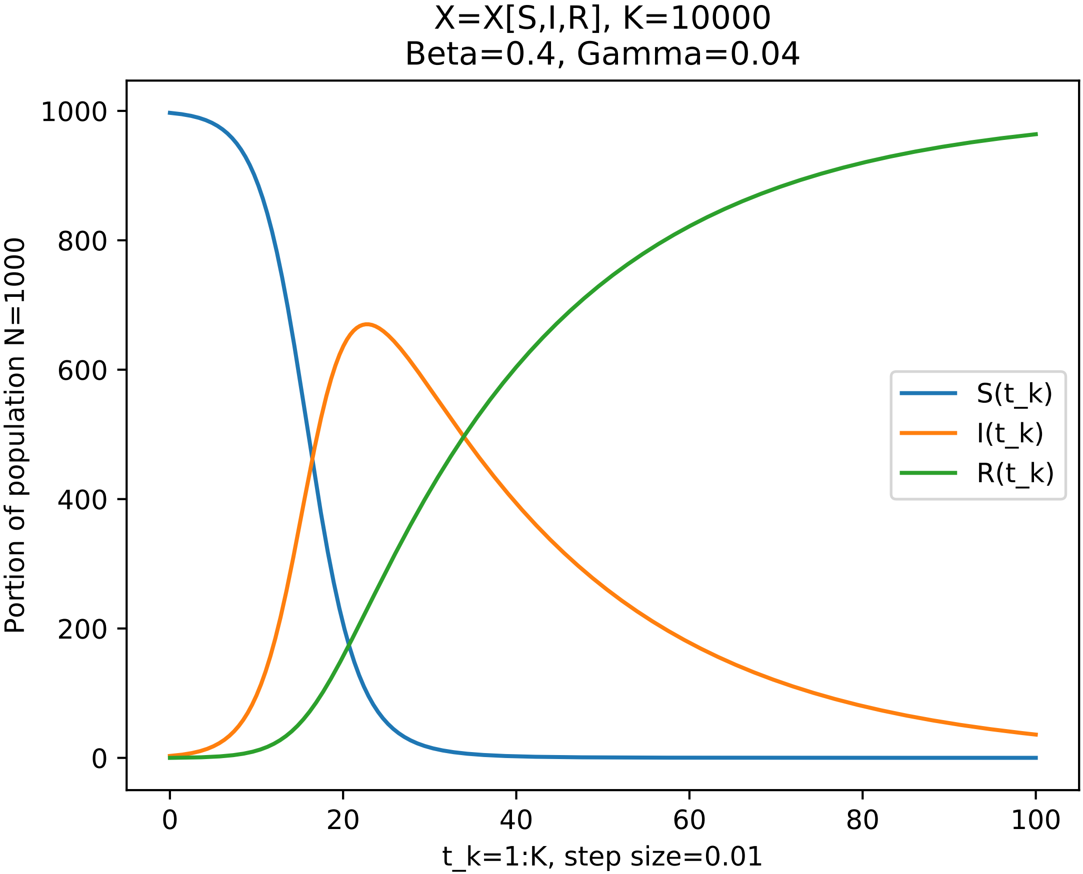
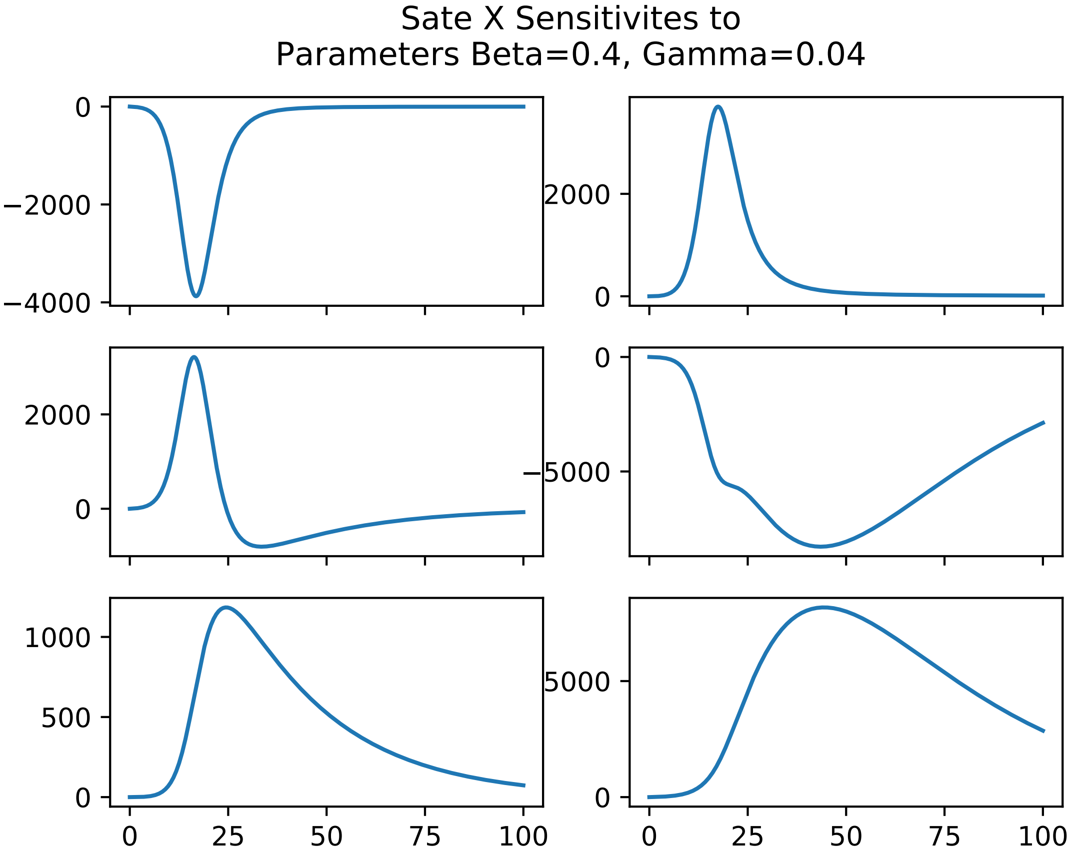
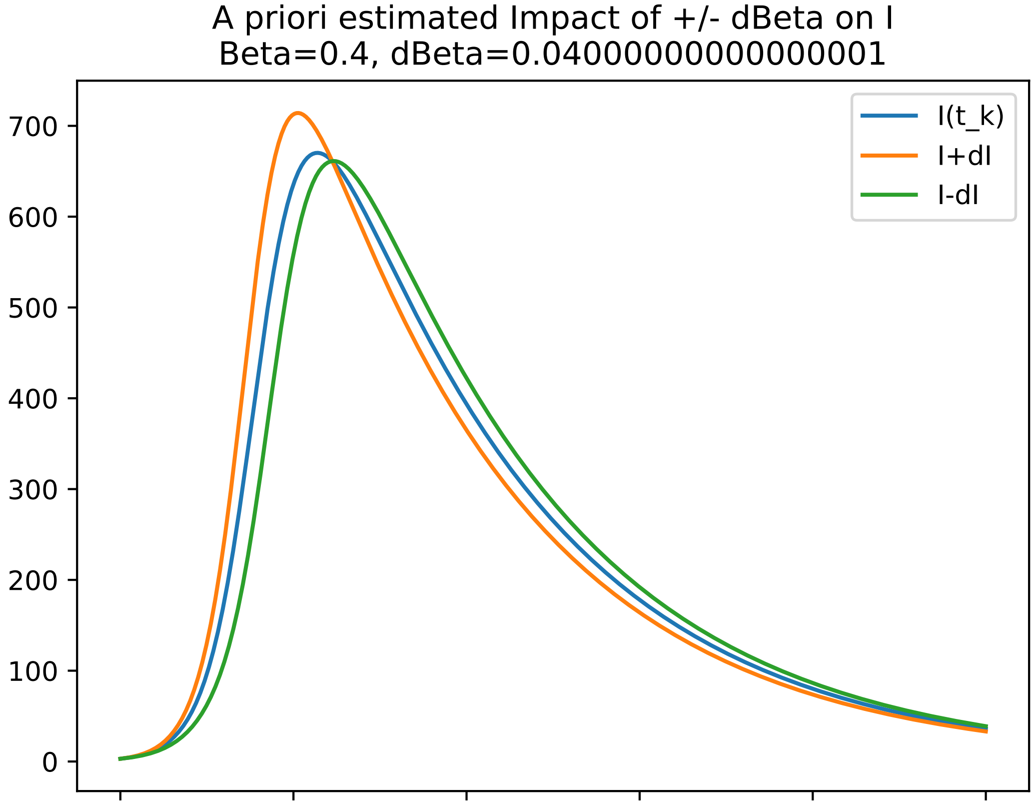
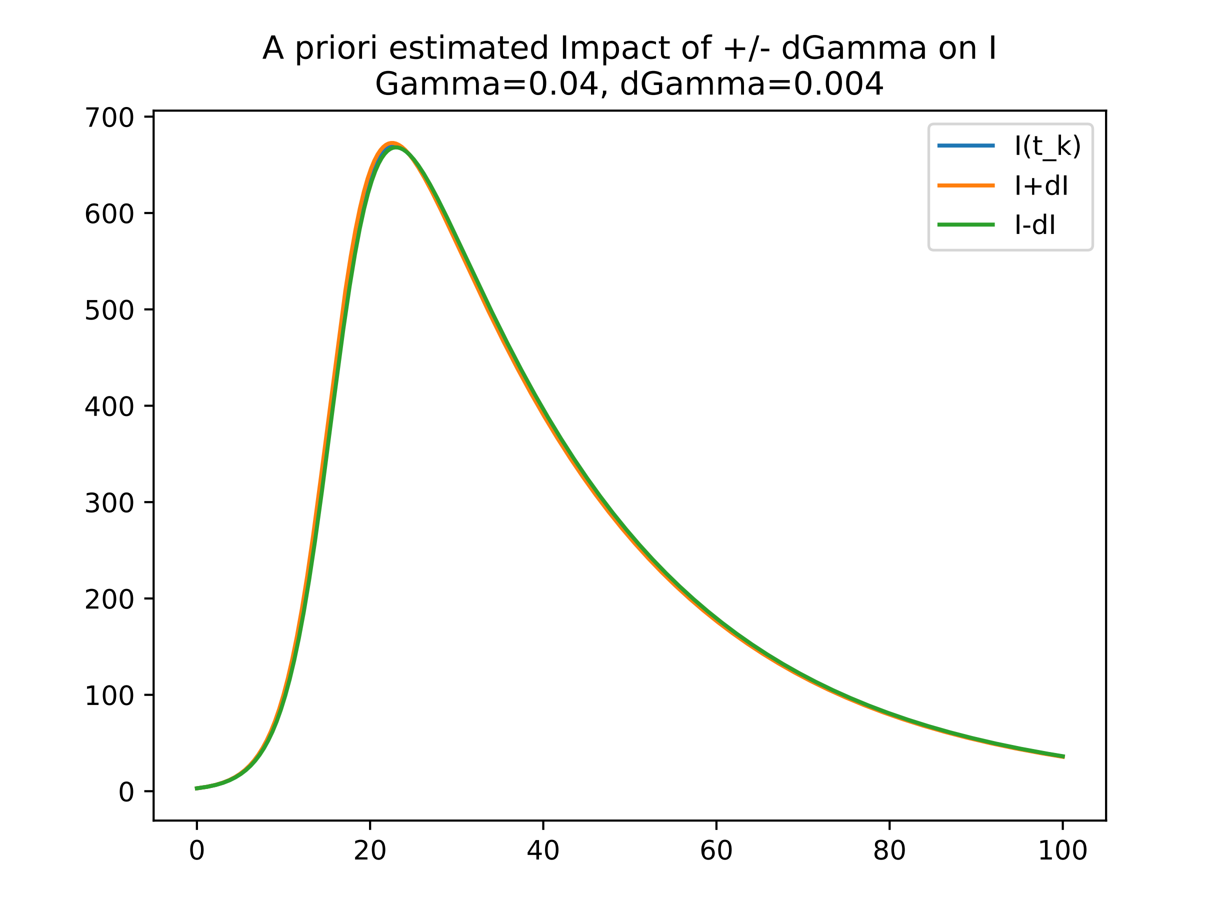

X=X[S,I,R], K=10000
Beta=0.4, Gamma=0.04
Portion of population N=1000
1000
800
600
400
200
0
0
20
40
60
80
100
t_k=1:K, step size=0.01
S(t_k)
I(t_k)
R(t_k)



All graphs share the same horizontal axis as in Fig 1.
Legend: [[dS/dBeta, dS/dGamma], [dI/dBeta, dI/dGamma], [dR/dBeta, dR/dGamma]]


A priori estimated Impact of +/- dBeta on I
Beta=0.4, dBeta=0.04000000000000001
I(t_k)
I+dI
I-dI
0
100
200
300
400
500
600
700
0
20
40
60
80
100


A priori estimated Impact of +/- dGamma on I
Gamma=0.04, dGamma=0.004
I(t_k)
I+dI
I-dI
0
100
200
300
400
500
600
700
0
20
40
60
80
100

Beta	0.4
Gamma	0.04
+/- dBeta	0.04
+/- dGamma	0.004
J	240.99
+/- dJ, from dBeta	0.7361
+/- dJ, from dGamma	0.07361

I don't exactly understand how I'm supposed to use which numbers for the last parts of Part B.

J is calculated using the quadrature model provided with I_k from Part A.

Both dJ's are calulated using the same quadrature model, just for the integral from of dJ with [dI/dBeta]_k also from Part A.

Branch: master ▾ Find file Copy path

610_Optimization / HW2 / hw2.py / <> Jump to ▾

JimVargas5 finished part b of hw2

bd3ca39 17 minutes ago

1 contributor

Raw Blame History

141 lines (113 sloc) 3.56 KB

```
# Jim Vargas
# MTH 610
# HW 2


import numpy as np
from tabulate import tabulate
from matplotlib import pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages
pdf=PdfPages('JimVargas_610_hw2_graphs.pdf')


Beta=0.4
Gamma=0.04
N=1000

X_0=np.array( [997, 3, 0] )
X=np.array(X_0) # [S I R]^T
X_storage=np.zeros((10001,3))
X_storage[0]=np.array([X])

dXdu_0=np.zeros((3,2))
dXdu=np.array(dXdu_0)
dXdu_storage=np.zeros((10001,3,2))
dXdu_storage[0]=np.array([dXdu])

K=10000
h=0.01
t=np.linspace(0, 100, K+1)


# model equations
def dXdt(S, I, R):  # 3x1
```

```python
    dSdt=-Beta*S*I/N
    dIdt=Beta*S*I/N - Gamma*I
    dRdt=Gamma*I
    return np.array([ dSdt, dIdt, dRdt ])
def dfdu(S, I, R):  # 3x2
    return np.array([
        [-S*I/N, 0],
        [S*I/N, -I],
        [0, I]
    ])
def dfdX(S, I, R):  # 3x3
    return np.array([
        [-Beta*I/N, -Beta*S/N, 0],
        [Beta*I/N, Beta*S/N - Gamma, 0],
        [0, Gamma, 0]
    ])


# main loop Part A ##############################################################
for k in range(1, K+1):
    X_next=X + h*dXdt(X[0], X[1], X[2])
    X_storage[k]=np.array([X_next])

    dXdu=dXdu +h*(
        np.matmul(dfdX(X[0], X[1], X[2]), dXdu)
        + dfdu(X[0], X[1], X[2])
    )
    dXdu_storage[k]=np.array([dXdu])

    X=X_next


# plots
fig_state=plt.figure()
plt.plot(t, X_storage[:,0], label='S(t_k)')
plt.plot(t, X_storage[:,1], label='I(t_k)')
plt.plot(t, X_storage[:,2], label='R(t_k)')
plt.title("X=X[S,I,R], K="+str(K)+'\n'+"Beta="+str(Beta)+", Gamma="+str(Gamma))
plt.xlabel("t_k=1:K, step size="+str(h))
plt.ylabel("Portion of population N="+str(N))
plt.legend(loc='best')
plt.close()
pdf.savefig(fig_state)

fig_params, ax=plt.subplots(3,2, sharex=True)
fig_params.suptitle(
    "Sate X Sensitivites to"+'\n'
    "Parameters Beta="+str(Beta)+", Gamma="+str(Gamma)
```

```python
)
fig_params.text(0.07,0.01,
    "All graphs share the same horizontal axis as in Fig 1.\n"+
    "Legend: [[dS/dBeta, dS/dGamma], [dI/dBeta, dI/dGamma], [dR/dBeta, dR/dGamma]]"
)
ax[0, 0].plot(t, dXdu_storage[:, 0, 0]) # dS/dBeta
ax[0, 1].plot(t, dXdu_storage[:, 0, 1]) # dS/dGamma
ax[1, 0].plot(t, dXdu_storage[:, 1, 0]) # dI/dBeta
ax[1, 1].plot(t, dXdu_storage[:, 1, 1]) # ...
ax[2, 0].plot(t, dXdu_storage[:, 2, 0])
ax[2, 1].plot(t, dXdu_storage[:, 2, 1])
plt.close()
pdf.savefig(fig_params)



# Part B ###########################################################################
dBeta=0.1*Beta
dGamma=0.1*Gamma

dI_Beta=dBeta*dXdu_storage[:,1,0]
dI_Gamma=dGamma*dXdu_storage[:,1,0]

J=(h/100)*np.sum(X_storage[:,1])

tempSum=(h/100)*np.sum(dXdu_storage[:,1,0])
dJ_Beta=dBeta*tempSum
dJ_Gamma=dGamma*tempSum


# plots and table
fig_apriori_dI_Beta=plt.figure()
plt.title("A priori estimated Impact of +/- dBeta on I\n"+
    "Beta="+str(Beta)+", dBeta="+str(dBeta))
plt.plot(t, X_storage[:,1], label="I(t_k)")
plt.plot(t, X_storage[:,1] + dI_Beta[:], label="I+dI")
plt.plot(t, X_storage[:,1] - dI_Beta[:], label="I-dI")
plt.legend(loc='best')
plt.close()
pdf.savefig(fig_apriori_dI_Beta)

fig_apriori_dI_Gamma=plt.figure()
plt.title("A priori estimated Impact of +/- dGamma on I\n"+
    "Gamma="+str(Gamma)+", dGamma="+str(dGamma))
plt.plot(t, X_storage[:,1], label="I(t_k)")
plt.plot(t, X_storage[:,1] + dI_Gamma[:], label="I+dI")
plt.plot(t, X_storage[:,1] - dI_Gamma[:], label="I-dI")
plt.legend(loc='best')
```

```
plt.close()
pdf.savefig(fig_apriori_dI_Gamma)


pdf.close()

Table=[
    ["Beta",Beta],["Gamma",Gamma],["+/- dBeta",dBeta],["+/- dGamma",dGamma],
    ["J",J],["+/- dJ, from dBeta",dJ_Beta],["+/- dJ, from dGamma",dJ_Gamma]
]
with open("JimVargas_610_hw2.txt", 'w') as output:
    print(tabulate(Table), file=output)
```