

Simulation

Jakob Klemm

16. September 2020

Inhaltsverzeichnis

1 Einführung	1
2 Implementierung	2

1 Einführung

Auch wenn sich die Flugbahn einer **Saturn V** gut in **Excel** oder einem ähnlichen Programm berechnen lässt, so sind diese Programme doch limitiert. Sowohl in ihrer Geschwindigkeit, als auch in den Funktionen und Einstellungen, welche für eine solche Simulation relevant sein könnten. Mit den wenigen Werten, welche uns für diese Simulationen zur Verfügung standen, war es schnell möglich neue Werte und Flugbahnen zu berechnen.

Und trotzdem ist es äusserst kompliziert, die Simulation interaktiv zu gestalten und dem Betrachter die Möglichkeit zu geben, selber die Kontrolle zu übernehmen. Dazu kommt ebenfalls noch die starke Systemgebundenheit solcher Dateien. Auch wenn **Excel** weit verbreitet ist und es genügend kostenlose Alternativen gibt, welche in der Lage sind **Excel-Dateien** zu öffnen, ist man doch an die Funktionen und Bedingungen dieser Programme gebunden. Obwohl einige *kostenlos* sind, findet man kaum *freie* oder gar *open-source* Varianten.

Die Formeln und Konzepte, mit welchen sich die Geschwindigkeiten und Flugbahnen einer **Saturn V** berechnet werden, lassen sich unter verschiedenen Bedienungen und für verschiedene Raketen oder Flugobjekte anwenden. Um den Prozess interaktiver und intuitiver zu machen, war es von Nöten die Simulation als ein ausführbares, interaktives Programm zu implementieren.

2 Implementierung

Der Begriff "*Raketensimulator*" verbinden die meisten im ersten Moment mit Supercomputern und hoch komplexen Berechnungen. Trotzdem lässt sich das meiste auf einfache, lineare oder exponentielle Gleichungen bringen, die von Hand oder bereits mit einem einfachen Taschenrechner gelöst werden können.

Daher war auch die Wahl der Programmiersprache eher offen, da die Berechnungen selbst nicht besonders aufwendig sind. Trotzdem gab es einige Kriterien, welche eine Programmiersprache für diese Aufgabe erfüllen musste.

- Die Fähigkeit Dateien zu lesen, zu interpretieren und zu schreiben.
- Einfache, lineare Algebra sowie häufige mathematische Funktionen.
- Zugriff auf externe Programme über eine standardisierte Schnittstelle.
- Kompilierung in eine alleinstehende, ausführbare Datei.

Auch wenn viele Sprachen und Programme zur Verfügung standen, fiel die Wahl auf **Golang**, eine Sprache die ursprünglich von **Google** entwickelt wurde. **Golang** erlaubt es Entwicklern schnelle, parallele und nebenläufige Programme systemnah zu schreiben.