Reverse Engineering the Eview Code in the FRB/US Model

Gary Young

June 29, 2016

Contents

Chapter 1

Introduction

From the Federal Reserve Board of Governors FRB/US model packages website page:

> FRB/US Model
>
> - About
> - Documentation and Research Papers
> - Technical Q&As
>
> Related Information
>
> - FEDS Note: The FRB/US Model: A Tool for Macroeconomic Policy Analysis
> - FEDS Note: November 2014 Update of the FRB/US Model
> - FEDS Note: Optimal-Control Monetary Policy in the FRB/US Model

> FRB/US model packages
>
> The main FRB/US model package is a self-contained set of equations, data, programs and documentation that enables various types of simulations and provides information about the model's structure. The package contains the following files and subdirectories:
>
> - A readme file with basic information about the organization of the package and how to get started;

- The subdirectory addins with EViews add-in commands that must be installed, following the instructions provided in the readme file;
- The subdirectory data containing the EViews database used in simulating the model as described in the readme file;
- The subdirectory mods containing model equations and parameter values;
- The subdirectory subs containing two libraries of subroutines, including the library mce_solve used for solving FRB/US under model-consistent (rational) expectations in EViews; the RE Solver Package is available for those wishing to download the mce_solve software and documentation by itself;
- The subdirectory programs containing some basic applications of the model including those described in the article "The FRB/US Model: A Tool for Macroeconomic Policy Analysis"
- The subdirectory documentation among whose documents are one with basic instructions for simulating FRB/US and another with a description of all of the model's equations.

FRB/US model package (ZIP) (Updated database: March 17, 2016)

The following zip file contains only the historical FRB/US database (csv format) and variable list.

FRB/US dataset and variable listing (ZIP) (Updated database: March 17, 2016)

Supply-side model

The program frbus_supply.prg estimates a multivariate state-space model that forms the basis of the FRB/US specification of potential output and its components. The zip file below contains the code and the latest data set used in estimation. The zip file also includes a note (latent_note.pdf) that provides documentation of the state-space model. The model is closely related to one presented in Charles Fleischman and John Roberts, "From Many Series, One Cycle: Improved Estimates of the Business Cycle from a Multivariate Unobserved Components Model," FEDS

Working Paper 2011-46.

FRB/US supply-side package (ZIP) (Updated database: December 30, 2015)

RE Solver Package

The software library mce_solve provides code for the solution of linear and non-linear models under model-consistent (MC) or rational (RE) expectations in EViews. This code is more reliable and efficient than the RE algorithm built into EViews (Fair-Taylor) at solving FRB/US when any of its expectations are MC. The mce_solve library includes two RE algorithms: E-Newton and E-QNewton. These algorithms iterate to find a model's RE solution with a sequence of updates to either exogenous estimates of the model's future-dated endogenous variables or exogenous components of such estimates. For single simulations of linear RE models, E-Newton is likely to be faster for models of small-to-medium size and E-QNewton is likely to be faster for larger models. For nonlinear models, E-Newton tends to be penalized relative to E-QNewton. The E-Newton algorithm has a substantial advantage over E-QNewton on experiments that involve a large number of RE solutions, as long as the same expectations Jacobian can be used for each E-Newton solution.

The solution algorithms are described in detail in Flint Brayton, "Two Practical Algorithms for Solving Rational Expectations Models, FEDS Working Paper 2011-44, and in the documentation included in the zip file. The zip file also contains some sample programs that illustrate how to use the algorithms.

mce_solve (ZIP)

NOTE: The programs for simulating the FRB/US model are written for use with the software EViews, available at www.EViews.com. The current version of FRB/US is compatible with EViews versions 7 and 8.

Model disclaimer

Last update: Apr 1, 2016

Home — Economic Research & Data

Chapter 2

FRB/US Package

2.1 example1 program

5 ⟨*frbus example1* 5⟩≡ (195c)

```
' Program for simple simulation under VAR expectations
'
' See FRB/US Simulation Basics document for information about
' this program

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2030q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model name and location
  %varmod = "stdver"
  %varpath = "../mods/"

' Input datbase
  %dbin  = "../data/longbase"

' Simulation range
  %simstart = "2020q1"
```

```
%simend   = "2025q4"


' ***************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ***************************************************************

' Load equations and coefficients
ld_frbus_eqs(modelname=%varmod,modelpath=%varpath)
ld_frbus_cfs(modelname=%varmod,modelpath=%varpath)

' Load data
dbopen %dbin as longbase
fetch(d=longbase) *

' Set monetary policy rule
smpl @all
call set_mp("dmpintay")

' Turn off zero bound and policy thresholds; hold policymaker's
' perceived equilibrium real interest rate constant
smpl @all
dmptrsh = 0
rffmin = -9999
drstar = 0

' Set fiscal policy
smpl @all
call set_fp("dfpsrp")

' Set _aerr variables to zero
smpl @all
{%varmod}.makegroup(a,n) endog @endog
call groupnew("endog","_aerr")
call group2zero("endog_aerr")

' Standard solution options
{%varmod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
%suftrk = "_0"
smpl %simstart %simend
{%varmod}.addassign @all
{%varmod}.addinit(v=n) @all
{%varmod}.scenario(n,a={%suftrk}) "track"
```

```
    {%varmod}.solve
    scalar mm = @max(@abs(xgap{%suftrk}-xgap))
    if mm > .0001 then
      statusline dynamic tracking simulation failed for {%varmod}
      stop
      endif

  ' ************************************************************
  ' Simulate a shock to monetary policy rule
  ' ************************************************************

    %sufsim = "_1"
    {%varmod}.scenario(n,a={%sufsim}) "sim"

    smpl %simstart %simstart
    rffintay_aerr = rffintay_aerr + 1

    smpl %simstart %simend
    {%varmod}.solve


  '************************************************************
  ' Make a graph
  '************************************************************

    smpl %simstart %simend
    series zero = 0
    series d_rff = rff{%sufsim} - rff
    series d_rg10 = rg10{%sufsim} - rg10
    series d_lur = lur{%sufsim} - lur
    series d_pic4 = pic4{%sufsim} - pic4

    graph fig1a.line zero d_rff
    fig1a.addtext(t,just(c),font("arial",12)) Federal Funds Rate
    fig1a.legend -display

    graph fig1b.line zero d_rg10
    fig1b.addtext(t,just(c),font("arial",12)) 10-Year Treasury Yield
    fig1b.legend -display

    graph fig1c.line zero d_lur
    fig1c.addtext(t,just(c),font("arial",12)) Unemployment Rate
    fig1c.legend -display

    graph fig1d.line zero d_pic4
    fig1d.addtext(t,just(c),font("arial",12)) Inflation Rate (4-Quarter)
```

```
fig1d.legend -display

graph fig1.merge fig1a fig1b fig1c fig1d
fig1.addtext(t,just(c),font("Arial",16)) Macroeconomic Effects of Funds Rate Pertu
fig1.align(2,1,1.25)
show fig1
```

Uses dbopen 217a, dfpsrp 219, dmpintay 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnew 92, ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, longbase 217b, lur 219, master_library 209a, pic4 219, rff 219, rffmin 219, rg10 219, set_fp 94, set_mp 95, solve 217a, wfcreate 217a, and xgap 219.

2.2 example2 program

9 ⟨*frbus example2* 9⟩≡ (196a)

```
' Program for simple simulation with MCE expectations
'
' The switch variables %mcvars_wp and %mcvars_all control whether
' the assumption of MC expectations extends beyond the financial
' sector

' See the Simulation Basics document for information about
' this program


' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library
  include ../subs/mce_solve_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model names and locations
  %varmod = "stdver"
  %varpath = "../mods/"
  %mcemod = "pfver"
  %mcepath = "../mods/"

' Input datbase
  %dbin  = "../data/longbase"

' Simulation range
  %simstart = "2020q1"
  %simend   = "2069q4"


' ************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ************************************************************
```

```
' Specify MC expectations variables
%mcvars_wp = "yes"
%mcvars_all = "no"

' MCE asset pricing
%zvars = "zdivgr zgap05 zgap10 zgap30 zrff5 zrff10 zrff30 zpi10 zpi10f zpic30 zpib5

' MCE elsewhere
if %mcvars_wp = "yes" and %mcvars_all = "no" then
  %zvars = %zvars + "zpicxfe zpieci "
  endif
if %mcvars_all = "yes" then
  %zvars = %zvars + "zpicxfe zpieci "
  %zvars = %zvars + "zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvpi zvps zxbd zxbi zxbs z
  endif

' Load equations and coefficients
call mce_load_frbus("mce_vars=%zvars,mod_b=%varmod,path_b=%varpath,mod_f=%mcemod,pa

' Load data
dbopen %dbin as longdata
fetch(d=longdata) *

' Data for extra variables associated with MC expectations
smpl @all
call make_frbus_mcevars(%zvars)

' Set monetary policy
smpl @all
call set_mp("dmpintay")

' Turn off zero bound and policy thresholds; hold policymaker's
' perceived equilibrium real interest rate constant
smpl @all
dmptrsh = 0
rffmin = -9999
drstar = 0

' Set fiscal policy
smpl @all
call set_fp("dfpsrp")

' Set _aerr variables to zero
smpl @all
{%varmod}.makegroup(a,n) endog @endog
call groupnew("endog","_aerr")
```

```
  call group2zero("endog_aerr")

' Standard solution options
  {%varmod}.solveopt(o=b,g=12,z=1e-12)
  {%mcemod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
  %suftrk = "_0"
  smpl %simstart %simend
  {%varmod}.addassign @all
  {%varmod}.addinit(v=n) @all
  {%varmod}.scenario(n,a={%suftrk}) "track"
  {%varmod}.solve
  scalar mm = @max(@abs(xgap{%suftrk}-xgap))
  if mm > .0001 then
    statusline dynamic tracking simulation failed for {%varmod}
    stop
    endif
  {%mcemod}.addassign @all
  {%mcemod}.addinit(v=n) @all

' ************************************************************
' Simulate a monetary policy shock
' ************************************************************

  %sufsim = "_1"
  {%varmod}.scenario(n,a={%sufsim}) "sim"
  {%mcemod}.scenario(n,a={%sufsim}) "sim"

  smpl %simstart %simstart
  rffintay_a = rffintay_a + 1

  %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
  %algstr = "meth=qnewton"
  %simstr = "type=single"
  smpl %simstart %simend
  call mce_run(%modstr,%algstr,%simstr)


'************************************************************
' Make a graph
'************************************************************

  smpl %simstart %simstart + 39
  series zero = 0
  series d_rff = rff{%sufsim} - rff
```

```
series d_rg10 = rg10{%sufsim} - rg10
series d_lur = lur{%sufsim} - lur
series d_pic4 = pic4{%sufsim} - pic4

graph fig1a.line zero d_rff
fig1a.addtext(t,just(c),font("arial",12)) Federal Funds Rate
fig1a.legend -display

graph fig1b.line zero d_rg10
fig1b.addtext(t,just(c),font("arial",12)) 10-Year Treasury Yield
fig1b.legend -display

graph fig1c.line zero d_lur
fig1c.addtext(t,just(c),font("arial",12)) Unemployment Rate
fig1c.legend -display

graph fig1d.line zero d_pic4
fig1d.addtext(t,just(c),font("arial",12)) Inflation Rate (4-Quarter)
fig1d.legend -display

graph fig1.merge fig1a fig1b fig1c fig1d
%title = " Macroeconomic Effects of Funds Rate Shock\r"
if %mcvars_wp = "no" and %mcvars_all = "no" then
  %title = %title + "(MC Expectations in Asset Pricing)"
  endif
if %mcvars_wp = "yes" and %mcvars_all = "no" then
  %title = %title + "(MC Expectations in Asset Pricing and Price-Wage Setting)"
  endif
if %mcvars_all = "yes" then
  %title = %title + "(MC Expectations in All Sectors)"
  endif
fig1.addtext(t,just(c),font("Arial",16)) {%title}
fig1.align(2,1,1.25)
show fig1
```

Uses dbopen 217a, dfpsrp 219, dmpintay 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnew 92, longbase 217b, lur 219, make_frbus_mcevars 175, master_library 209a, mce_load_frbus 171, mce_run 99, mce_solve_library 210 215b, pic4 219, rff 219, rffmin 219, rg10 219, sector 218, set_fp 94, set_mp 95, solve 217a, wfcreate 217a, xgap 219, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpic58 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zxbd 219, zxbi 219, zxbs 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

2.3 example3 program

13 ⟨*frbus example3* 13⟩≡ (196b)

```
' Program for simulation under VAR expectations that illustrates how
' to set the monetary policy options that impose the zero lower bound
' on the funds rate and delay the liftoff of the funds rate from the
' ZLB until either the unemployment rate falls below a threshold or
' inflation rises above a threshold.
'
' See FRB/US Simulation Basics document for general information about
' this program.

' Additional notes:

' 1. The scenario involves a set of negative aggregate demand
' shocks and a positive risk premium shock that start in 2003q3,
' when the baseline (historical) funds rate is about one percent.
' The shocks are equal to the equation errors actually observed
' in the four quarters starting in 2008q4.

' 2. To impose the ZLB set %zb = "yes" (rather than "no")

' 3. To impose the policy liftfoff threshold conditions set both
' %zb = "yes" and %threshold = "yes".  For illustrative purposes
' and reflecting the baseline conditions in 2003 and the years
' that immediately follow, the inflation threshold is set to 3.0
' and the unemployment threshold is set to 7.0, subject to the
' the adjustments described next.

' 4. Because the threshold conditions only make sense once the ZLB is
' binding, unemployment is above its threshold level (lurtrsh),
' and inflation is below its threshold (pitrsh), which is not the
' case in the initial simulation quarters, the program turns on the
' threshold code (using dmptrsh) in the 5th simulation quarter,
' at which point these conditions hold. In addition, for the threshold
' code to work properly, the endogenous switch variable dmptr must be
' zero in the quarter prior to the quarter in which the threshold code is
' turned on.  This is accomplished by setting the baseline data on dmptr
' to zero and by setting the unemployment and inflation thresholds
' (lurtrsh, pitrsh) to values in the first four simulation quarters that
' would not flip the dmptr switch to one.

' 4. Choose one of the five available policy rules by setting
' %policy to one of rffintay, rfftay, rfftlr, rffalt, or rffgen.

' 5. If neither the ZLB or thresholds are imposed, the monetary policy
```

```
' equations have baseline-tracking adds and the simulation is
' a standard deviations-from-baseline exercise.

' 6. If either the ZLB or thresholds are imposed, the add factors on
' monetary policy equations are set to zero after the tracking adds
' are computed so that the ZLB and threshold conditions are based on the
' actual simulated outcomes for the funds rate and inflation and unemployment,
' not their deviations from baseline.

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2012q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model name and location
  %varmod = "stdver"
  %varpath = "../mods/"

' Input datbase
  %dbin  = "../data/longbase"

' Simulation range
  %simstart = "2003q3"
  %simend   = "2008q2"

' Policy
  %zb = "yes"
  %threshold = "yes"
  %policy = "rfftay"


' ************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ************************************************************

' Load equations and coefficients
  ld_frbus_eqs(modelname=%varmod,modelpath=%varpath)
```

```
    ld_frbus_cfs(modelname=%varmod,modelpath=%varpath)

  ' Load data
    dbopen %dbin as longbase
    fetch(d=longbase) *

  ' Set monetary policy rule
    smpl @all
    %policydmp = @replace(%policy,"rff","dmp")
    call set_mp(%policydmp)

  ' Set ZLB
    if %zb = "yes" then
      rffmin = .125
      else
      rffmin = -9999
      endif

  ' Set threshold variables
    if %threshold = "yes" then
      if %zb = "no" then
        @uiprompt("When policy thresholds are imposed, the zero bound must also be imposed")
        stop
        endif
      smpl @all
      call dateshift(%simstart,%quarter4,3)
    ' thresholds (dmptrsh and dmptr) not active in first 4 qtrs
      smpl %simstart - 1 %quarter4
      dmptrsh = 0
      lurtrsh = -9999
      pitrsh = 9999
      dmptr = 0
    ' thresholds (dmptrsh and dmptr) active starting in qtr 5
      smpl %quarter4 + 1 %simend
      dmptrsh = 1
      lurtrsh = 7.0
      pitrsh = 3.0
      smpl @all
      else
      smpl @all
      dmptrsh = 0
      endif

    smpl @all
    drstar = 0
```

```
' Set fiscal policy
smpl @all
call set_fp("dfpsrp")

' Set _aerr variables to zero
smpl @all
{%varmod}.makegroup(a,n) endog @endog
call groupnew("endog","_aerr")
call group2zero("endog_aerr")

' Standard solution options
{%varmod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
%suftrk = "_0"
smpl %simstart 2012q4
{%varmod}.addassign @all
{%varmod}.addinit(v=n) @all
{%varmod}.scenario(n,a={%suftrk}) "track"
{%varmod}.solve
scalar mm = @max(@abs(xgap{%suftrk}-xgap))
if mm > .0001 then
  statusline dynamic tracking simulation failed for {%varmod}
  stop
  endif

' Set monetary policy add factors to zero when ZLB or threshold are
' imposed

if %zb = "yes" then
  smpl @all
  {%policy}_a = 0
  rffrule_a = 0
  rffe_a = 0
  if %threshold = "yes" then
    dmptpi_a = 0
    dmptlur_a = 0
    dmptmax_a = 0
    dmptr_a = 0
    endif
  endif

' ************************************************************
' Simulation
' ************************************************************
```

```
  %sufsim = "_1"
  {%varmod}.scenario(n,a={%sufsim}) "sim"

' shock values are taken from equation residuals for 2008q4-2009q3
  eco_a.fill(o=%simstart) -.006, -.006, -.011, -.001
  ecd_a.fill(o=%simstart) -.091, -.018, -.021,  .029
  eh_a.fill(o=%simstart)  -.076, -.078, -.040,  .073
  epd_a.fill(o=%simstart) -.096, -.062,  .014,  .032
  eps_a.fill(o=%simstart) -.018, -.046, -.036, -.017
  rbbbp_a.fill(o=%simstart) 2.70, 0.38, -0.89, -1.35

  smpl %simstart %simend
  {%varmod}.solve


'*************************************************************
' Make a graph
'*************************************************************

  smpl %simstart %simend

  graph fig1a.line rff rff{%sufsim}
  fig1a.addtext(t,just(c),font("arial",12)) Federal Funds Rate
  fig1a.legend -display

  graph fig1b.line rg10 rg10{%sufsim}
  fig1b.addtext(t,just(c),font("arial",12)) 10-Year Treasury Yield
  fig1b.legend -display

  graph fig1c.line lur lur{%sufsim}
  fig1c.addtext(t,just(c),font("arial",12)) Unemployment Rate
  fig1c.legend -display

  graph fig1d.line pic4 pic4{%sufsim}
  fig1d.addtext(t,just(c),font("arial",12)) Inflation Rate (4-Quarter)
  fig1d.legend -display

  %title = "Macroeconomic Effects of Negative AD Shock\r(VAR Expectations"
  %title = %title + "; Policy = " + %policy + ")"
  if %zb = "yes" and %threshold = "no" then
    %title = %title + "\r(ZLB Imposed)"
    endif
  if %zb = "yes" and %threshold = "yes" then
    %title = %title + "\r(ZLB and Thresholds Imposed)"
    endif
```

```
graph fig1.merge fig1a fig1b fig1c fig1d
fig1.addtext(t,just(c),font("Arial",16)) {%title}
fig1.addtext(b,just(c),font("Arial",16)) Blue:  Actual;  Red:  Simulated
fig1.align(2,1,1.25)
show fig1
```

Uses dateshift 89, dbopen 217a, dfpsrp 219, dmptr 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnew 92, ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, longbase 217b, lur 219, lurtrsh 219, master_library 209a, pic4 219, pitrsh 219, rff 219, rffalt 219, rffgen 219, rffintay 219, rffmin 219, rfftay 219, rfftlr 219, rg10 219, set_fp 94, set_mp 95, solve 217a, wfcreate 217a, and xgap 219.

2.4 example4 program

19 ⟨*frbus example4* 19⟩≡ (196c)

```
' This MCE example program illustrates:
'
' 1. how to use a monetary policy rule that is not one of the policy
'   alternatives included in FRB/US;
' 2. how to add new MCE expectations variables;
' 3. how to drop one of the regular FRB/US equations as part of the process
'   of loading the model

' Most of the code needed illustrate these issues is located between
' the "start of new code" and "end of new code" comments below

' The switch variables %mcvars_wp and %mcvars_all control whether
' the assumption of MC expectations extends beyond the financial
' sector

' See the Simulation Basics document for information about
' this program


' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library
  include ../subs/mce_solve_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model names and locations
  %varmod = "stdver"
  %varpath = "../mods/"
  %mcemod = "pfver"
  %mcepath = "../mods/"

' Input datbase
  %dbin  = "../data/longbase"

' Simulation range
```

```
%simstart = "2010q1"
%simend   = "2069q4"


' ******************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ******************************************************************

' Specify MC expectations variables
%mcvars_wp = "no"
%mcvars_all = "yes"

' MCE asset pricing
%zvars = "zdivgr zgap05 zgap10 zgap30 zrff5 zrff10 zrff30 zpi10 zpi10f zpic30 zpib5

' MCE elsewhere
if %mcvars_wp = "yes" and %mcvars_all = "no" then
  %zvars = %zvars + "zpicxfe zpieci "
  endif
if %mcvars_all = "yes" then
  %zvars = %zvars + "zpicxfe zpieci "
  %zvars = %zvars + "zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvpi zvps zxbd zxbi zxbs z
  endif

' Load equations and coefficients

' drop one of the FRB/US monetary policy rule equations (rffgen) so that it can be
' replaced below with an alternative rule
%allbut = "rffgen"
call mce_load_frbus("mce_vars=%zvars,mod_b=%varmod,path_b=%varpath,mod_f=%mcemod,pa

' Load data
dbopen %dbin as longdata
fetch(d=longdata) *

' ******************************************************************************
' ******************************************************************************
' start of new code (aside from the change above to mce_load_frbus and
'                    the change below to the call to set_mp)

' Code a first-difference interest rate rule as rffgen.  The first-difference rule de
' the expected output gap three quarters ahead (zgap3) and on expected 4-qtr inflatio
' quarters ahead (zpic43).  The name of each new expectation must start with a "z".
{%varmod}.append rffgen-rffgen_aerr = rffe(-1) + .5*(zpic43-pitarg) + .5*(zgap3-zga
```

```
' Add the MCE definitions of zgap3 and zpic43 to the forward-looking model,
' noting that the MCE names of these variables must start with a "w" rather than a "z".
  {%mcemod}.append wgap3-wgap3_aerr = xgap2(3)
  {%mcemod}.append wpic43-wpic43_aerr = picx4(3)
' Add expectations error equations to the MCE model
  {%mcemod}.append ezgap3 = zgap3-wgap3
  {%mcemod}.append ezpic43 = zpic43-wpic43

' Add to the backward-looking model simple equations for the new expectations variables.
' Technically, these equations should be the appropriate VAR expectations formulas, but
' because in this program these expectations will always be MCE, the form of their
' backward-looking identities is not very important.
  {%varmod}.append zgap3-zgap3_aerr = .5*xgap2(-1)
  {%varmod}.append zpic43-zpic43_aerr = .5*picx4(-1)+.5*ptr(-1)

' Add the new MCE variables to the %zvars string
  %zvars = %zvars + " zgap3 zpic43"

' Define baseline values of the new expectations variables
  smpl @all
  series zgap3 = xgap2(3)
  series zpic43 = picx4(3)

' Make sure that the baseline data for rffgen matches the baseline data for rffe
  rffgen = rffe

' end of new code
' ***************************************************************************
' ***************************************************************************

' Data for extra variables associated with MC expectations
  smpl @all
  call make_frbus_mcevars(%zvars)

' Set monetary policy to use the first-difference policy rule (coded as rffgen)
  smpl @all
  call set_mp("dmpgen")

' Turn off zero bound and policy thresholds; hold policymaker's
' perceived equilibrium real interest rate constant
  smpl @all
  dmptrsh = 0
  rffmin = -9999
  drstar = 0

' Set fiscal policy
```

```
  smpl @all
  call set_fp("dfpsrp")

' Set _aerr variables to zero
  smpl @all
  {%varmod}.makegroup(a,n) endog @endog
  call groupnew("endog","_aerr")
  call group2zero("endog_aerr")

' Standard solution options
  {%varmod}.solveopt(o=b,g=12,z=1e-12)
  {%mcemod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
  %suftrk = "_0"
  smpl %simstart %simend
  {%varmod}.addassign @all
  {%varmod}.addinit(v=n) @all
  {%varmod}.scenario(n,a={%suftrk}) "track"
  {%varmod}.solve
  scalar mm = @max(@abs(xgap{%suftrk}-xgap))
  if mm > .0001 then
    statusline dynamic tracking simulation failed for {%varmod}
    stop
    endif
  {%mcemod}.addassign @all
  {%mcemod}.addinit(v=n) @all

' ************************************************************
' Simulate the effects of a one-percent consumption shock
' ************************************************************

  %sufsim = "_1"
  {%varmod}.scenario(n,a={%sufsim}) "sim"
  {%mcemod}.scenario(n,a={%sufsim}) "sim"

  smpl %simstart %simstart
  eco_a = eco_a + .01

  %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
  %algstr = "meth=qnewton"
  %simstr = "type=single"
  smpl %simstart %simend
  call mce_run(%modstr,%algstr,%simstr)
```

```
'*************************************************************
' Make a graph
'*************************************************************

  smpl %simstart %simstart + 39
  series zero = 0
  series d_rff = rff{%sufsim} - rff
  series d_rg10 = rg10{%sufsim} - rg10
  series d_lur = lur{%sufsim} - lur
  series d_pic4 = pic4{%sufsim} - pic4

  graph fig1a.line zero d_rff
  fig1a.addtext(t,just(c),font("arial",12)) Federal Funds Rate
  fig1a.legend -display

  graph fig1b.line zero d_rg10
  fig1b.addtext(t,just(c),font("arial",12)) 10-Year Treasury Yield
  fig1b.legend -display

  graph fig1c.line zero d_lur
  fig1c.addtext(t,just(c),font("arial",12)) Unemployment Rate
  fig1c.legend -display

  graph fig1d.line zero d_pic4
  fig1d.addtext(t,just(c),font("arial",12)) Inflation Rate (4-Quarter)
  fig1d.legend -display

  graph fig1.merge fig1a fig1b fig1c fig1d
  %title = " Macroeconomic Effects of a Shock to Consumption\r"
  if %mcvars_wp = "no" and %mcvars_all = "no" then
    %title = %title + "(MC Expectations in Asset Pricing)"
    endif
  if %mcvars_wp = "yes" and %mcvars_all = "no" then
    %title = %title + "(MC Expectations in Asset Pricing and Price-Wage Setting)"
    endif
  if %mcvars_all = "yes" then
    %title = %title + "(MC Expectations in All Sectors)"
    endif
  fig1.addtext(t,just(c),font("Arial",16)) {%title}
  fig1.align(2,1,1.25)
  show fig1
```

Uses dbopen 217a, dfpsrp 219, dmpgen 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnew 92, longbase 217b, lur 219, make_frbus_mcevars 175, master_library 209a, mce_load_frbus 171, mce_run 99, mce_solve_library 210 215b,

2.5 ocpolicy program

25 ⟨*frbus ocpolicy* 25⟩≡ (196d)

```
' Routine to simulate how the SEP baseline forecast would change if
' policymakers commit to a path for the federal funds rate that is
' determined by optimal-control (OC) techniques to minimize a
' quadratic loss function.
'
' Detailed information on the mechanics of the OC algorithm and the
' various required and optional parameters that set up and guide its
' execution is available in the MCE Solve Users Guide in the
' documentation directory.  Most relevant is the part of section 5
' that describes the "opt" simulation type as well as table 7.

' As specified below, the loss function penalizes equally weighted
' squared deviations of the unemployment rate from the natural rate,
' squared deviations of inflation from a 2 percent, and squared
' quarterly changes in the funds rate.

' In the SEP baseline, agents with model-consistent (MC) expectations
' are initially assumed to project that the funds rate will follow the
' baseline path and to set their baseline expectations
' accordingly.  At the start of the optimal control simulation, however,
' these agents immediately and fully revise their expectations to be
' consistent with the revision to the funds rate path that occurs under
' optimal control -- that is, agents have rational expectations and
' announced policy actions are completely credible.
'
' The experiment can be run with the zero lower bound (ZLB) imposed
' (%zerobound = "yes") or not imposed (%zerobound = "no").  When
' the ZLB is imposed, a penalty term is added to the loss function.

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library
  include ../subs/mce_solve_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}
```

```
' FRB/US model names and locations
 %varmod = "stdver"
 %varpath = "../mods/"
 %mcemod = "pfver"
 %mcepath = "../mods/"

' Input datbase
 %dbin  = "../data/longbase"

' Simulation range
 %simstart = "2014q4"
 %simend   = "2070q4"

' Primary loss function parameters:  The value of the policy instrument
' is chosen optimally from %drvstart to %drvend (60 qtrs) to minimize
' the loss function from %evlstart to %evlend (80 qtrs).  The three
' arguments of the period loss function are weighted by the the
' weight parameters and over time losses are discounted at the rate
' %discount
 %evlstart = %simstart
 %drvstart = %simstart
 call dateshift(%evlstart,%evlend,79)
 call dateshift(%drvstart,%drvend,59)
 %discount = ".99"
 %u_weight = "1.0"
 %p_weight = "1.0"
 %r_weight = "1.0"

' Optionally impose the zero lower bound
 %zerobound = "yes"


' ***************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ***************************************************************

' Specify MC expectations variables
 %mcvars_wp = "yes"
 %mcvars_all = "no"

' MCE asset pricing
 %zvars = "zdivgr zgap05 zgap10 zgap30 zrff5 zrff10 zrff30 zpi10 zpi10f zpic30 zpib5

' MCE elsewhere
 if %mcvars_wp = "yes" and %mcvars_all = "no" then
```

```
    %zvars = %zvars + "zpicxfe zpieci "
    endif
  if %mcvars_all = "yes" then
    %zvars = %zvars + "zpicxfe zpieci "
    %zvars = %zvars + "zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvpi zvps zxbd zxnfbi zxnfbs zyh zyh
    endif

' Load equations and coefficients
call mce_load_frbus("mce_vars=%zvars,mod_b=%varmod,path_b=%varpath,mod_f=%mcemod,path_f=%mcep

' Add a ugap equation
{%varmod}.append ugap - ugap_aerr = lur - lurnat

' Load data
dbopen %dbin as longdata
fetch(d=longdata) *

' Define SEP-consistent ustar and ugap series; this step is needed because
' the baseline value of lurnat may not be fully SEP-consistent in the
' short-to-medium ruun
smpl @all
series ustar = lurnat
smpl %simstart 2025q4
ustar = 5.35
smpl @all
series ugap = lur-ustar
series ugap_aerr = 0

' Data for extra variables associated with MC expectations
smpl @all
call make_frbus_mcevars(%zvars)

' Set monetary policy option (the residual on the equation of
' the chosen option is the OC policy instrument)
smpl @all
call set_mp("dmptay")

' Initially turn off zero lower bound; if %zerobound = "yes", it will be
' imposed below by adding a penalty term to the loss function.
smpl @all
rffmin = -9999
' Turn off policy thresholds
dmptrsh = 0
' Let the perceived equilibrium real interest rate vary
drstar = 1
```

```
' Set fiscal policy so that it is exogenous for first 20 qtrs and then
' turns on debt targeting rule
  smpl %simstart %simstart + 19
  call set_fp("dfpex")
  smpl %simstart + 20 %simend
  call set_fp("dfpdbt")

' Set _aerr variables to zero
  smpl @all
  {%varmod}.makegroup(a,n) endog @endog
  call groupnew("endog","_aerr")
  call group2zero("endog_aerr")

' Standard solution options
  {%varmod}.solveopt(o=b,g=12,z=1e-12)
  {%mcemod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
  %suftrk = "_0"
  smpl %simstart %simend
  {%varmod}.addassign @all
  {%varmod}.addinit(v=n) @all
  {%varmod}.scenario(n,a={%suftrk}) "track"
  {%varmod}.solve
  scalar mm = @max(@abs(xgap{%suftrk}-xgap))
  if mm > .0001 then
    statusline dynamic tracking simulation failed for {%varmod}
    stop
    endif
  {%mcemod}.addassign @all
  {%mcemod}.addinit(v=n) @all



'*************************************************************
' optimal policy setup
'*************************************************************

' The policy instrument is a time varying constant in the equation
' for the selected policy rule
  group opt_instrus rfftay_aerr

' Loss function variables (unemployment gap, 4-qtr PCE inflation,
' and the first difference of the federal funds rate)
  group opt_targs ugap pic4 delrff
```

```
' The desired paths of the loss function variables are specified in
' series with "_t" suffix.
  smpl @all
  series ugap_t   = 0
  series delrff_t = 0
  series pic4_t   = 2.0

' The weights on the loss function arguments are specified in
' series with "_w" suffix.
  series ugap_w   = @val(%u_weight)
  series pic4_w   =  @val(%p_weight)
  series delrff_w =  @val(%r_weight)

  !discount = @val(%discount)
  smpl %simstart+1 %simend
  ugap_w   = !discount * ugap_w(-1)
  pic4_w   = !discount * pic4_w(-1)
  delrff_w = !discount * delrff_w(-1)

' Zero bound penalty function
  if %zerobound = "yes" then
    {%varmod}.append penalty - penalty_aerr = _
       @recode(rff<(rff_lo_bnd+rff_lo_shift), _
         3.0*((rff_lo_bnd+rff_lo_shift)-rff),  _
         0) _
     + @recode(rff<(rff_lo_bnd+rff_lo_shift), _
       .10*exp(10*(rff-(rff_lo_bnd+rff_lo_shift))), _
       .10*exp(-20*(rff-(rff_lo_bnd+rff_lo_shift))))
    smpl @all
    series rff_lo_bnd = .125
    series rff_lo_shift = .00
    series penalty = 0
    series penalty_aerr = 0
    series rffmin = -9999
    %penalty_weight = "10.0"
    opt_targs.add penalty
    smpl @all
    series penalty_a = 0
    series penalty_t = 0
    series penalty_w = @val(%penalty_weight)
    smpl %simstart+1 %simend
    penalty_w = !discount * penalty_w(-1)
    endif
```

```
'*************************************************************
' optimal policy simulation
'*************************************************************

  %sufcontrol = "_1"

' In %simstr, the "type=opt" string designates a commitment-based
' OC simulation.  The required "instrus" and "targs" keywords point to
' the groups containing the policy instrument(s) and target variables.
' FRB/US simulations of this type generally run much more quickly with
' the newton MCE algorithm than they do with qnewton.
  %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
  %algstr = "jinit=interp(4), meth=newton"
  %simstr = "type=opt,instrus=opt_instrus,targs=opt_targs"
  %simstr = %simstr + ",scen,suf=" + %sufcontrol+ ",solveopt=%sopt"
  %simstr = %simstr + ",lend=" + %evlend + ",iend=" + %drvend + ",lmax=20"
  smpl {%simstart} {%simend}
  call mce_run(%modstr,%algstr,%simstr)

' When the ZLB is imposed, run the OC algorithm a second time,
' after adjusting the intercept of the penalty function,
' to hit ZLB more closely
  if %zerobound = "yes" then
    smpl if rff{%sufcontrol} <.2
    rff_lo_shift = .125 - rff{%sufcontrol}
    %modstr = ""
    %algstr = ""
    %simstr = "type=opt,instrus=opt_instrus,targs=opt_targs"
    %simstr = %simstr + ",solveopt=%sopt"
    %simstr = %simstr + ",lend=" + %evlend + ",iend=" + %drvend + ",lmax=20"
    smpl {%simstart} {%simend}
    call mce_run(%modstr,%algstr,%simstr)
    endif


'*************************************************************
' graph results
'*************************************************************

  call dateshift(%simstart,%graphstart,-8)
  call dateshift(%simstart,%graphend,32)

  smpl %graphstart %graphend
  graph fig1a.line rff{%sufcontrol} rff
  fig1a.options size(7,4.2)
  fig1a.legend display -inbox position(3.8,2.8) font("arial",15)
```

```
fig1a.datelabel format(yy)
fig1a.addtext(6.4,-.30,font("arial",13),keep) percent
fig1a.axis(left) font("arial",15)
fig1a.axis(bottom) font("arial",15)
fig1a.setelem(1) lcolor(red)  legend("optimal control") lwidth(2)
fig1a.setelem(2) lcolor(black) legend("SEP-consistent baseline") lwidth(2)
fig1a.addtext(t,just(c),font("arial",18)) Federal Funds Rate


smpl %graphstart %graphend
graph fig1b.line rg10{%sufcontrol} rg10
fig1b.options size(7,4.2)
fig1b.legend display -inbox position(3.8,2.8) font("arial",15)
fig1b.datelabel format(yy)
fig1b.addtext(6.4,-.30,font("arial",13),keep) percent
fig1b.axis(left) font("arial",15)
fig1b.axis(bottom) font("arial",15)
fig1b.setelem(1) lcolor(red)  legend("optimal control") lwidth(2)
fig1b.setelem(2) lcolor(black) legend("SEP-consistent baseline") lwidth(2)
fig1b.addtext(t,just(c),font("arial",18)) 10-Year Treasury Yield


smpl %graphstart %graphend
graph fig1c.line lur{%sufcontrol} lur
fig1c.options size(7,4.2)
fig1c.legend display -inbox position(3.9,0.3) font("arial",15)
fig1c.datelabel format(yy)
fig1c.addtext(6.4,-.30,font("arial",13),keep) percent
fig1c.axis(left) font("arial",15)
fig1c.axis(bottom) font("arial",15)
fig1c.setelem(1) lcolor(red)  legend("optimal control") lwidth(2)
fig1c.setelem(2) lcolor(black) legend("SEP-consistent baseline") lwidth(2)
fig1c.addtext(t,just(c),font("arial",18)) Unemployment Rate


smpl %graphstart %graphend
graph fig1d.line pic4{%sufcontrol} pic4
fig1d.options size(7,4.2)
fig1d.legend display -inbox position(0.5,0.2) font("arial",15)
fig1d.datelabel format(yy)
fig1d.addtext(6.4,-.30,font("arial",13),keep) percent
fig1d.axis(left) font("arial",15)
fig1d.axis(bottom) font("arial",15)
fig1d.setelem(1) lcolor(red)  legend("optimal control") lwidth(2)
fig1d.setelem(2) lcolor(black) legend("SEP-consistent baseline") lwidth(2)
fig1d.addtext(t,just(c),font("arial",18)) PCE Inflation Rate (4-Quarter)
```

```
graph fig1.merge fig1a fig1b fig1c fig1d
if %mcvars_wp = "no" and %mcvars_all = "no" then
  %title = "Macroeconomic Effects of Optimal-Control Policy with Rational Expectati
  endif
if %mcvars_wp = "yes" and %mcvars_all = "no" then
  %title = "Macroeconomic Effects of Optimal-Control Policy\n With Rational Expecta
  endif
if %mcvars_all = "yes" then
  %title = "Macroeconomic Effects of Optimal-Control Policy With Full Rational Expe
  endif

if %zerobound = "yes" then
  %title = %title + "\rZLB Imposed"
  else
  %title = %title + "\rZLB not Imposed"
  endif

fig1.addtext(t,just(c),font("Arial",20)) {%title}
fig1.align(2,1,1.25)
show fig1
```

Uses dateshift 89, dbopen 217a, delrff 219, dfpdbt 219, dfpex 219, dmptay 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnew 92, left 219, longbase 217b, lur 219, lurnat 219, make_frbus_mcevars 175, master_library 209a, mce_load_frbus 171, mce_run 99, mce_solve_library 210 215b, pic4 219, rff 219, rffmin 219, rg10 219, set_fp 94, set_mp 95, solve 217a, wfcreate 217a, xgap 219, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpic58 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zxbd 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

2.6 pings program

33 ⟨*simulate six ping simulations, aka simple IRFs* 33⟩≡ (197a)

```
' Simulate six ping simulations (AKA simple IRFs)
'
' Notes:
'
' 1. Choose between VAR expectations and several MCE alternatives
' with the %mcevars parameter.
'
'   - %mcevars = "none"   => VAR expectations everywhere
'   - %mcevars = "mcap"   => MCE in asset pricing, VAR expectations elsewhere
'   - %mcevars = "mcapwp" => MCE in asset pricing and price-wage setting;
'        VAR expectations elsewhere
'   - %mcevars = "all"    => MCE everywhere
'
' Note that even when %mcevars = "none", the program does many
' of the setup steps for an MCE simulation even though it never
' uses what they create.

' 2. Seven of the pings are one-time shocks to the residual of an
' equation whose structure contains a large autoregressive
' element.  The remaining ping involves a permanent increase in the
' level of trend MFP.

' 3. The eight pings are:
'
'   - A 100 basis point upward shock to the rffintay monetary
'     policy rule
'   - An increase in federal purchases equal to one percent of
'     baseline GDP
'   - A one percent permanent increase in the level of trend MPF
'   - A 100 bp increase in the equity premium
'   - A $10 per barrel increase in the price of oil
'   - A 1 percent (ar) increase in the growth rate of
'     multifactor productivity
'   - Increases of 100 basis points to the 10-year Treasury term premium,
'     75 basis points to the 5-year premium, and 30 basis points to the
'     30-year premium
'   - A 10 percent increase in the (real) exchange rate

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
```

```
include ../subs/master_library
include ../subs/mce_solve_library

' Workfile
%wfstart = "1975q1"
%wfend = "2100q4"
%mainpage = "main"
wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model names and locations
%varmod = "stdver"
%varpath = "../mods/"
%mcemod = "pfver"
%mcepath = "../mods/"

' Input datbase
%dbin  = "../data/longbase"

' Simulation range
%simstart = "2020q1"
%simend   = "2069q4"

' Choose an expectations option ("none" => VAR expectations, "mcap", "mcapwp", "all")
%mcevars = "none"

' ******************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ******************************************************************

' MCE variable setup
if %mcevars = "none" then
  %zvars = "zpic58 "
  else
  %zvars = "zdivgr zgap05 zgap10 zgap30 zrff5 zrff10 zrff30 zpi10 zpi10f zpic30 zpi
  if %mcevars = "mcapwp" or %mcevars = "all" then
    %zvars = %zvars + "zpicxfe zpieci "
    endif
  if %mcevars = "all" then
    %zvars = %zvars + "zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvpi zvps zxbd zxbi zxbs
    endif
  endif

' Load equations and coefficients
call mce_load_frbus("mce_vars=%zvars,mod_b=%varmod,path_b=%varpath,mod_f=%mcemod,pa
```

```
' Load data
dbopen %dbin as longdata
fetch(d=longdata) *

' Data for extra variables associated with MC expectations
smpl @all
call make_frbus_mcevars(%zvars)

' Set monetary policy
smpl @all
call set_mp("dmpintay")

' Turn off zero bound and policy thresholds; hold policymaker's
' perceived equilibrium real interest rate constant
smpl @all
dmptrsh = 0
rffmin = -9999
drstar = 0

' Set fiscal policy
smpl @all
call set_fp("dfpsrp")

' Set _aerr variables to zero
smpl @all
{%varmod}.makegroup(a,n) endog @endog
call groupnew("endog","_aerr")
call group2zero("endog_aerr")

' Standard solution options
{%varmod}.solveopt(o=b,g=12,z=1e-12)
{%mcemod}.solveopt(o=b,g=12,z=1e-12)

' Assign baseline tracking add factors
%suftrk = "_0"
smpl %simstart %simend
{%varmod}.addassign @all
{%varmod}.addinit(v=n) @all
{%varmod}.scenario(n,a={%suftrk}) "track"
{%varmod}.solve
scalar mm = @max(@abs(xgap{%suftrk}-xgap))
if mm > .0001 then
  statusline dynamic tracking simulation failed for {%varmod}
  stop
  endif
{%mcemod}.addassign @all
```

```
  {%mcemod}.addinit(v=n) @all




' ******************************************************************
' Ping simulations
' ******************************************************************

  %suf = "_1"
  {%varmod}.scenario(n,a={%suf}) "ping"
  {%mcemod}.scenario(n,a={%suf}) "ping"

' *********************************
' Federal Funds Rate: RFF ping

  %ping = "rff"

  smpl %simstart %simstart
  rffintay_aerr = rffintay_aerr + 1
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  rffintay_aerr = rffintay_aerr - 1
  call copyit

' *********************************
' Treasury Term Premium: RG10P, RG5P, and RG30P ping

  %ping = "prem"
  smpl @all
  series rg30p_aerr = 0
  smpl %simstart %simstart
  rg10p_aerr = rg10p_aerr + 1
  rg5p_aerr = rg5p_aerr + .75
  rg30p_aerr = rg30p_aerr + .35
  smpl %simstart %simend
  if %mcevars = "none" then
```

```
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  rg10p_aerr = rg10p_aerr - 1
  rg5p_aerr = rg5p_aerr - 0.75
  rg30p_aerr = rg30p_aerr - .35
  call copyit
  smpl %simstart %simend
  series rg10p_{%ping} = rg10p{%suf} - rg10p

' *********************************
' Federal Purchases: EGFO ping

  %ping = "eg"
  smpl %simstart %simstart
  egfo_aerr = egfo_aerr + .01*xgdpn/egfon
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  egfo_aerr = egfo_aerr - .01*xgdpn/egfon
  call copyit
  smpl %simstart %simend
  series egfn_shr_{%ping} = 100*(egfn{%suf}/xgdpn{%suf} - egfn/xgdpn)

' *********************************
' Equity Premium: REQP ping

  %ping = "reqp"
  smpl %simstart %simstart
  reqp_aerr = reqp_aerr + 1
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
```

```
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  reqp_aerr = reqp_aerr - 1
  call copyit
  smpl %simstart %simend
  series reqp_{%ping} = reqp{%suf} - reqp


' *********************************
' Oil Prices: POILR ping

  %ping = "oil"
  smpl %simstart %simstart
  poilr_aerr = poilr_aerr + 10/pxb
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  poilr_aerr = poilr_aerr - 10/pxb
  call copyit
  smpl %simstart %simend
  series poil_{%ping} = poil{%suf} - poil



' *********************************
' Exchange Rate: FPXRR ping

  %ping = "exch"
  smpl %simstart %simstart
  series shock_fpxr = log(1.1)
  fpxrr_aerr = fpxrr_aerr + shock_fpxr

  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
```

```
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif

  series fpxr_{%ping} = fpxr{%suf} - fpxr
  smpl %simstart %simstart
  fpxrr_aerr = fpxrr_aerr - shock_fpxr
  call copyit


' *********************************
' HMFPT ping

  %ping = "hmfp"
  smpl %simstart %simstart
  hmfpt_aerr = hmfpt_aerr + 1
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  hmfpt_aerr = hmfpt_aerr - 1
  call copyit
  smpl %simstart %simend
  series hmfpt_{%ping} = hmfpt{%suf} - hmfpt


' *********************************
' MFPT ping

  %ping = "mfp"
  smpl %simstart %simstart
  mfpt_aerr = mfpt_aerr + .01
  smpl %simstart %simend
  if %mcevars = "none" then
    {%varmod}.solve
    else
    %modstr = "mod_b=%varmod,mod_f=%mcemod,mce_vars=%zvars"
```

```
    %algstr = "meth=qnewton"
    %simstr = "type=single,solveopt=%sopt,suf=" + %suf
    call mce_run(%modstr,%algstr,%simstr)
    endif
  smpl %simstart %simstart
  mfpt_aerr = mfpt_aerr - .01
  call copyit
  smpl %simstart %simend
  series mfpt_{%ping} = 100*(mfpt{%suf}/mfpt - 1)


' ****************************************************************
' Individual ping graphs
' ****************************************************************
  call graphit


' ****************************************************************
' Composite figures
' ****************************************************************

  if %mcevars = "none" then
    %exp = "VAR Expectations"
    endif
  if %mcevars = "mcap" then
    %exp = "MC (MCAP) Expectations"
    endif
  if %mcevars = "mcapwp" then
    %exp = "MC (MCAP+WP) Expectations"
    endif
  if %mcevars = "all" then
    %exp = "MC (ALL) Expectations"
    endif

  %t1 = "FRB/US Ping Simulations:  " + %exp + " -- I"
  %t2 = "FRB/US Ping Simulations:  " + %exp + " -- II"
  %t3 = "FRB/US Ping Simulations:  " + %exp + " -- III"

' Figure 1

  graph fig_1.merge gr_rff gr_eg gr_reqp
  fig_1.align(3,.4,1.0)
  fig_1.addtext(t,just(c),font(12)) %t1
  show fig_1
```

```
' Figure 2

graph fig_2.merge gr_oil gr_hmfp gr_mfp
fig_2.align(3,.4,1.0)
fig_2.addtext(t,just(c),font(12)) %t2
show fig_2


' Figure 3

graph fig_3.merge gr_prem gr_exch
fig_3.align(3,.4,1.0)
fig_3.addtext(t,just(c),font(12)) %t3
show fig_3
```

Uses copyit 197b, dbopen 217a, dfpsrp 219, dmpintay 219, dmptrsh 219, drstar 219, egfn 219, egfon 219, fetch 217a, fpxr 219, graphit 199, group2zero 91a, groupnew 92, hmfpt 219, longbase 217b, make_frbus_mcevars 175, master_library 209a, mce_load_frbus 171, mce_run 99, mce_solve_library 210 215b, mfpt 219, pings 197a, poil 219, pxb 219, reqp 219, rff 219, rffintay 219, rffmin 219, rg10p 219, set_fp 94, set_mp 95, solve 217a, wfcreate 217a, xgap 219, xgdpn 219, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpic58 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zxbd 219, zxbi 219, zxbs 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

2.7 stochsim program

42 ⟨*stochastic simulations under variable expectations* 42⟩≡ (207a)

```
' Program for stochastic sims under VAR expectations

' The stochastic shocks are bootstrapped from the de-meaned
' historical errors of stochastic equations.  The parameters
' %residstart and %residend declare the historical error range.
' A list of stochastic equations is extracted from the file
' pointed to by %varinfo.

' The bootstrap procedure randomly draws one historical quarter
' at a time when the parameter %errorblock = 1; alternatively, if
' %errorblock = 2, then the procedure would randomly draw two
' successive quarters at a time.

' The stochastic replications are simulated in a simple loop,
' rather than using the built-in EViews stochastic simulation
' procedure, so that shocks in the first simulation quarter can
' be scaled down by the parameter %q1_shock_damp.  This feature
' is useful when uncertainty about the first simulation quarter
' in real-time analysis by known information.  The shocks are
' not rescaled when %q1_shock_damp = 1.

' Similarly, the parameter %rff_weight_q1 is also designed to be
' used in real-time analysis when the first simulation quarter
' corresponds to a quarter that is already under way.  The
' parameter provides the fractional value to be given to the
' monetary policy rule; the remaining fractional value is given to
' an exogenous value.

' The document Simulation Basics discusses the effects of
' imposing the zero lower bound (ZLB) on the federal funds rate
' (%zerobound parameter) and imposing threshold conditions
' on the liftoff of the funds rate from a ZLB episode
' (%threshold parameter).

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
include ../subs/master_library

' Workfile
%wfstart = "1965q1"
```

```
  %wfend = "2020q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model name and location
  %varmod = "stdver"
  %varpath = "../mods/"
  %varinfo = "../mods/stdver_varinfo"

' Input database
  %dbin  = "../data/longbase"

' Simulation range
  %simstart = "2014q1"
  %simend   = "2018q4"

' Stochastic parameters
  rndseed 12345
  %errorblock = "1"
  %residstart = "1970q1"
  %residend   = "2012q4"
  %nsims      = "1000"
  %q1_shock_damp = ".5"
  %dbout_series = "rff lur picxfe picnia picx4 xgap2 hggdp anngr"

' Monetary policy
  %zerobound  = "yes"
  %threshold  = "yes"
  %rff_weight_q1  = ".25"

' ***************************************************************
' Retrieve data, model equations and coefficients, set
' policy options, and compute tracking residuals
' ***************************************************************

' Load equations, coefficients, and variable information
  ld_frbus_eqs(modelname=%varmod,modelpath=%varpath)
  ld_frbus_cfs(modelname=%varmod,modelpath=%varpath)
  ld_varinfo(pathname=%varinfo)

' add 4-qtr gdp growth equation
  {%varmod}.append anngr - anngr_aerr = 100*((xgdp/xgdp(-4))-1)

' Load data
  dbopen %dbin as longbase
  fetch(d=longbase) *
```

```
  smpl @all
  series anngr = 100*((xgdp/xgdp(-4))-1)

' Set monetary policy to inertial Taylor rule (dmpintay, rffintay)
  smpl @all
  call set_mp("dmpintay")
  if %zerobound = "yes" then
    rffmin = .250
    else
    rffmin = -9999
    endif
  if %threshold = "yes" and %zerobound = "no" then
    %err = "Error: policy threshold conditions can only be used when the ZLB is impos
    @uiprompt(%err)
    stop
    endif
  if %threshold = "yes" and %zerobound = "yes" then
    dmptrsh = 1
    dmptr = 0
    else
    dmptrsh = 0
    endif
  drstar = 0

  smpl {%simstart} {%simstart}
  dmpintay = @val(%rff_weight_q1)
  dmpex = 1 - dmpintay

' Set fiscal policy
  smpl @all
  call set_fp("dfpex")

  dmpstb = 1

' Set _aerr variables to zero
  smpl @all
  {%varmod}.makegroup(a,n) endog @endog
  call groupnew("endog","_aerr")
  call group2zero("endog_aerr")

' Standard solution options
  {%varmod}.solveopt(o=b,g=14,z=1e-14)

' Assign baseline tracking add factors
  %suftrk = "_0"
  smpl %residstart %simend
```

```
  {%varmod}.addassign @all
  {%varmod}.addinit(v=n) @all
  {%varmod}.scenario(n,a={%suftrk}) "track"
  {%varmod}.solve
  scalar mm = @max(@abs(xgap{%suftrk}-xgap))
  if mm > .0001 then
    statusline dynamic tracking simulation failed for {%varmod}
    stop
    endif

' ***************************************************************
' More monetary policy settings
' ***************************************************************


' if policy thresholds are turned on, set add factors on endogenous
' threshold switch variables to zero
  if %threshold = "yes" then
    smpl @all
    dmptpi_a = 0
    dmptlur_a = 0
    dmptmax_a = 0
    dmptr_a = 0
    endif

' if the zero bound is binding in part or all of the baseline, the
' adds (_a) on the policy rule and the funds rate equations are
' determined so as to satisfy the following conditions.
'
' a.  the stochastic funds rate equals the maximum of the zero bound and
'     the prediction of the chosen policy rule (this simply requires that
'     rffe_a and rffe_aerr be zero)
' b.  the prediction of the chosen policy rule is subject to _a add factors
'     that are determined as follows:
'     (1) in quarters when the zero bound is not binding in the baseline,
'         the associated add factors equal the values that make
'         the policy rule equation match the baseline funds rate under
'         baseline conditions (this is satisfied by the tracking adds on
'         the policy rule as long as the baseline value of the policy rule
'         variable equals the baseline funds rate);
'     (2) in quarters when the zero bound is binding in the baseline,
'         the associated add factors are determined by linear interpolation
'         of the add factors generated according to b(1) for the
'         unbound quarters;
'     (3) when the zero bound is binding in all baseline quarters, the
'         policy rule add factors are zero;
```

```
'     (4) the zero bound is assumed to be binding in the baseline whenever
'         the baseline funds rate (rffe) is within 25 basis points of the
'         zero bound variable (rffmin).

  smpl %simstart %simend
  series not_constrnd = ((rffe - rffmin) >= .25)
  !tmp_max = @max(not_constrnd)
  !tmp_min = @min(not_constrnd)

' zero bound binding in some quarters
  if (!tmp_max = 1) and (!tmp_min = 0) then
    smpl %simstart %simend
    rffe_a = 0
    rffe_aerr = 0
    rffintay_aerr = 0
    smpl %simstart %simend if (not_constrnd = 0)
    rffintay_a = NA
    %series_in = "rffintay_a"
    %series_out = %series_in + "_int"
    call interp_lin(%series_in,%series_out,%simstart,%simend)
    rffintay_a = {%series_out}
    endif

' zero bound binding in all quarters
  if (!tmp_max = 0) and (!tmp_min = 0) then
    smpl %simstart %simend
    rffe_a = 0
    rffe_aerr = 0
    rffintay_aerr = 0
    rffintay_a = 0
    endif

' ****************************************************************
' Stochastic shocks
' ****************************************************************

' copy historical residuals into series whose names have _err suffixes
  smpl %residstart %residend
  copy *_a *_err

' use vinfo table to create list/group of equations to receive shocks
  %tmp = " "
  for !i = 1 to vinfo_size
    %vname = @word(vinfo_vname,!i)
    %stoch = @word(vinfo_stoch,!i)
    if %stoch <> "NO" then
```

```
      %tmp = %tmp + " " + %vname
      endif
    next
  group shock {%tmp}

' demean historical residuals and store them in a matrix

  smpl %residstart %residend
  %error_names = " "
  for !i = 1 to shock.@count
    %temp = shock.@seriesname(!i) + "_err"
    scalar mm = @mean({%temp})
      series {%temp} =  {%temp} - mm
      %error_names = %error_names + " " + %temp
      next

  group errors {%error_names}
  smpl %residstart %residend
  stom(errors,errormat)

' create table of error statistics
  !nrows = 3 + errors.@count
  !ncols = 3
  table(!nrows,!ncols) error_tab
  error_tab.setjust(r1c1:r{!nrows}c1) left
  error_tab.setwidth(1) 20
  error_tab(1,1) = "error"
  error_tab(1,2) = "mean"
  error_tab(1,3) = "std-dev"
  smpl %residstart %residend
  for !i = 1 to errors.@count
    series tseries = errors(!i)
    error_tab(!i+3,1) = errors.@seriesname(!i)
    error_tab(!i+3,2) = @mean(tseries)
    error_tab(!i+3,3) = @stdev(tseries)
    next

' miscellaneous

  smpl %simstart %simend
  scalar nqtrs = @obssmpl
  scalar nrepl = {%nsims}
  scalar nsims = {%nsims}
  scalar nerrors = @rows(errormat)
  scalar bbbb = nqtrs-nerrors
```

```
  call groupnew("shock","_aerr")

  group track {%dbout_series}
  call groupnames2string("track",%tracknames)

' for tracked variables, set up matrices to hold stochastic results
  for !i = 1 to track.@count
    %temp = track.@seriesname(!i)
    matrix(nqtrs,nrepl) {%temp}_mat
    next


' ***************************************************************
' Stochastic sims
' ***************************************************************

  %sufstoch = "_1"
  {%varmod}.scenario(n,a={%sufstoch}) "stoch sims"


' *****************************************
' stochastic simulation loop (sims are run one at a time)

  smpl %simstart %simend
  for !i = 1 to nrepl
    statusline running stochastic sim number  !i
  ' draw nqtrs random rows from the matrix of historical errors,
  ' damp the shocks in the first drawn row, and load the shocks
  ' into the respective  _aerr error series
    matrix stocherrors = @resample(errormat,bbbb,{%errorblock})
    for !j = 1 to @columns(stocherrors)
      stocherrors(1,!j) = @val(%q1_shock_damp) * stocherrors(1,!j)
      next
    mtos(stocherrors,shock_aerr)
    {%varmod}.solve
  ' store solution values
    for !j = 1 to track.@count
      %temp = track.@seriesname(!j)
      %temp1 = %temp + "_mat"
      stom({%temp}{%sufstoch},tmp)
      colplace({%temp1},tmp,!i)
      next
    next

' ***************************************************************
' Statistics
```

```
' ****************************************************************

  statusline computing statistics

  smpl %simstart %simend
  !index = 2
  series year = @year
  series quarter = @quarter
  alpha yyyyqq = @str(year) + "Q" + @str(quarter)
  !lqtr = @dtoo(%simstart) - 1
  !nstats = 8

'create a summary table in which to store key results

  call tableform("summary_tab","100")

' loop over each tracked variable,
  for !ii1 = 1 to track.@count
    %trkname = track.@seriesname(!ii1)
  ' compute statistics
    call makestats(%trkname)

  ' load statistics into variable-specific table
    %tabname = %trkname + "_tab"
    call tableform(%tabname,@str(nqtrs+2))
    for !ii2 = 1 to nqtrs
      {%trkname}_tab(!ii2+2,1) = yyyyqq(!lqtr + !ii2)
      for !ii3 = 1 to !nstats
        {%trkname}_tab(!ii2+2,!ii3+1) = {%trkname}_stats(!ii2,!ii3)
        next
      next

  ' load statistics for each q4 observation into summary table
    !index = !index + 1
    summary_tab(!index,1) = %trkname
    !index = !index + 1
    for !ii2 = 1 to nqtrs
      if quarter(!lqtr+!ii2) = 4 then
        for !ii3 = 1 to !nstats + 1
          summary_tab(!index,!ii3) = {%trkname}_tab(!ii2+2,!ii3)
          next
        !index = !index + 1
        endif
      next

  ' make graph showing 70 and 90 percent bands
```

```
    graph {%trkname}_graph.band {%trkname}_lo90 {%trkname}_hi90 {%trkname}_lo70 _
        {%trkname}_hi70 {%trkname}_base
    {%trkname}_graph.addtext(t) %trkname
    {%trkname}_graph.options size(6,4.5)

  next


' ************************************
' summary graph

  lur_graph.addtext(t,just(c),font("arial",12)) Unemployment Rate
  rff_graph.addtext(t,just(c),font("arial",12)) Federal Funds Rate
  picx4_graph.addtext(t,just(c),font("arial",12)) 4-qtr Core Inflation Rate
  anngr_graph.addtext(t,just(c),font("arial",12)) 4-qtr Real GDP Growth Rate
  lur_graph.legend -display
  rff_graph.legend -display
  picx4_graph.legend -display
  anngr_graph.legend -display

  graph summary_graph.merge lur_graph rff_graph picx4_graph anngr_graph
  summary_graph.legend -display
  summary_graph.addtext(t,just(c),font("Arial",16)) Stochastic Simulations\r(70 and 9
  show summary_graph


' ************************************
' summary spool
  spool results
  summary_tab.deleterow(!index) 100
  results.append summary_tab
  results.append summary_graph
  results.append error_tab
  results.display
```

Uses dbopen 217a, dfpex 219, dmpex 219, dmpintay 219, dmpstb 219, dmptr 219, dmptrsh 219, drstar 219, fetch 217a, group2zero 91a, groupnames2string 91b, groupnew 92, hggdp 219, interp_lin 93, ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, ld_varinfo 193 194, left 219, longbase 217b, lur 219, makestats 208, master_library 209a, picnia 219, picx4 219, picxfe 219, rff 219, rffe 219, rffintay 219, rffmin 219, set_fp 94, set_mp 95, solve 217a, stoch 218, tableform 207b, varinfo 218, vinfo_stoch 218, vinfo_vname 218, vname 218, wfcreate 217a, xgap 219, xgap2 219, and xgdp 219.

Chapter 3

Model Consistent Expectation Solve Package

3.1 example1 program

51 ⟨*mce example1* 51⟩≡ (214a)

```
' This example illustrates:
'
'   - The automated approach to constructing the two operational
'     models
'   - How to define the multiplier shock in a text
'     file whose lines are executed from within the call to
'     mce_run
'   - How to declare the model scenario and the scenario
'     scenario alias within the call to mce_run
'   - The use of the "linear" option of the "newton" algorithm
'     for a linear model in which the maximum endogenous lead and
'     lag is one period

'************************************************************
'************************************************************
'************************************************************
' Section 1:  Workfile, model name, simulation range

  include  mce_solve_library

' Workfile
  %wfstart = "2000q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}
```

```
' Model name
  %mod = "simple"

' Simulation range
  %simstart = "2001q1"
  %simend = "2025q4"


'*************************************************************
'*************************************************************
'*************************************************************
' Section 2:  Model, coefficients, and data

' equations

  model {%mod}
  {%mod}.append pinf = cp(1) * pinf(-1) + (.98-cp(1))*pinf(1)+ cp(2) * ygap
  {%mod}.append rate = cr(1)*rate(-1)+(1-cr(1))*(cr(2)*pinf + cr(3)*ygap)
  {%mod}.append ygap = cy(1) * ygap(-1) + (.98-cy(1))*ygap(1) + cy(2) * (rate - pinf


' coefficients

  coef(2) cy
  cy.fill .50, -.02
  coef(2) cp
  cp.fill .50, .02
  coef(3) cr
  cr.fill .75, 1.5, 0.5

' set all data to zero
  smpl @all
  %vars = {%mod}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
    series {%tmp} = 0
    next


'*************************************************************
'*************************************************************
'*************************************************************
' Section 3: Simulation

  text shock1
```

```
shock1.append smpl {%simstart} {%simstart}
shock1.append series rate_a = rate_a + 1

%mopts = "create,mod=%mod,adds,track"
%aopts = "jinit=linear"
%sopts = "type=single,txt=shock1,scen,suf=_1"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)
copy mce_sim_spool mce_sim_spool_1
show mce_sim_spool_1

series zero = 0
smpl %simstart %simstart + 39
graph gr1.line zero rate_1 pinf_1 ygap_1
gr1.addtext(t,c,font(14)) "Positive interest rate shock"
show gr1
```

Uses mce_run 99, mce_solve_library 210 215b, and wfcreate 217a.

3.2 example2 program

54 ⟨*mce example2* 54⟩≡ (214b)

```
' This example illustrates:
'
'   - A manual approach to constructing the two operational
'     models that mimics what the automated approach does
'   - How to define the multiplier shock in a text
'     file whose lines are executed from within the call to
'     mce_run
'   - How to declare the model scenario and the scenario
'     scenario alias within the call to mce_run
'   - The use of the "linear" option of the "newton" algorithm
'     for a linear model in which the maximum endogenous lead and
'     lag is one period

'*************************************************************
'*************************************************************
'*************************************************************
' Section 1:  Workfile, model name, simulation range

  include  mce_solve_library

' Workfile
  %wfstart = "2000q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' Model names
  %modb = "simpleb"
  %modf = "simplef"

' Simulation range
  %simstart = "2001q1"
  %simend = "2025q4"


'*************************************************************
'*************************************************************
'*************************************************************
' Section 2:  Model, coefficients, and data

' equations in backward-looking model

  model {%modb}
```

```
  {%modb}.append pinf = cp(1) * pinf(-1) + (.98-cp(1))*zpinf+ cp(2) * ygap
  {%modb}.append rate = cr(1)*rate(-1)+(1-cr(1))*(cr(2)*pinf + cr(3)*ygap)
  {%modb}.append ygap = cy(1) * ygap(-1) + (.98-cy(1))*zygap + cy(2) * (rate - zpinf)

' equations in expectations errors model

  model {%modf}
  {%modf}.append ezpinf = zpinf - pinf(1)
  {%modf}.append ezygap = zygap - ygap(1)

' coefficients

  coef(2) cy
  cy.fill .50, -.02
  coef(2) cp
  cp.fill .50, .02
  coef(3) cr
  cr.fill .75, 1.5, 0.5

' set all data to zero
  smpl @all
  %vars = {%modb}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
    series {%tmp} = 0
    next
  %vars = {%modf}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
    series {%tmp} = 0
    next

' declare mce variables and instruments
  %instrus = "zpinf zygap"
  %errs = "ezpinf ezygap"

'************************************************************
'************************************************************
'************************************************************
' Section 3: Simulation


  text shock1
  shock1.append smpl {%simstart} {%simstart}
  shock1.append series rate_a = rate_a + 1
```

```
%mopts = "mod_b=%modb,mod_f=%modf,mce_instrus=%instrus,mce_errs=%errs,adds,track"
%aopts = "jinit=linear"
%sopts = "type=single,txt=shock1,scen,suf=_1"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)
copy mce_sim_spool mce_sim_spool_1
show mce_sim_spool_1

series zero = 0
smpl %simstart %simstart + 39
graph gr1.line zero rate_1 pinf_1 ygap_1
gr1.addtext(t,c,font(14)) "Positive interest rate shock"
show gr1
```

Uses mce_run 99, mce_solve_library 210 215b, and wfcreate 217a.

3.3 example3 program

57 ⟨*mce example3* 57⟩≡ (214c)

```
' This example illustrates:
'
'   - Another manual approach to constructing the two operational
'     models that introduces new endogenous variables for the
'     expectations leads along with simple equations for the new
'     endogenous variables.
'   - The MCE instruments are the add factors on the equations for
'     the new endogenous variables; this circumstance requires that
'     add factors be assigned to the operational models prior to
'     the call to mce_run.
'   - The option of defining the multiplier shock in commands that
'     are executed prior to the call to mce_run, when the manual
'     approach is used.
'   - The option of declaring model scenarios and the scenario
'     scenario alias prior to the call to mce_run, when the manual
'     approach is used.
'   - The model no longer satisfies the conditions for which the
'     "jinit=linear" option of the "newton" algorithm is designed.
'     This example uses "jinit=interp(4)" to specify a
'     particular approximate Jacobian.
'
'   - This example runs three simulations.  In the second and third
'     simulations, the assignment of null strings to the first two
'     arguments of the mce_run subroutine causes the simulations to be
'     run with the same internal models and algorithm (including
'     the Newton MCE Jacobian) that were created or declared in the
'     first simulation.

'*************************************************************
'*************************************************************
'*************************************************************
' Section 1:  Workfile, model name, simulation range

  include  mce_solve_library

' Workfile
  %wfstart = "2000q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' Model names
  %modb = "simpleb"
```

```
  %modf = "simplef"

' Simulation range
  %simstart = "2001q1"
  %simend = "2025q4"


'*************************************************************
'*************************************************************
'*************************************************************
' Section 2:  Model, coefficients, and data

' equations in backward-looking model

  model {%modb}
  {%modb}.append pinf = cp(1) * pinf(-1) + (.98-cp(1))*zpinf+ cp(2) * ygap
  {%modb}.append rate = cr(1)*rate(-1)+(1-cr(1))*(cr(2)*pinf + cr(3)*ygap)
  {%modb}.append ygap = cy(1) * ygap(-1) + (.98-cy(1))*zygap + cy(2) * (rate - zpinf)
  {%modb}.append zpinf = @movav(pinf(-1),4)
  {%modb}.append zygap = @movav(ygap(-1),4)

' equations in expectations errors model

  model {%modf}
  {%modf}.append ezpinf = zpinf - pinf(1)
  {%modf}.append ezygap = zygap - ygap(1)

' coefficients

  coef(2) cy
  cy.fill .50, -.02
  coef(2) cp
  cp.fill .50, .02
  coef(3) cr
  cr.fill .75, 1.5, 0.5

' set all data to zero
  smpl @all
  %vars = {%modb}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
    series {%tmp} = 0
    next
  %vars = {%modf}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
```

```
      series {%tmp} = 0
      next

  ' declare mce variables and instruments
    %instrus = "zpinf_a zygap_a"
    %errs = "ezpinf ezygap"

  ' assign tracking add factors
    smpl %simstart %simend
    {%modb}.addassign @all
    {%modb}.addinit(v=n) @all
    {%modf}.addassign @all
    {%modf}.addinit(v=n) @all

  '************************************************************
  '************************************************************
  '************************************************************
  ' Section 3: Simulations

    %sufm = "_1"
    {%modb}.scenario(n,a=%sufm) "multiplier"
    {%modf}.scenario(n,a=%sufm) "multiplier"


  ' Sim 1:  interest rate shock

    smpl {%simstart} {%simstart}
    rate_a = rate_a + 1

    %mopts = "mod_b=%modb,mod_f=%modf,mce_instrus=%instrus,mce_errs=%errs"
    %aopts = "jinit=interp(4)"
    %sopts = "type=single"
    smpl {%simstart} {%simend}
    call mce_run(%mopts,%aopts,%sopts)
    smpl {%simstart} {%simstart}
    rate_a = rate_a - 1

    series zero = 0
    smpl %simstart %simstart + 39
    graph gr1.line zero rate{%sufm} pinf{%sufm} ygap{%sufm}
    gr1.addtext(t,c,font(14)) "Positive interest rate shock"
    show gr1


  ' Sim 2:  output gap shock
```

```
smpl {%simstart} {%simstart}
ygap_a = ygap_a + 1

%mopts = ""
%aopts = ""
%sopts = "type=single"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)
smpl {%simstart} {%simstart}
ygap_a = ygap_a - 1

series zero = 0
smpl %simstart %simstart + 39
graph gr2.line zero rate{%sufm} pinf{%sufm} ygap{%sufm}
gr2.addtext(t,c,font(14)) "Positive output gap shock"
show gr2

' Sim 3:  inflation shock

smpl {%simstart} {%simstart}
pinf_a = pinf_a + 1

%mopts = ""
%aopts = ""
%sopts = "type=single"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)
smpl {%simstart} {%simstart}
pinf_a = pinf_a - 1

series zero = 0
smpl %simstart %simstart + 39
graph gr3.line zero rate{%sufm} pinf{%sufm} ygap{%sufm}
gr3.addtext(t,c,font(14)) "Positive inflation shock"
show gr3
```

Uses mce_run 99, mce_solve_library 210 215b, and wfcreate 217a.

3.4 example4 program

61 ⟨*mce example4* 61⟩≡ (214d)

```
' This example illustrates:
'
'   - The simulation of a nonlinear model (zero bound imposed)
'     (because the baseline data is set to zero; the zero-bound
'     is set illustratively to -1)
'   - The use of the qnewton algorithm

'************************************************************
'************************************************************
'************************************************************
' Section 1:  Workfile, model name, simulation range

  include  mce_solve_library

' Workfile
  %wfstart = "2000q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' Model names
  %modb = "simpleb"
  %modf = "simplef"

' Simulation range
  %simstart = "2001q1"
  %simend = "2025q4"


'************************************************************
'************************************************************
'************************************************************
' Section 2:  Model, coefficients, and data

' equations in backward-looking model

  model {%modb}
  {%modb}.append pinf = cp(1) * pinf(-1) + (.98-cp(1))*zpinf+ cp(2) * ygap
  {%modb}.append rate_u = cr(1)*rate(-1)+(1-cr(1))*(cr(2)*pinf + cr(3)*ygap)
  {%modb}.append rate = @recode( rate_u>rate_min,rate_u,rate_min)
  {%modb}.append ygap = cy(1) * ygap(-1) + (.98-cy(1))*zygap + cy(2) * (rate - zpinf)
  {%modb}.append zpinf = @movav(pinf(-1),4)
  {%modb}.append zygap = @movav(ygap(-1),4)
```

```
' equations in expectations errors model

model {%modf}
{%modf}.append ezpinf = zpinf - pinf(1)
{%modf}.append ezygap = zygap - ygap(1)

' coefficients

coef(2) cy
cy.fill .50, -.02
coef(2) cp
cp.fill .50, .02
coef(3) cr
cr.fill .75, 1.5, 0.5

' set all data to zero
smpl @all
%vars = {%modb}.@varlist
for !i = 1 to @wcount(%vars)
  %tmp = @word(%vars,!i)
  series {%tmp} = 0
  next
%vars = {%modf}.@varlist
for !i = 1 to @wcount(%vars)
  %tmp = @word(%vars,!i)
  series {%tmp} = 0
  next

' declare mce variables and instruments
%instrus = "zpinf_a zygap_a"
%errs = "ezpinf ezygap"

' assign tracking add factors
smpl %simstart %simend
{%modb}.addassign @all
{%modb}.addinit(v=n) @all
{%modf}.addassign @all
{%modf}.addinit(v=n) @all

'*************************************************************
'*************************************************************
'*************************************************************
' Section 3: Simulation

%zb = "yes"
```

```
%sufm = "_1"
{%modb}.scenario(n,a=%sufm) "multiplier"
{%modf}.scenario(n,a=%sufm) "multiplier"

if %zb = "yes" then
  smpl @all
  rate_min = -1
  else
  rate_min = -9999
  endif

smpl {%simstart} {%simstart}
ygap_a = ygap_a - 5

%mopts = "mod_b=%modb,mod_f=%modf,mce_instrus=%instrus,mce_errs=%errs"
%aopts = "meth=qnewton"
%sopts = "type=single"
smpl {%simstart} {%simend}
tic
call mce_run(%mopts,%aopts,%sopts)
scalar elapsed = @toc
show elapsed

series zero = 0
smpl %simstart %simstart + 39
graph gr1.line zero rate{%sufm} pinf{%sufm} ygap{%sufm}
%title = "Negative Output Shock"
if %zb = "yes" then
  %title = %title + "\r(zero bound imposed)"
  else
  %title = %title + "\r(zero bound not imposed)"
  endif
gr1.addtext(t,c,font(14)) %title
show gr1
```

Uses mce_run 99, mce_solve_library 210 215b, and wfcreate 217a.

3.5 example5 program

64 ⟨*mce example5* 64⟩≡ (215a)

```
' This example illustrates the two optimal policy simulation types
'
'   - Simulate the effects of a positive shock to the output gap
'     using sequentially
'
'     (a) the model's interest rate rule
'
'     (b) the opt simulation type to find the optimal interest rate
'     path under commitment
'
'     (c) the opttc simulation type to find the optimal time-consistent
'     or discretionary interest rate path; note that the solution in
'     this case is only approximate
'
'     In both (b) and (c) the policy instrument is the residual of
'     the interest rule
'
'   - The illustrative loss function penalizes equally weighted,
'     discounted, squared deviations of the output gap,
'     inflation, and the first difference of the interest rate.
'

'************************************************************
'************************************************************
'************************************************************
' Section 1:  Workfile, model name, simulation range

  include  mce_solve_library

' Workfile
  %wfstart = "2000q1"
  %wfend = "2100q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' Model name
  %mod = "simple"

' Simulation range
  %simstart = "2001q1"
  %simend = "2025q4"

'************************************************************
```

```
'************************************************************
'************************************************************
' Section 2:  Model, coefficients, and data

' equations

  model {%mod}
  {%mod}.append pinf = cp(1) * pinf(-1) + (.98-cp(1))*pinf(1)+ cp(2) * ygap
  {%mod}.append rate - rate_aerr = cr(1)*rate(-1)+(1-cr(1))*(cr(2)*pinf + cr(3)*ygap)
  {%mod}.append ygap = cy(1) * ygap(-1) + (.98-cy(1))*ygap(1) + cy(2) * (rate - pinf(1))
  {%mod}.append drate = rate - rate(-1)


' coefficients

  coef(2) cy
  cy.fill .50, -.02
  coef(2) cp
  cp.fill .50, .02
  coef(3) cr
  cr.fill .75, 1.5, 0.5

' set all data to zero
  smpl @all
  %vars = {%mod}.@varlist
  for !i = 1 to @wcount(%vars)
    %tmp = @word(%vars,!i)
    series {%tmp} = 0
    next

'************************************************************
'************************************************************
'************************************************************
' Section 3: Optimal policy setup

' targets, instruments
  group opt_instrus rate_aerr
  group opt_targs pinf ygap drate

' desired target trajectories
  smpl @all
  series pinf_t = 0
  series ygap_t = 0
  series drate_t = 0

' loss function weights
```

```
%discount = ".99"
%y_weight = "1.0"
%p_weight = "1.0"
%r_weight = "1.0"
smpl @all
series ygap_w = @val(%y_weight)
series pinf_w =  @val(%p_weight)
series drate_w  =  @val(%r_weight)
!discount = @val(%discount)
smpl %simstart+1 %simend
ygap_w = !discount * ygap_w(-1)
pinf_w = !discount * pinf_w(-1)
drate_w = !discount * drate_w(-1)


'*************************************************************
'*************************************************************
'*************************************************************
' Section 3: Simulations

text shock1
shock1.append smpl {%simstart} {%simstart}
shock1.append series ygap_a = ygap_a + 3

' run the simulation using the monetary policy rule

%mopts = "create,mod=%mod,adds,track"
%aopts = "jinit=linear"
%sopts = "type= single,txt=shock1,scen,suf=_1"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)

smpl @all
series rate_rule = rate_1
series pinf_rule = pinf_1
series ygap_rule = ygap_1


' run the simulation using opt

%mopts = ""
%aopts = ""
%sopts = "type=opt,instrus=opt_instrus,targs=opt_targs"
smpl {%simstart} {%simend}
call mce_run(%mopts,%aopts,%sopts)
```

```
    smpl @all
    series rate_opt = rate_1
    series pinf_opt = pinf_1
    series ygap_opt = ygap_1


  ' run the simulation using opttc

    %mopts = ""
    %aopts = ""
    %sopts = "type=opttc,instrus=opt_instrus,targs=opt_targs"
    smpl {%simstart} {%simend}
    call mce_run(%mopts,%aopts,%sopts)

    smpl @all
    series rate_opttc = rate_1
    series pinf_opttc = pinf_1
    series ygap_opttc = ygap_1

  'graph

    smpl %simstart %simstart + 39
    series zero = 0
    graph gr1.line zero rate_rule rate_opt rate_opttc
    graph gr2.line zero pinf_rule pinf_opt pinf_opttc
    graph gr3.line zero ygap_rule ygap_opt ygap_opttc
    graph gr4.merge gr1 gr2 gr3
    %title = "Effects of Positive Output Shock Under Three Policy Responses"
    %title = %title + "\r1.  An intertial interest rate rule (_rule)"
    %title = %title + "\r2.  Optimal policy under committment (_opt)"
    %title = %title + "\r3.  Optimal time-consistent policy (_opttc)"
    gr4.addtext(t,c,font(14)) %title
    show gr4
```

Uses mce_run 99, mce_solve_library 210 215b, and wfcreate 217a.

Chapter 4

State Space Package

4.1 Estimate Model

69 ⟨*estimation model* 69⟩≡ (215e)

```
' Program to estimate the FRB/US state-space model and
' generate estimates of model states.


' Subroutines:
'    To transform data
include "./data_transformations"

'    To estimate the model
include "./estimation_code"
include "./initial_values"

close @all
wfcreate kf_data q 1949:1 2020:4

%estend    = "2013q4"
%eststart  = "1963q2"
%datastart = "1949q1"



sample estsample  %eststart  %estend
sample datasample %datastart %estend
%datasmpl = %datastart + " " + %estend
%modname = "ss_model"
```

```
'*******************************************************************************
' Retrieve variables from the database

' Definitions of series in database
'    See FRB/US model documentation for more complete descriptions

' XGDP   - GDP, cw 2009$
' XGDPN  - GDP
' XB   - BLS Business output, 2009$
' XBN  - BLS Business output
' XGDIN  - GDI

' PGDP   - Price index for GDP, cw
' PXB  - BLS Business price

' LEP    - Employment in business sector (employee and self-employed)
' LHP    - Aggregate labor hours, business sector (employee and self-employed)
' LUR    - Civilian unemployment rate (break adjusted)
' LF     - Civilian labor force (break adjusted)
' N16    - Noninstitutional population, aged 16 and over (break adjusted)

' KS     - Capital services, 2009 $
' LQUALT - Labor quality, trend level
' VEOA   - Average energy-output ratio of existing capital stock

' PCXFE  - Price index for personal consumption expendits ex. food and energy, cw (NI
' PCER   - Price index for personal consumption expenditures on energy (relative to P

' PMO    - Price index for imports ex. petroleum, cw
' UCES   - Energy share of nominal consumption expenditures
' EMON   - Imports of goods and services ex. petroleum
' XGDEN  - Nominal Absorption
' PTR    - 10-year expected inflation (Hoey/Philadelphia survey)


%dbin = "state_space_data"
dbopen %dbin as dbin

string varlist = " xgdpn xbn pgdp pxb pcxfe pcer xgdp xb lep uces emon xgden ptr "
varlist = varlist + " ks lqualt veoa lhp lur lf n16 pmo xgdin "

fetch(d=dbin) {varlist}



'************************************************
```

```
' Make model observable series, set priors

smpl datasample

call data_transformations

'************************************************
' Specify and estimate model, save for re-use

smpl estsample

statusline Estimating model
call ss_estimation

{%modname}.makefilter saved_results



'************************************************************
' Create states

{%modname}.makestates(t=pred) *_prs        ' one-step ahead state predictions
{%modname}.makestates(t=filt) *_1          ' filter states
{%modname}.makestates(t=smooth) *_2        ' smoothed states
{%modname}.makestates(t=disturb) *err      ' estimate of the disturbances
{%modname}.makestates(t=filtse) *se        ' RMSE of the filtered states
{%modname}.makestates(t=smoothse) *se2     ' RMSE of the smoothed states

{%modname}.makesignals(t=pred) *_pr        ' one-step ahead signal predictions
{%modname}.makesignals(t=resid) *_res      ' error in one-step ahead signal predictions
{%modname}.makesignals(t=stdresid) *_sres  ' standardized one-step ahead prediction residual


'************************************************************
' Transformation into FRB/US mnemonics

series xbt  = exp((tmfp_2/.965 + 0.725*(terate_2 + tlfpr_2  + thtfactor_2 + tww_2) + _
                0.275*lks + 0.725*llqualt + (.035/.965)*lveoa +lpop)/100)
series hmfpt  = gtmfp_2
series qlfpr  = exp(tlfpr_2/100)
series hqlfpr = (gtlfpr_2/400)*qlfpr(-1)
series hqlww  = gtww_2
series huxb   = gtotfactor_2
series leppot = exp((tlfpr_2+terate_2+thtfactor_2)/100) * n16
series lurnat = 100*(1-exp(terate_2/100))
series qlww   = exp(tww_2/100)
```

```
series uxbt   = exp(totfactor_2/100)
series xgdpt  = xbt * uxbt
series xgdo   = exp(cycle_2/100)*xgdpt


wfsave saved_results_new
```

Defines:
saved_results_new, never used.
Uses close 217a, data_transformations 215c, dbopen 217a, emon 219, estimation_code 215d, ex 219, fetch 217a, hmfpt 219, hqlfpr 219, hqlww 219, huxb 219, initial_values 83 216b, ks 219, lep 219, leppot 219, lf 219, lhp 219, lqualt 219, lur 219, lurnat 219, n16 219, pcer 219, pcxfe 219, pgdp 219, pmo 219, ptr 219, pxb 219, qlfpr 219, qlww 219, sector 218, ss_estimation 76, uces 219, uxbt 219, veoa 219, wfcreate 217a, wfsave 217a, xb 219, xbn 219, xbt 219, xgden 219, xgdin 219, xgdo 219, xgdp 219, xgdpn 219, and xgdpt 219.

4.2 Data Transformations

73 ⟨*data transformations* 73⟩≡ (215c)

```
subroutine data_transformations()

' Subroutine for FRB/US state-space model package to
' transform raw data from EViews database to observables
' used in the estimation of the state-space model.

' 1. Data transformations

series lpop = 100*log(n16)

'Business - product side
series nbp = xbn
series bp = log(xb)*100 -lpop

'GDP
series gdp = log(xgdp)*100 -lpop

'Buseness - income side
series dsnst_q = xgdpn - xgdin
series nbi = nbp - dsnst_q
series rbi = nbi/(pxb/100)
series bi = log(rbi)*100 -lpop

' Employment Business sector
series eb = log(lep)*100  - lpop

' Workweek
series hb = log(lhp)*100 - lpop
series bww = hb - eb

' Employment Rate
series erate = 100*log((100-lur)/100)

' Participation Rate
series lfpr = 100*log(lf/n16)

' Variables for TMFP:
series lks = 100*log(ks) - lpop
series llqualt = 100*log(lqualt)
series lveoa = 100*log(veoa)

' Series for the price equation
series c1pcex = @pca(pcxfe)
```

```
series engylag_pcex = 0.5*(uces(-1)+uces(-2))*@pca(pcer(-1))
series sw_coreimp_pcex = 0.5*((emon/xgden)+(emon(-1)/xgden(-1)))*@pca(pmo/pcxfe)
series dum84 = @year>=1985
series frzbulg = 0   ''' Nixon wage-price control programs
smpl 1971q3 1974q1
frzbulg = 1
smpl 1974q2 1974q4
frzbulg = -3.666



' For the output-sector ratio, we use the median unbiased approach of Stock
' and Watson (1998). Thus, tau_oti is the ratio of the variances of the level
' and drift shocks.

scalar tau_oti = .033260  ' modifier for the drift totfactor (OSR*) error term


'*******************************************************************************

' 2. Prior starting values for states.

'    For mean zero level states, set prior to zero.
'    For other level states, set prior to a data-based
'        value near the start of the sample.
'    For most drift terms, set prior to zero.
'        Exception is trend MFP, set equal to sample average.

' Cycle
scalar     icycle  =      0
scalar     icycle1 =      0
scalar     icycle2 =      0

' Measurement error
scalar     ie3p =         0
scalar     ie3i =         0

' Levels of trends
scalar     itotfactor =         @elem(gdp,%eststart) - @elem(bp,%eststart)
scalar     itmfp =             .965*(@elem(bp,%eststart) - 0.725*(@elem(eb,%eststart)
                                  + @elem(bww,%eststart))  - 0.275*@elem(lks,%eststa
                                  - 0.725*@elem(llqualt,%eststart) _
                                  - (.035/.965)*@elem(lveoa,%eststart))
scalar     itww =               @elem(bww,%eststart)
scalar     ithtfactor =         @elem(eb,%eststart) - @elem(erate,%eststart) _
                                - @elem(lfpr,%eststart)
```

```
scalar     itlfpr =             @elem(lfpr,%eststart)
scalar     iterate =            @elem(erate,%eststart)

' Initial drift terms
scalar     igtotfactor =        0.0
scalar     igtmfp =             1.7
scalar     igtww =              0.0
scalar     igthtfactor =        0.0
scalar     igtlfpr =            0.0

scalar nstate = 25

' The mpriors *must* be in the same order as the states are in the model object
vector(nstate) mpriors
mpriors.fill   0, ie3p, ie3i, 0, 0, _
               0, 0, icycle, icycle1, icycle2, _
               itotfactor, igtotfactor, itmfp, igtmfp, itww, _
               itww, igtww, ithtfactor, ithtfactor, igthtfactor, _
               itlfpr, itlfpr, igtlfpr, iterate, iterate


' Set starting values for variance priors
'     Variance priors set at a high value.  In estimation, variance
'     drops sharply in early periods of estimation sample.


sym(nstate) vpriors
for !d = 1 to nstate
   vpriors(!d, !d) = 3
   next

' A tighter prior for drift variances,
vpriors(12,12) = 1    ' igtotfactor
vpriors(14,14) = 1    ' igtmfp
vpriors(17,17) = 1    ' igtww
vpriors(20,20) = 1    ' igthtfactor

endsub
```

Uses data_transformations 215c, emon 219, ks 219, lep 219, lf 219, lfpr 219, lhp 219, lqualt 219, lur 219, n16 219, pcer 219, pcxfe 219, pmo 219, pxb 219, sector 218, uces 219, veoa 219, xb 219, xbn 219, xgden 219, xgdin 219, xgdp 219, and xgdpn 219.

4.3 Estimation Code

76 ⟨*estimation code* 76⟩≡ (215d)

```
subroutine ss_estimation()

' Subroutine for FRB/US state-space model package to
' estimate the model parameters.

'*******************************************************************************

' 1. Starting values for parameters.

' All of the coefficients in equations appear as beta or phi.
' Note that some values that are set here are hard coded in the estimation code belo

call initial_values

'*******************************************************************************
'  Model

sspace {%modname}  ' Declare a new state-space model object

' 2. Output equations.

' In the output equations, trend output is related to the capital stock, energy inte
' and trends for labor input using a production function.  The parameters of the pro
' function are hard-coded to the values in the FRB/US model.

' GDP observable
{%modname}.append @signal gdp = totfactor + tmfp/.965 + 0.725*(terate + tlfpr + thtf
                              + 0.275*lks + 0.725*llqualt + (.035/.965)*lveoa _
                              + cycle + beta(11)*beta(6) + beta(11)*e3p  + rexo
{%modname}.append @state rexo = [var=0.0000001^2]

' Buisines sector product-side observable
{%modname}.append @signal bp = tmfp/.965 + 0.725*(terate + tlfpr  + thtfactor + tww)
                              + 0.275*lks + 0.725*llqualt + (.035/.965)*lveoa  _
                              + beta(10)*cycle  +   beta(6) + e3p
{%modname}.append @state e3p =  beta(602)*e3p(-1) + re3p
{%modname}.append @ename re3p 'business prod error
{%modname}.append @evar var(re3p) =  (beta(125)^2)

' Business sector income-side observable
{%modname}.append @signal bi =  tmfp/.965  + 0.725*(terate + tlfpr  + thtfactor + tw
                              + 0.275*lks + 0.725*llqualt + (.035/.965)*lveoa _
                              + beta(10)*cycle - beta(6) + e3i
```

```
{%modname}.append @state e3i =   beta(602)*e3i(-1) + re3i
{%modname}.append @ename re3i 'business income error
{%modname}.append @evar var(re3i) =  (beta(126)^2)

' 3. Labor equations.

' Workweek observable
' To make this equation more consistent with the FRB/US workweek specification,
' a contemporaneous change in the cycle is included as well as the level of the
' cycle.  In addition, the coefficient on the lagged gap term in this equation is
' hard-coded at a value similar to that implicit in the FRB/US model.

{%modname}.append @signal bww =  tww _
             + phi(20)*(cycle-cycle1) _
             + phi(22)*cycle1 _
             + 0.82*(bww(-1)-tww1) _
             + eww
{%modname}.append @state eww =  [var=beta(104)^2]

' Employment observable
{%modname}.append @signal eb = terate + tlfpr + thtfactor + _
             + phi(30)*cycle _
             + phi(31)*(eb(-1)-(terate1 + tlfpr1 +thtfactor1)) _
             + eeb
{%modname}.append @state eeb = [var=beta(105)^2]

' Employment rate observable
{%modname}.append @signal erate = terate + phi(50)*cycle _
                     + phi(51)*(erate(-1)-terate1) _
                     + eerate
{%modname}.append @state eerate =  [var=beta(106)^2]

' Labor force participation observable
{%modname}.append @signal lfpr =  tlfpr + phi(40)*cycle _
                     + phi(41)*(lfpr(-1)-tlfpr1) _
                     + elfpr
{%modname}.append @state elfpr = [var=beta(107)^2]

' 4. Price observable
        {%modname}.append @ename ec
        {%modname}.append @evar var(ec) = (beta(109)^2)
          {%modname}.append @signal c1pcex = _
          beta(401)*c1pcex(-1) + (1-beta(401))*(ptr(-1)+.1) _
        + beta(404)*(((.50*cycle + .33*cycle1 + .17*cycle2 )) )  _
                + beta(405)*@movav(engylag_pcex(-1),6) _
                + beta(406)*@movav(dum84*engylag_pcex(-1),6)  _
```

```
        + beta(407)*frzbulg _
        + beta(408)*sw_coreimp_pcex + beta(409)*sw_coreimp_pcex(-1) _
        + ec

' 5. State equations

' Cycle state
{%modname}.append @state cycle = beta(1)*cycle(-1) + beta(2)*cycle1(-1) + ecycle
{%modname}.append @state cycle1 = cycle(-1)
{%modname}.append @state cycle2 = cycle1(-1)
{%modname}.append @ename ecycle
{%modname}.append @evar var(ecycle) = (beta(111)^2)

' Trends

'  Totfactor = OSR* in F.R.(2011)
'  tau_oti is taken from FR(2011)
{%modname}.append @state totfactor = totfactor(-1) + 0.25*.95*(gtotfactor(-1) ) + 0.2
                                     etotfactor + 0.25*egtotfactor
{%modname}.append @state gtotfactor = .95*gtotfactor(-1) + .05*beta(213) + egtotfacto
{%modname}.append @ename etotfactor
{%modname}.append @evar var(etotfactor) = (beta(112)^2)
{%modname}.append @ename egtotfactor
{%modname}.append @evar var(egtotfactor) = (4*((tau_oti))*beta(112))^2

' Multi-Factor Productivity (OPH*)
{%modname}.append @state tmfp = tmfp(-1) +  etmfp + 0.25*.95*(gtmfp(-1)) + 0.25*0.05*
{%modname}.append @state gtmfp = 0.95*gtmfp(-1) + 0.05*beta(214) + egtmfp
{%modname}.append @ename etmfp
{%modname}.append @evar var(etmfp) = beta(114)^2
{%modname}.append @ename egtmfp
{%modname}.append @evar var(egtmfp) = 0.14^2 ' beta(115)^2

'     For the workweek, employment-sector ratio, and LFPR, the level variances are ha
'     This hard-coding is done for convenience in production work.  In particular, th
'     values are close to the values these parameters take on when they are freely es
'     However, the t-ratios for these parameters are not very high, and that imprecis
'     to sluggish convergence; imposing these values shortens the time needed for est

' Workweek (WW*)
{%modname}.append @state tww = tww(-1) + 0.25*.95*(gtww(-1) ) +  etww + 0.25*.05*beta
{%modname}.append @state tww1 = tww(-1)
{%modname}.append @state gtww = .95*gtww(-1) + .05*beta(216) + egtww
{%modname}.append @ename etww
{%modname}.append @ename egtww
{%modname}.append @evar var(etww)  =  0.1^2
```

```
{%modname}.append @evar var(egtww) =  beta(117)^2

' Employment Sector Ratio (ESR*)
{%modname}.append @state thtfactor = thtfactor(-1) + 0.25*.95*gthtfactor(-1) + _
                                    0.25*egthtfactor + ethtfactor
{%modname}.append @state thtfactor1 = thtfactor(-1)
{%modname}.append @state gthtfactor = 0.95*gthtfactor(-1) + egthtfactor
{%modname}.append @ename ethtfactor
{%modname}.append @evar var(ethtfactor) = .01^2 ' beta(118)^2
{%modname}.append @ename egthtfactor
{%modname}.append @evar var(egthtfactor) = (beta(119)^2)

' Labor Force Participation (LFPR*)
{%modname}.append @state tlfpr = tlfpr(-1) + 0.25*(.95*gtlfpr(-1) + egtlfpr) +  etlfpr
{%modname}.append @state tlfpr1 = tlfpr(-1)
{%modname}.append @state gtlfpr = 0.95*gtlfpr(-1) + egtlfpr
{%modname}.append @ename etlfpr
{%modname}.append @evar var(etlfpr) = .05^2 ' beta(122)^2
{%modname}.append @ename egtlfpr
{%modname}.append @evar var(egtlfpr) = beta(123)^2

' Employment Rate (ER*), no drift
{%modname}.append @state terate = terate(-1) + eterate
{%modname}.append @state terate1 = terate(-1)
{%modname}.append @ename eterate
{%modname}.append @evar var(eterate) = beta(124)^2


{%modname}.append @vprior vpriors

{%modname}.append @mprior mpriors

freeze({%modname}_results) {%modname}.ml(m=100,showopts)

endsub
```

Defines:
ss_estimation, used in chunk 69.
Uses close 217a, ec 219, initial_values 83 216b, lfpr 219, ptr 219, sector 218, and sspace 217a.

4.4 Supply Filter

80 ⟨*supply filter* 80⟩≡ (216a)

```
' Program is part of the FRB/US state-space model package.
' frbus_supply_filter takes a previously estimated state-space model and
' generates estimates of model states.


' Subroutines:
'    To transform data
include "./data_transformations"


close @all
wfcreate kf_data q 1949:1 2020:4

%estend    = "2014q3"
%eststart  = "1963q2"
%datastart = "1949q1"

sample estsample  %eststart  %estend
sample datasample %datastart %estend

%modname = "ss_model"


'*******************************************************************************
' Retrieve variables from the database

' Definitions of series in database
'    See FRB/US model documentation for more complete descriptions

' XGDP   - GDP, cw 2009$
' XGDPN  - GDP
' XB   - BLS NFB output, 2009$
' XBN  - BLS NFB output
' XGDIN  - GDI

' PGDP   - Price index for GDP, cw
' PXB  - BLS NFB price

' LEP    - Employment in nonfarm business sector (employee and self-employed)
' LHP    - Aggregate labor hours, nonfarm business sector (employee and self-employed
' LUR    - Civilian unemployment rate (break adjusted)
' LF     - Civilian labor force (break adjusted)
' N16    - Noninstitutional population, aged 16 and over (break adjusted)
```

```
' KS     - Capital services, 2009 $
' LQUALT - Labor quality, trend level
' VEOA   - Average energy-output ratio of existing capital stock

' PCXFE  - Price index for personal consumption expendits ex. food and energy, cw (NIA def.)
' PCER   - Price index for personal consumption expenditures on energy (relative to PCXFE)

' PMO    - Price index for imports ex. petroleum, cw
' UCES   - Energy share of nominal consumption expenditures
' EMON   - Imports of goods and services ex. petroleum
' XGDEN  - Nominal Absorption
' PTR    - 10-year expected inflation (Hoey/Philadelphia survey)


smpl datasample
%dbin = "./state_space_data"
dbopen %dbin as histdata
fetch(d=histdata) *

'************************************************
' Make model observable series, set priors

smpl datasample

call data_transformations

'************************************************
' Load saved model; run estimation step with previously estimated
' parameters as starting values (converges quickly).

smpl estsample

wfopen saved_results
copy saved_results::untitled\saved_results kf_data::untitled\{%modname}
wfclose saved_results
freeze({%modname}_results) {%modname}.ml(m=100,showopts)


'************************************************************
' Create states

{%modname}.makestates(t=pred) *_prs        ' one-step ahead state predictions
{%modname}.makestates(t=filt) *_1          ' filter states
{%modname}.makestates(t=smooth) *_2        ' smoothed states
{%modname}.makestates(t=disturb) *err      ' estimate of the disturbances
```

```
{%modname}.makestates(t=filtse) *se        ' RMSE of the filtered states
{%modname}.makestates(t=smoothse) *se2     ' RMSE of the smoothed states

{%modname}.makesignals(t=pred) *_pr        ' one-step ahead signal predictions
{%modname}.makesignals(t=resid) *_res      ' error in one-step ahead signal predictio
{%modname}.makesignals(t=stdresid) *_sres  ' standardized one-step ahead prediction r


'*************************************************************
' Transformation into FRB/US mnemonics



series xbt  = exp((tmfp_2/.965 + 0.725*(terate_2 + tlfpr_2  + thtfactor_2 + tww_2) +
               0.275*lks + 0.725*llqualt + (.035/.965)*lveoa +lpop)/100)
series hmfpt  = gtmfp_2
series qlfpr  = exp(tlfpr_2/100)
series hqlfpr = (gtlfpr_2/400)*qlfpr(-1)
series hqlww  = gtww_2
series huxb   = gtotfactor_2
series leppot = exp((tlfpr_2+terate_2+thtfactor_2)/100) * n16
series lurnat = 100*(1-exp(terate_2/100))
series qlww   = exp(tww_2/100)
series uxbt   = exp(totfactor_2/100)
series xgdpt  = xbt * uxbt
series xgdo   = exp(cycle_2/100)*xgdpt
```

Uses close 217a, data_transformations 215c, dbopen 217a, ex 219, fetch 217a, frbus_supply_filter 216a, hmfpt 219, hqlfpr 219, hqlww 219, huxb 219, leppot 219, lurnat 219, n16 219, qlfpr 219, qlww 219, sector 218, uxbt 219, wfclose 217a, wfcreate 217a, wfopen 217a, xbt 219, xgdo 219, and xgdpt 219.

4.5 Initial Values

83 ⟨*initial values* 83⟩≡ (216b)

```
subroutine initial_values


coef(703) beta
coef(60) phi
beta(1)          = 1.497516
beta(2)          = -0.540927
beta(213)        = -0.328852
beta(214)        = 1.061453
beta(216)        = -0.266658
beta(10)         = 1.325310
beta(401)        = 0.625693
beta(404)        = 0.090059
beta(405)        = 0.440360
beta(406)        = -0.324208
beta(407)        = -0.435237
beta(408)        = 0.293555
beta(409)        = 0.253904
beta(6)          = 0.331366
beta(602)        = 0.927692

phi(20)          = 0.301616
phi(22)          = 0.020651
phi(30)          = 0.452297
phi(31)          = 0.649965
phi(40)          = 0.042475
phi(41)          = 0.727876
phi(50)          = 0.299981
phi(51)          = 0.528947

beta(105)        = 0.176896
beta(106)        = 0.086226
beta(107)        = 0.210419
beta(104)        = 0.229143
beta(100)        = 0.000001
beta(109)        = 0.803236
beta(111)        = 0.566205
beta(119)        = 0.099423
beta(123)        = 0.129325
beta(115)        = 0.140001
beta(117)        = 0.081062
beta(124)        = 0.136532
beta(118)        = 0.010000
```

```
  beta(122)       = 0.050000
  beta(114)       = 0.107852
  beta(112)       = 0.044151
  beta(126)       = 0.476765
  beta(125)       = 0.534807
  beta(11) = 1/beta(10)


  endsub
```

Defines:
initial_values, used in chunks 69 and 76.

Chapter 5

Data Only Package

Chapter 6

Appendices

Routine Libraries

6.1 Master Library

6.1.1 quarterly date string shift

89 ⟨*quarterly date string shift* 89⟩≡ (209b)

```
subroutine dateshift(string %indate, string %outdate, scalar qtrshift)

' this subroutine takes the quarterly date string in %indate (ie, "2001q1") and shifts
' it qtrshift quarters, returning the result in %outdate

  !dddd1   = @dateval(%indate,"yyyyfq")
  !dddd2   = @dateadd(!dddd1,qtrshift,"q")
  %outdate = @datestr(!dddd2,"yyyyfq")

  endsub
```

Defines:
`dateshift`, used in chunks 13, 25, 93, and 154.

6.1.2 copy series into group

90 ⟨*copy series into group* 90⟩≡ (209b)

```
subroutine group2group(string %fromgroup, string %togroup, string %to_type)

' copies a group of series into another group.
'
' If %to_type = "suffix", then %togroup is interpreted as a suffix to be applied to
' %fromgroup and its series.
'
' If %to_type = "prefix", then %togroup is interpreted as a prefix to be applied to
' %fromgroup and its series.
'
' If %to_type = "group", the %togroup is interpreted as the name of a group that
' already exists.

  if %to_type = "group" then
    if {%fromgroup}.@count <> {%togroup}.@count then
      statusline ERROR in GROUP2GROUP:  the two groups do not contain the same numbe
      endif
    for !ik1 = 1 to {%fromgroup}.@count
      %tmp = {%fromgroup}.@seriesname(!ik1)
      %tmp1 = {%togroup}.@seriesname(!ik1)
      {%tmp1} = {%tmp}
      next
    endif

  if %to_type = "suffix" then
    %tmpa = " "
    %tmpb = %fromgroup + %togroup
    for !ik1 = 1 to {%fromgroup}.@count
      %tmp = {%fromgroup}.@seriesname(!ik1) + %togroup
      %tmpa = %tmpa + " " + %tmp
      if @isobject(%tmp) then
        {%tmp} = {%fromgroup}(!ik1)
        else
        series {%tmp} = {%fromgroup}(!ik1)
        endif
      next
    group {%tmpb} {%tmpa}
    endif

  if %to_type = "prefix" then
    %tmpa = " "
    %tmpb = %togroup + %fromgroup
    for !ik1 = 1 to {%fromgroup}.@count
```

```
      %tmp =  %togroup + {%fromgroup}.@seriesname(!ik1)
      %tmpa = %tmpa + " " + %tmp
      if @isobject(%tmp) then
        {%tmp} = {%fromgroup}(!ik1)
        else
        series {%tmp} = {%fromgroup}(!ik1)
        endif
      next
    group {%tmpb} {%tmpa}
    endif

  endsub
```

Defines:
group2group, never used.

6.1.3 set group to zero

91a ⟨*set group to zero* 91a⟩≡ (209b)

```
  subroutine group2zero(string %group)

  ' set all series in an existin %group to zero

    for !ik1 = 1 to {%group}.@count
      %tmp = {%group}.@seriesname(!ik1)
      {%tmp} = 0
      next

  endsub
```

Defines:
group2zero, used in chunks 5, 9, 13, 19, 25, 33, 42, and 204.

6.1.4 names of all series in group

91b ⟨*names of all series in group* 91b⟩≡ (209b)

```
  subroutine groupnames2string(string %group, string %groupnames)

  ' creates a string of the names of all the series in a group

    %groupnames = " "
    for !ik1 = 1 to {%group}.@count
      %groupnames = %groupnames + " " + {%group}.@seriesname(!ik1)
      next

  endsub
```

Defines:
groupnames2string, used in chunk 42.

6.1.5 create new group

92 ⟨*create new group* 92⟩≡ (209b)

```
subroutine groupnew(string %fromgroup, string %to)

' Creates a new group.  The names of the new group and associated series are built up
' from %fromgroup with %to as a suffix.  Series that do not yet exist are set to zero

  %tmpa = " "
  %tmpb = %fromgroup + %to
  for !ik1 = 1 to {%fromgroup}.@count
    %tmp = {%fromgroup}.@seriesname(!ik1) + %to
    %tmpa = %tmpa + " " + %tmp
    if @isobject(%tmp) <> 1 then
      series {%tmp} = 0
      endif
    next
  group {%tmpb} {%tmpa}

endsub
```

Defines:
groupnew, used in chunks 5, 9, 13, 19, 25, 33, 42, and 204.

6.1.6 interpolate unavailable observations

93 ⟨*interpolate unavailable observations* 93⟩≡ (209b)

```
subroutine interp_lin(string %series_in, string %series_out, string %substart, string %subend)

' Subroutine that replaces NA values in a series with
' interpolated observations.  NA values at the beginning or end of the series are
' replaced with the first or last non-NA value.

smpl %substart %subend

' ******************************************************
' check that series is not all NAs
series tmp_check = ({%series_in} = NA)
if @sum(tmp_check) = @obssmpl then
  statusline Error in interp_lin subroutine:  interpolation cannot be performed because serie
  stop
  endif

series tmp_ser = ({%series_in}<>NA)
series tmp_id = @cumsum(tmp_ser)
series tmp_next = @sumsby({%series_in},tmp_id(-1))
series tmp_prev = @sumsby({%series_in},tmp_id)

' ******************************************************
' check for NAs at either beginning or end of sample

' test for NAs at beginning of sample
series tmp_naprev = (tmp_prev = NA)
!flag_prev = @max(tmp_naprev)
if !flag_prev = 1 then
  'at this point, tmp_next will have an undesired NA in its first observation;
  'change it to equal its second observation
  smpl %substart %substart
  call dateshift(%substart,%nextqtr,1)
  tmp_next = @elem(tmp_next,%nextqtr)
  smpl %substart %subend if (tmp_prev = NA)
  tmp_prev = tmp_next
  smpl %substart %subend
  endif

' test for NAs at end of sample
series tmp_nanext = (tmp_next = NA)
!flag_next = @max(tmp_nanext)
if !flag_next = 1 then
  smpl %substart %subend if (tmp_next = NA)
```

```
    tmp_next = tmp_prev
    smpl %substart %subend
    endif


  ' ***************************************************
  series tmp_lambda = (@obsid-@minsby(@obsid,tmp_id))/@sumsby(1,tmp_id)
  series {%series_out} = tmp_lambda*tmp_next + (1-tmp_lambda)*tmp_prev


  delete tmp_ser tmp_prev tmp_next tmp_lambda tmp_id tmp_nanext tmp_naprev tmp_check

endsub
```

Defines:
interp_lin, used in chunk 42.
Uses dateshift 89.

6.1.7 set fiscal policy option

94 ⟨*set fiscal policy option* 94⟩≡ (209b)

```
subroutine set_fp(string dfpxxx)

  %policy_options = "dfpex dfpsrp dfpdbt"
  %dfpxxx = @lower(dfpxxx)
  %dfpxxx = @replace(%dfpxxx," ","")

  !kz = @wfind(%policy_options,%dfpxxx)
  if !kz > 0 then
    for !izzz = 1 to @wcount(%policy_options)
      %ppp = @word(%policy_options,!izzz)
      if !izzz = !kz then
        series {%ppp} = 1
        else
        series {%ppp} = 0
        endif
      next
    else
    %err = %dfpxxx + " is not a valid fiscal policy option; execution terminated"
    @uiprompt(%err)
    stop
    endif

endsub
```

Defines:
set_fp, used in chunks 5, 9, 13, 19, 25, 33, and 42.
Uses dfpdbt 219, dfpex 219, and dfpsrp 219.

6.1.8 set monetary policy option

95 ⟨*set monetary policy option* 95⟩≡ (209b)

```
subroutine set_mp(string dmpxxx)

  %policy_options = "dmpex dmprr dmptay dmptlr dmpintay dmpalt dmpgen"
  %dmpxxx = @lower(dmpxxx)
  %dmpxxx = @replace(%dmpxxx," ","")

  !kz = @wfind(%policy_options,%dmpxxx)
  if !kz > 0 then
    for !izzz = 1 to @wcount(%policy_options)
      %ppp = @word(%policy_options,!izzz)
      if !izzz = !kz then
        series {%ppp} = 1
        else
        series {%ppp} = 0
        endif
      next
    else
    %err = %dmpxxx + " is not a valid monetary policy option; execution terminated"
    @uiprompt(%err)
    stop
    endif

endsub
```

Defines:
set_mp, used in chunks 5, 9, 13, 19, 25, 33, and 42.
Uses dmpalt 219, dmpex 219, dmpgen 219, dmpintay 219, dmprr 219, dmptay 219, and dmptlr 219.

6.1.9 set monetary policy fed funds rate

96 ⟨*set monetary policy fed funds rate* 96⟩≡ (209b)

```
subroutine set_mpvars2rff

  rfffix = rff
  rfftay = rffe
  rfftlr = rffe
  rffalt = rff
  rffintay = rffe
  rffgen = rffe
  rrfix = rffe - @movav(picxfe,4)

endsub
```

Defines:
set_mpvars2rff, never used.
Uses picxfe 219, rff 219, rffalt 219, rffe 219, rfffix 219, rffgen 219, rffintay 219, rfftay 219, rfftlr 219, and rrfix 219.

6.2 mce solve library

6.2.1 mce solve library change history

97 ⟨*mce solve library change history* 97⟩≡ (211)

```
' Changes (1/25/14)
'
' 1.  Removed defaults from mce_load_frbus subroutine
'
' 2.  Added make_frbus_mcevars subroutine
'
' Changes (1/22/14)
'
' 1.  Added code so that the _$_sufsim string variable is assigned the
'     alias of the currently active scenario when the %mopts argument
'     is a null string.
'
' 2.  In subroutine mcz_sim, put the contents of string variable
'     mcz_sim_options into another string, a change which for unknown
'     reasons eliminates an unexplanined Eviews shutdown when running
'     a pair of simulations of which the first is type=single and the
'     second is type=opt.

' Changes (1/8/14)
'
' 1.  Added the "dontstop" option to the %sopt argument.  When invoked,
'     this option causes the eviews program that calls mce_run
'     to continue executing when running a type=single simulation if
'     (a) the solution iterations do not
'     converge within the maximum number of permitted iterations or
'     (b) the eviews solver generates an error when solving either
'     the backward-looking or forward-looking models in subroutine
'     mcz_solvit.  In the case of nonconvergence the call to mce_run
'     terminates with the string value %mce_finish = "no".  In the
'     case of a solver error, the call to mce_run terminates with the
'     string value %mce_finish = "failed_solve".  Otherwise,
'     %mce_finish = "yes".
'
'
' Changes (2/21/12)
'
' 1. Fixed problem with code for unconstrained TC policy
'    calculated via EViews matrices -- the solution at period t
'    (t = 1, ... !ndrv) must go out through period
'    the farthest period ever solved -- (!nevl + !ndrv - 1) --
'    which requires that the opt derivative
```

```
' matrix must span this many quarters.  The constrained TC
'
' 2. Wrote code for constrained TC policy in EViews when there is
'    a single instrument
'
' 3. Fixed code for constrained TC policy in R -- still need to do matlab

' Changes (2/16/12)
'
' 1. Fixed problem with constrained optimization when
'    number of evaluation periods is not the same as
'    the number of instrument periods
'    (subroutine mcz_opt_qp).  This undoes part of the
'    1/17/12 change #3
'
' Changes (2/13/12)
'
' 1. Added "d=" option for TC damping factor (!tcdamp)

' Changes (1/30/12)
'
' 1. Modified subroutine mca_opt_qp to set options in matlab
'    quadprog function call
'
' Changes (1/17/12)
'
' 1. Added new subroutine (mcz_opt_tc) and new simtype (opttc)
' 2. Added new keywords:  ideriv, for sopt string
'                /xopen, /xclose for R and matlab
' 3. Modified subroutine mcz_opt_qp so that the dimensions of the
'    initial and transformed constraint matrices are based on !ndrv not !nevl
' 4. Dropped the explicit optqp simtype
'
```

Uses make_frbus_mcevars 175, mce_load_frbus 171, mce_run 99, mcz_opt_qp 149, mcz_opt_tc 154, mcz_sim 116, and mcz_solvit 125.

6.2.2 run model consistent expectations

99 ⟨*run model consistent expectations* 99⟩≡ (211)

```
subroutine mce_run(string m_opts, string a_opts, string s_opts)

' Driver program

%mcestart = @word(@pagesmpl,1)
%mceend = @word(@pagesmpl,2)


' ***********************************************
' 1. Examine m_opts string (defines or creates models, mce errors and instruments)
' ***********************************************
'
' Case 1: null string (ie, "")
'         => use existing models whose names are contained in %_$_mod_b and
'            %_$_mod_f; use existing objects _$_mce_instrus
'            and _$_mce_errs.
' Case 2: string contains the keywords "create" and "mod=<modname>"
'         => a model named <modname> is in the workfile and contains explicit
'            leads; parse it to create the objects _$_mod_b, _$_mod_f,
'            _$_mce_instrus, and _$_mce_errs
' Case 3: string contains the keywords "mod_b=", "mod_f=", "mce_errs=", "mce_instrus=";
'         or the keywords "mod_b=", "mod_f=", "mce_vars="
'         => each keyword must be assigned to a string variable, whose contents are
'            used to define _$_mod_b, _$_mod_f, _$_mce_errs, and
'            _$_mce_instrus


if @isempty(m_opts) = 1 then
' ************************************
' case 1 code
  !m_case = 1
  !z1 = @isobject(%_$_mod_b)
  !z2 = @isobject(%_$_mod_f)
  !z3 = @isobject("_$_mce_instrus")
  !z4 = @isobject("_$_mce_errs")
  !zsum = !z1+!z2+!z3+!z4
  if !zsum <> 4 then
    %err = "When the first argument to mce_run is a null string, a previous call to "
    %err = %err + "mce_run must have placed the names of the backword and "
    %err = 5err + "forward looking models, list of mce instruments names "
    %err = %err + "and list of mce error names in various strings; at least one "
    %err = %err + "these strings either does not exist or contains the name of "
    %err = %err + "an object that does not exist.  Execution terminates."
```

```
  @uiprompt(%err)
  stop
  endif
'find alias of active scenario
%endog_active = {%_$_mod_b}.@endoglist("@active")
%endog_actual = {%_$_mod_b}.@endoglist
%word1_active = @word(%endog_active,1)
%word1_actual = @word(%endog_actual,1)
!a1 = @strlen(%word1_active)- @strlen(%word1_actual)
string _$_sufsim = @right(%word1_active,!a1)

else
m_opts = @lower(m_opts)
m_opts = @replace(m_opts," ","")
m_opts = @replace(m_opts,","," ")
m_opts = " " + m_opts + " "

if @instr(m_opts,"create") and @instr(m_opts,"mod=") then
' ****************************************
' case 2 code
  !m_case = 2
  call mcz_equalopt("mod",m_opts)
  if @len(%temp)>0 then
    %temp1 = @left(%temp,1)
    if %temp1 = "%" then
      %mod  = @lower({%temp})
      else
      %mod  = @lower(%temp)
      endif
    call mcz_parsemod({%mod})
    endif
  else
  !z1 = @instr(m_opts,"mod_b=")
  !z2 = @instr(m_opts,"mod_f=")
  !z3 = @instr(m_opts,"mce_errs=")
  !z4 = @instr(m_opts,"mce_instrus=")
  !z5 = @instr(m_opts,"mce_vars=")
  !zsum1 = (!z1>0)+(!z2>0)+(!z3>0)+(!z4>0)
  !zsum2 = (!z1>0)+(!z2>0)+(!z5>0)
  if !zsum1 = 4 or !zsum2 = 3 then
  ' ****************************************
  ' case 3 code
    !m_case = 3
    call mcz_equalopt("mod_b",m_opts)
    if @len(%temp)>0 then
      %temp = @lower({%temp})
```

```
        %_$_mod_b = %temp
        endif
      call mcz_equalopt("mod_f",m_opts)
      if @len(%temp)>0 then
        %temp = @lower({%temp})
        %_$_mod_f = %temp
        endif
      call mcz_equalopt("mce_errs",m_opts)
      if @len(%temp)>0 then
        %temp = @lower({%temp})
        group _$_mce_errs {%temp}
        endif
      call mcz_equalopt("mce_instrus",m_opts)
      if @len(%temp)>0 then
        %temp = @lower({%temp})
        group _$_mce_instrus {%temp}
        endif

      else
    ' ************************************
    ' m_opt string does not conform to a valid case
      @uiprompt("first argument to subroutine mce_run is incorrectly specified")
      stop
      endif
    endif
  endif

if !m_case = 2 or !m_case = 3 then
  call mcz_hasopt("adds",m_opts)
  if !hasflag = 1 then
    {%_$_mod_b}.addassign @all
    {%_$_mod_f}.addassign @all
    endif
  call mcz_hasopt("track",m_opts)
  if !hasflag = 1 then
    %track = "yes"
    %track_start = @word(@pagesmpl,1)
    %track_end = @word(@pagesmpl,2)
    call mcz_equalopt("tstart",m_opts)
    if @len(%temp)>0 then
      %track_start  = @lower(%temp)
      endif
    call mcz_equalopt("tend",m_opts)
    if @len(%temp)>0 then
      %track_end  = @lower(%temp)
      endif
```

```
      smpl %track_start %track_end
      {%_$_mod_b}.addinit(v=n) @all
      {%_$_mod_f}.addinit(v=n) @all
      endif
    endif

  if !m_case = 2 then
    group _$_mce_errs {{%mod}_targs}
    group _$_mce_instrus {{%mod}_instrus}
    endif

  if !m_case = 3 then
    call mcz_equalopt("mce_vars",m_opts)
    if @len(%temp)>0 then
      %temp = @lower({%temp})
      %errs = @wcross("e",%temp)
      group _$_mce_errs {%errs}
      %instrus = @wcross(%temp,"_a")
      group _$_mce_instrus {%instrus}
      endif
    endif

' ************************************************
' 2.  Examine a_opts string (specifies mce algorithm)
' ************************************************
'
' Case 1:  blank string and %existing_algos = "yes"
'          => do not call mcz_algo (use existing settings)
' Case 2:  blank string and %existing_algos <> "yes"
'          => call mcz_algo to set default options
' Case 3:  nonblank string
'          => call mcz_algo using string to set options overrides

  if @isempty(a_opts) = 1 then
    if %existing_algos = "yes" then
      !a_case = 1
      else
      !a_case = 2
      call mcz_algo(%_$_mod_b,%_$_mod_f," ",_$_mce_instrus,_$_mce_errs)
      endif
    else
    !a_case = 3
    call mcz_algo(%_$_mod_b,%_$_mod_f,a_opts,_$_mce_instrus,_$_mce_errs)
    endif
```

```
' ************************************************
' 3. Examine s_opts string (specifies type of simulation)
' ************************************************

  if @isempty(s_opts) = 1 then
    @uiprompt("error:  no simulation action requested")
    stop
    endif

' make a copy of s_opts for parsing within this subroutine
' (the original is passed to other subroutines for additional
' parsing)
  string _$_opts = s_opts
  _$_opts = @lower(_$_opts)
  _$_opts = @replace(_$_opts," ","")
  _$_opts = @replace(_$_opts,","," ")
  _$_opts = " " + _$_opts + " "

' check for keywords that pertain to all simulation types
  call mcz_hasopt("scen",_$_opts)
  if !hasflag = 1 then
    call mcz_equalopt("suf",_$_opts)
    if @len(%temp)>0 then
      string _$_sufsim  = @lower(%temp)
      else
      string _$_sufsim = "_1"
      endif
    %sufsim = _$_sufsim
    %scenname = "mce_sim" + _$_sufsim
    {%_$_mod_b}.scenario(n,a=%sufsim) %scenname
    {%_$_mod_f}.scenario(n,a=%sufsim) %scenname
    endif
  call mcz_equalopt("solveopt",_$_opts)
  {%_$_mod_b}.solveopt(o=n,g=12,z=1e-12)
  {%_$_mod_f}.solveopt(o=n,g=12,z=1e-12)
  if @len(%temp)>0 then
    {%_$_mod_b}.solveopt({{%temp}})
    {%_$_mod_f}.solveopt({{%temp}})
    endif
  call mcz_equalopt("txt",_$_opts)
  if @len(%temp)>0 then
    for !j = 1 to {%temp}.@linecount
      %tmp = @lower({%temp}.@line(!j))
      {%tmp}
      next
    endif
```

```
  !mceshow = 1
  call mcz_equalopt("o",_$_opts)
  if @len(%temp)>0 then
    !mceshow = @val(%temp)
    endif
  call mcz_equalopt("sstart",_$_opts)
  if @len(%temp)>0 then
    %mcestart = @lower(%temp)
    endif
  call mcz_equalopt("send",_$_opts)
  if @len(%temp)>0 then
    %mceend = @lower(%temp)
    endif
  smpl %mcestart %mceend
  !nqtrs = @obssmpl
  call mcz_hasopt("cleanup",_$_opts)
  if !hasflag = 1 then
    %cleanup = "yes"
    else
    %cleanup = "no"
    endif
  call mcz_hasopt("dontstop",_$_opts)
  if !hasflag = 1 then
    %dontstop = "yes"
    if @maxerrs - @errorcount < 2 then
      !tt = @errorcount + 2
      setmaxerrs !tt
      endif
    else
    %dontstop = "no"
    endif

' create various program variables and objects that are needed by
' all simulation types; compute initial Jacobian in some cases
  call mcz_sim_setup
  if %mcz_sim_setup = "err" then
    %mce_finish = "failed_solve"
    return
    endif

' determine simulation type
  call mcz_equalopt("type",_$_opts)
  if @len(%temp) > 0 then
    %simtype = @lower(%temp)
    if %simtype = "single" then
      call mcz_sim(_$_opts)
```

```
    if %mcz_sim = "err" then
      %mce_finish = "failed_solve"
      return
      endif
    else
    if %simtype = "opt" or %simtype = "opttc" then
      call mcz_equalopt("targs",_$_opts)
      if @len(%temp)>0 then
        %targs = @lower(%temp)
        else
        @uiprompt("error:  targs keyword is missing")
        stop
        endif
      call mcz_equalopt("instrus",_$_opts)
      if @len(%temp)>0 then
        %instrus = @lower(%temp)
        else
        @uiprompt("error:  instrus keyword is missing")
        stop
        endif
      call mcz_equalopt("cnstr",_$_opts)
      if @len(%temp)>0 then
        %cnstr = @lower(%temp)
        %cnstrflag = "yes"
        else
        text _$_blanktext
        %cnstr = "_$_blanktext"
        %cnstrflag = "no"
        endif
      call mcz_opt_setup(s_opts,{%instrus},{%targs},{%cnstr})
      else
      @uiprompt("error:  invalid simtype")
      stop
      endif
    endif
  else
  @uiprompt("error:  required simtype keyword not found")
  stop
  endif

endsub
```

Defines:
mce_run, used in chunks 9, 19, 25, 33, 51, 54, 57, 61, 64, 97, 114, and 116.
Uses mcz_algo 109, mcz_equalopt 113a, mcz_hasopt 113b, mcz_opt_setup 138, mcz_parsemod 106, mcz_sim 116, and mcz_sim_setup 114.

6.2.3 determine endogenous and exogenous variables

106 ⟨*determine endogenous and exogenous variables* 106⟩≡ (211)

```
subroutine mcz_parsemod(model modo)
'
' This subroutine takes modo, a model with explicit leads, and creates
' four objects.  The name of each created object starts with the name of the
' input model, which is denoted by <modname>.
'
' 1.  Model _$_<modname>_b is the same as modo but with all leads replaced
'     with exogenous variables
' 2.  Model _$_<modname>_f contains the MCE error equations
' 3.  String <modname>_instrus of the names of the added exogenous variables,
'     which are the instruments to be used to drive the MCE errors to zero
' 4.  String <modname>_targs of the names of the endogenous variables in
'     <modname>_f, which are the names of MCE error variables

' preliminaries
 %ok_chars = "=-+*(^ "
 freeze(_$_modtext) modo.text
 string _$_endog = @lower(modo.@endoglist)
 !nvars = @wcount(_$_endog)
 %leadnames = " "
 %errnames = " "

 %mm = @lower(modo.@name)
 %_$_mod_b = "_$_" + %mm + "_b"
 %_$_mod_f = "_$_" + %mm + "_f"
 model {%_$_mod_b}
 model {%_$_mod_f}


' create model _$_<modname>_b
 smpl @all
 for !i2 = 1 to _$_modtext.@linecount
     %tmp2 = @ltrim(@lower(_$_modtext.@line(!i2)))
     %aa = @left(%tmp2,1)
     if %aa <> "@" then
         for !i1 = 1 to !nvars
             !occurrence = 1
             %tmp1 = @word(_$_endog,!i1) + "("
             !kk = @instr(%tmp2,%tmp1)
             if !kk > 0 then
                 while !kk > 0
                     'three possibilities
                     '  1. it is part of another variable name
```

```
                        '  2. it is a lag
                        '  3. it is a lead
                        '
                        'look at character in %tmp2 just before %tmp1 to make sure that %tmp1
                        'is not part of a longer variable or coefficient name
                        %before = @mid(%tmp2, !kk-1,1)
                        if @instr(%ok_chars,%before) > 0 then
                            %tmp3 = @mid(%tmp2,!kk)
                            !kkll = @instr(%tmp3,"(")
                            !kkrr = @instr(%tmp3,")")
                            %laglead = @mid(%tmp3,!kkll, !kkrr-!kkll+1)
                            !ll = 0 + {%laglead}
                            if !ll <= 0 then   'it is a lag -- skip it
                               !occurrence = !occurrence + 1
                               endif
                            if !ll > 0 then   'it is a lead -- define a new variable
                                %aaa = @word(_$_endog,!i1) + %laglead
                                %bbb = @word(_$_endog,!i1) + "_ld_" + @str(!ll)
                                series {%bbb} = {%aaa}
                                %tmp4 = @replace(%tmp2,%aaa,%bbb,1)
                                %tmp2 = %tmp4
                                %leadnames = %leadnames + " " + %bbb
                                endif
                            else    'it is part of another variable name -- skip it
                            !occurrence = !occurrence + 1
                            endif
                        !kk = @instr(%tmp2,%tmp1,!occurrence)
                        wend
                    endif
                next
            endif
          {%_$_mod_b}.append {%tmp2}
        next

' create _$_<modname>_f
  %leadnames = @wunique(%leadnames)
  'smpl @all
  for !i1 = 1 to @wcount(%leadnames)
      %tmp1 = @word(%leadnames,!i1)
      !k1 = @instr(%tmp1,"_ld_")
      %tmp2 = @left(%tmp1,!k1-1)
      %tmp3 = @mid(%tmp1,!k1+4)
      %tmp4 =  "err_" + %tmp2 + "_" + @str(%tmp3)
      series {%tmp4} = 0
      %errnames = %errnames + " " + %tmp4
      %eqstring = %tmp4 + "=" + %tmp1 + "-" + %tmp2 + "(" + @str(%tmp3) + ")"
```

```
        {%_$_mod_f}.append {%eqstring}
        next

  ' create mce instrument and target strings

    string {%mm}_instrus = %leadnames
    string {%mm}_targs = %errnames


  endsub
```

Defines:
mcz_parsemod, used in chunk 99.

6.2.4 determine default method, linesearch, and other options

109 ⟨*determine default method, linesearch, and other options* 109⟩≡ (211)

```
subroutine mcz_algo(string mcz_mod_b, string mcz_mod_f, string mcz_algo_opts, group mcz_instr

%existing_algos = "yes"

mcz_algo_opts = @lower(mcz_algo_opts)
mcz_algo_opts = @replace(mcz_algo_opts," ","")
mcz_algo_opts = @replace(mcz_algo_opts,","," ")
mcz_algo_opts = " " + mcz_algo_opts + " "

' default values for method options
%meth = "newton"
%jinit = "interp"
!nskip = 12
!jtrigger = .5

' default values for linesearch options
%linemeth = "armijo"
!linetrigger  = .9
!mcelinemax = 10
!lambda = 1.0
!lrat = .5
!mcz_step_max = 1.0

' default values for other options
!mceconv = .00001
!mcemaxiter = 20
!mceptrb = .001
!broymax = 600

' just in case
%terminal = "no"

' are there overrides to defaults?
if @len(mcz_algo_opts) > 0 then

  ' *********************************
  ' look for method option
    call mcz_equalopt("meth",mcz_algo_opts)
    if @len(%temp)>0 then
      %meth  = @lower(%temp)
      if %meth = "broy" then
        %jinit = "bd"
```

```
      %jupdate = "na"
      %linemeth = "lmr"
      !mcemaxiter = 200
      endif
    if %meth = "qnewton" then
      %jinit = "bd"
      %jupdate = "na"
      %linemeth = "lmr"
      !mcemaxiter = 200
      call mcz_equalopt("broymax",mcz_algo_opts)
      if @len(%temp)>0 then
        !broymax = @val(%temp)
        endif
      endif
    if %meth = "ft" then
      %jinit = "identity"
      %jupdate = "na"
      %linemeth = "na"
      !mcemaxiter = 500
      endif
    endif

' *********************************
' look for jinit and jt options
  call mcz_equalopt("jinit",mcz_algo_opts)
  if @len(%temp)>0 then
    %temp = @lower(%temp)
    if @instr(%temp,"interp(") then
      !k1 = @instr(%temp,"(")
      !nskip = -@val(@mid(%temp,!k1))
      %tmp1 = @mid(%temp,!k1)
      %jinit = "interp"
      else
      %jinit = @lower(%temp)
      endif
    endif
  call mcz_equalopt("jt",mcz_algo_opts)
  if @len(%temp)>0 then
    !jtrigger  = @val(%temp)
    endif

' *********************************
' look for jupdate option
  %jupdate = %jinit
  call mcz_equalopt("jupdate",mcz_algo_opts)
  if @len(%temp)>0 then
```

```
    %temp = @lower(%temp)
    if @instr(%temp,"interp(") then
      !k1 = @instr(%temp,"(")
      !nskip = -@val(@mid(%temp,!k1))
      %tmp1 = @mid(%temp,!k1)
      %jupdate = "interp"
      else
      %jupdate = @lower(%temp)
      endif
    endif

' *********************************
' look for options related to linesearch
  call mcz_equalopt("lmeth",mcz_algo_opts)
  if @len(%temp)>0 then
    %linemeth  = @lower(%temp)
    endif
  call mcz_equalopt("lt",mcz_algo_opts)
  if @len(%temp)>0 then
    !linetrigger = @val(%temp)
    endif
  call mcz_equalopt("lmax",mcz_algo_opts)
  if @len(%temp)>0 then
    !mcelinemax = @val(%temp)
    endif
  call mcz_equalopt("lambda",mcz_algo_opts)
  if @len(%temp)>0 then
    !lambda = @val(%temp)
    endif
  call mcz_equalopt("stepmax",mcz_algo_opts)
  if @len(%temp)>0 then
    !mcz_step_max = @val(%temp)
    endif


' *********************************
' look for other options
  call mcz_equalopt("c",mcz_algo_opts)
  if @len(%temp)>0 then
    !mceconv = @val(%temp)
     endif
  call mcz_equalopt("m",mcz_algo_opts)
  if @len(%temp)>0 then
    !mcemaxiter = @val(%temp)
    endif
  call mcz_equalopt("p",mcz_algo_opts)
```

```
    if @len(%temp)>0 then
      !mceptrb = @val(%temp)
      endif

    endif


' ***********************************
' verify the MCE instrument and error arguments

  !nmceinstrus = _$_mce_instrus.@count
  !nmcetargs = _$_mce_errs.@count

  if !nmcetargs <> !nmceinstrus then
    @uiprompt("Error:  There must be as many mce errors as there are mce instruments
    stop
    endif

' ***********************************
' check that mce instruments are exogenous variables or add factors
' in the lag model
  %exog_vnames = {%_$_mod_b}.@exoglist
  %adds_vnames = {%_$_mod_b}.@addfactors
  %exog_vnames = %exog_vnames + " " + %adds_vnames
  %endog_vnames = {%_$_mod_b}.@endoglist
  for !i = 1 to _$_mce_instrus.@count
    %vvv = _$_mce_instrus.@seriesname(!i)
    !cc = @wfindnc(%exog_vnames,%vvv)
    if !cc = 0 then
      %errstring = "mce control variable " + %vvv + " is not an exogenous variable o
      @uiprompt(%errstring)
      stop
      endif
    next

' ***********************************
' check that mce errors are endogenous variables in the lead model
  %endog_lnames = {%_$_mod_f}.@endoglist
  for !i = 1 to _$_mce_errs.@count
    %vvv = _$_mce_errs.@seriesname(!i)
    !cc = @wfindnc(%endog_lnames,%vvv)
    if !cc = 0 then
      %errstring = "mce error variable " +  %vvv + " is not an endogenous variable i
      @uiprompt(%errstring)
      stop
      endif
```

```
    next

  endsub
```

Defines:
mcz_algo, used in chunk 99.
Uses mcz_equalopt 113a.

6.2.5 parse options containing equal signs

113a ⟨*parse options containing equal signs* 113a⟩≡ (211)

```
  subroutine mcz_equalopt(string optionstext,string opts)

' parse an option that contains an "=" sign
  optionstext = " " + optionstext + "="
  !k10 = @instr(opts,optionstext)
  if !k10 > 0 then
    !k11 = @len(optionstext)
    %tmp10 = @mid(opts,!k10+!k11)
    !k12 = @instr(%tmp10," ")
    %temp = @left(%tmp10,!k12-1)
    else
    %temp = ""
    endif

  endsub
```

Defines:
mcz_equalopt, used in chunks 99, 109, 116, 138, and 171.

6.2.6 parse options not containing equal signs

113b ⟨*parse options not containing equal signs* 113b⟩≡ (211)

```
  subroutine mcz_hasopt(string optionstext,string opts)

' parse an option that does not contain an "=" sign
  !k10 = @wfind(opts,optionstext)
  if !k10 > 0 then
    !hasflag = 1
    else
    !hasflag = 0
    endif

  endsub
```

Defines:
mcz_hasopt, used in chunks 99, 116, and 138.

6.2.7 create common variables, strings, matrices, vectors, and tables

114 ⟨*create common variables, strings, matrices, vectors, and tables* 114⟩≡ (211)

```
subroutine mcz_sim_setup

' create various program variables, strings, matrices, vectors, and tables
' that are common to all simulation types

!tmcetargs = !nmcetargs * !nqtrs
!tmceinstrus = !nmceinstrus * !nqtrs
%mce_targ_names = _$_mce_errs.@members
%mce_instru_names = _$_mce_instrus.@members

string _$_mod_f_exog = @lower({%_$_mod_f}.@exoglist)
string _$_mod_b_endog = @lower({%_$_mod_b}.@endoglist)
%fvars = @wintersect(_$_mod_f_exog,_$_mod_b_endog)
{%_$_mod_f}.override {%fvars}

!re_counter = 0


' ***************************************************************************
' create additional matrix/vector objects

vector(!mcemaxiter+1) _$_mce_loss_vec
vector(!mcemaxiter+1) _$_mce_conv_vec
vector( !nmceinstrus*!nqtrs) _$_mce_direction = 0
vector( !nmceinstrus*!nqtrs) _$_mce_instru_vec = 0
vector(!nmcetargs*!nqtrs) _$_mce_gap_vec
matrix _$_mce_ptrb_mat = @filledmatrix(!nqtrs,!nmceinstrus,!mceptrb)
matrix(!nmcetargs*!nqtrs,1) _$_mce_targ_vec
matrix(!tmcetargs,1) _$_mce_targ_dvec


' ***************************************************************************
'  misc

table(!mcemaxiter+2,6) mce_sim_stats


' ***************************************************************************
' compute initial jacobian except when it is an identity matrix
' or when its been created by a previous call to mce_run
if %jinit <> "identity" and !a_case <> 1 then
  !mcetry = 1
```

```
      smpl %mcestart %mceend
      call mcz_solvit
      if %mcz_solvit = "err" then
        %mcz_sim_setup = "err"
        return
        endif

      call mcz_derivs
      endif


    endsub
```

Defines:
  mcz_sim_setup, used in chunk 99.
Uses mce_run 99, mcz_derivs 127, and mcz_solvit 125.

6.2.8 model consistent coefficient simulation

116 ⟨*model consistent coefficient simulation* 116⟩≡ (211)

```
subroutine mcz_sim(string mcz_sim_options)

%mcz_sim = "ok"

' ************************************************
' 1. set options based on defaults and overrides in string mcz_sim_options
' ************************************************

' for some unknown reason, sometimes eviews bombs unless mcz_sim_options
' is assigned to another string before processing it
%mso = mcz_sim_options

%mso = @lower(%mso)
%mso = @replace(%mso," ","")
%mso = @replace(%mso,","," ")
%mso = " " + %mso + " "

%terminal = "no"
%mcevars_b = " "
%mcevars_f = " "

if @len(%mso) > 0 then
  call mcz_hasopt("terminal",%mso)
  if !hasflag = 1 then
    %terminal = "yes"
    call mcz_equalopt("mcevars_b",%mso)
    if @len(%temp)>0 then
      %mcevars_b  = @lower(%temp)
      call mcz_equalopt("mcevars_f",%mso)
      if @len(%temp)>0 then
        %mcevars_f  = @lower(%temp)
        if @wcount(%mcevars_b) <> @wcount(%mcevars_f) then
          %estring = "Error: mcevars_b and mcevars_f have different numbers of vari
          @uiprompt(%estring)
          stop
          endif
        endif
      endif
    endif
  endif

if %linemeth = "lmr" then
  !mhistory = 4
```

```
    !tmin = .1
    !tmax = .5
    !gammak = 10^(-4)
    endif


' ***********************************************
' 2. set up table of iteration-by-iteration statistics
' ***********************************************

  if @isobject("mce_sim_stats") then
    delete mce_sim_stats
    endif
  table(!mcemaxiter+2,6) mce_sim_stats
  mce_sim_stats.setwidth(1:1) 6
  mce_sim_stats.setwidth(2:6) 11
  mce_sim_stats.setlines(a2:f2) +b
  setcell(mce_sim_stats,1,1,"iter")
  setcell(mce_sim_stats,1,2,"converge")
  setcell(mce_sim_stats,2,2,"stat")
  setcell(mce_sim_stats,1,3,"SSR")
  setcell(mce_sim_stats,2,3,"stat")
  setcell(mce_sim_stats,1,4,"step")
  setcell(mce_sim_stats,2,4,"length")
  setcell(mce_sim_stats,1,5,"step")
  setcell(mce_sim_stats,2,5,"iters")
  setcell(mce_sim_stats,1,6,"Newton MCE")
  setcell(mce_sim_stats,2,6,"deriv's?")


' ***********************************************
' 3. information text file
' ***********************************************

  if @isobject("mce_sim_text") then
    delete mce_sim_text
    endif
  text mce_sim_text
  mce_sim_text.append Simulation start = {%mcestart}
  mce_sim_text.append Simulation end   = {%mceend}
  mce_sim_text.append MCE method = %meth
  if %meth = "newton" then
    mce_sim_text.append -- Initial jacobian = %jinit
    if %jinit = "interp" then
      mce_sim_text.append ---- Jacobian interpolation parameter = {!nskip}
      endif
```

```
    mce_sim_text.append  -- Recompute Jacobian based on jtrigger = {!jtrigger}
    mce_sim_text.append  -- Recompute jacobian using method = {%jupdate}
    endif
  if %meth = "broy" or %meth = "qnewton" then
    mce_sim_text.append -- Initial Jacobian approximation = %jinit
    if %jinit = "interp" then
      mce_sim_text.append ---- Interpolation parameter = {!nskip}
      endif
    if %meth = "qnewton" then
      mce_sim_text.append ---- QNewton iteration switch = {!broymax}
      endif
    endif
  if %meth = "ft" then
    mce_sim_text.append -- Fixed step size =   {!lambda}
    endif
  mce_sim_text.append Linesearch method = {%linemeth}
  if %linemeth <> "na" then
    mce_sim_text.append -- Linesearch trigger = {!linetrigger}
    mce_sim_text.append -- Maximum linesearch iterations =   {!mcelinemax}
    endif
  mce_sim_text.append Convergence criteria = {!mceconv}
  mce_sim_text.append Maximum number of MCE iterations = {!mcemaxiter}
  mce_sim_text.append MCE instrument perturbation factor = {!mceptrb}
  mce_sim_text.append Intermediate output level factor = {!mceshow}

  mce_sim_text.append MCE instrument variables  = {%mce_instru_names}
  mce_sim_text.append MCE error variables  = {%mce_targ_names}
  mce_sim_text.append There are {!tmceinstrus} instrument and {!tmcetargs} error obse

  !re_counter = !re_counter + 1

' ************************************************
' 4. solution iterations
' ************************************************

  '**********************************
  ' initialize counters, switches, etc.

  !mcetry = 0
  %mce_converge = "no"
  smpl %mcestart %mceend
  _$_mce_instru_vec = @vec(@convert(_$_mce_instrus))

  '**********************************
  ' start of iteration loop
```

```
  while !mcetry <= !mcemaxiter and %mce_converge = "no"

    !mcetry = !mcetry + 1
    setcell(mce_sim_stats,!mcetry+2,1,!mcetry-1,0)

    vector _$_instru_prev = _$_mce_instru_vec
    vector _$_gap_prev = _$_mce_gap_vec

    !mcz_step = !mcz_step_max
    call mcz_solvit
    if %mcz_solvit = "err" then
      %mcz_sim = "err"
      return
      endif

    if !mcetry > 1 then
      !gamma =  _$_mce_loss_vec(!mcetry)/_$_mce_loss_vec(!mcetry-1)
      !loss_prev = _$_mce_loss_vec(!mcetry-1)
      setcell(mce_sim_stats,!mcetry+2,5,1,0)
      if %linemeth <> "none" and !gamma > !linetrigger then
        if %linemeth = "lmr" then
          call mcz_lmr
          if %mcz_lmr = "err" then
            %mcz_sim = "err"
            return
            endif
          endif
        if %linemeth = "armijo"  then
          call mcz_armijo
          if %mcz_armijo = "err" then
            %mcz_sim = "err"
            return
            endif
          endif
        endif
      endif

    if !mceshow < 3 then
      statusline mce solution, iteration !mcetry, f(x) = !nconv
      endif

  ' test for convergence or iteration limit
    mce_sim_stats(!mcetry+2,2) =  _$_mce_conv_vec(!mcetry)
    if _$_mce_conv_vec(!mcetry) < !mceconv then
      %mce_converge = "yes"
      mce_sim_text.append At iteration {!mcetry}, convergence
```

```
    %mce_finish = "yes"
    endif
  if !mcetry = !mcemaxiter and  _$_mce_conv_vec(!mcetry) >= !mceconv then
    mce_sim_text.append No convergence in {!mcemaxiter} iterations
    if %dontstop = "yes" then
      %mce_finish = "no"
      %mce_converge = "yes"
      mce_sim_text.append Terminating call to mce_run, but execution continues
      else
      @uiprompt("No convergence in maximum number of iterations.  Execution termina
      stop
      endif
    endif


' continue if not converged
  if %mce_converge = "no" then

  ' ******************************
  ' Newton algorithm (optionally update MCE jacobian)
    if %meth = "newton" then
      if !mcetry = 1 then
        if %jinit = "identity" then
          matrix _$_mce_der_mat = !dfactor*@identity(!nmceinstrus*!nqtrs)
          endif
        if %jinit = "bd" then
          for !ijj = 1 to !nmcetargs
            !r = (!ijj-1)*!nqtrs
            matrix _$_mce_der_mat_{!ijj} = @subextract(_$_mce_der_mat,!r+1,!r+1,!r-
            next
          delete(noerr) _$_mce_der_mat
          endif
        if %jinit <> "identity" then
          mce_sim_stats(!mcetry+3,6) = "yes"
          endif
        else
        if (!gamma >= !jtrigger and %jupdate <> "none") then
          %jinit_bac = %jinit
          %jinit = %jupdate
          call mcz_derivs
          %jinit = %jinit_bac
          endif
        endif

      if %jinit = "bd" then
        for !ijj = 1 to !nmcetargs
```

```
      vector _$_vec_adds = -_$_mce_der_mat_{!ijj}*@subextract(_$_mce_gap_vec,(!ijj-1)*!nq
      matplace(_$_mce_direction,_$_vec_adds,(!ijj-1)*!nqtrs+1,1)
      next
    else
    _$_mce_direction = -(_$_mce_der_mat*_$_mce_gap_vec)
    endif

  endif

' ****************************
' Broyden algorithms
  if %meth = "broy" or %meth = "qnewton" then
    if !mcetry = 1 then
      if %jinit <> "identity" then
        _$_mce_direction = -(_$_mce_der_mat*_$_mce_gap_vec)
        else
        matrix _$_mce_der_mat = @identity(!nmceinstrus*!nqtrs)
        _$_mce_direction = -_$_mce_gap_vec
        endif
      if %jinit = "bd" then
        for !ijj = 1 to !nmcetargs
          !r = (!ijj-1)*!nqtrs
          matrix _$_mce_der_mat_{!ijj} = @subextract(_$_mce_der_mat,!r+1,!r+1,!r+!nqtrs,!r+
          next
        if %bmeth = "qnewton" then
          delete(noerr) _$_mce_der_mat
          endif
        endif
      endif
    if !mcetry > 1 then
      vector _$_instru_delta = _$_mce_instru_vec - _$_instru_prev
      vector _$_gap_delta = _$_mce_gap_vec - _$_gap_prev

    ' ***************
      if %meth = "broy" then
        matrix _$_jy = _$_mce_der_mat*_$_gap_delta
        matrix _$_sj = @transpose(_$_instru_delta)*_$_mce_der_mat
        scalar _$_sjf = @sum(@transpose(_$_instru_delta)*_$_jy)
        _$_mce_der_mat = _$_mce_der_mat + ((_$_instru_delta - _$_jy)*_$_sj)/_$_sjf
        _$_mce_direction = -(_$_mce_der_mat*_$_mce_gap_vec)
        endif

    ' ***************
      if %meth = "qnewton" then
        if !mcetry = 2 then
        ' create some matrices on first pass
```

```
  vector(!broymax) _$_stp_nrm
  matrix(!tmceinstrus,!broymax) _$_stp
  vector (!broymax) _$_lam_rec
  matrix(!tmceinstrus,1) _$_z
  vector(!broymax) _$_counter
  for !iq = 1 to !broymax
    _$_counter(!iq) = !iq
    next
  endif

if !mcetry <= !broymax + 1 then
  !q = !mcetry-1
  !f = !q
  else
  !q = @mod(!mcetry-1,!broymax) + 1
  !f = !broymax
  call shiftleft(_$_counter,1)
  endif

colplace(_$_stp,_$_instru_delta,!q)
_$_stp_nrm(!q) = @norm(_$_instru_delta,2)
_$_lam_rec(!q) = !mcz_step

if %jinit = "bd" and !mcetry <= !broymax + 1 then
  for !ijj = 1 to !nmcetargs
    vector _$_vec_adds = -_$_mce_der_mat_{!ijj}* @subextract(_$_mce_gap_v
    matplace(_$_z,_$_vec_adds,(!ijj-1)*!nqtrs+1,1)
    next
  else
  _$_z = -_$_mce_gap_vec
  endif

if !mcetry > 2 then
  for !kbroy = 2 to !f
    !k0 = _$_counter(!kbroy)
    !k1 = _$_counter(!kbroy - 1)
    !a = _$_lam_rec(!k1)/_$_lam_rec(!k0)
    !b = _$_lam_rec(!k1) - 1
    _$_z = _$_z + (!a*@columnextract(_$_stp,!k0)+!b*@columnextract(_$_stp
    next
  endif
!nrm2 = _$_stp_nrm(!q)^2
!lam = _$_lam_rec(!q)
!stz = @sum(@transpose(_$_instru_delta)*_$_z)
_$_mce_direction = (!nrm2*_$_z-(1-!lam)*!stz*_$_instru_delta)/(!nrm2-!lam
endif
```

```
          endif
        endif

    ' ****************************
    ' Fair-Taylor algorithm
      if %meth = "ft" then
        _$_mce_direction = -!lambda*_$_mce_gap_vec
        endif
    ' ****************************

      endif

    wend

' ***********************************************
' 5. final steps
' ***********************************************

  scalar _$_iterations = !mcetry

  if !mceshow = 2 then
    close mce_sim_stats
    endif

  if !mceshow = 1 or !mceshow = 2 then
    if @isobject("mce_sim_spool") then
      delete mce_sim_spool
      endif
    spool mce_sim_spool
    mce_sim_spool.append mce_sim_stats
    mce_sim_spool.append mce_sim_text
    mce_sim_spool.name untitled01 mce_sim_stats
    mce_sim_spool.name untitled02 mce_sim_text
    show mce_sim_spool
    endif


  if !mceshow < 3 then
    statusline mcz_sim finished
    endif


  if %simtype = "single" and %cleanup = "yes" then
    delete(noerr) _$_*
    endif
```

```
endsub
```

Defines:
mcz_sim, used in chunks 97, 99, 149, 154, 162, and 164.
Uses close 217a, mce_run 99, mcz_armijo 135, mcz_derivs 127, mcz_equalopt 113a, mcz_hasopt 113b, mcz_lmr 133, mcz_solvit 125, and shiftleft 169.

6.2.9 solve model consistent instrument values

125 ⟨*solve model consistent instrument values* 125⟩≡ (211)

```
subroutine mcz_solvit

' This subroutine first sets the MCE instrument values based on the current
' optimal direction and choice of step size, and then solves the models

  %mcz_solvit = "ok"

' update instrument values based on current direction and step size
  if !mcetry > 1 then
    mce_sim_stats(!mcetry+2,4) = !mcz_step
    _$_mce_instru_vec = _$_instru_prev + !mcz_step*_$_mce_direction
    matrix _$_tmp_mat = @unvec(_$_mce_instru_vec,!nqtrs)
    mtos(_$_tmp_mat,_$_mce_instrus)
    endif

' solve lag model
  smpl %mcestart %mceend
  !err_before = @errorcount
  {%_$_mod_b}.solve
  !err_after = @errorcount
  if !err_after > !err_before then
    if %dontstop = "yes" then
      %mcz_solvit = "err"
      return
      else
      @uiprompt("Error in solving lag model: execution terminating")
      stop
      endif
    endif

' optionally set terminal conditions on first iteration
  if !mcetry = 1 and %terminal = "yes" then
    call mcz_terminal
    endif

' solve lead model
  smpl %mcestart %mceend
  !err_before = @errorcount
  {%_$_mod_f}.solve
  !err_after = @errorcount
  if !err_after > !err_before then
    if %dontstop = "yes" then
      %mcz_solvit = "err"
```

```
      return
      else
      @uiprompt("Error in solving lead model: execution terminating")
      stop
      endif
    endif

' create group of the solution values of the mce target variables
' on the first iteration
  if !mcetry = 1 then
    {%_$_mod_f}.makegroup _$_mce_errs_sols {%mce_targ_names}
    endif

' compute mce error functions
  _$_mce_targ_vec = @vec(@convert(_$_mce_errs_sols))
  _$_mce_gap_vec = _$_mce_targ_vec
  !nloss = @norm(_$_mce_gap_vec,2)^2
  mce_sim_stats(!mcetry+2,3) = !nloss
  _$_mce_loss_vec(!mcetry) = !nloss
  !nconv =  @max(@abs(_$_mce_gap_vec))
  _$_mce_conv_vec(!mcetry) = !nconv
  if !mcetry > 1 then
    !gamma = !nloss/_$_mce_loss_vec(!mcetry-1)
    endif

  if !mceshow = 2 then
    show mce_sim_stats
    endif

endsub
```

Defines:
  mcz_solvit, used in chunks 97, 114, 116, 133, and 135.
Uses mcz_terminal 136 and solve 217a.

6.2.10 compute derivatives of mce targets wrt mce instruments

127 ⟨*compute derivatives of mce targets wrt mce instruments* 127⟩≡ (211)

```
subroutine mcz_derivs

'This subroutine computes the derivatives of the mce targets wrt the mce instruments

  mce_sim_stats(!mcetry+2,6) = "yes"
  matrix _$_mce_der_mat = @filledmatrix(!nmceinstrus*!nqtrs,!nmcetargs*!nqtrs,0)

  if !mcetry > 1 then
    '_$_mce_ptrb_mat = @abs(@unvec(!mcz_step*_$_mce_direction,!nqtrs)) + 1e-6
    _$_mce_ptrb_mat = @abs(@unvec(!mcz_step*_$_mce_direction,!nqtrs)) + 1e-4
    endif

  smpl %mcestart %mceend
  _$_mce_targ_vec = @vec(@convert(_$_mce_errs_sols))

  !maxlead = 1

' *********************************
' construct vector that determines when exact derivatives
' need to be computed

' _$_dvec > 0 => period in which derivatives are to be simulated
' _$_dvec = 1 => but derivatives do not have to be spread/interpolated forward or back
' _$_dvec = 2 => and derivatives have to be spread forward down diagonals
' _$_dvec = 3 => and derivatives have to be spread backward up diagonals
' _$_dvec = 4 => and derivaties have to be interpolated backward along diagonals
' _$_dvec = 5 => hybrid
  vector(!nqtrs) _$_dvec = 0

  ' *********************************
  if %jinit = "every" then
    _$_dvec = _$_dvec + 1
    endif

  ' *********************************
  if %jinit = "interp" then
    _$_dvec(1) = 1
    _$_dvec(!nqtrs-!maxlead) = 4
    for !ij0 = (1+!nskip) to (!nqtrs-!maxlead-1) step !nskip
      _$_dvec(!ij0) = 4
      next
    if !maxlead > 0 then
```

```
      for !ij0 = (!nqtrs-!maxlead+1) to !nqtrs
        _$_dvec(!ij0) = 1
        next
      endif
    if _$_dvec(!nqtrs-!maxlead-1) = 4 then
      _$_dvec(!nqtrs-!maxlead)  = 1
      endif
    endif

' *********************************
  if %jinit = "bd" then
    !bd_col = @floor(!nqtrs/2)
    _$_dvec(!bd_col) = 5
    endif


' *********************************
' code modified 5/27/10 to reduce the number of derivative
' sims by 1 per mc variable -- will work only when
' maxlead = 1
  if %jinit = "linear" then
    _$_dvec(1) = 2
    _$_dvec(!nqtrs) = 3
    '_$_dvec(!nqtrs-!maxlead) = 3
    'for !ij0 = !nqtrs - !maxlead + 1 to !nqtrs
    '  _$_dvec(!ij0) = 1
    '  next
    endif

' *********************************
' *********************************
' Outer loop:  specifies which instrument is shocked
  for !ij1 = 1 to !nmceinstrus
    %instru_name = _$_mce_instrus.@seriesname(!ij1)
    statusline computing MCE derivatives for instrument %instru_name

  ' *********************************
  ' Middle loop:  simulates effects of instrument shock for selected time periods
    !skip = 0
    for !ij2 = 1 to !nqtrs
      !skip = !skip + 1
      if _$_dvec(!ij2) > 0 then
        !perturbit =  _$_mce_ptrb_mat(!ij2,!ij1)
        !col = (!ij1-1)*!nqtrs + !ij2
        smpl %mcestart + !ij2-1 %mcestart + !ij2-1
        {%instru_name} = {%instru_name} + !perturbit
```

```
  smpl %mcestart %mceend
  {%_$_mod_b}.solve
  {%_$_mod_f}.solve
  _$_mce_targ_dvec = @vec(@convert(_$_mce_errs_sols))
  matplace(_$_mce_der_mat,(_$_mce_targ_dvec-_$_mce_targ_vec)/!perturbit,1,!col)
  smpl %mcestart + !ij2-1 %mcestart + !ij2-1
  {%instru_name} = {%instru_name} - !perturbit

' *********************************
' Inner loop:  place and/or interpolates derivatives
  if %jinit <> "every" then
    for !ij3 = 1 to !nmceinstrus
      !row = (!ij3-1)*!nqtrs + 1
      matrix _$_tempa = @subextract(_$_mce_der_mat,!row,!col,!row+!nqtrs-1,!col)

    ' ***************************************************************
      if %jinit = "linear" then

      ' forward loop moves the derivative diagonally down to the right
      ' when _$_dvec = 2
        if _$_dvec(!ij2) = 2 then
        for !ij4 = 1 to (!nqtrs - !maxlead - 1)
          matrix _$_tempb = @subextract(_$_tempa,1,1,!nqtrs-!ij4,1)
          matplace(_$_mce_der_mat,_$_tempb,!row+!ij4,!col+!ij4)
          next
        endif

      ' backward loop moves the derivative diagonally up to the left
      ' when _$_dvec = 3
        if _$_dvec(!ij2) = 3 then
          !ij5 =  !nqtrs - !maxlead - 1
          for !ij4 = 1 to !ij5
            matrix _$_tempb = @subextract(_$_tempa,(1+!ij4),1,!nqtrs-!maxlead,1)
            matplace(_$_mce_der_mat,_$_tempb,!row,!col-!ij4)
            next
          endif

      ' fill unassigned elements of last !maxlead rows with zeros when _$_dvec = 3
        if _$_dvec(!ij2) = 3 then
          !ij5 =  !nqtrs - 2
          matrix(1,!ij5) _$_tempc = 0
          for !ij4 = 1 to !maxlead
            matplace(_$_mce_der_mat,_$_tempc,!row+!nqtrs-!ij4,!col-!nqtrs+2)
            next
          endif
```

```
    endif

' ***************************************************************
  if (%jinit = "interp") and (_$_dvec(!ij2) = 4) then
    !nn2 = !nqtrs-!skip+1
    'fill in columns between current and previous derivative
    '_$_tempa holds current derivative, _$_tempb holds previous derivative

    matrix _$_tempa1 = @subextract(_$_tempa,1,1,!skip,1)
    matrix _$_tempa2 = @subextract(_$_tempa,!skip+1,1,!nqtrs,1)

    'previous derivative
    matrix _$_tempb = @subextract(_$_mce_der_mat,!row,!col-!skip,!row+!nqtr
    matrix _$_tempb1 = @subextract(_$_tempb,1,1,!nn2-1,1)
    matrix _$_tempb2 = @subextract(_$_tempb,!nn2,1,!nqtrs,1)

    'forward loop moves lower (ie, nonoverlapping) portion of _$_tempb diag
    for !ij4 = 1 to !skip-1
      matrix _$_tempb2a = @subextract(_$_tempb2,1,1,!skip-!ij4,1)
      matplace(_$_mce_der_mat,_$_tempb2a,!row+!nn2-1+!ij4,!col-!skip+!ij4)
      next

    'backward loop moves upper (ie, nonoverlapping) portion of _$_tempa dia
    for !ij4 = 1 to !skip-1
      matrix _$_tempa1a = @subextract(_$_tempa1,1+!ij4,1,!skip,1)
      matplace(_$_mce_der_mat,_$_tempa1a,!row,!col-!ij4)
      next

    'overlap loop interpolates upper part of _$_tempb and lower part of _$_
    for !ij4 = 1 to !skip-1
      matrix _$_tempc = ((!skip-!ij4)/!skip)*_$_tempb1 + (!ij4/!skip)*_$_te
      matplace(_$_mce_der_mat,_$_tempc,!row+!ij4,!col-!skip+!ij4)
      next

    'optionally interpolate elements of last !maxlead rows across each row
    !qqqq = 0
    if !qqqq = 1 then
      !nn3 = !nqtrs - !maxlead + 1
      matrix _$_tempa3 = @subextract(_$_tempa,!nn3,1,!nqtrs,1)
      matrix _$_tempb3 = @subextract(_$_tempb,!nn3,1,!nqtrs,1)
      for !ij4 = 1 to !skip-1
        matrix _$_tempc = ((!skip-!ij4)/!skip)*_$_tempb3 + (!ij4/!skip)*_$_
        matplace(_$_mce_der_mat,_$_tempc,!row+!nn3-1,!col-!skip+!ij4)
        next
      endif
```

```
            endif

          '*****************************************************************
          'The bd method takes account of only two own-derivative elements --
          'one on the diagonal and one on the first super diagonal.  The latter is
          'important for one-lead euler equations.
          '
          'there is some crude coding here, reflecting the fact that the full column
          'of the derivatives matrix has already been filled in and needs to be
          'zeroed out

           if (%jinit = "bd") and (_$_dvec(!ij2) = 5) and (!ij1=!ij3) then
             matplace(_$_mce_der_mat,0*_$_mce_targ_dvec,1,!col)
             !diag = _$_tempa(!bd_col,1)
             vector(!nqtrs-1) _$_tempb = _$_tempa(!bd_col-1,1)
             'vector(!nqtrs-1) _$_tempb1 = _$_tempa(!bd_col+1,1)
             matrix(!nqtrs,!nqtrs) _$_tempc
             matrix _$_tempc = !diag*@identity(!nqtrs) + @makediagonal(_$_tempb,1)
             'matrix _$_tempc = !diag*@identity(!nqtrs) + @makediagonal(_$_tempb,1) + @makediago
             _$_tempc = @inverse(_$_tempc)
             matplace(_$_mce_der_mat,_$_tempc,!row,!row)
             endif


            '*****************************************************************

             next
            !skip = 0
            endif
          endif
        next
      next
' *********************************

  delete(noerr) _$_tempa _$_tempb _$_tempc _$_tempd

  'recalculate current solution
  smpl %mcestart %mceend
  {%_$_mod_b}.solve
  {%_$_mod_f}.solve


' *********************************
' invert derivative matrix (unless using bd method, in which case
' it's already inverted)
  if %jinit <> "bd" then
```

```
      statusline inverting MCE derivative matrix
      _$_mce_der_mat = @inverse(_$_mce_der_mat)
      endif

  endsub
```

Defines:
  mcz_derivs, used in chunks 114 and 116.
Uses left 219 and solve 217a.

6.2.11 model consistent coefficient non-monotone step-length procedure

133 ⟨*model consistent coefficient non-monotone step-length procedure* 133⟩≡ (211)

```
subroutine mcz_lmr

  %mcz_lmr = "ok"

  !step_pos = !mcz_step
  !step_neg = !mcz_step
  %line_converge = "no"
  !nk = _$_mce_loss_vec(1)/(!mcetry)^2
  !loss_prev = _$_mce_loss_vec(!mcetry-1)
  if !mhistory >= (!mcetry-1) then
    vector losshist =  @subextract(_$_mce_loss_vec,1,1,!mcetry-1,1)
    else
    vector losshist = @subextract(_$_mce_loss_vec,!mcetry-!mhistory,1,!mcetry-1,1)
    endif
  !fkbar = @max(losshist)
  for !j = 1 to !mcelinemax
    if !j > 1 then
      !mcz_step = !step_pos
      call mcz_solvit
      if %mcz_solvit = "err" then
        %mcz_lmr = "err"
        return
        endif
      endif
    if !nloss <= (!fkbar + !nk - !gammak*!step_pos^2*!loss_prev) then
      %line_converge = "yes"
      exitloop
      else
      !nloss_pos = !nloss
      !mcz_step = -!step_neg
      call mcz_solvit
      if %mcz_solvit = "err" then
        %mcz_lmr = "err"
        return
        endif
      if !nloss <= (!fkbar + !nk - !gammak*!step_neg^2*!loss_prev) then
        %line_converge = "yes"
        exitloop
        endif
      !nloss_neg = !nloss
      endif
    !alphat = !step_pos^2*!loss_prev/(!nloss_pos + (2*!step_pos-1)*!loss_prev)
```

```
      if !alphat < (!tmin*!step_pos) then
        !step_pos = !tmin*!step_pos
        else
        if !alphat > !tmax*!step_pos then
          !step_pos = !tmax*!step_pos
          else
          !step_pos = !alphat
          endif
        endif
      !alphat = !step_neg^2*!loss_prev/(!nloss_neg + (2*!step_neg-1)*!loss_prev)
      if !alphat < (!tmin*!step_neg) then
        !step_neg = !tmin*!step_neg
        else
        if !alphat > !tmax*!step_neg then
          !step_neg = !tmax*!step_neg
          else
          !step_neg = !alphat
          endif
        endif
      next
    setcell(mce_sim_stats,!mcetry+2,5,!j,0)

  endsub
```

Defines:
  mcz_lmr, used in chunk 116.
Uses mcz_solvit 125.

6.2.12 model consistent armijo optimization rule

135 ⟨*model consistent armijo optimization rule* 135⟩≡ (211)

```
subroutine mcz_armijo

  %mcz_armijo = "ok"

  !iarm = 0
  while ((!nloss > (1-.01*!mcz_step)*!loss_prev) and !iarm < !mcelinemax)
    !mcz_step = !lrat * !mcz_step
    call mcz_solvit
    if %mcz_solvit = "err" then
      %mcz_armijo = "err"
      return
      endif
    !iarm = !iarm + 1
    wend
  setcell(mce_sim_stats,!mcetry+2,5,!iarm,0)

endsub
```

Defines:
`mcz_armijo`, used in chunk 116.
Uses `mcz_solvit` 125.

6.2.13 variable terminal values

136 ⟨*variable terminal values* 136⟩≡ (211)

```
subroutine mcz_terminal

' Set terminal values for all variables in _$_mod_f that might
' appear with leads, based on the simulated values of the corresponding
' variables in _$_mod_b at the end of the simulation period.
' Terminal values of stationary variables equal their last simulated
' level.  Terminal values of nonstationary are based on extrapolating their
' simulated growth rates.
'
' This subroutine is usually called only once per MCE simulation and only
' for MCE simulations of permanent shocks.  Multiple calls to this
' subroutine may result in convergence problems as a result of terminal
' values that drift from iteration to iteration.


' %fvars (variables that are exogenous in _$_mod_f and
' also endogenous in _$_mod_b)

  {%_$_mod_b}.makegroup _$_fvars_sols {%fvars}

  for !ijj = 1 to @wcount(%fvars)
    %tmpa = @lower(@word(%fvars,!ijj))
    %tmpb = _$_fvars_sols.@seriesname(!ijj)
    smpl  %mceend + 8 %mceend + 8
    !tmpmean = @abs(@mean({%tmpa}))
    if !tmpmean > .001 then
      series _$_tmpgrowth = @abs(@movav(d({%tmpa},0,1)/{%tmpa}(-1),8))
      else
      series _$_tmpgrowth = 0
      endif
    !tmpscal = @mean(_$_tmpgrowth)
    if (!tmpscal < .001) then
      'stationary case
      smpl %mceend + 1  @last
      {%tmpb} = {%tmpb}(-1)
      else
      'nonstationary case
      smpl %mceend  %mceend
      series _$_tmpgrowth = @movav(d({%tmpb},0,1)/{%tmpb}(-1),8)
      !tmpscal = @mean(_$_tmpgrowth) + 1
      smpl %mceend + 1  @last
      {%tmpb} = !tmpscal*{%tmpb}(-1)
      endif
```

```
      {%tmpa} = {%tmpb}
      next

  ' mcevars_f (mce variables in _$_mod_f, if any) have terminal conditions
  ' based on the values of mcevars_b (proxies for mce variables in _$_mod_b)

    if @wcount(%mcevars_f) > 0 then
      {%_$_mod_b}.makegroup _$_mcevars_b_sols {{%mcevars_b}}
      %mcevars_f1 = {%mcevars_f}
      for !ijj = 1 to  @wcount(%mcevars_f1)
        %tmpa = @lower(@word(%mcevars_f1,!ijj))
        %tmpb = _$_mcevars_b_sols.@seriesname(!ijj)
        smpl %mceend + 8 %mceend + 8
        !tmpmean = @abs(@mean({%tmpa}))
        if !tmpmean > .001 then
          series _$_tmpgrowth = @abs(@movav(d({%tmpa},0,1)/{%tmpa}(-1),8))
          else
          series _$_tmpgrowth = 0
          endif
        !tmpscal = @mean(_$_tmpgrowth)
        if (!tmpscal < .001) then
          'stationary case
          smpl %mceend + 1  @last
          {%tmpb} = {%tmpb}(-1)
          else
          'nonstationary case
          smpl %mceend  %mceend
          series _$_tmpgrowth = @movav(d({%tmpb},0,1)/{%tmpb}(-1),8)
          !tmpscal = @mean(_$_tmpgrowth) + 1
          smpl %mceend + 1  @last
          {%tmpb} = !tmpscal*{%tmpb}(-1)
          endif
        smpl %mceend + 1  %mceend + 8
        {%tmpa} = {%tmpb}
        next
      endif

    mce_sim_text.append Terminal conditions set at iteration {!mcetry}

  ' once terminal conditions have been set once, turn the switch that
  ' calls this subroutine off
    %terminal = "no"

  endsub
```

Defines:
mcz_terminal, used in chunk 125.

6.2.14 set options based on defaults and overrides

138 ⟨*set options based on defaults and overrides* 138⟩≡ (211)

```
subroutine mcz_opt_setup(string mcz_opt_options, group mcz_opt_instrus, group mcz_opt

  statusline mcz_opt_setup


' ************************************************************************
' set up values of options based on defaults and overrides in string mcz_opt_options

  mcz_opt_options = @lower(mcz_opt_options)
  mcz_opt_options = @replace(mcz_opt_options," ","")
  mcz_opt_options = @replace(mcz_opt_options,","," ")
  mcz_opt_options = " " + mcz_opt_options + " "

' default values for options

  if %simtype <> "opttc" then
    !optmaxiter = 15
    !optconv = 1e-05
    else
    !optmaxiter = 50
    !optconv = 1e-06
    !tcdamp = 1
    endif

  !optlinemax = 10
  !optptrb = .01
  !opt_step_max = 1.0
  !optshow = 3
  !mceshow = 3

  %evlstart = %mcestart
  %drvstart = %mcestart
  %freq = @pagefreq
  %evlend = @datestr(@dateadd(@dateval(%evlstart),59,%freq))
  %drvend = @datestr(@dateadd(@dateval(%drvstart),39,%freq))

  %ideriv = "yes"
  %xopen = "yes"
  %xclose = "yes"

' if imposing constraints (mcz_opt_qp)
  %qpswitch = "r"
```

```
' are there overrides to defaults?
if @len(mcz_opt_options) > 0 then
  %opts = "yes"
  else
  %opts = "no"
  endif

if %opts = "yes" then
  call mcz_equalopt("m",mcz_opt_options)
  if @len(%temp)>0 then
    !optmaxiter  = @val(%temp)
    endif
  call mcz_equalopt("d",mcz_opt_options)
  if @len(%temp)>0 then
    !tcdamp  = @val(%temp)
    endif
  call mcz_equalopt("c",mcz_opt_options)
  if @len(%temp)>0 then
    !optconv  = @val(%temp)
    endif
  call mcz_equalopt("lmax",mcz_opt_options)
  if @len(%temp)>0 then
    !optlinemax  = @val(%temp)
    endif
  call mcz_equalopt("p",mcz_opt_options)
  if @len(%temp)>0 then
    !optptrb  = @val(%temp)
    endif
  call mcz_equalopt("stepmax",mcz_opt_options)
  if @len(%temp)>0 then
    !optstepmax  = @val(%temp)
    endif
  call mcz_equalopt("oo",mcz_opt_options)
  if @len(%temp)>0 then
    !optshow  = @val(%temp)
    !mceshow  = @val(%temp)
    endif
  call mcz_equalopt("lstart",mcz_opt_options)
  if @len(%temp)>0 then
    %evlstart = @lower(%temp)
    endif
  call mcz_equalopt("lend",mcz_opt_options)
  if @len(%temp)>0 then
    %evlend = @lower(%temp)
    endif
  call mcz_equalopt("istart",mcz_opt_options)
```

```
    if @len(%temp)>0 then
      %drvstart = @lower(%temp)
      endif
    call mcz_equalopt("iend",mcz_opt_options)
    if @len(%temp)>0 then
      %drvend = @lower(%temp)
      endif
    call mcz_equalopt("ideriv",mcz_opt_options)
    if @len(%temp)>0 then
      %ideriv = @lower(%temp)
      endif
    call mcz_hasopt("terminal",mcz_opt_options)
    if !hasflag = 1 then
      %terminal = "yes"
      endif
    call mcz_hasopt("matlab",mcz_opt_options)
    if !hasflag = 1 then
      %qpswitch = "matlab"
      endif
    call mcz_hasopt("/xopen",mcz_opt_options)
    if !hasflag = 1 then
      %xopen = "no"
      endif
    call mcz_hasopt("/xclose",mcz_opt_options)
    if !hasflag = 1 then
      %xclose = "no"
      endif

    endif


' ***************************************************************************
' Some preliminaries

' copy group subroutine arguments into objects with fixed names so that they can be
' easily accessed by other subroutines
  copy mcz_opt_instrus     _$_opt_instrus
  copy mcz_opt_targs       _$_opt_targs

  smpl %evlstart %evlend
  !nevl = @obssmpl
  smpl %drvstart %drvend
  !ndrv = @obssmpl


' ************************************
```

```
' Examine the inequality constraints, put them in a table,
' and, if necessary, augment the list of target variables to
' include all constraint variables
  if %cnstrflag = "yes" then
    svector _$_opt_cnstr = mcz_opt_cnstr.@svectornb
    !nconstraints = @rows(_$_opt_cnstr)
    if !nconstraints > 0 then
      table(1,1) _$_opt_cnstr_tab
      string _$_opt_cnstr_vars = " "
      call mcz_constraints(_$_opt_cnstr,_$_opt_cnstr_tab,_$_opt_cnstr_vars)
      %extra_targets = @wnotin(@upper(_$_opt_cnstr_vars),_$_opt_targs.@members)
      %extra_targets = @wunique(%extra_targets)
      !nextra = @wcount(%extra_targets)
      if !nextra > 0 then
        for !qq = 1 to !nextra
          %newtarg = @word(%extra_targets,!qq)
          _$_opt_targs.add {%newtarg}
          %newtarg_t = %newtarg + "_t"
          %newtarg_w = %newtarg + "_w"
          smpl @all
          series {%newtarg_t} = 0
          series {%newtarg_w} = 0
          next
        endif
      else
      @uiprompt("Error:  The cnstr keyword is assigned to an empty text file")
      stop
      endif
    endif



' ***************************************************************************
' More preliminaries

  !noptinstrus = _$_opt_instrus.@count
  !nopttargs = _$_opt_targs.@count

  !topttargs = !nopttargs * !nevl
  !toptinstrus = !noptinstrus * !ndrv
  if !topttargs < !toptinstrus then
    statusline Error: more instruments than targets
    stop
    endif

  matrix _$_opt_ptrb_mat = @filledmatrix(!ndrv,!noptinstrus,!optptrb)
```

```
  vector(!optmaxiter+1) _$_opt_loss_vec

  if %ideriv <> "no" then
    matrix _$_opt_der_mat = @filledmatrix(!toptinstrus,!topttargs,0)
    endif

  %opt_targ_names = _$_opt_targs.@members
  %opt_des_names = @wcross(%opt_targ_names,"_t")
  group _$_opt_des {%opt_des_names}
  smpl %evlstart %evlend
  matrix _$_opt_des_vec = @vec(@convert(_$_opt_des))

' if number of instruments and targets is the same, the optimal loss is zero,
' shortcut formulas can be used, and weights are not necessary
  if !toptinstrus = !topttargs then
    !zero_loss = 1
    matrix _$_opt_wt_mat = @identity(!topttargs)
    else
    !zero_loss = 0
    smpl %evlstart %evlend
    %opt_wt_names = @wcross(%opt_targ_names,"_w")
    group _$_opt_wts {%opt_wt_names}
    matrix _$_opt_wt_mat = @makediagonal(@vec(@convert(_$_opt_wts)))
    endif




' ************************************************************************
' text file

  text mce_opt_text

  if %cnstrflag = "yes" then
    mce_opt_text.append constrained optimization using {%qpswitch}:  xopen = {%xopen}
    else
    mce_opt_text.append unconstrained optimization (EViews)
    endif
  if %simtype = "opt" then
    mce_opt_text.append optimization type = committment
    mce_opt_text.append simulation period:  {%mcestart} - {%mceend}
    mce_opt_text.append loss evalation period: {%evlstart} - {%evlend}
    mce_opt_text.append instrument setting period: {%drvstart} - {%drvend}
    mce_opt_text.append max number of optimization iterations = !optmaxiter
    mce_opt_text.append max number of line search steps per iteration = !optlinemax
    endif
```

```
  if %simtype = "opttc" then
    mce_opt_text.append optimization type = time-consistent nash (discretion)
    mce_opt_text.append first instrument setting period: {%drvstart}
    mce_opt_text.append first simulation period:  {%mcestart} - {%mceend}
    mce_opt_text.append first loss evalation period: {%evlstart} - {%evlend}
    mce_opt_text.append last instrument setting period: {%drvend}
    mce_opt_text.append max number of backward-induction iterations = !optmaxiter
    endif
  mce_opt_text.append convergence criteria = !optconv
  mce_opt_text.append output control parameter = !optshow
  mce_opt_text.append compute instrument derivs? = {%ideriv}
  mce_opt_text.append instrument perturbation factor = !optptrb


' ************************************************************************
' table of iteration-by-iteration statistics

  if @isobject("mce_opt_stats") then
    delete mce_opt_stats
    endif
  table(!optmaxiter+2,7) mce_opt_stats
  mce_opt_stats.setwidth(1:1) 5
  mce_opt_stats.setwidth(1:5) 12
  mce_opt_stats.setlines(a2:g2) +b
  setcell(mce_opt_stats,1,1,"iter")
  setcell(mce_opt_stats,1,2,"f(x)")
  setcell(mce_opt_stats,1,3,"step size")
  setcell(mce_opt_stats,1,4,"convergence")
  setcell(mce_opt_stats,2,4,"statistic")
  setcell(mce_opt_stats,1,5,"linearity")
  setcell(mce_opt_stats,2,5,"statistic")

' ************************************************************************
' call appropriate optimization subroutine

  if %simtype = "opt" and %cnstrflag = "no" then
    call mcz_opt
    endif
  if %simtype = "opt" and %cnstrflag = "yes" then
    call mcz_opt_qp
    endif
  if %simtype = "opttc" then
    call mcz_opt_tc
    endif
```

```
' ***************************************************************************
' spool
  if @isobject("mce_opt_spool") then
    delete mce_opt_spool
    endif
  spool mce_opt_spool
  if %simtype = "opt" then
    mce_opt_spool.append mce_opt_stats
    mce_opt_spool.append mce_opt_text
    mce_opt_spool.name untitled01 mce_opt_stats
    mce_opt_spool.name untitled02 mce_opt_text
    endif
  if %simtype = "opttc" then
    mce_opt_spool.append mce_opt_text
    mce_opt_spool.name untitled01 mce_opt_text
    endif

  if !optshow <> 4 then
    show mce_opt_spool
    close mce_opt_stats
    endif

  if %cleanup = "yes" then
    delete(noerr) _$_*
    endif


  endsub
```

Defines:
mcz_opt_setup, used in chunk 99.
Uses close 217a, mcz_constraints 166, mcz_equalopt 113a, mcz_hasopt 113b, mcz_opt 145, mcz_opt_qp 149, and mcz_opt_tc 154.

6.2.15 main unconstrained optimal control simulation

145 ⟨*main unconstrained optimal control simulation* 145⟩≡ (211)

```
subroutine mcz_opt

'Main subroutine for unconstrained optimal control simulations


' ***************************************************************************
' initialize counters and switches

  !opttry = 0
  %opt_converge = "no"
  !optnonlin = 0
  !optpchloss = 100



' ***************************************************************************
' initial solution

  smpl %mcestart %mceend
  !opt_step = 0
  !opt_step_prev = 0
  call mcz_opt_solve
  if !opttry = 0 and !optloss = 0 then
    %opt_converge = "yes"
    mce_opt_text.append At iteration {!opttry}, convergence
    endif



' ***************************************************************************
' iterate to minimize loss
  smpl %mcestart %mceend
  while !opttry <= !optmaxiter and %opt_converge = "no"

    !opttry = !opttry + 1

  ' ******************************
  ' compute instrument derivatives, hessian, gradient, direction, and
  ' predicted loss assuming the model is linear
    if (%ideriv <> "no" and (!opttry = 1 or @abs(!optnonlin) > .1)) then
      call mcz_opt_deriv
      endif
    if !zero_loss = 0 then
```

```
    matrix _$_opt_hess = 2*_$_opt_der_mat*_$_opt_wt_mat*@transpose(_$_opt_der_mat)
    else
    matrix _$_opt_hess = 2*@transpose(_$_opt_der_mat)
    endif
  if @issingular(_$_opt_hess) = 1 then
    %errstring = "Hessian is singular at iteration " + @str(!opttry)
    @uiprompt(%errstring)
    stop
    endif
  matrix _$_opt_hessinv = @inverse(_$_opt_hess)
  if !zero_loss = 0 then
    matrix _$_opt_grad = 2*_$_opt_der_mat*_$_opt_wt_mat*_$_opt_gap_vec
    else
    matrix _$_opt_grad = 2*_$_opt_gap_vec
    endif
  vector _$_opt_direction = _$_opt_hessinv*_$_opt_grad
  matrix _$_opt_gap_vec_p = _$_opt_gap_vec - @transpose(_$_opt_der_mat) * _$_opt_di
  !optloss_p = @sum(@transpose(_$_opt_gap_vec_p)*_$_opt_wt_mat*_$_opt_gap_vec_p)

' *******************************
' solve model and compute loss
  !opt_step = !opt_step_max
  !opt_step_prev = 0
  call mcz_opt_solve

' *******************************
' test for nonlinearity
' !optnonlin is the ratio of the actual to predicted percentage reduction in loss l
'  the closer the model is to being linear, the closer !optnonlin is to zero
  !ddloss = (_$_opt_loss_vec(!opttry)-_$_opt_loss_vec(!opttry+1))
  if !ddloss <> 0 then
    !optnonlin = (_$_opt_loss_vec(!opttry+1)-!optloss_p) / !ddloss
    else
    !optnonlin = 100
    endif
  mce_opt_stats(!opttry+3,5) = !optnonlin

' *******************************
' Search for a better a step size (Armijo condition and backtracking)
  !opt_loss_dderiv = -@transpose(_$_opt_grad)*_$_opt_direction
  !kk = .01
  !iarm = 0
  while (_$_opt_loss_vec(!opttry+1) > _$_opt_loss_vec(!opttry) + !kk*!opt_step*!opt
    and !iarm < !optlinemax
    !iarm = !iarm + 1
    !opt_step_prev = !opt_step
```

```
    !opt_step = .5*!opt_step
    call mcz_opt_solve
    wend

  ' ******************************
  ' Test for convergence
    if !zero_loss = 1 then
    ' equal number of targets and controls, full rank derivative matrix
    ' => loss has to be less than !optconv
    mce_opt_stats(!opttry+3,4) =  _$_opt_loss_vec(!opttry+1)
    if _$_opt_loss_vec(!opttry+1) < !optconv then
      %opt_converge = "yes"
      endif
    else
    ' !zero_loss = 0 => percentage change in loss from previous iteration has
    ' to be less than !optconv
    !optpchloss =  1-(_$_opt_loss_vec(!opttry+1)/_$_opt_loss_vec(!opttry))
    mce_opt_stats(!opttry+3,4) =  !optpchloss
    if !optpchloss < !optconv then
      %opt_converge = "yes"
      mce_opt_text.append At iteration {!opttry}, convergence
      endif
    endif

  if !opttry = 1 and @abs(!optnonlin) < 1e-7 and !opt_step = 1 then
    %opt_converge = "yes"
    mce_opt_text.append At iteration 1, convergence assumed because model is linear
    endif

  if !opttry = !optmaxiter then
    mce_opt_text.append At iteration {!opttry}, no convergence in {!optmaxiter} iterations
    !continue = @uiprompt("Maximum number of optimization iterations reached:  Continue execu
    if !continue = 2 then
      stop
      else
      %opt_converge = "yes"
      endif
    endif

  if !optshow <> 4 then
    show mce_opt_stats
    endif
  wend
```

```
endsub
```

Defines:
mcz_opt, used in chunk 138.
Uses mcz_opt_deriv 162, mcz_opt_solve 164, and solve 217a.

6.2.16 main optimal control simulation with inequality constraints

149 ⟨*main optimal control simulation with inequality constraints* 149⟩≡ (211)

```
subroutine mcz_opt_qp

' Main subroutine for optimal control simulations with inequality constraints
' (requires either R or matlab)


' The original problem:
'
'   choose x to minimize z = (y-y*)'W(y-y*)
'   under the constraint Cy >= c
'
'   where at each iteration the linearized relationship between
'   the target variables (y) and instrument variables (x) is
'   given by
'
'   y = B'x + k
'
' The transformed problem:  Solve out y
'
'   choose x to minimize z = x'BWB'x + 2(k-y*)'WB'x + const
'   under the constraint CB'x >= c - Ck
'
' The R command -- quadprog::solve.QP(D,d,A,b) -- solves
'
'   min [0.5 * x' D x - d'x] with the constraint A'x >= b
'
' The matlab command -- x = quadprog(D,d,A,b) -- solves
'
'   min [0.5 * x' D x + d'x] with the constraint Ax <= b
'
'             R             matlab
'   D     2BWB'             same
'   d'    2(k-y*)'WB'      -2(k-y*)'WB'
'   A'    CB'              -CB'
'   b     c-Ck             -(c-Ck)
'
'   where k = y - B'x
'


' ************************************
' Create constraint matrices (C,c)-- dimensions are based on the number
' of periods in which the instruments are set (!ndrv) not the number of
```

```
' periods in which the loss function is evaluated (!nevl)

if %cnstrflag = "yes" then
  matrix(!nevl * !nconstraints,!nevl*!nopttargs) _$_opt_cnstr_mat = 0
  matrix(!nevl * !nconstraints,1) _$_opt_cnstr_vec = 0
  for !i = 1 to !nconstraints
    !nterms = @val(_$_opt_cnstr_tab(!i,1))
    for !k = 1 to !nterms
      %var = @upper(_$_opt_cnstr_tab(!i,2*!k))
      !coef =  @val(_$_opt_cnstr_tab(!i,2*!k+1))
      !j = @wfindnc(%opt_targ_names,%var)
      for !l = 1 to !nevl
        _$_opt_cnstr_mat((!i-1)*!nevl+!l,(!j-1)*!nevl+!l) = !coef
        next
      next
    for !l = 1 to !nevl
      !coef =  @val(_$_opt_cnstr_tab(!i,2*!nterms+2))
      _$_opt_cnstr_vec((!i-1)*!nevl+!l) = !coef
      next
    next
  else
  !nconstraints = 1
  matrix(!nevl,!nevl*!nopttargs) _$_opt_cnstr_mat = 0
  matrix(!nevl,1) _$_opt_cnstr_vec = 0
  endif


' ***************************************************************************
' initialize counters and switches

  !opttry = 0
  %opt_converge = "no"
  !optnonlin = 0
  !optpchloss = 100

  if %qpswitch = "r" and %xopen = "yes" then
    xopen(type=r, case=lower)
    endif
  if %qpswitch = "matlab" and %xopen = "yes" then
    xopen(type=m, case=lower)
    endif


' ***************************************************************************
' compute an initial solution, without reference to any constraints
```

```
  smpl %mcestart %mceend
  !opt_step = 0
  !opt_step_prev = 0
  call mcz_opt_solve


' ***************************************************************************
' iterate to minimize loss
  smpl %mcestart %mceend
  while !opttry <= !optmaxiter and %opt_converge = "no"

    !opttry = !opttry + 1

  ' ******************************
  ' compute instrument derivatives
    if %ideriv <> "no" then
      call mcz_opt_deriv
      endif

  ' ******************************
  ' compute qp matrices
    smpl %drvstart %drvend
    vector _$_opt_instru_vec = @vec(@convert(_$_opt_instrus))
    matrix _$_qp_k_vec = _$_opt_targ_vec-@transpose(_$_opt_der_mat)*_$_opt_instru_vec
    matrix qp_d_mat = 2 * _$_opt_der_mat * _$_opt_wt_mat * @transpose(_$_opt_der_mat)
    matrix qp_d_vec = -2 * _$_opt_der_mat*@transpose(_$_opt_wt_mat) _
                      * (_$_qp_k_vec - _$_opt_des_vec)

    matrix qp_a_mat = @transpose(_$_opt_cnstr_mat * @transpose(_$_opt_der_mat))
    matrix qp_b_vec = _$_opt_cnstr_vec - _$_opt_cnstr_mat _
                     * (_$_qp_k_vec - _$_opt_des_vec)

    if !ndrv <> !nevl then
      qp_a_mat = @subextract(qp_a_mat,1,1,!ndrv,!ndrv)
      qp_b_vec = @subextract(qp_b_vec,1,1,!ndrv,1)
      endif

    xput qp_d_mat qp_a_mat qp_d_vec qp_b_vec

  ' ******************************
  ' ******************************
  ' R code

    if %qpswitch = "r" then
      if ((!ndrv = 1) and (!noptinstrus = 1)) then
        xrun "dim(qp_d_mat) <- c(1,1);"
```

```
      xrun "dim(qp_a_mat) <- c(1,1);"
      xrun "dim(qp_d_vec) <- c(1,1);"
      xrun "dim(qp_b_vec) <- c(1,1);"
      endif
    xrun "QP.results <- quadprog::solve.QP(qp_d_mat,qp_d_vec,qp_a_mat,qp_b_vec);"
    xrun "r_xhat <- QP.results$solution;"
    xrun "r_xhat <- as.matrix(r_xhat);"
    xget(type = matrix, name = _$_qp_instru_vec) r_xhat
    endif

' ********************************
' ********************************
' matlab code

  if %qpswitch = "matlab" then
    xrun "qp_d_vec = -qp_d_vec;"
    xrun "qp_a_mat = -qp_a_mat';"
    xrun "qp_b_vec = -qp_b_vec;"
    xrun  "aeq = [];"
    xrun  "beq = [];"
    xrun  "lb = [];"
    xrun  "ub = [];"
    xrun  "xx0 = [];"
    xrun "options = optimset('Algorithm','active-set','LargeScale','off');"
    xrun "matlab_xhat = quadprog(qp_d_mat,qp_d_vec,qp_a_mat,qp_b_vec,aeq,beq,lb,ub,
    xget matlab_xhat
    %tmp = matlab_xhat.@type
    if %tmp = "SCALAR" then
      matrix(1,1) _$_qp_instru_vec = matlab_xhat
      else
      matrix _$_qp_instru_vec = matlab_xhat
      delete matlab_xhat
      endif
    endif

' ********************************
' ********************************

' values of target variables in qp (linearized) solution
  matrix _$_qp_targ_vec = @transpose(_$_opt_der_mat)*_$_qp_instru_vec + _$_qp_k_vec

' values of target variables in original model
  smpl %drvstart %drvend
  matrix _$_tt1 = @unvec(@vec(_$_qp_instru_vec),!ndrv)
  mtos(_$_tt1,_$_opt_instrus)
  smpl %mcestart %mceend
```

```
      call mcz_sim(" ")
      smpl %evlstart %evlend
      matrix _$_opt_targ_vec = @vec(@convert(_$_opt_targs_sols))
      matrix _$_opt_gap_vec = _$_opt_targ_vec - _$_opt_des_vec
      !optloss = @sum(@transpose(_$_opt_gap_vec)*_$_opt_wt_mat*_$_opt_gap_vec)
      setcell(mce_opt_stats,!opttry+2,1,!opttry,0)

    ' convergence test is based on the maximum difference between
    ' the target variable values in the linearized and original models
      !conv_stat = @max(abs(_$_qp_targ_vec-_$_opt_targ_vec))
      mce_opt_stats(!opttry+2,2) = !optloss
      mce_opt_stats(!opttry+2,4) = !conv_stat

      if !optshow <> 4 then
        show mce_opt_stats
        endif

      if !conv_stat < !optconv then
        %opt_converge = "yes"
        else
        endif

      wend

    delete(noerr) qp_d_mat qp_a_mat qp_d_vec qp_b_vec


  endsub
```

Defines:
mcz_opt_qp, used in chunks 97 and 138.
Uses mcz_opt_deriv 162, mcz_opt_solve 164, mcz_sim 116, solve 217a, xget 217a, and xrun 217a.

6.2.17 model consistent optimal time-consistent solution

154 ⟨*model consistent optimal time-consistent solution* 154⟩≡ (211)

```
subroutine mcz_opt_tc


' 1. For a linear model, this code finds the exact time-consistent solution;
'    for a nonlinear model, the calculated solution is based on the
'    linearization of the model as given by the derivatives of the target
'    variables wrt the instrument variables along the baseline.
'
' 2. The basic data matrices include observations for (!nevl+!ndrv-1) periods,
'    which is also the interval over which tracking adds are required.
'
' 3. The simulation, derivative and evaluation periods are assumed to have the
'    same starting date
'
' 4. The solution method is backward induction; if there are no inequality constraint
'    the method executes in EViews (using matrices); if inequality constraints
'    are present, the method executes in EViews if there is a single
'    instrument and in Matlab or R (depending on the
'    setting of a switch) if there are multiple instruments (which requires
'    a quadratic programming algorithm),


' ***************************************
' Create constraint matrices (C,c)-- dimensions are based on the fact that
' each policymaker sets the instruments for a single period

  if %cnstrflag = "yes" then
    matrix(!nconstraints,!nopttargs) _$_opt_cnstr_mat = 0
    matrix(!nconstraints,1) _$_opt_cnstr_vec = 0
    for !i = 1 to !nconstraints
      !nterms = @val(_$_opt_cnstr_tab(!i,1))
      for !k = 1 to !nterms
        %var = @upper(_$_opt_cnstr_tab(!i,2*!k))
        !coef =  @val(_$_opt_cnstr_tab(!i,2*!k+1))
        !j = @wfindnc(%opt_targ_names,%var)
        _$_opt_cnstr_mat(!i,!j) = !coef
        next
      !coef =  @val(_$_opt_cnstr_tab(!i,2*!nterms+2))
      _$_opt_cnstr_vec(!i) = !coef
      next
    endif
```

```
' ********************************************************************************
' initial solution, instrument derivatives, and data matrices


' initial simulation
  smpl %mcestart %mceend
  if %terminal = "yes" then
    call mcz_sim("terminal")
    %terminal = "no"
    else
    call mcz_sim("  ")
    endif
  {%_$_mod_b}.makegroup _$_opt_targs_sols {%opt_targ_names}
  !ntot = !ndrv + !nevl - 1

' derivative matrices and submatrices
' _$_opt_der_mat (this is the basic matrix; tc_dmat is its transpose)
' tc_dmat        effect of !ndrv periods of instruments on !ntot periods of targets
' tc_dmat_short  effect of !ndrv periods of instruments on !nevl periods of targets
' tc_dmat1       effect of period 1 instrument on !nevl periods of targets
' tc_dmat2       effect of period 1 instrument on period 1 targets
  if %ideriv <> "no" then
    !opttry = 1
    matrix _$_opt_der_mat = @filledmatrix(!toptinstrus,!nopttargs*!ntot,0)
    %evlend_b = %evlend
    %freq = @pagefreq
    !ntot1 = !ntot-1
    %evlend = @datestr(@dateadd(@dateval(%simstart),!ntot1,%freq))
    'call dateshift(%simstart,%evlend,!ntot-1)
    call mcz_opt_deriv
    %evlend = %evlend_b
    matrix tc_dmat = @transpose(_$_opt_der_mat)
    matrix tc_dmat_short = @filledmatrix(!nevl*!nopttargs,!ndrv*!noptinstrus,0)
    for !i = 1 to !toptinstrus
      for !j = 1 to !nopttargs
        !r1 = !ntot*(!j-1)+1
        !r2 = !r1 + !nevl - 1
        matplace(tc_dmat_short,@subextract(tc_dmat,!r1,!i,!r2,!i),!nevl*(!j-1)+1,!i)
        next
      next
    matrix tc_dmat1 = @filledmatrix(!nevl*!nopttargs,!noptinstrus,0)
    for !i = 1 to !noptinstrus
      matplace(tc_dmat1,@columnextract(tc_dmat_short,(!i-1)*!ndrv+1),1,!noptinstrus)
      next
    matrix tc_dmat2 = @filledmatrix(!noptinstrus,!nopttargs,0)
    for !i = 1 to !noptinstrus
```

```
      for !j = 1 to !nopttargs
        tc_dmat2(!i,!j) = tc_dmat1((!j-1)*!nevl+1,!i)
        next
      next
    endif

' initial simulation again (is this necessary)
  call mcz_sim(" ")

' data matrices
  smpl %simstart {%evlend} + !ndrv - 1
  stom(_$_opt_targs_sols,tc_ymat)
  stom(_$_opt_des,tc_ystarmat)
  stom(_$_opt_instrus,tc_xmat)
  matrix tc_wmat = _$_opt_wt_mat

' (no constraints) => eviews;
' (1 constraint -- single period) => eviews
' (multiple constraints) => R or Matlab
  if %cnstrflag = "no" then
    %iii = "eviews"
    else
    if !noptinstrus = 1 then
      %iii = "eviews"
      else
      %iii = %qpswitch
      endif
    endif

  scalar tc_iter = 0
  scalar tc_stat = 100
  scalar tc_itmax = !optmaxiter
  scalar tc_conv = !optconv


' ***************************************************************************
' unconstrained backward induction => EViews

  if %cnstrflag = "no" then
    while (tc_stat > tc_conv) and tc_iter <= tc_itmax

      matrix tc_xb = tc_xmat
      matrix tc_yb = tc_ymat

      for !i = !ndrv to 1 step -1
        !n1 = !i+!nevl-1
```

```
        !n2 = !i+!ndrv-1
        vector tc_y0v = @vec(@subextract(tc_ymat,!i,1,!n1,!nopttargs))
        vector tc_ys0v = @vec(@subextract(tc_ystarmat,!i,1,!n1,!nopttargs))
        vector tc_x0v =  @vec(@subextract(tc_xmat,!i,1,!n2,!noptinstrus))
        vector tc_x0_1v = @vec(@subextract(tc_xmat,!i,1,!i,!noptinstrus))
        vector tc_xbv =  @vec(@subextract(tc_xb,!i,1,!n2,!noptinstrus))

        vector tc_y1v = tc_y0v + tc_dmat_short*(tc_x0v-tc_xbv)

        matrix tc_g = @transpose(tc_dmat1)*tc_wmat*(tc_y1v-tc_ys0v)
        matrix tc_h = @transpose(tc_dmat1)*tc_wmat*tc_dmat1
        vector tc_x1_1v = tc_x0_1v - !tcdamp*@inverse(tc_h)*tc_g
        matrix tc_x1_1 = @unvec(tc_x1_1v,!noptinstrus)
        matplace(tc_xmat,tc_x1_1,!i,1)
        next

      vector tc_x0v = @vec(@subextract(tc_xmat,1,1,!ndrv,!noptinstrus))
      vector tc_y0v = @vec(@subextract(tc_ymat,1,1,!ntot,!nopttargs))
      vector tc_y1v = tc_y0v + tc_dmat*(tc_x0v-tc_xbv)
      matrix tc_y1 = @unvec(tc_y1v,!ntot)
      matplace(tc_ymat,tc_y1,1,1)

      tc_iter = tc_iter + 1
      if tc_iter > 1 then
        tc_stat = @max(@abs(tc_xmat-tc_xmatprev))
        endif

      matrix tc_xmatprev = tc_xmat
      wend
    endif


' ***************************************************************************
' constrained backward induction

  if %cnstrflag = "yes" then

    matrix tc_cmat = _$_opt_cnstr_mat
    matrix tc_cvec =  _$_opt_cnstr_vec


  ' ************************************************************************
  ' single-period constraint => eviews
    if %iii = "eviews" then

      statusline constrained TC optimization in EViews
```

```
matrix mat_a = tc_dmat2*@transpose(tc_cmat)

while (tc_stat > tc_conv) and tc_iter <= tc_itmax

  matrix tc_xb = tc_xmat
  matrix tc_yb = tc_ymat

  for !i = !ndrv to 1 step -1
    !n1 = !i+!nevl-1
    !n2 = !i+!ndrv-1

  ' vectors of observations in the current loss and instrument periods
    vector tc_y0v = @vec(@subextract(tc_ymat,!i,1,!n1,!nopttargs))
    vector tc_ys0v = @vec(@subextract(tc_ystarmat,!i,1,!n1,!nopttargs))
    vector tc_x0v =  @vec(@subextract(tc_xmat,!i,1,!n2,!noptinstrus))
    vector tc_x0_1v = @vec(@subextract(tc_xmat,!i,1,!i,!noptinstrus))
    vector tc_xbv =  @vec(@subextract(tc_xb,!i,1,!n2,!noptinstrus))

  ' vector of target variable values based on current instrument values
  ' (ie, solve model)
    vector tc_y1v = tc_y0v + tc_dmat_short*(tc_x0v-tc_xbv)

  ' unconstrained current optimal instrument value
    matrix tc_g = @transpose(tc_dmat1)*tc_wmat*(tc_y1v-tc_ys0v)
    matrix tc_h = @transpose(tc_dmat1)*tc_wmat*tc_dmat1
    vector tc_x1_1v = tc_x0_1v - @inverse(tc_h)*tc_g

  ' check constraint
    matrix tc_y1m = @unvec(tc_y1v,!nevl)
    matrix tc_ysm = @unvec(tc_ys0v,!nevl)
    matrix mattemp = @rowextract(tc_y1m,1) - (@transpose(tc_x0_1v)*tc_dmat2) -
    matrix mat_b = tc_cvec - tc_cmat*@transpose(mattemp)
    scalar testit = @sum(mat_b)/@sum(mat_a)
    if @sum(tc_x1_1v) < testit then
      tc_x1_1v = testit
      !tc_iter = tc_iter
      endif
    vector tc_x1_1v = (1-!tcdamp)*tc_x0_1v + !tcdamp* tc_x1_1v

  ' place current optimal instrument value in instrument matrix
    matrix tc_x1_1 = @unvec(tc_x1_1v,!noptinstrus)
    matplace(tc_xmat,tc_x1_1,!i,1)

    next
```

```
    ' solve model
      vector tc_x0v = @vec(@subextract(tc_xmat,1,1,!ndrv,!noptinstrus))
      vector tc_y0v = @vec(@subextract(tc_ymat,1,1,!ntot,!nopttargs))
      vector tc_y1v = tc_y0v + tc_dmat*(tc_x0v-tc_xbv)
      matrix tc_y1 = @unvec(tc_y1v,!ntot)
      matplace(tc_ymat,tc_y1,1,1)

      tc_iter = tc_iter + 1

      if tc_iter > 1 then
        tc_stat = @max(@abs(tc_xmat-tc_xmatprev))
        endif

      matrix tc_xmatprev = tc_xmat
      wend
    endif


' ***************************************************************************
' multiple constraints => R or matlab

  if %iii = "r" or %iii = "matlab" then

    scalar tc_noptinstrus = !noptinstrus
    scalar tc_nopttargs = !nopttargs
    scalar tc_nevl = !nevl
    scalar tc_ndrv = !ndrv
    scalar tc_damp = !tcdamp

    if %xopen = "yes" then
      if %iii = "r" then
        xopen(type=r)
        xrun library("quadprog")
        %wd = """ + %rpath + """
        xrun setwd({%wd})
        else
        xopen(type=m)
        xrun addpath {%mpath}
        endif
      endif

    xput tc_ymat tc_ystarmat tc_xmat tc_wmat tc_cmat tc_cvec
    xput  tc_nopttargs tc_noptinstrus tc_nevl tc_ndrv tc_itmax tc_conv
    xput tc_dmat tc_dmat1 tc_dmat2 tc_dmat_short tc_damp
```

```
      if %iii = "r" then
        xrun source("tcoc_r.R")
        else
        xrun tcoc_m
        endif

      xget tc_xmat
      xget tc_ymat
      xget tc_iter
      !tc_iter = tc_iter

      if %xclose = "yes" then
        xclose
        endif

      delete(noerr) tc_ystarmat tc_dmat tc_wmat tc_cmat tc_cvec
      delete(noerr) tc_nopttargs tc_noptinstrus tc_nevl tc_ndrv tc_conv
      delete(noerr) tc_damp tc_dmat1 tc_dmat2 tc_dmatshort

      endif
    endif

' ***************************************************************************
' examine linearized solution

  if tc_iter >= tc_itmax then
    @uiprompt("iteration limit reached:  time-consistent iterations did not converge'
    'stop
    endif

' write instrument and target values back into their corresponding series
  smpl %evlstart {%evlstart} + !nevl + !ndrv - 2
  mtos(tc_xmat,_$_opt_instrus)
  mtos(tc_ymat,_$_opt_targs_sols)

' value of loss function for linearized solution
  smpl %evlstart %evlend
  matrix _$_opt_targ_vec = @vec(@convert(_$_opt_targs_sols))
  matrix _$_opt_gap_vec = _$_opt_targ_vec - _$_opt_des_vec
  !optloss_lin = @sum(@transpose(_$_opt_gap_vec)*_$_opt_wt_mat*_$_opt_gap_vec)
  stom(_$_opt_targs_sols,tc_ymat_lin)

' simulate EViews model
  smpl %mcestart %mceend
  call mcz_sim("  ")
  smpl %evlstart %evlend
```

```
    stom(_$_opt_targs_sols,tc_ymat_ev)
    !maxdiff = @max(@abs(tc_ymat_ev-tc_ymat_lin))
    matrix _$_opt_targ_vec = @vec(@convert(_$_opt_targs_sols))
    matrix _$_opt_gap_vec = _$_opt_targ_vec - _$_opt_des_vec
    !optloss_ev = @sum(@transpose(_$_opt_gap_vec)*_$_opt_wt_mat*_$_opt_gap_vec)

    !tciter = tc_iter
    mce_opt_text.append time-consistent solution
    mce_opt_text.append --- backward-induction iterations = !tciter
    mce_opt_text.append --- TC damping factor = !tcdamp
    mce_opt_text.append --- linearized solution loss = !optloss_lin
    mce_opt_text.append --- EViews solution loss = !optloss_ev
    mce_opt_text.append --- max diff btwn target vars in linear and EViews sols = !maxdiff

    delete(noerr) tc_ymat tc_xmat tc_iter tc_ymat_lin tc_ymat_ev
    delete(noerr) tc_itmax tc_stat

  endsub
```

Defines:
  mcz_opt_tc, used in chunks 97 and 138.
Uses dateshift 89, mcz_opt_deriv 162, mcz_sim 116, solve 217a, xget 217a, and xrun 217a.

6.2.18 compute derivatives of loss function targets wrt instruments

162 ⟨*compute derivatives of loss function targets wrt instruments* 162⟩≡ (211)

```
subroutine mcz_opt_deriv
'
' requires that the following be defined
'
' !optshow, mce_opt_stats, %evlstart, %evlend
' _$_opt_targs_sols, !noptinstrus, _$_opt_instrus, !drv,
' _$_opt_ptrb_mat, _$_opt_der_mat

' This subroutine computes the derivatives of the loss function targets wrt the instr

  smpl @all

  !optshow_bac = !optshow
  !optshow = 3

  'make a copy of the current values of the mce instruments;
  'they will be restored after each derivative is computed
  smpl %mcestart %mceend
  stom(_$_mce_instrus,_$_instrus_bac)

  'do not reset terminal conditions when computing derivatives
  %set_terminal = "no"

  smpl %evlstart %evlend
  matrix _$_opt_targ_vec = @vec(@convert(_$_opt_targs_sols))

  '*****************************************
  'derivatives loop
  statusline computing instrument derivatives at optimization iteration !opttry
  for !zi = 1 to !noptinstrus
    %instru_name = _$_opt_instrus.@seriesname(!zi)
    for !zj = 1 to !ndrv
      !perturbit =  _$_opt_ptrb_mat(!zj,!zi)
      smpl %mcestart + !zj-1 %mcestart + !zj-1
      {%instru_name} = {%instru_name} + !perturbit
      call mcz_sim(" ")
      smpl %evlstart %evlend
      matrix _$_opt_targ_dvec = @vec(@convert(_$_opt_targs_sols))
      matplace(_$_opt_der_mat,@transpose(_$_opt_targ_dvec-_$_opt_targ_vec)/!perturbit
      smpl %mcestart + !zj-1 %mcestart + !zj-1
      {%instru_name} = {%instru_name} - !perturbit
      'restore the original values of the add factors on
```

```
        'the z variable equations
        smpl %mcestart %mceend
        mtos(_$_instrus_bac,_$_mce_instrus)
        next
      next

    smpl %mcestart %mceend

    !optshow = !optshow_bac

  endsub
```

Defines:
mcz_opt_deriv, used in chunks 145, 149, and 154.
Uses mcz_sim 116.

6.2.19 solve model consistent expectations model

164 ⟨*solve model consistent expectations model* 164⟩≡ (211)

```
subroutine mcz_opt_solve

' This subroutine first sets the instrument values, based on the current
' optimal direction and choice of step size, and then solves the model

  statusline optimization solution, iteration !opttry
  smpl %mcestart %mceend

  'compute instrument values based on current direction and step size
  if !opttry > 0 then
    mce_opt_stats(!opttry+3,3) = !opt_step
    smpl %drvstart %drvend
    for !i = 1 to !noptinstrus
      vector _$_temp_adj = @subextract(_$_opt_direction,(!i-1)*!ndrv+1,1,!i*!ndrv,1)
      mtos(_$_temp_adj,_$_temp_ser)
      %y1 = _$_opt_instrus.@seriesname(!i)
      {%y1} = {%y1} - (!opt_step-!opt_step_prev)*_$_temp_ser
      next
    endif

  'solve model
  if !opttry = 0 and %terminal = "yes" then
    call mcz_sim("terminal")
    %terminal = "no"
    else
    call mcz_sim("  ")
    endif

  if !opttry = 0 then
    {%_$_mod_b}.makegroup _$_opt_targs_sols {%opt_targ_names}
    endif

  'compute value of loss function
  smpl %evlstart %evlend
  matrix _$_opt_targ_vec = @vec(@convert(_$_opt_targs_sols))
  matrix _$_opt_gap_vec = _$_opt_targ_vec - _$_opt_des_vec
  !optloss = @sum(@transpose(_$_opt_gap_vec)*_$_opt_wt_mat*_$_opt_gap_vec)
  setcell(mce_opt_stats,!opttry+3,1,!opttry,0)
  mce_opt_stats(!opttry+3,2) = !optloss
  _$_opt_loss_vec(!opttry+1) = !optloss

  if !optshow = 2 then
    show mce_opt_stats
```

```
      endif

  endsub
```

Defines:
  mcz_opt_solve, used in chunks 145 and 149.
Uses mcz_sim 116.

6.2.20 convert mcz inequality constraints

166 ⟨*convert mcz inequality constraints* 166⟩≡ (211)

```
subroutine local mcz_constraints(svector sss, table ctable,string cvarnames)


'This subroutine converts the inequality constraint text into a table

' subroutine arguments:
'
' inputs:
'
'   sss                svector of constraint text
'
' outputs:
'
'   _$_opt_cnstr_tab      table of constraint variables and coefficients
'   _$_opt_cnstr_vars     string of names of variables in constraints

' ***************************************

  !nconstraints = @rows(sss)
  scalar loc
  scalar sign

  for !i = 1 to !nconstraints
    %rrr = sss(!i)

  ' ****************************************
  ' ****************************************
  ' separate the left and right sides of each constraint

    !kk1 = @instr(%rrr,">=")
    if !kk1 = 0 then
      @uiprompt("each constraint must contain an ""="" term")
      stop
      endif
    %rrrl = @left(%rrr,!kk1-1)
    if @isempty(%rrrl) = 1 then
      @uiprompt("each constraint must contain terms to the left of ""="""")
      stop
      endif
    %rrrr = @mid(%rrr,!kk1+2)
    if @isempty(%rrrr) = 1 then
      @uiprompt("each constraint must contain a value to the right of ""="""")
      stop
```

```
    endif


' ****************************************
' ****************************************
' process the left side of each constraint
'
' 1. split %rrrl into individual terms (based on + and - characters);
'    the main challenge concerns the first term, which may or may not
'    have a leading + or - attached
' 2. then split each term into 2 parts (based on * character)
' 3. then identify which part is coefficient and which is variable

  !zz = 0    'flag to indicate end of left hand side of a constraint
  !nterms = 1
  vector(100) split_locs = 0
  vector(100) split_signs = 0
  %rrrlx = %rrrl

' find boundaries and signs of each term
  while !zz = 0

  ' first term: determine whether its sign is explicit or implicit
    if !nterms = 1 then
      call find_next_delimit(%rrrlx,loc,sign)
      if loc = 0 then                  'implicit leading sign
        split_locs(!nterms) = 0
        split_signs(!nterms) = 1
        else
        %rrrll = @left(%rrrlx,loc-1)
        if @isempty(%rrrll) = 1 then   'explicit leading sign
          split_locs(!nterms) = loc
          split_signs(!nterms) = sign
          %rrrlx = @mid(%rrrlx,loc+1)
          else                         'implicit leading sign
          split_locs(!nterms) = 0
          split_signs(!nterms) = 1
          endif
        endif
      endif

    call find_next_delimit(%rrrlx,loc,sign)
    if loc = 0 then                  'last term
      split_locs(!nterms+1) = @length(%rrrlx)
      !zz = 1
      else
```

```
      split_locs(!nterms+1) = loc
      split_signs(!nterms+1) = sign
      %rrrlx = @mid(%rrrlx,loc+1)
      endif

    !nterms = !nterms + 1
    wend

  for !k = 2 to !nterms
    split_locs(!k) = split_locs(!k-1) + split_locs(!k)
    next

  !nterms = !nterms - 1


' parse each term into coefficient times variable (they must
' appear in that order, although coefficients = 1 can be
' omitted), and store coefficient and variable in ctable
  for !k = 1 to !nterms
    %term = @mid(%rrrl,split_locs(!k)+1,split_locs(!k+1)-split_locs(!k)-1)
    if @isempty(%term) = 1 then
      @uiprompt("term is empty:  illegal constraint specification")
      stop
      else
      !ii = @instr(%term,"*")
      if !ii > 0 then
        %coef = @left(%term,!ii-1)
        !coef = split_signs(!k) * @val(%coef)
        %var =  @mid(%term,!ii+1)
        else
        !coef = split_signs(!k)
        %var = %term
        endif
      %var = @trim(%var)
      endif

    ctable(!i,2*!k) = %var
    ctable(!i,2*!k+1) = !coef
    cvarnames = cvarnames + " " + %var
    next


' ****************************************
' ****************************************
' process the right side of each constraint
```

```
      ctable(!i,2*!nterms+2) = @val(%rrrr)
      ctable(!i,1) = !nterms

    next

    cvarnames = @wunique(cvarnames)

  endsub
```

Defines:
mcz_constraints, used in chunk 138.
Uses find_next_delimit 170 and left 219.

6.2.21 shift left

169 ⟨*shift left* 169⟩≡ (211)

```
  subroutine local shiftleft(vector abc, scalar nshift)

    !rows = @rows(abc)
    vector v1 = @subextract(abc,1,1,nshift,1)
    vector v2 = @subextract(abc,nshift+1,1,!rows,1)
    matplace(abc,v2,1)
    matplace(abc,v1,!rows - nshift + 1)

  endsub
```

Defines:
shiftleft, used in chunk 116.

6.2.22 find next delimiter

170 ⟨*find next delimiter* 170⟩≡ (211)

```
subroutine local find_next_delimit(string %instring,scalar loc,scalar sign)

  !ttp = @instr(%instring,"+")
  !ttm = @instr(%instring,"-")
  if !ttp= 0 or !ttm = 0 then
    loc = !ttp + !ttm
    if !ttp = 0 and !ttm = 0 then
      sign = 1
      else
      sign = (!ttp > 0) - (!ttm > 0)
      endif
    else
    loc = !ttp*(!ttp < !ttm) + !ttm*(!ttm < !ttp)
    sign = (!ttp < !ttm) - (!ttm < !ttp)
  endif

endsub
```

Defines:
find_next_delimit, used in chunk 166.

6.2.23 load frbus with transformed subsidiary model

171 ⟨*load frbus with transformed subsidiary model* 171⟩≡ (211)

```
subroutine mce_load_frbus(string frbus_opts)

' take two corresponding FRB/US models (eg, stdver and pfver) and make
' the transformations needed to the second model so that the pair of
' models can be used with the mcz_solve_subs programs

' parameters
'   required:  mce_vars, mod_b, mod_f, path_b, path_f
'   optional:  allbut, only
'

  !allbut = 0
  !only = 0

  frbus_opts = @lower(frbus_opts)
  frbus_opts = @replace(frbus_opts," ","")
  frbus_opts = @replace(frbus_opts,","," ")
  frbus_opts = " " + frbus_opts + " "

  if @isempty(frbus_opts) = 0 then
    call mcz_equalopt("mce_vars",frbus_opts)
    if @len(%temp)>0 then
      %mce_vars  = @lower({%temp})
      else
      @uiprompt("Error:  mce_load_frbus sub requires the mce_vars argument")
      stop
      endif
    call mcz_equalopt("mod_b",frbus_opts)
    if @len(%temp)>0 then
      %mod_b  = {%temp}
      else
      @uiprompt("Error:  mce_load_frbus sub requires the mod_b argument")
      stop
      endif
    call mcz_equalopt("mod_f",frbus_opts)
    if @len(%temp)>0 then
      %mod_f  = {%temp}
      else
      @uiprompt("Error:  mce_load_frbus sub requires the mod_f argument")
      stop
      endif
    call mcz_equalopt("path_b",frbus_opts)
    if @len(%temp)>0 then
```

```
    %path_b  = {%temp}
    else
    @uiprompt("Error:  mce_load_frbus sub requires the path_b argument")
    stop
    endif
  call mcz_equalopt("path_f",frbus_opts)
  if @len(%temp)>0 then
    %path_f  = {%temp}
    else
    @uiprompt("Error:  mce_load_frbus sub requires the path_f argument")
    stop
    endif
  call mcz_equalopt("allbut",frbus_opts)
  if @len(%temp)>0 then
    !allbut = 1
    %allbut  = {%temp}
    else
    call mcz_equalopt("only",frbus_opts)
    if @len(%temp)>0 then
      !only = 1
      %only  = {%temp}
      endif
    endif
  endif

' Added so that users may ask directly for a group of forward-looking equations inste
' passing in a list that may change in the future.

' Model consistent Asset Pricing
  %s_mcap = " zdivgr zgap05 zgap10 zpi10f zpic30 zrff10 zrff5 zgap30 zrff30 zpi10 zpi

' Wages and prices
  %s_wp = " zpicxfe zpieci "

' Others - all PAC expectations
  %s_other = " zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvps zvpi zxnfbd zxnfbs zxnfbi zyh

  if %mce_vars = "-all" then
    %mce_vars = %s_mcap + %s_wp + %s_other
    endif
  if %mce_vars = "-mcap" then
    %mce_vars = %s_mcap
    endif
  if %mce_vars = "-wp" then
    %mce_vars = %s_wp
    endif
```

```
  if %mce_vars = "-mcap+wp" then
    %mce_vars = %s_mcap + %s_wp
    endif
  if @left(%mce_vars, 7) = "-allbut" then
    %tmp = %mce_vars
    %s_remove = @replace(%tmp, "-allbut", "")
    %mce_vars = @wnotin(%s_mcap + %s_wp + %s_other, %s_remove)
    endif
  string zvar_list = %mce_vars
  %zvars = %mce_vars


' backward-looking model

  ld_frbus_cfs(modelname=%mod_b,modelpath=%path_b)
  if !allbut = 1 then
    %tmp = "allbut " + %allbut
    ld_some_eqs(modelname=%mod_b,modelpath=%path_b,eqnames=%tmp)
    endif
  if !only = 1 then
    %tmp = %only
    ld_some_eqs(modelname=%mod_b,modelpath=%path_b,eqnames=%tmp)
    endif
  if !allbut = 0 and !only = 0 then
    ld_frbus_eqs(modelname=%mod_b,modelpath=%path_b)
    endif

' model with mce equations and errors

  ld_mce_eqs(pfname=%mod_f,pfpath=%path_f,mcename=%mod_f,mceeqs=%mce_vars)
  ld_mce_cfs(pfname=%mod_f,pfpath=%path_f,mceeqs=%mce_vars)

  !nmcevars = @wcount(%mce_vars)
  %evars = @wcross("e",%mce_vars)
  for !i = 1 to !nmcevars
    %tmp = @word(%mce_vars,!i)
    %tmpw = " w" + @mid(%tmp,2)
    if %tmp <> "zyh" and %tmp <> "zyhp" and %tmp <> "zyht" then
      %tmp1 = @word(%evars,!i) + "=" + @word(%mce_vars,!i) + "-" + %tmpw
      else
      %tmp1 = @word(%evars,!i) + "= log(" + @word(%mce_vars,!i) + "/" + %tmpw + ")"
      endif
    {%mod_f}.append {%tmp1}
    next

endsub
```

Defines:
mce_load_frbus, used in chunks 9, 19, 25, 33, and 97.
Uses ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, ld_mce_cfs 181b 182, ld_mce_eqs 184 185, ld_some_eqs 189 190, mcz_equalopt 113a, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpic58 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

6.2.24 create wage and expectation variables in forward looking model

175 ⟨*create wage and expectation variables in forward looking model* 175⟩≡ (211)

```
subroutine make_frbus_mcevars(string frbus_mcevars)

' Create data for the w and e variables in the operational forward looking
' model over the workfile sample currently in effect
'
' The input string contains the names of the expectations variables that
' are to have MC solutions

  %mce_vars = frbus_mcevars

' The user may ask directly for a group of forward-looking equations instead of
' passing in a list that may change in the future.

' Model consistent Asset Pricing
  %s_mcap = " zdivgr zgap05 zgap10 zpi10f zpic30 zrff10 zrff5 zgap30 zrff30 zpi10 zpib5 zpic58"

' Wages and prices
  %s_wp = " zpicxfe zpieci "

' Others - all PAC expectations
  %s_other = " zecd zeco zeh zgapc2 zlhp zpi5 zvpd zvps zvpi zxnfbd zxnfbs zxnfbi zyh zyhp zyht

  if %mce_vars = "-all" then
    %mce_vars = %s_mcap + %s_wp + %s_other
    endif
  if %mce_vars = "-mcap" then
    %mce_vars = %s_mcap
    endif
  if %mce_vars = "-wp" then
    %mce_vars = %s_wp
    endif
  if %mce_vars = "-mcap+wp" then
    %mce_vars = %s_mcap + %s_wp
    endif
  if @left(%mce_vars, 7) = "-allbut" then
    %tmp = %mce_vars
    %s_remove = @replace(%tmp, "-allbut", "")
    %mce_vars = @wnotin(%s_mcap + %s_wp + %s_other, %s_remove)
    endif


' Data for extra variables associated with MC expectations
```

```
    smpl @all
    for !i = 1 to @wcount(%mce_vars)
      %tmp = @word(%mce_vars,!i)
      %wtmp = "w" + @mid(%tmp,2)
      %wtmp_aerr = %wtmp + "_aerr"
      %etmp = "e" + @word(%mce_vars,!i)
      series {%wtmp} = {%tmp}
      series {%wtmp_aerr} = 0
      series {%etmp} = 0
      next

  endsub
```

Defines:
  make_frbus_mcevars, used in chunks 9, 19, 25, 33, and 97.
Uses zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpic58 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

File Contents

6.3 frbus package

6.3.1 srcEview/frbus.package/addins/ld.frbus.cfs/ld.frbus.cfs.prg

177 ⟨*srcEview/frbus.package/addins/ld.frbus.cfs/ld.frbus.cfs.prg* 177⟩≡
⟨*load frbus coefficients* 178a⟩

⟨*load frbus coefficients call* 178b⟩

This code is written to file `srcEview/frbus.package/addins/ld.frbus.cfs/ld.frbus.cfs.prg`.
Defines:
`ld_frbus_cfs`, used in chunks 5, 13, 42, 171, 178b, 195b, and 204.

178a ⟨*load frbus coefficients* 178a⟩≡ (177)

```
subroutine ld_frbus_cfs(string %mname, string %mpath)

'Load coefficients for frbus version %mname from a text file in directory %mpath
'that has been previously created by the script eq_docs2eviews.

  %cpath = %mpath + %mname  + "_coeffs.txt"
  delete(noerr) coefpath
  text coefpath
  coefpath.append(file) %cpath
  svector coefpathv = coefpath.@svectornb

  for !i = 1 to 900
    svector cofname = @wsplit(coefpathv(!i))
    %y1 = cofname(1)
      if @left(%y1,6) = "theend" then
        exitloop
      endif
    %y2 = cofname(2)
    %y3 = cofname(3)
    coef({%y2}) {%y1}
    {%y1}.fill {%y3}
    next

endsub
```

Defines:
ld_frbus_cfs, used in chunks 5, 13, 42, 171, 178b, 195b, and 204.

178b ⟨*load frbus coefficients call* 178b⟩≡ (177)

```
if @len(@option(1)) < 1  or @len(@option(2)) < 1 then
  @uiprompt("Error: ld_frbus_cfs requires model name and model path")
  stop
endif

%temp = @equaloption("modelname")
if @len(%temp)>0 then
        %mname = %temp
endif
%temp = @equaloption("modelpath")
if @len(%temp)>0 then
        %mpath = %temp
endif

call ld_frbus_cfs(%mname, %mpath)
```

Uses ld_frbus_cfs 177 178a.

6.3.2 srcEview/frbus.package/addins/ld.frbus.eqs/ld.frbus.eqs.prg

179 ⟨*srcEview/frbus.package/addins/ld.frbus.eqs/ld.frbus.eqs.prg* 179⟩≡
⟨*load frbus equations* 180⟩

⟨*load frbus equations call* 181a⟩

This code is written to file `srcEview/frbus.package/addins/ld.frbus.eqs/ld.frbus.eqs.prg`.
Defines:
`ld_frbus_eqs`, used in chunks 5, 13, 42, 171, 181a, 195b, and 204.

180 ⟨*load frbus equations* 180⟩≡ (179)

```
subroutine ld_frbus_eqs(string %mname, string %mpath)

'Create eviews model %mname and load it with the equations for frbus version
'%mname that are in a text file in directory %mpath that was previously created
'by the script eq_docs2eviews.

  %epath = %mpath + %mname
  if @fileexist(%epath) <> 1 then
   %epath = %mpath + %mname  + "_eqs.txt"
  endif
  delete(noerr) eqtext
  text eqtext
  eqtext.append(file) %epath
  svector eqtextv = eqtext.@svectornb

  model {%mname}

  !eqnum = 0
  %eqcode = " "

  for !i = 1 to 3000
    %y = eqtextv(!i)
    if @isempty(%y) = 0 then
      'string is not blank
      if @left(%y,6) = "theend" then
        %x = @replace(%eqcode," _"," ")
        {%mname}.append {%x}
        exitloop
        endif

      !k = @instr(%y,":")
      if !k > 0 then
        'string contains the start of a new equation
        !eqnum = !eqnum + 1
        if !eqnum > 0 then
          %x = @replace(%eqcode," _"," ")
          {%mname}.append {%x}
          endif
        %eqcode = @mid(%y,!k+1)

        else
        'string contains the continuation of an equation
        %eqcode = %eqcode + %y
        endif
```

```
        endif
      next

  endsub
```

Defines:
ld_frbus_eqs, used in chunks 5, 13, 42, 171, 181a, 195b, and 204.

181a ⟨*load frbus equations call* 181a⟩≡ (179)

```
  if @len(@option(1)) < 1  or @len(@option(2)) < 1 then
    @uiprompt("Error: ld_frbus_eqs requires model name and model path")
    stop
  endif

  %temp = @equaloption("modelname")
  if @len(%temp)>0 then
          %mname = %temp
  endif
  %temp = @equaloption("modelpath")
  if @len(%temp)>0 then
          %mpath = %temp
  endif

  call ld_frbus_eqs(%mname, %mpath)
```

Uses ld_frbus_eqs 179 180.

6.3.3 srcEview/frbus.package/addins/ld.mce.cfs/ld.mce.cfs.prg

181b ⟨*srcEview/frbus.package/addins/ld.mce.cfs/ld.mce.cfs.prg* 181b⟩≡

```
  ⟨load mce coefficients 182⟩

  ⟨load mce coefficients call 183⟩
```

This code is written to file srcEview/frbus.package/addins/ld.mce.cfs/ld.mce.cfs.prg.
Defines:
ld_mce_cfs, used in chunks 171, 183, and 195b.

182 ⟨*load mce coefficients* 182⟩≡ (181b)

```
subroutine ld_mce_cfs(string %pfname, string %pfpath, string %mceeqs)

'This subroutine is used for setting up the particular type of frbus simulations
'with model-consistent expectations in which a separate model is created
'containing only those expectations equations chosen to have model-consistent
'solutions.  For those expectations variables, the initial z character is replaced
'with w.

'Load coefficients for frbus version %pfname from a text file in directory %pfpath
'that has been previously created by the script eq_docs2eviews.  Only those
'coefficient vectors whose names are in the string %mceeqs are stored.



  %cpath = %pfpath + %pfname  + "_coeffs.txt"
  delete(noerr) coefpathpv
  text coefpathpv
  coefpathpv.append(file) %cpath
  svector coefpathv = coefpathpv.@svectornb

  for !i = 1 to 900
    svector cofname = @wsplit(coefpathv(!i))
    %y1 = cofname(1)
      if @left(%y1,6) = "theend" then
        exitloop
      endif
    %y2 = cofname(2)
    %y3 = cofname(3)
    for !j = 1 to @wcount(%mceeqs)
      %z = @lower(@word(%mceeqs,!j))
      %z1 = @mid(%y1,3)
      if %z = %z1 then
        %y1 = @replace(%y1,"z","w")
        coef({%y2}) {%y1}
        {%y1}.fill {%y3}
        exitloop
      endif
    next
  next


endsub
```

Defines:
ld_mce_cfs, used in chunks 171, 183, and 195b.

183 ⟨*load mce coefficients call* 183⟩≡ (181b)

```
if @len(@option(1)) < 1  or @len(@option(2)) < 1 or @len(@option(3)) < 1  then
  @uiprompt("Error: ld_mce_cfs requires four parameters: pf model  name, pf model path, mceeqs
  stop
endif

%temp = @equaloption("pfname")
if @len(%temp)>0 then
        %pfname = %temp
endif
%temp = @equaloption("pfpath")
if @len(%temp)>0 then
        %pfpath = %temp
endif
%temp = @equaloption("mceeqs")
if @len(%temp)>0 then
        %mceeqs = %temp
endif
'    ' Added so that users may ask directly for a group of forward-looking equations instead of
'    ' passing in a list that may change in the future.
' '   ' Model consistent Asset Pricing
'    %s_mcap = " zdivgr zgap05 zgap10 zpi10f zpic30 zrff10 zrff5 zgap30 zrff30 zpi10 zpib5 "
' '   ' Wages and prices
'    %s_wp = " zpicxfe zpieci "
' '   ' Others - all PAC expectations
'    %s_other = " zecd zeco zeh zgapc2 zlhp zpi5 zpi10 zpib5 zvpd zvps zvpi zxnfbd zxnfbs zxnfb
'    if %mceeqs = "-all" then
'      %mceeqs = %s_mcap + %s_wp + %s_other
'    endif
'    if %mceeqs = "-mcap" then
'      %mceeqs = %s_mcap
'    endif
'    if %mceeqs = "-wp" then
'      %mceeqs = %s_wp
'    endif
'    if %mceeqs = "-mcap+wp" then
'      %mceeqs = %s_mcap + %s_wp
'    endif
'    if @left(%mceeqs, 7) = "-allbut" then
'      %tmp = %mceeqs
'      %s_remove = @replace(%tmp, "-allbut", "")
'      %mceeqs = @wnotin(%s_mcap + %s_wp + %s_other, %s_remove)
'    endif
'    string zvar_list = %mceeqs
'    scalar nzvars = @wcount(%mceeqs)
'    group zvars {%mceeqs}
```

```
call ld_mce_cfs(%pfname, %pfpath, %mceeqs)
```

Uses ld_mce_cfs 181b 182, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

6.3.4 srcEview/frbus.package/addins/ld.mce.eqs/ld.mce.eqs.prg

184 ⟨*srcEview/frbus.package/addins/ld.mce.eqs/ld.mce.eqs.prg* 184⟩≡

⟨*load mce equations* 185⟩

⟨*load mce equations call* 188⟩

This code is written to file srcEview/frbus.package/addins/ld.mce.eqs/ld.mce.eqs.prg.
Defines:

ld_mce_eqs, used in chunks 171, 188, and 195b.

185 ⟨*load mce equations* 185⟩≡ (184)

```
subroutine ld_mce_eqs(string %pfname, string %pfpath, string %mcename, string %mceeqs)

'This subroutine is used for setting up the particular type of frbus simulations
'with model-consistent expectations in which a separate model is created
'containing only those expectations equations chosen to have model-consistent
'solutions.  For those expectations variables, the initial z character is replaced
'with w.

'Create eviews model %mcename and load it with the equations for frbus version
'%pfname that are in a text file in directory %pfpath that was previously created
'by the script eq_docs2eviews.  Only those equations whose names are in the
'string %mceeqs are included.

' revised 2/11/13 to add a check that version %pfname contains an equation for each
' name in %mceeqs


  %epath = %pfpath + %pfname  + "_eqs.txt"
  delete(noerr) eqtextp
  text eqtextp
  eqtextp.append(file) %epath
  svector eqtextpv = eqtextp.@svectornb

  %coded = " "

  model {%mcename}

  !eqnum = 0
  %eqnew = " "
  %eqold = " "
  %eqcode = " "
  %eqcodeold = " "

  for !i = 1 to 3000

    %appendit = "no"
    %exitloop = "no"
    %y = eqtextpv(!i)
    if @isempty(%y) = 0 then
      'string is not blank
      if @left(%y,6) = "theend" then
        'string contains end-of-file flag
        %appendit = "yes"
        %exitloop = "yes"
        %eqold = %eqnew
```

```
      %eqcodeold = %eqcode
      endif

    !k = @instr(%y,":")
    if !k > 0 then
      'string contains the start of a new equation
      %appendit = "yes"
      !eqnum = !eqnum + 1
      if !eqnum > 0 then
        %eqold = %eqnew
        %eqcodeold = %eqcode
        endif
      %eqnew  = @left(%y,!k-1)
      %eqcode = @mid(%y,!k+1)
      else
      'string contains the continuation of an equation
      %eqcode = %eqcode + %y
      endif

    if %appendit = "yes" then
      'add equation to model only if it is one with mce expectations
      for !j = 1 to @wcount(%mceeqs)
        %z = @lower(@word(%mceeqs,!j))
        if %z = %eqold then
          %x = @replace(%eqcodeold," _"," ")
          %x = @replace(%x,"z","w")
          {%mcename}.append {%x}
          %coded = %coded + " " + %z
          exitloop
          endif
        next
      endif

    if %exitloop = "yes" then
      exitloop
      endif

    endif
  next

' check that model %mcename contains an equation for each variable in %mceeqs
  !j = @wcount(%mceeqs)
  !k = @wcount(%coded)
  if !j <> !k then
    %z = @wnotin(@lower(%mceeqs),@lower(%coded))
    %errstring = "Error in ld_mce_eqs addin: Model " + %mcename
```

```
    %errstring = %errstring + " does not contain equation(s) for variable(s):  " + %z
    %errstring = %errstring + ".  Execution stopped."
    @uiprompt(%errstring)
    stop
    endif

  endsub
```

Defines:
ld_mce_eqs, used in chunks 171, 188, and 195b.

188 ⟨*load mce equations call* 188⟩≡ (184)

```
if @len(@option(1)) < 1  or @len(@option(2)) < 1 or @len(@option(3)) < 1 or @len(@opt
  @uiprompt("Error: ld_mce_eqs requires five parameters: pf model  name, pf model pat
  stop
endif

%temp = @equaloption("pfname")
if @len(%temp)>0 then
        %pfname = %temp
endif
%temp = @equaloption("pfpath")
if @len(%temp)>0 then
        %pfpath = %temp
endif
%temp = @equaloption("mcename")
if @len(%temp)>0 then
        %mcename = %temp
endif
%temp = @equaloption("mceeqs")
if @len(%temp) >0 then
        %mceeqs = %temp
endif
'    ' Added so that users may ask directly for a group of forward-looking equations
'    ' passing in a list that may change in the future.
' '   ' Model consistent Asset Pricing
'    %s_mcap = " zdivgr zgap05 zgap10 zpi10f zpic30 zrff10 zrff5 zgap30 zrff30 zpi10
' '   ' Wages and prices
'    %s_wp = " zpicxfe zpieci "
' '   ' Others - all PAC expectations
'    %s_other = " zecd zeco zeh zgapc2 zlhp zpi5 zpi10 zpib5 zvpd zvps zvpi zxbd zxbs
'    if %mceeqs = "-all" then
'      %mceeqs = %s_mcap + %s_wp + %s_other
'    endif
'    if %mceeqs = "-mcap" then
'      %mceeqs = %s_mcap
'    endif
'    if %mceeqs = "-wp" then
'      %mceeqs = %s_wp
'    endif
'    if %mceeqs = "-mcap+wp" then
'      %mceeqs = %s_mcap + %s_wp
'    endif
'    if @left(%mceeqs, 7) = "-allbut" then
'      %tmp = %mceeqs
'      %s_remove = @replace(%tmp, "-allbut", "")
'      %mceeqs = @wnotin(%s_mcap + %s_wp + %s_other, %s_remove)
```

```
'    endif
'    string zvar_list = %mceeqs
'    scalar nzvars = @wcount(%mceeqs)
'    group zvars {%mceeqs}

call ld_mce_eqs(%pfname, %pfpath, %mcename, %mceeqs)
```

Uses ld_mce_eqs 184 185, zdivgr 219, zecd 219, zeco 219, zeh 219, zgap05 219, zgap10 219, zgap30 219, zgapc2 219, zlhp 219, zpi10 219, zpi10f 219, zpi5 219, zpib5 219, zpic30 219, zpicxfe 219, zpieci 219, zrff10 219, zrff30 219, zrff5 219, zvpd 219, zvpi 219, zvps 219, zxbd 219, zxbi 219, zxbs 219, zyh 219, zyhp 219, zyht 219, and zynid 219.

6.3.5 srcEview/frbus.package/addins/ld.some.eqs/ld.some.eqs.prg

189 ⟨*srcEview/frbus.package/addins/ld.some.eqs/ld.some.eqs.prg* 189⟩≡

⟨*load some equations* 190⟩

⟨*load some equations call* 192⟩

This code is written to file srcEview/frbus.package/addins/ld.some.eqs/ld.some.eqs.prg.
Defines:

ld_some_eqs, used in chunks 171, 192, and 195b.

190 ⟨*load some equations* 190⟩≡ (189)

```
subroutine ld_some_eqs(string %mname, string %mpath, string %eqnames)

' Create eviews model %mname and load it with selected equations for frbus version
' %mname that are in a text file in directory %mpath that was previously created
' by the script eq_docs2eviews.  Only those equations whose names match or do not mat
' equation names in the string %eqnames are included.  If the first "word" in %eqname
' "allbut", then all equations but those listed will be included.

'**********************************
'parse equation string and put equation names in a table

  %allbut = "no"

  string zlist = " "
  zlist = zlist + %eqnames
  zlist = @trim(zlist)
     if @isempty(zlist) = 1 then
      @uiprompt("Error:  input string to subroutine load_selected_eqtions is empty!
      stop
      endif

  if  @wfind(@upper(zlist),"ALLBUT") = 1 then
     %allbut = "yes"
     zlist = @wdrop(zlist,"ALLBUT")
     zlist = @wdrop(zlist,"allbut")
  endif


'**********************************
'parse equation file

  %epath = %mpath + %mname  + "_eqs.txt"
  delete(noerr) eqtext
  text eqtext
  eqtext.append(file) %epath
  svector eqtextv = eqtext.@svectornb

  model {%mname}

  !eqnum = 0
  %eqnew = " "
  %eqold = " "
  %eqcode = " "
  %eqcodeold = " "
```

```
for !i = 1 to 3000
  %appendit = "no"
  %exitloop = "no"
  %y = eqtextv(!i)
  if @isempty(%y) = 0 then
    'string is not blank
    if @left(%y,6) = "theend" then
      %appendit = "yes"
      %exitloop = "yes"
      %eqold = %eqnew
      %eqcodeold = %eqcode
      endif

    !k = @instr(%y,":")
    if !k > 0 then
      'string contains the start of a new equation
      %appendit = "yes"
      !eqnum = !eqnum + 1
      if !eqnum > 0 then
        %eqold = %eqnew
        %eqcodeold = %eqcode
        endif
      %eqnew  = @left(%y,!k-1)
      %eqcode = @mid(%y,!k+1)

      else
      'string contains the continuation of an equation
      %eqcode = %eqcode + %y
      endif

    if %appendit = "yes" then
      'check whether equation should be added to model
      !zswitch = 0
      for !j = 1 to @wcount(zlist)
        %z = @lower(@word(zlist,!j))
        if %z = %eqold then
          !zswitch = 1
          endif
        if  %allbut = "no" and !zswitch = 1 then
          %x = @replace(%eqcodeold," _"," ")
          {%mname}.append {%x}
          exitloop
          endif
        if  %allbut = "yes" and !j = @wcount(zlist) and !zswitch = 0 then
          %x = @replace(%eqcodeold," _"," ")
```

```
            {%mname}.append {%x}
            endif
          next
        endif

       if %exitloop = "yes" then
         exitloop
         endif

       endif
     next



  endsub
```

Defines:
ld_some_eqs, used in chunks 171, 192, and 195b.

192 ⟨*load some equations call* 192⟩≡ (189)

```
  if @len(@option(1)) < 1  or @len(@option(2)) < 1 or @len(@option(3)) < 1 then
    @uiprompt("Error: ld_some_eqs requires model name and model pathand  and eqnames")
    stop
  endif

  %temp = @equaloption("modelname")
  if @len(%temp)>0 then
          %mname = %temp
  endif
  %temp = @equaloption("modelpath")
  if @len(%temp)>0 then
          %mpath = %temp
  endif
  %temp = @equaloption("eqnames")
  if @len(%temp)>0 then
          %eqnames = %temp
  endif

  call ld_some_eqs(%mname, %mpath, %eqnames)
```

Uses ld_some_eqs 189 190.

6.3.6 srcEview/frbus.package/addins/ld.varinfo/ld.varinfo.prg

193 ⟨*srcEview/frbus.package/addins/ld.varinfo/ld.varinfo.prg* 193⟩≡
⟨*load variable information* 194⟩

⟨*load variable information call* 195a⟩

This code is written to file `srcEview/frbus.package/addins/ld.varinfo/ld.varinfo.prg`.
Defines:
`ld_varinfo`, used in chunks 42, 195, and 204.

The ld_varinfo subroutine parses the frbus_package/mods/stdver_varinfo file.

194 ⟨*load variable information* 194⟩≡ (193)

```
subroutine ld_varinfo(string %pathname)

'Load varinfo information from %pathname in strings

  text vinfo_text
  vinfo_text.append(file) %pathname
  svector varinfo = vinfo_text.@svectornb
  string vinfo_vname = " "
  string vinfo_vtype = " "
  string vinfo_vrule = " "
  string vinfo_sector = " "
  string vinfo_stoch = " "
  string vinfo_decomp = " "
  for !i = 1 to 900
    %y1 = varinfo(!i)
    !ss = @instr(%y1," ")
    %vname = @mid(%y1,!ss+1,8)
    %vname = @rtrim(%vname)
    %vtype = @mid(%y1,!ss+107,1)
    %vrule = @mid(%y1,!ss+109,1)
    if %vrule = " " then
      %vrule = "0"
    endif
    %sector = @mid(%y1,!ss+120,1)
    if %sector = " " then
      %sector = "0"
    endif
    %stoch = @mid(%y1,!ss+128,2)
    %decomp = @mid(%y1,!ss+135,2)
    if %vname = "ZZZBLANK" then
      scalar vinfo_size = !i-1
      exitloop
      endif
   vinfo_vname = vinfo_vname + " " + %vname
   vinfo_vtype = vinfo_vtype + " " + %vtype
   vinfo_vrule = vinfo_vrule + " " + %vrule
   vinfo_sector = vinfo_sector + " " + %sector
   vinfo_stoch = vinfo_stoch + " " + %stoch
   vinfo_decomp = vinfo_decomp + " " + %decomp
  next

endsub
```

Defines:
ld_varinfo, used in chunks 42, 195, and 204.
Uses decomp 218, sector 218, stoch 218, varinfo 218, vinfo_decomp 218, vinfo_sector 218, vinfo_stoch 218, vinfo_text 218, vinfo_vname 218, vinfo_vrule 218, vinfo_vtype 218, vname 218, vrule 218, and vtype 218.

195a ⟨*load variable information call* 195a⟩≡ (193)

```
if @len(@option(1)) < 1  then
  @uiprompt("Error: ld_varinfo requires varinfo path")
  stop
endif

%temp = @equaloption("pathname")
if @len(%temp)>0 then
        %pathname = %temp
endif


call ld_varinfo(%pathname)
```

Uses ld_varinfo 193 194 and varinfo 218.

6.3.7 srcEview/frbus.package/addins/regadd.prg

195b ⟨*srcEview/frbus.package/addins/regadd.prg* 195b⟩≡

```
addin(proc="ld_frbus_eqs",desc="Load FRB/US equations") ".\ld_frbus_eqs/ld_frbus_eqs.prg"
addin(proc="ld_frbus_cfs",desc="Load FRB/US coefficients") ".\ld_frbus_cfs/ld_frbus_cfs.prg"
addin(proc="ld_mce_eqs",desc="Load MCE equations") ".\ld_mce_eqs/ld_mce_eqs.prg"
addin(proc="ld_mce_cfs",desc="Load MCE coefficients") ".\ld_mce_cfs/ld_mce_cfs.prg"
addin(proc="ld_some_eqs",desc="Load some FRB/US equations") ".\ld_some_eqs/ld_some_eqs.prg"
addin(proc="ld_varinfo",desc="Load variable information file") ".\ld_varinfo/ld_varinfo.prg"
```

This code is written to file srcEview/frbus.package/addins/regadd.prg.
Defines:
regadd, never used.
Uses ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, ld_mce_cfs 181b 182, ld_mce_eqs 184 185, ld_some_eqs 189 190, and ld_varinfo 193 194.

6.3.8 srcEview/frbus.package/programs/example1.prg

195c ⟨*srcEview/frbus.package/programs/example1.prg* 195c⟩≡
⟨*frbus example1* 5⟩

This code is written to file srcEview/frbus.package/programs/example1.prg.
Defines:
example1, never used.

6.3.9 srcEview/frbus.package/programs/example2.prg

196a ⟨*srcEview/frbus.package/programs/example2.prg* 196a⟩≡
⟨*frbus example2* 9⟩

This code is written to file `srcEview/frbus.package/programs/example2.prg`.
Defines:
`example2`, never used.

6.3.10 srcEview/frbus.package/programs/example3.prg

196b ⟨*srcEview/frbus.package/programs/example3.prg* 196b⟩≡
⟨*frbus example3* 13⟩

This code is written to file `srcEview/frbus.package/programs/example3.prg`.
Defines:
`example3`, never used.

6.3.11 srcEview/frbus.package/programs/example4.prg

196c ⟨*srcEview/frbus.package/programs/example4.prg* 196c⟩≡
⟨*frbus example4* 19⟩

This code is written to file `srcEview/frbus.package/programs/example4.prg`.
Defines:
`example4`, never used.

6.3.12 srcEview/frbus.package/programs/ocpolicy.prg

196d ⟨*srcEview/frbus.package/programs/ocpolicy.prg* 196d⟩≡
⟨*frbus ocpolicy* 25⟩

This code is written to file `srcEview/frbus.package/programs/ocpolicy.prg`.
Defines:
`ocpolicy`, never used.

6.3.13 srcEview/frbus.package/programs/pings.prg

197a ⟨*srcEview/frbus.package/programs/pings.prg* 197a⟩≡
⟨*simulate six ping simulations, aka simple IRFs* 33⟩
```
'*************************************************************
'*************************************************************
'*************************************************************
```
⟨*copy it* 197b⟩

```
'*************************************************************
'*************************************************************
'*************************************************************
```
⟨*plot it* 198⟩

```
'*************************************************************
'*************************************************************
'*************************************************************
```
⟨*graph it* 199⟩

This code is written to file `srcEview/frbus.package/programs/pings.prg`.
Defines:
`pings`, used in chunk 33.

197b ⟨*copy it* 197b⟩≡ (197a)
```
subroutine copyit

smpl %simstart %simend
series picnia_{%ping} = picnia{%suf} - picnia
series pic4_{%ping} = pic4{%suf} - pic4
series picx4_{%ping} = picx4{%suf} - picx4
series picxfe_{%ping} = picxfe{%suf} - picxfe
series xgap2_{%ping} = xgap2{%suf} -xgap2
series lur_{%ping} = lur{%suf} - lur
series rff_{%ping} = rff{%suf} - rff

endsub
```
Defines:
`copyit`, used in chunk 33.
Uses `lur` 219, `pic4` 219, `picnia` 219, `picx4` 219, `picxfe` 219, `rff` 219, and `xgap2` 219.

198 ⟨*plot it* 198⟩≡ (197a)

```
subroutine plotit(string %grname, string %width, string %height, string %var1, stri


graph {%grname}.line  {%var1} zero
{%grname}.options size({%width},{%height}) -inbox
{%grname}.setelem(1) linewidth(3) linepattern(1) linecolor(black)
{%grname}.addtext(t,just(c),font(9)) %title
{%grname}.addtext(0,-.15,font(8),just("r")) %units
{%grname}.datelabel format(yyyy)
{%grname}.legend -display
{%grname}.axis(b) font(9)
{%grname}.axis(l) font(9)

endsub
```

Defines:
plotit, used in chunk 199.

199 ⟨*graph it* 199⟩≡ (197a)

```
subroutine graphit

smpl %simstart %simstart + 39
series zero = 0

delete(noerr) gr_*


' RFF ping

%ping = "rff"

%name = %ping + "a"
%var1 =  "xgap2_" + %ping
%tt = "Response of Output Gap\rto Funds Rate"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "b"
%var1 =  "picxfe_" + %ping
%tt = "Response of Core Inflation\rto Funds Rate"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "c"
%var1 =  "rff_" + %ping
%tt = "Response of Funds Rate\rto Funds Rate"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
gr_{%ping}.align(3,.40,.40)



' EGFO ping


%ping = "eg"

%name = %ping + "a"
%var1 =  "xgap2_" + %ping
%tt = "Response of Output Gap\rto Federal Purch"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "b"
%var1 =  "picxfe_"  + %ping
%tt = "Response of Core Inflation\rto Federal Purch"
```

```
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "c"
%var1 =  "egfn_shr_" + %ping
%tt = "Response of Federal Purch\rto Federal Purch"
call plotit(%name,"2","1.5",%var1,%tt,"percent of GDP")

graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
gr_{%ping}.align(3,.40,.40)


' REQP ping

%ping = "reqp"

%name = %ping + "a"
%var1 =  "xgap2_" + %ping
%tt = "Response of Output Gap\rto Equity Premium"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "b"
%var1 =  "picxfe_" + %ping
%tt = "Response of Core Inflation\rto Equity Premium"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "c"
%var1 =  "reqp_" + %ping
%tt = "Response of Equity Premium\rto Equity Premium"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
gr_{%ping}.align(3,.40,.40)



' POILR ping

%ping = "oil"

%name = %ping + "a"
%var1 =  "xgap2_" + %ping
%tt = "Response of Output Gap\rto Oil Price"
call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

%name = %ping + "b"
%var1 =  "picxfe_" + %ping
```

```
  %tt = "Response of Core Inflation\rto Oil Price"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "c"
  %var1 =  "poil_" + %ping
  %tt = "Response of Oil Price\rto Oil Price"
  call plotit(%name,"2","1.5",%var1,%tt,"dollars per barrel")

  graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
  gr_{%ping}.align(3,.40,.40)



' HMFPT ping

  %ping = "hmfp"

  %name = %ping + "a"
  %var1 =  "xgap2_" + %ping
  %tt = "Response of Output Gap\rto MFP Growth"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "b"
  %var1 =  "picxfe_" + %ping
  %tt = "Response of Core Inflation\rto MFP Growth"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "c"
  %var1 =  "hmfpt_" + %ping
  %tt = "Response of MFP Growth\rto MFP Growth"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
  gr_{%ping}.align(3,.40,.40)



' MFPT ping

  %ping = "mfp"

  %name = %ping + "a"
  %var1 =  "xgap2_" + %ping
  %tt = "Response of Output Gap\rto MFP Level"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")
```

```
  %name = %ping + "b"
  %var1 =  "picxfe_" + %ping
  %tt = "Response of Core Inflation\rto MFP Level"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "c"
  %var1 =  "mfpt_" + %ping
  %tt = "Response of MFP Level\rto MFP Level"
  call plotit(%name,"2","1.5",%var1,%tt,"percent")

  graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
  gr_{%ping}.align(3,.40,.40)


%ping = "prem"

  %name = %ping + "a"
  %var1 =  "xgap2_" + %ping
  %tt = "Response of Output Gap\rto Term Premium"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "b"
  %var1 =  "picxfe_" + %ping
  %tt = "Response of Core Inflation\rto Term Premium"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "c"
  %var1 =  "rg10p_" + %ping
  %tt = "Response of Term Premium (10-year) \rto Term Premium"
  call plotit(%name,"2","1.5",%var1,%tt,"percent")

  graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
  gr_{%ping}.align(3,.40,.40)

%ping = "exch"

  %name = %ping + "a"
  %var1 =  "xgap2_" + %ping
  %tt = "Response of Output Gap\rto Exchange Rate"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")

  %name = %ping + "b"
  %var1 =  "picxfe_" + %ping
  %tt = "Response of Core Inflation\rto Exchange Rate"
  call plotit(%name,"2","1.5",%var1,%tt,"percentage points")
```

```
%name = %ping + "c"
%var1 =  "fpxr_" + %ping
%tt = "Response of  Exchange Rate \rto Exchange Rate"
call plotit(%name,"2","1.5",%var1,%tt,"percent")

graph gr_{%ping}.merge  {%ping}a {%ping}b {%ping}c
gr_{%ping}.align(3,.40,.40)
endsub
```

Defines:
graphit, used in chunk 33.
Uses plotit 198, reqp 219, and rff 219.

6.3.14 srcEview/frbus.package/programs/plot.resids.prg

203 ⟨*srcEview/frbus.package/programs/plot.resids.prg* 203⟩≡
⟨*plot historical residuals of key equations* 204⟩

```
' ***************************************************************
' ***************************************************************
```

⟨*find variable description* 206⟩

This code is written to file srcEview/frbus.package/programs/plot.resids.prg.
Defines:
plot_resids, never used.

204 ⟨*plot historical residuals of key equations* 204⟩≡ (203)

```
' Program to plot the historical residuals of key FRB/US equations
'
' Each residual has the same units of measurement as the
' the left hand side of its equation

' ************************************************************
' Initial filename and parameter settings
' ************************************************************

' Subroutines
  include ../subs/master_library

' Workfile
  %wfstart = "1975q1"
  %wfend = "2030q4"
  %mainpage = "main"
  wfcreate(wf=aaa,page={%mainpage}) q {%wfstart} {%wfend}

' FRB/US model name and location
  %varmod = "stdver"
  %varpath = "../mods/"
  %varinfo = "../mods/stdver_varinfo"

' Input datbase
  %dbin  = "../data/longbase"

' Plot range
  %plotstart = "1980q1"
  %plotend = "2014q2"


' ************************************************************
' Retrieve data, model equations and coefficients
' ************************************************************

' Load equations and coefficients
  ld_frbus_eqs(modelname=%varmod,modelpath=%varpath)
  ld_frbus_cfs(modelname=%varmod,modelpath=%varpath)
  ld_varinfo(pathname=%varinfo)

' Load data
  dbopen %dbin as longbase
  smpl %plotstart-12 %plotend
  fetch(d=longbase) *
```

```
' Set _aerr variables to zero
  smpl @all
  {%varmod}.makegroup(a,n) endog @endog
  call groupnew("endog","_aerr")
  call group2zero("endog_aerr")

' Compute baseline tracking add factors
  smpl %plotstart %plotend
  {%varmod}.addassign @all
  {%varmod}.addinit(v=n) @all

' ************************************************************
' Plots
' ************************************************************

  %plotvars = "eco ecd eh epd epi eps ki ex emo lfpr lhp leo lww"
  %plotvars = %plotvars + " picxfe pieci pcer pcfr pcengr"
  %plotvars = %plotvars + " rg5p rg10p rg30p rbbbp rcar rme rtbe rcgain"
  %plotvars = %plotvars + " ynidn yniin yhibn"

  smpl %plotstart %plotend
  series zero = 0

  spool plot_vars

  !counter = 0
  for !i = 1 to @wcount(%plotvars)
    %vname = @word(%plotvars,!i)
    call find_var_description
    !counter = !counter + 1
    graph gr_{%vname}.line zero {%vname}_a
    %title = %vname + ": " + %desc
    gr_{%vname}.addtext(t) %title
    gr_{%vname}.axis range(minmax)
    gr_{%vname}.options size(4,3)
    gr_{%vname}.legend -display

    plot_vars.append gr_{%vname}
    %index = "000" + @str(!counter)
    if !counter < 100 then
      %index = @right(%index,2)
      else
      %index = @right(%index,3)
      endif
    %name = "untitled" + %index
    plot_vars.name {%name} {%vname}
```

```
    next

  plot_vars.display
```

Uses dbopen 217a, ecd 219, eco 219, eh 219, emo 219, epd 219, epi 219, eps 219, ex 219, fetch 217a, find_var_description 206, group2zero 91a, groupnew 92, ki 219, ld_frbus_cfs 177 178a, ld_frbus_eqs 179 180, ld_varinfo 193 194, left 219, leo 219, lfpr 219, lhp 219, longbase 217b, lww 219, master_library 209a, pcengr 219, pcer 219, pcfr 219, picxfe 219, pieci 219, rbbbp 219, rcar 219, rcgain 219, rg10p 219, rg30p 219, rg5p 219, rme 219, rtbe 219, varinfo 218, vname 218, wfcreate 217a, yhibn 219, ynidn 219, and yniin 219.

This was originally written in Fortran! Look at the ii, jj, kk variable names. I had forgotten about that from when I briefly helped the programmer debug the original model some 45 years ago. Who would have expected nostalgia from something so prosaic. I never saw the original source code because I was working with only a hex dump.

206 ⟨*find variable description* 206⟩≡ (203)

```
  subroutine find_var_description

  for !j = 1 to 500
    %vline = vinfo_text.@line(!j)
    !eq = @instr(%vline,"=")
    !zz = !eq-4
    %name = @lower(@rtrim(@ltrim(@mid(%vline,4,!zz))))

    if %vname = %name then
      %desc = " "
      !ii = !eq + 1
      !kk = @instr(%vline,"sector_")
      if !kk = 0 then
        !kk = @instr(%vline,"X.")
        !jj = !kk - 1 - !ii
        else
        !jj = !kk - 7 - !ii
        endif
      %desc = @rtrim(@mid(%vline,!ii,!jj))
      exitloop
      endif
    next
  endsub
```

Defines:
find_var_description, used in chunk 204.
Uses vinfo_text 218 and vname 218.

6.3.15 srcEview/frbus.package/programs/stochsim.prg

207a ⟨*srcEview/frbus.package/programs/stochsim.prg* 207a⟩≡
⟨*stochastic simulations under variable expectations* 42⟩

```
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
'Subroutines


'*******************************************************************************
'*******************************************************************************
```
⟨*form table* 207b⟩

```
'*******************************************************************************
'*******************************************************************************
```
⟨*make statistics* 208⟩

This code is written to file `srcEview/frbus.package/programs/stochsim.prg`.
Defines:
`stochsim`, never used.

207b ⟨*form table* 207b⟩≡ (207a)

```
subroutine tableform(string %tabname, string %nrows)

  table(@val(%nrows),9) {%tabname}

  {%tabname}.setwidth(1:9) 8
  {%tabname}.setjust(r1c1:r{%nrows}c9) right
  {%tabname}.setformat(r2c2:r{%nrows}c9) f.3
  {%tabname}(1,1) = "qtr"
  {%tabname}(1,2) = "baseline"
  {%tabname}(1,3) = "mean"
  {%tabname}(1,4) = "median"
  {%tabname}(1,5) = "stdev"
  {%tabname}(1,6) = "90%-low"
  {%tabname}(1,7) = "90%-hi"
  {%tabname}(1,8) = "70%-low"
  {%tabname}(1,9) = "70%-hi"

  endsub
```

Defines:
`tableform`, used in chunk 42.

208 ⟨*make statistics* 208⟩≡ (207a)

```
subroutine makestats(string %trkname)

  %trkmat = %trkname + "_mat"
  %statsmat = %trkname + "_stats"
  matrix(nqtrs,8) {%statsmat}

  smpl {%simstart} {%simend}

' loop over each simulation quarter
  for !ii2 = 1 to nqtrs
  ' put simulation replications for this quarter into matrix tempm1
    matrix tempm1 = @sort(@rowextract({%trkmat},!ii2))
    {%statsmat}(!ii2,1) = {%trkname}(!lqtr + !ii2)
    {%statsmat}(!ii2,2) = @mean(tempm1)
    {%statsmat}(!ii2,3) = tempm1(1,@floor(.50*nrepl))
    {%statsmat}(!ii2,4) = @stdev(tempm1)
    {%statsmat}(!ii2,5) = tempm1(1,@floor(.05*nrepl))
    {%statsmat}(!ii2,6) = tempm1(1,@floor(.95*nrepl))
    {%statsmat}(!ii2,7) = tempm1(1,@floor(.15*nrepl))
    {%statsmat}(!ii2,8) = tempm1(1,@floor(.85*nrepl))

    next

' also create individual series for each statistic
  series {%trkname}_base = 0
  series {%trkname}_mn = 0
  series {%trkname}_med  = 0
  series {%trkname}_se  = 0
  series {%trkname}_lo90 = 0
  series {%trkname}_hi90 = 0
  series {%trkname}_lo70 = 0
  series {%trkname}_hi70 = 0

  group {%trkname}_group {%trkname}_base {%trkname}_mn {%trkname}_med _
        {%trkname}_se {%trkname}_lo90 {%trkname}_hi90 {%trkname}_lo70 _
        {%trkname}_hi70
  mtos({%statsmat},{%trkname}_group)

  endsub
```

Defines:
makestats, used in chunk 42.

6.3.16 srcEview/frbus.package/subs/master.library.prg

209a ⟨*srcEview/frbus.package/subs/master.library.prg* 209a⟩≡
⟨*master library routines* 209b⟩

This code is written to file `srcEview/frbus.package/subs/master.library.prg`.
Defines:
`master_library`, used in chunks 5, 9, 13, 19, 25, 33, 42, and 204.

209b ⟨*master library routines* 209b⟩≡ (209a)

```
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨quarterly date string shift 89⟩


'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨copy series into group 90⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨set group to zero 91a⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨names of all series in group 91b⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨create new group 92⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨interpolate unavailable observations 93⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨set fiscal policy option 94⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨set monetary policy option 95⟩
'*******************************************************************************
'*******************************************************************************
'*******************************************************************************
⟨set monetary policy fed funds rate 96⟩
```

6.3.17 srcEview/frbus.package/subs/mce.solve.library.prg

210 ⟨*srcEview/frbus.package/subs/mce.solve.library.prg* 210⟩≡
⟨*mce solve library* 211⟩

This code is written to file `srcEview/frbus.package/subs/mce.solve.library.prg`.
Defines:
`mce_solve_library`, used in chunks 9, 19, 25, 33, 51, 54, 57, 61, and 64.

211 ⟨*mce solve library* 211⟩≡ (210 215b)
⟨*mce solve library change history* 97⟩

```
' ********************************************************************* '
' ********************************************************************* '
' *********************************************************************
```
⟨*run model consistent expectations* 99⟩

```
'*************************************************************************
'*************************************************************************
'*************************************************************************
```
⟨*determine endogenous and exogenous variables* 106⟩

```
' ********************************************************************* '
' ********************************************************************* '
' *********************************************************************
```
⟨*determine default method, linesearch, and other options* 109⟩

```
' ************************************************************
' ************************************************************
' ************************************************************
```
⟨*parse options containing equal signs* 113a⟩

```
' ************************************************************
' ************************************************************
' ************************************************************
```
⟨*parse options not containing equal signs* 113b⟩

```
' ************************************************************************
' ************************************************************************
' ************************************************************************
' ************************************************************************
```
⟨*create common variables, strings, matrices, vectors, and tables* 114⟩

```
' ********************************************************************* '
' ********************************************************************* '
' *********************************************************************
```
⟨*model consistent coefficient simulation* 116⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*solve model consistent instrument values* 125⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*compute derivatives of mce targets wrt mce instruments* 127⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*model consistent coefficient non-monotone step-length procedure* 133⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*model consistent armijo optimization rule* 135⟩

```
'*****************************************************************************
'*****************************************************************************
'*****************************************************************************
```
⟨*variable terminal values* 136⟩

```
'*****************************************************************************
'*****************************************************************************
'*****************************************************************************
```

⟨*set options based on defaults and overrides* 138⟩

```
'*****************************************************************************
'*****************************************************************************
'*****************************************************************************
```

⟨*main unconstrained optimal control simulation* 145⟩

```
'*************************************************************************
'*************************************************************************
'*************************************************************************
```
⟨*main optimal control simulation with inequality constraints* 149⟩

```
'*************************************************************************
'*************************************************************************
'*************************************************************************
```
⟨*model consistent optimal time-consistent solution* 154⟩

```
'*************************************************************************
'*************************************************************************
'*************************************************************************
```
⟨*compute derivatives of loss function targets wrt instruments* 162⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*solve model consistent expectations model* 164⟩

```
'*****************************************************************
'*****************************************************************
'*****************************************************************
```
⟨*convert mcz inequality constraints* 166⟩

```
'*************************************************************
'*************************************************************
'*************************************************************
```
⟨*shift left* 169⟩

```
'*************************************************************
'*************************************************************
'*************************************************************
```
⟨*find next delimiter* 170⟩

```
' ********************************************************************* '
' ********************************************************************* '
' *********************************************************************
```
⟨*load frbus with transformed subsidiary model* 171⟩

```
' ********************************************************************* '
' ********************************************************************* '
' *********************************************************************
```
⟨*create wage and expectation variables in forward looking model* 175⟩

6.4 mce solve package

6.4.1 srcEview/mce.solve.package/example1.prg

214a ⟨*srcEview/mce.solve.package/example1.prg* 214a⟩≡
⟨*mce example1* 51⟩

This code is written to file `srcEview/mce.solve.package/example1.prg`.
Defines:
`example1`, never used.

6.4.2 srcEview/mce.solve.package/example2.prg

214b ⟨*srcEview/mce.solve.package/example2.prg* 214b⟩≡
⟨*mce example2* 54⟩

This code is written to file `srcEview/mce.solve.package/example2.prg`.
Defines:
`example2`, never used.

6.4.3 srcEview/mce.solve.package/example3.prg

214c ⟨*srcEview/mce.solve.package/example3.prg* 214c⟩≡
⟨*mce example3* 57⟩

This code is written to file `srcEview/mce.solve.package/example3.prg`.
Defines:
`example3`, never used.

6.4.4 srcEview/mce.solve.package/example4.prg

214d ⟨*srcEview/mce.solve.package/example4.prg* 214d⟩≡
⟨*mce example4* 61⟩

This code is written to file `srcEview/mce.solve.package/example4.prg`.
Defines:
`example4`, never used.

6.4.5 srcEview/mce.solve.package/example5.prg

215a ⟨*srcEview/mce.solve.package/example5.prg* 215a⟩≡
⟨*mce example5* 64⟩

This code is written to file `srcEview/mce.solve.package/example5.prg`.
Defines:
`example5`, never used.

6.4.6 srcEview/mce.solve.package/mce.solve.library.prg

215b ⟨*srcEview/mce.solve.package/mce.solve.library.prg* 215b⟩≡
⟨*mce solve library* 211⟩

This code is written to file `srcEview/mce.solve.package/mce.solve.library.prg`.
Defines:
`mce_solve_library`, used in chunks 9, 19, 25, 33, 51, 54, 57, 61, and 64.

6.5 state space package

6.5.1 srcEview/state.space.package/data.transformations.prg

215c ⟨*srcEview/state.space.package/data.transformations.prg* 215c⟩≡
⟨*data transformations* 73⟩

This code is written to file `srcEview/state.space.package/data.transformations.prg`.
Defines:
`data_transformations`, used in chunks 69, 73, and 80.

6.5.2 srcEview/state.space.package/estimation.code.prg

215d ⟨*srcEview/state.space.package/estimation.code.prg* 215d⟩≡
⟨*estimation code* 76⟩

This code is written to file `srcEview/state.space.package/estimation.code.prg`.
Defines:
`estimation_code`, used in chunk 69.

6.5.3 srcEview/state.space.package/frbus.supply.estimation.prg

215e ⟨*srcEview/state.space.package/frbus.supply.estimation.prg* 215e⟩≡
⟨*estimation model* 69⟩

This code is written to file `srcEview/state.space.package/frbus.supply.estimation.prg`.
Defines:
`frbus_supply_estimation`, never used.

6.5.4 srcEview/state.space.package/frbus.supply.filter.prg

216a ⟨*srcEview/state.space.package/frbus.supply.filter.prg* 216a⟩≡
⟨*supply filter* 80⟩

This code is written to file `srcEview/state.space.package/frbus.supply.filter.prg`.
Defines:
`frbus_supply_filter`, used in chunk 80.

6.5.5 srcEview/state.space.package/initial.values.prg

216b ⟨*srcEview/state.space.package/initial.values.prg* 216b⟩≡
⟨*initial values* 83⟩

This code is written to file `srcEview/state.space.package/initial.values.prg`.
Defines:
`initial_values`, used in chunks 69 and 76.

Notes, Bibliography and Indexes

6.6 Eview Functions

217a ⟨*Presumed Eview Functions* 217a⟩≡

Defines:
`close`, used in chunks 69, 76, 80, 116, and 138.
`dbopen`, used in chunks 5, 9, 13, 19, 25, 33, 42, 69, 80, and 204.
`fetch`, used in chunks 5, 9, 13, 19, 25, 33, 42, 69, 80, and 204.
`solve`, used in chunks 5, 9, 13, 19, 25, 33, 42, 125, 127, 145, 149, and 154.
`sspace`, used in chunk 76.
`wfclose`, used in chunk 80.
`wfcreate`, used in chunks 5, 9, 13, 19, 25, 33, 42, 51, 54, 57, 61, 64, 69, 80, and 204.
`wfopen`, used in chunk 80.
`wfsave`, used in chunk 69.
`xget`, used in chunks 149 and 154.
`xrun`, used in chunks 149 and 154.

6.7 Data File Name

217b ⟨*index data file name* 217b⟩≡

Defines:
`longbase`, used in chunks 5, 9, 13, 19, 25, 33, 42, and 204.

6.8 Variable Information

218 ⟨*Presumed varinfo Schema* 218⟩≡

Defines:
- `decomp`, used in chunk 194.
- `sector`, used in chunks 9, 19, 69, 73, 76, 80, and 194.
- `stoch`, used in chunks 42 and 194.
- `varinfo`, used in chunks 42, 194, 195a, and 204.
- `vdesc`, never used.
- `vinfo_decomp`, used in chunk 194.
- `vinfo_sector`, used in chunk 194.
- `vinfo_stoch`, used in chunks 42 and 194.
- `vinfo_text`, used in chunks 194 and 206.
- `vinfo_vname`, used in chunks 42 and 194.
- `vinfo_vrule`, used in chunk 194.
- `vinfo_vtype`, used in chunk 194.
- `vname`, used in chunks 42, 194, 204, and 206.
- `vrule`, used in chunk 194.
- `vtype`, used in chunk 194.

6.9 Series Names

219 ⟨*Model Data Series Names* 219⟩≡

Defines:
- ceng, never used.
- d01q4, never used.
- d2002, never used.
- d2003, never used.
- d69, never used.
- d79a, never used.
- d8095, never used.
- d81, never used.
- d83, never used.
- d86, never used.
- d87, never used.
- dcon, never used.
- ddockm, never used.
- ddockx, never used.
- delrff, used in chunk 25.
- deuc, never used.
- dfmprr, never used.
- dfpdbt, used in chunks 25 and 94.
- dfpex, used in chunks 25, 42, and 94.
- dfpsrp, used in chunks 5, 9, 13, 19, 33, and 94.
- dglprd, never used.
- dmpalt, used in chunk 95.
- dmpex, used in chunks 42 and 95.
- dmpgen, used in chunks 19 and 95.
- dmpintay, used in chunks 5, 9, 33, 42, and 95.
- dmprr, used in chunk 95.
- dmpstb, used in chunk 42.
- dmptay, used in chunks 25 and 95.
- dmptlr, used in chunk 95.
- dmptlur, never used.
- dmptmax, never used.
- dmptpi, never used.
- dmptr, used in chunks 13 and 42.
- dmptrsh, used in chunks 5, 9, 13, 19, 25, 33, and 42.
- dpadj, never used.
- dpgap, never used.
- drstar, used in chunks 5, 9, 13, 19, 25, 33, and 42.
- ec, used in chunk 76.
- ecd, used in chunk 204.
- ech, never used.
- ecnia, never used.
- ecnian, never used.
- eco, used in chunk 204.
- egf, never used.
- egfi, never used.
- egfin, never used.
- egfit, never used.
- egfl, never used.
- egfln, never used.
- egflt, never used.
- egfn, used in chunk 33.
- egfo, never used.
- egfon, used in chunk 33.

egfot, never used.
egpdin, never used.
egs, never used.
egsi, never used.
egsin, never used.
egsit, never used.
egsl, never used.
egsln, never used.
egslt, never used.
egsn, never used.
egso, never used.
egson, never used.
egsot, never used.
eh, used in chunk 204.
ehn, never used.
ei, never used.
ein, never used.
em, never used.
emn, never used.
emo, used in chunk 204.
emon, used in chunks 69 and 73.
emp, never used.
empn, never used.
empt, never used.
epd, used in chunk 204.
epdn, never used.
epi, used in chunk 204.
epin, never used.
eps, used in chunk 204.
epsn, never used.
ex, used in chunks 69, 80, and 204.
exn, never used.
fcbn, never used.
fcbrn, never used.
fgdp, never used.
fgdpt, never used.
fnicn, never used.
fniln, never used.
fnin, never used.
fnirn, never used.
fpc, never used.
fpcm, never used.
fpi10, never used.
fpi10t, never used.
fpic, never used.
fpitrg, never used.
fpx, never used.
fpxm, never used.
fpxr, used in chunk 33.
fpxrr, never used.
fpxrrt, never used.
frl10, never used.
frs10, never used.
frstar, never used.
ftcin, never used.
fxgap, never used.
fynicn, never used.

fyniln, never used.
fynin, never used.
gfdbtn, never used.
gfdrt, never used.
gfintn, never used.
gfs, never used.
gfsn, never used.
gfsrpn, never used.
gfsrt, never used.
gfsub, never used.
gfsubn, never used.
gft, never used.
gftn, never used.
gftrd, never used.
gftrt, never used.
gsdbtn, never used.
gsdrt, never used.
gsintn, never used.
gssrpn, never used.
gssrt, never used.
gssub, never used.
gssubn, never used.
gst, never used.
gstn, never used.
gstrd, never used.
gstrt, never used.
hgemp, never used.
hggdp, used in chunk 42.
hggdpt, never used.
hgpcdr, never used.
hgpdr, never used.
hgpir, never used.
hgpkir, never used.
hgppsr, never used.
hgvpd, never used.
hgvpi, never used.
hgvps, never used.
hgx, never used.
hgynid, never used.
hks, never used.
hksr, never used.
hlept, never used.
hlprdt, never used.
hmfpt, used in chunks 33, 69, and 80.
hqlfpr, used in chunks 69 and 80.
hqlww, used in chunks 69 and 80.
huqpct, never used.
huxb, used in chunks 69 and 80.
hxbt, never used.
jccacn, never used.
jccan, never used.
jkcd, never used.
jrcd, never used.
jrh, never used.
jrpd, never used.
jrpi, never used.
jrps, never used.

pic4, used in chunks 5, 9, 13, 19, 25, and 197b.
picngr, never used.
picnia, used in chunks 42 and 197b.
picx4, used in chunks 19, 42, and 197b.
picxfe, used in chunks 42, 96, 197b, and 204.
pieci, used in chunk 204.
pigdp, never used.
pipl, never used.
pipxnc, never used.
pitarg, used in chunk 19.
pitrsh, used in chunk 13.
pkir, never used.
pkpdr, never used.
pl, never used.
plmin, never used.
plminr, never used.
pmo, used in chunks 69 and 73.
pmp, never used.
poil, used in chunk 33.
poilr, never used.
poilrt, never used.
ppdr, never used.
ppir, never used.
ppsr, never used.
ptr, used in chunks 19, 69, and 76.
pwstar, never used.
pxb, used in chunks 33, 69, and 73.
pxg, never used.
pxnc, never used.
pxp, never used.
pxr, never used.
qec, never used.
qecd, never used.
qeco, never used.
qeh, never used.
qepd, never used.
qepi, never used.
qeps, never used.
qkir, never used.
qleor, never used.
qlep, never used.
qlf, never used.
qlfpr, used in chunks 69 and 80.
qlhp, never used.
qlww, used in chunks 69 and 80.
qpcnia, never used.
qpl, never used.
qpmo, never used.
qpxg, never used.
qpxnc, never used.
qpxp, never used.
qynidn, never used.
rbbb, never used.
rbbbe, never used.
rbbbp, used in chunk 204.
rcar, used in chunk 204.
rccd, never used.

tfsin, never used.
trfci, never used.
trfcim, never used.
trfib, never used.
trfp, never used.
trfpm, never used.
trfpt, never used.
trfptx, never used.
trfsi, never used.
trsci, never used.
trscit, never used.
trsib, never used.
trsibt, never used.
trsp, never used.
trspp, never used.
trspt, never used.
trsptx, never used.
trssi, never used.
trssit, never used.
tryh, never used.
tscin, never used.
tsibn, never used.
tspn, never used.
tssin, never used.
uces, used in chunks 69 and 73.
ucfs, never used.
uemot, never used.
uemp, never used.
ufcbr, never used.
ufnir, never used.
ufpcm, never used.
ufpxm, never used.
uftcin, never used.
ugfdbt, never used.
ugsdbt, never used.
ugsint, never used.
ugssub, never used.
ujcca, never used.
ujccac, never used.
ujygfe, never used.
ujygfg, never used.
ujygse, never used.
ujygsg, never used.
ulef, never used.
ules, never used.
upcpi, never used.
upcpix, never used.
upgfl, never used.
upgsl, never used.
upkpd, never used.
upmp, never used.
upxb, never used.
uqpct, never used.
uveoa, never used.
uvpd, never used.
uvpi, never used.
uvps, never used.

yhp, never used.
yhpcd, never used.
yhpgap, never used.
yhpntn, never used.
yhpshr, never used.
yhptn, never used.
yhshr, never used.
yhsn, never used.
yht, never used.
yhtgap, never used.
yhtn, never used.
yhtshr, never used.
ykin, never used.
ykpdn, never used.
ykpsn, never used.
ymsdn, never used.
ynicpn, never used.
ynidn, used in chunk 204.
yniin, used in chunk 204.
yniln, never used.
ynin, never used.
ynisen, never used.
ypn, never used.
zdivgr, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zecd, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zeco, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zeh, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zgap05, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zgap10, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zgap30, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zgapc2, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zlhp, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpi10, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpi10f, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpi5, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpib5, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpic30, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpic58, used in chunks 9, 19, 25, 33, 171, and 175.
zpicxfe, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zpieci, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zrff10, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zrff30, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zrff5, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zvpd, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zvpi, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zvps, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zxbd, used in chunks 9, 19, 25, 33, and 188.
zxbi, used in chunks 9, 19, 33, and 188.
zxbs, used in chunks 9, 19, 33, and 188.
zyh, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zyhp, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zyhpst, never used.
zyhst, never used.
zyht, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.
zyhtst, never used.
zynid, used in chunks 9, 19, 25, 33, 171, 175, 183, and 188.

6.10 Chunks

⟨*compute derivatives of loss function targets wrt instruments* 162⟩
⟨*compute derivatives of mce targets wrt mce instruments* 127⟩
⟨*convert mcz inequality constraints* 166⟩
⟨*copy it* 197b⟩
⟨*copy series into group* 90⟩
⟨*create common variables, strings, matrices, vectors, and tables* 114⟩
⟨*create new group* 92⟩
⟨*create wage and expectation variables in forward looking model* 175⟩
⟨*data transformations* 73⟩
⟨*determine default method, linesearch, and other options* 109⟩
⟨*determine endogenous and exogenous variables* 106⟩
⟨*estimation code* 76⟩
⟨*estimation model* 69⟩
⟨*find next delimiter* 170⟩
⟨*find variable description* 206⟩
⟨*form table* 207b⟩
⟨*frbus example1* 5⟩
⟨*frbus example2* 9⟩
⟨*frbus example3* 13⟩
⟨*frbus example4* 19⟩
⟨*frbus ocpolicy* 25⟩
⟨*graph it* 199⟩
⟨*index data file name* 217b⟩
⟨*initial values* 83⟩
⟨*interpolate unavailable observations* 93⟩
⟨*load frbus coefficients* 178a⟩
⟨*load frbus coefficients call* 178b⟩
⟨*load frbus equations* 180⟩
⟨*load frbus equations call* 181a⟩
⟨*load frbus with transformed subsidiary model* 171⟩
⟨*load mce coefficients* 182⟩
⟨*load mce coefficients call* 183⟩
⟨*load mce equations* 185⟩
⟨*load mce equations call* 188⟩
⟨*load some equations* 190⟩
⟨*load some equations call* 192⟩
⟨*load variable information* 194⟩
⟨*load variable information call* 195a⟩
⟨*main optimal control simulation with inequality constraints* 149⟩
⟨*main unconstrained optimal control simulation* 145⟩
⟨*make statistics* 208⟩
⟨*master library routines* 209b⟩
⟨*mce example1* 51⟩

⟨*srcEview/mce.solve.package/example1.prg* 214a⟩
⟨*srcEview/mce.solve.package/example2.prg* 214b⟩
⟨*srcEview/mce.solve.package/example3.prg* 214c⟩
⟨*srcEview/mce.solve.package/example4.prg* 214d⟩
⟨*srcEview/mce.solve.package/example5.prg* 215a⟩
⟨*srcEview/mce.solve.package/mce.solve.library.prg* 215b⟩
⟨*srcEview/state.space.package/data.transformations.prg* 215c⟩
⟨*srcEview/state.space.package/estimation.code.prg* 215d⟩
⟨*srcEview/state.space.package/frbus.supply.estimation.prg* 215e⟩
⟨*srcEview/state.space.package/frbus.supply.filter.prg* 216a⟩
⟨*srcEview/state.space.package/initial.values.prg* 216b⟩
⟨*stochastic simulations under variable expectations* 42⟩
⟨*supply filter* 80⟩
⟨*variable terminal values* 136⟩

6.11 Index