



TESIS - RG 092509

**TENSOR MARUSSI DAN PEMODELAN GAYABERAT
INVERSI TIGA DIMENSI (3D) UNTUK
MEMBUKTIKAN PATAHAN WATUKOSEK DENGAN
MENGUNAKAN DATA EIGEN-6C4**

JAMHUR
3515203002

DOSEN PEMBIMBING
Dr. DWA DESA WARNANA
IRA MUTIARA ANJASMARA, ST, M.Phil, Ph.D

PROGRAM MAGISTER
BIDANG KEAHLIAN GEOFISIKA
JURUSAN TEKNIK GEOMATIKA
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

Tesis disusun untuk memenuhi salah satu syarat memperoleh
gelar
Magister Teknik (MT)
di
Institut Teknologi Sepuluh Nopember

Oleh :
Jamhur
NRP. 3515203002

Tanggal Ujian : 24 Juli 2017
Periode Wisuda : September

Disetujui oleh:

1. Dr. Dwa Desa Warnana
NIP: 19760123 200003 1001
(Pembimbing I)
2. Ira Mutiara Anjasmara, ST, M.Phil, Ph.D
NIP: 19781231 200212 2001
(Pembimbing II)
5. Dr. Ir. Muhammad Taufik
NIP: 19550919 198603 1001
(Penguji)
6. Dr. Ayi Syaeful Bahri, S.Si, MT
NIP: 19690906 199702 1001
(Penguji)



Dekan Fakultas Teknik Sipil dan Perencanaan,

Dr. Purwanita Setijanti, M.Sc., Ph.D

NIP. 19590427 198503 2001

“Halaman sengaja dikosongkan”

TENSOR MARUSSI DAN PEMODELAN GAYABERAT INVERSI TIGA DIMENSI (3D) UNTUK MEMBUKTIKAN PATAHAN WATUKOSEK DENGAN MENGGUNAKAN DATA EIGEN-6C4

Nama Mahasiswa : Jamhur
NRP : 3515203002
Dosen Pembimbing : 1. Dr. Dwa Desa Warnana
2. Ira Mutiara Anjasmara, ST, M.Phil, Ph.D.

ABSTRAK

Daerah Porong, Sidoarjo, adalah daerah semburan gunung api lumpur yang sering disebut LUSI (Lumpur Sidoarjo). Pada daerah tersebut terdapat kemungkinan adanya Patahan Watukosek yang membentang dari Gunung Penanggungan melewati pusat semburan LUSI sampai dengan Selat Madura. Penelitian ini bertujuan untuk membuktikan keberadaan patahan tersebut dengan menggunakan pemodelan gayaberat inversi tiga dimensi yang didukung dengan hasil interpretasi peta Tensor Marussi dari data EIGEN-6C4.

Model EIGEN-6C4 adalah model geopotensial global yang memiliki anomali yang bersifat regional sehingga dalam menginterpretasi struktur geologi hanya hanya pada zona dalam.

Pemodelan gayaberat inversi tiga dimensi dibuat dari data anomali bouguer regional yang menghasilkan model kontras densitas bawah permukaan yang memiliki struktur geologi perlipatan, *graben*, *half-graben*, dan patahan normal. sedangkan Tensor Marussi adalah turunan kedua dari *disturbing potential* dalam bentuk enam peta dua dimensi yang terdiri dari T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} , dan T_{zz} . Struktur geologi daerah Jawa Timur di-interpretasi berupa patahan geser *sinstral* dengan jurus timur laut. Patahan watukosek di-interpretasikan di sebelah timur laut Gunung Penanggungan mengarah ke timur laut namun tidak terdapat kemenerusan sampai ke daerah gunungapi lumpur Gunung Anyar.

Kata kunci: Gayaberat, EIGEN-6C4, Anomali Bouguer, Pemodelan Inversi Tiga Dimensi, Tensor Marussi.

“Halaman sengaja dikosongkan”

**MARUSSI TENSOR AND THREE-DIMENSIONAL (3D) GRAVITY
INVERSION MODELLING TO PROVE WATUKOSEK FAULT USING
EIGEN-6C4 DATA**

By : Jamhur
Student Identity Number : 3515203002
Supervisors : 1. Dr. Dwa Desa Warnana
2. Ira Mutiara Anjasmara, ST, M.Phil, Ph.D

ABSTRACT

Porong region, Sidoarjo, is the area of mud volcano eruption which often called LUSI (Lumpur Sidoarjo). In the area there is the possibility of Watukosek fault extending from Penanggungan Mountain passing through the center of LUSI towards Madura Strait. This study attempts to show the existence of that fracturing by three-dimensional gravity inversion modelling backed with the results of Marussi tensor map interpretation from EIGEN-6C4 data.

EIGEN-6C4 is a model of global geopotential containing regional anomaly so that the interpretation of geological structure is only been done in the deep zone.

Three-dimensional gravity inversion modelling is made from regional bouguer anomaly data that produced model of density contrast that has a geological structure of folding, grabens, half-grabens, and normal fault, while Tensor Marussi is a second derivative of disturbing potential that could describe the geological structures in the form of six two-dimensional maps consist of T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} , and T_{zz} . The geologic structures is interpreted as a sinistral shear fault with strike towards northeast. The watukosek fault is interpreted in the north-east of Mount Penanggungan towards the northeast but there is no continuity until of Gunung Anyar mud volcano.

Keywords: Gravity, EIGEN-6C4, Bouguer Anomaly, Three-Dimensional Inversion Modelling, Tensor Marussi

“Halaman sengaja dikosongkan”

KATA PENGANTAR

Puji dan rasa syukur mendalam penulis panjatkan ke hadirat Allah SWT, karena berkat limpahan rahmat, hidayah, dan inayah-Nya maka tesis ini dapat diselesaikan dengan baik. Salam dan salawat semoga selalu tercurah pada baginda Rasulullah Muhammad SAW.

Penulis mengucapkan rasa terimakasih yang sebesar-besarnya atas semua bantuan yang telah diberikan, baik secara langsung maupun tidak langsung selama penyusunan tesis ini hingga selesai. Ucapan terimakasih penulis sampaikan kepada:

1. Orangtua yang selalu memberi do'a dan dukungan moril maupun materil.
2. Istri yang selalu membantu dalam penulisan tesis ini.
3. Dosen wali, Dr. Widya Utama, DEA, yang selalu memberikan arahan dan bimbingan semasa perkuliahan hingga setelah lulus kuliah.
4. Dr. Dwa Desa Warnana dan Ira Mutiara Anjasmara, ST, M.Phil, Ph.D selaku dosen pembimbing yang telah memberikan bimbingan dan dorongan dalam penyusunan tesis ini.
5. Seluruh staf pengajar yang telah membimbing dan memberikan materi perkuliahan kepada penulis.
6. Seluruh pegawai yang ada di lingkungan jurusan Teknik Geomatika.
7. Seluruh sahabat yang selalu memberikan semangat dan dukungan.
8. Semua pihak yang tidak bisa disebutkan satu per satu.

Akhir kata penulis berharap semoga tesis ini dapat memberikan hal yang bermanfaat dan menambah wawasan bagi pembaca dan khususnya bagi penulis.

Surabaya, Juli 2017

Penulis

“Halaman sengaja dikosongkan”

DAFTAR ISI

	Hal
LEMBAR PENGESAHAN	i
ABSTRAK	iii
KATA PENGANTAR	v
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR SIMBOL	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	5
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	6
BAB II KAJIAN PUSTAKA	7
2.1. Tektonik dan Stratigrafi.....	7
2.2. Potensial Gayaberat.....	10
2.3. Gayaberat dari <i>Spherical Harmonics Expansion</i>	11
2.4. Gayaberat Normal	12
2.5. Koreksi Terhadap Gayaberat Normal.....	12
2.5.1. Koreksi Udara Bebas (KUB).....	12
2.5.2. Koreksi Bouguer Sederhana (<i>Bullard A</i>)	13
2.5.3. Koreksi <i>Bouguer Spherical Cap (Bullard B)</i>	13
2.5.4. Koreksi Medan (<i>Bullard C</i>)	15
2.6. Anomali Udara Bebas (AUB)	18
2.7. Anomali Bouguer Lengkap (ABL)	19
2.8. Pemisahan Anomali Regional dan Residual.....	19
2.8.1. Analisis Spektrum.....	19
2.8.2. Perata-rataan Bergerak (<i>Moving Average</i>)	20
2.9. Metodologi Inversi	21
2.10. Tensor Marussi.....	24
BAB III METODOLOGI PENELITIAN	27
3.1. Daerah Penelitian	27
3.2. Data penelitian.....	28
3.3. Diagram Alir Penelitian.....	29
3.4. Diagram Alir Pemodelan Inversi 3D.....	31
BAB IV HASIL DAN PEMBAHASAN	35
4.1. Geologi Regional.....	35
4.2. Peta Ketinggian, Gayaberat, dan Anomali Bouguer	35
4.3. Plot <i>Power Spectrum</i> dan <i>low-pass filter</i>	39
4.4. Pemodelan Gayaberat Inversi 3D.....	44
4.5. Peta Tensor Marussi	49
4.6. Struktur geologi hasil interpretasi pada daerah penelitian.....	57

BAB V KESIMPULAN DAN SARAN	59
DAFTAR PUSTAKA.....	61
LAMPIRAN	65

DAFTAR GAMBAR

Gambar 1.1	Kemungkinan patahan Watukosek (garis putih putus-putus) dan gunungapi lumpur yang ditandai dengan titik putih. (Mazzini, <i>et al.</i> , 2009).....	1
Gambar 1.2	Peta ketinggian Pulau Jawa bagian timur. Posisi beberapa gunungapi lumpur yang diketahui, gunungapi utama dan celah vulkanik (Mazinni, <i>et al.</i> , 2012).....	3
Gambar 2.1	Modifikasi dari Peta Geologi Lembar Jawa Bagian Timur (Gafoer & Ratman, 1999). Sistem perlipatan Kendeng yang memiliki <i>plunge</i> kearah timur tertutup oleh sedimen alluvial delta Brantas.....	8
Gambar 2.2	A) Penyederhanaan kerangka tektonik Asia Tenggara. B) Penyederhanaan kerangka tektonik Jawa Timur. (Panzeria, <i>et al.</i> , 2017).....	9
Gambar 2.3	Kolom stratigrafi di sumur BJP-1 (Mazinni <i>et al.</i> , 2007).....	9
Gambar 2.4	Potensial di titik P terhadap suatu massa (Blakely, 1995).....	10
Gambar 2.5	<i>Infinite Slab</i> pada koreksi Bouguer Sederhana (Lafer, 1991b).....	14
Gambar 2.6	<i>Spherical cap</i> pada koreksi <i>Bullard B</i> (Lafer, 1991b).....	14
Gambar 2.7	Grid regional dan ruas grid yang akan dihitung koreksinya pada zona 0, zona 1, dan zona 2. (Nagy, 1966).....	17
Gambar 2.8	(a) Koreksi medan pada Zona 0, (b) koreksi medan pada Zona 1 (Nagy, 1966), dan (c) koreksi medan pada zona 2 (Kane, 1962).....	18
Gambar 3.1	Area Penelitian (kotak merah). Patahan Watukosek berada pada daerah desa Watukosek, Kabupaten Pasuruan, Provinsi Jawa Timur, Indonesia.....	27
Gambar 3.2	Diagram alir penelitian.....	29
Gambar 3.3	Program GrafLab berbahasa MATLAB (Bucha & Janák, 2013)....	30
Gambar 3.4	Contoh proses analisis spectrum untuk lintasan AA' dengan menggunakan program MATLAB.....	31
Gambar 3.5	Diagram alir pemodelan inversi 3D.....	32

Gambar 3.6	GRAV3D <i>User Interface</i> . Versi untuk pendidikan (gratis) dengan keterbatasan jumlah data dan sel.....	33
Gambar 4.1	Peta geologi daerah penelitian hasil modifikasi dari Santosa <i>et al.</i> , 1992 dan Supardjono <i>et al.</i> , 1992.....	36
Gambar 4.2	Ketinggian ortometrik daerah penelitian dari data <i>EARTH2014</i> (Hirt and Rexer, 2015).....	37
Gambar 4.3	Peta gayaberat A) model <i>EGM2008</i> , dan B) model <i>EIGEN-C4</i> pada daerah yang sama.....	37
Gambar 4.4	Peta Gayaberat hasil model <i>EIGEN-6C4</i> pada daerah penelitian....	38
Gambar 4.5	Peta Anomali Udara Bebas (AUB) daerah penelitian.....	38
Gambar 4.6	Peta Anomali Bouguer Lengkap dan penampang AA', BB', CC', DD', dan EE' untuk analisis spektrum.....	39
Gambar 4.7	Plot $\ln(\text{power})$ terhadap bilangan gelombang untuk penampang AA' (a), BB' (b), CC' (c), DD' (d), dan EE' (e). garis merah adalah radien regional dan garis biru adalah gradien residual.....	41
Gambar 4.8	Peta Anomali Bouguer Regional.....	42
Gambar 4.9	Peta Anomali Bouguer Residual.....	43
Gambar 4.10	Daerah yang akan dibuat model gayaberat inversi 3D dan sayatan N-O, P-Q, R-S, T-U, dan V-W untuk interpretasi struktur geologi....	44
Gambar 4.11	Data masukan anomali Bouguer regional sebelum proses inversi.....	46
Gambar 4.12	Pengurangan error pada data masukan anomali Bouguer regional.....	46
Gambar 4.13	Data masukan setelah pengurangan error.....	47
Gambar 4.14	Hasil inversi pada kedalaman 900 m hingga 10900 m.....	47
Gambar 4.15	Sayatan pada penampang AA'.....	48
Gambar 4.16	Sayatan pada penampang BB'.....	48
Gambar 4.17	Sayatan pada penampang CC'.....	49
Gambar 4.18	Sayatan pada penampang DD'.....	49
Gambar 4.19	Sayatan pada penampang EE'.....	50

Gambar 4.20	Peta T_{xx} . Pola anomali berarah ke sumbu x-x atau utara sehingga struktur geologi terlihat apabila dilihat dari sisi tegak lurus nya, arah barat atau timur.	51
Gambar 4.21	Peta T_{xy} . Pola anomali berarah x-y (utara-barat) sehingga struktur geologi terlihat dari sisi tegak lurus nya, timur laut atau barat daya.....	52
Gambar 4.22	Peta T_{xz} . Keadaan struktur geologi yang mirip dengan peta T_{xx} , namun dipengaruhi oleh anomali arah vertikal ke atas, sehingga interpretasi pada peta ini untuk melihat struktur yang memiliki bidang miring dengan jurus barat-timur.....	53
Gambar 4.23	Peta T_{yy} . Pola anomali menunjukkan arah barat-timur, sedangkan untuk interpretasi struktur geologi dilihat dari arah utara-selatan.....	54
Gambar 4.24	Peta T_{yz} . Anomali menunjukkan struktur yang memiliki kemiringan dengan jurus utara-selatan... ..	55
Gambar 4.25	Peta T_{zz} . Anomali yang paling mendekati dengan keadaan struktur geologi... ..	56
Gambar 4.26	Struktur geologi hasil interpretasi pada daerah penelitian.....	57

“Halaman sengaja dikosongkan”

DAFTAR TABEL

Tabel 4.1	Perhitungan nilai lebar jendela.....	41
-----------	--------------------------------------	----

“Halaman sengaja dikosongkan”

DAFTAR SIMBOL

Simbol	Definisi
x, y, z	Sistem koordinat kartesian
r, φ, λ	Jari-jari, lintang, dan bujur <i>spherical</i>
n, m	Derajat dan orde <i>spherical harmonic</i>
n_{min}, n_{max}	Derajat minimum dan maksimum dari <i>spherical harmonic expansion</i>
$\bar{P}_{n,m}$	Fungsi Legendre jenis pertama derajat n dan orde m yang diasosiasikan dengan normalisasi penuh 4π
$P_{n,m}$	Fungsi Legendre jenis pertama derajat n dan orde m yang diasosiasikan.
P_n	Polinomial Legendre derajat n
$\bar{C}_{n,m}, \bar{S}_{n,m}$	Koefisien <i>spherical harmonic</i> derajat n dan orde m yang dinormalisasi penuh 4π pada model geopotensial global
$C_{n,m}, S_{n,m}$	Koefisien <i>harmonic</i> derajat n dan orde m
$C_{n,m}^{ell}, S_{n,m}^{ell}$	Koefisien <i>spherical harmonic</i> derajat n dan orde m mengacu pada referensi elipsoid (WGS84/GRS80) yang dinormalisasi penuh 4π
GM^{ell}, a^{ell}	Konstanta gravitasi geosentris dan sumbu semi mayor referensi elipsoid
GM, R	Konstanta gravitasi geosentris dan sumbu semi mayor referensi bola (<i>sphere</i>)

“Halaman sengaja dikosongkan”

BAB I

PENDAHULUAN

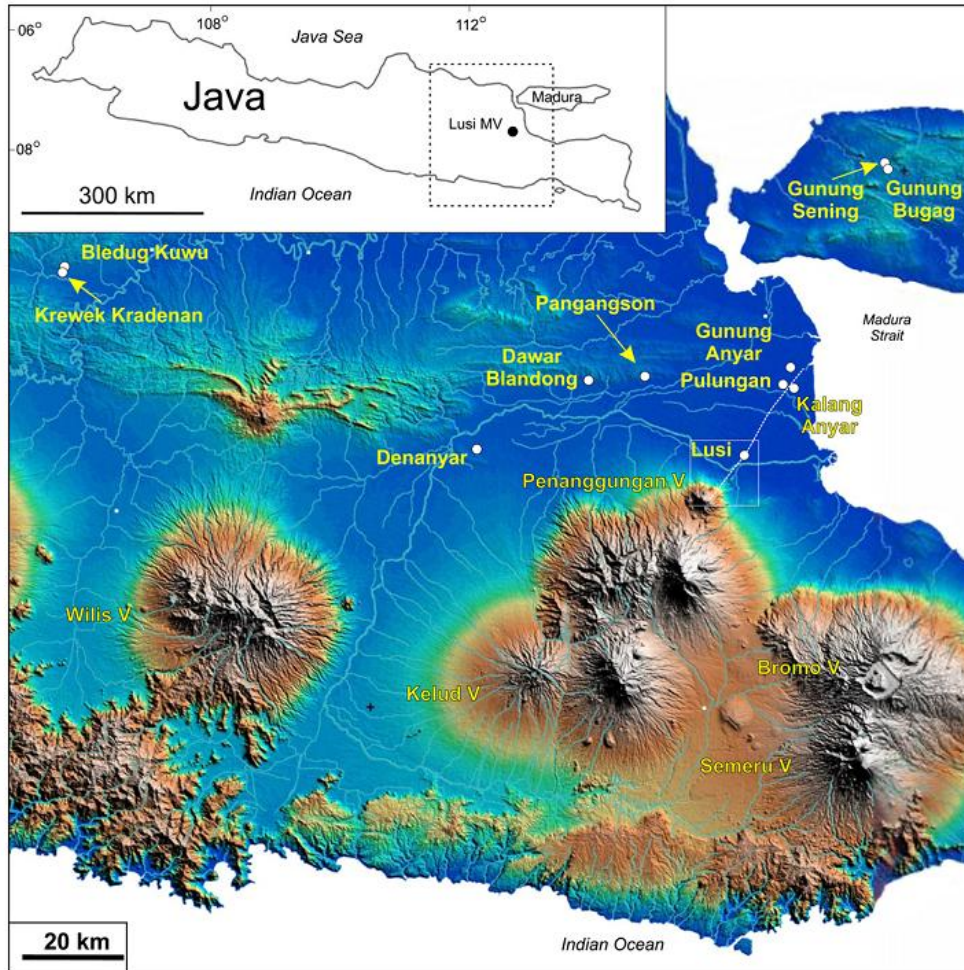
1.1 Latar Belakang

Di daerah Jawa Timur diindikasikan terdapat patahan geser Watukosek (Gambar 1.1) yang membentang dari gunung penanggungan, memotong sungai porong yang ditandai dengan pembelokan tajam Sungai Porong, kemudian melewati tepat pada semburan LUSI (lumpur Sidoarjo), sampai ke Selat Madura (Mazzini, *et al.*, 2009). Selain itu, celah vulkanik dan gunung api lumpur (Gambar 1.2) di bagian Jawa ini selaras dengan LUSI dan memiliki jurus yang sama dengan patahan Watukosek (Mazinni, *et al.*, 2012).

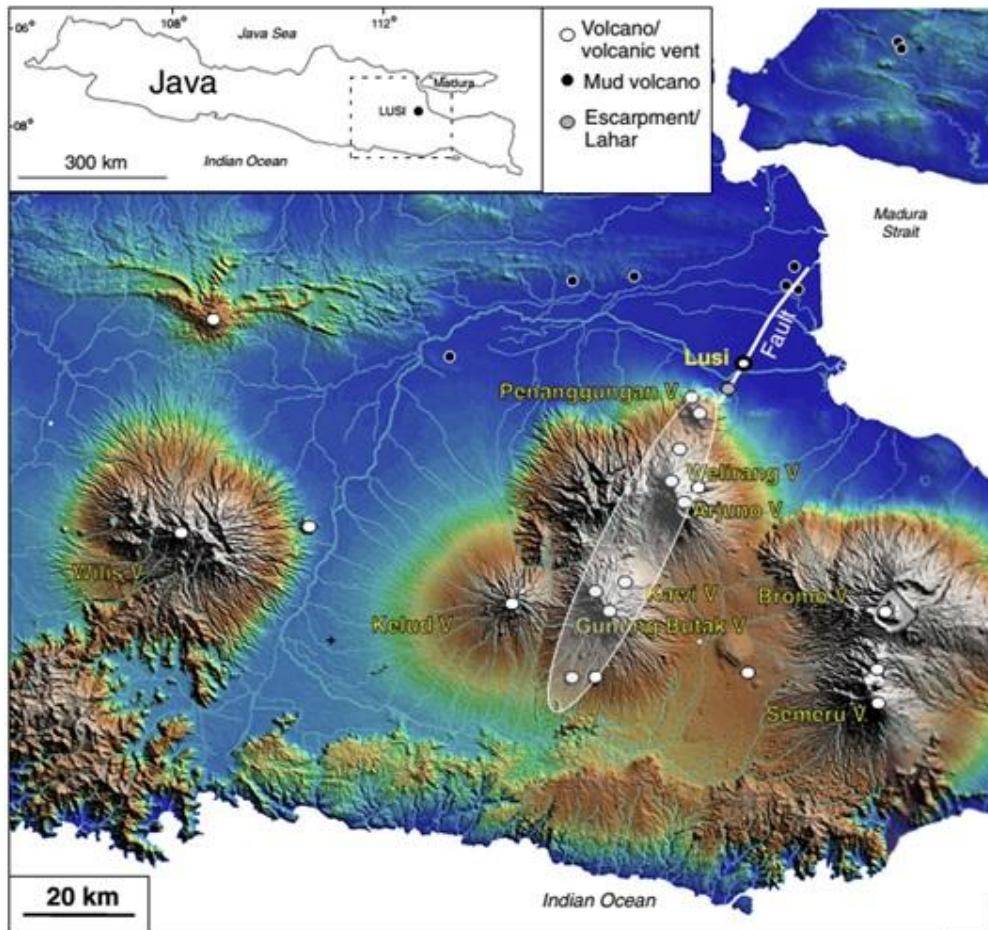
Gunung api lumpur LUSI menambah banyak gunung api lumpur yang ada di Indonesia, seperti struktur keruntuhan Porong (timur laut LUSI), Kalang Anyar & Pulungan (Sedati, Sidoarjo), Gunung Anyar (kampus UPN, Surabaya), Bleduk Kuwu & Keradenan (Purwodadi), Wringin Anom / Pengangson (Gresik), Semolowaru (kampus Unitomo, Surabaya), Dawar Blandong (Mojokerto), Sangiran (Jawa Tengah), Socah (Bangkalan, Madura) dan lainnya. Semua gunung api lumpur ini terjadi secara alami, beberapa di antaranya masih aktif dan masih mengeluarkan lumpur. Di bagian timur Jawa, di sebelah timur zona Kendeng, sejumlah gunung api lumpur ini menunjukkan sebuah pola yang mengikuti Patahan Watukosek. Patahan ini dimulai dari kompleks gunung berapi Arjuno - Welirang di sepanjang arah barat daya - timur laut (Sawolo, *et al.*, 2009)

Penyebab erupsi LUSI masih dalam perdebatan. Menurut Manga (2007) bahwa pernah terjadi gempa di tahun 1976 dan 1998 yang lebih dekat dengan daerah LUSI yang tidak menimbulkan terjadinya erupsi di daerah tersebut. Pendapat lain seperti Davies *et al* (2008) dan Davies *et al* (2011) menyatakan bahwa pemicu semburan LUSI itu disebabkan oleh operasi pemboran Banjar Panji-1 (BJP-1) yang berjarak 150 m dari pusat semburan LUSI. Peristiwa utamanya adalah pada saat pengangkatan mata bor dan pipa bor pada tanggal 27-28 Mei yang menyebabkan masuknya fluida formasi dan gas ke dalam lubang bor. Hasil dari

penutupan dari peristiwa *kick*, tingginya kelebihan tekanan pipa bor pada tanggal 28 Mei 2006 dipadukan dengan tekanan pengeboran lumpur akan cukup untuk menyebabkan rekahan hidrolik yang berada di bawah sepatu casing 33,97 cm. Rekahan hidrolik disebarkan ke permukaan dan letusan dimulai.



Gambar 1.1 Kemungkinan patahan Watukosek (garis putih putus-putus) dan gunung api lumpur yang ditandai dengan titik putih. (Mazzini, *et al.*, 2009).



Gambar 1.2 Peta ketinggian Pulau Jawa bagian timur. Posisi beberapa gunung api lumpur yang diketahui, gunung api utama dan celah vulkanik (Mazinni, *et al.*, 2012).

Mekanisme ini dikenal sebagai *subsurface blowout*. Faktanya bahwa bagian paling bawah di 1734 m dari Sumur eksplorasi tidak memiliki casing pelindung sebagai faktor pendukung, namun *kick* yang ter-induksi dengan menarik pipa bor dan mata bor memicu gunung api lumpur. Selain itu, Tanikawa *et al* (2010) berpendapat bahwa erupsinya LUSI akibat dari tekanan fluida pori berubah akibat gempa Yogyakarta 27 Mei 2006 sehingga menyebabkan terjadinya *liquefaction* dari lempung dan aliran fluida yang terlihat pada rekahan alami.

Berbagai metode geofisika telah dilakukan pada daerah LUSI. Interpretasi gayaberat oleh Istadi *et al* (2009) dengan area mencakup $7 \times 7 \text{ km}^2$, dan alat yang digunakan dalam survey detail tersebut adalah *microgravity*. Penelitian tersebut dilakukan pada Juni 2006, sebulan setelah erupsi LUSI pada bulan Mei 2006. Hasilnya adalah peta anomali Bouguer residual yang menunjukkan anomali negatif pada daerah LUSI, namun analisis patahan Watukosek dari hasil anomali Bouguer residual tidak dijelaskan secara mendetail. Penelitian di daerah yang sama juga dilakukan oleh Padmawidjaja (2013) yang menampilkan interpretasi secara regional maupun secara lokal pada daerah LUSI, namun analisis mengenai patahan Watukosek juga tidak dijelaskan secara mendetail. Yusuf *et al* (2015) melakukan penelitian daerah LUSI pada skala regional dengan menggunakan data *TOPEX*. Data tersebut diolah untuk mendapatkan analisis jenis patahan Watukosek dengan menggunakan metode *SVD* (*Second Vertical Derivative*).

Pada penelitian ini digunakan data gayaberat global (*Combined Global Gravity Model*) dari model *European Improved Gravity model of the Earth by New techniques-6C4* (Foerste, *et al.*, 2014) atau disingkat *EIGEN-6C4*, data ketinggian *EARTH2014* (Hirt dan Rexer, 2015).

Data *EIGEN-6C4* diolah untuk mendapatkan gayaberat absolut dan tensor Marussi. Data gayaberat absolut dari *EIGEN-6C4* dan data ketinggian dari *EARTH2014* digunakan untuk mendapatkan Anomali Bouguer Lengkap (ABL). Anomali Bouguer adalah anomali gayaberat yang menunjukkan keadaan geologi daerah penelitian. Intrusi, cekungan, patahan maupun lipatan dapat terdeteksi dengan menggunakan metode ini. Dari data yang ada kemudian dibuat peta Anomali Bouguer Lengkap (ABL) yang kemudian di-*filter* dengan metode *low-pass filter* (*moving average*) untuk mendapatkan anomali Bouguer regional dan residual. Dalam pemodelan inversi biasanya menggunakan data anomali residual, tetapi dikarenakan data yang digunakan bersifat regional maka data masukan untuk pemodelan inversi 3D menggunakan anomali Bouguer regional. Sedangkan Tensor Marussi adalah turunan parsial kedua dari *disturbing potensial*, atau disebut *Disturbing Tensor* Hasil dari model inversi 3D dikombinasi dengan hasil Tensor Marussi gambaran struktur geologi bawah permukaan.

1.1 Rumusan Masalah

Perumusan masalah dalam penulisan Tesis ini sebagai berikut:

- a. Data *EIGEN-6C4* adalah data gayaberat yang menghasilkan anomali gayaberat yang bersifat regional, sehingga dapat digunakan untuk proses analisis dan interpretasi pada daerah yang luas.
- b. Patahan Watukosek tidak muncul dipermukaan sebelumnya, namun ketika gempa Yogyakarta 27 Mei 2006 patahan tersebut muncul ditandai dengan rekahan-rekahan (Mazzini, *et al.*, 2009).
- c. Pada peta geologi tahun 1992 tidak menunjukkan adanya patahan Watukosek, sehingga perlu adanya pembuktian dengan metode-metode yang digunakan dalam tesis ini.

1.3 Batasan Masalah

Batasan masalah pada penulisan Tesis ini adalah sebagai berikut:

- a. Data gayaberat yang digunakan adalah data dari model *EIGEN-6C4*.
- b. Lokasi penelitian berada pada 112⁰ BT – 113⁰ BT, 7⁰ LS – 8⁰ LS, dengan luas sekitar 111 km x 111 km.
- c. Data ketinggian merupakan model *EARTH2014*.
- d. Untuk proses digitasi dan kontrol geologi digunakan peta geologi tahun 1992 lembar Surabaya-Sapulu dan lembar Malang dengan skala 1:100.000.
- e. Untuk mendapatkan gayaberat absolut dan tensor Marussi digunakan perangkat lunak GrafLab berbahasa MATLAB (Bucha dan Janak, 2013).
- f. Untuk pemisahan anomali regional-residual ABL digunakan metode perata-rataan bergerak (*moving Average*).
- g. Untuk pemodelan gayaberat inversi tiga dimensi menggunakan perangkat lunak GRAV3D yang dikembangkan *UBC - Geophysical Inversion Facility*. Perangkat lunak ini dapat di download di <http://www.flintbox.com/public/project/1605/>.

1.3 Tujuan Penelitian

Tujuan dari Tesis ini adalah sebagai berikut:

- a. Membuat model gayaberat inversi tiga dimensi pada daerah yang diperkirakan terdapat Patahan Watukosek (berdasarkan interpretasi Mazinni, *et al.*, 2009; Istadi, *et al.*, 2009; Mazinni, *et al.*, 2012).
- b. Membuat peta tensor Marussi yang terdiri dari 6 peta ($T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}, T_{zz}$).
- c. Interpretasi struktur geologi patahan Watukosek dari hasil model inversi gayaberat tiga dimensi dan tensor Marussi.

1.4 Manfaat Penelitian

Manfaat dari Tesis ini adalah:

- a. Mengembangkan ilmu dan pengetahuan geologi, geodesi, dan geofisika dalam pengolahan data dan interpretasi metode gayaberat.
- b. Sebagai bagian dari bahan kajian kondisi geologi regional daerah penelitian yang berhubungan dengan mitigasi bencana maupun eksplorasi minyak dan gas bumi.

BAB II

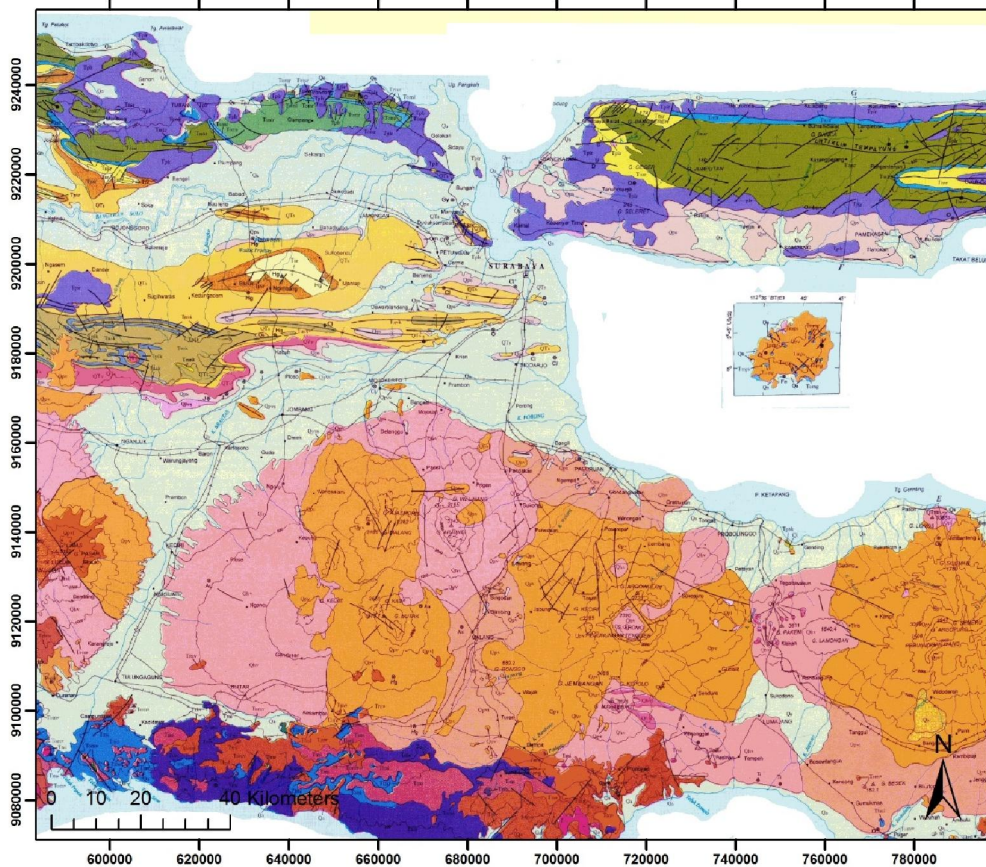
KAJIAN PUSTAKA

2.1 Tektonik dan Stratigrafi

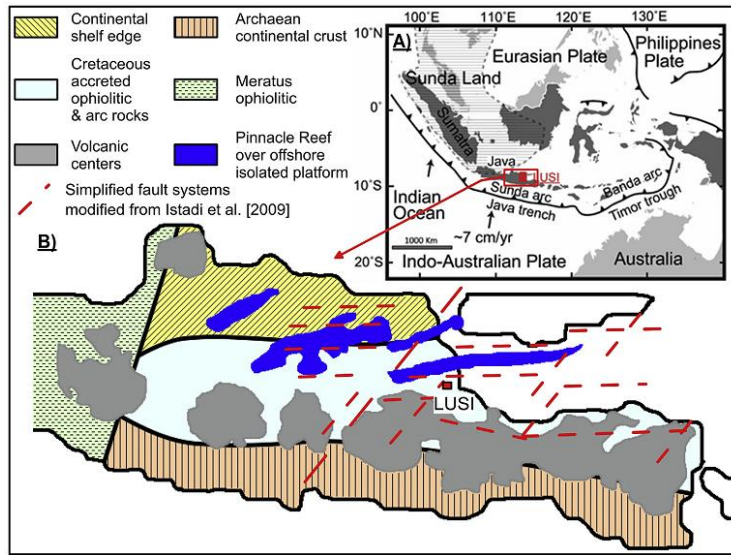
Daerah Jawa Timur (Gambar 2.1 dan Gambar 2.2), secara geologis terbagi menjadi tiga provinsi struktural (Smyth, *et al.*, 2007; Clements, *et al.*, 2009; Hall, 2011) yang terdiri dari paparan benua di bagian utara, ophiolit yang terkonsentrasi berumur cretaceous dan batuan busur vulkanik (cekungan Kendeng), dan kerak benua Archaean di daerah selatan. Sebelah tenggara Cekungan Kendeng, dimana LUSI berada, mengalami fase ekstensional dari Eosen Tengah ke Oligosen dimana *grabens* dan *half-grabens* dibentuk. Hal ini diikuti oleh fase kompresional dari Neogen sampai seterusnya, ditandai dengan patahan naik dan patahan geser (Clements *et al.*, 2009). Dari Pliosen Akhir sampai Holosen, Cekungan ini mengalami fase sedimentasi yang paling baru (De Genevraye dan Samuel, 1972; Sawolo, *et al.*, 2009). Selama ini bagian selatan cekungan mengalami pergerakan tektonik kompresional yang mengarah pada pembentukan *horst* dan daerah *uplift*. Selama fase *uplift* material magmatik mengisi *graben* membentuk endapan vulkaniklastik. Selanjutnya, endapan laut dangkal, lempung dan karbonat diendapkan diatas endapan vulkaniklastik dari Oligosen ke Holosen (De Genevraye dan Samuel, 1972). Penurunan cepat yang dialami daerah setelah Holosen ditambah dengan pematangan dari bahan organik dan laju sedimentasi tinggi yang terjadi di wilayah inilah yang dianggap telah menyebabkan kelebihan tekanan yang diukur dalam formasi Kalibeng (Willumsen dan Schiller, 1994; Schiller *et al.*, 1994).

LUSI terletak di daerah busur belakang (*back-arc*), 10 km timur laut dari Gunung Penanggungan. Profil seismik daerah dan pengamatan lapangan menunjukkan adanya patahan regional melalui area LUSI meskipun sebagian terkubur oleh sedimen aluvial. Stratigrafi di wilayah Sidoarjo (Gambar 2.3) terdiri dari (atas-bawah) sedimen alluvial, selang seling batupasir dan serpih Formasi Pucangan sampai kedalaman 900 m berumur Pleistosen, tanah liat abu-abu liat Pleistosen dari Formasi Kalibeng Atas sampai kedalaman 1871 m, dan pasir vulkaniklastik setidaknya setebal 962 m. Sumur Porong #1 yang dibor 6,5 km arah

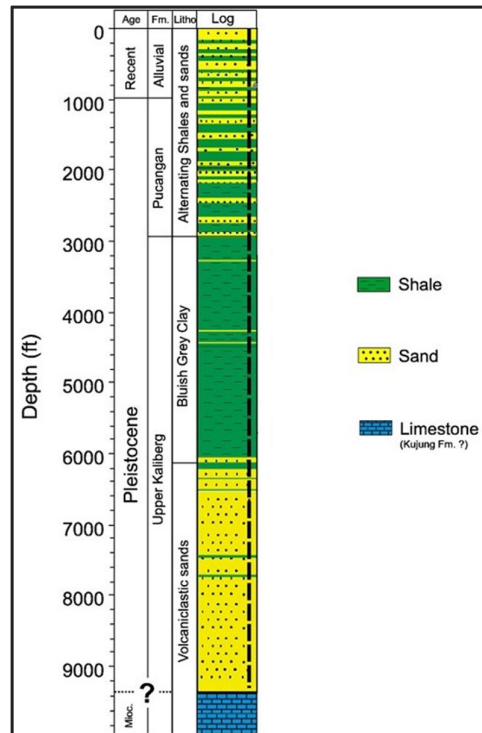
timur laut dari Sidoarjo menunjukkan bahwa Plio-Pleistosen (?) Sedimen langsung menutupi bagian atas batu kapur koral berumur Miosen (Mazinni *et al*, 2007).



Gambar 2.1 Modifikasi dari Peta Geologi Lembar Jawa Bagian Timur (Gafner & Ratman, 1999). Sistem perlipatan Kendeng yang memiliki *plunge* kearah timur tertutup oleh sedimen alluvial delta Brantas.



Gambar 2.2 A) Penyederhanaan kerangka tektonik Asia Tenggara. B) Penyederhanaan kerangka tektonik Jawa Timur. (Panzera, *et al.*, 2017)



Gambar 2.3 Kolom stratigrafi di sumur BJP-1 (Mazinni *et al.*, 2007).

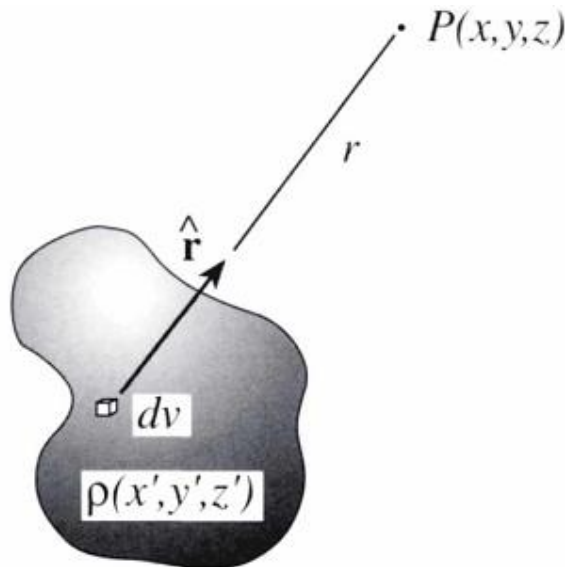
2.2 Potensial Gayaberat

Suatu partikel yang terdapat dalam sistem ruang tertentu akan menimbulkan medan potensial di sekitarnya. Medan potensial bersifat konservatif, artinya usaha yang dilakukan dalam suatu medan tidak tergantung pada lintasan yang ditempuhnya, tetapi tergantung pada posisi awal dan akhir (Blakely, 1995).

Dasar dari metode gayaberat adalah hukum Newton tentang gravitasi dan teori medan potensial.

Potensial $V(x, y, z)$ di titik $P(x, y, z)$ akibat suatu massa yang memiliki densitas $\rho(x', y', z')$ pada jarak tertentu, yaitu (Barthelmes, 2013):

$$V(x, y, z) = G \iiint \frac{\rho(x', y', z')}{((x-x')^2 + (y-y')^2 + (z-z')^2)^{1/2}} dx' \cdot dy' \cdot dz' \quad (2.1)$$



Gambar 2.4 Potensial di titik P terhadap suatu massa (Blakely, 1995).

Dimana G adalah konstanta gravitasi umum, (x, y, z) dan (x', y', z') masing-masing adalah sistem koordinat kartesian dari potensial dan koordinat kartesian sumber benda, dan $dv = dx' dy' dz'$ (Gambar 2.4).

Potensial dalam bentuk *spherical harmonics* (Bucha dan Janak, 2013):

$$V = \frac{GM}{r} \sum_{n=n_{min}}^{n_{max}} \left(\frac{R}{r}\right)^n \sum_{m=0}^n (\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda) \bar{P}_{n,m}(\sin \varphi) \quad (2.2)$$

Dimana, $\bar{P}_{n,m}$ adalah fungsi Legendre jenis pertama derajat n dan orde m yang diasosiasikan dengan normalisasi penuh 4π , sedangkan $\bar{C}_{n,m}$ dan $\bar{S}_{n,m}$ adalah koefisien *spherical harmonic* derajat n dan orde m yang dinormalisasi penuh 4π pada model geopotensial global. Uraikan secara detail diberikan oleh Torge (2001) sebagai berikut (Torge, 2001):

$$\bar{P}_{n,m}(\sin\varphi) = \sqrt{k(2n+1) \frac{(n-m)!}{(n+m)!}} P_{n,m}(\sin\varphi), \quad k = \begin{cases} 1 & \text{untuk } m = 0 \\ 2 & \text{untuk } m \neq 0 \end{cases} \quad (2.3)$$

$$P_{n,m}(\sin\varphi) = (1 - (\sin\varphi)^2)^{m/2} \frac{d^m}{d(\sin\varphi)^m} P_n(\sin\varphi) \quad (2.4)$$

$$P_n(\sin\varphi) = \frac{1}{2^n n!} \times \frac{d^n}{dt^n} ((\sin\varphi)^2 - 1)^n \quad (2.5)$$

$$\begin{cases} \bar{C}_{n,m} \\ \bar{S}_{n,m} \end{cases} = \sqrt{\frac{(n+m)!}{k(2n+1)(n-m)!}} \begin{cases} C_{n,m} \\ S_{n,m} \end{cases}, \quad k = \begin{cases} 1 & \text{untuk } m = 0 \\ 2 & \text{untuk } m \neq 0 \end{cases} \quad (2.6)$$

2.3 Gayaberat dari *Spherical Harmonics Expansion*

Bumi yang berotasi terdapat potensial sentrifugal (V_C) dengan persamaan (Barthelmes, 2013):

$$V_C(x, y, z) = \frac{1}{2} \omega^2 d_z^2 \quad (2.7)$$

Dimana ω adalah kecepatan sudut/angular dari rotasi bumi dan $d_z = \sqrt{x^2 + y^2}$ yang merupakan jarak ke sumbu rotasi z . Apabila dirubah kedalam system koordinat *spherical* maka:

$$V_C(r, \varphi, \lambda) = \frac{1}{2} \omega^2 r^2 \cos^2 \varphi \quad (2.8)$$

Sehingga potensial gayaberat (W) dinyatakan dengan:

$$W = V + V_C \quad (2.9)$$

Gayaberat (g) adalah bentuk skalar dari gradien dari potensial gayaberat:

$$g = |\nabla W| \quad (2.10)$$

Bentuk *spherical harmonics* persamaan 2.10 (Bucha dan Janak, 2013) adalah:

$$g(r, \varphi, \lambda) = \sqrt{\left(\frac{\partial V}{\partial r} + \frac{\partial V_C}{\partial r}\right)^2 + \left[\frac{1}{r} \left(\frac{\partial V}{\partial \varphi} + \frac{\partial V_C}{\partial \varphi}\right)\right]^2 + \left[\frac{1}{r \cos \varphi} \left(\frac{\partial V}{\partial \lambda} + \frac{\partial V_C}{\partial \lambda}\right)\right]^2} \quad (2.11)$$

Dimana,

$$\left. \begin{aligned}
\frac{\partial V}{\partial r} &= -\frac{GM}{r^2} \sum_{n=n_{min}}^{n_{max}} \left(\frac{R}{r}\right)^n (n+1) \sum_{m=0}^n (\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda) \bar{P}_{n,m} \sin \varphi \\
\frac{\partial V}{\partial \varphi} &= \frac{GM}{r^2} \sum_{n=n_{min}}^{n_{max}} \left(\frac{R}{r}\right)^n \sum_{m=0}^n (\bar{C}_{n,m} \cos m\lambda + \bar{S}_{n,m} \sin m\lambda) \frac{d\bar{P}_{n,m}(\sin \varphi)}{d\varphi} \\
\frac{\partial V}{\partial \lambda} &= \frac{GM}{r^2} \sum_{n=n_{min}}^{n_{max}} \left(\frac{R}{r}\right)^n \sum_{m=0}^n (\bar{C}_{n,m} \cos m\lambda - \bar{S}_{n,m} \sin m\lambda) m \bar{P}_{n,m} \sin \varphi
\end{aligned} \right\}$$

(2.12)

$$\frac{\partial V_c}{\partial r} = \omega^2 r^2 \cos^2 \varphi, \quad \frac{\partial V_c}{\partial \varphi} = -\omega^2 r^2 \cos \varphi \sin \varphi, \quad \frac{\partial V_c}{\partial \lambda} = 0$$

(2.13)

2.4 Gayaberat Normal

Gayaberat normal adalah nilai gayaberat teoritis tegak lurus terhadap bidang elipsoid bumi.

Gayaberat normal dihitung dengan menggunakan persamaan Somigliana-Pizetti (Moritz, 1980) dinyatakan dengan:

$$\gamma_0 = \gamma_e \left(\frac{1+k\sin^2\varphi}{\sqrt{1-e^2\sin^2\varphi}} \right)$$

(2.14)

Dimana,

γ_0 = Gayaberat Normal pada bidang referensi elipsoid (mgal)

γ_e = Gayaberat (normal) di equator

k = konstanta

e^2 = Eksentrisitas pertama dari elipsoid referensi (WGS84)

φ = Sudut lintang geosentris (rad)

Dengan menggunakan parameter WGS84 (NIMA, 2000), gayaberat normal dari persamaan 2.14 adalah:

$$\gamma_0 = 978032,67715 \left(\frac{1+0,00193185265241\sin^2\varphi}{\sqrt{1-0,0066943799901490\sin^2\varphi}} \right)$$

(2.15)

2.5 Koreksi terhadap Gayaberat Normal

2.5.1 Koreksi Udara Bebas (KUB)

Orde kedua dari koreksi udara bebas (Featherstone, 1995a) pada ketinggian diatas geoid (H), dengan persamaan:

$$KUB = \frac{2\gamma_0}{a} (1 + f + m - 2f \sin^2 \Phi) H - \frac{3\gamma_0}{a^2} H^2 \quad (2.16)$$

dimana,

Φ = sudut lintang geodetik (rad)

f (kecepatan) = $\frac{a-b}{a}$,

m (parameter geodetik) = $\frac{\omega^2 a^2 b}{GM}$, dan

γ_0 = Gayaberat normal (mgal).

Persamaan 2.16 menggunakan parameter WGS84 (NIMA, 2000) dimana,

a (sumbu semi mayor) = 6378137,0 meter,

b (sumbu semi minor) = 6356752,3142 meter,

ω (kecepatan angular bumi) = $7292115,0 \times 10^{-11}$ rad/s, dan

GM (konstanta gravitasi geosentris) = 3986004.418×10^8 m³/s²

2.5.2 Koreksi Bouguer Sederhana (*Bullard A*)

Koreksi udara bebas dan gayaberat normal mengabaikan massa yang mungkin ada antara posisi pengamatan dengan geoid. Koreksi Bouguer sederhana mendekati semua massa di atas permukaan laut dalam bentuk homogen, yaitu *infinite slab*, dimana ketebalan sama dengan tinggi titik pengamatan di atas geoid (Blakely, 1995).

Koreksi Bouguer Sederhana atau *Bullard A* (Lafehr, 1991) pada ketinggian diatas geoid (H) adalah koreksi pengukuran akibat densitas *slab* (ρ) batuan diatas geoid dengan nilai rata-rata 2,67 gr/cc (Blakely, 1995):

$$Bullard A = 2\pi\rho GH \quad (2.17)$$

2.5.3 Koreksi Bouguer *Spherical Cap* (*Bullard B*)

Koreksi Bouguer *Spherical Cap* mengkonversi *infinite slab* (Gambar 2.5) pada perhitungan *Bullard A* ke dalam bentuk *spherical cap* (Gambar 2.6). Koreksi ini disebut *Bullard B* atau *curvature*.

$$Bullard B = 2\pi\rho G(\mu h - \lambda R) \quad (2.18)$$

Dimana μ dan λ adalah koefisien yang dihitung dari persamaan 2.19 dimana,

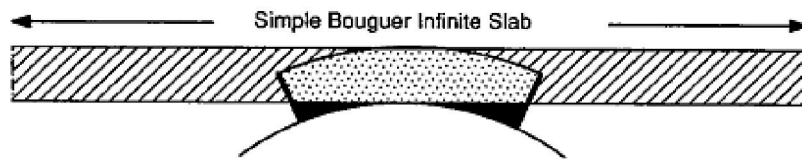
$$\mu = (1/3\eta^2 - \eta)$$

$$\lambda = \frac{1}{3} \left\{ (d + f\delta + \delta^2)[(f - \delta)^2 + k]^{1/2} + p + m \log_e \frac{n}{f - \delta + [(f - \delta)^2 + k]^{1/2}} \right\}$$

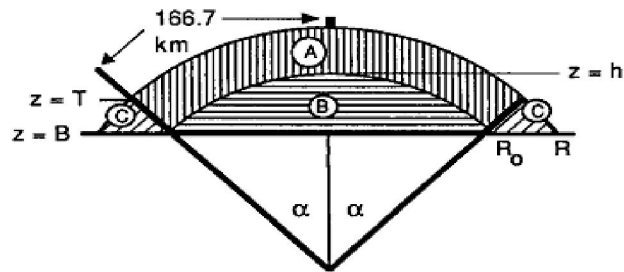
$$\eta = h/R, \quad \delta = R_0/R, \quad d = 3 \cos^2 \alpha - 2, \quad f = \cos \alpha, \quad k = \sin^2 \alpha$$

$$p = -6 \cos^2 \alpha \sin(\alpha/2) + 4 \sin^3(\alpha/2), \quad m = -3 \sin^2 \alpha \cos \alpha,$$

$$n = 2 [\sin(\alpha/2) - \sin^2(\alpha/2)]$$



Gambar 2.5 *Infinite Slab* pada koreksi Bouguer Sederhana (Lafer, 1991b).



Gambar 2.6 *Spherical cap* pada koreksi Bullard B (Lafer, 1991b).

Dimana,

R_0 = Jari-jari normal permukaan laut (6371 km)

R = Jari-jari dari stasiun

h = ketinggian stasiun, tetapi dihitung dari R_0 (permukaan laut)

- B = jarak vertikal stasiun terhadap bidang horizontal
- T = jarak vertikal stasiun ke bidang horizontal melewati batas atas zona ©
- Ⓑ = zona pertama yang akan dihilangkan dalam koreksi *curvature*
- Ⓒ = zona kedua yang akan dihilangkan dalam koreksi *curvature*
- α = setengah sudut dari panjang busur 166,7 km

Nilai 166,7 km untuk jari-jari permukaan *spherical cap* (yang merupakan jari-jari luar zona 0 Hayford-Bowie) untuk meminimalisir perbedaan antara pengaruh *cap* dan *infinite slab* untuk rentang ketinggian yang signifikan. (Lafehr, 1991b)

2.5.4 Koreksi Medan (*Bullard C*)

Koreksi medan adalah koreksi akibat dari pengaruh massa di luar stasiun. Dalam tesis ini digunakan gabungan dari metode prisma (Nagy, 1966) dan metode cincin (Kane, 1962).

Untuk menghitung koreksi lokal, data DEM (*Digital Elevation Model*) lokal “*disampling*” menjadi ruas grid dimana titik stasiun berada ditengah (Gambar 2.7). Pada dasarnya koreksi ini mengitung pada *near zone* (zona 0), *intermediate zone* (zona 1), dan *far zone* (zona 2). Pada *near zone* (jumlah sel 0 sampai 1 dari stasiun) algoritma menjumlahkan efek dari empat segitiga (gambar 2.8a) yang menunjam/miring (*sloping triangle*). Untuk menghitung nilai koreksi dari zona 0 digunakan persamaan:

$$g_0 = G\rho\Phi\left(R - \sqrt{R^2 + H^2} + \frac{H^2}{\sqrt{R^2 + H^2}}\right) \quad (2.19)$$

dimana,

- g_0 = nilai koreksi pada zona 0 (mgal)
- ρ = densitas batuan (kg/m^3)
- Φ = sudut pada segitiga mendatar

Pada *intermediate zone* (jumlah sel 1 sampai 8 dari stasiun), setiap stasiun dihitung koreksi medannya dengan menggunakan metode dari Nagy (1966), yaitu suatu prisma berbentuk kubus datar di bagian atas (Gambar 2.8b). Persamaan koreksi medan untuk zona 1 adalah:

$$g_1 = -G\rho \left[\frac{z_2}{z_1} \frac{y_2}{y_1} \frac{x_2}{x_1} x \bullet \ln(y+R) + y \bullet \ln(x+R) + Z \arctan \frac{Z \bullet R}{x \bullet y} \right] \quad (2.20)$$

Dimana,

g_1 = koreksi medan total pada zona 1 (mgal)

x, y, Z = koordinat kartesian suatu massa berdimensi $dz_1 dy_1 dz_2$ terhadap stasiun

R = jari-jari dari stasiun terhadap massa pada prisma

Pada *far zone* (lebih besar dari 8 sel), koreksi medan dihitung pada suatu prisma dengan luas A pada suatu segmen antara cincin 1, R_1 , dan cincin 2, R_2 (Gambar 2.8c). Koreksi ini merupakan penerapan metode Kane (1962), dengan persamaan:

$$g_2 = 2G\rho A^2 \left(\frac{R_2 - R_1 \sqrt{R_1^2 + H^2} - \sqrt{R_2^2 + H^2}}{(R_2^2 - R_1^2)} \right) \quad (2.21)$$

Dimana,

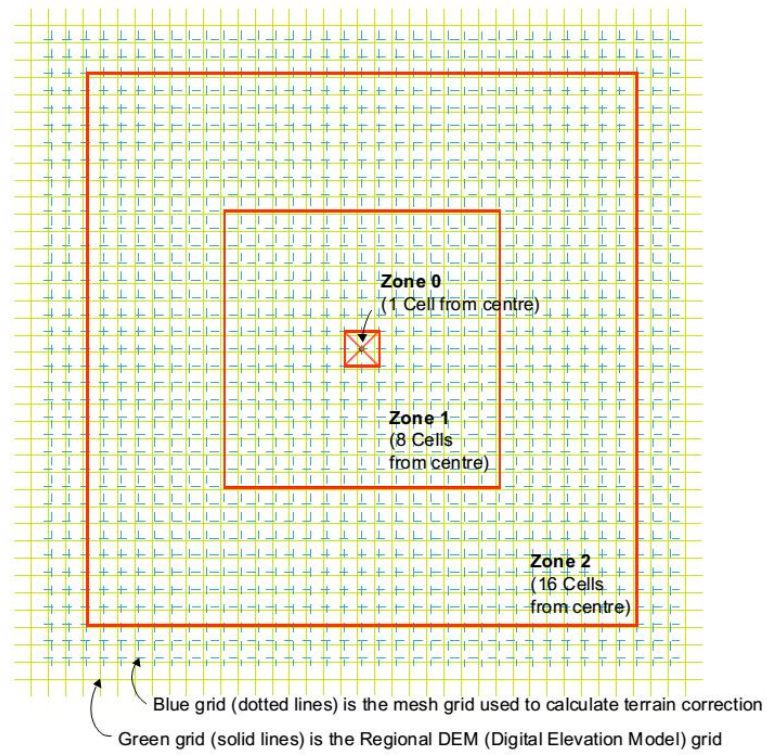
g_2 = koreksi medan pada zona 2 (mgal)

A = luas prisma

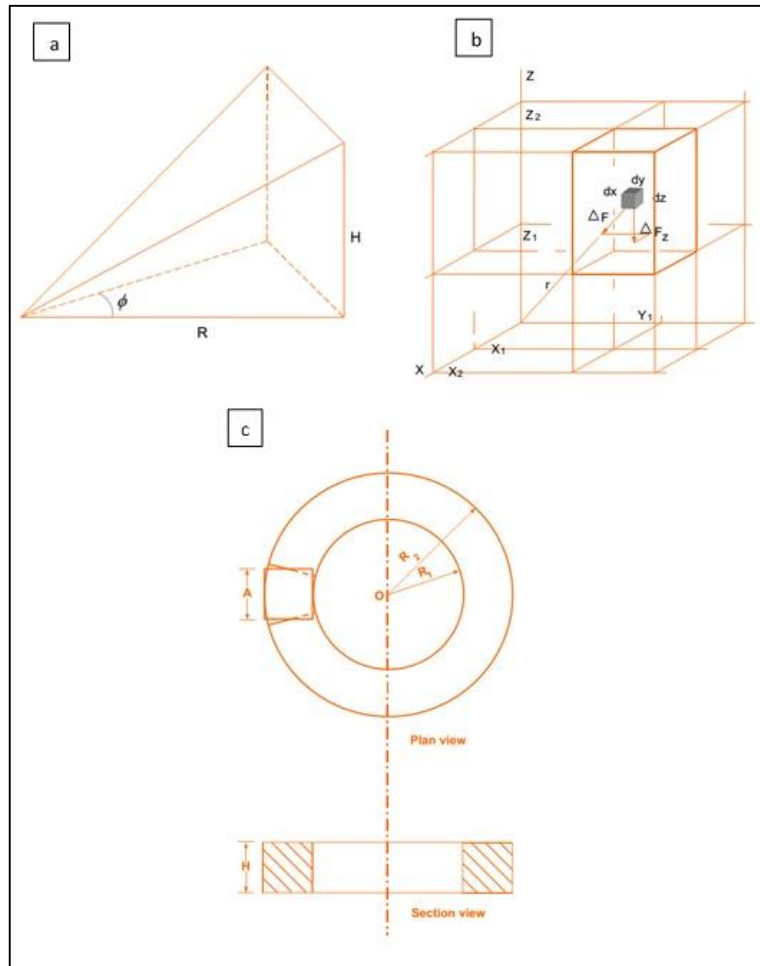
R_2 = Jari-jari luar

R_1 = Jari-jari dalam

H = Ketinggian prisma terhadap stasiun



Gambar 2.7 Grid DEM (*Digital Elevation Model*) regional (garis hijau) dan grid *mesh* (garis biru) yang akan dihitung koreksinya pada zona 0, zona 1, dan zona 2. (Nagy, 1966).



Gambar 2.8 (a) Koreksi medan pada Zona 0, (b) koreksi medan pada Zona 1 (Nagy, 1966), dan (c) koreksi medan pada zona 2 (Kane, 1962).

2.6 Anomali Udara Bebas (*AUB*)

Anomali Udara Bebas (*AUB*) adalah selisih nilai gayaberat observasi pada titik pengukuran dengan nilai gayaberat normal yang telah dikoreksi dengan koreksi udara bebas. Persamaannya adalah (Blakely, 1995):

$$AUB = g_{obs} - \gamma_0 + KUB \quad (2.22)$$

2.7 Anomali Bouguer Lengkap (ABL)

Menurut Lafehr (1991) bahwa hasil koreksi Bouguer *spherical cap* harus ditambah dengan koreksi Bouguer *planar* sehingga Anomali Bouguer Lengkap dihitung dengan persamaan sebagai berikut (Lafehr, 1991):

$$ABL = AUB - (Bullard A + Bullard B - bullard C) \quad (2.23)$$

2.8 Pemisahan Anomali Regional dan Anomali Residual

Anomali regional adalah anomali akibat pengaruh densitas pada zona yang dalam sedangkan anomali residual akibat pengaruh densitas pada zona yang dangkal. Anomali regional memiliki panjang gelombang yang tinggi sedangkan anomali residual memiliki panjang gelombang yang rendah. Dalam memisahkan anomali regional dan residual tersebut perlu dilakukan analisis spektrum untuk mengetahui batas antara kedua anomali tersebut.

2.8.1 Analisis Spektrum

Mengacu pada Blakely (1995), analisis spektrum dilakukan untuk melihat respon anomali yang berasal dari zona dalam (regional), zona dangkal (residual) dan *noise*. Analisis spektrum bertujuan untuk mendapatkan gradien regional dan gradien residual. Hasil perpotongan kedua gradien adalah nilai bilangan gelombang *cut-off* (k_c). Nilai k_c untuk mendapatkan estimasi lebar jendela pada metode perataan bergerak.

Transformasi Fourier (persamaan 2.24) dalam metoda gayaberat adalah men-transformasi dari domain ruang ($F(x)$) ke domain bilangan gelombang ($F(k)$) seperti halnya men-transformasi dari domain waktu ke domain frekuensi di dalam metode seismik.

$$F(x) \leftrightarrow F(k) \quad (2.24)$$

Fungsi dalam domain bilangan gelombang merupakan penjumlahan dari real ($ReF(k)$) dan imajineranya ($iImF(k)$) seperti pada persamaan 2.25 atau bisa ditulis dalam bentuk lain seperti persamaan 2.26.

$$F(k) = ReF(k) + iImF(k) \quad (2.25)$$

$$F(k) = |F(k)|e^{i\phi(k)} \quad (2.26)$$

Amplitudo ($|F(k)|$) dihitung dengan menggunakan persamaan 2.27, sedangkan fase ($\phi(k)$) dihitung dengan persamaan 2.28, dan $\ln|F[k]|^2$ adalah *power-density spectrum* yang akan digunakan dalam proses analisis.

$$|F(k)| = [(ReF(k))^2 + (ImF(k))^2]^{1/2} \quad (2.27)$$

$$\phi(k) = \arctan \frac{ImF(k)}{ReF(k)} \quad (2.28)$$

Kedalaman anomali rata-rata benda (\bar{d}) dapat dihitung setelah mendapatkan gradien regional dan residual hasil dari plot *power-density spectrum* terhadap bilangan gelombang dengan menggunakan persamaan (Spector & Grant, 1970):

$$\bar{d} = -\frac{1}{4\pi} \left(\frac{p_2 - p_1}{k_2 - k_1} \right) \quad (2.29)$$

Dimana $\frac{p_2 - p_1}{k_2 - k_1}$ adalah gradien hasil regresi linear pada plot *power-density spectrum* terhadap bilangan gelombang.

2.8.2 Perata-rataan Bergerak (*Moving Average*)

Salah satu cara untuk memisahkan anomali regional dan anomali residual adalah dengan metode perata-rataan bergerak, metode ini dilakukan dengan cara merata-ratakan nilai anomalinnya. Hasil dari perata-rataan bergerak ini merupakan anomali regionalnya. Sedangkan anomali residualnya didapatkan dengan mengurangi data hasil pengukuran gravitasi dengan anomali regionalnya.

Lebar jendela (N) didapatkan dari hasil analisis spectrum dengan persamaan:

$$N = \frac{1}{k_c \Delta x} \quad (2.30)$$

Dimana,

$$k = \frac{1}{\lambda}, \quad \lambda = N \cdot \Delta x,$$

λ = panjang gelombang (m),

k_c = bilangan gelombang *cut-off* hasil dari analisis spectrum (putaran/m),

Δx = interval sampling (m).

Anomali Bouguer Lengkap (ABL_{reg}) dihitung dengan persamaan:

$$ABL_{reg} \left(i + \frac{N+1}{2} \right) = \frac{ABL(i)+ABL(i+1)+ABL(i+2)+\dots+ABL\left(i+\frac{N+1}{2}\right)+\dots+ABL(i+n)}{N} \quad (2.31)$$

Dimana,

$$i = 0, 1, 2, 3 \dots$$

Selanjutnya, ABL_{res} dapat dihitung dengan menggunakan persamaan berikut.

$$ABL_{res} = ABL - ABL_{reg} \quad (2.32)$$

Dimana,

ABL_{res} = Anomali Bouguer Lengkap Residual (mgal)

ABL = Anomali Bouguer Lengkap (mgal)

ABL_{reg} = Anomali Bouguer Lengkap Regional (mgal)

2.9 Metodologi Inversi

Data anomali gayaberat (d), dimana $d = (d_1, d_2, d_3 \dots d_N)^T$ dan data kontras densitas (ρ), dimana $\rho = (\rho_1, \rho_2, \rho_3 \dots \rho_N)^T$ dari sel-sel yang akan dimodelkan, keduanya berhubungan oleh *sensitivity matrix* (G), dengan persamaan:

$$d = G\rho \quad (2.33)$$

sensitivity matrix memiliki unsur $g_{i,j}$ yang mengukur kontribusi terhadap datum i^{th} karena unit densitas yang ada di dalam sel.

Masalah inversi diformulasikan untuk mengoptimasi permasalahan dimana fungsi objektif global, ϕ , diminimalisir terhadap batasan-batasan dalam persamaan 2.32. Fungsi objektif global terdiri dari dua komponen: fungsi objektif model, ϕ_m , dan data fungsi *misfit*, ϕ_d , dimana

$$\min \phi = \phi_d + \beta \phi_m \quad (2.34)$$

$$\text{untuk } \rho^t \leq \rho \leq \rho^u$$

Dimana β adalah parameter yang mengontrol kehalusan (*smoothness*) model melalui fungsi objektif dan fungsi *misfit* data. Ketika standar deviasi dari data diketahui, misfit yang dapat diterima diberikan dari nilai ϕ_d dan dengan mencari nilai β melalui *L-curve criterion* akan didapat *misfit* yg diharapkan. Jika tidak, maka pengguna menentukan sendiri nilainya. Batas dipakai melalui metode gradien yang

diproyeksikan sehingga model yang dipulihkan terletak di antara ρ^t sampai dengan ρ^u .

Konstruksi fungsi objektif dari model ketika diminimalisir menghasilkan model yang dapat ditafsirkan secara geofisika. Fungsi obyektif memberi keleluasaan untuk memasukkan informasi sesedikit mungkin atau sebanyak mungkin. Paling tidak, fungsi ini mendorong solusinya menuju model referensi ρ_0 dan mensyaratkan agar modelnya relatif lancar dalam tiga arah spasial. Dengan mengadopsi sistem koordinat Cartesian tangan kanan dimana positif ke utara dan positif turun, persamaan fungsi objektif model adalah:

$$\begin{aligned} \phi_m(\rho) = & \alpha_s \int_V w_s \{w(\mathbf{r})[\rho(\mathbf{r}) - \rho_0]\}^2 dv + \alpha_x \int_V w_x \left\{ \frac{\partial w(\mathbf{r})[\rho(\mathbf{r}) - \rho_0]}{\partial x} \right\}^2 dv \\ & + \alpha_y \int_V w_y \left\{ \frac{\partial w(\mathbf{r})[\rho(\mathbf{r}) - \rho_0]}{\partial y} \right\}^2 dv + \alpha_z \int_V w_z \left\{ \frac{\partial w(\mathbf{r})[\rho(\mathbf{r}) - \rho_0]}{\partial z} \right\}^2 dv, \end{aligned} \quad (2.35)$$

Dimana w_s, w_x, w_y dan w_z *dependent* secara spasial, sedangkan $\alpha_s, \alpha_y, \alpha_x$, dan α_z adalah koefisien yang mempengaruhi dari komponen yang berbeda dalam fungsi objektif. Model referensi ρ_0 dan $w(\mathbf{r})$ adalah fungsi *depth weighting* umum. Tujuan dari fungsi ini adalah untuk menangkal peluruhan geometris dari *sensitivity* dengan jarak dari lokasi pengamatan sehingga kontras densitas yang dipulihkan tidak terkonsentrasi di dekat lokasi pengamatan.

Fungsi objektif dalam persamaan 2.35 memiliki eksistensi untuk menggabungkan banyak jenis pengetahuan sebelumnya ke dalam inversi. Model referensi mungkin merupakan model latar belakang umum yang diperkirakan dari penyelidikan sebelumnya atau akan menjadi model nol. Model referensi umumnya dimasukkan ke komponen pertama dari fungsi objektif namun dapat dilepaskan jika diinginkan dari persyaratan yang tersisa; biasanya ditentukan nilai model pada titik tertentu daripada memberikan perkiraan gradien. Pilihan apakah harus disertakan atau tidak ρ_0 dalam derivatif dapat memberi efek signifikan pada model yang dipulihkan. Kedekatan model akhir dengan model referensi di lokasi manapun dikendalikan oleh fungsi w_s . Apabila interpreter memiliki ketergantungan yang tinggi pada model referensi di wilayah tertentu, dia dapat menentukan peningkatan

amplitudo di sana dibandingkan dengan wilayah model lainnya. Fungsi *weighting* w_x , w_y , dan w_z dapat dirancang untuk meningkatkan atau menipiskan gradien di berbagai wilayah dalam domain model. Jika geologi menunjukkan zona transisi yang tajam dalam model, maka penurunan *weighting* pada turunan tertentu dari model akan memungkinkan gradien yang lebih tinggi, dengan demikian memberikan model geologis yang lebih baik.

Secara numerik, fungsi objektif model pada persamaan 2.34 di-diskret ke dalam mesh model kontras densitas menggunakan pendekatan *finite difference*. Sehingga:

$$\begin{aligned}\phi_m(\boldsymbol{\rho}) &= (\boldsymbol{\rho} - \boldsymbol{\rho}_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s + \alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z) (\boldsymbol{\rho} - \boldsymbol{\rho}_o), \\ &\equiv (\boldsymbol{\rho} - \boldsymbol{\rho}_o)^T \mathbf{W}_m^T \mathbf{W}_m (\boldsymbol{\rho} - \boldsymbol{\rho}_o), \\ &= \|\mathbf{W}_m (\boldsymbol{\rho} - \boldsymbol{\rho}_o)\|^2\end{aligned}\tag{2.36}$$

Dimana m dan m_0 adalah panjang vektor M dari model referensi yang dipulihkan dan model referensi. Demikian pula, ada pilihan untuk menghapus model referensi dari turunan spasial dalam persamaan 2.36, yaitu:

$$\begin{aligned}\phi_m(\boldsymbol{\rho}) &= (\boldsymbol{\rho} - \boldsymbol{\rho}_o)^T (\alpha_s \mathbf{W}_s^T \mathbf{W}_s) (\boldsymbol{\rho} - \boldsymbol{\rho}_o) + \boldsymbol{\rho}^T (\alpha_x \mathbf{W}_x^T \mathbf{W}_x + \alpha_y \mathbf{W}_y^T \mathbf{W}_y + \alpha_z \mathbf{W}_z^T \mathbf{W}_z) \boldsymbol{\rho}, \\ &\equiv (\boldsymbol{\rho} - \boldsymbol{\rho}_o)^T \mathbf{W}_s^T \mathbf{W}_s (\boldsymbol{\rho} - \boldsymbol{\rho}_o) + \boldsymbol{\rho}^T \mathbf{W}_m^T \mathbf{W}_m \boldsymbol{\rho}, \\ &= \|\mathbf{W}_s (\boldsymbol{\rho} - \boldsymbol{\rho}_o) + \mathbf{W}_m \boldsymbol{\rho}\|^2\end{aligned}\tag{2.37}$$

Matriks individu \mathbf{W}_s , \mathbf{W}_x , \mathbf{W}_y , dan \mathbf{W}_z secara langsung menghitung setelah model *mesh* dan fungsi *weighting* w_s , w_x , w_y , w_z dibuat. Matriks kumulatif $\mathbf{W}_m^T \mathbf{W}_m$ kemudian dibentuk untuk konfigurasi yang dipilih.

Langkah selanjutnya dalam menyiapkan inversi adalah menentukan perhitungan *misfit*. Di sini kita menggunakan l_2 -norm, dengan persamaan:

$$\phi_d = \|\mathbf{W}_d (\mathbf{G}\boldsymbol{\rho} - d)\|^2\tag{2.38}$$

Dengan mengasumsikan adanya noise yang *independent* dalam data dan Gaussian memiliki rata-rata nol. Menentukan \mathbf{W}_d menjadi matriks diagonal yang memiliki elemen $1/\sigma_i$, dimana σ_i adalah standar deviasi datum i^{th} membuat ϕ_d terdistribusi *chi-square* dengan N derajat kebebasan. Data *misfit* yang optimal terkontaminasi

dengan *independent, noise* Gaussian memiliki nilai yang diharapkan $E[\chi^2] = N$ menghasilkan *misfit* untuk inversi (Li & Oldenburg, 1998).

2.10 Tensor Marussi

Persamaan *spherical harmonics* untuk *disturbing potential* (Bucha dan Janak, 2013):

$$T = \frac{GM}{r} \sum_{n=n_{min}}^{n_{max}} \left(\frac{R}{r}\right)^n \sum_{m=0}^n (\Delta \bar{C}_{n,m} \cos m\lambda + \Delta \bar{S}_{n,m} \sin m\lambda) \bar{P}_{n,m}(\sin \varphi) \quad (2.39)$$

Dimana,

$$\Delta \bar{C}_{n,m} = \bar{C}_{n,m} - \bar{C}_{n,m}^{Ell} \frac{GM^{Ell}}{GM} \left(\frac{a^{Ell}}{R}\right)^n$$

$$\Delta \bar{S}_{n,m} = \bar{S}_{n,m} - \bar{S}_{n,m}^{Ell} \frac{GM^{Ell}}{GM} \left(\frac{a^{Ell}}{R}\right)^n = \bar{S}_{n,m}$$

$$\bar{C}_{n,m}^{Ell} \begin{cases} = (-1)^n \frac{3e^{2n}}{(2n+1)(2n+3)\sqrt{4n+1}} \left(1 - n - 5^{3/2} n \frac{\bar{C}_{2,0}^{Ell}}{e^2}\right) & \text{if } n = 0, 2, 4, \dots, 20, m = 0 \\ = 0 & \text{else} \end{cases}$$

$C_{n,m}^{ell}$ dan $S_{n,m}^{ell}$ berturut-turut adalah koefisien *spherical harmonic* derajat n dan orde m mengacu pada referensi elipsoid (WGS84/GRS80) yang dinormalisasi penuh 4π , sedangkan GM^{ell} dan a^{ell} berturut-turut adalah konstanta gravitasi geosentris, sumbu semi mayor referensi elipsoid.

Istilah Tensor Marussi merujuk pada Kostelecky *et al* (2014), adalah hasil turunan kedua dari *disturbing potential* (\mathbf{T}) (Bucha dan Janak, 2013; Petrovskaya, 2006; Pedersen dan Rasmussen, 1990; Beiki dan Pedersen, 2010). Persamaan Tensor Marussi (\mathbf{T}) yang merujuk pada adalah:

$$\mathbf{T} = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix} = \begin{bmatrix} \frac{d^2T}{dx^2} & \frac{d^2T}{dxdy} & \frac{d^2T}{dxdz} \\ \frac{d^2T}{dydx} & \frac{d^2T}{dy^2} & \frac{d^2T}{dydz} \\ \frac{d^2T}{dzdx} & \frac{d^2T}{dzdy} & \frac{d^2T}{dz^2} \end{bmatrix} \quad (2.40)$$

Dimana,

$$T_{xx}(r, \varphi, \lambda) = \frac{GM}{r^3} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R}{r}\right)^n \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_m(\lambda) \left(a_{n,m} \bar{P}_{n,|m|-2}(\sin \varphi) \right. \\ \left. + [b_{n,m} - (n+1)(n+2)] \bar{P}_{n,|m|}(\sin \varphi) + c_{n,m} \bar{P}_{n,|m|+2}(\sin \varphi) \right)$$

$$T_{xy}(r, \varphi, \lambda) = \frac{GM}{r^3} \sum_{n=0}^{n_{\max}} \left(\frac{R}{r}\right)^n \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_{-m}(\lambda) \left(d_{n,m} \bar{P}_{n-1,|m|-2}(\sin \varphi) \right. \\ \left. + g_{n,m} \bar{P}_{n-1,|m|}(\sin \varphi) + h_{n,m} \bar{P}_{n-1,|m|+2}(\sin \varphi) \right), \quad m \neq 0$$

$$T_{xz}(r, \varphi, \lambda) = \frac{GM}{r^3} \sum_{n=0}^{n_{\max}} \left(\frac{R}{r}\right)^n \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_m(\lambda) \left(\beta_{n,m} \bar{P}_{n,|m|-1}(\sin \varphi) \right. \\ \left. + \gamma_{n,m} \bar{P}_{n,|m|+1}(\sin \varphi) \right)$$

$$T_{yy}(r, \varphi, \lambda) = -\frac{GM}{r^3} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R}{r}\right)^n \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_m(\lambda) \left(a_{n,m} \bar{P}_{n,|m|-2}(\sin \varphi) \right. \\ \left. + b_{n,m} \bar{P}_{n,|m|}(\sin \varphi) + c_{n,m} \bar{P}_{n,|m|+2}(\sin \varphi) \right)$$

$$T_{yz}(r, \varphi, \lambda) = \frac{GM}{r^3} \sum_{n=0}^{n_{\max}} \left(\frac{R}{r}\right)^n \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_{-m}(\lambda) \left(\mu_{n,m} \bar{P}_{n-1,|m|-1}(\sin \varphi) \right. \\ \left. + \nu_{n,m} \bar{P}_{n-1,|m|+1}(\sin \varphi) \right), \quad m \neq 0$$

$$T_{zz}(r, \varphi, \lambda) = \frac{GM}{r^3} \sum_{n=n_{\min}}^{n_{\max}} \left(\frac{R}{r}\right)^n (n+1)(n+2) \sum_{m=-n}^n \Delta \bar{C}_{n,m} Q_m(\lambda) \bar{P}_{n,|m|}(\sin \varphi)$$

$$Q_m(\lambda) = \begin{cases} \cos m\lambda, & m \geq 0 \\ \sin |m|\lambda, & m < 0. \end{cases}$$

Kerangka referensi yang digunakan untuk Tensor Marussi adalah *LNOF* (*Local North-Oriented reference Frame*), dimana x mengarah ke utara, y mengarah ke barat, dan z kearah atas. Dimana, T_{xx} , T_{xy} , dan T_{xz} berturut-turut adalah turunan sebagian (*partial*) dari *disturbing potensial* berarah x (T_x) yang diturunkan terhadap

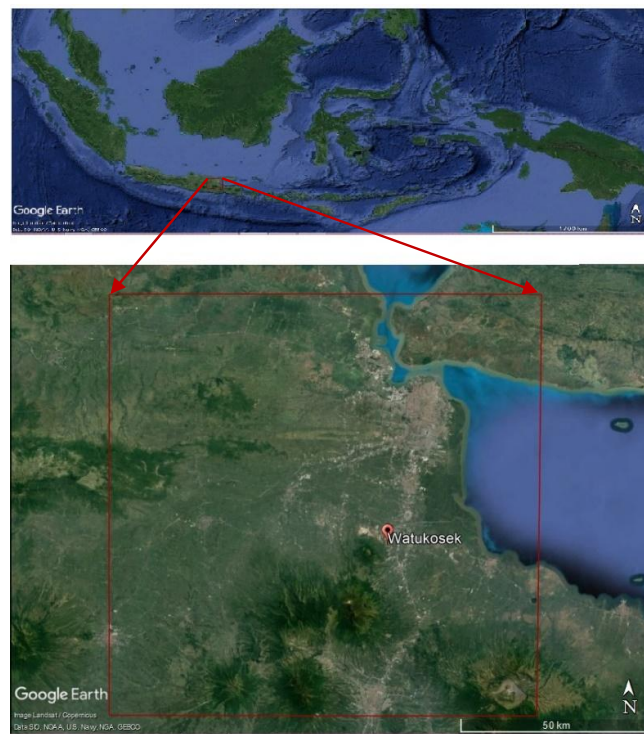
x , y dan z . Sedangkan T_{yy} dan T_{yz} berturut-turut adalah turunan sebagian (*partial*) dari *disturbing potensial* berarah y (T_y) yang diturunkan terhadap y dan z . Kelima gradien ini disebut gradien gayaberat horizontal yang menunjukkan anomali secara horizontal. Dan yang terakhir, T_{zz} adalah turunan kedua vertikal dari *disturbing potensial* yang menunjukkan anomali pada arah vertikal. (Saad, 2006)

Dalam tesis ini peta hasil perhitungan tensor Marussi digunakan sebagai bahan pendukung terhadap hasil pemodelan inversi 3D.

BAB III METODOLOGI PENELITIAN

3.1 Lokasi Penelitian

Penelitian dilakukan di daerah Jawa Timur, Indonesia (Gambar 3.1) yang mencakup area yang sangat luas dengan batasan koordinat 7° LS – 8° LS dan 112° BT – 113° BT, namun difokuskan pada daerah di sekitar Sidoarjo, Mojokerto, dan Pasuruan.



Gambar 3.1 Area Penelitian (kotak merah). Patahan Watukosek berada pada daerah Desa Watukosek, Kabupaten Pasuruan, Provinsi Jawa Timur, Indonesia.

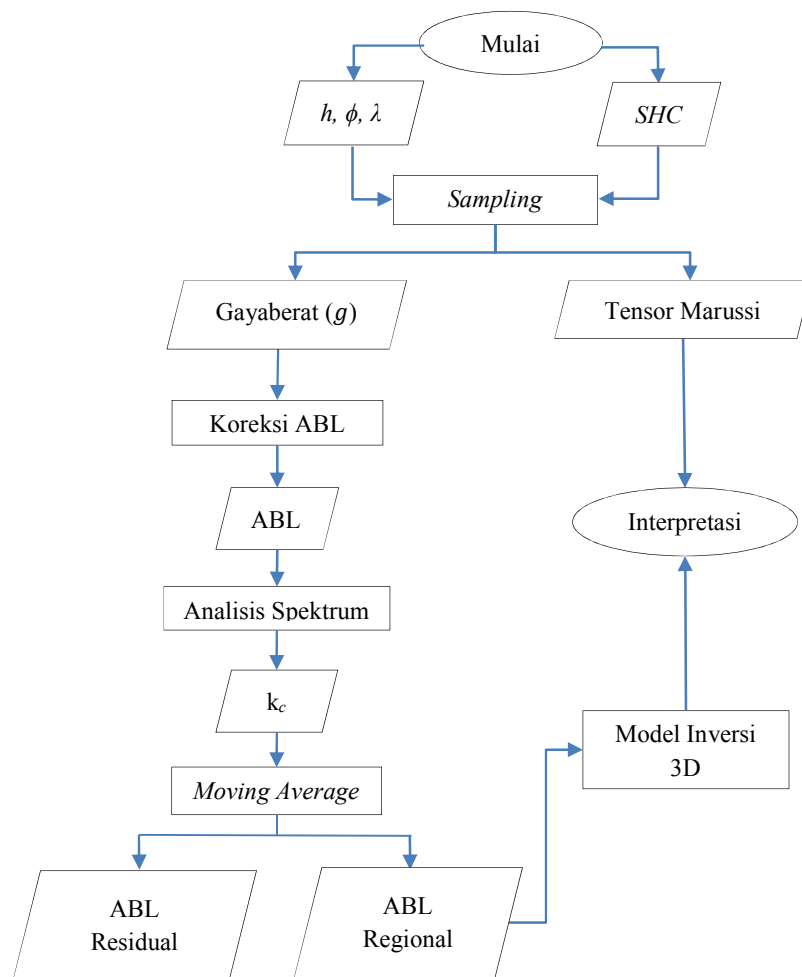
3.2 Data Penelitian

Data yang digunakan dalam penelitian ini adalah:

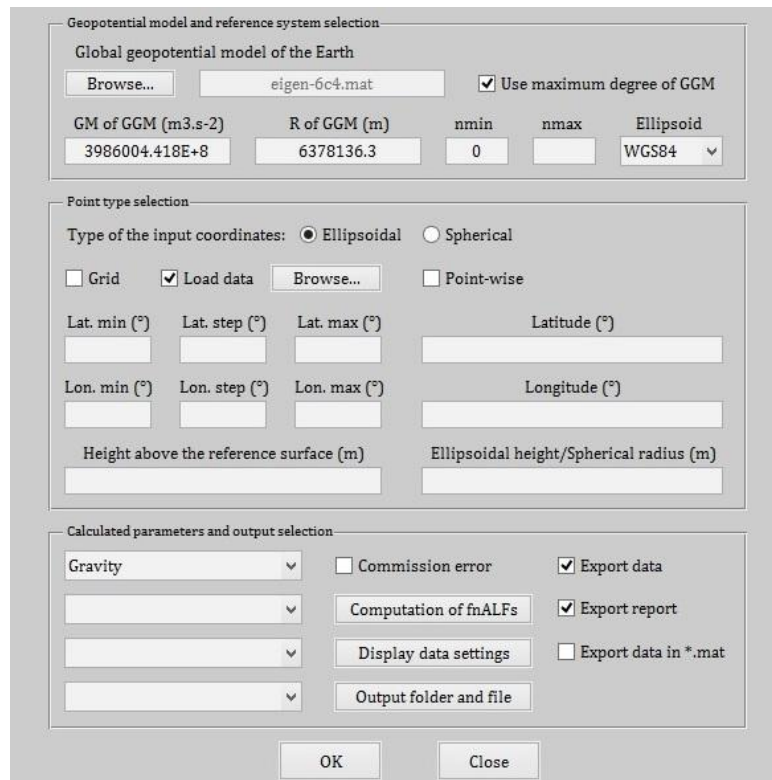
1. Data *DEM (Digital Elevation Model)* dari *Earth2014* (Hirt and Rexer, 2015). Data ini adalah data ketinggian ortometrik (H) didapat dari *TBI (Topography, Bedrock, Ice) Earth2014* yang merupakan gabungan dari data *Bedmap2*, *BGT v3*, *SRTM30_PLUS v9 Bathymetry*, dan *SRTM v4.1 Topography*. Data tersebut adalah data grid dengan resolusi 1 *arc-minutes* berformat GMT yang diproses dengan menggunakan perangkat lunak The Generic Mapping Tools (Wessel, et al., 2013).
2. Data *European Improved Gravity model of the Earth by New techniques-6C4 (EIGEN-6C4)*. *EIGEN-6C4* adalah hasil kombinasi medan gravitasi global statis. Elaborasi secara bersama-sama oleh GFZ Potsdam dan GRGS Toulouse yang mengandung data satelit dan darat, berupa:
 - a. Data LAGEOS tahun 1985 - 2010
 - b. Data GRACE RL03 GRGS tahun 2003 – 2012
 - c. Data GOCE-SGG (termasuk data T_{xx} , T_{yy} , T_{zz} , T_{xz}) dari 837 hari pada tanggal 01 Nopember 2009 sampai 01 Agustus 2012 dan 422 pada fase orbit yang lebih rendah pada tanggal 01 September 2012 sampai dengan 24 Juni 2013
 - d. Data terrestrial yang terdiri dari data geoid laut DTU12 dan data ketinggian benua dari EGM2008.Data *EIGEN-6C4* memiliki derajat dan orde hingga 2190 yang terdiri dari koefisien $\bar{C}_{n,m}$ dan koefisien $\bar{S}_{n,m}$ (Foerste, et al., 2014).
3. Undulasi (N) dari *EGM2008 (Earth Gravity Model 2008)* yang didapat dengan menggunakan program *hsynth_WGS84.exe* (Pavlis, 2013).

3.3 Diagram Alir Penelitian

Nilai ketinggian elipsoid (h), posisi (lintang (ϕ) dan bujur (λ)), beserta data *Spherical Harmonics Coefficient* (SHC) dari model *EIGEN-6C4* (Gambar 3.3) di-*sampling* pada setiap titik koordinat dengan jarak spasi 0,01 derajat dengan menggunakan program GrafLab (Bucha & Janák, 2013; Bucha & Janák, 2014) sehingga didapat nilai gayaberat (g) dan tensor Marussi (Gambar 3.4) berupa $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$, dan T_{zz} . Nilai gayaberat dari model *EIGEN-6C4* dijadikan nilai gayaberat hasil observasi (g_{obs}).



Gambar 3.2 Diagram alir penelitian



Gambar 3.3 Program GrafLab berbahasa MATLAB (Bucha & Janák, 2013).

Untuk menghitung ketinggian ellipsoid (h) digunakan persamaan:

$$h = N + H \quad (3.1)$$

dimana,

h = ketinggian elipsoid dalam meter

N = Undulasi dalam meter

H = ketinggian ortometrik dalam meter

Data ketinggian ortometrik (H) adalah data ketinggian yang didapat dari *EARTH2014* (Hirt and Rexer, 2015). Sedangkan Undulasi (N) didapat dari *EGM2008*.

Untuk perhitungan koreksi ABL dan perhitungan nilai ABL (Gambar 3.2) digunakan persamaan 2.16 sampai persamaan 2.23, kemudian peta ABL diberi sayatan yaitu AA', BB', CC', DD', EE' untuk proses analisis spektrum dengan menggunakan program MATLAB (Gambar 3.4). Analisis spektrum bertujuan

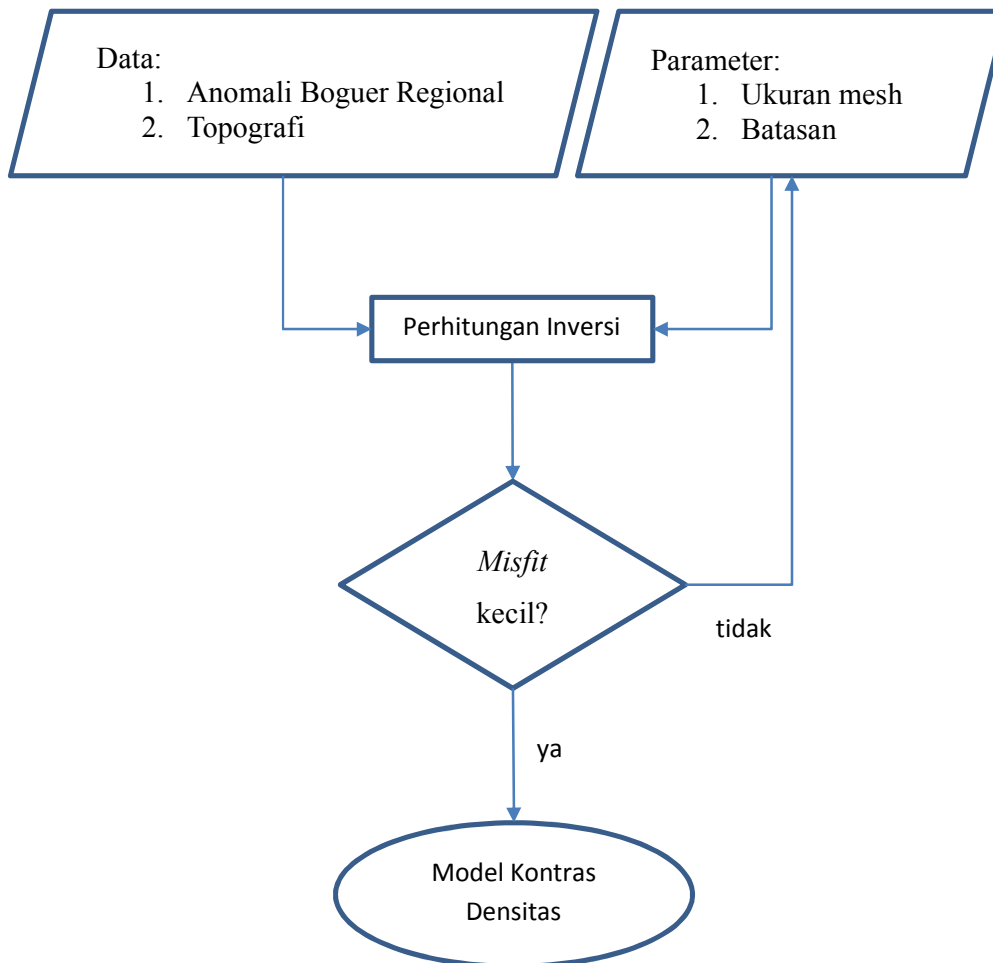
untuk mendapatkan nilai k_c , setelah itu lebar jendela dihitung dengan menggunakan persamaan 2.29, kemudian dilakukan *filter* dengan metode *moving average* menggunakan persamaan 2.30. Selanjutnya, pemisahan anomali regional dan residual menggunakan persamaan 2.31.

```
clc;
clear all;
AA = xlsread('AA.xlsx',1);
x= AA(:,1); %jarak lintasan
y= AA(:,2); %nilai anomali bouguer
sp = 500; %spasi dari tiap titik
xi = min(x):sp:max(x);
yi = spline(x,y,xi);
plot(xi, yi, 'ro');
data = yi;
n = length(y);
D=fft(data);
D(1)=0;
pwr=abs(D(1:round(n/2))).^2;
freq=(1:length(pwr))/(n/2)*1/(2*sp);
semilogy(freq, pwr, '.');
ylabel('power spektrum');
xlabel('cycle/meter');
title('Analisis power spektrum lintasan AA');
```

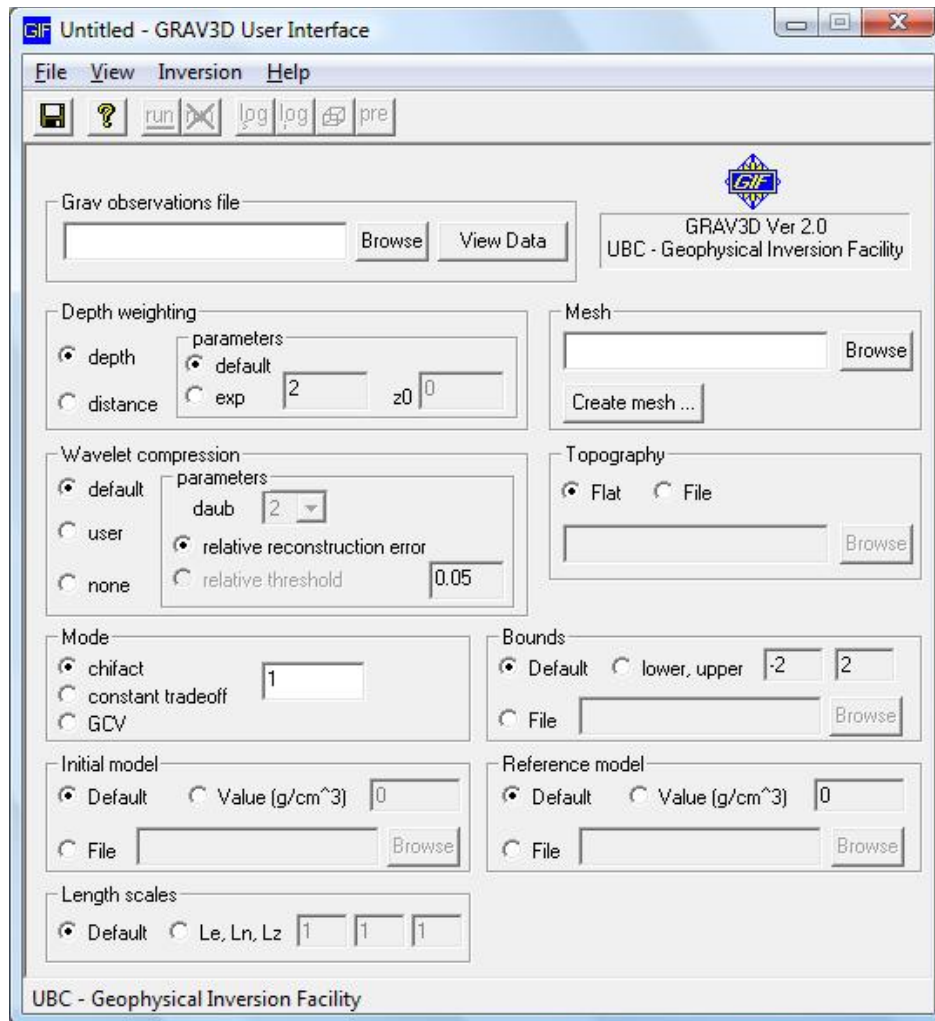
Gambar 3.4 Contoh proses analisis spektrum untuk lintasan AA' dengan menggunakan program MATLAB.

3.4 Diagram Alir Pemodelan Inversi 3D

Pembuatan model kontras densitas (Gambar 3.5) pada metode inversi didapat menggunakan perangkat lunak GRAV3D (Gambar 3.6). Data masukan untuk proses pemodelan terdiri dari anomali Bouguer regional dan topografi. Di bagian lain harus dipersiapkan terlebih dahulu model *mesh* dan batasan densitasnya.



Gambar 3.5 Diagram alir pemodelan inversi 3D



Gambar 3.6 GRAV3D *User Interface*. Versi pendidikan (gratis) dengan keterbatasan jumlah data dan sel.

Langkah-langkah menjalankan program GUI GRAV3D adalah sebagai berikut:

1. Buat *file input* untuk *Grav observation file*. *File* di *save* dengan *extension* *.grv.
2. Buat *file input* untuk *Mesh*. *File* di *save* dengan *extension* *.mesh.
3. Untuk *Mode* pilih *radio button* GCV
4. Pilih menu *File* di bagian atas GUI GRAV3D, kemudian pilih *Save*.

5. Jalankan inversi dengan memilih menu *Inversion* di bagian atas GUI GRAV3D,
6. Pilih *Run*,
7. Setelah proses inversi selesai, hasilnya dapat ditampilkan dengan memilih menu *View* di bagian atas GUI GRAV3D, kemudian pilih *Model*. Untuk membandingkan hasil observasi dan perhitungan inversi pilih menu *Predicted*.

Proses ini tidak cukup sekali karena perlu penyesuaian dengan prinsip geologi dengan nilai *misfit* sekecil mungkin.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Geologi Regional Jawa Timur

Peta geologi yang digunakan dalam interpretasi terdiri dari tiga peta. Peta pertama (Gambar 4.1) adalah peta geologi Jawa Bagian Timur dengan skala 1:500.000 (Gafoer & Ratman, 1999), kedua, Peta lembar Surabaya-Sapulu (Supardjono, *et al.*, 1992), dan ketiga, peta lembar Malang (Santosa, *et al.*, 1992). Peta lembar Surabaya-Sapulu digabungkan dengan peta lembar Malang (Gambar 4.2).

Kondisi struktur geologi di permukaan memperlihatkan struktur perlipatan yang sumbu lipatannya berjurus barat-timur. Patahan geser banyak terlihat di berbagai tempat, hampir seluruhnya memiliki jurus timur laut

Dalam peta tersebut tidak menunjukkan adanya patahan Watukosek di permukaan daerah Watukosek sebelah timur laut gunung penanggungan, namun beberapa peneliti meyakini ketika terjadi gempa tektonik daerah selatan Yogyakarta pada tanggal 27 Mei 2006 mengaktifkan kembali patahan tersebut sehingga muncul dipermukaan berupa rekahan yang diyakini sebagai hasil pergeseran sesar geser yang membentang dari dari penanggungan hingga daerah hilir sungai Buntung daerah segorotambak seperti yang digambarkan oleh Mazzini *et al.*, 2007.

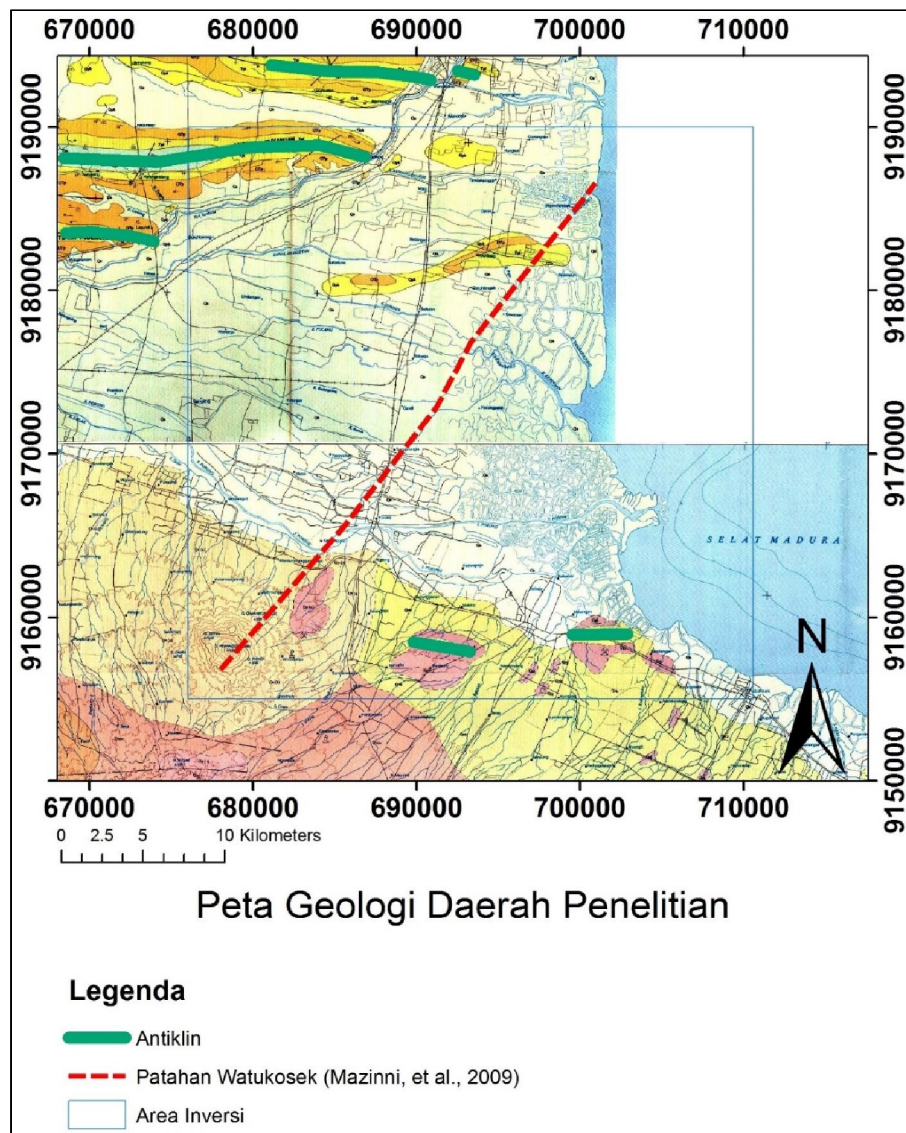
4.2 Peta Ketinggian, Gayaberat, dan Anomali Bouguer

Peta ketinggian (Gambar 4.2) diatas geoid atau yang disebut ketinggian ortometrik didapat dari *EARTH2014*. Data ketinggian ini digunakan untuk proses perhitungan ketinggian elipsoid dan koreksi ABL.

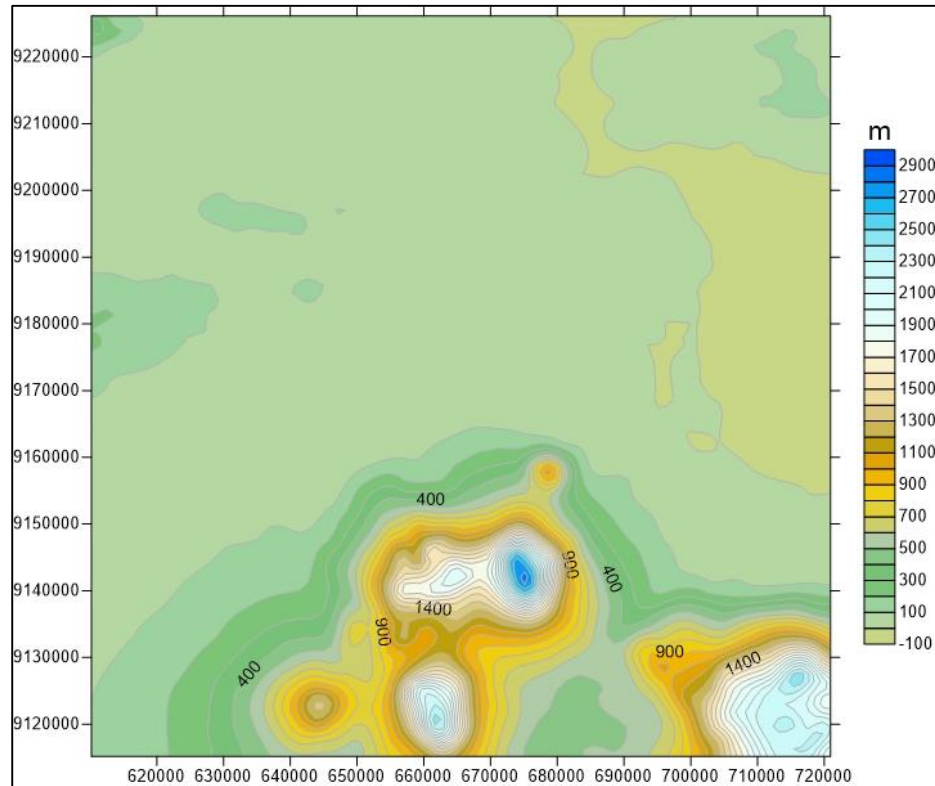
Perbandingan data gayaberat (Gambar 4.3) antara *EGM2008* (Pavlis, *et al* 2008; Pavlis, *et al* 2012) dengan dengan *EIGEN-6C4* yang dihitung pada 1100 meter di atas ellipsoid terlihat bahwa keduanya tidak memiliki perbedaan jauh dalam kaitannya dengan struktur geologi. Dalam tesis ini menggunakan data *EIGEN-6C4* berdasarkan hasil evaluasi Förste, *et al* (2014) menggunakan

GPS/levelling dan analisis statistik perbandingan dengan *GEOHALO Airborne Gravimetry* yang menunjukkan bahwa *EIGEN-6C4* memiliki RMS error lebih kecil dibandingkan dengan *EGM2008*.

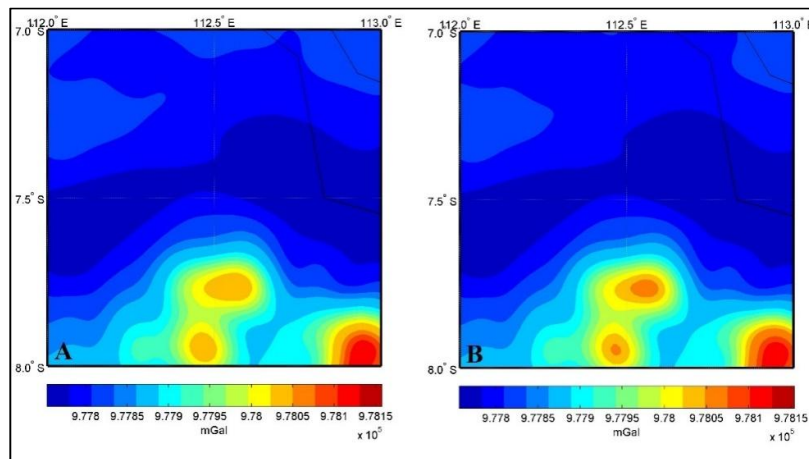
Peta gayaberat (Gambar 4.4) merupakan hasil perhitungan dari persamaan 2.11 model *EIGEN-6C4* pada 1100 meter ketinggian ellipsoid dengan spasi sampling 0,01 derajat.



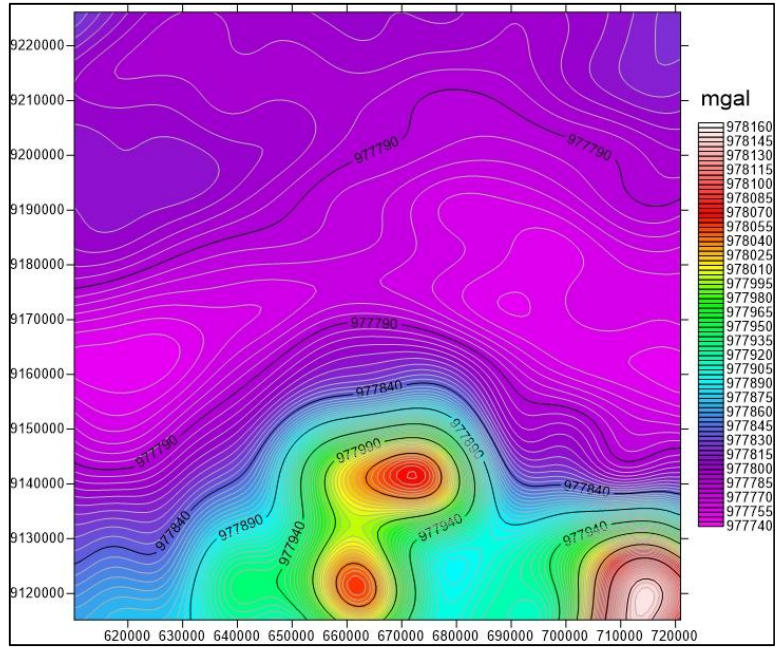
Gambar 4.1 Peta geologi dengan skala 1:100.000 hasil modifikasi dari Santosa *et al* (1992) dan Supardjono *et al* (1992).



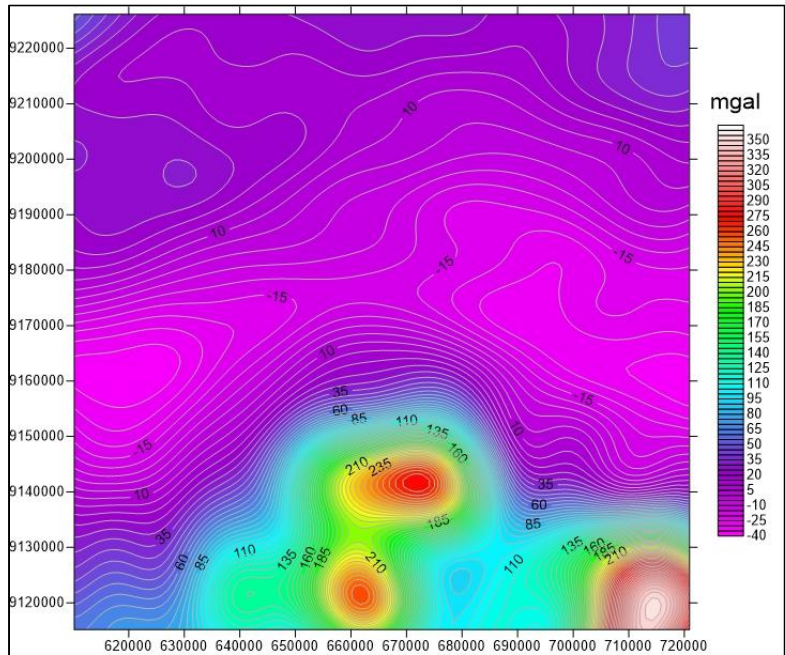
Gambar 4.2 Ketinggian ortometrik daerah penelitian dari data *EARTH2014* (Hirt dan Rexer, 2015).



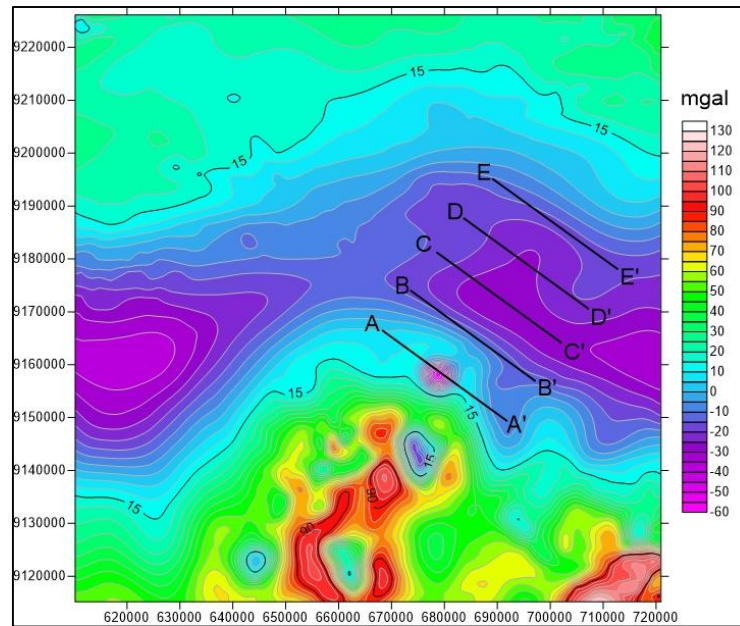
Gambar 4.3 Peta gayaberat A) model *EGM2008*, dan B) model *EIGEN-C4* pada daerah yang sama.



Gambar 4.4 Peta Gayaberat hasil model *EIGEN-6C4* pada daerah penelitian



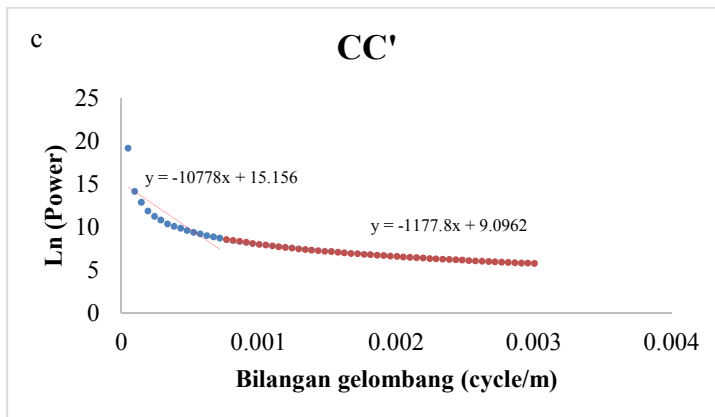
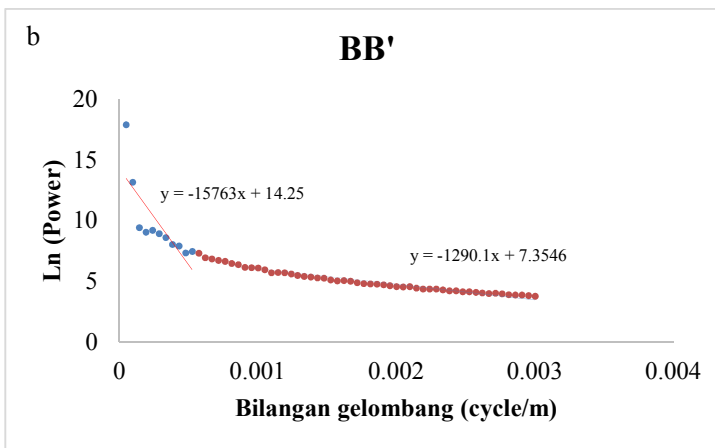
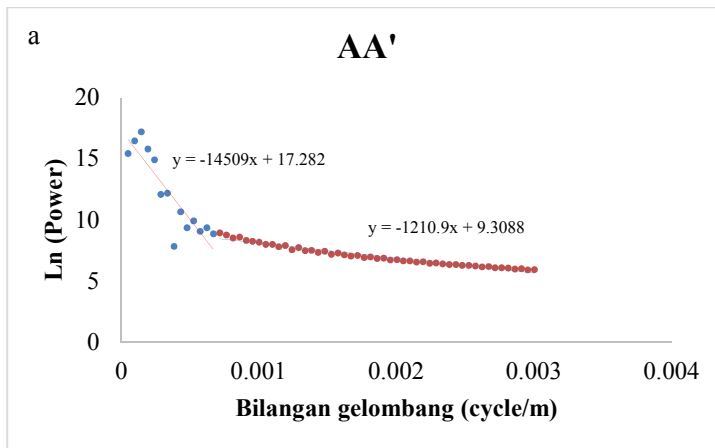
Gambar 4.5 Peta Anomali Udara Bebas (AUB) daerah penelitian.

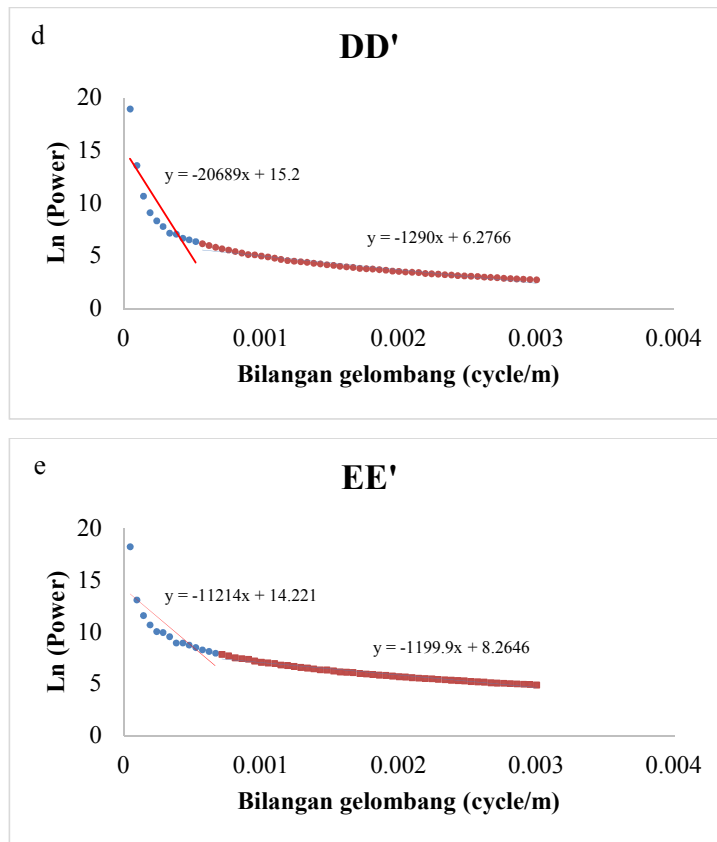


Gambar 4.6 Peta Anomali Bouguer Lengkap dan penampang AA', BB', CC', DD', dan EE' untuk analisis spektrum.

4.3 Plot Power Spectrum dan low-pass filter

Analisis spektrum menggunakan aplikasi *Fast Fourier Transform (FFT)* yang bertujuan untuk mengetahui batas anomali regional, residual, maupun *noise*, serta untuk mengetahui kedalaman rata-rata benda anomali dari peta anomali Bouguer. Dalam peta Anomali Bouguer Lengkap (gambar 4.6) dibuat lima penampang AA', BB', CC', DD', dan EE' untuk dilakukan plot $\ln(\text{power})$ terhadap bilangan gelombang (k) Perpotongan antara gradien regional dan gradien residual (Gambar 4.7) adalah nilai bilangan gelombang *cut-off* (k_c). Lebar jendela dihitung dengan persamaan 2.30 dan dirata-ratakan.





Gambar 4.7 Plot $\ln(\text{power})$ terhadap bilangan gelombang (k) untuk penampang AA' (a), BB' (b), CC' (c), DD', (d), dan EE' (e). Garis merah adalah gradien regional dan garis biru adalah gradien residual. Semakin curam gradien itu, semakin dalam anomalnya.

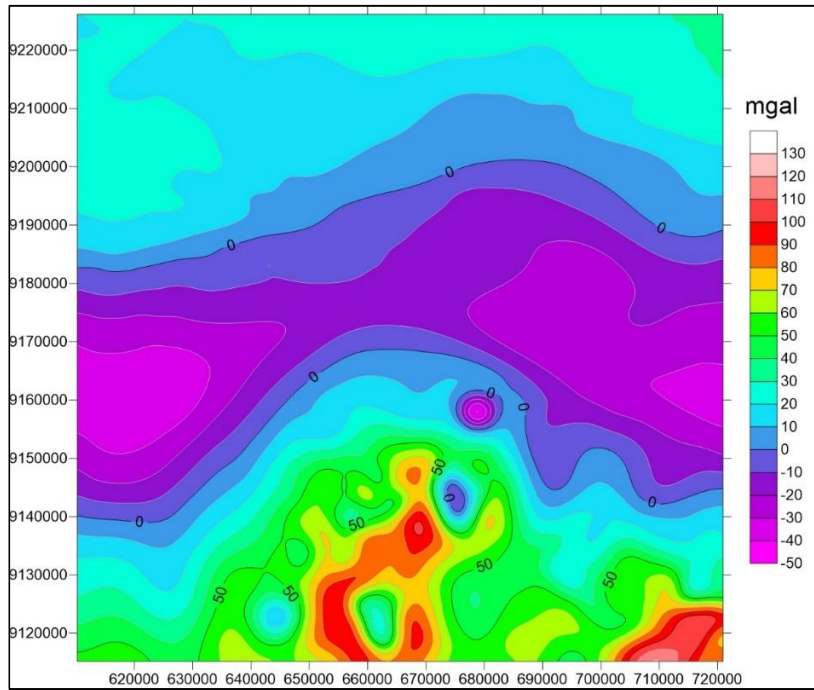
Tabel 4.1 Perhitungan nilai lebar jendela.

	Penampang				
	AA'	BB'	CC'	DD'	EE'
m-reg	14509	15763	10778	20689	11214
m-res	1210,9	1290,1	1177,8	1290	1199,9
reg-d	1154,59	1254,38	857,686	1646,378	892,3818
res-d	96,36036	102,6629	93,72635	102,6549	95,48501
c-reg	17,282	14,25	15,156	15,2	14,221

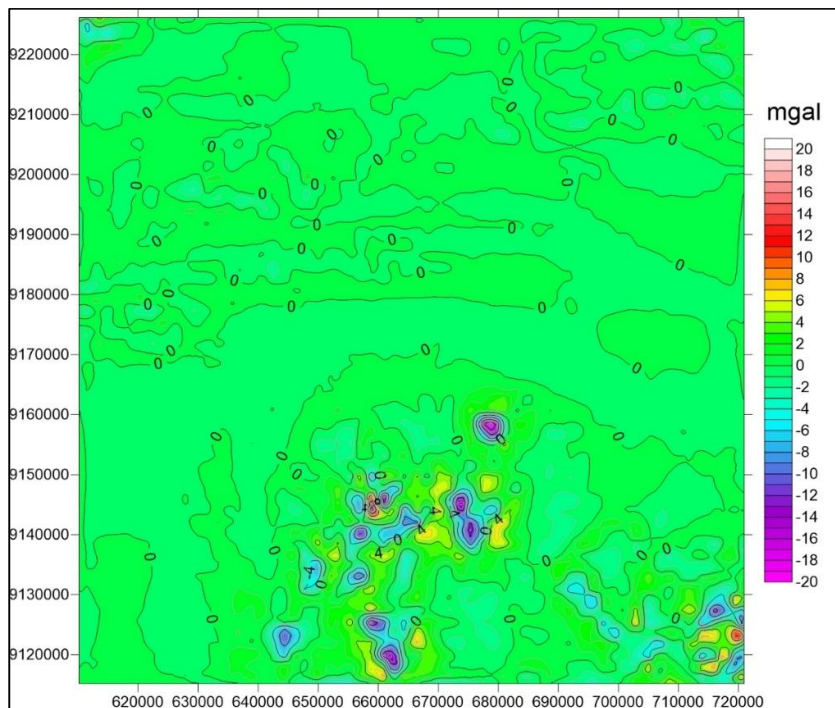
c-res	9,3088	7,3546	9,0962	6,2766	8,2646
k_c	0,0006	0,000476	0,000631	0,00046	0,000595
N	33,357	41,97842	31,68487	43,47894	33,62467

Dimana, m-reg, m-res, reg-d, res-d, c-reg, c-res, k_c , dan N berturut-turut adalah gradien regional, gradien residual, kedalaman anomali regional, kedalaman anomali residual, konstanta gradien regional, konstanta gradien residual, bilangan gelombang *cut-off*, dan lebar jendela. Kedalaman anomali dihitung dengan persamaan 2.29.

Lebar jendela rata-rata pada Tabel 3 adalah 36,82478, sehingga untuk proses pemisahan digunakan lebar jendela sebesar 37. Kemudian anomali regional (gambar 4.8) diperoleh dari persamaan 2.30 dibantu dengan perangkat lunak *surfer*, sedangkan anomali residual (gambar 4.9) didapat dari persamaan 2.31. Pada peta anomali residual lebih menunjukkan anomali topografi dibandingkan sinyal yang diharapkan, sedangkan pada peta anomali regionalnya dapat menunjukkan keadaan struktur bawah permukaan, oleh sebab itu pada pemodelan inversi digunakan data anomali Bouguer regional.



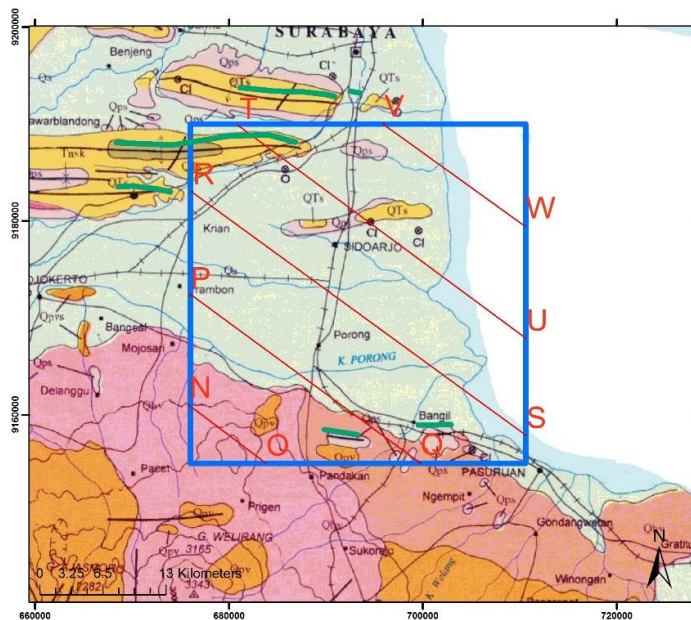
Gambar 4.8 Peta Anomali Bouguer Regional



Gambar 4.9 Peta Anomali Bouguer Residual.

4.4 Pemodelan Gayaberat Inversi 3D

Pemodelan gayaberat inversi 3D dalam tesis ini menggunakan program GRAV3D. Data masukan untuk dalam proses inversi adalah anomali Bouguer regional dan topografi daerah yang akan di inversi, dikarenakan keterbatasan perangkat lunak GRAV3D, ukuran mesh hanya 23 x 24 x 15 dengan spasi 2000 x 2000 x 1000 m. Daerah yang akan di inversi adalah daerah sekitar patahan Watukosek (Gambar 4.10). Data anomali Bouguer regional (Gambar 4.11) dan *absolute error* (Gambar 4.12). Hasil inversi terdiri dari dua hal yang utama yaitu *Predicted data* (Gambar 4.13) dan model inversi (Gambar 4.14). Gambar 4.14 sampai gambar 4.19 yang ditampilkan dalam model inversi berada di kedalaman 900 hingga 10900 meter dari atas permukaan model *mesh*, hal ini dilakukan karena anomali bersifat regional yang hanya memperlihatkan daerah yang dalam, bila dilihat dari hasil analisis spektrum kedalaman rata-rata anomali regional berkisar antara 800 hingga 1500 meter.

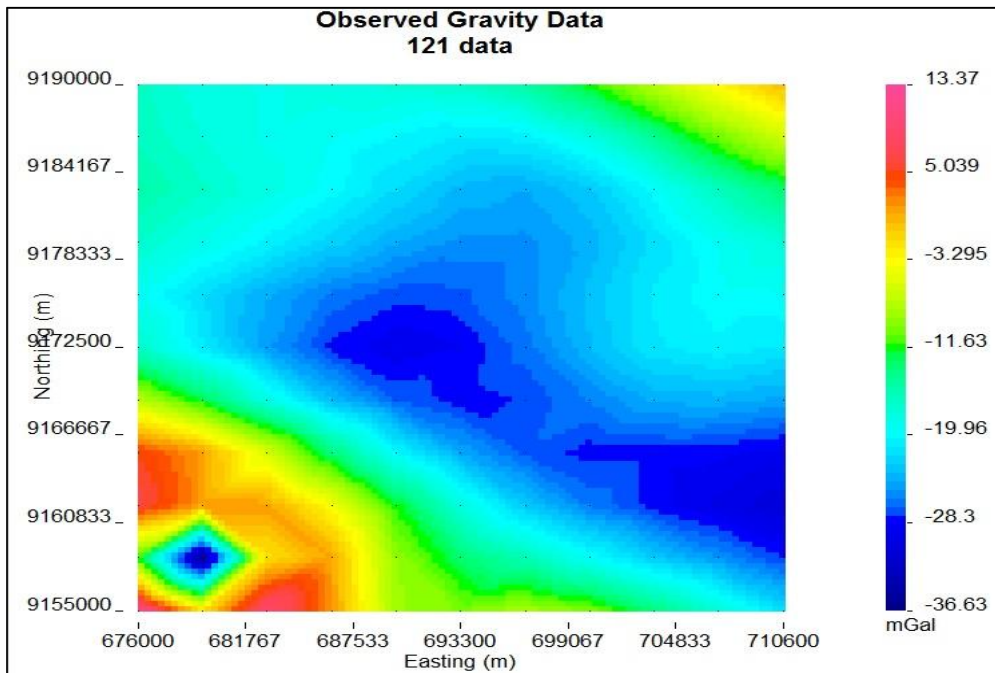


Gambar 4.10 Daerah yang akan dibuat model gayaberat inversi 3D dan sayatan N-O, P-Q, R-S, T-U, dan V-W untuk interpretasi struktur geologi.

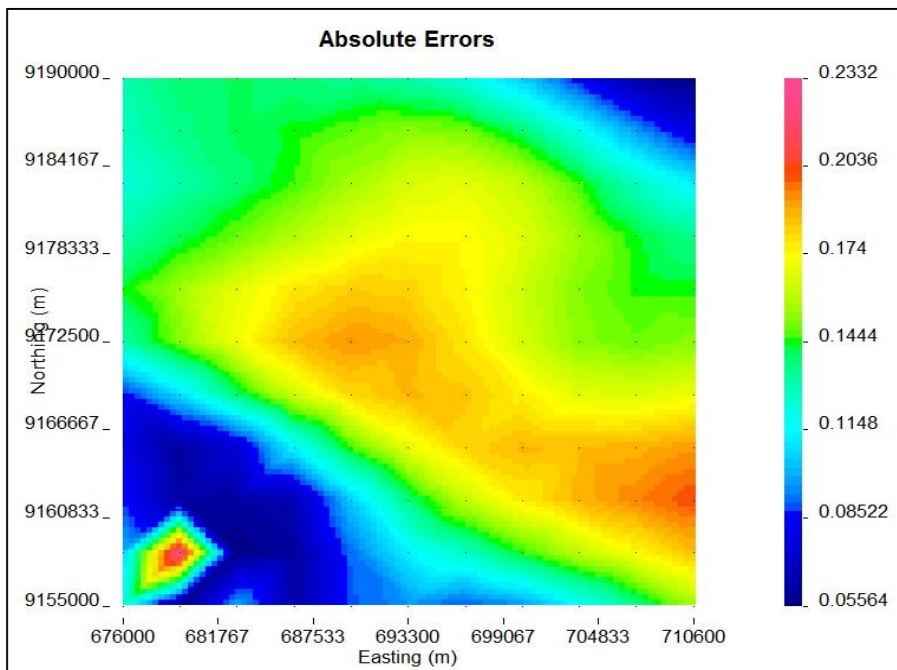
Ketidakhadiran anomali positif intrusi Gunung Penanggungan menunjukkan error yang sangat besar (Gambar 4.12). Hal ini disebabkan ketika dilakukan koreksi Bouguer menghasilkan anomali negatif yang sangat besar.

Hasil inversi yang memperlihatkan kondisi geologi daerah penelitian berupa perlipatan dimana antiklin yang memiliki nilai kontras densitas negatif berkisar antara -0,1 gr/cc hingga -0,03 gr/cc tetapi lebih besar dari kontras densitas yang dimiliki sinklin yang berkisar antara -0,2 gr/cc hingga -0,15 gr/cc, sedangkan Gunung Penanggungan memiliki kontras densitas sekitar 0,1 gr/cc diatas densitas rata-rata (2,67 gr/cc).

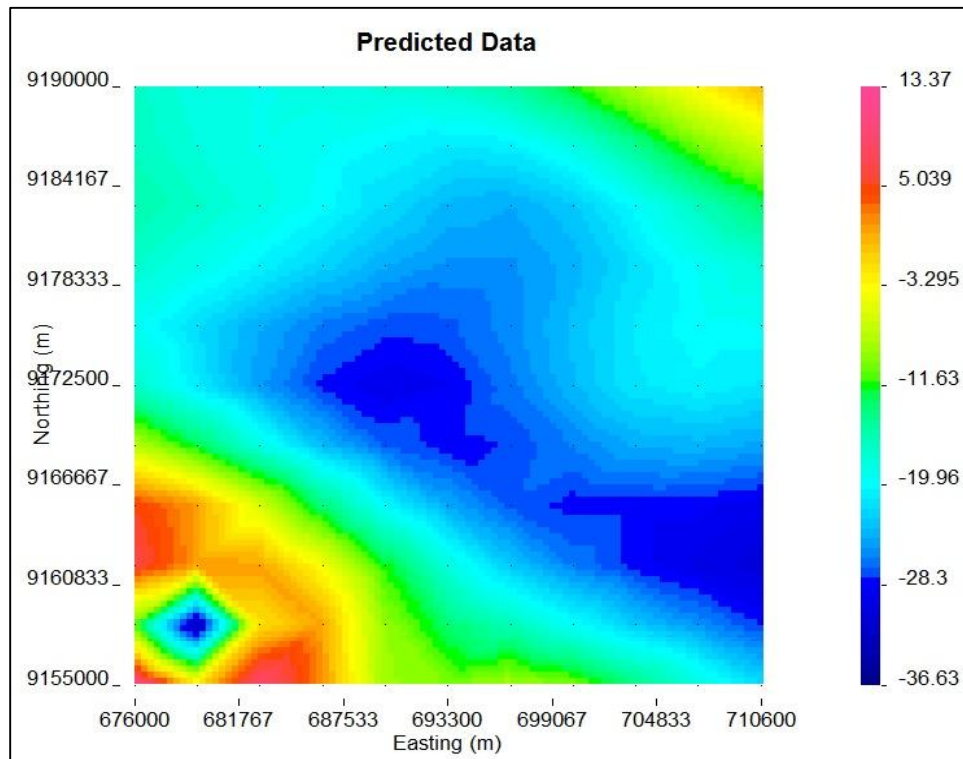
Hasil model inversi 3D (Gambar 4.14) didapat struktur perlipatan, intrusi Gunung Penanggungan, dan cekungan, dan patahan. Dari model tersebut dibuat 5 sayatan yang terdiri dari sayatan N-O, P-Q, R-S, T-U, dan V-W (. Pada sayatan N-O (Gambar 4.15) adalah intrusi Gunung Penanggungan dan beberapa patahan diinterpretasikan sebagai patahan normal, namun sulit untuk menentukan patahan geser Watukosek. Pada sayatan P-Q (Gambar 4.16) masih terlihat intrusi batuan Gunung Penanggungan dan terlihat cekungan dan kemungkinan terdapat patahan-patahan sedangkan pada sayatan R-S (Gambar 4.17), T-U (Gambar 4.18), dan V-W (Gambar 4.19) diinterpretasikan terdapat suatu patahan namun belum bisa dipastikan merupakan kemenerusan dari patahan Watukosek.



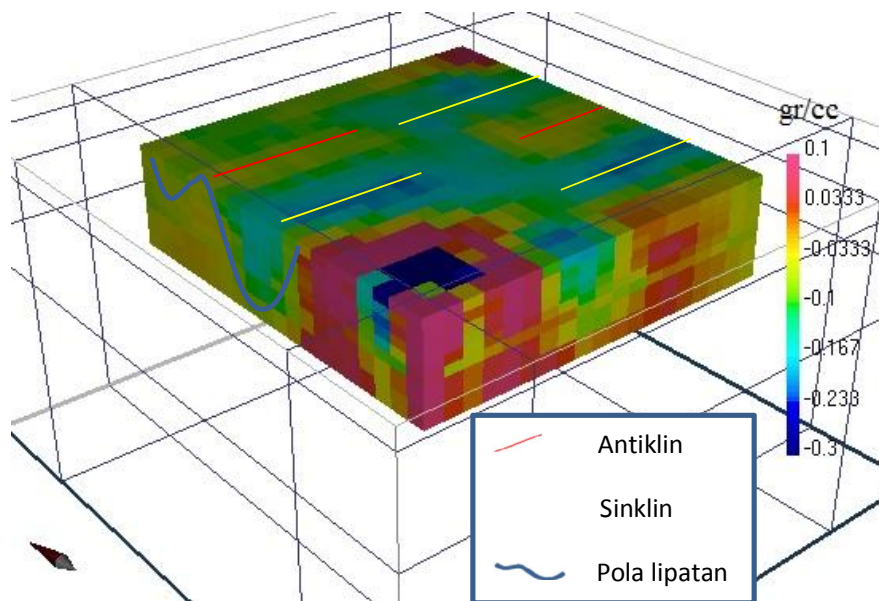
Gambar 4.11 Data masukan anomali Bouguer regional sebelum proses inversi.



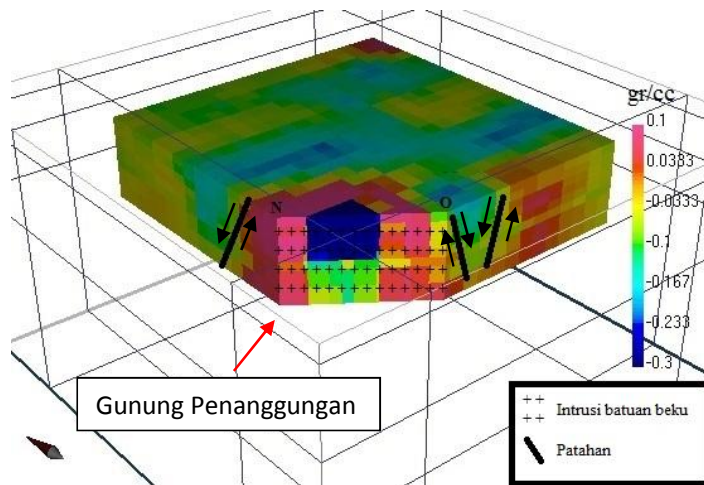
Gambar 4.12 Pengurangan error pada data masukan anomali Bouguer regional.



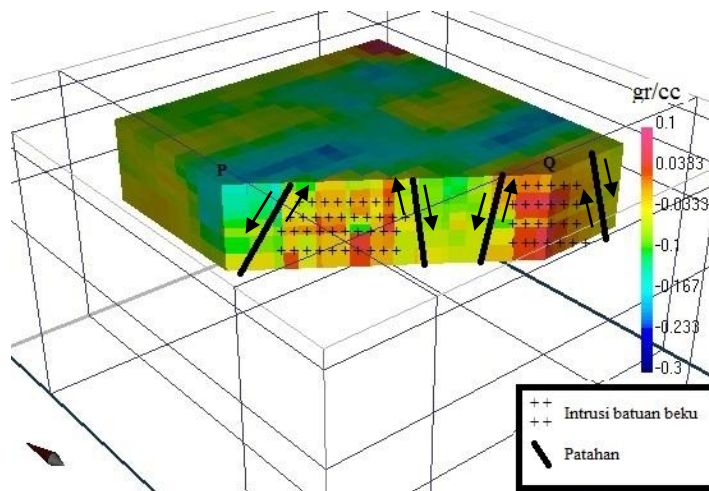
Gambar 4.13 Data masukan setelah pengurangan error.



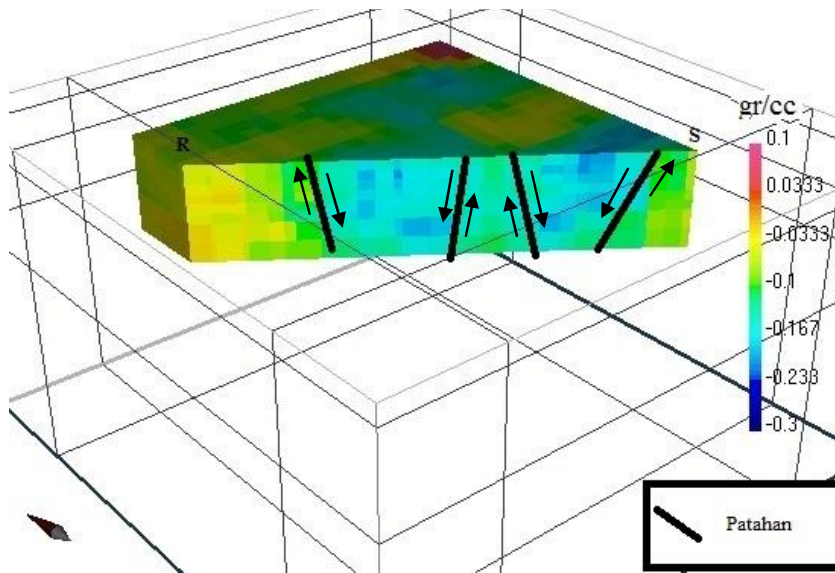
Gambar 4.14 Model gayaberat inversi 3D pada kedalaman 900 m hingga 10900 m terhadap bidang horizontal *mesh* paling atas.



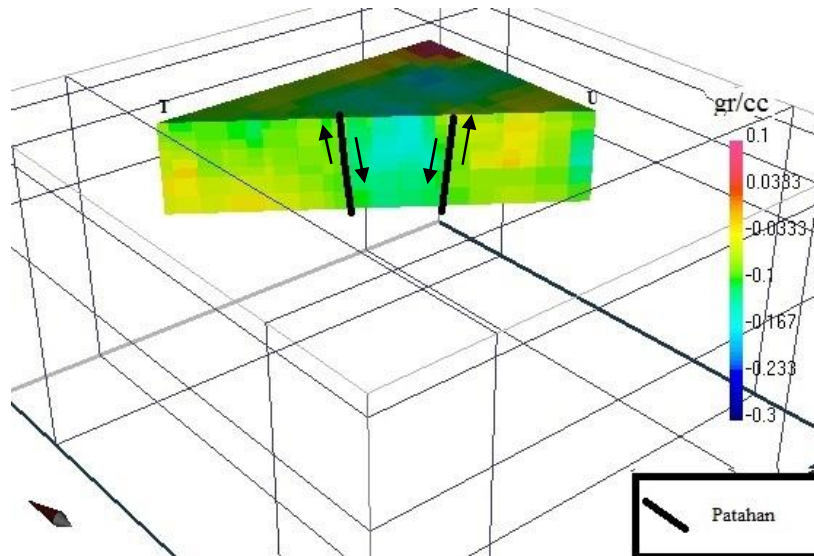
Gambar 4.15 Model kontras densitas 3D. Interpretasi struktur geologi untuk sayatan N-O berupa intrusi batuan beku Gunung Penanggungan, patahan normal, dan *graben*.



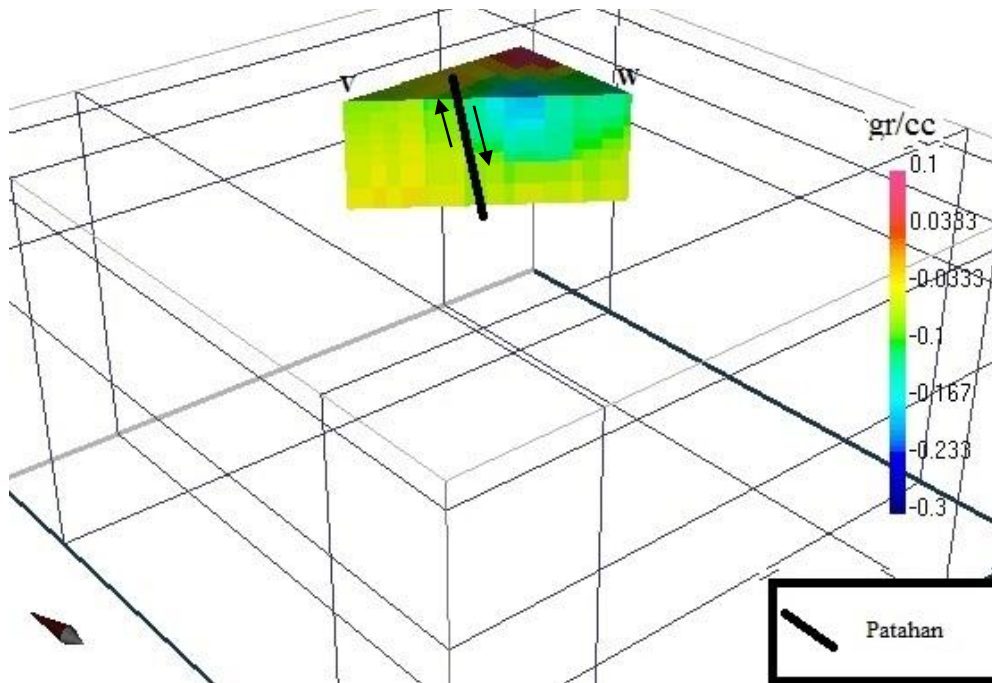
Gambar 4.16 Model kontras densitas 3D untuk sayatan P-Q. Intrusi batuan beku, patahan normal dan *graben*.



Gambar 4.17 Model kontras densitas 3D untuk sayatan R-S. Struktur *horst* dan *graben* yang berasosiasi dengan patahan geser regional dan proses pengendapan sedimen.



Gambar 4.18 Model kontras densitas 3D untuk sayatan T-U. Struktur *graben* yang memotong struktur perlipatan.

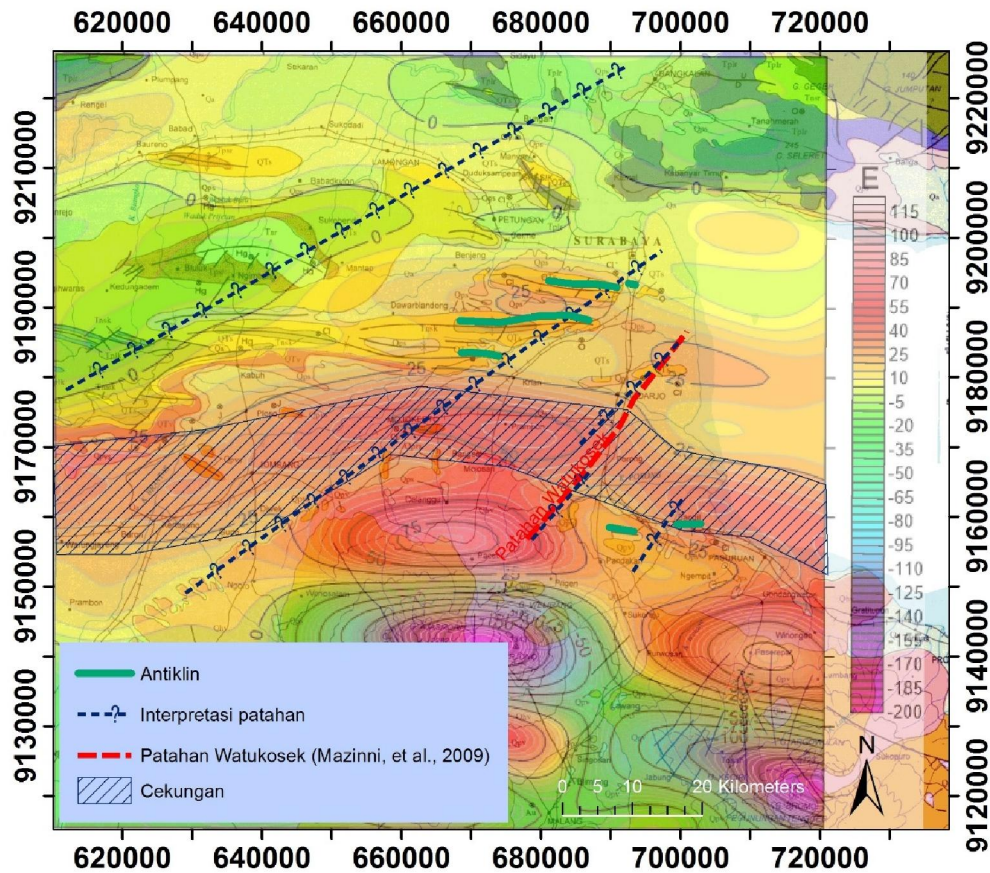


Gambar 4.19 Model kontras densitas 3D untuk sayatan V-W.

Struktur half graben (Gambar 4.18) diakibatkan oleh deformasi *extesional* dan secara bersamaan dengan proses pengendapan sehingga cekungan terbentuk.

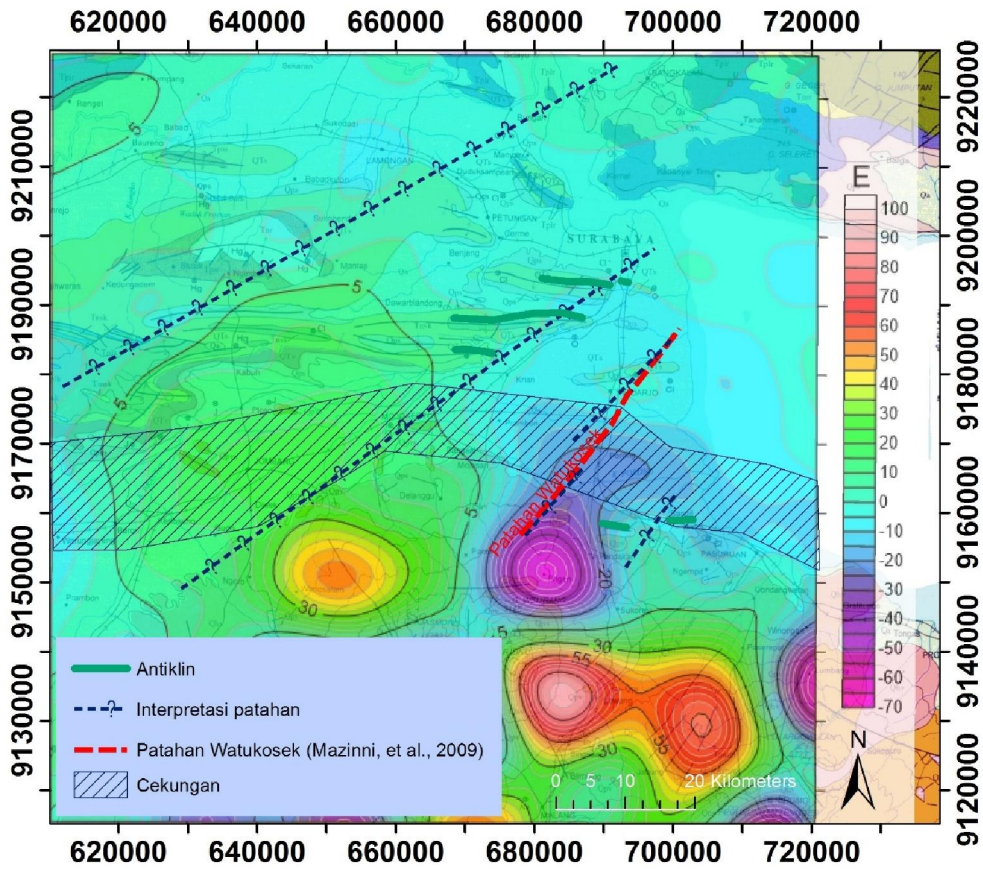
4.5 Peta Tensor Marussi

Peta *disturbing tensor* (Tensor Marussi) yang terdiri dari 6 peta (T_{xx} , T_{xy} , T_{xz} , T_{yy} , T_{yz} , T_{zz}). Sesuai dengan system *LNOF* bahwa sumbu x mengarah ke utara, sumbu y mengarah ke barat, dan sumbu z mengarah ke atas. Peta yang dihasilkan kemudian di-interpretasi struktur geologinya yang di-*overlay* dengan peta geologi regional Jawa Bagian Timur (modifikasi dari Gafoer & Ratman, 1999).



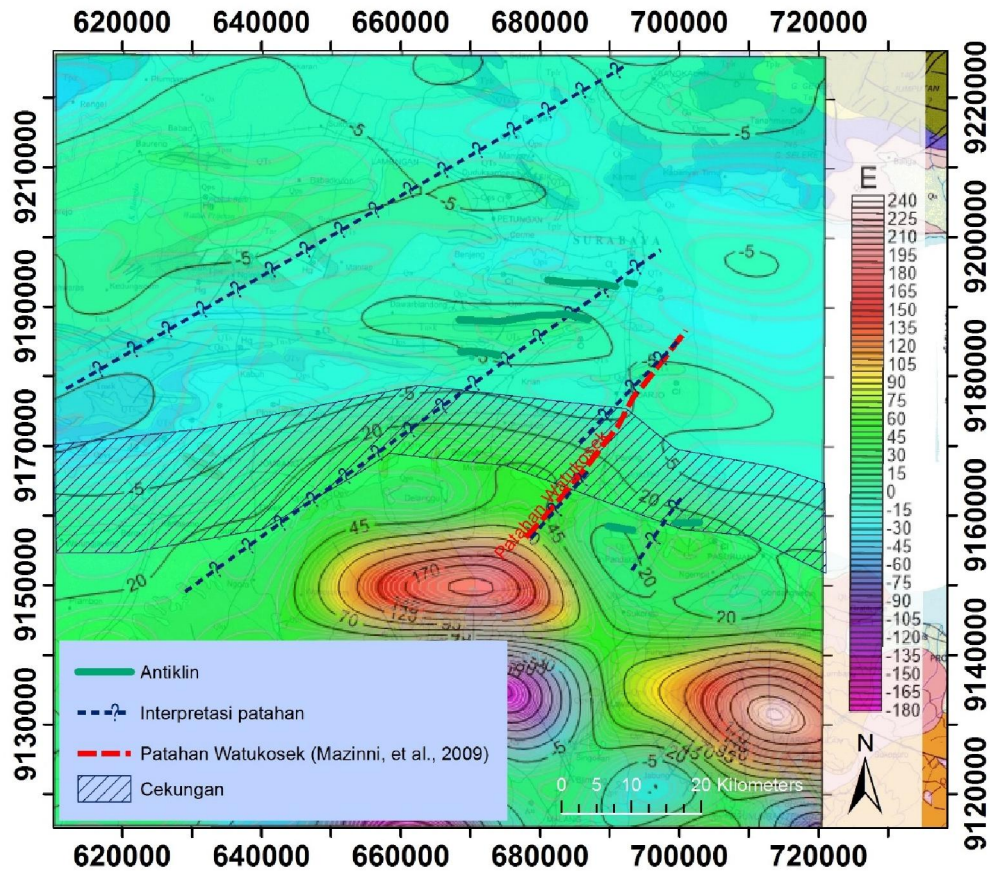
Gambar 4.20 Peta T_{xx} . Pola anomali berarah ke sumbu x-x atau utara sehingga struktur geologi terlihat apabila dilihat dari sisi tegak lurus nya, arah barat atau timur.

Peta T_{xx} (Gambar 4.20) menggambarkan kondisi struktur geologi berupa perlipatan (antiklin dan sinklin). Zona cekungan terbentuk akibat proses perlipatan di ikuti dengan pengendapan sedimen terus-menerus. Hal ini menyebabkan tekanan formasinya tinggi. Selain perlipatan terdapat pula patahan-patahan geser yang terdapat pada peta geologi maupun hasil interpretasi. Patahan geser *sinistral* (mengiri) di Jawa Timur secara regional memiliki jurus ke arah timur laut. Patahan geser ini diikuti deformasi ekstensional dan pengendapan sedimen, sehingga terbentuklah struktur *graben* dan *horst*. Seperti daerah cekungan tepat di utara Gunung Penanggungan yang kemungkinan terdapat banyak struktur tersebut.



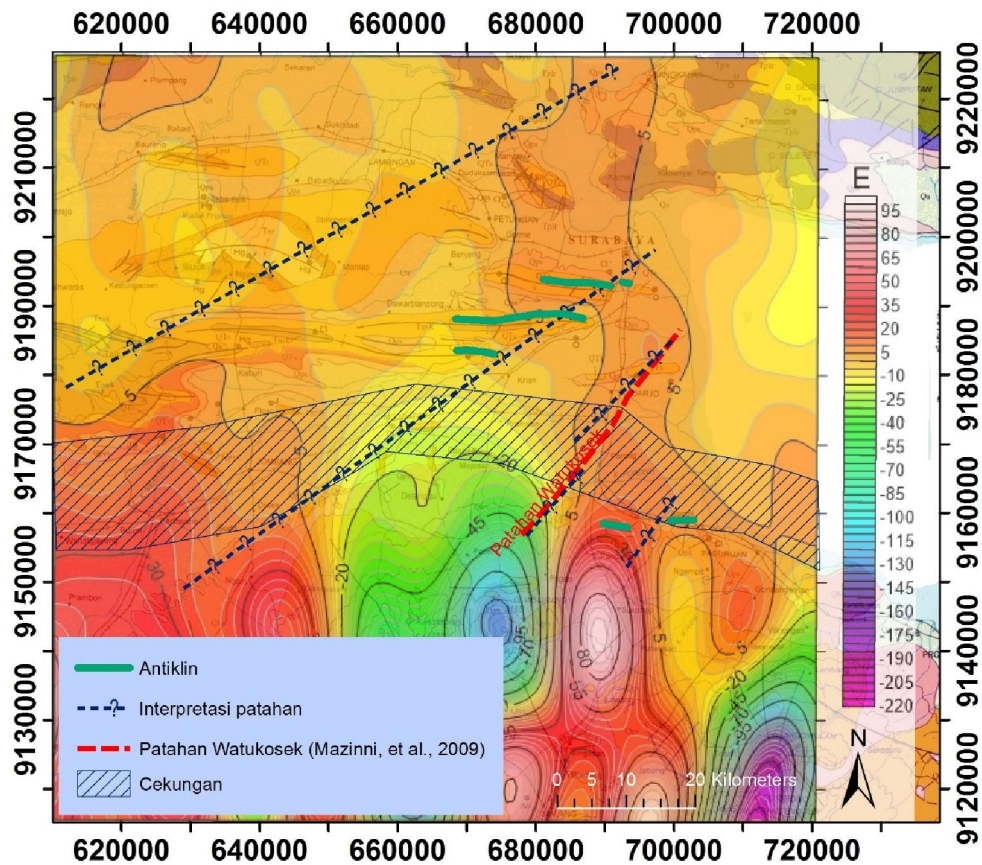
Gambar 4.21 Peta T_{xy} . Pola anomali berarah x-y (utara-barat) sehingga struktur geologi terlihat dari sisi tegak lurus nya, timur laut atau barat daya.

Berdasarkan anomalnya Peta T_{xy} (Gambar 4.21) terdapat kesesuaian dengan Patahan Watukosek, namun tidak menunjukkan adanya kemenerusan. Interpretasi struktur dari peta ini adalah pada bidang batas antara anomali positif dan negatif atau tinggi dengan rendah, namun hanya untuk jurus yang berarah timur laut atau barat daya.



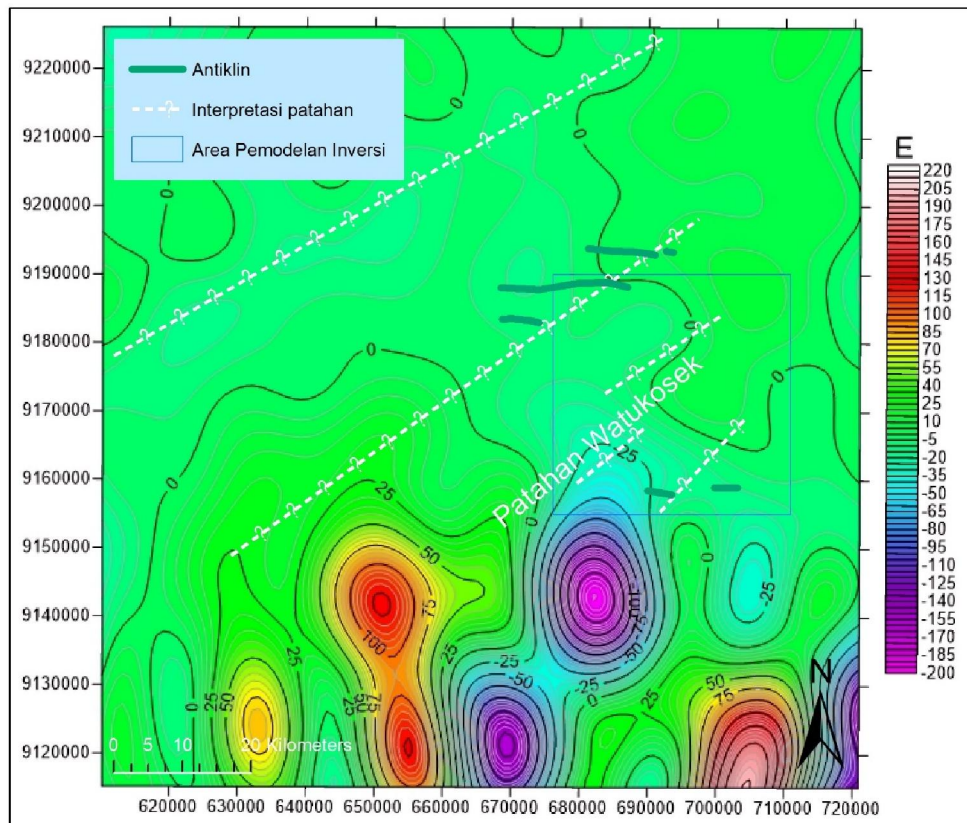
Gambar 4.22 Peta T_{xz} . Keadaan struktur geologi yang mirip dengan peta T_{xx} , namun dipengaruhi oleh anomali arah vertikal ke atas, sehingga interpretasi pada peta ini untuk melihat struktur yang memiliki bidang miring dengan jurus barat-timur.

Patahan watukosek pada peta T_{xz} (Gambar 4.22) dapat terlihat namun tidak dapat menunjukkan kemenerusannya. Hal ini dilihat dari kemiringan kontur anomali bidang patahan Watukosek yang memiliki *dip* ke arah tenggara, namun ketika ditarik garis seperti yang digambarkan Mazinni *et al* (2009) kemiringan kontur berubah dengan *dip* ke arah barat laut.



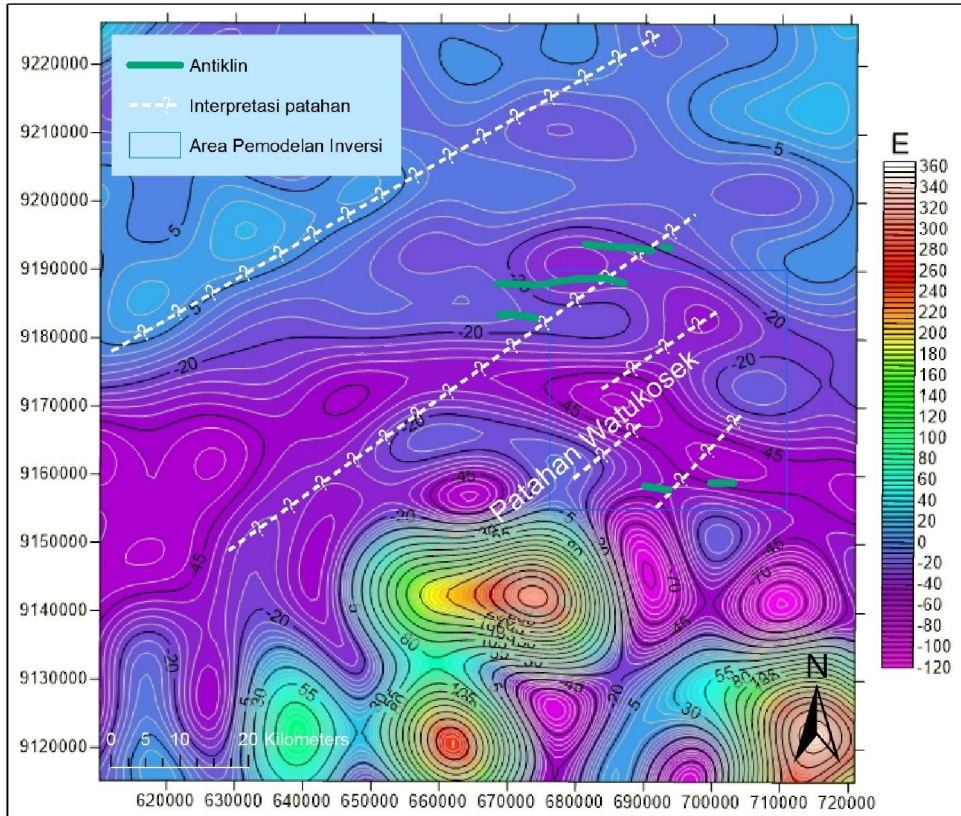
Gambar 4.23 Peta T_{yy} . Pola anomali menunjukkan arah barat-timur, sedangkan untuk interpretasi struktur geologi dilihat dari arah utara-selatan.

Struktur geologi yang dapat di-interpretasikan (Gambar 4.23) merupakan struktur yang memiliki jurus ke utara sampai timur laut. *Graben*, *half-graben*, *horst*, dan patahan normal dapat dilihat dan dipadukan dengan hasil dari pemodelan inversi (Gambar 4.15 hingga Gambar 4.19).



Gambar 4.24 Peta T_{yz} . Anomali menunjukkan struktur yang memiliki kemiringan dengan jurus utara-selatan.

Patahan watukosek pada peta T_{yz} (Gambar 4.24) dapat diinterpretasikan namun sama halnya dengan yang telah dibahas pada Gambar 4.22 bahwa kemiringan kontur anomali patahan Watukosek di dekat Gunung Penanggungan memiliki dip ke arah tenggara sedangkan apabila diteruskan menuju timur laut *dip* ini berubah ke arah barat laut.

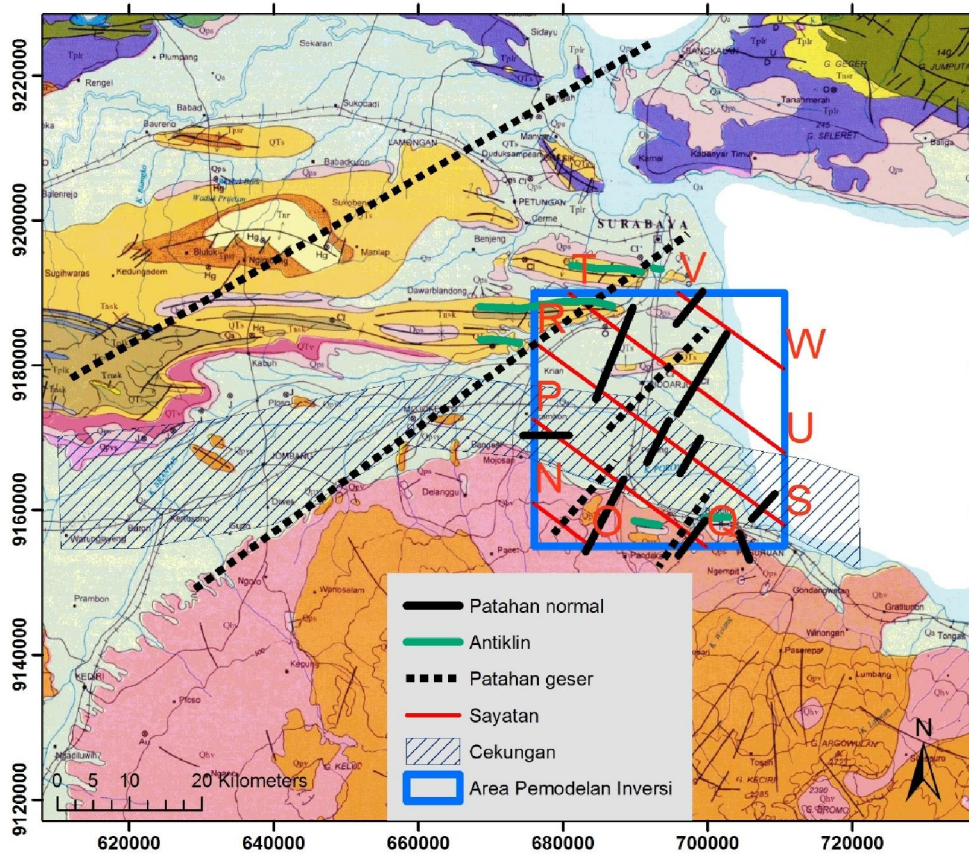


Gambar 4.25 Peta T_{zz} . Anomali yang paling mendekati dengan keadaan struktur geologi.

Peta T_{zz} (Gambar 4.25) memperlihatkan pola anomali yang mirip dengan anomali Bouguer residual namun anomali cekungan, lipatan, dan patahan terlihat lebih jelas.

4.6 Interpretasi Struktur Geologi

Struktur geologi yang di interpretasi dalam pemodelan gayaberat inversi 3D digabungkan dengan hasil interpretasi dari tensor Marussi dalam bentuk peta (Gambar 4.26).



Gambar 4.26 Struktur geologi hasil interpretasi pada daerah penelitian.

“Halaman ini sengaja dikosongkan”

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil dari interpretasi struktur geologi pada model inversi 3D dan peta tensor Marusi dapat disimpulkan bahwa:

1. Hasil pemodelan inversi 3D sulit untuk menentukan kemenerusan patahan Watukosek namun terdapat kemungkinan patahan lain yang dapat diinterpretasi. Struktur perlipatan yang terbentuk akibat deformasi kompresional selama proses subduksi lempeng Australia terhadap lempeng Eurasia. Struktur graben, half-graben, patahan normal yang telah diinterpretasikan terbentuk sebagai akibat proses perlipatan. Ketika terdapat tekanan subduksi dari arah selatan ke utara, alam mengimbangi dengan deformasi ekstensional ke arah barat-timur. Selama proses ini sedimen diendapkan secara terus menerus sehingga terbentuk seperti sekarang ini.
2. Patahan-patahan yang telah diinterpretasikan dalam peta tensor Marusi merupakan patahan geser *sinistral* (mengiri) dengan jurus timur laut. Patahan geser ini secara alami terbentuk akibat proses subduksi di selatan Jawa. Patahan Watukosek pada daerah Gunung Penanggungan dapat diinterpretasi, namun diragukan kemenerusannya hingga daerah Gunung Anyar. Hal ini terlihat adanya perbedaan orientasi jurus patahan geser yang telah diinterpretasikan dengan hasil interpretasi patahan Watukosek oleh Mazinni *et al* (2009).

5.2 Saran

Penelitian lebih lanjut mengenai keadaan struktur geologi daerah Jawa Timur sangatlah penting, khususnya pada daerah cekungan kendeng, karena selain potensi bencana alam di Jawa Timur cukup besar, potensi sumber daya alamnya juga cukup besar.

“Halaman ini sengaja dikosongkan”

DAFTAR PUSTAKA

- Barthelmes, F., 2013. Definition of Functionals of the Geopotential and Their Calculation from Spherical Harmonic Models. *Scientific Technical Report STR09/02. GFZ German Research Centre for Geosciences. Potsdam, Germany, 32pp.*
- Blakely, R.J., 1995. Potential theory in gravity and magnetic applications, Cambridge University Press, New York.
- Beiki, M. and Pedersen, L.B., 2010. Eigenvector analysis of gravity gradient tensor to locate geologic bodies. *Geophysics, vol. 75, pp. 137-149. DOI: 10.1190/1.3484098.*
- Bucha, B., Jan'ak, J., 2013. A MATLAB-based graphical user interface program for computing functionals of the geopotential up to ultra-high degrees and orders. *Computers & Geosciences 56, 186–196.*
- Clements, B., Hall, R., Smyth, H. R. & Cottam, M. A., 2009. Thrusting of a volcanic arc: a new structural model for Java. *Pet. Geosci., 15(2), pp. 159-174.*
- Davies, R. et al., 2008. The East Java mud volcano (2006 to present): An earthquake or drilling trigger?. *Earth and Planetary Science Letters, Volume 272, pp. 627-638.*
- Davies, R., Manga, M., Tingay, M. & Swarbrick, R., 2011. Fluid transport properties and estimation of overpressure at the Lusi mud volcano, East Java Basin (Tanikawa et al., 2010). *Engineering Geology, Volume 121, pp. 97-99.*
- De Genevraye, P. & Samuel, L., 1972. *Geology of the Kendeng zone (central and east Java)*. s.l., Proceedings of the Indonesian Petroleum Association, 1st Annual Convention, pp. 17-30.
- Featherstone, W. E., 1995a. On the use of Australian geodetic datum in gravity field determination. *Geomatics Research Australia 62, 19-29.*
- Förste, Ch., Bruinsma, S.L., Abrikosov, O., Lemoine, J.- M., Marty, J.C., Flechtner, F., Balmino, G. 2014. EIGEN-6C4 The latest combined global gravity field model including GOCE data up to degree and order 2190 of GFZ Potsdam and GRGS Toulouse. *GFZ Data Services.*
- Gafoer, S. & Ratman, N., 1999. *Peta Geologi Lembar Jawa Bagian Timur, Bandung: Pusat Penelitian dan Pengembangan Geologi.*

- Hall, R., 2011. Australia-SE Asia collision: plate tectonics and crustal flow. *Geol. Soc. Lond.*, Volume 355, pp. 75-109.
- Hirt, C. and Rexer, M., 2015. Earth2014: 1 arc-min shape, topography, bedrock and ice-sheet models – available as gridded data and degree-10,800 spherical harmonics. *International Journal of Applied Earth Observation and Geoinformation* vol. 39, pp. 103-112. doi:10.1016/j.jag.2015.03.001.
- Istadi, B. P., Pramono, G. H., Sumintadireja, P. & Alam, S., 2009. Modeling study of growth and potential geohazard for LUSI mud volcano: East Java, Indonesia. *Marine and Petroleum Geology*, Volume 26, p. 1724–1739.
- Istadi, B.P., Wibowo, H.T., Sunardi, E., Hadi, S., Sawolo, N., 2012. Mud Volcano and Its Evolutions, *Earth Sciences*, ISBN:978-953-307-861-8
- Kane, M. F., 1962. A Comprehensive System of Terrain Using a Digital Computer. *Geophysics*, 27(4), pp. 455-462.
- Kostecký, J., Eppelbaum, L. V., 2014. Gravity Disturbances, Marussi Tensor, Invariants and Other Functions of the Geopotential Represented by EGM 2008. *Journal of Earth Science Research*, Vol. 2 Iss. 3, PP. 88-101.
- Lafehr, T.R., 1991 Standardization in gravity reduction. *Geophysics*, vol. 56 no.8, pp. 1170-1178.
- Lafehr, T.R., 1991b. An exact solution for the gravity curvature (Bullard B) correction. *Geophysics*, Vol. 56 no.8, pp. 1179-1184.
- Li, Y. & Oldenburg, D., 1998. 3-D inversion of gravity data. *Geophysics*, Volume 63, p. 109–119.
- Manga, M., 2007. Did an Earthquake Trigger the May 2006 Eruption of the Lusi Mud Volcano?. *EOS*, 88(18), p. 201.
- Mazzini, A. et al., 2007. Triggering and dynamic evolution of the LUSI mud volcano, Indonesia. *Earth and Planetary Science Letters*, Volume 261, p. 375–388.
- Mazzini, A. et al., 2009. Strike-slip faulting as a trigger mechanism for overpressure release through piercement structures. Implications for the Lusi mud volcano, Indonesia. *Marine and Petroleum Geology*, Volume 26, pp. 1751-1765.
- Mazinni, A., Etiope, G. & Svensen, H., 2012. A new hydrothermal scenario for the 2006 Lusi eruption, Indonesia. Insights from gas geochemistry. *Earth and Planetary Science Letters*, p. 317–318.

- Moritz, H., 1980. Geodetic Reference System 1980. *Geodesique*, 54(4), pp. 395-405.
- Nagy, D., 1966. The Gravitational Attraction of a Right Rectangular Prism. *Geophysics*, 31(2), pp. 362-371.
- NIMA, 2000. Departement of Defense World Geodetic System 1984. Technical report TR8350.2, National Imagery and Mapping Agency, Washington D.C., USA
- Padmawidjaja, T., 2013. Analisis data gayaberat daerah Porong dalam studi kasus struktur dan deformasi geologi bawah permukaan. *Jurnal Lingkungan dan Bencana Geologi*, Vol. 4 No. 3 Desember 2013: 237 – 251
- Pavlis, N., 2013. *NGA.mil*. [Online] Available at: http://earth.info.nga.mil/GandG/wgs84/gravitymod/egm2008/hsynth_WGS84.exe [Accessed wednesday october 2016].
- Pedersen. B.D. and Rasmussen, T.M., 1990. The gradient tensor of potential field anomalies: Some implications on data collection and data processing of maps. *Geophysics*, vol. 55, pp. 1558-1566.
- Petrovskaya, M.S., Vershkov, A.N., 2006. Non-singular expressions for the gravity gradients in the local north-oriented and orbital reference frames. *Journal of Geodesy*, 80, pp. 117–127.
- Panzer, F., D'Amico, S., Lupi, M. & Mauri, G., 2017. Lusi hydrothermal structure inferred through ambient vibration measurements. *Marine and Petroleum Geology*, Volume xxx, pp. 1-9.
- Saad, A., 2006. Understanding gravity gradients - a tutorial. *The Leading Edge*, 25(8), pp. 942-949.
- Santosa, S., Suwanti, T., 1992. Peta Geologi Lembar Malang, Jawa. Pusat Penelitian dan Pengembangan Geologi, Bandung.
- Sawolo, N., Sutriyono, E., Istadi, B. P. & Darmoyo, A. B., 2009. The LUSI mud volcano triggering controversy: Was it caused by drilling?. *Marine and Petroleum Geology*, 26, pp. 1766–1784.
- Schiller, D. M., Seubert, B. W., Musliki, S. & Abdullah, M., 1994. The reservoir potential of globigerinid sands in Indonesia. *Indon. Pet. Assoc. 23rd Ann. Conv.*
- Smyth, H. R., Hamilton, P. J., Hall, R. & Kinny, P. D., 2007. The deep crust beneath island arcs: inherited zircons reveal a Gondwana continental fragment beneath East Java, Indonesia. *Earth Planet. Sci. Lett.*, 258, pp. 269-282.

Spector, A. & Grant, F. S., 1970. STATISTICAL MODELS FOR INTERPRETING AEROMAGNETIC DATA. *Geophysics*, 35(2), pp. 293-302.

Supardjono, J.B., Hasan, K., Panggabean, H., Satria, D., Sukardi, 1992. Peta Geologi Lembar Surabaya dan Sapulu, Jawa. Pusat Penelitian dan Pengembangan Geologi, Bandung.

Tanikawa, W. et al., 2010. Fluid transport properties and estimation of overpressure at the Lusi mud volcano. East Java Basin. *Engineering Geology*, Volume 116, pp. 73-85.

Torge, W., 2001. *Geodesy*. 3rd ed. Berlin: Walter de Gruyter.

Wessel, P. et al., 2013. Generic Mapping Tools: Improved version released,. *EOS Trans. AGU*, Volume 94, pp. 409-410

Willumsen, P. & Schiller, D. M., 1994. High quality volcanoclastic sandstone reservoirs in East Java, Indonesia. *IPA*, Volume 1, pp. 101-111.

Yusuf, M., 2015. Pemodelan tiga dimensi anomaly gravitasi dan identifikasi sesar lokal dalam penentuan jenis sesar di daerah sidoarjo. Conference: STMKG Conference. <https://www.researchgate.net>

LAMPIRAN

```

function GrafLab(vstpar)
R1=0.8; G1=0.8; B1=0.8;
R2=0.95; G2=0.95; B2=0.95;
if nargin==0
    %Main window
    M=figure('units','pixels','numbertitle','off','name','GrafLab
1.2.2',...
        'color',[R1 G1 B1],'position',[300 100 600 600],...
        'tag','okno','menubar','none');
    a=0.01; b=0.04; c=0.045; d=-0.02;

    %Panels
    %=====
    %Geopotential model and reference system selection panel
    uipanel('Units','normalized','position',[0.06 0.77 0.88
0.21],...
        'backgroundcolor',[R1 G1 B1],'HighlightColor',[R2 G2
B2],'tag',...
        'GMaRSSpanel');
    %Point type selection panel
    uipanel('Units','normalized','position',[0.06 0.355 0.88
0.4],...
        'backgroundcolor',[R1 G1 B1],'HighlightColor',[R2 G2
B2],'tag',...
        'PTspanel');
    %Calculated parameters and output selection panel
    uipanel('Units','normalized','position',[0.06 0.08 0.88
0.26],...
        'backgroundcolor',[R1 G1 B1],'HighlightColor',[R2 G2
B2],'tag',...
        'CPaOSpanel');

    %Geopotential model and reference system selection
    %=====
    uicontrol('Units','normalized','position',[0.08 0.865+a 0.14
0.035],...
        'style','pushbutton','string','Browse...','tag','GGM',...
        'callback','GrafLab import_GGM'); %Browse... button
    uicontrol('Units','normalized','position',[0.6 0.865+a 0.32
0.035],...
        'style','checkbox','string','Use maximum degree of
GGM',...
        'value',1,'backgroundcolor',[R1 G1 B1],'tag','use'); %Use
maximum degree of GGM
    uicontrol('Units','normalized','position',[0.08 0.91+a 0.375
0.025],...
        'style','text','string','Global geopotential model of the
Earth',...
        'backgroundcolor',[R1 G1 B1]); %Text Global geopotential
model of the Earth
    uicontrol('Units','normalized','position',[0.075 0.82+a 0.22
0.025],...
        'style','text','string','GM of GGM (m3.s-2)',...
        'backgroundcolor',[R1 G1 B1]); %Text GM of GGM (m^3.s^-2)

```

```

    uicontrol('Units','normalized','position',[0.08 0.78+a 0.21
0.035],...
'style','edit','string','3986004.415E+8','backgroundcolor',...
[R2 G2 B2],'tag','GM'); %Value of GM
    uicontrol('Units','normalized','position',[0.325 0.82+a 0.2
0.025],...
'style','text','string','R of GGM (m)',...
'backgroundcolor',[R1 G1 B1],'tag','R_text'); %Text R of
GGM (m)
    uicontrol('Units','normalized','position',[0.32 0.78+a 0.21
0.035],...
'style','edit','string','6378136.3','backgroundcolor',[R2
G2 B2],...
'tag','R'); %Value of R
    uicontrol('Units','normalized','position',[0.565 0.82+a 0.06
0.025],...
'style','text','string','nmin','backgroundcolor',[R1 G1
B1],...
'tag','text_nmin'); %Text nmin
    uicontrol('Units','normalized','position',[0.56 0.78+a 0.08
0.035],...
'style','edit','string','0','backgroundcolor',[R2 G2
B2],...
'tag','nmin'); %Value of nmin
    uicontrol('Units','normalized','position',[0.675 0.82+a 0.06
0.025],...
'style','text','string','nmax','backgroundcolor',[R1 G1
B1],...
'tag','text_nmax'); %Text nmax
    uicontrol('Units','normalized','position',[0.67 0.78+a 0.08
0.035],...
'style','edit','backgroundcolor',[R2 G2 B2],'tag','nmax');
%Value of nmax
    uicontrol('Units','normalized','position',[0.795 0.82+a 0.09
0.025],...
'style','text','string','Ellipsoid','backgroundcolor',[R1
G1 B1],...
'tag','ell_text'); %Text Ellipsoid
    uicontrol('Units','normalized','position',[0.78 0.717+a 0.13
0.1],...
'style','popup','string','GRS80|WGS84','backgroundcolor',...
[R2 G2 B2],'tag','ell'); %Ellipsoid - pop-up menu
    uicontrol('units','normalized','position',[0.25 0.865+a 0.31
0.035],...
'style','edit','tag','nameGGM','enable','off'); %Name of
the imported GGM file
    uicontrol('Units','normalized','position',[0.06 0.03 0.28
0.025],...
'style','text','backgroundcolor',[R1 G1
B1],'tag','hlasky'); %Status line

    %Point type selection
    %=====
    %Text Input coordinates

```

```

    uicontrol('Units','normalized','position',[0.08 0.65+b 0.28
0.025],...
    'style','text','string','Type of the input
coordinates:','backgroundcolor',...
    [R1 G1 B1]);
    %Radio button group
    c0 = uibuttongroup('visible','on','units','normalized',...
    'Position',[0.375 0.618+b 0.4
0.06], 'bordertype','none',...
    'backgroundcolor',[R1 G1 B1], 'tag','coordinates');
    %Radio button: Ellipsoidal
    c1 = uicontrol('units','normalized','Style','Radio','pos',...
    [0.0 0.5 0.4
0.5], 'parent',c0, 'HandleVisibility','on',...
    'backgroundcolor',[R1 G1
B1], 'tag','rbutton1coord');
set(c1, 'String','Ellipsoidal', 'fontname','cambria', 'fontsize',10);
    %Radio button: Spherical
    c2 = uicontrol('units','normalized','Style','Radio','pos',...
    [0.388 0.5 0.4
0.5], 'parent',c0, 'HandleVisibility','on',...
    'backgroundcolor',[R1 G1
B1], 'tag','rbutton2coord');
set(c2, 'String','Spherical', 'fontname','cambria', 'fontsize',10);
    uicontrol('Units','normalized','position',[0.08 0.59+b 0.1
0.035],...
    'style','checkbox','string','Grid','backgroundcolor',...
    [R1 G1 B1], 'tag','gridcheck'); %Grid checkbox
    uicontrol('units','normalized','position',[0.2 0.59+b 0.13
0.035],...
    'style','checkbox','string','Load data','background',[R1
G1 B1],...
    'tag','loaddata'); %Load data checkbox
    uicontrol('Units','normalized','position',[0.34 0.59+b 0.14
0.035],...
    'style','pushbutton','string','Browse...','tag','import',...
    'callback','GrafLab input'); %Browse... button
    uicontrol('Units','normalized','position',[0.55+d 0.59+b 0.2
0.035],...
    'style','checkbox','string','Point-wise',...
    'backgroundcolor',[R1 G1 B1], 'tag','diskcheck'); %Point-
wise checkbox

    %Grid
    %-----
    %phi
    uicontrol('Units','normalized','position',[0.08 0.5+b 0.11
0.035],...
    'style','edit','backgroundcolor',[R2 G2
B2], 'tag','fimin'); %phi min
    uicontrol('Units','normalized','position',[0.225 0.5+b 0.11
0.035],...
    'style','edit','backgroundcolor',[R2 G2
B2], 'tag','fistep'); %phi step
    uicontrol('Units','normalized','position',[0.37 0.5+b 0.11
0.035],...

```

```

        'style','edit','backgroundcolor',[R2 G2
B2], 'tag','fimax'); %phi max

%lambda
uicontrol('Units','normalized','position',[0.08 0.418+b 0.11
0.035],...
        'style','edit','backgroundcolor',[R2 G2
B2], 'tag','lambdamin'); %lambda min
uicontrol('Units','normalized','position',[0.225 0.418+b 0.11
0.035],...
        'style','edit','backgroundcolor',[R2 G2
B2], 'tag','lambdastep'); %lambda step
uicontrol('Units','normalized','position',[0.37 0.418+b 0.11
0.035],...
        'style','edit','backgroundcolor',[R2 G2
B2], 'tag','lambdamax'); %lambda max

%h
uicontrol('Units','normalized','position',[0.08 0.335+b 0.4
0.035],...
        'style','edit','backgroundcolor',[R2 G2
B2], 'tag','hgrid'); %h

%Text for latitudes
uicontrol('Units','normalized','position',[0.075 0.54+b 0.12
0.025],...
        'style','text','string','Lat. min
(°)', 'backgroundcolor',...
        [R1 G1 B1], 'tag','fimin_string'); %Text Lat. min (°)
uicontrol('Units','normalized','position',[0.22 0.54+b 0.12
0.025],...
        'style','text','string','Lat. step
(°)', 'backgroundcolor',...
        [R1 G1 B1], 'tag','fistep_string'); %Text Lat. step (°)
uicontrol('Units','normalized','position',[0.365 0.54+b 0.12
0.025],...
        'style','text','string','Lat. max
(°)', 'backgroundcolor',...
        [R1 G1 B1], 'tag','fimax_string'); %Text Lat. max (°)

%Text for longitudes
uicontrol('Units','normalized','position',[0.075 0.458+b 0.12
0.025],...
        'style','text','string','Lon. min
(°)', 'backgroundcolor',...
        [R1 G1 B1]); %Text Lon. min (°)
uicontrol('Units','normalized','position',[0.22 0.458+b 0.12
0.025],...
        'style','text','string','Lon. step
(°)', 'backgroundcolor',...
        [R1 G1 B1]); %Text Lon. step (°)
uicontrol('Units','normalized','position',[0.365 0.458+b 0.12
0.025],...
        'style','text','string','Lon. max
(°)', 'backgroundcolor',...
        [R1 G1 B1]); %Text Lon. max (°)

```



```

    %Text Height above the reference surface (m)
    uicontrol('Units','normalized','position',[0.08 0.375+b 0.4
0.025],...
    'style','text','string','Height above the reference
surface (m)',...
    'backgroundcolor',[R1 G1 B1],'tag','h_string');

    %Point-wise
    %-----
    uicontrol('Units','normalized','position',[0.55+d 0.5+b 0.38
0.035],...
    'style','edit','backgroundcolor',[R2 G2 B2],'tag','fi');
%phi
    uicontrol('Units','normalized','position',[0.55+d 0.418+b 0.38
0.035],...
    'style','edit','backgroundcolor',[R2 G2
B2],'tag','lambda'); %lambda
    uicontrol('Units','normalized','position',[0.55+d 0.335+b 0.38
0.035],...
    'style','edit','backgroundcolor',[R2 G2
B2],'tag','hdisk'); %h

    uicontrol('Units','normalized','position',[0.55+d 0.54+b 0.38
0.025],...
    'style','text','string','Latitude
(°)','backgroundcolor',[R1 G1 B1]); %Text Latitude (°)
    uicontrol('Units','normalized','position',[0.55+d 0.458+b 0.38
0.025],...
    'style','text','string','Longitude
(°)','backgroundcolor',[R1 G1 B1]); %Text Longitude (°)
    uicontrol('Units','normalized','position',[0.55+d 0.375+b 0.38
0.025],...
    'style','text','string','Ellipsoidal height/Spherical
radius (m)',...
    'backgroundcolor',[R1 G1 B1]); %Ellipsoidal
height/Spherical radius (m)

    %Calculated parameters and output selection
    %=====
    %The first functional
    uicontrol('Units','normalized','position',[0.08 0.16+c 0.3
0.1],...
    'style','popup','string',...

'|Deflection_of_the_vertical_eta|Deflection_of_the_vertical_xi|Def
lection_of_the_vertical_Theta|Disturbing_potential|Disturbing_tens
or_Trr_Tpp_Tll|Disturbing_tensor_Trp_Trl_Tpl|Disturbing_tensor_Txx
_Tyy_Tzz|Disturbing_tensor_Txy_Txz_Tyz|Geoid_undulation|Gravitatio
nal_potential|Gravitational_tensor_Vrr_Vpp_Vll|Gravitational_tenso
r_Vrp_Vrl_Vpl|Gravitational_tensor_Vxx_Vyy_Vzz|Gravitational_tenso
r_Vxy_Vxz_Vyz|Gravity|Gravity_sa|Gravity_potential|Gravity_anomaly
_sa|Gravity_disturbance|Gravity_disturbance_sa|Height_anomaly_ell|
Height_anomaly|Second_radial_derivative_of_disturbing_potential|Se
cond_radial_derivative_of_gravity_potential',...
    'backgroundcolor',[R2 G2 B2],'tag','P1');
    %The second functional

```

```

    uicontrol('Units','normalized','position',[0.08 0.105+c 0.3
0.1],...
        'style','popup','string',...

'|Deflection_of_the_vertical_eta|Deflection_of_the_vertical_xi|Def
lection_of_the_vertical_Theta|Disturbing_potential|Disturbing_tens
or_Trr_Tpp_Tll|Disturbing_tensor_Trp_Trl_Tpl|Disturbing_tensor_Txx
_Tyy_Tzz|Disturbing_tensor_Txy_Txz_Tyz|Geoid_undulation|Gravitatio
nal_potential|Gravitational_tensor_Vrr_Vpp_Vll|Gravitational_tenso
r_Vrp_Vrl_Vpl|Gravitational_tensor_Vxx_Vyy_Vzz|Gravitational_tenso
r_Vxy_Vxz_Vyz|Gravity|Gravity_sa|Gravity_potential|Gravity_anomaly
_sa|Gravity_disturbance|Gravity_disturbance_sa|Height_anomaly_ell|
Height_anomaly|Second_radial_derivative_of_disturbing_potential|Se
cond_radial_derivative_of_gravity_potential',...
    'backgroundcolor',[R2 G2 B2],'tag','P2');
%The third functional
    uicontrol('Units','normalized','position',[0.08 0.05+c 0.3
0.1],...
        'style','popup','string',...

'|Deflection_of_the_vertical_eta|Deflection_of_the_vertical_xi|Def
lection_of_the_vertical_Theta|Disturbing_potential|Disturbing_tens
or_Trr_Tpp_Tll|Disturbing_tensor_Trp_Trl_Tpl|Disturbing_tensor_Txx
_Tyy_Tzz|Disturbing_tensor_Txy_Txz_Tyz|Geoid_undulation|Gravitatio
nal_potential|Gravitational_tensor_Vrr_Vpp_Vll|Gravitational_tenso
r_Vrp_Vrl_Vpl|Gravitational_tensor_Vxx_Vyy_Vzz|Gravitational_tenso
r_Vxy_Vxz_Vyz|Gravity|Gravity_sa|Gravity_potential|Gravity_anomaly
_sa|Gravity_disturbance|Gravity_disturbance_sa|Height_anomaly_ell|
Height_anomaly|Second_radial_derivative_of_disturbing_potential|Se
cond_radial_derivative_of_gravity_potential',...
    'backgroundcolor',[R2 G2 B2],'tag','P3');
%The fourth functional
    uicontrol('Units','normalized','position',[0.08 -0.005+c 0.3
0.1],...
        'style','popup','string',...

'|Deflection_of_the_vertical_eta|Deflection_of_the_vertical_xi|Def
lection_of_the_vertical_Theta|Disturbing_potential|Disturbing_tens
or_Trr_Tpp_Tll|Disturbing_tensor_Trp_Trl_Tpl|Disturbing_tensor_Txx
_Tyy_Tzz|Disturbing_tensor_Txy_Txz_Tyz|Geoid_undulation|Gravitatio
nal_potential|Gravitational_tensor_Vrr_Vpp_Vll|Gravitational_tenso
r_Vrp_Vrl_Vpl|Gravitational_tensor_Vxx_Vyy_Vzz|Gravitational_tenso
r_Vxy_Vxz_Vyz|Gravity|Gravity_sa|Gravity_potential|Gravity_anomaly
_sa|Gravity_disturbance|Gravity_disturbance_sa|Height_anomaly_ell|
Height_anomaly|Second_radial_derivative_of_disturbing_potential|Se
cond_radial_derivative_of_gravity_potential',...
    'backgroundcolor',[R2 G2 B2],'tag','P4');

%-----
%Commission error checkbox
    uicontrol('Units','normalized','position',[0.42 0.228+c 0.2
0.025],...
        'style','checkbox','string','Commission
error','backgroundcolor',[R1 G1 B1],...
        'tag','STD'); %STD
%Computation of fnALFs
    uicontrol('Units','normalized','position',[0.42 0.167+c 0.25
0.038],...
        'style','pushbutton','string','Computation of fnALFs',...

```

```

        'callback','GrafLab fnALFs','tag','fnALFs');

    %Display data settings
    uicontrol('Units','normalized','position',[0.42 0.112+c 0.25
0.038],...
        'style','pushbutton','string','Display data settings',...
        'callback','GrafLab Display_data_settings','tag','DDS');
    %Output folder and file
    uicontrol('Units','normalized','position',[0.42 0.057+c 0.25
0.038],...
        'style','pushbutton','string','Output folder and file',...
        'callback','GrafLab Output_folder','tag','outfolder');
    %Export data checkbox
    uicontrol('Units','normalized','position',[0.72+d 0.228+c 0.2
0.025],...
        'style','checkbox','string','Export
data','tag','export',...
        'callback','GrafLab output','backgroundcolor',[R1 G1
B1],'value',1);
    %Export report checkbox
    uicontrol('Units','normalized','position',[0.72+d 0.175+c 0.2
0.025],...
        'style','checkbox','string','Export
report','backgroundcolor',...
        [R1 G1 B1],'tag','report','value',1);
    %Export data in *.mat
    uicontrol('Units','normalized','position',[0.72+d 0.12+c 0.23
0.025],...
        'style','checkbox','string','Export data in
*.mat','backgroundcolor',...
        [R1 G1 B1],'tag','datamat','value',0);
    %OK button
    uicontrol('Units','normalized','position',[0.35 0.015 0.13
0.05],...
        'style','pushbutton','string','OK','Callback','GrafLab
OK');
    %Close button
    uicontrol('Units','normalized','position',[0.55 0.015 0.13
0.05],...
        'style','pushbutton','string','Close','Callback','GrafLab
Close');
    %Set font to Cambria
    set(get(M,'children'),'fontname','cambria','fontsize',10);
    set(findobj('tag','PTspanel'),'title','Point type
selection',...
        'fontname','cambria','fontsize',8);
    set(findobj('tag','GMarSSpanel'),'title',...
        'Geopotential model and reference system
selection','fontname',...
        'cambria','fontsize',8);
    set(findobj('tag','CPaOSpanel'),'title',...
        'Calculated parameters and output
selection','fontname',...
        'cambria','fontsize',8);
else
    switch(vstpar)

```

```

        case('import_GGM') %Click on the Browse... button in the
Geopotential model and reference system selection panel

        [GGMname,GGMadresar]=uigetfile('*.','Select GGM
File');
        if GGMname==0
        else
            set(findobj('tag','R'),'userdata',GGMname);
set(findobj('tag','ell'),'userdata',GGMadresar);
set(findobj('tag','nameGGM'),'string',GGMname); %Display the name
of the imported GGM file
        end
        case('input') %Click on the Browse... button in the Point
type selection panel
            [loadname,loadadresar]=uigetfile('*.','Select File
Containing Computational Points');
            if loadname==0
            else
                set(findobj('tag','use'),'userdata',loadname);
set(findobj('tag','diskcheck'),'userdata',loadadresar);
            end
        case('Output_folder') %Click on the Output folder and file
button
            [outname,outadresar]=uiputfile('*.');
            if outname==0
            else
                if find(outname=='.')>0
                    outname=outname(1:(find(outname=='.')-1));
                end
                set(findobj('tag','R_text'),'userdata',outname);
set(findobj('tag','ell_text'),'userdata',outadresar);
            end
        case('Display_data_settings') %Click on the Display data
settings
            %Main window
D=figure('units','pixels','numbertitle','off','name',...
'Display data settings','color',[R1 G1
B1],'position',...
[320 150 550
350],'tag','oknoDDS','menubar','none');
            %Panel
            uipanel('Units','normalized','position',[0 0 0.26
1],...
                'backgroundcolor',[R1 G1 B1],'HighlightColor',[R2
G2 B2],...
                'fontsize',8,'fontname','cambria');
            %Display data checkbox
            uicontrol('Units','normalized','position',[0.05 0.87
0.2 0.05],...
                'style','checkbox','string','Display
data','backgroundcolor',...
                [R1 G1 B1],'tag','Display');
            display_data=get(findobj('tag','DDS'),'userdata');
            if display_data==0
            elseif display_data==1
                set(findobj('tag','Display'),'value',1);
            end
end

```

```

    %Text next to the Display data checkbox
    g0=uicontrol('Units','normalized','position',[0.3 0.83
0.65 0.13],...
    'style','text','backgroundcolor',[R1 G1
B1],'string',...
    'To export a graphic file, select this checkbox.
The data will be depicted on a map using automatically selected
cartographic projection. ');
    set(g0,'HorizontalAlignment','left');
    %Text Graphic format
    uicontrol('Units','normalized','position',[0.05 0.75
0.16 0.05],...
    'style','text','string','Graphic
format','backgroundcolor',...
    [R1 G1 B1],'tag','format');
    %Graphic format pop-up menu
    uicontrol('Units','normalized','position',...
    [0.05 0.64 0.16 0.1],'style','popup','string',...
    '*.bmp|*.emf|*.eps|*.jpeg|*.pdf|*.png|*.tiff',...
    'backgroundcolor',[R2 G2 B2],'tag','pripona');
    pripona=get(findobj('tag','nmin'),'userdata');
    if isempty(pripona)
        set(findobj('tag','pripona'),'value',6);
    else
        set(findobj('tag','pripona'),'value',pripona);
    end
    %Text next to the Graphic format file
    g1=uicontrol('Units','normalized','position',[0.3 0.67
0.65 0.13],...
    'style','text','backgroundcolor',[R1 G1
B1],'string',...
    'Select one of the graphic format files. For a
vector output it is recommended to use *.eps graphic file and
*.png format for a bitmap output. ');
    set(g1,'HorizontalAlignment','left');

    %Text Colormap
    uicontrol('Units','normalized','position',[0.05 0.57
0.16 0.05],...
    'style','text','string','Colormap',...
    'backgroundcolor',[R1 G1 B1]);
    %Colormap pop-up menu
    uicontrol('Units','normalized','position',[0.05 0.46
0.16 0.1],...
    'style','popup','string',...
    'jet|HSV|hot|cool|spring|summer|autumn|winter|gray|bone|copper|pin
k|lines',...
    'backgroundcolor',[R2 G2 B2],'tag','colormap');
    set(findobj('tag','colormap'),'value',2);
    color=get(findobj('tag','nmax'),'userdata');
    if isempty(color)
        set(findobj('tag','colormap'),'value',1);
    else
        set(findobj('tag','colormap'),'value',color);
    end
    %Text next to the colormap

```

```

        g1=uicontrol('Units','normalized','position',[0.3
0.417 0.65 0.2],...
        'style','text','backgroundcolor',[R1 G1
B1],'string',...
        'Select a colormap of the output file. Mostly it
is recommended to use the jet colormap, which ranges from blue to
red, and passes through the colors cyan, yellow, and orange.');
```

```

        set(g1,'HorizontalAlignment','left');
        %Text Number of colors
        uicontrol('Units','normalized','position',[0.04 0.39
0.19 0.05],...
        'style','text','string','Number of colors',...
        'backgroundcolor',[R1 G1 B1]);
        %Value of number of colors
        uicontrol('Units','normalized','position',[0.08 0.32
0.1 0.06],...
        'style','edit','string',15,'backgroundcolor',...
        [R2 G2 B2],'tag','skala');
        %Text next to the number of colors
        g1=uicontrol('Units','normalized','position',[0.3 0.31
0.65 0.13],...
        'style','text','backgroundcolor',[R1 G1 B1]);
        set(g1,'HorizontalAlignment','left','string',...
        'Enter a number of colors of the selected
colormap. Note that processing time may increase to a several
minutes, if a large number of colors has been entered for a large
data set.');
```

```

        ncolor=get(findobj('tag','text_nmin'),'userdata');
        if isempty(ncolor)
        else
            set(findobj('tag','skala'),'string',ncolor);
        end
        %Text DPI
        uicontrol('Units','normalized','position',[0.08 0.24
0.1 0.05],...
        'style','text','string','DPI','backgroundcolor',[R1 G1 B1]);
        %Value of DPI
        uicontrol('Units','normalized','position',[0.08 0.17
0.1 0.06],...
        'style','edit','string',300,'backgroundcolor',[R2
G2 B2],...
        'tag','DPI');
```

```

        DPI=get(findobj('tag','text_nmax'),'userdata');
        if isempty(DPI)
        else
            set(findobj('tag','DPI'),'string',DPI);
        end
        %Text next to the DPI
        g1=uicontrol('Units','normalized','position',[0.3 0.03
0.65 0.2],...
        'style','text','backgroundcolor',[R1 G1
B1],'string',...
        'Enter a value of dots per inch of the output
file.');
```

```

        set(g1,'HorizontalAlignment','left');
        %OK button

```

```

        uicontrol('Units','normalized','position',[0.35 0.02
0.13 0.08],...
'style','pushbutton','string','OK','Callback','GrafLab OKDDS');
%Close button
        uicontrol('Units','normalized','position',[0.55 0.02
0.13 0.08],...

'style','pushbutton','string','Close','Callback',...
'GrafLab CloseDDS');
%setting font to Cambria
set(get(D,'children'),'fontname','cambria','fontsize',10)
    case('fnALFs') %Click on the Computation of fnALFs
        %Main window
F=figure('units','pixels','numbertitle','off','name',...
'Approaches to compute fully normalized associated
Legendre functions','color',...
[R1 G1 B1],'position',[320 150 550 350],'tag',...
'oknofnALFs','menubar','none');
%Radio button group
u0 =
uibuttongroup('visible','on','units','normalized',...
'Position',[0 0 .26 1],'tag','volbaALFs');
%The first radiobutton
u1 =
uicontrol('units','normalized','Style','Radio','pos',...
[0.1 0.7 0.9
0.3],'parent',u0,'HandleVisibility','on',...
'tag','rbutton1');
set(u1,'String','<html>Standard forward<br>column
method',...
'fontname','cambria','fontsize',10);
note1=uicontrol('Units','normalized','position',...
[0.3 0.74 0.65
0.2],'style','text','backgroundcolor',...
[R1 G1 B1]);
set(note1,'HorizontalAlignment','left','string',...
'It is recommended to use the standard forward
column method for all latitudes up to the maximum degree 1800.
However, this method may also be used for the latitudes <0°,56°>
and <80°,90°> up to the maximum degree 2190.');
```

```

        %The second radiobutton
u2 =
uicontrol('units','normalized','Style','Radio','pos',...
[0.1 0.42 0.9
0.3],'parent',u0,'HandleVisibility','on',...
'tag','rbutton2');
set(u2,'String',...
'<html>Modified forward<br>column
method<br>combined with<br>Horner's scheme',...
'fontname','cambria','fontsize',10);
note2=uicontrol('Units','normalized','position',[0.3
0.38 0.65 0.3],...
'style','text','backgroundcolor',[R1 G1 B1]);
set(note2,'HorizontalAlignment','left','string',...
'It is recommended to use the modified forward
column method combined with Horner's scheme for all latitudes and
maximum degrees ranging from 1801 to 2700. This method may also be
```

```

used for lower degrees than 1801, but cannot be applied to degrees
higher than 2700 due to the overflow problem.');
```

```

    %The third radiobutton
    u3 =
uicontrol('units','normalized','Style','Radio','pos',...
          [0.1 0.12 0.9
0.3], 'parent',u0,'HandleVisibility','on',...
          'tag','rbutton3');
    set(u3,'String','<html>Extended-
range<br>arithmetic',...
        'fontname','cambria','fontsize',10);
    note3=uicontrol('Units','normalized','position',...
                   [0.3 0.018 0.65
0.3], 'style','text','backgroundcolor',...
                   [R1 G1 B1]);
    set(note3,'HorizontalAlignment','left','string',...
        'The extended-range arithmetic approach may be
used for all latitudes up to an arbitrary degree essentially.');
```

```

    %OK button
    uicontrol('Units','normalized','position',[0.35 0.02
0.13 0.08],...
            'style','pushbutton','string','OK','Callback','GrafLab OKfnALFs');
    %Close button
    uicontrol('Units','normalized','position',[0.55 0.02
0.13 0.08],...
            'style','pushbutton','string','Close','Callback','GrafLab
ClosefnALFs');
```

```

    %The chosen approach for computation of fnALFs
    volbaALFs=get(findobj('tag','fnALFs'),'userdata');
    if isempty(volbaALFs)
        volbaALFs=1;
    end

    %Mark the chosen radiobutton
    if volbaALFs==1
        set(u1,'value',1);
    elseif volbaALFs==2
        set(u2,'value',1);
    elseif volbaALFs==3
        set(u3,'value',1);
    end

set(get(F,'children'),'fontname','cambria','fontsize',10);
    case('ClosefnALFs') %Click on the Close button in the
Computation of fnALFs window
    close
    case('OKfnALFs') %Click on the OK button in the
Computation of fnALFs window
volbaALFs=get(findobj('tag','volbaALFs'),'selectedobject');
    if get(findobj('tag','rbutton1'),'value')==1
        volbaALFs=1;
    elseif get(findobj('tag','rbutton2'),'value')==1
        volbaALFs=2;
    elseif get(findobj('tag','rbutton3'),'value')==1
        volbaALFs=3;
    end
    set(findobj('tag','fnALFs'),'userdata',volbaALFs);

```



```

close
case('OKDDS') %Click on the OK button in the Display data
settings window
display_data=get(findobj('tag','Display'),'value');
set(findobj('tag','DDS'),'userdata',display_data);
prip=get(findobj('tag','pripona'),'value');
set(findobj('tag','nmin'),'userdata',prip);
color=get(findobj('tag','colormap'),'value');
set(findobj('tag','nmax'),'userdata',color);
ncolor=get(findobj('tag','skala'),'string');
ncolor=str2double(ncolor);
set(findobj('tag','text_nmin'),'userdata',ncolor);

DPI=get(findobj('tag','DPI'),'string');
DPI=str2double(DPI);
set(findobj('tag','text_nmax'),'userdata',DPI);
%Check of the entered value of DPI and number of
colors
if isnan(DPI)==1 || DPI<0
    errordlg('Value of DPI must be larger than
zero.',...
            'Display data settings');
    error('Value of DPI must be larger than zero.');
```

```

end
if isnan(ncolor)==1
    errordlg('Entered value of number of colors is not
correct.',...
            'Display data settings');
    error('Entered value of number of colors is not
correct.');
```

```

end
if rem(DPI,1)~=0
    errordlg('Value of DPI must be a positive
integer.',...
            'Display data settings');
    error('Value of DPI must be a positive integer.');
```

```

end
if ncolor<2
    errordlg('Value of number of colors must be larger
than 1.',...
            'Display data settings');
    error('Value of number of colors must be larger
than 1.');
```

```

end
if rem(ncolor,1)~=0
    errordlg('Value of number of colors must be a
positive integer.',...
            'Display data settings');
    error('Value of number of colors must be a
positive integer.');
```

```

end
close

case('CloseDDS') %Click on the Close button in the Display
data settings window
close

```

```

        case('OK') %Click on the OK button in the main GrafLab
window
        set(findobj('tag','hlasky'),'string',...
            ','fontsize',8,'foregroundcolor','k');
drawnow;
        %Identification of point type selection
volbagridcheck=get(findobj('tag','gridcheck'),'value'); %Grid
        volbaloadcheck=get(findobj('tag','loaddata'),'value');
%Load
volbadiskcheck=get(findobj('tag','diskcheck'),'value'); %Diskretne
body
        %Error messages for point type selection
        if volbagridcheck==0 && volbadiskcheck==0 &&
volbaloadcheck==0
            errordlg('Please select a point type.','Point type
selection error');
            error('Please select a point type. ');
        elseif volbagridcheck==1 && volbadiskcheck==1 &&
volbaloadcheck==1
            errordlg('Please select one point type only.',...
                'Error in point type selection');
            error('Please select one point type only. ');
        elseif volbagridcheck==1 && volbadiskcheck==1
            errordlg('Please select one point type only.',...
                'Error in point type selection');
            error('Please select one point type only. ');
        elseif volbadiskcheck==1 && volbaloadcheck==1
            errordlg('Please select one point type only.',...
                'Error in point type selection');
            error('Please select one point type only. ');
        elseif volbagridcheck==1 && volbaloadcheck==1
            errordlg('Please select one point type only.',...
                'Error in point type selection');
            error('Please select one point type only. ');
        end
        %Ellipsoid
        if get(findobj('tag','ell'),'value')==1; %GRS80
            GME1=3986005*10^8; %Geocentric gravitational
constant of GRS80
            aE1=6378137; %Semimajor axis of GRS80
            eE1=sqrt(0.006694380022903416); %First
eccentricity of GRS80
            omegaE1=7292115*10^-11; %Angular velocity of GRS80
            CE1_20=-108263*10^-8/sqrt(5); %Fully normalized
C_20 of GRS80
        elseif get(findobj('tag','ell'),'value')==2; %WGS84
            GME1=3986004.418*10^8; %Geocentric gravitational
constant of WGS84
            aE1=6378137; %Semimajor axis of WGS84
            fE1=1/298.257223563; %Flattening of WGS84
            omegaE1=7292115*10^-11; %Angular velocity of WGS84
            CE1_20=-0.484166774985*10^-3; %Fully normalized
C_20 of WGS84
            eE1=sqrt(fE1*(2-fE1)); %First eccentricity of
WGS84
        end
        %GM and R of GGM
        GM=str2double(get(findobj('tag','GM'),'string'));

```

```

R=str2double(get(findobj('tag','R'),'string'));
if GM<=0
    errordlg('Value of GM must be larger than
zero.',...
            'Error in geopotential model and reference
system selection')
    error('Value of GM must be larger than zero.')
end
if isnan(GM)
    errordlg('Please input the value of GM.',...
            'Error in geopotential model and reference
system selection')
    error('Please input the value of GM.')
end
if R<=0
    errordlg('R must be larger than zero.',...
            'Error in geopotential model and reference
system selection')
    error('R value must be larger than zero.')
end
if isnan(R)
    errordlg('Please input the value of R.',...
            'Error in geopotential model and reference
system selection')
    error('Please input the value of R.')
end
%Selection of functionals of the geopotential
volbapar1=get(findobj('tag','P1'),'value'); %ID of the
first functional
volbapar2=get(findobj('tag','P2'),'value'); %ID of the
second functional
volbapar3=get(findobj('tag','P3'),'value'); %ID of the
third functional
volbapar4=get(findobj('tag','P4'),'value'); %ID of the
fourth functional
volbapar=[volbapar1;volbapar2;volbapar3;volbapar4];
pocetpar=length(find(volbapar>1));
%Error messages for selection of functionals of the
geopotential
if pocetpar==1
    if volbapar1>1 && volbapar2==1 && volbapar3==1 &&
volbapar4==1
        else
            errordlg('Please select the functional of the
geopotential in the first pop-up menu.',...
                    'Calculated parameters and output
selection');
            error('Please select the functional of the
geopotential in the first pop-up menu.');
```

```

                                'Calculated parameters and output
selection');
                                error('An identical functional of the
geopotential cannot be computed simultaneously more than once.');
```

```

                                end
                                else
                                errordlg('Please select the functionals of the
geopotential in the first and the second pop-up menu.',...
                                'Calculated parameters and output
selection');
```

```

                                error('Please select the functionals of the
geopotential in the first and the second pop-up menu.');
```

```

                                end
                                if length(nonzeros(volbapar==10 |
volbapar==23))==1 %If geoid or height anomaly is to be computed
                                errordlg('Geoid undulation and/or
Height_anomaly cannot be computed simultaneously with other
functionals. However, Geoid_undulation and Height_anomaly alone
can be computed simultaneously.',...
                                'Calculated parameters and output
selection');
```

```

                                error('Geoid_undulation and/or Height_anomaly
cannot be computed simultaneously with other functionals. However,
Geoid_undulation and Height_anomaly alone can be computed
simultaneously.');
```

```

                                end
                                elseif pocetpar==3
                                if volbapar1>1 && volbapar2>1 && volbapar3>1 &&
volbapar4==1
                                if (volbapar1 == volbapar2) || (volbapar1 ==
volbapar3) || (volbapar2 == volbapar3)
                                errordlg('An identical functional of the
geopotential cannot be computed simultaneously more than
once.',...
                                'Calculated parameters and output
selection');
```

```

                                error('An identical functional of the
geopotential cannot be computed simultaneously more than once.');
```

```

                                end
                                else
                                errordlg('Please select the functionals of the
geopotential in the first, second and third pop-up menu.',...
                                'Calculated parameters and output
selection');
```

```

                                error('Please select the functionals of the
geopotential in the first, second and third pop-up menu.');
```

```

                                end

                                if length(nonzeros(volbapar==10 |
volbapar==23))==1 || length(nonzeros(volbapar==10 |
volbapar==23))==2 %If geoid or height anomaly is to be computed
                                errordlg('Geoid undulation and/or
Height_anomaly cannot be computed simultaneously with other
functionals. However, Geoid_undulation and Height_anomaly alone
can be computed simultaneously.',...
                                'Calculated parameters and output
selection');
```

```

        error('Geoid_undulation and/or Height_anomaly
cannot be computed simultaneously with other functionals. However,
Geoid_undulation and Height_anomaly alone can be computed
simultaneously.');
```

end

```

    elseif pocetpar==4
        if (volbapar1 == volbapar2) || (volbapar1 ==
volbapar3) || (volbapar1 == volbapar4) || (volbapar2 == volbapar3)
|| (volbapar2 == volbapar4) || (volbapar3 == volbapar4)
            errordlg('An identical functional of the
geopotential cannot be computed simultaneously more than
once.',...
                    'Calculated parameters and output
selection');
```

error('An identical functional of the
geopotential cannot be computed simultaneously more than once.');

end

```

        if length(nonzeros(volbapar==10 |
volbapar==23))==1 || length(nonzeros(volbapar==10 |
volbapar==23))==2 || length(nonzeros(volbapar==10 |
volbapar==23))==3 %If geoid or height anomaly is to be computed
            errordlg('Geoid_undulation and/or
Height_anomaly cannot be computed simultaneously with other
functionals. However, Geoid_undulation and Height_anomaly alone
can be computed simultaneously.',...
                    'Calculated parameters and output
selection');
```

error('Geoid_undulation and/or Height_anomaly
cannot be computed simultaneously with other functionals. However,
Geoid_undulation and Height_anomaly alone can be computed
simultaneously.');

end

end

```

%Identification of type of the input coordinates
coord=get(findobj('tag','rbutton2coord'),'value');
```

if volbagridcheck==1

if coord==1 %Entered spherical coordinates

if any(volbapar==10) || any(volbapar==23)

errordlg('Geoid_undulation and
Height_anomaly cannot be computed in the spherical coordinates.
Reference surface to which geoid undulation and height anomaly are
expressed is conventionally the surface of the reference
ellipsoid. Grid that has been entered in the spherical coordinates
(constant value of the radius r) does not refer to this surface.
If you wish to compute these functionals, please use ellipsoidal
type of the input coordinates and set value in the array Height
above the reference surface (m) to zero.','Calculated parameters
and output selection')

error('Geoid_undulation and Height_anomaly
cannot be computed in the spherical coordinates. Reference surface
to which geoid undulation and height anomaly are expressed is
conventionally the surface of the reference ellipsoid. Grid that
has been entered in the spherical coordinates (constant value of
the radius r) does not refer to this surface. If you wish to
compute these functionals, please use ellipsoidal type of the

```

input coordinates and set value in the array Height above the
reference surface (m) to zero.')
    end
    end
elseif volbadiskcheck==1
    if coord==1 %Entered spherical coordinates
        if any(volbapar==10) || any(volbapar==23)
            errordlg('Geoid undulation and
Height_anomaly cannot be computed in the spherical coordinates.
Reference surface to which geoid undulation and height anomaly are
expressed is conventionally the surface of the reference
ellipsoid. If you wish to compute these functionals, please use
ellipsoidal type of the input coordinates and set value(s) in the
array Ellipsoidal height/Spherical radius (m) to
zero.','Calculated parameters and output selection')
            error('Geoid undulation and Height_anomaly
cannot be computed in the spherical coordinates. Reference surface
to which geoid undulation and height anomaly are expressed is
conventionally the surface of the reference ellipsoid. If you wish
to compute these functionals, please use ellipsoidal type of the
input coordinates and set value(s) in the array Ellipsoidal
height/Spherical radius (m) to zero.')
        end
    end
    end
    if volbapar1==1 && volbapar2==1 && volbapar3==1 &&
volbapar4==1
        errordlg('Please choose at least one functional of
the geopotential.','...
'Error in calculated paramters and output
selection')
        error('Please choose at least one functional of
the geopotential.')
    end
    %Identification of computation of fnALFs approach
    volbaALFs=get(findobj('tag','fnALFs'),'userdata');
    if isempty(volbaALFs)
        volbaALFs=1;
    end
    %Identification of Display data
    display_data=get(findobj('tag','DDS'),'userdata');
    if isempty(display_data)
        display_data=0;
    end
    %Identification of computation of commission error
    STD=get(findobj('tag','STD'),'value');
    %Identification of the output folder and file
    outname=get(findobj('tag','R_text'),'userdata');
    outadresar=get(findobj('tag','ell_text'),'userdata');
    if isempty(outname)
        set(findobj('tag','hlasky'),'string','Select
output folder and output file',...
'fontsize',8,'foregroundcolor','k');
drawnow;
        warn2=warndlg('Output folder and output file were
not specified. Click OK and then select an output folder and
output file. After the selection, the computation will start.');
```

```

        [outname,outadresar]=uinputfile('*.');
        if outname==0
            errordlg('Output folder and output file must
be specified!');

            set(findobj('tag','hlasky'),'string','',...
                'fontsize',8,'foregroundcolor','k');
drawnow;
            error('Output folder and output file must be
specified!');
        else
            if find(outname=='.')>0
                outname=outname(1:(find(outname=='.')-1));
            end
            end
            set(findobj('tag','hlasky'),'string','',...
                'fontsize',8,'foregroundcolor','k');
drawnow;
        end
        %Check of the entered values of DPI a number of colors
        if display_data==1
            ncolor=get(findobj('tag','text_nmin'),'userdata');
            DPI=get(findobj('tag','text_nmax'),'userdata');
            if isnan(DPI)==1 || DPI<0
                errordlg('Entered DPI value must be larger
than zero.',...
                    'Display data settings');
                error('Entered DPI value must be larger than
zero.');
```

```

            elseif isnan(ncolor)==1 || ncolor<2
                errordlg('Value of number of colors must be
larger than 1.',....
                    'Display data settings');
                error('Value of number of colors must be
larger than 1.');
```

```

            end
        end
        %Check, if computation of tensors in the LNOF has been
        selected (not allowed)
        if any(volbapar==8) || any(volbapar==9) ||
any(volbapar==14) || any(volbapar==15)
            if volbaALFs==2 && volbagridcheck==1 && STD==0
                errordlg('The following functionals of the
geopotential cannot be computed on a grid using the modified
forward column method combined with Horner's scheme:
Disturbing_tensor_Txx_Tyy_Tzz, Disturbing_tensor_Txy_Txz_Tyz,
Gravitational_tensor_Vxx_Vyy_Vzz and
Gravitational_tensor_Vxy_Vxz_Vyz. In the case of a high maximum
degree (1800 and higher), it is recommended to use the extended-
range arithmetic approach. However, for the point-wise computation
of the above mentioned functionals, the modified forward column
method combined with Horner's scheme can be applied.',...
                    'Calculated parameters and output
selection');
```

```

                error('The following functionals of the
geopotential cannot be computed on a grid using the modified
forward column method combined with Horner's scheme:

```

```

Disturbing_tensor_Txx_Tyy_Tzz, Disturbing_tensor_Txy_Txz_Tyz,
Gravitational_tensor_Vxx_Vyy_Vzz and
Gravitational_tensor_Vxy_Vxz_Vyz. In the case of a high maximum
degree (1800 and higher), it is recommended to use the extended-
range arithmetic approach. However, for the point-wise computation
of the above mentioned functionals, the modified forward column
method combined with Horner's scheme can be applied.');
```

```

    end
    end
    %Check, if display data in point-wise or load data
approach has been
    %selected (not allowed)
    if display_data==1 && volbagridcheck~=1
        warn4=warndlg('Only data computed on a grid can be
displayed. After clicking OK, the computation will start, although
the computed data will not be displayed.');
```

```

        waitfor(warn4);
        display_data=0;
    end
    if STD==0
        tic %Start clock to measure computation time
        %Loading DMR
        if get(findobj('tag','P1'),'value')==10 ||
get(findobj('tag','P2'),'value')==10 ||
get(findobj('tag','P3'),'value')==10 ||
get(findobj('tag','P4'),'value')==10 ||
get(findobj('tag','P1'),'value')==23 ||
get(findobj('tag','P2'),'value')==23 ||
get(findobj('tag','P3'),'value')==23 ||
get(findobj('tag','P4'),'value')==23
            set(findobj('tag','hlasky'),'string','Please
select DTM file',...
                'fontsize',8,'foregroundcolor','k');
```

```

drawnow;
[loadnameDMR,loadadresarDMR]=uigetfile('*.mat','Select File Containg
Spherical Harmonic Coefficients of the Topography');
```

```

        if loadnameDMR==0
            errordlg('To compute
Geoid_undulation/Height_anomaly, DTM file must be imported!',...
                'Error in geopotential model and
reference system selection');
```

```

            error('To compute
Geoid_undulation/Height_anomaly, DTM file must be imported!')
        else
            set(findobj('tag','hlasky'),'string',...
                'Loading DTM file...', 'fontsize',8,...
                'foregroundcolor','k'); drawnow;
            if strcmp(loadnameDMR(end-3:end),'.mat')
                DMR=load([loadadresarDMR,loadnameDMR]);
                DMR=struct2cell(DMR);
                DMR=cell2mat(DMR);
            else
                DMR=load([loadadresarDMR,loadnameDMR]);
            end

            [rows_DMR,cols_DMR]=size(DMR); %#ok<*ASGLU>
            if cols_DMR<4
```



```

                                errorrdlg('Wrong format of the input DTM
file.',...
                                'Geopotential model and reference
system selection');
                                error('Wrong format of the input DTM
file.')
                                end

                                DMR=DMR(:,1:4);
                                set(findobj('tag','hlasky'),'string','',...
'foregroundcolor','k'); drawnow;
                                end
                                end
                                %Loading GGM
                                GGMname=get(findobj('tag','R'),'userdata');
                                GGMadresar=get(findobj('tag','ell'),'userdata');
                                if isempty(GGMname) %Error message, if GGM file
has not been imported
                                errorrdlg('Please input geopotential model
file.',...
                                'Error in point type selection');
                                error('Please input geopotential model file.')
                                end

                                set(findobj('tag','hlasky'),'string',...
'Loading GGM file...','fontsize',8,...
'foregroundcolor','k'); drawnow;
                                %Loading GGM
                                if strcmp(GGMname(end-3:end),'.gfc') %Input data
in ICGEM format
                                fGGMid=fopen([GGMadresar,GGMname]);
                                while(~feof(fGGMid))
                                s=fgetl(fGGMid);
                                if strncmpi(s,'earth_gravity_constant',22)
                                GM=str2num(s(23:end)); %#ok<*ST2NM>
                                end
                                if strncmpi(s,'radius',6)
                                R=str2num(s(7:end));
                                end
                                if strncmpi(s,'end_of_head',11)
                                break
                                end
                                end
                                GGM=textscan(fGGMid,'%s%f%f%f%f%f');
                                fclose(fGGMid);
                                GGM=GGM(2:end);
                                GGM=cell2mat(GGM);
                                elseif strcmp(GGMname(end-3:end),'.mat') %Input
data in MAT format
                                GGM=load([GGMadresar,GGMname]);
                                GGM=struct2cell(GGM);
                                GGM=cell2mat(GGM);
                                else
                                GGM=load([GGMadresar,GGMname]);
                                end
                                [rows_GGM,cols_GGM]=size(GGM);
                                if cols_GGM<4

```

```

                                errordlg('Wrong format of the input GGM
file.',...
                                'Geopotential model and reference system
selection');
                                error('Wrong format of the input GGM file.')
end
GGM=GGM(:,1:4);
GGM=sortrows(GGM,1);
stupen=GGM(:,1);
rad=GGM(:,2);
C=GGM(:,3);
S=GGM(:,4);
nmaxGGM=max(GGM(:,1));
del00=find(stupen==0 & rad==0,1); %If the
coefficients of the degree 0 and order 0 are missing
if isempty(del00)
    stupen=[0;stupen];
    rad=[0;rad];
    C=[1;C];
    S=[0;S];
end
del10=find(stupen==1 & rad==0,1); %If the
coefficients of the degree 1 and order 0 are missing
if isempty(del10)
    stupen=[stupen(1);1;stupen(2:end)];
    rad=[rad(1);0;rad(2:end)];
    C=[C(1);0;C(2:end)];
    S=[S(1);0;S(2:end)];
end
del11=find(stupen==1 & rad==1,1); %If the
coefficients of the degree 1 and order 1 are missing
if isempty(del11)
    stupen=[stupen(1:2);1;stupen(3:end)];
    rad=[rad(1:2);1;rad(3:end)];
    C=[C(1:2);0;C(3:end)];
    S=[S(1:2);0;S(3:end)];
end
clear del00 del10 del11
%Identification of GGM file format
if stupen(1)==0 && stupen(2)==1 && stupen(3)==1 &&
stupen(4)==2 && stupen(5)==2 && rad(1)==0 && rad(2)==0 &&
rad(3)==1 && rad(4)==0 && rad(5)==1
else
                                errordlg('Wrong format of the input GGM
file.',...
                                'Geopotential model and reference system
selection');
                                error('Wrong format of the input GGM file.')
end
set(findobj('tag','hlasky'),'string',...
    '', 'fontsize',8, 'foregroundcolor','k');
drawnow;
                                %Value of nmin and error messages

nmin=str2double(get(findobj('tag','nmin'),'string'));
if nmin<0
                                errordlg('Value of nmin cannot be
negative.',...

```

```

                                'Error in geopotential model and reference
system selection')
                                error('Value of nmin cannot be negative.')
                                elseif nmin>nmaxGGM
                                error('Value of nmin exceeds nmax value of
GGM.',...
                                'Error in geopotential model and reference
system selection')
                                error('Value of nmin exceeds nmax value of
GGM.')
                                end
                                if isnan(nmin)==1
                                error('Please input the nmin
value.',...
                                'Error in geopotential model and
reference system selection')
                                error('Please input the nmin value.')
                                end
                                if rem(nmin,1)~=0
                                error('Value of nmin must be an
integer.',...
                                'Error in geopotential model and reference
system selection')
                                error('Value of nmin must be an integer.')
                                end
                                %Value of nmax and error messages
                                if get(findobj('tag','use'),'value')==1
                                nmax=nmaxGGM;
                                else
nmax=str2double(get(findobj('tag','nmax'),'string'));
                                if nmax>nmaxGGM
                                error('Entered value of nmax exceeds
nmax value of GGM.',...
                                'Error in geopotential model and
reference system selection')
                                error('Entered value of nmax exceeds nmax
value of GGM.')
                                elseif nmax<2
                                error('Value of nmax must be at least
2.',...
                                'Error in geopotential model and
reference system selection')
                                error('Value of nmax must be at least 2.')
                                elseif nmin>nmax
                                error('Value of nmin cannot be larger
than nmax value.',...
                                'Error in geopotential model and
reference system selection')
                                error('Value of nmin cannot be larger than
nmax value.')
                                end
                                if isnan(nmax)==1
                                error('Please input the nmax
value.',...
                                'Error in geopotential model and
reference system selection')
                                error('Please input the nmax value.')

```

```

end
if rem(nmax,1)~=0
    errordlg('Value of nmax must be an
integer.',...
            'Error in geopotential model and
reference system selection')
    error('Value of nmax must be an integer.')
end
end
if nmin>0
    if any(volbapar==20)
        errordlg('Gravity disturbance cannot be
computed if nmin>0.',...
                'Calculated parameters and output
selection');
        error('Gravity disturbance cannot be
computed if nmin>0.');
```

end

```

end
if nmin>0
    if any(volbapar==10) || any(volbapar==23)
        errordlg('Geoid_undulation and
Height_anomaly cannot be computed if nmin>0.');
```

cannot be computed if nmin>0.');

```

    end
    if any(volbapar==9) || any(volbapar==15)
        errordlg('The following functionals of the
geopotential cannot be computed if nmin>0:
Disturbing_tensor_Txy_Txz_Tyz and
Gravitational_tensor_Vxy_Vxz_Vyz.',...
                'Calculated parameters and output
selection');
```

error('The following functionals of the
geopotential cannot be computed if nmin>0:
Disturbing_tensor_Txy_Txz_Tyz and
Gravitational_tensor_Vxy_Vxz_Vyz.');

```

    end
    if any(volbapar==8) || any(volbapar==14)
        LNOFnmin=1; %Logical 1

        if any(volbapar~=8 & volbapar~=9 &
volbapar~=14 & volbapar~=15 & volbapar~=1)
            errordlg('The following functionals of
the geopotential cannot be computed simultaneously with other
functionals if nmin>0: Disturbing_tensor_Txx_Tyy_Tzz and
Gravitational_tensor_Vxx_Vyy_Vzz.',...
                    'Calculated parameters and output
selection');
```

error('The following functionals of
the geopotential cannot be computed simultaneously with other
functionals if nmin>0: Disturbing_tensor_Txx_Tyy_Tzz and
Gravitational_tensor_Vxx_Vyy_Vzz.');

```

        end
    else
        LNOFnmin=0; %Logical 0
    end
end

```

```

end
%Sorting spherical harmonic coefficients of DTM
if get(findobj('tag','P1'),'value')==10 ||
get(findobj('tag','P2'),'value')==10 ||
get(findobj('tag','P3'),'value')==10 ||
get(findobj('tag','P4'),'value')==10 ||
get(findobj('tag','P1'),'value')==23 ||
get(findobj('tag','P2'),'value')==23 ||
get(findobj('tag','P3'),'value')==23 ||
get(findobj('tag','P4'),'value')==23
    DMR(DMR(:,1)>nmaxGGM,:)=[];
    if DMR(1,1)==0 && DMR(2,1)==1 && DMR(3,1)==1
&& DMR(4,1)==2 && DMR(1,2)==0 && DMR(2,2)==0 && DMR(3,2)==1 &&
DMR(4,2)==0
        else
            DMR=sortrows(DMR,1);
        end

        HC=DMR(:,3);
        HS=DMR(:,4);

        clear DMR
    end
%% Computation of functionals of the geopotential
on a regular grid
if volbagridcheck==1
    %Entered coordinates of the grid
    fimin=str2num(get(findobj('tag','fimin'),'string'));
    fistep=str2num(get(findobj('tag','fistep'),'string'));
    fimax=str2num(get(findobj('tag','fimax'),'string'));
    lambdamin=str2num(get(findobj('tag','lambdamin'),'string'));
    lambdastep=str2num(get(findobj('tag','lambdastep'),'string'));
    lambdamax=str2num(get(findobj('tag','lambdamax'),'string'));
    h=str2num(get(findobj('tag','hgrid'),'string'));
    %Check of the entered coordinates
    if isempty(fimin) || isempty(fistep) ||
isempty(fimax) || isempty(lambdamin) || isempty(lambdastep) ||
isempty(lambdamax) || isempty(h) || ~isreal(fimin) ||
~isreal(fistep) || ~isreal(fimax) || ~isreal(lambdamin) ||
~isreal(lambdastep) || ~isreal(lambdamax) || ~isreal(h)
        errordlg('Entered grid is not
correct.',...
                'Error in point type selection');
        error('Entered grid is not correct.');
```

```

elseif lambdamin>lambdamax
    errordlg('Value of Lon. min must be
smaller than Lon. max value.',...
            'Error in point type selection');
    error('Value of Lon. min must be smaller
than Lon. max value.');
```

```

elseif lambdastep<=0
    errordlg('Lon. step must be larger than
zero.',...
            'Error in point type selection');
    error('Lon. step must be larger than
zero.');
```

```

end
if fimin>90 || fimin<-90
    errordlg('Value of Lat. min must be within
the interval <-90°,90°>.',...
            'Error in point type selection');
    error('Value of Lat. min must be within
the interval <-90°,90°>.');
```

```

end
if fimax>90 || fimax<-90
    errordlg('Value of Lat. max must be within
the interval <-90°,90°>.',...
            'Error in point type selection');
    error('Value of Lat. max must be within
the interval <-90°,90°>.');
```

```

end
if lambdamin>360 || lambdamin<-180
    errordlg('Value of Lon. min must be within
the interval <-180°,180°> or <0°,360°>.',...
            'Error in point type selection');
    error('Value of Lon. min must be within
the interval <-180°,180°> or <0°,360°>.');
```

```

end
if lambdamax>360 || lambdamax<-180
    errordlg('Value of Lon. max must be within
the interval <-180°,180°> or <0°,360°>.',...
            'Error in point type selection');
    error('Value of Lon. max must be within
the interval <-180°,180°> or <0°,360°>.');
```

```

end
if (lambdamax-lambdamin)>360
    errordlg('Longitude must be in the range
<-180°,180°> or <0°,360°>.',...
            'Error in point type selection');
    error('Longitude must be in the range <-
180°,180°> or <0°,360°>.');
```

```

end
%Vectors phi and lambda in one longitude and
latitude

%parallel, respectively
fi=(fimin:fistep:fimax)';
lambda=(lambdamin:lambdastep:lambdamax)';
fi=pi/180*(fi(:));
lambda=pi/180*(lambda(:));
%Grid, which is to be displayed has to have at
least two
```

```

                                %points in latitude parallels and at least two
points in                                %longitude parallels.
                                if display_data==1
                                    if length(fi)<2 || length(lambda)<2
                                        warn3=warndlg('To display computed
data on a grid, it must contain at least two distinct points in
one parallel and two distinct points in one meridian. After
clicking OK, the computation will start, although the data will
not be displayed.');
```

waitfor(warn3);
display_data=0;

```

                                end
                                end
                                if coord==1 %Entered spherical coordinates
                                    %Spherical radius
                                    r=(R+h)*ones(length(fi),1);
                                    hsph=h;
                                    %Spherical latitude
                                    fiG=fi;
                                %Transformation of spherical latitude into
the ellipsoidal
                                %latitude
                                [X,Y,Z]=sph2cart(0*fiG,fiG,r);
                                [fi,lambda_del,h]=ecef2geodetic(X,Y,Z,[aEl
eEl]);

                                clear X Y Z lambda_del
                                elseif coord==0 %Entered ellipsoidal
coordinates
                                %Trasformation of (fi, lambda, h) into (X,
Y, Z)
                                [X,Y,Z]=geodetic2ecef(fi,0*zeros(length(fi),1),h*ones(length(fi),1
),[aEl eEl]);

                                r=sqrt(X.*X+Y.*Y+Z.*Z); %Radius
                                %Spherical latitude
                                fiG=atan(Z./sqrt(X.*X+Y.*Y));
                                clear X Y Z
                                end
                                %Computation of the coefficients C0,0; C2,0;
...; C20,0

                                %of the selected ellipsoid
                                CE1=zeros(length(C),1);
                                for n=0:10
                                    CE1(2*n==stupen & rad==0,1)=((-
1)^n*(3*eEl^(2*n))/(2*n+1)*(2*n+3)*sqrt(4*n+1))*(1-n-
5^(3/2)*n*CE1_20/eEl^2)).*(aEl./R).^ (2*n).*(GME1/GM);
                                end
                                if any(volbapar==11) || any(volbapar==12) ||
any(volbapar==13) || any(volbapar==14) || any(volbapar==15) ||
any(volbapar==16) || any(volbapar==17) || any(volbapar==18) ||
any(volbapar==20) || any(volbapar==25)
                                    grav=1;
                                else
                                    grav=0;
                                end
                                if any(volbapar==2) || any(volbapar==3) ||
any(volbapar==4) || any(volbapar==5) || any(volbapar==6) ||
any(volbapar==7) || any(volbapar==8) || any(volbapar==9) ||

```

```

any(volbapar==10) || any(volbapar==19) || any(volbapar==21) ||
any(volbapar==22) || any(volbapar==23) || any(volbapar==24)
    por=1;
    deltaC=C-CEl;
else
    por=0;
end
if any(volbapar==20)
    normal=1;
else
    normal=0;
end
clear GGM stupen rad
if normal==0
    clear CE1
end
%Initialization
eta=0; ksi=0; Theta=0; T=0; T_rr=0; Trr=0;
Trf=0; Trl=0; Tff=0; %#ok<*NASGU>
Tfl=0; Tll=0; Tzz=0; Txx=0; Tyy=0; N=0; V=0;
Vrr=0; Vrf=0; Vrl=0; Vff=0;
Vfl=0; Vll=0; g=0; g_sa=0; W=0; anomalia_sa=0;
porucha=0;
porucha_sa=0; zetaEl=0; zeta=0; Wrr=0; Wr=0;
Wfi=0; Wlambda=0;
Ur=0; Ufi=0; N1c=0; N2c=0; H=0;
%If geoid/quasigeoid is to be computed
if any(volbapar==10) || any(volbapar==23)
    geoid=1;
    if h~=0
        errordlg('To compute Geoid_undulation
or Height_anomaly, the ellipsoidal height must be set to
zero.',...
                'Error in point type selection');
        error('To compute Geoid_undulation or
Height_anomaly, the ellipsoidal height must be set to zero.');
```



```

bit=bit(end-1:end);
bit=str2double(bit);
if bit==32
    bit=32;
elseif bit==64
    bit=64;
else
    bit=64;
end
nmax23=nmax*2+3;
rr=zeros(nmax23,1); ri=rr;
dd=zeros(nmax,1); am=dd; bm=am;

m1=1:nmax23;
rr(m1)=sqrt(m1);
ri(m1)=1./rr;
m2=1:nmax;
dd(m2)=rr(2*m2+3).*ri(2*m2+2);
IND=960;
BIG=2^IND;
BIGI=2^(-IND);
BIGS=2^(IND/2);
BIGSI=2^(-IND/2);
ROOT3=1.732050807568877;
if bit==32
    pm=am;
    ps1=zeros(length(fiG),nmax);
    ips1=ps1;
    x=ROOT3*u.*q;
    ix=zeros(size(x));
    ps1(:,1)=x;
    ips1(:,1)=ix;
    for m3=2:nmax
        x=(dd(m3-1)*u).*x.*q;
        y=abs(x);
        iy=y>=BIGS;
        if any(iy)
            x(iy)=x(iy)*BIGI;
            ix(iy)=ix(iy)+1;
        end
        iy=y<BIGSI;
        if any(iy)
            x(iy)=x(iy)*BIG;
            ix(iy)=ix(iy)-1;
        end
        ps1(:,m3)=x;
        ips1(:,m3)=ix;
    end
elseif bit==64
    tq=t.*q;
    temp1=zeros(length(fiG),1);
    temp2=ones(length(fiG),1);
    temp3=temp2;
    temp4=temp1;
    temp5=temp1+BIGI;
    ps1b=zeros(length(fiG),nmax);
    ips1b=ps1b;

```

```

        xb=ROOT3*u.*q;
        ixb=zeros(size(xb));
        pslb(:,1)=xb;
        ipslb(:,1)=ixb;
        for m3=2:nmax
            xb=(dd(m3-1)*u).*xb.*q;
            yb=abs(xb);
            iyb=yb>=BIGS;
            if any(iyb)
                xb(iyb)=xb(iyb)*BIGI;
                ixb(iyb)=ixb(iyb)+1;
            end
            iyb=yb<BIGSI;
            if any(iyb)
                xb(iyb)=xb(iyb)*BIG;
                ixb(iyb)=ixb(iyb)-1;
            end
            pslb(:,m3)=xb;
            ipslb(:,m3)=ixb;
        end
    end
    clear dd
end
%Initialization of the matrices and vectors
for the
fnALFs
%computation of the first-order derivatives of
    if any(volbapar==3) || any(volbapar==4) ||
any(volbapar==6) || any(volbapar==7) || any(volbapar==12) ||
any(volbapar==13) || any(volbapar==16) || any(volbapar==20)
        dALFs=1;
        dPnm=zeros(length(fi),nmax+1);
        qu=q./u;
        tu=t./u;
        %Treatment of the dPnm singularity
        singdPnm=fi==pi/2 | fi==-pi/2;
    else
        dALFs=0;
    end
end
%Initialization of the matrices and vectors
for the
of fnALFs
%computation of the second-order derivatives
    if any(volbapar==6) || any(volbapar==12)
        ddALFs=1;
        ddPnm=zeros(length(fi),nmax+1);
        %Treatment of the ddPnm singularity
        singddPnm=fi==pi/2 | fi==-pi/2;
    else
        ddALFs=0;
    end
end
%Status line
progressbar=findobj('tag','hlasky');
%% Summation over m
for m=nmax:-1:0
    %Update of the progress bar
    if rem(m,10)==0
        set(progressbar,'string',...

```

```

                                sprintf('Progress: m =
%5.0d',m),...
                                'fontsize',8); drawnow;
                                end

                                %Selection of the spherical harmonic
                                coefficients of order m
                                %=====
                                if grav==1 %C's spherical harmonic
                                coefficients for the functionals without the normal gravity field
                                    Cm=C(index((m+1):end)+m);
                                end
                                if por==1 %C's spherical harmonic
                                coefficients for the functionals with the disturbing field
                                    deltaCm=deltaC(index((m+1):end)+m);
                                end
                                if normal==1 %C's spherical harmonic
                                coefficients for the functionals with the normal field
                                    if m==0
                                        CE1m=CE1(index((m+1):end)+m);
                                    end
                                end
                                if geoid==1
                                    HCm=HC(index((m+1):end)+m);
                                    HSm=HS(index((m+1):end)+m);
                                end
                                end
                                Sm=S(index((m+1):end)+m);

                                %=====
                                %% Computation of the modified fnALFs
                                %=====
                                if volbaALFs==1 %Standard forward column
                                method
                                    if m==0
                                        Pnm(:,1)=1;
                                    elseif m==1
                                        Pnm(:,1)=sqrt(3)*u.*q;
                                    elseif m>1
                                        i=2*(2:m);
                                        i1=sqrt((i+ones(size(i)))./i);
                                        Pnm(:,1)=u.^m*sqrt(3)*prod(i1).*q.^m;
                                    end
                                    if m==nmax
                                        elseif m<=(nmax-1)
                                            n=m+1;
                                            anm=sqrt((2*n-1)*(2*n+1)/((n-
                                m)*(n+m)));

                                            Pnm(:,2)=anm*t.*Pnm(:,1).*q;
                                    end
                                    if m<(nmax-1)
                                        j=3;
                                        for n=m+2:nmax
                                            anm=sqrt((2*n-1)*(2*n+1)/((n-
                                m)*(n+m)));

                                            bnm=sqrt((2*n+1)*(n+m-1)*(n-m-
                                1)/((n-m)*(n+m)*(2*n-3)));

```

```

Pnm(:,j)=anm*t.*Pnm(:,j-1).*q-
bnm*Pnm(:,j-2).*q2;
j=j+1;
end
end
elseif volbaALFs==2 %Modified forward
column method
if m==0
Pnm(:,1)=1e-280;
elseif m==1
Pnm(:,1)=sqrt(3)*q*1e-280;
elseif m>1
i=2*(2:m);
i1=sqrt((i+ones(size(i)))./i);
Pnm(:,1)=sqrt(3)*prod(i1)*(q.^m)*1e-280;
end
if m==nmax
elseif m<=(nmax-1)
n=m+1;
anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
Pnm(:,2)=anm*t.*Pnm(:,1).*q;
end
if m<(nmax-1)
j=3;
for n=m+2:nmax
anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
bnm=sqrt((2*n+1)*(n+m-1)*(n-m-
1)/((n-m)*(n+m)*(2*n-3)));
Pnm(:,j)=anm*t.*Pnm(:,j-1).*q-
bnm*Pnm(:,j-2).*q2;
j=j+1;
end
end
elseif volbaALFs==3 %Extended-range
arithmetic
if bit==32 %32 bit version of Matlab
am(m+1)=rr(2*m+3);
for n=m+2:nmax
ww=rr(2*n+1)*ri(n-m)*ri(n+m);
am(n)=rr(2*n-1)*ww;
bm(n)=rr(n-m-1)*rr(n+m-
1)*ri(2*n-3)*ww;
end
if m~=0
for i=1:length(fiG)
x=ps1(i,m);
ix=ips1(i,m);
if ix==0
pm(m)=x;
elseif ix<-1
pm(m)=0;
elseif ix<0

```

```

pm(m)=x*BIGI;
else
pm(m)=x*BIG;
end

if m==nmax
Pnm(i,1:(nmax-
m+1))=pm(m:end);

continue;
end
y=x;
iy=ix;
x=(am(m+1)*t(i)*q(i))*y;
ix=iy;
w=abs(x);
if w>=BIGS
x=x*BIGI;
ix=ix+1;
elseif w<BIGSI
x=x*BIG;
ix=ix-1;
end
if ix==0
pm(m+1)=x;
elseif ix<-1
pm(m+1)=0;
elseif ix<0
pm(m+1)=x*BIGI;
else
pm(m+1)=x*BIG;
end
for n=m+2:nmax
id=ix-iy;
if id==0

zz=(am(n)*t(i)*q(i))*x-bm(n)*q2(i)*y;
iz=ix;
elseif id==1
zz=(am(n)*t(i)*q(i))*x-bm(n)*q2(i)*(y*BIGI);
iz=ix;
elseif id==-1
zz=(am(n)*t(i)*q(i))*(x*BIGI)-bm(n)*q2(i)*y;
iz=iy;
elseif id>1
zz=(am(n)*t(i)*q(i))*x;
iz=ix;
else
zz=-bm(n)*q2(i)*y;
iz=iy;
end
w=abs(zz);
if w>=BIGS
zz=zz*BIGI;
iz=iz+1;
elseif w<BIGSI
zz=zz*BIG;
iz=iz-1;

```

```

end
    if iz==0
        pm(n)=zz;
    elseif iz<-1
        pm(n)=0;
    elseif iz<0
        pm(n)=zz*BIGI;
    else
        pm(n)=zz*BIG;
    end
    y=x;
    iy=ix;
    x=zz;
    ix=iz;
end
Pnm(i,1:(nmax-
m+1))=pm(m:end);
end
elseif m==0
    Pnm(:,1)=1;
    Pnm(:,2)=sqrt(3)*t.*q;
    for i=2:nmax
Pnm(:,i+1)=Pnm(:,i).*sqrt((2*i+1)*(2*i-1))./i.*t.*q-q2.*Pnm(:,i-
1).*(i-1).*sqrt(2.*i+1)./(i.*sqrt(2.*i-3));
    end
clear rr ri am bm pm ps1 ips1
m1 m2 ...
        dd ix x y iy w iz zz
    end
elseif bit==64 %64 bit version of
Matlab
    am(m+1)=rr(2*m+3);
    for n=m+2:nmax
        ww=rr(2*n+1)*ri(n-m)*ri(n+m);
        am(n)=rr(2*n-1)*ww;
        bm(n)=rr(n-m-1)*rr(n+m-
1)*ri(2*n-3)*ww;
    end
    if m==0 %Zonal modified fnALFs
        Pnm(:,1)=1;
        Pnm(:,2)=sqrt(3)*tq;
        for i=2:nmax
Pnm(:,i+1)=Pnm(:,i).*sqrt((2*i+1)*(2*i-1))./i.*tq-q2.*Pnm(:,i-
1).*(i-1).*sqrt(2.*i+1)./(i.*sqrt(2.*i-3));
        end
clear rr ri am bm ps1b ips1b
m1 m2 dd ...
        ixb xb yb iyb wb izb zzb
pmxb pm0b pmxBIGIb ...
        pmxBIGb wb wBIGSb wBIGSIb
pmlxb pml0b ...
        pmlxBIGIb pmlxBIGb idb
id0b id1b id_1b ...
        idv1b idmlb iz0b izm_1b
izm0b izv0b tq templ...
        temp2 temp3 temp4 temp5
elseif m~=0 %Non-zonal modified
fnALFs

```

```

xb=ps1b(:,m);
ixb=ips1b(:,m);
temp5(ixb==0)=1;
temp5(ixb<-1)=0;
%temp5(izb>=-1 & izb<0)=BIGI;
%The condition "izb>=-1 &
izb<0"
already
%is useless, as "izb" is
%initialized as "izb=BIGI".
temp5(ixb>0)=BIG;
Pnm(:,1)=xb.*temp5;
temp5=temp5.*0+BIGI;
if m<nmax
    yb=xb;
    iyb=ixb;
    xb=(am(m+1).*tq).*yb;
    ixb=iyb;
    wb=abs(xb);
    wBIGSb=wb>=BIGS;
    wBIGSIb=wb<BIGSI;
    temp3(wBIGSb)=BIGI;
    temp3(wBIGSIb)=BIG;
    temp4(wBIGSb)=1;
    temp4(wBIGSIb)=-1;
    xb=xb.*temp3;
    ixb=ixb+temp4;
    temp3=temp2;
    temp4=temp4.*0;
    temp5(ixb==0)=1;
    temp5(ixb<-1)=0;
    %temp5(izb>=-1 &
izb<0)=BIGI;
%The condition "izb>=-1 &
izb<0"
already
%is useless, as "izb" is
%initialized as "izb=BIGI".
temp5(ixb>0)=BIG;
Pnm(:,2)=xb.*temp5;
temp5=temp5.*0+BIGI;
for n=m+2:nmax
    idb=ixb-iyb;
    id0b=idb==0;
    id1b=idb==1;
    id_1b=idb==-1;
    idv1b=idb>1;
    temp1(id0b)=1;
    temp1(id1b)=1;
    temp2(id1b)=BIGI;
    temp1(id_1b)=BIGI;
    temp1(idv1b)=1;
    temp2(idv1b)=0;
zzb=(am(n).*tq).*(xb.*temp1)-bm(n).*((yb.*q2).*temp2);
    izb=iyb;
    id0b_id1b_idv1b=id0b |
id1b | idv1b;
izb(id0b_id1b_idv1b)=ixb(id0b_id1b_idv1b);

```

```

temp1=temp1.*0;
temp2=temp1+1;
wb=abs(zzb);

wBIGSb=wb>=BIGS;
wBIGSIb=wb<BIGSI;
temp3(wBIGSb)=BIGI;
temp3(wBIGSIb)=BIG;
temp4(wBIGSb)=1;
temp4(wBIGSIb)=-1;
zzb=zzb.*temp3;
izb=izb+temp4;
temp3=temp2;
temp4=temp4.*0;
temp5(izb==0)=1;
temp5(izb<-1)=0;
%temp5(izb>=-1 &

izb<0)=BIGI;

& izb<0"

is already

"izb=BIGI".

m+1)=zzb.*temp5;

%The condition "izb>=-1

%is useless, as "izb"

%initialized as

temp5(izb>0)=BIG;
Pnm(:,n-

temp5=temp1+BIGI;
yb=xb;
iyb=ixb;
xb=zzb;
ixb=izb;

end
end
end
end
end
%If nmin~=0
=====
if nmin~=0
    if LNOFnmin==0
        if por==1
            deltaCm(1:(nmin-m))=0;
        end
        if grav==1
            Cm(1:(nmin-m))=0;
        end
        if geoid==1
            HCm(1:(nmin-m))=0;
            HSm(1:(nmin-m))=0;
        end
        Sm(1:(nmin-m))=0;
    elseif LNOFnmin==1
        Pnm(:,1:(nmin-m))=0;
    end
end
=====

```



```

                                %% Computation of the first-order
derivatives of the modified fnALFs
                                if dALFs==1
                                    if volbaALFs==1 || volbaALFs==3

enm=sqrt((2*(m:nmax)+1).*(m:nmax)-m).*(m:nmax)+m)./(2*(m:nmax)-
1));

                                if m==0 %Zonal modified dALFs
                                    dPnm(:,1)=0.*u;
                                    dPnm(:,2)=sqrt(3)*u.*q;
                                    dPnm(:,3:end)=-
bsxfun(@times,2:nmax,tu).*Pnm(:,3:end)+(bsxfun(@times,enm(3:end),q
u)).*Pnm(:,2:(end-1)));

                                elseif m==nmax %Sectorial modified
dALFs
                                    dPnm(:,1)=-m*(tu).*Pnm(:,1);
                                else
                                    dPnm(:,1)=-m*(tu).*Pnm(:,1);

%Tesseral modified dALFs
                                    dPnm(:,2:(nmax-m+1))=-
bsxfun(@times,(m+1):nmax,tu).*Pnm(:,2:(nmax-
m+1))+bsxfun(@times,enm(2:end),qu).*Pnm(:,1:(nmax-m)); %Tesimalne
ALFs

                                end
                                elseif volbaALFs==2

enm=sqrt((2*(m:nmax)+1).*(m:nmax)-m).*(m:nmax)+m)./(2*(m:nmax)-
1));

                                if m==0 %Zonal modified dALFs
                                    dPnm(:,1)=0.*u*1e-280;
                                    dPnm(:,2)=sqrt(3)*u.*q*1e-280;
                                    dPnm(:,3:end)=-
bsxfun(@times,2:nmax,tu).*Pnm(:,3:end)+(bsxfun(@times,enm(3:end),q
u)).*Pnm(:,2:(end-1)));

                                elseif m==nmax
                                    dPnm(:,1)=-m*(tu).*Pnm(:,1);

%Sectorial modified dALFs
                                else
                                    dPnm(:,1)=-m*(tu).*Pnm(:,1);

%Tesseral modified dALFs
                                    dPnm(:,2:(nmax-m+1))=-
bsxfun(@times,(m+1):nmax,tu).*Pnm(:,2:(nmax-
m+1))+bsxfun(@times,enm(2:end),qu).*Pnm(:,1:(nmax-m)); %Tesimalne
dALFs

                                end
                                end
                                %Treatment of the dALFs singularity
                                dPnm(singdPnm,:)=0;
                                if ddALFs==1 %If the second-order
derivatives of the modified fnALFs are to be computed
                                    if m==0 %Zonal modified ddALFs
                                        ddPnm=bsxfun(@times,tu,dPnm)-
bsxfun(@times,(0:nmax).*(0:nmax)+1),Pnm);
                                    else
                                        ddPnm(:,1:end-
m)=bsxfun(@times,tu,dPnm(:,1:end-
m))+bsxfun(@times,m^2./u.^2,Pnm(:,1:end-m))-
bsxfun(@times,(m:nmax).*(m:nmax)+1),Pnm(:,1:end-m));
                                    end

```

```

                                %Treatment of the ddALFs
singularity                      ddPnm(singddPnm, :)=0;
                                end
                                end
                                %% Loop for 1:Nf (number of computing
functional)
                                for i=1:pocetpar
                                if volbapar(i)==1
                                elseif volbapar(i)==2 %Deflection of
the vertical eta
                                Lm=m*Pnm(:, 1:(nmax-m+1));
                                if volbaALFs==1 || volbaALFs==3
                                %Computation using the standard
                                %forward column method or
                                extended-range arithmetic
                                if m==nmax
                                end
                                Aeta(:, m+1)=Lm*deltaCm;
                                Beta(:, m+1)=Lm*Sm;
                                elseif volbaALFs==2 %Computation
using the modified forward column method combined with Horner's
scheme
                                if m==nmax
                                eta=zeros(length(fiG), length(lambda));
                                end
                                eta=bsxfun(@times, eta, u)+(-
Lm*deltaCm*sin(m*lambda')+Lm*Sm*cos(m*lambda'));
                                end
                                elseif volbapar(i)==3 %Deflection of
the vertical xi
                                if m==nmax
                                Aksi=zeros(length(fiG), nmax+1);
                                Bksi=zeros(length(fiG), nmax+1);
                                end
                                dLm=dPnm(:, 1:(nmax-m+1));
                                if volbaALFs==1 || volbaALFs==3
                                if m==nmax
                                Aksi=zeros(length(fiG), nmax+1);
                                Bksi=zeros(length(fiG), nmax+1);
                                end
                                Aksi(:, m+1)=dLm*deltaCm;
                                Bksi(:, m+1)=dLm*Sm;
                                elseif volbaALFs==2
                                if m==nmax
                                kxi=zeros(length(fiG), length(lambda));
                                end
                                kxi=bsxfun(@times, kxi, u)+(dLm*deltaCm*cos(m*lambda')+dLm*Sm*sin(m*
lambda'));
                                end
                                elseif volbapar(i)==4 %Deflection of
the vertical Theta
                                Lm=m*Pnm(:, 1:(nmax-m+1));
                                dLm=dPnm(:, 1:(nmax-m+1));
                                if volbaALFs==1 || volbaALFs==3

```

```

        if m==nmax

ATeta=zeros (length (fiG) ,nmax+1);

BTeta=zeros (length (fiG) ,nmax+1);

        ATksi=ATeta;
        BTksi=BTeta;
    end
    ATeta (:,m+1)=Lm*deltaCm;
    BTeta (:,m+1)=Lm*Sm;
    ATksi (:,m+1)=dLm*deltaCm;
    BTksi (:,m+1)=dLm*Sm;
elseif volbaALFs==2
    if m==nmax
Teta=zeros (length (fiG) ,length (lambda));
Tksi=zeros (length (fiG) ,length (lambda));
        end
        Teta=bsxfun (@times,Teta,u) + (-
Lm*deltaCm*sin (m*lambda') +Lm*Sm*cos (m*lambda'));
Tksi=bsxfun (@times,Tksi,u) + (dLm*deltaCm*cos (m*lambda') +dLm*Sm*sin (
m*lambda'));
        end
elseif volbapar (i)==16 %Gravity
    if m==nmax
        ampl_Wr= ((0:nmax)+1);
    end
    LmWr=bsxfun (@times,Pnm (:,1:(nmax-
m+1)),ampl_Wr (:,(m+1):end));
    LmWlambda=m*Pnm (:,1:(nmax-m+1));
    LmWfi=dPnm (:,1:(nmax-m+1));
    if volbaALFs==1 || volbaALFs==3
        if m==nmax
            AWlambda=AWr;
            BWlambda=BWr;
            AWfi=AWr;
            BWfi=BWr;
        end
        AWr (:,m+1)=LmWr*Cm;
        BWr (:,m+1)=LmWr*Sm;
        AWlambda (:,m+1)=LmWlambda*Cm;
        BWlambda (:,m+1)=LmWlambda*Sm;
        AWfi (:,m+1)=LmWfi*Cm;
        BWfi (:,m+1)=LmWfi*Sm;
    elseif volbaALFs==2
        if m==nmax
Wr=zeros (length (fiG) ,length (lambda));

            Wlambda=Wr;
            Wfi=Wr;
        end
        Wr=bsxfun (@times,Wr,u) + (LmWr*Cm*cos (m*lambda') +LmWr*Sm*sin (m*lambda'));

        Wlambda=bsxfun (@times,Wlambda,u) + (-
LmWlambda*Cm*sin (m*lambda') +LmWlambda*Sm*cos (m*lambda'));
        Wfi=bsxfun (@times,Wfi,u) + (LmWfi*Cm*cos (m*lambda') +LmWfi*Sm*sin (m*lambda'));
    end
end

```

```

end
elseif volbapar(i)==8 %Disturbing tensor Txx_Tyy_Tzz

    if m==nmax
    ampl_Tzz=(0:nmax)+1.*(0:nmax)+2);
    end
    if m<2
    Lm=bsxfun(@times,Pnm(:,1:(nmax-m+1)),ampl_Tzz(:,(m+1):end));
    %Txx
    bnm=((m:nmax)+m+1).*(m:nmax)+m+2)./2./(m+1);
    LmTxx1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),bnm-
    ((m:nmax)+1).*(m:nmax)+2));
    LmTyy1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),bnm);
    if m==0
    anm=sqrt(2)/4.*sqrt((m:nmax).^2-(m+2-
    1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)-(m+2)+2);
    else
    anm=1./4.*sqrt((m:nmax).^2-(m+2-
    1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)-(m+2)+2);
    end
    LmTxx2=bsxfun(@times,Pnm(:,1:(nmax-m+1)),anm);
    LmTxx3=zeros(length(fi),nmax-
    1); %Coefficients cnm are equal to zeros, if m==0 a m==1
    else
    Lm=bsxfun(@times,Pnm(:,1:(nmax-m+1)),ampl_Tzz(:,(m+1):end));
    %Txx
    bnm=((m:nmax).^2+m^2+3.*(m:nmax)+2)./2;
    LmTxx1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),bnm-
    ((m:nmax)+1).*(m:nmax)+2));
    LmTyy1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),bnm);
    anm=1/4.*sqrt((m:nmax).^2-
    (m+2-1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)-(m+2)+2);
    LmTxx2=bsxfun(@times,Pnm(:,1:(nmax-m+1)),anm);
    if m==2
    cnm=sqrt(2)/4.*sqrt((m:nmax).^2-(m-2+1).^2).*sqrt((m:nmax)-(m-
    2)).*sqrt((m:nmax)+m-2+2);
    else
    cnm=1./4.*sqrt((m:nmax).^2-(m-2+1).^2).*sqrt((m:nmax)-(m-
    2)).*sqrt((m:nmax)+m-2+2);
    end
    LmTxx3=bsxfun(@times,Pnm(:,1:(nmax-m+1)),cnm);
    end
    if m==nmax
    ATzz=zeros(length(fiG),nmax+1);
    BTzz=zeros(length(fiG),nmax+1);

    ATxx1=ATzz;
    BTxx1=ATzz;
    ATxx2=ATzz;
    BTxx2=ATzz;
    ATxx3=ATzz;
    BTxx3=BTzz;
    ATyy1=ATzz;
    BTyy1=BTzz;
    end
    ATzz(:,m+1)=Lm*deltaCm;
    BTzz(:,m+1)=Lm*Sm;
    %Txx

```

```

ATxx1(:,m+1)=LmTxx1*deltaCm;
BTxx1(:,m+1)=LmTxx1*Sm;
if m<2

ATxx2(:,m+1)=LmTxx2*deltaC(index((m+1):end)+m+2);
BTxx2(:,m+1)=LmTxx2*S(index((m+1):end)+m+2);
    ATxx3(:,m+1)=0;
    BTxx3(:,m+1)=0;
elseif m>nmax-2
    ATxx2(:,m+1)=0;
    BTxx2(:,m+1)=0;

ATxx3(:,m+1)=LmTxx3*deltaC(index((m+1):end)+m-2);
BTxx3(:,m+1)=LmTxx3*S(index((m+1):end)+m-2);
    else
ATxx2(:,m+1)=LmTxx2*deltaC(index((m+1):end)+m+2);
BTxx2(:,m+1)=LmTxx2*S(index((m+1):end)+m+2);
ATxx3(:,m+1)=LmTxx3*deltaC(index((m+1):end)+m-2);
BTxx3(:,m+1)=LmTxx3*S(index((m+1):end)+m-2);
    end
    %Tyy
    ATyy1(:,m+1)=LmTyy1*deltaCm;
    BTyy1(:,m+1)=LmTyy1*Sm;

%Modified forward column method cannot be
%applied
elseif volbapar(i)==9 %Disturbing

tensor Txy_Txz_Tyz

    if m<2
        if m==0
betanm=( (m:nmax)+2) ./2.*sqrt(1+ones(1,nmax+1)).*sqrt((m:nmax)+m+1)
.*sqrt((m:nmax)-(m+1)+1);
            gamanm=zeros(1,nmax+1);
            %Txy
dnm=1/4.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt(2).*sqrt((m:nmax).^2-(m+2-
1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)+m+2-2);
LmTxy1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],dnm);
LmTxy2=zeros(length(fi),nmax+1);
LmTxy3=zeros(length(fi),nmax+1);
            %Tyz
minm=( (m:nmax)+2) ./2.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt(2).*sqrt((m:nmax)+m+1).*sqrt((m:nmax)+m+1-1);
LmTyz1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],minm);
LmTyz2=zeros(length(fi),nmax+1);
        else
betanm=( (m:nmax)+2) ./2.*sqrt((m:nmax)+m+1).*sqrt((m:nmax)-
(m+1)+1);
            gamanm=-
((m:nmax)+2).*sqrt((m:nmax).*((m:nmax)+1)/2);
            %Txy
dnm=1/4.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-1)).*sqrt((m:nmax).^2-
(m+2-1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)+m+2-2);
LmTxy1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],dnm);

```

```

                                gnm=-
1/4*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt((m:nmax)+1).*sqrt((m:nmax)-1).*(m:nmax)+2);
LmTxy2=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],gnm);
LmTxy3=zeros(length(fi),nmax);

                                %Tyz
minm=((m:nmax)+2)/2.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt((m:nmax)+m+1).*sqrt((m:nmax)+m+1-1);
LmTyz1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],minm);
LmTyz2=zeros(length(fi),nmax);

                                end
                                %Tzx
LmTzx1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),betanm);

LmTzx2=bsxfun(@times,Pnm(:,1:(nmax-m+1)),gamanm);
                                else
                                %Tzx
betanm=((m:nmax)+2)/2.*sqrt((m:nmax)+m+1).*sqrt((m:nmax)-
(m+1)+1);

                                gamanm=-
((m:nmax)+2)/2.*sqrt((m:nmax)-(m-1)).*sqrt((m:nmax)+m-1+1);
LmTzx1=bsxfun(@times,Pnm(:,1:(nmax-m+1)),betanm);
LmTzx2=bsxfun(@times,Pnm(:,1:(nmax-m+1)),gamanm);
                                %Txy
dnm=1/4.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-1)).*sqrt((m:nmax).^2-
(m+2-1).^2).*sqrt((m:nmax)+m+2).*sqrt((m:nmax)+m+2-2);
LmTxy1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],dnm);

                                gnm=-
m/2*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt((m:nmax)+m).*sqrt((m:nmax)-m);
LmTxy2=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],gnm);

                                if m==2
                                %Txy
LmTxy3=zeros(length(fi),nmax-1);

                                elseif m==3
                                %Txy
                                hnm=-
1/4*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-1)).*sqrt((m:nmax)-
3).*sqrt((m:nmax)-2).*sqrt((m:nmax)-1).*sqrt((m:nmax)+2);
LmTxy3=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],hnm);

                                else
                                %Txy
                                hnm=-
1/4*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-1)).*sqrt((m:nmax).^2-(m-
2+1).^2).*sqrt((m:nmax)-(m-2)).*sqrt((m:nmax)-(m-2)-2);
LmTxy3=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],hnm);

                                end
                                %Tyz
minm=((m:nmax)+2)/2.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt((m:nmax)+m+1).*sqrt((m:nmax)+m+1-1);
LmTyz1=bsxfun(@times,[zeros(length(fiG),1) Pnm(:,1:(nmax-
m))],minm);

```

```

ninm=(m:nmax)+2)./2.*sqrt((2*(m:nmax)+1)/(2*(m:nmax)-
1)).*sqrt((m:nmax)-(m-1)).*sqrt((m:nmax)-(m-1)-1);
LmTy2=bsxfun(@times,[zeros(length(fiG),1) Prm(:,1:(nmax-
m))],ninm);
end
if m==nmax
ATxz1=zeros(length(fiG),nmax+1);
BTxz1=zeros(length(fiG),nmax+1);
ATxz2=ATxz1;
BTxz2=ATxz1;
ATxy1=ATxz1;
BTxy1=ATxz1;
ATxy2=ATxz1;
BTxy2=ATxz1;
ATxy3=ATxz1;
BTxy3=ATxz1;
ATyz1=ATxz1;
BTyz1=ATxz1;
ATyz2=ATxz1;
BTyz2=ATxz1;
end
ATxy2(:,m+1)=(LmTxy2*deltaCm).*q;
BTxy2(:,m+1)=(LmTxy2*Sm).*q;
if m<2
ATxz1(:,m+1)=LmTxx1*deltaC(index((m+1):end)+m+1);
BTxz1(:,m+1)=LmTxx1*S(index((m+1):end)+m+1);
if m==1
%Txx
ATxz2(:,m+1)=LmTxx2*deltaC(index((m+1):end)+m-1);
BTxz2(:,m+1)=LmTxx2*S(index((m+1):end)+m-1);
%Tyx
ATyz2(:,m+1)=LmTyx2*deltaC(index((m+1):end)+m-1).*q;
BTyz2(:,m+1)=LmTyx2*S(index((m+1):end)+m-1).*q;
else
%Txx
ATxz2(:,m+1)=0;
BTxz2(:,m+1)=0;
%Tyx
ATyz2(:,m+1)=0;
BTyz2(:,m+1)=0;
end
%Txy
ATxy1(:,m+1)=(LmTxy1*deltaC(index((m+1):end)+m+2)).*q;
BTxy1(:,m+1)=(LmTxy1*S(index((m+1):end)+m+2)).*q;
ATxy3(:,m+1)=0;
BTxy3(:,m+1)=0;
%Tyx
ATyz1(:,m+1)=(LmTyx1*deltaC(index((m+1):end)+m+1)).*q;
BTyz1(:,m+1)=(LmTyx1*S(index((m+1):end)+m+1)).*q;
elseif m>nmax-2
if m==nmax
%Txx
ATxz1(:,m+1)=0;
BTxz1(:,m+1)=0;
%Tyx

```

```

ATyz1(:,m+1)=0;
BTyz1(:,m+1)=0;
else
    %Txz
ATxz1(:,m+1)=LmTxz1*deltaC(index((m+1):end)+m+1);
BTxz1(:,m+1)=LmTxz1*S(index((m+1):end)+m+1);
    %Tyz
ATyz1(:,m+1)=LmTyz1*deltaC(index((m+1):end)+m+1).*q;
BTyz1(:,m+1)=LmTyz1*S(index((m+1):end)+m+1).*q;
end
ATxz2(:,m+1)=LmTxz2*deltaC(index((m+1):end)+m-1);
BTxz2(:,m+1)=LmTxz2*S(index((m+1):end)+m-1);
    %Txy
ATxy1(:,m+1)=0;
BTxy1(:,m+1)=0;
ATxy3(:,m+1)=(LmTxy3*deltaC(index((m+1):end)+m-2)).*q;
BTxy3(:,m+1)=(LmTxy3*S(index((m+1):end)+m-2)).*q;
    %Tyz
ATyz2(:,m+1)=LmTyz2*deltaC(index((m+1):end)+m-1).*q;
BTyz2(:,m+1)=LmTyz2*S(index((m+1):end)+m-1).*q;
else
ATxz1(:,m+1)=LmTxz1*deltaC(index((m+1):end)+m+1);
BTxz1(:,m+1)=LmTxz1*S(index((m+1):end)+m+1);
ATxz2(:,m+1)=LmTxz2*deltaC(index((m+1):end)+m-1);
BTxz2(:,m+1)=LmTxz2*S(index((m+1):end)+m-1);
    %Txy
ATxy1(:,m+1)=(LmTxy1*deltaC(index((m+1):end)+m+2)).*q;
BTxy1(:,m+1)=(LmTxy1*S(index((m+1):end)+m+2)).*q;
ATxy3(:,m+1)=(LmTxy3*deltaC(index((m+1):end)+m-2)).*q;
BTxy3(:,m+1)=(LmTxy3*S(index((m+1):end)+m-2)).*q;
    %Tyz
ATyz1(:,m+1)=LmTyz1*deltaC(index((m+1):end)+m+1).*q;
BTyz1(:,m+1)=LmTyz1*S(index((m+1):end)+m+1).*q;
ATyz2(:,m+1)=LmTyz2*deltaC(index((m+1):end)+m-1).*q;
BTyz2(:,m+1)=LmTyz2*S(index((m+1):end)+m-1).*q;
end
%% Computation of functionals in point-wise
%=====
    %Identification of point-wise approach
    if volbadiskcheck==1
        %Ellipsoidal coordinates
        fi=str2num(get(findobj('tag','fi'),'string'))';
        lambda=str2num(get(findobj('tag','lambda'),'string'))';
        h=str2num(get(findobj('tag','hdisk'),'string'))';
        end
        %Identification of load data approach
        if volbaloadcheck==1
            %Import of data file containing ellipsoidal
coordinates
            loadname=get(findobj('tag','use'),'userdata');
            loadadresar=get(findobj('tag','diskcheck'),'userdata');
            if isempty(loadname) %Error message, if GGM
file has not been imported
                errordlg('Please input the data file
containing coordinates of the computing points.',...
                    'Error in point type selection');
                error('Please input the data file
containing coordinates of the computing points.')
            end
        end

```



```

end

if strcmp(loadname(end-3:end),'.mat') %Loading
MAT file
    Import=load([loadadresar,loadname]);
    Import=struct2cell(Import);
    Import=cell2mat(Import);
    Import=Import(:,1:3);
else
    Import=load([loadadresar,loadname]);
end
[rows_Import,cols_Import]=size(Import);
if cols_Import<3
    error('The input data file containing
coordinates of the computational points must have three
columns.',...
        'Error in point type selection');
    error('The input data file containing
coordinates of the computational points must have three columns.')
end
clear rows_Import cols_Import
%Ellipsoidal coordinates
fi=Import(:,1);
lambda=Import(:,2);
h=Import(:,3);
volbadiskcheck=1;
end

%=====
if volbadiskcheck==1
    %Error message for input ellipsoidal
coordinates
    if isempty(fi) || isempty(lambda) ||
isempty(h)
        error('At least one ellipsoidal
coordinate is empty or incorrectly entered.',...
            'Error in point type selection');
        error('At least one ellipsoidal coordinate
is empty or incorrectly entered.');
```

```

        'Error in point type selection');
        error('Values of Longitude min must be
within the interval <-180°,180°> or <0°,360°>');
    end
    fi=pi/180*(fi(:));
    lambda=pi/180*(lambda(:));
    if coord==1 %Entered spherical coordinates
        %Spherical radius
        r=h;
        %Spherical latitude
        fiG=fi;
        %Transform spherical latitude into the
ellipsoidal
        %latitude
        [X,Y,Z]=sph2cart(lambda,fiG,r);
        [fi,lambda_del,h]=ecef2geodetic(X,Y,Z,[aEl
eEl]);
        clear X Y Z lambda_del
    elseif coord==0 %Entered ellipsoidal
coordinates
        %Trasformation of (fi, lambda, h) into (X,
Y, Z)
        [X,Y,Z]=geodetic2ecef(fi,lambda,h,[aEl
eEl]);
        r=sqrt(X.*X+Y.*Y+Z.*Z); %Radius
        %Spherical latitude
        fiG=atan(Z./sqrt(X.*X+Y.*Y));
        clear X Y Z
    end
    %Computation of the coefficients C0,0; C2,0;
...; C20,0
    %of the selected ellipsoid
    CE1=zeros(length(C),1);
    for n=0:10
        CE1(2*n==stupen & rad==0,1)=((-
1)^n*(3*eEl^(2*n))/((2*n+1)*(2*n+3)*sqrt(4*n+1))*(1-n-
5^(3/2)*n*CE1_20/eEl^2)).*(aEl./R).^ (2*n).*(GME1/GM);
    end
    if any(volbapar==11) || any(volbapar==12) ||
any(volbapar==13) || any(volbapar==14) || any(volbapar==15) ||
any(volbapar==16) || any(volbapar==17) || any(volbapar==18) ||
any(volbapar==20) || any(volbapar==25)
        grav=1;
    else
        grav=0;
    end
    if any(volbapar==2) || any(volbapar==3) ||
any(volbapar==4) || any(volbapar==5) || any(volbapar==6) ||
any(volbapar==7) || any(volbapar==8) || any(volbapar==9) ||
any(volbapar==10) || any(volbapar==19) || any(volbapar==21) ||
any(volbapar==22) || any(volbapar==23) || any(volbapar==24)
        por=1;
        deltaC=C-CE1;
    else
        por=0;
    end
    if any(volbapar==20)
        normal=1;
    end

```

```

else
    normal=0;
end
clear GGM stupen rad
if normal==0
    clear CEL
end
%Initialization
eta=0; ksi=0; Teta=0; Tksi=0; T=0; T_rr=0;
Trr=0; Trfi=0; Trl=0; Tfifi=0;
Tfil=0; Tll=0; N=0; V=0; Vrr=0; Vrfi=0; Vrl=0;
Vfifi=0;
Vfil=0; Vll=0; g=0; g_sa=0; W=0;
anomalια_sa=0; porucha=0;
porucha_sa=0; zetaEl=0; zeta=0; Wrr=0; Wr=0;
Wfi=0; Wlambda=0;
Ur=0; Ufi=0; N1c=0; N2c=0; H=0; zeta_N1c=0;
zeta_H=0; zeta_dg=0;
etaH=0; ksiH=0; TetaH=0; TksiH=0; TH=0;
T_rrH=0; TrrH=0; TrfiH=0; TrlH=0; TfifiH=0;
TfilH=0; TllH=0; NH=0; VH=0; VrrH=0; VrfiH=0;
VrlH=0; VfifiH=0;
VfilH=0; VllH=0; gH=0; g_saH=0; WH=0;
anomalια_saH=0; poruchaH=0;
porucha_saH=0; zetaElH=0; zetaH=0; WrrH=0;
WrH=0; WfiH=0; WlambdaH=0;
UrH=0; UfiH=0; N1cH=0; N2cH=0; HH=0;
zeta_N1cH=0; zeta_HH=0; zeta_dgH=0;
Tzz=0; Txx=0; Tyy=0; Txy=0; Txz=0; Tyz=0;
TzzH=0; TxxH=0; TyyH=0; TxyH=0; TxzH=0;
TyzH=0;
Vzz=0; Vxx=0; Vyy=0; Vxy=0; Vxz=0; Vyz=0;
VzzH=0; VxxH=0; VyyH=0; VxyH=0; VxzH=0;
VyzH=0;
%Indices of the spherical harmonic
coefficients
index=zeros(nmax+1,1);
index(1,1)=1;
for i=1:nmax
    index(i+1,1)=index(i,1)+i;
end
%Initialization of the matrices and vectors
for the computation of fnALFs
Pnm=zeros(length(fi),nmax+1);
q=(R./r);
q2=(R./r).^2;
u=cos(fiG);
t=sin(fiG);
%Initialization for extended-range arithmetic
approach
if volbaALFs==3
    bit=mexext; %Bit version of Matlab
    bit=bit(end-1:end);
    bit=str2double(bit);
    if bit==32
        bit=32;
    elseif bit==64
        bit=64;
    end
end

```

```

else
    bit=64;
end

nmax23=nmax*2+3;
rr=zeros(nmax23,1); ri=rr;
dd=zeros(nmax,1); am=dd; bm=am;
if bit==32
    pm=am;
else
    zz=zeros(length(fiG),1);
end
ps1=zeros(length(fiG),nmax); ips1=ps1;
m1=1:nmax23;
rr(m1)=sqrt(m1);
ri(m1)=1./rr;
m2=1:nmax;
dd(m2)=rr(2*m2+3).*ri(2*m2+2);
IND=960;
BIG=2^IND;
BIGI=2^(-IND);
BIGS=2^(IND/2);
BIGSI=2^(-IND/2);
ROOT3=1.732050807568877;
x=ROOT3*u.*q;
ix=zeros(size(x));
ps1(:,1)=x;
ips1(:,1)=ix;
for m3=2:nmax
    x=(dd(m3-1)*u).*x.*q;
    y=abs(x);
    iy=y>=BIGS;
    if any(iy)
        x(iy)=x(iy)*BIGI;
        ix(iy)=ix(iy)+1;
    end
    iy=y<BIGSI;
    if any(iy)
        x(iy)=x(iy)*BIG;
        ix(iy)=ix(iy)-1;
    end
    ps1(:,m3)=x;
    ips1(:,m3)=ix;
end
end
%If geoid/quasigeoid is to be computed
if any(volbapar==10) || any(volbapar==23)
    geoid=1;
    if any(h~=0)
        errordlg('To compute Geoid_undulation
or Height_anomaly, the ellipsoidal height must be set to
zero.',...
                'Error in point type selection');
        error('To compute Geoid_undulation or
Height_anomaly, the ellipsoidal height must be set to zero.');
```

```

        geoid=0;
    end
    %Initialization of the matrices and vectors
for the
    %computation of the first-order derivatives of
fnALFs
    if any(volbapar==3) || any(volbapar==4) ||
any(volbapar==6) || any(volbapar==7) || any(volbapar==12) ||
any(volbapar==13) || any(volbapar==16) || any(volbapar==20)
        dALFs=1;
        dPnm=zeros(length(fi),nmax+1);
        qu=q./u;
        tu=t./u;
        %Treatment of the dPnm singularity
        singdPnm=fi==pi/2 | fi==-pi/2;
    else
        dALFs=0;
    end
    %Initialization of the matrices and vectors
for the
    %computation of the second-order derivatives
of fnALFs
    if any(volbapar==6) || any(volbapar==12)
        ddALFs=1;
        ddPnm=zeros(length(fi),nmax+1);
        %Treatment of the ddPnm singularity
        singddPnm=fi==pi/2 | fi==-pi/2;
    else
        ddALFs=0;
    end
    %Status line
    progressbar=findobj('tag','hlasky');
    %% Summation over m
    for m=nmax:-1:0
        %Update of the progress bar
        if rem(m,10)==0
            set(progressbar,'string',...
                sprintf('Progress: m =
%5.0d',m),...
                'fontsize',8); drawnow;
        end
        %Selection of the spherical harmonic
coefficients of order m
        %=====
        if grav==1 %C's spherical harmonic
coefficients for the functionals without the normal gravity field
            Cm=C(index((m+1):end)+m);
        end
        if por==1 %C's spherical harmonic
coefficients for the functionals with the disturbing field
            deltaCm=deltaC(index((m+1):end)+m);
        end
        if normal==1 %C's spherical harmonic
coefficients for the functionals with the normal field
            if m==0
                CELm=CEl(index((m+1):end)+m);
            end
        end
    end
end

```

```

        if geoid==1
            HCm=HC(index((m+1):end)+m);
            HSm=HS(index((m+1):end)+m);
        end
        Sm=S(index((m+1):end)+m);
%=====
        %% Computation of modified fnALFs
        if volbaALFs==1 %Standard forward column
method
            if m==0
                Pnm(:,1)=1;
            elseif m==1
                Pnm(:,1)=sqrt(3)*u.*q;
            elseif m>1
                i=2*(2:m);
                i1=sqrt((i+ones(size(i)))./i);
Pnm(:,1)=u.^m*sqrt(3)*prod(i1).*q.^m;
            end
            if m==nmax
            elseif m<=(nmax-1)
                n=m+1;
                anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
                Pnm(:,2)=anm*t.*Pnm(:,1).*q;
            end
            if m<(nmax-1)
                j=3;
                for n=m+2:nmax
                    anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
                    bnm=sqrt((2*n+1)*(n+m-1)*(n-m-
1)/((n-m)*(n+m)*(2*n-3)));
                    Pnm(:,j)=anm*t.*Pnm(:,j-1).*q-
bnm*Pnm(:,j-2).*q2;
                    j=j+1;
                end
            end
        elseif volbaALFs==2 %Modified forward
column method
            if m==0
                Pnm(:,1)=1e-280;
            elseif m==1
                Pnm(:,1)=sqrt(3)*q*1e-280;
            elseif m>1
                i=2*(2:m);
                i1=sqrt((i+ones(size(i)))./i);
Pnm(:,1)=sqrt(3)*prod(i1)*(q.^m)*1e-280;
            end
            if m==nmax
            elseif m<=(nmax-1)
                n=m+1;
                anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
                Pnm(:,2)=anm*t.*Pnm(:,1).*q;
            end
            if m<(nmax-1)

```

```

j=3;
for n=m+2:nmax
    anm=sqrt((2*n-1)*(2*n+1)/((n-
m)*(n+m)));
    bnm=sqrt((2*n+1)*(n+m-1)*(n-m-
1)/((n-m)*(n+m)*(2*n-3)));
    Pnm(:,j)=anm*t.*Pnm(:,j-1).*q-
bnm*Pnm(:,j-2).*q2;
    j=j+1;
end
end
elseif volbaALFs==3 %Extended-range
arithmetic
    if bit==32 %32 bit version of Matlab
        am(m+1)=rr(2*m+3);
        for n=m+2:nmax
            w=rr(2*n+1)*ri(n-m)*ri(n+m);
            am(n)=rr(2*n-1)*w;
            bm(n)=rr(n-m-1)*rr(n+m-
1)*ri(2*n-3)*w;
        end
        if m~=0
            for i=1:length(fiG)
                x=ps1(i,m);
                ix=ips1(i,m);
                if(ix==0)
                    pm(m)=x;
                elseif (ix<-1)
                    pm(m)=0;
                elseif (ix<0)
                    pm(m)=x*BIGI;
                else
                    pm(m)=x*BIG;
                end
                if(m>=nmax)
                    Pnm(i,1:(nmax-
m+1))=pm(m:end);
                    continue;
                end
                y=x;
                iy=ix;
                x=(am(m+1)*t(i)*q(i))*y;
                ix=iy;
                w=abs(x);
                if(w>=BIGS)
                    x=x*BIGI;
                    ix=ix+1;
                elseif (w<BIGSI)
                    x=x*BIG;
                    ix=ix-1;
                end
                if(ix==0)
                    pm(m+1)=x;
                elseif (ix<-1)
                    pm(m+1)=0.;
                elseif (ix<0)
                    pm(m+1)=x*BIGI;
                else

```

```

pm(m+1)=x*BIG;
end
for n=m+2:nmax
id=ix-iy;
if(id==0)
zz=(am(n)*t(i)*q(i))*x-bm(n)*q2(i)*y;
iz=ix;
elseif(id==1)
zz=(am(n)*t(i)*q(i))*x-bm(n)*q2(i)*(y*BIGI);
iz=ix;
elseif(id==-1)
zz=(am(n)*t(i)*q(i))*(x*BIGI)-bm(n)*q2(i)*y;
iz=iy;
elseif(id>1)
iz=ix;
else
zz=-bm(n)*q2(i)*y;
iz=iy;
end

w=abs(zz);
if(w>=BIGS)
zz=zz*BIGI;
iz=iz+1;
elseif(w<BIGSI)
zz=zz*BIG;
iz=iz-1;
end
if(iz==0)
pm(n)=zz;
elseif(iz<-1)
pm(n)=0.;
elseif(iz<0)
pm(n)=zz*BIGI;
else
pm(n)=zz*BIG;
end
y=x;
iy=ix;
x=zz;
ix=iz;
end
Pnm(i,1:(nmax-
m+1))=pm(m:end);
end
elseif m==0
Pnm(:,1)=1;
Pnm(:,2)=sqrt(3)*t.*q;
for i=2:nmax
Pnm(:,i+1)=Pnm(:,i).*sqrt((2*i+1)*(2*i-1))./i.*t.*q-q2.*Pnm(:,i-
1).*(i-1).*sqrt(2.*i+1)./(i.*sqrt(2.*i-3));
end
clear rr ri dd am bm pm ps1
ips1 m1 m2 ...
dd ix x y iy w iz z
end

```



```

else %64 bit version of Matlab
am(m+1)=rr(2*m+3);
for n=m+2:nmax
w=rr(2*n+1)*ri(n-m)*ri(n+m);
am(n)=rr(2*n-1)*w;
bm(n)=rr(n-m-1)*rr(n+m-
1)*ri(2*n-3)*w;
end
if m==0 %Zonal modified fnALFs
Pnm(:,1)=1;
Pnm(:,2)=sqrt(3)*t.*q;
for i=2:nmax
Pnm(:,i+1)=Pnm(:,i).*sqrt((2*i+1)*(2*i-1))./i.*t.*q-q2.*Pnm(:,i-
1).*(i-1).*sqrt(2.*i+1)./(i.*sqrt(2.*i-3)));
end
clear rr ri am bm ps1 ips1 m1
m2 dd ...
ix x y iy w iz zz pmx pm0
pmxBIGI ...
pmxBIG w wBIGS wBIGSI pmlx
pm10 ...
pmlxBIGI pmlxBIG id id0
id1 id_1 ...
idv1 idm1 iz0 izm_1 izm0
izv0
elseif m~=0 %Non-zonal modified
x=ps1(:,m);
ix=ips1(:,m);
pmx=ix==0;
pm0=ix<-1;
pmxBIGI=(ix>=-1 & ix<0);
pmxBIG=ix>0;
Pnm(pmx,1)=x(pmx);
Pnm(pm0,1)=0;
Pnm(pmxBIGI,1)=x(pmxBIGI)*BIGI;
Pnm(pmxBIG,1)=x(pmxBIG)*BIG;
if m<nmax
y=x;
iy=ix;
x=(am(m+1).*t).*y.*q;
ix=iy;
w=abs(x);
wBIGS=w>=BIGS;
x(wBIGS)=x(wBIGS)*BIGI;
ix(wBIGS)=ix(wBIGS)+1;
wBIGSI=w<BIGSI;
x(wBIGSI)=x(wBIGSI)*BIG;
ix(wBIGSI)=ix(wBIGSI)-1;
pmlx=ix==0;
pm10=ix<-1;
pmlxBIGI=(ix>=-1 & ix<0);
pmlxBIG=ix>0;
Pnm(pmlx,2)=x(pmlx);
Pnm(pm10,2)=0;
Pnm(pmlxBIGI,2)=x(pmlxBIGI)*BIGI;
Pnm(pmlxBIG,2)=x(pmlxBIG)*BIG;
for n=m+2:nmax

```

```

id=ix-iy;
id0=id==0;
id1=id==1;
id_1=id==-1;
idv1=id>1;
idm1=id<-1;
zz(id0)=(am(n).*t(id0).*q(id0)).*x(id0)-bm(n).*y(id0).*q2(id0);
iz(id0,1)=ix(id0);
zz(id1)=(am(n).*t(id1).*q(id1)).*x(id1)-
bm(n).*y(id1).*q2(id1).*BIGI);
iz(id1)=ix(id1);
zz(id_1)=(am(n).*t(id_1).*q(id_1)).*(x(id_1).*BIGI)-
bm(n).*q2(id_1).*y(id_1);
iz(id_1)=iy(id_1);
zz(idv1)=(am(n).*t(idv1).*q(idv1)).*x(idv1);
iz(idv1)=ix(idv1);
zz(idm1)=-
bm(n).*y(idm1).*q2(idm1);
iz(idm1)=iy(idm1);
w=abs(zz);
wBIGS=w>=BIGS;

zz(wBIGS)=zz(wBIGS)*BIGI;
iz(wBIGS)=iz(wBIGS)+1;
wBIGSI=w<BIGSI;

zz(wBIGSI)=zz(wBIGSI)*BIG;
iz(wBIGSI)=iz(wBIGSI)-
1;

iz0=iz==0;
izm_1=iz<-1;
izm0=(iz>=-1 & iz<0);
izv0=iz>0;
Pnm(iz0,n-m+1)=zz(iz0);
Pnm(izm_1,n-m+1)=0;
Pnm(izm0,n-
m+1)=zz(izm0)*BIGI;
Pnm(izv0,n-
m+1)=zz(izv0)*BIG;

y=x;
iy=ix;
x=zz;
ix=iz;
end
end
end
end
end
%If nmin~=0
=====
if nmin~=0
if LNOFnmin==0
if por==1
deltaCm(1:(nmin-m))=0;
end
if grav==1
Cm(1:(nmin-m))=0;
end
end

```

```

        if geoid==1
            HcM(1:(nmin-m))=0;
            HsM(1:(nmin-m))=0;
        end
        Sm(1:(nmin-m))=0;
    elseif LNOFnmin==1
        Pnm(:,1:(nmin-m))=0;
    end
end
end
=====
%% Computation of the first-order
derivatives of modified fnALFs
    if dALFs==1
        if volbaALFs==1 || volbaALFs==3
            enm=sqrt((2*(m:nmax)+1).*(m:nmax)-m).*(m:nmax)+m)./(2*(m:nmax)-1));
            if m==0 %Zonal modified dALFs
                dPnm(:,1)=0.*u;
                dPnm(:,2)=sqrt(3)*u.*q;
                dPnm(:,3:end)=-
                bsxfun(@times,2:nmax,tu).*Pnm(:,3:end)+(bsxfun(@times,enm(3:end),q
                u)).*Pnm(:,2:(end-1)));
            elseif m==nmax %Sectorial modified
dALFs
                dPnm(:,1)=-m*(tu).*Pnm(:,1);
            else
                dPnm(:,1)=-m*(tu).*Pnm(:,1);
            %Tesseral modified dALFs
                dPnm(:,2:(nmax-m+1))=-
                bsxfun(@times,(m+1):nmax,tu).*Pnm(:,2:(nmax-
                m+1))+bsxfun(@times,enm(2:end),qu).*Pnm(:,1:(nmax-m)); %Tesimalne
                ALFs
            end
            elseif volbaALFs==2
                enm=sqrt((2*(m:nmax)+1).*(m:nmax)-m).*(m:nmax)+m)./(2*(m:nmax)-1));
                if m==0 %Zonal modified dALFs
                    dPnm(:,1)=0.*u*1e-280;
                    dPnm(:,2)=sqrt(3)*u.*q*1e-280;
                    dPnm(:,3:end)=-
                    bsxfun(@times,2:nmax,tu).*Pnm(:,3:end)+(bsxfun(@times,enm(3:end),q
                    u)).*Pnm(:,2:(end-1)));
                elseif m==nmax
                    dPnm(:,1)=-m*(tu).*Pnm(:,1);
                %Sectorial modified dALFs
                else
                    dPnm(:,1)=-m*(tu).*Pnm(:,1);
                %Tesseral modified dALFs
                    dPnm(:,2:(nmax-m+1))=-
                    bsxfun(@times,(m+1):nmax,tu).*Pnm(:,2:(nmax-
                    m+1))+bsxfun(@times,enm(2:end),qu).*Pnm(:,1:(nmax-m)); %Tesimalne
                    dALFs
                end
            end
            %Treatment of the dALFs singularity
            dPnm(singdPnm,:)=0;

```

```

                                if ddALFs==1 %If the second-order
derivatives of the modified fnALFs are to be computed
                                if m==0 %Zonal modified ddALFs
                                    ddPnm=bsxfun(@times,tu,dPnm)-
bsxfun(@times,(0:nmax).*((0:nmax)+1),Pnm);
                                else
                                    ddPnm(:,1:end-
m)=bsxfun(@times,tu,dPnm(:,1:end-
m))+bsxfun(@times,m^2./u.^2,Pnm(:,1:end-m))-
bsxfun(@times,(m:nmax).*((m:nmax)+1),Pnm(:,1:end-m));
                                end
                                %Treatment of the ddALFs
singularity
                                    ddPnm(singddPnm,:)=0;
                                end
                                end
                                cosla=cos(m*lambda);
                                sinla=sin(m*lambda);
                                if any(volbapar==8) || any(volbapar==9) ||
any(volbapar==14) || any(volbapar==15)
                                    if m<2
                                        coslaplus2=cos((m+2)*lambda);
                                        sinlaplus2=sin((m+2)*lambda);
                                        coslaminus2=cos(0*lambda);
                                        sinlaminus2=sin(0*lambda);
                                    elseif m>nmax-2
                                        coslaplus2=cos(0*lambda);
                                        sinlaplus2=sin(0*lambda);
                                        coslaminus2=cos((m-2)*lambda);
                                        sinlaminus2=sin((m-2)*lambda);
                                    else
                                        coslaplus2=cos((m+2)*lambda);
                                        sinlaplus2=sin((m+2)*lambda);
                                        coslaminus2=cos((m-2)*lambda);
                                        sinlaminus2=sin((m-2)*lambda);
                                    end
                                end
                                if any(volbapar==9) || any(volbapar==15)
                                    if m<1
                                        coslaplus1=cos((m+1)*lambda);
                                        sinlaplus1=sin((m+1)*lambda);
                                        coslaminus1=cos(0*lambda);
                                        sinlaminus1=sin(0*lambda);
                                    elseif m>nmax-1
                                        coslaplus1=cos(0*lambda);
                                        sinlaplus1=sin(0*lambda);
                                        coslaminus1=cos((m-1)*lambda);
                                        sinlaminus1=sin((m-1)*lambda);
                                    else
                                        coslaplus1=cos((m+1)*lambda);
                                        sinlaplus1=sin((m+1)*lambda);
                                        coslaminus1=cos((m-1)*lambda);
                                        sinlaminus1=sin((m-1)*lambda);
                                    end
                                end
                                end
end

```

BIODATA PENULIS



Jamhur lahir di Bandung pada tanggal 23 April 1981. Mulai menempuh pendidikan Sarjana Teknik S1 jurusan Teknik Geofisika, Fakultas Ilmu Kebumihan dan Teknologi Mineral di Institut Teknologi Bandung pada tahun 2000. Pada tahun 2008 bekerja di salah satu perusahaan di Jakarta sebagai *mud logger* selama setahun. Kemudian pada tahun 2009 menjalankan perusahaan warnet/*game online* selama 1,5 tahun. Tahun 2011 menjalankan perusahaan penggilingan plastik selama 1,5 tahun. Tahun 2013 sampai sekarang mengajar di salah satu bimbingan belajar di Surabaya. Tahun 2015 diterima di program magister S2 Institut Teknologi Sepuluh Nopember dan lulus pada Juli 2017.