ROGRAM CODE FOR THE MULTIVARIATE AUTOREGRESSIVE DISTRIBUTED LAG UNIT ROOT TEST

For FRGS/1/2020/SS0/USM/02/25

Main Program

```
' ========== ARDL unit root test =============
' *** This program using n=10,000 bootstrap replication to generate bootstrap critical values.
' *** Declare your variables and testing period. Put them between the " ". For example, "GDP".
' Declare your variables.
%dependent = "aust"
%independent1 = "us"
' Declare your testing period.
%firstperiod = "2000m01"
%lastperiod = "2023m12"
' Declare your regression specification, whether it is no-intercept, intercept or trend.
' If no-intercept, regtype = n; If intercept, regtype = i; If trend, regtype = t
%regtype = "i"
' Declare your workfile data type. It can be annual, semi-annual,quarterly, monthly, weekly, daily, or undated.
' For annual, put "a"; for semi-annual, put "s"; for quarterly, put "q"; for monthly, put "m"; for weekly, put "w"; for
daily, put "d"; for undated, put "u".
%datatype = "m"
' If you wish to set your own lag length for ARDL(p,q), put "Y". Otherwise, put "N" and the program will help you to
determine the optimal lag length.
%selfspec = "Y"
' If you put "N" for %selfspec, set your equation maximum lag length. Maximum 10 lags.
!maxlag = 2
' If you put "Y" for %selfspec, set your prefer lag lengths for dy and dx.
' The lag length p for dy.
!fix_p = 2
' The lag length q for dx.
!fix_q= 2
' Number of bootstrap replication. Recomended replication = 1000 or 2000 or 5000 or 10000.
!nrep_b = 5000

' ================== Program start here ========================
' ========= DO NOT CHANGE ANYTHING START FROM HERE! ==========
' *** Quiet mode.
mode quiet

pagecopy(page=ARDLurt_{%dependent}) {%dependent} {%independent1}
rndseed 12345                    ' Seed to generate random numbers.

genr y = {%dependent}
genr dy = d(y)
genr x1 = {%independent1}
genr dx1 = d(x1)

table(!maxlag+2, !maxlag+2) table_aic_y
setcell(table_aic_y, 1, 1, "ARDL(p,q)")
for !a=1 to !maxlag+1
setcell(table_aic_y, 1, 1+!a, @str(!a-1))
next

for !b=1 to !maxlag+1
setcell(table_aic_y, 1+!b, 1, @str(!b-1))
```

```
next

scalar opt_aic_y = na

vector(!nrep_b) tstat_b_dist = na
vector(!nrep_b) fstat_b_dist = na

        if %regtype = "n" then                  ' Case for regression without intercept.
            include .\ardlurt02_none
else    if %regtype = "i" then                      ' Case for regression with intercept only.
            include .\ardlurt03_intercept
else    if %regtype = "t" then                      ' Case for regression with intercept and trend.
            include .\ardlurt04_trend
endif
endif
endif

' ================================ End of Bootstrap ================================
vector(4) critical_value_y = na
vector(4) critical_value_x = na
%smplperiod = %firstperiod + " : " + %lastperiod
smpl @all
!k = 1
for %1 0.90 0.95 0.975 0.99
critical_value_x(!k) = @quantile(fstat_b_dist, %1)
!k = !k + 1
next

!m = 1
for %2 0.10 0.05 0.025 0.01
critical_value_y(!m) = @quantile(tstat_b_dist, %2)
!m = !m + 1
next


table(11,5) table_result
setcell(table_result, 1, 1, "ARDL Unit Root Test.")
setcell(table_result, 2, 1, "Dependent variable:")
setcell(table_result, 2, 2, %dependent)
setcell(table_result, 3, 1, "Independent variable:")
setcell(table_result, 3, 2, %independent1)
setcell(table_result, 4, 1, "Regression:")
if %regtype = "n" then
setcell(table_result, 4, 2, "None")
else
if %regtype = "i" then
setcell(table_result, 4, 2, "Intercept")
else
if %regtype = "t" then
setcell(table_result, 4, 2, "Trend")
endif
endif
endif
setcell(table_result, 3, 4, "Sample:")
setcell(table_result, 3, 5, %smplperiod)
setcell(table_result, 4, 4, "Optimal Model:")
setcell(table_result, 4, 5, "ARDL(" + @str(!opt_p) + ", " + @str(!opt_q) + ")")
setcell(table_result, 5, 4, "Akaike info criterion:")
setcell(table_result, 5, 5, opt_aic_y)
setcell(table_result, 6, 4, "Bootstrap replication")
setcell(table_result, 6, 5, !nrep_b)

setcell(table_result, 5, 1, "t-statistic:")
setcell(table_result, 6, 1, "F-statistic:")
setcell(table_result, 5, 2, tstat_y)
setcell(table_result, 6, 2, fstat_y)
```

```
setcell(table_result, 8, 1, "Sig. Level")
setcell(table_result, 8, 2, "0.100")
setcell(table_result, 8, 3, "0.050")
setcell(table_result, 8, 4, "0.025")
setcell(table_result, 8, 5, "0.010")

setcell(table_result, 9, 1, "t-critical value")
setcell(table_result, 10, 1, "F-critical value")

for !t=1 to 4
table_result(9, !t+1) = critical_value_y(!t)
next

for !t=1 to 4
table_result(10, !t+1) = critical_value_x(!t)
next

show table_result

stop



' ================ Sub-procedures for ARDL unit root test ==================
' ============== ARDL unit root test for no intercept model ================

if %selfspec = "N" then
    ' Determine the optimal lag length of ARDL model.
    if !maxlag = 0 then
        smpl %firstperiod %lastperiod
        equation eq_y.ls dy y(-1) x1(-1)
        table_aic_y(2,2) = eq_y.@aic
    else
        ' If maximum lag length more than 0, then determine the optimal model using algorithm below.
    if !maxlag > 0 then
        smpl %firstperiod %lastperiod
        equation eq_y.ls dy y(-1) x1(-1)       ' Calculate aic for ARDL(0,0) model.
        table_aic_y(2,2) = eq_y.@aic
        opt_aic_y = eq_y.@aic
        !opt_p = 0                             ' Optimal lag length for dy.
        !opt_q = 0                             ' Optimal lag length for dx.

        for !q=1 to !maxlag      ' Calculate the aic for 0 lag length for dy.
        smpl %firstperiod %lastperiod
        equation eq_y.ls dy y(-1) x1(-1) dx1(-1 to -!q)
        table_aic_y(2,2+!q) = eq_y.@aic
            if table_aic_y(2,2+!q) < opt_aic_y then
            opt_aic_y = eq_y.@aic
            !opt_p = 0
            !opt_q = !q
            endif
        next

        for !p=1 to !maxlag      ' Calculate the aic for 0 lag length for dx.
        smpl %firstperiod %lastperiod
        equation eq_y.ls dy y(-1) x1(-1) dy(-1 to -!p)
        table_aic_y(2+!p,2) = eq_y.@aic
            if table_aic_y(2+!p, 2) < opt_aic_y then
            opt_aic_y = eq_y.@aic
            !opt_p = !p
            !opt_q = 0
            endif
        next

        for !p=1 to !maxlag ' Calculate the aic for ARDL(m,n) model.
        for !q=1 to !maxlag
        smpl %firstperiod %lastperiod
```

```
        equation eq_y.ls dy y(-1) x1(-1) dy(-1 to -!p) dx1(-1 to -!q)
        table_aic_y(2+!p, 2+!q) = eq_y.@aic
            if table_aic_y(2+!p,2+!q) < opt_aic_y then
            opt_aic_y = eq_y.@aic
            !opt_p = !p
            !opt_q = !q
            endif
        next
        next
    endif
    endif

else
if %selfspec = "Y" then
    !opt_p = !fix_p
    !opt_q = !fix_q
endif
endif

' Determine the longest lag length of the opimal model.
if !opt_p >= !opt_q then
!opt_lag = !opt_p
else
if !opt_p <= !opt_q then
!opt_lag = !opt_q
endif
endif

table(1,1) check_opt_ardl                    ' Check the optimal lag length for the ARDL.
setcell(check_opt_ardl, 1, 1, "ARDL(" + @str(!opt_p) + ", " + @str(!opt_q) + ")")

if !opt_p = 0 and !opt_q = 0 then
%cond_y = "00"
else
if !opt_p = 0 and !opt_q <> 0 then
%cond_y = "01"
else
if !opt_p <> 0 and !opt_q = 0 then
%cond_y = "10"
else
if !opt_p <> 0 and !opt_q <> 0 then
%cond_y = "11"
endif
endif
endif
endif

if %cond_y = "00" then
    smpl %firstperiod %lastperiod
    equation eq_opt_y.ls dy y(-1) x1(-1)
    scalar tstat_y = eq_opt_y.@tstat(1)              ' Obtain the testing t statistic.
    freeze(mode=overwrite, fstat_x_table) eq_opt_y.wald c(2)=0
    scalar fstat_y = @val(fstat_x_table(7,2))        ' Obtain the testing F statistic.

    ' ============== Bootstrap start here =========================
    smpl %firstperiod %lastperiod
    ' Estimate equation with restriction of null y(-1) for t test and x(-1) for F test.
    equation restricted_t_y.ls dy x1(-1)             ' Impose y(-1) = 0.
    equation restricted_f_y.ls dy y(-1)              ' Impose x(-1) = 0.

    restricted_t_y.makeresids resids_t_y
    restricted_f_y.makeresids resids_f_y

    ' Recenter residuals
    scalar sum_resids_t_y = @csum(resids_t_y)
    scalar sum_resids_f_y = @csum(resids_f_y)
```

```
    scalar nobs_t = resids_t_y.@obs
    scalar nobs_f = resids_f_y.@obs

    for !c=@dtoo(%firstperiod) to @dtoo(%lastperiod)
    resids_t_y(!c) = resids_t_y(!c) - (sum_resids_t_y/nobs_t)
    resids_f_y(!c) = resids_f_y(!c) - (sum_resids_f_y/nobs_f)
    next

    for !id=1 to 1
    vector(1) coef_restrict_t_y(!id) = restricted_t_y.@coef(!id)
    next

    for !ie=1 to 1
    vector(1) coef_restrict_f_y(!ie) = restricted_f_y.@coef(!ie)
    next

    ' ========== Bootstrap replication start here ==================
    for !ib=1 to !nrep_b                    ' Bootstrap replication set for 10,000.
    smpl @all
    series y_t_b = y
    series dy_t_b = dy

    series y_f_b = y
    series dy_f_b = dy

    group gu resids_t_y resids_f_y
    gu.resample(dropna, outsmpl=%firstperiod %lastperiod, name=gu_b)

model a
    !start = @dtoo(%firstperiod) + 1
    %start = @otod(!start)
    smpl %start %lastperiod

    a.append dy_t_b = coef_restrict_t_y(1)*x1(-1) + resids_t_y_b
    a.append y_t_b = y_t_b(-1) + dy_t_b

    a.append dy_f_b = coef_restrict_f_y(1)*y_f_b(-1) + resids_f_y_b
    a.append y_f_b = y_f_b(-1) + dy_f_b

    a.scenario "actuals"
    a.solve

    smpl %firstperiod %lastperiod
    equation bootstrap_t_y.ls dy_t_b y_t_b(-1) x1(-1)
    scalar tstat_b = bootstrap_t_y.@tstat(1)

    tstat_b_dist(!ib) = tstat_b

    smpl %firstperiod %lastperiod
    equation bootstrap_f_y.ls dy_f_b y_f_b(-1) x1(-1)
    freeze(mode=overwrite,  fstat_x_b_table) bootstrap_f_y.wald c(2)=0
    scalar fstat_b = @val(fstat_x_b_table(7,2))

    fstat_b_dist(!ib) = fstat_b
    next
else
if %cond_y = "01" then
    smpl %firstperiod %lastperiod
    equation eq_opt_y.ls dy y(-1) x1(-1) dx1(-1 to -!opt_q)
    scalar tstat_y = eq_opt_y.@tstat(1)                    ' Obtain the testing t statistic.
    freeze(mode=overwrite, fstat_x_table) eq_opt_y.wald c(2)=0
    scalar fstat_y = @val(fstat_x_table(7,2))              ' Obtain the testing F statistic.

    ' ============== Bootstrap start here ==========================
    smpl %firstperiod %lastperiod
    ' Estimate equation with restriction of null y(-1) for t test and x(-1) for F test.
    equation restricted_t_y.ls dy x1(-1) dx1(-1 to -!opt_q)            ' Impose y(-1) = 0.
```

```
    equation restricted_f_y.ls dy y(-1) dx1(-1 to -!opt_q)              ' Impose x(-1) = 0.

    restricted_t_y.makeresids resids_t_y
    restricted_f_y.makeresids resids_f_y

    ' Recenter residuals
    scalar sum_resids_t_y = @csum(resids_t_y)
    scalar sum_resids_f_y = @csum(resids_f_y)

    scalar nobs_t = resids_t_y.@obs
    scalar nobs_f = resids_f_y.@obs

    for !c=@dtoo(%firstperiod) to @dtoo(%lastperiod)
    resids_t_y(!c) = resids_t_y(!c) - (sum_resids_t_y/nobs_t)
    resids_f_y(!c) = resids_f_y(!c) - (sum_resids_f_y/nobs_f)
    next

    !in = 1+!opt_q
    for !id=1 to !in
    vector(!id) coef_restrict_t_y(!id) = restricted_t_y.@coef(!id)
    next

    for !ie=1 to !in
    vector(!ie) coef_restrict_f_y(!ie) = restricted_f_y.@coef(!ie)
    next

    ' ============ Bootstrap replication start here ==================
    for !ib=1 to !nrep_b                    ' Bootstrap replication set for 10,000.
    smpl @all
    series y_t_b = y
    series dy_t_b = dy
    series dify_t_b = 0

    series y_f_b = y
    series dy_f_b = dy
    series dify_f_b = 0

    group gu resids_t_y resids_f_y
    gu.resample(dropna, outsmpl=%firstperiod %lastperiod, name=gu_b)

    model a
    !start = @dtoo(%firstperiod) + !opt_lag + 1
    %start = @otod(!start)
    smpl %start %lastperiod
    if !opt_q = 1 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1)
    else
    if !opt_q = 2 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2)
    else
    if !opt_q = 3 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3)
    else
    if !opt_q = 4 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4)
```

```
    else
    if !opt_q = 5 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4) + coef_restrict_t_y(6)*dx1(-5)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5)
    else
    if !opt_q = 6 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4) + coef_restrict_t_y(6)*dx1(-5) + coef_restrict_t_y(7)*dx1(-6)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5) + coef_restrict_f_y(7)*dx1(-6)
    else
    if !opt_q = 7 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4) + coef_restrict_t_y(6)*dx1(-5) + coef_restrict_t_y(7)*dx1(-6) +
coef_restrict_t_y(8)*dx1(-7)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5) + coef_restrict_f_y(7)*dx1(-6) +
coef_restrict_f_y(8)*dx1(-7)
    else
    if !opt_q = 8 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4) + coef_restrict_t_y(6)*dx1(-5) + coef_restrict_t_y(7)*dx1(-6) +
coef_restrict_t_y(8)*dx1(-7) + coef_restrict_t_y(9)*dx1(-8)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5) + coef_restrict_f_y(7)*dx1(-6) +
coef_restrict_f_y(8)*dx1(-7) + coef_restrict_f_y(9)*dx1(-8)
    else
    if !opt_q = 9 then
        a.append dify_t_b = coef_restrict_t_y(2)*dx1(-1) + coef_restrict_t_y(3)*dx1(-2) + coef_restrict_t_y(4)*dx1(-
3) + coef_restrict_t_y(5)*dx1(-4) + coef_restrict_t_y(6)*dx1(-5) + coef_restrict_t_y(7)*dx1(-6) +
coef_restrict_t_y(8)*dx1(-7) + coef_restrict_t_y(9)*dx1(-8) + coef_restrict_t_y(10)*dx1(-9)

        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5) + coef_restrict_f_y(7)*dx1(-6) +
coef_restrict_f_y(8)*dx1(-7) + coef_restrict_f_y(9)*dx1(-8) + coef_restrict_f_y(10)*dx1(-9)
    else
    if !opt_q = 10 then
        a.append dify_f_b = coef_restrict_f_y(2)*dx1(-1) + coef_restrict_f_y(3)*dx1(-2) + coef_restrict_f_y(4)*dx1(-
3) + coef_restrict_f_y(5)*dx1(-4) + coef_restrict_f_y(6)*dx1(-5) + coef_restrict_f_y(7)*dx1(-6) +
coef_restrict_f_y(8)*dx1(-7) + coef_restrict_f_y(9)*dx1(-8) + coef_restrict_f_y(10)*dx1(-9) +
coef_restrict_f_y(11)*dx1(-10)
endif
endif
endif
endif
endif
endif
endif
endif
endif
endif

a.append dy_t_b = coef_restrict_t_y(1)*x1(-1) + dify_t_b + resids_t_y_b
a.append y_t_b = y_t_b(-1) + dy_t_b

a.append dy_f_b = coef_restrict_f_y(1)*y_f_b(-1) + dify_f_b + resids_f_y_b
a.append y_f_b = y_f_b(-1) + dy_f_b

a.scenario "actuals"
a.solve
```

```
    smpl %firstperiod %lastperiod
    equation bootstrap_t_y.ls dy_t_b y_t_b(-1) x1(-1) dx1(-1 to -!opt_q)
    scalar tstat_b = bootstrap_t_y.@tstat(1)

    tstat_b_dist(!ib) = tstat_b

    smpl %firstperiod %lastperiod
    equation bootstrap_f_y.ls dy_f_b y_f_b(-1) x1(-1) dx1(-1 to -!opt_q)
    freeze(mode=overwrite,  fstat_x_b_table) bootstrap_f_y.wald c(2)=0
    scalar fstat_b = @val(fstat_x_b_table(7,2))

    fstat_b_dist(!ib) = fstat_b
    next
else
if %cond_y = "10" then
    smpl %firstperiod %lastperiod
    equation eq_opt_y.ls dy y(-1) x1(-1) dy(-1 to -!opt_p)
    scalar tstat_y = eq_opt_y.@tstat(1)                    ' Obtain the testing t statistic.
    freeze(mode=overwrite, fstat_x_table) eq_opt_y.wald c(2)=0
    scalar fstat_y = @val(fstat_x_table(7,2))              ' Obtain the testing F statistic.

    ' ============== Bootstrap start here ==========================
    smpl %firstperiod %lastperiod
    ' Estimate equation with restriction of null y(-1) for t test and x(-1) for F test.
    equation restricted_t_y.ls dy x1(-1) dy(-1 to -!opt_p)          ' Impose y(-1) = 0.
    equation restricted_f_y.ls dy y(-1) dy(-1 to -!opt_p)               ' Impose x(-1) = 0.

    restricted_t_y.makeresids resids_t_y
    restricted_f_y.makeresids resids_f_y

    ' Recenter residuals
    scalar sum_resids_t_y = @csum(resids_t_y)
    scalar sum_resids_f_y = @csum(resids_f_y)

    scalar nobs_t = resids_t_y.@obs
    scalar nobs_f = resids_f_y.@obs

    for !c=@dtoo(%firstperiod) to @dtoo(%lastperiod)
    resids_t_y(!c) = resids_t_y(!c) - (sum_resids_t_y/nobs_t)
    resids_f_y(!c) = resids_f_y(!c) - (sum_resids_f_y/nobs_f)
    next

    !in = 1+!opt_p
    for !id=1 to !in
    vector(!id) coef_restrict_t_y(!id) = restricted_t_y.@coef(!id)
    next

    for !ie=1 to !in
    vector(!ie) coef_restrict_f_y(!ie) = restricted_f_y.@coef(!ie)
    next

    ' ============ Bootstrap replication start here ==================
'   pagestruct(end = 5000+@dtoo(%lastperiod))
    for !ib=1 to !nrep_b                    ' Bootstrap replication set for 10,000.
    smpl @all
    series y_t_b = y
    series dy_t_b = dy
    series dify_t_b = 0

    series y_f_b = y
    series dy_f_b = dy
    series dify_f_b = 0

    group gu resids_t_y resids_f_y
    gu.resample(dropna, outsmpl=%firstperiod %lastperiod, name=gu_b)

model a
```

```
    !start = @dtoo(%firstperiod) + !opt_lag + 1
    %start = @otod(!start)
    smpl %start %lastperiod
    if !opt_p = 1 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1)
    else
    if !opt_p = 2 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2)
    else
    if !opt_p = 3 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3)
    else
    if !opt_p = 4 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4)
    else
    if !opt_p = 5 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5)
    else
    if !opt_p = 6 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6)
    else
    if !opt_p = 7 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7)
    else
    if !opt_p = 8 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8)
    else
    if !opt_p = 9 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8) +
coef_restrict_t_y(10)*dy_t_b(-9)
```

```
        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8) +
coef_restrict_f_y(10)*dy_f_b(-9)
    else
    if !opt_p = 10 then
        a.append dify_t_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8) +
coef_restrict_t_y(10)*dy_t_b(-9) + coef_restrict_t_y(11)*dy_t_b(-10)

        a.append dify_f_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8) +
coef_restrict_f_y(10)*dy_f_b(-9) + coef_restrict_f_y(11)*dy_f_b(-10)
endif
endif
endif
endif
endif
endif
endif
endif
endif
endif

a.append dy_t_b = coef_restrict_t_y(1)*x1(-1) + dify_t_b + resids_t_y_b
a.append y_t_b = y_t_b(-1) + dy_t_b

a.append dy_f_b = coef_restrict_f_y(1)*y_f_b(-1) + dify_f_b + resids_f_y_b
a.append y_f_b = y_f_b(-1) + dy_f_b

a.scenario "actuals"
a.solve

    smpl %firstperiod %lastperiod
    equation bootstrap_t_y.ls dy_t_b y_t_b(-1) x1(-1) dy_t_b(-1 to -!opt_p)
    scalar tstat_b = bootstrap_t_y.@tstat(1)

    tstat_b_dist(!ib) = tstat_b

    smpl %firstperiod %lastperiod
    equation bootstrap_f_y.ls dy_f_b y_f_b(-1) x1(-1) dy_f_b(-1 to -!opt_p)
    freeze(mode=overwrite,  fstat_x_b_table) bootstrap_f_y.wald c(2)=0
    scalar fstat_b = @val(fstat_x_b_table(7,2))

    fstat_b_dist(!ib) = fstat_b
    next
else
if %cond_y = "11" then
    smpl %firstperiod %lastperiod
    equation eq_opt_y.ls dy y(-1) x1(-1) dy(-1 to -!opt_p) dx1(-1 to -!opt_q)
    scalar tstat_y = eq_opt_y.@tstat(1)                    ' Obtain the testing t statistic.
    freeze(mode=overwrite, fstat_x_table) eq_opt_y.wald c(2)=0
    scalar fstat_y = @val(fstat_x_table(7,2))              ' Obtain the testing F statistic.

    ' ============== Bootstrap start here ==========================
    smpl %firstperiod %lastperiod
    ' Estimate equation with restriction of null y(-1) for t test and x(-1) for F test.
    equation restricted_t_y.ls dy x1(-1) dy(-1 to -!opt_p) dx1(-1 to -!opt_q)          ' Impose y(-1) = 0.
    equation restricted_f_y.ls dy y(-1) dy(-1 to -!opt_p)   dx1(-1 to -!opt_q)             ' Impose x(-1) = 0.

    restricted_t_y.makeresids resids_t_y
    restricted_f_y.makeresids resids_f_y

    ' Recenter residuals
    scalar sum_resids_t_y = @csum(resids_t_y)
```

```
    scalar sum_resids_f_y = @csum(resids_f_y)

    scalar nobs_t = resids_t_y.@obs
    scalar nobs_f = resids_f_y.@obs

    for !c=@dtoo(%firstperiod) to @dtoo(%lastperiod)
    resids_t_y(!c) = resids_t_y(!c) - (sum_resids_t_y/nobs_t)
    resids_f_y(!c) = resids_f_y(!c) - (sum_resids_f_y/nobs_f)
    next

    !in = 1+!opt_p+!opt_q
    for !id=1 to !in
    vector(!id) coef_restrict_t_y(!id) = restricted_t_y.@coef(!id)
    next

    for !ie=1 to !in
    vector(!ie) coef_restrict_f_y(!ie) = restricted_f_y.@coef(!ie)
    next

    ' ========== Bootstrap replication start here ==================
    for !ib=1 to !nrep_b                    ' Bootstrap replication set for 10,000.
    smpl @all
    series y_t_b = y
    series dy_t_b = dy
    series dify_ty_b = 0
    series dif_tx_b = 0

    series y_f_b = y
    series dy_f_b = dy
    series dify_fy_b = 0
    series dify_fx_b = 0

    group gu resids_t_y resids_f_y
    gu.resample(dropna, outsmpl=%firstperiod %lastperiod, name=gu_b)

model a
    !start = @dtoo(%firstperiod) + !opt_lag + 1
    %start = @otod(!start)
    smpl %start %lastperiod
    if !opt_p = 1 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1)
    else
    if !opt_p = 2 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2)
    else
    if !opt_p = 3 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3)
    else
    if !opt_p = 4 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4)
    else
    if !opt_p = 5 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5)
    else
    if !opt_p = 6 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6)
    else
    if !opt_p = 7 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7)
    else
```

```
    if !opt_p = 8 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8)
    else
    if !opt_p = 9 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8) +
coef_restrict_t_y(10)*dy_t_b(-9)
    else
    if !opt_p = 10 then
        a.append dify_ty_b = coef_restrict_t_y(2)*dy_t_b(-1) + coef_restrict_t_y(3)*dy_t_b(-2) +
coef_restrict_t_y(4)*dy_t_b(-3) + coef_restrict_t_y(5)*dy_t_b(-4) + coef_restrict_t_y(6)*dy_t_b(-5) +
coef_restrict_t_y(7)*dy_t_b(-6) + coef_restrict_t_y(8)*dy_t_b(-7) + coef_restrict_t_y(9)*dy_t_b(-8) +
coef_restrict_t_y(10)*dy_t_b(-9) + coef_restrict_t_y(11)*dy_t_b(-10)
endif
endif
endif
endif
endif
endif
endif
endif
endif
endif

    if !opt_q = 1 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1)
    else
    if !opt_q = 2 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2)
    else
    if !opt_q = 3 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3)
    else
    if !opt_q = 4 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4)
    else
    if !opt_q = 5 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5)
    else
    if !opt_q = 6 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5) +
coef_restrict_t_y(!opt_p+7)*dx1(-6)
    else
    if !opt_q = 7 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5) +
coef_restrict_t_y(!opt_p+7)*dx1(-6) + coef_restrict_t_y(!opt_p+8)*dx1(-7)
    else
    if !opt_q = 8 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5) +
coef_restrict_t_y(!opt_p+7)*dx1(-6) + coef_restrict_t_y(!opt_p+8)*dx1(-7) + coef_restrict_t_y(!opt_p+9)*dx1(-8)
    else
if !opt_q = 9 then
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5) +
coef_restrict_t_y(!opt_p+7)*dx1(-6) + coef_restrict_t_y(!opt_p+8)*dx1(-7) + coef_restrict_t_y(!opt_p+9)*dx1(-8) +
coef_restrict_t_y(!opt_p+10)*dx1(-9)
    else
    if !opt_q = 10 then
```

```
        a.append dify_tx_b = coef_restrict_t_y(!opt_p+2)*dx1(-1) + coef_restrict_t_y(!opt_p+3)*dx1(-2) +
coef_restrict_t_y(!opt_p+4)*dx1(-3) + coef_restrict_t_y(!opt_p+5)*dx1(-4) + coef_restrict_t_y(!opt_p+6)*dx1(-5) +
coef_restrict_t_y(!opt_p+7)*dx1(-6) + coef_restrict_t_y(!opt_p+8)*dx1(-7) + coef_restrict_t_y(!opt_p+9)*dx1(-8) +
coef_restrict_t_y(!opt_p+10)*dx1(-9) + coef_restrict_t_y(!opt_p+11)*dx1(-10)
endif
endif
endif
endif
endif
endif
endif
endif
endif
endif

a.append dy_t_b = coef_restrict_t_y(1)*x1(-1) + dify_ty_b + dify_tx_b + resids_t_y_b
a.append y_t_b = y_t_b(-1) + dy_t_b

    if !opt_p = 1 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1)
    else
    if !opt_p = 2 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2)
    else
    if !opt_p = 3 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3)
    else
    if !opt_p = 4 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4)
    else
    if !opt_p = 5 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5)
    else
    if !opt_p = 6 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6)
    else
    if !opt_p = 7 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7)
    else
    if !opt_p = 8 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8)
    else
    if !opt_p = 9 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8) +
coef_restrict_f_y(10)*dy_f_b(-9)
    else
    if !opt_p = 10 then
        a.append dify_fy_b = coef_restrict_f_y(2)*dy_f_b(-1) + coef_restrict_f_y(3)*dy_f_b(-2) +
coef_restrict_f_y(4)*dy_f_b(-3) + coef_restrict_f_y(5)*dy_f_b(-4) + coef_restrict_f_y(6)*dy_f_b(-5) +
coef_restrict_f_y(7)*dy_f_b(-6) + coef_restrict_f_y(8)*dy_f_b(-7) + coef_restrict_f_y(9)*dy_f_b(-8) +
coef_restrict_f_y(10)*dy_f_b(-9) + coef_restrict_f_y(11)*dy_f_b(-10)
endif
endif
endif
endif
endif
```

```
endif
endif
endif
endif
endif

    if !opt_q = 1 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1)
    else
    if !opt_q = 2 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2)
    else
    if !opt_q = 3 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3)
    else
    if !opt_q = 4 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4)
    else
    if !opt_q = 5 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5)
    else
    if !opt_q = 6 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5) +
coef_restrict_f_y(!opt_p+7)*dx1(-6)
    else
    if !opt_q = 7 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5) +
coef_restrict_f_y(!opt_p+7)*dx1(-6) + coef_restrict_f_y(!opt_p+8)*dx1(-7)
    else
    if !opt_q = 8 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5) +
coef_restrict_f_y(!opt_p+7)*dx1(-6) + coef_restrict_f_y(!opt_p+8)*dx1(-7) + coef_restrict_f_y(!opt_p+9)*dx1(-8)
    else
    if !opt_q = 9 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5) +
coef_restrict_f_y(!opt_p+7)*dx1(-6) + coef_restrict_f_y(!opt_p+8)*dx1(-7) + coef_restrict_f_y(!opt_p+9)*dx1(-8) +
coef_restrict_f_y(!opt_p+10)*dx1(-9)
    else
    if !opt_q = 10 then
        a.append dify_fx_b = coef_restrict_f_y(!opt_p+2)*dx1(-1) + coef_restrict_f_y(!opt_p+3)*dx1(-2) +
coef_restrict_f_y(!opt_p+4)*dx1(-3) + coef_restrict_f_y(!opt_p+5)*dx1(-4) + coef_restrict_f_y(!opt_p+6)*dx1(-5) +
coef_restrict_f_y(!opt_p+7)*dx1(-6) + coef_restrict_f_y(!opt_p+8)*dx1(-7) + coef_restrict_f_y(!opt_p+9)*dx1(-8) +
coef_restrict_f_y(!opt_p+10)*dx1(-9) + coef_restrict_f_y(!opt_p+11)*dx1(-10)
endif
endif
endif
endif
endif
endif
endif
endif
endif
endif

a.append dy_f_b = coef_restrict_f_y(1)*y_f_b(-1) + dify_fy_b + dify_fx_b + resids_f_y_b
a.append y_f_b = y_f_b(-1) + dy_f_b

a.scenario "actuals"
a.solve
```

```
    smpl %firstperiod %lastperiod
    equation bootstrap_t_y.ls dy_t_b y_t_b(-1) x1(-1) dy_t_b(-1 to -!opt_p) dx1(-1 to -!opt_q)
    scalar tstat_b = bootstrap_t_y.@tstat(1)

    tstat_b_dist(!ib) = tstat_b

    smpl %firstperiod %lastperiod
    equation bootstrap_f_y.ls dy_f_b y_f_b(-1) x1(-1) dy_f_b(-1 to -!opt_p) dx1(-1 to -!opt_q)
    freeze(mode=overwrite,  fstat_x_b_table) bootstrap_f_y.wald c(2)=0
    scalar fstat_b = @val(fstat_x_b_table(7,2))

    fstat_b_dist(!ib) = fstat_b
    next
endif
endif
endif
endif
```