Marco Mid-term HW Calibration

Group2

2018/12/6

```r
rho2 <- 0.01035
delta2 <- 0.04
tau_k2 <- 0.03
alpha2 <- 0.64
sigma2 <- 0.7
A2 <- 1
epsilon2 <- 0.77
tau_l2 <- 0.12
tau_c2 <- 0.05
```

A. Before the Comparative Statics

As we know the l* = 0.3525, then we can solve initial k from the `return` function.

```r
lzero <- 0.3525

funtok <- function(k) {
  return(lzero-(((rho2+delta2*(1-tau_k2))/((1-tau_k2)*(1-alpha2)*A2))^(1/alpha2))*k)

}

uniroot(funtok, c(-3,100), tol = 0.000000001) -> solofk

solofk$root -> kzero

kzero
```

```
## [1] 7.545832
```

After we get kzero(initial k*), we can solve initial c.

```r
funtoc <- function(c) {

  return(c-(((rho2+delta2*(1-tau_k2))/((1-tau_k2)*(1-alpha2)))-delta2)*kzero)
}

uniroot(funtoc, c(-3,50), tol = 0.000000001) -> solofc

solofc$root -> czero

czero
```

```
## [1] 0.7602447
```

After getting the initial l* and k*, then we can solve initial y.

```r
yzero <- A2*(lzero^alpha2)*(kzero^(1-alpha2))
yzero
```

```
## [1] 1.062078
```

So far, we can get the endogenous variables in the model.

```
lzero
kzero
czero
yzero
```

B. Solve the chi in this model

```
funTochi <- function(chi){
    rho2 <- 0.01035
    delta2 <- 0.04
    tau_k2 <- 0.03
    alpha2 <- 0.64
    sigma2 <- 0.7
    A2 <- 1
    epsilon2 <- 0.77
    tau_l2 <- 0.12
    tau_c2 <- 0.05

    w <- A2*alpha2*(lzero^(alpha2-1))*(kzero^(1-alpha2))

M <- ((1-tau_l2)*w)/(chi*(1+tau_c2))

P <- (1-(((rho2+delta2*(1-tau_k2))/((1-tau_k2)*(1-alpha2)*A2))^(1/alpha2))*kzero)
Q <- (((1-tau_k2)*(1-alpha2))/(rho2+alpha2*delta2*(1-tau_k2)))

return(kzero-(((M^(1/sigma2))*(P^(epsilon2/sigma2))*Q)))
}

uniroot(funTochi, c(0,10000), tol = 0.0001, extendInt = "yes") -> solofchi2


chi2 <- solofchi2$root
chi2
```

```
## [1] 1.401078
```

At the same time, we have initial c and l, so we can slove initial u.

```
R <- ((czero^(1-sigma2)-1)/(1-sigma2))
S <- chi2*(((1-lzero)^(1-epsilon2))/(1-epsilon2))
uzero <- R+S
uzero
```

```
## [1] 5.248984
```

Now, we know all we need before the comparative statics.

```
lzero
kzero
czero
yzero
chi2
```

```r
c(c("kzero", "lzero", "czero", "yzero", "uzero"), round(c(kzero,lzero,czero,yzero,uzero), digits = 6)) -
Beforestatics
```

```
##  [1] "kzero"    "lzero"    "czero"    "yzero"    "uzero"    "7.545832"
##  [7] "0.3525"   "0.760245" "1.062078" "5.248984"
```

C. Comparative Statics

tau_k decrease from 0.03 to 0.025

```r
rho3 <- 0.01035
delta3 <- 0.04
tau_k3 <- 0.025
alpha3 <- 0.64
sigma3 <- 0.7
A3 <- 1
epsilon3 <- 0.77
tau_l3 <- 0.12
tau_c3 <- 0.05
chi3 <- chi2
```

Solve new k, and we name it *kone1*

```r
funToFindkone <- function(kone){

    l <- (((rho3+delta3*(1-tau_k3))/((1-tau_k3)*(1-alpha3)*A3))^(1/alpha3))*kone
    w <- A3*alpha3*l^(alpha3-1)*kone^(1-alpha3)

M <- ((1-tau_l3)*w)/(chi3*(1+tau_c3))

P <- (1-(((rho3+delta3*(1-tau_k3))/((1-tau_k3)*(1-alpha3)*A3))^(1/alpha3))*kone)
Q <- (((1-tau_k3)*(1-alpha3))/(rho3+alpha3*delta3*(1-tau_k3)))

return(kone-(((M^(1/sigma3))*(P^(epsilon3/sigma3))*Q)))
}

uniroot(funToFindkone, c(1,10), tol = 0.000000001) -> solofkone

kone1 <- solofkone$root
kone1
```

```
## [1] 7.561775
```

After getting k* (the `kone1`) from comparative statics, we can solve other endogenous variables.

```r
funoflcone <- function(kvalue) {

  lone <- (((rho3+delta3*(1-tau_k3))/((1-tau_k3)*(1-alpha3)*A3))^(1/alpha3))*kvalue
  cone <- (((rho3+delta3*(1-tau_k3))/((1-tau_k3)*(1-alpha3)))-delta3)*kvalue
  yone <- A3*(lone^alpha3)*(kvalue^(1-alpha3))

  R <- ((cone^(1-sigma3)-1)/(1-sigma3))
  S <- chi3*(((1-lone)^(1-epsilon3))/(1-epsilon3))
  uone <- R+S

  c(c("kone", "lone", "cone", "yone", "uone"), round(c(kvalue,lone,cone,yone,uone), digits = 6)) -> Afte
```

```
  Afterstatics
}




funoflcone(kone1)
```

```
## [1] "kone"     "lone"     "cone"     "yone"     "uone"     "7.561775"
## [7] "0.352649" "0.760702" "1.063173" "5.249246"
```

```
Beforestatics
```

```
## [1] "kzero"    "lzero"    "czero"    "yzero"    "uzero"    "7.545832"
## [7] "0.3525"   "0.760245" "1.062078" "5.248984"
```