

Übungsaufgaben zur Potenzialtheorie

WiSe 2012/2013

Janina Kammann
Marius kriegewski
Moritz Nieschlag
Doreen Kasper

3. November 2012

Inhaltsverzeichnis

1	Aufgabenteil	1
2	Aufgabenteil 2	2
3	Aufgabe 2	4
A	Quellcode	8
A.1	Aufgabe 1	8
A.2	Aufgabe 2	10
A.3	Aufgabe 3	12
A.3.1	Hauptprogramm	12
A.3.2	Subprogramm: Berechnen der Kreispunkte	14
A.3.3	Subprogramm: Berechnen der Kugel	14

Abbildungsverzeichnis

1	Legendre-Polynom P_9^0 - zonal	1
2	Legendre-Polynom P_6^6 - sektoral	1
3	Legendre-Polynom P_9^3 - tesseral	2
4	Das Schwerepotential für nmax=5 - sehr geringe Auflösung	2
5	Das Schwerepotential für nmax=200 - geringe Anomalien sichtbar	3
6	Approximierte Kugel aus 100 Schichten und 100-eckigen Polygonen	4
7	analytische Lösung für das Schwerefeld einer Kugel	6
8	Schwerefeld einer durch 100 Schichten 100-eckeniger Polygonen approximierten Kugel	7
9	Differenz zwischen analytischer Lösung und Näherung bei x=0 und y=-15:15	8

1 Aufgabenteil

Im Rahmen der Masterveranstaltung *GP-M-POTTHEO* wurden uns die mathematisch-physikalischen Grundlagen der Potentialtheorie vermittelt, welche nun in den folgenden Übungen an ausgewählten geophysikalischen Beispielen veranschaulicht werden. Die hierfür verwendeten Programme sowie graphischen Darstellungen wurden mittels der Software Matlab erstellt und sind dem anschließenden Anhang dieser Arbeit zu entnehmen. Die von uns verwendeten Gleichungen zur Aufgabenbearbeitung basieren auf dem Vorlesungeskript oder in der Vorlesung bereitgestellter Literatur.

Im ersten Aufgabenteil des Übungsblattes sollen drei Sätze Legendre Polynome als Funktion der geographischen Länge und Breite berechnet und programmiert werden. Die jeweiligen Polynome unterscheiden sich durch verschiedene Ordnungen n und Grade m , wobei für diese folgende Bedingung gilt: $m \leq n$.

Die in den anschließenden Darstellungen berechneten Polynome definieren sich aus folgender Beziehung:

$$(A_{n,m} \cos m\Phi + B_{n,m} \sin m\Phi) * P_n^m(\theta) \quad (1)$$

Laut Aufgabenstellung gilt für die Koeffizienten der Zusammenhang: $A_{n,m} = B_{n,m} = 1$. Da die Legendre-Polynome auf der Verwendung beliebig orthogonaler Funktionen basieren haben wir für die Normalisierung der zonalen Kugelfunktionen eine Schmidt-Normalisierung mittels der Matlabfunktion 'sch' durchgeführt, sodass aus der Basis des aufgespannten Vektorraums eine Orthonormalbasis konstruiert wird. Durch die Normalisierung wird des Weiteren eine Wichtung der Koeffizienten in einem Intervall von $[-1, 1]$ vorgenommen. Die nachfolgenden Darstellungen zeigen die Ergebnisse der programmierten Legendre-Polynome. Der entsprechend dokumentierte Programmcode ist dem Anhang A zu entnehmen.

Die erste Graphik veranschaulicht eine zonale Darstellung der Legendre-Polynome. In diesem Fall ist die Ordnung des Polynoms stets durch $m=0$ definiert und der Grad des Polynoms variierbar (hier $n=9$). Da Polynom P_9^0 definiert sich über 9 Nullstellen und ist unabhängig von den Längengraden (θ) (da $m=0$). Verschiedene Beispiele zeigen, dass durch Erhöhung der Koeffizienten m und n die beschriebenen Gebiete der Funktion kleiner 'gefächert' sind. Die zweite Abbildung für P_6^6 ist sektoriell, das heißt der Grad des Polynoms entspricht seiner Ordnung sodass gilt: $m=n=6$. In diesem Fall ist das Polynom unabhängig von dem Breitengrad und durch 12 Nullstellen definiert. Die dritte Graphik veranschaulicht die allgemeine Kugelflächenfunktion P_9^3 ; eine tesserale Darstellung der unterschiedlichen Koeffizienten.

Abbildung 1: Legendre-Polynom P_9^0 - zonal

Abbildung 2: Legendre-Polynom P_6^6 - sektoral

2 Aufgabenteil 2

In dieser Aufgabe soll das Schwerefeld der Erde, welches beispielsweise aus Satellitenmessungen bekannt ist mittels der in Aufgabe 1 berechneten Legendre Polynome dargestellt werden. Die dazugehörigen Datensätze zur Berechnung notwendiger Parameter haben wir der Institution des Helmholtz Centre Potsdam (GFZ German Research Centre For Geosciences) aus folgender Internetseite entnommen: <http://icgem.gfz-potsdam.de/ICGEM>.

Für die Aufgabenbearbeitung wurde das Datenfile *osu89a.gfc* verwendet, welches die Koeffizienten der Kugelfunktionsentwicklung bis zu verschiedenen Graden und Ordnungen mit dazugehörigen Standardabweichungen enthält. Das Einlesen des Files erfolgte via `load('osu.txt')`. Für die Berechnung des Potentials ist zu beachten, dass die Koeffizienten der Polynome bis zum 10. Grad korrigiert werden (siehe Aufgabenblatt). Aufgrund der linearen Approximation des Potentials wird durch die Korrektur der Koeffizienten erreicht, dass die ersten Terme aus der Reihenentwicklung eine stärkere Gewichtung erhalten, sodass ellipsoide Anteile (bzgl. des Referenzellipsoids) im Potentialfeld nicht überwiegen und auch sehr geringe Anomalien zu erkennen sind. Das Programm zur Berechnung des Schwerepotentials ist folgend strukturiert:

- Einlesen des Datenfiles
- Deklaration der Variablen
- Anordnung der Koeffizienteneinträge in neue Matrizen C_{nm} und S_{nm} mit dazugehörigem Grad und Ordnung
- Korrektur der Koeffizienten
- Schleife über θ und Φ
- Berechnung der Legendre Polynome (+ Schmidt-Normalisierung)
- Berechnung des Potentials über Schleife (m,n) und Summenbildung
- graphische Darstellung des Schwerepotentials

Eine detaillierte Beschreibung ist dem Programmcode im Anhang zu entnehmen. Die Berechnung des Potentials der Erde basiert auf folgendem Zusammenhang:

$$U_g = \frac{\gamma M}{r} \sum_{n=0}^{\infty} \left(\frac{a}{r}\right)^n \sum_{m=0}^n P_{mn}(\theta) [C_{nm} \cos m\Phi + S_{nm} \sin m\Phi] \quad (2)$$

Die zwei folgenden Abbildungen zeigen, dass variierende Ordnungen (n) der Legendre Polynome unterschiedliche Ergebnisse für das Schwerepotential liefern. Demnach bestimmt die Höhe der Ordnungen n und damit die Anzahl berechneter Koeffizienten das Auflösungsvermögen der zu ermittelnden Größe. Bei der Berechnung des Potentials mit n=200 werden auch kleinste Anomalien sichtbar, während für n=5 kleinskalige Undulationen nicht berücksichtigt werden. Dies ist auf die kurzwelligen Anteile der Koeffizienten zurückzuführen, welche durch eine hohe Ordnung der Kugelflächenfunktionen sichtbar beziehungsweise für Koeffizienten $n \leq 5$ unterdrückt werden.

Abbildung 4: Das Schwerepotential für nmax=5 - sehr geringe Auflösung

Abbildung 5: Das Schwerepotential für $n_{\max}=200$ - geringe Anomalien sichtbar

In den anschließenden Abbildungen sind drei Ausschnitte dargestellt die besonders starke Schwereanomalien aufweisen. Die Ursache für die langwelligen Anomalien sind auf großräumige Dichteveriationen im Erdmantel beziehungsweise in der Erdkruste zurückzuführen. Eine höhere Gesteinsdichte erzeugt demnach eine zusätzliche Gravitationsbeschleunigung, wodurch der Massenüberschuss eine positive Schwereanomalie bewirkt und das Geoid "ausbeult". Andererseits werden durch geringere Dichten negative Schwereanomalien verursacht die Eindellungen "des Geoids hervorrufen. Die Dichteveriationen im Erdmantel werden durch die Geodynamik der Mantelkonvektion begründet. Demnach weist extrem heißes Mantelmaterial eine geringe Dichte auf und steigt nach oben, wogegen sich kalte Regionen durch eine hohe Dichte ausweisen und absinken. Folglich sind abtauchende Konvektionsströme ursächlich für positive Schwereanomalien (Beulen) und aufsteigende Konvektionsströme verantwortlich für negative Anomalien (Dellen). Auch die Topographie führt zu lateral variablen Massenvariationen und führt zu Schwereanomalien (siehe Abb.?? Himalaya) bla bla beispiele:

3 Aufgabe 2

Im dritten Aufgabenteil der Übung geht es darum das berechnete Schwerfeld einer approximierten Kugel mit der analytischen Lösung zu vergleichen. Hierbei beziehen wir uns auf das Paper *gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections* von Plouff (1976), in dem die Schwerewirkung eines n-eckigen Polygons berechnet wird.

Die Kugel (mit $R = 2m$ bei $[0\ 0\ 4m]$) in unserem Modell soll durch k Schichten n-eckiger Polygone angenähert werden und den Dichteunterschied $\delta\rho = 200 \frac{kg}{m^3}$ zu der Umgebung aufweisen. In Abbildung 3 ist eine durch 100 Schichten von 100-eckigen Polygonen approximierte Kugel zu erkennen. Optisch ist bei einer Abtastung durch viele Polygonecken und Schichten kein Unterschied zu einer Kugel zu erkennen. Es gilt nun festzustellen, wie stark das Schwerfeld von der Anzahl der Schichten und Polygonecken abhängt.

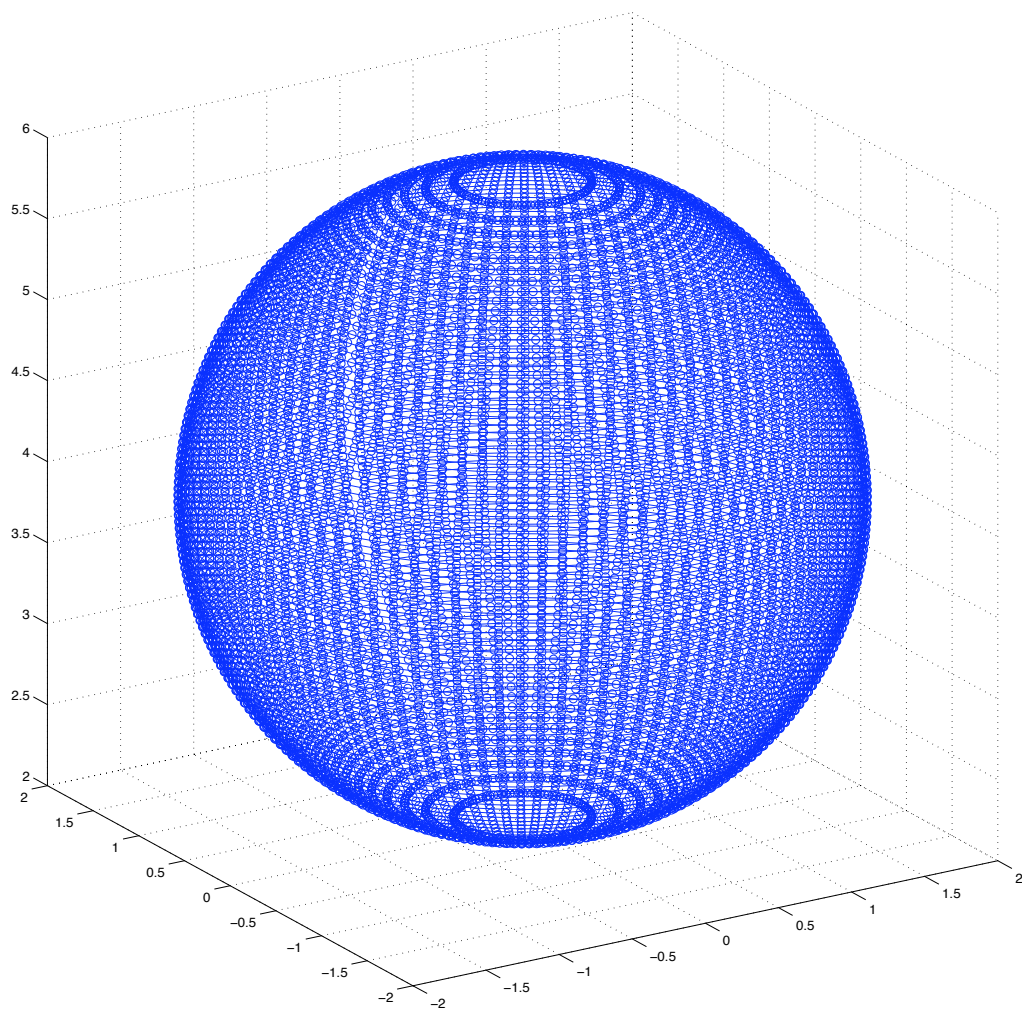


Abbildung 6: Approximierte Kugel aus 100 Schichten und 100-eckigen Polygonen

Um die Schwere zu berechnen wird die Schwerebeschleunigung aller Kanten und aller Polygone aufsummiert. Dafür wird die von Plouff (1976) eingeführte Formel verwendet (3) verwendet.

$$g = \gamma \rho s_m \sum_{i=1}^n \left\{ s_p A(z_1 - z_2) + z_2 \left(\tan^{-1} \left(\frac{z_2 d_1}{P R_{12}} \right) - \tan^{-1} \left(\frac{z_2 d_2}{P R_{22}} \right) \right) \right. \\ \left. - z_1 \left(\tan^{-1} \left(\frac{z_1 d_1}{P R_{11}} \right) - \tan^{-1} \left(\frac{z_1 d_2}{P R_{21}} \right) \right) - P \cdot \ln \left(\frac{R_{22} + d_2}{R_{12} + d_1} \frac{R_{11} + d_1}{R_{21} + d_2} \right) \right\} \quad (3)$$

Die analytische Lösung ergibt sich aus der Schwerebeschleunigung einer Punktmasse M

$$g = -\frac{\gamma \cdot M}{r^2} \quad (4)$$

mit

$$M = -\frac{4}{3} \pi R^3 d\rho \quad (5)$$

Abbildung 3 zeigt die analytische Lösung für das Schwerfeld der Kugel. Die Form und Größenordnung stimmt weitestgehend mit der approximierten Lösung in Abbildung 3 überein. Bei genauerer Betrachtung fällt jedoch auf, dass die Schwerebeschleunigung bei zunehmendem Abstand zum Mittelpunkt bei der analytischen Lösung wesentlich schneller abflacht. Dieser Effekt rührt vermutlich daher, dass bei der analytischen Lösung von einer Punktmasse in der Mitte der Kugel ausgegangen wird. Der Abstand zur Kugel geht demnach als Abstand zum Mittelpunkt der Kugel ein.

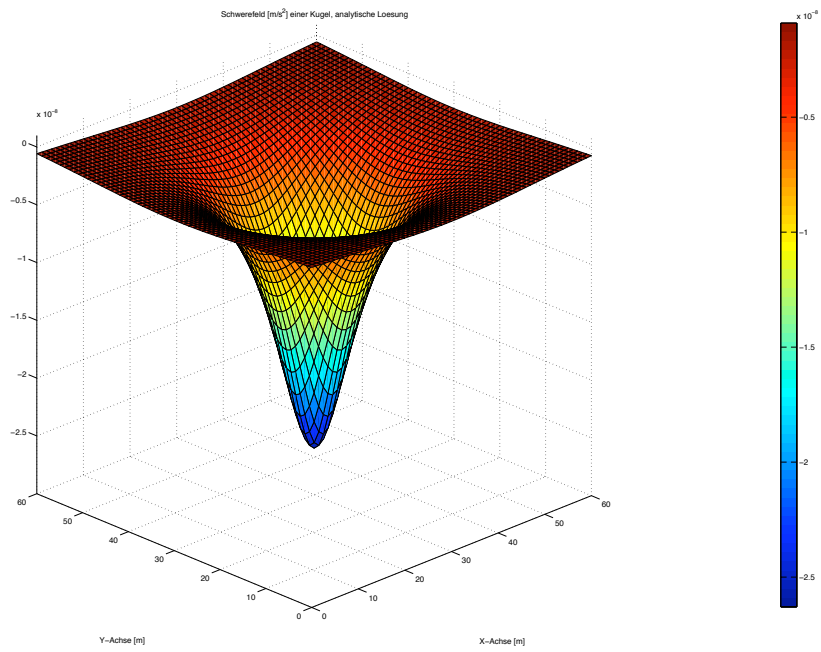


Abbildung 7: analytische Lösung für das Schwerfeld einer Kugel

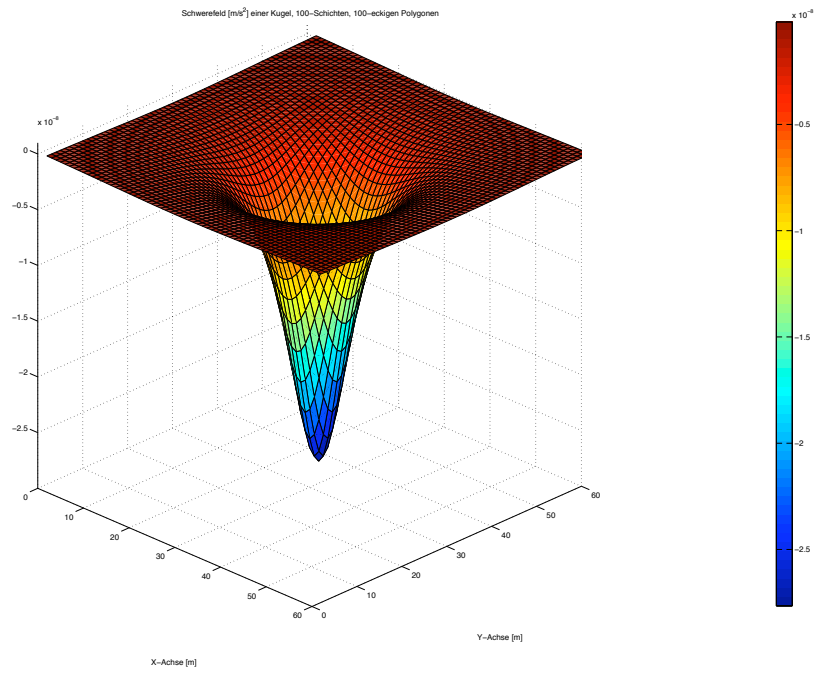


Abbildung 8: Schwerefeld einer durch 100 Schichten 100-eckiger Polygonen approximierten Kugel

bla bla

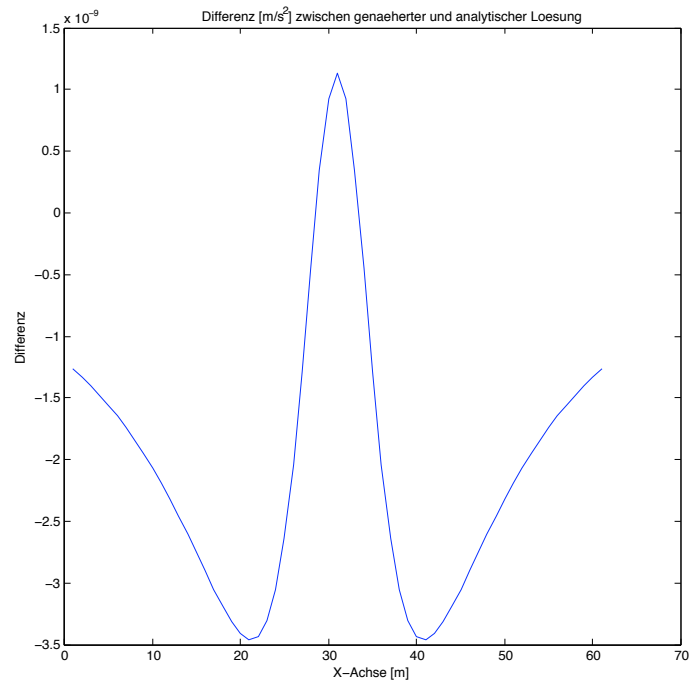


Abbildung 9: Differenz zwischen analytischer Lösung und Näherung bei $x=0$ und $y=-15:15$

A Quellcode

A.1 Aufgabe 1

```
%clear all
close all

Grad = 3;
Ordnung = 3;

delta = pi/200;
theta = 0 : delta : pi;
phi = 0 : 2*delta : 2*pi;
[phi,theta] = meshgrid(phi,theta);

% Legendre ausrechnen
Ymm = legendre(Grad,cos(theta(:,1)),'sch');

%Polynome gesuchter Ordnung extrahieren:
Ymm = Ymm(Ordnung+1,:);

%Grid erstellen:
yy = Ymm;
for kk = 2: size(theta,1)
    yy = [yy Ymm];
end;
```

```

%Werte  $f\tilde{A}_{\frac{1}{4}}r$  phi ausrechnen:
yy = yy.*cos(Ordnung*phi);
K=ones(length(yy));
[X,Y,Z]=sphere(length(yy))
surf(X,Y,Z,yy)

```

A.2 Aufgabe 2

```

clear all
close all

%% Einlesen des Datenfiles
filename = 'osu89a-mod.gfc';
[A,delimiterOut]=importdata(filename)

gamma = 6.67384E-11;% Gravitationskonstante
M = 5.974E24;        % Erdmasse
r = 6371000.;        % mittlerer Erdradius
a = 149.6E6;         % grosse Halbachse
V = gamma*M/r;       % Vorfaktor

delta = pi/100;
theta = 0 : delta : pi;    % Breite
phi = 0 : 2*delta : 2*pi;   % Laenge

for line_i=1:length(A.data)
    Cnm(A.data(line_i,1)+1,A.data(line_i,2)+1)=A.data(line_i,3);
    Snm(A.data(line_i,1)+1,A.data(line_i,2)+1)=A.data(line_i,4);
end

%Korrekturen
Cnm(3,1) = Cnm(3,1) + 0.108262982131 * 10^(-2)/sqrt(5);
Cnm(5,1) = Cnm(5,1) - 0.237091120053 * 10^(-5)/sqrt(9);
Cnm(7,1) = Cnm(7,1) + 0.608346498882 * 10^(-8)/sqrt(13);
Cnm(9,1) = Cnm(9,1) - 0.142681087920 * 10^(-10)/sqrt(17);
Cnm(11,1) = Cnm(11,1) + 0.121439275882 * 10^(-13)/sqrt(21);

n=100;
yy=zeros(length(theta));
xy=zeros(length(theta));
asdf(1:length(phi),1:length(phi))=0;

for n_i=0:n
    test=n_i
    Pnm = legendre(n_i,cos(theta),'sch');
    for phi_i=1:length(phi)
        parfor m_i=0:n_i

            tmp(m_i+1,:) = Pnm(m_i+1,:).*(Cnm(n_i+1,m_i+1)*cos(m_i*phi(phi_i))+...
            Snm(n_i+1,m_i+1)*sin(m_i*phi(phi_i)));

        end
        asdf(phi_i,:)=sum(tmp);
        xy(phi_i,:)=xy(phi_i,:)+ asdf(phi_i,:);
    end
    qwer(:, :) = xy(:, :)*((a/r)^n_i);
    yy(:, :)=yy(:, :)+qwer(:, :);
end
yy(:, :)=yy(:, :)*V;
yy=yy';

```

```
[phiplot ,thetaplot]=meshgrid(phi ,theta );  
contourf(theta ,phi , yy, 100, 'linestyle ','None')  
colorbar()
```

A.3 Aufgabe 3

A.3.1 Hauptprogramm

```
%% POTENZIALTHEORIE
% Marius Kriegerowski, Moritz Nieschlag, Doreen Kasper und Janina Kammann

clear all
close all
%matlabpool open
%% Deklarationen

drho = 200;
Sm = 1;
gamma = 6.67384E-11; % Gravitationskonstante
ne = 100; % Anzahl der Ecken
ns = 100; % Anzahl der Schichten
PU = [0 0 4]; % Mittelpunkt der Kugel
R = 2; % Radius

Xrange = -15:0.5:15;
Yrange = Xrange;

k=1;

%for ne=5:5:200

    ns=ne;
    dz = 2*R/(ns-1); % Schichtdicke

    [X,Y] = meshgrid(Xrange, Yrange);
    V = gamma * drho * Sm;

    XYZ = zeros(length(Xrange), length(Yrange));

    % Erzeugung der Kugel:
    B=kugel(R,ns,ne,PU);

    for x_i=1:length(Xrange)
        asd=x_i
        for y_i=1:length(Yrange)

            gout=0;
            g=0;

            parfor ns_i=1:ns-2

                for w_i=1:ne

                    x1=B(ns_i,1,w_i)-Xrange(x_i);
                    y1=B(ns_i,2,w_i)-Yrange(y_i);
                    z1=B(ns_i,3,w_i);
```

```

x2=B( ns_i ,1,mod( w_i ,ne)+1)-Xrange( x_i );
y2=B( ns_i ,2,mod( w_i ,ne)+1)-Yrange( y_i );
%z2=B( ns_i +1,3,mod( w_i ,ne)+1);

g=graviPunkt( x1 ,x2 ,y1 ,y2 ,z1 ,dz );

gout=gout+g;

end

end

XYZ( x_i , y_i ) = gout*V;
g_analyt( x_i , y_i )=-4/3*pi*gamma*drho*(R-dz)^3*1/...
((PU(1)-Xrange( x_i ))^2+(PU(2)-Yrange( y_i ))^2+PU(3)^2);

gdiff( x_i , y_i )=g_analyt( x_i , y_i )-XYZ( x_i , y_i );

end
end

diff(k,1)=gdiff(30,30);
diff(k,2)=ns;

k=k+1

%matlabpool close

figure(1)
contourf(XYZ)
xlabel('X-Achse [m]')
ylabel('Y-Achse [m]')
title('Schwerefeld [m/s^2] einer Kugel, 100-Schichten, 100-eckigen Polygonen')
colorbar
axis('square')

figure(2)
contourf(g_analyt)
xlabel('X-Achse [m]')
ylabel('Y-Achse [m]')
title('Schwerefeld [m/s^2] einer Kugel, analytische Loesung')
colorbar
axis('square')

figure(3)
plot(gdiff(30,:))
xlabel('X-Achse [m]')
ylabel('Differenz ')
%ylabel('Y-Achse [m]')
title('Differenz [m/s^2] zwischen genaehelter und analytischer Loesung')
colorbar
axis('square')

```



```

figure(4)
surf(XYZ)
xlabel('X-Achse [m]')
ylabel('Y-Achse [m]')
title('Schwerefeld [m/s^2] einer Kugel, 100-Schichten, 100-eckigen Polygonen')
colorbar
axis([-30 30 -30 30 -3E-8 0])

```

```

figure(5)
surf(g_analyt)
xlabel('X-Achse [m]')
ylabel('Y-Achse [m]')
title('Schwerefeld [m/s^2] einer Kugel, analytische Loesung')
colorbar
axis('square')

```

```

figure(6)
plot(diff(:,2), diff(:,1))
xlabel('Anzahl der Schichten/Ecken')
ylabel('Differenz Analytische und Gerechnete Loesung')
title('Fehlerentwicklung in Abhaengigkeit von der Anzahl der Ecken/Schichten')

```

A.3.2 Subprogramm: Berechnen der Kreispunkte

```

function [P] = kreis(n,PU,r)

P=zeros(3,n);
gamma=2*pi/n;

for n_i=1:n

    P(1,n_i) = r*sin(n_i*gamma)+PU(1);
    P(2,n_i) = r*cos(n_i*gamma)+PU(2);
    P(3,n_i) = PU(3);

end

```

A.3.3 Subprogramm: Berechnen der Kugel

```

function [A]=kugel(R, ns, ne, PM)

dz =2*R/(ns-1);

for ns_i=1:ns-1

```

```

PU(1)=PM(1);
PU(2)=PM(2);
PU(3)=PM(3)+R-ns_i*dz;

r=sqrt(R^2-(R-ns_i*dz)^2);

A(ns_i, :, :) = kreis(ne,PU,r);
end

```