Análise matemática epidemiológica do Coronavírus (modelo SIR)

Ângelo Augusto Matteoni de Souza;
Markus Alexandre Alves da Silva;
Leonardo Mazzamboni Colussi;
Gabriela Haddad Grubl;
Amanda Rocha Faga

Julho 2020

1 Introdução

Nesse projeto, apresentaremos um breve estudo de análise matemática da epidemia do Coronavírus no primeiro semestre de 2020 no estado de São Paulo (SP). Para isso, utilizaremos os dados disponíveis na página Painel Coronavírus do professor Dr. Alberto Saa[1] e do Ministério da Saúde (MS)[2] aplicados ao modelo SIR (Sustíveis, Infectados e Recuperados) recorrendo ao método matemático de Runge-Kutta de quarta ordem[3] para solucionar o sistema de equações diferenciais ordinárias (EDOs) presente no modelo. Todavia, aproveitaremos a flexibilidade do método para simular diversos cenários e condições iniciais.

2 Desenvolvimento

2.1 Modelo SIR

O modelo matemático SIR consiste de um sistema de equações diferenciais ordinárias (EDOs) o que possibilita a análises da progressão de uma determinada doença, avaliando casos possíveis de uma epidemia. Assim, o modelo SIR segue a seguinte dinâmica:

$$\frac{dS}{dt} = -\gamma ro\frac{IS}{N} \tag{1}$$

$$\frac{dI}{dt} = \gamma(ro\frac{IS}{N} - I) \tag{2}$$

$$\frac{dR}{dt} = \gamma I \tag{3}$$

Em que N é um valor fixo que representa a população total de indivíduos estudada dividida em três classes: S, o número de indivíduos suscetíveis, I, o número de indivíduos infectados e R, o número de indivíduos recuperados. Desse modo, segue que:

$$N = S + I + R$$

Ainda, podemos determinar o número básico reprodutivo como $ro = \frac{\beta}{\gamma}$, onde:

- $\beta > 0$ é a taxa de infecção $[T^{-1}]$;
- $\gamma > 0$ é a taxa de remoção $[T^{-1}]$.

Sendo assim, se $\beta > \gamma$ ($ro > 1$), o cenário é propício ao aumento da epidemia. Consequentemente, se $\gamma > \beta$ ($ro < 1$), o cenário não é propício à epidemia e o número de infectados tende a diminuir conforme o passar do tempo.

2.2 Método de Runge-Kutta

O método de Runge-Kutta é uma ferramenta bastante útil para resolver numéricamente os problemas de valor inicial (PVI), uma vez que ele nos isenta de cálculos de derivadas. Desse modo, para esse projeto, adotaremos o método de Runge-Kutta de quarta ordem (RK4), com passos de 1 dia (ou seja, h = 1 dia), e, para efeito de cálculo e de conveniência, utilizaremos recursivamente a forma matricial a seguir:

$$\begin{bmatrix} S \\ I \\ R \end{bmatrix}^{(i+1)} \approx \begin{bmatrix} S \\ I \\ R \end{bmatrix}^{(i)} + \frac{h}{6}\Big[(k_1 + 2k_2 + 2k_3 + k_4)\Big]^{(i)} \tag{4}$$

sendo,

$$\Big[k_1\Big]^{(i)} = \begin{bmatrix} -\gamma ro\dfrac{I^{(i)}S^{(i)}}{N} \\ \gamma(ro\dfrac{I^{(i)}S^{(i)}}{N} - I^{(i)}) \\ \gamma I^{(i)} \end{bmatrix} \tag{5}$$

$$\Big[k_2\Big]^{(i)} = \begin{bmatrix} -\dfrac{\gamma ro}{N}(I^{(i)} + \dfrac{k_1^{(2,i)}}{2})(S^{(i)} + \dfrac{k_1^{(1,i)}}{2}) \\ \gamma(\dfrac{ro}{N}(I^{(i)} + \dfrac{k_1^{(2,i)}}{2})(S^{(i)} + \dfrac{k_1^{(1,i)}}{2}) - (I^{(i)} + \dfrac{k_1^{(2,i)}}{2})) \\ \gamma(I^{(i)} + \dfrac{k_1^{(2,i)}}{2}) \end{bmatrix} \tag{6}$$

$$\Big[k_3\Big]^{(i)} = \begin{bmatrix} -\dfrac{\gamma ro}{N}(I^{(i)} + \dfrac{k_2^{(2,i)}}{2})(S^{(i)} + \dfrac{k_2^{(1,i)}}{2}) \\ \gamma(\dfrac{ro}{N}(I^{(i)} + \dfrac{k_2^{(2,i)}}{2})(S^{(i)} + \dfrac{k_2^{(1,i)}}{2}) - (I^{(i)} + \dfrac{k_2^{(2,i)}}{2})) \\ \gamma(I^{(i)} + \dfrac{k_2^{(2,i)}}{2}) \end{bmatrix} \tag{7}$$

$$\Big[k_4\Big]^{(i)} = \begin{bmatrix} -\dfrac{\gamma ro}{N}(I^{(i)} + k_3^{(2,i)})(S^{(i)} + k_3^{(1,i)}) \\ \gamma(\dfrac{ro}{N}(I^{(i)} + k_3^{(2,i)})(S^{(i)} + k_3^{(1,i)}) - (I^{(i)} + k_3^{(2,i)})) \\ \gamma(I^{(i)} + k_3^{(2,i)}) \end{bmatrix} \tag{8}$$

3 Análises e Resultados

3.1 Curva de infectados e removidos em função do tempo com r_0 constante

Os gráficos a seguir mostram a quantidade de infectados e removidos ao longo do tempo a partir do dia 14/07/2020. Seguindo esses parâmetros, temos que os tempos calculados para o pico e para o fim da pandemia são de 265 e 1138 dias, respectivamente.

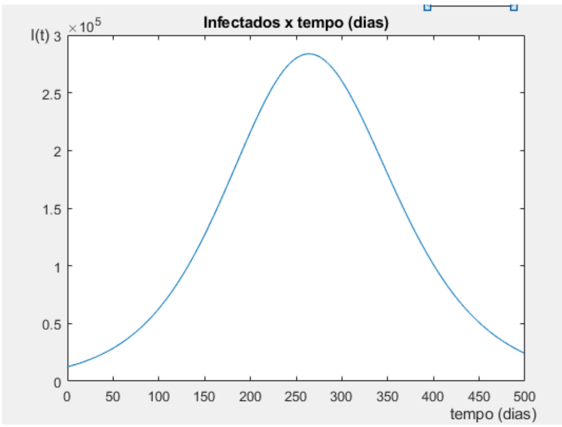

Figura 1: Infectados por tempo com $r_o = 1,13$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

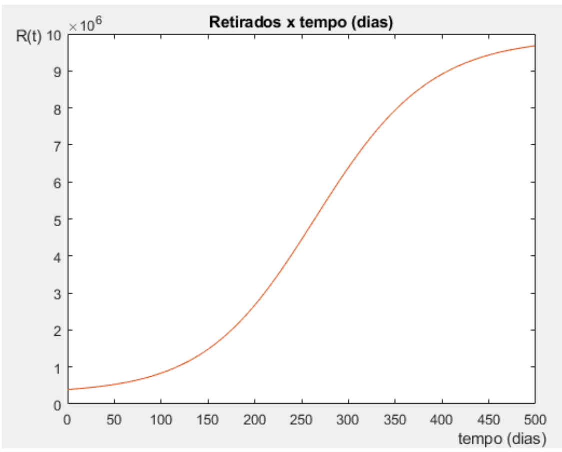

Figura 2: Removidos por tempo com $r_o = 1,13$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

Para os próximos gráficos foi utilizado uma valor de 3,1 para r_o. Desse modo, os tempos calculados para o pico e o fim da pandemia são de 31 e 179 dias, respectivamente. Note que, com um valor maior de r_o, a pandemia seria mais ”curta”. Isso se deve ao fato de que um valor maior de r_o implica em uma quantidade maior de infecções por dia e, como observado na terceira equação (3), quanto maior o número de infectados maior será o número de recuperados. Por essa razão, a pandemia seria encurtada com r_o maior, fato conhecido como ”imunização de rebanho”.

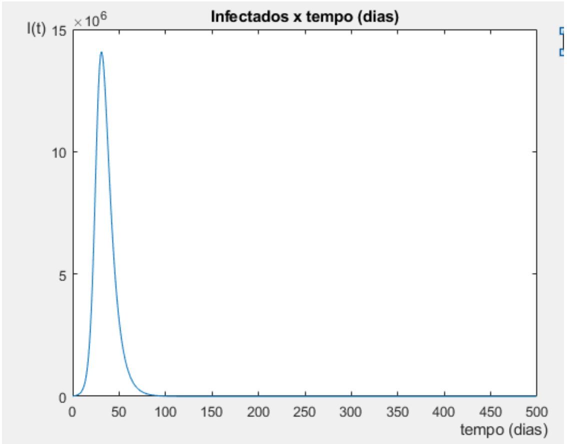

Figura 3: Infectados por tempo com $r_o = 3,1$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

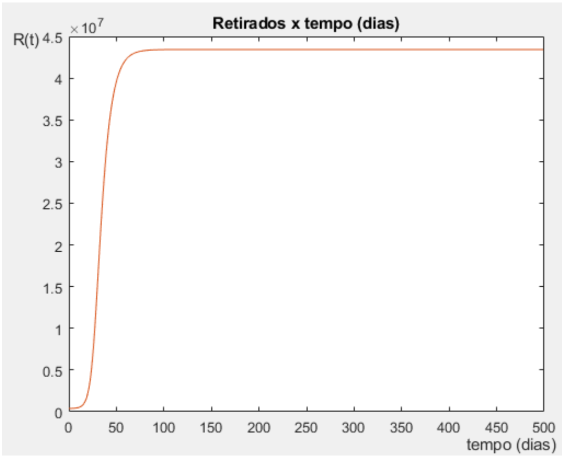

Figura 4: Removidos por tempo com $r_o = 3,1$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

A seguir, foi utilizado r_o sendo 0,88 e o tempo calculado para o fim da pandemia foi de 522 dias. Dessa forma, como r_o é menor que 1, o número de recuperados supera o de infectados. Porém, pelo equação (3) o número de recuperados é influenciado pelo número de infectados, assim, como a quantidade de infectados diminui com o tempo, o de recuperados também, estendendo o fim da pandemia.

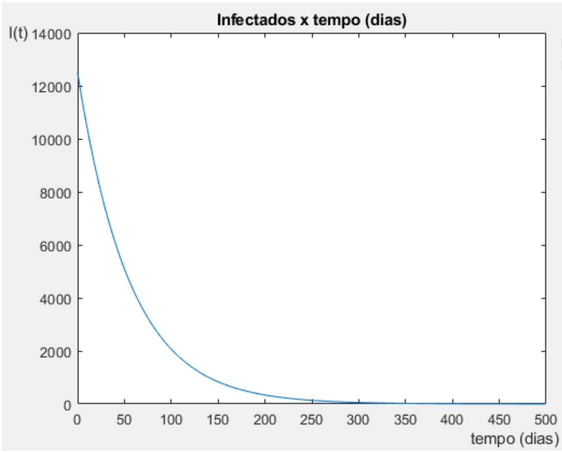

Figura 5: Infectados por tempo com $r_o = 0,88$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

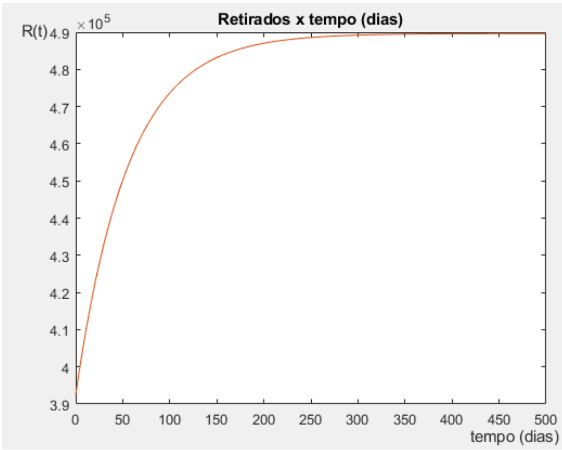

Figura 6: Removidos por tempo com $r_o = 0,88$ do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

3.2 Número básico reprodutivo (r_0) em função do tempo (t)

Usamos o software *GetDatadigitizer* para extrair os dados do gráfico de r_o ilustrado abaixo.

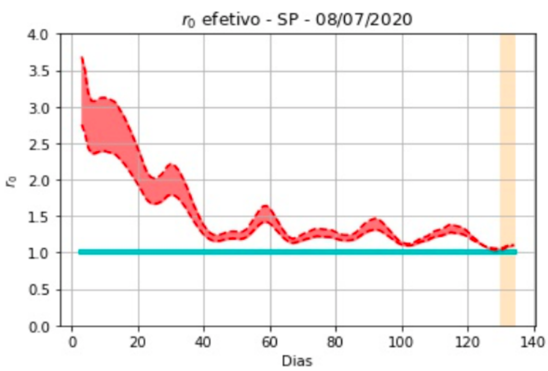

Figura 7: r_o do estado de São Paulo[1]

O software nos devolveu mais de um valor de r_o por dia e por isso fizemos uma seleção dos dados optando por aqueles que correspondiam à parte superior da curva. Usando o *excel* foi possível plotar os gráficos abaixo.

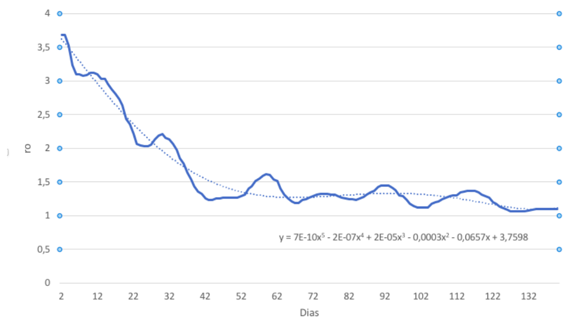

Figura 8: Fetting dos dados de r_o por um polinomio de sexta ordem

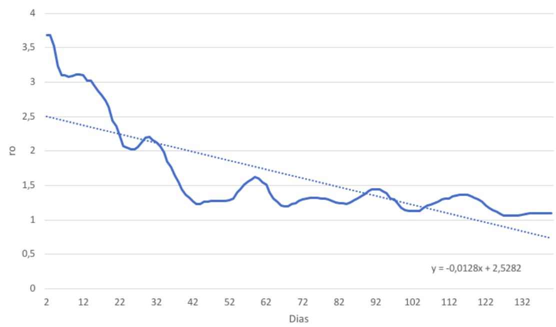

Figura 9: Fetting linear dos dados de r_0

Assim, conseguimos as seguintes equações para r_0 como uma função do tempo:

$$r_0 = -0,0128t + 2,5282 \tag{9}$$

$$r_0 = 7 \cdot 10^{-10}t^5 - 2 \cdot 10^{-7}t^4 + 2 \cdot 10^{-5}t^3 - 3 \cdot 10^{-4}t^2 - 6,57 \cdot 10^{-2}t + 3,7598 \tag{10}$$

Os próximos gráficos simulam um cenário futuro provável com r_o variável, como calculado anteriormente nesta seção, a partir do dia 14/07/2020, e apresenta uma previsão para o fim da pandemia em 139 dias.

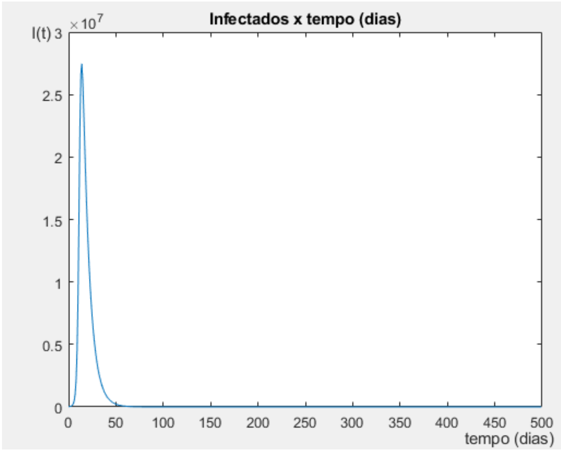

Figura 10: Infectados por tempo com r_o variando em função do tempo do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

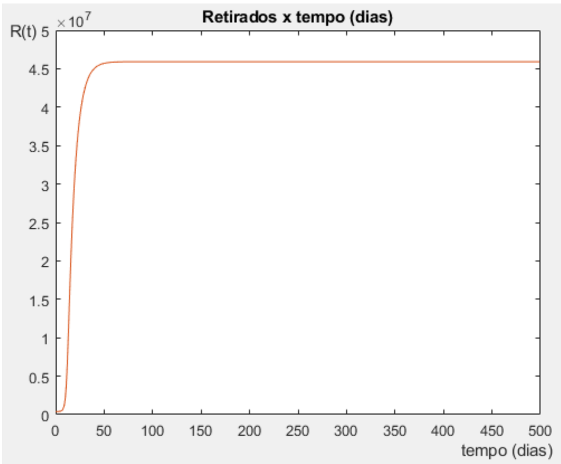

Figura 11: Removidos por tempo com r_o variando em função do tempo do estado de São Paulo a partir do dia 14 de Julho de 2020[1]

O grupo analisou que, a partir da comparação dos dados e gráficos das seções 3.1 e 3.2, não é eficiente considerar r_o constante, pois isso implicaria em considerar que a taxa de infecção é constante durante todo o período da pandemia, fato que não ocorre na prática, uma vez que a quantidade alta de infectados resulta em uma baixa taxa de infecção uma vez que a quantidade de pessoas suscetíveis à infecção diminui e, consequentemente, a taxa de infecção também, diminuindo o r_o. Sendo assim, não faz sentido considerar r_o fixo, quando na prática ele é variável.

4 Conclusão

Por fim, levando em consideração o caso mais realista, ou seja, com o r_o variando com o tempo (Figura 10 e Figura 11), foi possível observar que a pandemia acabaria somente em 139 dias a partir do dia 14 de Julho de 2020. Essa conclusão é válida somente se o r_o continuar diminuindo segundo a Equação 10, caso contrário, esse tempo tende a aumentar, pelo fato do r_o também aumentar ao invés de diminuir, pois o r_o não seguirá mais a Equação 10, isso poderia acontecer no caso da quarentena acabar antes do fim da pandemia.

Referências

[1] Painel Coronavírus, organizado pelo Professor Alberto Saa.
http://vigo.ime.unicamp.br/COVID/

[2] Dados da epidemia de COVID-19 publicados diariamente pelo Ministério da Saúde.
https://covid.saude.gov.br/

[3] Ruggiero, M. A. G e Lopes, V. L. R.: Cálculo Numérico, Segunda Edição, Pearson 1998

Anexo - Código MATLAB para a obtenção dos gráficos I(t) e R(t)

```
dt=500;
sir=zeros(3,dt);
r0=_; % para r0 constante
c=0.14;
n=45919049; % população inicial de sp
h=1;
sir(1,1)=n-_-(_-_)-_; % população - casos no dia - (numeros acumulado de casos - numero de casos no dia) - numero acumulado de mortes
sir(2,1)=_; % numero de casos no dia
sir(3,1)=(_-_)+_; % (numeros acumulado de casos - numero de casos no dia) + numero acumulado de mortes
i=0;
t0=141; % para r0 inconstante
for e=randn(0,dt)
  t0=t0+1; % para r0 inconstante
  r0=0.0000000007*(t0^5)-0.0000002*(t0^4)+0.00002*(t0^3)-0.0003*(t0^2)-0.0657*t0+3.7598; % para r0 inconstante
  i=i+1;
  if i~=1
    k1=[h*((-c)*r0*sir(2,i-1)*sir(1,i-1)/n)
      h*c*((r0*sir(2,i-1)*sir(1,i-1)/n)-sir(2,i-1))
      h*c*sir(2,i-1)];
    k2=[h*((-c)*r0*(sir(2,i-1)+(k1(2,1)/2))*(sir(1,i-1)+(k1(1,1)/2))/n)
      h*c*((r0*(sir(2,i-1)+(k1(2,1)/2))*(sir(1,i-1)+(k1(1,1)/2))/n)-(sir(2,i-1)+(k1(2,1)/2)))
      h*c*(sir(2,i-1)+(k1(2,1)/2))];
    k3=[h*((-c)*r0*(sir(2,i-1)+(k2(2,1)/2))*(sir(1,i-1)+(k2(1,1)/2))/n)
      h*c*((r0*(sir(2,i-1)+(k2(2,1)/2))*(sir(1,i-1)+(k2(1,1)/2))/n)-(sir(2,i-1)+(k2(2,1)/2)))
      h*c*(sir(2,i-1)+(k2(2,1)/2))];
    k4=[h*((-c)*r0*(sir(2,i-1)+(k3(2,1)))*(sir(1,i-1)+(k3(1,1)))/n)
      h*c*((r0*(sir(2,i-1)+(k3(2,1)))*(sir(1,i-1)+(k3(1,1)))/n)-(sir(2,i-1)+(k3(2,1))))
      h*c*(sir(2,i-1)+(k3(2,1)))];
    sir(1,i)=sir(1,i-1)+(1/6)*(k1(1,1)+2*k2(1,1)+2*k3(1,1)+k4(1,1));
    sir(2,i)=sir(2,i-1)+(1/6)*(k1(2,1)+2*k2(2,1)+2*k3(2,1)+k4(2,1));
    sir(3,i)=sir(3,i-1)+(1/6)*(k1(3,1)+2*k2(3,1)+2*k3(3,1)+k4(3,1));
  end
end
T=zeros(1,dt);
i=0;
t=-1;
for e=randn(0,dt)
  i=i+1;
  t=t+1;
  T(1,i)=t;
end
x1=T';
y1=sir(_,:)'; % 2 para plotar I(t) , 3 para plotar R(t)
plot(x1,y1);
```