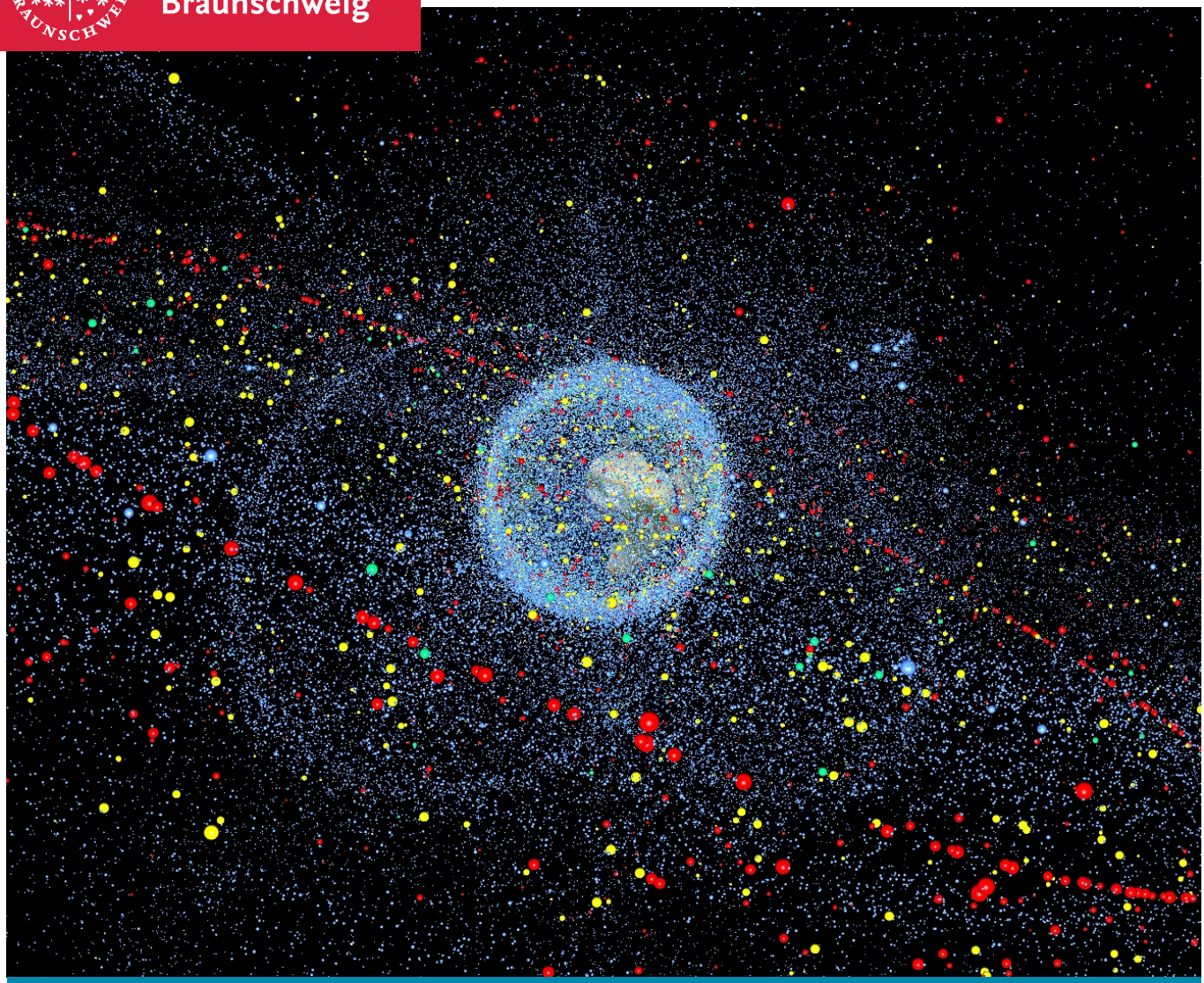
Technische Universität Braunschweig

Institut für Raumfahrtsysteme



R XXXX X (beim Betreuer beantragen!) Titel der Arbeit

Institut für Raumfahrtsysteme

Max Mustermann

22. Dezember 2017

Aufgabenstellung

Die Originalaufgabenstellung ist bei Studienarbeiten dem ungebundenen Institutsexemplar beizufügen, bei Bachelor-, Master- und Diplomarbeiten dem gebundenen Exemplar zur Vorlage bei der Fakultät. Die Aufgabenstellung bei Bachelor-, Master- und Diplomarbeiten wird vom Fachbereich ausgegeben (bei CSE-Masterarbeit vom CSE Office), dieser registriert den Beginn und die Abgabe der Arbeit und stempelt diese Angaben auf das letzte Blatt der Original-Aufgabenstellung.

Eine Diplom-, Studien-, Bachelor- bzw. Masterarbeit soll zeigen, dass man in der Lage ist, in begrenzter Frist eine Aufgabe nach wissenschaftlichen Methoden selbständig zu bearbeiten.

Die Aufgabenstellung kann Literaturhinweise enthalten, die als Einstieg in die Aufgabe gedacht sind. Es wird erwartet, daß weitere Literatur selbständig gesammelt wird (Bibliotheken der TU, des Instituts, etc.).

Wichtig: Schriftverkehr mit Dritten bei Nennung des die Arbeit betreuenden Instituts bedarf der vorherigen Genehmigung.

In der Abgabeversion dann dieses Blatt entfernen und an dieser Stelle durch die Aufgabenstellung ersetzen!

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die nachfolgende Arbeit selbständig und nur unter Zuhilfenahme der angegebenen Literatur angefertigt habe.

_____

Datum, Unterschrift

Übersicht

Die Übersicht enthält kurz gefasste Angaben über die Zielsetzung, die angewandten Methoden und die gewonnenen Ergebnisse. Sie soll das Wesentliche aus dem Inhalt der Arbeit in wenigen Sätzen zusammenfassen und ist der eigentlichen Arbeit voranzustellen (höchstens 1/2 bis 1 Seite). Sie soll also nicht lediglich die Aufgabenstellung wiedergeben.

Inhaltsverzeichnis

1 Einleitung

Die Einleitung soll einen Überblick über den Stand der Technik geben, das zu unter- suchende System beschreiben und die Aufgabenstellung mit eigenen Worten näher erläutern.

1.1 Ziele der Arbeit

1.2 Vorgehensweise

Hier wiedergeben, wie die zuvor definierten Ziele in den folgenden Kapiteln umgesetzt werden, etwa:

In Kapitel **??** werden die theoretischen Grundlagen behandelt, wonach dann in Kapitel **??** die Software-Architektur festgelegt und beschrieben wird.

2 Theoretische Grundlagen

2.1 Literaturrecherche

Zu jeder guten Arbeit gehört eine umfangreiche Literaturrecherche. Während in den ersten Gesprächen mit dem Betreuer bereits einige Tipps zu empfohlener Literatur erhalten werden, ist es in den meisten Fällen notwendig, selbst nach Literatur zu suchen. Dazu bietet sich zunächst die UB an, mit der man als Student bereits vertraut sein sollte. Darüber hinaus ist folgendes Vorgehen zu empfehlen:

- Suche nach Literatur, auf die in den Referenzen bereits erhaltener Literatur verwiesen wird.
- Suche bei *Science Direct*, wobei zahlreiche Paper aus dem Uni-Netzwerk kostenlos heruntergeladen werden können: `http://www.sciencedirect.com/`
- Suche in digitaler Bibliothek des ILR

Für den letzteren Punkt wird ein Account am ILR benötigt, den man beim Betreuer beantragen kann. Die Nutzung der digitalen Bibliothek funktioniert folgendermaßen:

1. Kopiere die Datei **JabRef-2.10.jar** z.B. auf deinen Desktop (oder unter Linux **JabRef** nach */bin/*.). Hier ggf. den Betreuer nach dem aktuellen Pfad der Datei fragen.
2. Durch anklicken kann JabRef jetzt gestartet werden (oder unter Linux durch den Befehl **JabRef** aus der Konsole)
3. Oben links befindet sich der Button zum Öffnen: *Öffnen einer BibTex-Database*
4. Öffne die Datei */home/export/biblio/e-biblio.bib* (unter Windows ggf. das Laufwerk *biblio an ILR-Server (Jupiter)* einbinden)
5. Die elektronische Bibliothek ist geöffnet (diese wird beim nächsten Starten von JabRef automatisch wieder geladen)
6. Zugriff auf die Literatur:
   - Elektronische Literatur lässt sich direkt aus JabRef öffnen
   - Literatur, die nur als Buch oder Kopiervorlage existiert, kann über das Feld *Note* gefunden werden, in dem die entsprechende Referenznummer abgelegt ist. Hier am besten den Betreuer fragen, wo im Institut sich das Dokument dann befindet (Archiv).

2.2 Bewertung der Arbeit

Alle Arbeiten am ILR werden nach einem vorgegebenen Schema bewertet, so dass vorkommen kann, dass Studenten, die eine perfekte Software als Lösung des Problems liefern, oder etwa eine perfekte

Konstruktionslösung bieten, dennoch Abzüge in der Note erhalten können, wenn etwa die Dokumentation Schwächen zeigt (äußere Form, *roter Faden* in der Arbeit, Unvollständigkeit, etc.).

Die Beurteilungskriterien sind in der folgenden Tabelle aufgelistet:

Inhalt (Einführung, Theorie, Lösungsweg, Ergebnisse, Diskussion)	50%
Äußere Form der Arbeit	15%
Arbeitsweise (Eigene Ideen, Selbständigkeit, Methodik)	30%
Zeiteinteilung	5%

2.3 Formelzeichen

Ein Symbol- und Abkürzungsverzeichnis sind, neben den bereits automatisch durch diese Vorlage erstellten Tabellen- und Abbildungsverzeichnissen, zu erstellen. Dabei wird empfohlen, von dem Paket **glossaries** Gebrauch zu machen, welches ebenfalls in dieser Vorlage eingebunden ist. Hier ein paar Beispiele:

- Ein Symbol, welches im Symbolverzeichnis auftauchen soll: J_2
- Eine Abkürzung wird beim ersten Mal ausgeschrieben: Institut für Luft- und Raumfahrtsysteme (ILR) und ab dem zweiten Mal nicht mehr: ILR

In Gleichungen verwendet man die Formelzeichen analog, wie an Gleichung **??** zu sehen ist:

$$E = m \cdot c^2 \tag{2.1}$$

Zunächst werden die als Formelzeichen benutzten lateinischen bzw. deutschen Buchstaben in alphabetischer Reihenfolge geordnet, wobei jeweils die großen Buchstaben den kleinen voranzustellen sind. Dann in gleicher Reihenfolge die dem griechischen Alphabet entnommenen Symbole. Am Schluss sind die häufig benutzten Indizes in alphabetischer Reihenfolge anzugeben. Beinhaltet die Arbeit die Erstellung eines Programms oder von Teilen hierzu, so sollen die im Programm verwendeten Bezeichnungen den Formelzeichen zugeordnet werden. Es sind ausschließlich SI-Einheiten zu verwenden.

Die Sortierung kann ebenfalls automatisch vorgenommen werden, indem im Symbolverzeichnis (Datei: *symbolverzeichnis.tex*) jedem Eintrag der entsprechende **sort**-Schlüssel gegeben wird, z.B.:

- Das Symbol **c** soll im Symbolverzeichnis auftauchen. Es handelt sich um einen lateinischen Buchstaben, der Schlüssel zum Sortieren wird mit *sort=xc* vergeben (Präfix *x*, der gleich ersichtlich wird - s.u.).
- Das Symbol α ist ein griechischer Buchstabe und soll erst nach den lateinischen Buchstaben kommen. Entsprechend stellt man den Schlüssel auf *sort=yalpha* (Präfix *y*, sodass griechische Buchstaben immer nach den lateinischen Buchstaben erscheinen werden.

Existieren benötigte Herleitungen bereits in der Literatur, so ist auf diese zu verweisen. Nur zum unmittelbaren Verständnis notwendige Gleichungen sollten dargestellt werden. Es ist darauf zu achten, dass sonstige theoretische Herleitungen schlüssig sind (kontrollieren, ob jede Größe eindeutig definiert ist und keine Gedankensprünge vorhanden sind). Herleitungen soweit wie möglich mit allgemeinen Bezeichnungen durchführen, Zahlenwerte erst im Ergebnisteil (ab Kapitel **??**) einsetzen.

3 Hauptteil

Hier werden die zur Lösung der gestellten Aufgaben erforderlichen Arbeiten ausführlich beschrieben. Je nach Aufgabenstellung sind benutzte Theorien und Rechenmethoden zu erläutern, Versuchseinrichtungen zu beschreiben, Festigkeitsnachweise zu führen, Konstruktionsbeschreibungen anzufertigen usw. Der Text soll es dem Leser ermöglichen, die Arbeit inhaltlich zu verstehen, ohne zusätzliche Spezialliteratur zu benötigen, wobei davon ausgegangen wird, dass der Leser einen Bachelor-Abschluss mit einigen Grundlagenkenntnissen zur Luft- und Raumfahrt besitzt. Allgemein zugänglicher bzw. bekannter Lehrbuch- oder Vorlesungsstoff soll daher nicht ausführlich wiederholt, sondern nur, soweit zum Verständnis unbedingt nötig, kurz zusammengefasst und zitiert werden.

Es ist darauf zu achten, dass theoretische Herleitungen schlüssig sind (kontrollieren, ob jede Größe eindeutig definiert ist; keine Gedankensprünge). Theoretische Herleitungen mit allgemeinen Bezeichnungen. Zahlenwerte erst im Ergebnisteil. Die im Rahmen der Arbeit gefundenen Ergebnisse sind in Kapitel **??** ausführlich zu diskutieren. Evtl. auftretende Abweichungen zwischen Theorie und Messung bzw. zwischen verschiedenen Rechenverfahren sind zu deuten. Besonderes Gewicht ist auf die physikalische Interpretation mathematischer bzw. experimenteller Ergebnisse zu legen. Computerprogramme sind bezüglich ihrer Ein- und Ausgabedateien sorgfältig zu dokumentieren. Unterroutinen sollen mit Zweck und Ein- und Ausgabeparametern dokumentiert werden. Ein Struktogramm, das den logischen Programmablauf darstellt, ist zu empfehlen. Der Quellcode der entwickelten Computerprogramme ist in einem textuellen Dateiformat (ASCII) zusammen mit der Arbeit einzureichen.

Für den Hauptteil sind weiterhin folgende Punkte zu beachten:

- Eigennamen sollen in Großbuchstaben geschrieben werden, z. B. EULER-Winkel. Verweise auf Bilder und Tabellen sind entsprechend dieser Vorlage zu erstellen, z.B.: Abbildung **??**, Tabelle **??**.
- Der Raum für Text und Bilder soll möglichst gut ausgenutzt werden, so soll nicht etwa eine Seite nur eine Abbildung in der Mitte enthalten und oben und unten große Leerräume bleiben.
- Nur neue Kapitel (Einleitung, Theoretische Grundlagen, etc.) beginnen auf neuen Seiten.
- Formeln werden automatisch nummeriert, ein Verweis ist folgendermaßen möglich: Gleichung **??**
- Zu jeder Abbildung und jeder Tabelle gehört eine Bildunterschrift (Abbildung) bzw. -überschrift (Tabelle).
- In Abbildungen sollen die Achsen mit der entsprechenden Größe und der Dimension bezeichnet werden.

- Zahlen entsprechend der ISO-31 mit Tausender-Trennzeichen (Leerzeichen) schreiben: 10 000 und nicht 10000

Abbildung 3.1: Ein Bild von Kepler.

In Tabellen ist darauf zu achten, dass möglichst wenig vertikale Striche verwendet werden, wie im Beispiel in Tabelle **??** gezeigt.

Tabelle 3.1: Bahntypen als Funktion der Exzentrizität.

Wert der Exzentrizität	**Bahntyp**
$\epsilon = 0$	Kreis
$0 < \epsilon < 1$	Ellipse
$\epsilon = 1$	Parabel
$\epsilon > 1$	Hyperbel

3.1 Referenzieren und Zitieren

Das richtige Referenzieren und Zitieren ist Grundvoraussetzung für eine erfolgreiche Arbeit. Die TU Braunschweig setzt seit dem 15.07.2014 routinemäßig die Plagiatserkennungssoftware **docoloc** ein, um alle Arbeiten von Studenten auf Plagiarismus zu prüfen.

Es ist also mit äußerster Sorgfalt bei der Verwendung von Textstellen, Herleitungen, Bildern und Daten aus der Literatur bzw. externen Quellen vorzugehen.

Für jede Wiedergabe aus der Literatur (z.B. Zitat) folgt die Referenz direkt im Anschluss (nicht erst am Ende des Satzes!). Eine Autor-Jahr-Notation ist empfohlen, welche mit folgendem Beispiel einfach in dieser Vorlage umgesetzt werden kann:

- Ein Zitat mit Klammern: (**?**)
- Ein Zitat ohne Klammern: **?**

Literaturangaben werden durch **bibtex** (Datei: *literatur.bib*) übernommen und müssen enthalten:

- Bei Artikeln aus Zeitschriften: Verfassernamen, Titel des Artikels, Name der Zeitschrift, Bandnummer, Erscheinungsjahr, Nummer des Heftes, Anfangs- und Schlussseite des Artikels
- Bei Büchern: Verfassername, Buchtitel, Bandnummer, Auflage, Verlagsort, Verlag und Erscheinungsjahr
- Bei Internet-Seiten: URL sowie Zugriffsdaten
- Informationen, die man etwa aus persönlichen Gesprächen erhalten hat, lassen sich ebenfalls eintragen, z.B.: *Max Mustermann, persönliches Gespräch, Datum*

3.2 Programmieren

Basierend auf einer Top-Down-Analyse des vorgegebenen Problems können zunehmend verfeinerte Flussdiagramme erstellt werden, die bereits eine Programmstruktur implizieren und die Übersicht bei komplexen Programmen erhöhen.

Programme sollten allgemein so geschrieben werden, daß ein Interessierter deren Funktion und deren Ablauf in groben Schritten ohne Dokumentation nachvollziehen kann (selbstdokumentierend). Hierzu dienen Kommentare im Quelltext, eine optische Gliederung des Programmtextes und die weitestgehende Verwendung von strukturierter Programmierung. Ein Anwender sollte vom Programm geleitet und über den Ablauf informiert werden. Benutzereingaben sollten möglichst vom Programm auf Plausibilität gecheckt werden.

Im Folgenden ist ein Code-Beispiel (Fortran) gezeigt, welches die typischen Elemente jedes Haupt- und Unterprogramms zeigt, wobei Anmerkungen darin zwischen "<<" und ">>" gefasst und damit nicht Teil des Quellcodes bzw. der Kommentare des Codes sind. Es handelt sich bei dem Kopfteil um eine Struktur, die es ermöglicht, eine automatische Dokumentation mittels der Software **Doxygen** zu erstellen:

```
!------------------------------------------------------------------------
!
!> @anchor      initGravityPotential  << Doxygen-Kommentare beginnen in Fortran
!!                                    << mit !!, der erste jedoch mit !>. Ein
!!                                    << @anchor stellt später einen Link auf
!!                                    << diese Funktion zur Verfügung >>
!!
!! @brief       Initialization...     << Kurzbeschreibung der Funktion >>
!! @author      Max Mustermann        << Name des Autors >>
!!
!! @date        <ul>                  << @date erlaubt eine Revisionshistorie
!!                                    << zu führen >>
!!               <li> 02.10.2012 (initial design)    </li>
!!               <li> 31.05.2013 (code optimization) </li>
!!               <li> 19.08.2013 (added ...)         </li>
!!             </ul>
!!
!! @param[in]   cpath       Path to...       << Beschreibung der Inputgrößen
```

```
!! @param[in]   imodel      Model to be...
!! @param[out]  cout        Output string... << 'out' für Outputgrößen
!!
!! @details     This routine initializes the... << Detaillierte Funktions-
!!               ....                          << beschreibung, in die z.B. auch
!!                                             << auch die verwendeten Quellen
!!                                             << bzw. Literaturangaben gehören.
!!
!!-------------------------------------------------------------------------
subroutine initGravityPotential(cpath,imodel,cout)

  implicit none              << In Fortran immer gut, damit keine Variablen
                             << implizit deklariert sind, z.B. wäre dann
                             << ein 'a' automatisch ein integer


  << DEKLARATIONSTEIL>>


  !** interface              << Zuerst die Schnittstellenvariablen
  !----------------------------------------
  character(len=*), intent(in)  :: cpath
  integer,          intent(in)  :: imodel
  !----------------------------------------

  !** local                  << Dann die lokalen Variablen
  !-------------------------------------------------
  character(len=255)           :: cbuf      ! character buffer
  character(len=*), parameter  :: csubid = "initGravityPotential"
  ...

  integer :: i                   ! loop counter
  integer :: ich                 ! input channel
  integer :: ierr                ! error flag
  ...

  real(dp)  :: fac               ! multiplication factor
  ...
  !-------------------------------------------------------

  << Nun beginnt der eigentliche PROGRAMMTEXT >>

  coeffInitialized = .false.    ! as a new initialization is started...
```

```
!=======================================================================
!
! Decide on which model to use (default: EIGEN-GL04C)
!
!-------------------------------------------------------

<< Verwendung von logischen 'Blöcken', um Lesbarkeit zu erhöhen...>>

!** check imodel validity
if(imodel == EGM96 .or. imodel == EGM08) then
  nmodel = imodel  << Einrückung erhöhen die Lesbarkeit! >>
else
  nmodel = EIGEN_GL04C
end if

!=======================================================================
!
!   Read earth radius and...
!
!-----------------------------------------------------------------------

flag_mu   = .false.
flag_rekm = .false.

do i = 1,imax

  read(ich,'(a)',iostat=ios) cbuf

  !** earth gravity constant
  if(index(cbuf, "earth_gravity_constant") /= 0) then

    read(cbuf,*) ctemp, mu
    ...

  end if

  ...

end do

...
```

Während die obige Darstellung bereits eine gute Möglichkeit darstellt, um Ausschnitte von Quellcode auch in der eigenen Arbeit zwecks Beschreibung wiederzugeben, gibt es auch weitere Pakete, die etwa auch Syntax-Highlighting unterstützen. Eines davon ist z.B. **minted**.

3.2.1 Unterunterkapitel

4 Ergebnisse

4.1 Grafiken und Text

Im Ergebnis-Teil sollen die Ergebnisse vorgestellt werden, für das Beispiel einer Software-Entwicklung also die mit der entwickelten Software erzielbaren Ergebnisse. Im Falle einer Simulationsentwicklung könnte man hier verschiedene Simulationsszenarien definieren und unter verschiedenen Randbedingungen vorstellen.

Es ist dabei darauf zu achten, dass ein ausgewogenes Verhältnis von Bild und Text besteht. Alle in den Plots vorkommenden Linien sind zu erklären. Jeder Plot benötigt eine vollständige Achsenbeschriftung und ggf. eine Legende. Eine Grafik muss immer selbsterklärend sein, d.h. man muss mit Grafik und Bildunterschrift allein in der Lage sein, das Gezeigte zu verstehen.

Auch bei den Ergebnissen soll weiterhin auf den roten Faden geachtet werden. Der Leser soll Stärken und Schwächen des Tools / der Simulation / der Konstruktion kennenlernen.

Um Plots zu erstellen, lässt sich ebenfalls TikZ nutzen (wie auch für WBS, Zeitplan, etc.). Der Vorteil ist, dass alle Beschriftungen einer Grafik auch dieselbe Schrift und -größe tragen, wie auch der Haupttext. Darüber hinaus sind TikZ-Grafiken auch Vektorgrafiken, sodass Qualitätsverluste nicht zu erwarten sind, wie das etwa bei PNG oder JPG der Fall wäre.

Beispiel für das Plotten von zwei verschiedenen Kurven aus Datenfiles, welches das Paket **pgfplots** nutzt (siehe auch sehr ausführliche Dokumentation mit zahlreichen Beispielen online!):

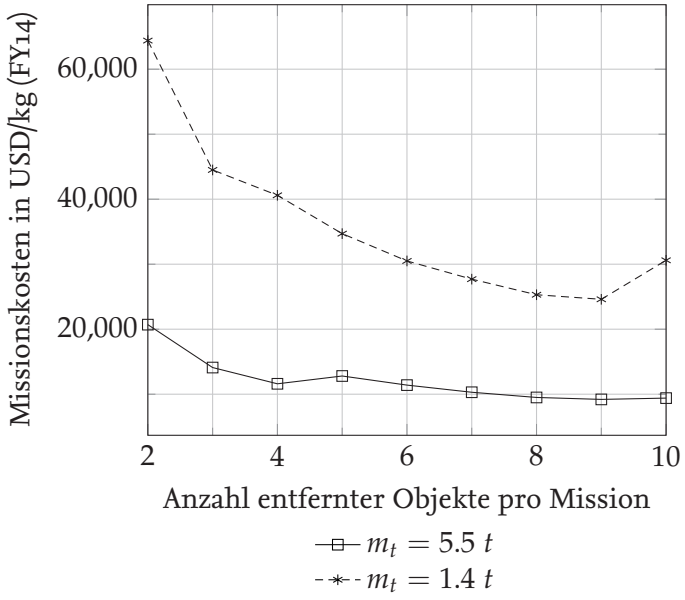


Abbildung 4.1: Vergleich der Kosten pro kg entfernten Schrotts für hohe und mittlere Objektmassen.

Auch das Plotten von Funktionen ist sehr einfach, wie Abbildung **??** für das simple Beispiel einer quadratischen Funktion zeigt.

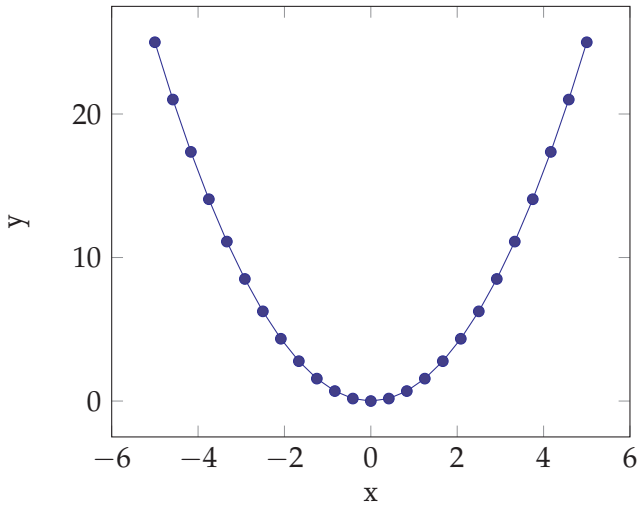


Abbildung 4.2: Die Funktion $y = x^2$.

Sollte nicht TikZ verwendet werden, ist auch **Gnuplot** zu empfehlen. Allerdings ist dann darauf zu achten, dass sämtliche Beschriftungen gut lesbar sind und auch die Grafiken in der entsprechenden Auflösung, ohne Artefakte, etc. im Dokument erscheinen.

4.2 Erstellen dieses Dokuments

Um das vorliegende Dokument zu erstellen, muss die Datei *vorlage.tex* mit **pdflatex** kompiliert werden. Alle üblichen Entwicklungsumgebungen für Latex stellen diese Funktion zur Verfügung (z.B. Kile unter Linux, MikTeX unter Windows, oder TeXShop auf dem Mac).

Dazu wird die Dokumentenklasse **tubsreprt** benötigt, welche das Corporate Design der TU Braunschweig umsetzt. Diese findet man, samt Installationsanleitung unter `http://www.enricojoerns.de/tubslatex/` (aktueller Release: 1.0.3).

Wichtig, damit auch das Symbol- und Abkürzungverzeichnis funktionieren, ist die Ausführung des Befehls **makeglossaries** ebenfalls auf die Datei *vorlage.tex* angewandt. Die Erstellung des Dokuments verläuft also in mehreren Schritten:

- pdflatex auf *vorlage.tex* (erzeugt die für makeglossaries benötigten Dateien!)
- makeglossaries auf *vorlage.tex* (erstellt das Abkürzungs- und Symbolverzeichnis)
- pdflatex auf *vorlage.tex* (ggf. mehrmals!) erstellt nun das Dokument mit allen Referenzen

Der Befehl makeglossaries muss in der Regel manuell in der Entwicklungsumgebung eingestellt werden, was für die verschiedenen Typen unterschiedlich ist. Daher am besten Google befragen (z.B. “Kile makeglossaries”).

5 Zusammenfassung

In der Zusammenfassung (mindestens 1,5 Seiten) sollen die theoretische Herleitung und die wesentlichen Ergebnisse so aufgelistet werden, dass sie ohne Kenntnis der vorherigen Abhandlung verständlich sind. Dabei wird in der Vergangenheit geschrieben und die wichtigsten Ergebnisse der Arbeit wiedergegeben.

6 Fazit und Ausblick

Ein kurzer Ausblick (max. ca. 1 Seite) kann dazu dienen, bei Bearbeitung der gestellten Aufgabe entstandene neue Fragestellungen für zukünftige Untersuchungen zu nennen.

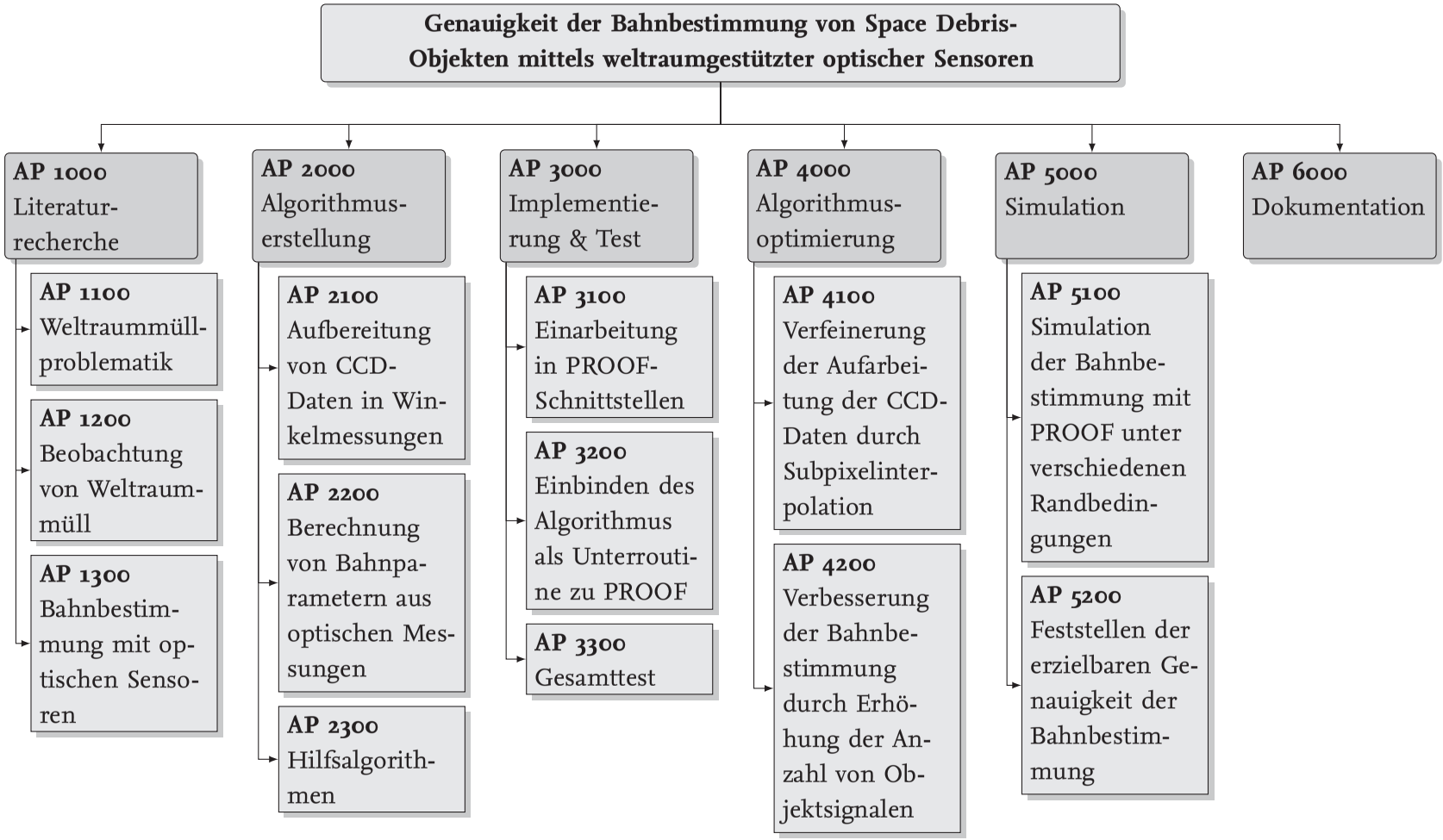
Abbildungsverzeichnis

Tabellenverzeichnis

Symbolverzeichnis

A Projektmanagement

A.1 Work Breakdown Structure


Genauigkeit der Bahnbestimmung von Space Debris-Objekten mittels weltraumgestützter optischer Sensoren
AP 1000 Literatur-recherche
AP 1100 Weltraummüll-problematik
AP 1200 Beobachtung von Weltraum-müll
AP 1300 Bahnbestim-mung mit op-tischen Senso-ren
AP 2000 Algorithmus-erstellung
AP 2100 Aufbereitung von CCD-Daten in Win-kelmessungen
AP 2200 Berechnung von Bahnpa-rametern aus optischen Mes-sungen
AP 2300 Hilfsalgorith-men
AP 3000 Implementie-rung & Test
AP 3100 Einarbeitung in PROOF-Schnittstellen
AP 3200 Einbinden des Algorithmus als Unterroutine zu PROOF
AP 3300 Gesamttest
AP 4000 Algorithmus-optimierung
AP 4100 Verfeinerung der Aufarbei-tung der CCD-Daten durch Subpixelinter-polation
AP 4200 Verbesserung der Bahnbe-stimmung durch Erhö-hung der An-zahl von Ob-jektsignalen
AP 5000 Simulation
AP 5100 Simulation der Bahnbe-stimmung mit PROOF unter verschiedenen Randbedin-gungen
AP 5200 Feststellen der erzielbaren Ge-nauigkeit der Bahnbestim-mung
AP 6000 Dokumentation

A.2 Zeitplan

AP 1000: Literaturrecherche

AP 1100: Weltraummüllproblematik

AP 1200: Beobachtung von Weltraummüll

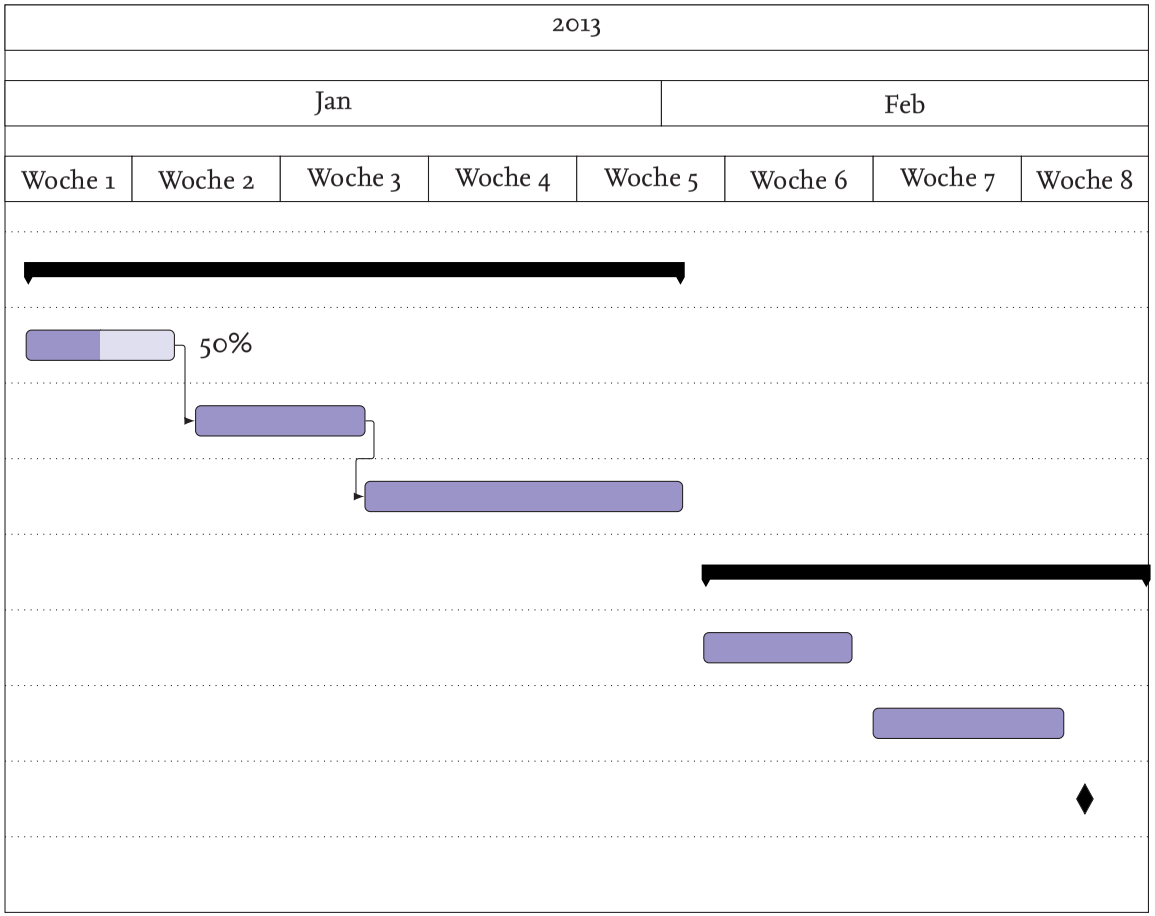
AP 1300: Bahnbestimmung mit optischen Sensoren

AP 2000: Algorithmuserstellung

AP 2100: ...

AP 2200: ...

Meilenstein



A.3 Work Package Description

		AP 1100
Titel	**Genauigkeit der Bahnbestimmung von Space Debris-Objekten mittels weltraumgestützter optischer Sensoren**	**Seite:** 1 von 1
Verantwortlicher	Sebastian Stabroth	**Version:** 1.0
		Datum: 25.06.2003
Beginn	T_0	
Ende	T_0+1 Woche	**Dauer:** 1 Woche
Bearbeiter	Sebastian Stabroth	

Ziele:
• Kenntnis über die Weltraummüllumgebung, Bahnbereiche von Space Debris-Populationen, Objektanzahlen und -größen

Input:
• Literatur zum Thema Weltraummüll

Schnittstellen zu anderen APs:
• **AP 5100** zur Simulation der Bahnbestimmung von Weltraummüll

Aufgaben:
• Einlesen in die Thematik Weltraummüll

Ergebnisse:
• Verständnis der Weltraummüllproblematik

		AP 1200
Titel	**Titel des Arbeitspakets**	**Seite:** X von Y
Verantwortlicher	Dein Name	**Version:** 1.1
		Datum: DD.MM.YYYY
Beginn	T_0	
Ende	T_0+X Wochen	**Dauer**: X Wochen
Bearbeiter	Dein Name	

Ziele:
- Ziel 1
- Ziel 2
- ...

Input:
- Input 1
- ...

Schnittstellen zu anderen APs:
- **AP XXXX** Beschreibung
- **AP ....** ...

Aufgaben:
- Aufgabe 1
- ...

Ergebnisse:
- Ergebnis 1
- ...