

A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time

*Português: Um Novo Procedimento de Simulação de Monte Carlo para Modelar a Disseminação da COVID-19 ao Longo do Tempo

Autor da obra original, **Gang Xie Research Office**
Charles Sturt University
Wagga Wagga, Australia

1nd Revisor: Adrian Alejandro Chavez Alanes, *Programa de Pós-Graduação (PPG) em Telecomunicações*
Instituto Nacional de Telecomunicações
Santa Rita do Sapucaí, Brasil

2nd Revisor: Everton Vilhena Cardoso, *Programa de Pós-Graduação (PPG) em Telecomunicações*
Instituto Nacional de Telecomunicações
Santa Rita do Sapucaí, Brasil

Resumo—Este projeto mostra a reprodução, em Python, do modelo estocástico sugerido por Gang Xie (2020) para simular a propagação da COVID-19 por meio do método de Monte Carlo. Ao contrário dos modelos determinísticos convencionais (como SIR), a metodologia de Xie considera cada pessoa como uma entidade independente, empregando distribuições de Poisson para o total de infecções produzidas e binomial negativa para o intervalo até a infecciosidade. A simulação reproduz a disseminação diária do vírus, levando em conta variados valores de R_t (taxa de reprodução), μT (tempo médio até infecção) e $sizeV$ (variabilidade), com e sem medidas interventivas. Realizaram-se 50 experimentos para cada situação, sendo analisadas as médias diárias de casos ativos. Os resultados reproduziram com precisão as curvas esperadas, demonstrando que intervenções precoces (diminuição de R_t) aplanam a curva epidêmica. A modelagem por Monte Carlo provou ser eficaz para integrar aleatoriedade e analisar diversos cenários de maneira flexível, confirmando a consistência do modelo original no novo ambiente computacional.

I. INTRODUÇÃO

A modelagem matemática tem sido amplamente utilizada para estudar a propagação de doenças infectocontagiosas, auxiliando na compreensão da dinâmica epidêmica e na formulação de políticas públicas.

Durante a pandemia da doença causada pelo coronavírus SARS-CoV-2 (COVID-19), modelos computacionais tornaram-se ferramentas fundamentais para prever a evolução de casos, estimando a carga sobre os sistemas de saúde mundiais, tornando possível avaliar o impacto de medidas de contenção.

Modelos determinísticos baseados em compartimentos, como o modelo **Susceptível-Infectado-Recuperado (SIR)**[Sat90], assumem transições médias e homogêneas entre grupos populacionais. Embora eficientes em muitos contextos, tais modelos nem sempre representam a natureza aleatória de eventos como o número de pessoas que um

indivíduo pode infectar ou o tempo necessário para que uma pessoa recém-infectada se torne contagiosa.

Diante dessas limitações, Gang Xie [Xie20] propôs em 2020 um modelo baseado em simulação de Monte Carlo, utilizando um processo estocástico para representar a disseminação da COVID-19. No modelo, cada indivíduo é tratado como uma unidade independente, e a transmissão do vírus é modelada com distribuições probabilísticas, como a distribuição de Poisson para o número de novas infecções e a distribuição binomial negativa para o tempo até que um novo infectado se torne infeccioso.

II. METODOLOGIA

A. Descrição do artigo

O artigo “A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time” de Gang Xie, propõe uma abordagem alternativa aos modelos compartimentais clássicos (como SIR) para simular a propagação da COVID-19. E em vez de dividir a população em compartimentos determinísticos (suscetível, infectado, recuperado), o autor adota um modelo estocástico baseado em processos de ponto aleatórios, onde cada indivíduo é tratado como uma unidade independente que pode transmitir a infecção segundo determinadas distribuições probabilísticas.

Princípios do modelo: O modelo é fundamentado em dois aspectos-chave da dinâmica da doença:

1) Número de pessoas infectadas por cada indivíduo infeccioso ativo

Isso é modelado usando uma **distribuição de Poisson** com média R_t , que representa a **taxa de reprodução** ou número médio de pessoas que um indivíduo *infecta* durante seu período infeccioso.

Um valor alto de R_t (ex: 3) implica em alta transmissibilidade, gerando uma curva de crescimento epidêmico rápida. Já valores baixos de R_t (ex: 0,5) indicam controle da disseminação da doença.

2) Tempo até que cada novo indivíduo seja infectado

Este tempo é modelado com uma **distribuição binomial negativa**, que depende de dois parâmetros:

- `muT`: valor médio de dias para que uma transmissão ocorra
- `sizeV`: parâmetro de dispersão, onde quanto menor o valor de `sizeV`, maior a variabilidade do tempo de infecção

Principais achados:

- `n0`: número inicial de pessoas infecciosas no início da simulação. %
- `pp`: proporção da população imune (default: 0, ou seja, toda população é suscetível). %
- `limit`: tamanho máximo da população (por exemplo, 1 milhão). %
- `days`: duração da janela de observação total (por exemplo, 300 dias). %
- `nd`: período da simulação ativa (por exemplo, 100 dias). %
- `Rt`: vetor de valores de R_t para cada dia de simulação. Pode ser constante ou variável ao longo do tempo. %

2) *Processo de Simulação*: O modelo simula dia após dia, rastreando quais indivíduos estão infecciosos e quantas novas transmissões ocorrem. Quando uma nova infecção acontece, um tempo até a infecção ser efetivada é sorteado da distribuição binomial negativa. Ao atingir esse tempo, o indivíduo passa a ser infeccioso e pode gerar novas infecções, até que expire seu potencial de transmissão (também modelado estatisticamente).

A simulação *continua* por uma quantidade de dias definida, acumulando o número de casos ativos por dia. Por ser um processo estocástico, o modelo é executado centenas ou milhares de vezes (usualmente com *bootstrap* de 1000 execuções) para estimar a mediana e o intervalo interquartil dos resultados.

3) Importância dos parâmetros::

- Um R_t maior \rightarrow mais casos novos por dia \rightarrow pico mais alto.
- Um `muT` menor \rightarrow infecções mais rápidas \rightarrow curva mais íngreme.
- Um `sizeV` menor \rightarrow maior variabilidade \rightarrow comportamento mais imprevisível (picos mais agudos ou dispersos).

Essa abordagem permite incorporar intervenções (mudanças em R_t), imunidade adquirida (`pp`) e testar cenários “what-if” com maior flexibilidade que modelos determinísticos tradicionais.

B. Simulação com Monte Carlo

O objetivo desta seção é descrever detalhadamente a implementação em Python do modelo proposto por Xie[Xie20], com foco na lógica estatística envolvida, nas distribuições probabilísticas utilizadas e no papel central do método de Monte Carlo em todo o processo.

Monte Carlo é uma técnica de simulação baseada em geração de números aleatórios para modelar sistemas complexos ou incertos. No contexto da COVID-19, é usada para

simular como a infecção se propaga em uma população, levando em conta a variabilidade natural do processo (ex: nem todos infectam o mesmo número de pessoas, nem todos infectam no mesmo tempo).

A essência do método está em repetir muitas vezes um experimento computacional (neste caso, o espalhamento do vírus), utilizando distribuições de probabilidade para gerar os eventos relevantes. Cada execução do modelo (uma “corrida” da simulação) gera uma possível realidade. Ao executar o modelo diversas vezes, podemos observar a distribuição de possíveis cenários e tirar conclusões com base na mediana, médias, percentis etc.

As distribuições utilizadas no modelo são:

1) *Distribuição de Poisson*: A distribuição de Poisson é utilizada para modelar quantas pessoas um indivíduo infeccioso poderá contagiar. O parâmetro da Poisson é a taxa de reprodução (R_t), que define o número médio de novos infectados.

- Antes da intervenção: $R_t = 3.2$
- Após a intervenção: $R_t = 0.8$

Valores mais altos de R_t geram crescimento acelerado da epidemia, enquanto valores abaixo de 1 indicam contenção da transmissão.

- R_t maior \rightarrow maior número médio de novos infectados por indivíduo \rightarrow crescimento acelerado da epidemia.
- R_t menor \rightarrow propagação mais controlada ou até decrescente.

2) *Distribuição Binomial Negativa*: A distribuição binomial negativa é utilizada para definir o número de dias que leva até que um novo infectado se torne infeccioso.

- `muT` = 4.0 (tempo médio entre infecções, mantido nos dois cenários)
- `sizeV` = 1.0 antes da intervenção; `sizeV` = 0.6 após a intervenção (maior variabilidade com intervenção)

Valores menores de `sizeV` e `muT` aumentam a variabilidade do tempo de infecção.

- `muT` pequeno \rightarrow infecção ocorre mais rapidamente \rightarrow curvas com picos mais cedo.
- `sizeV` pequeno \rightarrow maior variabilidade \rightarrow imprevisibilidade no momento do pico.

C. Estrutura da simulação

A função *simular cenário* executa a simulação de propagação dia a dia, seguindo os seguintes passos:

- 1) Inicializa-se com 1 indivíduo infeccioso.
- 2) A cada dia, são verificados os indivíduos que se tornam infecciosos. Para cada um deles:
 - Sorteia-se quantas pessoas ele infectará (Poisson).
 - Sorteia-se em quantos dias essas novas infecções se tornarão ativas (Binomial Negativa).
 - Define-se o tempo de recuperação (entre 10 e 15 dias sem intervenção; entre 4 e 6 dias com intervenção).
- 3) O processo se repete por 80 dias, controlando o número máximo de infectados (4000).

- 4) Ao final de cada simulação, é armazenado o vetor de casos ativos por dia.

Esse processo é repetido 50 vezes para cada cenário, com os resultados armazenados em matrizes 2D (simulacoes sem intervencao e simulacoes com intervencao). Em seguida, calcula-se a média por dia entre as 50 execuções, obtendo curvas médias suaves e representativas para análise comparativa.

A geração de números aleatórios ocorre em dois pontos-chave do processo: na determinação do número de novas infecções (via distribuição de Poisson) e na definição do tempo até que essas infecções se tornem ativas (via binomial negativa). O uso de múltiplas execuções permite suavizar a variabilidade natural do modelo e obter curvas médias representativas para cada situação.

Tabela 1. Resumo das variáveis principais no código.

Variável	Descrição
TOTAL_DIAS	Duração da simulação (80 dias)
INFECTADOS_INICIAIS	Número de pessoas infectadas no início (1)
NUM_SIMULACOES	Número de simulações independentes por cenário (50)
Rt_antes / Rt_depois	Taxa de reprodução antes e depois da intervenção
muT_antes / muT_depois	Tempo médio entre infecções (mantido igual nos dois cenários: 4.0)
sizeV_antes / sizeV_depois	Dispersão da infecção (reduzida após intervenção)
DIA_INTERVENCAO	Dia em que os parâmetros mudam para refletir a intervenção (dia 20)

III. RESULTADOS

A simulação do modelo estocástico proposto por Gang Xie [Xie20] foi implementada em Python com dois cenários distintos: um com intervenção no dia 20 da simulação e outro sem qualquer intervenção. Cada cenário foi executado 50 vezes, e os resultados médios de casos ativos por dia foram utilizados para gerar os gráficos comparativos.

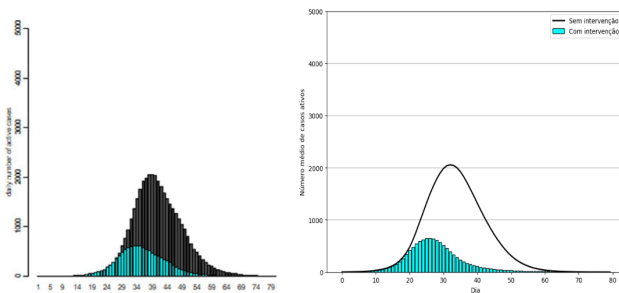


Figura 1. (a) Gráfico original do artigo. (b) Gráfico gerado na simulação em Python.

Ambos os gráficos ilustram a evolução do número de casos ativos de COVID-19 ao longo do tempo, destacando o impacto de medidas de intervenção. No gráfico gerado neste trabalho, observa-se que:

- O cenário sem intervenção resulta em um crescimento rápido e sustentado dos casos ativos.

- Já o cenário com intervenção no dia 20 mostra uma redução significativa na propagação do vírus, com um pico mais baixo e uma queda progressiva no número de casos ativos a partir do momento da intervenção.
- O comportamento geral das curvas está de acordo com o que foi descrito no artigo: a intervenção precoce, representada pela queda na taxa de reprodução (R_t), leva ao achatamento da curva.

IV. CONCLUSÃO

A simulação desenvolvida em Python reproduziu com sucesso os principais comportamentos do modelo original proposto por Gang Xie [Xie20], demonstrando coerência nas curvas de casos ativos sob diferentes combinações de parâmetros epidemiológicos.

Apesar das limitações, como pequenas diferenças numéricas e a conversão entre linguagens (R[R C24] para Python[Pyt24]), o modelo replicado preservou as respostas esperadas à variação de R_t , μT e sizeV . A abordagem de Monte Carlo mostrou-se adequada para representar a aleatoriedade natural da propagação viral e permite testar múltiplos cenários de forma flexível.

Com base nisso, conclui-se que o gráfico original pôde ser reproduzido de maneira satisfatória utilizando simulação de Monte Carlo em Python[Pyt24], respeitando os princípios estatísticos descritos no artigo e adaptando-os corretamente ao ambiente Colab[Goo24].

REFERÊNCIAS

- [Goo24] GOOGLE. *Google Colaboratory*. <https://colab.research.google.com/>. Acesso em: 4 maio 2025. 2024.
- [Pyt24] PYTHON SOFTWARE FOUNDATION. *Python Programming Language*. [urlhttps://www.python.org/](https://www.python.org/). Accessed: 2025-05-04. 2024.
- [R C24] R CORE TEAM. *R: A Language and Environment for Statistical Computing*. [urlhttps://www.r-project.org/](https://www.r-project.org/). Accessed: 2025-05-04. 2024.
- [Sat90] SATTENSPIEL, L. «Yearbook of Physical Anthropology». Em: *Yearbook of Physical Anthropology*. Hoboken: Wiley-Liss Inc., 1990, pp. 245–276.
- [Xie20] XIE, G. «A novel Monte Carlo simulation procedure for modelling COVID-19 spread over time». Em: *Scientific Reports* 10 (2020), p. 13120. DOI: 10.1038/s41598-020-70091-1. URL: <https://doi.org/10.1038/s41598-020-70091-1>.