hw5_code

September 29, 2021

Michael Goforth
CAAM 550
HW 5
9/29/21

Problem 1

Part ii

[16]:
```
import math
# I0
I = math.exp(1) - 1
print('  0 %13.6e ' % I)

for i in range(1,21):
    I = math.exp(1) - i*I
    print(' %2d %13.6e ' % (i, I))
```

```
 0  1.718282e+00
 1  1.000000e+00
 2  7.182818e-01
 3  5.634363e-01
 4  4.645365e-01
 5  3.955995e-01
 6  3.446845e-01
 7  3.054900e-01
 8  2.743615e-01
 9  2.490280e-01
10  2.280015e-01
11  2.102652e-01
12  1.950999e-01
13  1.819831e-01
14  1.705191e-01
15  1.604959e-01
16  1.503482e-01
17  1.623631e-01
18 -2.042536e-01
19  6.599099e+00
20 -1.292637e+02
```

Problem 3

[49]:
```
import numpy as np
import pandas as pd
nvec = [2*i for i in range(1, 21)]
columnnames = ["n", "||x-xcomp||2", "||x-xcomp||2/||x||2", "    kappa2(A)"]
data = pd.DataFrame(columns = columnnames)
for n in nvec:
    # Construct A matrix
    A = np.zeros((n, n))
    for i in range(0, n):
        ti = -1 + 2 * (i) / (n - 1)
        for j in range(0, n):
            A[i, j] = ti**j
    xex = np.ones((n, 1))
    b = A @ xex
    xcomp = np.linalg.solve(A, b)
    comp1 = np.linalg.norm(xex-xcomp, 2)
    comp2 = comp1/np.linalg.norm(xex, 2)
    kappa2 = np.linalg.norm(A, 2) * np.linalg.norm(np.linalg.inv(A), 2)
    data = data.append({'n': n, "||x-xcomp||2" : comp1, "||x-xcomp||2/||x||2" :␣
 ↪comp2, "    kappa2(A)" : kappa2}, ignore_index=True)
data['n'] = data['n'].apply('{:.0f}'.format)
print(data.to_string(index=False))
```

```
 n  ||x-xcomp||2  ||x-xcomp||2/||x||2      kappa2(A)
 2    0.0000e+00           0.0000e+00     1.0000e+00
 4    3.1402e-16           1.5701e-16     8.0116e+00
 6    6.2892e-15           2.5676e-15     6.3827e+01
 8    3.9111e-15           1.3828e-15     5.3535e+02
10    3.8002e-13           1.2017e-13     4.6264e+03
12    2.1609e-12           6.2379e-13     4.0755e+04
14    1.9775e-11           5.2850e-12     3.6383e+05
16    1.1501e-10           2.8753e-11     3.2800e+06
18    4.1977e-09           9.8940e-10     2.9794e+07
20    2.6001e-08           5.8140e-09     2.7224e+08
22    1.9828e-07           4.2273e-08     2.4997e+09
24    2.2190e-06           4.5296e-07     2.3043e+10
26    1.4532e-05           2.8499e-06     2.1314e+11
28    4.0202e-05           7.5975e-06     1.9772e+12
30    6.3281e-04           1.1553e-04     1.8388e+13
32    1.9835e-03           3.5063e-04     1.7151e+14
34    2.0962e-01           3.5950e-02     1.6057e+15
36    2.0214e+00           3.3690e-01     1.6189e+16
38    3.3398e+01           5.4178e+00     1.2159e+17
40    4.5722e+02           7.2293e+01     1.9694e+18
```

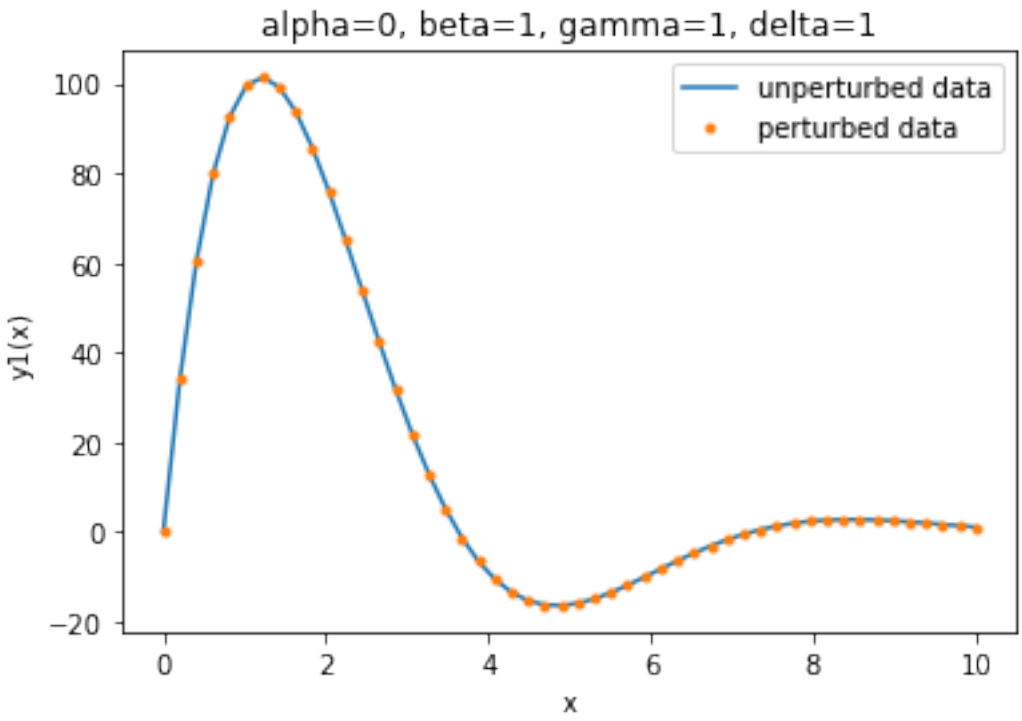
Problem 4

Part i.

[29]:
```
from scipy.linalg import expm
import numpy as np
import matplotlib.pyplot as plt

np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

alpha = 0
beta = 1
gamma = 1
delta = 1

E = expm(10 * np.array([[0, 1], [-delta, -gamma]]))
e1t = np.array([[1, 0]])
s = (beta - e1t @ E[:, 0] * alpha) / (e1t @ E[:, 1])
print('s = ' + str(s[0]))
x = np.linspace(0, 10)
y = np.zeros(x.size)
for i in range(x.size):
    y[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s]]))[0]
plt.plot(x, y, label='unperturbed data')
plt.xlabel('x')
plt.ylabel('y1(x)')
plt.title('alpha=0, beta=1, gamma=1, delta=1')
s2 = s + 1e-8
y2 = np.zeros(x.size)
for i in range(x.size):
    y2[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s + 1e-8]]))[0]
plt.plot(x, y2, '.', label='perturbed data')
plt.legend()
```

```
s = 185.68444885271484
```

[29]:
```
<matplotlib.legend.Legend at 0x1d3f1190280>
```



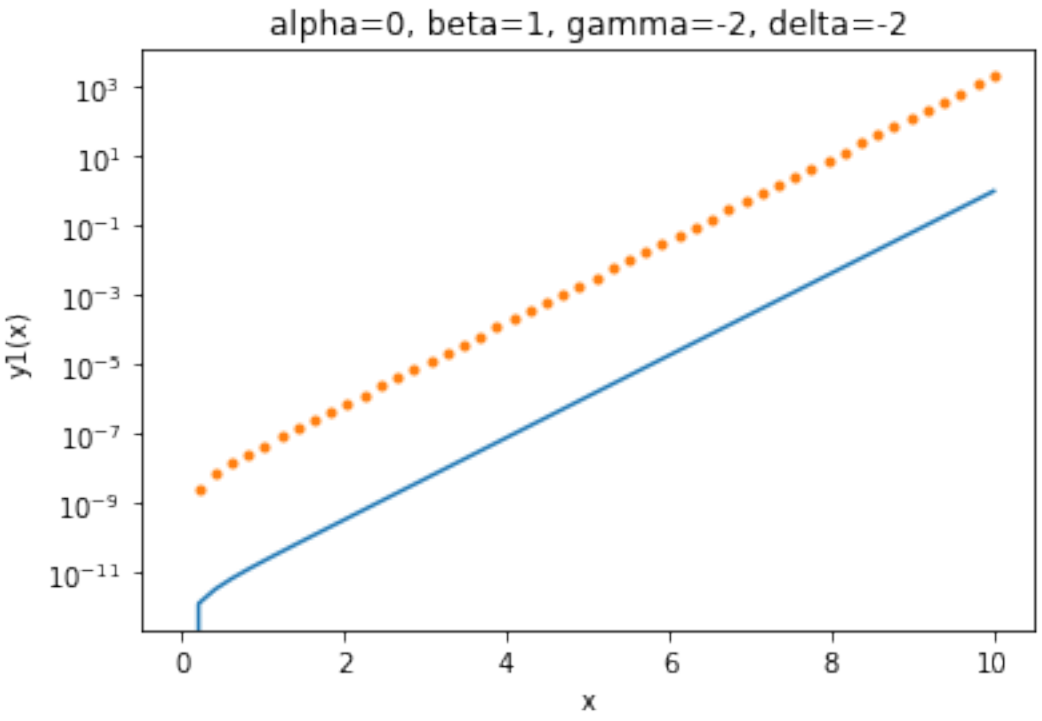
part ii.

[28]:
```
from scipy.linalg import expm
import numpy as np
import matplotlib.pyplot as plt

np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

alpha = 0
beta = 1
gamma = -2
delta = -2

E = expm(10 * np.array([[0, 1], [-delta, -gamma]]))
e1t = np.array([[1, 0]])
s = (beta - e1t @ E[:, 0] * alpha) / (e1t @ E[:, 1])
print('s = ' + str(s[0]))
x = np.linspace(0, 10)
y = np.zeros(x.size)
for i in range(x.size):
    y[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s]]))[0]
plt.semilogy(x, y, label='unperturbed data')
plt.xlabel('x')
```

```
plt.ylabel('y1(x)')
plt.title('alpha=0, beta=1, gamma=-2, delta=-2')
s2 = s + 1e-8
s2 = s + 1e-8
y2 = np.zeros(x.size)
for i in range(x.size):
    y2[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s + 1e-8]]))[0]
plt.semilogy(x, y2, '.', label='perturbed data')
```

```
s = 4.725466872852787e-12
```

[28]:
```
[<matplotlib.lines.Line2D at 0x1d3f14c7cd0>]
```



part iv

[23]:
```
import math
import numpy as np
from scipy.linalg import expm
alpha = 0
beta = 1
gamma = 1
delta = 1
E = expm(10 * np.array([[0, 1], [-delta, -gamma]]))
```

```
e1t = np.array([[1, 0]])
s = (beta - e1t @ E[:, 0] * alpha) / (e1t @ E[:, 1])
mu = -gamma/2
theta = (4 * delta - gamma**2)**.5 / 2
x = 10
fprime1 = math.exp(mu * x)* math.sin(theta * x) / theta
print('part i: dy(10)/ds = ' + str(fprime1))
gamma = -2
delta = -2
E = expm(10 * np.array([[0, 1], [-delta, -gamma]]))
e1t = np.array([[1, 0]])
s = (beta - e1t @ E[:, 0] * alpha) / (e1t @ E[:, 1])
lambda1 = -gamma / 2 + ((gamma**2 - 4 * delta) / 4)**.5
lambda2 = -gamma / 2 - ((gamma**2 - 4 * delta) / 4)**.5
x = 10
fprime2 = (-math.exp(lambda1 * x) + math.exp(lambda2 * x)) / (lambda2 - lambda1)
print('part ii: dy(10)/ds = ' + str(fprime2))
```

```
part i: dy(10)/ds = 0.005385480616059573
part ii: dy(10)/ds = 211619301733.9473
```

Problem 5 part i

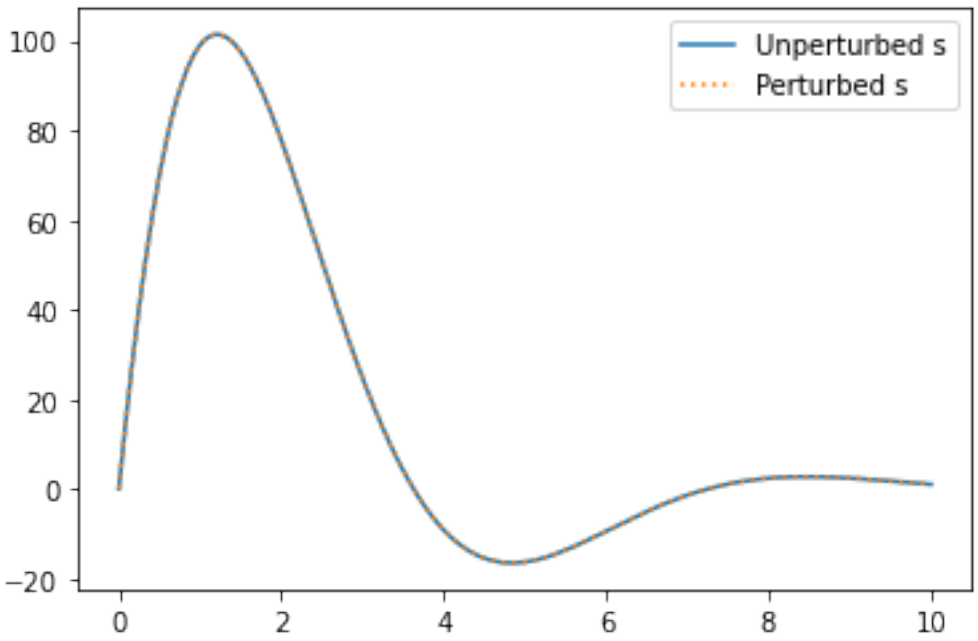
[27]:
```
import math
import numpy as np
from scipy.linalg import expm
import matplotlib.pyplot as plt

np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

alpha = 0
beta = 1
gamma = 1
delta = 1
xjvec = [i for i in range(0, 11)]
# See written pdf for derivation of A, b
E = expm(np.array([[0, 1], [-delta, -gamma]]))
e11 = E[0, 0]
e12 = E[0, 1]
e21 = E[1, 0]
e22 = E[1, 1]
b = np.zeros((19, 1))
b[0] = -e11 * alpha
b[1] = -e21 * alpha
b[18] = beta
A = np.zeros((19, 19))
# From eqn 18a
A[0, 0] = e12
```

```
A[0, 1] = -1
A[1, 0] = e22
A[1, 2] = -1
# From eqn 18b
for n in range(1, 9):
    A[2*n, 2*n-1] = e11
    A[2*n, 2*n] = e12
    A[2*n, 2*n+1] = -1
    A[2*n+1, 2*n-1] = e21
    A[2*n+1, 2*n] = e22
    A[2*n+1, 2*n+2] = -1
# from eqn 18c
A[18, 17] = e11
A[18, 18] = e12

s = np.linalg.solve(A, b)
print('s2,0 = ' + str(s[0]))
print('s1,1 = ' + str(s[1]))
print('s2,1 = ' + str(s[2]))
print('s1,2 = ' + str(s[3]))
print('s2,2 = ' + str(s[4]))
print('s1,3 = ' + str(s[5]))
print('s2,3 = ' + str(s[6]))
print('s1,4 = ' + str(s[7]))
print('s2,4 = ' + str(s[8]))
print('s1,5 = ' + str(s[9]))
print('s2,5 = ' + str(s[10]))
print('s1,6 = ' + str(s[11]))
print('s2,6 = ' + str(s[12]))
print('s1,7 = ' + str(s[13]))
print('s2,7 = ' + str(s[14]))
print('s1,8 = ' + str(s[15]))
print('s2,8 = ' + str(s[16]))
print('s1,9 = ' + str(s[17]))
print('s2,9 = ' + str(s[18]))

# solution from eqn 17a
x = np.linspace(xjvec[0], xjvec[1])
y = np.zeros(x.size)
for i in range(x.size):
    y[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s[0]]]))[0]
# solutions from eqn 17b
for j in xjvec[1:-1]:
    xnew = np.linspace(j, j+1)
    ynew = np.zeros(xnew.size)
    for i in range(xnew.size):
```

```
        ynew[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * (xnew[i] - j)) @␣
 ↪np.array([[s[2*j-1][0]], [s[2*j][0]]]))[0]
    x = np.append(x, xnew)
    y = np.append(y, ynew)
plt.plot(x, y, label='Unperturbed s')

sperturb = s + 1e-8
# solution from eqn 17a
xperturb = np.linspace(xjvec[0], xjvec[1])
yperturb = np.zeros(xperturb.size)
for i in range(xperturb.size):
    yperturb[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * xperturb[i]) @␣
 ↪np.array([[alpha], [sperturb[0]]]))[0]
# solutions from eqn 17b
for j in xjvec[1:-1]:
    xnew = np.linspace(j, j+1)
    ynew = np.zeros(xnew.size)
    for i in range(xnew.size):
        ynew[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * (xnew[i] - j)) @␣
 ↪np.array([[sperturb[2*j-1][0]], [sperturb[2*j][0]]]))[0]
    xperturb = np.append(xperturb, xnew)
    yperturb = np.append(yperturb, ynew)
plt.plot(xperturb, yperturb, ':', label='Perturbed s')
plt.legend()
```

```
s2,0 = [185.68444885]
s1,1 = [99.06398948]
s2,1 = [23.43206991]
s1,2 = [77.85370695]
s2,2 = [-49.89438895]
s1,3 = [24.74108692]
s2,3 = [-47.83183337]
s1,4 = [-9.19692842]
s2,4 = [-19.23558844]
s1,5 = [-16.32953992]
s2,5 = [2.47923168]
s1,6 = [-9.44991206]
s2,6 = [9.02478862]
s1,7 = [-1.41931877]
s2,7 = [6.18046085]
s1,8 = [2.36099552]
s2,8 = [1.53714741]
s1,9 = [2.37762831]
s2,9 = [-1.06563092]
```

[27]: <matplotlib.legend.Legend at 0x1d3f14c0670>



part ii.

[26]:
```
import math
import numpy as np
from scipy.linalg import expm
import matplotlib.pyplot as plt

np.warnings.filterwarnings('ignore', category=np.VisibleDeprecationWarning)

alpha = 0
beta = 1
gamma = -2
delta = -2
xjvec = [i for i in range(0, 11)]
# See written pdf for derivation of A, b
E = expm(np.array([[0, 1], [-delta, -gamma]]))
e11 = E[0, 0]
e12 = E[0, 1]
e21 = E[1, 0]
e22 = E[1, 1]
b = np.zeros((19, 1))
b[0] = -e11 * alpha
b[1] = -e21 * alpha
b[18] = beta
A = np.zeros((19, 19))
# From eqn 18a
```

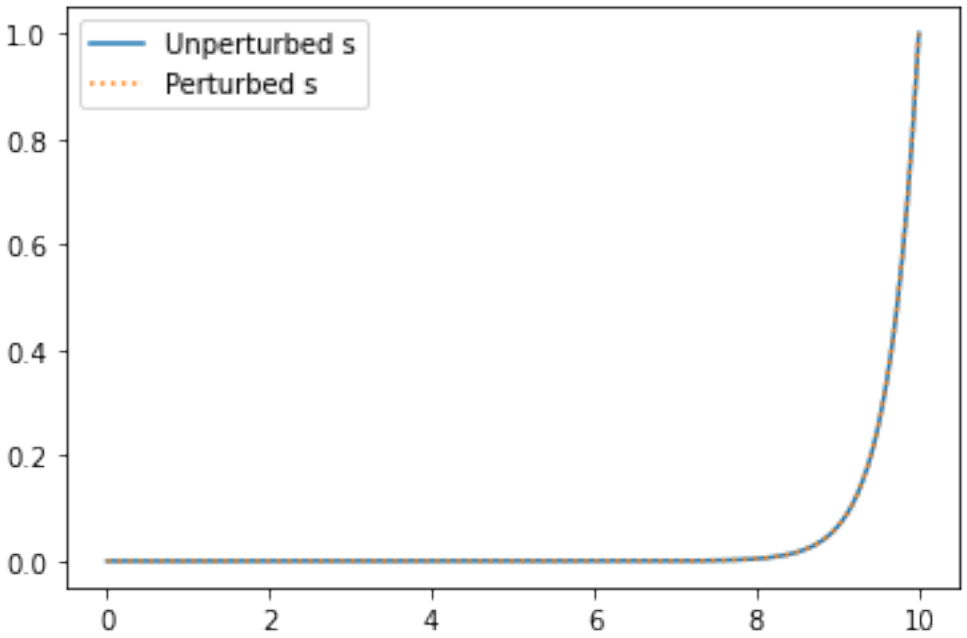
```
A[0, 0] = e12
A[0, 1] = -1
A[1, 0] = e22
A[1, 2] = -1
# From eqn 18b
for n in range(1, 9):
    A[2*n, 2*n-1] = e11
    A[2*n, 2*n] = e12
    A[2*n, 2*n+1] = -1
    A[2*n+1, 2*n-1] = e21
    A[2*n+1, 2*n] = e22
    A[2*n+1, 2*n+2] = -1
# from eqn 18c
A[18, 17] = e11
A[18, 18] = e12

s = np.linalg.solve(A, b)
print('s2,0 = ' + str(s[0]))
print('s1,1 = ' + str(s[1]))
print('s2,1 = ' + str(s[2]))
print('s1,2 = ' + str(s[3]))
print('s2,2 = ' + str(s[4]))
print('s1,3 = ' + str(s[5]))
print('s2,3 = ' + str(s[6]))
print('s1,4 = ' + str(s[7]))
print('s2,4 = ' + str(s[8]))
print('s1,5 = ' + str(s[9]))
print('s2,5 = ' + str(s[10]))
print('s1,6 = ' + str(s[11]))
print('s2,6 = ' + str(s[12]))
print('s1,7 = ' + str(s[13]))
print('s2,7 = ' + str(s[14]))
print('s1,8 = ' + str(s[15]))
print('s2,8 = ' + str(s[16]))
print('s1,9 = ' + str(s[17]))
print('s2,9 = ' + str(s[18]))

# solution from eqn 17a
x = np.linspace(xjvec[0], xjvec[1])
y = np.zeros(x.size)
for i in range(x.size):
    y[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * x[i]) @ np.
 ↪array([[alpha], [s[0]]]))[0]
# solutions from eqn 17b
for j in xjvec[1:-1]:
    xnew = np.linspace(j, j+1)
    ynew = np.zeros(xnew.size)
```

```
    for i in range(xnew.size):
        ynew[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * (xnew[i] - j)) @␣
 ↪np.array([[s[2*j-1][0]], [s[2*j][0]]]))[0]
    x = np.append(x, xnew)
    y = np.append(y, ynew)
plt.plot(x, y, label='Unperturbed s')

sperturb = s + 1e-8
print(sperturb)
# solution from eqn 17a
xperturb = np.linspace(xjvec[0], xjvec[1])
yperturb = np.zeros(xperturb.size)
for i in range(xperturb.size):
    yperturb[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * xperturb[i]) @␣
 ↪np.array([[alpha], [sperturb[0]]]))[0]
# solutions from eqn 17b
for j in xjvec[1:-1]:
    xnew = np.linspace(j, j+1)
    ynew = np.zeros(xnew.size)
    for i in range(xnew.size):
        ynew[i] = (expm(np.array([[0, 1], [-delta, -gamma]]) * (xnew[i] - j)) @␣
 ↪np.array([[sperturb[2*j-1][0]], [sperturb[2*j][0]]]))[0]
    xperturb = np.append(xperturb, xnew)
    yperturb = np.append(yperturb, ynew)
plt.plot(xperturb, yperturb, ':', label='Perturbed s')
plt.legend()
```

```
s2,0 = [4.72546687e-12]
s1,1 = [2.03028727e-11]
s2,1 = [5.77410592e-11]
s1,2 = [3.21704819e-10]
s2,2 = [8.80006845e-10]
s1,3 = [4.94748574e-09]
s2,3 = [1.3517308e-08]
s1,4 = [7.60172305e-08]
s2,4 = [2.07683189e-07]
s1,5 = [1.16795749e-06]
s2,5 = [3.19091933e-06]
s1,6 = [1.79449247e-05]
s2,6 = [4.9026446e-05]
s1,7 = [0.00027571]
s2,7 = [0.00075326]
s1,8 = [0.00423615]
s2,8 = [0.01157336]
s1,9 = [0.06508567]
s2,9 = [0.17781737]
[[1.00047255e-08]
```

```
[1.00203029e-08]
[1.00577411e-08]
[1.03217048e-08]
[1.08800068e-08]
[1.49474857e-08]
[2.35173080e-08]
[8.60172305e-08]
[2.17683189e-07]
[1.17795749e-06]
[3.20091933e-06]
[1.79549247e-05]
[4.90364460e-05]
[2.75722356e-04]
[7.53270166e-04]
[4.23615503e-03]
[1.15733734e-02]
[6.50856845e-02]
[1.77817380e-01]]
```

[26]: `<matplotlib.legend.Legend at 0x1d3f11cc5e0>`



[10]:
```
from scipy.linalg import expm
import numpy as np

alpha = 0
beta = 1
```

```
gamma = -2
delta = -2

E = expm(np.array([[0, 1], [-delta, -gamma]]))
e11 = E[0, 0]
e12 = E[0, 1]
e21 = E[1, 0]
e22 = E[1, 1]

A = np.zeros((19, 19))
# From eqn 18a
A[0, 0] = e12
A[0, 1] = -1
A[1, 0] = e22
A[1, 2] = -1
# From eqn 18b
for n in range(1, 9):
    A[2*n, 2*n-1] = e11
    A[2*n, 2*n] = e12
    A[2*n, 2*n+1] = -1
    A[2*n+1, 2*n-1] = e21
    A[2*n+1, 2*n] = e22
    A[2*n+1, 2*n+2] = -1
# from eqn 18c
A[18, 17] = e11
A[18, 18] = e12

b = np.zeros((19, 1))
b[0] = -e11 * alpha
b[1] = -e21 * alpha
b[18] = beta

kappa_2 = np.linalg.norm(A, 2) * np.linalg.norm(np.linalg.inv(A),2)
print('kappa_2 = ' + str(kappa_2))

s = np.linalg.solve(A, b)

print('Max error: ' + str(kappa_2 * np.linalg.norm(np.ones(19) * 1e-8,2) / np.
 ↪linalg.norm(s, 2)))
```

```
kappa_2 = 31.70067813307599
Max error: 7.281949918325125e-06
```