

## 1 Vorwort

Einer der großen Vorteile des Textsatzes mit  $\text{\LaTeX}$  ist das Vermeiden von Fehlern und der verringerte Aufwand beim Aktualisieren von Dokumenten. Durch die automatisierte Zählung von Abbildungen, Tabellen und Kapiteln wird der menschliche Fehler aus den entsprechenden Prozessen eliminiert und es müssen keine weiteren Änderungen vorgenommen werden, wenn die Reihenfolge von Kapiteln vertauscht wird. Diese Vorteile erwachsen aus der Trennung von Formatierung und Inhalt, sodass der Nutzer im Idealfall nur den neuen Inhalt liefern muss und keinerlei weiteren Formatierungsaufwand hat. Dieser Gedanke soll diesem letzten Termin zugrunde liegen, sodass die folgenden Prozesse in der Anzahl ihrer Schritte und dem ihrem Formatierungsaufwand optimiert werden:

- Erstellen und Einfügen von Grafiken
- Einbinden von Messdaten in Tabellenform oder als Plot
- Anpassen von Kalenderdaten

## 2 Anpassbarkeit von Kalenderdaten

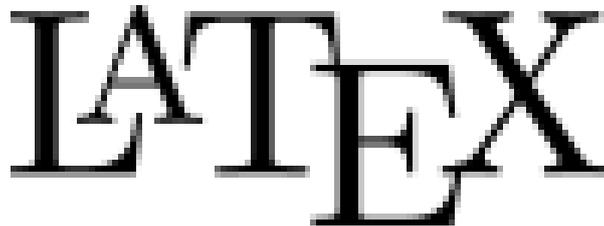
Wenn Dokumente mit Informationen über geplante oder vergangene Veranstaltungen oder Texte mit vielen häufig vertretenen Daten und Uhrzeiten (Geschichtswissenschaftliche Texte, Protokolle, Missionspläne,...) verfasst werden, kann es von Vorteil sein, diese Daten nur einmal eingeben zu müssen. Bei einer Wiederverwendung des Dokumentes im kommenden Jahr muss so an nur einer Stelle ein Datum geändert werden und Tippfehler werden vermieden.

Für diese Zwecke existiert das Paket `datetime2`. Es ersetzt das `datetime`-Paket und bringt Unterstützung für diverse Spracheinstellungen mit um Kalenderdaten und Uhrzeiten den spracheigenen Konventionen entsprechend darzustellen.

In dieser Datei sind die Methoden um Daten und Uhrzeiten für den späteren Gebrauch zu speichern in selbsterstellte Befehle eingefasst, die in der Präambel definiert werden.

## 3 Grafiken in $\text{\LaTeX}$

Das „Portable Graphics Format“ PGF ist ein Format für eine Vektorgrafik-Beschreibungssprache. Sie ermöglicht es, wie Dokumente mit  $\text{\LaTeX}$  beschrieben werden, Vektorgrafiken mit PGF zu beschreiben. Vektorgrafiken sind ein Gegenstück zu Rastergrafiken, die man im allgemeinen als digitale Bilder kennt. Während Rastergrafiken für ein Raster aus Pixeln einzeln Farben definieren, beschreiben Vektorgrafiken das Aussehen der Grafik mithilfe von



(a) Rastergrafik(PNG) -  $100 \times 37$  pixel - 1,83kb



(b) Rastergrafik(PNG) -  $200 \times 74$  pixel - 3,83kb



(c) Rastergrafik(PNG) -  $300 \times 112$  pixel - 5,79kb



(d) Vektorgrafik(PDF) - 2,89kb

Vektoren und mathematisch beschreibbaren Kurven. Als Resultat sind Vektorgrafiken kleinere Dateien und können beliebig vergrößert werden ohne jemals unscharf zu werden. Die Abbildungen 1a bis 1c zeigen dieselbe Grafik in drei unterschiedlichen Auflösungen, während Abbildung 1d die passende Vektorgrafik zeigt. Für Fotos sind Vektorgrafiken ineffizient, weil hier für jeden Pixel anstelle einer Farbe vier Eckpunkte eines Quadrates und eine Farbe gespeichert werden müssten. Für alle Arten von Diagrammen, Logos, Funktionsplots und dergleichen sind Vektorgrafiken das Mittel der Wahl.

Bedauerlicherweise ist PGF sehr umständlich zu benutzen, weswegen das  $\text{\LaTeX}$ -Paket `tikz` (auch TikZ für „TikZ ist kein Zeichenprogramm“) existiert. TikZ ermöglicht es, im  $\text{\LaTeX}$ -Dokument Vektorgrafiken durch Code zu

beschreiben. Die Vorstellungskraft liefert hier die Grenzen des Machbaren und es wegen der Vielfältigkeit möglicher Abbildungen ist hier eine Erläuterung der Funktionsweise noch weniger angebracht als bei  $\text{\LaTeX}$ . Unter <http://www.texample.net/tikz/examples/all/> sind hunderte Beispielfrafiken mit TikZ-Code aufgeführt, von denen man sich für seine eigenen Schaubilder inspirieren lassen kann. Ein kleines Beispiel finden Sie in Abschnitt 5. Eine Auswahl einfacher Befehle und Grundlagen liefern diese Folien von einem Lehrstuhl der Uni Graz: [Link](#).

## 4 Einbinden von Messdaten

Gerade in den Ingenieurwissenschaften sind Forschungsergebnisse häufig die Interpretationen von Messdaten, die dem Leser kunstfertig präsentiert werden wollen. Während der häufig langwierigen Produktionsphasen dieser Forschungsarbeiten müssen Messdaten und Ihre Repräsentationen im Dokument unter Umständen mehrfach aktualisiert werden und es wäre von Vorteil, wenn damit nicht jedes Mal ein Eintippen von  $\text{\LaTeX}$ -Tabellen oder ein Import in z.B. MATLAB, eine Durchführung eines Plotting-Skriptes, der Export als Grafik und das Einbinden in das Dokument verbunden wären.

### 4.1 Tabellen

Tabelle 1 führt die Messdaten aus `myData.csv` auf und enthält in den Parametern für den `\csvreader`-Aufruf einen Filter, der die Tabelle auf die ersten 5 Zeilen beschränkt. Eine deutlich längere Tabelle würde viel Platz einnehmen, bei Seitenumbrüchen Darstellungsfehler erzeugen und dem Leser keinerlei verwertbare Information vermitteln. Für diese Darstellung wurde das `simplecsv`-Paket genutzt und sie erlaubtes, die Datei `myData.csv` einfach durch eine neue zu ersetzen und dann das Dokument neu zu kompilieren um die Tabelle zu aktualisieren.

Ein Kursteilnehmer hat mich neulich auch auf ein Plugin für Microsoft-Excel hingewiesen, das  $\text{\LaTeX}$ -Code für existierende Tabellen erzeugt, was zwar ein Ersetzen von  $\text{\LaTeX}$ -Code erfordert, für einige Anwender aber vielleicht unkomplizierter ist. Einen Link zu einem entsprechenden Tutorial-Video finden Sie im Kurs.

### 4.2 Graph-Plots

Um dem Leser nicht einfach nur Daten hinzuwerfen und sich selbst zu überlassen, bietet es sich an, die Daten so aufzubereiten, dass der Bezug zum umgebenden Text deutlicher wird und die Darstellungsform Argumentation und Sachverständnis unterstützt. Unsere 629 Messpunkte lassen sich bedeutend besser verarbeiten, wenn man sie platzsparend in einem Graph-Plot darstellt. Abbildung 2 zeigt alle Daten aus `myData.csv` in einem Linienplot

Zeitpunkt [s]	Messwert [ $\frac{l}{min}$ ]
0	0.09754
0.01	0.2885
0.02	0.56688
0.03	0.9875
0.04	1.0049

Tabelle 1: Die Messdaten aus `myData.csv`

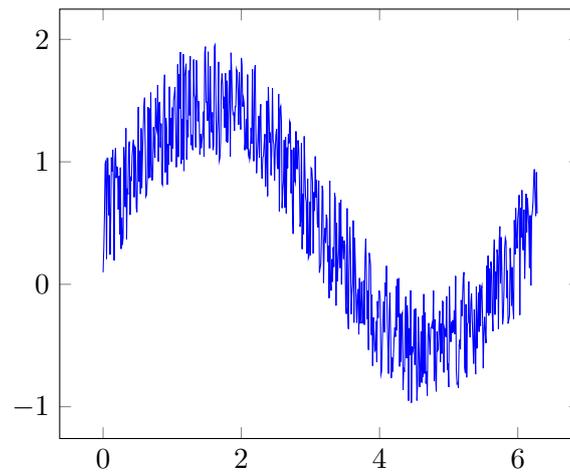


Abbildung 2: Die Messdaten aus `myData.csv`

auf ähnlich wenig Platz, wie die ersten 5 Zeilen als Tabelle eingenommen haben und ist bedeutend weniger ätzend anzuschauen als 629 Zeilen Tabelle.

Diese Beispiel demonstriert nur die einfachste Methode, Daten mit PGFplots zu plotten. Weitere Beispiele finden Sie unter <http://pgfplots.sourceforge.net/gallery.html>.

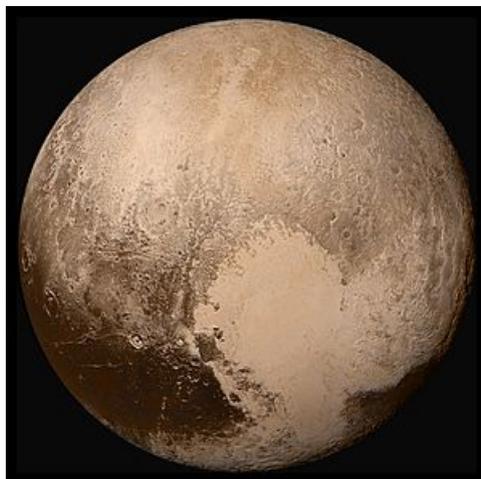


Abbildung 3: Eine „Nahaufnahme“ von Kuipergürtel-Objekt „(134340) Pluto“ vom 14. Juli 2015.

## 5 Ein Beispieltext mit Kalenderdaten

Die Raumsonde „New Horizons“ startete am 19. Januar 2006 um 19:00 Uhr von Cape Canaveral an Bord einer Atlas V Trägerrakete. Ihre Primärmission, den Kuipergürtel und im speziellen das Kuipergürtel-Objekt „(134340) Pluto“ zu untersuchen, war am 25. Oktober 2016 beendet. Auf ihrem Weg stellte sie am 13. Juni 2006 Beobachtungen des Asteroiden „(132524) APL“ an, bevor sie die ihre Flugbahn durch einen Vorbeiflug an Jupiter am 28. Januar 2007 korrigierte und am 14. Februar 2015 die Beobachtung des Pluto. „New Horizons“ erreichte die nächste Annäherung an Pluto am 14. Juli 2015 und sendete bis zum 25. Oktober 2016 die gesammelten Daten zurück zur Erde. Unter diesen Daten befand sich auch das bisher beste Bild des ehemaligen Planeten, das in Abbildung 3 zu sehen ist. Nach dem Vorbeiflug an Pluto wurden die Triebwerke von „New Horizons“ gezündet um mehr als drei Jahre später, am 1. Januar 2019 am damals inoffiziell „Ultima Thule“ genannten Kuipergürtelobjekt „Arrokoth“ vorbeizufiegen und das in Abbildung 5 gezeigte Farbbild zu machen. 2035 wird „New Horizons“ den Terminationsshock passieren und in die Heliohülle eintreten.

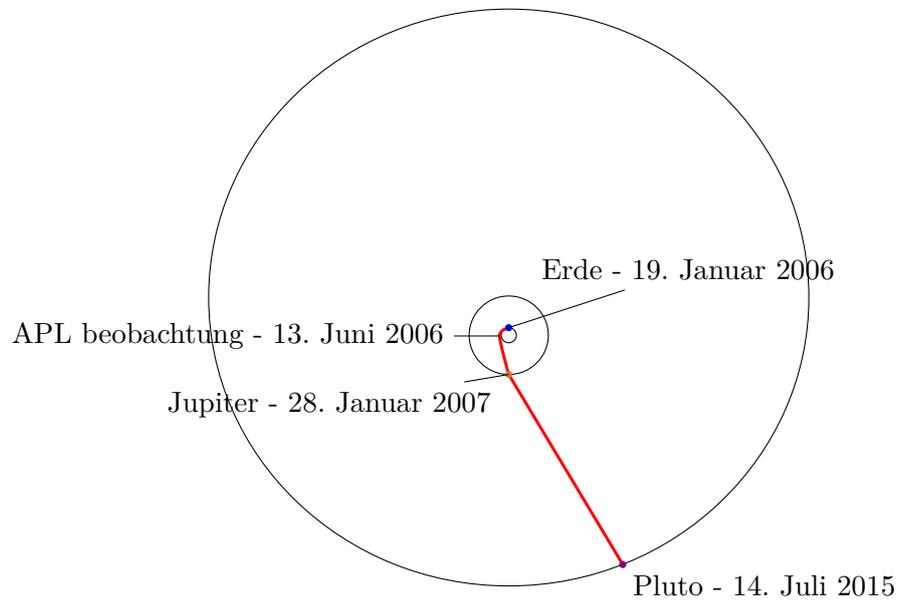


Abbildung 4: „New Horizons“ Primärmission

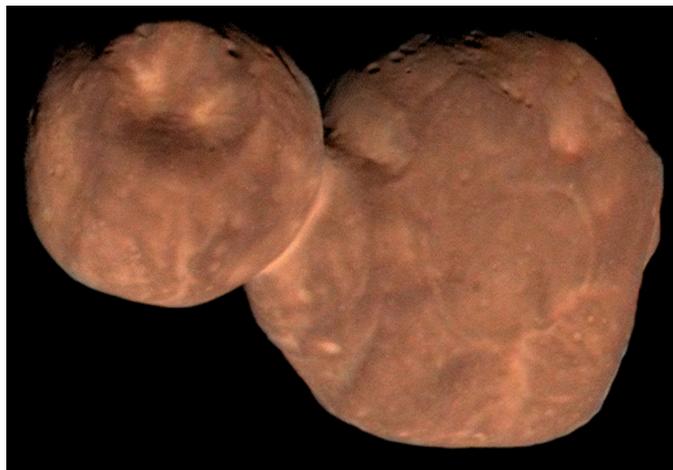


Abbildung 5: „Arrokoth“ besteht aus zwei umeinander kreisenden Körpern, die einander berühren.