

## Projektarbeit zur Vorlesung Computergestütztes Wissenschaftliches Rechnen

Abgabedatum: 08.08.2023 23.59 Uhr      Betreuung: Emily Klass

### Projekt 6: Modell für die Ausbreitung von Infektionskrankheiten

#### Einleitung

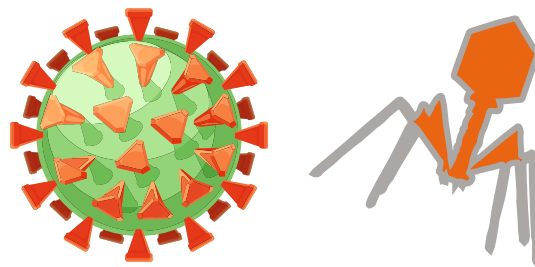


Abbildung 1: Beispiele für Viren.

Wir untersuchen ein vereinfachtes Gittermodell (ein sog. zellulärer Automat) zur Modellierung der epidemischen Ausbreitung einer durch Viren übertragenen Infektionskrankheit. Wir betrachten dazu ein zweidimensionales Quadratgitter mit  $L \times L$  Gitterplätzen, wobei sich auf jedem Gitterplatz eine Person aufhalten soll. Der Einfachheit halber sollen die Personen sich nicht fortbewegen (in einer realistischeren Modellierung müsste man dies noch mit berücksichtigen, und auch, dass nicht auf jedem Gitterplatz eine Person sitzt). Die Personen sollen dabei folgende Zustände durchlaufen können:

Anfällig  $A \rightarrow$  Infiziert  $I \rightarrow$  Genesen  $G \rightarrow$  Anfällig  $A$ ,

d.h., in diesem Modell ist es möglich, dass bereits genesene Personen wieder anfällig werden und sich wieder infizieren können. Jede Person auf dem entsprechenden Gitterpunkt ist entweder im Zustand  $A$ ,  $I$  oder  $G$ .

Die Ausbreitung der Infektion wird durch folgende Regeln modelliert:

- Ist eine Person anfällig und einer der nächsten Nachbarn krank, dann ist die Wahrscheinlichkeit, dass diese Person erkrankt,  $p_1$ .
- Ist eine Person erkrankt, dann kann sie mit Wahrscheinlichkeit  $p_2$  wieder gesund werden.
- Ist die Person genesen, dann wird sie mit Wahrscheinlichkeit  $p_3$  wieder anfällig für eine Infektion.

## Aufgaben

1. **(30 P)** Schreiben Sie einen Code, der die oben genannten Regeln implementiert.

Da alle Personen, bevor sie in unser abgeschlossenes System gekommen sind, mit weiteren Personen in Kontakt gewesen sind, können bereits zu Beginn der Simulation alle drei Zustände  $A$ ,  $I$  und  $G$  realisiert sein. Erstellen Sie daher den Anfangszustand, indem Sie jedem Gitterplatz mit gleicher Wahrscheinlichkeit zufällig den Zustand  $A$ ,  $I$  oder  $G$  zuordnen.

Die Größen  $p_1$ ,  $p_2$ ,  $p_3$  und  $L$  sollen Eingabeparameter sein. Stellen Sie sicher, dass bei jedem Simulations(zeit)schritt das gesamte Gitter nach den oben genannten Regeln aktualisiert worden ist. Dies können Sie erreichen, indem Sie sequentiell durch das Gitter laufen und jeden Gitterpunkt laut den obigen Regeln aktualisieren, oder  $L^2$  mal zufällig einen Gitterpunkt wählen und diesen dann entsprechend obiger Regeln aktualisieren. Führen Sie die folgenden Simulationen mit einem dieser Aktualisierungsschemen durch und begründen Sie Ihre Wahl. Erwarten Sie einen wesentlichen Unterschied zwischen beiden Aktualisierungsschemen?

2. **(30 P)** Um zu messen, wie viele Personen nach einer vollständigen Aktualisierung des Gitters zum Simulationszeitpunkt  $t$  infiziert sind, berechnen wir das Infektionsverhältnis zu diesem Zeitpunkt,

$$\langle I \rangle_t := \frac{1}{L^2} \sum_{i=1, j=1}^L I_{i,j}, \quad (1)$$

wobei  $i$  und  $j$  die Koordinaten des Gitterpunktes,  $L$  die Größe des Gitters und  $t$  die Simulationszeit sind. Es sei  $I_{i,j} = 1$ , wenn die Person akut infiziert ist und  $I_{i,j} = 0$  andernfalls. Um die durchschnittliche Infektionsrate über einen bestimmten Simulationszeitraum  $T$  zu berechnen, müssen wir den zeitlichen Mittelwert über  $\langle I \rangle_t$  bilden,

$$\overline{\langle I \rangle} := \frac{1}{T} \sum_{t=0}^T \langle I \rangle_t. \quad (2)$$

Betrachten Sie zunächst ein Gitter der Größe  $L = 16$ . Setzen Sie  $p_2 = 0.3$  und  $p_3 = 0.3$  und berechnen Sie jeweils  $\overline{\langle I \rangle}$  für  $p_1 = 0, 0.02, 0.04, \dots, 1$ . Plotten Sie die durchschnittliche Infektionsrate  $\overline{\langle I \rangle}$  als Funktion von  $p_1$ .

Wiederholen Sie die Rechnung für  $L = 32$  und  $L = 64$ , sowie für einen weiteren, grösseren Wert von  $L$ , den Sie noch mit ‘vernünftigem’ Rechenaufwand behandeln können. Was ist dann der grösstmögliche  $L$ -Wert, den Sie simulieren? Begründen Sie Ihre Wahl. Plotten Sie die mit den verschiedenen Gittergrößen erhaltenen Ergebnisse für  $\overline{\langle I \rangle}$  als Funktion von  $p_1$  und beschreiben Sie, wie sich die Ergebnisse mit grösser werdendem  $L$  ändern. Gibt es einen kritischen Wert, ab dem die durchschnittliche Infektionsrate null wird?

Wiederholen Sie die selbe Untersuchung mit  $p_2 = 0.6$  und  $p_3 = 0.3$ . Beschreiben und erläutern Sie Ihre Ergebnisse und vergleichen Sie mit den Ergebnissen für  $p_2 = 0.3$  und  $p_3 = 0.3$ . Diskutieren Sie, ob die Ergebnisse bei Änderung von  $p_2$  Ihren Erwartungen entsprechen.

*Hinweis:* Als Simulationszeitpunkt  $t$  können Sie eine ganze Zahl nehmen, nämlich die Anzahl der Durchläufe, bei denen das gesamte Gitter aktualisiert wurde.

3. **(20 P)** Nehmen wir nun an, dass eine gut wirkende Impfung gegen das Virus entwickelt wurde, so dass jede Person nun auch einen weiteren Zustand  $V$  (‘vaccinated’) annehmen kann. Zu Beginn der Simulation sei jede Person nun mit Wahrscheinlichkeit  $p_4$  geimpft und somit im Zustand  $V$ . Die Impfung sei so wirksam, dass die geimpften Personen während des gesamten Simulationsverlaufs unverändert im Zustand  $V$  verbleiben und somit niemanden in ihrer Umgebung anstecken können.

Modifizieren Sie Ihren Code entsprechend und schließen Sie nun die Präsenz geimpfter Personen mit ein. Wählen Sie  $p_1 = p_2 = p_3 = 0.5$  und berechnen Sie nun die durchschnittliche Infektionsrate  $\langle I \rangle$  als Funktion von  $p_4$ , wobei die Simulationen für  $p_4 = 0, 0.02, 0.04, \dots, 1$  durchgeführt werden sollen. Erstellen Sie für die in Teilaufgabe 2 genutzten Systemgrößen  $L$  einen Plot für  $\langle I \rangle$  als Funktion von  $p_4$ , in dem die Ergebnisse für die verschiedenen Gittergrößen miteinander verglichen werden. Diskutieren Sie die Ergebnisse.

## Allgemeine Hinweise zur Abgabe

### • Code

- Soweit nicht in der Aufgabenstellung anders angegeben, müssen Sie die in Ihrem Programm benötigten Algorithmen selbst implementieren und dürfen nicht vorgefertigte Routinen aus Bibliotheken (z.B. GSL) nutzen. Die Nutzung der Standard-Bibliothek von C ist aber selbstverständlich erlaubt.
- Der C-Code, den Sie abgeben, muss zuverlässig die (ebenfalls abzugebenden) Daten generieren. Insbesondere muss er auf dem im CIP-Pool der Physik installierten Linux-System kompilierbar sein und ohne Laufzeitfehler durchlaufen. Die Kompilation des C-Programms muss durch ein Makefile automatisiert sein.
- Auch die im Bericht gezeigten Plots müssen mit ebenfalls abzugebendem Python-Code reproduzierbar sein. Auch dieser muss im CIP-Pool lauffähig sein. Die eigentliche Berechnung/Simulation soll im C-Code stattfinden, der Python-Code soll also wirklich nur zum Plotten und ggf. zur vorbereitenden Strukturierung der Daten verwendet werden.
- Der abgegebene Code muss gut lesbar und sinnvoll strukturiert sein. Kompliziertere Code-Blöcke sollen dabei mit Code-Kommentaren erläutert werden.

### • Bericht

Der Umfang des abgegebenen Berichts soll 10 Seiten sein, nicht mehr. Stellen Sie den im Folgenden dargestellten Inhalt darin knapp auf deutsch oder englisch dar. Beachten Sie, dass ein Bericht ein zusammenhängender Text ist; eine Sammlung von Stichworten ist kein Bericht. Der Bericht ist außerdem ein wissenschaftlicher Text, d.h. im Prinzip muss es möglich sein, die beschriebene Simulation anhand des Berichts nachzuprogrammieren und auf dieselben Ergebnisse zu kommen. Das heißt insbesondere, dass Sie *eigene* Parameterwahlen dokumentieren müssen. (Sie dürfen aber davon ausgehen, dass dem/der Leser/in das Aufgabenblatt vorliegt, Sie müssen also nicht unbedingt alle darin gegebenen Parameterwahlen, physikalische Größen usw. wiederholen, es kann jedoch für die Lesbarkeit vorteilhaft sein.)

1. **Einleitung:** Kurze Einführung in das Thema bzw. das zu simulierende Modellsystem, sowie Nennung der untersuchten Fragestellung(en) und der verwendeten Methodik.
2. **Theorie:** Ggf. in der Aufgabenstellung geforderte analytische Rechnungen und Herleitungen, Diskretisieren von Differenzialgleichungen usw.
3. **Methodik:** Kurze Darstellung der verwendeten Algorithmen und wofür sie im Projekt eingesetzt wurden (jeweils 1–2 Absätze).
4. **Implementation:** Kurze Beschreibung des C-Programms (Struktur und Programmflow, Verwendung von Bibliotheken, Besonderheiten). Falls die typische Laufzeit Ihres Programms länger als etwa eine Minute dauert, sollten Sie dies hier ebenfalls erwähnen.
5. **Ergebnisse und Diskussion:** Der wichtigste Teil sind natürlich die Darstellung der in der Aufgabenstellung geforderten Ergebnisse, Plots und Diskussionen bzw. Antworten auf die

gestellten Fragen. Plots müssen korrekte Achsenbeschriftungen tragen und alle darin dargestellten Datensätze müssen z.B. durch eine Legende und/oder in der Bildunterschrift eindeutig identifizierbar sein. Alle Plots müssen im Haupttext eingeführt und die darin dargestellten Ergebnisse diskutiert werden. Achten Sie darauf, dass die Plots klar lesbar sind! Das kann die Schriftgröße der Achsenbeschriftung betreffen, aber auch zu viele dargestellte Linien können die Lesbarkeit erheblich beeinträchtigen – lieber zwei klare Plots machen, als einen, in dem man nichts erkennt.

Diskutieren Sie immer auch Fehlerquellen der numerischen Verfahren und quantifizieren Sie diese, wenn verlangt.

Die Aufgaben müssen eigenständig bearbeitet werden. Die Verwendung von Literatur (auch aus dem Internet) wird ausdrücklich begrüßt. Geben Sie aber stets die von Ihnen verwendeten Quellen an. **Sollte sich Ihre Lösung in wesentlichen Teilen als Plagiat herausstellen, wird Ihr Projekt als “nicht bestanden” gewertet.**

- **Bewertung**

Die maximale Punktzahl ist **100**, die Projektarbeit gilt als bestanden, wenn mindestens **50 Punkte** erreicht werden. 80 Punkte entfallen auf die projektspezifischen Aufgabenstellungen. Jeweils 10 Punkte werden für Lesbarkeit des Programmcodes und für die allgemeine Qualität der Darstellung von Daten in Plots vergeben. Die Aufgaben sind nicht gleich und damit ggf. auch nicht gleich schwierig. Der Schwierigkeitsgrad der Aufgabe wird jedoch bei der Bepunktung der Aufgabenteile und der Bewertung berücksichtigt. **Wenn keine ausreichende Fehlerdiskussion im Bericht vorhanden ist, können (unabhängig von der Verteilung der Punkte in der Aufgabenstellung) global bis zu 10 Punkte abgezogen werden.**

- **Abgabe**

Abzugeben sind der Programmcode mit Makefile, der Bericht als PDF, generierte Datensätze und der Plotting-Code. (Plots müssen nicht einzeln abgegeben werden, sondern sind Teil des Berichts; eventuelle  $\text{\LaTeX}$ -Quelldateien müssen auch nicht mit abgegeben werden.) Falls noch nicht geschehen, erstellen Sie für dieses Projekt ein `git`-Repository auf <https://gitlab.gwdg.de> und geben Sie Ihrem betreuenden Tutor / Ihrer betreuenden Tutorin (Name siehe erste Seite dieses Dokuments) und PD Dr. Manmana Zugang zu diesem Repository. **Achten Sie darauf, dass keine weiteren Personen Zugang zu diesem Repository haben!** Laden Sie alle abzugebenden Dateien in dieses Repository und informieren Sie Ihren Tutor/Ihre Tutorin, wenn Sie das Projekt einreichen. Falls Sie Fragen oder Probleme mit `git` haben, wenden Sie sich an den betreuenden Tutor/betreuende Tutorin. Das Abgabedatum finden Sie am Anfang dieses Dokuments.