

Modeling & Simulation

SIR Model - Mass Tests

Christian Göth

Christian Sallinger

Florian Schager

Paul Winkler

31. Januar 2021

Betreuer: Martin Bicher

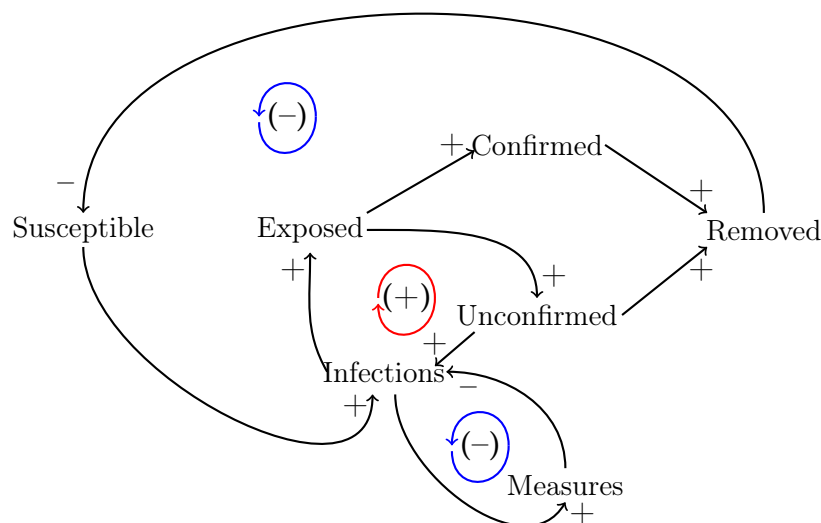
Abstract

In 2020 the COVID-19 pandemic caused worldwide suffering and deaths and the whole world is waiting for a vaccine. In winter 2020/2021 countries in Europe came up with the idea of executing nationwide mass tests to reduce the number of unconfirmed cases. This strategy should serve as an alternative to lockdown measures that force people to reduce their contacts. In this project we want to contrast these alternatives qualitatively and quantitatively by constructing a modified SIR Model to simulate the spread of the disease. With our model we want to answer the question: How many days of lockdown are necessary with different strategies of mass-testing?

Inhaltsverzeichnis

1	Model Description	3
2	Modell	3
2.1	Unterkapitel 2.1	3
2.1.1	Unterkapitel 2.1.1	3
2.2	Unterkapitel 2.2	4
3	Implementation	4
3.1	Source Code	4
4	Simulation Results	4
5	Conclusio	5

Abbildung 1: Causal Loop Diagram



1 Model Description

Our model is based on the classical SIR Model by Kermack and McKendrick, but in addition to the standard compartments Susceptible, Infectious, Recovered we introduce an extra Exposed compartment between the Susceptible and the Infectious. Furthermore we split the Infectious compartment into two separate compartments: Confirmed and Unconfirmed.

In our model we assume, that the persons in the Confirmed compartment are in quarantine and do not contribute to the infection rate anymore.

Stock and Flow Diagram:

2 Modell

2.1 Unterkapitel 2.1

Literaturquellen müssen im File *References.bib* definiert werden und können dann hier einfach mittels `\cite{QUELLE}` referenziert werden.

Empfehlenswerte Literatur zum Thema *Kontinuierliche Simulation* ist z.B.: [CK06].

2.1.1 Unterkapitel 2.1.1

Verwenden Sie die Latex Distribution MikTeX (unter Windows) bzw. TeXLive (unter Linux) und als IDE z.B. *TeXnicCenter*¹ (unter Windows) oder z.B. *Kile*² (unter Linux).

¹<http://www.texniccenter.org/>

²<http://kile.sourceforge.net>

2.2 Unterkapitel 2.2

Hier möchte ich auf Kapitel 2.1 Bezug nehmen.

3 Implementation

3.1 Source Code

Als nächstes fügen wir Source-Code aus einer Datei ein:

```
1 function [x, y] = polar2cartesian(r, phi)
2 %POLAR2CARTESIAN converts polar into cartesian coordinates
3
4 if r < 0
5     disp('Error in ''polar2cartesian''! Negative values for ''r'' are not allowed!');
6 end
7
8 x = r*cos(phi);
9 y = r*sin(phi);
10 end
```

Listing 1: Die Funktion *polar2cartesian*.

Listing 1 zeigt den Source-Code der Funktion *polar2cartesian*. Hierbei wurde ganz am Beginn des .tex-Files mittels `\lstset` angegeben, dass es sich bei dem in Folge einzufügenden Code um Matlab-Code handelt. Aus diesem Grund sind die Schlüsselwörter so schön eingefärbt.

Source Code ohne Datei:

```
1 class InOutputVector:public std::vector<InOutput> {
2     public:
3         int untreated_entry_changes;
4
5         InOutputVector() {
6             untreated_entry_changes = 0;
7         }
8         void setAt(int c, double val, double t) {
9             if (true==(*this)[c].already_treated) {
10                 untreated_entry_changes++;
11             }
12             (*this)[c].set(val, t);
13         }
14         double* treatAt(int c, double val) {
15             if (false==(*this)[c].already_treated) {
16                 untreated_entry_changes--;
17             }
18             return((*this)[c].treat());
19         }
20         void treatAll() {
21             for (int i=0; i<this->size(); i++){
22                 (*this)[i].already_treated = true;
23             }
24             untreated_entry_changes=0;
25         }
26     };
```

Listing 2: Definition der Klasse *InOutputVector*.

In Zeile 5 in Listing 2 ist der Kontruktor der Klasse *InOutputVector* implementiert.

4 Simulation Results

Tabelle 1 zeigt den verwendeten Parametersatz.

t_{end} [s]	X_0 [m]	V_0 [m/s]	d	relTol	absTol	Refine	maxdist	Tol
100	$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 0 \end{pmatrix}$	1	1e-3	1e-6	8	2	$\begin{pmatrix} 1e-3 \\ 1e-6 \end{pmatrix}$

Tabelle 1: Parametersatz A.

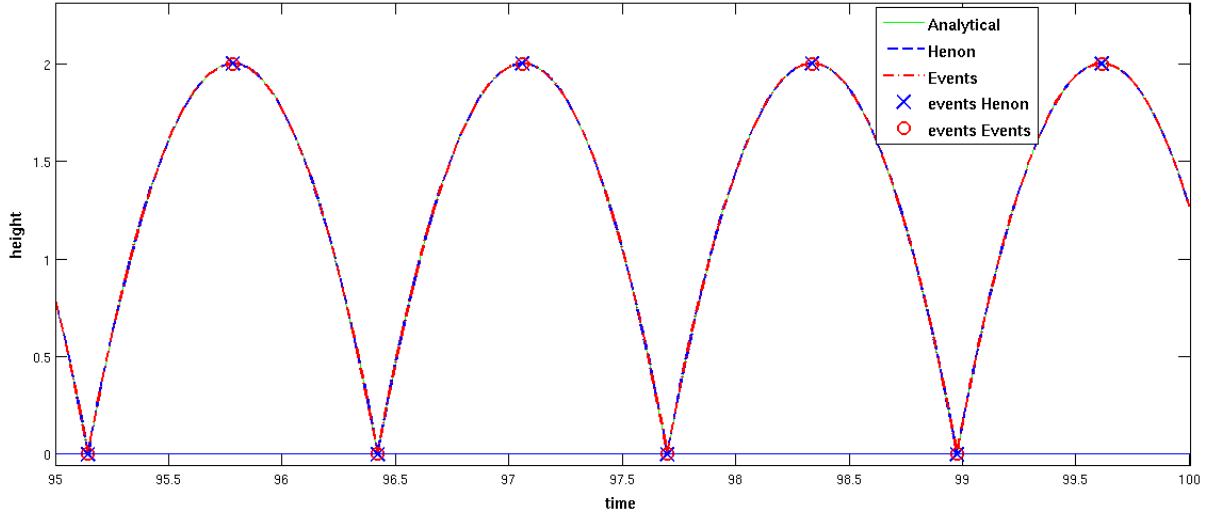


Abbildung 2: Simulationsergebnisse unter Verwendung von Parametersatz A.

5 Conclusio

Es wurde gezeigt, wie in Latex einzeilige, mehrzeilige, nummerierte und nicht nummerierte Formeln erzeugt werden können, sowie wie auf Formeln verwiesen werden kann und wie Matrizen erzeugt werden können. Weiters wurde gezeigt wie auf verwendete Literatur verwiesen werden kann, wie Source-Code eingefügt werden kann und wie auf einzelne Zeilen im Source Code verwiesen werden kann. Das Einfügen von Bildern und das Erzeugen von Tabellen wurde ebenfalls demonstriert.

Die grobe Struktur Ihres Projektprotokolls soll wie in diesem Dokument sein.

Zusammenfassend ist zu sagen, dass Sie die meisten \LaTeX - Befehle die Sie für ihr Protokoll brauchen werden, hier in diesem Template finden sollten. Es wird Ihnen aber trotzdem nicht erspart bleiben, sich einmal etwas genauer mit der \LaTeX Syntax auseinander zu setzen. Literatur und Foren zu \LaTeX sind im Internet ausreichend vorhanden.

Literatur

[CK06] F.E. Cellier and E. Kofman. *Continuous System Simulation*. Springer, New York, 2006.