Epidemics Plugin

18.03.2021

Abstract

Epidemics is a UG4 Plugin that provides various algorithms useful for epidemics modeling. The following algorithms are currently implemented:

SIR	Standard Susceptible-Recovered-Infected ODE Model
SEIRD	Extended SRI model accounting for an incubation period and more

Usage of the plugin assumes a working knowledge with Lua and C++. Strictly speaking, for most models UG4 is not required. Expert users can evaluate many models with only a working C++ compiler. This is shown in **Running Epidemics without UG4**.

For questions etc. contact devanshr, CowFreedom or others involved in the project.

Table of Contents

Installation

Epidemics requires a C++ compiler compatible with the C++17 standard. It has been tested with MSVC and GCC.

Most models can be run standalone with UG4. If you decide to link it with UG4, add the entire Epidemics folder into your plugin directory and follow the usual instructions for recompiling UG4. If you use older compilers, compilation of UG4 may halt with an error. Historically, this was mostly caused because the older compilers did not link activate the C++17 standard by default. Below are instructions on how to fix such situations. Only try this if compilation with UG4 did not work.

Activate additional compilation flags in GCC

Follow the steps on the ughub GitHub page. If errors occur proceed with this text. The plugin makes use of C++ std::threads. This might necessitates activating the -pthread flag to the build process for ug4. Within your UG4 install library, go to

$ cd ug4/ugcore/cmake/

and open

$ ug_includes.cmake

Now search for

```

elseif("${CMAKE_CXX_COMPILER_ID}" STREQUAL "GNU")

  add_cxx_flag("-Wall")

  add_cxx_flag("-Wno-multichar")

  add_cxx_flag("-Wno-unused-local-typedefs")

  add_cxx_flag("-Wno-maybe-uninitialized")

```

and add

```
   add_cxx_flag("-pthread")
```

to the GNU include statements. Now rebuild UG4 as usual. The plugin should now be installed without any issues.

Activate additional compilation flags in Clang

Follow the steps on the ughub GitHub page. If errors occur proceed with this text. The plugin makes use of C++11 features, like std::threads and constexpr. This might necessitates activating the -std=c++11 flag to the build process for ug4. Within your UG4 install library, go to

$ cd ug4/ugcore/cmake/

and open

$ ug_includes.cmake

Now search for

```

elseif("${CMAKE_CXX_COMPILER_ID}" STREQUAL "Clang")

  add_cxx_flag("-Wall")

  add_cxx_flag("-Wno-multichar")

```

and add

  set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++17")

to the Clang include statements. This ensures that the -std=c++17 flag is only added to the C++ compiler and not the C compiler in the build process. Now rebuild UG4 as usual. The plugin should

now be installed without any issues.

Running Epidemics without UG4

Compilation of UG4 takes a long time and is cumbersome. Fortunately, it is not needed to have a compiled UG4 in order to use most models in *Epidemics*. You can just download the *Epidemics* folder and include the header files of the respective models you want into your program. The samples in the examples folder in *Epidemics* do not make use of UG4 and can be studied.

Models

SEIRD

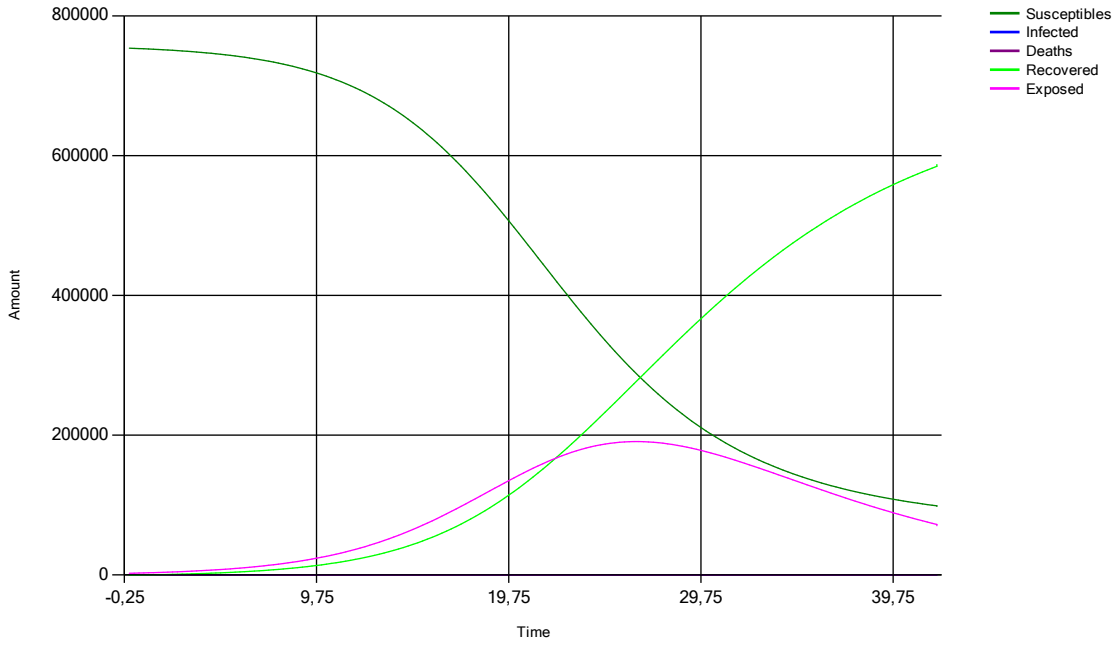

Figure 1 Example plot of the SEIRD model.

$S(t)$	Number of susceptibles at time t
$E(t)$	Number of exposed at time t
$I(t)$	Number of infected at time t
$R(t)$	Number of recovered at time t
$D(t)$	Number of deaths at time t
$\alpha \in [0,1]$	Exposed growth rate
$\kappa \in [0,1], \kappa \leq qq$	Conversion rate exposed to infected
$\theta \in [0,1]$	Death growth rate
$pp \in \mathbb{N}$	TODO
$qq \in [0,1]$	Exposed decay rate

The SEIRD model is an extension of the previously defined SIR model. In *Epidemics* it is defined in *seird.h* as such:

$$\frac{dS}{dt} = -\alpha S \cdot E$$

$$\frac{dE}{dt} = \alpha S \cdot E - (1 - qq)E$$

$$\frac{dI}{dt} = \left(\frac{\kappa}{qq}\right)E - \left(\frac{1}{pp}\right)I$$

$$\frac{dR}{dt} = \frac{(1-\kappa)}{qq} \cdot E + \left(\frac{1-\theta}{pp}\right)I$$

$$\frac{dD}{dt} = \frac{\theta}{pp} \cdot I$$

Internally, the system of ODEs is computed using an explicit fourth order Runge-Kutta method. The model is very sensitive to changes in α and it is not unusual for the model to blow up in finite time.

“EpidemicsRunner” User Interface

The current interface is programmed using WinForms. The Linux and Mac Versions will utilize GTK.

Future

License