This is a program for “Mathematica”.

```
Clear["Global`*"]
```
クリア

1. Global variables and specification of the functional form

Global variable α

```
α = 0.4;
```

Dynamical system of laisse-faire economy under n = ng; $H(k_t, k_{t+1}) = 0$ (Equation (36)).

$$w[k\_, A\_] := A (1-\alpha) \left((1-\alpha) + \alpha * k^{1-\frac{1}{\tau}}\right)^{\frac{1}{\tau-1}};$$

$$r[k\_, A\_] := A * \alpha \left((1-\alpha)\ k^{-\left(1-\frac{1}{\tau}\right)} + \alpha\right)^{\frac{1}{\tau-1}};$$

$$H[k1\_, k2\_, ng\_, A\_, \beta\_] := w[k1, A] - ng * k2 \left(\beta^{-\mu}\ r[k2, A]^{1-\mu} + 1\right);$$

The slope of $H(k_t, k_{t+1}) = 0$ at k_t=k_{t+1}=k STB(k)(Equation (37)).

```
H1[k1_, k2_, ng_, A_, β_] = D[H[k1, k2, ng, A, β], k1];
```
微分係数

```
H2[k1_, k2_, ng_, A_, β_] = D[H[k1, k2, ng, A, β], k2];
```
微分係数

$$STB[k\_, ng\_, A\_, \beta\_] := -\frac{H1[k, k, ng, A, \beta]}{H2[k, k, ng, A, \beta]}$$

2. Main Program

Setting the range of τ, and the length of grid as follows:

```
τmin = 0.4; τmax = 0.7; τgrid = 0.01;
```

Main program 1 (benchmark case): The case where A=1 and β=0.5.

Given τ, we seek μ which satisfies dk(t+1)/dk(t)=1 (i.e., STB=1) under the MGG steady state. Remember that the following two equation hold in the MGG steady state:

$\alpha\left(\beta^{-\mu}\ n^{1-\mu} + 1\right) == (1-\alpha)\ k^{-\frac{\tau-1}{\tau}}$ and $\left((1-\alpha)\ k^{-\frac{\tau-1}{\tau}} + \alpha\right)^{\frac{1}{\tau-1}} == \frac{n}{A * \alpha}$.

We seek the triplet (μ, k, n) which satisfies the above three equations. We denote the solution by (μg, kg, ng).

Running this program, you will see alerts which indicates that the accuracy of the numerical simulation is low when the value of τ is high. So, I employ only the results which satisfies $|STB-1|<10^{-8}$ (that is, the accuracy are sufficiently high).

The main result, the list of pairs of (τ, μg) in the MGG steady state, is stored in the list entitled

"resultBorder1."
In addition, the list of the pair of (τ, STB) obtained under (μg, kg, ng) is stored in the list entitled "resultSTB1." This is useful to check the accuracy of the result of the simulation.

```
Module[{A = 1, β = 0.5, resultBorder = {}, resultSTB = {}}, Do[solMGG = FindRoot[
モジュール                                                  反復指定          根を求める
    {α (β^-μ n^(1-μ) + 1) == (1 - α) k^(-(τ-1)/τ), ((1 - α) k^(-(τ-1)/τ) + α)^(1/(τ-1)) == n/(A * α), STB[k, n, A, β] == 1},
    {n, A * α^(τ/(τ-1)) / 10}, {k, (α/(1 - α) (β^-0.5 (A * α^(τ/(τ-1)) / 10)^0.5 + 1))^2}, {μ, 0.5}];
   {ng, kg, μg} = Chop[{n, k, μ} /. solMGG];
                  近い数にする
   STBval = STB[kg, ng, A, β] /. μ → μg;
   If[0 ≤ μg && Abs[STBval - 1] < 10^-8, resultBorder = Append[resultBorder, {τ, μg}]];
               絶対値                                  追加
   resultSTB = Append[resultSTB, {τ, STBval}],
               追加
   {τ, τmin, τmax, τgrid}];
  resultBorder1 = resultBorder;
  resultSTB1 = resultSTB];
```

Main program 2 (Effect of an increase in β): The case where A=1 and β=1.

Main result is stored in "resultBorder2." Moreover, we make the list entitled "resultSTB2" to check the accuracy of the simulation result.

```
Module[{A = 1, β = 1, resultBorder = {}, resultSTB = {}}, Do[solMGG = FindRoot[
モジュール                                                反復指定          根を求める
    {α (β^-μ n^(1-μ) + 1) == (1 - α) k^(-(τ-1)/τ), ((1 - α) k^(-(τ-1)/τ) + α)^(1/(τ-1)) == n/(A * α), STB[k, n, A, β] == 1},
    {n, A * α^(τ/(τ-1)) / 10}, {k, (α/(1 - α) (β^-0.5 (A * α^(τ/(τ-1)) / 10)^0.5 + 1))^2}, {μ, 0.5}];
   {ng, kg, μg} = Chop[{n, k, μ} /. solMGG];
                  近い数にする
   STBval = STB[kg, ng, A, β] /. μ → μg;
   If[0 ≤ μg && Abs[STBval - 1] < 10^-8, resultBorder = Append[resultBorder, {τ, μg}]];
               絶対値                                  追加
   resultSTB = Append[resultSTB, {τ, STBval}],
               追加
   {τ, τmin, τmax, τgrid}];
  resultBorder2 = resultBorder;
  resultSTB2 = resultSTB];
```

Main program 3 (Effect of an increase in A): The case where A=5 and β=0.3.

Main result is stored in "resultBorder3." Moreover, we make the list entitled "resultSTB3" to check the accuracy of the simulation result.

```
Module[{A = 5, β = 0.3, resultBorder = {}, resultSTB = {}}, Do[solMGG = FindRoot[
モジュール                                                  反復指定          根を求める
    {α (β^-μ n^(1-μ) + 1) == (1 - α) k^(-(τ-1)/τ), ((1 - α) k^(-(τ-1)/τ) + α)^(1/(τ-1)) == n/(A * α), STB[k, n, A, β] == 1},
    {n, A * α^(τ/(τ-1)) / 10}, {k, (α/(1 - α) (β^-0.5 (A * α^(τ/(τ-1)) / 10)^0.5 + 1))^2}, {μ, 0.5}];
  {ng, kg, μg} = Chop[{n, k, μ} /. solMGG];
                 近い数にする
  STBval = STB[kg, ng, A, β] /. μ → μg;
  If[0 ≤ μg && Abs[STBval - 1] < 10^-8, resultBorder = Append[resultBorder, {τ, μg}]];
              絶対値                                   追加
  resultSTB = Append[resultSTB, {τ, STBval}],
              追加
  {τ, τmin, τmax, τgrid}];
 resultBorder3 = resultBorder;
 resultSTB3 = resultSTB];
```

4. Program to illustrate the main result on the τ-μ plane

The function “graph[resultBorder, dashlevel, thickness]” illustrates the main result on the τ-μ plane.

The functions “auxLine[dashlevel, thickness]” and “fixedPoint[size]” are not essential, but they are useful to show clearly that
the border always passes (τ, μ)=(0.5, 0.5) (Lemma 1).

```
graph[resultBorder_, dashlevel_, thickness_] := ListPlot[resultBorder,
                                                リストプロット
  PlotJoined → True, PlotRange → {{0.4, 0.71}, {0.1, 0.71}}, AxesOrigin → {0.4, 0.1},
               真      プロット範囲                               軸の原点
  AxesLabel → {Style["t", FontFamily → "Symbol", FontSize → 22, Italic], Style["m",
  軸のラベル     スタイル     フォントファミリ  記号       フォントサイズ    斜体     スタイル
     FontFamily → "Symbol", FontSize → 22, Italic]}, AspectRatio → 0.75, PlotStyle →
                  記号       フォントサイズ    斜体       縦横比                プロットスタイル
   {{AbsoluteDashing[{10, dashlevel}], AbsoluteThickness[thickness], Black}},
     破線の長さ                          絶対的な太さ                     黒
  AxesStyle → AbsoluteThickness[1.5], LabelStyle → Directive[Black, 16],
  軸のスタイル  絶対的な太さ                 ラベルスタイル  指示子     黒
  ImageSize → 140 * 300 / 25.4];
  画像サイズ

auxLine[dashlevel_, thickness_] :=
 ListPlot[{{{0, 0.5}, {0.5, 0.5}}, {{0.5, 0}, {0.5, 0.5}}}, PlotJoined → True,
 リストプロット                                                          真
  PlotRange → {{0.4, 0.71}, {0.1, 0.71}}, AxesOrigin → {0.4, 0.1},
  プロット範囲                               軸の原点
  AspectRatio → 0.75, PlotStyle → {{AbsoluteDashing[{4, dashlevel}],
                      プロットスタイル  破線の長さ
     AbsoluteThickness[thickness], Black}}, ImageSize → 140 * 300 / 25.4];
                                   黒       画像サイズ
```

```
fixedPoint[size_] := ListPlot[{{0.5, 0.5}},
                     リストプロット
  PlotRange → {{0.4, 0.71}, {0.1, 0.71}}, AxesOrigin → {0.4, 0.1}, AspectRatio → 0.75,
  プロット範囲                              軸の原点                    縦横比
  PlotStyle → {{PointSize[size / 100], Black}}, ImageSize → 140 * 300 / 25.4];
  プロットスタイル  ポイントサイズ          黒        画像サイズ
```

5 Results

Figure 9

```
fig09 = {graph[resultBorder1, 0, 1.5], auxLine[4, 0.6], fixedPoint[2]};

Show[fig09]
示す
```

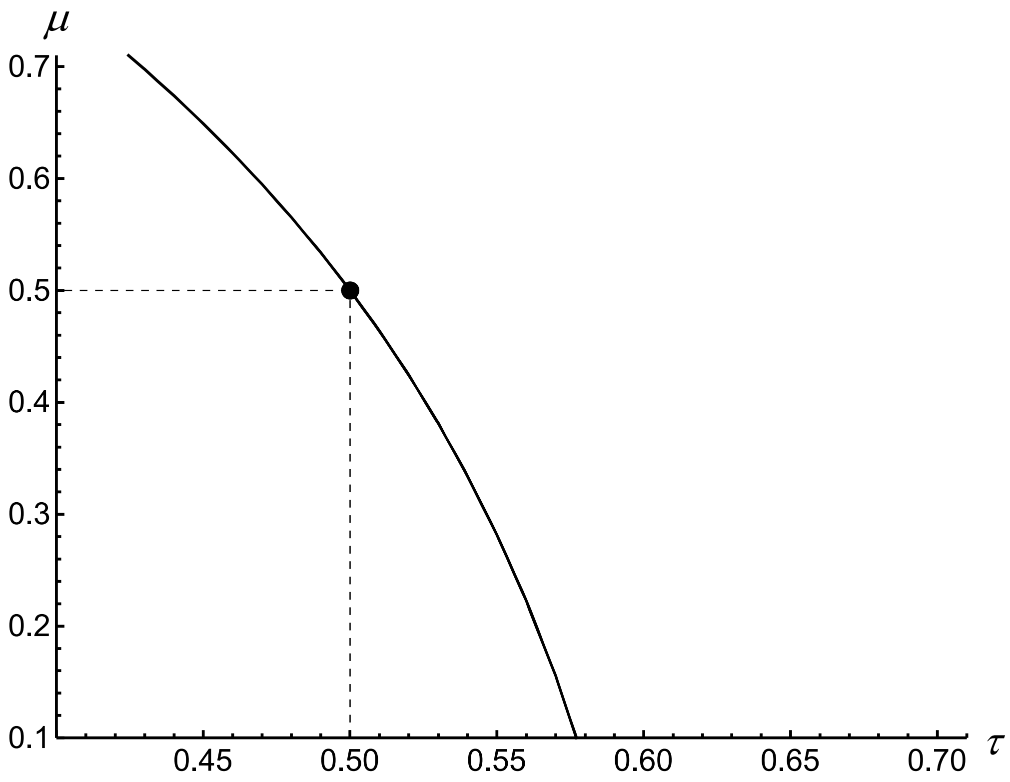

Figure 10

```
fig10 = {graph[resultBorder1, 5, 1.5],
  graph[resultBorder2, 0, 1.5], auxLine[4, 0.6], fixedPoint[2]};
```

```
Show[fig10]
```

示す

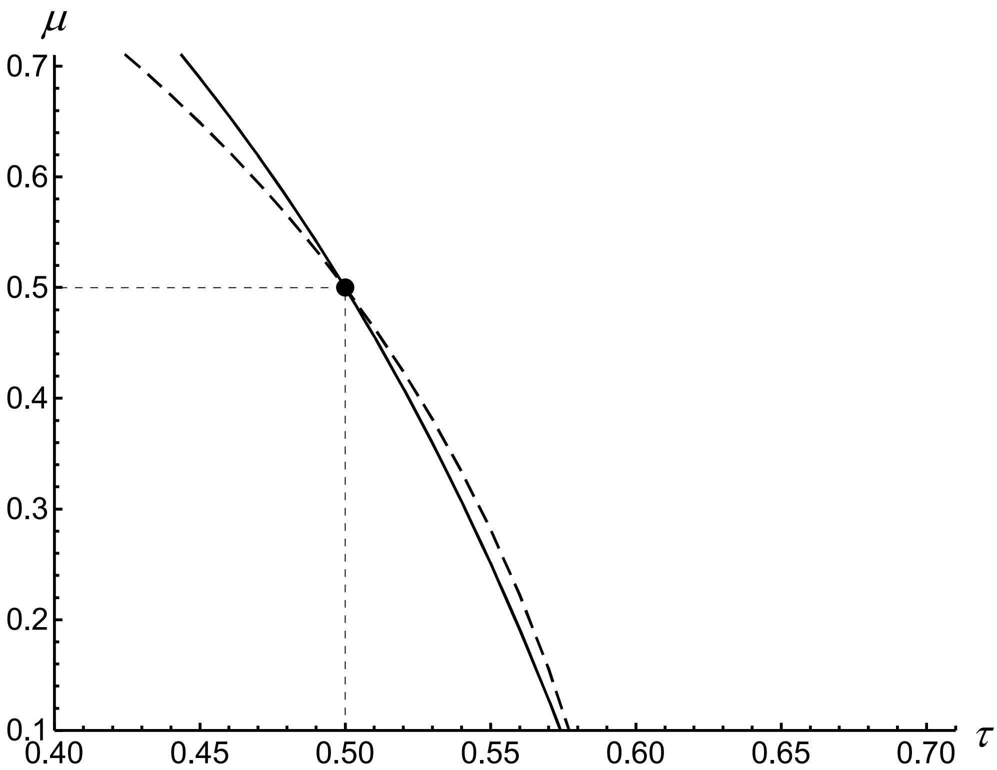

Figure 11

```
fig11 = {graph[resultBorder1, 5, 1.5],
   graph[resultBorder3, 0, 1.5], auxLine[4, 0.6], fixedPoint[2]};
```

```
Show[fig11]
```

示す