



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di Laurea Magistrale in Informatica

## AGENT-BASED MODELING AND LEARNING FOR EPIDEMIOLOGICAL STUDIES

**Relatore:** Prof. Antoniotti Marco

**Correlatore:** Prof.

**Relazione della prova finale di:**

Matteo Stievano

Matricola 829535

June 18, 2023

**Anno Accademico 2022-2023**

### **Abstract**

L'abstract, che probabilmente dovrà essere presentato in doppia lingua (ITA, ENG) rappresenta un breve sunto di tutto l'elaborato di cui fa da cappello. Ovviamente non dovrà entrare nel dettaglio, e anzi dovrà essere scorrevole e chiaro, senza soffermarsi troppo su alcun dettaglio, in quanto ci saranno le sezioni successive per espandere tali conoscenze. Ad occhio e croce potrebbe prendere al massimo mezza pagina (così che se necessario ci stiano le due versioni su una pagina).

# Contents

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Stato dell'Arte</b>	<b>4</b>
2.1	Epidemiologia . . . . .	4
2.2	Causalità . . . . .	6
2.3	Modelli Compartmentali . . . . .	7
2.3.1	Derivazioni del modello SIR . . . . .	9
2.3.2	Modelli Deterministici . . . . .	13
2.3.3	Modelli Stocastici . . . . .	16
2.4	Modelli ad Agente . . . . .	17
2.4.1	Discretizzazione . . . . .	19
2.5	Julia . . . . .	21
2.5.1	Agents.jl . . . . .	21
2.5.2	SciML.ai . . . . .	25
2.5.3	SafeBlues . . . . .	29
<b>3</b>	<b>Metodi e Modelli</b>	<b>31</b>
3.1	Modello ad Agente . . . . .	31
3.2	Monitoraggio e Intervento . . . . .	44
3.2.1	Predizione . . . . .	44
3.2.2	Intervento . . . . .	46
<b>4</b>	<b>Risultati Ottenuti</b>	<b>47</b>
<b>5</b>	<b>Sviluppi Futuri</b>	<b>48</b>
5.1	Altre epidemie . . . . .	48
5.1.1	Ebola . . . . .	48
5.1.2	Aviaria . . . . .	48
5.1.3	Influenza . . . . .	48
5.2	Miglioramento integrazione SciML.ai . . . . .	48
5.3	Accuratezza nella descrizione del modello . . . . .	48
5.3.1	Happiness . . . . .	48
5.3.2	Interventi localizzati . . . . .	48
5.3.3	Migliore flessibilità generale . . . . .	48
<b>6</b>	<b>Conclusioni</b>	<b>49</b>

# 1 Introduzione

L'utilizzo di metodi e tecniche sempre più sofisticati da parte della comunità scientifica, in particolar modo da parte di epidemiologi e medici, è sempre stato argomento di grande dibattito e interesse. Negli ultimi anni il mondo si è espanso esponenzialmente divenendo sempre più connesso, incrementando drasticamente la probabilità che un virus affligga a livello mondiale la popolazione creando un disastro senza precedenti.

La storia dell'uomo è costellata di epidemie, e solamente alcune tra loro si sono guadagnate il privilegio di essere ricordate e tra queste solamente una piccola parte ha ottenuto il primato di essere ricordata come una catastrofe. Forse è proprio questo che le ha rese così salde nell'immaginario comune aumentando la loro già imponente aurea di terrore.

Per fare degli esempi possiamo citare: la peste nera [71] che a partire dalla metà del 14esimo secolo ritornò in Europa uccidendo venti milioni di persone in soli sei anni, l'epidemia di tifo [79] che non solo fu fatale durante il periodo delle crociate, ma anche durante la seconda guerra mondiale all'interno dei campi di concentramento nazisti, oppure le varie influenze, prima tra tutte quella spagnola [66] la quale nel periodo del primo dopo guerra, ovvero tra il 1918 e il 1920, uccise 50 milioni di persone in tutto il mondo, oppure l'epidemia di AIDS [78], che ha all'attivo dal 1981 oltre 75 milioni di casi e 35 milioni di morti.

Se ci si soffrema un attimo pensando proprio a questo tipo di pandemia, quella influenzale, ci viene da tirare un sospiro di sollievo, in quanto oramai come cittadini del primo mondo l'idea di influenza non ci fa più così paura. Eppure dovrebbe, e sfortunatamente lo abbiamo ricordato nel peggior modo possibile.

La pandemia di SARS-CoV-2 [70] scoppiata negli ultimi mesi del 2019 condizionò l'intera umanità per circa 3 anni, e nel momento che sto scrivendo queste righe continua a condizionarla. Questa pandemia si è macchiata di aver mietuto, allo stato attuale delle cose, quasi 7 milioni di vite accertate. Questa tragedia rimarrà impressa nella memoria umana in quanto capace di aver messo in crisi l'intero sistema governativo mondiale, creando uno stato di allarme, panico e alle volte perfino isteria, che pochi altri avvenimenti sono stati in grado di fare.

Osservando le statistiche proprie di questa epidemia, ciò che balza subito all'occhio è sicuramente il numero associato alle morti e agli infetti: quasi 7 milioni di morti e più di 700 milioni di infetti [41]. Numeri che da soli basterebbero a mettere a disagio qualsiasi lettore, eppure altri dati, nascosti a prima vista, possono fornire ulteriori macabre informazioni. Un esempio tra tutti è l'effetto che un'epidemia del genere ha avuto sull'economia [25], portando disagi generalizzati ovunque.

Legato al disagio economico vi è un altro dato preoccupante definito come poverty trap [9] [60]. Questo fenomeno nasce in quegli ambienti in cui le condizioni di povertà economica e la prevalenza di malattie possono imprigionare una società in uno stato persistente di bassa sanità e sempre maggiore povertà; questo fenomeno ciclico si auto sostiene e prende il nome di positive feedback [72]. Per ultimo, ma non per questo meno importante, l'effetto di una pandemia può portare ad instabilità governative le quali possono portare a un arretramento del sistema sanitario e di welfare [16], ricadendo come sopra descritto all'interno del fenomeno di positive feedback.

Questi sono solamente alcuni esempi dei problemi che possono sorgere, e che sono sorti con lo scoppio di una pandemia globale come è stata quella del COVID-19. Per questo la comunità scientifica, in

particolare gli epidemiologi cercano soluzioni sempre più efficaci e accurate per prevenire, arginare e contrastare avvenimenti del genere.

L'epidemiologia è una disciplina nata di recente evolutasi insieme alle esigenze della società ogni qualvolta una nuova emergenza sanitaria faceva irruzione nella quotidianità. La prima definizione di epidemiologia è stata data da Lilienfeld [19] nel 1978, e cita:

*l'epidemiologia è un modo di ragionare riguardo le malattie, e si occupa di effettuare inferenza biologica derivata dall'osservazione di fenomeni patologici all'interno di una popolazione.*

Con il tempo questa definizione ha subito molti cambiamenti, derivati anche e soprattutto dall'espansione degli ambiti relazionati all'epidemiologia; con l'aggiunta ad esempio della farmacoepidemiologia, dell'epidemiologia molecolare e dell'epidemiologia genetica. Non solo, ambiti come etica, filosofia ed epistemologia sono estremamente importanti ed influenti nella crescita e sviluppo di questa materia [19].

Attualmente con il termine epidemiologia si intende la disciplina biomedica che studia la distribuzione e la frequenza delle malattie ed eventi di rilevanza sanitaria nella popolazione. L'epidemiologia si occupa di analizzare le cause, il decorso e le conseguenze delle malattie [62]. Secondo Last et al (1998) l'epidemiologia è definita come:

*lo studio della distribuzione e dei determinanti delle situazioni o degli eventi collegati alla salute in una specifica popolazione, e l'applicazione di questo studio al controllo dei problemi di salute.*

Uno degli strumenti più utilizzati in epidemiologia sono le simulazioni software [74]. Dato lo scopo dell'epidemiologia, questa necessita di avvalersi di modelli matematici [50] che aiutano i ricercatori a trarre conclusioni sul sistema che analizza. Sistemi simili vengono definiti come complessi, [20] [33] ovvero sistemi dinamici a multicomponenti che tipicamente interagiscono tra loro, e che sono descrivibili tramite modelli matematici.

I primi modelli chiamati compartmentali [8] basano il proprio funzionamento sullo studio di gruppi di individui disgiunti che interagiscono tra loro, analizzabili tramite un sistema di equazioni ordinarie differenziali (ODE) [10]. Questo approccio è stato teorizzato da Kermack e McKendrick nel 1927 applicando una modellazione matematica al comportamento delle malattie infettive su un gruppo di individui, tenendo in considerazione la variabile del tempo. Da qui è nato il famoso modello Susceptible-Infectious-Recovered (SIR) [58], il quale tuttora viene utilizzato ampiamente insieme alle sue varianti. Successivamente con l'unione di svariate discipline come: la teoria dei giochi, i sistemi complessi, i comportamenti emergenti, la sociologia computazionale, i sistemi multiagente e la programmazione evoluzionaria sono nati i modelli ad agente [54] [7], modelli computazionali autonomi per la simulazione di sistemi complessi.

Grazie alla loro teorizzazione negli anni 1940 ma soprattutto grazie al loro uso concreto dagli anni 1990 il mondo della simulazione ha avuto un grande balzo in avanti. Ciò che rende questi sistemi molto flessibili e potenti è la capacità di far emergere spontaneamente dei comportamenti complessi da un insieme di regole semplici. Ovviamente però la richiesta di risorse e capacità computazionale è estremamente elevata rispetto alla controparte di modelli puramente matematici.

Ogni approccio ha dei pro e dei contro e la vera potenza di questi modelli sta principalmente nella acutezza di chi deve poi farne uso. In base ai compromessi e le assunzioni fatte durante la fase

di modellazione ogni approccio può rivelarsi vincente. Uno dei compromessi maggiori che viene generalmente applicato a questi modelli è quello della discretizzazione dell'ambiente [32]. La realtà è a tutti gli effetti un insieme continuo di eventi, ma essendo tutti i dispositivi di calcolo discreti è impossibile simulare avvenimenti continui (siano essi nel tempo o nello spazio) in maniera diretta e perciò bisogna fare dei compromessi, trasformando il proprio spazio di lavoro in uno più adatto alle macchine che lo devono simulare.

Un'altra assunzione generalista e approssimativa per definizione, ma necessaria per la costruzione di un modello che analizzi il decorso di un'epidemia all'interno di una società, è quella legata alla tipologia di comportamento che verrà mostrato dagli esseri umani in condizioni di pericolo [52] [17]. Queste sono solo alcune delle assunzioni e compromessi che bisogna fare quando ci si approccia al mondo della simulazione. Tuttavia applicare delle assunzioni, alle volte anche forti e controiduttive, non sempre è sinonimo di errore. Alle volte tramite lo studio della causalità degli eventi [20] [43] è possibile astrarre un set minimale di assunzioni che se applicate danno la capacità al modello di rappresentare molto bene il comportamento desiderato.

Certamente ci saranno alcune discrepanze soprattutto in casi estremi, ma è un'eventualità che viene tenuta in considerazione ogni qualvolta si parla di simulazione, e che non è possibile eliminare del tutto.

*La causalità o causa effetto, è quell'influenza per cui un evento, un processo, uno stato o un oggetto contribuiscono nella produzione di un nuovo evento, processo, stato o effetto, dove la causa è parzialmente responsabile dell'effetto e l'effetto è parzialmente dipendente dalla causa.*

A prima vista non sembra una tematica molto complessa o di difficile approccio, complice il fatto che in quanto esseri umani siamo una specie che si è evoluta per trovare una correlazione tra gli eventi, ma correlazione non significa, e soprattutto non implica, causalità [4]. Questo tema si riferisce all'incapacità legittima di dedurre la relazione di causa - effetto tra due eventi o variabili solamente sulla base di una osservazione della loro associatività o correlazione.

Diventa chiaro come una delle parti più complesse dell'epidemiologia sia proprio quella di stabilire le cause di un dato fenomeno e comprendere come un determinato intervento su di esse influenzi quest'ultimo [20] [43].

Questa rapida introduzione all'epidemiologia e alla simulazione di sistemi complessi tramite l'utilizzo in particolare di modelli ad agente sarà l'argomento cardine dell'intero elaborato. Nelle sezioni successive verrà proposta un'analisi dello stato dell'arte dell'attuale panorama epidemiologico e simulativo, con alcune rapide digressioni sul problema della discretizzazione, problema assai sentito nell'ambito della simulazione.

Successivamente verranno portati alla luce alcuni esempi e modelli di sistemi ad agente che si occupano di modellare differenti tipi di epidemie, tutte caratterizzate però dal fatto di essere epidemie infettive a diffusione principalmente aerea, come ad esempio l'ebola, l'aviaria e l'influenza.

Successivamente vi sarà un'analisi più dettagliata riguardo la pandemia da COVID-19 che recentemente ci ha colpito. Infine verranno analizzati alcuni metodi di monitoraggio di queste simulazioni con un focus su alcuni metodi di intervento.

## 2 Stato dell'Arte

### 2.1 Epidemiologia

L'epidemiologia è la disciplina biomedica che studia la distribuzione e la frequenza delle malattie ed eventi di rilevanza sanitaria nella popolazione [62]. Si occupa di analizzare le cause, il decorso e le conseguenze delle malattie.[20] [43]

L'epidemiologia è una disciplina molto pratica, che visto l'obiettivo che si pone, ovvero quello di trovare le cause relative ad un dato effetto, non può esentarsi dagli svariati problemi che gravitano e definiscono questo nobile obiettivo, primo tra tutti: cosa vuol dire che un evento è causa di un altro e come definisco questo tipo di rapporto in maniera inequivocabile?

Questi interrogativi possono sembrare banali in quanto come specie ci siamo evoluti per trovare una correlazione di causalità tra eventi anche quando questi non ne hanno. Ad esempio se fossimo in un bosco, al buio e soli con l'unico rumore ad accompagnarcici che è quello di una piacevole brezza estiva, se percepissimo un rumore tra i cespugli, molto probabilmente penseremmo che c'è qualcosa che non va, che ci sia un pericolo in agguato, un predatore, anche se magari la motivazione è la suddetta brezza.

Questo adattamento evolutivo ci ha permesso di sopravvivere in situazioni di pericolo, ma sfortunatamente quando si parla di scienza e di dati, non sempre l'istinto è qualcosa a cui affidarsi, in quanto i dati di per loro non dicono assolutamente nulla, siamo noi in quanto individui dotati di intelletto, tecniche e metodi a dover estrapolare dei significati che verifichiamo essere corretti e inequivocabili.

Se ci soffermiamo su questa ultima affermazione, possiamo essere facilmente tratti in inganno. Prendiamo per esempio il seguente grafico che mostra in maniera *inequivocabile* come la spesa da parte degli USA sulla ricerca aerospaziale sia direttamente collegata al numero di suicidi per strangolamento.

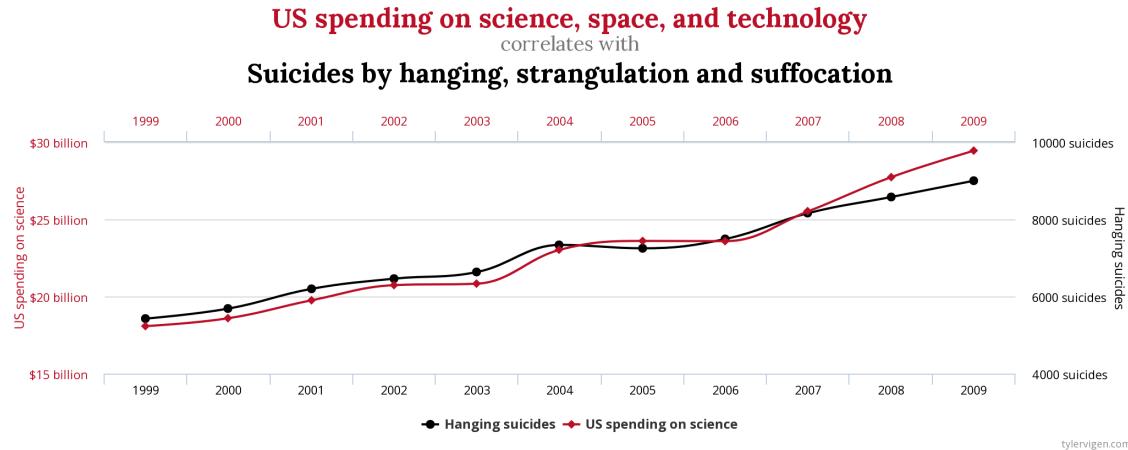


Figure 1: Esempio di correlazione spuria da <https://www.tylervigen.com/spurious-correlations>

Affidandoci solamente al grafico e ai dati riportati in figura, ci verrebbe da pensare che le due categorie siano in qualche modo correlate, e che il governo americano debba essere fermato, in quanto respondabile di incitamento al suicidio. Ebbene non è così, questo è un caso di **relazione spuria** [77], ovvero che due o più variabili sono associate ma non causalmente correlate.

Associatività e causalità infatti non sono la stessa cosa, ed è bene quando si studia l'una, non confondersi con l'altra. In statistica, una correlazione tra dati è una qualsiasi relazione che vi è tra due o più variabili, sia essa di tipo causale oppure non [59]. Nell'esempio di prima la correlazione tra le due variabili potrebbe essere banalmente il tempo, infatti con il passare del tempo la spesa media per la ricerca aerospaziale è continuata a salire per via del sempre più alto interesse e investimento nel settore e sfortunatamente con il passare degli anni si è registrato un aumento costante del numero di suicidi.

Il **bias** che abbiamo verso la ricerca di un intreccio tra gli eventi, in maniera che questi siano sempre contigui l'uno con l'altro, in maniera che si possa tracciare una chiara e distinta linea dal primo all'ultimo ci trae in inganno quando questi sembrano esserlo ma in realtà non lo sono. Molto spesso la risposta più semplice è anche quella meno interessante, benché corretta, ovvero che due eventi completamente slegati tra loro possono avere andamenti simili, così come differenti addendi possono portare lo stesso risultato;

Solamente perché cinque più cinque fa dieci non vuol dire che dieci sia il risultato solamente di cinque più cinque.

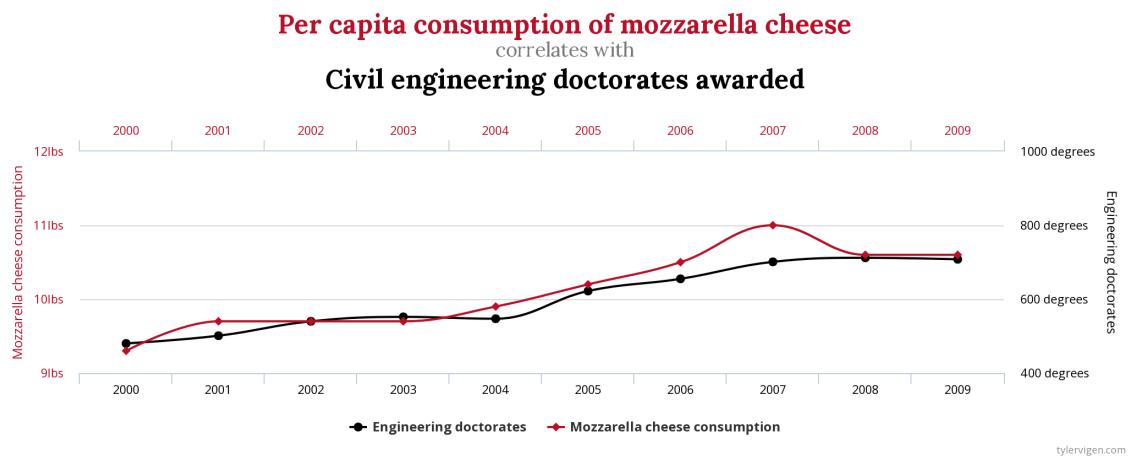


Figure 2: Un altro esempio di correlazione spuria da <https://www.tylervigen.com/spurious-correlations>

## 2.2 Causalità

Il problema della causalità non è da prendere sotto gamba ed è uno dei problemi cardine quando si vogliono determinare e applicare degli interventi all'interno di una popolazione per cercare di mitigare ad esempio la diffusione di un agente patogeno [43].

Come è stato già introdotto precedentemente, i dati di per loro non dicono nulla, siamo noi che dobbiamo imparare a capire il significato di ciò che esprimono. Un esempio estremamente calzante di come pur avendo bene in mente il problema sopra citato delle correlazioni *spurie*, si possa comunque essere ingannati dai dati, è il seguente:

Poniamo di essere un medico e di dover decidere se prescrivere o meno ad un paziente un determinato medicinale. Per aiutarci nella decisione abbiamo la storia clinica del paziente e i risultati di uno studio su una nuova medicina che attesta di curare il malessere del paziente. Questa nuova medicina è stata testata su un gruppo di settecento persone divise in due pari sottogruppi in cui 350 pazienti decidevano autonomamente se prendere o meno la medicina e 350 decidevano autonomamente il contrario. I risultati sono i seguenti:

	Drug			No Drug		
	patients	recovered	% recovered	patients	recovered	% recovered
<b>Men</b>	87	81	93%	270	234	87%
<b>Women</b>	263	192	73%	80	55	69%
<b>Combined data</b>	350	273	78%	350	289	83%

Figure 3: Risultati medicina

Questi risultati sembrano suggerire come la prescrizione di questa nuova medicina non aiuti i pazienti a stare meglio. Tuttavia questo risultato cade nel cosi detto *paradosso di Simpson* [73] per cui i dati aggregati relativi ad uno specifico trattamento sembrano descrivere una sua perdita di efficacia relativamente ai singoli dati delle singole categorie in esame, portando un lettore non attento a cadere nell'inganno di pensare che ci sia una perdita di efficacia. Questo esempio pone l'accento sul fatto che non sempre estrapolare informazioni da dei dati aggregati può risultare efficace, e anzi, alle volte questi possono ingannare. In questi casi bisogna estrapolare le informazioni di causalità dai dati singoli.

E' chiaro che non sia così semplice comprendere le cause di un determinato effetto, o insieme di effetti. Conoscere l'agente patogeno, o quanto meno la sua natura può aiutare, ma non sempre è sufficiente. L'utilizzo di modelli di Machine Learning per l'estrapolazione di dati, di correlazioni e successivamente per la definizione di policy di intervento può risultare in un rischio non indifferente ma al contempo descrive un potente alleato per la definizione di policy di intervento all'interno di settori estremamente delicati come quelli sanitari [49].

## 2.3 Modelli Compartmentali

In epidemiologia i modelli compartmentali sono una tecnica di modellezione generica che si pre-dispone molto bene allo studio complessivo del comportamento di una malattia infettiva [58]. Questa tecnica di modellazione si applica anche ad altre branche della scienza, come ad esempio la finanza.

Questa tecnica di modellazione matematica basa il proprio funzionamento sull'assunzione che, data una popolazione di individui, questi vengano etichettati in maniera differente, in base allo stato di progressione della malattia che hanno, o non hanno, contratto. Così facendo si vanno a definire dei compartimenti ben separati che possono interagire tra loro, ma che rimangono comunque chiaramente distinti l'uno dagli altri.

Il modello che tutt'ora viene usato come riferimento e come base per lo studio e modellazione è il così detto modello **Susceptible, Infectious, Recovered** (SIR):

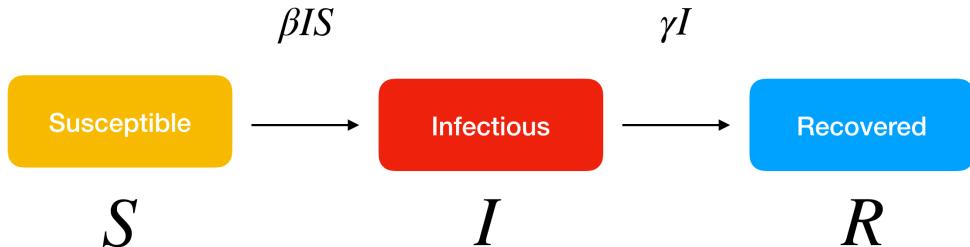


Figure 4: Struttura modello SIR

Questo modello è stato ideato all'inizio del 20esimo secolo, più precisamente nel 1917, da Kermack e McKendrick. Come introdotto questo modello si basa sull'assunzione che all'interno di una popolazione durante il decorso di una malattia vi possano esistere solamente tre stadi in cui un individuo può essere inserito:

- Susceptible: Questo stadio rappresenta lo stato iniziale per la maggior parte degli individui all'interno di una popolazione. Rappresenta il numero di persone che possono contrarre la malattia.
- Infectious: Questo stadio rappresenta tutti quegli individui che dallo stato di Susceptible, dopo essere venuti in contatto con un individuo infetto, diventano a loro volta individui infetti.
- Recovered: Questo stadio rappresenta una duplice categoria, quella degli individui che alla fine del docorso della malattia sopravvivono ad essa, e quelli che invece muoiono a causa di questa. Generalmente questo stato viene anche definito come Removed.

Da questa semplice idea poi si è andato a sviluppare un modello matematico per descrivere come queste 3 categorie separate ma che si influenzano vicendevolmente, cambiano nel corso del tempo. Questo approccio si basa sull'utilizzo di un sistema di Equazioni Ordinarie Differenziali (ODE) [10]. Una ODE è un equazione differenziale, ovvero un equazione che lega una funzione incognita

alle sue derivate, che coinvolge una funzione di una variabile e le sue derivate di ordine qualsiasi. Questo oggetto viene utilizzato estensivamente in molti ambiti della scienza e in epidemiologia viene utilizzato per descrivere un sistema dinamico [63].

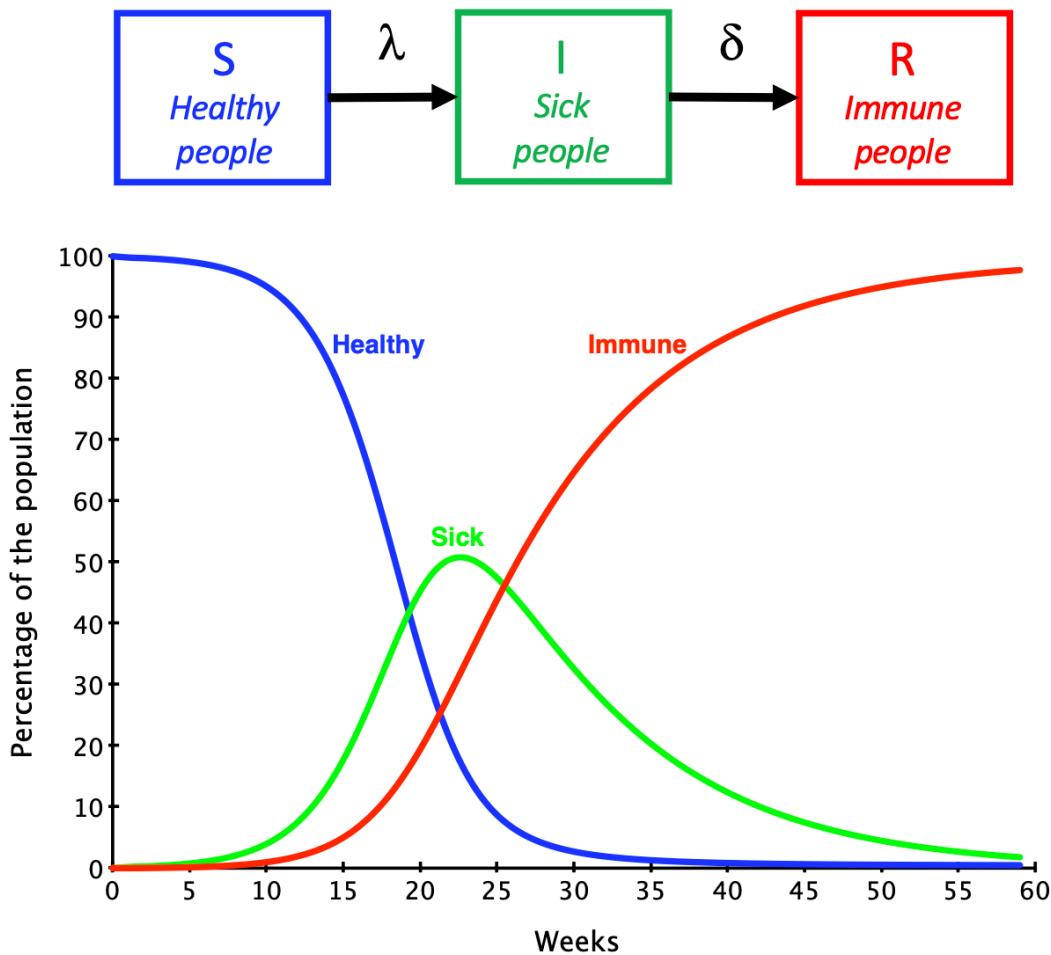


Figure 5: Visualizzazione grafica di un modello SIR

Come succede per la maggior parte di tutte le equazioni differenziali, queste non possono essere solitamente risolte in maniera esatta, e per questo ci si limita a studiarne il comportamento qualitativo della soluzione senza essere capaci di ottenere una'espressione analitica.

Nell'ambito epidemiologico tuttavia non sempre l'utilizzo di un sistema di ODE è preferito come metodo generale di modellazione di un sistema dinamico, in quanto il più delle volte questo sistema al suo interno utilizza un insieme di variabili rappresentanti processi stocastici, le quali è bene

mantenere tali. A questo scopo vengono utilizzati i sistemi di Equazioni Differenziali Stocastiche (SDE) [3].

Queste equazioni si basano sulla teoria del moto Browniano il quale descrive il movimento randomico delle particelle sospese all'interno di un medium [56]. In questo modo è possibile modellare in maniera più granulare ad esempio la diffusione di un agente patogeno tramite il medium aereo, come può essere il COVID-19.

### 2.3.1 Derivazioni del modello SIR

Con il tempo questo il modello SIR è stato espanso per tenere in considerazione comportamenti differenti sia della popolazione che delle malattie, andando a definire una moltitudine di modelli utili a differenti scopi. In epidemiologia il modello di riferimento maggiormente utilizzato è il modello SEIR (Susceptible, Exposed, Infectious, Recovered) con le sue varianti proprie di ogni approccio.

Questa tipologia di modello basa il proprio punto di forza sull'osservazione che da quando un individuo viene infettato tramite un agente patogeno a quando quest'ultimo diventa infettivo, passa un periodo di latenza in cui l'individuo non può ne infettare ne essere infettato. Questo periodo viene anche conosciuto come *periodo di incubazione* [65]. Con questa conoscenza pregressa è possibile sviluppare modelli e policy che tenendo conto di questo comportamento lo sfruttino per arginare l'epidemia.

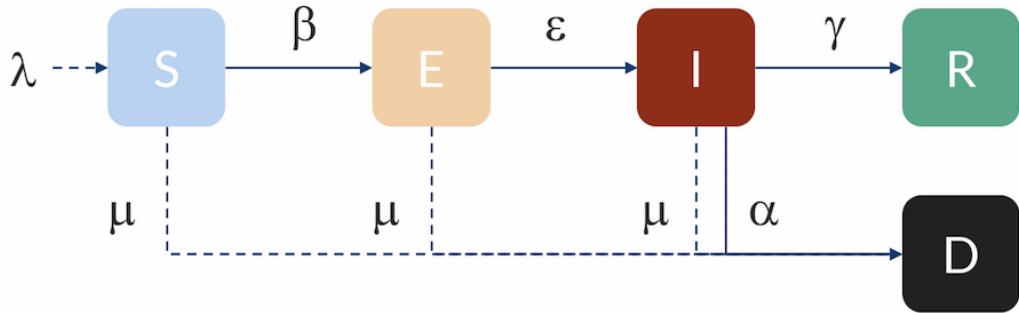


Figure 6: Modello schematico SEIR

Alcuni modelli definiscono i propri stati in maniera da considerare come stato interno al sistema anche l'agente patogeno, così da poter modellare e simulare l'andamento dell'infettività della pandemia in relazione alle contromisure prese, siano esse farmaceutiche o non. Ne è un esempio il modello proposto da [37] nel quale il modello viene proposto con l'idea di incorporare le misure di distanziamento sociale come variabili per misurare la loro efficacia contro la recente pandemia da COVID-19.

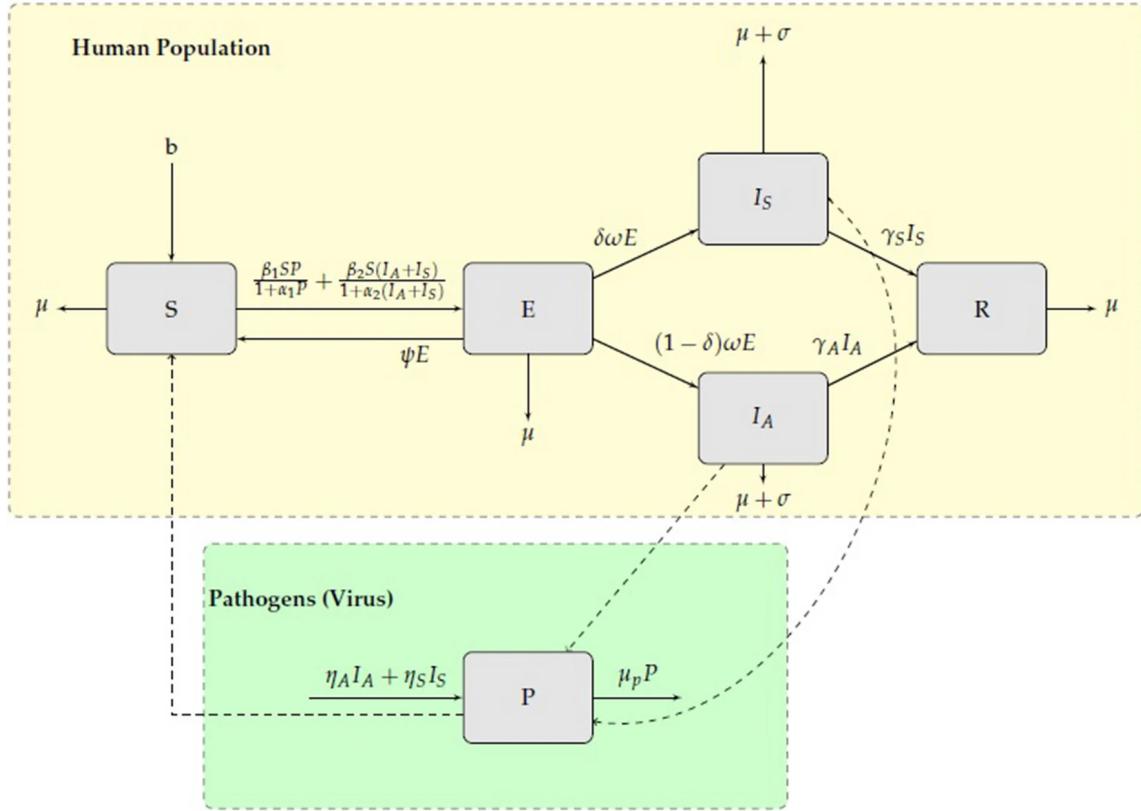


Figure 7: Esempio di modello SEIR preso dall'articolo [37]

Altri modelli, come quello proposto da [22] mantengono la stessa filosofia, ovvero quella di analizzare l'efficacia delle misure di prevenzione non farmaceutiche sull'andamento di un'epidemia, ma non modellano esplicitamente l'agente patogeno come stato del modello, bensì variando i parametri di infettività e contagio, arrivano allo stesso risultato. Un'altra differenza tra i due approcci è quella della tipologia di equazioni differenziali utilizzate, [37] hanno utilizzato delle ODE mentre [22] delle SDE.

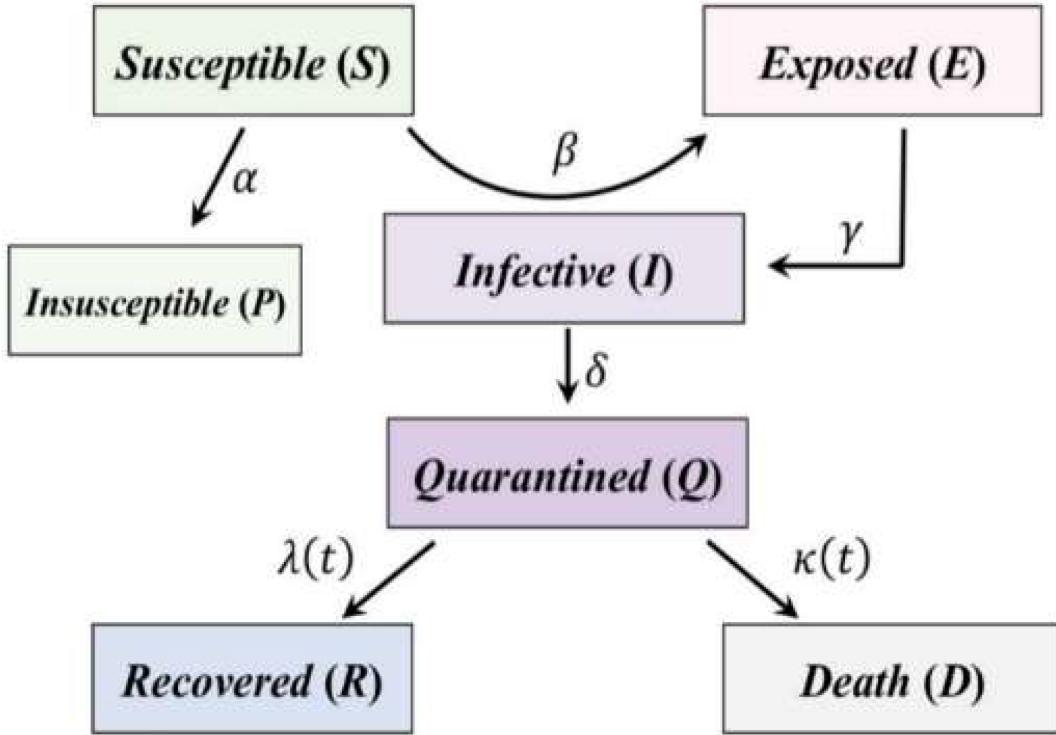


Figure 8: Esempio di modello SEIR preso dall'articolo [22]

Il motivo per cui viene utilizzato il modello SEIR come base è perché permette di modellare una caratteristica intrinseca di una malattia infettiva, ovvero il periodo di latenza che un individuo appena infettato ha prima di diventare infettivo a sua volta e mostrare i sintomi di infezione. Questo permette di osservare quanto le misure di sicurezza e prevenzione non farmaceutiche sono efficaci sulla popolazione tenendo in considerazione un tempo di ritardo intrinseco nel feedback tra l'attuamento delle misure di prevensione e i risultati positivi di queste ultime.

Una delle modifiche più utilizzate a questo modello è quella di avere un sistema ciclico, ovvero in cui gli individui che entrano nello stato R non diventano immuni alla malattia a tempo indefinito, ma perdono questa loro caratteristica di immunità dopo un dato periodo di tempo. Questo permette di modellare con più accuratezza le malattie infettive stagionali come ad esempio la comune influenza o il raffreddore, oppure mostrare l'andamento ad ondate di altre malattie che hanno la caratteristica di mutare molto velocemente, come è stato per il COVID-19 e le sue innumerevoli varianti.

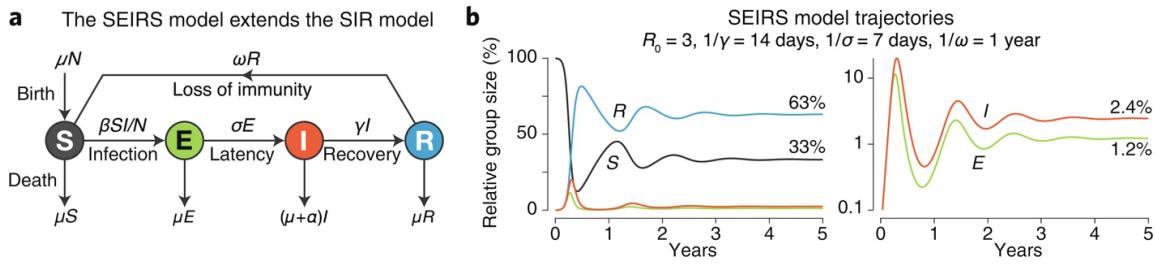


Figure 9: Modello SEIRS preso dall'articolo [8]

Questa variante denominata SEIRS permette invece di osservare, non solo l'andamento e l'efficacia delle contromisure non farmaceutiche, ma anche di quelle farmaceutiche, come ad esempio i vaccini; o più in generale l'andamento della così detta immunità di gregge [8].

Rimanendo sull'idea di voler analizzare l'efficacia di un vaccino, una modifica comune al modello SEIR è quella legata all'aggiunta dello stato V, Vaccinated, come stato esplicito oppure oppure implicito al modello. Questa variazione permette di modellare con più attenzione l'efficacia di un vaccino una volta introdotto all'interno della popolazione, ma più in generale permette di osservare l'efficacia di una politica di vaccinazione in relazione al numero di vaccinazioni effettuate in un determinato periodo di tempo. Questo viene solitamente affiancato con un modello ciclico, così da poter osservare come bisogna modificare le proprie politiche vaccinali in vista di ondate cicliche più o meno intense di infezioni.

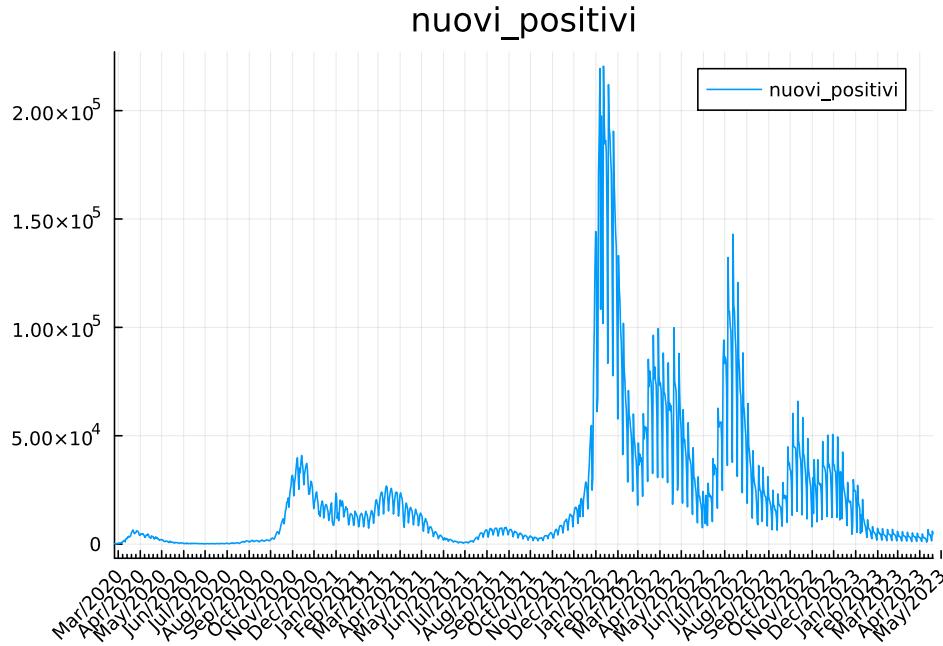


Figure 10: Esempio di ondate di infettività. Dati del Dipartimento di Protezione Civile Italiana

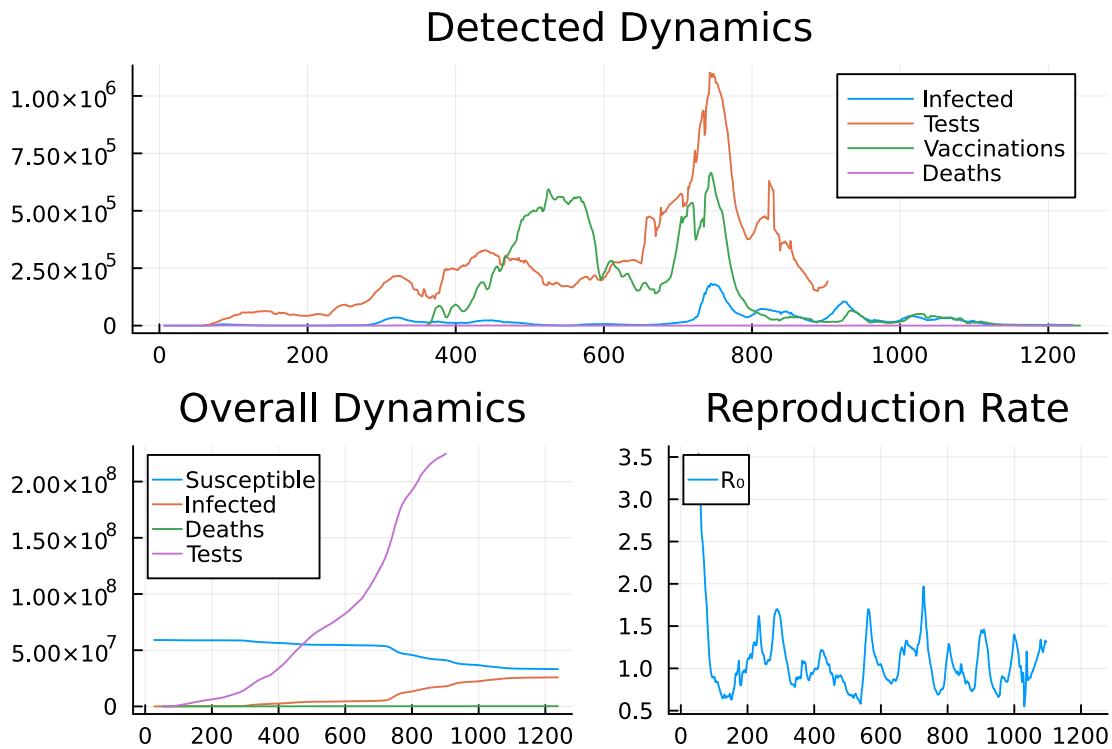


Figure 11: Grafico cumulativo dei dati della pandemia da COVID-19 in Italia. Dati ottenuti da Our World in Data

Ai fini pratici di una simulazione avere uno stato esplicitamente definito oppure ricavabile dalle probabilità di transizione degli altri stati è pressoché indifferente, e potrebbe essere richiesta una differenziazione solamente in caso in cui si avrebbe una differenza sostanziale tra lo stato  $R$  e  $V$ , ad esempio in termini di protezione dalla malattia, durata immunità etc.... .

Non essendoci un numero massimo di stati, e quindi di equazioni, utilizzabili all'interno del modello, ogni individuo è libero di definire un numero di equazioni arbitrario che rispecchia la sua idea di modellazione del sistema. Ne è un esempio il modello riportato in [21].

Come precedentemente introdotto esistono due grandi famiglie di modelli per la simulazione, e sono rispettivamente la famiglia di modelli deterministici e quella di modelli stocastici.

### 2.3.2 Modelli Deterministici

I modelli deterministici vengono principalmente utilizzati per la loro immediatezza e riproducibilità. Infatti un modello deterministico, una volta impostati i parametri necessari riprodurrà sempre lo stesso risultato. Questo tipo di approccio, seppur utilizzato in larga scala come ad esempio da [8] [37], [21] si basa su delle assunzioni molto forti che non sempre rispecchiano la realtà.

Infatti i modelli deterministici hanno il grosso problema di essere affidabili solamente nel caso in cui vi siano dati sufficientemente grandi, cosa che non sempre è possibile avere [58]. Essendo modelli

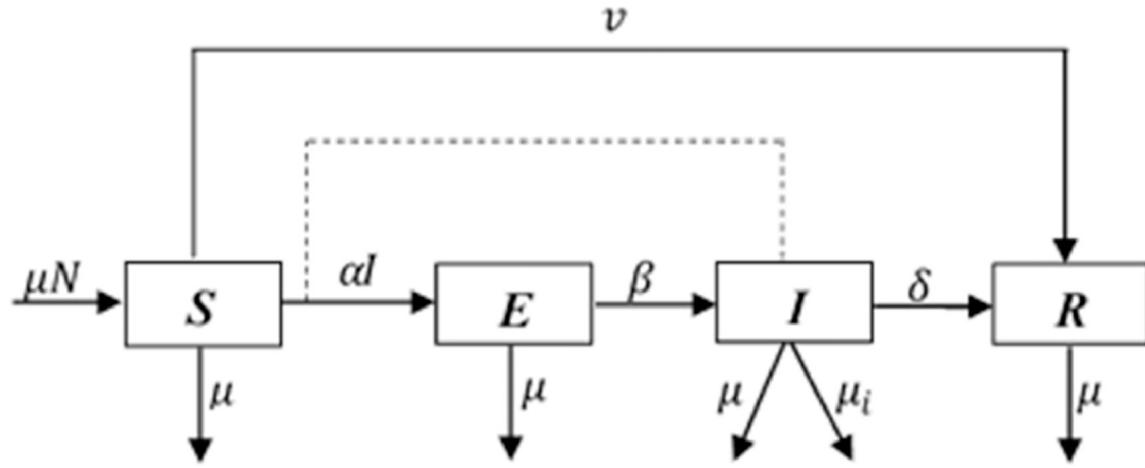


Figure 12: Esempio di modello SEIRV con stato隐式 per la condizione V

di tipo deterministico, avendo dei parametri di infettività maggiori di zero, con un numero di individui infetti anch'esso maggiore di zero, si tenderà ad avere nel lungo periodo un andamento di equilibrio endemico derivato dalle equazioni e dal modello utilizzato. Questo comportamento però non sempre rispetta la realtà, ma come precedentemente accennato, in casi in cui si hanno grandi quantità di dati legati principalmente alla popolazione, questi modelli si comportano in maniera affidabile.

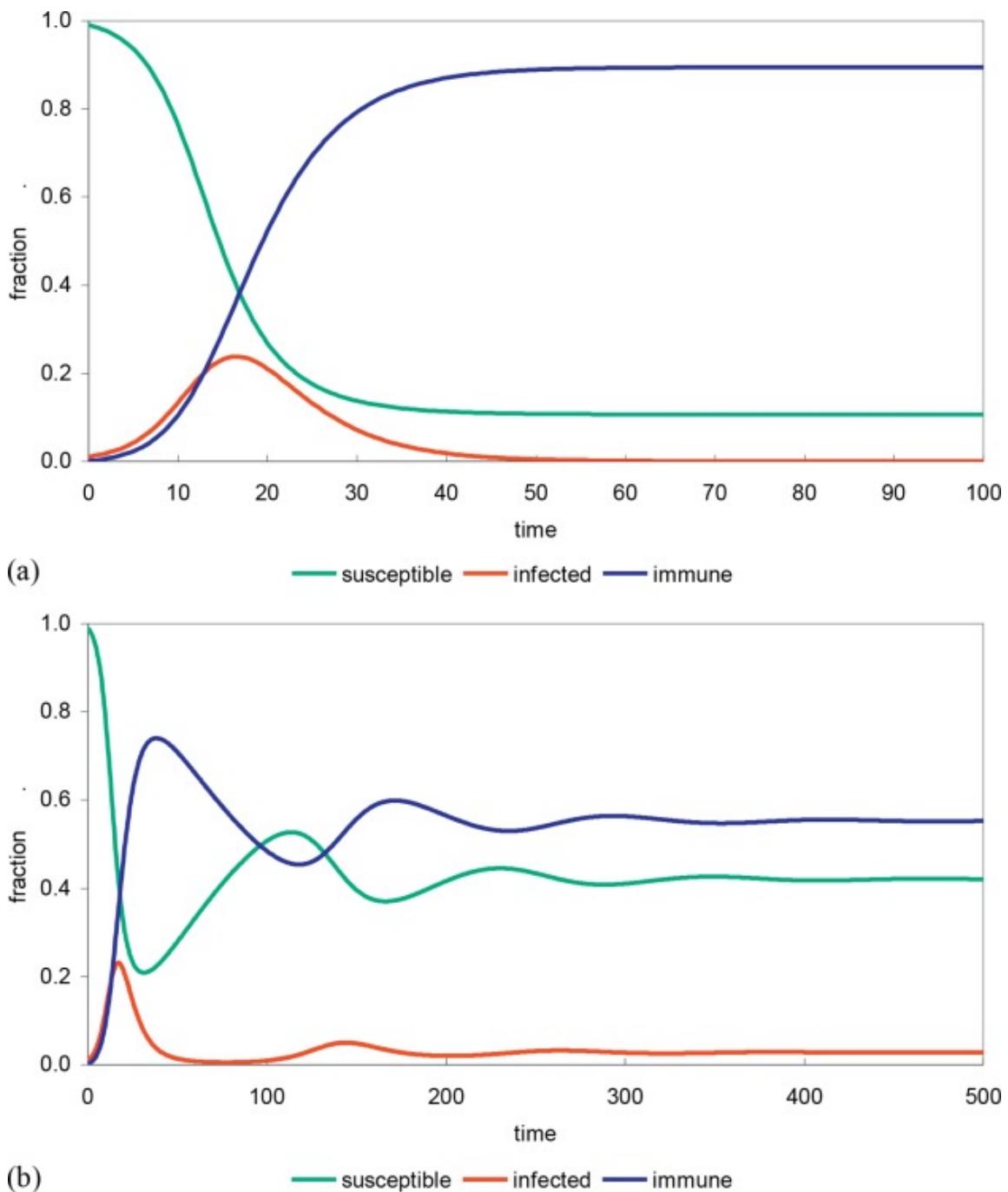


Figure 13: Esempio equilibrio endemico

### 2.3.3 Modelli Stocastici

I modelli stocastici, seppur più complessi e non determinabili a priori, permettono una modellazione più veritiera e simile alla realtà in quanto tengono in considerazione variazioni randomiche che possono capitare durante il decorso di una pandemia. Tuttavia questa loro caoticità richiede che per ottenere risultati robusti debbano essere eseguiti e computati molteplici volte, e la media dei loro risultati è il valore vero da tenere in considerazione. Questi modelli sono stati applicati durante la pandemia da COVID-19 come ad esempio da [22].

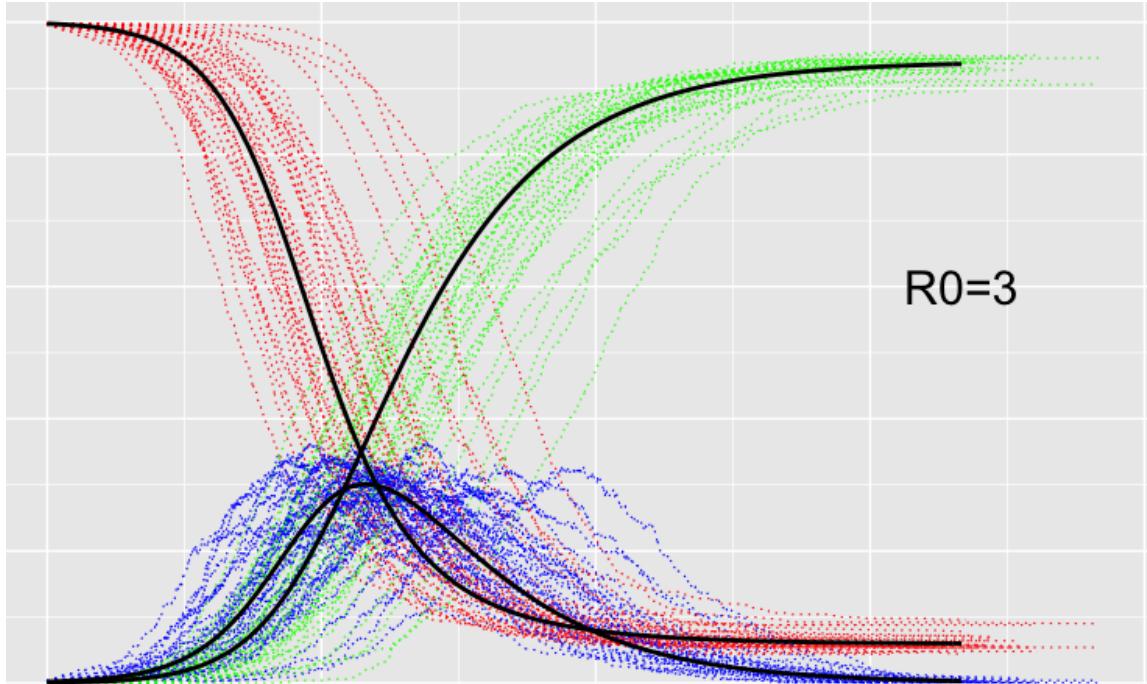


Figure 14: Modello SIR stocastico

E' immediato notare come il comportamento delle curve sia imprevedibile se preso singolarmente, e che non sempre esiste uno stato di equilibrio endemico chiaro e definito come quello ottenibile da un modello deterministico. Ciò nonostante effettuando molte simulazioni è possibile vedere come il comportamento generale del modello sia comunque simile a quello di un modello deterministico.

## 2.4 Modelli ad Agente

Un modello ad agente e' un modello computazionale per la simulazione delle azioni e interazioni di un insieme di agenti autonomi, siano essi individui o gruppi di individui, con l'obiettivo di comprendere il comportamento del sistema e la relazione che vige con i suoi risultati [55] [54].

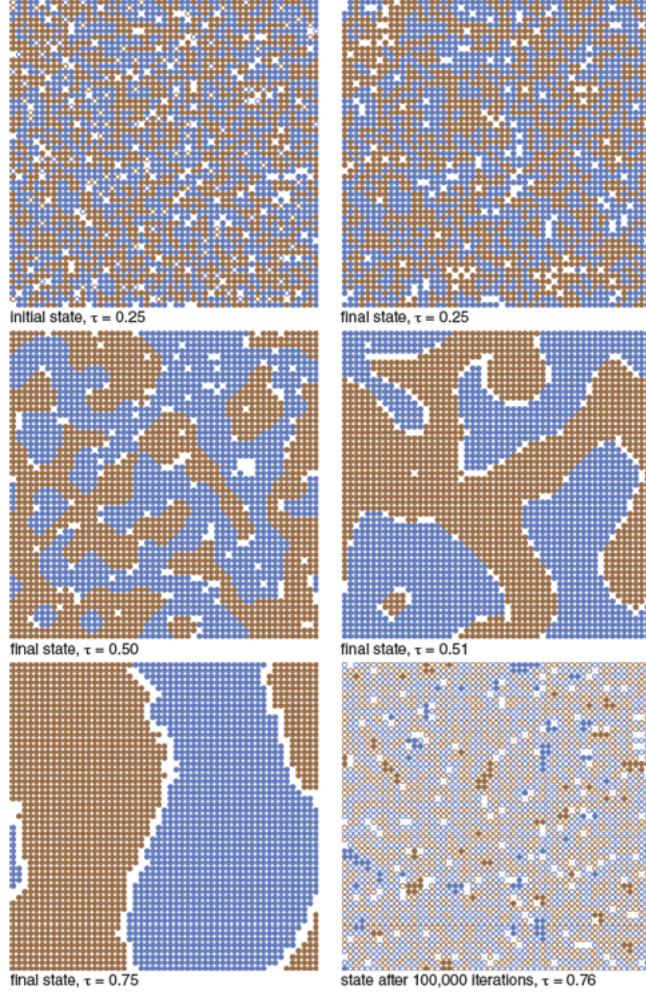


Figure 15: Schelling's model of segregation simulato tramite ABM

L'utilizzo di modelli ad agente in epidemiologi e' una tecnica nota che da piu' di un decennio viene impiegata per simulare e comprendere le situazioni e i problemi piu' disparati, tutti pero' generalmente accomunati dal fatto che come parametro portante vi sia il comportamento umano [24] [17] [52] [7]. Uno dei parametri piu' caratterizzanti che da sempre sono stati tenuti in considerazione e assunti come ristretti ad una piccola cerchia, sono le interazioni sociali tra individui. Questo sovrainsieme di parametri racchiude molteplici sottoinsiemi di parametri che descrivono delle interazioni piu' specifiche ma che possono essere raggruppate come macro categoria se si scende a

compromessi.

Questa specifica e' importante in quanto uno delle sfide piu' grandi che il mondo della simulazione, e quindi quello epidemiologico devono affrontare e' proprio quello di trovare un modo efficiente e soprattutto realistico di simulare le interazioni sociali tra individui, in quanto queste possono influenzare notevolmente i risultati di una simulazione definendola utile oppure inutile [51]. Non soltanto, un'altra sfida e' il modo con cui si decide di rappresentare lo spazio (e il tempo) all'interno della simulazione. In base al tipo di discretizzazione effettuata una simulazione potrebbe essere utile in un campo ma totalmente inutile in un altro [32].

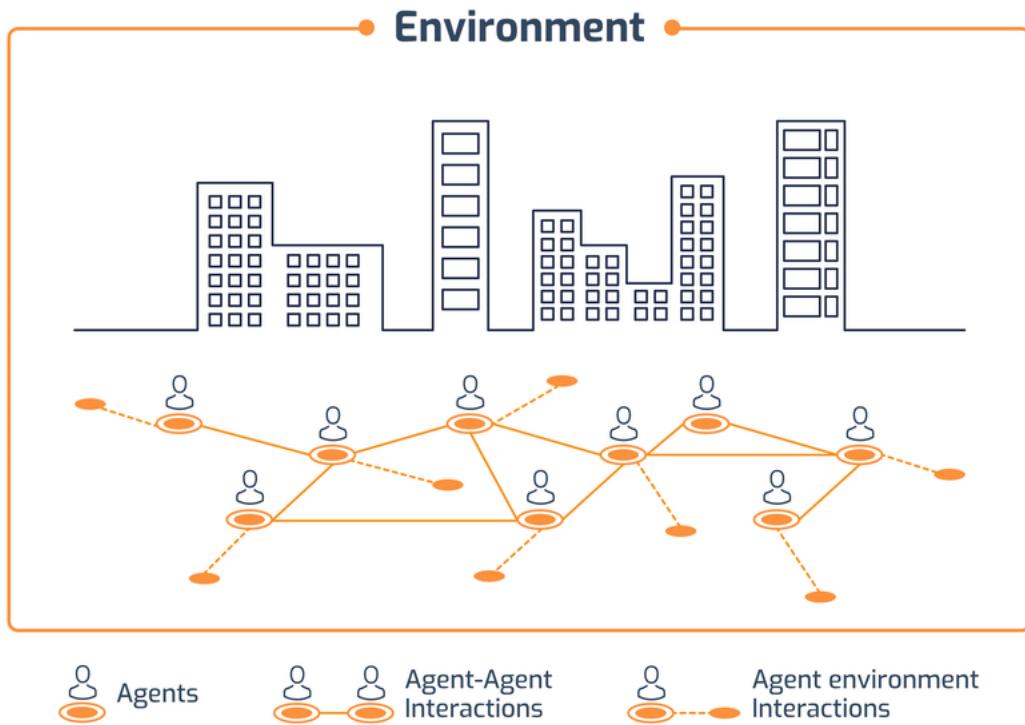


Figure 16: Rappresentazione schematica di un modello ad agente

Con l'arrivo della pandemia da COVID-19 molti ricercatori hanno focalizzato la propria attenzione sull'idea di sviluppare un modello ad agente puro oppure ibrido [36] (ibridato con delle ODE oppure delle Partial DE), con l'obiettivo di trovare un modello di simulazione in grado di simulare in maniera affidabile il decorso di una pandemia tenendo in considerazione le variabili piu' stocastiche e imprevedibili come il comportamento umano. L'ambiente di lavoro simulato era generalmente un ambiente controllato che poteva essere una citta' come in [44],

Un'altro obiettivo e' quello di osservare l'impatto degli interventi non farmaceutici sulla popolazione come riporta [40] [30]. Altri ancora invece utilizzano l'approccio simulativo tramite modelli ad agente per estrarre delle ODE tramite l'analisi del modello, come riportato da [38].

#### 2.4.1 Discretizzazione

La tematica della discretizzazione e' una delle proprietà fondamentali e al contempo uno dei problemi atavici della simulazione. Il mondo in cui viviamo e' un mondo continuo, ma gli strumenti che attualmente abbiamo per simularlo sono discreti, per cui ogni qualvolta che vogliamo simulare un evento dobbiamo decidere in che modo adattare la realta' alla simulazione, andando inequivocabilmente a perdere informazioni nel processo [32].

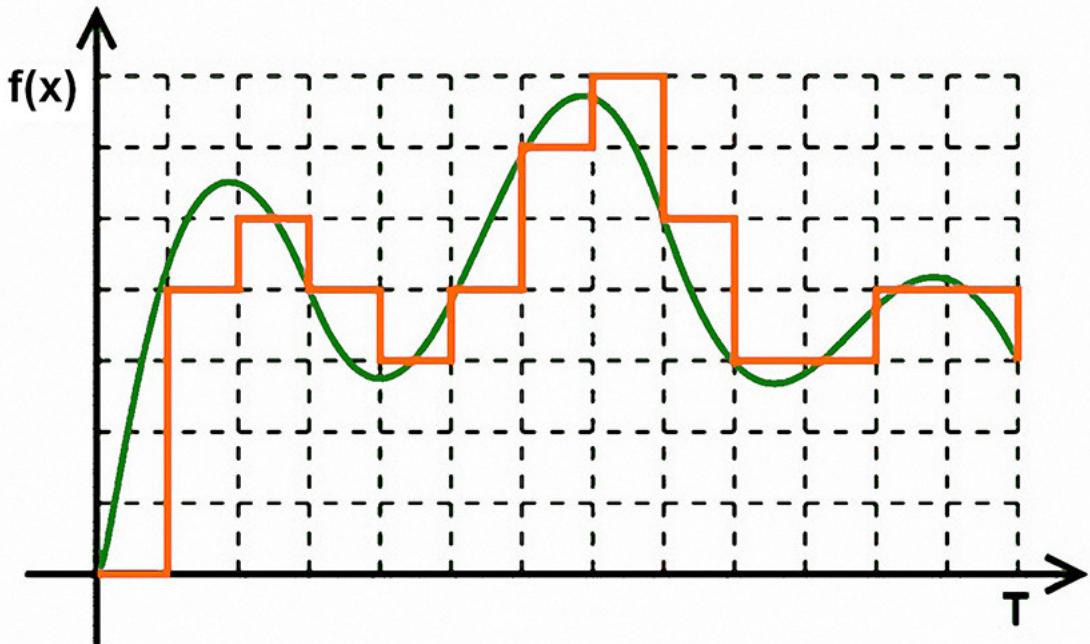


Figure 17: Esempio di discretizzazione dei dati

Il processo di discretizzazione in una simulazione puo' prendere principalmente due macro aree che sono: lo spazio e il tempo. Molti framework di simulazione implementano al loro interno diversi trucchetti per simulare in maniera quanto piu' accurata un insieme di dati continuo, permettendo all'utente di utilizzare parole chiave come ad esempio `ContinuousSpace`. Quello che praticamente viene fatto e' creare un ambiente per il modello che e' il piu' preciso e granulare possibile, per cui servono allo scopo fintanto che quest'ultimo non richiede una maggiore precisione.

Il processo di discretizzazione comunque non e' sempre negativo, in quanto alcuni problemi possono essere simulati in maniera estremamente fedele anche effettuando questi accorgimenti, e anzi alle volte non e' perfino necessario avere una precisione troppo alta per la simulazione di determinati eventi.

Se ad esempio si volesse simulare tramite un agente una partita a Risiko, non e' necessario richiedere uno spazio e un tempo continuo della simulazione, in quanto questi possono essere rimpiazzati dalla loro controparte discreta che sono caselle e turni.



Figure 18: Plancia di gioco di Risiko

Se si volesse simulare il traffico aereo si potrebbe utilizzare uno spazio a grafo dove ogni nodo rappresenta un aeroporto e gli archi rappresentano la tratta in partenza o in arrivo. Anche in questo caso l'applicazione della discretizzazione non sarebbe qualcosa di problematico.

## 2.5 Julia

Julia è un linguaggio di programmazione ad alto livello, multi-paradigma e open-source ideato per compiere analisi numerica ed effettuare operazioni di computer science in maniera rapida e stabile. Julia è nato ufficialmente come linguaggio di programmazione nell'anno 2012 con lo scopo di fornire uno strumento potente, robusto e veloce tanto se non più dei linguaggi considerati in questo ambito lo stato dell'arte, ovvero C e Fortran; ma anche facile da approcciare, al contrario dei linguaggi sopra citati. Julia è un linguaggio di programmazione scritto in C++ e Scheme, ma gran parte della sua composizione è scritta in Julia stesso [67].

le caratteristiche principali di questo linguaggio sono principalmente:

- Alte performance: lo scopo per cui Julia è nato è stato quello di offrire un linguaggio estremamente performante con la capacità di poter compilare programmi in codice nativo per molteplici piattaforme grazie all'utilizzo di LLVM
- Dinamico: la scelta di rendere Julia un linguaggio dinamicamente tipizzato lo rende di facile utilizzo in quanto rende molto più semplice il suo approccio anche a chi non ha una base solida di programmazione, in quanto ritorna la stessa sensazione di immediatezza di un linguaggio di scripting. Inoltre questo permette un alto supporto per l'uso interattivo
- Ambiente riproducibile: lo scopo del linguaggio è quello di poter permettere all'utente di ricreare le stesse condizioni ogni volta su ogni macchina su cui un programma viene eseguito. Questo può essere ottenuto tramite l'utilizzo di file binari pre compilati
- Componibile: Julia utilizza l'approccio multiple dispatch come paradigma, permettendo una grande flessibilità nell'esprimere una elevata quantità di pattern di programmazione, dall'object-oriented al funzionale
- General Purpose: lo scopo del linguaggio è quello di creare un ecosistema in grado di poter soddisfare qualsiasi esigenza di un utente, permettendo la creazione di applicativi e microservizi senza dover ricorrere ad integrazioni con codice non nativo Julia
- Open source: Julia abbraccia la filosofia open source, e il codice sorgente dell'intero linguaggio, così come di tutte le librerie è disponibile sulla piattaforma GitHub sotto la licenza MIT. Questo permette una crescita eterogenea grazie al contributo di più di 1000 utenti che si impegnano a migliorare il linguaggio

### 2.5.1 Agents.jl

Seguendo la filosofia propria del linguaggio di programmazione in cui è sviluppata, la libreria Agents.jl [15] viene sviluppata con l'obiettivo di essere facile da imparare e usare ed estendibile, con forte attenzione sulla creazione ed evoluzione di modelli veloci e soprattutto scalabili. Molteplici esempi comparativi sono stati effettuati mostrando come il framework sviluppato permetta di avere un notevole guadagno prestazionale rispetto ai maggiori competitor attualmente presenti sul mercato (Mesa, NetLogo, MASON) [1].

		Repast HPC	MATSIM, PDES-MAS, Swarm
	Extreme-scale		
	High /Large-scale	Altreva Adaptive Modeler, SeSAM	AnyLogic (2D/3D), AOR Simulation, CloudSim, CybelePro, FLAME, LSD (2D/3D), MASS, Pandora, UrbanSim
		NetLogo (2D/3D)	Ascape, CRAFTY, GAMA (2D/3D), SimEvents (MATLAB®), Simio (2D/3D), Simul8 (2D/3D)
	Medium-scale	JAS, VSEdit	Agent Factory, Breve (3D), Cormas, Envision, GALATEA, IDEA, JAMSIM, Janus, JASA, JAS-mine, MACSimJX, Mathematica® (Wolfram), Mimosa, MIMOSE, Mobility Testbed, Modgen, OBEUS, SimAgent, SimBioSys, TerraME, Xholon (2D/3D)
		AgentSheets, BehaviourComposer (2D/3D), FlexSim (2D/3D)	Eve, ExtendSim (2D/3D), GROWLab, Insight Maker, Mesa
	Light-weight /Small-scale	AgentScript, Framsticks (2D/3D), JAMEL, JCASim (1D/2D/3D), jES, MOBIDYC, PedSim, PS-I, Scratch (2D/3D), SimJr, SimSketch, SOARS, StarLogo, StarLogoTNG (3D), Sugarscape, VisualBots	SEAS (2D/3D)
		Simple/Easy	Moderate
			Complex/Hard
		Model Development Effort	

Figure 19: Tabella comparativa

La facilità di interazione con questa libreria non è da confondersi con una mancanza di opzioni durante lo sviluppo, in quanto nativamente Agents.jl permette l'integrazione con altre librerie che in maniera altrettanto semplice e veloce offrono all'utente la possibilità di addentrarsi nel mondo

del machine learning, in particolar modo il mondo del Scientific Machine Learning [47], branca che soprattutto grazie alla pandemia da Covid-19 ha visto un sempre piu' crescente interesse.

Agents.jl offre molteplici opzione di configurazione, ma principalmente quello su cui si basa sono i seguenti principi:

- definizione di un tipo di agente, generalmente viene raccomandato di estendere la tipologia *StandardABM* la quale e' la piu' concreta implementazione, nonche' l'implementazione di default, di un costruttore generico di un **AgentBasedModel**.
- definizione di una tipologia di spazio, esistono principalmente 2 tipologie di spazio da poter utilizzare come base e si basano sull'utilizzo di uno spazio *discreto* oppure *continuo*.
  - spazio discreto a grafo: un *GraphSpace* rappresenta uno spazio del modello rappresentato da un grafo arbitrario in cui ogni nodo puo' contenere una quantita' di agenti arbitraria. Per funzionare correttamente questa tipologia di spazio richiede che gli agenti implementino al loro interno specifici attributi per rappresentare la loro posizione all'interno dello spazio. Questa tipologia di spazio si appoggia alla libreria **Graphs.jl** [18] per gestire tutte le operazioni relative alla struttura dati del grafo.

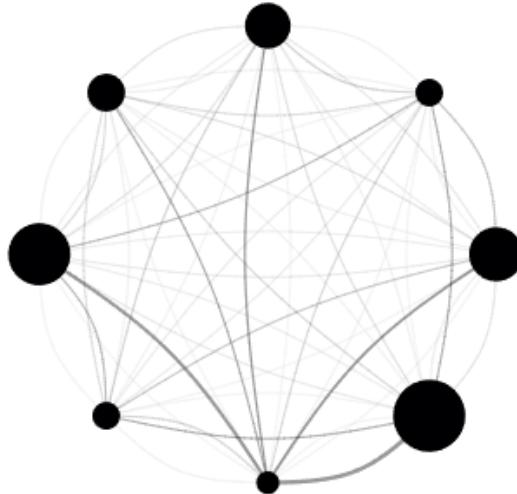


Figure 20: Rappresentazione di uno spazio a grafo

- spazio discreto a griglia: un *GridSpace* rappresenta uno spazio del modello rappresentato da una griglia di dimensione  $D \geq 1$ . Questa tipologia di spazio richiede l'utilizzo di una metrica per la definizione della distanza tra celle di una griglia. Ci sono attualmente tre tipologie di metriche supportate e sono: *Euclidean*, *Manhattan* e *Chebyshev*.

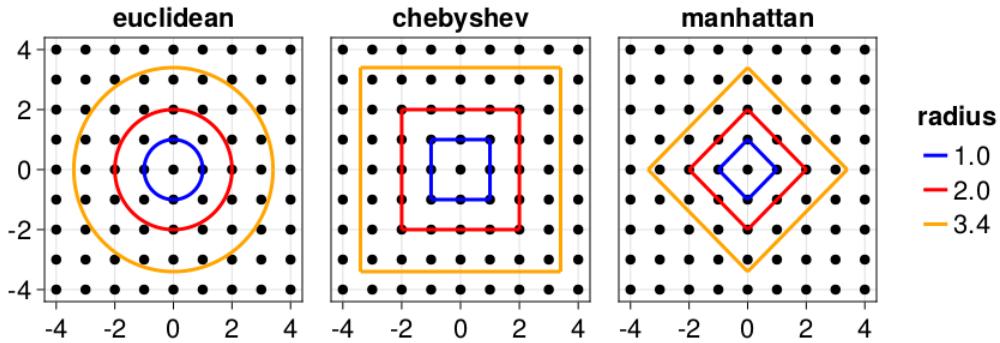


Figure 21: Metriche di distanza di una griglia

- spazio continuo: un *ContinuousSpace* rappresenta uno spazio di dimensione  $D \in (0, \infty)$ . E' fortemente consigliato di attribuire ad un agente all'interno di questo spazio due caratteristiche fondamentali, una posizione e una velocita'. Questa tipologia di spazio permette di rappresentare delle proprietà spaziali tramite valori finiti oppure tramite *funzioni*, i quali rappresentano una discretizzazione di valori spaziali che potrebbero non essere disponibili in maniera analitica. Utilizzando questa tipologia di spazio la metrica di distanza utilizzata sara' sempre *Euclidian*. Per velocizzare il calcolo della posizione degli agenti, viene effettuata una discretizzazione implicita dello spazio, ma questa puo' essere forzata a rimanere nello spazio continuo ottenendo un calo di prestazioni.
- spazio misto: un *OpenStreetMapSpace* rappresenta una mappa come un'entità continua che preferisce l'accuratezza alle prestazioni. La mappa viene rappresentata come un grafo connesso. I nodi non rappresentano necessariamente intersezioni.

### 2.5.2 SciML.ai



<https://sciml.ai>

Figure 22: Logo SciML.ai

SciML.ai è una collezione di librerie dedite all'analisi numerica e al calcolo scientifico. Questo framework permette di avere tutti gli strumenti per poter utilizzare facilmente, velocemente e in maniera robusta tecniche di analisi numerica molto avanzata, così da poter sviluppare applicazioni complesse in maniera semplice e concreta [47] [45] [46].

Il principale utilizzo che e' stato fatto di queste librerie si concentra principalmente sull'implementazione di metodi di analisi numerica, come ad esempio l'utilizzo di risolutori per sistemi di *Equazioni Ordinarie Differenziali* (ODE) che si possono trovare nel package *OrdinaryDiffEq.jl* [47] uniti a metodi di *Machine Learning* (ML) [42] [26] [27] per lo sviluppo di un modello di *Scientific Machine Learning* [45] [46]. La principale differenza che esiste tra i modelli classici di ML e quelli di SciML e' che i secondi richiedono un numero molto meno elevato di dati per la comprensione delle dinamiche che li governano, rendendo questi modelli moltio piu' scalabili.

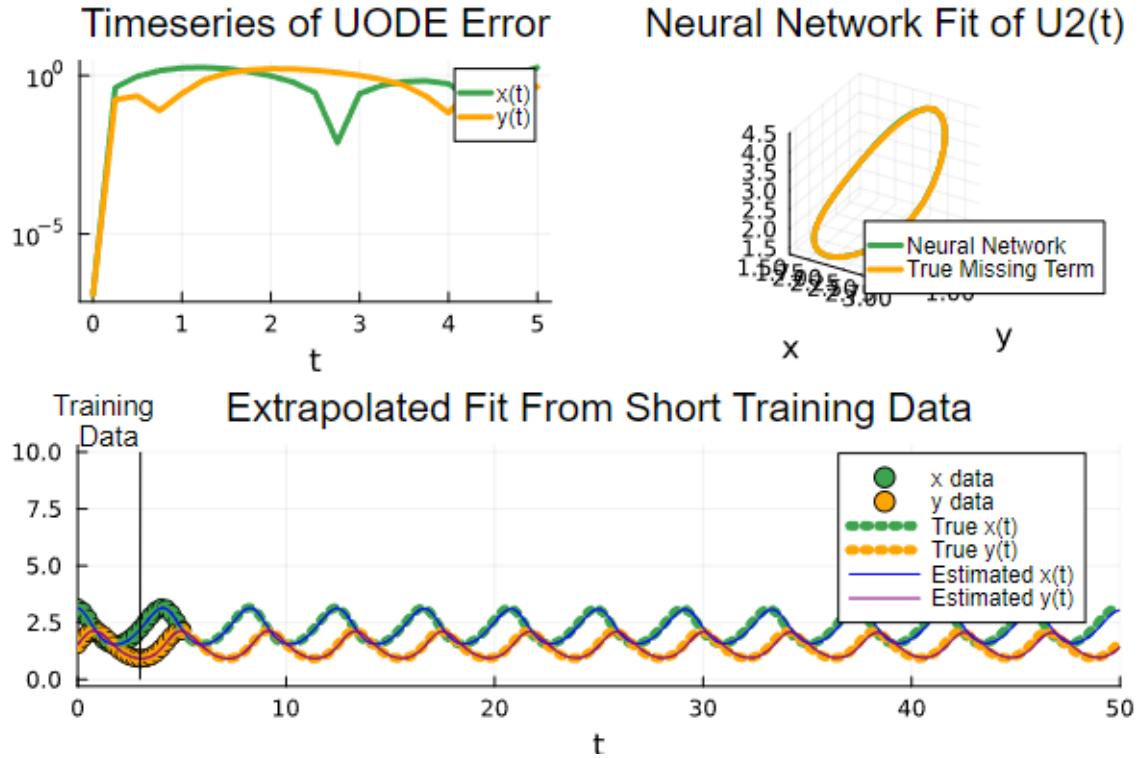


Figure 23: Esempio di Scientific Machine Learning

### Equazioni Ordinarie Differenziali

Nell'ambito matematico, una *equazione ordinaria differenziale* (ODE) e' un'equazione differenziale (DE) dipendente da un singolo valore indipendente, generalmente il tempo. All'interno di questa grande famiglia di equazioni, il gruppo delle *equazioni lineari differenziali* gioca un ruolo predominante in quanto la maggior parte dei fenomeni fisici e di matematica applicata possono essere descritti dalla soluzione di questo tipo di equazioni.

Una equazione lineare differenziale e' definita da un *polinomio lineare* e la sua derivata e' un'equazione della forma:

$$\alpha_0(x)y + \alpha_1(x)y' + \alpha_2(x)y'' + \dots + \alpha_n(x)y^{(n)} + b(x) = 0$$

dove  $\alpha_0(x), \dots, \alpha_n(x)$  e  $b(x)$  sono funzioni differenziabili arbitrarie che non richiedono di essere lineari, e  $y', \dots, y^{(n)}$  sono le successive derivate della funzione incognita  $y$  della variabile  $x$ .

L'utilizzo di *equazioni non lineari differenziali* puo' essere generalmente approssimato con la controparte lineare cosi' da ottenere una soluzione piu' semplice.

La suite di SciML.ai offre molteplici framework per la risoluzione di sistemi di equazioni lineari differenziali prevalentemente all'interno delle librerie *DifferentialEquations.jl* [47] [48] [34] [23], questi sono separati nelle seguenti categorie:

- Standard Non-Stiff ODEs Solver
- Standard Stiff ODEs Solver
- Second Order and Dynamical ODEs Solver
- IMEX Solvers
- Semilinear ODEs Solver
- DAE Solver
- Misc Solver

### Equazioni Differenziali Universali

Un *equazione differenziale differenziale* (UDE) e' una *equazione algebrica differenziale*[61] non triviale, ovvero un sistema di equazioni che contiene delle equazioni differenziali ed equazioni algebriche oppure e' un sistema equivalente, con la propriet'a che la sua soluzione puo' approssimare *qualsiasi* funzione continua su un qualunque intervallo  $\in R$  a qualsiasi livello di precisione desiderata. [80]

Per essere precisi, una equazione differenziale (possibilmente in forma implicita)  $P(y', y'', y''', \dots, y^{(n)}) = 0$  e' una UDE se, per ogni funzione a valori relative continua  $f$  e per ogni funzione continua positiva  $\epsilon$  esiste una soluzione liscia[75] (una funzione e' considerabile liscia se e' differenziabile in ogni suo punto, perci' continua)  $y$  di  $P(y', y'', y''', \dots, y^{(n)}) = 0$  con  $|y(x) - f(x)| < \epsilon(x) \forall x \in R$ .

Il concetto di UDE puo' essere analogo all'idea di una *Macchina di Turing Universale* [81] con la differenza che le UDE non dettano l'evoluzione di un sistema, ma si limitano a imporre determinate regole che ogni sistema che si evolve deve soddisfare. Questo permette di avere un modello robusto per l'analisi di dati e la predizione dell'interazione che hanno vari fenomeni tra loro.

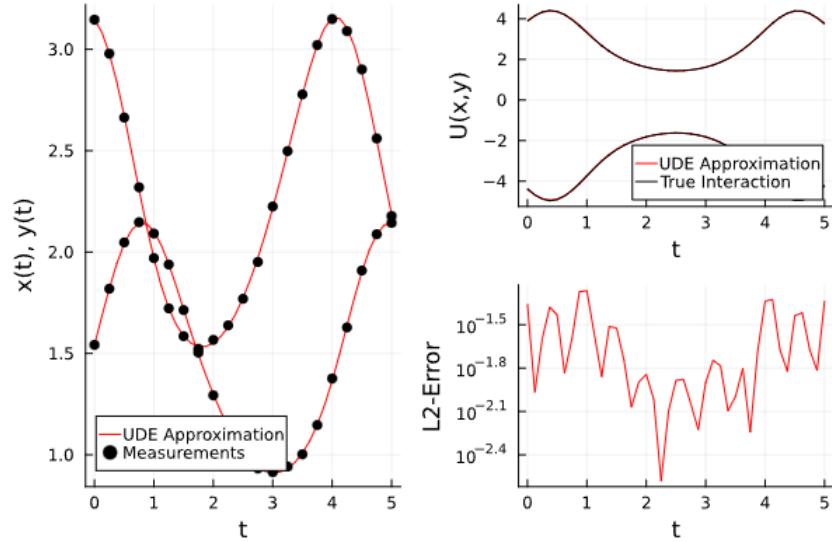


Figure 24: Comportamento UDE nell'approssimazione di fenomeni non lineari

Questo approccio viene spesso unito a tecniche *Data-Driven* [29] per l'identificazione sparsa di dinamiche non lineari. In particolare uno degli approcci utilizzati e' quello tramite l'algoritmo *SINDy* [76]. Questo algoritmo performa una serie di operazioni di regressione come ad esempio *LASSO* su una libreria di funzioni candidate non lineari ottenute da uno snapshot del sistema dinamico che si sta analizzando e delle sue derivate, con l'obiettivo di trovare le equazioni che lo governano. Questo procedimento si basa sull'assunzione che molti sistemi fisici hanno solamente una manciata di termini che ne dettano le dinamiche e l'evoluzione. Questo metodo e' stato largamente utilizzato nell'identificazione della *dinamica dei fluidi* cosi' come nelle *reti biologiche* e altri sistemi dinamici complessi.

### **Equazioni Neurali Differenziali**

Recentemente si e' iniziato a ibridare due paradigmi di modellazione come le ODE e le reti neurali (NN) che hanno sempre avuto ambiti applicativi ben distinti, per cercare di ottenere il massimo da entrambe minimizzando gli effetti indesiderati. [31] [11]

L'idea e' quella di inserire una NN come parte sinistra di una ODE, e successivamente inserire a sua volta la ODE all'interno di una NN piu' grande. Consideriamo il seguente esempio:

$$z(0) = z_0, \frac{dz}{dt}(t) = f_\theta(t, z(t))$$

dove  $z_0$  e' un qualsiasi tipo di input, mentre  $f_\theta$  e' la nostra rete neurale, e l'output del modello puo' essere successivamente utilizzato come input di  $z(T)$  per qualche  $T > 0$ . Potrebbe sembrare che questo approccio sia solamente una chimera di due tecniche distinte, quando in verita' non e' cosi.

Utilizzando i dati ottenuti da una rete neurale, i quali sono stati estrapolati dai dati osservati, questi possono essere utilizzati come parametri delle equazioni differenziali parametriche che costituiscono il modello matematico, unendo la flessibilita' di una rete neurale con la robustezza di un modello matematico di equazioni differenziali. Questo approccio vede utilizzo in molteplici applicazioni dal *deep learning* alla tradizionale modellazione matematica.

Questo tipo di approccio e' *memory efficient*, ha la capacita' di gestire *dati irregolari* con forti priori sullo spazio del modello, ha un'elevata capacita' di approssimare funzioni lineari e non lineari e si poggia su solide basi teoriche che pesca da entrambi i lati.

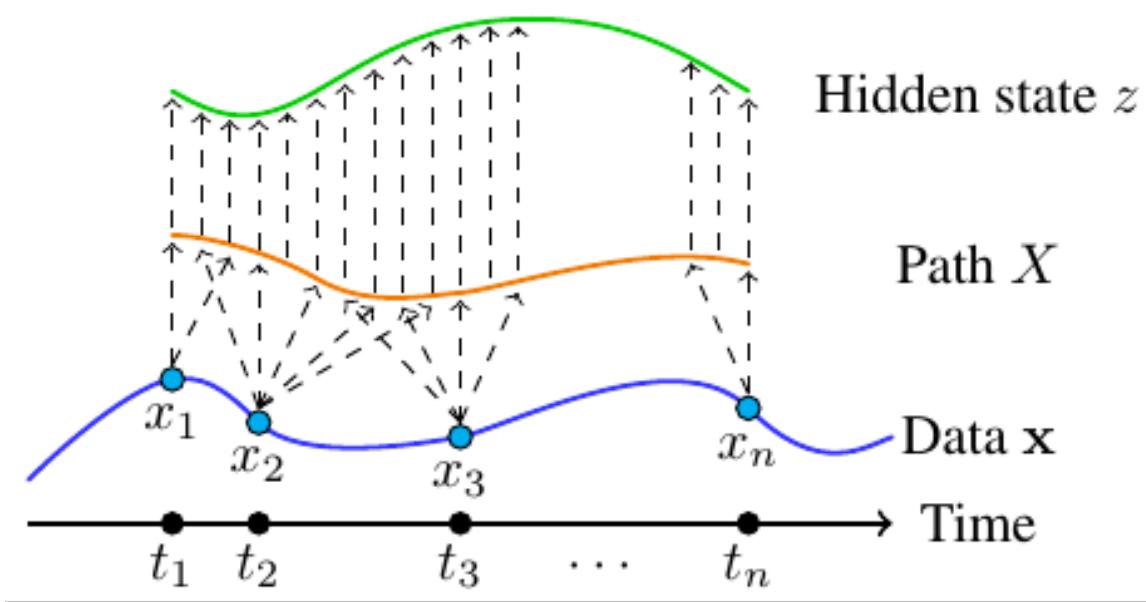


Figure 25: Esempio di Equazioni Differenziali Neurali

### 2.5.3 SafeBlues

Durante la pandemia da Covid-19 il framework SciML.ai è stato utilizzato per sviluppare applicazioni le quali grazie all'utilizzo di tecniche di scientific machine learning riuscivano a prevedere in maniera molto accurata l'andamento dell'epidemia, seppur in presenza di una scarsa quantità di dati, e le stesse presentavano misure di contenimento e prevenzione che si sono dimostrate essere efficaci nel loro utilizzo [6] [14].

Un esempio può essere il modello denominato **SafeBlues** [6] [14] il quale simulando una rete bluetooth in cui gli individui potevano venire infettati da un virus e poi infettare a loro volta gli individui circostanti nella rete con una certa probabilità, aveva riprodotto fedelmente l'andamento della pandemia da Covid-19. In aggiunta questa soluzione, aveva mostrato come l'applicazione di policy per il contenimento del virus bluetooth erano perfettamente applicabili anche al caso reale della pandemia.

Questa applicazione e' stata sviluppata e rilasciata per dispositivi mobili con lo scopo di sperimentare se il sistema sviluppatto (*Safe Blues System*) potesse aiutare e migliorare i tradizionali metodi di previsione del decorso di una pandemia.

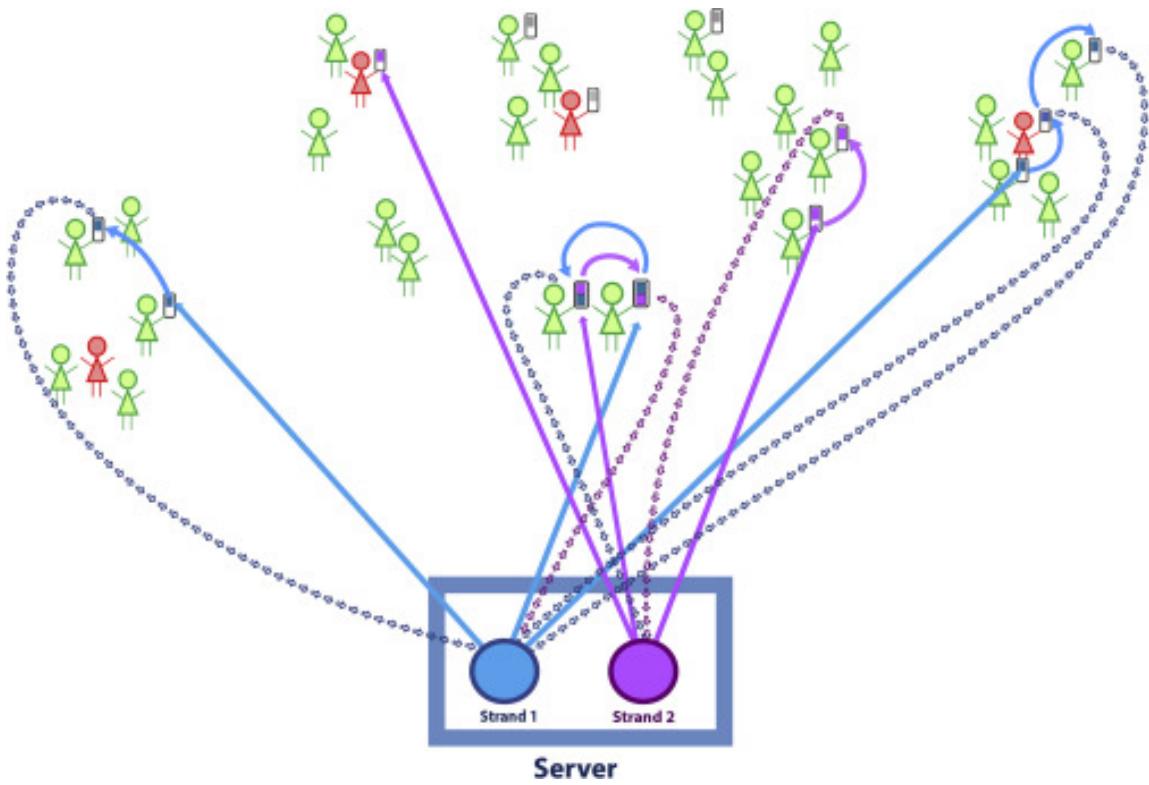


Figure 26: Esempio funzionamento SafeBlues

## 3 Metodi e Modelli

### 3.1 Modello ad Agente

Il modello ad agente implementato pone l'accento sull'idea di avere un controllo granulare sulla localizzazione dei possibili focolari o punti di interesse durante il ciclo di vita di una pandemia. Per modellare uno spazio simile e' stato utilizzato uno **spazio discreto a grafo**.

L'idea e' quella di rappresentare differenti punti di interesse astratti dal livello di dettaglio che si vuole andare a modellare, andando a generalizzare uno scenario tipo nel quale N agenti si muovono all'interno di una rete (in questo caso un grafo) e con questo monitorare diversi tipologie di interventi e il loro grado di successo nella lotta contro una possibile pandemia.

La struttura offerta dal framework **Agents.jl** e' quella di uno spazio *GraphSpace* in coppia con il relativo tipo di agente di default *GraphAgent*. Non vi e' alcuna differenza tra la tipologia GraphAgent e ad esempio l'interfaccia AbstractAgent, in quanto il primo deriva dal secondo. Tuttavia GraphAgent ha già al suo interno tutti gli attributi utili per gestire un agente all'interno di uno spazio a grafo, lasciando al programmatore una preoccupazione in meno.

#### Agente

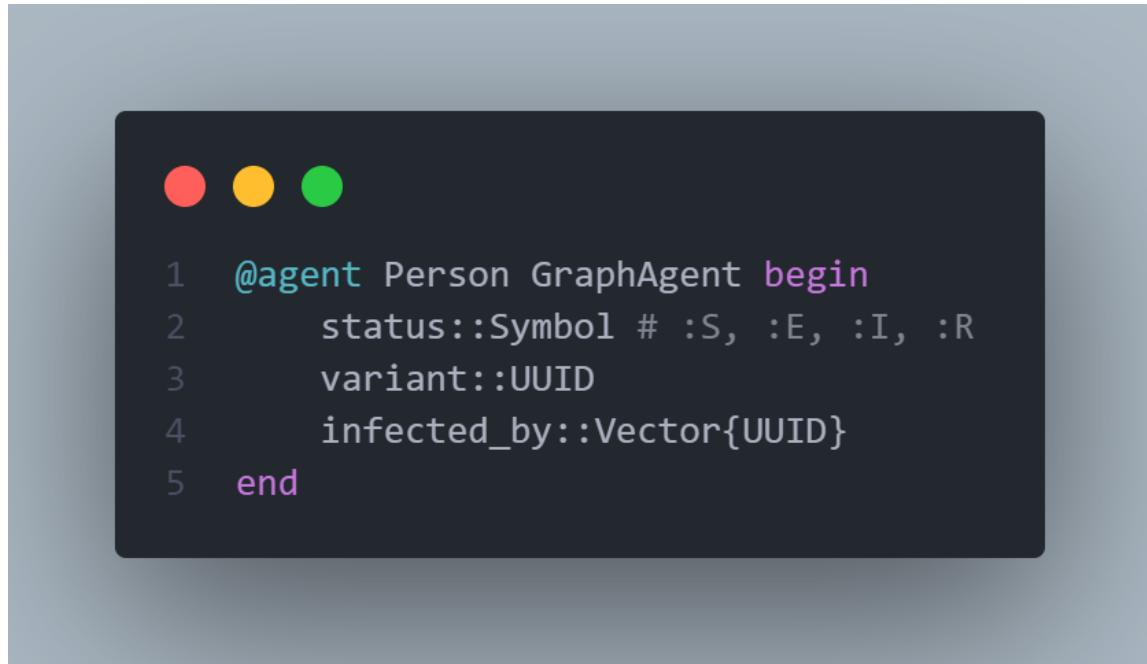


Figure 27: Codice Agente

Come e' possibile vedere dalla Figura 27, il codice che si occupa di creare l'agente e' estremamente semplice. Al suo interno troviamo tutto quello che un agente deve sapere per poter agire in maniera corretta all'interno dell'ambiente. Essendo che l'agente *Person* deriva dall'agente *GraphAgent*, questo eredita automaticamente i campi fondamentali **pos** e **id**, i quali serviranno per riconoscere

la posizione all'interno del grafo dell'agente e distinguere l'agente dagli altri.

I campi principali sono:

- status: e' un campo di tipo **Symbol** che rappresenta lo stato di un determinato agente in un preciso momento del tempo. Questo stato puo' mutare nel tempo ed e' strettamente correlato a tutti e soli gli stati che un individuo puo' assumere all'interno di un modello matematico di tipologia **SEIR**
- variant: e' un campo di tipo **UUID** (Universally Unique ID) che rappresenta un identificatore univoco associato ad una delle possibili varianti dell'agente patogeno attualmente in circolazione. Questo attributo serve principalmente a gestire la possibilita' di non avere una completa immunita' dopo aver contratto e aver debellato il virus dal proprio sistema immunitario, rendendo gli individui identificati come :**R** passibili di reinfezione
- infected by: e' un campo di tipo **Vector** che rappresenta quali varianti sono state contratte dall'agente. Un agente non puo' piu' essere infettato da queste varianti. Questo approccio e' stato ideato per la modellazione di un vaccino, il quale non e' detto che copra tutte le varianti che sono uscite, o comunque non e' detto che, pur coprendo tutte le varianti conosciute copra anche quelle non conosciute.

Utilizzare la macro **@agent** e' il modo che viene incentivato dal linguaggio, in quanto permette di creare un agente da un super tipo come `AbstractAgent` e permette di avere già inclusi tutti i campi necessari. Inoltre, utilizzando questa **macro** e' possibile includere tutti i campi di un altro agente, dovendo solamente specificare quelli nuovi.

### Spazio e Modello

Lo spazio e' di tipo `GraphSpace` e viene definito direttamente in fase di creazione del modello.



```

1 model = StandardABM(
2     Person, #agent
3     GraphSpace(Agents.Graphs.complete_graph(C)); # space
4     properties=@dict(
5         number_point_of_interest, # vector
6         migration_rate, # matrix
7         new_migration_rate = migration_rate, # matrix
8         step_count = 0, # counter
9         R₀, # float
10        ξ, # float ∈ [0,1]
11        Iₛ, # vector
12        C, # integer
13        γ, # integer
14        σ, # integer
15        ω, # integer
16        δ, # float ∈ [0,1]
17        η = [zeros(Float64, length(number_point_of_interest))...],
18        Rᵢ, # float
19        happiness, # vector
20    ),
21    rng
22 )

```

Figure 28: Codice Modello

In questo caso come si puo' osservare dalla Figura 28, lo spazio e' un *grafo completo* con C nodi. I nodi corrispondono ai punti di interesse del grafo. Per dettagliare maggiormente questo spazio altrimenti scarso, vengono definite delle *proprietà spaziali* del modello, ovvero proprietà che ci sono sempre e ovunque. Queste proprietà possono essere divise in due macro categorie: proprietà dei punti di interesse e proprietà della pandemia.

Le proprietà dei punti di interesse sono le seguenti:

- number point of interest: e' di tipo **Vector** e rappresenta il numero di agenti che quel determinato nodo puo' contenere. Gli elementi all'interno di questo vettore vengono determinati tramite una funzione esponenziale.
- migration rate: e' di tipo **Matrix** e rappresenta una matrice di transizione tra i vari nodi del grafo. Questa matrice di transizione codifica la probabilità di un agente di spostarsi dal suo nodo ad un qualsiasi altro nodo presente. Essendo il grafo di tipo completo, ogni nodo e' collegato a tutti gli altri, tuttavia non tutti i collegamenti sono equamente probabili. La matrice

infatti tiene in considerazione la probabilita' di un agente di spostarsi piu' facilmente verso un nodo popoloso, piuttosto che verso uno che non lo e', questo a mimica degli spostamenti umani.

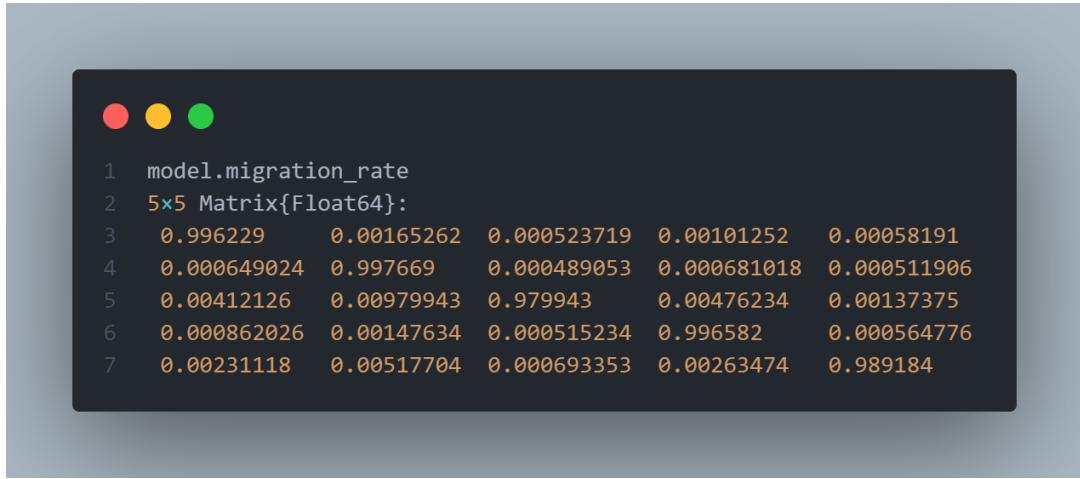


Figure 29: Esempio di matrice di transizione

- $I_0$ : e' di tipo **Vector** e rappresenta il vettore degli infetti iniziali. Generalmente viene inizializzato randomicamente nella posizione dei nodi, ma fisso nel numero di infetti ovvero con  $I = 1$ .
- $\eta$ : e' di tipo **Vector** e rappresenta il vettore delle contromisure implementate. L'esperienza della pandemia ci ha insegnato che non sempre una contromisura generalizzata e' qualcosa che funziona, alle volte e' giusto trattare in maniera locale il problema, applicando contromisure mirate alla tipologia di esigenza locale. Questo vettore raccoglie quindi le contromisure associate ad ogni nodo del grafo. Una contromisura e' un valore  $\in [0, 1]$  che identifica l'efficacia della contromisure applicate. La sfida e' comprendere in che modo una contromisura impatti sulla diffusione della pandemia.
- $happiness$ : e' di tipo **Vector** e rappresenta il vettore della felicità della popolazione di un determinato nodo. Questo vettore contiene valori  $\in [-1, 1]$ . Questo parametro e' stato inserito per evitare una caduta in una strategia funzionale ma insostenibile. Se l'obiettivo di un controllore e' quello di minimizzare il numero di infetti (e morti), la scelta piu' ragionevole (non dovendo tenere in conto di null'altro), e' quella di applicare un lockdown generalizzato a tutta la popolazione. Terminato il periodo di vita del virus si e' risolta la crisi pandemica. Questa soluzione e' pero' **umanamente insostenibile**. Da qui un termine aggiuntivo che il controllore deve bilanciare, ovvero la felicità della popolazione. Infatti questo termine deve essere massimizzato, e dipende principalmente dal termine  $\eta$ .

Le proprieta' della pandemia sono invece:

- $R_0$ : e' di tipo **Float64** e rappresenta l'indice di infettività di un virus
- $\gamma$ : e' di tipo **Int** e rappresenta il numero di giorni dopo di cui un paziente che ha contratto il virus puo' considerarsi guarito

- $\sigma$ : e' di tipo **Int** e rappresenta il numero di giorni per cui un agente dopo essere stato infettato viene considerato *esposto*, per cui non ancora in grado di infettare, ma che una volta terminato questo periodo diventera' infettivo
- $\omega$ : e' di tipo **Int** e rappresenta il numero di giorni per cui un agente e' considerato immune alla malattia dopo essere guarito da essa
- $\delta$ : e' di tipo **Float64** e rappresenta la probabilita' di un agente infetto di morire al termine del proprio periodo di infettività ( $\gamma$ ).
- $\xi$ : e' di tipo **Float64**  $\in [0, 1]$  e rappresenta, una volta che e' stato introdotto un vaccino come metodo di prevenzione contro il virus, il rateo di individui S che che possono essere vaccinati, e quindi diventare R, al giorno.
- $R_i$ : e' di tipo **Float64** e rappresenta l'indice di infettività desiderato, che si vuole raggiungere per arginare la malattia e fermarne l'avanzata. Generalmente questo valore si attesta al piu' intorno a 1.0

### Funzione di avanzamento agente

Un agente finche' all'interno di un modello segue ciclicamente il comportamento riportato in figura 30



```

1  function agent_step!(agent, model::StandardABM)
2      migrate!(agent, model)
3      transmit!(agent, model)
4      update!(agent, model)
5  end

```

Figure 30: Comportamento agente

Durante la funzione di avanzamento un agente segue un insieme di regole, che dettano un comportamento semplice ma completo. Un agente ha il compito di provare a spostarsi di nodo, passando da quello attuale ad uno dei molteplici collegati; la scelta del nodo e' casuale ma la sua transizione dipende dalla matrice di transizione definita precedentemente. Successivamente l'agente ha il compito di fare da veicolo per le infezioni `sse` il suo stato e' infetto, altrimenti non e' necessario che venga effettuato questo punto. Infine il compito di un agente e' quello di aggiornare il proprio stato

passando potenzialmente da quello attuale ad uno nuovo; questo implica che e' possibile che un agente venga rimosso totalmente dal modello se dovesse morire.

La funzione di trasmissione basa il proprio funzionamento sull' *assunzione* che il numero di contatti di un individuo infetto siano distribuiti seguendo una **Poisson** con media il parametro  $R_0$ . Ogni contatto poi ha la possibilita' di essere infettato data unicamente da due fattori:

- stato dell'agente: solamente un agente con stato S (Suscettibile) puo' essere infettato, oppure un agente con stato R (Recovered) che non ha mai contratto la variante con cui attualmente e' entrato in contatto
- probabilita' di infezione: questo valore e' dato dal rapporto tra le variabili  $R_0$  e  $\gamma$  e indica la probabilita' di un infezione secondaria dato un contatto positivo all'essere infetto



```

1  function transmit!(agent, model::StandardABM)
2      agent.status != :I && return
3      ncontacts = rand(model.rng, Poisson(model.R₀))
4      for i = 1:ncontacts
5          contact = model[rand(model.rng, ids_in_position(agent, model))]
6          if (
7              contact.status == :S ||
8              (contact.status == :R && !(agent.variant ∈ contact.infected_by))
9          ) && (rand(model.rng) < model.R₀ / model.γ)
10             contact.status = :E
11             contact.variant = agent.variant
12         end
13     end
14 end

```

Figure 31: Funzione di trasmissione

La funzione di aggiornamento dello stato degli agenti prende spunto dalla funzione di transizione descritta dal modello compartmentale SEIR in figura 9 tranne che non viene modellata ne la morte naturale ne tantomeno la nascita di nuovi agenti all'interno del modello. Viene pero' aggiunta la possibilita' di passare da uno stato S ad uno R tramite la probabilita' di **vaccinarsi** definita con il parametro  $\xi$ .



```

1  function update!(agent, model::StandardABM)
2      # possibilità di vaccinazione
3      if agent.status == :S && (rand(model.rng) < model.ξ)
4          agent.status = :R
5          # fine periodo di latenza
6      elseif agent.status == :E && (rand(model.rng) < 1 / model.σ)
7          agent.status = :I
8          # fine malattia
9      elseif agent.status == :I && (rand(model.rng) < 1 / model.γ)
10         # probabilità di morte
11         if rand(model.rng) < model.δ
12             remove_agent!(agent, model)
13             return
14         end
15         # probabilità di guarigione
16         agent.status = :R
17         push!(agent.infected_by, agent.variant)
18         # perdita immunità
19     elseif agent.status == :R && (rand(model.rng) < 1 / model.ω)
20         agent.status = :S
21     end
22 end

```

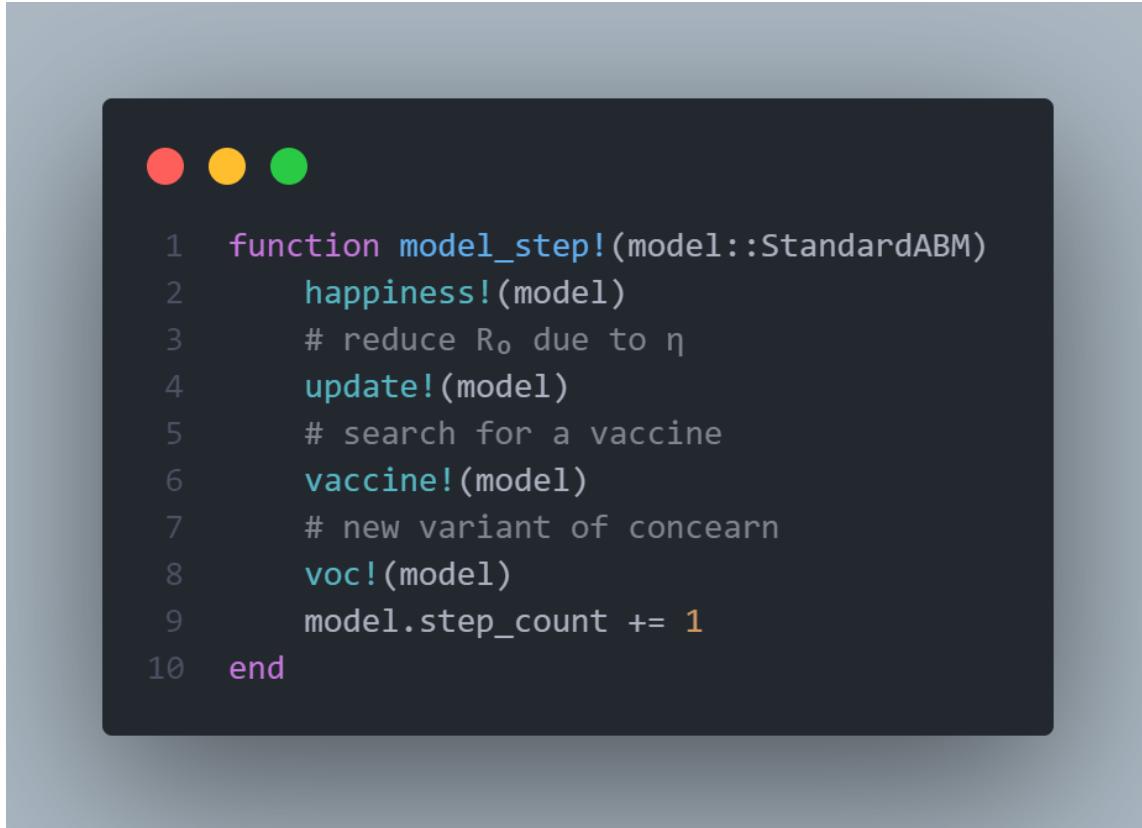
Figure 32: Funzione di aggiornamento dello stato di un agente

Come mostrato nella sezione riguardante le proprietà del modello in figura 28, le proprietà che descrivono le probabilità di transizione da uno stato all'altro sono prevalentemente di tipo **Int**. Queste però vengono poi implicitamente trasformate in tipo **Float64** quando bisogna calcolare la probabilità di transizioni. La scelta di inizializzarle come tipo Int è stata fatta solamente per essere più **Human Understandable**.

### Funzione di avanzamento modello

Ogni passo di avanzamento del modello segna un passo di avanzamento dell'intero sistema. Questo vuol dire che all'interno di un passo del modello vi è un passo per ogni agente. Di default il framework Agents.jl effettua prima l'avanzamento degli agenti e successivamente quello del modello. Questo comportamento può essere modificato quando viene chiamata la funzione di **step!** che si occupa di far girare l'intero sistema.

Il modello si occupa ad ogni passo di effettuare tutte quelle operazioni che non sono legate ad un singolo agente ma che sono legate o alla comunità di agenti oppure al modello in quanto sistema complesso.



```
1  function model_step!(model::StandardABM)
2      happiness!(model)
3      # reduce R₀ due to η
4      update!(model)
5      # search for a vaccine
6      vaccine!(model)
7      # new variant of concern
8      voc!(model)
9      model.step_count += 1
10 end
```

Figure 33: Funzione di avanzamento del modello

Come è possibile osservare dalla figura 33, il sistema si occupa principalmente di aggiornare la felicità generale degli agenti, aggiornare l'indice  $R_0$  riducendo il suo valore in caso in cui sono state prese delle contromisure adeguate, controllare se è disponibile un vaccino, generare una nuova *Variant of Concern* (VOC).

Questa sezione è quella che più presta il fianco ad *assunzioni* riguardo il comportamento generale del sistema. Tutte le funzioni sopra descritte si basano su un insieme più o meno nutrito di assunzioni per funzionare correttamente. Questo perché al momento in cui sto redigendo questo documento June 18, 2023, non sono ancora riuscito ad implementare metodi migliori.

Attualmente la funzione che si occupa di generare la felicità complessiva di un nodo si basa sull'utilizzo di una *tangente iperbolica* ( $\tanh$ ) che prende come parametro il valore attuale di happiness del nodo e lo bilancia con il valore delle contromisure  $\eta$  associate e attualmente uso. Questo approccio è sicuramente problematico e fallace per numerosi motivi ma attualmente adempie al suo obiettivo.

```

● ● ●

1 function happiness!(model::StandardABM)
2   for n = 1:model.C
3     # agents = filter(x -> x.pos == n, [a for a in allagents(model)])
4     # dead = length(agents) / model.number_point_of_interest[n]
5     # infects = filter(x -> x.status == :I, agents)
6     # infects = length(infects) / length(agents)
7     # very bad estimator for happiness
8     model.happiness[n] =
9       tanh(model.happiness[n] - model.η[n]) #- tanh(dead + infects) / length(agents))
10    end
11 end

```

Figure 34: Funzione di stima della felicità generale di un nodo del grafo

La funzione di aggiornamento dell'indice  $R_0$  in relazione a  $\eta$  riduce ad ogni passo del modello il valore di  $R_0$  proporzionalmente al valore di  $\eta$  e del valore di  $R_i$ , ovvero il valore atteso di  $R_0$  che si vuole ottenere.

```

● ● ●

1 function update!(model::StandardABM)
2   if model.R₀ > model.Rᵢ
3     model.R₀ -= mean(model.η) * (model.R₀ - model.Rᵢ)
4   end
5 end

```

Figure 35: Funzione di riduzione  $R_0$

Come mostrato in figura 35 questo metodo si basa su un aspetto macroscopico del sistema, cercando di rimanere quanto più fedele possibile all'idea alla base del modello. Tuttavia rimane molto approssimativo come approccio, soprattutto quando la quantificazione delle contromisure o misure di prevenzione ( $\eta$ ) non è chiara né facile da stimare.

La funzione che si occupa di creare un vaccino è la seguente



```
1 function vaccine!(model::StandardABM)
2     if model.ξ == 0 && rand(model.rng) < 1 / 365
3         # heard immunity over vaccine effectiveness
4         v = (1 - (1 / model.R₀)) / 0.83
5         # voglio arrivare ad avere una herd immunity
6         # entro model.ω tempo
7         model.ξ = v / model.ω
8     end
9 end
```

Figure 36: Funzione che si occupa di creare un vaccino

Questa si occupa di generare il flusso di vaccinazioni giornaliere date dal calcolo di una *immunità di gregge* ottenibile tramite un percorso vaccinale.[64] L'efficacia del vaccino e' stata assegnata seguendo lo studio di meta-analisi [82].

Infine la funzione che si occupa di generare la *variant of concern* (VOC) e' la seguente

```

1  function voc!(model::StandardABM)
2      if rand(model.rng) ≤ 8 * 10E-4 # condizione di attivazione
3          variant = uuid1(model.rng)
4          model.R₀ = abs(rand(model.rng, Uniform(3.3, 5.7)))
5          model.γ = round(Int, abs(rand(model.rng, Normal(model.γ))))
6          model.σ = round(Int, abs(rand(model.rng, Normal(model.σ))))
7          model.ω = round(Int, abs(rand(model.rng, Normal(model.ω, model.ω / 10))))
8          model.δ = abs(rand(model.rng, Normal(model.δ, model.δ / 10)))
9          model.R₀ᵢ = model.R₀
10         # new infect
11         new_infect = random_agent(model)
12         new_infect.status = :I
13         new_infect.variant = variant
14     end
15 end

```

Figure 37: Funzione che si occupa di generare la VOC

Si possono notare molte assunzioni:

- la condizione di attivazione della funzione si basa su un valore che sembra totalmente casuale. Quel valore, ovvero  $8 \cdot 10^{-4}$  deriva dai seguenti articoli [35] [53] [2] i quali descrivono prevalentemente il tasso di mutazione casuale delle basi che compongono il DNA del virus SARS-COV2. Questo però non implica che tali mutazioni creino una VOC. Per semplicità è stato scelto di usare l'approccio per cui se una mutazione avviene, questa è un a VOC. In linea di massima sembra che questo approccio semplicistico crei un numero di VOC generalmente sensato e in linea con quanto abbiamo potuto osservare durante la pandemia
- la distribuzione dei parametri associati alla pandemia viene calcolata seguendo una distribuzione **Normale**, tranne per la distribuzione dell'indice  $R_0$  il quale segue una distribuzione di tipo **Uniforme** [68]

## Grafici

Il seguente grafico 38 mostra l'andamento delle curve del modello quando questo viene eseguito senza alcuna tipologia di intervento.

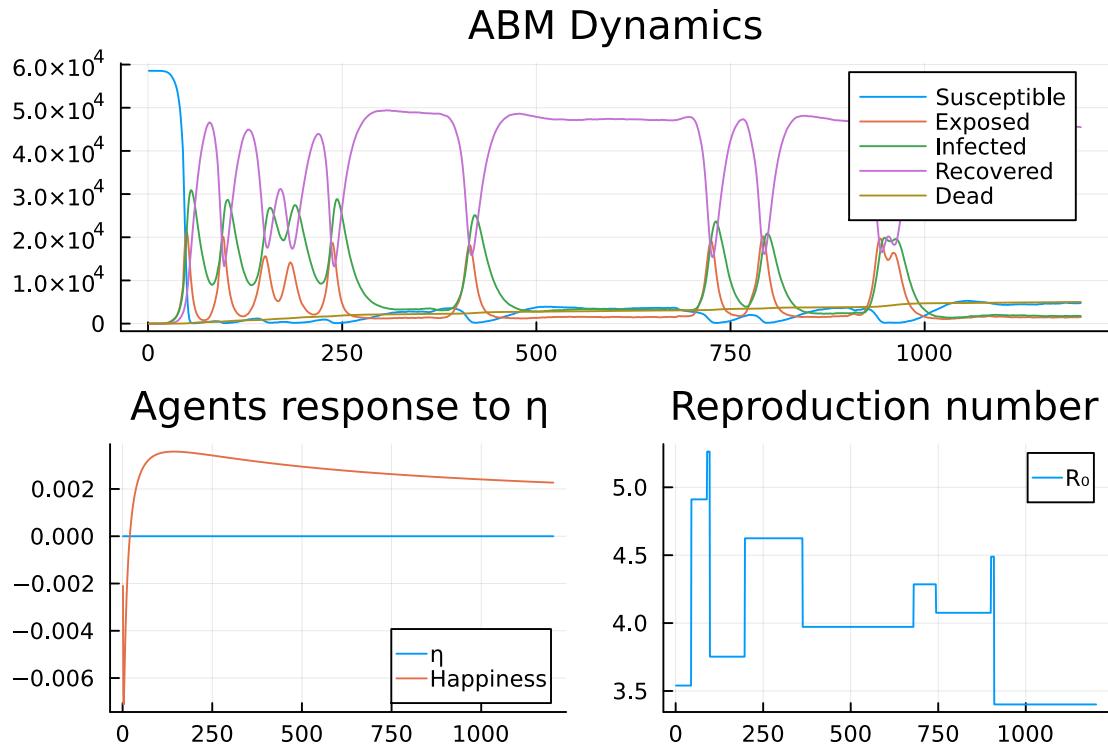


Figure 38: Grafico cumulativo del risultato del modello senza intervento del controllore

Come e' possibile notare, il numero di individui suscettibili crolla drasticamente per via della diffusione rapida e simil esponenziale che ha il virus. Questa viene emulata dall'altrettanto rapida crescita di individui guariti (recovered) che pero', per via di come e' stato definita la condizione di guariti, non sono immuni alle varianti del virus. Queste proprietà contribuiscono ad un andamento ciclico delle curve, in particolar modo di quelle  $E$ ,  $I$ ,  $R$ . La curva associata al comportamento  $S$  ha un suo andamento ciclico seppur molto sottile; questa sottigliezza e' dovuta all'assenza di misure di controllo all'interno del sistema.

Si puo' infine notare come la curva associata all'andamento degli individui nella classe  $D$  abbia una crescita lineare.

A seguire si puo' osservare come la curva associata alla variabile di happiness del modello, valore che serve per bilanciare la durezza delle misure di controllo per evitare di cadere in un ciclo funzionale ma insostenibile, mostra un iniziale decrescita molto ripida, e una successiva crescita estremamente rapida fino poi a ottenere un *plateau* poco sopra il livello 0.0, che rappresenta uno stato di normalità'.

Successivamente questa curva tende a decrescere lentamente. Questo comportamento e' irrealistico e associato alla definizione che e' stata fatta della funzione **happiness!**<sup>34</sup>. Tuttavia anche se il comportamento di questa curva non e' completamente realistico, si puo' comunque utilizzare per mantenere sotto controllo le contromisure  $\eta$ .

Infine si nota come, seppur la definizione della funzione associata alla creazione di una nuova voc 37 sia semplicistica e irrealistica, il grafico mostra come su un periodo di circa 3 anni, associabile alla durata del periodo covid, le voc sono al piu' una decina (ogni picco o depressione e' una voc).

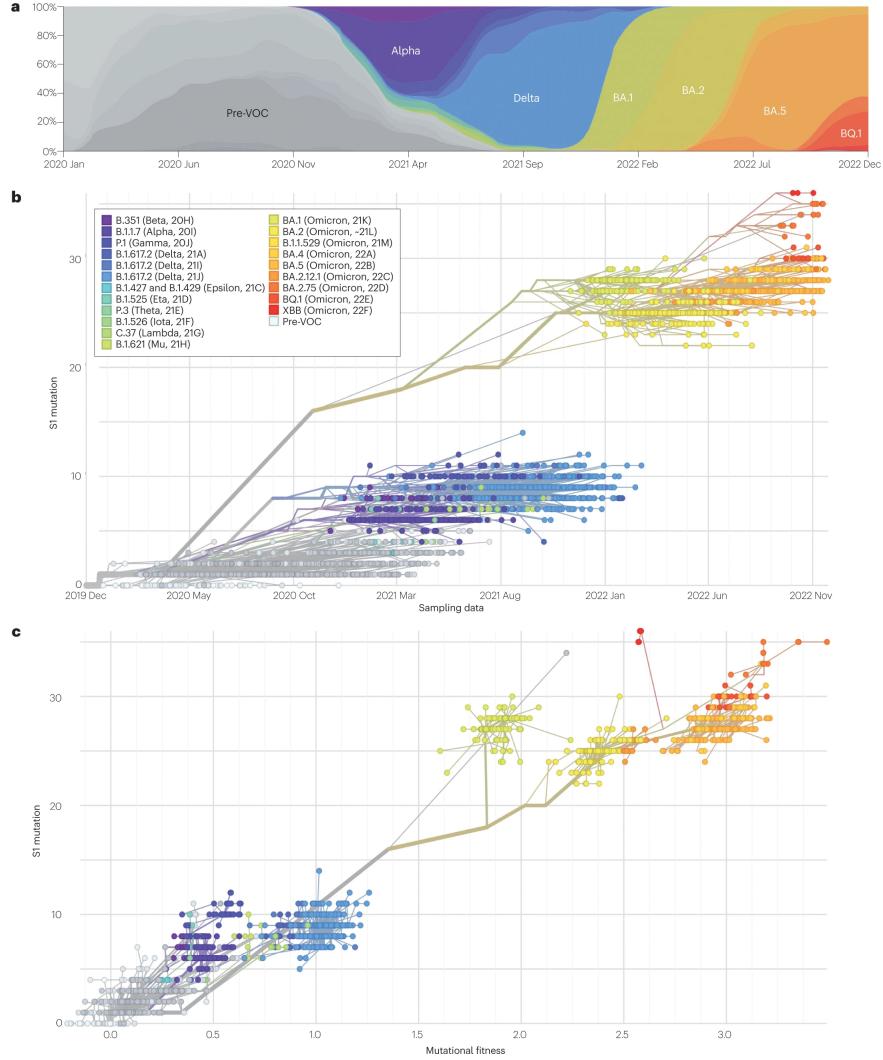


Figure 39: Grafico delle mutazioni casuali del virus SARS-COV2 preso dall'articolo [35]

Si nota come le voc scoperte del covid siano una quantita' pressoché' similare seppur con una distribuzione all'interno della linea temporale differente. Tuttavia il comportamento semplicistico applicato nel modello sembra essere una buona approssimazione.

## 3.2 Monitoraggio e Intervento

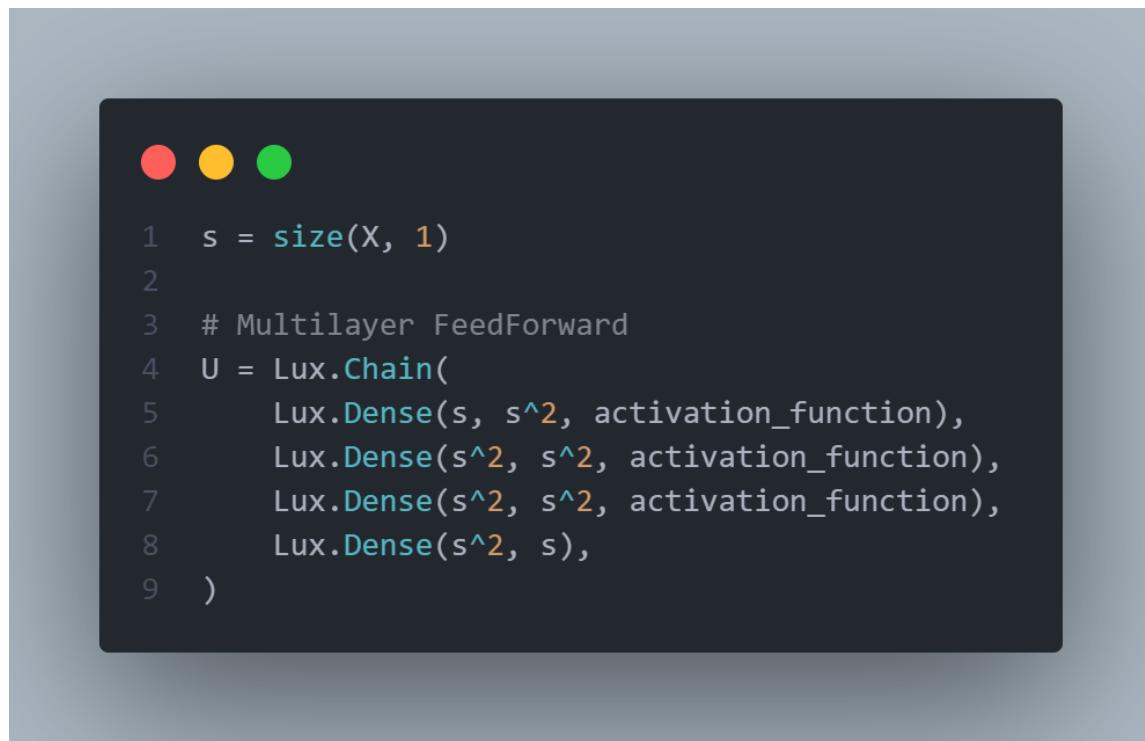
L'utilizzo della suite **SciML.ai** e' stato fatto per l'integrazione di un modello di Machine Learning in un sistema di monitoraggio e possibilmente di intervento.

### 3.2.1 Predizione

Applicare un algoritmo di ML per predire l'andamento di un sistema non lineare dinamico senza avere una grande base di dati puo' risultare problematico e i risultati possono risultare poco affidabili. Lo sviluppo di tecniche che congiungono la modellazione puramente matematica alla flessibilita' delle reti neurali ha permesso di sviluppare approcci *data-driven domain-specific* [46] [31] [13] che permettono di ottenere risultati affidabili pur avendo una base di dati scarsa.

### ODE e Neural Network

L'approccio iniziale al problema e' stato quello di definire una rete neurale con cui andare ad effettuare le predizioni che serviranno per determinare l'andamento delle curve e i possibili interventi.



```
1 s = size(X, 1)
2
3 # Multilayer FeedForward
4 U = Lux.Chain(
5     Lux.Dense(s, s^2, activation_function),
6     Lux.Dense(s^2, s^2, activation_function),
7     Lux.Dense(s^2, s^2, activation_function),
8     Lux.Dense(s^2, s),
9 )
```

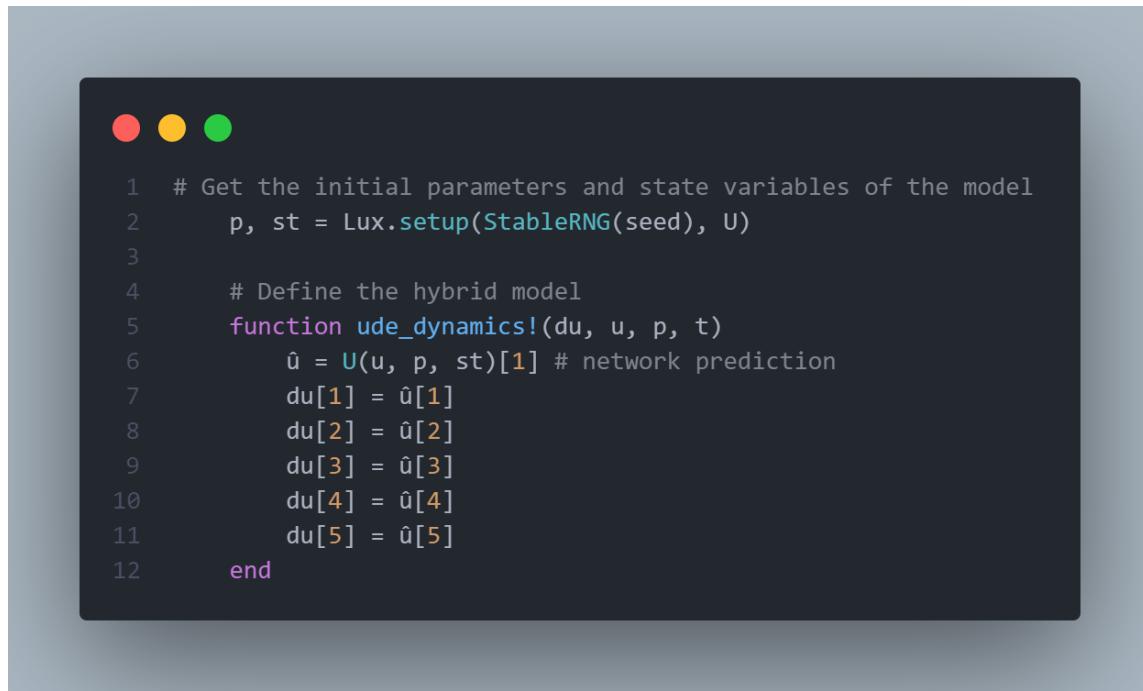
Figure 40: Definizione della rete neurale utilizzando la libreria Lux.jl [42]

L'idea principale alla base dell'approccio delle **Neural ODE** e' quello di accorpare due approcci per ottenere il meglio da entrambi. Partendo ad analizzare la definizione della rete neurale, si puo' osservare come sia del tipo *FeedForward*. L'approccio utilizzato per definire quanti *hidden nodes* dovessero essere presenti all'interno di ogni *layer* della rete e' basato sul numero di equazioni che si

vogliono ricavare. Questo numero viene ricavato dai dati in input, che sono nella forma **DataFrame** e che vengono successivamente convertiti in una forma di tipo **Array**. Le righe che compongono la matrice **X** sono le mie equazioni incognite.

Per cui si ottiene che il numero di nodi in input e output sara' esattamente il numero di equazioni incognite, mentre il numero di hidden nodes sara' pari al quadrato di quest'ultimo. La scelta di questo valore ricade principalmente sul fatto che si sta analizzando un sistema SEIRD, quindi a 5 stati (equazioni). Avere 25 hidden nodes per ogni hidden layer mi e' sembrata una scelta ragionevole. Cosi' come avere un numero di hidden layer decisamente ristretto.

L'idea alla base e' quella comunque di avere un modello molto snello, sia in memoria che in tempo di computazione, che permetta di arrivare ad avere risultati affidabili con un numero contenuto di iterazioni, senza richiedere al contempo un elevato quantitativo di memoria.



```

1 # Get the initial parameters and state variables of the model
2 p, st = Lux.setup(StableRNG(seed), U)
3
4 # Define the hybrid model
5 function ude_dynamics!(du, u, p, t)
6     ū = U(u, p, st)[1] # network prediction
7     du[1] = ū[1]
8     du[2] = ū[2]
9     du[3] = ū[3]
10    du[4] = ū[4]
11    du[5] = ū[5]
12 end

```

Figure 41: Modello ibrido che utilizza l'idea di rete neurale insieme all'idea di ODE

L'utilizzo successivo del modello di NN precedentemente introdotto 40 viene fatto utilizzandolo all'interno di un sistema di ODE. Questo viene utilizzato come predittore dei valori attualmente non conosciuti che modificano il sistema. Si puo' notare come non esista un vero e proprio aggiornamento dei pesi della rete, questo perche' i pesi sono implicitamente i parametri del sistema di ODE e ogni iterazione li aggiorna per definizione.

## **Raffinazione con SINDy**

### **3.2.2 Intervento**

Descrizione del controllore e di come viene implementato all'interno della logica del modello.  
Chiarire perche' vengono utilizzate determinate tecniche e non altre.

Risultati grafici.

Bug e problematiche

## **4 Risultati Ottenuti**

Analisi riassuntiva dei risultati ottenuti con qualche pensiero personale

## **5 Sviluppi Futuri**

### **5.1 Altre epidemie**

#### **5.1.1 Ebola**

#### **5.1.2 Aviaria**

#### **5.1.3 Influenza**

### **5.2 Miglioramento integrazione SciML.ai**

### **5.3 Accuratezza nella descrizione del modello**

#### **5.3.1 Happiness**

#### **5.3.2 Interventi localizzati**

#### **5.3.3 Migliore flessibilità generale**

## **6 Conclusioni**

## References

- [1] Sameera Abar, Georgios K. Theodoropoulos, Pierre Lemarinier, and Gregory M.P. O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24:13–33, 2017.
- [2] Mohammad Abavisani, Karim Rahimian, Bahar Mahdavi, Samaneh Tokhanbigli, Mahsa Mollapour Siasakht, Amin Farhadi, Mansoor Kodori, Mohammadamin Mahmanzar, and Zahra Meshkat. Mutations in sars-cov-2 structural proteins: a global analysis. *Virology Journal*, 19(1):220, Dec 2022.
- [3] Linda J. S. Allen. *An Introduction to Stochastic Epidemic Models*, pages 81–130. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [4] Naomi Altman and Martin Krzywinski. Association, correlation and causation. *Nature Methods*, 12(10):899–900, Oct 2015.
- [5] Justin Angevaare, Zeny Feng, and Rob Deardon. Pathogen.jl: Infectious disease transmission network modeling with julia. *Journal of Statistical Software*, 104(4):1–30, 2022.
- [6] Azam Asanjarani, Aminath Shausan, Keng Chew, Thomas Graham, Shane G. Henderson, Hermanus M. Jansen, Kirsty R. Short, Peter G. Taylor, Aapeli Vuorinen, Yuvraj Yadav, Ilze Ziedins, and Yoni Nazarathy. Emulation of epidemics via bluetooth-based virtual safe virus spread: Experimental setup, software, and data. *PLOS Digital Health*, 1(12):1–23, 12 2022.
- [7] Keith R. Bissett, Jose Cadena, Maleq Khan, and Chris J. Kuhlman. Agent-based computational epidemiological modeling. *Journal of the Indian Institute of Science*, 101(3):303–327, Jul 2021.
- [8] Ottar N. Bjørnstad, Katriona Shea, Martin Krzywinski, and Naomi Altman. The seirs model for infectious disease dynamics. *Nature Methods*, 17(6):557–558, Jun 2020.
- [9] Matthew H Bonds, Donald C Keenan, Pejman Rohani, and Jeffrey D Sachs. Poverty trap formed by the ecology of infectious diseases. *Proc Biol Sci*, 277(1685):1185–1192, December 2009.
- [10] Fred Brauer. *Compartmental Models in Epidemiology*, pages 19–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [11] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.
- [12] P. Ciunkiewicz, W. Brooke, M. Rogers, and S. Yanushkevich. Agent-based epidemiological modeling of covid-19 in localized environments. *Computers in Biology and Medicine*, 144:105396, 2022.
- [13] Raj Dandekar, Karen Chung, Vaibhav Dixit, Mohamed Tarek, Aslan Garcia-Valadez, Krishna Vishal Vemula, and Chris Rackauckas. Bayesian neural ordinary differential equations, 2022.
- [14] Raj Dandekar, Shane G. Henderson, Hermanus M. Jansen, Joshua McDonald, Sarat Moka, Yoni Nazarathy, Christopher Rackauckas, Peter G. Taylor, and Aapeli Vuorinen. Safe blues:

The case for virtual safe virus spread in the long-term fight against epidemics. *Patterns*, 2(3):100220, 2021.

- [15] George Datseris, Ali R. Vahdati, and Timothy C. DuBois. Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity. *SIMULATION*, 0(0):003754972110688, January 2022.
- [16] Bernhard Ebbinghaus, Lukas Lehner, and Elias Naumann. Welfare state support during the covid-19 pandemic: Change and continuity in public attitudes towards social policies in germany. *European Policy Analysis*, 8(3):297–311, 2022.
- [17] Abdulrahman M El-Sayed, Peter Scarborough, Lars Seemann, and Sandro Galea. Social network analysis and agent-based modeling in social epidemiology. *Epidemiol Perspect Innov*, 9(1):1, February 2012.
- [18] James Fairbanks, Mathieu Besançon, Schöllly Simon, Júlio Hoffman, Nick Eubank, and Stefan Karpinski. Juliagraphs/graphs.jl: an optimized graphs package for the julia programming language, 2021.
- [19] Mathilde Frérot, Annick Lefebvre, Simon Aho, Patrick Callier, Karine Astruc, and Ludwig Serge Aho Glélé. What is epidemiology? changing definitions of epidemiology 1978-2017. *PLOS ONE*, 13(12):1–27, 12 2018.
- [20] Sandro Galea, Matthew Riddle, and George A Kaplan. Causal thinking and complex system approaches in epidemiology. *Int J Epidemiol*, 39(1):97–106, October 2009.
- [21] Giulia Giordano, Franco Blanchini, Raffaele Bruno, Patrizio Colaneri, Alessandro Di Filippo, Angela Di Matteo, and Marta Colaneri. Modelling the covid-19 epidemic and implementation of population-wide interventions in italy. *Nature Medicine*, 26(6):855–860, Jun 2020.
- [22] Alberto Godio, Francesca Pace, and Andrea Vergnano. Seir modeling of the italian epidemic of sars-cov-2 using computational swarm intelligence. *International Journal of Environmental Research and Public Health*, 17(10), 2020.
- [23] Shashi Gowda, Yingbo Ma, Valentin Churavy, Alan Edelman, and Christopher Rackauckas. Sparsity programming: Automated sparsity-aware optimizations in differentiable programming. 2019.
- [24] Elizabeth R. Groff, Shane D. Johnson, and Amy Thornton. State of the art in agent-based modeling of urban crime: An overview. *Journal of Quantitative Criminology*, 35(1):155–193, Mar 2019.
- [25] World Bank Group. Wdr 2022 chapter 1. introduction, Mar 2023.
- [26] Michael Innes, Elliot Saba, Keno Fischer, Dhairyra Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. Fashionable modelling with flux. *CoRR*, abs/1811.01457, 2018.
- [27] Mike Innes. Flux: Elegant machine learning with julia. *Journal of Open Source Software*, 2018.
- [28] Masoud Jalayer, Carlotta Orsenigo, and Carlo Vercellis. Cov-abm: A stochastic discrete-event agent-based framework to simulate spatiotemporal dynamics of covid-19, 2020.
- [29] Julius Martensen, Christopher Rackauckas, et al. Datadrivendiffeq.jl, July 2021.

- [30] Cliff C. Kerr, Robyn M. Stuart, Dina Mistry, Romesh G. Abeysuriya, Katherine Rosenfeld, Gregory R. Hart, Rafael C. Núñez, Jamie A. Cohen, Prashanth Selvaraj, Brittany Hagedorn, Lauren George, Michał Jastrzębski, Amanda S. Izzo, Greer Fowler, Anna Palmer, Dominic Delport, Nick Scott, Sherrie L. Kelly, Caroline S. Bennette, Bradley G. Wagner, Stewart T. Chang, Assaf P. Oron, Edward A. Wenger, Jasmina Panovska-Griffiths, Michael Famularo, and Daniel J. Klein. Covasim: An agent-based model of covid-19 dynamics and interventions. *PLOS Computational Biology*, 17(7):1–32, 07 2021.
- [31] Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9):093122, sep 2021.
- [32] Garyfallos Konstantoudis, Dominic Schuhmacher, Håvard Rue, and Ben D Spycher. Discrete versus continuous domain models for disease mapping. *Spatial and Spatio-temporal Epidemiology*, 32:100319, 2020.
- [33] James Ladyman, James Lambert, and Karoline Wiesner. What is a complex system? *European Journal for Philosophy of Science*, 3(1):33–67, Jan 2013.
- [34] Yingbo Ma, Vaibhav Dixit, Michael J Innes, Xingjian Guo, and Chris Rackauckas. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9, 2021.
- [35] Peter V. Markov, Mahan Ghafari, Martin Beer, Katrina Lythgoe, Peter Simmonds, Nikolaos I. Stilianakis, and Aris Katzourakis. The evolution of sars-cov-2. *Nature Reviews Microbiology*, 21(6):361–379, Jun 2023.
- [36] Sadegh Marzban, Renji Han, Nóra Juhász, and Gergely Röst. A hybrid PDE-ABM model for viral dynamics with application to SARS-CoV-2 and influenza. *R Soc Open Sci*, 8(11):210787, November 2021.
- [37] Samuel Mwalili, Mark Kimathi, Viona Ojiambo, Duncan Gathungu, and Rachel Mbogo. Seir model for covid-19 dynamics incorporating the environment and social distancing. *BMC Research Notes*, 13(1):352, Jul 2020.
- [38] John T Nardini, Ruth E Baker, Matthew J Simpson, and Kevin B Flores. Learning differential equation models from stochastic agent-based model simulations. *J R Soc Interface*, 18(176):20200987, March 2021.
- [39] Hiroshi Nishiura. Correcting the actual reproduction number: a simple method to estimate  $r(0)$  from early epidemic growth data. *Int J Environ Res Public Health*, 7(1):291–302, January 2010.
- [40] Aleksandar Novakovic and Adele H. Marshall. The cp-abm approach for modelling covid-19 infection dynamics and quantifying the effects of non-pharmaceutical interventions. *Pattern Recognition*, 130:108790, 2022.
- [41] World Health Organization. Who coronavirus (covid-19) dashboard.
- [42] Avik Pal. Lux: Explicit Parameterization of Deep Neural Networks in Julia, 2023. If you use this software, please cite it as below.

- [43] M Parascandola and D L Weed. Causation in epidemiology. *J Epidemiol Community Health*, 55(12):905–912, December 2001.
- [44] Sunny Prakash Prajapati, Rahul Bhaumik, and Tarun Kumar. An intelligent abm-based framework for developing pandemic-resilient urban spaces in post-covid smart cities. *Procedia Computer Science*, 218:2299–2308, 2023. International Conference on Machine Learning and Data Engineering.
- [45] Chris Rackauckas, Mike Innes, Yingbo Ma, Jesse Bettencourt, Lyndon White, and Vaibhav Dixit. Diffeqflux.jl-a julia library for neural differential equations. *arXiv preprint arXiv:1902.02376*, 2019.
- [46] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, and Ali Ramadhan. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.
- [47] Christopher Rackauckas and Qing Nie. Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1):15, 2017.
- [48] Christopher Rackauckas and Qing Nie. Confederated modular differential equation apis for accelerated algorithm development and benchmarking. *Advances in Engineering Software*, 132:1–6, 2019.
- [49] Pedro Sanchez, Jeremy P. Voisey, Tian Xia, Hannah I. Watson, Alison Q. O’Neil, and Sotirios A. Tsaftaris. Causal machine learning for healthcare and precision medicine. *Royal Society Open Science*, 9(8):220638, 2022.
- [50] Constantinos I. Siettos and Lucia Russo. Mathematical modeling of infectious disease dynamics. *Virulence*, 4(4):295–306, 2013. PMID: 23552814.
- [51] Eric Silverman, Umberto Gostoli, Stefano Picascia, Jonatan Almagor, Mark McCann, Richard Shaw, and Claudio Angione. Situating agent-based modelling in population health research. *Emerging Themes in Epidemiology*, 18(1):10, Jul 2021.
- [52] Melissa Tracy, Magdalena Cerdá, and Katherine M Keyes. Agent-Based modeling in public health: Current applications and future directions. *Annu Rev Public Health*, 39:77–94, January 2018.
- [53] Shihang Wang, Xuanyu Xu, Cai Wei, Sicong Li, Jingying Zhao, Yin Zheng, Xiaoyu Liu, Xiaomin Zeng, Wenliang Yuan, and Sihua Peng. Molecular evolutionary characteristics of sars-cov-2 emerging in the united states. *Journal of Medical Virology*, 94(1):310–317, 2022.
- [54] Christopher W. Weimer, J. O. Miller, and Raymond R. Hill. Agent-based modeling: An introduction and primer. In *2016 Winter Simulation Conference (WSC)*, pages 65–79, 2016.
- [55] Wikipedia. Agent-based model — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Agent-based%20model&oldid=1148935023>, 2023. [Online; accessed 30-April-2023].
- [56] Wikipedia. Brownian motion — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Brownian%20motion&oldid=1155409035>, 2023. [Online; accessed 21-May-2023].

- [57] Wikipedia. Causality — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Causality&oldid=1150225747>, 2023. [Online; accessed 27-April-2023].
- [58] Wikipedia. Compartmental models in epidemiology — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Compartmental%20models%20in%20epidemiology&oldid=1150020798>, 2023. [Online; accessed 27-April-2023].
- [59] Wikipedia. Correlation — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Correlation&oldid=1149809031>, 2023. [Online; accessed 21-May-2023].
- [60] Wikipedia. Cycle of poverty — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Cycle%20of%20poverty&oldid=1151766525>, 2023. [Online; accessed 27-April-2023].
- [61] Wikipedia. Differential-algebraic system of equations — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Differential-algebraic%20system%20of%20equations&oldid=1159353421>, 2023. [Online; accessed 16-June-2023].
- [62] Wikipedia. Epidemiologia — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Epidemiologia&oldid=124857046>, 2023. [Online; accessed 27-April-2023].
- [63] Wikipedia. Equazione differenziale ordinaria — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Equazione%20differenziale%20ordinaria&oldid=130359533>, 2023. [Online; accessed 21-May-2023].
- [64] Wikipedia. Immunità di gregge — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Immunit%C3%A0%20di%20gregge&oldid=133955463>, 2023. [Online; accessed 17-June-2023].
- [65] Wikipedia. Incubation period — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Incubation%20period&oldid=1152591975>, 2023. [Online; accessed 21-May-2023].
- [66] Wikipedia. Influenza spagnola — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Influenza%20spagnola&oldid=132751127>, 2023. [Online; accessed 27-April-2023].
- [67] Wikipedia. Julia (programming language) — Wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Julia%20\(programming%20language\)&oldid=1151847776](http://en.wikipedia.org/w/index.php?title=Julia%20(programming%20language)&oldid=1151847776), 2023. [Online; accessed 27-April-2023].
- [68] Wikipedia. Numero di riproduzione di base — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Numero%20di%20riproduzione%20di%20base&oldid=133906348>, 2023. [Online; accessed 17-June-2023].
- [69] Wikipedia. Pandemia — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Pandemia&oldid=132968684>, 2023. [Online; accessed 27-April-2023].
- [70] Wikipedia. Pandemia di COVID-19 — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Pandemia%20di%20COVID-19&oldid=133076664>, 2023. [Online; accessed 27-April-2023].

- [71] Wikipedia. Peste nera — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Peste%20nera&oldid=132822978>, 2023. [Online; accessed 27-April-2023].
- [72] Wikipedia. Positive feedback — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Positive%20feedback&oldid=1147282594>, 2023. [Online; accessed 27-April-2023].
- [73] Wikipedia. Simpson's paradox — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Simpson's%20paradox&oldid=1154565479>, 2023. [Online; accessed 21-May-2023].
- [74] Wikipedia. Simulation software — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Simulation%20software&oldid=1149041989>, 2023. [Online; accessed 27-April-2023].
- [75] Wikipedia. Smoothness — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Smoothness&oldid=1142265373>, 2023. [Online; accessed 16-June-2023].
- [76] Wikipedia. Sparse identification of non-linear dynamics — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Sparse%20identification%20of%20non-linear%20dynamics&oldid=1157228086>, 2023. [Online; accessed 16-June-2023].
- [77] Wikipedia. Spurious relationship — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Spurious%20relationship&oldid=1139468569>, 2023. [Online; accessed 30-April-2023].
- [78] Wikipedia. Storia della pandemia di AIDS — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Storia%20della%20pandemia%20di%20AIDS&oldid=132423486>, 2023. [Online; accessed 27-April-2023].
- [79] Wikipedia. Tifo esantematico — Wikipedia, the free encyclopedia. <http://it.wikipedia.org/w/index.php?title=Tifo%20esantematico&oldid=132730114>, 2023. [Online; accessed 27-April-2023].
- [80] Wikipedia. Universal differential equation — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Universal%20differential%20equation&oldid=1138859269>, 2023. [Online; accessed 16-June-2023].
- [81] Wikipedia. Universal Turing machine — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Universal%20Turing%20machine&oldid=1149333304>, 2023. [Online; accessed 16-June-2023].
- [82] Nana Wu, Keven Joyal-Desmarais, Paula A B Ribeiro, Ariany Marques Vieira, Jovana Stojanovic, Comfort Sanuade, Doro Yip, and Simon L Bacon. Long-term effectiveness of COVID-19 vaccines against infections, hospitalisations, and mortality in adults: findings from a rapid living systematic evidence synthesis and meta-analysis up to December, 2022. *Lancet Respir. Med.*, 11(5):439–452, May 2023.
- [83] Hualei Xin, Yu Li, Peng Wu, Zhili Li, Eric H Y Lau, Ying Qin, Liping Wang, Benjamin J Cowling, Tim K Tsang, and Zhongjie Li. Estimating the Latent Period of Coronavirus Disease 2019 (COVID-19). *Clinical Infectious Diseases*, 74(9):1678–1681, 09 2021.

- [84] J H Zaccai. How to assess epidemiological studies. *Postgrad Med J*, 80(941):140–147, March 2004.