Esame programmazione per la fisica

Pierbattisti Goffredo

Colombo Giordano

# Introduzione:

Il progetto che verrà presentato consiste nell’implementazione di un modello SIR in un software sviluppato utilizzando come linguaggio di programmazione C++ e alcune librerie del framework ROOT sviluppato dal CERN che ci consentono di generare un grafico in cui viene riprodotto l’andamento temporale dei parametri del modello SIR in esame.

Analizzeremo in questo elaborato le scelte implementative utilizzate in ogni unità di implementazione ed intestazione.

Volendo mantenere separate la logica con le parti competenti al puro linguaggio C++ e quelle che utilizzano ROOT, abbiamo deciso di dividere il programma nei seguenti file di intestazione ed implementazione:

# main.cpp

# sirlibrary.cpp - sirlibrary.hpp

# rootsir.cpp - rootsir.hpp

Iniziamo l’analisi delle scelte di implementazione e di architettura software.

# main.cpp

Le due librerie, sirlibrary.hpp e rootsir.hpp, sono state inserite entrambe nel main anche se interdipendenti in quanto per l’architettura del nostro software abbiamo optato per la possibilità di scorporare la parte contenente le librerie di ROOT, ovvero deputata al grafico del modello e quella contenente la pura dinamica del modello SIR.

Nel main troviamo la parte di codice scritta per accettare da standard input i parametri iniziali del modello SIR, in particolare vengono richiesti: 1) i parametri beta ed alfa che rappresentano le probabilità normalizzate di infezione e di guarigione, 2) suscettibili, infetti e rimossi al tempo t = 0, 3) la durata in giorni del modello.

Inoltre il main si occupa del controllo della validità dei dati iniziali, ovvero vogliamo che sia S, I, R e T (durata pandemia) siano numeri interi positivi, e inoltre richiediamo che alfa e beta siano due valori double compresi tra 0 e 1.

Infine al main spetta il compito di chiamare le funzioni che generano il modello ed il grafico del nostro modello.

# sirlibrary.cpp - sirlibrary.hpp

# GIORDS COMPLETA QUI

# rootsir.cpp - rootsir.hpp

In questi due file vengono incluse tutte le librerie root necessarie alla generazione di grafici tramite ROOT, inoltre viene inclusa la libreria sirlibrary.hpp in quanto abbiamo optato per l’utilizzo dell’ereditarietà tra classi del C++.

In particolare, viene generato un oggetto della classe SIRGraph che in pratica non è altro che un’estensione con metodi di ROOT della classe SIRModel presente nel file sirlibrary.hpp fatta per rendere possibile un accesso diretto tramite gli attributi protected ai membri della classe madre, in modo da avere una classe figlia totalmente equivalente a quella di partenza, con in più l’estensione fornita dal framework di ROOT.

Questo sempre in spirito della totale indipendenza della parte che descrive la dinamica del modello da quella che ne permette la visualizzazione su canvas ROOT.

Questo fa si che per noi accedere ai membri del vettore<SIR> per inserirli nel grafico è molto semplice in quanto vengono passati direttamente dalla classe madre alla classe figlia e li possiamo estrarre con un semplice ciclo for che accederà ai vari membri del vettore record\_ tramite il formalismo della struct, ovvero record\_.S, record\_.I e record\_.R