AppendixA

July 7, 2015

1 Definitions

```
In [52]: from sympy import *
         t, m, a, a_1, a_2, m_1, m_2, b = symbols('t m a a_1 a_2 m_1 m_2 b')
         t_star = symbols('t_star')
```

```
In [54]: ##### This works
         ##### DO NOT CHANGE!
         ##### Copy and modify only
         from sympy.stats import Uniform, density
         X = Uniform("x", 0,1)

         part1 = -(a_1 - t - (1-t)*b)**2
         part2 = -(a_2 - t - (1-t)*b)**2

         t_sol = solve(diff(integrate(part1*density(X)(t), (t, 0, t_star)) + integrate(part2*density(X)
         print t_sol[0].args[0][0]
```

```
(-a_1 - a_2 + 2*b)/(2*(b - 1))
```

```
In [55]: ##### This works
         ##### DO NOT CHANGE!
         ##### Copy and modify only
         X = Uniform("x", 0,1)

         part1 = -(a_1 - t - (1-t)*b)**2
         part2 = -(a_2 - t - (1-t)*b)**2

         EU_S = integrate(part1*density(X)(t), (t, 0, t_star)) + integrate(part2*density(X)(t), (t, t_s
         t_diff = diff(EU_S, t_star)
         t_sol = solve(t_diff, t_star)
         t_sol_f = t_sol[0].args[0][0]
         print t_sol_f
         EU_R = EU_S.subs(b, 0)
         a1_diff = diff(EU_R, a_1)
         a1_sol = solve(a1_diff, a_1)
         a1_sol_f = a1_sol[1].args[0][0]
         a2_diff = diff(EU_R, a_2)
         a2_sol = solve(a2_diff, a_2)
         a2_sol_f = a2_sol[1].args[0][0]
         # Solve system of equations
         solve([Eq(t_star, t_sol_f), Eq(a_1, a1_sol_f), Eq(a_2, a2_sol_f)], [t_star, a_1, a_2])
```

```
(-a_1 - a_2 + 2*b)/(2*(b - 1))
```

```
Out[55]: {t_star: (4*b - 1)/(2*(2*b - 1)),
          a_1: (4*b - 1)/(4*(2*b - 1)),
          a_2: (8*b - 3)/(4*(2*b - 1))}
```

```
In [9]: X = Uniform("x", 0,1)

        part1 = -(a_1 - t - (1-t)*b)**2
        part2 = -(a_2 - t - (1-t)*b)**2

        EU_S = integrate(part1*density(X)(t), (t, 0, t_star), conds='none') + integrate(part2*density(X)
        t_diff = diff(EU_S, t_star)
        t_sol = solve(t_diff, t_star)
        t_sol_f = t_sol[0].args[0][0]
        print t_sol_f
        EU_R = EU_S.subs(b, 0)
        a1_diff = diff(EU_R, a_1)
        a1_sol = solve(a1_diff, a_1)
        print a1_sol
        a1_sol_f = a1_sol[1].args[0][0]
        print a1_sol_f
        a2_diff = diff(EU_R, a_2)
        a2_sol = solve(a2_diff, a_2)
        a2_sol_f = a2_sol[1].args[0][0]
        print a2_sol_f
        # Solve system of equations
        sys_sols = solve([Eq(t_star, t_sol_f), Eq(a_1, a1_sol_f), Eq(a_2, a2_sol_f)], [t_star, a_1, a_2
```

```
(-a_1 - a_2 + 2*b)/(2*(b - 1))
[1/2, Piecewise((t_star/2, And(0 <= t_star, t_star <= 1)), (nan, True))]
t_star/2
t_star/2 + 1/2
```

```
In [10]: t_plot = sys_sols[t_star]
         a1_plot = sys_sols[a_1]
         a2_plot = sys_sols[a_2]
         from matplotlib import pyplot as plt
         % matplotlib inline
         print t_plot
         plot(t_plot, a1_plot, a2_plot, .5, (b, 0, .24999), ylim=(0,1))
         #plt.axvline(x = .25)
```

```
(4*b - 1)/(2*(2*b - 1))
```

max size=0.90.9AppendixA$_f$*iles/AppendixA$_{51}$.png*

```
Out[10]: <sympy.plotting.plot.Plot at 0x105cc7f90>
```