yo

Abstract

uadnf+ads

1 Introduction

blalba

2 Algorithms, implementation and testing

As this project is new this year (as far as I know), some trial and error, as well as some creative freedom to change equations for more realistic implementation was taken.

2.1 Runge-Kutta model

When the population is stable, such that N is constant, we have that

$$N = I + R + S \tag{1}$$

Which can be simplified such that

$$R = N - R - S$$

So that R is plugged into the equation for S, and we only need 2 equations. However, as is the case with the vital parameters where N is changing, we now have to update N and R, which depend on each other, which is no longer possible. This version of the RK-model turns out to be signifficantly slowe, as R now needs to be calculated for every step.

2.2 Monte Carlo method

2.3 Tests

2.3.1 Testing the RK solver with expected values

2.3.2 Testing the population class in the MC solver

2.4 Optimizations and runtimes

2.4.1 2 vs 3 equations in RK solver

2.4.2 Using array of functions to avoid if-statements in MC solver

2.4.3 Compiler flag optimizations

2.4.4 OpenMP for multiple consecutive runs of MC solver

3 Analysis

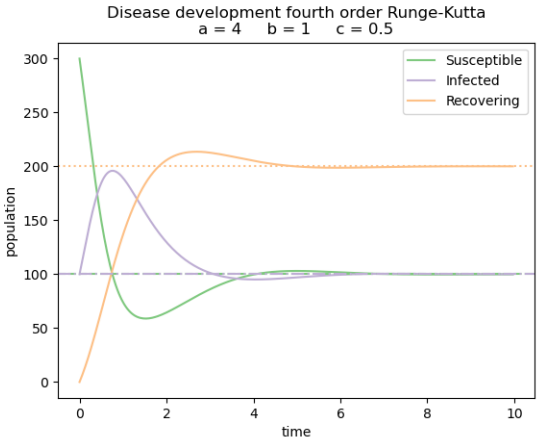

Figure 1: kamsdoka

maybe only include infected and compare that?

4 Discussion

5 Conclusion