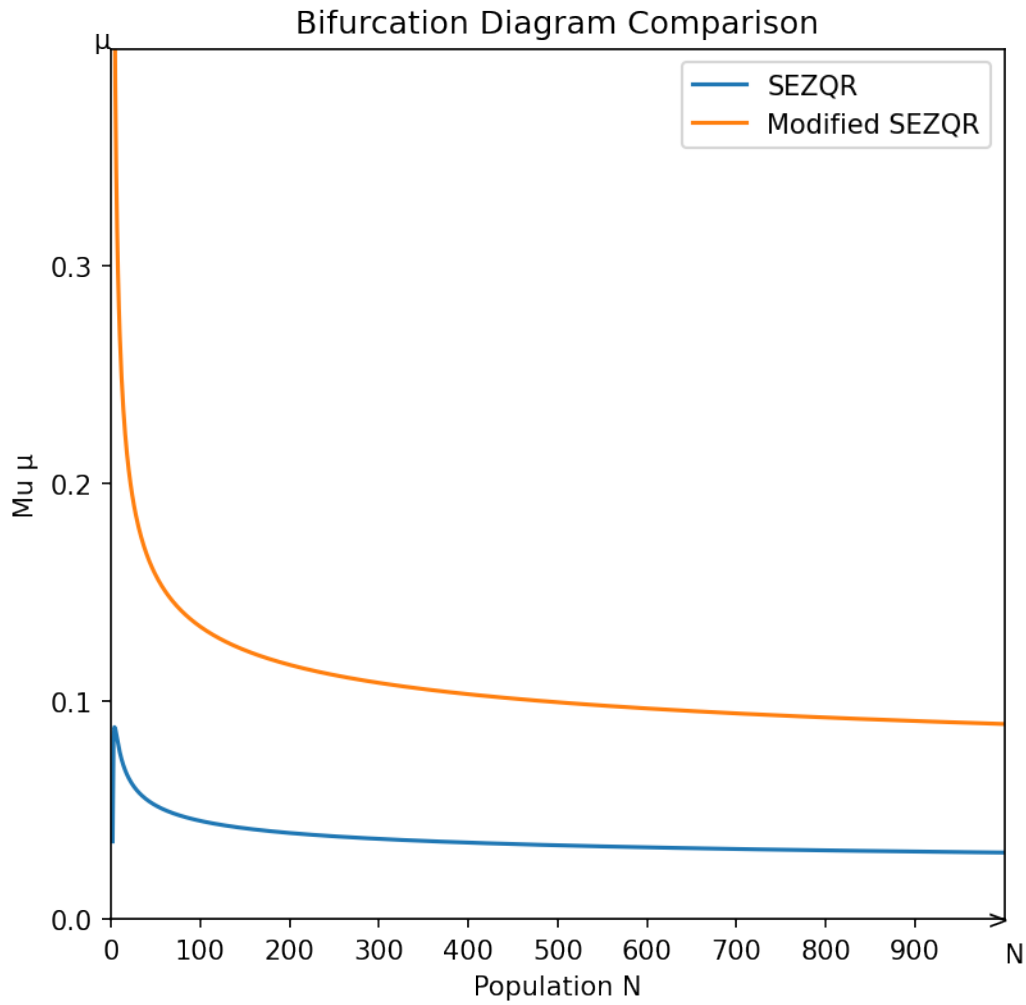
SEZQR_model

September 28, 2021

[17]:
```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""This program plots the bifurcation diagram for the SEZQR Model

    Reference: Allen, Robert F., Cassandra Jens, and Theodore J. Wendt. 2014.
   "Perturbations in Epidemiological Models". Letters in Biomathematics 1 (2),␣
 ↪173-80.
   Link- https://doi.org/10.1080/23737867.2014.11414478.

   Part of MS Thesis at Universität Koblenz-Landau
   Note: this program uses the following libraries-  Numpy, Matplotlib
   Python Version 3.7
"""
# Bifurcation Diagram For SEZQR Model

# SEZQR MODEL

# Import our modules that we are using
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.pyplot import figure
figure(figsize=(6, 6), dpi=150)


beta = 0.0095
alpha = 0.005
zeta = 0.0001
rho = 0.005
kappa = 0.001
sigma = 0.001
gamma = 0.0001
omega = 0.009
N = 1000


# Create the vectors X and Y
```

```
x = np.array(range(2,1000))
temp03 = - alpha*gamma*(rho + kappa)
temp04 = (rho+kappa)*((zeta*sigma)+gamma*(zeta+sigma))
y = (np.log( (beta*x*(zeta*kappa + gamma*(zeta - rho)) + temp04) / (temp03))/
 ↪(np.log(x))) -1



# Modoified SEZQR
temp01 = ((zeta-rho)*(gamma+omega)+kappa*(zeta-omega))
temp02 = (rho+kappa)*(sigma*x*(zeta - omega) + (gamma+omega)*(zeta + sigma*x))
z = ((np.log((beta*x*temp01 + temp02 )/(-alpha*(gamma+omega)*(rho+kappa))))/(np.
 ↪log(x))) - 1


# Create the plot
plt.plot(x,y, label='SEZQR')
plt.plot(x,z, label='Modified SEZQR')
plt.ylim([0, 0.4])
plt.xlim([0, 1000])
plt.yticks(np.arange(0, 0.4, 0.1))
plt.xticks(np.arange(0, 1000, 100))
plt.xlabel("Population N")
plt.ylabel("Mu " + u"\u03bc")
plt.legend(loc='best')

plt.text(1000, -0.02, "N")
plt.text(980, -0.004, ">")
#plt.text(0, 0.4, ">")
plt.text(-20, 0.4, u"\u03bc")

#plt.text(600, 0.05, "Stable", fontsize= 14, color= 'green')
#plt.text(600, 0.005, "Unstable", fontsize= 14, color= 'red')


# Show the plot
plt.title("Bifurcation Diagram Comparison")
plt.show()


__author__ = "Md Tariqul Islam"
__version__ = "1.0"
__maintainer__ = "Tariqul"
__email__ = "tariquldipu@uni-koblenz.de"
__status__ = "Final"
```

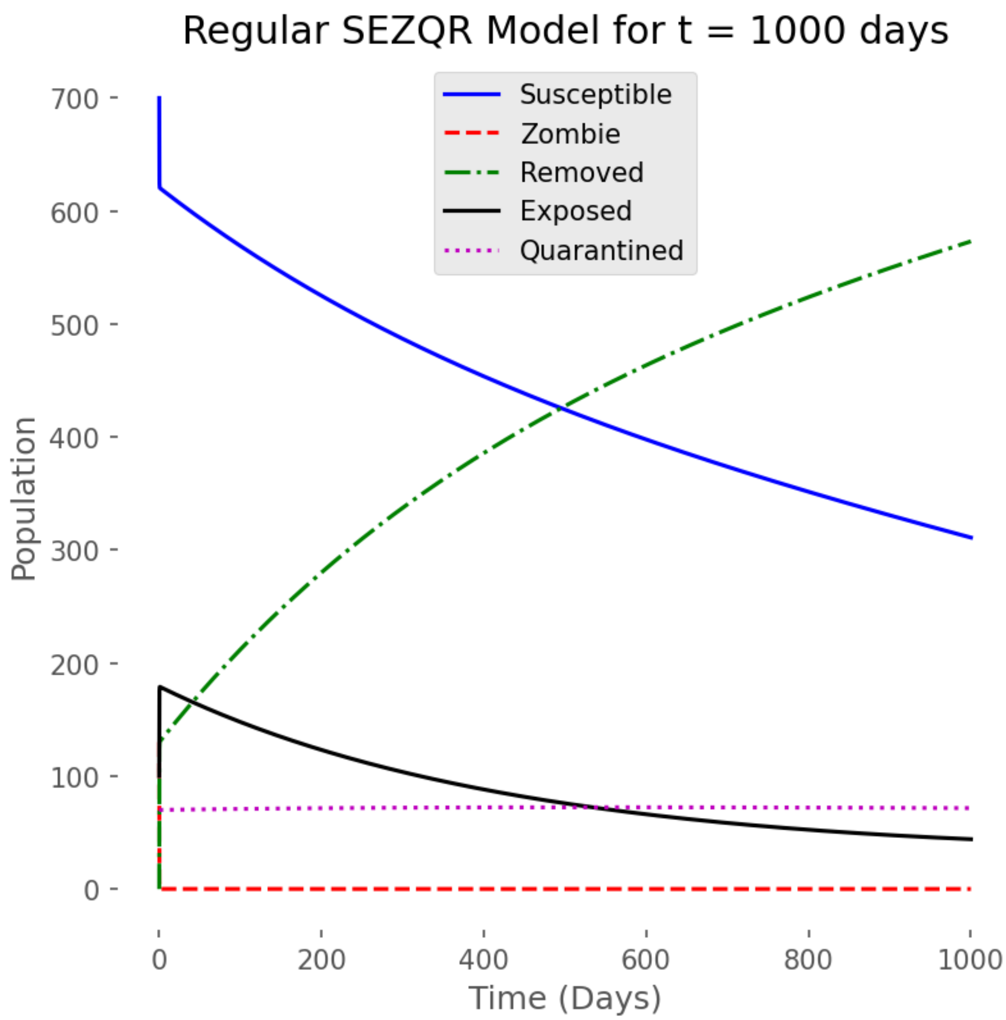


[ ]:

[23]:
```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

"""This simple program tries to solve the ODE of the SEZQR model with␣
↪perturbation and create a numerical solution.
   After that, this program draws the solution graph for the Susceptible,␣
↪Exposed, Zombie, Quarantined and Removed
   Population. This simple graph helps us to understand the actual scenario of␣
↪the population
   from the SEZQR model after a certain number of days. We can change the␣
↪value of the Perturbation
   Parameter Mu to investigate the change of behaviour and stabality of the␣
↪SZR model.
```

```
    Reference: Allen, Robert F., Cassandra Jens, and Theodore J. Wendt. 2014.
   "Perturbations in Epidemiological Models". Letters in Biomathematics 1 (2),␣
 ↪173-80.
   Link- https://doi.org/10.1080/23737867.2014.11414478.

   Part of MS Thesis at Universität Koblenz-Landau
   Note: this program uses the following libraries-  Numpy, Matplotlib, Scipy
   Python Version 3.7
"""

# SEZQR MODEL

# importing libraries
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
figure(figsize=(6, 6), dpi=150)

# I'm using this style for a pretier plot, but it's not actually necessary
plt.style.use('ggplot')

# Parameter values
beta = 0.0095
alpha = 0.005
zeta = 0.0001
rho = 0.005
kappa = 0.001
sigma = 0.001
gamma = 0.0001
kappa = 0.0001
omega = 0.009
N = 1000  # initial population

# Perturbation Parameter Mu, change this value to see different results
mu = 0.175

# Defining Function of the SEZQR ODE
def model(t, x):
    S = x[0] # Intital Susceptible Population
    E = x[1] # exposed
    Z = x[2] # zombies
    Q = x[3] # quarentined
    dsdt = - beta*S*Z # Susceptible
    dedt = beta*S*Z - (rho + kappa)*E # Exposed
```

```
    dzdt = rho*E + zeta*(N - S - E - Z - Q) - sigma*Z - alpha*(S**(1 + mu))*Z #
↪Zombies
    dqdt = kappa*E + sigma*Z - gamma*Q # Quarantined
    dxdt = [dsdt, dedt, dzdt, dqdt]
    return dxdt

# Initial Condition. Susceptible= 700, Exposed= 100, Zombie= 130, Quarantined =
↪70
x0 = [700, 100, 130, 70]

# Time, as in number of days. Total days= 1000
t = np.array([0, 1000])
tspan = np.linspace(t[0], t[1], 10000)

# Calculating numerical solution of the given ODEs
x = solve_ivp(model, t, x0, t_eval=tspan)

time = x.t
susceptible = x.y[0];
exposed = x.y[1];
zombie = x.y[2]
quarantined  = x.y[3]
removed = (N - susceptible - exposed - zombie - quarantined)

# Plot
plt.plot(time, susceptible, 'b-', label="Susceptible") # Susceptible Population
↪Graph
plt.plot(time, zombie, 'r--', label="Zombie") # Zombie Population Graph
plt.plot(time, removed, 'g-.', label="Removed") # Recovered Population Graph
plt.plot(time, exposed, 'k', label="Exposed") # Exposed Population graph
plt.plot(time, quarantined , 'm:', label="Quarantined")
plt.title('Regular SEZQR Model for t = 1000 days')
plt.ylabel('Population')
plt.xlabel('Time (Days)')
plt.legend(loc='best')
ax = plt.gca()
ax.set_facecolor('w')
plt.show()

__author__ = "Md Tariqul Islam"
__version__ = "1.0"
__maintainer__ = "Tariqul"
__email__ = "tariquldipu@uni-koblenz.de"
__status__ = "Final"
```


Regular SEZQR Model for t = 1000 days
Susceptible
Zombie
Removed
Exposed
Quarantined
Population
Time (Days)
0
100
200
300
400
500
600
700
0
200
400
600
800
1000

```
[ ]:
```