

Implementierung einer generischen Starrflügelflugzeug-Simulation in C++

Abschlussbericht für das Modul Effizient Programmieren I & II
von
Jan Olucak

durchgeführt am
Institut für Aerodynamik und Gasdynamik
der Universität Stuttgart

Stuttgart, im April 2018

Inhaltsverzeichnis

1	Problemstellung	1
1.1	Ausgangslage	1
1.2	Zielsetzung	1
2	Kurzvorstellung Programm Aircraft Designer	2
3	Aufbau der Simulation	3
3.1	Grundidee des Simulation-Frameworks	3
3.1.1	Beispiel Navigation-Klasse	3
3.2	Ablauf der Simulation	3
4	Durchführung der Implementierung	4
4.1	Entwicklungsumgebung und verwendete Tools	4
4.2	Dokumentation, User Manual und Coding Standard	4
4.3	Unit- und Modultests	4
4.4	Verifikation der Simulation	4
	Literaturverzeichnis	5
	Abbildungsverzeichnis	6
	Tabellenverzeichnis	7

1 Problemstellung

1.1 Ausgangslage

Der Prozess des Flugzeugentwurfs erstreckt sich über mehrere Schleifen in denen das Flugzeug immer detaillierter Ausgearbeitet wird. Um das Leistungsvermögen früh im Entwicklungsprozess zu beurteilen, können numerische Simulationen genutzt werden. Zum einen werden mathematische Modelle benötigt, um das physikalische Verhalten von spezifischen Domänen des Flugzeuges abzubilden, zum anderen werden Parameter benötigt, um besagte Modelle zu beschreiben. Viele Parameter sind erst im Laufe des Entwurfsprozesses verfügbar. Um das Flugverhalten dennoch abzubilden, wird eine Simulation mit verschiedenen Ausbaustufen und Fehlermodellen benötigt. Aus den meisten Ingenieursanwendungen ist MATLAB nicht mehr wegzudenken. Grund hierfür ist das breite Anwendungsspektrum und die Vielzahl an zusätzlichen Toolboxes. Dabei handelt es sich um eine proprietäre Programmiersprache die auf dem jeweiligen Rechner interpretiert wird. Trotz der vielseitigen Anwendungsmöglichkeiten benötigt MATLAB für komplexere Anwendung eine größere Laufzeit, im Vergleich zu Programmiersprachen, welche direkt in Maschinsprache übersetzt werden. Die Laufzeit ist mitunter einer sehr kritische Größe bei der Nachweisführung und Leistungsrechnung. Zudem ist eine objektorientierte Programmierung nur bedingt möglich, was den Aufbau einer generischen Simulation erschwert.

1.2 Zielsetzung

Innerhalb dieser Ausarbeitung soll eine generische Flugzeugsimulation in der Hochsprache C++ implementiert werden. Generisch bedeutet in diesem Zusammenhang, dass die Simulation in der Lage ist, verschiedene, mit dem Aircraft Designer modellierte Flugzeuge simulieren zu können. Jede Ausbaustufe des Entwurfsprozesses soll simulierbar sein, ohne dass dabei der Code verändert wird. Über Input-Files sollen verschiedene Domänen-Modelle ausgewählt werden können. Ein modularer Aufbau soll zudem eine einfache Erweiterung des Simulation gewährleisten. Um das Simulations-Framework zu verifizieren, wird die in [1] implementierte Simulation in C++ übersetzt. Der gesamte Entwicklungsprozess stützt sich dabei auf die in [2] vorgestellten Methoden und Anregungen.

2 Kurzvorstellung Programm Aircraft Designer

In der Arbeit nach [1] wurde eine MATLAB basierendes Programm entwickelt, mit dessen Hilfe Simulationsmodelle für Flugzeuge modelliert werden können. Die fertigen Modelle werden sowohl in Text-Files, als auch in MATLAB spezifischen .mat-files gespeichert. Um die Modelle zu verifizieren wurde zudem eine Simulation mit 6 Freiheitsgraden implementiert. Das Programm Aircraft Designer kann jederzeit um weitere Methoden ergänzt werden, um so den Entwurfsprozess fortzusetzen.

3 Aufbau der Simulation

3.1 Grundidee des Simulation-Frameworks

3.1.1 Beispiel Navigation-Klasse

3.2 Ablauf der Simulation

4 Durchführung der Implementierung

4.1 Entwicklungsumgebung und verwendete Tools

4.2 Dokumentation, User Manual und Coding Standard

4.3 Unit- und Modultests

4.4 Verifikation der Simulation

[3]

Literaturverzeichnis

- [1] Jan Olucak. *Modellierung und Implementierung eines Starflüglermodells zur Untersuchung von Außenlasten*. Bachelorarbeit, Technische Hochschule, Ingolstadt, 2017-02-15.
- [2] Manuel Kessler. Effizient programmieren i, Sommersemester 2017.
- [3] Manuel Kessler. Effizient programmieren ii, Wintersemester 2017/18.

Abbildungsverzeichnis

Tabellenverzeichnis