Using Monte-Carlo simulations for modeling the Coronavirus Disease (COVID-19)

Hilla De-Leon (ECT*, INFN-TIPFA),
Francesco Pederiva (UNIT, INFN-TIPFA)

The Monte-Carlo algorithm

- The Monte-Carlo (MC) algorithm is based on transforming integrals into sums of functions evaluated on samples (usually called "*walkers*") of some distribution.

- The absence of truncations or uncontrolled approximations makes MC a powerful ab-initio method that can perform reliable integration of complex many-body systems and complex interactions.

The Monte-Carlo algorithm

- In our model, each demographic area contains two types of populations: healthy and sick people.

- A healthy person (i) can become sick with a daily probability

  $P_i = \sum_j P_{ij}$ where P_{ij} is a function of:

  1. The distance between each sick people (j) in the area and the healthy person (i).
  2. The amount of time that the healthy person spends with each sick people (t_{ij}).

The Monte-Carlo algorithm

- We are using a "one-way" Ising-model Monte Carlo: A healthy person become sick as a result of an interaction with sick person (or people), but a sick person stops being sick (i.e. recovered or died) within an average time of ~14 up to 40 days for the severs cases.
- After that time, the recovered person cannot infect another person anymore.
- We repeat this simulation

The parameters and Preliminary assumptions:

- Unfortunately, currently there's still a lot of unknowns regarding the spread of the novel Coronavirus, therefore we had to take some assumptions based on the current data, which may be more stringent than the real nature of the virus.

- In our work we modeled the spread of the Coronavirus by running many MC simulations using the following parameters:

The parameters and Preliminary assumptions:

- For all simulations, we used an area of 1 squared-km (with periodic boundary conditions, which allow us to treat this area as a bulk) with different population density.

- The population density is a function of the number of households in a certain area (rather than the number of people), since it is almost impossible to prevent contagion in the family.

The parameters and Preliminary assumptions:

- The population density is an "**effective**" density which represents not only the number of families in a certain area, but also the "kinetic energy (or temperature)" of the population .

- For example, a nursing home with a common dining area is denser than the ones where all the residents eat in their rooms, and a neighborhood with many young families (which meet in the playground etc.) is denser than the same neighborhood with less social activities.

The parameters and Preliminary assumptions:

- The average time for a symptomatic patient to develop symptoms is 5 days with $\sigma = 1 \text{ day}$.

- The number of effective households is denoted by: **N**

- Some of the carries are "silent carriers" which have no symptoms (AKA asymptomatic) but can infect other people with a chance of 50%. Their fraction in the population is denoted by: $\boldsymbol{a_{silent}}$.

The parameters and Preliminary assumptions:

- In day 0 there is only one sick person: **patient_zero**

- In some of the models we have infections from an unknown sources (a health person can become sick without an interaction with a known sick person).

- In the first 14 days the population without symptoms walks freely without any restrictions.

- For each sick person, the infection period is between the 3th day to the 7th day.

The parameters and Preliminary assumptions:

- The probably of the ith healthy person to get infected is:

$$P_i = int\left\{\sum_{j=1}^{n_{sick}} P_{ij} + \xi\right\} = int\left\{\sum_{j=1}^{n_{sick}} \exp\left[-\frac{(|\vec{r_i}-\vec{r_j}|)^2}{2 \cdot \sigma_r^2}\right] \times f(a_{silent}, n_{out}) + \xi\right\}$$

Where:

- $r_i(x_i, y_i)$ is the location of the ith healthy person and $r_j(x_j, y_j)$ is the location of the jth sick person, so $|\vec{r_i} - \vec{r_j}|$ is the distance between them

- n_{sick} is the total number of sick people in the area.

- σ_r is the standard deviation (here $\sigma_r = 2.4$ meters).

The parameters and Preliminary assumptions:

○ $f(a_{silent}, n_{out})$ is a function that considers the social activity of the sick person and whether he has symptoms, which affects the spread of the virus outside the house.

○ Each sick will infect some of his household member (uniformly distributed between 1 and 5).

○ ξ is a random number between 0 and 1, which allows us to consider some violation of the lockdown and the fact that even during full lockdown, people continue visiting supermarket etc.

Different models of lockdown:

We considered different models of lockdown.

In all models:

- Days 1-14: no restrictions.

- Days 15-50: full lockdown with social distancing (SD), i.e., people are forced to keep 4 meters from each other.

- Days 51-200: people must wear face-masks so that the daily infection (outside) probability is reduced to:

$$P_i = int\left\{0.7 \times \sum_{j=1}^{n_{sick}} \exp\left[-\frac{(|\vec{r_i}-\vec{r_j}|)^2}{2\cdot\sigma_r^2}\right] \times f(a_{silent}, n_{out}) + \xi\right\}$$

Different models of lockdown:

- Model #1: Days 51-200: no restrictions.

- Model #2: Days 51-200: two weeks with no restrictions and then two weeks with full lockdown and so on.

- Model #3: Days 51-200: one week with no restrictions and then one week with full lockdown and so on.

- Model #4: Days 51-200: one week with no restrictions and then two weeks with full lockdown.

Each model was tested with and without SD on days 51-200

The algorithm

- For each model on day 0 there is only one sick person in the entire area, i.e. $n_{\text{healthy}} = N - 1, n_{\text{sick}} = 1$

- R_{sick} is a $2 \times n_{sick}$ vector of the positions of all sick people (x and y coordinates).

- R_{healthy} is a $2 \times n_{\text{healthy}}$ vector of the positions of all healthy people (x and y coordinates).

- V_{day} is a $1 \times n_{\text{sick}}$ vector of the number of the day in which each sick person was infected

The algorithm

For each day ($t = [1,2 \ldots 200]$):

1. We check if the jth sick person is still sick by:

   - Computing $t_{recovery}$ which is normally distributed, so that:

   $$P(t_{recovery}) \sim \frac{1}{2} \times \left\{ \exp\left[\frac{(t_{recovery}-14)^2}{2 \cdot 4^2}\right] + \exp\left[\frac{(t_{recovery}-18)^2}{2 \cdot 6^2}\right] \right\}$$

   - If $(t^{th} - V_{day}^j) > t_{recovery}$, we remove him from the R_{sick} vector

The algorithm

2. All can change their position in the area such that:

$$R^i_{\text{healthy}} \rightarrow R^i_{\text{healthy}} + \rho$$

where ρ distributes normally with $\sigma_1 = 500$ meter for no restrictions period and with $\sigma_0 = 0.5 \times 10^{0.5}$ meter (a) or $\sigma_0 = 0.5 \times 10^{0.5}$ (b) for full lockdown period.

The algorithm

3. The Probably of the ith healthy person to get infected is:

$$P_i = int\left\{\sum_{j=1}^{n_{sick}} \exp\left[-\frac{\left(R_{free}^i(x) - R_{sick}^j(x)\right)^2 + \left(R_{free}^i(y) - R_{sick}^j(y)\right)^2}{2 \cdot \sigma_r^2}\right] \times \right.$$
$$f(a_{silent}, n_{out}) + \xi \left.\right\}$$

- If P_i (the sum of the partial probabilities, P_{ij}) exceeds 1, we set it to be $P_i = 1$
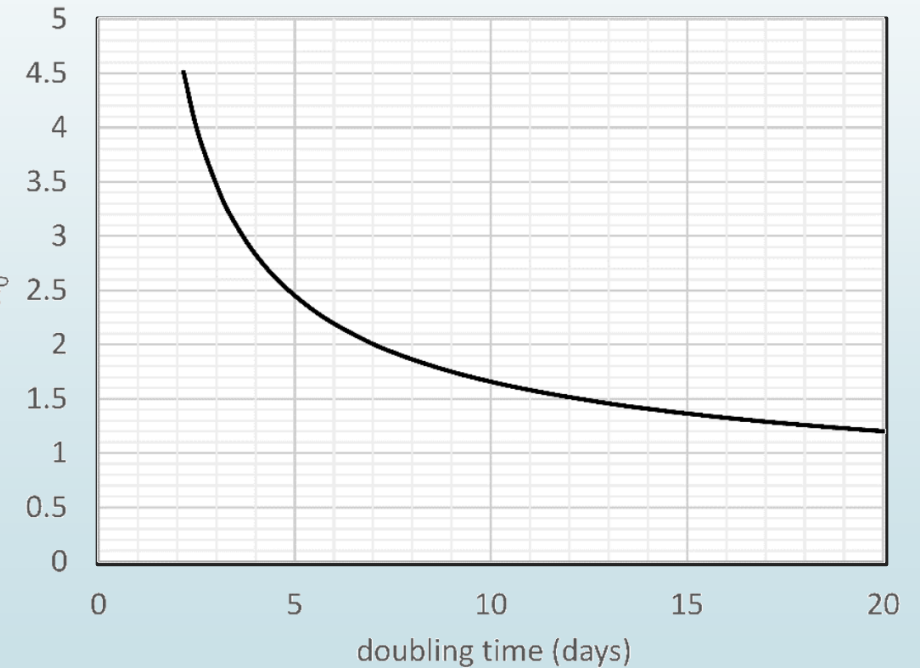- We used $\sigma_r = 2.4$ meter

The algorithm

3. The Probably of the ith healthy person to get infected is:

$$P_i = int \left\{ \sum_{j=1}^{n_{sick}} \exp\left[-\frac{\left(R_{free}^i(x) - R_{sick}^j(x)\right)^2 + \left(R_{free}^i(y) - R_{sick}^j(y)\right)^2}{2 \cdot \sigma_r^2} \right] \times \right.$$
$$\left. f(a_{silent}, n_{out}) + \xi \right\}$$

- For the case of SD: If $\left(R_{healthy}^i - R_{sick}^j\right) < 3$ meter $\rightarrow$ $(R_{healthy}^i - R_{sick}^j) = 3$ meter

- If $P_i = 1, R_{healty}^i \rightarrow R_{sick}^{j+i}$

- The algorithm runs until day 200 or when there are no more sick people.

Basic reproduction number

o For the coronavirus, the basic reproduction number, R_0 , suggests each infection directly generates 2-4 more infections in the absence of countermeasures like social distancing.

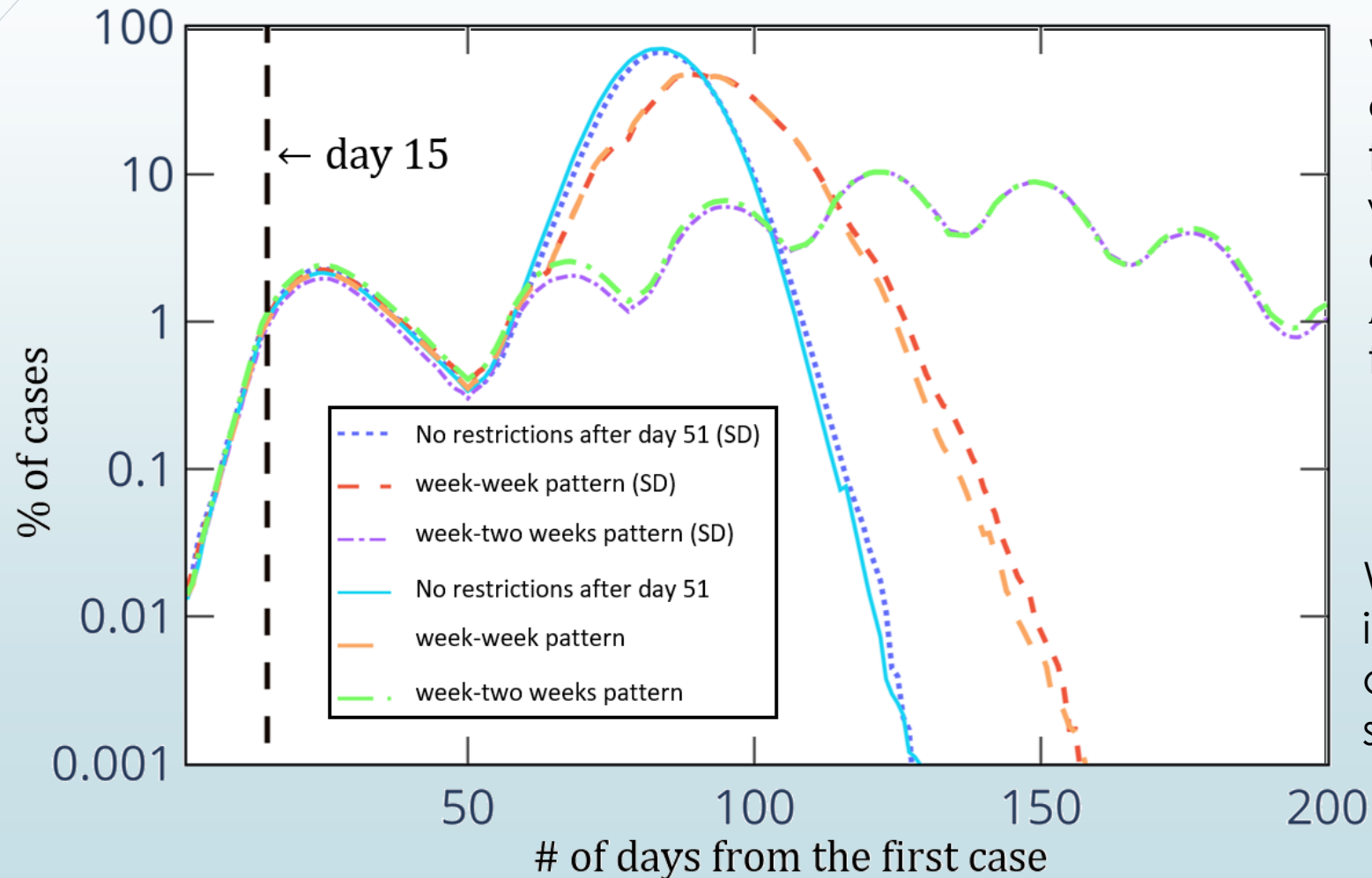o R_0 is a function of doubling time T_d (see arXiv:2003.08824).

Results: doubling time vs population density
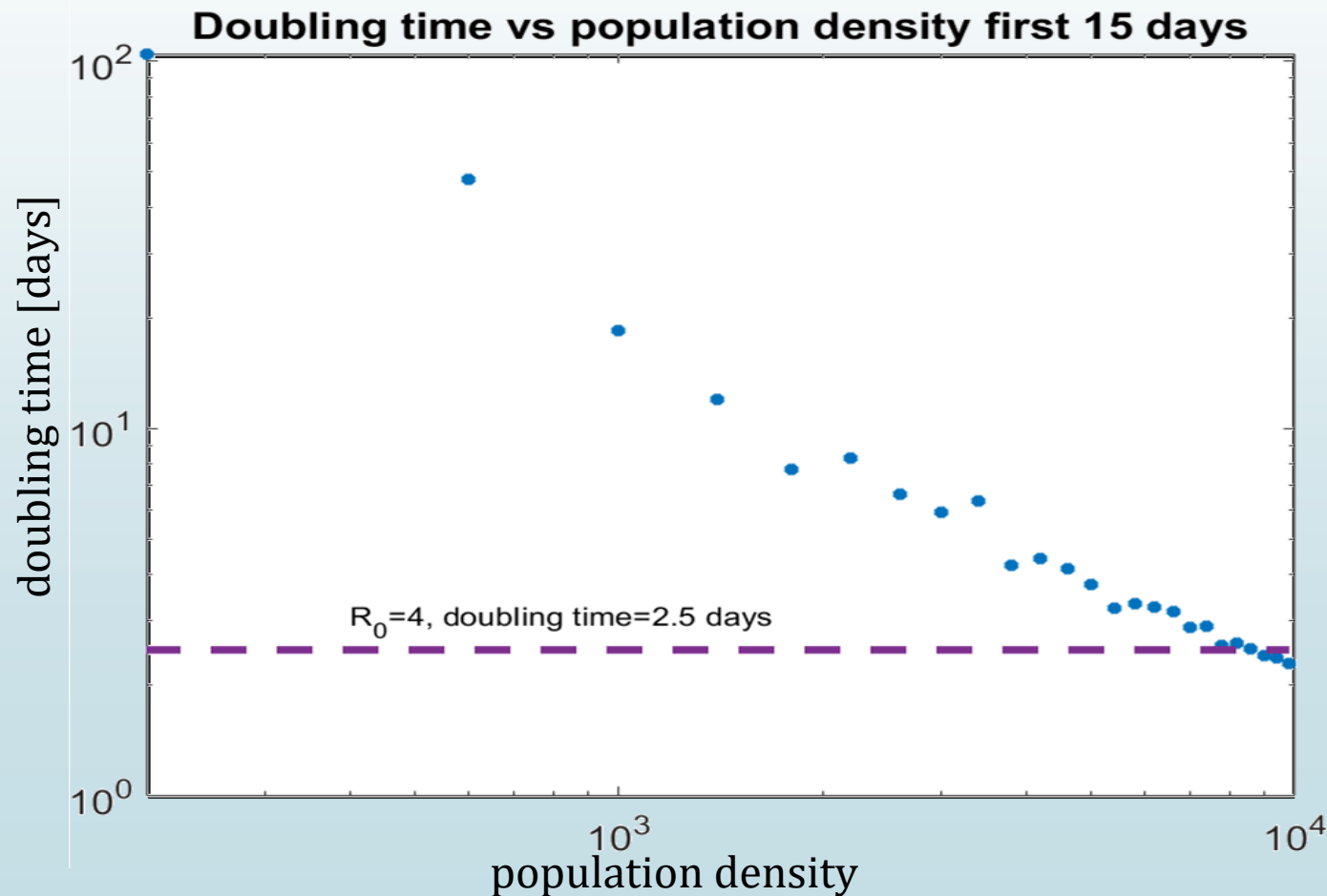
Doubling time vs population density first 15 days

Results: $R_0(1-15) = 4$

← day 15

% of cases

No restrictions after day 51 (SD)

week-week pattern (SD)

week-two weeks pattern (SD)

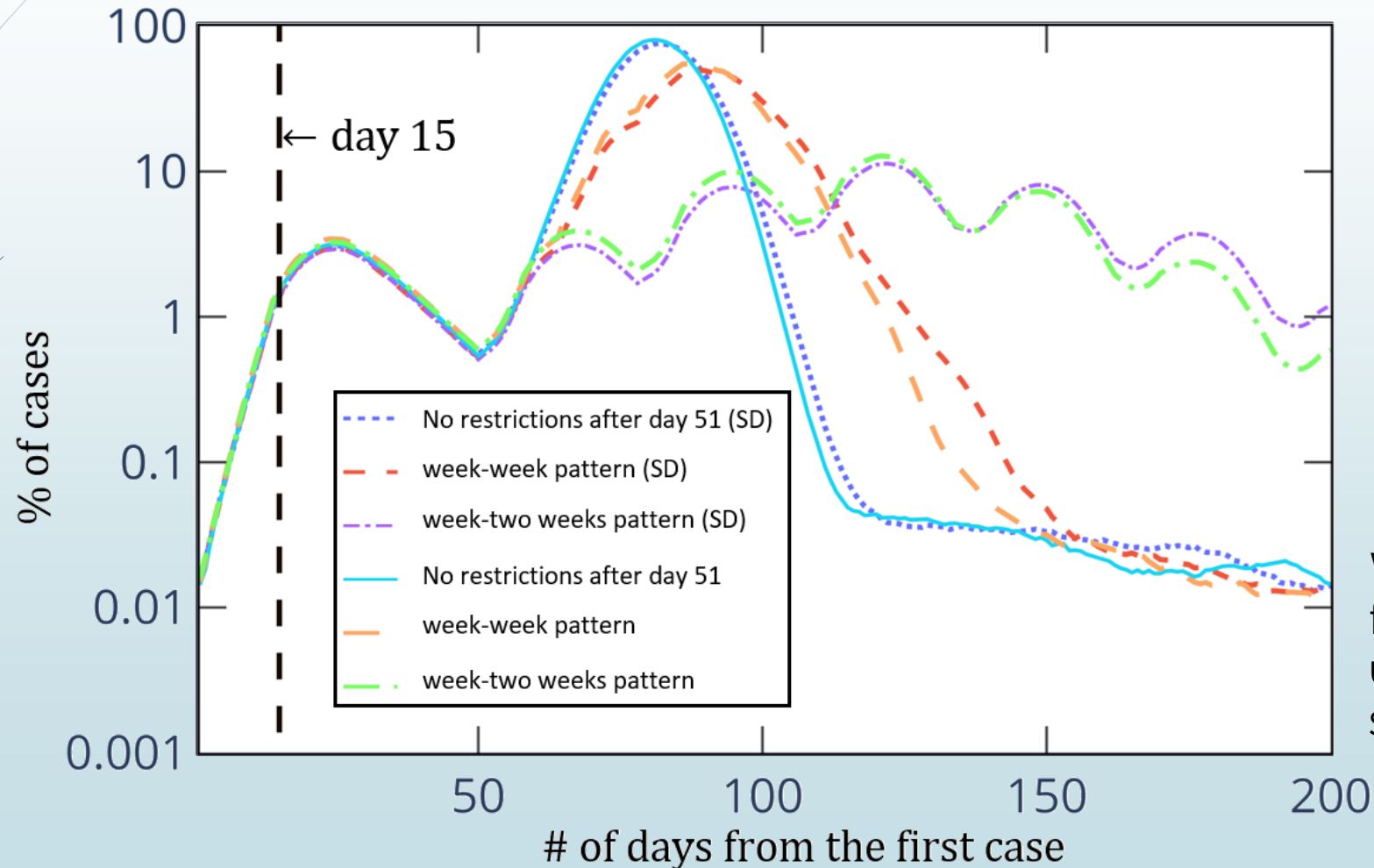No restrictions after day 51

week-week pattern

week-two weeks pattern

# of days from the first case

We check different models for N=9000, which is associated with $R_0 = 4$ (for the first 14 days).

Without infections from an unknown sources

Results: $R_0(1-15) = 4$

We check different models for N=9000, which is associated with $R_0 = 4$ (for the first 14 days).
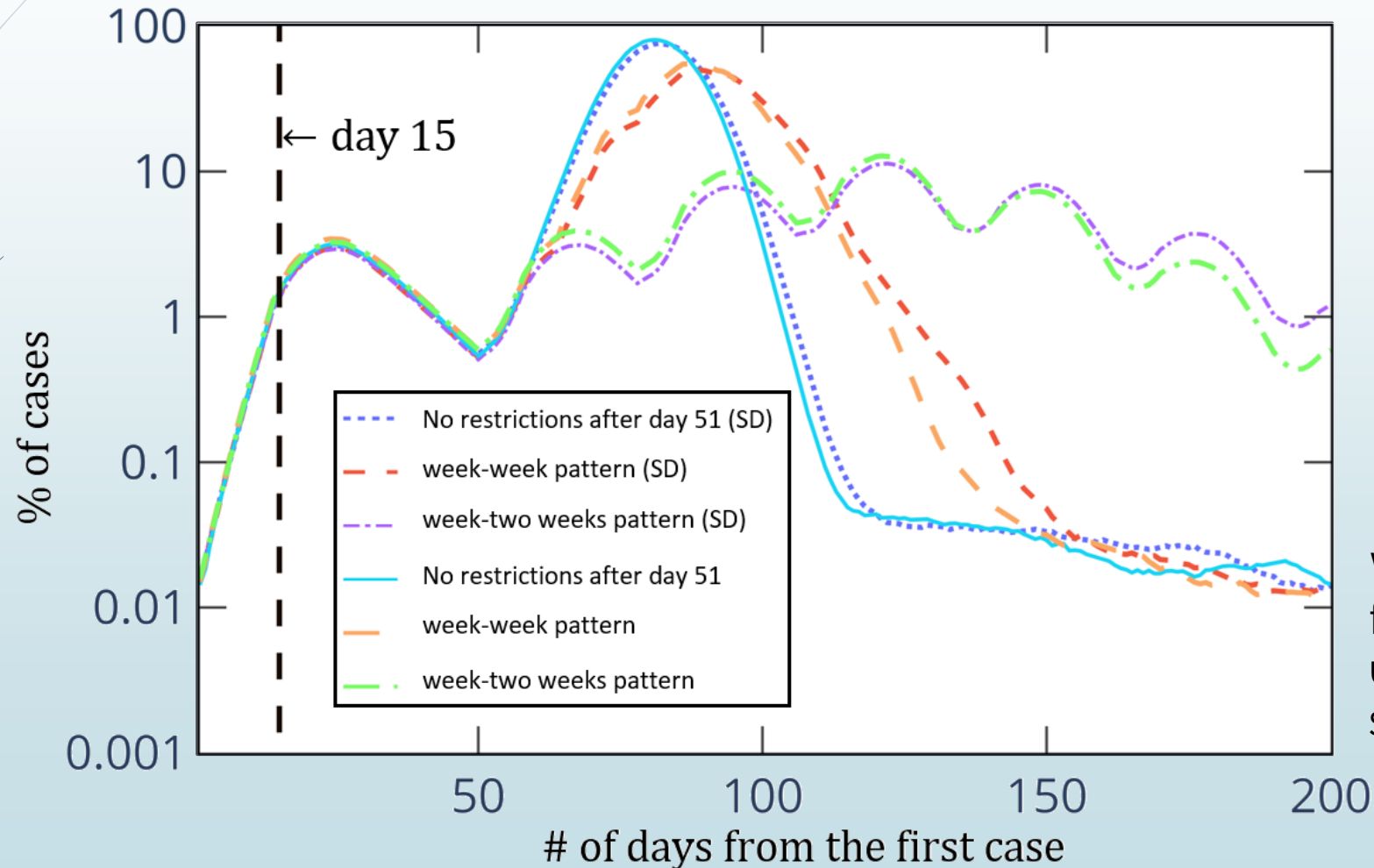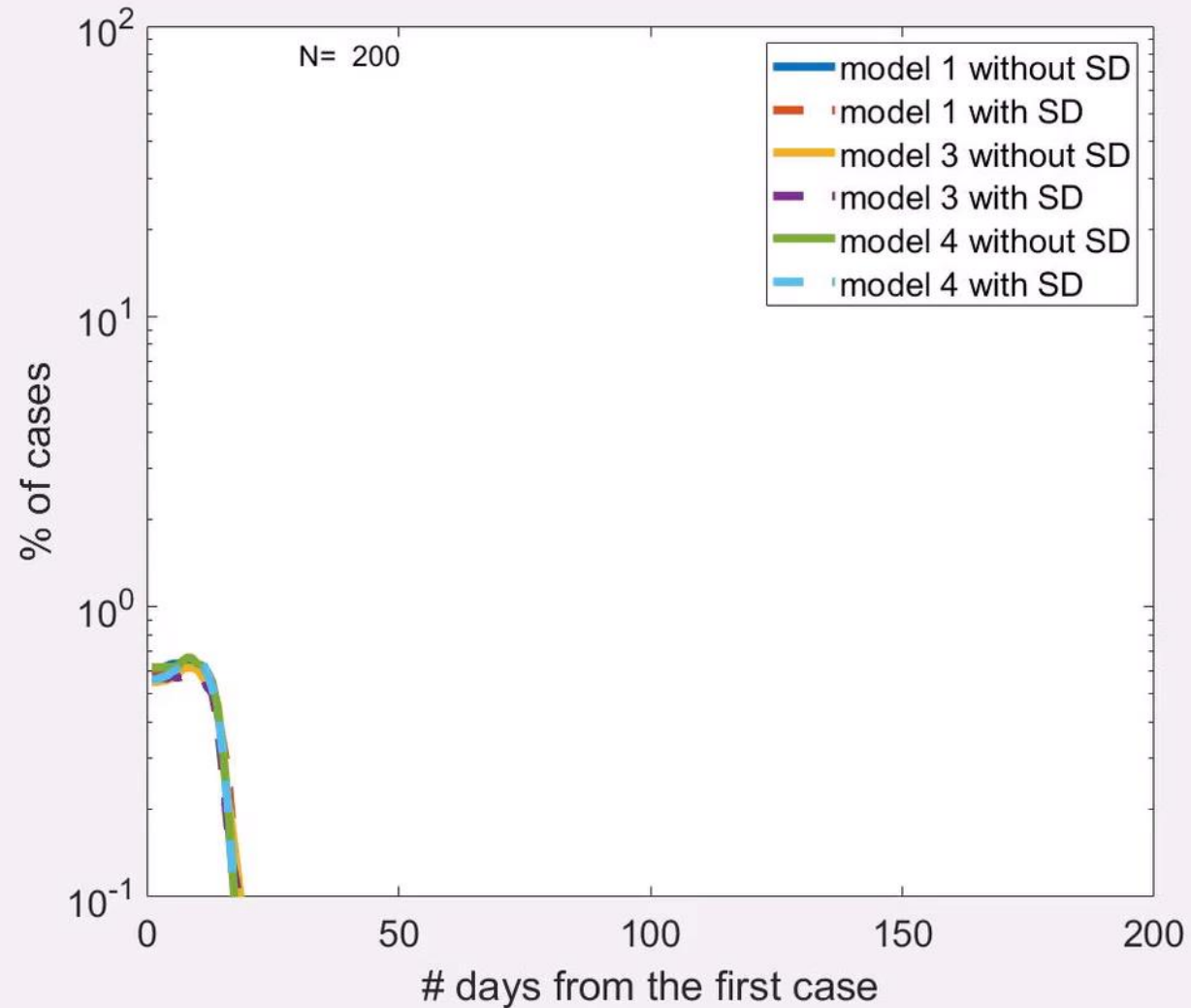
With infections from an unknown sources

Results: $R_0(1-15) = 4$

We check different models for N=9000, which is associated with $R_0 = 4$ (for the first 14 days).

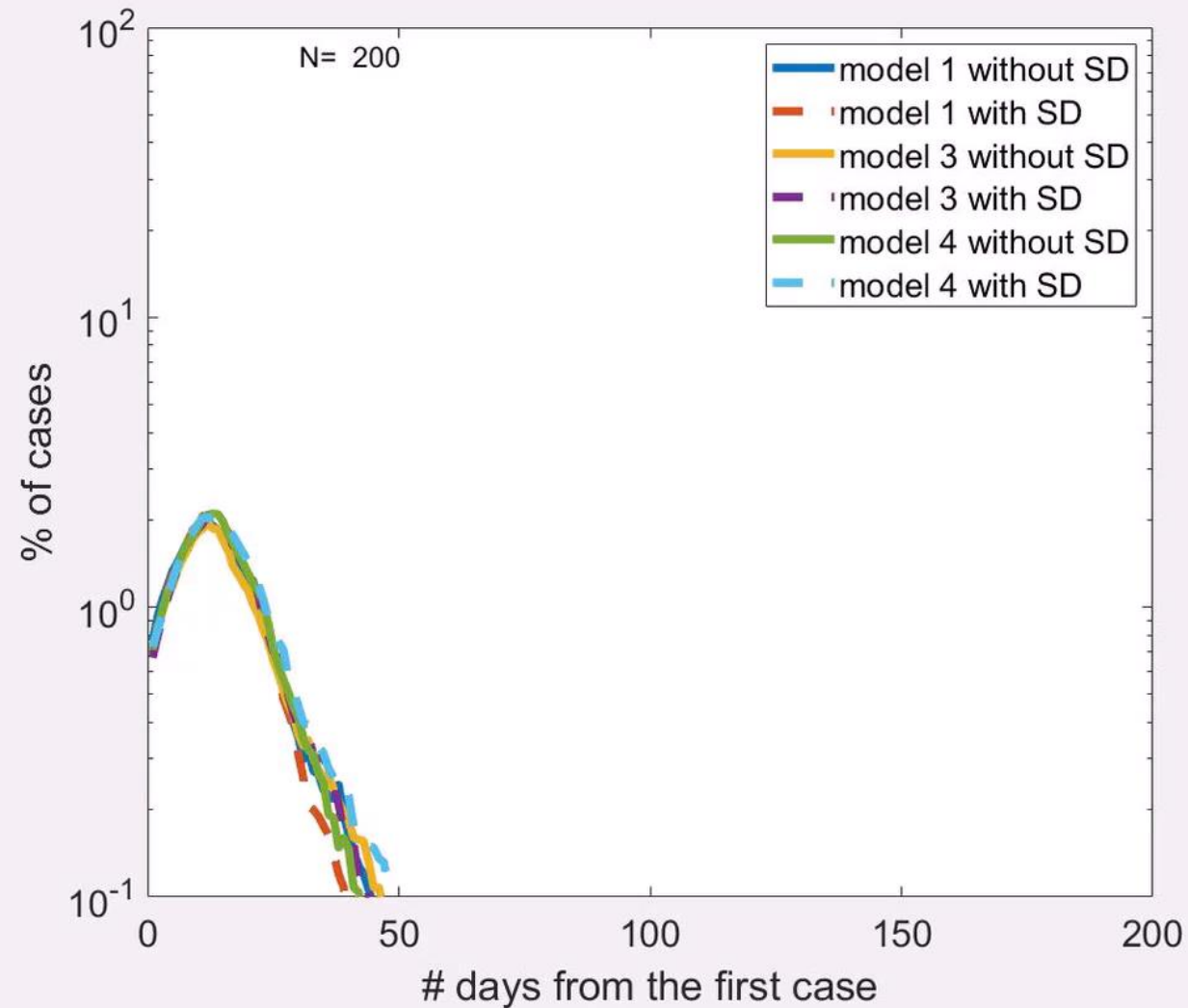With infections from an unknown sources

Results: all R_0

Without infections from an unknown sources

Results: all R_0

With infections from an unknown sources

Discussion and conclusions

- Our Monte-Carlo algorithm enables us to model different scenarios of the infection rate of the novel Coronavirus.

- The main advantage of our model is R_0 is obtained from the simulation and not pre-assumed.

- Although it is challenging to model the specific characters of each Coronavirus infected area, we have noticed that social distancing and a week-two weeks pattern will help keeping the infection rate under control.

Discussion and conclusions

- The main task of the decision makers is to lower the "kinetic energy" of each area as possible by keeping social distancing and minimizing the high "kinetic energy" regions.

- High "kinetic energy" regions are: schools, playground, kindergartens, houses of prayer etc.