentos del Golfo de México (Figura 4.2).

En la Tabla 4.2 se muestran los parámetros asociados al rendimiento y ejecución del método de inversión implementado:

TABLA 4.2: Parámetros relacionados con el algoritmo de inversión 3D de datos gravimétricos aplicados a la anomalía de cuerpos salinos del Golfo de México.

Parámetros	Valores
Número de parámetros invertidos:	27,000
Número de puntos de observación:	2,601
Temperatura inicial:	1.0
Temperatura final:	0.16×10^{-8}
Energía del modelo inicial:	33.484543
Energía del modelo final:	0.0178259
Etapas de reducción de temperatura:	1,000
Factor de reducción de temperatura:	0.98
Número de ciclos previos al autoajuste de VM:	10
Número de ciclos de equilibrio térmico:	5
Número total de modelos probados:	149,400,000
Número de modelos aceptados:	115,281,390
Número de modelos rechazados:	34,118,610

La configuración 3D del modelo final generado por el proceso de inversión, se muestra en la Figura 4.9 y la superficie de anomalía gravimétrica generada por dicho modelo, se muestra en la Figura 4.10.

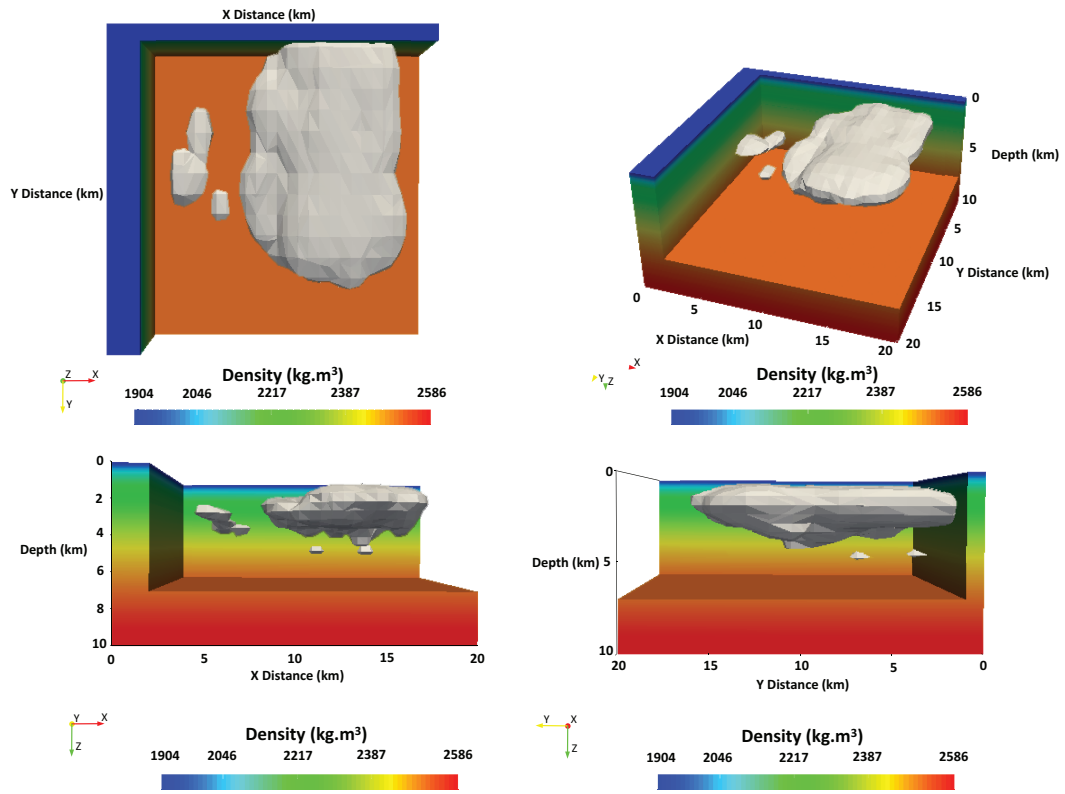


FIGURA 4.9: Modelo final de densidades 3D, construido a partir de la inversión de la anomalía gravimétrica de cuerpos salinos mostrada en la Figura 4.3.

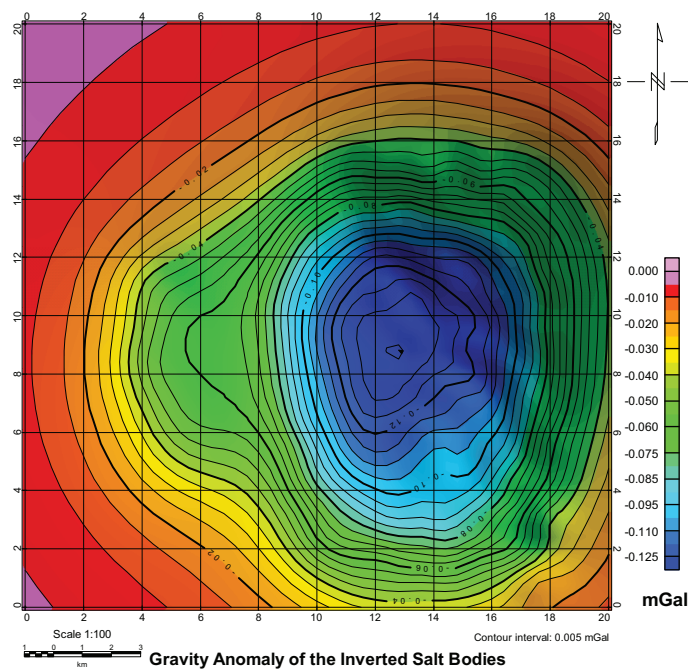


FIGURA 4.10: Anomalía gravimétrica generada por el modelo final de densidades mostrado en la Figura 4.9.

La anomalía gravimétrica generada por el modelo final, resultante de la inversión (Figura 4.10), muestra que, a pesar de existir algunas discrepancias en la forma de la parte central de esta, con respecto a la de la anomalía gravimétrica observada (Figura 4.3), sus amplitudes son similares.

Para cuantificar la calidad del modelo final invertido (Figura 4.9), se realizó un análisis de residuales (diferencias) entre datos de las mallas de anomalía observada (Figura 4.3) y calculada (Figura 4.10).

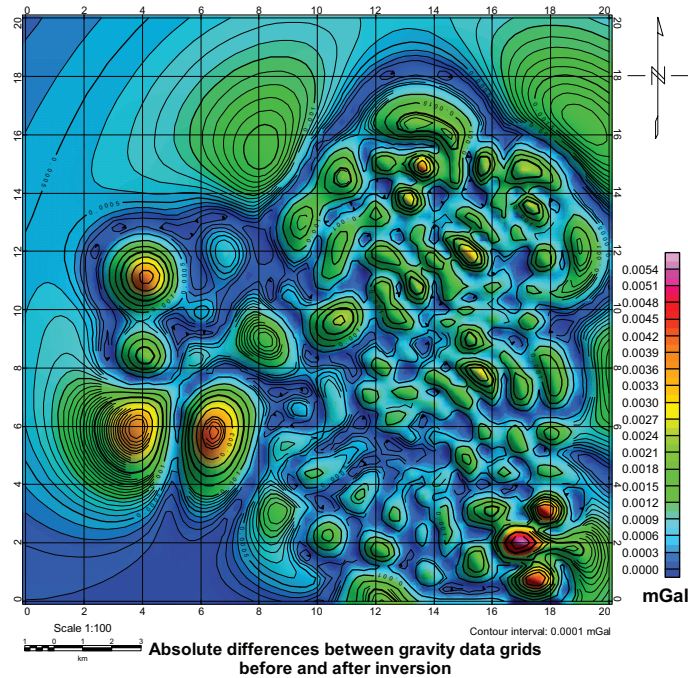


FIGURA 4.11: Distribución espacial de los valores absolutos de los residuales en la malla de observaciones.

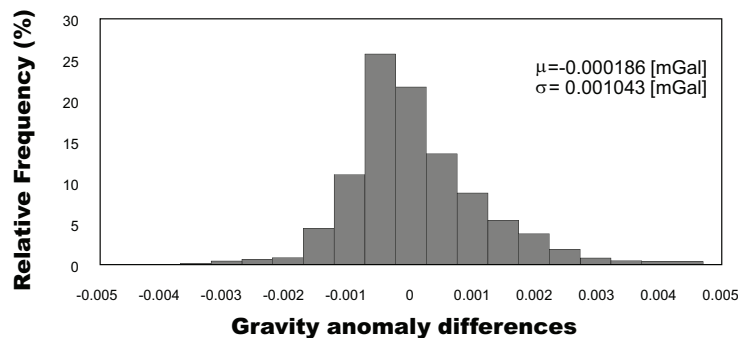


FIGURA 4.12: Distribución estadística de residuales entre las mallas de anomalía observada (Figura 4.3) y calculada (Figura 4.10).

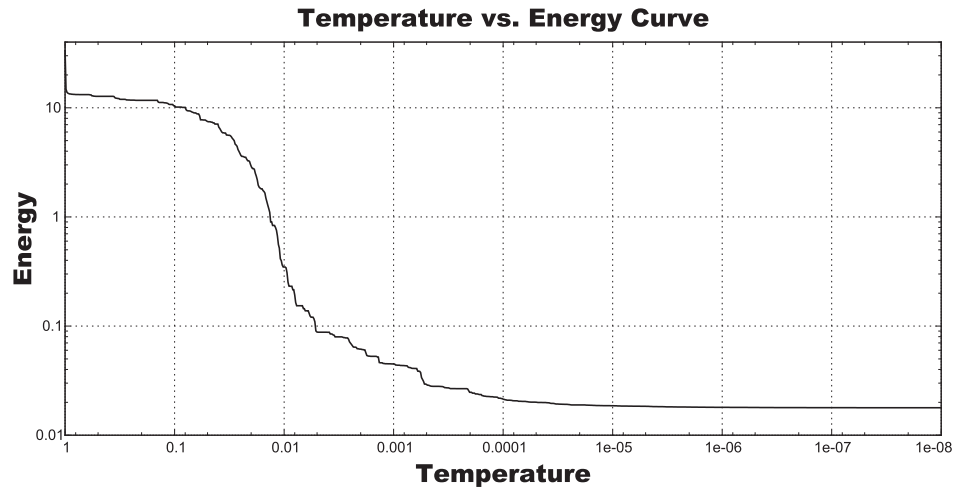


FIGURA 4.13: Curva de convergencia del proceso de inversión de datos gravimétricos.

La malla de valores absolutos de los residuales (Figura 4.11) ilustra su distribución espacial a lo largo de la malla de observaciones, así como sus amplitudes, cuyo máximo es $0.005624884 [mGal]$, equivalente al 4.26 % del rango total de la anomalía observada (Figura 4.3), que tiene un valor de $0.132137625 [mGal]$. El histograma de residuales generado, así como su media y desviación estándar se muestran en la Figura 4.12. Los valores de estos parámetros estadísticos son: $\mu = -0.000186 [mGal]$ y $\sigma = 0.001043 [mGal]$, indicando que la anomalía invertida reproduce exitosamente a la anomalía observada.

Finalmente, la curva de desajuste, que representa la relación entre temperatura y energía a lo largo del proceso de inversión (Figura 4.13), exhibe una forma caracterizada por tres tasas de convergencia, cuyo comportamiento se analizó en el subcapítulo 3.3.3 y es reflejo de la correcta selección de parámetros que controlan el proceso de inversión, así como del buen funcionamiento del mismo: un decaimiento gradual al inicio de la inversión, una zona de convergencia rápida y una zona de decremento progresivo y muy lento del error, hasta el final del proceso.

4.2 Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.

Esta aplicación consistió en modelar la capa de roca del domo salino Vinton, ubicado en Louisiana, E.U.A., a partir del procesamiento e inversión de mallas de gradiometría gravimétrica aérea, adquiridas por la empresa Bell Geospace Inc. en esa zona, a principios de julio de 2008. La metodología empleada incluyó la generación de un modelo inicial, formado a partir de la integración de resultados de procesos de delimitación de fuentes y estimación de profundidades, así como la depuración de dicho modelo y obtención de un modelo final, invirtiendo los datos de gradiometría gravimétrica observados, con un algoritmo tipo Simulated Annealing con Sensibilidad Aumentada (SA-SA).

4.2.1 Mallas de las componentes del tensor gravimétrico de la capa de roca del domo Vinton

Los datos utilizados para el modelado de la capa de roca del domo Vinton corresponden a la porción central de las mallas de gradiometría gravimétrica aérea, adquiridas por la empresa Bell Geospace Inc. en el período comprendido entre el 3 y 8 de julio de 2008, cuyo levantamiento fue realizado con las siguientes características:

Latitud:	30.07° a 30.23° N
Longitud:	-93.66° a -93.53° W
Altitud de vuelo:	53 a 116 [m]
Número de líneas de vuelo:	53
Número de líneas de amarre:	17
Total de kilómetros lineales:	1,087.5
Superficie cubierta:	196.2 [km ²]

En la Figura 4.14 se muestra la ubicación geográfica de la zona cubierta por el levantamiento gradiométrico.

Cabe mencionar que los datos proporcionados por la empresa ya incluyen un preproceso consistente en la edición de picos y datos ruidosos, micronivelación, corrección topográfica y filtrado de ruido de alta frecuencia (Selman, 2010).

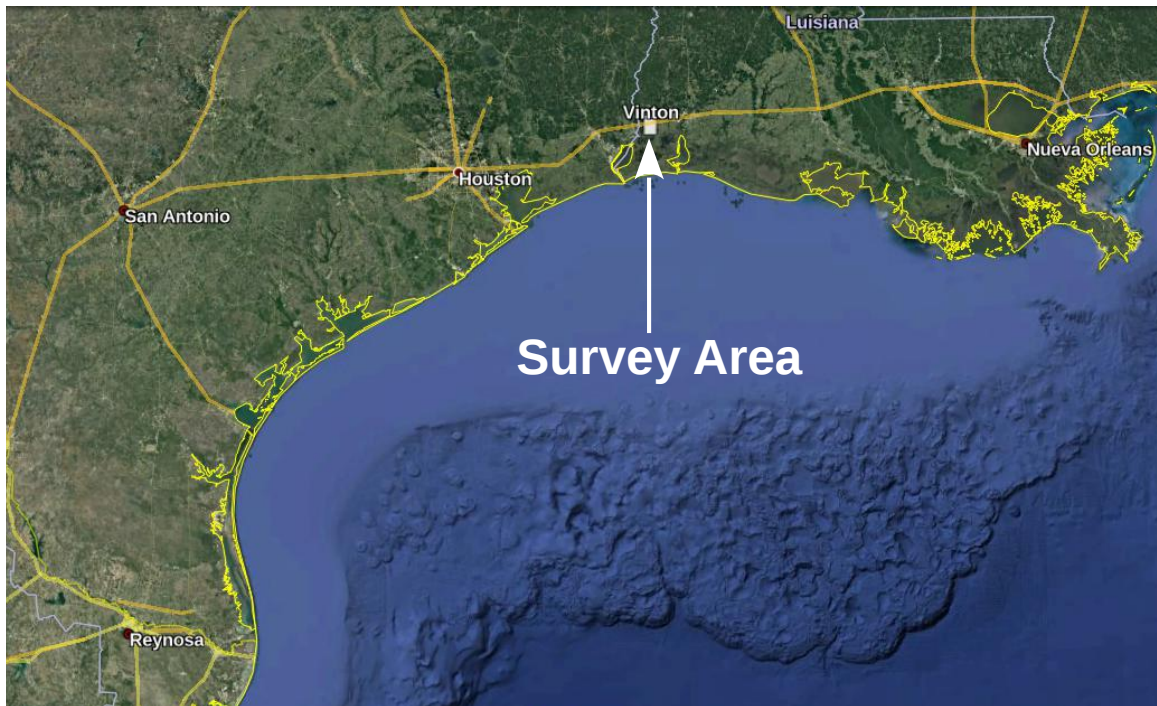


FIGURA 4.14: Ubicación geográfica del área del levantamiento gradiométrico realizado por la empresa Bell Geospace.

El área seleccionada para realizar el modelado cubre una superficie de $25.0 \text{ [km}^2\text{]}$ y está delimitada por las siguientes coordenadas:

	Coordenada Mínima	Coordenada Máxima
X [UTM]:	440,000	445,000
Y [UTM]:	3,332,000	3,337,000

En la Figura 4.15 se despliegan las mallas de las componentes tensoriales de la capa de roca del domo Vinton. Todas fueron extraídas a partir de las mallas de datos originales, previa aplicación de continuación ascendente a 150 [m] para remover efectos de fuentes superficiales.

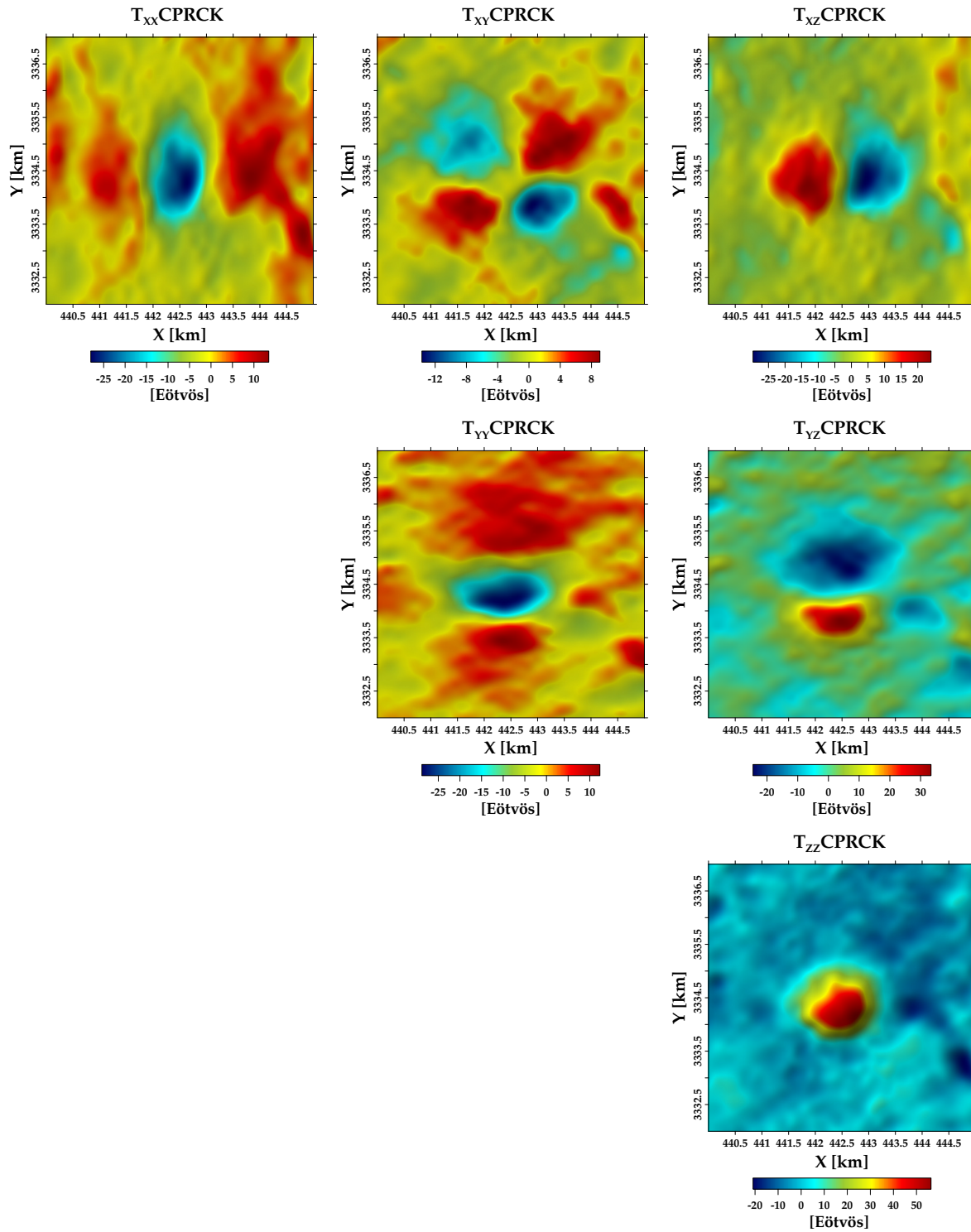


FIGURA 4.15: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional de la capa de roca del domo Vinton.

4.2.2 Interpretación de las mallas del tensor gravimétrico de la capa de roca del domo Vinton

La interpretación de la capa de roca del domo Vinton, realizada a partir de sus mallas de gradiometría gravimétrica, consistió en inferir su geometría y profundidad de sepultamiento utilizando métodos de realce de bordes y estimación de profundidades.

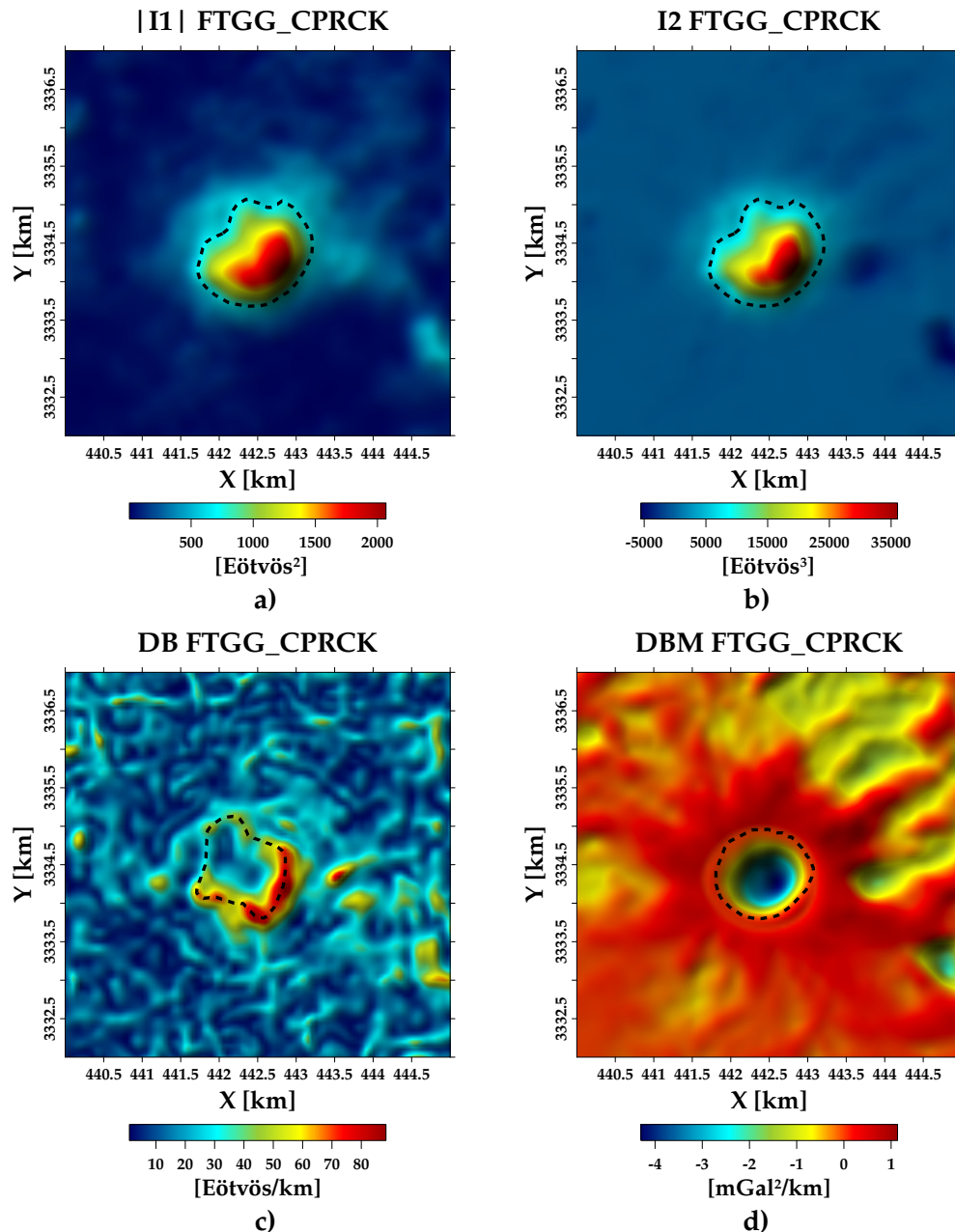


FIGURA 4.16: Mallas resultantes de métodos de realce de bordes para definir la extensión lateral de la capa de roca del domo Vinton. a) Valor absoluto del invariante I_1 , b) Invariante I_2 , c) Función detectora de bordes y d) Función detectora de bordes mejorada.

Para definir la extensión lateral de la capa de roca del domo Vinton se calcularon las mallas de invariantes I_1 e I_2 (Pedersen y Rasmussen, 1990), función detectora de bordes (Beiki, 2010) y función detectora de bordes mejorada (Zhou et al., 2013). En la Figura 4.16 se despliegan las mallas de estos procesos, así como una interpretación de la extensión lateral de la capa de roca para cada uno de ellos.

A partir de los límites laterales inferidos para la capa de roca de acuerdo a los métodos aplicados (líneas discontinuas superpuestas a las mallas de la Figura 4.16), se generó una interpretación unificada, la cual se muestra en la Figura 4.17 y representa la vista en planta de esta fuente de campo.

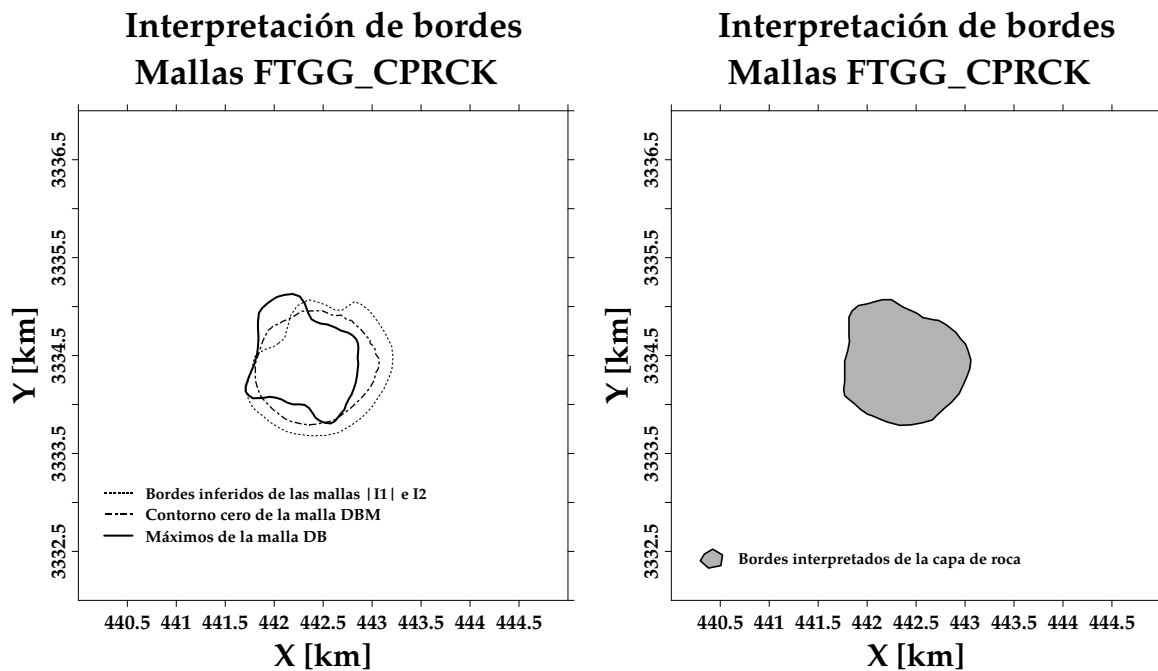


FIGURA 4.17: **Izquierda:** Límites laterales de la capa de roca del domo Vinton inferidos por los métodos de realce de bordes mostrados en la Figura 4.16. **Derecha:** Interpretación unificada de la extensión lateral de la capa de roca del domo Vinton, a partir de estos límites.

Una vez interpretada la extensión lateral de la capa de roca (Figura 4.17), se estimó su profundidad, a través de la deconvolución de Euler 3D del tensor gravimétrico (Zhang et al., 2000), utilizando una ventana cuadrada de 500 [m] por lado (equivalente a 10 puntos de malla) y un índice estructural $N = 1$, correspondiente a fuentes tipo contacto (Stavrev y Reid, 2006). Las soluciones generadas por este algoritmo se muestran en la Figura 4.18.

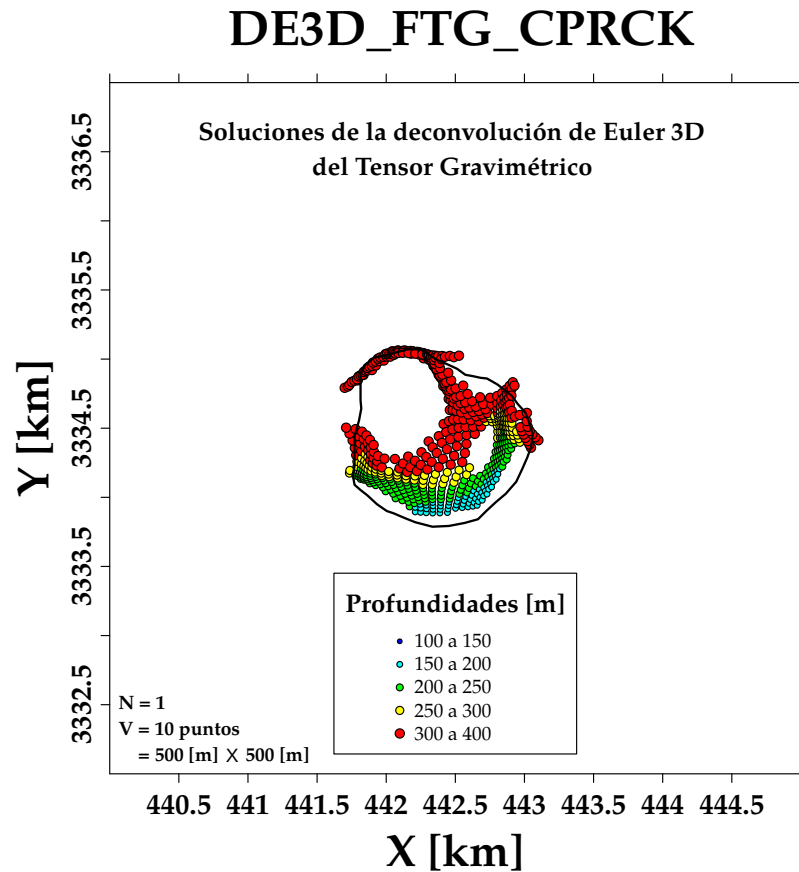


FIGURA 4.18: Soluciones de la deconvolución de Euler 3D del tensor gravimétrico de la capa de roca del domo Vinton (Figura 4.15). La línea negra representa la extensión lateral interpretada para la capa de roca.

Las profundidades estimadas con la deconvolución de Euler concuerdan con estudios previos realizados en la zona (Thompson y Eichelberger, 1928; Gherasim et al., 2005; Coker et al., 2007; Ennen y Hall, 2011), en los que se determinó que la capa de roca tiene una forma irregular aproximada a una pirámide de poca altura con base ovoidal, ápice achatado y desplazado hacia el sureste, cuyas profundidades varían desde aproximadamente 150 [m] al sureste, hasta 400 [m] al noroeste.

La superficie generada por el invariante I_2 , mostrada en el inciso b) de la Figura 4.16, que de acuerdo a Murphy y Brewster (2007) tiende a revelar la forma de la cima de las fuentes de campo, sirvió como base para interpretar la geometría de la cima de la capa de roca, realizando un escalamiento de las amplitudes de esta malla con las profundidades arrojadas por la deconvolución de Euler (Figura 4.18) y limitando su extensión lateral con la interpretación realizada a partir de los algoritmos de realce de bordes mostrados en la Figura 4.17. La geometría resultante de la cima se muestra en la Figura 4.19.

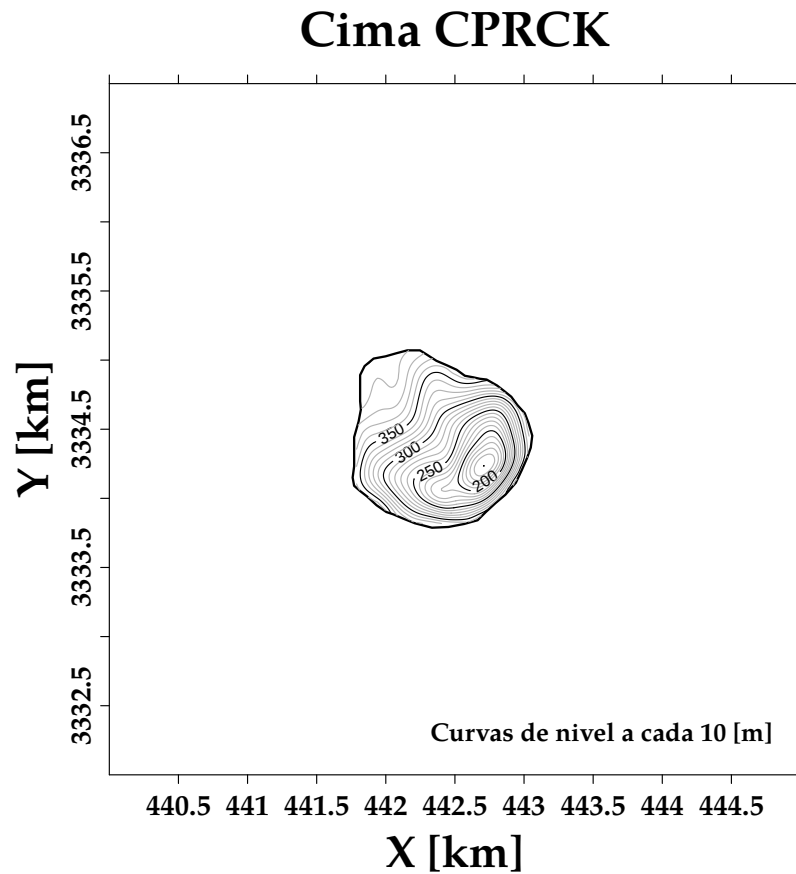


FIGURA 4.19: Geometría de la cima de la capa de roca del domo Vinton, estimada a partir de las profundidades del algoritmo de deconvolución de Euler 3D del tensor gravimétrico, la superficie del invariante I_2 y la extensión lateral interpretada y mostrada en la Figura 4.17.

A partir de la geometría de la cima de la capa de roca, se generó un modelo 3D con base horizontal y plana, alcanzando una profundidad de 600 [m] (Figura 4.20), que posteriormente se utilizó como modelo inicial en un proceso de inversión de las mallas del tensor gravimétrico mostradas en la Figura 4.15 con un algoritmo tipo SA-SA.

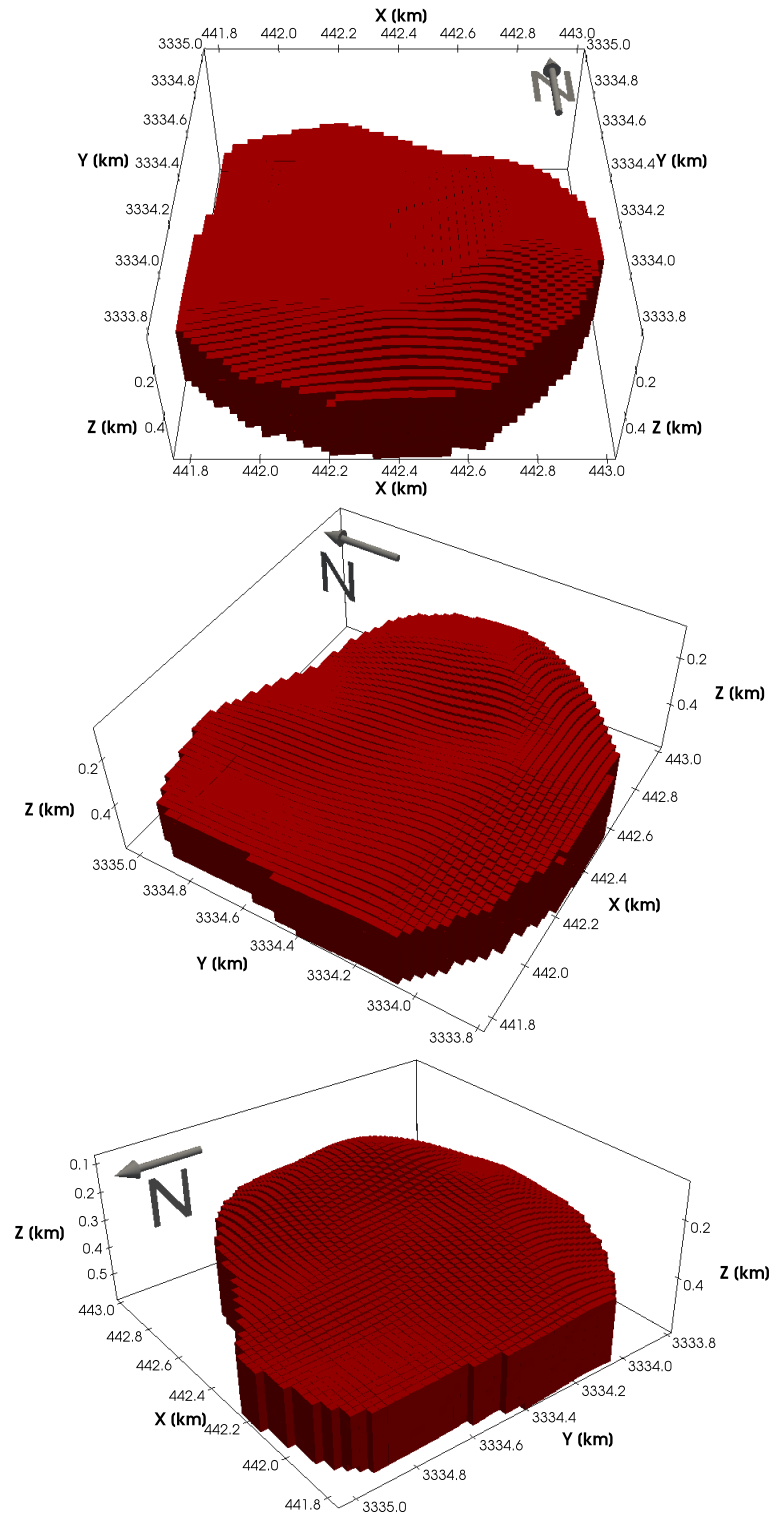


FIGURA 4.20: Visualización 3D del modelo de la capa de roca del domo Vinton, formado a partir de la geometría estimada de la cima (Figura 4.19) y las profundidades calculadas con la deconvolución de Euler (Figura 4.18).

El modelo mostrado en la Figura 4.20 se generó discretizando la capa de roca en un ensamble de prismas con las características que se indican en la Tabla 4.3, asignándole una densidad de $2,750 \left[\frac{kg}{m^3} \right]$, correspondiente a su composición reportada: caliza, yeso y anhidrita en sucesión, de acuerdo a Thompson y Eichelberger (1928) y Coker et al. (2007), mientras que la densidad de los sedimentos circundantes se asignó de acuerdo a la curva de compactación de sedimentos del Golfo de México (Hudec y Jackson, 2007; Nelson y Fairchild, 1989) dada por la ecuación 4.1, reescrita a continuación:

$$\rho(z) = 1400.0 + 172.0z^{0.21}$$

Donde: ρ es la densidad de los sedimentos $\left[\frac{kg}{m^3} \right]$ y z es la profundidad $[m]$.

TABLA 4.3: Características del modelo 3D inicial del algoritmo de inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton.

Dimensiones del modelo	Discretización del ensamble	Rango de densidades
Dirección X: 1,250 [m]	Dirección X: ≤ 50 prismas	$1,890$ a $2,750 \left[\frac{kg}{m^3} \right]$
Dirección Y: 1,250 [m]	Dirección Y: ≤ 50 prismas	
Dirección Z: 450 [m]	Dirección Z: ≤ 18 prismas	
Número total de prismas: 23,305 Tamaño de cada prisma: $25.0 [m] \times 25.0 [m] \times 25.0 [m]$ (Direcciones X, Y y Z)		

Las componentes tensoriales del modelo inicial de la capa de roca (Figura 4.21) fueron calculadas sumando las contribuciones individuales de cada prisma del ensamble, en los puntos de observación (Ec. 2.42), considerando que dichas contribuciones están dadas por las expresiones 2.36 a 2.41 (Nagy et al., 2000).

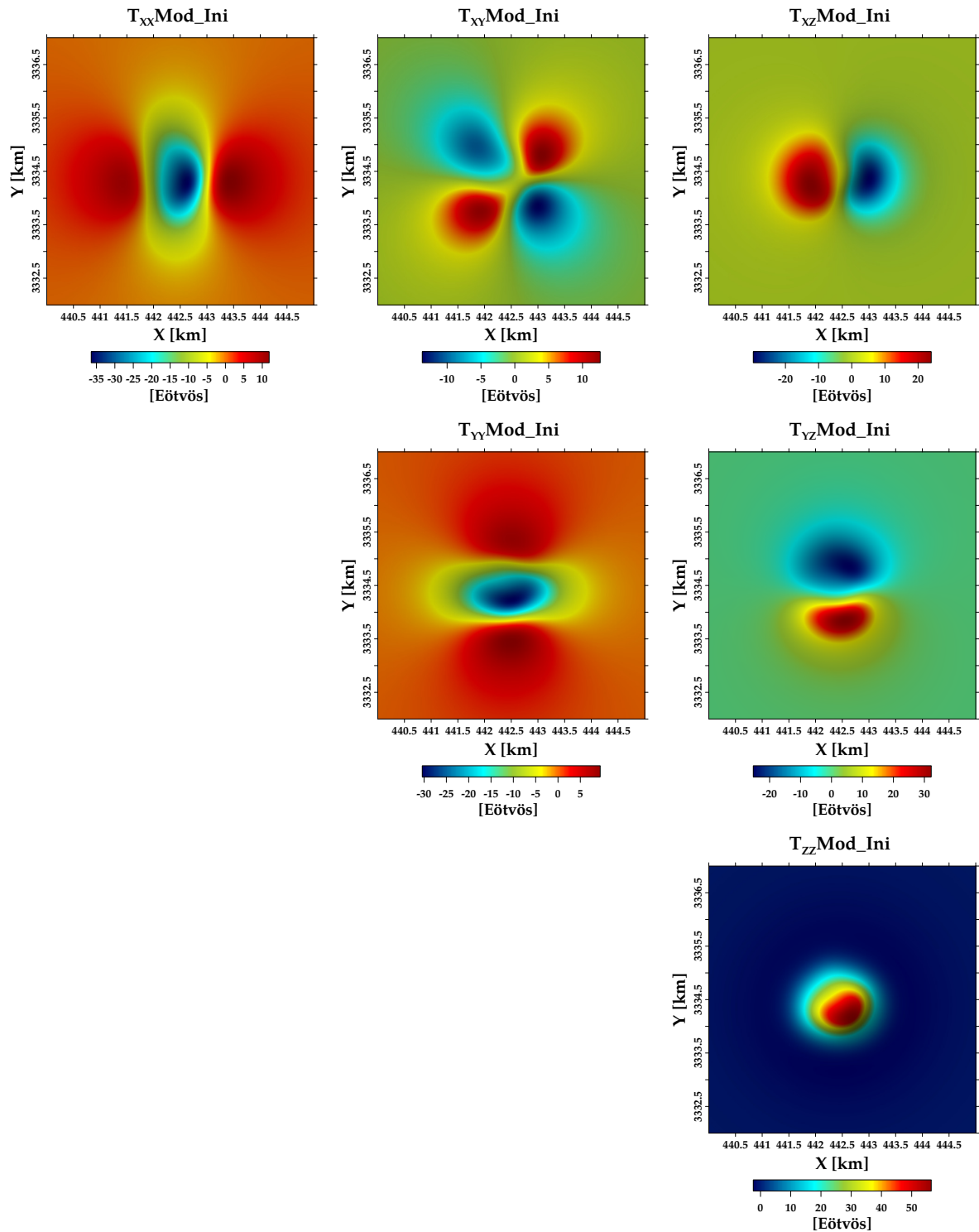


FIGURA 4.21: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional del modelo inicial de la capa de roca del domo Vinton (Figura 4.20).

4.2.3 Inversión de las mallas del tensor gravimétrico de la capa de roca del domo Vinton

Las componentes tensoriales del modelo inicial de la capa de roca (Figura 4.21), exhiben una forma correlacionable con las componentes observadas (Figura 4.15), aunque sus amplitudes difieren en magnitud desde 8.8 [Eötvös] en la componente T_{xy} , hasta 20.4 [Eötvös] en la componente T_{zz} . Con el objeto de obtener un modelo cuya respuesta gravimétrica exhiba diferencias mínimas entre componentes tensoriales observadas y calculadas, se invirtieron las mallas de datos observados con un algoritmo tipo SA-SA, utilizando como modelo inicial, el modelo interpretado que se muestra en la Figura 4.20.

Previo al proceso de inversión, se realizó un análisis similar al descrito en el subcapítulo 3.3.3 para definir los valores de los parámetros T_0 , RT y VM , obteniéndose los valores mostrados en la Tabla 4.4, donde se incluyen también los valores de otros parámetros que controlan el rendimiento y ejecución del algoritmo.

TABLA 4.4: Parámetros iniciales del algoritmo SA-SA utilizados en la inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton.

Parámetros Iniciales	Valores
T_0	0.001
RT	0.98
VM	5 %*
Descensos totales de temperatura	2,000
Ciclos de estabilidad termodinámica	5

* % del rango posible de densidades del modelo

Una vez concluido el proceso de inversión, se obtuvieron los valores de la Tabla 4.5, que resumen el desempeño del algoritmo de inversión utilizado.

TABLA 4.5: Resultados del algoritmo de inversión de mallas del tensor gravimétrico de la capa de roca del domo Vinton.

Parámetros	Valores
Número de parámetros invertidos:	23,305
Número de puntos de observación:	10,404
Temperatura inicial:	0.001
Temperatura final:	2.832×10^{-21}
Energía del modelo inicial:	0.436277
Energía del modelo final:	0.167857
Etapas de reducción de temperatura:	2,000
Factor de reducción de temperatura:	0.98
Número de ciclos de equilibrio térmico:	5
Número total de modelos evaluados:	233,050,000
Número de modelos aceptados:	58,695,910
Número de modelos rechazados:	174,354,090
Tiempo empleado en el cálculo:	7 hr, 48 min, 29 s

La configuración 3D del modelo final generado por el proceso de inversión se muestra en la Figura 4.22 y las superficies de componentes tensoriales resultantes del mismo, se despliegan en la Figura 4.23.

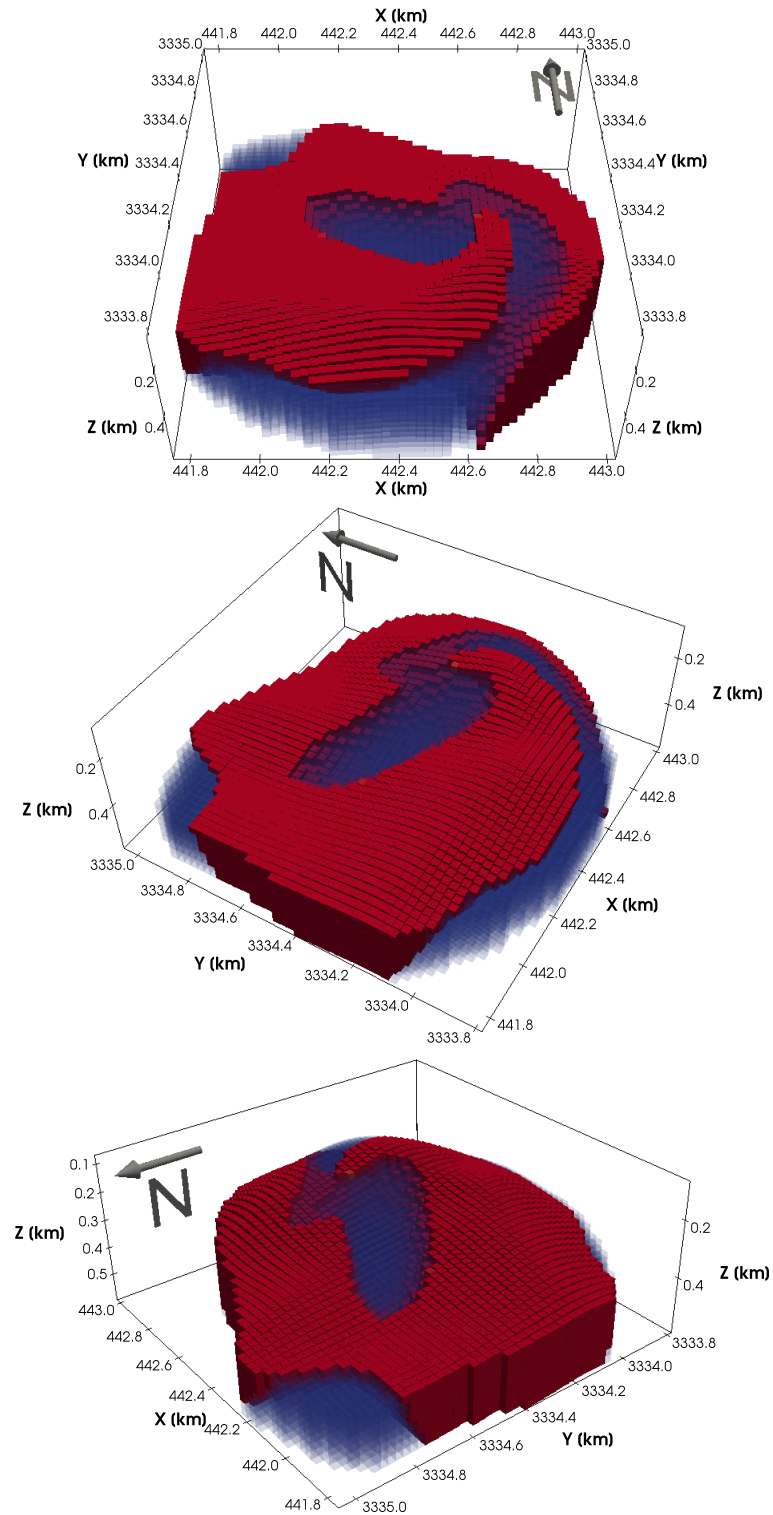


FIGURA 4.22: Visualización 3D del modelo final de la capa de roca del domo Vinton, generado a partir del método de inversión aplicado. Los prismas azules traslúcidos representan elementos de la capa de roca cuya densidad fue reasignada a densidad de sedimentos en el proceso de inversión.

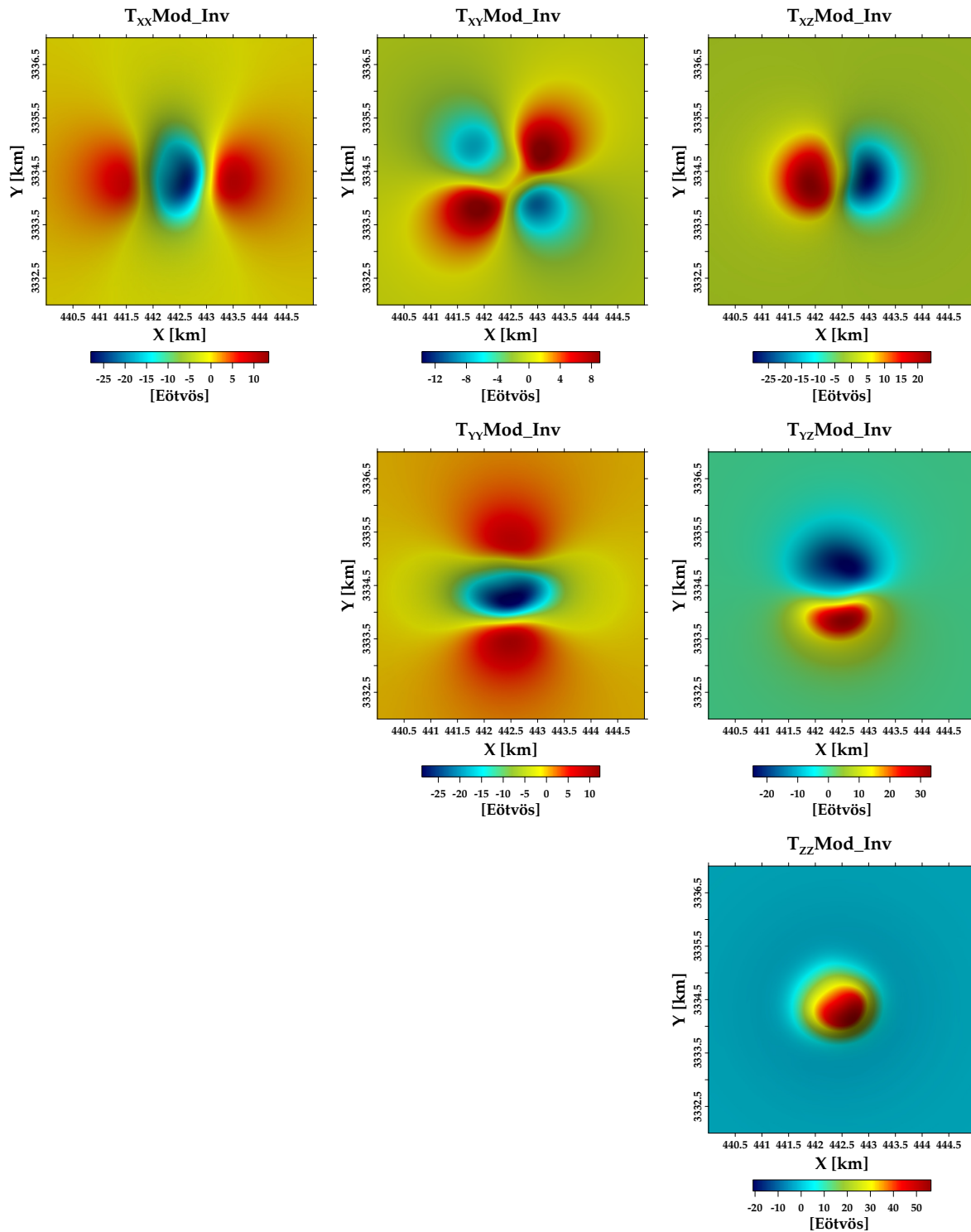


FIGURA 4.23: Superficies de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} del tensor de gradiometría gravitacional del modelo final de la capa de roca del domo Vinton (Figura 4.22), generado a partir del método de inversión aplicado.

Para evaluar la calidad del modelo final generado (Figura 4.22), se realizó un análisis estadístico y espacial de las diferencias entre datos observados y calculados (residuales), así como la generación e inspección de la curva de convergencia del método de inversión.

La distribución estadística de los residuales se determinó a partir de sus histogramas, obteniéndose uno por cada componente, acompañado por su media (μ) y desviación estándar (σ), lo cual se muestra en la Figura 4.24.

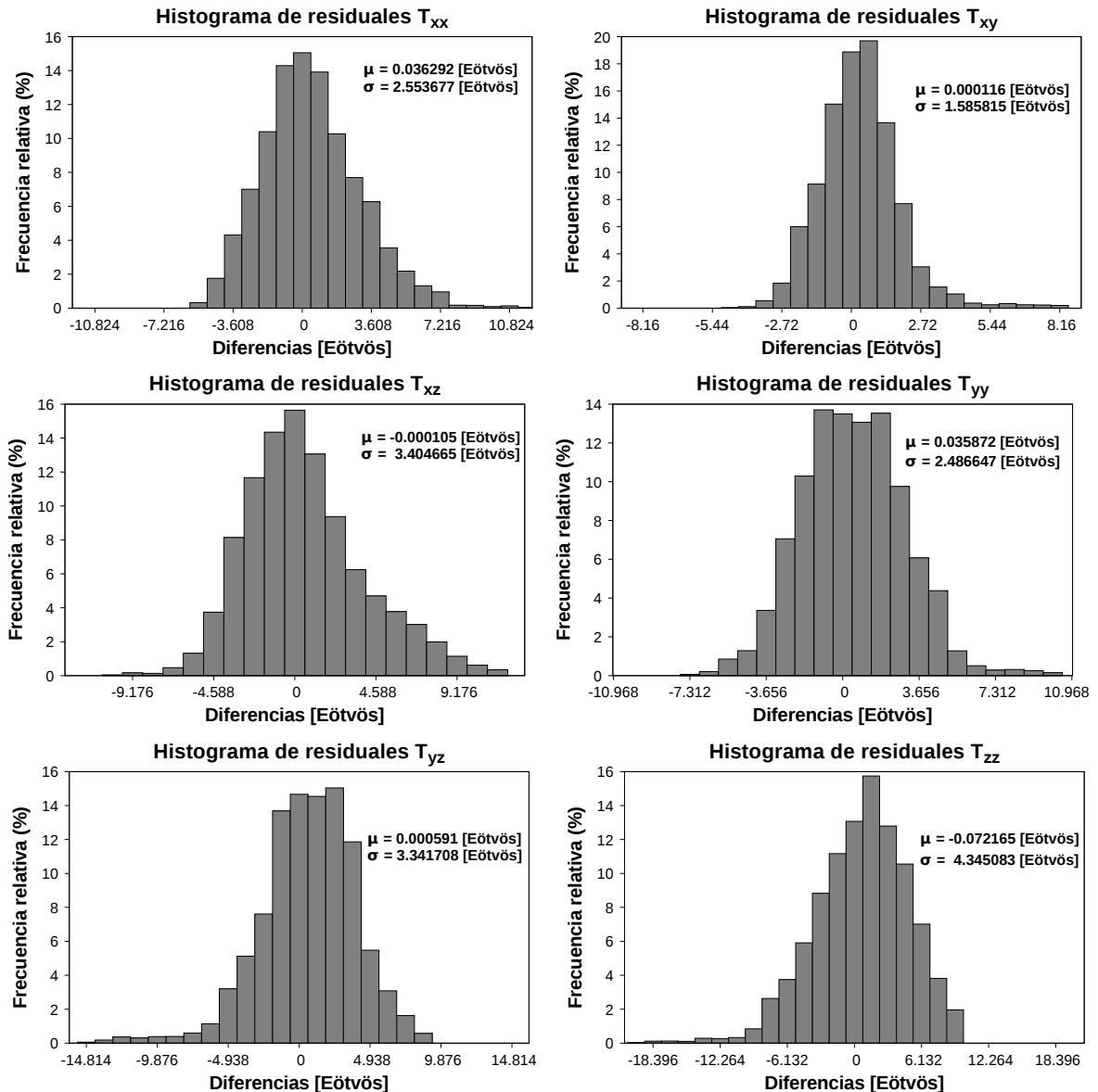


FIGURA 4.24: Distribución estadística de residuales entre las mallas observadas (Figura 4.15) y calculadas (Figura 4.23) del tensor de gradiometría gravimétrica de la capa de roca del domo Vinton.

Los valores de la media para todos los residuales son cercanos a cero, variando desde $-0.000105 [E]$ (Componente T_{xz}), hasta $-0.072 [E]$ (Componente T_{zz}), mientras que sus desviaciones estándar toman valores desde $1.58 [E]$ (Componente T_{xy}), hasta $4.34 [E]$ (Componente T_{zz}), lo que indica que las mallas de gradientes gravitacionales generadas por el modelo final se ajustan adecuadamente a las mallas de datos observados.

En cuanto a la distribución espacial de los residuales absolutos, se generaron las mallas mostradas en la Figura 4.26, cuya inspección corrobora que el modelo final reproduce adecuadamente los efectos gravitacionales de la capa de roca del domo Vinton, ya que los máximos registrados en estas mallas, dentro de los límites laterales de la capa de roca interpretada, alcanzan valores en un rango de 6.06 % (componente T_{xz}) a 15.53 % (componente T_{xy}) de la amplitud total observada de dichas componentes. Los valores numéricos de los máximos de las mallas de residuales absolutos, así como los porcentajes equivalentes con respecto a la amplitud máxima de cada componente se despliegan en la Tabla 4.6.

TABLA 4.6: Valores máximos y mínimos de las diferencias absolutas entre mallas de gradiometría gravimétrica observadas y calculadas para el modelo final de la capa de roca del domo Vinton.

Residuales absolutos máximos	% Amp. Máx.
$T_{xx} : 3.302905 [E]$	7.96 %
$T_{xy} : 3.535734 [E]$	15.53 %
$T_{xz} : 3.250536 [E]$	6.06 %
$T_{yy} : 5.118259 [E]$	12.44 %
$T_{yz} : 5.931692 [E]$	10.23 %
$T_{zz} : 7.298645 [E]$	9.46 %

De forma complementaria al análisis de residuales y como parte de la valoración del rendimiento del algoritmo de inversión empleado, en la Figura 4.25 se muestra la curva de convergencia, que refleja ser resultado de un proceso gradual muy lento de modificación del modelo inicial hasta su forma final.

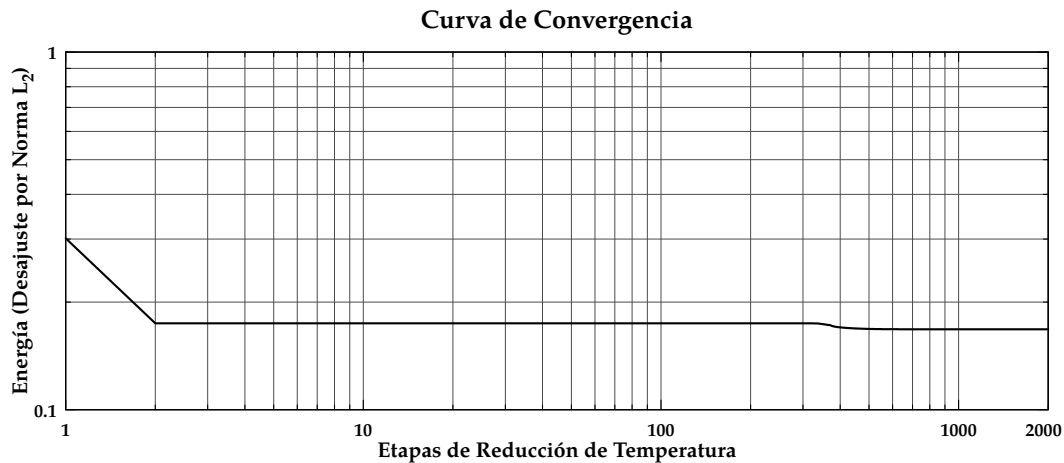


FIGURA 4.25: Curva de convergencia del proceso de inversión de mallas de gradiometría gravimétrica de la capa de roca del domo Vinton.

La relativamente baja tasa de reducción del error en el proceso de inversión que se refleja en la curva de convergencia, es un indicador de que el modelo inicial introducido al algoritmo de inversión, que a su vez fue generado a partir del procesamiento de las mallas de tensor gravimétrico e información previa del área de estudio, tiene la calidad suficiente para generar una respuesta gravitacional con poco error.

Finalmente, en lo que respecta a la morfología del modelo final generado (Figura 4.22), se observa que tiene una forma y profundidad acorde con modelos generados en trabajos recientes como los de Wang et al. (2017), Qin et al. (2016), Geng et al. (2014) y Oliveira y Barbosa (2013), aunque a diferencia de todos ellos, la resolución del ensamble generado para reproducir la forma de la capa de roca, es mayor en este trabajo.

Es notable la presencia de un rasgo en la superficie de la capa de roca similar a una hendidura rellena por material con la misma densidad de los sedimentos encajonantes, que sigue la dirección preferencial NW-SE de los fallamientos principales reportados en esta zona (Coker et al., 2007). Aunque no se puede asegurar que esta hendidura tenga las dimensiones exactas con que se muestran en el modelo generado a través del proceso de inversión, sí es una estructura congruente con la geología regional de la zona (Thompson y Eichelberger, 1928; Ennen y Hall, 2011; Coker et al., 2007).

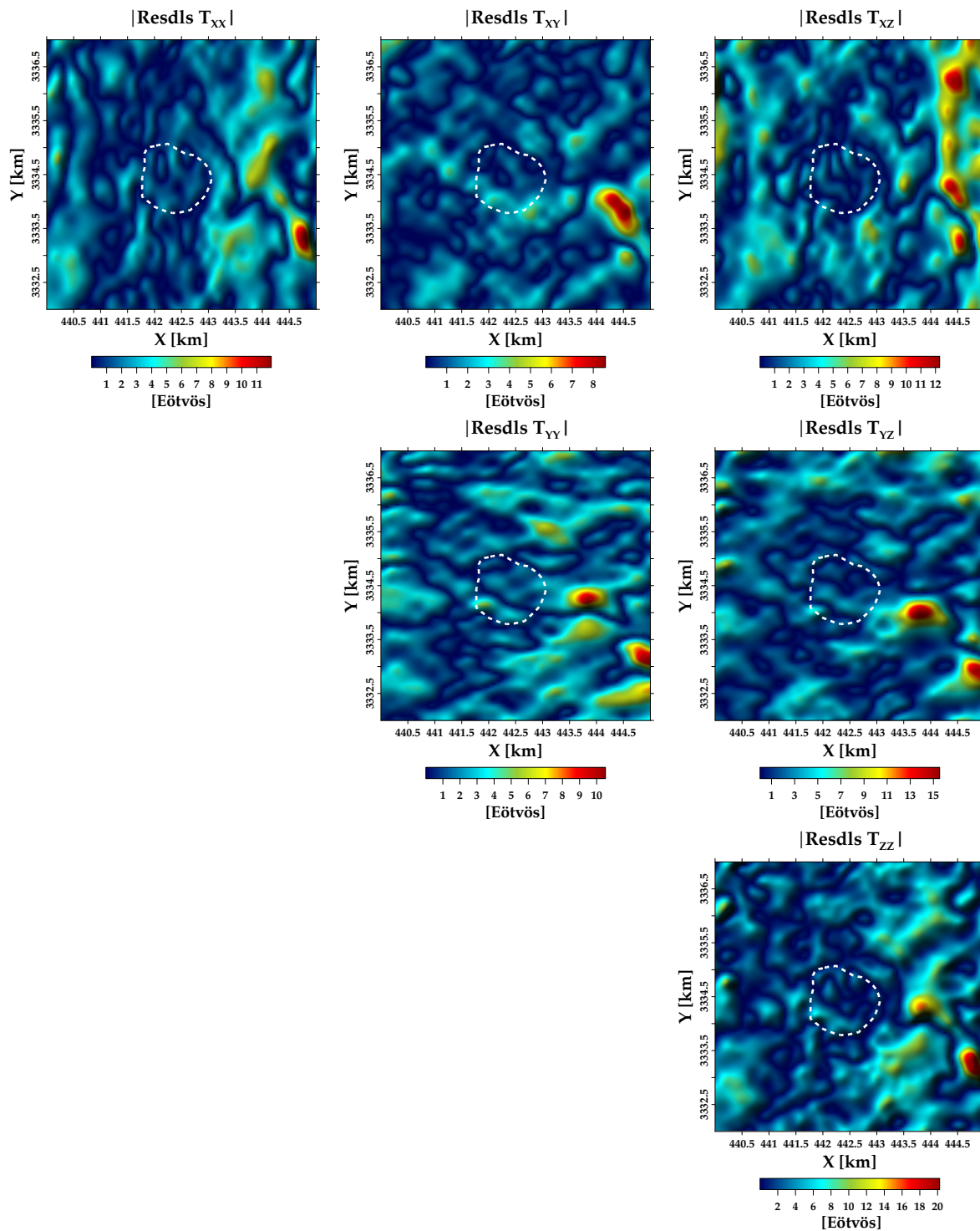


FIGURA 4.26: Mallas de residuales absolutos de las componentes del tensor de gradiometría gravimétrica del modelo final de la capa de roca del domo Vinton. Los límites laterales de la capa de roca interpretada se indican en línea discontinua.

5

CONCLUSIONES Y RECOMENDACIONES

Se presentan a continuación, las principales conclusiones derivadas de este trabajo de tesis, así como las recomendaciones para trabajos futuros relacionados con el modelado gravimétrico 3D, de datos de anomalía o gradiometría gravimétrica en zonas de tectónica salina.

5.1 Conclusiones

La metodología propuesta para realizar el modelado gravimétrico 3D ha probado ser exitosa en la exploración del subsuelo en escenarios geológicos complejos caracterizados por la presencia de cuerpos salinos alóctonos emplazados en sedimentos terrígenos asociados a la presencia de hidrocarburos.

La integración de resultados de los métodos de delimitación de bordes y estimación de profundidades aplicados a las mallas de anomalía y gradiometría gravimétrica es esencial para realizar la interpretación y configuración del modelo inicial de densidades del subsuelo de las fuentes de anomalías y su posterior inversión, aportando información útil para restringir el espacio de búsqueda del método de inversión empleado.

El método de inversión Simulated Annealing con Sensitividad Aumentada propuesto en este trabajo, que incluye el cálculo optimizado del producto matriz-vector en cada iteración presentado por Ortiz-Alemán y Martín (2005a) y un factor de incremento de sensibilidad que ayuda a contrarrestar el efecto de la obtención de soluciones cercanas a los puntos de observación, arrojó resultados aceptables, tanto en las pruebas con datos sintéticos, como en las aplicaciones, probando que es relativamente insensible ante el ruido y computacionalmente eficiente, invirtiendo modelos formados por decenas de miles de parámetros en tiempos relativamente cortos, logrando realizar más de 200 millones de cálculos del problema directo en 7 hr, 48 min, 29 s.

El análisis del algoritmo de inversión y los criterios propuestos para seleccionar los valores iniciales de los parámetros que controlan su rendimiento y ejecución, probaron ser herramientas necesarias para determinar dichos valores y aumentar la probabilidad de que al fin de su ejecución, se generaran modelos potencialmente útiles para la interpretación.

La aplicación publicada, **Modelado gravimétrico 3D de estructuras salinas complejas en el sur del Golfo de México** (Nava-Flores et al., 2016), mostró que la metodología propuesta para el modelado gravimétrico 3D es adecuada en escenarios reales comúnmente encontrados en el Golfo de México, mientras que la aplicación denominada **Modelado 3D de datos de gradiometría gravimétrica aplicado al domo Vinton, LA, E.U.A.** muestra que esta metodología es válida si se extiende a mallas del tensor gravimétrico, logrando obtener un modelo 3D para la capa de roca del domo con una geometría general y profundidad estimada concordante con otros modelos previamente generados por estudios en esa misma zona (Wang et al., 2017; Qin et al., 2016; Geng et al., 2014; Oliveira y Barbosa, 2013), pero con mayor resolución que estos, además de revelar un rasgo estructural superficial, que si bien no se ha corroborado, parece ser coherente con la geología de la región.

5.2 Recomendaciones para trabajos futuros

Aunque los métodos de cálculo del problema directo e inverso mostrados han probado lograr un desempeño aceptable, se recomienda realizar análisis dirigidos a incrementar su desempeño con técnicas de supercómputo al aplicarse a modelos realistas de grandes dimensiones, como se muestra en los trabajos de Couder-Castañeda et al. (2013), Couder-Castañeda et al. (2015), Martin et al. (2013) y Hou et al. (2015).

Aún se debe analizar a profundidad el método de inversión propuesto (SA-SA) para mejorar su funcionamiento y adaptación a diferentes problemas. Se recomienda inspeccionar el efecto en el algoritmo que pudieran tener diferentes esquemas de enfriamiento, diferentes factores amplificadores de sensibilidad y diferentes funciones de costo.

Independientemente del algoritmo de inversión empleado, si está basado en algún método de optimización global, es recomendable realizar análisis paramétricos para determinar los valores iniciales óptimos que puedan aplicarse a cada problema específico para ahorrar tiempo de cómputo y aumentar la probabilidad de generación de modelos útiles en la solución de dichos problemas, tal como se propuso en este trabajo.

Referencias

- Barnett, C. T., 1976. Theoretical modeling of the magnetic and gravitational fields of an arbitrarily shaped three-dimensional body. *Geophysics*, 41, 1353–1364.
- Beiki, M., 2010. Analytic signals of gravity gradient tensor and their application to estimate source location. *Geophysics*, 75, I59–I74.
- Bhattacharyya, B. K., 1978. Computer modeling in gravity and magnetic interpretation. *Geophysics*, 43, 912–929.
- Blakely, R. J., 1996. *Potential theory in gravity and magnetic applications*. Cambridge University Press.
- Blakely, R. J. and Simpson, R. W., 1986. Approximating edges of source bodies from magnetic or gravity anomalies. *Geophysics*, 51, 1494–1498.
- Boashash, B., 2015. *Time-frequency signal analysis and processing: a comprehensive reference*. Academic Press.
- Carter, N. L., Horseman, S. T., Russell, J. E., and Handin, J., 1993. Rheology of rocksalt. *Journal of Structural Geology*, 15, 1257–1271.
- Chirila, D. B. and Lohmann, G., 2015. *Introduction to Modern Fortran for the Earth System Sciences*. Springer.
- Coker, M. O., Bhattacharya, J. P., and Marfurt, K. J., 2007. Fracture patterns within mudstones on the flanks of a salt dome: Syneresis or slumping? *Gulf Coast Association of Geological Societies Transactions*, 125–137.
- Colombo, D. and De Stefano, M., 2007. Geophysical modeling via simultaneous joint inversion of seismic, gravity, and electromagnetic data: Application to prestack depth imaging. *The Leading Edge*, 26, 326–331.
- Cooper, G. R. J., 2011. The semiautomatic interpretation of gravity profile data. *Computers & geosciences*, 37, 1102–1109.
- Corana, A., Marchesi, M., Martini, C., and Ridella, S., 1987. Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 13, 262–280.
- Cordell, L., 1979. Gravimetric expression of graben faulting in Santa Fe country and the Española basin, New Mexico. *New Mexico Geol. Soc. Guidebook, 30th Field Conf*, 59–64.

- Couder-Castañeda, C., Ortiz-Alemán, C., Orozco-del-Castillo, M. G., and Nava-Flores, M., 2013. TESLA GPUs versus MPI with OpenMP for the forward modeling of gravity and gravity gradient of large prisms ensemble. *Journal of Applied Mathematics*, 2013.
- Couder-Castañeda, C., Ortiz-Alemán, C., Orozco-del-Castillo, M. G., and Nava-Flores, M., 2015. Forward modeling of gravitational fields on hybrid multi-threaded cluster. *Geofísica internacional*, 54, 31–48.
- Debeglia, N. and Corpel, J., 1997. Automatic 3-D interpretation of potential field data using analytic signal derivatives. *Geophysics*, 62, 87–96.
- Dribus, J. R., Jackson, M. P. A., Kapoor, J., and Smith, M. F., 2008. The prize beneath the salt. *Oilfield Review*, 15, 4–17.
- D'Urso, M. G., 2014. Analytical computation of gravity effects for polyhedral bodies. *Journal of Geodesy*, 88, 13–29.
- Ennen, C., Hall, S., et al., 2011. Structural mapping of the Vinton salt dome, Louisiana, using gravity gradiometry data. *2011 SEG Annual Meeting*. Society of Exploration Geophysicists.
- Evjen, H. M., 1936. The place of the vertical gradient in gravitational interpretations. *Geophysics*, 1, 127–136.
- Ferreira, F. J. F., de Souza, J., de B. e S. Bongiolo, A., and de Castro, L. G., 2013. Enhancement of the total horizontal gradient of magnetic anomalies using the tilt angle. *Geophysics*, 78, J33–J41.
- Garcia-Abdeslem, J., 1992. Gravitational attraction of a rectangular prism with depth-dependent density. *Geophysics*, 57, 470–473.
- Gardner, G. H. F., Gardner, L. W., and Gregory, A. R., 1974. Formation velocity and density—The diagnostic basics for stratigraphic traps. *Geophysics*, 39, 770–780.
- Geng, M., Huang, D., Yang, Q., and Liu, Y., 2014. 3D inversion of airborne gravity-gradiometry data using cokriging. *Geophysics*, 79, G37–G47.
- Gherasim, M., Hoelting, C., and Marfurt, K., 2005. 3-D VSP elastic Kirchhoff pre-stack depth migration—Vinton Dome, Louisiana. *Geophysics*, 67, 1886–1894.
- Goffe, W. L., Ferrier, G. D., and Rogers, J., 1994. Global optimization of statistical functions with simulated annealing. *Journal of econometrics*, 60, 65–99.
- Grant, F. S. and West, G. F., 1965. *Interpretation theory in applied geophysics*. McGraw-Hil Book Co. New York.
- Hinze, W. J., Von Frese, R. R. B., and Saad, A. H., 2013. *Gravity and magnetic exploration: Principles, practices, and applications*. Cambridge University Press.

- Hou, Z.-L., Wei, X.-H., Huang, D.-N., and Sun, X., 2015. Full tensor gravity gradiometry data inversion: Performance analysis of parallel computing algorithms. *Applied Geophysics*, 12, 292–302.
- Hsu, S. K., Sibuet, J. C., and Shyu, C. T., 1996. High-resolution detection of geologic boundaries from potential-field anomalies: An enhanced analytic signal technique. *Geophysics*, 61, 373–386.
- Hudec, M. R. and Jackson, M. P. A., 2007. Terra infirma: Understanding salt tectonics. *Earth-Science Reviews*, 82, 1–28.
- Kellogg, O. D., 1967. *Foundations of Potential Theory*. Springer-Verlag.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., 1983. Optimization by simulated annealing. *Science*, 220, 671–680.
- Krahenbuhl, R. A., 2005. “Binary inversion of gravity data for salt imaging.” PhD thesis. Colorado School of Mines. Colorado, USA.
- Li, X., 2006. Understanding 3D analytic signal amplitude. *Geophysics*, 71, L13–L16.
- Li, Y. and Oldenburg, D. W., 1998. 3-D inversion of gravity data. *Geophysics*, 63, 109–119.
- Lourenco, J. S. and Morrison, H. F., 1973. Vector magnetic anomalies derived from measurements of a single component of the field. *Geophysics*, 38, 359–368.
- Marson, I. and Klingele, E. E., 1993. Advantages of using the vertical gradient of gravity for 3-D interpretation. *Geophysics*, 58, 1588–1595.
- Martin, R., Monteiller, V., Komatitsch, D., Perrouy, S., Jessell, M., Bonvalot, S., and Lindsay, M., 2013. Gravity inversion using wavelet-based compression on parallel hybrid CPU/GPU systems: application to southwest Ghana. *Geophysical Journal International*, 195, 1594–1619.
- Menke, W., 1989. *Geophysical data analysis: Discrete inverse theory*. Academic Press, INC.
- Mickus, K. L. and Hinojosa, J. H., 2001. The complete gravity gradient tensor derived from the vertical component of gravity: a Fourier transform technique. *Journal of Applied Geophysics*, 46, 159–174.
- Miller, H. G. and Singh, V., 1994. Potential field tilt—a new concept for location of potential field sources. *Journal of Applied Geophysics*, 32, 213–217.
- Montgomery, S. L. and Moore, D. C., 1997. Subsalt play, Gulf of Mexico: A review. *AAPG Bulletin*, 81.
- Murphy, C. A. and Brewster, J., 2007. Target delineation using full tensor gravity gradiometry data. *ASEG Extended Abstracts*, 2007, 1–3.
- Nabighian, M. N., 1972. The analytic signal of two-dimensional magnetic bodies with polygonal cross-section: Its properties and use for automated anomaly interpretation. *Geophysics*, 37, 507–517.

- Nabighian, M. N., 1984. Toward a three-dimensional automatic interpretation of potential field data via generalized Hilbert transforms: Fundamental relations. *Geophysics*, 49, 780–786.
- Nagihara, S. and Hall, S. A., 2001. Three-dimensional gravity inversion using simulated annealing: Constraints on the diapiric roots of allochthonous salt structures. *Geophysics*, 66, 1438–1449.
- Nagy, D., Papp, G., and Benedek, J., 2000. The gravitational potential and its derivatives for the prism. *Journal of Geodesy*, 74, 552–560.
- Nava-Flores, M., Ortiz-Alemán, C., Orozco-del-Castillo, M. G., Urrutia-Fucugauchi, J., Rodríguez-Castellanos, A., Couder-Castañeda, C., and Trujillo-Alcántara, A., 2016. 3D Gravity Modeling of Complex Salt Features in the Southern Gulf of Mexico. *International Journal of Geophysics*, 2016.
- Nelson, T. H. and Fairchild, L. H., 1989. Emplacement and evolution of salt sills in northern Gulf of Mexico. *AAPG (Am. Assoc. Pet. Geol.) Bull.:(United States)*, 73.
- Newton, I., 1687. *Philosophiae Naturalis Principia Mathematica (Mathematical Principles of Natural Philosophy)*.
- Okabe, M., 1979. Analytical expressions for gravity anomalies due to homogeneous polyhedral bodies and translations into magnetic anomalies. *Geophysics*, 44, 730–741.
- Oliveira, V. C. and Barbosa, V. C. F., 2013. 3-D radial gravity gradient inversion. *Geophysical Journal International*, 195, 883–902.
- Orozco-del-Castillo, M. G., Ortiz-Alemán, C., Urrutia-Fucugauchi, J., Martin, R., Rodríguez-Castellanos, A., and Villaseñor-Rojas, P. E., 2014. A genetic algorithm for filter design to enhance features in seismic images. *Geophysical Prospecting*, 62, 210–222.
- Ortiz-Alemán, C. and Martin, R., 2005a. Inversion of electrical capacitance tomography data by simulated annealing: Application to real two-phase gas–oil flow imaging. *Flow Measurement and Instrumentation*, 16, 157–162.
- Ortiz-Alemán, C. and Martin, R., 2005b. Two-phase oil–gas pipe flow imaging by simulated annealing. *Journal of Geophysics and Engineering*, 2, 1–6.
- Ortiz-Alemán, C., Martin, R., and Gamio, J. C., 2004. Reconstruction of permittivity images from capacitance tomography data by using very fast simulated annealing. *Measurement Science and Technology*, 15, 1382–1390.
- Ortiz-Alemán, C. and Urrutia-Fucugauchi, J., 2010. Aeromagnetic anomaly modeling of central zone structure and magnetic sources in the Chicxulub crater. *Physics of the Earth and Planetary Interiors*, 179, 127–138.
- Oruç, B., Sertçelik, I., Kafadar, Ö., and Selim, H. H., 2013. Structural interpretation of the Erzurum Basin, eastern Turkey, using curvature gravity gradient tensor and gravity inversion of basement relief. *Journal of Applied Geophysics*, 88, 105–113.

- Oruç, B., 2011. Edge detection and depth estimation using a tilt angle map from gravity gradient data of the Kozaklı-Central Anatolia Region, Turkey. *Pure and applied geophysics*, 168, 1769–1780.
- Pedersen, L. B. and Rasmussen, T. M., 1990. The gradient tensor of potential field anomalies: Some implications on data collection and data processing of maps. *Geophysics*, 55, 1558–1566.
- Plouff, D., 1976. Gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections. *Geophysics*, 41, 727–741.
- Qin, P., Huang, D., Yuan, Y., Geng, M., and Liu, J., 2016. Integrated gravity and gravity gradient 3D inversion using the non-linear conjugate gradient. *Journal of Applied Geophysics*, 126, 52–73.
- Reid, A. B., Allsop, J. M., Granser, H., Millett, A. J., and Somerton, I. W., 1990. Magnetic interpretation in three dimensions using Euler deconvolution. *Geophysics*, 55, 80–91.
- René, R. M., 1986. Gravity inversion using open, reject, and “shape-of-anomaly” fill criteria. *Geophysics*, 51, 988–994.
- Rodríguez-Zúñiga, J. L., Ortiz-Alemán, C., Padilla, G., and Gaulon, R., 1997. Application of genetic algorithms to constrain shallow elastic parameters using in situ ground inclination measurements. *Soil Dynamics and Earthquake Engineering*, 16, 223–234.
- Roest, W. R., Verhoef, J., and Pilkington, M., 1992. Magnetic interpretation using the 3-D analytic signal. *Geophysics*, 57, 116–125.
- Roy, L., Sen, M. K., Blankenship, D. D., Stoffa, P. L., and Richter, T. G., 2005. Inversion and uncertainty estimation of gravity data using simulated annealing: An application over Lake Vostok, East Antarctica. *Geophysics*, 70, J1–J12.
- Salem, A., Masterton, S., Campbell, S., Fairhead, J. D., Dickinson, J., and Murphy, C., 2013. Interpretation of tensor gravity data using an adaptive tilt angle method. *Geophysical Prospecting*, 61, 1065–1076.
- Selman, D., 2010. *Processing and Acquisition of Air-FTG® Data*. Final Report. Houston, Texas, USA: Bell Geospace, Inc.
- Sen, M. K. and Stoffa, P. L., 2013. *Global optimization methods in geophysical inversion*. Cambridge University Press.
- Singh, B. and Gupthasarma, D., 2001. New method for fast computation of gravity and magnetic anomalies from arbitrary polyhedra. *Geophysics*, 66, 521–526.
- Stavrev, P. and Reid, A., 2006. Degrees of homogeneity of potential fields and structural indices of Euler deconvolution. *Geophysics*, 72, L1–L12.
- Talwani, M. and Ewing, M., 1960. Rapid computation of gravitational attraction of three-dimensional bodies of arbitrary shape. *Geophysics*, 25, 203–225.

- Tarantola, A., 2005. *Inverse problem theory and methods for model parameter estimation*. SIAM.
- Telford, W. M., Geldart, L. P., and Sheriff, R. E., 1990. *Applied geophysics*. Vol. 1. Cambridge University Press.
- Thompson, D. T., 1982. EULDPH: A new technique for making computer-assisted depth estimates from magnetic data. *Geophysics*, 47, 31–37.
- Thompson, S. A. and Eichelberger, O. H., 1928. Vinton salt dome, Calcasieu Parish, Louisiana. *AAPG Bulletin*, 12, 385–394.
- Uieda, L. and Barbosa, V. C. F., 2012. Robust 3D gravity gradient inversion by planting anomalous densities. *Geophysics*, 77, G55–G66.
- Verduzco, B., Fairhead, J. D., Green, C. M., and MacKenzie, C., 2004. New insights into magnetic derivatives for structural mapping. *The Leading Edge*, 23, 116–119.
- Wang, T. H., Huang, D. N., Ma, G. Q., Meng, Z. H., and Li, Y., 2017. Improved preconditioned conjugate gradient algorithm and application in 3D inversion of gravity - gradiometry data. *Applied Geophysics*, 14, 301–313.
- Zhang, C., Mushayandebvu, M. F., Reid, A. B., Fairhead, J. D., and Odegard, M. E., 2000. Euler deconvolution of gravity tensor gradient data. *Geophysics*, 65, 512–520.
- Zhang, J., Wang, C. Y., Shi, Y., Cai, Y., Chi, W. C., Dreger, D., Cheng, W. B., and Yuan, Y. H., 2004. Three-dimensional crustal structure in central Taiwan from gravity inversion with a parallel genetic algorithm. *Geophysics*, 69, 917–924.
- Zhou, W., Du, X., and Li, J., 2013. The limitation of curvature gravity gradient tensor for edge detection and a method for overcoming it. *Journal of Applied Geophysics*, 98, 237–242.

ANEXOS

A

PUBLICACIÓN DERIVADA DE ESTA TESIS Y ARTÍCULOS RELACIONADOS

En este anexo se puede consultar la publicación derivada de este trabajo de tesis, así como otras dos publicaciones relacionadas que tratan acerca de modelado gravimétrico 3D a través de ensambles de grandes dimensiones, implementado con técnicas de cómputo de alto rendimiento, en las que ha participado el autor de esta.

Artículo derivado de este trabajo de tesis:

3D Gravity Modeling of Complex Salt Features in the Southern Gulf of Mexico.
(Nava-Flores et al., [2016](#)).

Artículos relacionados con participación del autor de esta tesis:

Forward modeling of gravitational fields on hybrid multi-threaded cluster.
(Couder-Castañeda et al., [2015](#)).

TESLA GPUs versus MPI with OpenMP for the forward modeling of gravity and gravity gradient of large prisms ensemble.
(Couder-Castañeda et al., [2013](#)).

Research Article

3D Gravity Modeling of Complex Salt Features in the Southern Gulf of Mexico

Mauricio Nava-Flores,¹ Carlos Ortiz-Aleman,² Mauricio G. Orozco-del-Castillo,² Jaime Urrutia-Fucugauchi,³ Alejandro Rodriguez-Castellanos,² Carlos Couder-Castañeda,⁴ and Alfredo Trujillo-Alcantara²

¹Facultad de Ingeniería, Universidad Nacional Autónoma de México (UNAM), Avenida Universidad No. 3000, CU, Coyoacán, 04510 Ciudad de México, DF, Mexico

²Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas No. 152, San Bartolo Atepehuacan, Gustavo A. Madero, 07730 Ciudad de México, DF, Mexico

³Programa de Perforaciones en Océanos y Continentes, Instituto de Geofísica, Universidad Nacional Autónoma de México, 04510 Ciudad de México, DF, Mexico

⁴Centro de Desarrollo Aeroespacial, Instituto Politécnico Nacional, Belisario Domínguez No. 22, 06010 Ciudad de México, DF, Mexico

Correspondence should be addressed to Mauricio G. Orozco-del-Castillo; maorca42@yahoo.com

Received 30 November 2015; Accepted 3 February 2016

Academic Editor: Robert Tenzer

Copyright © 2016 Mauricio Nava-Flores et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a three-dimensional (3D) gravity modeling and inversion approach and its application to complex geological settings characterized by several allochthonous salt bodies embedded in terrigenous sediments. Synthetic gravity data were computed for 3D forward modeling of salt bodies interpreted from Prestack Depth Migration (PSDM) seismic images. Density contrasts for the salt bodies surrounded by sedimentary units are derived from density-compaction curves for the northern Gulf of Mexico's oil exploration surveys. By integrating results from different shape- and depth-source estimation algorithms, we built an initial model for the gravity anomaly inversion. We then applied a numerically optimized 3D simulated annealing gravity inversion method. The inverted 3D density model successfully retrieves the synthetic salt body ensemble. Results highlight the significance of integrating high-resolution potential field data for salt and subsalt imaging in oil exploration.

1. Introduction

Hydrocarbon exploration is largely based on geophysical methods among which seismic reflection is the most intensely employed. Increased interest in subsalt related plays in the Gulf of Mexico and in other sedimentary basins around the world has turned oil and gas prospecting within these regions into a major challenge. Physical property contrasts of salt features such as highly contrasting seismic velocities relative to the surrounding media lead to complex wave diffraction patterns and lack of illumination near and below them.

In this context, gravity methods are well suited to support seismic prospecting and improve subsalt imaging by taking full advantage of the density contrasts between salt bodies and surrounding sedimentary targets. Salt bodies retain low

densities, whereas upon burial sediments compact increasing the density contrast. Table 1 shows typical ranges for seismic wave velocity, density, and permeability of salt. The seismic wave high velocity contrasts at salt-sediment interface result in strong refractions and reflections, making it difficult to image the bottom and structures beneath salt bodies. Major advances with improved images of subsalt plays have resulted from prestack imaging, with velocity-depth modeling and Prestack Depth Migration (PSDM). Nevertheless, subsalt structural complexities present major barriers, requiring new approaches and integrative analyses of seismic and potential field data.

Complex geological imaging using modeling and inversion of potential field anomalies has been examined in recent studies. Ortiz-Alemán and Urrutia-Fucugauchi [1] applied

TABLE 1: P-wave velocity, density, and permeability of the rock salt bodies placed in oil and gas prospecting zones (SI units).

Property	Range of values	Reference
Seismic velocity (P-wave)	4270 to 5190 [m/s]	Grant and West, 1967 [8]
Density	2,100 to 2,200 [kg/m ³]	Gardner et al., 1974 [9]
Permeability	<10 ⁻²⁰ [m ²]	Carter et al., 1993 [10]

3D magnetic field modeling to the study of the central zone of the Chicxulub impact structure, which had proved difficult to image from seismic reflection data. Nagihara and Hall [2] applied the simulated annealing (SA) global optimization method to invert synthetic gravity data due to a simplified salt diapir. They constrained the shape of the diapir at depth through inverse modeling in 3D. Zhang et al. [3] determined the crustal structure of central Taiwan through gravity inversion with a parallel genetic algorithm, using an initial model derived from 3D P-wave velocity tomography. Roy et al. [4] inverted gravity data using SA over Lake Vostok in East Antarctica in order to estimate the water-sediment and sediment-basement interfaces. Krahenbuhl [5] introduced an approach called binary inversion, which uses an assemblage of equal-volume prisms as model space, and density contrasts as model parameters, which only could take one of two possible values: zero for prisms located in salt-free zones and one for prisms placed in salt areas. Rene [6] developed a method of gravity inversion by iteratively applying open, reject, and fill criteria within a modeling procedure based on the use of prisms ensembles with density contrasts previously assigned and the “shape-of-anomaly” fill criterion. Uieda and Barbosa [7] performed a 3D gravity inversion method by planting anomalous densities. They used the “shape-of-anomaly” data misfit in conjunction with the L_2 -norm data-misfit function achieving better delineation of elongated sources and the recovery of compact geologic bodies.

In this work we built a 3D gravity model including several allochthonous salt bodies as interpreted from a Prestack Depth Migration seismic volume, integrating results from different potential field techniques such as edge-source detection, depth-to-source estimation, and 3D gravity inversion.

2. Materials and Methods

2.1. Forward Gravity Modeling of Salt Structures. The whole computational domain, including several salt features, was discretized into an ensemble of rectangular prismatic elements. Its gravity response, that is, g_z , was calculated by summing the individual responses of every single prismatic element, on all points belonging to the observation grid. Figure 1 shows a sketch of the gravity response generated by an ensemble formed by M prisms, computed over an observation plane.

The total gravity response calculated at some observation point was the sum of the gravity contributions generated by the M prisms of the ensemble:

$$f_i = \sum_{j=1}^M f_i^j, \quad (1)$$

where f_i represents the gravity response on the observation point i and f_i^j is the gravity response due to the particular prism j , observed on the position i .

Now, the vertical component g_z of the gravity vector, due to each individual rectangular prism, with constant density ρ , according to Plouff [11], Blakely [12], and Nagy et al. [13], is

$$g_z = \frac{\partial U}{\partial z} = \gamma \rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 S_i S_j S_k \left[z_k \arctan \left(\frac{x_i y_i}{z_k R_{ijk}} \right) - x_i \ln (R_{ijk} + y_i) - y_i \ln (R_{ijk} + x_i) \right], \quad (2)$$

where γ is the universal gravitational constant, $R_{ijk} = \sqrt{x_i^2 + y_j^2 + z_k^2}$, $S_1 = -1$, and $S_2 = 1$. Figure 1(b) shows the elements involved in (2).

As illustrated in Figure 1(b), and according to (2), to calculate the gravity response of a rectangular prismatic body it is necessary to determine the density and coordinates of its extreme vertices. In this approach it is just necessary to know the density, the coordinates of only one point, and the volume grid interval of the ensemble to which it belongs.

Taking into account the above and the fact that salt bodies embedded in terrigenous sediments are geometrically irregular, they can be modeled as ensembles of regular rectangular prisms, formed by discrete points. Analyzed salt bodies (Figure 2) were imaged from digital 3D Prestack Depth Migration seismic velocity modeling, using real data from areas located in the southeastern sector of the Gulf of Mexico.

On computing the gravity response of the salt bodies illustrated in Figure 2, a background density, representative of the surrounding sedimentary rocks, is needed. Curves of sedimentary rocks densities, representative of the Gulf of Mexico, were published by the Applied Geodynamics Laboratory (AGL) by Hudec and Jackson [16], based on the work of Nelson and Fairchild [14], in which they propose that the density of those sedimentary units could be modeled as a function of depth (Figure 3) by the exponential curve:

$$\rho(z) = 1400 + 172z^{0.21}, \quad (3)$$

where ρ is the sediments density [kg/m³] and z is depth [m].

The gravity response is therefore calculated by first obtaining a density contrast between the salt bodies and the surrounding sediments, in the position of each point source (prism), subtracting (3) from each single salt prism, assuming that its density ρ_s is constant, with the value $\rho_s = 2,180$ [kg/m³].

Figure 4 shows the gravity anomaly calculated from the ensemble formed by all the prismatic sources simulating the

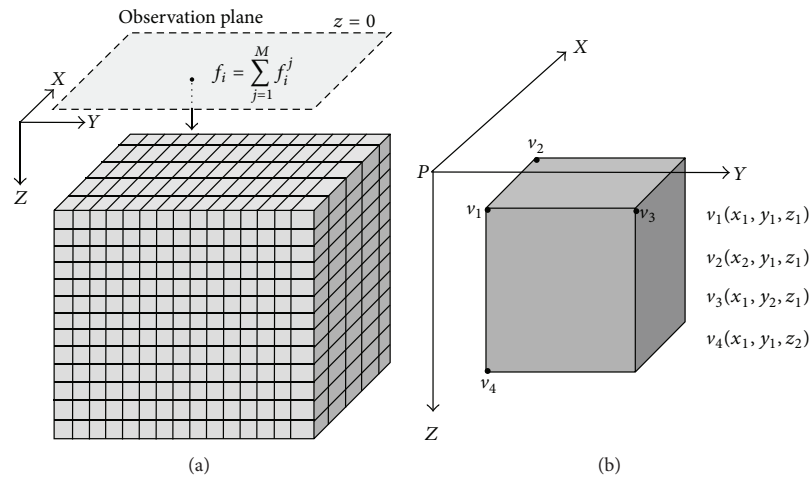


FIGURE 1: (a) Discretized media formed by M rectangular prisms and the vertical component of the gravity response calculated over an observation point at the surface observation plane $Z = 0$ and (b) prismatic body located in a right rectangular coordinate system. P is the observation point, where the gravity response is computed.

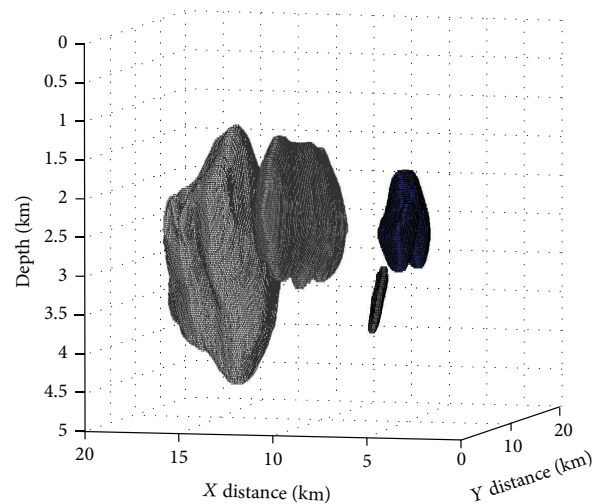


FIGURE 2: Salt bodies interpreted from a 3D PSDM velocity model. Each body is composed of many discrete points with a regular volume grid interval.

salt bodies depicted in Figure 3. The number of rectangular prisms considered in the calculation of the anomaly was 201,540; each prism size is $50 \text{ [m]} \times 50 \text{ [m]} \times 25 \text{ [m]}$ (X , Y , and Z directions). The grid interval for the observation grid in both directions, X and Y , was 0.4 km , and the number of observation points in both directions was 51, that is, 2,601 grid observation points.

2.2. Shape and Depth Estimation of Salt Structures. Several methods especially suited to enhance anomalies and estimate depth to source are commonly applied to potential field data. While there are methods that use systematic search algorithms to find a solution of the distribution of the densities of the model [7], and others which use a great

amount of rectangular prisms to obtain a good approximation of the gravimetric anomalies [6], in this work we applied a series of those approaches to gravity gridded data (Figure 4), in order to infer an initial 3D density distribution for inverse modeling. These methods include the Horizontal Gradient (HG), as proposed by Cordell [17], the 3D Analytic Signal Amplitude (AS), developed by Nabighian [18], the Enhanced Analytic Signal (EAS) introduced by Hsu et al. [19], and the 3D Euler deconvolution (3DED) developed by Reid et al. [20].

Figure 5 shows the results obtained after applying the HG, AS, and EAS methods to the gravity anomaly of Figure 4. Location of the maxima in those grids was estimated by the method of Blakely and Simpson [21] and roughly corresponds to the lateral extent of the gravity field sources, that is, the salt

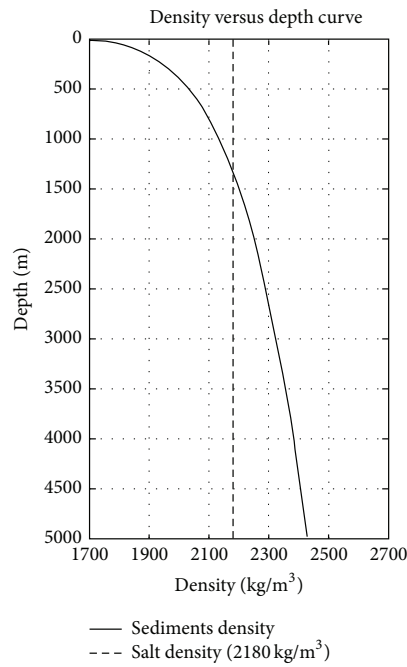


FIGURE 3: Density versus depth curve, representative of sediments of Gulf of Mexico and rock salt density. Based on Nelson and Fairchild [14].

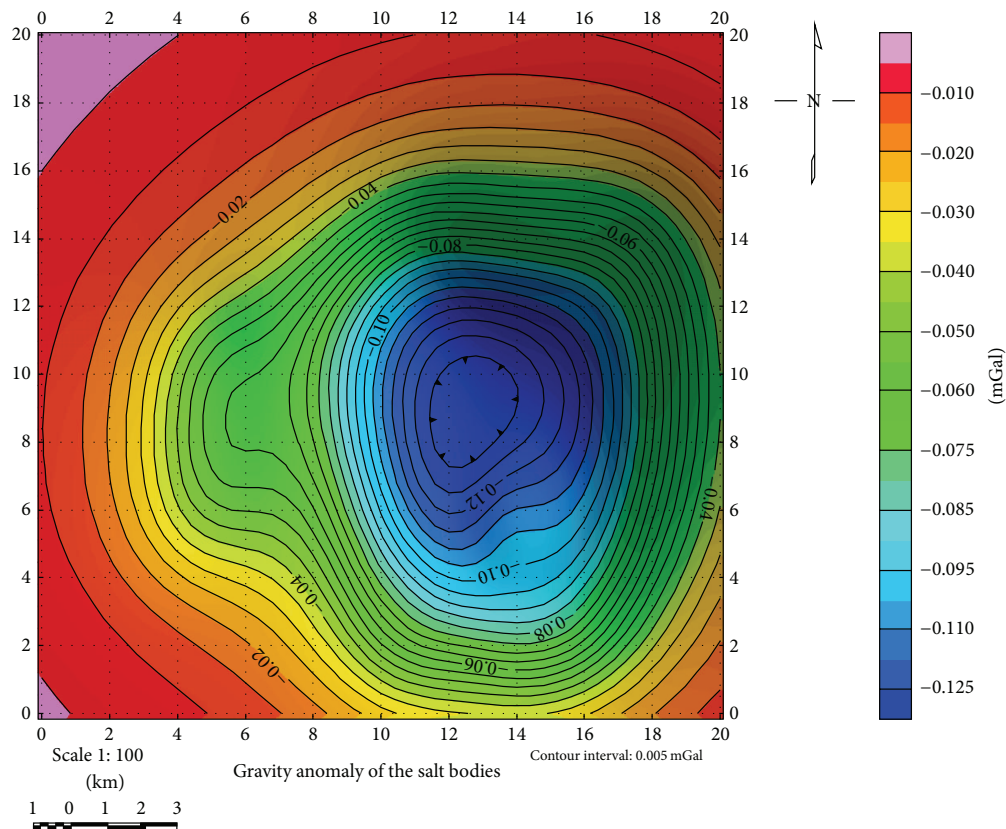


FIGURE 4: Gravity anomaly caused by the salt bodies depicted in Figure 2.

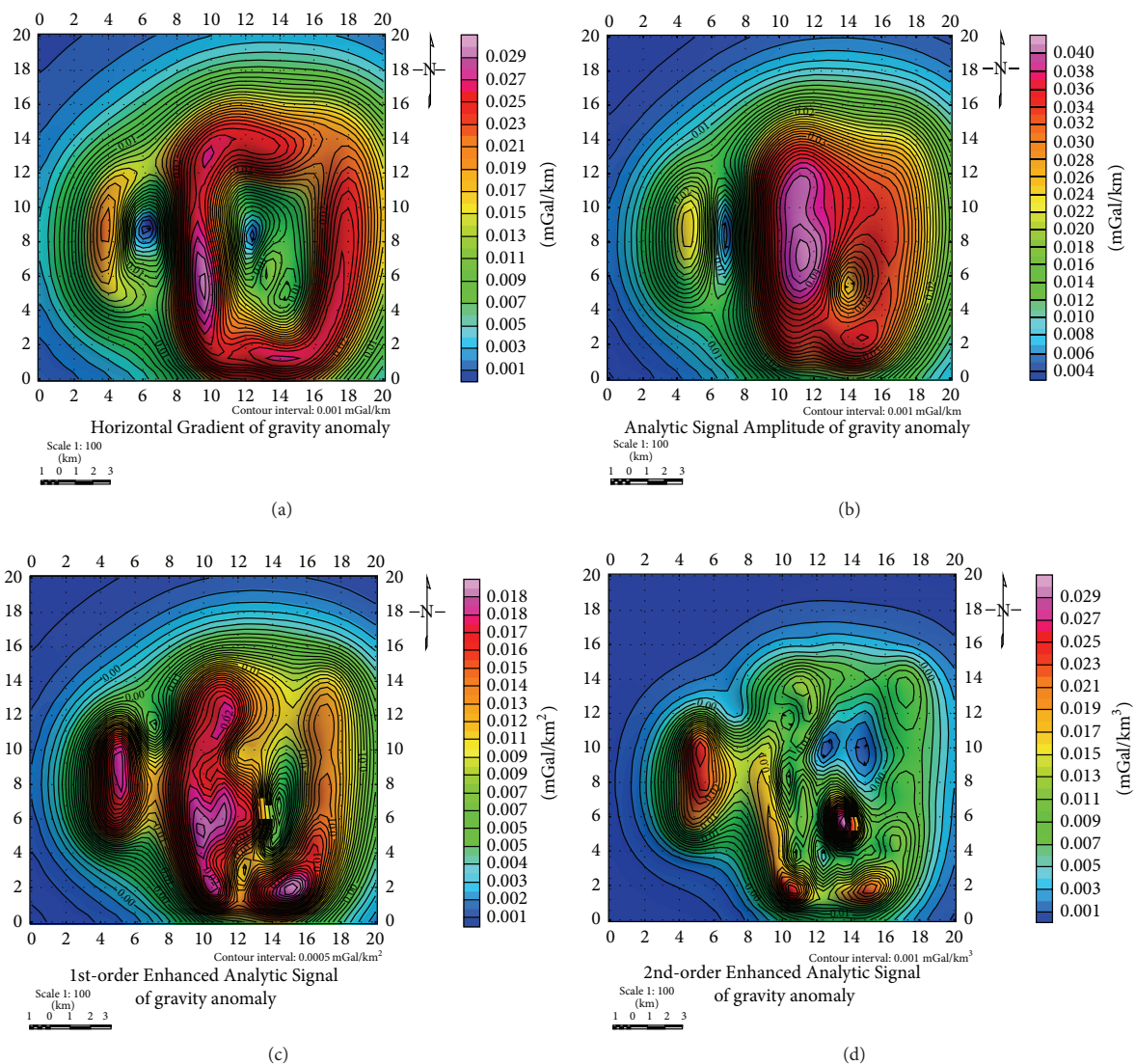


FIGURE 5: Lateral extent of source bodies from a set of depth-to-source estimation methods. (a) Horizontal Gradient, (b) Analytic Signal Amplitude, (c) 1st-order EAS, and (d) 2nd-order Enhanced Analytic Signal.

bodies in a plan view. The edges of salt bodies as interpreted from these maxima are shown in Figure 6.

After estimation of projected source location on a horizontal plane (plant-view of sources), the corresponding distribution of the sources with depth was inferred in order to build an initial 3D structural model. For this purpose, we applied the 3DED algorithm to the gravity anomaly grid, considering a structural index $N = 0$, assuming a geologic contact-type of source [22], and a 4 km size square window. The computed solutions are shown in Figure 7.

We built a 3D structural model including two huge salt diapirs surrounded by sedimentary rocks with relative density contrast assigned as a function of depth (3), by considering source location in plant (Figure 6) and the 3D Euler deconvolution solutions (Figure 7). We then computed

the gravity anomaly for this 3D model, in order to evaluate how well it correlates with the gravity anomaly response from the originally postulated salt bodies, shown in Figure 4.

Figure 8 shows this 3D model, labeled as 3D initial model (3DIniM), in three orthogonal projections, and a 3D perspective oblique view. The salt masses were presented gray colored for display purposes, and their gravity anomaly is shown in Figure 9.

Table 2 summarizes the main characteristics of this 3D initial model.

The computed gravity anomaly for the 3D initial model qualitatively resembles the shape of the anomaly produced by the salt bodies interpreted from a PSDM volume (Figure 4) but in terms of the amplitude of relative error remains still quite large. To minimize such error, we inverted the

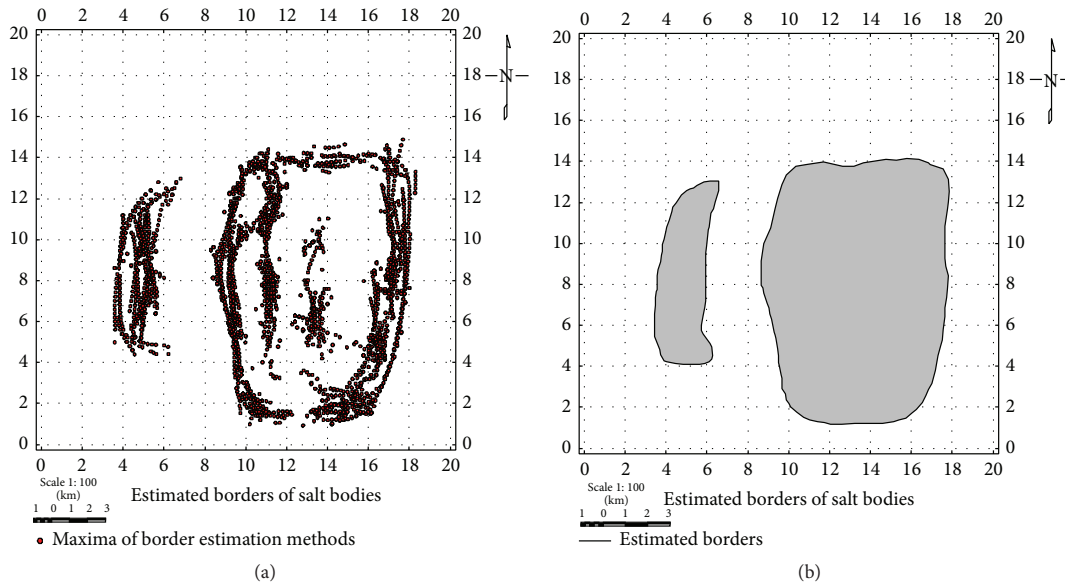


FIGURE 6: (a) Maxima location estimated from the methods depicted in Figure 5 and (b) sources in a plan view as interpreted from those maxima.

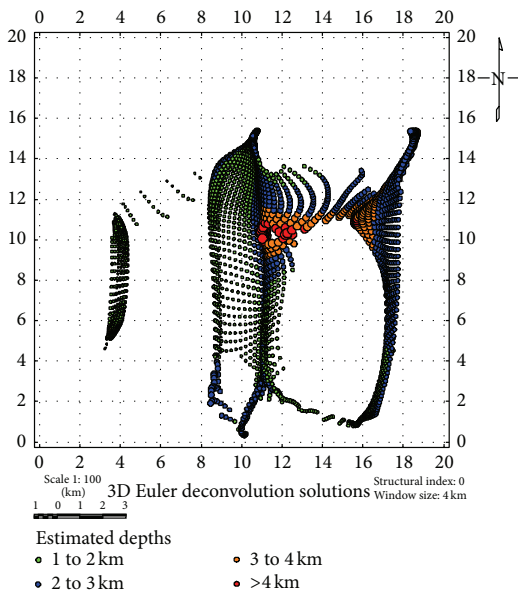


FIGURE 7: Distribution with depth of gravity anomaly sources estimated by the 3DED algorithm.

gravity anomaly by using a numerically optimized simulated annealing algorithm, as discussed in the next section.

2.3. 3D Inversion of Gravity Data by Simulated Annealing. According to (2), the gravity anomaly of the entire prism ensemble is the sum of the gravity anomalies generated by each of the individual prisms, so we can rewrite (2) as

$$f_i = \sum_{j=1}^M g_i(\rho_j), \quad \text{for } i = 1, \dots, N. \quad (4)$$

This is a linear system of equations, where N denotes the amount of observation points and M is the number of prisms in the ensemble. The linear system can be also represented as

$$f_i = G_{ij} \cdot \rho_j. \quad (5)$$

Here, G_{ij} are the elements in the sensitivity kernel or sensitivity matrix. Each one of its elements accounts for the contribution to the complete gravity anomaly in the i observation point, due to the prism located on the j position inside the ensemble.

To solve the inverse problem, we chose the simulated annealing global optimization method. A main drawback of global optimization lies in the excessive amount of forward problem computations required to solve the inverse problem. In the past decades, global optimization has been successfully applied to several geophysical exploration issues, where dimensionality of the inverse problem did not represent a bottleneck [23, 24].

The simulated annealing method was conceived as a mathematical analogy with the natural optimization process of crystal formation from a mineral fluid at high temperature. Its basic concepts were taken from the statistical mechanics.

The simulated annealing optimization process emulates the evolution of a physical system as it slowly cools down and crystallizes at a state of minimum energy. If temperature, T , is gradually reduced after the thermal equilibrium has been reached, then in the limit, as $T \rightarrow 0$, the minimum energy state becomes predominantly likely, as well as the crystal formation, and therefore the parameter configuration could be considered as optimum model.

Following Kirkpatrick et al. [25], we used the Metropolis algorithm as central part of our simulated annealing method, taking advantage of its ability to escape from local minima,

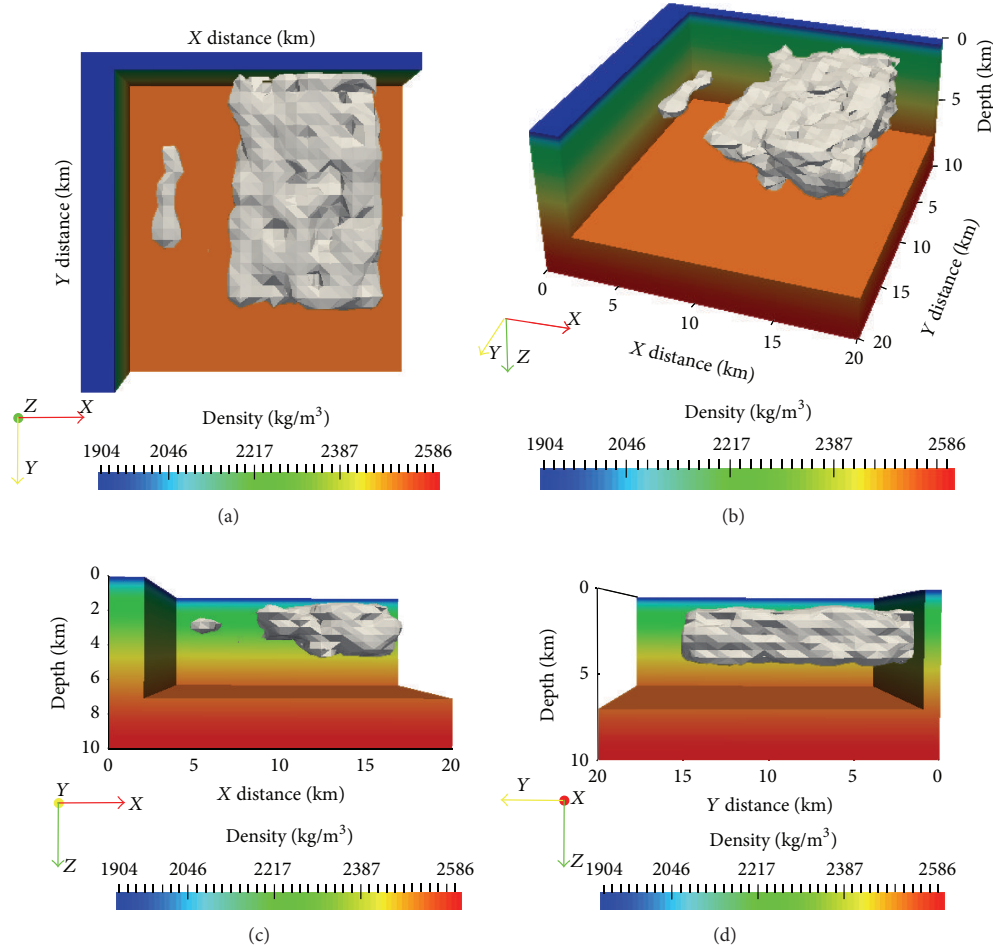


FIGURE 8: 3D initial model built from 3D Euler deconvolution solutions and the maxima of the lateral extent estimation methods previously applied to the gravity anomaly.

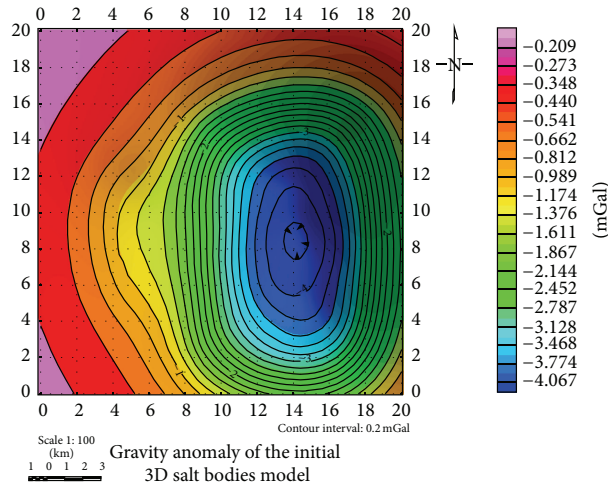


FIGURE 9: Gravity anomaly caused by the 3D initial model shown in Figure 8.

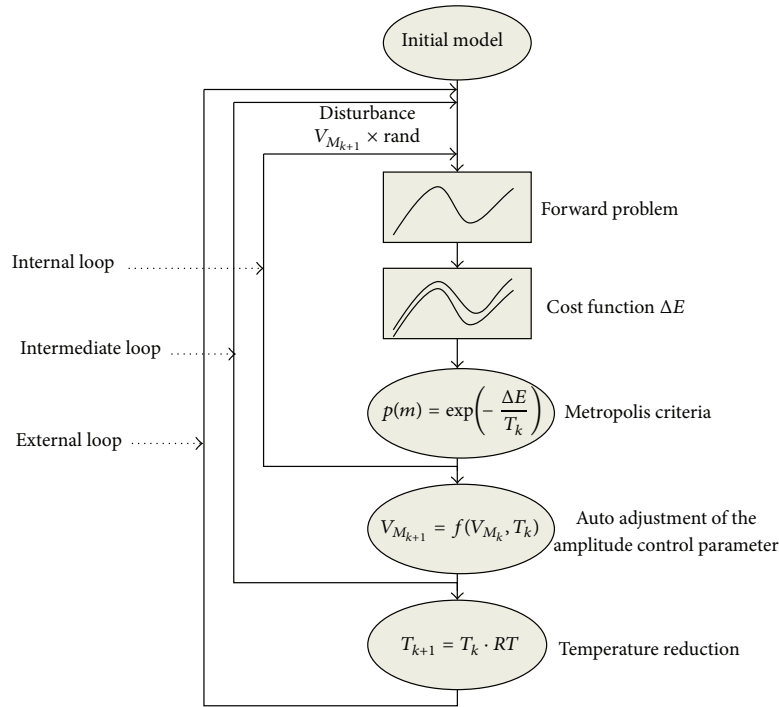


FIGURE 10: Flux diagram representation of the simulated annealing algorithm applied in this work (modified from Ortiz-Alemán and Martin [15]).

TABLE 2: 3D initial model main characteristics.

Model size	Ensemble discretization	Density range
X direction: 20,000 [m] Y direction: 20,000 [m] Z direction: 10,000 [m]	X direction: 30 prisms Y direction: 30 prisms Z direction: 30 prisms Total number of prisms: 27,000 Individual prism size: 666.66 [m] × 666.66 [m] × 333.33 [m] (X, Y, and Z directions)	1,900 to 2,590 [kg/m ³]

which increases the chances to reach the global optimum, and at the same time approaches to the Boltzmann probability density function asymptotically [26]. It consists basically in disturbing some initial model, m_i , which already has the energy content $E(m_i)$, and getting a new model, m_j , with energy $E(m_j)$, and then calculating the energy level change due to the disturbance applied, $\Delta E_{ij} = E(m_j) - E(m_i)$, and accepting or rejecting m_j on the basis of the value of ΔE_{ij} calculated: if $\Delta E_{ij} \leq 0$, then m_j will be unconditionally accepted, but in the case that $\Delta E_{ij} > 0$, m_j will be accepted with the probability $P(m_j) = \exp(-\Delta E_{ij}/T)$.

This acceptance-rejection procedure is repeated several times for a fixed temperature, T_i , until the thermal equilibrium is reached, which is characterized by not exhibiting substantial changes in the energy level before the temperature reduction.

To compute the energy level in each stage, we used a normalized L_2 norm [27, 28], given by

$$E = L_2 = \frac{\sum_{k=1}^N (d_k^{\text{obs}} - d_k^{\text{calc}})^2}{\sum_{k=1}^N (d_k^{\text{obs}})^2}, \quad (6)$$

where d_k^{obs} is the observed gravity anomaly and d_k^{calc} is the gravity anomaly calculated for the m_k model.

The cooling schedule we choose reduces the temperature in an exponential fashion by multiplying the actual temperature by some parameter RT in each temperature cycle pass and, according to Nagihara and Hall [2], is characterized by ensuring convergence to the global minimum:

$$T_k = T_i (RT)^k, \quad (7)$$

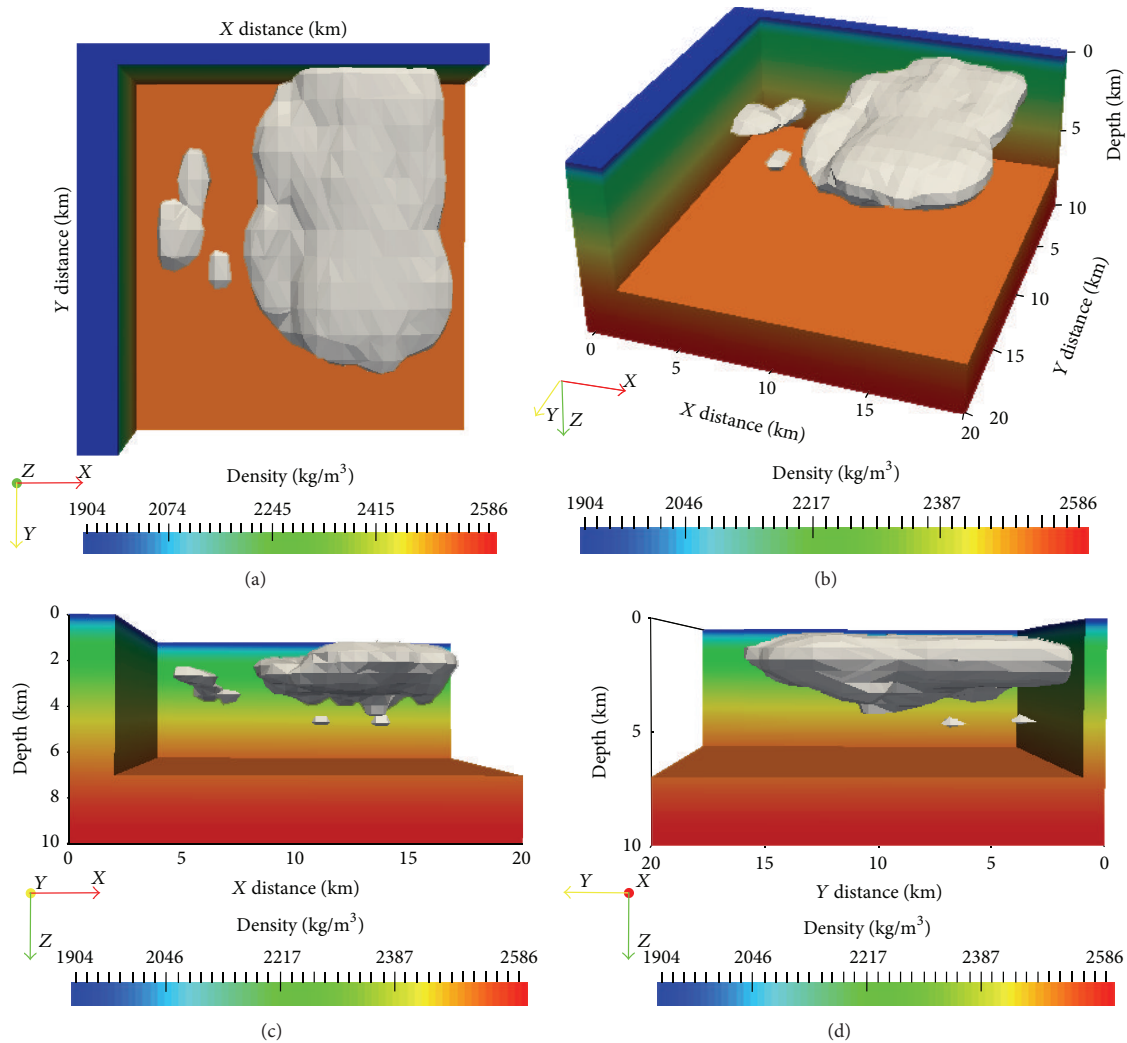


FIGURE 11: 3D Inverted Model generated from the 3D gravity anomaly inversion.

where T_i is the initial temperature of the system, T_k is the temperature at k th stage, and RT is the reduction temperature parameter ($0 < RT < 1$).

Finally, this process is repeated until reaching the limit $T \rightarrow 0$ or controlled by some stop criterion given as a tolerance error with respect to $E(m_j)$ and, at the same time, by a maximum number of temperature reductions.

One first improvement made to the basic simulated annealing method in this work was to accelerate the product $G_{ij} \cdot \rho_j$, as proposed by Ortiz-Alemán and Martín [15] by using a previously computed forward problem, and using it to update the actual one, summing it to the product $\Delta m \cdot G_{ij}$ (Δm is the disturbance applied to the model parameter m_i). This improvement justifies the employment of a global optimization method in the inversion of a quite large linear system, as after the first iteration it is no longer required to compute the forward problem for a complete ensemble, and hence the numerical burden will be dramatically reduced.

The final improvement made to the SA method consisted in applying an auto adjustment to the amplitude control parameter, V_M from iteration k to $k+1$, proposed by Corana et al. [29]. Let $r = N_A/N_R$ be the relation between the numbers of accepted (N_A) and rejected models (N_R) by the Metropolis criterion. If $r > 0.6$, then $V_{M_{k+1}} = V_{M_k} (1 + c((r - 0.6)/0.4))$, and if $r < 0.4$, then $V_{M_{k+1}} = V_{M_k} / (1 + c((0.4 - r)/0.4))$, where c is a constant value fixed at 2.0.

The final SA inversion algorithm is summarized in the diagram shown in Figure 10. Its three-nested-loop structure is based on the algorithm presented by Goffe et al. [30].

This modified SA algorithm was applied to the gravity anomaly data generated by the postulated set of salt bodies, with the following restrictions:

- (1) The lateral extent of all models generated by the inversion procedure was restricted to the interpreted source borders (Figure 7).

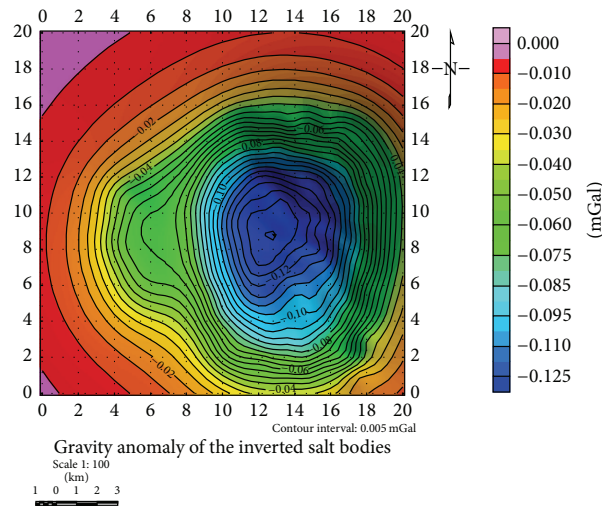


FIGURE 12: Gravity anomaly caused by the 3DInvM shown in Figure 11.

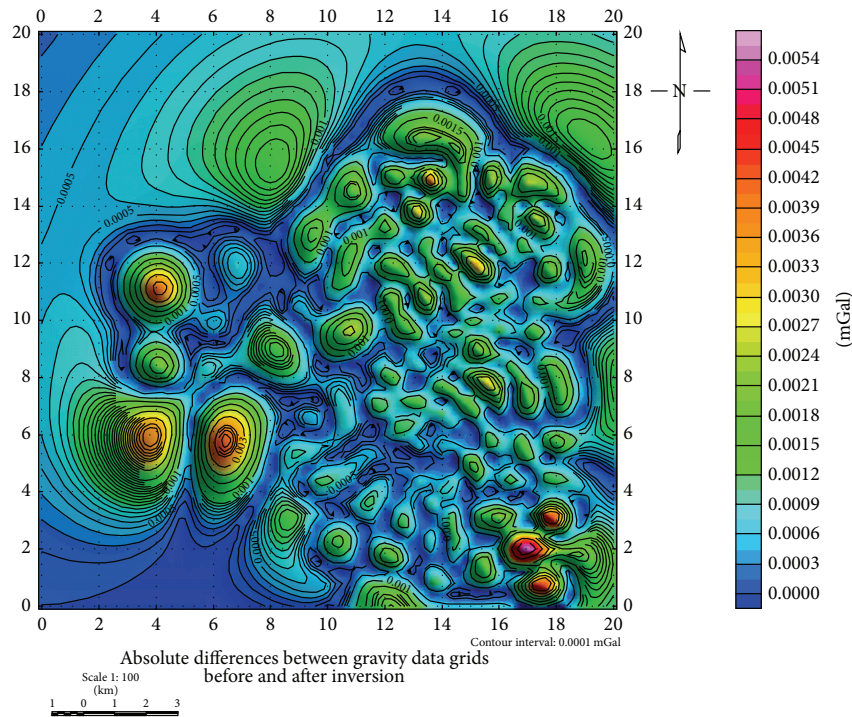


FIGURE 13: Spatial distribution of residuals along the grid and their amplitudes, computed as the absolute differences between gravity data before and after inversion.

- (2) The model space was bounded according to the salt and sediments density contrast (Figure 4).

Table 3 shows the 3D gravity data inversion parameters of the SA algorithm method as applied in this work.

3. Results and Discussion

The 3D density model resulting from the inversion procedure (Figure 11) is shown in the same projections and perspective

angles as displayed in the 3DIniM. In order to differentiate this inverted model from the initial one, it is labeled as 3D Inverted Model (3DInvM).

The gravity anomaly grid generated by the 3DInvM shows that, despite the apparent differences in the central part of the grid corresponding to the gravity minimum, the amplitudes are similar to the observed gravity (Figure 12).

In order to quantify the quality of the 3DInvM, we calculated the differences between the gridded gravity anomaly

TABLE 3: Parameter values related with the 3D gravity data inversion procedure.

Parameters	Values
Number of inverted parameters	27,000
Number of observed data points	2,601
Initial temperature	1.0
Final temperature	$0.16e - 8$
Energy of the initial model	33.484543
Energy of the final model	0.0178259
Number of temperature reduction steps	1,000
Reduction temperature factor	0.98
Previous cycles to the V_M parameter auto adjustment	10
Cycles of thermal equilibrium	5
Total number of tried models	149,400,000
Number of accepted models	115,281,390
Number of rejected models	34,118,610

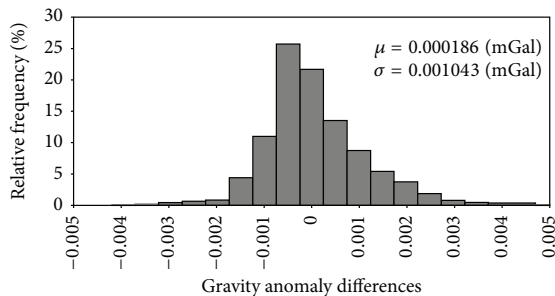


FIGURE 14: Relative frequency distribution of residuals shown in Figure 13, including mean and standard deviation values.

data of the salt bodies (Figure 4) and the 3DInvM (Figure 12). The absolute values of these differences are shown in Figure 13, illustrating the spatial distribution of residuals along the grid and their amplitudes, whose maximum difference (0.005624884 [mGal]) is 4.26% of Figure 12 total range (0.132137625 [mGal]). The histogram and the mean and standard deviation values of the residuals are shown in Figure 14, where the mean value (-0.000186 [mGal]) and low standard deviation (0.001043 [mGal]) indicate that the inverted gravity anomaly successfully resembles the observed gravity anomaly. Based on this last fact we find our inversion results as very encouraging.

The misfit curve, representing the relationship between temperature and energy parameters along the inversion process, exhibits three different kinds of convergence rates: a gradual decay in the beginning of the inversion, an intermediate region of sharp decreasing misfit, and a zone of progressively slower convergence rates until final entrapment (Figure 15).

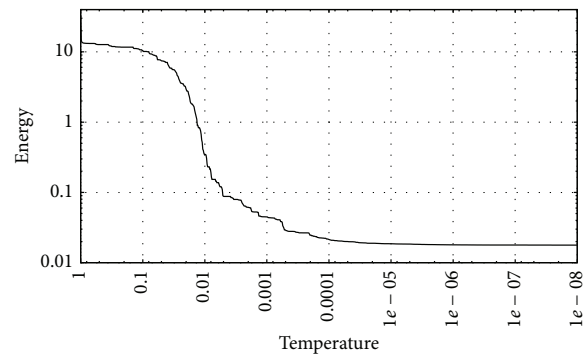


FIGURE 15: Temperature versus energy curve in the 3D gravity inversion process.

4. Conclusions

In this study we applied 3D gravity modeling and inversion in a complex geological setting involving several allochthonous salt features embedded in terrigenous sediments, representing a challenging and quite realistic scenario commonly found in the southern Gulf of Mexico.

Several methods especially suited to enhance anomalies and estimate depth to source are used in this work to determine an initial 3D density distribution for inverse modeling. These methods include the Horizontal Gradient (HG), the 3D Analytic Signal Amplitude (AS), the Enhanced Analytic Signal (EAS), and the 3D Euler deconvolution (3DED). We built an initial density model by integrating results from this set of shape- and depth-source estimation algorithms. We finally applied a numerically optimized three-dimensional simulated annealing gravity inversion approach. As the total amount of evaluated forward models in this study case was quite large (~150 million), application to other realistic gravity modeling efforts should consider the use of high performance computing for the forward and inverse problems, as recently introduced by Couder-Castañeda et al. [31, 32] and Martin et al. [33]. Results highlight the significance of integrating high-resolution potential field data to imaging complex salt tectonics media for oil exploration.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors of this work acknowledge the financial support provided by SENER-CONACyT Project 128376 and Mexican Institute of Petroleum Projects D.00475 and H.61006.

References

- [1] C. Ortiz-Alemán and J. Urrutia-Fucugauchi, "Aeromagnetic anomaly modeling of central zone structure and magnetic sources in the Chicxulub crater," *Physics of the Earth and Planetary Interiors*, vol. 179, no. 3-4, pp. 127-138, 2010.

- [2] S. Nagihara and S. A. Hall, "Three-dimensional gravity inversion using simulated annealing: constraints on the diapiric roots of allochthonous salt structures," *Geophysics*, vol. 66, no. 5, pp. 1438–1449, 2001.
- [3] J. Zhang, C.-Y. Wang, Y. Shi et al., "Three-dimensional crustal structure in central Taiwan from gravity inversion with a parallel genetic algorithm," *Geophysics*, vol. 69, no. 4, pp. 917–924, 2004.
- [4] L. Roy, M. K. Sen, D. D. Blankenship, P. L. Stoffa, and T. G. Richter, "Inversion and uncertainty estimation of gravity data using simulated annealing: an application over Lake Vostok, East Antarctica," *Geophysics*, vol. 70, no. 1, pp. J1–J12, 2005.
- [5] R. A. Krahenbuhl, *Binary Inversion of Gravity Data for Salt Imaging*, Colorado School of Mines, 2005.
- [6] R. M. Rene, "Gravity inversion using open, reject, and 'shape-of-anomaly' fill criteria," *Geophysics*, vol. 51, no. 4, pp. 988–994, 1986.
- [7] L. Uieda and V. C. F. Barbosa, "Use of the 'shape-of-anomaly' data misfit in 3D inversion by planting anomalous densities," in *SEG Technical Program Expanded Abstracts*, vol. 4, pp. 1–6, 2012.
- [8] F. S. Grant and G. F. West, *Interpretation Theory in Applied Geophysics*, vol. 130, McGraw Hill, New York, NY, USA, 1967.
- [9] G. H. F. Gardner, L. W. Gardner, and A. R. Gregory, "Formation velocity and density—the diagnostic basics for stratigraphic traps," *Geophysics*, vol. 39, no. 6, pp. 770–780, 1974.
- [10] N. L. Carter, S. T. Horseman, J. E. Russell, and J. Handin, "Rheology of rocksalt," *Journal of Structural Geology*, vol. 15, no. 9–10, pp. 1257–1271, 1993.
- [11] D. Plouff, "Gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections," *Geophysics*, vol. 41, no. 4, pp. 727–741, 1976.
- [12] R. J. Blakely, *Potential Theory in Gravity and Magnetic Applications*, Cambridge University Press, Cambridge, UK, 1995.
- [13] D. Nagy, G. Papp, and J. Benedek, "The gravitational potential and its derivatives for the prism," *Journal of Geodesy*, vol. 74, no. 7–8, pp. 552–560, 2000.
- [14] T. H. Nelson and L. Fairchild, "Emplacement and evolution of salt sills in the northern Gulf of Mexico," *Houston Geological Society Bulletin*, vol. 32, no. 1, pp. 6–7, 1989.
- [15] C. Ortiz-Alemán and R. Martin, "Inversion of electrical capacitance tomography data by simulated annealing: application to real two-phase gas-oil flow imaging," *Flow Measurement and Instrumentation*, vol. 16, no. 2–3, pp. 157–162, 2005.
- [16] M. R. Hudec and M. P. A. Jackson, "Terra infirma: understanding salt tectonics," *Earth-Science Reviews*, vol. 82, no. 1–2, pp. 1–28, 2007.
- [17] L. Cordell, "Gravimetric expression of graben faulting in Santa Fe country and the Espanola Basin," in *New Mexico Geological Guidebook, 30th Field Conference*, no. 1977, pp. 59–64, 1979.
- [18] M. N. Nabighian, "Toward a three-dimensional automatic interpretation of potential field data via generalized Hilbert transforms: fundamental relations," *Geophysics*, vol. 49, no. 6, pp. 780–786, 1984.
- [19] S. K. Hsu, J. C. Sibuet, and C. T. Shyu, "High-resolution detection of geologic boundaries from potential-field anomalies: an enhanced analytic signal technique," *Geophysics*, vol. 61, no. 2, pp. 373–386, 1996.
- [20] A. B. Reid, J. M. Allsop, H. Granser, A. J. Millett, and I. W. Somerton, "Magnetic interpretation in three dimensions using Euler deconvolution," *Geophysics*, vol. 55, no. 1, pp. 80–91, 1990.
- [21] R. J. Blakely and R. W. Simpson, "Approximating edges of source bodies from magnetic or gravity anomalies," *Geophysics*, vol. 51, no. 7, pp. 1494–1498, 1986.
- [22] P. Stavrev and A. Reid, "Degrees of homogeneity of potential fields and structural indices of Euler deconvolution," *Geophysics*, vol. 72, no. 1, pp. L1–L12, 2007.
- [23] J. L. Rodríguez-Zúñiga, C. Ortiz-Alemán, G. Padilla, and R. Gaulon, "Application of genetic algorithms to constrain shallow elastic parameters using in situ ground inclination measurements," *Soil Dynamics and Earthquake Engineering*, vol. 16, no. 3, pp. 223–234, 1997.
- [24] M. G. Orozco-del-Castillo, C. Ortiz-Alemán, J. Urrutia-Fucugauchi, R. Martin, A. Rodriguez-Castellanos, and P. E. Villaseñor-Rojas, "A genetic algorithm for filter design to enhance features in seismic images," *Geophysical Prospecting*, vol. 62, no. 2, pp. 210–222, 2014.
- [25] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [26] M. K. Sen and P. L. Stoffa, *Global Optimization Methods in Geophysical Inversion*, Cambridge University Press, Cambridge, UK, 2013.
- [27] C. Ortiz-Aleman, R. Martin, and J. C. Gamio, "Reconstruction of permittivity images from capacitance tomography data by using very fast simulated annealing," *Measurement Science and Technology*, vol. 15, no. 7, pp. 1382–1390, 2004.
- [28] J. C. Ortiz-Alemán and R. Martin, "Two-phase oil-gas pipe flow imaging by simulated annealing," *Journal of Geophysics and Engineering*, vol. 2, no. 1, pp. 32–37, 2005.
- [29] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the 'simulated annealing' algorithm," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, 1987.
- [30] W. L. Goffe, G. D. Ferrier, and J. Rogers, "Global optimization of statistical functions with simulated annealing," *Journal of Econometrics*, vol. 60, no. 1–2, pp. 65–99, 1994.
- [31] C. Couder-Castañeda, C. Ortiz-Alemán, M. G. Orozco-del-Castillo, and M. Nava-Flores, "TESLA GPUs versus MPI with OpenMP for the forward modeling of gravity and gravity gradient of large prisms ensemble," *Journal of Applied Mathematics*, vol. 2013, Article ID 437357, 15 pages, 2013.
- [32] C. Couder-Castañeda, J. C. Ortiz-Alemán, M. G. Orozco-del-Castillo, and M. Nava-Flores, "Forward modeling of gravitational fields on hybrid multi-threaded cluster," *Geofísica Internacional*, vol. 54, no. 1, pp. 31–48, 2015.
- [33] R. Martin, V. Monteiller, D. Komatitsch et al., "Gravity inversion using wavelet-based compression on parallel hybrid CPU/GPU systems: application to southwest Ghana," *Geophysical Journal International*, vol. 195, no. 3, pp. 1594–1619, 2013.

Forward modeling of gravitational fields on hybrid multi-threaded cluster

Carlos Couder-Castañeda*, José Carlos Ortiz-Alemán, Mauricio Gabriel Orozco-del-Castillo and Mauricio Nava-Flores

Received: October 18, 2013; accepted: March 11, 2014; published on line: December 12, 2014

Resumen

La solución analítica de las componentes del tensor gravimétrico, utilizando la ecuación del potencial gravitacional para un ensamble volumétrico compuesto de prismas de densidad constante, requiere un alto costo computacional. Esto se debe a que el potencial gravitacional de cada uno de estos prismas tiene que ser calculado para todos los puntos de una malla de observación previamente definida, lo cual resulta en una carga computacional de gran escala. En este trabajo introducimos un diseño híbrido y su implementación paralela basada en OpenMP y MPI, para el cálculo de las componentes vectoriales del campo gravimétrico (G_x, G_y, G_z) y las componentes del tensor gravimétrico ($G_{xx}, G_{xy}, G_{xz}, G_{yy}, G_{yz}, G_{zz}$). El rendimiento obtenido conlleva a óptimas relaciones del speed-up, ya que el tiempo de cómputo es drásticamente reducido. La técnica de paralelización aplicada consiste en descomponer el problema en grupos de prismas y utilizar diferentes espacios de memoria por núcleo de procesamiento, con el fin de evitar los problemas de cuello de botella cuando se accesa a la memoria compartida de un nodo del cluster, que se producen generalmente cuando varios hilos de ejecución acceden a la misma región en OpenMP. Debido a que OpenMP solo puede utilizarse en sistemas de memoria compartida es necesario utilizar MPI para la distribución del cálculo entre los nodos del cluster, dando como resultado un código híbrido OpenMP+MPI altamente eficiente con un speed-up prácticamente perfecto. Adicionalmente los resultados numéricos fueron validados con respecto a su contraparte secuencial.

Palabras clave: gravedad, gradiometría, OpenMP, MPI, hyper-threading, clusters.

Abstract

The analytic solution of the gravimetric tensor components, making use of the gravitational potential equation for a three-dimensional volumetric assembly composed of unit prisms of constant density, demands a high computational cost. This is due to the gravitational potential of each one of these prisms must be calculated for all of the points of a previously defined observation grid, which turns out in a large scale computational cost. In this work we introduce a hybrid design and its parallel implementation, based on OpenMP and MPI, for the calculation of the vectorial components of the gravimetric field and the components of the gravimetric tensor. Since the computing time is drastically reduced, the obtained performance leads close to optimal speed-up ratios. The applied parallelization technique consists of decomposing the problem into groups of prisms and using different memory allocations per processing core to avoid bottleneck issues when accessing the main memory in one cluster node, which are generally produced when using too many execution threads over the same region in OpenMP. Due OpenMP can be only used on shared memory systems is necessary to use MPI for the calculation distribution among cluster nodes, giving as a result a hybrid code (OpenMP+MPI) highly efficient and with a nearly perfect speed-up. Additionally the numerical results were validated with respect to its sequential counterpart.

Keywords: gravity, gradiometry, OpenMP, MPI, hyper-threading, clusters.

C. Couder-Castañeda*
 J. C. Ortiz-Alemán
 M. Gabriel Orozco-del-Castillo
 Mexican Petroleum Institute
 Eje Central Lázaro Cárdenas, 152
 San Bartolo Atepehuacan, Gustavo A. Madero
 07730, Ciudad de México
 *Corresponding author: ccouder@esfm.ipn.mx

M. Nava Flores
 División de Ingeniería en Ciencias de la Tierra
 Facultad de Ingeniería
 Universidad Nacional Autónoma de México
 Ciudad Universitaria
 Delegación Coyoacán, 04510
 México D.F., México

Introduction

The shared memory architecture is becoming more common every day in the high-performance computing market. With the hardware technology advances allowing us to have a great number of cores with access to the same memory locations, nowadays it is not that expensive to have systems with forty or sixty cores using shared memory. OpenMP is now a standard for symmetric multiprocessing systems (SMP) (even can be used transparently in the Xeon Phi architecture (Calvin *et al.*, 2013)) sustained by a combination of function and compiler directives, a standard for the symmetric multiprocessing (SMP) systems (Dagum and Menon, 1998; Curtis-Maury *et al.*, 2008). OpenMP has proven to be a powerful tool for SMP due to several reasons: it is highly portable; it allows fine and medium granularity, each thread can access to the same global memory; and has their own private memory, and it also has a greater level of abstraction than MPI model (Brunst and Mohr, 2008).

MPI is a library supported on the Same Program Multiple Data (SPMD) model and on the message passing model, with an explicit control of the parallelism. The processes can only read and write in their respective local memories and the data in these memories is transferred through calls to functions or procedures which implement the message passing model. Among the principal characteristics of MPI are that it can run in architectures of shared and distributed memory, is convenient for medium to coarse granularity and that employment is widely extended, making it extremely portable among platforms (Krpic *et al.*, 2012).

Using a hybrid programming model we can take advantage of the benefits of two programming models OpenMP and MPI. MPI is normally used to control the parallelism among cluster nodes, while OpenMP is applied in the creation of threads of fine granularity tasks within each node. Most applications developed in hybrid model involves a hierarchical model: MPI is for the higher level and OpenMP for the lower one (Smith, 2000).

One of the potential benefits of using hybrid model programming consists of getting rid of the barrier of scaling that each model has. Generally, in MPI the scaling is limited by the communications cost, because an application is affected by the overload of communication when the number of processes is increased. In OpenMP the performance of an application is affected by cache coherence problems and access to shared memory

which may lead to bottleneck issues between the execution threads when trying to access memory. By mixing these methodologies of parallel programming (OpenMP and MPI), we can obtain a more diverse granularity of the application and therefore a better performance than by using each one on its own.

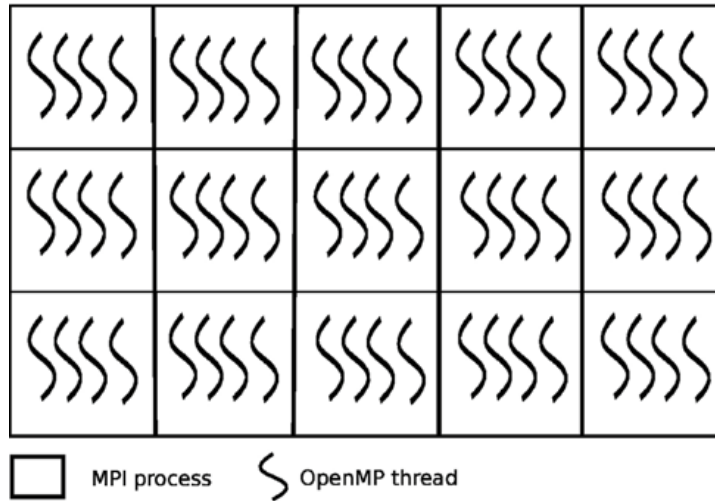
There are different applications which use this programming paradigm: OpenMP with MPI. For example, in the solution of sparse linear systems (Mitin *et al.*, 2012), in graph-coloring algorithms (Sariyuce *et al.*, 2012), in some models of fluid dynamics (Amritkar *et al.*, 2012; Couder-Castañeda, 2009) and finite element methods (Boehmer *et al.*, 2012), in the simulation of turbulent fluids (Jagannathan and Donzis, 2012), even in the simulation of combustion chambers (Környei, 2012) and the implementation of neural networks (Gonzalez *et al.*, 2012). As can be observed, there are numerous computational implementations using OpenMP with MPI, nevertheless, this type of design is supported on a natural decomposition of the domain (Carrillo-Ledesma *et al.*, 2013), based on data. For our particular problem, each one of the processing units accesses all of the computational domain points.

In Figure 1 is depicted a domain decomposition, where each task (process or thread) is given some data subset on which to work. This domain decomposition is commonly used for example in finite differences problems where computational domains divided disjointly among the different tasks.

On the other hand, in the direct conformation of gravimetric data, an initial model for the source body is constructed from geological-geophysical information. The anomaly of such model is calculated and compared to the observed anomaly, after which the parameters are adapted to improve the adjustment between them. These three steps that arrange the model properties — *anomalies calculation*, *comparison* and *adjustment* — are repeated up to the observed and calculated anomalies are similar enough.

A mass volume can be approximated by a set of rectangular prisms; if chosen sufficiently small, each prism can be considered to have a constant density. Because of the superposition principle, the gravitational anomaly of a body can be approximated at any point by summing the effects of all the prisms over that point. Even though this methodology appears simple (by reducing the size of the prisms to better adjust the source body),

Figure 1. The domain decomposition based on data for an OpenMP+MPI application.



computing time is considerably increased. There are other approaching methods of the gravitational anomaly that can simplify the required computation (mass points or tesseroids approximations), however, they may complicate the construction of the geological model (Heck and Seitz, 2007).

Application design

The application consists of calculating the gravimetric anomaly produced by a rectangular prismatic body with constant density with respect to a group of observation points (see Figure 2). The set of prisms is known as an ensemble of prisms, which is not necessarily regular. A set of irregular prisms can be configured as long as the prisms are not superimposed. Because the gravitational field complies with the superposition principle with respect to the observation points, if f is the calculated response at a point (x, y) , then the

observed response at the point $f(x, y)$ is given by:

$$f(x, y) = \sum_{k=1}^M G(\rho_k, x, y) \quad (1)$$

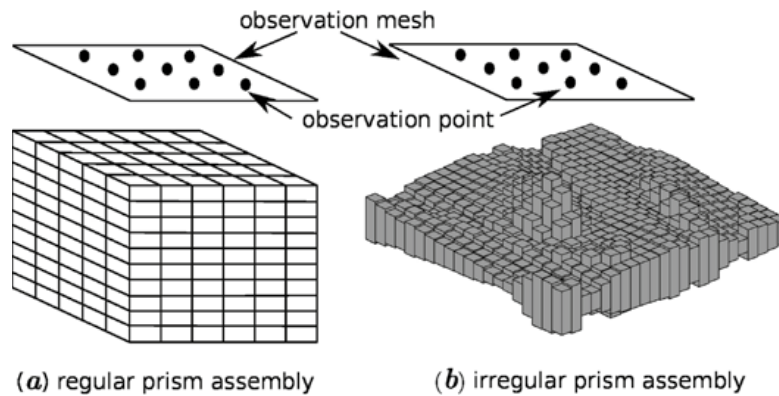
where M is the number of total prisms and ρ is the density of the prism.

It is well known that the function that calculates the anomaly for a given prism from an observation point is written as follows (Nagy *et al.*, 2000):

$$g = f(x_p, y_p, z_p, x_r, y_r, z_r, x_p, y_p, z_p, \rho) \quad (2)$$

where (x_p, y_p, z_p) is the top left vertex of the prism, (x_r, y_r, z_r) is the bottom right prism and (x_p, y_p, z_p) is the observation point and ρ the density, as shown in Figure 3.

Figure 2. Decomposition of the calculation of M prisms with respect to the observation grid: (a) regular prism assembly, (b) irregular prism assembly.



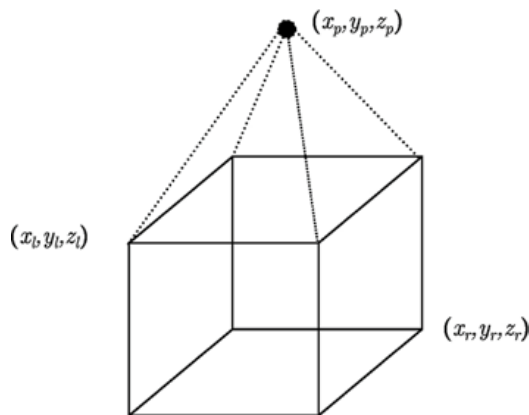


Figure 3. Calculation of a prism with respect to a point of observation.

The aforementioned is a large scale problem since, for example, a synthetic problem conformed by a set of prisms of $300 \times 300 \times 150 = 13,500,000$ elements, against an observation grid of $100 \times 100 = 10,000$ points, results in the calculation of 135,000,000,000 integrals or differentials to solve the entire problem. The formulations we used are included in appendix A.

Computing time reduction in a numerical simulation is of great importance to diminish research costs. A simulation which lasts a week is likely to be costly, not only because the machine time is expensive, but also because it prohibits the quick acquisition of results to make modifications and predictions.

In many projects to be parallelized, several times the serial algorithm does not show a natural decomposition which allows easily porting it to a parallel environment, or the trivial decomposition does not yield good performance results. For such reasons it is convenient to use a hybrid programming methodology, as the one developed and presented in this paper. This methodology provides an adequate programming design to obtain a superior performance.

To develop a parallel program it is fundamental to search for the finest granularity, as in the methodology proposed by Foster (Foster, 1995). In this case it is possible to parallelize by prisms or by observation points. One of the requirements of the design is that it must be scalable, therefore the use of hybrid systems is quite appropriate; these systems are the most commonly used nowadays. Following Foster's methodology, it is necessary to begin with the finest granularity, in this case

corresponds to OpenMP because it is in the lowest level. Subsequently the implementation follows with MPI, due to its coarse granularity.

Implementation in OpenMP

We started our design with OpenMP because it handles shared memory and it is also the finest granularity. First we partitioned the domain into prisms, and for each prism we parallelized the calculation by observation points, as shown in Figure 4.

This parallelization by observation points is trivial and does not offer a great design challenge, since we simply partition the calculation with respect to the observation grid for each prism (see the pseudo-code 1). However, this scheme has several drawbacks. One of them is that the performance is not optimal since the number of prisms is much greater than the number of observation points. In other words, this partitioning is efficient as long as there are not too many threads working upon the observation grid, thus avoiding a bottleneck issue as a consequence of the threads works in the same memory allocation. Maybe the worst drawback lies in the fact that the parallel environment is created and closed, i.e. for each prism, a function which parallelly calculates the anomalies is executed, but such environment is closed once the execution is over, and reopened for the following prism, which results in an unnecessary overload and therefore decreases the performance.

Listing 1. Parallelization by observation points

```
For each prism from 1 to M
!$OMP PARRALLEL DO COLLAPSE(2)
  For each j from 1 to Ny
    For each i from 1 to Nx
      G(i,j)=Gz(parameters)+G(i,j)
    End For
  End For
!$OMP END PARALLEL DO
End For
```

Listing 2. Parallelization by observation points

```
!$OMP PARRALLEL DO
For each prism from 1 to M
  For each j from 1 to Ny
    For each i from 1 to Nx
      G(i,j)=Gz(parameters)+G(Thread,
      i,j)
    End For
  End For
End For
!$OMP END PARALLEL DO
```

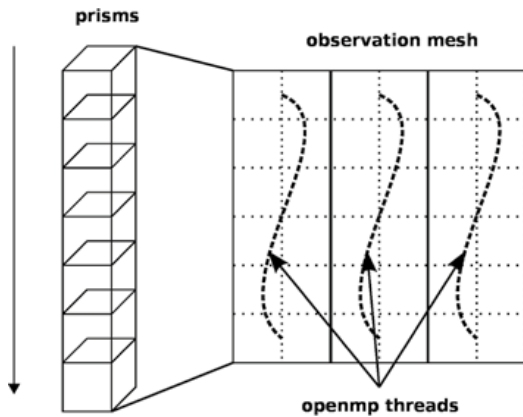


Figure 4. Partitioning by observation points.

The other parallelization option is to use prisms i.e., making the threads divide the work per number of prisms (see pseudo-code 2). To avoid the coherence problems of the cache it is necessary to create a different memory space for each execution thread, because it is not feasible to create a single memory space for a unique observation grid, shared by all the threads.

As observed in Figure 6, it is required to create an observation grid for each execution thread to avoid memory consistency problems. Bottleneck memory access issues are avoided since every thread writes in a different direction of the memory space. If only one grid were to be used, there would be access problems to the shared grid, which would create numerical inconsistencies.

One of the characteristics of OpenMP is that the computing is distributed in an implicit

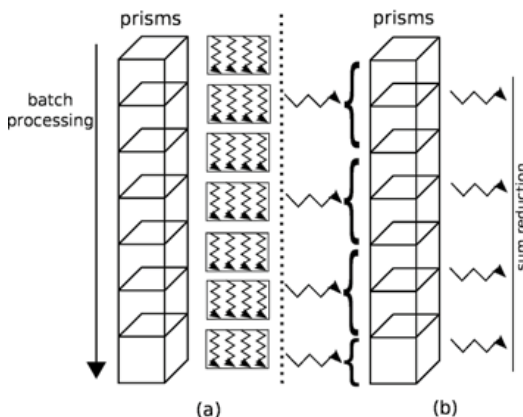


Figure 5. Parallel region behavior: (a) pseudo-code (1), (b) pseudo-code (2).

manner, therefore the partitioning of the M prisms, which composes the problem, is done automatically using a balancing algorithm included in OpenMP. In this case the decision is left to the compiler, which is optimum 99% of the cases (Zhang *et al.*, 2004).

OpenMP+MPI Implementation

One of the advantages of the prism parallelization is that it is easier to implement in MPI, producing tasks of coarse granularity using the same design previously applied in OpenMP. Having the observation grid partitioned would result in a more complicated and less efficient design using MPI. Since the parallelization in MPI is explicit, we need to manually distribute the number of prisms through a modular expression. If M is the number of prisms to calculate and p is the MPI process number (numbered from 0 to $p-1$), then for each process p we define the beginning and end of the prisms to be processed by p as p_{start} and p_{end} , respectively. We define the integer s as the quotient of the number of prisms M between the total number of processes p_n , and

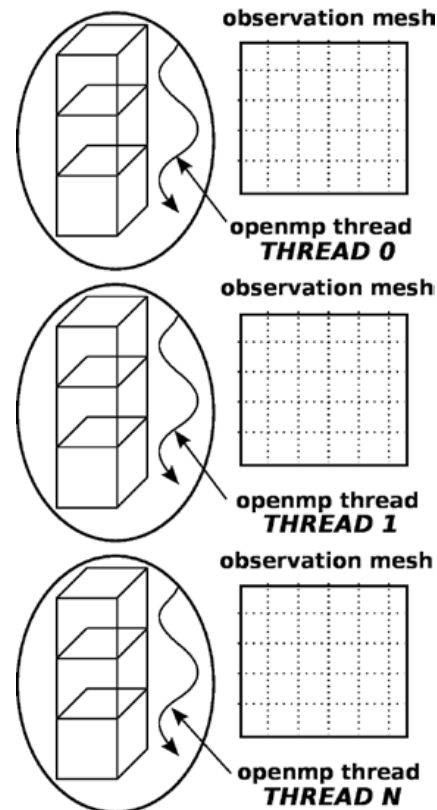


Figure 6. Partitioning by prisms.

r as the remainder, the procedure to determine p_{start} and p_{end} proceed as follows:

$$s = M/p_n, \quad (3)$$

$$r = \text{mod}(M/p_n). \quad (4)$$

Therefore

$$p_{start} = p \times s + 1 \quad (5)$$

and

$$p_{end} = (p + 1) \times s. \quad (6)$$

If $r \neq 0$ and $p < r$, then we adjust as:

$$p_{start} = p_{start} + p \quad (7)$$

and

$$p_{end} = p_{end} + (p + 1). \quad (8)$$

If $r \neq 0$ and $p \geq r$, then:

$$p_{start} = p_{start} + r \quad (9)$$

and

$$p_{end} = p_{end} + r. \quad (10)$$

This way we can distribute the number of prisms M over p_n processes in a balanced manner; once this distribution is made, we can use the OpenMP implementation in each node. In other words, we occupy MPI to distribute the number of prisms in each node, and at the same time in each node we employ OpenMP to reduce the number of MPI processes, reducing communication time.

In consequence, the application is partitioned by the number of prisms M , both in OpenMP as in MPI. Another option is to parallelize by prisms in MPI and by observation points in OpenMP. Even though this is a viable option, it is not very scalable due the drawback discussed in the previous subsection.

Basically the design consists of allocating an observation grid per execution thread and a global observation grid in the master thread per computing node, subsequently the reduction of the sum of the grids per thread is done and stored in the global grid contained in the master thread, and finally at the end of the parallel calculation, every master thread will add their grid values to update the master thread of the master node using a MPI reduction method (see Figure 7).

It is necessary to mention that the implementation of the code was made with

the FORTRAN 2003 specification, using as development tool the Intel Cluster Toolkit version 2013 of Intel Corporation.

Performance experiments

For the synthetic experiment we used a case composed by a cube of $700 \times 700 \times 50$ prisms, with 7 contrasting spheres of variable density (see Figure 8). The spheres were conformed by 251,946 prisms and an observation grid of $150 \times 100 = 15,000$ points, to an elevation of 100 m. Therefore, the number of calls to a procedure required, to calculate the vector/tensor component of the gravity are 3,779,190,000; this classifies the experiment into a high-performance computing problem.

We tested the parallelized code by observation points versus the version by prisms using OpenMP. The first parallel scheme is technically easier to implement because for each one of the prisms the calculation of the cycles corresponding to the tracking of the observation grid is parallelized. The second scheme has a more complex implementation because it requires different space memory allocations. The performance experiments that calculate the components of the gravimetric

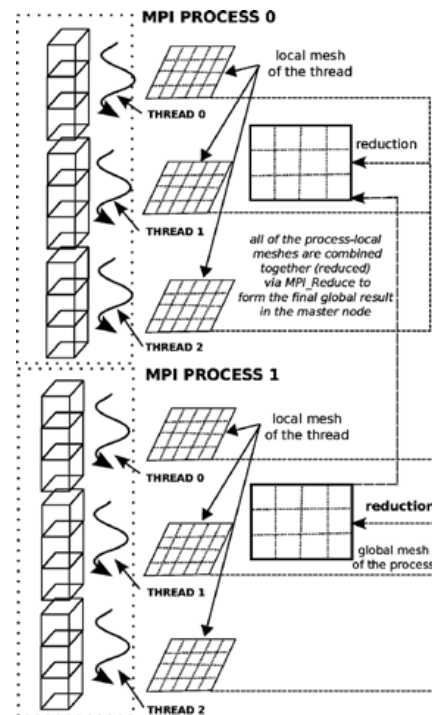


Figure 7. OpenMP+MPI design.

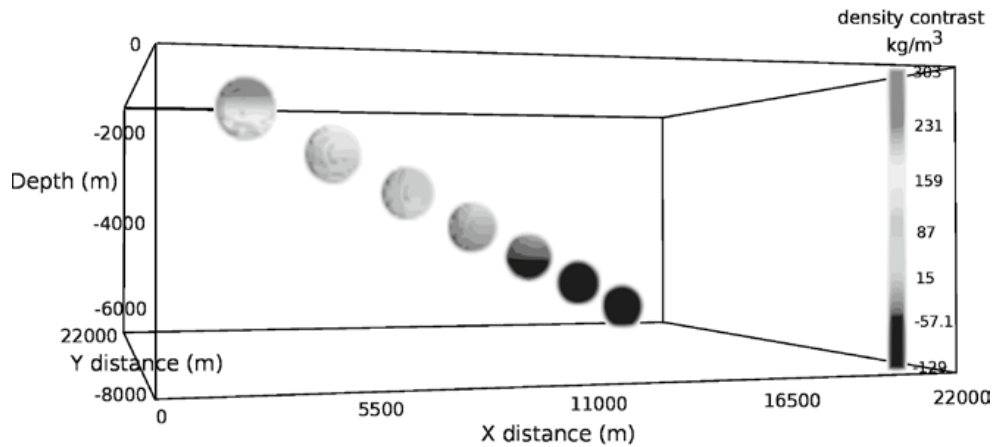


Figure 8. Synthetic problem setup with 7 spheres of variable density contrast (not scaled). Ensemble size of 22 km x 22km x 8km, 251,946 prisms conform the spheres.

tensor $G_{xx}, G_{yy}, G_{zz}, G_{xy}, G_{xz}, G_{yz}$ using both versions were carried out in the server described below. We did not include the performance analysis for the vectorial components G_x, G_y and G_z , since its behavior is very similar.

The characteristics of the server where the tests took place with OpenMP are as follows:

- 4 Xeon Intel (R) Xeon (R) E7-4850 Processors
- 10 processing cores per processor
- Hyperthreading Technology deactivated
- 512 GB of RAM memory
- Red Hat 6.3 as operating system

To interfere as least as possible with the processes of the operating system, we used 35 of the 40 cores available in the server. Initially

we can say that the prisms implementation and with independent memory per core was 3.22X faster than its counterpart of observation points. Therefore, while the observation points version uses 757 s, the version partitioned by prisms only consumes 235 s.

The comparison of the computing times per thread in the partition by prisms against the partition by observation points is shown in Figure 9.

In Figure 9 it can be seen that the performance behavior is kept stable in both types of partitioning; however, by prisms the best reduction in time is obtained. To prove that the partitioning by prisms keeps reduction time practically linear, we graphed the *speed-up* of the performance by prisms.

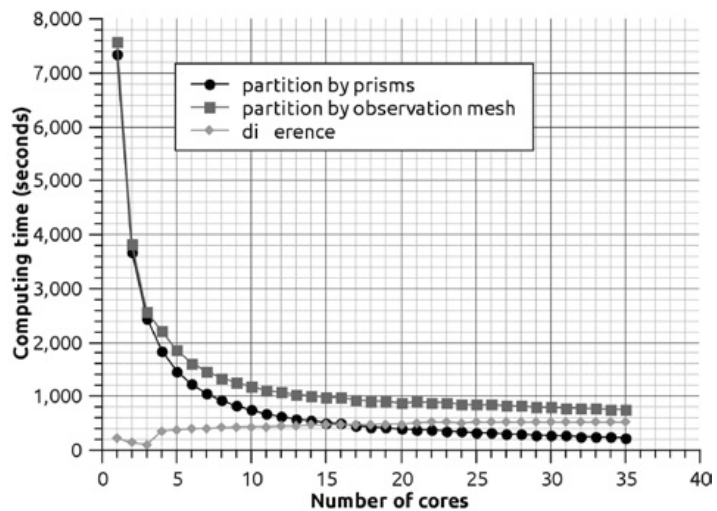


Figure 9. Comparison between execution time used between the partition by prisms against the partition by observation points (one thread per core).

For the *speed-up* shown in Figure 10, we considered a serial fraction of 5% ($f=0.05$). In this fraction the necessary reductions to sum the grid points for each core are contemplated, the total result of the anomaly is calculated as:

$$O_f(i,j) = \sum_{j=1}^{N_i} O_i(i,j) \quad (11)$$

where, for each (i,j) O_j is the final observation, O_i is the calculated grid by core t and N_i is the total number of cores. Therefore, we considered that 95% of the code is parallel, and according to Gustafson's law, the maximum textitspeed-up that can be obtained with 35 processing units, in this case cores, is $35 + (1-35) \times (0.05) = 33.30$. The experimentally obtained

speed-up result was 31.31, which represents an absolute difference of 1.99 and a relative difference of 0.06, which shows the efficiency of the implementation.

Another indicator which must be contemplated is the efficiency E , defined as:

$$E = \frac{S(n)}{n} \times 100\% \quad (12)$$

where $S(n)$ is the obtained *speed-up* with n tasks, and indicates how busy the processors or cores are during execution. Figure 11 shows that the efficiency by prisms is high since on average every processing core is kept busy 94% of the time. The efficiency E also indicates

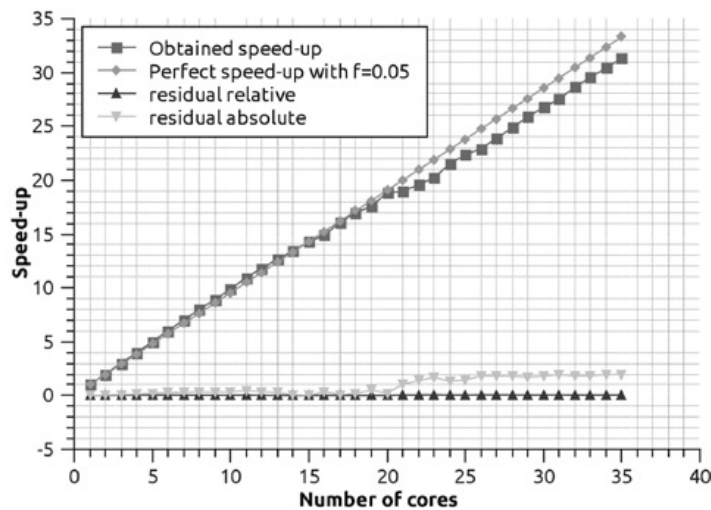


Figure 10. *Speed-up* of the partitioning by prisms (one thread per core).

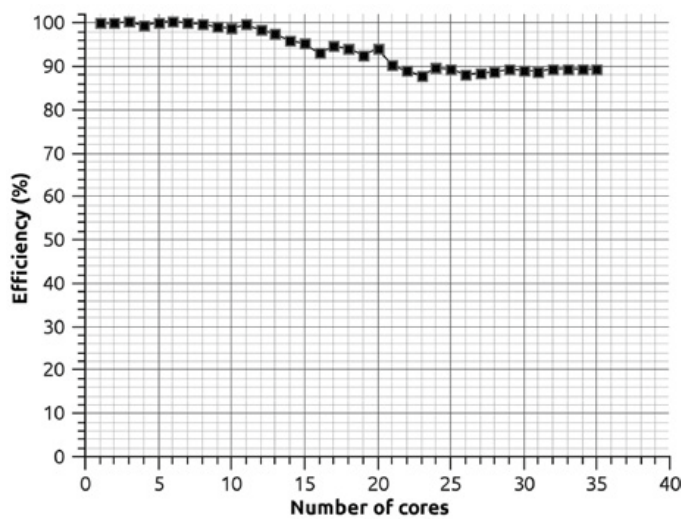


Figure 11. Efficiency of the partitioning by prisms.

that the partitioning by prisms is scalable, which means that we can increase the number of processors to improve time reduction while not losing efficiency in the use of many cores. The scalability must be contemplated as a good design of the parallel program since it allows scaling the algorithm, so we could expect when the number of processing units is increasing the performance is not affected.

The design using OpenMP is limited to architectures of machines of shared memory, therefore we are now making experiments using a hybrid machine commonly known as *cluster*, mixing OpenMP+MPI with the methodology described in subsection 2.2.

The characteristics of the cluster where the numerical experiments were carried out are as follows:

- Node: Intel(R) Xeon(R) model X5550 processors with four physical cores processor.
- 44 processing nodes
- Hyperthreading Technology enabled
- 40 GB of RAM memory per node
- Red Hat 6.3 as operating system
- InfiniBand 300Gbps

We started by evaluating the performance of each cluster node, as opposed to the experiments done with the 40 cores server, where hyperthreading technology (HT) was disabled. In this case HT is enabled, so each

node reports the handling of 8 execution threads instead of 4, but we only have 4 physical floating point units (FPUs). Since our program is computationally intensive, we have to find out if we benefit from the use of HT; some studies have reported the use of HT in numerical applications can modify the performance by 30% (Curtis-Maury *et al.*, 2008).

The behavior obtained using one node containing 1 processor with four real cores with HT enabled/disabled can be exposed by an analysis of the computing time graph, shown in Figure 12, the problems analyzed is setup with 13,997 prism conforming a sphere with a mesh of 150×100 observations points.

As can be observed, the best run-time performance that we can obtain from the processor in HT mode is not produced with 4 execution threads, the best performance is obtained with 8 threads, but the time is not doubly improved. This occurs since two threads share the same FPU and the HT technology is designed to quickly switch between threads, and therefore there is not a double improvement in time but the performance gain is approximately 30%, which means that the two threads make better use of the FPU, therefore is necessary to create two threads per core to obtain the maximum performance when the HT is enabled. When the HT is disabled we have an asymptotic behavior after 4 threads but did not reach the performance obtained using the HT mode.

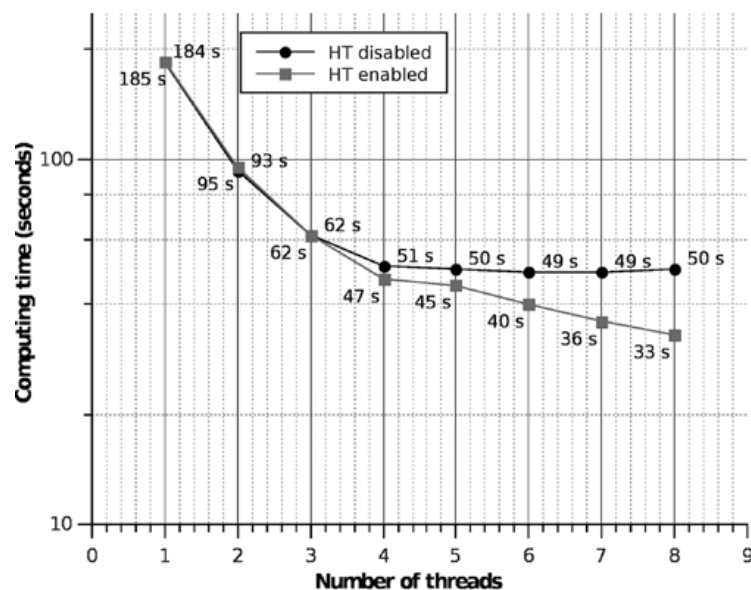


Figure 12. Computing time using only one node with HT enabled/disabled, calculating a problem of 13,997 prisms with 10,000 observation points.

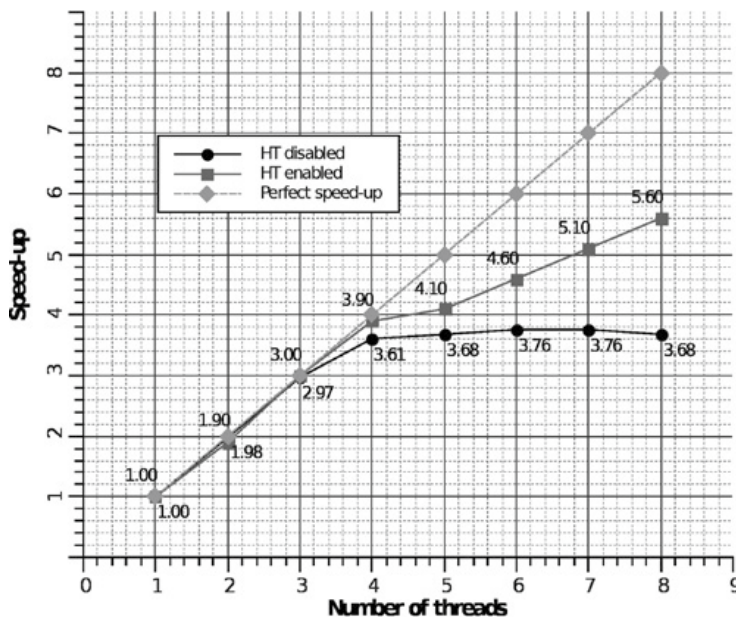


Figure 13. Speed-up using only one node with HT enabled/disabled corresponding to the execution times shown in the Figure 11.

In Figure 13 it can be observed that when HT technology is enabled we obtain a linear speed-up up to 4 execution threads; this is obvious since there are only 4 physical CPUs. Nevertheless, with the HT we can have a better use of the CPUs improving the speed-up up to 5.60, this is, 1.6 more processing units. With the HT disabled, a similar performance is observed up to 4 threads, although this performance is below the one with the HT enabled. For more than 4 threads, the performance with the HT disabled begins to decrease.

The efficiency corresponding to the speed-up shown in Figure 13 is graphed in Figure 14; notice how HT is able to increase the efficiency of some intensive floating point applications up to 30% when the number of threads equals the number of physical cores. Of course, the best efficiency is obtained with 4 threads because we have 4 CPUs, nevertheless we can get a better performance creating 4 threads more using the additional virtual processors created by the HT.

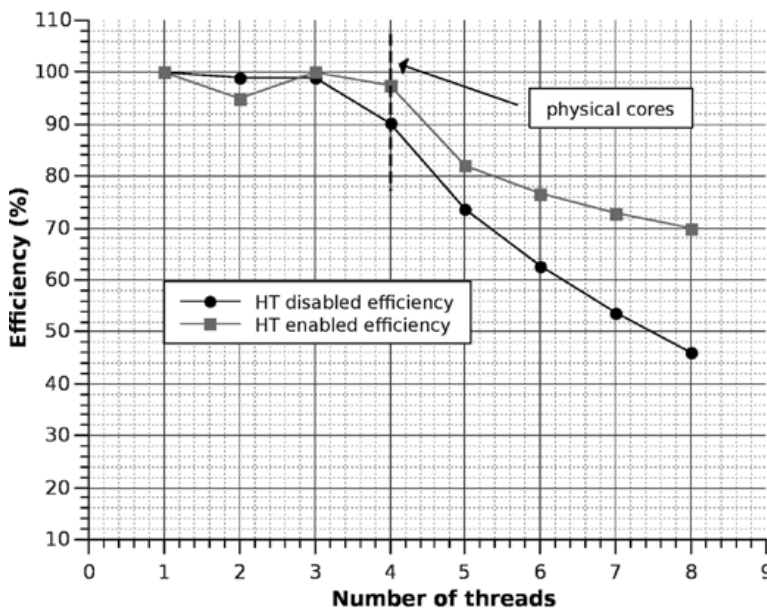


Figure 14. Efficiency using only one node with 4 cores with the HT enabled/disabled for the problem of 13,997 prisms.

To analyze the performance in a node with the original problem (shown in Figure 8), we added a processor in the second socket to one of the nodes. In other words, we created a node with eight real cores to compare it against a node with four real cores with HT enabled. The results of execution time are shown in Figure 15.

It must be taken in consideration our cluster nodes are composed of a single processor with HT enabled, we only added another processor in the second socket to a node for experimental purposes. To have a better perspective of the

performance, we determined the *speed-up* through both node configurations we showed in Figure 16. A nearly perfect *speed-up* can be observed for the node with 8 real cores, but a increase of 1.8 processing units for the node with 4 real cores with HT enabled. Evidently, if we enable HT in the machine with 8 real cores we would have 16 reported processors, and to get its maximum performance we would have to create 16 threads. However, the experimentation with 8 real cores was only for comparison purposes, since the cluster configuration is made of one node with 4 real

Figure 15. Eight real cores with HT disabled vs four real cores with HT enabled for the problem setup in the Figure 8.

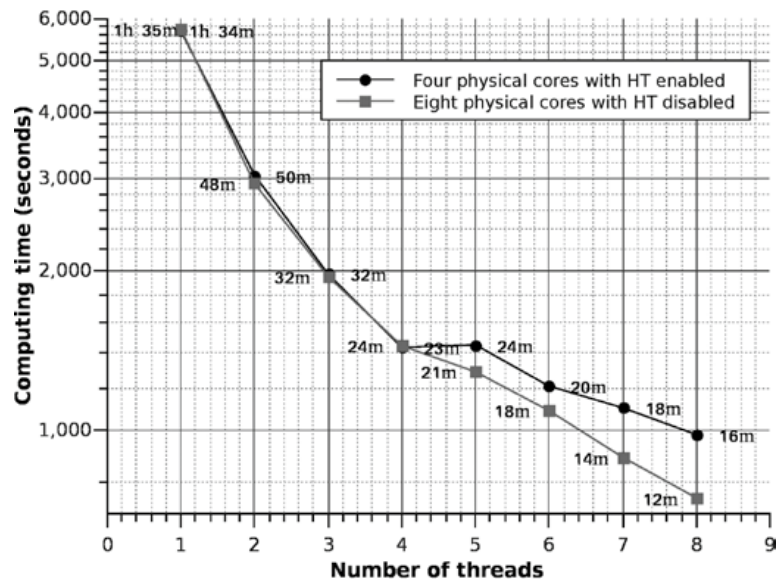
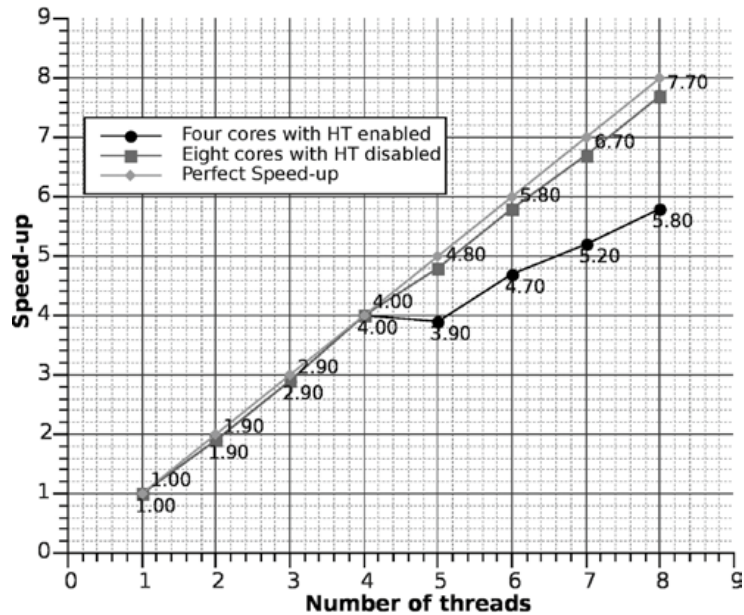


Figure 16. Eight real cores with the HT disabled vs four real cores with HT enabled for the problem setup in the Figure 8.



cores with HT enabled. It can also be observed that each node of the cluster reduces the time by a factor of 5,8X against the serial version.

Once it is known that the best node performance is achieved with 8 execution threads for a node with 4 real cores with HT enabled and with the partition by prisms, we can consider each node as a processing unit and distribute the computing with MPI, obtaining a code with a hybrid programming model.

The *speed-up* results using 25 cluster nodes are displayed in Figure 17; a serial fraction of 5% ($f = 0.05$) is considered since in MPI there needs to be reductions in the sum for each node. The results show that a nearly perfect *speed-up* is obtained up to 22 nodes. From this point on, the speed-up starts declining because the application performance is affected by the communication time between nodes. In other words, the granularity of the tasks begins to decrease for this problem of 249,946 prisms

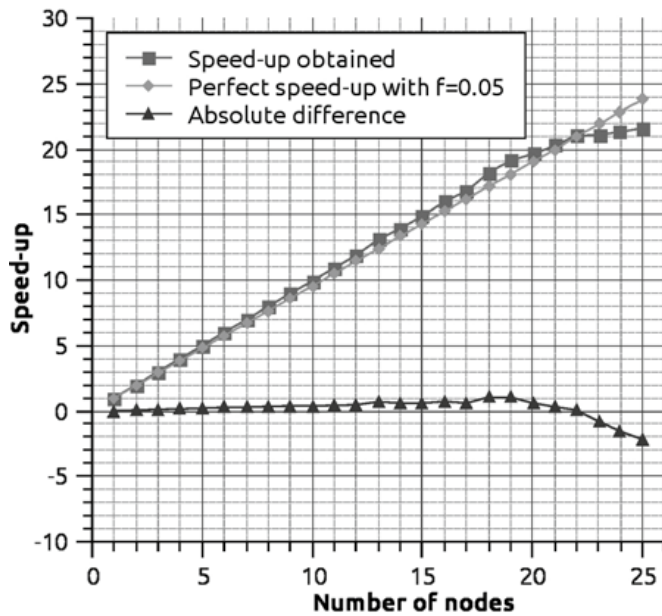


Figure 17. *Speed-up* obtained using 25 cluster nodes.

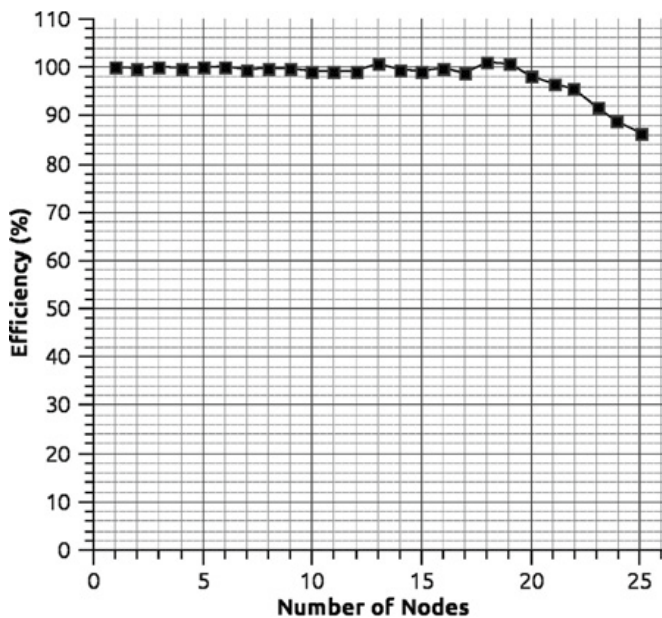


Figure 18. Cluster efficiency using 25 nodes.

for 30 nodes. This implies that by increasing the granularity of the problem (increasing the number of prisms), the *speed-up* is also increased until it becomes stable, to decrease again later on.

The efficiency graph related with the *speed-up* of Figure 16 is shown in Figure 18. Notice how the efficiency is below 90% after node 23. If we consider that we have an increase in speed 5.8 times per node (from Figure 15) with respect to the serial version, then the optimum speed factor for this cluster (for a problem of 251,946 prisms) is approximately $5.8 \times 22 = 127.6X$, i.e. 127 times faster than the serial version. Obviously, as previously stated, if we increase the granularity (number of prisms), the efficiency increases as well. In fact, we reduce the computation time of the spheres problem from 1 h 34 m 56 s to 34 s

Comparison with similar programs

To provide a better perspective of the obtained performance with the parallel implementation of our code, we compared against an open source code called *tesseractoids* (Uieda *et al.*, 2011), which can be downloaded from <http://dx.doi.org/10.6084/m9.figshare.786514>. We chose the problem of 13,997 prisms which form an sphere against 10,000 observation points, since *tesseractoids* is not distributed (can not be executed on a cluster) and can only accelerate the computation in shared memory machines. The execution times are shown using the bar chart in Figure 19, where it can be observed that with HT disabled we have

a speed improvement of 2.14X and with HT enabled of 2.51X with respect to *tesseractoids*. This performance improvement is due to our program design takes a better advantage of the processor technology and keeps the cores occupied to the maximum by using a prisms parallelization scheme based on different memory allocations. This can be observed in the CPU history graph shown in the Figure 20.

Numerical code validation

The main challenge of the parallel programming is to decompose the program into components which can be simultaneously executed to reduce computing time. The decomposition level is highly influenced by the type of architecture of the parallel machine. In this case the design was made with a hybrid programming strategy to get the maximum out of the architecture. Although the reduction of the execution time is the main objective of the parallel programming, the validation of the code is a topic that should be covered since inherent parallelism programming errors can occur.

To measure the error, we compared the previously validated sequential counterpart in the synthetic experiment with the analytical solution. We used the L2 norm error or RMS (Mickus and Hinojosa, 2001; Menke, 1989), defined as:

$$e = \sqrt{\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |g_{i,j}^p - g_{i,j}^s|^2}, \quad (13)$$

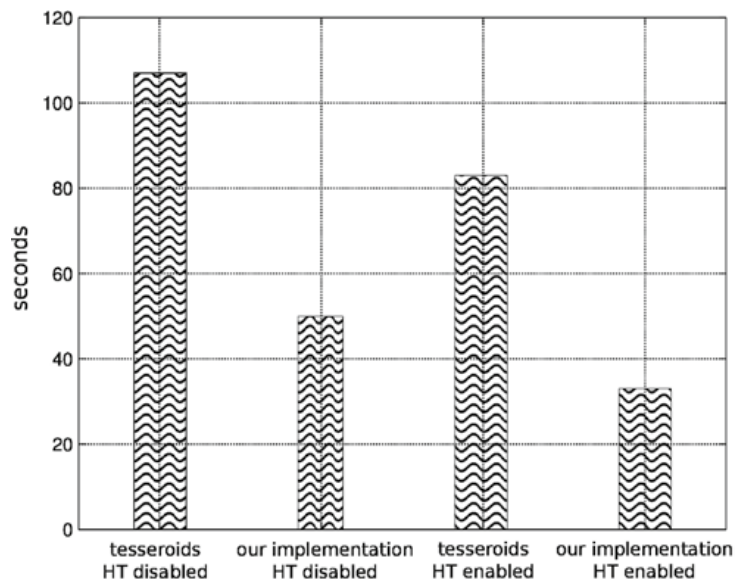


Figure 19. Computing time of our implementation vs tesseractoids on one node.

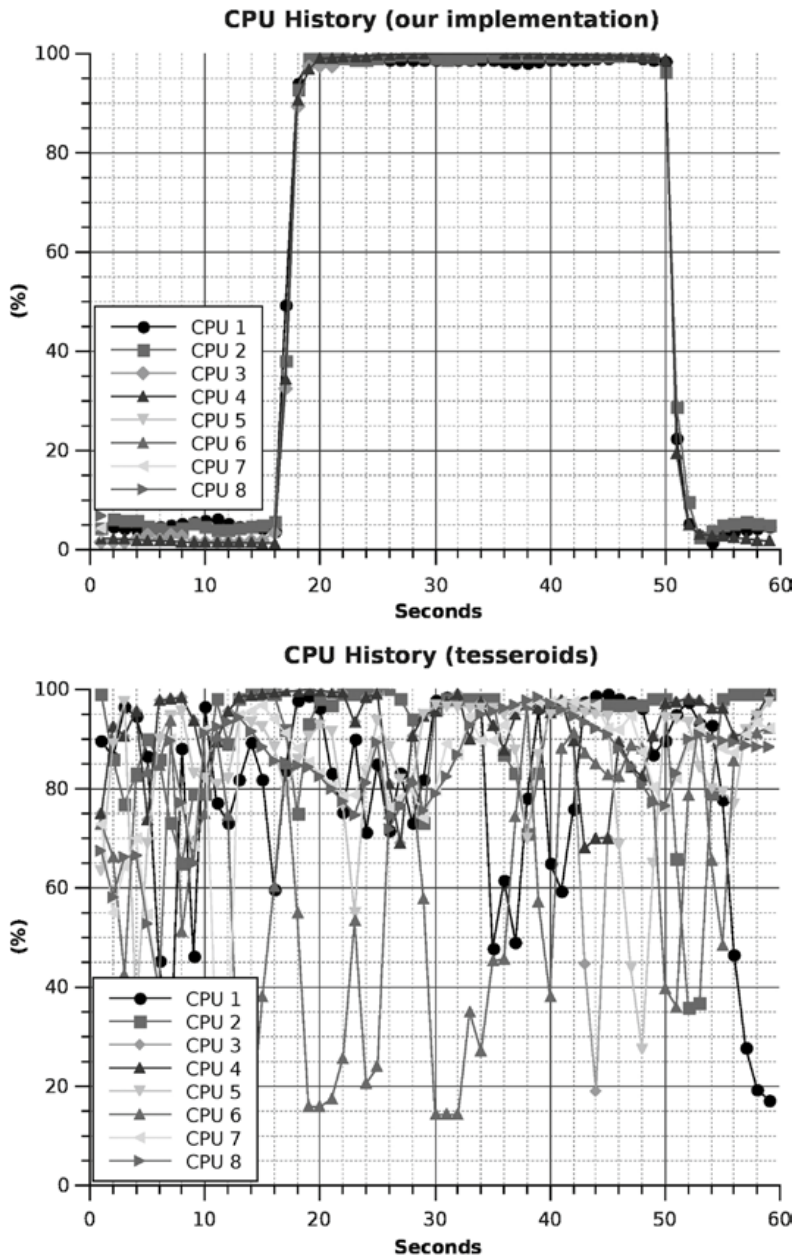


Figure 20. Behavior of the CPU utilization produced by our implementation vs tesseractoids. The HT is enabled and note how the cores are used to maximum efficiency in our implementation compared with the partial use of tesseractoids.

where $g_{i,j}^p$ is the tensor component, parallelly computed, and $g_{i,j}^s$ is the serially calculated component.

In Table 1 the errors of the gravimetric tensor components are shown, parallelly calculated with respect to the serial form.

From the errors obtained it can be noticed that there is no numerical difference, therefore the parallel version is correctly implemented.

Table 1. Errors of the tensor components with respect to its sequential counterpart.

Gravity gradient tensor components	Error L2
G_{xx}	6.3136e-12
G_{yy}	6.3054e-12
G_{zz}	2.8367e-12
G_{xy}	1.0244e-14
G_{xz}	1.5518e-14
G_{yz}	1.5581e-14

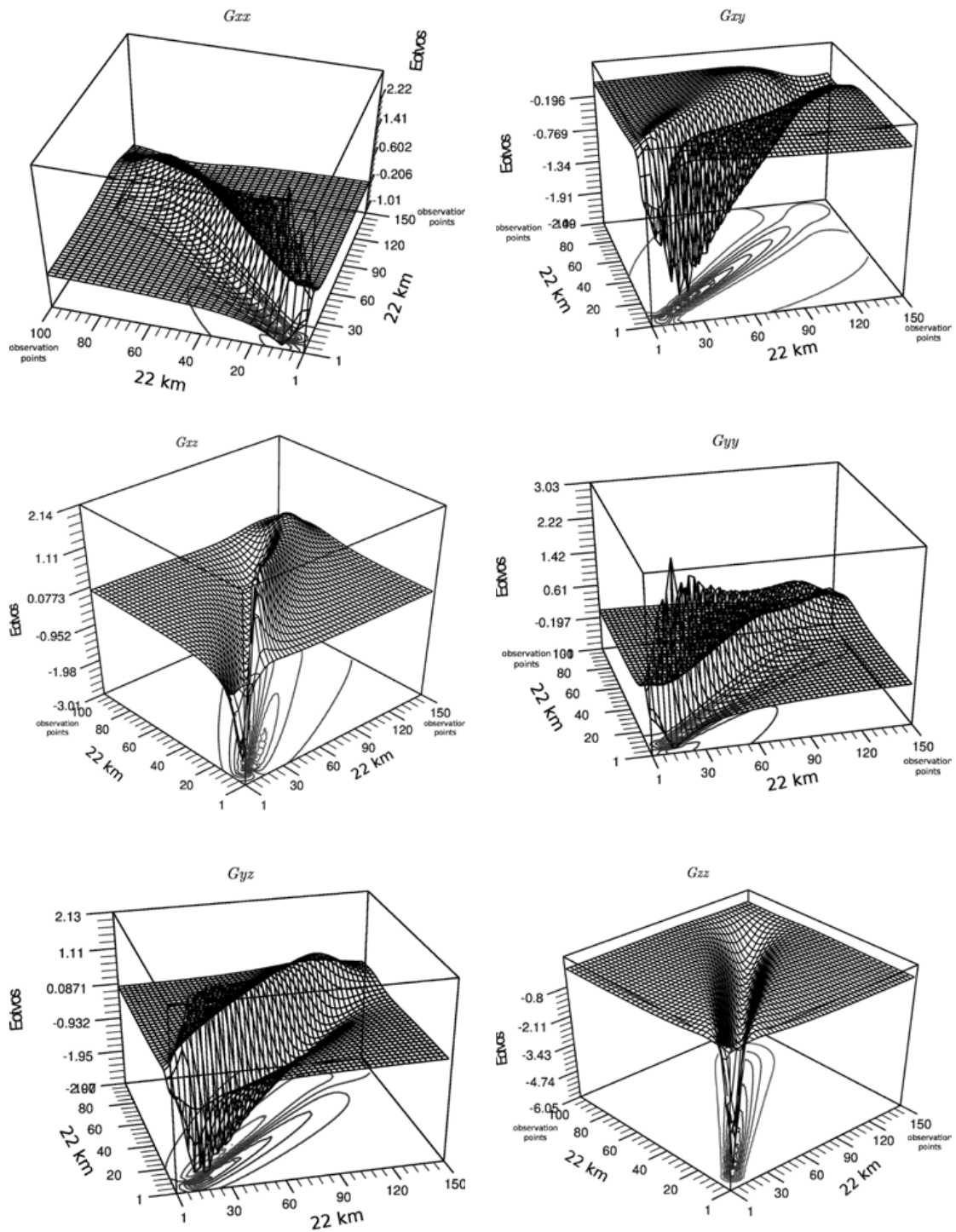


Figure 21. Behavior of the CPU utilization produced by our implementation vs tesseroids. The HT is enabled and note how the cores are used to maximum efficiency in our implementation compared with the partial use of tesseroids.

The surface graphs of the gravitational fields are shown in Figure 21. These graphs correspond to the components of the gravimetric tensor, calculated for the synthetic case studied in Figure 8.

Conclusions

A parallel design for the calculation of the vectorial and tensorial components of the gravity anomaly was implemented and validated using a hybrid methodology with OpenMP and MPI. The numerical experiments and the obtained indicators validate that the implementation is very efficient and that it also yields good results with respect to the numerical solution.

We show that using the simplest or most trivial parallelization form does not contribute to the attainment of the best performance or the greatest exploitation of the platform. For our case, even though the partitioning by prisms requires a greater investment in the design and implementation, it was the most advantageous with respect to performance.

The HT technology could improve some numerical intensive applications up to 30%, nevertheless, to get the best performance it is necessary to create two threads per core when the HT is enabled.

We also conclude that this design can serve as a benchmark for solving problems which require the parallelization of schemes where the decomposition of the domain is not trivial or is shared by the processing units, as is the case of the observation grid. Finally the correct exploitation of OpenMP and MPI, jointly, can become a fundamental tool for parallel programming in clusters.

Future work

As future work we pretend to implement the code in CUDA NVIDIA with TESLA technology and compare these results with the cluster performance results presented in this paper, as the measurement of the error introduced by CUDA in single and double precision. The implementation in CUDA is a work of interest since the reduction of the variable values in CUDA technology is very complicated when used in shared form, as is the case with the observation grid.

Appendix A. Calculation of gravitational quantities

The Earth's gravitational potential G is a scalar quantity, its shape can be constrained by its slope in the x , y and z directions, called the gravitational attraction G_x , G_y and G_z (gravity vector field). In this work, we have investigated how to parallelize the analytical calculation of the components of the gravity field vector and the gravity gradients represented by a nine component tensor, because of the symmetrical or irrotational attribute, the gravity gradient tensor is reduced to only six independent components: G_{xx} , (the vertical gravity gradient), and For the right rectangular prism model, the analytical formulae for the three components vectors and the six gravity gradient components, corresponding to the Eq. (2) are given by:

$$G_x = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times [y_j \ln(z_k + r_{ijk}) + z_k \ln(y_j + r_{ijk}) - x_i \arctan \frac{y_j z_k}{x_i r_{ijk}}]$$

$$G_y = -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times [z_k \ln(x_i + r_{ijk}) + x_i \ln(z_k + r_{ijk}) - y_j \arctan \frac{z_k x_i}{y_j r_{ijk}}]$$

$$G_z = -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times [z_i \ln(x_i + r_{ijk}) + x_i \ln(z_k + r_{ijk}) - y_j \arctan \frac{z_k x_i}{y_j r_{ijk}}]$$

$$G_{xx} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{y_j z_k}{x_i r_{ijk}}$$

$$G_{yy} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{x_i z_k}{y_j r_{ijk}}$$

$$G_{zz} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{x_i y_j}{z_k r_{ijk}}$$

$$G_{xy} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(z_k + r_{ijk})$$

$$G_{xz} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(y_k + r_{ijk})$$

$$G_{yz} = \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(x_i + r_{ijk})$$

Acknowledgment

The authors thank the support provided by the Mexican Institute of Petroleum (IMP, www.imp.mx) in allowing access to its computing equipment, as well as the financial support through project Y.00107, jointly created by IMP-SENER-CONACYT number 128376. Also, we would like to express our gratitude to the two anonymous reviewers for their helpful comments.

References

- Amritkar A., Tafti D., Liu R., Kufirin R., Chapman B., 2012, OpenMP parallelism for fluid and fluid-particulate systems. *Parallel Computing*, 38, 9, 501-517.
- Boehmer S., Cramer T., Hafner M., Lange E., Bischof C., Hameyer K., 2012, Numerical simulation of electrical machines by means of a hybrid parallelisation using MPI and OpenMP for finite-element method. *Science, Measurement & Technology*, IET, 6, 5, 339-343.
- Brunst H., Mohr B., 2008, Performance analysis of large-scale OpenMP and hybrid MPI/OpenMP applications with Vampir NG. In OpenMP Shared Memory Parallel Programming (pp. 5-14). Springer Berlin Heidelberg.
- Calvin C., Ye F., Petiton S., 2013, October, The Exploration of Pervasive and Fine-Grained Parallel Model Applied on Intel Xeon Phi Coprocessor. In P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on (pp. 166-173). IEEE.
- Carrillo-Ledesma A., Herrera I., de la Cruz L.M., 2013, Parallel algorithms for computational models of geophysical systems. *Geofísica Internacional*, 52, 3, 293-309.
- Couder-Castañeda C., 2010. Simulation of supersonic flow in an ejector diffuser using the jsvm. *Journal of Applied Mathematics*, 2009.
- Curtis-Maury M., Ding X., Antonopoulos C.D., Nikolopoulos D.S., 2008, An evaluation of OpenMP on current and emerging multithreaded/multicore processors. In OpenMP Shared Memory Parallel Programming (pp. 133-144). Springer Berlin Heidelberg.
- Dagum L., Menon R., 1998, OpenMP: an industry standard API for shared-memory programming. *Computational Science & Engineering*, IEEE, 5(1), 46-55.
- Foster I., 1995, Designing and building parallel programs (pp. 83-135). Addison Wesley Publishing Company.
- Gonzalez B., Donate J.P., Cortez P., Sánchez G., De Miguel A., 2012, May, Parallelization of an evolving Artificial Neural Networks system to Forecast Time Series using OPENMP and MPI. In Evolving and Adaptive Intelligent Systems (EAIS), 2012 IEEE Conference on (pp. 186-191). IEEE.
- Heck B., Seitz K., 2007, A comparison of the tesseract, prism and point-mass approaches for mass reductions in gravity field modelling. *Journal of Geodesy*, 81, 2, 121-136.
- Jagannathan S., Donzis D.A., 2012, July, Massively parallel direct numerical simulations of forced compressible turbulence: a hybrid MPI/OpenMP approach. In Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond (p. 23). ACM.
- Krpic Z., Martinovic G., Crnkovic I., 2012, May). Green HPC: MPI vs. OpenMP on a shared memory system. In MIPRO, 2012 Proceedings of the 35th International Convention (pp. 246-250). IEEE.
- Korneye L., 2012, May, Parallel implementation of a combustion chamber simulation with MPI-OpenMP hybrid techniques. In MIPRO, 2012 Proceedings of the 35th International Convention (pp. 356-361). IEEE.
- Menke W., 2012, Geophysical data analysis: discrete inverse theory. Academic press.
- Mickus K.L., Hinojosa J.H., 2001, The complete gravity gradient tensor derived from the vertical component of gravity: a Fourier transform technique. *Journal of Applied Geophysics*, 46, 3, 159-174.
- Mitin I., Kalinkin A., Laevsky Y., 2012, A parallel iterative solver for positive-definite systems with hybrid MPI-OpenMP parallelization for multi-core clusters. *Journal of Computational Science*, 3, 6, 463-468.

C. Couder-Castañeda, J. Carlos Ortiz-Alemán, M. G. Orozco-del-Castillo and M. Nava-Flores

Nagy D., Papp G., Benedek J., 2000, The gravitational potential and its derivatives for the prism. *Journal of Geodesy*, 74(7-8), 552-560.

Sariyuce A.E., Saule E., Catalyurek U.V., 2012, May, Scalable hybrid implementation of graph coloring using mpi and openmp. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 2012 IEEE 26th International (pp. 1744-1753). IEEE.

Smith L.A., 2000, Mixed mode MPI/OpenMP programming. *UK High-End Computing Technology Report*, 1-25.

Uieda L., Bomfim E., Braitenberg C., Molina E., 2011, July, Optimal forward calculation method of the Marussi tensor due to a geologic structure at GOCE height. In *Proceedings of GOCE User Workshop 2011*.

Zhang Y., Burcea M., Cheng V., Ho R., Voss M., 2004, September, An Adaptive OpenMP Loop Scheduler for Hyperthreaded SMPs. In *ISCA PDCS* (pp. 256-263).

Research Article

TESLA GPUs versus MPI with OpenMP for the Forward Modeling of Gravity and Gravity Gradient of Large Prisms Ensemble

Carlos Couder-Castañeda,¹ Carlos Ortiz-Alemán,¹
Mauricio Gabriel Orozco-del-Castillo,¹ and Mauricio Nava-Flores²

¹ Mexican Petroleum Institute, Eje Central Lázaro Cárdenas 152, Colonia San Bartolo Atepehuacan, 07730 México, DF, Mexico

² División de Ingeniería en Ciencias de la Tierra, Facultad de Ingeniería, Universidad Nacional Autónoma de México, Circuito Interior S/N, Colonia Ciudad Universitaria, 04510 México, DF, Mexico

Correspondence should be addressed to Carlos Couder-Castañeda; ccouder@imp.mx

Received 28 May 2013; Revised 16 September 2013; Accepted 17 September 2013

Academic Editor: Luca Formaggia

Copyright © 2013 Carlos Couder-Castañeda et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An implementation with the CUDA technology in a single and in several graphics processing units (GPUs) is presented for the calculation of the forward modeling of gravitational fields from a tridimensional volumetric ensemble composed by unitary prisms of constant density. We compared the performance results obtained with the GPUs against a previous version coded in OpenMP with MPI, and we analyzed the results on both platforms. Today, the use of GPUs represents a breakthrough in parallel computing, which has led to the development of several applications with various applications. Nevertheless, in some applications the decomposition of the tasks is not trivial, as can be appreciated in this paper. Unlike a trivial decomposition of the domain, we proposed to decompose the problem by sets of prisms and use different memory spaces per processing CUDA core, avoiding the performance decay as a result of the constant calls to kernels functions which would be needed in a parallelization by observations points. The design and implementation created are the main contributions of this work, because the parallelization scheme implemented is not trivial. The performance results obtained are comparable to those of a small processing cluster.

1. Introduction

In recent years the number of publications about parallel computing applications using the GPUs architecture has remarkably increased. These applications represent an economic and powerful way to access high-performance computing [1, 2]. However, since the architecture of the GPU is different to that of a conventional CPU, the programming paradigm should be changed. This had led to the development of a new research field within scientific computing which explores the performance of the GPU to general purpose applications, such as acoustic simulation [3], propagation of seismic waves [4], seismic migration [5], molecular engineering [6], fluid dynamics [7], even for astrophysical simulations [8] and many other implementations. In a few

words, the objective of the general purpose computing in GPU (GPGPU) is to develop new applications for those who pretend to solve problems of numerical simulation requiring as less computing time as possible.

Even though the GPUs have become an accessible platform for general purpose programming, they still have some limitations and its programming entails some difficulties [2]. Compute unified device architecture (CUDA) is a set of tools that includes mainly a compiler for an extension of the C language, a set of libraries, and drivers for the specific programming of NVIDIA cards. Despite that these tools have eased the programming, it is still needed to know with precision the architecture of the card with its several memory levels to obtain the maximum performance. One of the greatest drawbacks that can occur in CUDA is the

handling of critical sections or shared memory, since there are no proper instructions of exclusions, as is the case in OpenMP [13]. Therefore, when these difficulties are present in the application design, it is necessary to modify the strategy to achieve a good performance. Implementing this application in CUDA for the calculation of the direct model through a prisms ensemble represents a big challenge; this is because the memory region where the calculations are made (observation grid) is shared and therefore requires different memory allocations to avoid the data coherency problems produced when two or more processing cores access the same memory location at the same time. This drawback is not present, for example, when a model is solved through finite differences since the domain is divided between the cores and there is no overlap in the data handled by each core [9].

One of the simplest parallelization options for this problem using CUDA consists of partitioning by the number of prisms and to consequently divide the domain of the observation grid between the device cores; this approach avoids the handling of shared regions. However, this design is extremely inefficient since for each prism it would be necessary to make a call to the GPU. A typical problem of fourteen million prisms would imply the execution of a kernel by the same number, and each kernel call is computationally expensive [11]. Therefore, we propose an efficient design based on the partition by groups of prisms.

Additionally, we make experiments with double and single precision to calculate the errors that can introduce the single precision, and even when its use reduces the computing time from 30% to 50%, it is necessary to evaluate the effect of the introduced error by using only seven significant digits in the floating numbers and investigate if this error affects considerably the result of the modeling. This analysis is necessary since recently NVIDIA introduced TESLA K10 cards which handle single precision and are more economical than the TESLA K20 cards for double precision.

1.1. Related Work. Some related research works which implement an approach to calculate scalar and tensor gravity utilizing the massively parallel architecture of GPU can be found in [12], in which a parametrization based on rectilinear blocks with constant density within each block is used; however, the results show that our design yields a better performance using different memory allocations. Also a parallel program was developed to estimate the correlation imaging for gravity and gravity gradiometry data to provide a rapid approach to equivalent estimation of objective bodies with different density contrasts in the subsurface [17]; however, neither is multi-GPU implementation.

1.2. Paper Organization. This paper is organized as follows: in Section 2 we present the characteristics of the CUDA platform and the tools we used, in Section 3 the application design is explained, in Section 4 we present some numerical experiments that were made, in Section 5 we detail the validation of the code for double as well as single precision, and finally in Section 6 we compare against a 29-nodes cluster and finally we present our conclusions.

2. Architecture of the Platform

As a general-purpose architecture, CUDA includes the hardware that can have dedicated processing cards or cards which also control the visualization of the monitor and the software that includes the compiler, the card drivers, and the libraries. The programming model in CUDA consists of functions called kernels which are executed concurrently by several light threads (CUDA threads). These threads are grouped into blocks which can be of one, two, or even three dimensions. Each block can contain a maximum number of threads, defined by the architecture of the card which is being used. The blocks are executed concurrently by the stream multiprocessors (SMs) and the execution order is nondeterministic. Each SM contains a set of microprocessors which can be thought of as arithmetic logic units (ALUs) and are known as CUDA cores. The threads within a block are divided into groups of 32, called warps. A warp is executed concurrently by the CUDA cores, and the number of cores can be less than the size of the warp, as happens in the TESLA C1060 card, which has eight cores per SM but supports one warp. This configuration implies that 8 threads are executed in parallel, but 32 are processed concurrently, and this means that each core will process 4 threads previously assigned.

There is an implicit synchronization between kernel calls; that is, the next kernel cannot be executed until the previous one has finalized. There are some cards of a more advanced architecture which allow the concurrent execution of kernels, but this must be specified by the programmer. The threads within the same block can be synchronized, but synchronization between blocks cannot be achieved.

Understanding the different types and hierarchies of memory of the NVIDIA cards is essential to be able to take advantage of them. There are four types of memory: global, constant, texture, and shared. The global memory is analogous to the RAM memory used in a CPU. A CUDA application requires several data transfers from the global memory of the GPU to the CPU memory. The constant and the texture memories are cache memories and read-only by the SM. The content of the texture memory can be updated through special functions. The shared memory is included in each block of threads and is shared only by the threads in the block and is extremely fast in comparison with the global memory, but its deficiency lies in the fact that it is very limited and its size is defined by the architecture.

The key, in general, to achieve an efficient code for GPU is to correctly handle the access times (latency) to the memory, that is, to carry out the least possible data transfers between the global memory of the GPU and the principal memory of the CPU [15], followed by few calls to the kernel functions. This implies having a great amount of blocks to process or having blocks with a great amount of threads. It is also necessary to avoid an excessive read-write access to the global memory, and preferable to use the shared memory, even though a lot of times this is not possible since a great amount of data is being handled and the shared memory becomes insufficient.

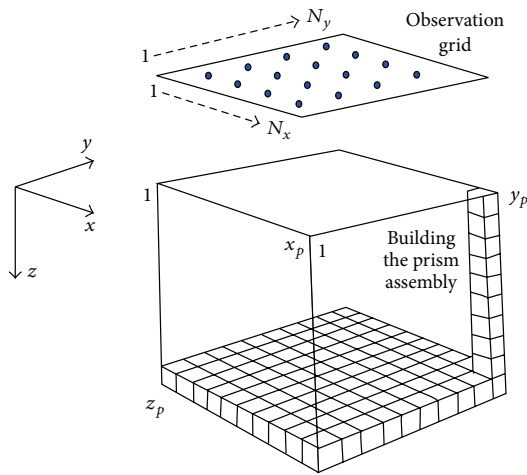


FIGURE 1: Construction of a prism of densities and its calculation with respect to an observation grid.

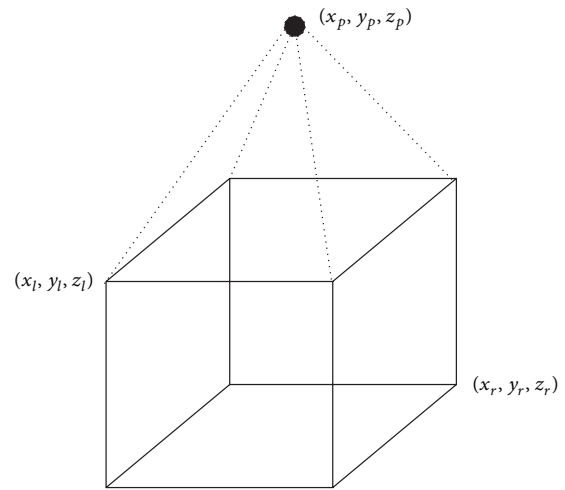


FIGURE 2: Illustration of the calculation of the anomaly produced by a prism with respect to an observation point.

3. Design of the Application for GPU

The application consists of calculating the gravimetric or gradiometric response produced by a rectangular prismatic body with constant density, in reference to a set of points called observation points (see Figure 1) [16]. The set of prisms is known as prisms ensemble and is not necessarily regular. An ensemble of nonregular prisms can be configured (Figure 3), with the only requisite being that they are not superimposed. Since the gravitational field complies with the superposition principle with respect to the observation, if f is the calculated response at a point (x, y) , then the observed response at the point $f(x, y)$ is given by

$$f(x, y) = \sum_{k=1}^M G(\rho_k, x, y), \quad (1)$$

where M is the total number of prisms and ρ the density of the prism.

It is well known that the function which can calculate the gravimetric or gradiometric contribution for a given prism and a set point can be rewritten as follows:

$$g = f(x_1, y_1, z_1, x_2, y_2, z_2, x, y, z, \rho), \quad (2)$$

where (x_1, y_1, z_1) is the upper left vertex and (x_2, y_2, z_2) the lower right vertex of the prism; (x, y, z) is the observation point and ρ the density, as shown in Figure 2.

To be able to discretise the cube, we define x_p , y_p , and z_p as the face numbers in the directions x , y , and z , respectively. If the cube is discretized in an homogeneous way, then we can define M , the number of prisms, as $M = x_p \times y_p \times z_p$, and the consecutive numbering of the prisms would be first in x , then y and finally in z . In case the ensemble is irregular, x_1 , y_1 , z_1 , x_2 , y_2 , z_2 , and ρ should be provided through a file for each prism. We define O as the number of observation points, which is determined by $O = N_x \times N_y$, where N_x and N_y are the number of observation points in the x and y directions,

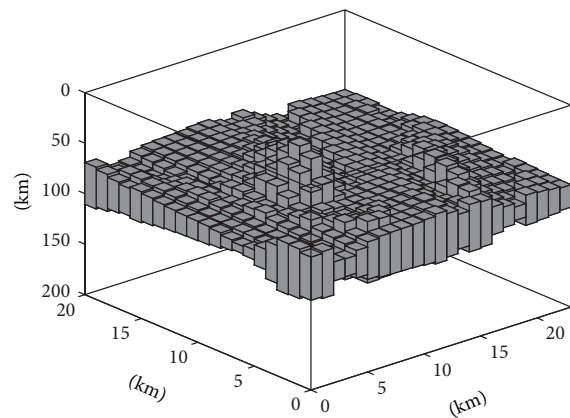


FIGURE 3: Illustration of an ensemble of irregular prisms.

respectively. Therefore, the number of times which is required to call the function defined in (2) to calculate the anomaly produced by a component is $M \times O$.

The first step to develop a parallel program is to search the finest granularity. This is important since CUDA handles a fine granularity paradigm. In this case it can be thought of parallelizing by prisms or by observation points, (see Figure 4), that is, by the number of elements in M or by the number of elements in O . One of the requisites which must be taken into consideration in the design is that it must be scalable, and hybrid systems must be considered since they are the most commonly used nowadays. However, many times design and type of architecture cannot be separated, especially when it is as specific as the CUDA architecture. Following the methodology proposed by Foster [10], it is necessary to analyze both parallelization schemes and examine which yields the best performance. Nevertheless, since $M \gg O$, in principle the best option of partitioning is by M , as

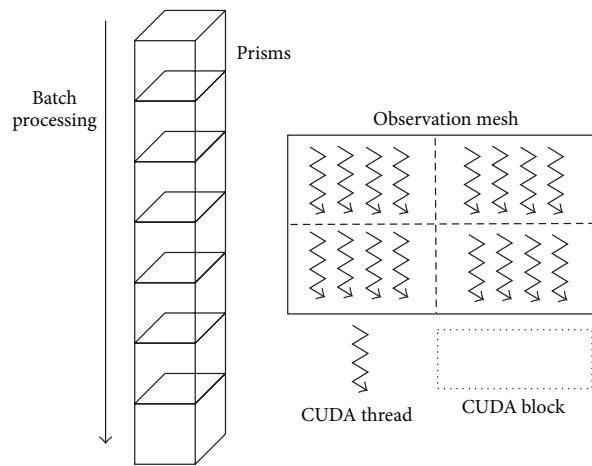


FIGURE 4: Partitioning by observation points.

detailed in the next section, even though this design requires a greater effort for its implementation.

3.1. Implementation in a Single GPU. The design for a GPU will consist of generating an observation grid (memory space) for each execution thread created in the TESLA card; in other words, if we create 14 blocks, each one containing 32 execution threads, we would have generated 448 observation grids. This design obeys the fact that we will select a partition by prisms, which implies less calls to kernel functions and is therefore more efficient in terms of execution time.

To give the correct dimension to our design, we analyze the simplest option of parallelization, which consists of partitioning the observation grid in the memory card for each prism.

This method of parallelization is the most trivial since it is enough to simply parallelize the cycle of the observations, which can even be done by using OpenACC, simplifying even more the work and avoiding the creation of the kernel by the programmer and leaving it to the compiler. However, a big drawback of this method is the excessive number of calls to the kernel function, which decreases performance since the parallel region is created and closed on each call. Additionally, it does not represent a big challenge design-wise to the point that the scheme can be solved with a compiler which could automatically generate parallel code.

On the other hand, the other parallelization option is to do it by prisms; in other words, make the threads divide the work by prisms. However, to avoid the coherence problems it is necessary to create a different space of memory for each execution thread, since it is not feasible to create just one memory space for a single observation grid, shared by all of the threads, since one of the principal problems which is not easy to handle in CUDA is the mutual exclusion of the threads in shared zones.

As can be seen in Figure 5, it is required to create an observation grid for each execution thread to avoid numerical consistency problems; if only one grid is occupied for all

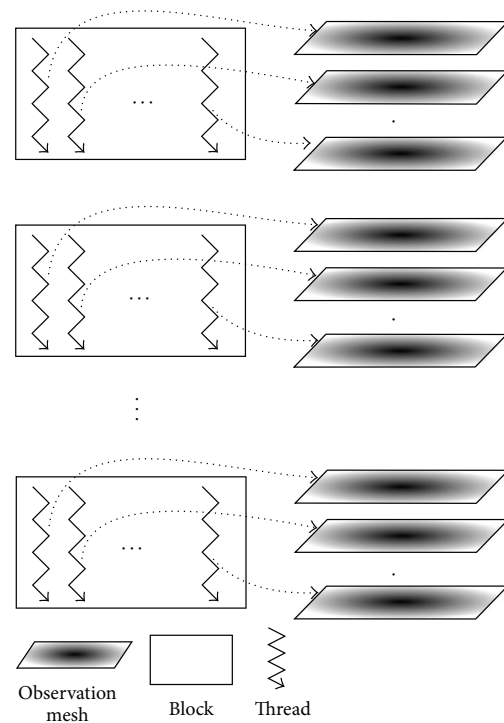


FIGURE 5: Partitioning by prisms using different memory spaces.

the threads, access conflicts will appear since several threads would write in the same memory location.

The number of created grids is equal to the number of created execution threads, and therefore the number of grids would be equal to the number of blocks times the number of threads contained in each block. This design allows the process at the same time as many prisms as threads are created, and therefore, for example, if 448 threads are created, then the same number of prisms will be processed in parallel in one kernel function execution and in the following call to the kernel function another 448 prisms will be processed and so successively until finalizing the process. In this way, the thread 1 will process the prisms set {1, 449, 897, 1345, ...}; in fact, the number of times that the kernel function is called is determined by

$$p = \left\lceil \frac{M}{T} \right\rceil, \tag{3}$$

where M is the number of prisms and T is the number of created threads, and consequently $p \ll M$. To exemplify that the prisms partitioning is better, let us suppose that 14 blocks are created, each one containing 512 execution threads, and then a total of 7,168 observation grids are generated. If we have a problem of 200,000 prisms, the number of calls to the kernel will be $\lceil 200,000 / 7,168 \rceil = 30$, and 30 is much smaller than 200,000. Therefore, we reduced the number of calls to the kernel in a 6,666X factor with respect to the partition by observation points.

```

If (K<=M) then
  For each j from 1 to Ny
    For each i from 1 to Nx
      G(Thread,i,j) = Gz(parameters) + G(Thread,i,j);
    End For
  End For
Else
  For each j from 1 to Ny
    For each i from 1 to Nx
      G(Thread,i,j) = G(Thread,i,j) + 0.0
    End For
  End For
End if

```

PSEUDOCODE 1

```

DO k = 1, T
!$acc parallel loop present (Gd, Gshared.d)
collapse (2)
DO i = 1, Nx
  DO j = 1, Ny
    Gshared_D(i,j) = Gd(k,i,j) + Gshared_D(i,j);
  ENDDO
ENDDO
!$acc end parallel loop
ENDDO

```

PSEUDOCODE 2

Each thread will follow a scheme of processing over the prisms as follows: a thread t will process the sequence of prisms

$$T \times (n - 1) + t, \quad (4)$$

where $n = 1, 2, 3, \dots, p$.

The implementation will initially consist of coding in device mode the functions which calculate the vectorial components (g_x, g_y, g_z) and the tensorial components $(g_{xx}, g_{yy}, g_{zz}, g_{xy}, g_{xz}, g_{yz})$, following the definition of (2). The functions of type device can only be called by kernels and are executed by a single CUDA thread.

To calculate any component it is necessary to allocate the memory for a tridimensional array G of size $T \times N_x \times N_y$. We define $ID = t \cdot x + (b \cdot x - 1) \times s \cdot x$, where $t \cdot x$ is the thread identifier, $b \cdot x$ the block identifier, and $s \cdot x$ the block size. Notice that we write $(b \cdot x - 1)$ because the thread and block identifiers in FORTRAN-CUDA are numbered starting from 1. We also define K as $K = (T) * (I - 1) + ID$, where T is the number of created threads (equal to the number of observation grids), while I is the number of partition over the set of prisms M in which it is working upon. The I partition of the set M is a division of M into nonoverlapping and nonempty subsets of size T that cover all of M . The subsets are collectively exhaustive and mutually exclusive with respect to the set M . The cardinality of the partition is determined by T (4) and the number of partitions is defined by p (3) and I can take the values $1, 2, 3, \dots, p$. The general scheme of the

computing kernel would be defined in Pseudocode 1, where Gz receives the parameters defined in (2), which are:

- (i) $Xa(K)$, $Ya(K)$, and $Za(K)$, the position in x , y , and z , respectively, of the upper left vertex of the prism,
- (ii) $Xb(K)$, $Yb(K)$, and $Zb(K)$, the position in x , y , and z , respectively, of the lower right vertex of the prism,
- (iii) $Xm(i)$, $Ym(j)$, and $Elev(i, j)$, the location in x , y , and z , respectively, of the observation point,
- (iv) $Rho(K)$, the density of the prism.

To calculate the final result of the anomaly it is necessary to add all of the obtained results from the threads, this is, the final anomaly G_f at a point (i, j) is approximated as:

$$G_f(i, j) = \sum_{k=1}^T G(k, i, j). \quad (5)$$

To generate the reduction (the sum), we can make use of the OpenACC which automatically generates the kernel, and the structure in FORTRAN is as in Pseudocode 2.

In Pseudocode 2, $Gshared$ is the bidimensional array where the reduction is made, and Gd is the tridimensional which contains the data of the anomaly generated by the different threads.

3.2. Multi-GPU Implementation. There are diverse possibilities to do the implementation using several GPUs [14]; one of

them is to use the same libraries provided by CUDA 4.0 to the development of peer-to-peer applications to communicate the GPUs within the same workstation; another possibility would be to use OpenMP. Nevertheless, we consider that the best option in this case is to use MPI since the number of required messages is not intensive and, unlike OpenMP or the peer-to-peer connection, it allows distributing the work to several GPUs which are not even connected in the same machine. In fact, the number of messages required to be sent for a problem with q MPI processes, each one controlling a GPU, is q itself; this is because each process sends its observation grid. The estimated communication time t_{comm} to the complete application is

$$t_{\text{comm}} = q(t_{\text{startup}} + t_{\text{data}}), \quad (6)$$

where t_{startup} is the necessary time to initialize the sending of the data and $t_{\text{data}} = N_x \times N_y$ is the size of the message, which in this case is the number of observation points.

Due to the fact that the parallelization in MPI is explicit, we need to manually distribute the number of prisms through a modular expression. Let us suppose that M is the number of prisms to calculate and that q is the number of cards that we are going to use. If every card is controlled by a process, we define the start and end of prisms to process by q as q_{start} and q_{end} , respectively. We then calculate the integer s as the quotient of M and the total number of processes q_n , and we determine the remainder r , both as

$$s = \frac{M}{q_n}, \quad (7)$$

$$r = \text{mod} \left(\frac{M}{q_n} \right).$$

Therefore

$$q_{\text{start}} = q \times s + 1, \quad (8)$$

$$q_{\text{end}} = (q + 1) \times s.$$

If $r \neq 0$ and $q < r$, then we adjust as

$$q_{\text{start}} = q_{\text{start}} + q, \quad (9)$$

$$q_{\text{end}} = q_{\text{end}} + (q + 1).$$

If $r \neq 0$ and $q \geq r$, then

$$q_{\text{start}} = q_{\text{start}} + r, \quad (10)$$

$$q_{\text{end}} = q_{\text{end}} + r.$$

In this way we can distribute the number of prisms M over q_n GPUs, in a balanced way.

Once the precedent distribution is done, we can occupy the implementation of the previous section to process the subset of local prisms for each GPU. Let us suppose that we have two workstations containing 4 GPUs each one; then in each station the cards will be numbered as 0, 1, 2, 3. To correctly select a device for each process q , numbered from 0 to 7, we use the function

$$f(q) = \text{mod}(q, n_d), \quad (11)$$

where n_d (constant) is the number of devices per machine (in this case 4). If $q = 4$ then $f(4) = 0$, and this means that the fifth process, numbered as 4, will be responsible of controlling the device 0 of the second team. It is necessary to note that this procedure works if the workload of processes is orderly distributed; in other words, in the first machine the processes 0, 1, 2, 3 are addressed and in the second 4, 5, 6, 7. If this distribution is not ordered, then the algorithm does not work properly, which is why for some MPI implementation the ordered flag can be specified as execution parameter, which orderly distributes the workload.

After selecting a device per process, we proceed to distribute the prisms between the 8 processes using (7). In this way the balance for a problem of 251,946 prisms results as follows:

$$q_0 = \{0, \dots, 31494\},$$

$$q_1 = \{31495, \dots, 62988\},$$

$$q_2 = \{62989, \dots, 94481\},$$

$$q_3 = \{94482, \dots, 125974\},$$

$$q_4 = \{125975, \dots, 157467\},$$

$$q_5 = \{157468, \dots, 188960\},$$

$$q_6 = \{188961, \dots, 220453\},$$

$$q_7 = \{220454, \dots, 251946\}.$$

After the subsets of prisms in which every card will work are defined, we apply the scheme used in Section 3.1.

4. Performance Experiments

As experiment we use a synthetic case composed by an ensemble of $700 \times 700 \times 50$ prisms with 7 spheres of contrast of variable density (Figure 6). The spheres are conformed by 251,946 prisms and an observation grid of $150 \times 100 = 15,000$ points at an elevation of 100 m. Therefore, the number of calls to a function to calculate a component of the tensor or vector is 3,779,190,000, which represents a high-performance computing problem.

We performed experiments to calculate the vectorial components G_x , G_y , and G_z , and the tensorial components G_{xx} , G_{yy} , G_{zz} , G_{xy} , G_{xz} , and G_{yz} in a single GPU, in a workstation with 4 GPUs and with two workstations containing 4 GPUs each one.

The characteristics of the workstations where the experiments took place are as follows:

- (i) 2 Intel(R) Xeon(R) CPU X5690 @ 3.47 GHz,
- (ii) 6 real cores per processor,
- (iii) hyperthreading technology disabled,
- (iv) 12 GB of RAM memory,
- (v) 4 TESLA Cards model C2070,
- (vi) operating system Red Hat 6.3.

The principal characteristics that we can highlight of the TESLA card C2070 are

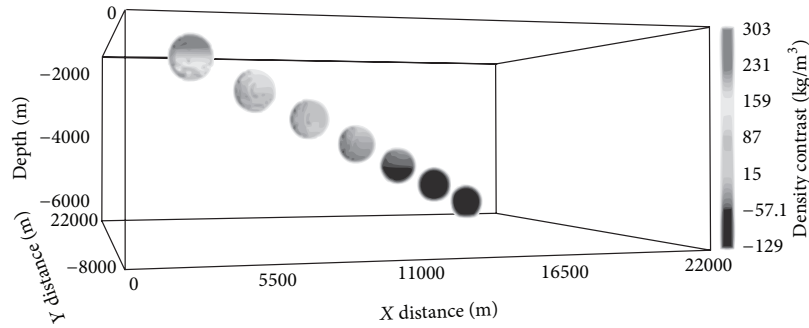


FIGURE 6: Synthetic problem setup with 7 spheres of variable density contrast (not scaled). Ensemble size of 22 km × 22 km × 8 km, 251,946 conform the spheres.

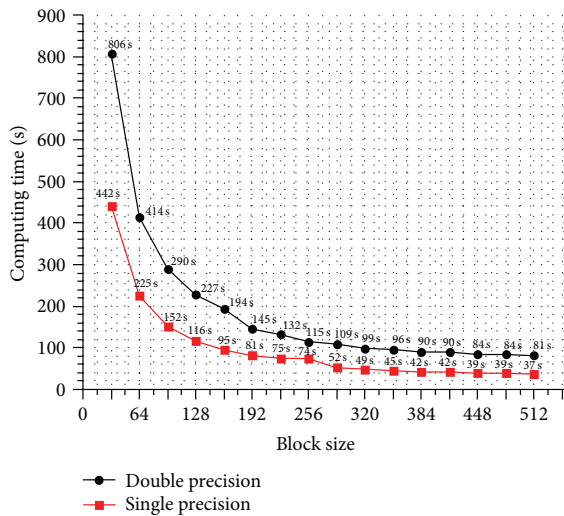


FIGURE 7: Comparison of the execution time using a variable block size in multiples of 32, in double and single precision.

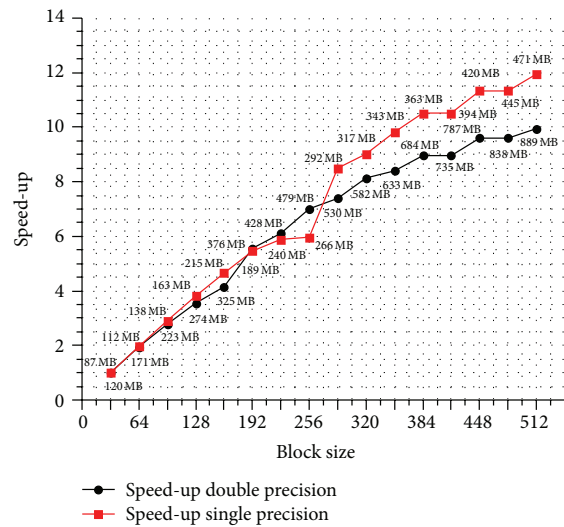


FIGURE 8: Speed-up of the behavior by increasing the number of threads per block in double and single precision, with its respective quantity of global memory required.

- (i) 14 multiprocessors (SM),
- (ii) 32 cores per multiprocessor (448 total CUDA cores),
- (iii) 6 GB of global memory DDR5,
- (iv) 515 theoretical GLOPS in double precision,
- (v) 1.03 theoretical TFLOPS in single precision,
- (vi) Frequency of the CUDA cores 1.15 GHz.

4.1. Performance of One Single Card. The first experiment was made in a C2070 card and consists of testing with different block sizes, keeping the number of blocks fixed at 14. The size of the selected block obeys the number of SMs available in the card and varies in multiples of 32 (warp size) until 512 threads per block, which means 16 experiments.

In Figure 7, the obtained computing times to solve the problem of the 7 spheres in double and single precision are shown. Notice how by increasing the size of the block, the execution time decreases exponentially to a limit or an

asymptotic time which in double precision is of 81 s and in single precision of 37 s. However, even when the best computing time is obtained with a block of 512 threads, this is the most memory-consuming setup. To improve the understanding of the behavior we used the speed-up as a metric, considering a block of 32 threads as the processing unit, and we label the quantity of required memory for each case.

By increasing the block size in multiples of the warp, we increase what is known as multiprocessor occupancy. In general, by increasing the occupancy we improve the use of the SM, and in this application this phenomenon is observed.

The speed-up graph depicted in Figure 8 shows that the computing time reduction is practically linear by increasing the occupancy for both cases, single and double precision, and later on it starts to stabilize, which means that in fact we do obtain an increase in the performance if we increase the block size in multiples of 32 threads. In the C2070 card,

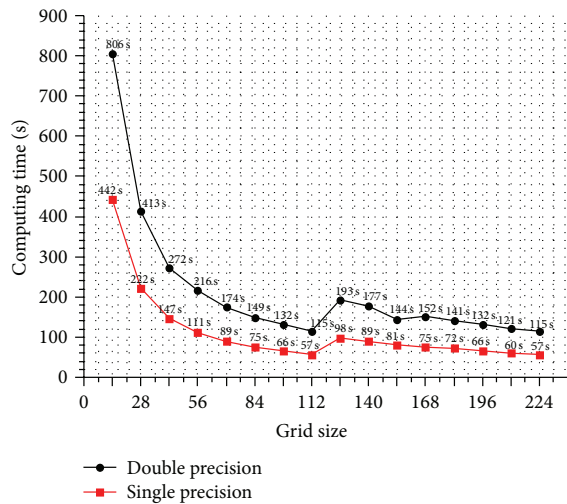


FIGURE 9: Comparison of the execution time using a variable grid size in multiples of 14, in double and single precision.

the maximum number of threads with their own memory space which could be created was 7,168 (14 blocks \times 512 threads), with a size of $150 \times 100 = 15,000$ elements, which along with the number of prisms which must be previously stored in the memory of the card add up 889 MB in double precision and 471 MB in single precision. We observe that the amount of used memory in GPU increases, but the performance is improved. The number of blocks and their size—the number of created threads—will depend on the type of the card which is being used. There are several cards of medium range which do not exceed 500 MB of memory; thus so many grids cannot be created.

We now examine the behavior of the performance if we set the number of threads fixed at 32 per block and vary the number of blocks from 14 to 224 in multiples of 14. The results of the execution times are shown in Figure 9, where it can be noted that the minimum in execution time is reached, both for single and double precision, when 112 blocks of 32 threads are created. Later on the time increases and starts decreasing again when 224 blocks are created. Comparing with the graph in Figure 7, the behavior produced by increasing the number of blocks is not as stable as increasing the number of threads per block since occupancy is not increased as only one warp per block is handled.

It is necessary to note that even though 32 blocks of 512 threads and 224 blocks of 32 threads sum up 7,168 execution threads, the second option is slower because it increases the computing time in a 42% for double precision and 54% for single precision. To simplify the writing, if we create n blocks with m threads, using double precision we write $\langle n, m, d \rangle$ and $\langle n, m, s \rangle$ in single precision.

In this particular case, the creation of many blocks is not as efficient as increasing the number of threads, as shown in the speed-up graph (Figure 10), considering every 14 blocks as a processing unit. In this graph a decrease of the speed-up can be clearly seen when reaching 126 blocks, but after this point

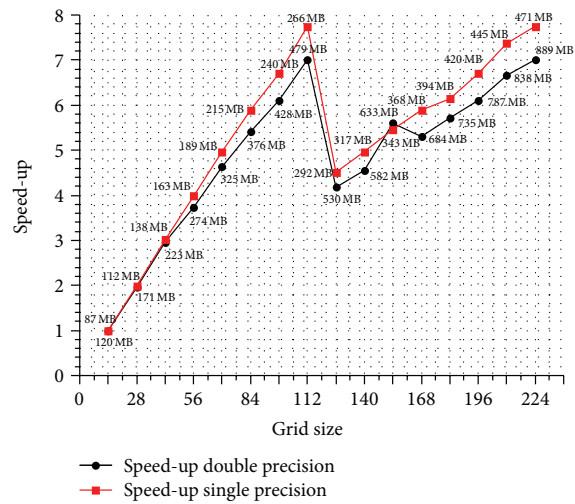


FIGURE 10: Speed-up of the behavior by increasing the quantity of blocks in double and single precision for a constant block size of 32 threads, with the respective quantity of required memory.

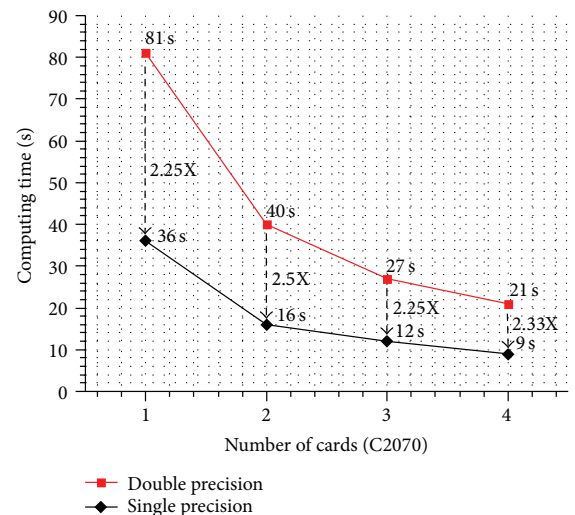


FIGURE 11: Execution time using up to four graphic cards in the same workstation in double and single precision.

the speed-up increases again until reaching at 224 blocks the same value obtained at 112 blocks.

4.2. Performance Using Multi-GPUs. In this subsection we analyze the performance using several GPUs integrated into the same workstation and distributed in two workstations, as was mentioned in Section 3.2; the choice to distribute the work between several GPUs was MPI. The best configuration found for this problem using only one C2070 card was to create 14 blocks with 512 threads, so this configuration was used. First we did experiments using four GPUs integrated into the same workstation. The results with respect to computing time are shown in Figure 11, where we considered a card as

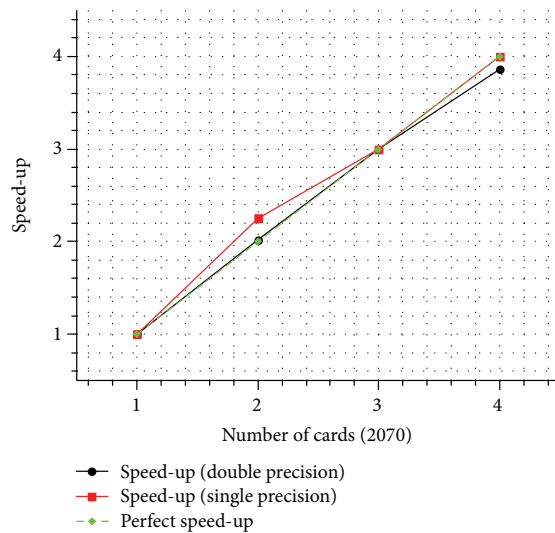


FIGURE 12: Speed-up of the behavior by increasing the number of processing cards in double and single precision. The number of blocks is 14 and the number of threads per block is 512 in every card, the optimal setup found for one single card.

a processing unit. It can be observed that the time decrease is proportional to the number of cards and that in single precision we reduced even more the time by a factor greater than 2X. The speed-up graph shows that the decrease of time is almost linear for double and single precision (Figure 12).

We then analyzed the performance using the cards in a distributed fashion. For this we used two workstations each one with four internal cards interconnected by a network of 1000 Gbytes. The results show that the communication latency is negligible since there is no perceivable overload because of the use of MPI. This is because the used MPI functions are only required at the end of the calculation to do the reductions. The configurations which are used in a distributed fashion are (1 + 1) (A setup (1 + 1) means the use of two cards in a distributed fashion, so a configuration (m+n) implies m cards in one workstation and n in other), (2 + 1), (2 + 2), (3 + 3), (4 + 4), and the execution times are shown in Figure 13.

In the graph shown in Figure 13 it can be observed that the execution times are practically the same for cases where the application is executed both in a local or distributed fashion. Thus, there is no difference between executing with four cards in the same workstation or with two in each workstation. With respect to the execution time, this decreases proportionally to the number of cards; however, in the cases where six and seven GPUs are used in one configuration (3 + 3) and (4 + 3), respectively, the execution times are the same since the work distribution requires the same number of executions of the kernel for each card, in this case 36. This happens because the number of prisms for the case of 7 cards is not reduced for less than 7168 per card.

The behavior of the speed-up for a distributed execution is represented in Figure 14. It can be observed a practically linear speed-up and in some particular cases up to a super

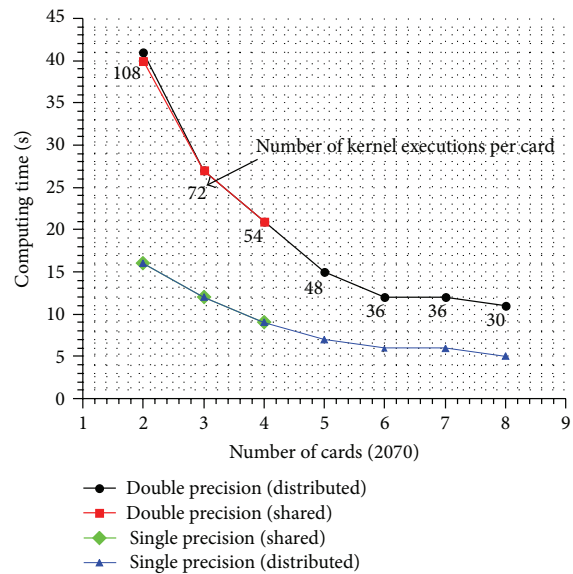


FIGURE 13: Execution time obtained using shared and distributed graphics cards, in double and single precision. We can see that MPI is not introducing overhead in the computing time.

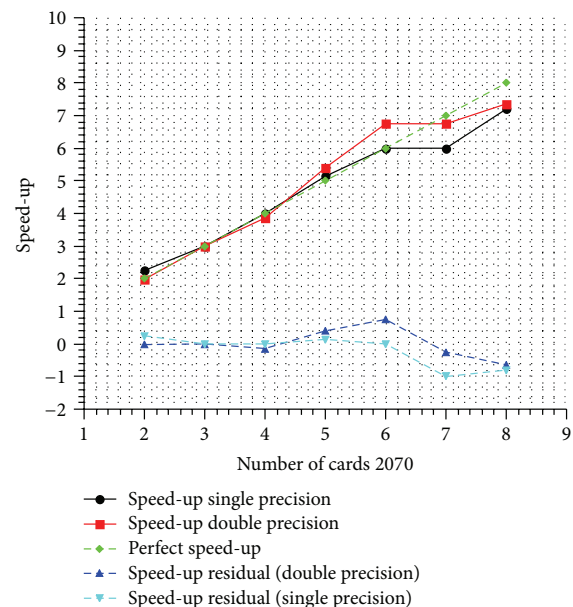


FIGURE 14: Speed-up obtained using shared and distributed graphics cards, in double and single precision.

speed-up, as is the case with the (3+3) configuration. Because of this we consider that the performance is excellent.

4.3. Comparison against a Cluster. To get a better perspective of the obtained performance with this CUDA implementation we compared it with the development made in OpenMP

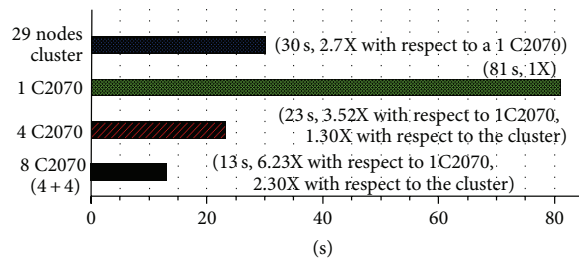


FIGURE 15: Comparison of the execution times for the calculation of the tensorial components in double precision between a small cluster against a single C2070 card, four C2070 cards in the same workstation, and 8 C2070 cards distributed in two workstations.

with MPI for cluster, and we also compared the results obtained in a cluster with the following characteristics:

- (i) node: 1 Intel Xeon processors model X5550 with four physical cores per processor (2 threads per core),
- (ii) 29 processing nodes,
- (iii) hyperthreading technology enabled,
- (iv) 40 GB of RAM per node,
- (v) Red Hat 6.3 as operating system.

The results against which this implementation is compared were obtained from a very efficient hybrid design and its implementation using MPI and OpenMP for the calculation of the gravimetry and gradiometry. The comparison of the results shows that, using 29 nodes of the cluster, it is required an execution time of 30 s; in a C2070 card using a $\langle 14, 512, d \rangle$ configuration, the execution time is of 81 s; four C2070 cards in the same machine communicated by MPI require 23 s; and 8 distributed CUDA cards (4 per workstation) demand 13 s. With these times we composed the bar chart shown in Figure 15.

The comparison is made against 29 nodes because it is the optimal number of nodes for the distribution in a problem of 251,946 prisms with an observation grid of 15,000 points. It turns out that the cluster is 2.7X faster than a single CUDA 2070 card, but if we occupy 4 cards these are 1.3X times faster than the cluster, and we can say that there is an approximate equivalence, for this problem in particular, between 29 nodes and 4 C2070 cards, but if we occupy 8 cards distributively, these are 2.3X times faster than the cluster.

5. Validation of the Numerical CUDA Code

We now proceed to verify the quality of the numerical solution produced using single card C2070 and four C2070 cards. The main objective of using the CUDA programming is to reduce the computing time; however, the quality of the numerical results must be validated and verified.

TABLE 1: Errors of the components of the gravimetric tensor, calculated with CUDA in double precision, with respect to its sequential counterpart in double precision.

Gravity components	Error L2 1-GPU (DP)	Error L2 4-GPU (DP)
G_z	$2.0582e - 09$	$2.0581e - 09$
G_x	$1.7107e - 09$	$1.7106e - 09$
G_y	$1.1162e - 09$	$1.1163e - 09$

TABLE 2: Errors of the components of the gravimetric tensor, calculated with CUDA in single precision, with respect its sequential counterpart in double precision.

Gravity components	Error L2 1-GPU (SP)	Error L2 4-GPU (SP)
G_z	1.3080	1.3080
G_x	0.8929	0.8929
G_y	0.9275	0.9275

TABLE 3: Errors of the components of the gradient tensor of gravity in double precision, with respect to its sequential counterpart.

Gravity gradient tensor components	Error L2 1-GPU (DP)	Error L2 4-GPU (DP)
G_{zz}	$4.6290e - 11$	$4.6373e - 11$
G_{xx}	$7.7994e - 11$	$7.8030e - 11$
G_{yy}	$8.8905e - 11$	$8.8935e - 11$
G_{xy}	$5.3476e - 11$	$5.3503e - 11$
G_{xz}	$9.2797e - 10$	$9.2792e - 10$
G_{yz}	$2.7225e - 10$	$2.7207e - 10$

TABLE 4: Errors of the components of the gradient tensor of gravity in single precision, with respect to its sequential counterpart.

Gravity gradient tensor components	Error L2 1-GPU (SP)	Error L2 4-GPU (SP)
G_{zz}	0.0310	0.0310
G_{xx}	0.0385	0.0385
G_{yy}	0.0392	0.0392
G_{xy}	0.0743	0.0743
G_{xz}	0.3085	0.3085
G_{yz}	0.3148	0.3148

To measure the error we use the formula of L2 norm, or RMS, defined as [18]

$$e = \sqrt{\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |g_{i,j}^p - g_{i,j}^s|^2}, \quad (12)$$

where $g_{i,j}^c$ is the component of the tensor calculated using the GPU, and $g_{i,j}^s$ is the component calculated serially in the CPU.

In Table 1 the errors of the components of the gravimetric tensor parallelly calculated with a GPU are shown, using the configuration of $\langle 14, 512, d \rangle$, with respect to the serial version.

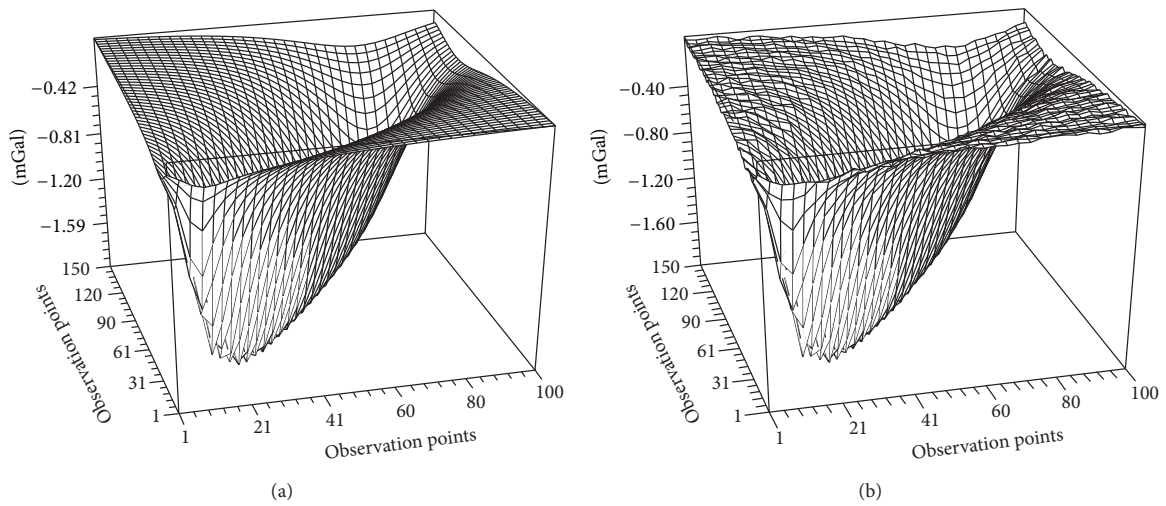


FIGURE 16: Component z of the vector G_z , (a) calculated in double precision, (b) calculated in single precision. Apparently no significant differences are seen between the graphs; however there is the introduction of roughness in single precision.

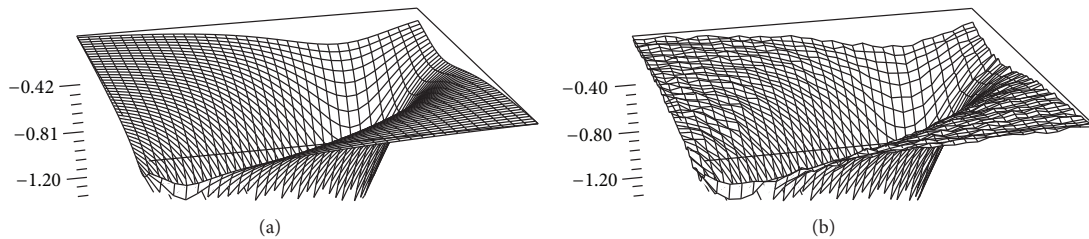


FIGURE 17: Zooming the component of the vector G_z , (a) calculated in double precision, (b) calculated in single precision. We can observe clearly the roughness introduced by the single precision.

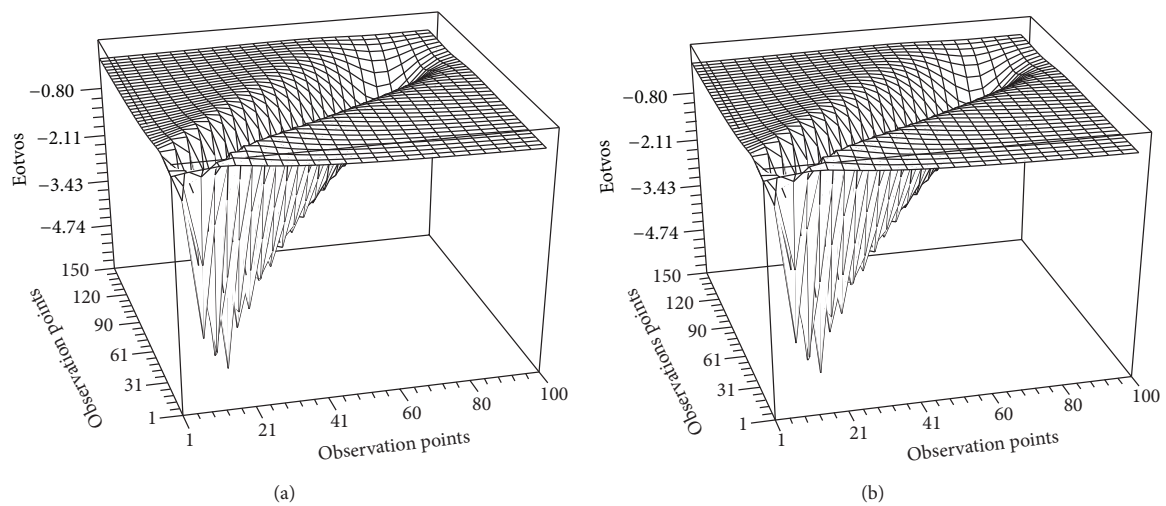


FIGURE 18: Component zz of the tensor G_{zz} , (a) calculated in double precision, (b) calculated in single precision. No significant differences are noticed between the plots.

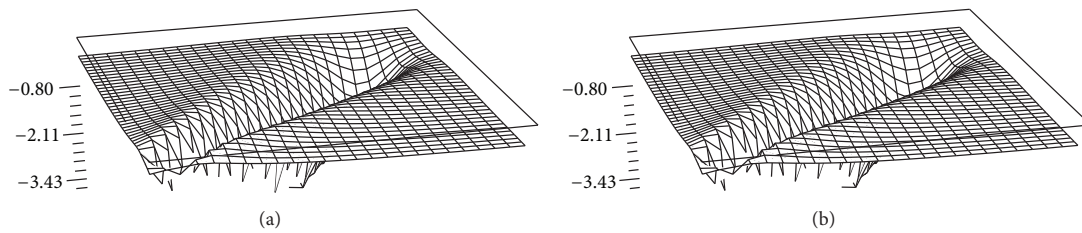


FIGURE 19: Zooming the component of the tensor G_{zz} , (a) calculated in double precision, (b) calculated in single precision. In this case the rugosity is not introduced by the single precision.

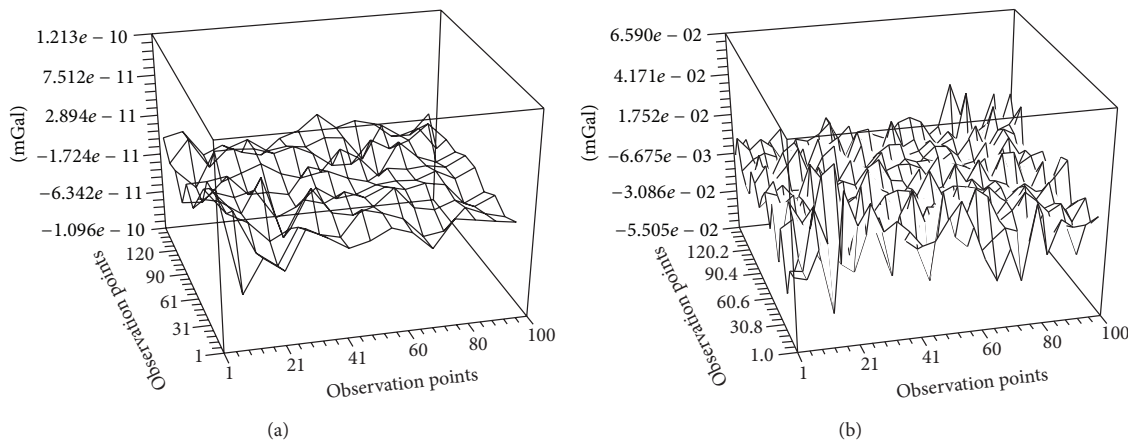


FIGURE 20: Absolute error of the components G_z calculated using four C2070 cards in (a) double precision and (b) single precision with respect to double precision CPU-cluster results.

The errors in single precision for the same configuration (14, 512, s) are shown in Table 2.

The errors of the components of the gradient tensor of gravity calculated with CUDA in double precision, with respect to its sequential counterpart, are shown in Table 3.

And finally in Table 4 the errors of the components of the gradient tensor of gravity are shown, calculated with CUDA in single precision, with respect to its sequential counterpart.

It is necessary to mention that the sequential version is calculated in double precision and, as can be seen, there is practically no difference between the CUDA version in double precision and the reference solution. Nevertheless, in single precision the calculation of the vectors produces an error more significant than the calculation of the tensors. To observe how the error propagates in single precision, we show in Figure 16 the calculation of G_z in double and single precision, using a single GPU.

As can be noted, apparently the same anomaly result is reproduced both in double precision as in single, but in this last one the numerical roughness is pronounced. To examine this phenomenon in greater detail, we can zoom in Figure 16 to see with more detail the introduced rugosity depicted in Figure 17.

However, the numerical rugosity problem in single precision is left to the criterion of anyone interested in the result, since this precision requires practically 40% and 50% less in computing time and global memory, respectively.

To note that the single precision does not always introduce pronounced numerical rugosities, we can observe the behavior of the calculation of G_{zz} , which does not introduce rugosity problems. Figure 18 shows the G_{zz} tensor calculated in double and single precision, and Figure 19 is zoomed to focus on the detail.

Finally we mention that single precision can be used if done properly, depending on the requirements of the particular problem. The maximum absolute error found in G_z with single precision is of 0.0566 mGal and for G_{zz} of 0.0011 Eotvos (see Figures 20 and 21). As we can see in the calculation of G_{zz} the introduced error is less than in G_z .

6. Conclusions

A parallel design for the calculation of the vectorial and tensorial components of the gravity using CUDA was implemented and validated. The numerical experiments and the obtained metrics validate that the implementation is very efficient and that it also produces good results with respect to the numerical solution.

We showed that selecting the simplest or most trivial parallelization technique does not necessarily leads to the best performance or the best use of the platform. In our particular case, even though the partitioning by prisms requires a greater inversion in the design and implementation, this

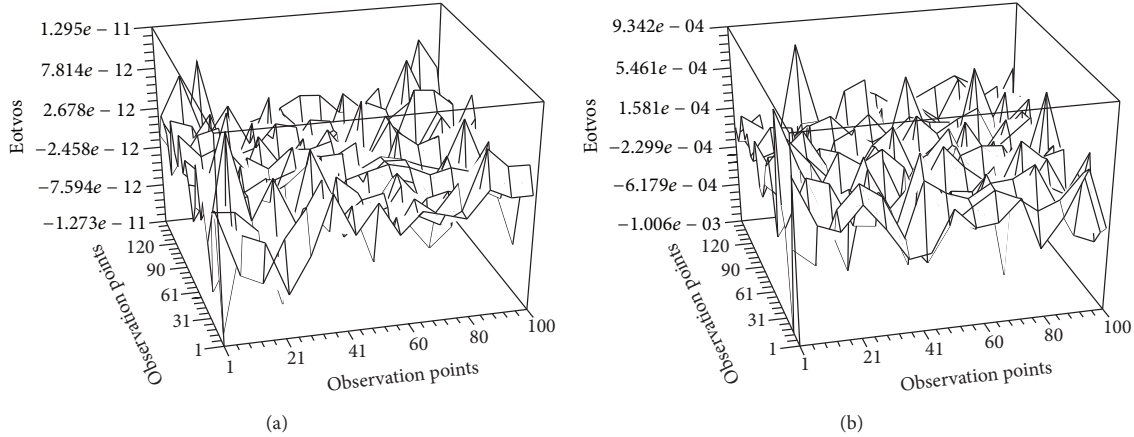


FIGURE 21: Absolute error of the components G_{zz} calculated using four C2070 cards in (a) double precision and (b) single precision with respect to double precision CPU-cluster results.

method is the most advantageous with respect to performance because the transfers between the CPU and the GPU are minimized, moving more executable code to the GPU and also the number of calls to the kernel functions is reduced.

The multi-GPU version using MPI as controller and balancer of the workload was correctly implemented since it produces practically the same results as those of the version for just one GPU. In double precision we can say that there is no difference between the calculations by the CPU and the GPU. The single precision can be used with confidence in the calculation of the tensorial components and, with appropriate considerations, in the calculation of the vectorial components as well.

It was shown that for our synthetic problem, approximately 29 nodes are equivalent to 4 four C2070 cards. This obviously shows the economical benefit of using CUDA, since it is cheaper to acquire 4 graphic cards than 29 nodes of processing, and clearly the maintenance and energetic consume is considerably smaller. Nevertheless, we consider that the CUDA implementation is much more costly from the point of view of the design and the required time for its programming.

We can also conclude that this implementation can serve as a design pattern to parallelize numerical schemes where the computational space cannot be disjointly divided between the processing cores, therefore minimizing the execution of the number of kernels calls.

Finally we expect that GPU computing will enable us, in a near future, to optimize the numerical burden of large scale geophysical applications such as potential field modeling of impact craters [19] and multiparameter geophysical global optimization by heuristic methods [20, 21].

Appendix

Calculation of Gravitational Quantities

The Earth's gravitational potential G is a scalar quantity and its shape can be constrained by its slope in the x , y , and z

directions, called the gravitational attraction G_x , G_y , and G_z (gravity vector field). In this work, we have investigated how to parallelize the analytical calculation of the components of the gravity field vector and the gravity gradients represented by a nine-component tensor; because of the symmetrical or irrotational attribute, the gravity gradient tensor is reduced to only six independent components: G_{xx} , G_{yy} , G_{zz} (the vertical gravity gradient), G_{xy} , G_{xz} , and G_{yz} . For the right rectangular prism model, the analytical formulae for the three components vectors and the six gravity gradient components, corresponding to (2), are given by

$$\begin{aligned}
 G_x &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \left[y_j \ln(z_k + r_{ijk}) \right. \\
 &\quad \left. + z_k \ln(y_j + r_{ijk}) \right. \\
 &\quad \left. - x_i \arctan \frac{y_j z_k}{x_i r_{ijk}} \right], \\
 G_y &= -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \left[z_k \ln(x_i + r_{ijk}) \right. \\
 &\quad \left. + x_i \ln(z_k + r_{ijk}) \right. \\
 &\quad \left. - y_j \arctan \frac{z_k x_i}{y_j r_{ijk}} \right], \\
 G_z &= -\gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \left[x_i \ln(y_j + r_{ijk}) \right. \\
 &\quad \left. + x_i \ln(z_k + r_{ijk}) \right. \\
 &\quad \left. - y_j \arctan \frac{z_k r_{ijk}}{x_i y_j} \right],
 \end{aligned}$$

$$\begin{aligned}
 G_{xx} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{y_j z_k}{x_i r_{ijk}}, \\
 G_{yy} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{x_i z_k}{y_j r_{ijk}}, \\
 G_{zz} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \arctan \frac{x_i y_j}{z_k r_{ijk}}, \\
 G_{xy} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(z_k + r_{ijk}), \\
 G_{xz} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(y_j + r_{ijk}), \\
 G_{yz} &= \gamma\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \ln(x_i + r_{ijk}),
 \end{aligned}
 \tag{A.1}$$

where $r_{ijk} = \sqrt{x_i^2 + y_j^2 + z_k^2}$, $\gamma = 6.673 \times 10^{-11}$ and $x_1 = (x_l - x_p) \times 10^3$, $x_2 = (x_r - x_p) \times 10^3$, $y_1 = (y_l - y_p) \times 10^3$, $y_2 = (y_r - y_p) \times 10^3$, $z_1 = (z_l - z_p) \times 10^3$, $z_2 = (z_r - z_p) \times 10^3$.

Glossary

CPU:	Central processing unit, of one or several cores in shared memory
Device or GPU:	Graphics processing unit. For instance, the Tesla C2070 card
Device function:	Is a function that can only be executed by a CUDA thread and is called by a kernel
Grid:	A set of blocks of threads. A kernel is executed in a grid of blocks
CUDA thread:	A CUDA thread is a light process which executes a distinct sequence of the contained code in a kernel and resides in the GPU
Block identifier:	It is analogous to the thread identifier and identifies the block within a grid. It is accessed with the variable <code>blockIdx</code>

Thread identifier:	It is a value between 0 and n in C language or between 1 and n in FORTRAN language, which functions as thread identifier within a block. It is accessed with the variable <code>threadIdx</code> . This variable is very useful to distribute the work among different threads and can be handled up to 3 components (x, y, z), coinciding with the dimensions of blocks of threads, but not necessarily
Kernel:	A function or procedure executed parallelly in the device which is executed by the CUDA threads
Global memory:	Uncached off-chip DRAM memory
Multiprocessor:	It is a processing unit containing 8 CUDA cores
CUDA core:	It is a core of processing contained inside a multiprocessor and dispatches the threads contained in a block
Numerical rugosity:	It is defined for this work as numerical rugosity to the effect produced by truncate and round to 7 decimals the precision of the floating-point numbers, which produces values above or below the exact solution
Block size:	It indicates the number of threads contained inside the block and it is accessed through the variable <code>blockDim</code> . It can contain the three dimensions
Multiprocessor occupancy:	It is the quotient of the number of warps executing concurrently in a multiprocessor, divided between the maximum number of warps which can be executed concurrently
Warp:	It is a group of 32 threads which execute concurrently in a GPU multiprocessor.

Acknowledgments

The authors want to acknowledge the support provided by the Mexican Institute of Petroleum to access its computing equipment, based on the financing received through project SERNER-CONACYT 128376 (IMP Y.00107) created jointly by IMP-SENER-CONACYT. They want to express their special

gratitude to Dr. Raul del Valle and Dr. Oleg Titov for all the support received in their laboratory.

References

- [1] R. Vuduc and K. Czechowski, "What GPU computing means for high-end systems," *IEEE Micro*, vol. 31, no. 4, pp. 74–78, 2011.
- [2] J. D. Owens, D. Luebke, N. Govindaraju et al., "A survey of general-purpose computation on graphics hardware," *Computer Graphics Forum*, vol. 26, no. 1, pp. 80–113, 2007.
- [3] C. J. Webb and S. Bilbao, "Computing room acoustics with CUDA-3D FDTD schemes with boundary losses and viscosity," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 317–320, May 2011.
- [4] N. Nakata, T. Tsuji, and T. Matsuoka, "Acceleration of computation speed for elastic wave simulation using a Graphic Processing Unit," *Exploration Geophysics*, vol. 42, no. 1, pp. 98–104, 2011.
- [5] R. Abdelkhalek, H. Calandra, O. Coulaud, J. Roman, and G. Latu, "Fast seismic modeling and reverse time migration on a GPU cluster," in *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS '09)*, pp. 36–43, June 2009.
- [6] J. Yang, Y. Wang, and Y. Chen, "GPU accelerated molecular dynamics simulation of thermal conductivities," *Journal of Computational Physics*, vol. 221, no. 2, pp. 799–804, 2007.
- [7] T. Brandvik and G. Pullan, "Acceleration of a two-dimensional Euler flow solver using commodity graphics hardware," *Journal of Mechanical Engineering Science*, vol. 221, no. 12, pp. 1745–1748, 2007.
- [8] R. Capuzzo-Dolcetta, A. Mastrobuono-Battisti, and D. Maschietti, "NBSymple, a double parallel, symplectic N-body code running on graphic processing units," *New Astronomy*, vol. 16, no. 4, pp. 284–295, 2011.
- [9] L.-G. Du, K. Li, F.-M. Kong, and Y. Hu, "Parallel 3D finite-difference time-domain method on multi-GPU systems," *International Journal of Modern Physics C*, vol. 22, no. 2, pp. 107–121, 2011.
- [10] I. Foster, *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*, Addison-Wesley Longman, Boston, Mass, USA, 1995.
- [11] D. Michéa and D. Komatitsch, "Accelerating a three-dimensional finite-difference wave propagation code using GPU graphics cards," *Geophysical Journal International*, vol. 182, no. 1, pp. 389–402, 2010.
- [12] M. Moorkamp, M. Jegen, A. Roberts, and R. Hobbs, "Massively parallel forward modeling of scalar and tensor gravimetry data," *Computers and Geosciences*, vol. 36, no. 5, pp. 680–686, 2010.
- [13] L. Dagum and R. Menon, "Openmp: an industry-standard api for shared-memory programming," *IEEE Computing in Science and Engineering*, vol. 5, no. 1, pp. 46–55, 1998.
- [14] T.-Y. Liang, H.-F. Li, and J.-Y. Chiu, "Enabling mixed openmp/mpi programming on hybrid cpu/gpu computing architecture," in *Proceedings of the IEEE 26th International Parallel and Distributed Processing Symposium Workshops (IPDPSW '12)*, pp. 2369–2377, 2012.
- [15] D. Komatitsch, "Fluid-solid coupling on a cluster of GPU graphics cards for seismic wave propagation," *Comptes Rendus Mécanique*, vol. 339, no. 2-3, pp. 125–135, 2011.
- [16] B. Heck and K. Seitz, "A comparison of the tesseroid, prism and point-mass approaches for mass reductions in gravity field modelling," *Journal of Geodesy*, vol. 81, no. 2, pp. 121–136, 2007.
- [17] Z. Chen, X. Meng, and L. Guo, "Gicuda: a parallel program for 3d correlation imaging of large scale gravity and gravity gradiometry data on graphics processing units with cuda," *Computers and Geosciences*, vol. 46, pp. 119–128, 2012.
- [18] K. L. Mickus and J. H. Hinojosa, "The complete gravity gradient tensor derived from the vertical component of gravity: a Fourier transform technique," *Journal of Applied Geophysics*, vol. 46, no. 3, pp. 159–174, 2001.
- [19] C. Ortiz-Alemán and J. Urrutia-Fucugauchi, "Aeromagnetic anomaly modeling of central zone structure and magnetic sources in the Chicxulub crater," *Physics of the Earth and Planetary Interiors*, vol. 179, no. 3-4, pp. 127–138, 2010.
- [20] M. G. Orozco-del Castillo, C. Ortiz-Alemán, J. Urrutia-Fucugauchi, R. Martin, A. Rodriguez-Castellanos, and P. E. Villaseñor-Rojas, "A genetic algorithm for filter design to enhance features in seismic images," *Geophysical Prospecting*, 2013.
- [21] J. L. Rodríguez-Zúñiga, C. Ortiz-Alemán, G. Padilla, and R. Gaulon, "Application of genetic algorithms to constrain shallow elastic parameters using in situ ground inclination measurements," *Soil Dynamics and Earthquake Engineering*, vol. 16, no. 3, pp. 223–234, 1997.

B

IMPLEMENTACIÓN COMPUTACIONAL

Para realizar el modelado gravimétrico en forma eficiente, se diseñaron programas y subprogramas en el lenguaje compilado FORTRAN, estándar 2008, óptimo para hacer cómputo numérico.

Para el caso de la esfera, se diseñó un solo subprograma en el que se incluyó el cálculo de todas las componentes vectoriales y tensoriales, mientras que para el prisma rectangular, se diseñaron subprogramas por separado para el cálculo de las componentes vectoriales y tensoriales en un punto de observación, con una estructura basada en la de la subrutina **gbox.f**, codificada en FORTRAN 77 y publicada por Blakely (1996).

A continuación se enlistan y explican los parámetros requeridos por los subprogramas **EsferaG.f08**, **VectorGBox.f08** y **TectorGBox.f08**, diseñados para el cálculo de las cantidades gravitacionales mencionadas:

Esfera (EsferaG.f08):

$$[g_i, T_{ij}] (x_0, y_0, z_0) = f(x_0, y_0, z_0, a, x_c, y_c, z_c, \rho)$$

donde:

- g_i : Son las componentes del vector de atracción gravitacional (g_x, g_y y g_z).
- T_{ij} : Son las componentes del tensor de gradientes gravitacionales ($T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ y T_{zz}).
- x_0, y_0 y z_0 : Son las coordenadas del punto de observación.
- a : Es el radio de la esfera.
- x_c, y_c y z_c : Son las coordenadas del centro de la esfera.
- ρ : Es la densidad de la esfera.

Prisma (VectorGBox.f08 y TensorGBox.f08):

$$g_i(x_0, y_0, z_0) = f(x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, \rho, i) \quad i = x, y, z$$

$$T_{ij}(x_0, y_0, z_0) = f(x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, \rho, ij) \quad ij = xx, xy, xz, yy, yz, zz$$

donde:

- x_0, y_0 y z_0 : Son las coordenadas del punto de observación.
- x_1, y_1 y z_1 : Punto inicial para la delimitación del prisma (ver Figura 2.8).
- x_2, y_2 y z_2 : Punto final para la delimitación del prisma (ver Figura 2.8).
- ρ : Es la densidad del prisma.

En cuanto al costo computacional empleado para realizar el modelado directo a través de ensamblajes, en las Figuras B.1 y B.2 se ilustra la relación de proporcionalidad entre resolución del ensamblaje y tiempo de cómputo de cada una de las componentes del vector y tensor gravitacional en un punto de observación:

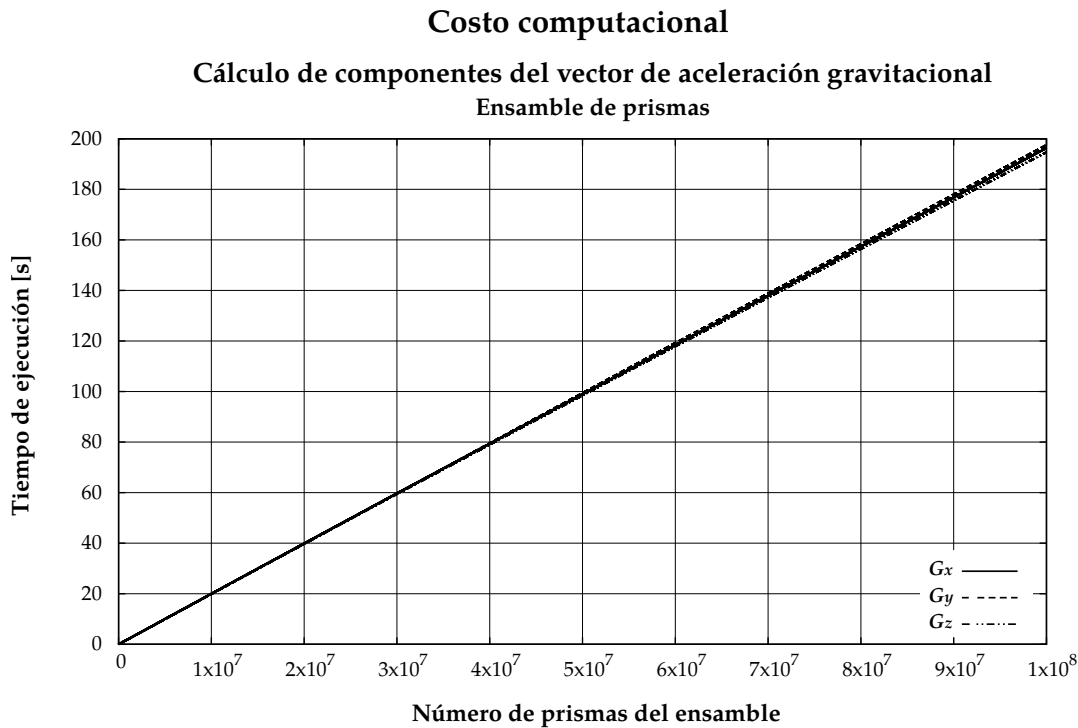


FIGURA B.1: Curvas de costo computacional empleado para el cálculo de las componentes vectoriales G_x , G_y y G_z , de ensamblajes de hasta 100 millones de prismas, en un punto de observación. Procesador Utilizado: Intel[®] Xeon[®] CPU E5-2687W v2 @ 3.40GHz × 16.

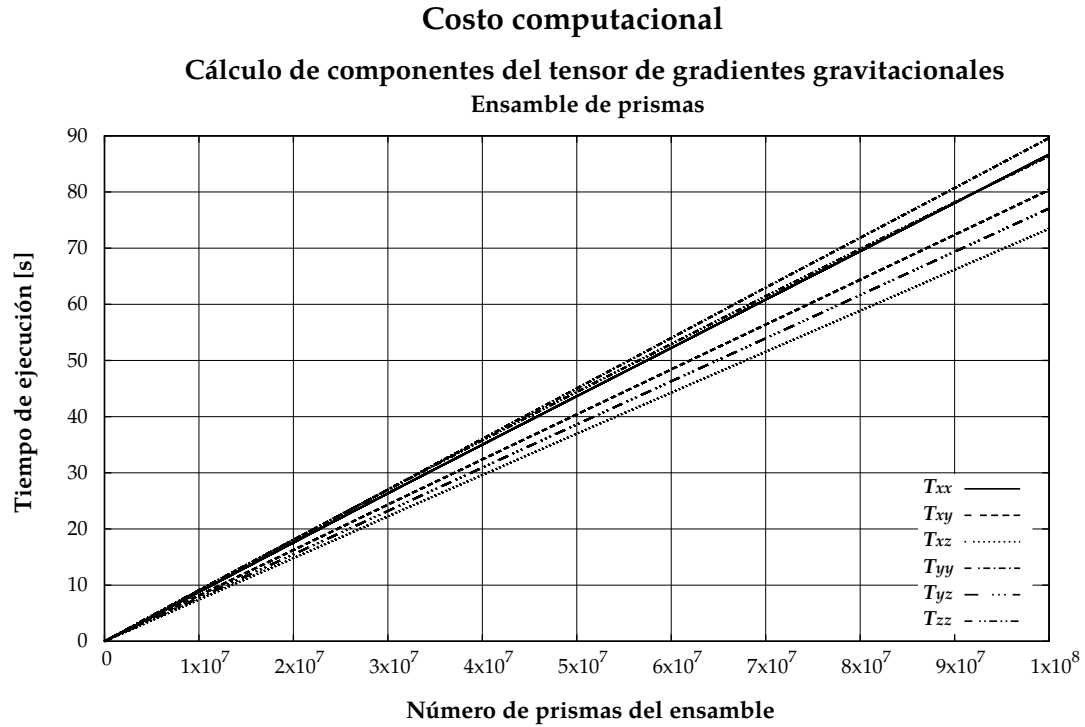


FIGURA B.2: Curvas de costo computacional empleado para el cálculo de las componentes tensoriales T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} y T_{zz} , de ensamblajes de hasta 100 millones de prismas, en un punto de observación. Procesador Utilizado: Intel[®] Xeon[®] CPU E5-2687W v2 @ 3.40GHz × 16.

Las especificaciones del procesador Intel[®] Xeon[®] CPU E5-2687W v2 @ 3.40GHz × 16 son las siguientes:

Procesador : Intel[®] Xeon[®] CPU E5-2687W v2 @ 3.40GHz × 16
Número de núcleos : 8
Número de hilos : 16
Frecuencia Base del Procesador : 3.40 GHz
Frecuencia Turbo Máxima : 4.00 GHz
Caché : 25 MB
Velocidad de BUS : 4 GHz

De las Figuras B.1 y B.2 se hace evidente la relación de proporcionalidad entre la resolución del ensamble y el tiempo de cómputo, así como el gran costo computacional que implica el cálculo para ensambles de grandes dimensiones: Para ensambles de 100 millones de prismas ($1000 \times 1000 \times 100$ prismas en direcciones X, Y y Z, respectivamente), se requieren prácticamente de 200 [s] para calcular cada componente del vector de aceleración gravitacional y de 73 [s] a 90 [s] para cada componente del tensor de gradientes gravitacionales en un solo punto de observación, lo que se traduciría en más de 23.5 días para calcular una malla de 101×101 puntos, por cada una de las componentes G_x , G_y o G_z y cerca de 10.5 días para cada malla con el mismo número de puntos, de cada una de las componentes T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} o T_{zz} , utilizando la unidad de procesamiento descrita anteriormente y utilizada para realizar los cálculos en esta tesis.

Por otro lado, de las expresiones que permiten calcular las componentes del vector de atracción gravitacional y las componentes del tensor de gradientes gravitacionales, se sabe que el cálculo de la contribución individual de cada prisma del ensamble a la respuesta gravimétrica total en cada uno de los puntos observados, es independiente, lo cual, aunado a la relación directa existente entre resolución del ensamble y tiempo de cómputo, refleja la necesidad y posibilidad de emplear cómputo de alto rendimiento para reducir tiempos de ejecución en este tipo de modelado. Se recomienda consultar las publicaciones relacionadas con el tema (Couder-Castañeda et al., 2013; Couder-Castañeda et al., 2015), incluidas en el Anexo A.

A continuación se pueden consultar los códigos de algunos de los subprogramas auxiliares diseñados para el modelado gravimétrico 3D, directo e inverso. Todos han sido codificados en lenguaje **FORTRAN 2008** y se compilan y ejecutan adecuadamente con el compilador libre **gfortran**, versión **5.4.1** y posteriores:

NumKind.f08

Establece los “números KIND” para variables enteras y reales utilizadas por el resto de los subprogramas incluidos en este anexo, con el propósito de unificar la precisión entre diferentes plataformas y compiladores FORTRAN en que se podrían implementar.

EsferaG.f08

Calcula el potencial, componentes vectoriales y tensoriales gravitacionales generadas por un cuerpo esférico con densidad uniforme, en un punto de observación.

PotenGBox.f08

Calcula el potencial gravitacional de un cuerpo prismático rectangular con densidad uniforme, en un punto de observación.

VectorGBox.f08

Calcula las componentes del vector de atracción gravitacional de un cuerpo prismático rectangular con densidad uniforme, en un punto de observación.

TensorGBox.f08

Calcula las componentes del tensor de gradientes gravitacionales de un cuerpo prismático rectangular con densidad uniforme, en un punto de observación.

NumKind.f08

```

1  !-----!
2  !Módulo NumKind [Nava–Flores, 2017]
3  !– Basado en el módulo NumericKinds [Chirila, D. B., & Lohmann, G. (2015)].
4  !
5  !Calcula y establece los números "KIND" y sus "alias", para los siguientes tipos de variables:
6  ! – Integer:
7  ! > Simple (IS)
8  ! > Long (IL)
9  ! > Long Long (ILL)
10 !
11 ! – Real:
12 ! > Single Precision (SP)
13 ! > Double Precision (DP)
14 ! > Quadruple Precision (QP)
15 !
16 !El objetivo de este módulo es definir la precisión de este tipo de variables, independientemente
17 !del compilador utilizado.
18 !
19 !Una vez calculados los números kind, se pueden utilizar a través de sus "alias"
20 !(IS, IL, ILL, SP, DP y QP)
21 !
22 !Se incluye una subrutina para realizar el diagnóstico completo de los tipos de variables que se
23 !pueden utilizar en la computadora en la que se corre:
24 ! – Para correr el diagnóstico, hay que hacer la llamada desde el programa principal a la subrutina
25 ! ImprimeKind (sin parámetros).
26 !
27 !Última modificación: Septiembre de 2017
28 !-----!
29 module NumKind
30
31     implicit none
32
33     !Números KIND para diferentes tipos de enteros:
34     integer, parameter:: IS = selected_int_kind(4), &
35                          IL = selected_int_kind(9), &
36                          ILL= selected_int_kind(18)
37
38     !Números KIND para diferentes tipos de reales:
39     integer, parameter:: SP = selected_real_kind(6,37), &
40                          DP = selected_real_kind(15,307), &
41                          QP = selected_real_kind(33,4931)
42
43     contains
44
45     !-----!
46     !Subrutina ImprimeKind [Nava–Flores, 2017]
47     !– Basada en la subrutina printSupportedRealKinds [Chirila, D. B., & Lohmann, G. (2015)].
48     !
49     !Realiza un diagnóstico para determinar si la computadora en la que se utiliza el módulo
50     !NumKind.f08, soporta todos los tipos de variables especificados en él y lo despliega en
51     !pantalla.
52     !
53     !La subrutina ImprimeKind no tiene parámetros.
54     !
55     !Última modificación: Septiembre de 2017
56     !-----!
57     subroutine ImprimeKind()

```

```

58
59     implicit none
60
61     integer(IS):: intIS
62     integer(IL):: intIL
63     integer(ILL):: intILL
64     real(SP):: reSP
65     real(DP):: reDP
66     real(QP):: reQP
67
68     write(*,10)
69
70     if (IS > 0) then
71         write(*,'(a)') '_____Enteros_tipo_"IS"_soportados!!!'
72         write(*, '(a,i0)') '_____Número_KIND:_' , IS
73         write(*, '(a,i0)') '_____Número_de_bits_de_int (IS):_' , bit_size(intIS)
74         write(*,'(a,i0,/)' ) '_____Máximo_entero (IS):_' , huge(intIS)
75     else
76         write(*, '(a,/)' ) ' _Enteros_tipo_"IS"_NO_SOPORTADOS!!!'
77     end if
78
79     if (IL > 0) then
80         write(*,'(a)') '_____Enteros_tipo_"IL"_soportados!!!'
81         write(*, '(a,i0)') '_____Número_KIND:_' , IL
82         write(*, '(a,i0)') '_____Número_de_bits_de_int (IL):_' , bit_size(intIL)
83         write(*,'(a,i0,/)' ) '_____Máximo_entero (IL):_' , huge(intIL)
84     else
85         write(*, '(a,/)' ) ' _Enteros_tipo_"IL"_NO_SOPORTADOS!!!'
86     end if
87
88     if (ILL > 0) then
89         write(*,'(a)') '_____Enteros_tipo_"ILL"_soportados!!!'
90         write(*, '(a,i0)') '_____Número_KIND:_' , ILL
91         write(*, '(a,i0)') '_____Número_de_bits_de_int (ILL):_' , bit_size(intILL)
92         write(*,'(a,i0,/)' ) '_____Máximo_entero (ILL):_' , huge(intILL)
93     else
94         write(*, '(a,/)' ) ' _Enteros_tipo_"ILL"_NO_SOPORTADOS!!!'
95     end if
96
97     if (SP > 0) then
98         write(*,'(a)') '_____Reales_tipo_"SP"_soportados!!!'
99         write(*, '(a,i0)') '_____Número_KIND:_' , SP
100        write(*, '(a,i0)') '_____Número_de_Bytes_de_real (SP):_' , sizeof(reSP)
101        write(*, '(a,i0)') '_____Decimales_de_precisión_real (SP):_' , precision(reSP)
102        write(*, '(a,i0)') '_____Rango_real (SP):_' , range(reSP)
103        write(*, '(a,i0)') '_____Mínimo_exponente_de_real (SP):_' , minexponent(reSP)
104        write(*,'(a,i0,/)' ) '_____Máximo_exponente_de_real (SP):_' , maxexponent(reSP)
105    else
106        write(*, '(a,/)' ) ' _Reales_tipo_"SP"_NO_SOPORTADOS!!!'
107    end if
108
109    if (DP > 0) then
110        write(*,'(a)') '_____Reales_tipo_"DP"_soportados!!!'
111        write(*, '(a,i0)') '_____Número_KIND:_' , DP
112        write(*, '(a,i0)') '_____Número_de_Bytes_de_real (DP):_' , sizeof(reDP)
113        write(*, '(a,i0)') '_____Decimales_de_precisión_real (DP):_' , precision(reDP)
114        write(*, '(a,i0)') '_____Rango_real (DP):_' , range(reDP)
115        write(*, '(a,i0)') '_____Mínimo_exponente_de_real (DP):_' , minexponent(reDP)
116        write(*,'(a,i0,/)' ) '_____Máximo_exponente_de_real (DP):_' , maxexponent(reDP)

```

```

117     else
118     write(*, '(a,/)' ) '_Reales_tipo_"DP"_NO_SOPORTADOS!!!'
119     end if
120
121     if (QP > 0) then
122     write(*, '(a)')      '___Reales_tipo_"QP"_soportados!!!'
123     write(*, '(a,i0)') '_____Número_KIND:_' , QP
124     write(*, '(a,i0)') '_____Número_de_Bytes_de_real (QP):_' , sizeof(reQP)
125     write(*, '(a,i0)') '_____Decimales_de_precisión_real (QP):_' , precision(reQP)
126     write(*, '(a,i0)') '_____Rango_real (QP):_' , range(reQP)
127     write(*, '(a,i0)') '_____Mínimo_exponente_de_real (QP):_' , minexponent(reQP)
128     write(*, '(a,i0)') '_____Máximo_exponente_de_real (QP):_' , maxexponent(reQP)
129     else
130     write(*, '(a)')      '_Reales_tipo_"QP"_NO_SOPORTADOS!!!'
131     end if
132
133     write(*,11)
134
135
136     10 format (/, '-----', /, &
137     '_____DIAGNÓSTICO_NUMKIND_____ ', /, &
138     '-----')
139     11 format ('-----')
140
141     return
142 end subroutine ImprimeKind
143 !-----!
144
145 end module NumKind
146 !-----!

```

```

EsferaG.f08

1  !-----!
2  !Función EsferaG [Nava–Flores, 2017]
3  !
4  !Útil para calcular el potencial, componentes vectoriales y tensoriales debidas a una esfera con
5  !densidad constante en un punto de observación.
6  !
7  !Parámetros de entrada:
8  ! – Coordenadas del punto de observación: x0,y0,z0
9  ! – Coordenadas del centro de la esfera: xc,yc,zc
10 ! – Radio de la esfera: a
11 ! – Contraste de densidad de la esfera: Rho
12 ! – Cantidad gravitacional por calcular: Comp
13 ! Potencial: Comp = 'Pot'
14 ! Componentes vectoriales: Comp = 'GrX', 'GrY' o 'GrZ'
15 ! Componentes tensoriales: Comp = 'Txx', 'Txy', 'Tzx', 'Tyy', 'Tyz' o 'Tzz'
16 !
17 !Unidades consideradas:
18 ! – Longitud: [km]
19 ! – Densidad: [kg/m**3]
20 ! – Potencial Gravitacional: [mGal*m]
21 ! – Vector de Atracción Gravitacional: [mGal]
22 ! – Tensor de Gradiometría Gravitacional: [Eötvös] (1 [E] = 0.1 [mGal/km])
23 !
24 !Última modificación: Enero de 2018
25 !-----!
26 function EsferaG(x0,y0,z0,xc,yc,zc,a,Rho,Comp)
27
28     use NumKind
29
30     implicit none
31     character (len=3), intent(in):: Comp
32     real (DP), intent(in):: x0,y0,z0,xc,yc,zc,a,Rho
33     real (DP):: k,R
34     real (DP), parameter:: gama=6.67e-11_DP, pi=4._DP*atan(1._DP), si2mg=1.e5_DP, &
35         km2m=1.e3_DP, mGkm2E=10._DP, eps=1.e-15_DP
36     real (DP):: EsferaG
37
38     k=(4._DP*pi*gama*Rho*a**3)/3._DP
39     R=sqrt((x0-xc)**2 + (y0-yc)**2 + (z0-zc)**2)
40
41     if (R < eps) then
42         print *, ('Argumento_erróneo_detectado!!!_EsferaG')
43         stop
44     end if
45
46     EsferaG=0._DP
47
48     if (Comp=='Pot') then
49         EsferaG=(k/R)*si2mg*km2m
50     elseif (Comp=='GrX') then
51         EsferaG=(-k*(x0-xc)/(R**3))*si2mg*km2m
52     elseif (Comp=='GrY') then
53         EsferaG=(-k*(y0-yc)/(R**3))*si2mg*km2m
54     elseif (Comp=='GrZ') then
55         EsferaG=(-k*(z0-zc)/(R**3))*si2mg*km2m
56     elseif (Comp=='Txx') then
57         EsferaG=(k*(2._DP*(x0-xc)**2-((y0-yc)**2+(z0-zc)**2))/(R**5))*si2mg*km2m*mGkm2E

```



```
58  elseif (Comp=='Txy') then
59      EsferaG=(3._DP*k*((x0-xc)*(y0-yc))/(R**5))*si2mg*km2m*mGkm2E
60  elseif (Comp=='Txz') then
61      EsferaG=(3._DP*k*((x0-xc)*(z0-zc))/(R**5))*si2mg*km2m*mGkm2E
62  elseif (Comp=='Tyy') then
63      EsferaG=(k*(2._DP*(y0-yc)**2-((x0-xc)**2+(z0-zc)**2))/(R**5))*si2mg*km2m*mGkm2E
64  elseif (Comp=='Tyz') then
65      EsferaG=(3._DP*k*((y0-yc)*(z0-zc))/(R**5))*si2mg*km2m*mGkm2E
66  elseif (Comp=='Tzz') then
67      EsferaG=(k*(2._DP*(z0-zc)**2-((x0-xc)**2+(y0-yc)**2))/(R**5))*si2mg*km2m*mGkm2E
68  end if
69
70  return
71 end function EsferaG
72 !-----!
```

```

PotenGBox.f08

1  !-----!
2  !Función PotenGBox [Nava–Flores, 2015]
3  !– Basada en el trabajo de Nagy et al (2000) y gbox [Blakely, 1996].
4  !
5  !Calcula el potencial gravimétrico de un prisma rectangular, observado en un punto.
6  !
7  !El prisma considerado se encuentra en posición paralela con respecto a los ejes X y Y.
8  !
9  !Parámetros de entrada:
10 !– Coordenadas del punto de observación (sistema fijo): x0,y0,z0
11 !– El prisma se extiende desde (x1,y1,z1) hasta (x2,y2,z2)
12 !– Densidad del prisma:
13 !   Rho
14 !
15 !Unidades consideradas:
16 !   – Longitud: [km]
17 !   – Densidad: [kg/m^3]
18 !   – Potencial Gravimétrico: [mGal*m]
19 !
20 !Última modificación: Enero de 2018
21 !-----!
22 function PotenGBox(x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho)
23
24     use NumKind
25
26     implicit none
27     integer (IL) :: i,j,k,signo(2)
28     real (DP), intent (in) :: x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho
29     real (DP) :: x(2),y(2),z(2),rij,k,ijk,arg1,arg2,arg3,arg4,arg5,arg6,suma
30     real (DP), parameter :: gama=6.674e-11_DP, twopi=8._DP*atan(1._DP), si2mg=1.e5_DP, &
31         km2m=1.e3_DP
32     real (DP) :: PotenGBox
33
34     signo(1)=-1_IL
35     signo(2)=1_IL
36
37     x(1)=x0-x1
38     x(2)=x0-x2
39     y(1)=y0-y1
40     y(2)=y0-y2
41     z(1)=z0-z1
42     z(2)=z0-z2
43
44     suma=0._DP
45     do i=1,2
46         do j=1,2
47             do k=1,2
48                 rij=dsqrt(x(i)**2+y(j)**2+z(k)**2)
49                 ijk=signo(i)*signo(j)*signo(k)
50                 arg4=atan2((y(j)*z(k)),(x(i)*rij))
51                 arg5=atan2((z(k)*x(i)),(y(j)*rij))
52                 arg6=atan2((x(i)*y(j)),(z(k)*rij))
53                 if (arg4<0._DP) arg4=arg4+twopi
54                 if (arg5<0._DP) arg5=arg5+twopi
55                 if (arg6<0._DP) arg6=arg6+twopi
56                 arg1=rij+z(k)
57                 arg2=rij+x(i)

```

```
58     arg3=rij+k+y(j)
59     if (arg1<=0._DP) print *, ('Error_en_el_punto!!!_PotenGBox')
60     arg1=dlog(arg1)
61     if (arg2<=0._DP) print *, ('Error_en_el_punto!!!_PotenGBox')
62     arg2=dlog(arg2)
63     if (arg3<=0._DP) print *, ('Error_en_el_punto!!!_PotenGBox')
64     arg3=dlog(arg3)
65     suma=suma+ijk*((x(i)**2)*arg4/2._DP + (y(j)**2)*arg5/2._DP + &
66                 (z(k)**2)*arg6/2._DP - x(i)*y(j)*arg1 - &
67                 y(j)*z(k)*arg2 - z(k)*x(i)*arg3)
68     end do
69     end do
70     end do
71     PotenGBox=Rho*gama*suma*si2mg*km2m
72
73     return
74 end function PotenGBox
75 !-----!
```

```

VectorGBox.f08

1  !-----!
2  !Función VectorGBox [Nava–Flores, 2014]
3  !– Basada en el trabajo de Nagy et al (2000) y gbox [Blakely, 1996].
4  !
5  !Calcula las componentes el vector gravimetrico de un prisma rectangular, observado en un punto.
6  !
7  !El prisma considerado se encuentra en posición paralela con respecto a los ejes X, Y y Z.
8  !
9  !Parámetros de entrada:
10 !– Coordenadas del punto de observación (sistema fijo): x0,y0,z0
11 !– El prisma se extiende desde (x1,y1,z1) hasta (x2,y2,z2)
12 !– Densidad del prisma: Rho
13 !– Componente gravimétrica por calcular: Gx, Gy, Gz
14 !
15 !Unidades consideradas:
16 !   – Longitud: [km]
17 !   – Densidad: [kg/m^3]
18 !   – Anomalía gravimétrica: [mGal]
19 !
20 !Última modificación: Enero de 2018
21 !-----!
22 function VectorGBox(x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho,Gcomp)
23
24     use NumKind
25
26     implicit none
27     character (len=2), intent (in):: Gcomp
28     integer (IL):: i,j,k
29     real (DP), intent (in):: x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho
30     real (DP):: x(2),y(2),z(2),rij,signo(2),ijk,signo(2),ijk,arg1,arg2,arg3,suma
31     real (DP), parameter:: gama=6.674e-11_DP, twopi=8._DP*atan(1._DP), si2mg=1.e5_DP, &
32         km2m=1.e3_DP
33     real (DP):: VectorGBox
34
35     signo(1)=-1._DP
36     signo(2)=1._DP
37     x=0._DP
38     y=0._DP
39     z=0._DP
40
41     if (Gcomp=='Gx') then
42         x(1)=y0-y1;      x(2)=y0-y2
43         y(1)=z0-z1;      y(2)=z0-z2
44         z(1)=x0-x1;      z(2)=x0-x2
45     else if (Gcomp=='Gy') then
46         x(1)=z0-z1;      x(2)=z0-z2
47         y(1)=x0-x1;      y(2)=x0-x2
48         z(1)=y0-y1;      z(2)=y0-y2
49     else if (Gcomp=='Gz') then
50         x(1)=x0-x1;      x(2)=x0-x2
51         y(1)=y0-y1;      y(2)=y0-y2
52         z(1)=z0-z1;      z(2)=z0-z2
53     end if
54
55     if (abs(z(2))-abs(z(1)) < 0._DP) z=-z
56
57     suma=0._DP

```

```
58  do i=1,2
59      do j=1,2
60          do k=1,2
61              rijk=sqrt(x(i)**2+y(j)**2+z(k)**2)
62              ijk=signo(i)*signo(j)*signo(k)
63              arg1=atan2((x(i)*y(j)),(z(k)*rijk))
64              if (arg1<0._DP) arg1=arg1+twopi
65              arg2=rijk+y(j)
66              arg3=rijk+x(i)
67              if (arg2<=0._DP) print *, ('Error_en_el_punto!!!_VectorGBox')
68              arg2=log(arg2)
69              if (arg3<=0._DP) print *, ('Error_en_el_punto!!!_VectorGBox')
70              arg3=log(arg3)
71              suma=suma+ijk*(z(k)*arg1-x(i)*arg2-y(j)*arg3)
72          end do
73      end do
74  end do
75  VectorGBox=Rho*gama*suma*si2mg*km2m
76
77  return
78 end function VectorGBox
79 !-----!
```

```

TensorGBox.f08

1  !-----!
2  !Función TensorGBox [Nava–Flores, 2014]
3  !– Basada en el trabajo de Nagy et al (2000) y gbox [Blakely, 1996].
4  !
5  !Calcula las componentes del tensor gravimétrico, observado en un punto.
6  !
7  !El prisma considerado se encuentra fijo con respecto a los ejes X y Y.
8  !
9  !Parámetros de entrada de la función TensorGBox:
10 !– Coordenadas del punto de observación (sistema fijo): x0,y0,z0
11 !– El prisma se extiende desde (x1,y1,z1) hasta (x2,y2,z2)
12 !– Densidad del prisma:
13 !   Rho
14 !– Componente tensorial requerida: Txx, Tyy, Tzz, Txy, Txz, Tyz
15 !
16 !Unidades consideradas:
17 !   – Longitud: [km]
18 !   – Densidad: [kg/m^3]
19 !   – Tensor gravimétrico: [Evs] (1 [E] = 0.1 [mGal/km])
20 !
21 !Última modificación: Enero de 2018
22 !-----!
23 function TensorGBox(x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho,T)
24
25     implicit none
26     character (len=3), intent (in):: T
27     integer (IL):: i,j,k,signo(2)
28     real (DP), intent (in):: x0,y0,z0,x1,y1,z1,x2,y2,z2,Rho
29     real (DP):: x(2),y(2),z(2),rij,k,ijk,arg1,arg2,arg3,arg4,arg5,arg6,suma
30     real (DP), parameter:: gama=6.674e-11_DP, twopi=8._DP*atan(1._DP), si2mg=1.e5_DP, &
31         km2m=1.e3_DP, mGkm2E=10._DP
32     real (DP):: TensorGBox
33
34     TensorGBox=0._DP
35
36     signo(1)=-1_IL
37     signo(2)=1_IL
38
39     x(1)=x0-x1
40     x(2)=x0-x2
41     y(1)=y0-y1
42     y(2)=y0-y2
43     z(1)=z0-z1
44     z(2)=z0-z2
45
46     suma=0._DP
47     do i=1,2
48         do j=1,2
49             do k=1,2
50                 rijk=sqrt(x(i)**2+y(j)**2+z(k)**2)
51                 ijk=signo(i)*signo(j)*signo(k)
52                 select case (T)
53                     case ('Txx')
54                         arg4=atan2((y(j)*z(k)),(x(i)*rijk))
55                         if (arg4<0._DP) arg4=arg4+twopi
56                         suma=suma+ijk*arg4
57                     case ('Txy')

```

```

58         arg3=rijk+z(k)
59         if (arg3<=0._DP) print *, ('Error_en_el_punto!!!_TensorGBox')
60         arg3=log(arg3)
61         suma=suma+ijk*(-arg3)
62         case('Tzx')
63         arg1=rijk+y(j)
64         if (arg1<=0._DP) print *, ('Error_en_el_punto!!!_TensorGBox')
65         arg1=log(arg1)
66         suma=suma+ijk*(-arg1)
67         case('Tyy')
68         arg5=atan2((z(k)*x(i)),(y(j)*rijk))
69         if (arg5<0._DP) arg5=arg5+twopi
70         suma=suma+ijk*arg5
71         case('Tyz')
72         arg2=rijk+x(i)
73         if (arg2<=0._DP) print *, ('Error_en_el_punto!!!_TensorGBox')
74         arg2=log(arg2)
75         suma=suma+ijk*(-arg2)
76         case('Tzz')
77         arg6=atan2((x(i)*y(j)),(z(k)*rijk))
78         if (arg6<0._DP) arg6=arg6+twopi
79         suma=suma+ijk*arg6
80     end select
81 end do
82 end do
83 end do
84     TensorGBox=Rho*gama*suma*si2mg*km2m*mGkm2E
85
86     return
87 end function TensorGBox
88 !-----!

```

COMPLEMENTO DE PROCESAMIENTO DE DATOS GRAVIMÉTRICOS

En este anexo se incluyen todas las mallas procesadas de datos del modelo sintético G1, cuyos resultados fueron reportados en el capítulo 3. También se incluye información complementaria de cada método aplicado y definido en el mismo capítulo, así como algunos comentarios relacionados con los resultados obtenidos.

C.1 Procesos aplicados a mallas de anomalía gravimétrica

Gradiente Horizontal

Reescribiendo la definición del gradiente horizontal (Ec. 2.45) aplicado a una malla regular de datos de anomalía gravimétrica $g(x, y)$:

$$GH(x, y) = \sqrt{\left[\frac{\partial}{\partial x}g(x, y)\right]^2 + \left[\frac{\partial}{\partial y}g(x, y)\right]^2}$$

Las mallas de derivadas direccionales: $\frac{\partial}{\partial x}g(x, y)$ y $\frac{\partial}{\partial y}g(x, y)$ se pueden aproximar en el dominio espacial a través de operadores de diferencias centradas en el interior de la malla y diferencias hacia adelante y atrás en las fronteras. A continuación se muestran los operadores de segundo orden para la aproximación de las derivadas:

Diferencias centradas:

$$\frac{\partial}{\partial x}g(x, y) \approx \frac{g(x + \Delta x, y) - g(x - \Delta x, y)}{2\Delta x}$$

$$\frac{\partial}{\partial y}g(x, y) \approx \frac{g(x, y + \Delta y) - g(x, y - \Delta y)}{2\Delta y}$$

Diferencias hacia adelante:

$$\frac{\partial}{\partial x}g(x, y) \approx \frac{4g(x + \Delta x, y) - g(x + 2\Delta x, y) - 3g(x, y)}{2\Delta x}$$

$$\frac{\partial}{\partial y}g(x, y) \approx \frac{4g(x, y + \Delta y) - g(x, y + 2\Delta y) - 3g(x, y)}{2\Delta y}$$

Diferencias hacia atrás:

$$\frac{\partial}{\partial x}g(x, y) \approx \frac{g(x - 2\Delta x, y) + 3g(x, y) - 4g(x - \Delta x, y)}{2\Delta x}$$

$$\frac{\partial}{\partial y}g(x, y) \approx \frac{g(x, y - 2\Delta y) + 3g(x, y) - 4g(x, y - \Delta y)}{2\Delta y}$$

(Δx y Δy son los intervalos de muestreo de la malla de anomalía)

Si se prefiere trabajar en el dominio del número de onda, estas mallas se pueden calcular a través de la propiedad de derivación de la Transformada de Fourier:

$$\frac{\partial^n}{\partial x^n}g(x, y) = \mathcal{F}^{-1}\{(ip)^n G(p, q)\}$$

$$\frac{\partial^n}{\partial y^n}g(x, y) = \mathcal{F}^{-1}\{(iq)^n G(p, q)\}$$

donde:

$$i = \sqrt{-1}$$

$$G(p, q) = \mathcal{F}\{g(x, y)\} \text{ (Transformada de Fourier 2D)}$$

$$p = 2\pi k_x \text{ (números de onda en dirección X)}$$

$$q = 2\pi k_y \text{ (números de onda en dirección Y)}$$

En la Figura C.1 se muestran las mallas de gradiente horizontal de la componente G_z del modelo sintético **G1**, considerando datos libres de ruido, así como con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuados ascendentemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

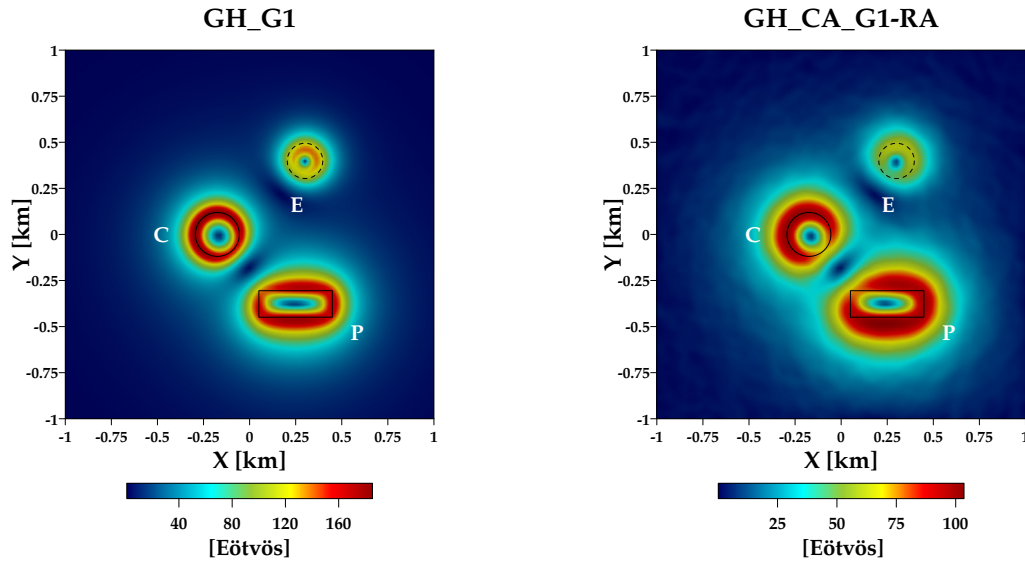


FIGURA C.1: Izquierda: Malla GH_G1, de gradiente horizontal de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla GH_CA_G1-RA, de gradiente horizontal de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

Tal como se muestra en la Figura C.1, el gradiente horizontal permite delimitar los bordes o contornos de las fuentes de campo P y C, aún en datos contaminados por ruido aleatorio, el cual se atenúa lo suficiente a través de la continuación ascendente aplicada previamente al cálculo del gradiente horizontal y debido al suavizamiento generado por este proceso de continuación, los máximos del gradiente horizontal también son suavizados y se alejan ligeramente hacia el exterior de los contornos de la fuente prismática (P), sin embargo, aún es posible interpretar una fuente de campo en esa zona.

Derivada Vertical

Está definida a través de la ecuación 2.46, reescrita a continuación:

$$\frac{\partial}{\partial z} g(x, y) = \mathcal{F}^{-1} \{ |k| G(p, q) \}$$

En la Figura C.2 se muestra la malla de gradiente vertical de la anomalía gravimétrica del modelo sintético G1, libre de ruido y con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuada ascendentemente 2 UM, mientras que en la Figura C.3 se muestra el gradiente horizontal de dichas mallas. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras E (esfera), P (prisma) y C (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

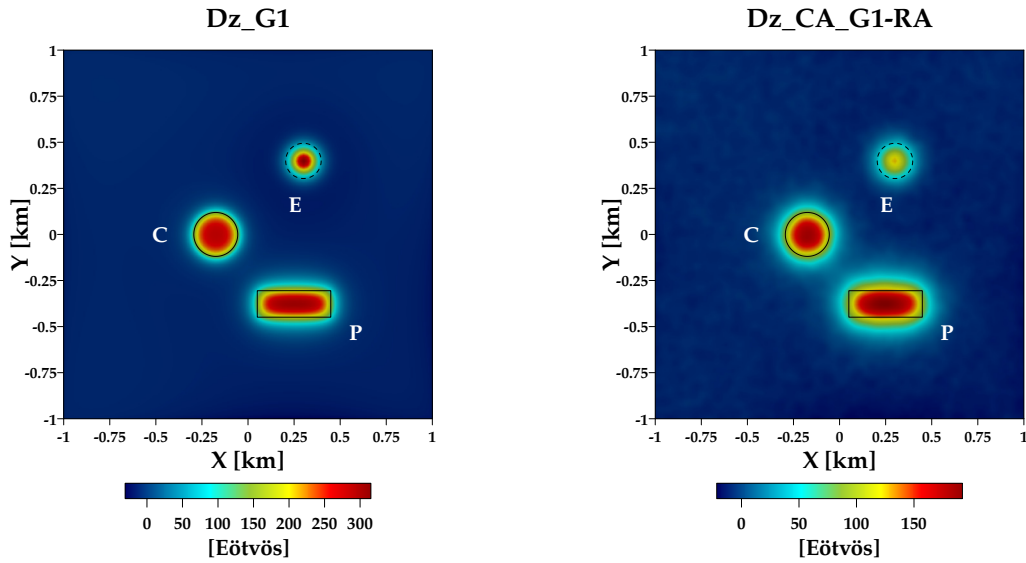


FIGURA C.2: **Izquierda:** Malla Dz_G1 , de gradiente vertical de la componente G_z del modelo sintético $G1$ libre de ruido. **Derecha:** Malla Dz_CA_G1-RA , de gradiente vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

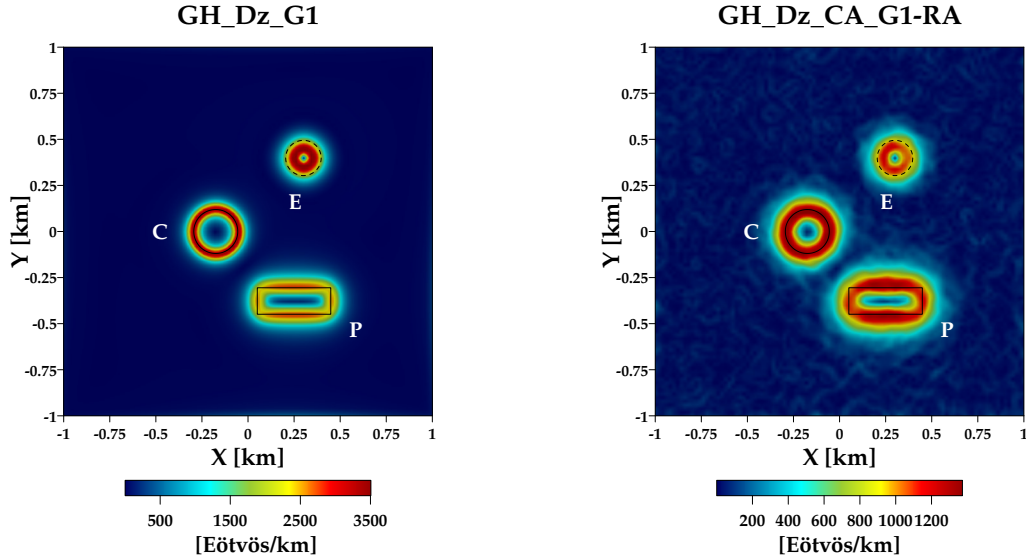


FIGURA C.3: **Izquierda:** Malla GH_Dz_G1 , de gradiente horizontal de la derivada vertical de la componente G_z del modelo sintético $G1$ libre de ruido. **Derecha:** Malla $GH_Dz_CA_G1-RA$, de gradiente horizontal de la derivada vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

Es notable la mejora en la definición individual de las anomalías después de la aplicación del gradiente vertical (Figura C.2), por lo cual, al aplicar métodos de realce de bordes como el gradiente horizontal a la malla de gradiente vertical (Figura C.3), los bordes de las estructuras serán más fácilmente discernibles que al aplicarse directamente en la malla de anomalía gravimétrica (Figura C.1), tal como fue reportado por Marson and Klingele (1993).

Amplitud de la Señal Analítica

La amplitud de la señal analítica 3D quedó representada a través de la ecuación 2.49, reescrita a continuación:

$$|A(x, y)| = \sqrt{\left[\frac{\partial\phi}{\partial x}\right]^2 + \left[\frac{\partial\phi}{\partial y}\right]^2 + \left[\frac{\partial\phi}{\partial z}\right]^2}$$

y la señal analítica mejorada, a través de la ecuación 2.50:

$$|A_n(x, y)| = \sqrt{\left[\frac{\partial^n}{\partial z^n} G_x\right]^2 + \left[\frac{\partial^n}{\partial z^n} G_y\right]^2 + \left[\frac{\partial^n}{\partial z^n} G_z\right]^2}$$

En las Figuras C.4 y C.5 se muestran las mallas de Amplitud de la Señal Analítica y Señal Analítica Mejorada de primer orden ($n=1$ en la expresión 2.50), aplicadas a la malla de anomalía gravimétrica del modelo **G1**, sin ruido y con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuada ascendientemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua.

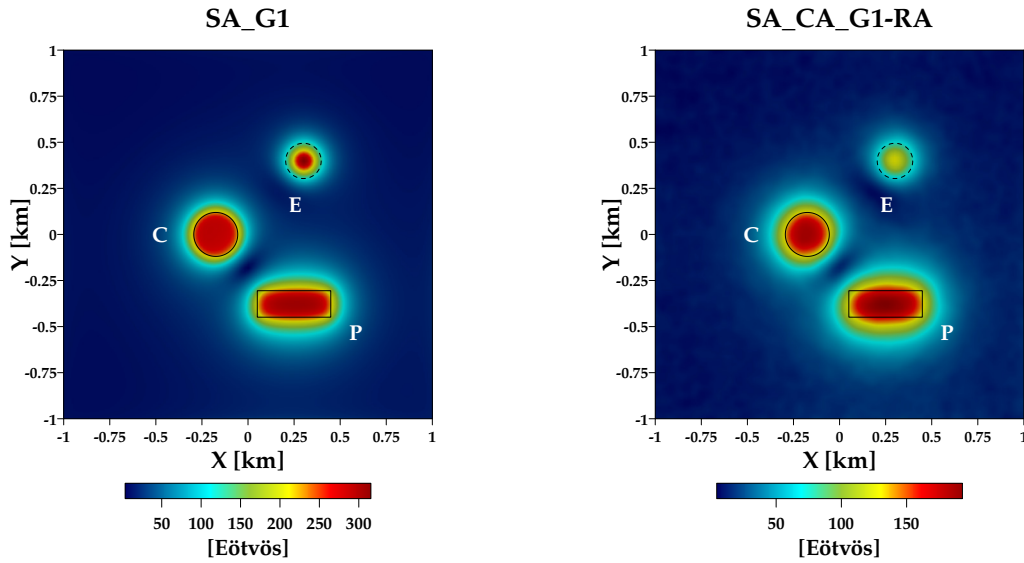


FIGURA C.4: Izquierda: Malla SA_G1, de señal analítica de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla SA_CA_G1-RA, de señal analítica de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

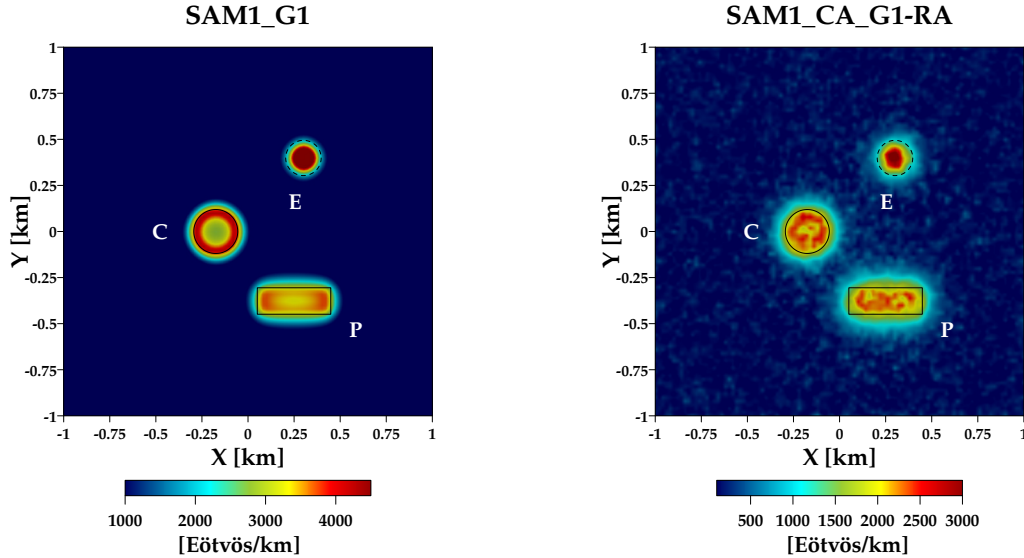


FIGURA C.5: Izquierda: Malla SAM1_G1, de señal analítica mejorada de primer orden de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla SAM1_CA_G1-RA, de señal analítica mejorada de primer orden de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

En la Figura C.5 se muestra una clara mejora en definición de los bordes de las fuentes del modelo **G1** por la aplicación de la señal analítica mejorada, con respecto a la señal analítica simple (Figura C.4), especialmente en los datos libres de ruido. En los datos contaminados por ruido, los máximos de la señal analítica mejorada no se ubican directamente sobre los bordes de los cuerpos **C** y **P**, aunque sí se puede interpretar claramente la presencia de fuentes con formas circular y prismática (cuerpos **C** y **P**, respectivamente), proyectadas en el plano de observación de datos.

Ángulo de Inclinación de Gradientes

Está definido a través de la ecuación 2.51, reescrita a continuación:

$$AIG(x, y) = \tan^{-1} \left[\frac{\frac{\partial f}{\partial z}}{\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}} \right]$$

En la Figura C.6 se muestra la malla del Ángulo de Inclinación de Gradientes de la derivada vertical de la anomalía gravimétrica del modelo **G1**, libre de ruido y con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuada ascendentemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera en línea discontinua. Adicionalmente, se marca el contorno cero de las mallas procesadas en línea punteada:

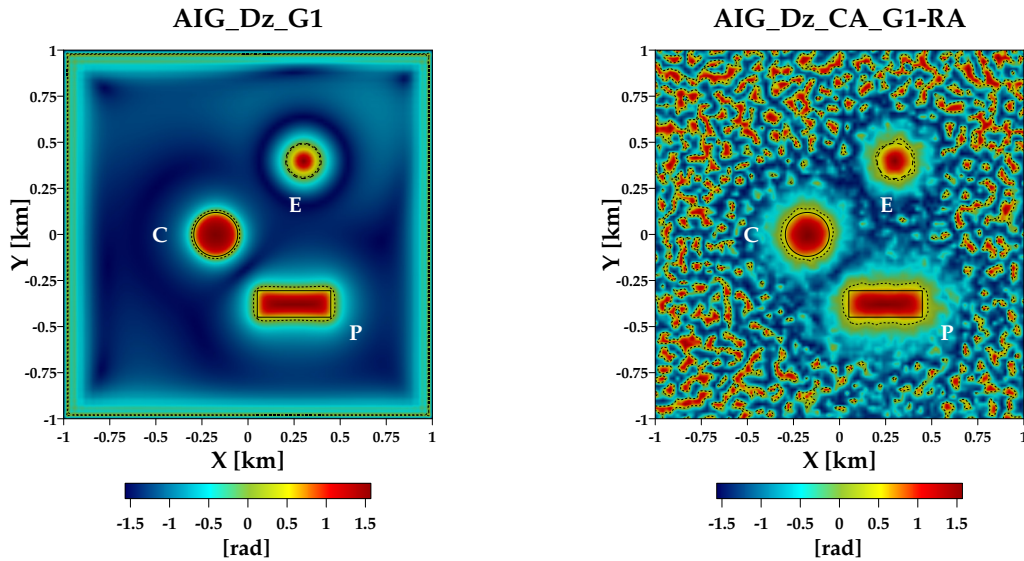


FIGURA C.6: Izquierda: Malla AIG_Dz_G1, de ángulo de inclinación de gradientes de la derivada vertical de la componente G_z del modelo sintético G1 libre de ruido. Derecha: Malla AIG_Dz_CA_G1-RA, de ángulo de inclinación de gradientes de la derivada vertical de la continuación ascendente de G_z con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro) en línea continua, así como la proyección de la esfera (E) en línea discontinua y los contornos cero de las mallas procesadas en línea punteada.

En la Figura C.6 se nota claramente que el contorno cero de las mallas procesadas se aproxima muy bien a los bordes de las fuentes, tanto en datos libres de ruido como en los datos contaminados por ruido aleatorio, en los que el efecto ecualizador del operador ángulo de inclinación de gradientes ha realizado el ruido aleatorio notablemente y esto podría causar errores en la interpretación, sin embargo, la formación de una zona de mínimos que bordea las fuentes a modo de “halo”, se podría utilizar como guía en la interpretación para descartar las fuentes espurias generadas por la aplicación de este proceso.

C.2 Procesos aplicados a mallas de gradiometría gravimétrica

Invariantes

Reescribiendo las ecuaciones que permiten obtener las mallas de invariantes que reflejan la forma de las fuentes que causan las anomalías del *FTGG* (ecuaciones 2.53 e 2.54):

$$I_1 = T_{xx}T_{yy} + T_{yy}T_{zz} + T_{xx}T_{zz} - (T_{xy}^2 + T_{yz}^2 + T_{xz}^2)$$

$$I_2 = \det(T_{ij})$$

$$= T_{xx}(T_{yy}T_{zz} - T_{yz}^2) + T_{xy}(T_{yz}T_{xz} - T_{xy}T_{zz}) + T_{xz}(T_{xy}T_{yz} - T_{xz}T_{yy})$$

En la Figuras C.7 y C.8 se muestran mallas de valor absoluto del invariante I_1 y mallas del invariante I_2 del *FTGG* del modelo **G1**, en datos libres de ruido y con ruido aleatorio añadido, $N(0, 0.05\sigma)$. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

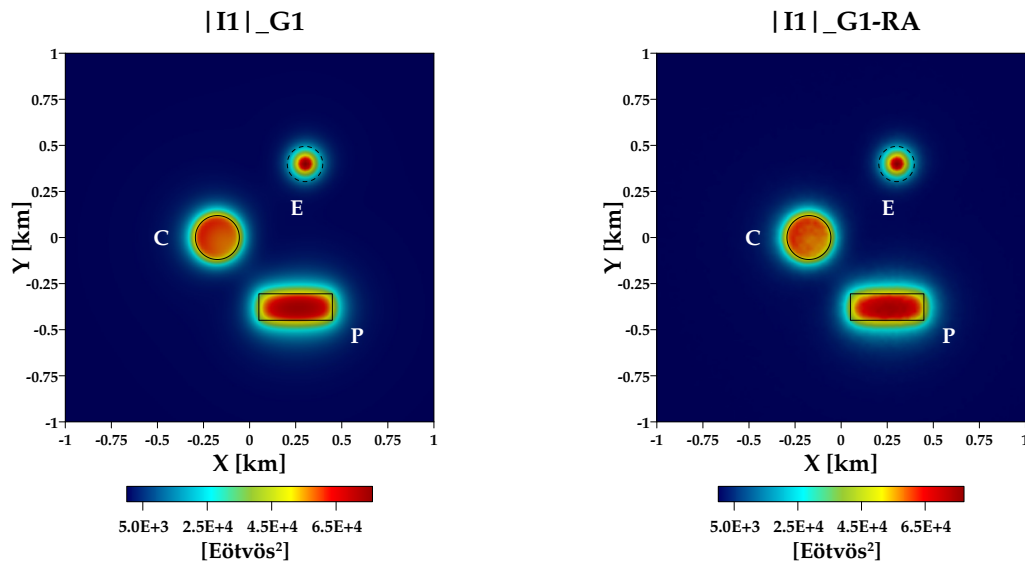


FIGURA C.7: Izquierda: Malla $|I1|_{G1}$, de valor absoluto del invariante I_1 del *FTGG* del modelo sintético **G1** libre de ruido. Derecha: Malla $|I1|_{G1-RA}$, de valor absoluto del invariante I_1 del *FTGG* con ruido aleatorio añadido. Se muestran los contornos de las fuentes **P** (prisma) y **C** (cilindro), así como la proyección de la esfera (**E**).

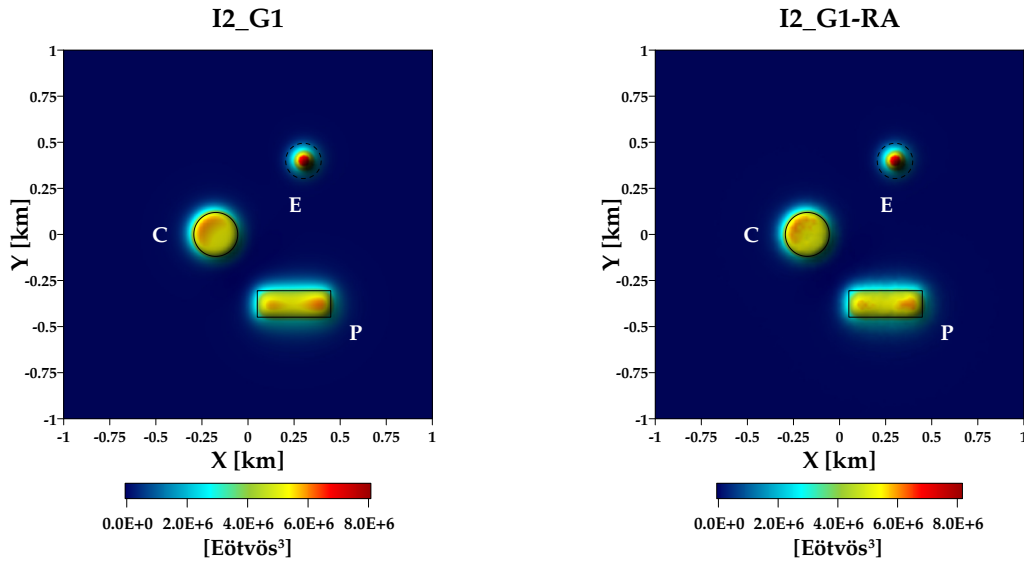


FIGURA C.8: Izquierda: Malla $I2_G1$, del invariante I_2 del $FTGG$ del modelo sintético $G1$ libre de ruido. Derecha: Malla $I2_G1-RA$, del invariante I_2 del $FTGG$ con ruido aleatorio añadido. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

Es notable la poca sensibilidad que tienen los invariantes I_1 e I_2 hacia el ruido aleatorio añadido, lo cual se puede deducir de las expresiones que los definen (Ecs. 2.53 y 2.54), debido a que no requieren de derivación direccional, proceso que amplifica componentes de longitud de onda corta, incluyendo el ruido. Esto se confirma a través de la inspección de las Figuras C.7 y C.8 mostradas, razón por la cual no fue necesario aplicar continuación ascendente a las mallas de las componentes tensoriales del modelo $G1$ con ruido añadido, previo al cálculo de estos invariantes.

Adicionalmente, en las Figuras C.7 y C.8 se refleja la utilidad práctica del uso conjunto de los invariantes I_1 e I_2 en la interpretación de fuentes de campo, debido a que, mientras que el invariante I_1 enfatiza las regiones del área de estudio bajo las que existe un contraste de densidad marcado, el invariante I_2 tiende a resaltar la forma de la cima de las fuentes de campo que exhiben ese contraste de densidad, con respecto al medio circundante, tal como fue reportado por Murphy and Brewster (2007).

Señales Analíticas Direccionales y Función Detectora de Bordes

Las ecuaciones que permiten obtener las mallas de señales analíticas direccionales (2.56 a 2.58) son reescritas a continuación:

$$A_x(x, y, z) = \sqrt{(T_{xx})^2 + (T_{xy})^2 + (T_{xz})^2}$$

$$A_y(x, y, z) = \sqrt{(T_{xy})^2 + (T_{yy})^2 + (T_{yz})^2}$$

$$A_z(x, y, z) = \sqrt{(T_{xz})^2 + (T_{yz})^2 + (T_{zz})^2}$$

y la función detectora de bordes (Ec. 2.59) es:

$$|DB| = \sqrt{|A_{x,z}|^2 + |A_{y,z}|^2}$$

En la Figura C.9 se muestran las mallas de las señales analíticas direccionales del *FTGG* del modelo **G1** en datos libres de ruido, mientras que en la Figura C.10 se muestran las señales analíticas direccionales de las mallas con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuadas ascendientemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

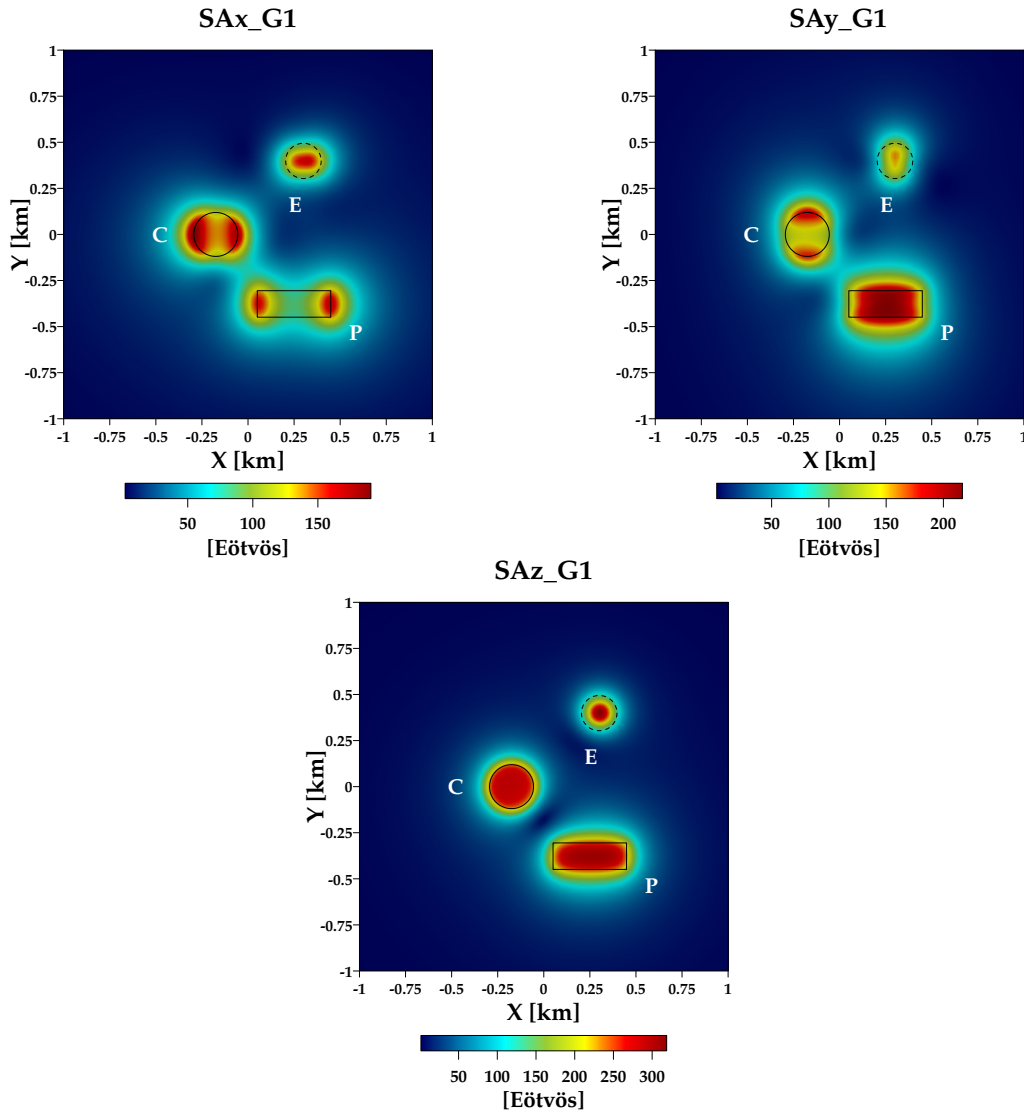


FIGURA C.9: Mallas de señal analítica direccional en X , Y y Z del $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla SAx_G1 , de señal analítica direccional en X . **Derecha:** Malla SAy_G1 , de señal analítica direccional en Y . **Abajo:** Malla SAz_G1 , de señal analítica direccional en Z . Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

La Figura C.9 muestra que los máximos de las señales analíticas direccionales en X y Y se ubican directamente sobre los bordes de los cuerpos C y P (un poco difusos en este último cuerpo), mientras que en el cuerpo E , no se correlacionan con el centro de la esfera. Por otro lado, la señal analítica direccional en Z tiene un máximo ubicado directamente sobre el centro de la esfera (E) y no forma máximos bien definidos sobre los bordes de los cuerpos C y P . Esto mismo aplica para el caso de la Figura C.10, en las que las mallas exhiben un suavizamiento resultante del proceso de continuación ascendente aplicado antes del cálculo de la señal analítica direccional.

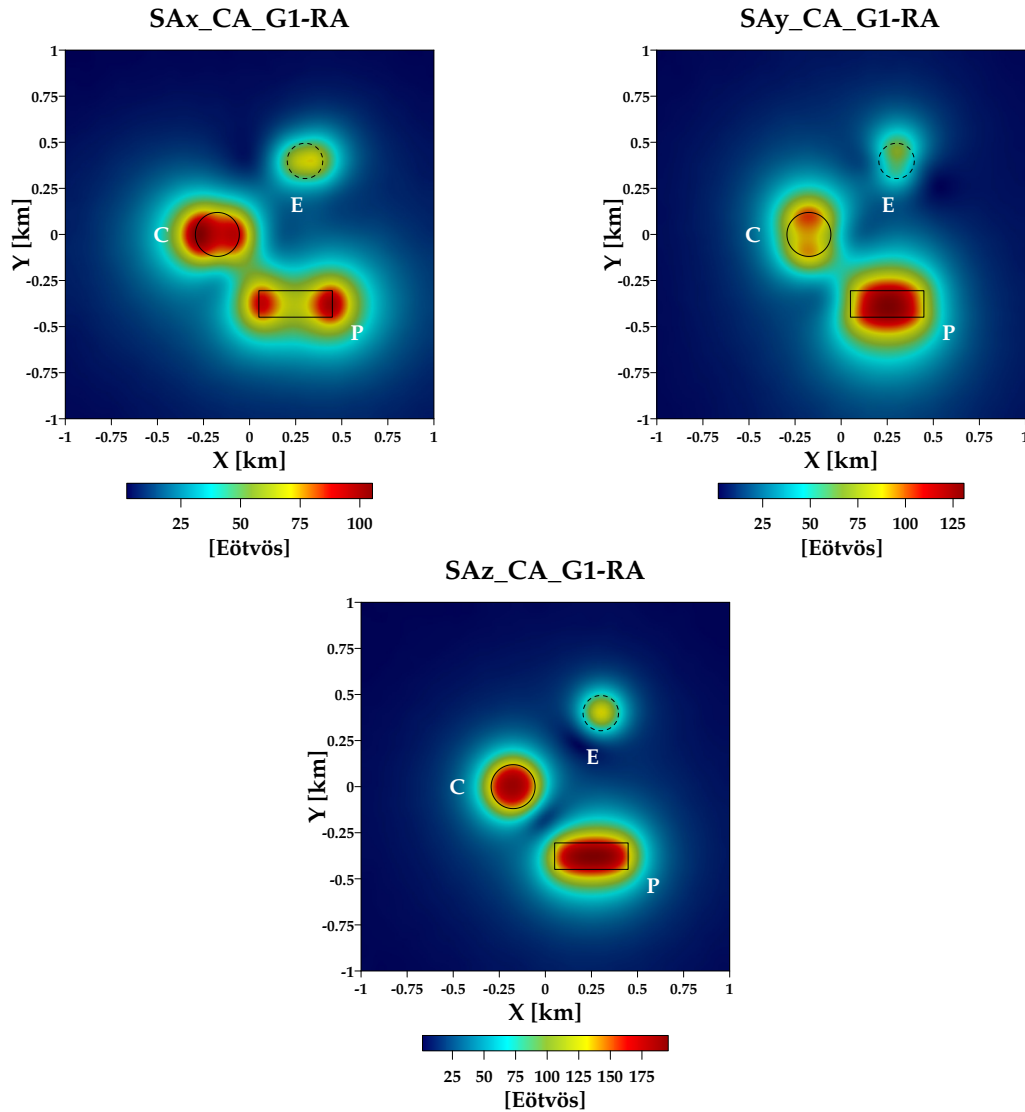


FIGURA C.10: Mallas de señal analítica direccional en X , Y y Z de la continuación ascendente del $FTGG$ del modelo sintético $G1$ con ruido aleatorio añadido. **Izquierda:** Malla SAx_CA_G1-RA , de señal analítica direccional en X . **Derecha:** Malla SAy_CA_G1-RA , de señal analítica direccional en Y . **Abajo:** Malla SAz_CA_G1-RA , de señal analítica direccional en Z . Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

Ahora, en la Figura C.11 se muestran las mallas de los gradientes verticales de las señales analíticas direccionales y la función detectora de bordes del $FTGG$ del modelo $G1$ en datos libres de ruido, mientras que en la Figura C.12 se muestran estos mismos procesos a mallas con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuadas ascendientemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras E (esfera), P (prisma) y C (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua:

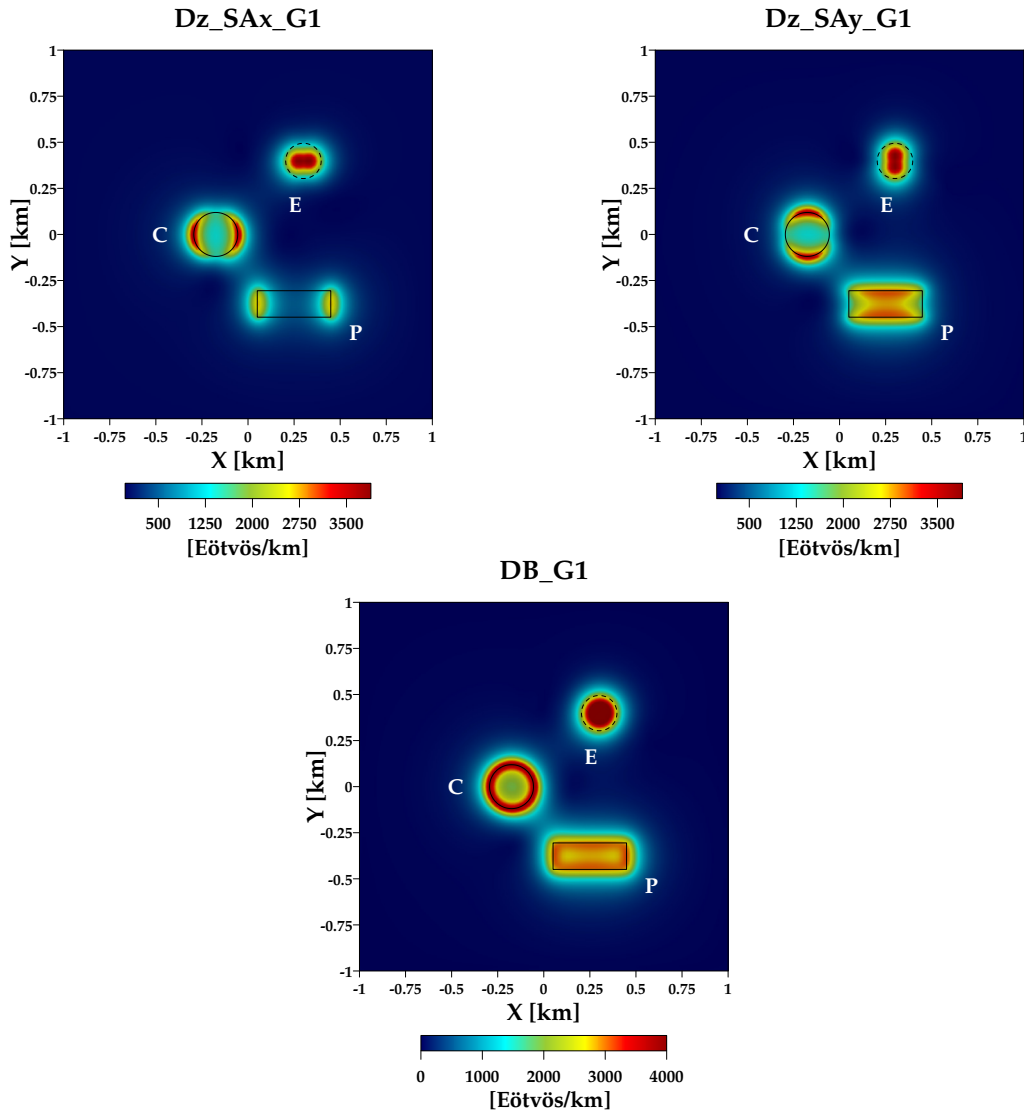


FIGURA C.11: Mallas de gradiente vertical de la señal analítica direccional en X y Y , así como de la función detectora de bordes del $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla Dz_SAx_G1 , de gradiente vertical de la señal analítica direccional en X . **Derecha:** Malla Dz_SAy_G1 , de gradiente vertical de la señal analítica direccional en Y . **Abajo:** Malla DB_G1 , de la función detectora de bordes. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

La Figura C.11 muestra la efectividad que tiene el uso de las mallas de gradiente vertical de la señal analítica direccional y la función detectora de bordes en el realce de los límites laterales de las fuentes y la ubicación de la esfera. Mientras que para los cuerpos C y P , los máximos del gradiente vertical de la señal analítica direccional y de la función detectora de bordes se ubican sobre sus bordes, para el cuerpo E , los máximos del gradiente vertical de la señal analítica direccional se ubican alrededor del centro de la esfera, donde la función detectora de bordes forma su máximo.

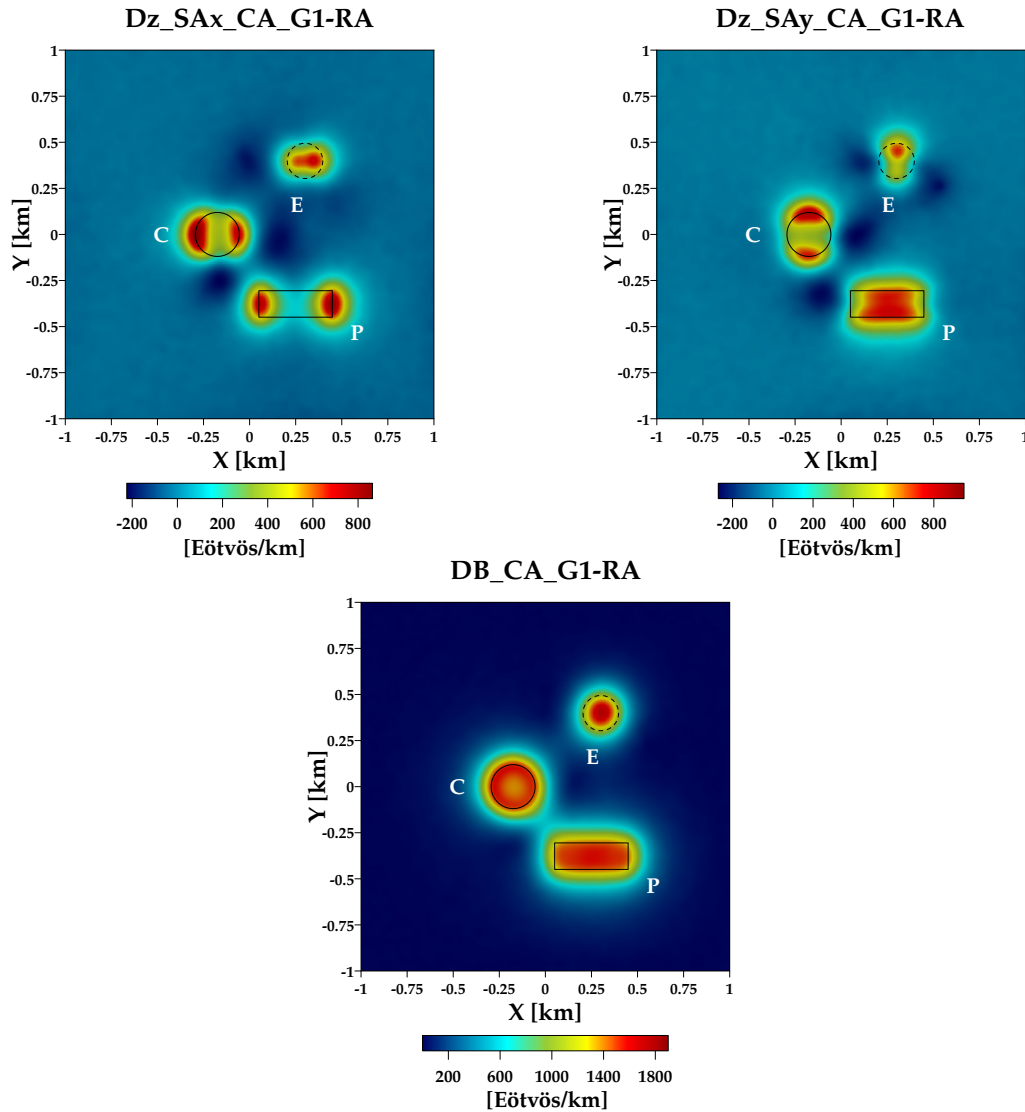


FIGURA C.12: Mallas del gradiente vertical de la señal analítica direccional en X y Y , así como de la función detectora de bordes de la continuación ascendente $FTGG$ del modelo sintético $G1$ libre de ruido. **Izquierda:** Malla $Dz_SAx_CA_G1-RA$, de gradiente vertical de la señal analítica direccional en X . **Derecha:** Malla $Dz_SAy_CA_G1-RA$, de gradiente vertical de la señal analítica direccional en Y . **Abajo:** Malla DB_G1-RA , de la función detectora de bordes. Se muestran los contornos de las fuentes P (prisma) y C (cilindro), así como la proyección de la esfera (E).

La aplicación de los gradientes verticales de las señales analíticas y la función detectora de bordes a las mallas de componentes tensoriales continuadas ascendentemente (Figura C.12), demuestra que estos procesos realzan los bordes de los cuerpos C y P , así como el centro de la esfera (E) lo suficiente como para permitir su interpretación, a pesar de exhibirse de una forma relativamente más suavizada que los resultados en mallas sin continuar (Figura C.11).

Función Detectora de Bordes Mejorada

La ecuación que describe este proceso está dada por la ecuación 2.60, reescrita a continuación:

$$DBM = \frac{1}{2} \left[T_{xx}g_z + T_{yy}g_z + \sqrt{(T_{xx}g_z - T_{yy}g_z)^2 + 4(T_{xy}g_z)^2} \right]$$

En la Figura C.13 se muestra la malla de la Función Detectora de Bordes Mejorada del *FTGG* del modelo **G1**, libre de ruido y con ruido aleatorio añadido, $N(0, 0.05\sigma)$, continuado ascendentemente 2 UM. Se marcan las proyecciones al plano horizontal de los contornos reales de las fuentes con líneas y se indican con las letras **E** (esfera), **P** (prisma) y **C** (cilindro). Los contornos del prisma y del cilindro se marcan con línea sólida y la extensión de la esfera, en línea discontinua. Adicionalmente, se marca el contorno cero de las mallas procesadas en línea punteada:

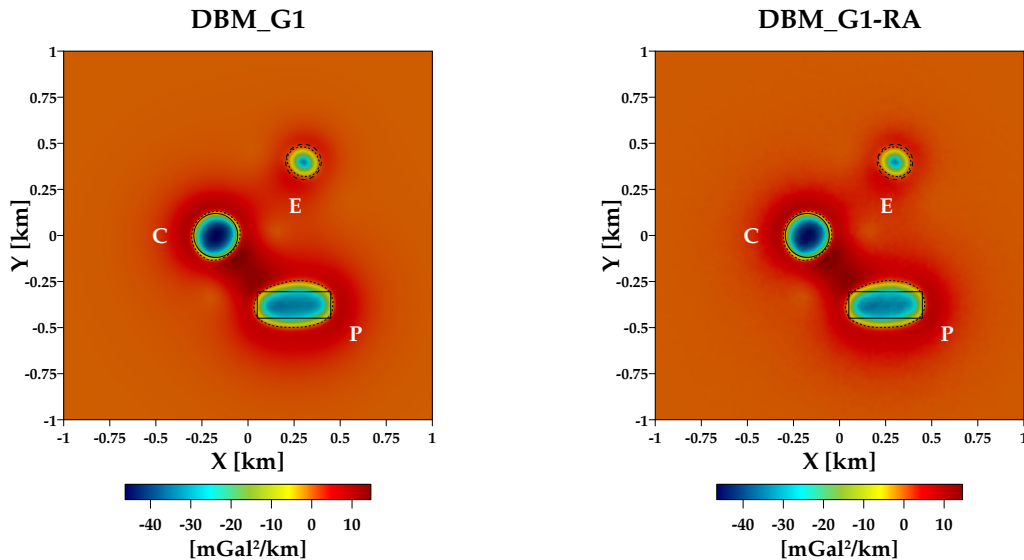


FIGURA C.13: Izquierda: Malla *DBM_G1*, de la función detectora de bordes mejorada del *FTGG* del modelo sintético **G1** libre de ruido. Derecha: Malla *DBM_G1-RA*, de la función detectora de bordes mejorada del *FTGG* con ruido aleatorio añadido. Se muestran los contornos de las fuentes **P** (prisma) y **C** (cilindro) en línea continua, así como la proyección de la esfera (**E**) en línea discontinua y los contornos cero de las mallas procesadas en línea punteada.

Del mismo modo que ocurre con los invariantes I_1 e I_2 , la función detectora de bordes mejorada presenta baja sensibilidad hacia el ruido aleatorio añadido, por lo que no fue necesario aplicar el proceso en mallas continuadas ascendentemente. Adicionalmente, en la Figura C.13 se muestra que los bordes detectados tienen una forma suavizada, causando que los bordes rectos del cuerpo \mathbf{P} no sean correctamente definidos por este proceso, sin embargo, a pesar de este efecto, los bordes identificados son ubicados correctamente, sobre las fuentes de campo.