

SIR Simulation

Lorenzo Manini

Nicolò Montalti

A.A. 2019/2020

1 Descrizione del progetto

2 Compilazione ed esecuzione

```
cmake ..  
make
```

3 Risultati

L'applicazione restituisce due tipi di output grafici. Nella prima finestra vengono mostrate le persone, schematizzate come cerchi colorati, che si muovono in uno spazio quadrato. Lo stato delle persone è indicato con colori diversi: verde per i sani, rosso per gli infettivi, arancione se il virus è in incubazione, bianco per la quarantena e blu per i recuperati. Un'immagine d'esempio è mostrata in fig. 1. Contemporaneamente, in una seconda finestra, viene mostrato un grafico aggiornato in tempo reale in cui viene plottato il numero di sani, infetti e recuperati.

Di seguito vengono riportati gli esiti di alcune simulazioni effettuate variando i parametri delle classi *Infection* e *Motion*. Se non diversamente indicato, i parametri dello stato iniziale sono $size = 600$, $S = 400$, $I = 10$ e $R = 0$. Si è scelto di iniziare la simulazione con 10 individui infettivi per velocizzarne l'esecuzione. Inoltre si è notato che con un solo infetto può capitare che questo guarisca prima di infettare altre persone. Quest'ultimo caso, sebbene possibile, è stato ritenuto di scarso interesse.

La classe *Motion* è stata inizializzata con una deviazione standard pari a 0.2. Si è notato che variare questo parametro influisce poco sulla simulazione. L'unica differenza apprezzabile è nella durata dell'epidemia, che diminuisce all'aumentare della varianza.

La classe *Simple Infection* è stata inizializzata con una distanza critica di 10, una probabilità di infezione di 0.05 e un tempo di recupero di media 200 e deviazione standard 50. Inoltre, alla variante *Incubation Infection*, si sono assegnati un tempo di incubazione di 50 e una probabilità di essere costretti alla quarantena di 0.005.

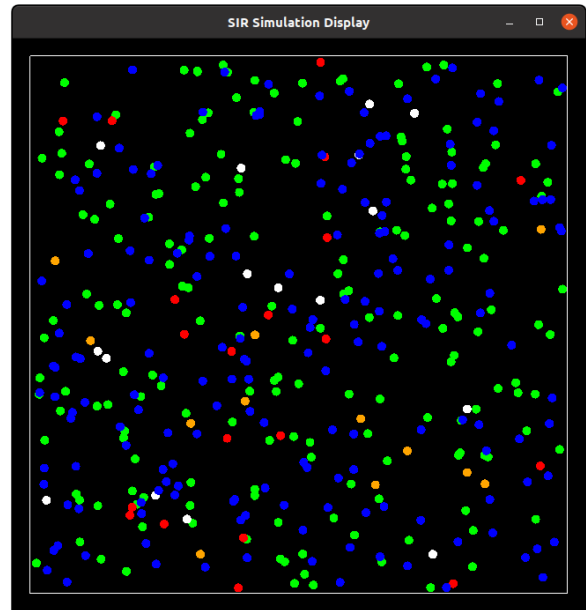


Figura 1: Finestra grafica ottenuta dalla classe *Display*. Il verde corrisponde ai sani, il rosso agli infettivi, l'arancione al virus in incubazione, il bianco alla quarantena e il blu ai guariti

L'esito di una simulazione con i parametri sopra descritti e la classe *Simple Infection* a gestire l'infezione è riportato in fig. 2a. In fig. 2b si può vedere come aumentando la probabilità di infezione a 0.08 il picco si alzi sensibilmente. Diminuire la *size* da 600 a 400, come mostrato in fig- 3a, determina un'epidemia molto più violenta, con un picco alto e la totalità della popolazione che contrae il virus. Aumentarla a 800 (fig. 3b), simulando una sorta di distanziamento sociale, provoca invece l'effetto opposto. Introducendo un periodo di incubazione si ottiene il grafico in fig. 4a, in cui si può notare come l'intensità dell'epidemia sia più modesta. Infine, aggiungendo la possibilità di essere costretti alla quarantena, si ottiene il grafico in fig. 4b, in cui il numero di infetti è costantemente sotto controllo e il numero finale di individui che non contraggono il virus consistente.

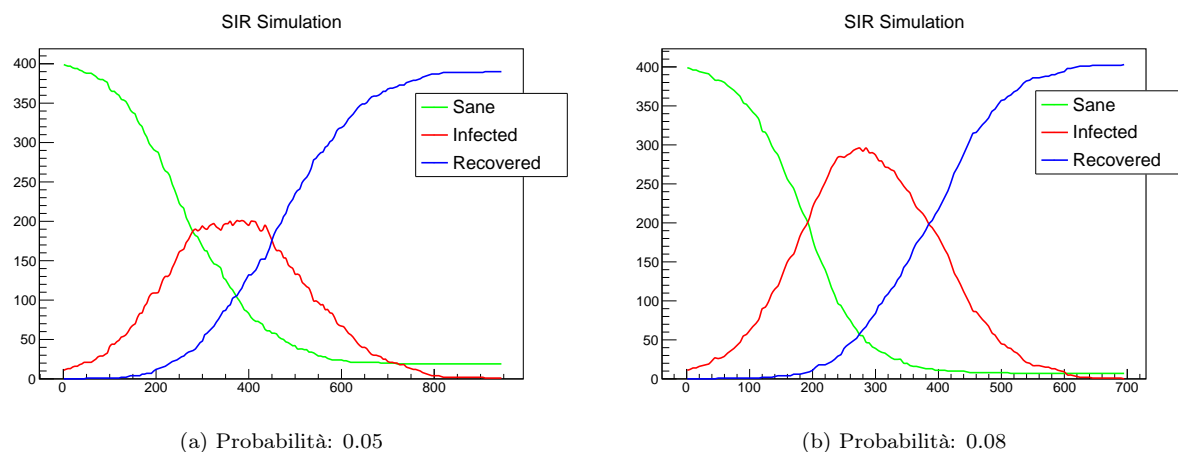


Figura 2: Grafici di una simulazione con *Simple Infection* a gestire l'infezione. I parametri della classe differiscono solo per la probabilità di infettarsi venendo a contatto con un individuo infettivo

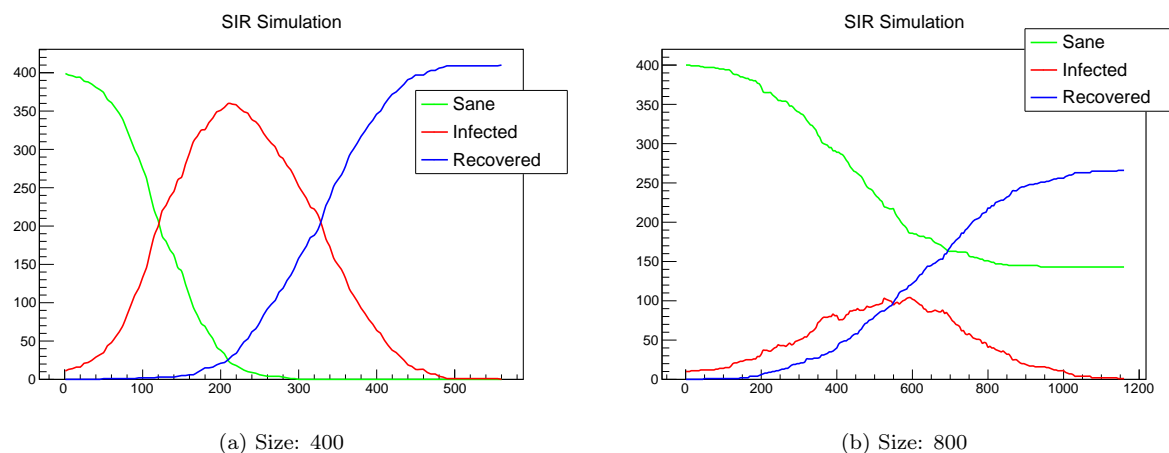


Figura 3: Grafici di una simulazione con *Simple Infection* a gestire l'infezione. I parametri della classe differiscono solo per la dimensione dello spazio a disposizione

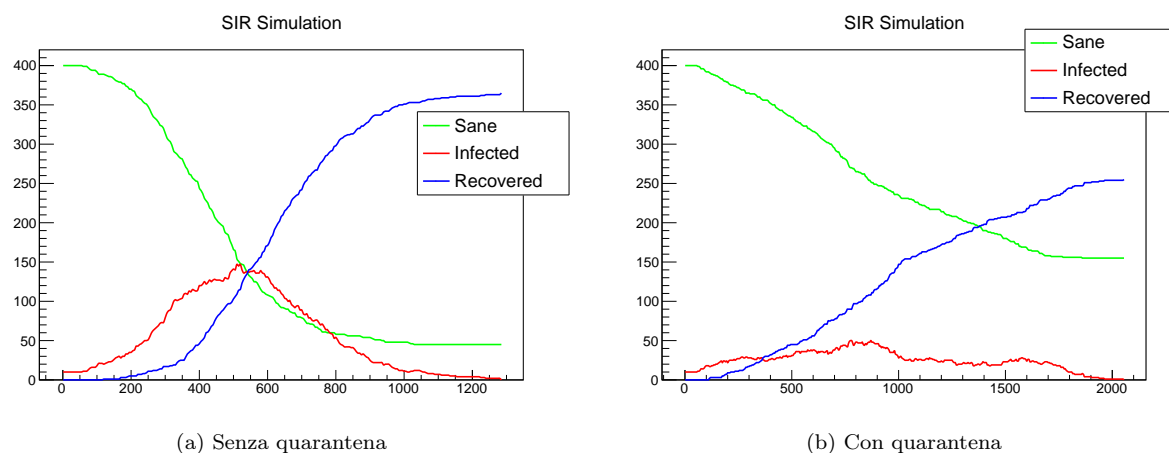


Figura 4: Grafici di una simulazione con *Incubation Infection* a gestire l'infezione. I parametri della classe sono gli stessi, ma nel secondo si è aggiunta la possibilità di essere costretti alla quarantena

4 Strategia di testing

Per testare la correttezza del programma si è utilizzato *Doctest*. Si è scelto di testare esplicitamente solo le classi *Motion* e *Infection*, dato che costituiscono il cuore del programma e sono le più suscettibili ad errori. Per le classi *Display* e *Plot*, che gestiscono l'output grafico, ci si è limitati a verificare il corretto funzionamento durante l'esecuzione del programma. Tutti i file utili al testing sono contenuti nella directory `/tests`.

Motion Per testare la classe *Motion*, si è creata un'istanza con deviazione standard nulla, così da poter prevedere lo spostamento delle persone. Si è poi generata una popolazione di un solo individuo e si è verificato che la posizione venisse aggiornata correttamente, tenendo conto della velocità iniziale, della presenza dell'attrito e degli urti con le pareti. In seguito si è creata una seconda istanza di *Motion* con deviazione standard non nulla e si è controllato che le medie delle posizioni, velocità e accelerazioni della popolazione fossero compatibili con una distribuzione delle accelerazioni di media zero.

Infection Le due classi che gestiscono l'infezione sono state testate con popolazioni ridotte. Gli individui sono stati distanziati in modo da poter prevedere i contatti e i parametri probabilistici delle classi sono stati impostati in modo da generare eventi certi o impossibili. Si è poi verificato che i contagi avvenissero come previsto, controllando sia la composizione dei vettori sia il *Sub Status* degli individui.