min_esssup

December 23, 2019

[1]:
```python
from sympy import *
from sympy.plotting import plot_parametric
```

[2]:
```python
A = Matrix([[-7, 5],
            [-6, 4]])
A
```

[2]: $\begin{bmatrix}-7 & 5\\-6 & 4\end{bmatrix}$

[3]:
```python
B = Matrix([[-2],
            [3]])
B
```

[3]: $\begin{bmatrix}-2\\3\end{bmatrix}$

[4]:
```python
x0 = Matrix([[3],
             [2]])
x0
```

[4]: $\begin{bmatrix}3\\2\end{bmatrix}$

[5]:
```python
x1 = Matrix([[2],
             [2]])
x1
```

[5]: $\begin{bmatrix}2\\2\end{bmatrix}$

[6]:
```python
t0 = 0
t1 = log(2)
```

[8]:
```python
eigs = A.eigenvects()
pprint(eigs)
```

```
         1             5/6
-2, 1,     ,  -1, 1,
```

1 1

[48]:
```python
t, tau = symbols('t tau', real=True, nonnegative=True)
```

[49]:
```python
Phi = (exp(eigs[0][0] * t) * eigs[0][2][0]).col_insert(1, exp(eigs[1][0] * t) *␣
↪eigs[1][2][0])
expA = Phi @ Phi.subs(t, 0).inv()
```

[50]:
```python
expA
```

[50]:

$$\begin{bmatrix} -5e^{-t} + 6e^{-2t} & 5e^{-t} - 5e^{-2t} \\ -6e^{-t} + 6e^{-2t} & 6e^{-t} - 5e^{-2t} \end{bmatrix}$$

[51]:
```python
c = x1 - expA.subs(t, t1 - t0) @ x0
c
```

[51]:

$$\begin{bmatrix} \frac{5}{2} \\ 3 \end{bmatrix}$$

[52]:
```python
H = expA.subs(t, t1 - tau) @ B
H
```

[52]:

$$\begin{bmatrix} -\frac{27e^{2\tau}}{4} + \frac{25e^{\tau}}{2} \\ -\frac{27e^{2\tau}}{4} + 15e^{\tau} \end{bmatrix}$$

[53]:
```python
B.col_insert(1, A @ B)
```

[53]:

$$\begin{bmatrix} -2 & 29 \\ 3 & 24 \end{bmatrix}$$

[54]:
```python
l1, l2 = symbols('l1 l2', real=True)
l = Matrix([[l1],
            [l2]])
l
```

[54]:

$$\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$$

[55]:
```python
integrate(abs(H.T @ l), (tau, t0, t1))[0]
```

[55]:

$$\frac{\int\limits_{0}^{\log(2)} e^{\tau} \left|27l_1e^{\tau} - 50l_1 + 27l_2e^{\tau} - 60l_2\right| d\tau}{4}$$

[56]:
```python
c
```

[56]:

$$\begin{bmatrix} \frac{5}{2} \\ 3 \end{bmatrix}$$

[57]:
```python
l
```

[57]:
$$\begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$$

[58]:
```python
sol = solve(l.dot(c) - 1, l2)
```

[59]:
```python
sol
```

[59]:
```
[1/3 - 5*l1/6]
```

[60]:
```python
tmp = simplify((H.T @ l).subs(l2, sol[0])[0])
tmp
```

[60]:
$$\frac{\left(-9 l_1 e^{\tau} - 18 e^{\tau} + 40\right) e^{\tau}}{8}$$

[61]:
```python
integrate(H, (tau, t0, t1))
```

[61]:
$$\begin{bmatrix} \frac{19}{8} \\ \frac{39}{8} \end{bmatrix}$$

[62]:
```python
taus = symbols('tau_s', real=True)
taus
```

[62]:
$$\tau_s$$

[63]:
```python
mu, sig = symbols('mu sigma', real=True)
```

[64]:
```python
expr_l = expA.subs(t, taus - t1) @ c + integrate(expA.subs(t, taus - tau) @ B *
 ↪mu * sig, (tau, t1, taus))
expr_l
```

[64]:
$$\begin{bmatrix} \frac{23\mu\sigma}{2} - 50\mu\sigma e^{-\tau_s} + 54\mu\sigma e^{-2\tau_s} + 5e^{-\tau_s} \\ \frac{33\mu\sigma}{2} - 60\mu\sigma e^{-\tau_s} + 54\mu\sigma e^{-2\tau_s} + 6e^{-\tau_s} \end{bmatrix}$$

[65]:
```python
expr_r = -integrate(expA.subs(t, taus - tau) @ B * mu * sig, (tau, t0, taus))
expr_r
```

[65]:
$$\begin{bmatrix} -\frac{23\mu\sigma}{2} + 25\mu\sigma e^{-\tau_s} - \frac{27\mu\sigma e^{-2\tau_s}}{2} \\ -\frac{33\mu\sigma}{2} + 30\mu\sigma e^{-\tau_s} - \frac{27\mu\sigma e^{-2\tau_s}}{2} \end{bmatrix}$$

[66]:
```python
sol_1 = solve((expr_l - expr_r).subs(sig, 1), (mu, taus))
pprint(sol_1)
```

```
  √10   3              log(10)
 ─── - ─ , -log(2) + ───────
   5    5                2
```

[67]:
```python
sol_2 = solve((expr_l - expr_r).subs(sig, -1), (mu, taus))
pprint(sol_2)
```

```
 3   √10              log(10)
  +    , -log(2) +
 5    5                  2
```

[68]:
```python
mu_opt = -sol_2[0][0]
mu_opt
```

[68]:
$$-\frac{\sqrt{10}}{5} - \frac{3}{5}$$

[69]:
```python
u_opt = mu_opt * sign(tau - sol_2[0][1])
u_opt
```

[69]:
$$\left(-\frac{\sqrt{10}}{5} - \frac{3}{5}\right) \operatorname{sign}\left(\tau - \frac{\log(10)}{2} + \log(2)\right)$$

[70]:
```python
simplify(u_opt)
```

[70]:
$$-\frac{\left(3 + \sqrt{10}\right) \operatorname{sign}\left(2\tau + \log\left(\frac{2}{5}\right)\right)}{5}$$

[72]:
```python
x_opt = simplify(expA.subs(t, t - t0) @ x0 + integrate(expA.subs(t, t - tau) @
 ↪B * u_opt, (tau, t0, t)))
x_opt
```

[72]:
$$\begin{bmatrix} \frac{((-54(3+\sqrt{10})e^{-2t+2\min(t,-\log(2)+\frac{\log(10)}{2})}+100(3+\sqrt{10})e^{-t+\min(t,-\log(2)+\frac{\log(10)}{2})}-23\sqrt{10}-69)e^{2t}-50(\sqrt{10}+4)e^{t}+27\sqrt{10}+161)e^{-2t}}{10} \\ \frac{(3(-18(3+\sqrt{10})e^{-2t+2\min(t,-\log(2)+\frac{\log(10)}{2})}+40(3+\sqrt{10})e^{-t+\min(t,-\log(2)+\frac{\log(10)}{2})}-11\sqrt{10}-33)e^{2t}-60(\sqrt{10}+4)e^{t}+27\sqrt{10}+161)e^{-2t}}{10} \end{bmatrix}$$

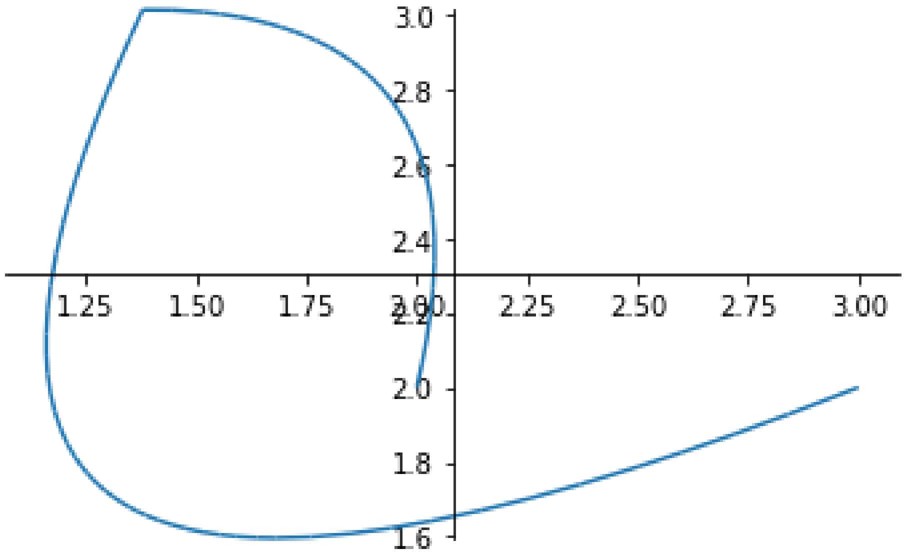
[79]:
```python
mu_opt
```

[79]:
$$-\frac{\sqrt{10}}{5} - \frac{3}{5}$$

[80]:
```python
ts = sol_2[0][1]
ts
```

[80]:
$$-\log(2) + \frac{\log(10)}{2}$$

[90]:
```python
plot_parametric(x_opt[0], x_opt[1], (t, t0, t1))
```




```
[90]: <sympy.plotting.plot.Plot at 0x7f2924a1cbd0>
```

```
[ ]:
```