Universidade Federal do Espírito Santo

Trabalhos

Códigos

Algoritmos numéricos - Explicação dos códigos do trabalho

Italo Jezo de Oliveira Silva e Agenor Andre Almeida Dias

Vitória

2022

Problema 1) Primeiro trabalho computacional

- **PYTHON: savgol_filter ( ), da biblioteca scipy.signal (Trabalho1.py)**

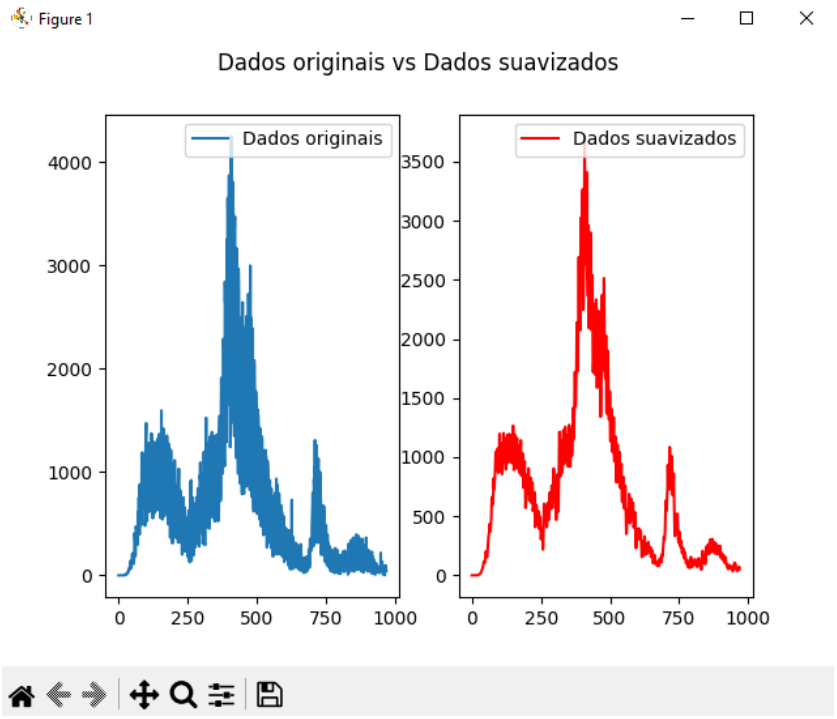
Conseguimos inserir no código uma forma automatizada de descobrir qual é o delimitador do arquivo .csv. Isso foi essencial para a evolução do código, já que em alguns casos, o arquivo .csv é limitado por vírgulas (“.”) ou tabs (“\t”). No esquema atual, o código está configurado para identificar qual o delimitador, para então, continuar os cálculos.

Esse código lê um arquivo CSV (CovidNumerico.csv) e usa as bibliotecas "pandas" e "scipy.signal" para processar os dados. Primeiro, é feita a leitura do arquivo usando a função "pd.read_csv", identificando automaticamente o delimitador. Em seguida, são verificados os valores nulos e interpolados. Em seguida, é criada uma lista com os nomes das colunas. O filtro de Savitsky-Golay é aplicado aos dados com base na largura, ordem polinomial e número de passadas especificadas.

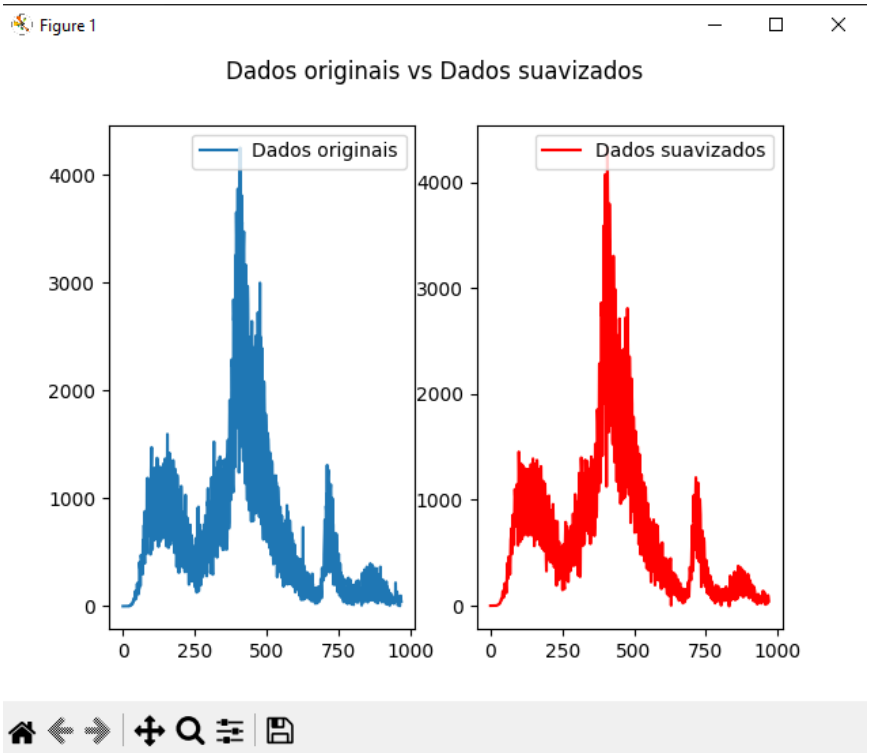
O usuário é convidado a escolher a coluna que deseja suavizar usando um menu interativo. As opções incluem "CasosNovos", "ObitosNovos", "Recuperadosnovos" e "EmAcompanhamentoNovos". A coluna selecionada é filtrada e os dados originais e filtrados são plotados usando "matplotlib".

A seguir, iremos aplicar combinações, na coluna de ObitosNovos do arquivo .csv, conforme solicitado:

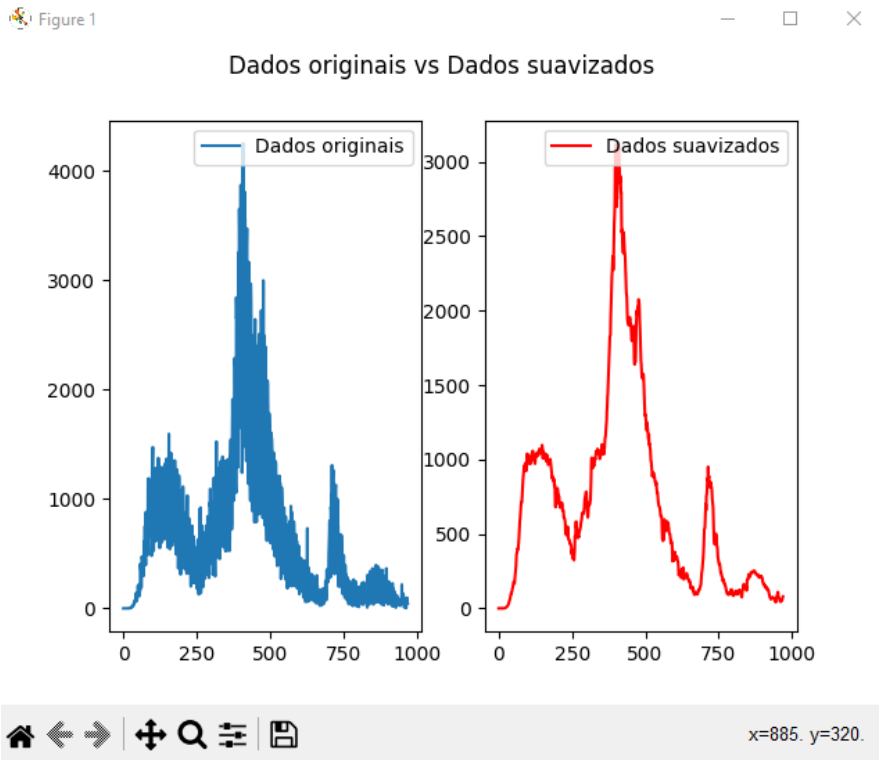
Teste 1:



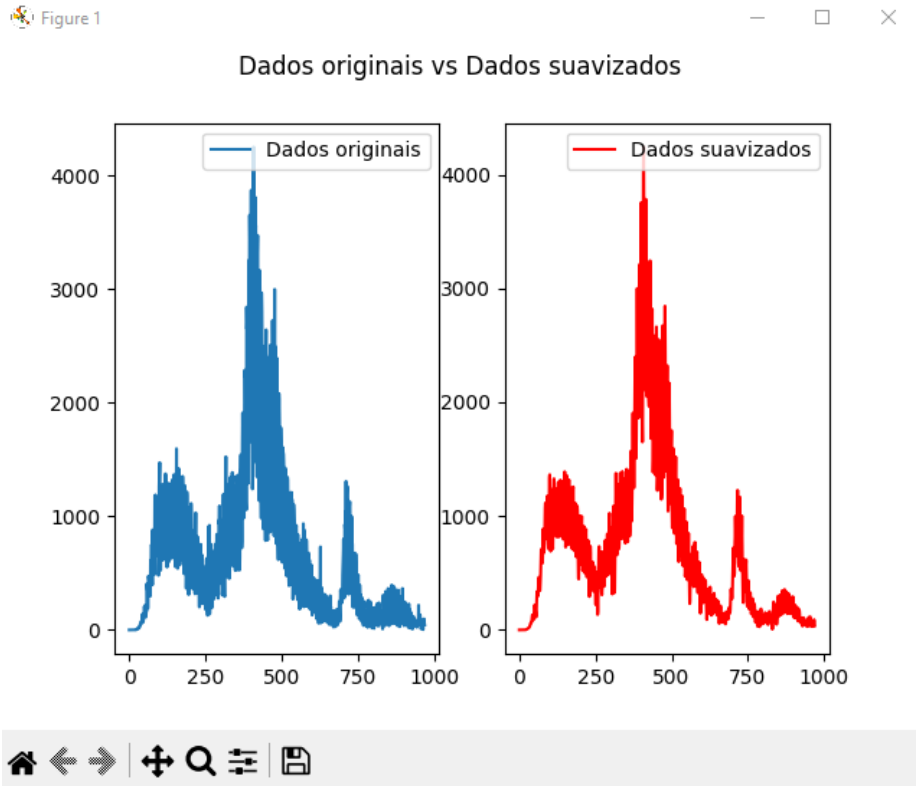
Teste 2:



Teste 3:

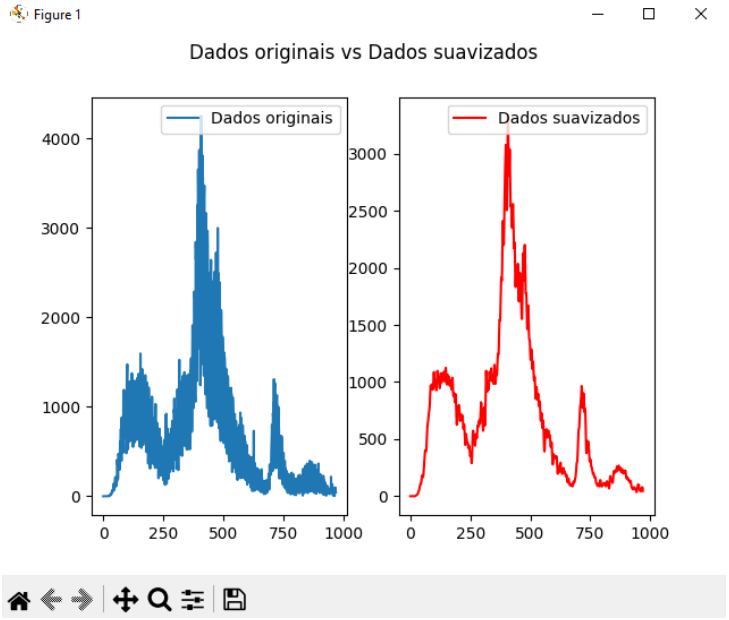


Teste 4:



Teste outras (escolha do grupo):

- **Largura: 10**
- **Grau: 3**
- **Número de passadas: 3**



- **OCTAVE: sgolayfilt (x,p,n) (trab1.m)**

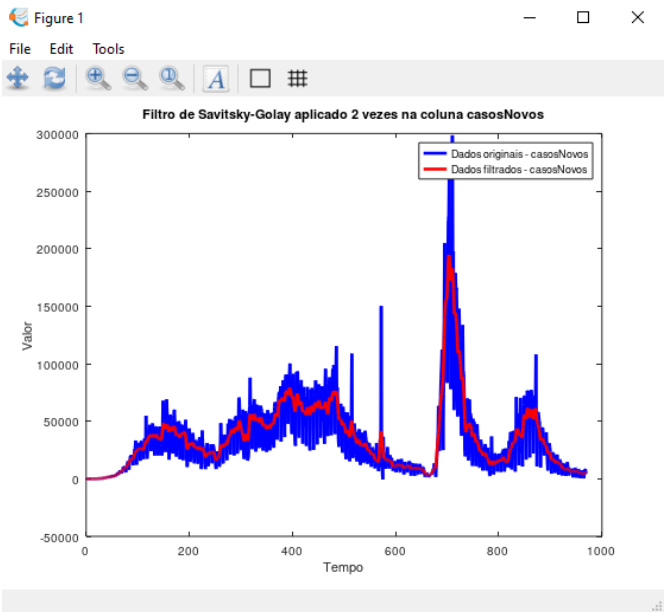
O primeiro passo do código foi listar os principais delimitadores de um arquivo .csv e identificar qual está sendo utilizado. Assim, não terá problemas caso o delimitador seja algum diferente da máquina criada para o desenvolvimento do código.

Este código em Octave é uma aplicação para filtrar dados em um arquivo CSV usando o filtro de Savitsky-Golay. O objetivo é identificar o delimitador correto do arquivo CSV (possivelmente separado por vírgulas, tabulações, ponto-e-vírgulas ou barras), ler as colunas e títulos, solicitar ao usuário qual coluna ele deseja filtrar e quantas vezes o filtro deve ser aplicado. O filtro de Savitsky-Golay é então aplicado aos dados selecionados e exibido em um gráfico comparando os dados originais e filtrados. Por fim, é perguntado ao usuário se ele deseja finalizar o programa.

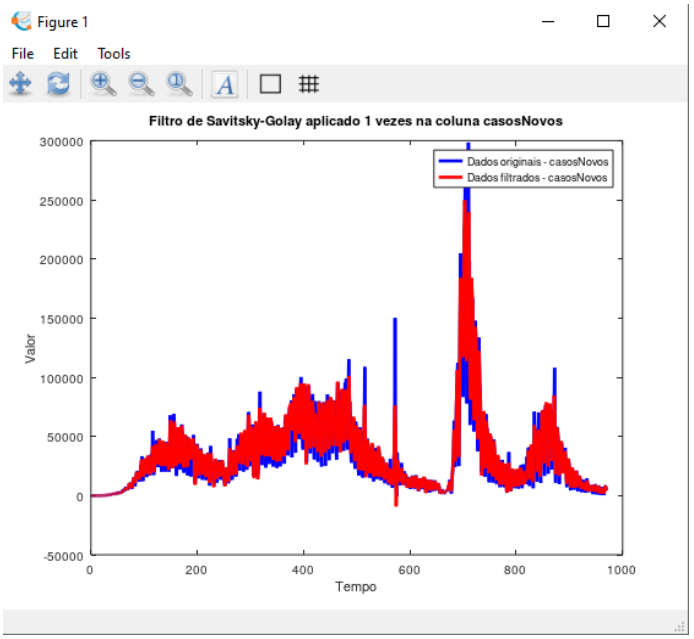
A seguir, iremos aplicar combinações, na coluna de CasosNovos do arquivo .csv, conforme solicitado:

Teste	LARGURA	GRAU	NÚMERO DE “PASSADAS”
1	5	1	2
2	5	3	1
3	7	1	2
4	7	3	1
outras	à escolha do grupo	à escolha do grupo	à escolha do grupo

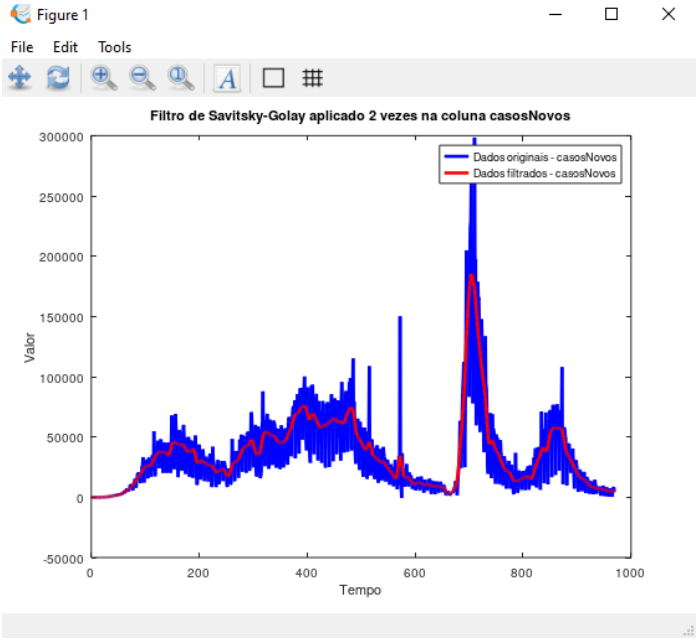
Teste 1:



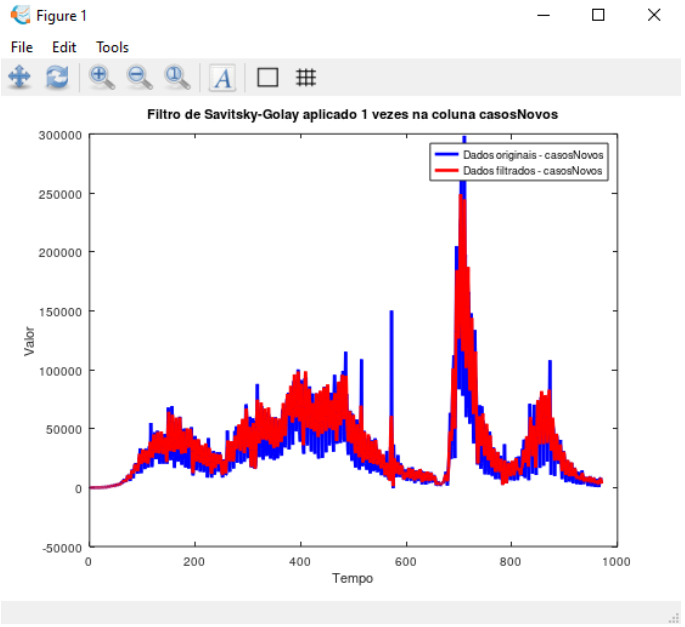
Teste 2:



Teste 3:

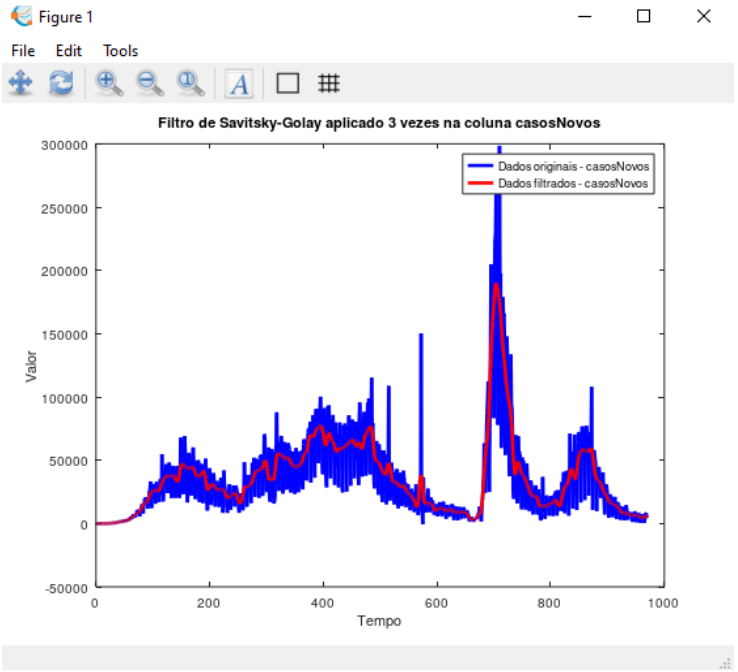


Teste 4:



Teste outras (escolha do grupo):

- **Largura: 10**
- **Grau: 3**
- **Número de passadas: 3**



Problema 2) Segundo trabalho computacional

Este é um programa em Python que oferece três opções para o usuário escolher: (1) Algoritmo de mínimos erros quadrados ordinário, (2) Algoritmo de mínimos erros quadrados recursivo e (3) Algoritmo de mínimos quadrados total. Quando o usuário seleciona uma opção, o programa executa a correspondência algoritmo de ajuste de curva e imprime o resultado dos coeficientes A, B e C.

A biblioteca Numpy é usada para criar arrays numéricos, enquanto a biblioteca Scipy.odr é usada para realizar o ajuste de curva por mínimos erros quadrados ordinários (ODR). A função "f" define uma função polinomial de segundo grau, que é usada como modelo de ajuste de curva. Os dados são armazenados em arrays x e y, e um objeto ODR é criado usando esses dados e o modelo. O resultado da execução do objeto ODR é armazenado na variável "result", que contém os coeficientes A, B e C.

- ### Resultados do código:

```
=======BEM-VINDO AO MENU DE ESCOLHAS!========

Selecione a opção desejada:

0- Sair
1- Algoritmo de mínimos erros quadrados ordinário.
2- Algoritmo de mínimos erros quadrados recursivo.
3- Algoritmo de mínimos quadrados total.

Escolha um número para definir com qual variação você irá calcular:
```

Menu de escolhas

```
Utilizando algoritmo de mínimos erros quadrados ordinário, temos:

A: 2.001126533646429
B: 2.8306741825401414
C: 2.616059806934307

Aperte qualquer tecla para continuar...
```

```
Utilizando algoritmo de mínimos erros quadrados recursivo, temos:

a: 36.10107599999999
b: 7.134599999999999
c: 1.41

Aperte qualquer tecla para continuar...
```

```
Utilizando algoritmo de mínimos erros quadrados total, temos:

A: 1.9931191549681553
B: 3.6508736713346983
C: -11.812383285548506

Aperte qualquer tecla para continuar...
```

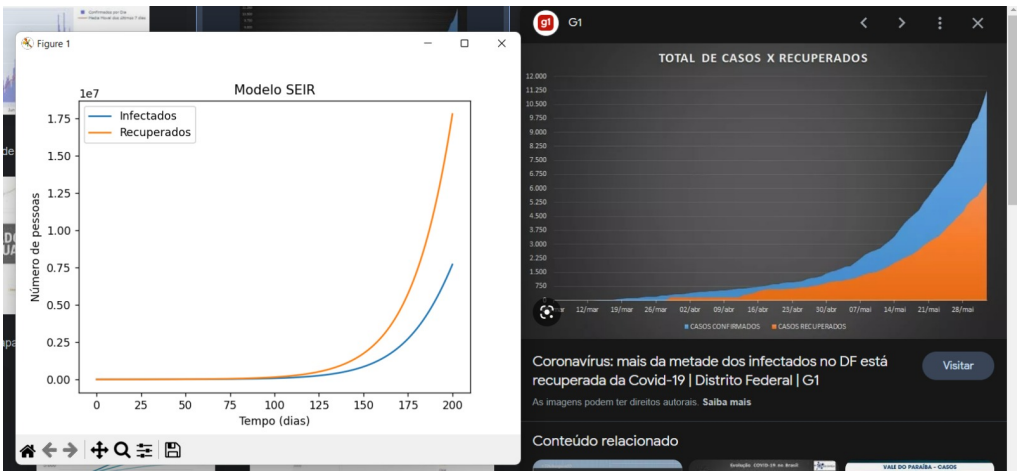
Problema 3) Terceiro trabalho computacional

Este código implementa um modelo matemático SEIR (susceptível, exposto, infectado, recuperado) para simular o crescimento de casos de COVID-19 em uma população. O usuário pode escolher entre três opções de exemplos, que possuem diferentes valores para as taxas de transmissão, recuperação, incubação e decadência da proteção vacinal. As equações diferenciais que descrevem o modelo são resolvidas com a função "odeint" da biblioteca scipy e os resultados são plotados com o módulo matplotlib. A cada vez que o usuário escolhe um exemplo, o gráfico é mostrado na tela e a tela é limpa.

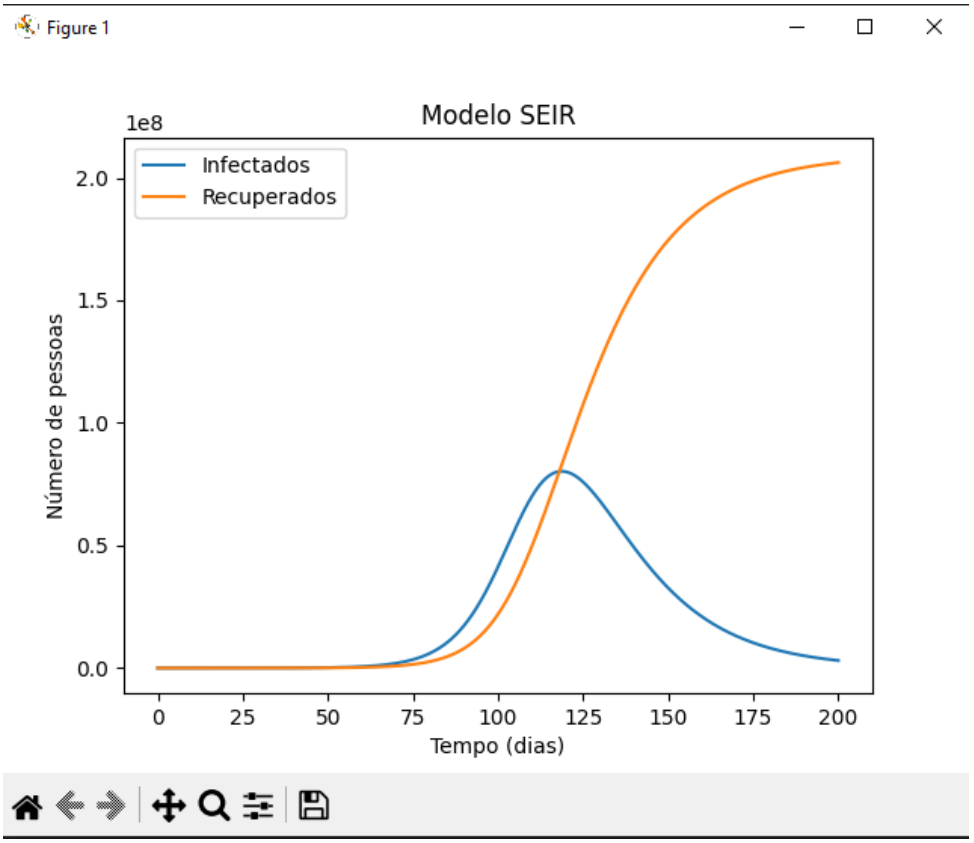
Ajustamos os parâmetros manualmente para infecção por Covid-19. Perceba que alteramos o eixo do tempo nos gráficos plotados.

```
=======BEM-VINDO AO MENU DE EXEMPLOS!========

Selecione a opção desejada:

0- Sair
1- Exemplo 1
2- Exemplo 2
3- Exemplo 3

Escolha um numero para definir o exemplo de gráfico de infecção por Covid-19 que você deseja visualizar:
```

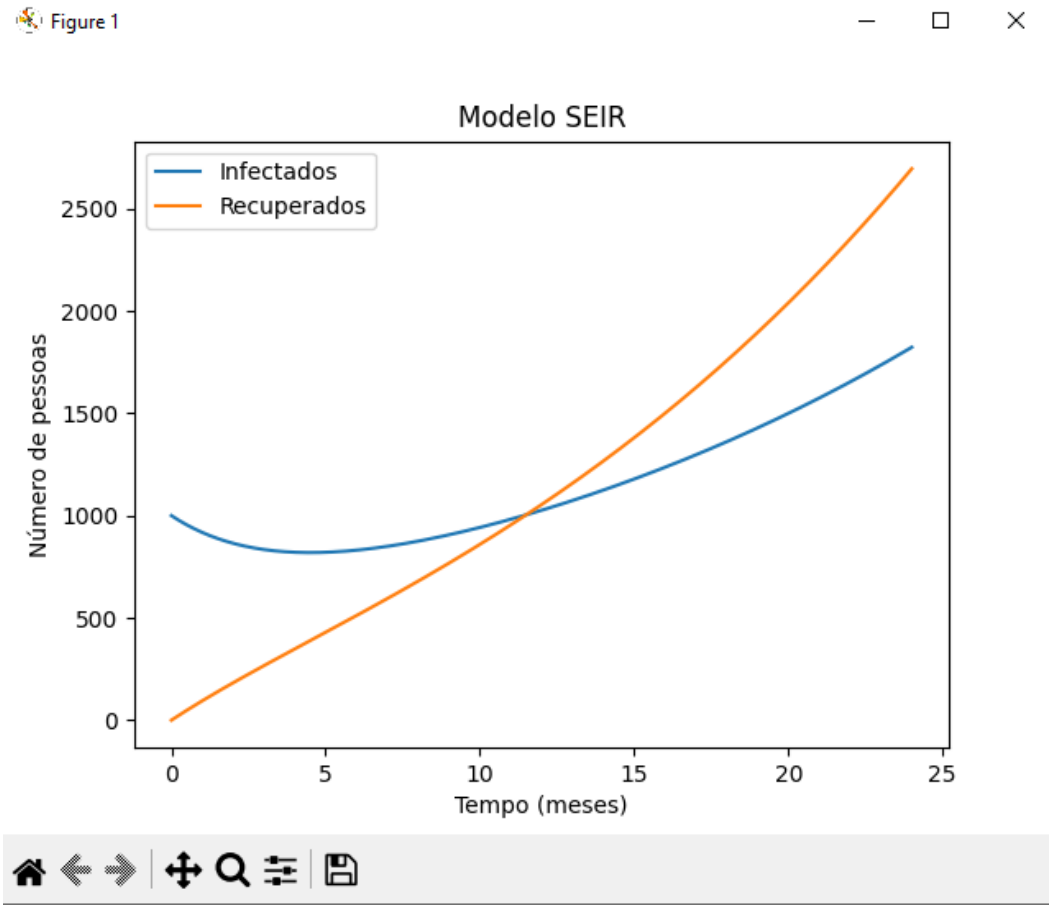
Menu de exemplos



Exemplo 01: conseguimos ajustar parâmetros para ficar próximo ao que é fornecido no mundo real.



Exemplo 02: Simulação que após a vacina (entre os dias 100 e 125) o número de recuperados começa a aumentar.



Exemplo 03: Simulação que após a vacina (entre os meses 10 e 15) o número de recuperados começa a aumentar.