

Modello SIR

Andrea Farina, Matteo Martelli

a.a. 2021-2022

1 Introduzione

Il progetto simula l'andamento dei contagi di una pandemia tramite le equazioni del modello SIR:

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{S}{N} I \\ \frac{dI}{dt} &= \beta \frac{S}{N} I - \gamma I \\ \frac{dR}{dt} &= \gamma I\end{aligned}$$

Tale modello permette di simulare la variazione dei suscettibili all'infezione (S), la variazione degli infetti (I) e la variazione dei rimossi (R) durante lo scorrere dei giorni tramite i parametri γ e β . I suscettibili sono le persone sane che non hanno contratto la malattia e possono essere infettate, gli infetti sono le persone malate che possono contagiare le persone suscettibili e i rimossi sono la parte di popolazione che dopo essersi infettata è o guarita o deceduta. I rimossi non possono essere ricontagiati. $\beta \in [0, 1]$ è una misura della probabilità di contagio in seguito a contatti tra infetti e suscettibili, $\gamma \in [0, 1]$ è la probabilità che ha un infetto di guarire (o morire) dalla malattia. Il progetto include funzionalità grafiche che permettono di visualizzare l'andamento dell'epidemia simulata tramite immagine dove vengono rappresentati gli andamenti dei parametri della pandemia.

2 Compilazione ed esecuzione del programma

2.1 Compilazione ed esecuzione del modello SIR

Prima di compilare il programma accertarsi di aver aggiornato le librerie di Ubuntu. Accedere tramite il comando "cd" nella directory del progetto. Per la compilazione eseguire il comando:

- `g++ -Wall -Wextra -fsanitize=address main.cpp epidemic.cpp`

Per poter eseguire il programma, immettere il comando:

- ./a.out

e seguire le indicazioni sul terminale per completare l'esecuzione. Verrà chiesto di fornire valori per i parametri epidemici γ e β , per la durata della simulazione t e per i valori iniziali S I R. Infine verrà chiesto all'utilizzatore del programma di selezionare come desidera visualizzare i dati, tra stampa su terminale e/o stampa su file di testo.

2.2 Parte grafica

Se è stata sfruttata la porzione di programma che permette di creare un file di testo, è possibile graficare in un file png l'andamento dei suscettibili, infetti e rimossi. Per permettere il disegno del grafico è necessario avere installato Gnuplot su terminale, possiamo installarlo immettendo i seguenti comandi da terminale:

- `sudo apt-get install gnuplot`
- `sudo apt-get install libgnuplot-iostream-dev`

Per visualizzare i dati generati dal programma bisogna eseguire il comando:

- `gnuplot graph.gnu`

il comando appena utilizzato permette di eseguire Gnuplot nel file `graph.gnu`.

3 Funzionamento del programma

La logica del programma è suddivisa in alcuni file.

In particolare i file "epidemic.hpp" e "epidemic.cpp" contengono le variabili e implementano i metodi che simulano lo sviluppo dell'epidemia, mentre il file "printing.hpp" contiene le funzioni che governano la visualizzazione dei dati.

Il file "epidemic.hpp" definisce la struct `EpidemicPar`, che contiene i parametri dell'epidemia β e γ , e la classe `EpidemicSIR`, che contiene i dati iniziali delle variabili epidemiche e le dichiarazioni dei metodi che ne governano l'evoluzione e che ritornano le variabili. Infine sono qui presenti le funzioni che permettono di chiedere le variabili al terminale.

Il file "epidemic.cpp" contiene l'implementazione di alcuni metodi della classe `EpidemicSIR`, in particolare il costruttore e il metodo `Evolve`.

- Il costruttore riceve come input i parametri e i valori SIR iniziali, controlla che abbiamo valori accettabili e inizializza le variabili private dell'oggetto.
- Il metodo "Evolve" simula l'evoluzione dell'epidemia tramite le equazioni differenziali del modello SIR utilizzando un delta t pari a un giorno. Alla fine di ogni giorno di simulazione approssima i valori di SIR a interi e copia questi valori all'interno di vettori appropriati, che conterranno informazioni sull'intera evoluzione dell'epidemia.

Il file "printing.hpp" contiene le funzioni che governano la stampa dei dati su terminale e su file di testo. Queste funzioni prendono come argomento un oggetto della classe "EpidemicSIR", accedono ai vettori che contengono l'evoluzione di S I e R e li stampano a schermo e/o su file di testo.

Il file "main.cpp" utilizza la logica della classe e delle funzioni viste per chiedere i valori dei parametri all'utilizzatore, creare un oggetto della classe con questi parametri e fargli simulare l'epidemia. Infine utilizza le funzioni di "printing.hpp" per stampare i dati dell'oggetto creato.

Il file "graph.gnu" contiene le istruzioni necessarie per la cura dell'estetica del grafico generato e utilizza il file di testo "datiSir.txt" precedentemente creato per rappresentare in un file "SIRmodel.png" l'andamento dei contagi simulati dal programma. Rieseguendo Gnuplot con nuovi dati, il grafico verrà sovrascritto andando a così perdere la generazione precedente.

3.1 Strategia di testing

Per testare il corretto funzionamento del programma abbiamo utilizzato il file "doctest.h" per creare il documento "epidemic.test.cpp". Questo programma controlla che la classe EpidemicSIR si comporti come desideriamo in determinate situazioni.

Abbiamo prima controllato che il programma si interrompesse correttamente qualora gli fornissimo dei dati non accettabili. Per esempio abbiamo controllato che β e γ fossero compresi tra zero e uno e che i valori di S, I e R fossero non negativi. Poi abbiamo creato alcuni casi con dati corretti e abbiamo controllato che il valore di $S+I+R$ si conservasse e che i valori ottenuti dopo l'evoluzione fossero pari a quelli attesi.

Per utilizzare il file di testing bisogna compilare "epidemic.test.cpp" tramite il comando da terminale:

- `g++ -Wall -Wextra -fsanitize=address epidemic.test.cpp epidemic.cpp`

E poi eseguire il programma eseguendo il comando:

- `./a.out`

Se i test osservano il comportamento corretto comparirà sul terminale una lista con il numero di test fatti e passati durante l'esecuzione.

4 Interpretazione dei risultati

Per verificare che il nostro programma lavorasse in maniera corretta, abbiamo confrontato il grafico generato dal nostro progetto (Fig. 1) con un grafico generato tramite modello SIR (Fig. 2). Possiamo quindi notare che c'è una analogia tra l'andamento dei suscettibili, infetti e rimossi.

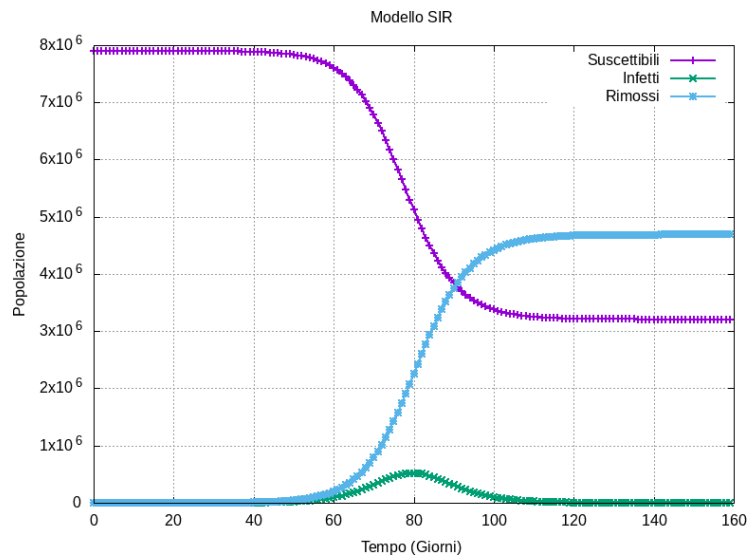


Figure 1: Grafico generato dalla simulazione del nostro programma

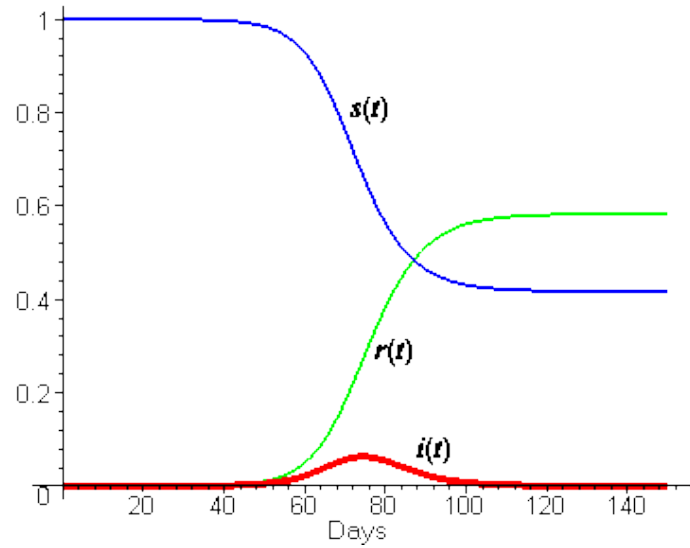


Figure 2: Simulazione modello SIR