Reverse Engineering the FRB/US Model in R

Gary Young

June 12, 2016

Contents

Chapter 1

Introduction

I am starting to reverse engineer[1] the Federal Reserve's FRB/US model packages to create my own version in the R Language. I quote their about page:

> The FRB/US model is a large-scale estimated general equilibrium model of the U.S. economy that has been in use at the Federal Reserve Board since 1996. The model is designed for detailed analysis of monetary and fiscal policies. One distinctive feature compared to dynamic stochastic general equilibrium (DSGE) models is the ability to switch between alternative assumptions about expectations formation of economic agents. Another is the models level of detail: FRB/US contains all major components of the product and income sides of the U.S. national accounts. Since its original development, the model has continuously undergone changes to cope with the evolving structure of the economy, including conceptual revisions to sectoral definitions of the national accounts.
>
> The article "The FRB/US Model: A Tool for Macroeconomic Policy Analysis" provides a brief overview of the structure of FRB/US, and presents some key properties of the model and some applications, code for which is included with the main FRB/US model package. The article "November 2014 Update of the FRB/US Model" presents some model properties of the most recently released version of FRB/US.

This is an evolving document, where I will initially create the Fed's model files byte for byte and reverse engineer the structure of the model. Then I

[1] The pdf was created with noweb, the literate programming tool: "noweb frbus.nw — pdflatex -synctex=1 -interaction=nonstopmode frbus.tex"

plan to morph it into the R software environment for statistical computing and graphics, to use to create my own models. I'm using the literate programming method of Donald Knuth to combine the documentation with the actual code.

Appendices

Appendix A

Original Files

I'll produce the files so that it can be compared byte for byte to the originals; "variables.txt", "stdver_eqs.txt", and "stdver_coeffs.txt" in the FRB/US dataset and variable listing (ZIP) (Updated database: March 17, 2016). Because of file name limitations with noweb, I've had to modify the file names that I create somewhat.

A.1 Standard Version Variable Information File

9 ⟨*stdver.varinfo* 9⟩≡

```
CENG     = Consumption of crude energy (oil, coal, natural gas),  2009 $
D01Q4    = Dummy, destruction of World Trade Center
D2002    = Dummy,
D2003    = Dummy,
D69      = Dummy, post-1968 indicator
D79A     = Dummy, post-1979 indicator
D8095    = Dummy, 1980-1995 indicator
D81      = Dummy, post-1980 indicator
D83      = Dummy, post-1983 indicator
D86      = Dummy, post-1985 indicator
D87      = Dummy, post-1986 indicator
DCON     = Dummy, 0 prior to 1986, 1 after 1988, with a linear trend in between
DDOCKM   = Dock strike dummy, import equation
DDOCKX   = Dock strike dummy, export equation
DELRFF   = Federal funds rate, first diff
DEUC     = EUC switch:  1 for including EUC, 0 for not including
DFMPRR   = Dummy, Foreign monetary policy switch:  Exogenous real interest rate
DFPDBT   = Fiscal policy switch:  1 for debt ratio stabilization
DFPEX    = Fiscal policy switch:  1 for exogenous personal income trend tax rates
DFPSRP   = Fiscal policy switch:  1 for surplus ratio stabilization
DGLPRD   = Switch to control for long-run productivity growth in the government sector
```

```
22 DMPALT   = Monetary policy switch: MA rule
23 DMPEX    = Monetary policy switch:  exogenous federal funds rate
24 DMPGEN   = Monetary policy switch:  Generalized reaction function
25 DMPINTAY = Monetary policy switch:  inertial taylor rule
26 DMPRR    = Monetary policy switch:  exogenous real federal funds rate
27 DMPSTB   = Stabilization  switch:  0 for standard applications, 1 for stochastic
28 DMPTAY   = Monetary policy switch:  Taylor's reaction function
29 DMPTLR   = Monetary policy switch:  Taylor's reaction function with unemployment
30 DMPTLUR  = Monetary policy indicator for unemployment threshold
31 DMPTMAX  = Monetary policy indicator for both thresholds
32 DMPTPI   = Monetary policy indicator for inflation threshold
33 DMPTR    = Monetary policy indicator for policy rule thresholds
34 DMPTRSH  = Monetary policy threshold switch:  0 for no threshold,  1 for thresho
35 DPADJ    = Price inflation aggregation adjustment
36 DPGAP    = Price inflation aggregation discrepancy
37 DRSTAR   = RSTAR updating switch: 1 is on, 0 is off
38 EC       = Consumption, cw 2009$ (FRB/US definition)
39 ECD      = Consumer expenditures on durable goods, cw 2009$
40 ECH      = Consumer expenditures on housing services, cw 2009$
41 ECNIA    = Personal consumption expenditures, cw 2009$ (NIPA definition)
42 ECNIAN   = Personal consumption expenditures, current $ (NIPA definition)
43 ECO      = Consumer expenditures on non-durable goods and non-housing services, c
44 EGF      = Federal government consumption and gross investment, cw 2009$
45 EGFI     = Federal government gross investment, cw 2009$
46 EGFIN    = Federal government gross investment, current $
47 EGFIT    = Federal government gross investment, cw 2009$, trend
48 EGFL     = Federal government employee compensation, cw 2009$
49 EGFLN    = Federal government employee compensation, current $
50 EGFLT    = Federal government employee compensation, cw 2009$, trend
51 EGFN     = Federal government consumption and gross investment, current $
52 EGFO     = Federal government consumption ex. employee comp., cw 2009$
53 EGFON    = Federal government consumption ex. employee comp., current $
54 EGFOT    = Federal government consumption ex. employee comp., cw 2009$, trend
55 EGPDIN   = Gross private domestic investment
56 EGS      = S&L government consumption and gross investment, cw 2009$
57 EGSI     = S&L government gross investment, cw 2009$
58 EGSIN    = S&L government gross investment, current $
59 EGSIT    = S&L government gross investment, cw 2009$, trend
60 EGSL     = S&L government employee compensation, cw 2009$
61 EGSLN    = S&L government employee compensation, current $
62 EGSLT    = S&L government employee compensation, cw 2009$, trend
63 EGSN     = S&L government consumption and gross investment, current $
64 EGSO     = S&L government consumption ex. employee comp., cw 2009$
65 EGSON    = S&L government consumption ex. employee comp., current $
66 EGSOT    = S&L government consumption ex. employee comp., cw 2009$, trend
67 EH       = Residential investment expenditures, cw 2009$
```

```
 68 EHN      = Residential investment expenditures
 69 EI       = Change in private inventories, cw 2009$
 70 EIN      = Change in business inventories, current $
 71 EM       = Imports of goods and services, cw 2009$
 72 EMN      = Imports of goods and services, current $
 73 EMO      = Imports of goods and services ex. petroleum, cw 2009$
 74 EMON     = Imports of goods and services ex. petroleum
 75 EMP      = Petroleum imports, cw 2009$
 76 EMPN     = Petroleum imports, current $
 77 EMPT     = Petroleum imports trend, cw 2009$
 78 EPD      = Investment in equipment, cw 2009$
 79 EPDN     = Investment in equipment, current $
 80 EPI      = Investment in intellectual property, cw 2009$
 81 EPIN     = Investment in intellectual property, current $
 82 EPS      = Investment in nonresidential structures, cw 2009$
 83 EPSN     = Investment in nonresidential structures, current $
 84 EX       = Exports of goods and services, cw 2009 $
 85 EXN      = Exports of goods and services, current $
 86 FCBN     = US current account balance, current $
 87 FCBRN    = US current account balance residual, current $
 88 FGDP     = Foreign aggregate GDP (world, bilateral export weights)
 89 FGDPT    = Foreign aggregate GDP (world, bilateral export weights), trend
 90 FNICN    = Gross stock of claims of US residents on the rest of the world, current $
 91 FNILN    = Gross stock of liabilities of US residents to the rest of the world, current $
 92 FNIN     = Net stock of claims of US residents on the rest of the world, current $
 93 FNIRN    = Net stock of claims of US residents on the rest of the world, residual
 94 FPC      = Foreign aggregate consumer price (G39, import/export trade weights)
 95 FPCM     = Foreign aggregate consumer price (G39, bilateral non-oil import trade weights)
 96 FPI10    = Foreign consumer price inflation (G10)
 97 FPI10T   = Foreign consumer price inflation, trend (G10)
 98 FPIC     = Foreign consumer price inflation (G39, bilateral export trade weights)
 99 FPITRG   = Foreign target consumer price inflation (G10)
100 FPX      = Nominal exchange rate (G39, import/export trade weights)
101 FPXM     = Nominal exchange rate (G39, bilateral import trade weights)
102 FPXR     = Real exchange rate (G39, import/export trade weights)
103 FPXRR    = Real exchange rate residual
104 FPXRRT   = Real exchange rate residual, trend
105 FRL10    = Foreign long-term interest rate (G10)
106 FRS10    = Foreign short-term interest rate (G10)
107 FRSTAR   = Equilibrium real short-term interest rate used in foreign Taylor rule
108 FTCIN    = Corporate taxes paid to rest of world, current $
109 FXGAP    = Foreign output gap (world, bilateral export weights)
110 FYNICN   = Gross investment income received from the rest of the world, current $
111 FYNILN   = Gross investment income paid to the rest of the world, current $
112 FYNIN    = Net investment income received from the rest of the world, current $
113 GFDBTN   = Federal government debt stock, current $
```

```
114 GFDRT    = Federal government target debt-to-GDP ratio
115 GFINTN   = Federal government net interest payments, current $
116 GFS      = Federal government grants-in-aid to S&L government, deflated by PGDP
117 GFSN     = Federal government grants-in-aid to S&L government, current $
118 GFSRPN   = Federal government budget surplus, current $
119 GFSRT    = Federal government target surplus-to-GDP ratio
120 GFSUB    = Federal government subsidies less surplus, deflated by PGDP
121 GFSUBN   = Federal government subsidies less surplus, current $
122 GFT      = Federal government net transfer payments, deflated by PGDP
123 GFTN     = Federal government net transfer payments, current $
124 GFTRD    = Deviation of ratio of federal transfers to GDP from trend ratio
125 GFTRT    = Federal government, trend ratio of transfer payments to GDP
126 GSDBTN   = S&L government debt stock, current $
127 GSDRT    = S&L government target debt-to-GDP ratio
128 GSINTN   = S&L government net interest payments, current $
129 GSSRPN   = S&L government budget surplus, current $
130 GSSRT    = State and local government, target surplus-to-GDP ratio
131 GSSUB    = S&L government subsidies less surplus, deflated by PGDP
132 GSSUBN   = S&L government subsidies less surplus, current $
133 GST      = S&L government net transfer payments, deflated by PGDP
134 GSTN     = S&L government net transfer payments, current $
135 GSTRD    = Deviation of ratio of S&L transfers to GDP from trend ratio
136 GSTRT    = S&L government, trend ratio of transfer payments to GDP
137 HGEMP    = Petroleum imports, cw 2009$, trend growth rate
138 HGGDP    = Growth rate of GDP, cw 2009$ (annual rate)
139 HGGDPT   = Trend growth rate of XGDP, cw 2009$ (annual rate)
140 HGPCDR   = Trend growth rate of price of consumer durable goods (relative to PCNI
141 HGPDR    = Trend Price Growth of PPDR
142 HGPIR    = Trend Price Growth of PPIR
143 HGPKIR   = Trend growth rate of PKIR
144 HGPPSR   = Trend growth rate of PPSR
145 HGVPD    = Trend Growth of VPD
146 HGVPI    = Trend growth rate of VPI
147 HGVPS    = Trend growth rate of VPS
148 HGX      = Trend growth rate of XG, cw 2009$ (annual rate)
149 HGYNID   = Growth rate of real after-tax corporate profits
150 HKS      = Growth rate of KS, cw 2009$ (compound annual rate)
151 HKSR     = Residual growth of capital services
152 HLEPT    = Trend growth rate of LEP (annual rate)
153 HLPRDT   = Trend growth rate of output per hour
154 HMFPT    = Trend growth rate of multifactor productivity
155 HQLFPR   = Drift component of change in QLFPR
156 HQLWW    = Trend growth rate of workweek
157 HUQPCT   = Drift term in stochastic component of trend ratio of PCNIA to PXP
158 HUXB     = Drift term in UXBT
159 HXBT     = Trend rate of growth of XB  , cw 2009$ (annual rate)
```

```
160 JCCACN   = Consumption of fixed capital, corporate, current $
161 JCCAN    = Consumption of fixed capital, current $
162 JKCD     = Consumption of fixed capital, consumer durables
163 JRCD     = Depreciation rate, consumer durables
164 JRH      = Depreciation rate, housing
165 JRPD     = Depreciation rate, equipment
166 JRPI     = Depreciation rate, intellectual property
167 JRPS     = Depreciation rate, nonresidential structures
168 JYGFEN   = CFC, federal government enterprises, current $
169 JYGFGN   = CFC, federal government, general, current $
170 JYGSEN   = CFC, state and local government enterprises, current $
171 JYGSGN   = CFC, state and local government, general, current $
172 JYNCN    = Noncorporate business CFC, current $
173 KCD      = Stock of consumer durables, cw 2009$
174 KH       = Stock of residential structures, cw 2009$
175 KI       = Stock of private inventories, cw 2009$
176 KPD      = Capital stock - Equipment, 2009$
177 KPI      = Capital Stock - Intellectual Property, 2009$
178 KPS      = Capital stock - nonresidential structures, 2009$
179 KS       = Capital services, 2009 $
180 LEF      = Federal civilian employment ex. gov. enterprise
181 LEFT     = Federal civilian employment ex. gov. enterprise, trend
182 LEH      = Civilian employment (break adjusted)
183 LEO      = Difference between household and business sector payroll employment, less gov't
184 LEP      = Employment in  business sector (employee and  self-employed)
185 LEPPOT   = Potential employment in  business sector
186 LES      = S&L government employment ex. gov. enterprise
187 LEST     = S&L government employment ex. gov. enterprise, trend
188 LEUC     = Emergency unemployment compensation (EUC)
189 LF       = Civilian labor force (break adjusted)
190 LFPR     = Labor force participation rate
191 LHP      = Aggregate labor hours,  business sector (employee and  self-employed)
192 LPRDT    = Trend labor productivity
193 LQUALT   = Labor quality, trend level
194 LUR      = Civilian unemployment rate (break adjusted)
195 LURBLS   = Civilian unemployment rate (published)
196 LURNAT   = Natural rate of unemployment
197 LURTRSH  = Unemployment threshold
198 LWW      = Workweek,  business sector (employee and self-employed)
199 MEI      = Multiplicative discrepancy for the difference between XGDI and XGDO
200 MEP      = Multiplicative discrepancy for the difference between XGDP and XGDO
201 MFPT     = Multifactor productivity, trend level
202 N16      = Noninstitutional population, aged 16 and over (break adjusted)
203 PCDR     = Price index for consumer durables, cw (relative to to PCNIA)
204 PCENG    = Price index for aggregate energy consumption
205 PCENGR   = Price index for aggregate energy consumption (relative to PXB  )
```

```
206 PCER     = Price index for personal consumption expenditures on energy (relative
207 PCFR     = Price index for personal consumption expenditures on food (relative to
208 PCFRT    = Real PCE price of food, trend
209 PCHR     = Price index for housing services, cw (relative to to PCNIA)
210 PCNIA    = Price index for personal consumption expenditures, cw (NIPA definitior
211 PCOR     = Price index for non-durable goods and non-housing services, cw (relati
212 PCPI     = Consumer price index,total
213 PCPIX    = Consumer price index,excluding food and energy
214 PCSTAR   = Target consumption price level (used in RFFGEN policy rule)
215 PCXFE    = Price index for personal consumption expendits ex. food and energy, cw
216 PGDP     = Price index for GDP, cw
217 PGFIR    = Price index for federal gov. investment, cw (relative to PXP)
218 PGFL     = Price index for federal government employee compensation, cw
219 PGFOR    = Price index for federal governemnt consumption ex. emp. comp., cw (rel
220 PGSIR    = Price index for S&L government investment (relative to PXP)
221 PGSL     = Price index for S&L government employee compensation, cw
222 PGSOR    = Price index for S&L government consumption ex. emp. comp., cw (relativ
223 PHOUSE   = Loan Performance House Price Index
224 PHR      = Price index for residential investment, cw (relative to PXP)
225 PIC4     = Four-quarter percent change in PCE prices
226 PICNGR   = Weighted growth rate of relative energy price
227 PICNIA   = Inflation rate, personal consumption expenditures, cw
228 PICX4    = Four-quarter percent change core in PCE prices
229 PICXFE   = Inflation rate, personal consumption expenditures, ex. food and energy
230 PIECI    = Annualized rate of growth of EI hourly compensation
231 PIGDP    = Inflation rate, GDP, cw
232 PIPL     = Rate of growth of PL
233 PIPXNC   = Inflation rate, price of adjusted final sales excluding consumption (a
234 PITARG   = Target rate of consumption price inflation (used in policy reaction fu
235 PITRSH   = Inflation threshold
236 PKIR     = Price index for stock of inventories, cw (relative to PXP)
237 PKPDR    = Ratio of price of equipment stock (KPD) to PXP
238 PL       = Compensation per hour,  business
239 PLMIN    = Minimum wage
240 PLMINR   = Ratio of hourly minimum wage to compensation per hour (times 100)
241 PMO      = Price index for imports ex. petroleum, cw
242 PMP      = Price index for petroleum imports
243 POIL     = Price of imported oil ($ per barrel)
244 POILR    = Price of imported oil, relative to price index for bus. sector output
245 POILRT   = Price of imported oil, relative to price index for bus. sector output,
246 PPDR     = Price level of EPD compared to PXP
247 PPIR     = Price level of EPI compared to PXP
248 PPSR     = Price index for nonresidential structures, cw (relative to PXP)
249 PTR      = 10-year expected PCE price inflation (Survey of Professional Forecaste
250 PWSTAR   = Equilibrium NFB price markup
251 PXB      = Price index for NFB output
```

```
252 PXG      = Price index for  business output plus oil imports
253 PXNC     = Price of adjusted final sales excluding consumption
254 PXP      = Price index for final sales plus imports less gov. labor
255 PXR      = Price index for exports, cw (relative to PXP)
256 QEC      = Desired level of consumption (FRBUS definition)
257 QECD     = Target level of consumption of durable goods, trending component
258 QECO     = Desired level of consumption of nondurable goods and nonhousing services
259 QEH      = Target level of residential investment
260 QEPD     = Desired level of investment in equipment
261 QEPI     = Desired level of investment in intellectual property
262 QEPS     = Desired level of investment in structures
263 QKIR     = Desired Inventory Sales Ratio
264 QLEOR    = Desired ratio of employment discrepancy to the labor force
265 QLEP     = Desired level of  business employment
266 QLF      = Desired level of civilian labor force
267 QLFPR    = Trend labor force participation rate
268 QLHP     = Desired level of  business labor hours
269 QLWW     = Trend workweek,  business sector (employee and  self-employed)
270 QPCNIA   = Desired level of consumption price
271 QPL      = Desired level of compensation per hour, trending component
272 QPMO     = Random walk component of non-oil import prices
273 QPXG     = Desired price level of private output ex. energy, housing, and farm
274 QPXNC    = Desired level of nonconsumption price
275 QPXP     = Desired price level of adjusted final sales
276 QYNIDN   = Desired level of dividends
277 RBBB     = S&P BBB corporate bond rate
278 RBBBE    = S&P BBB corporate bond rate (effective ann. yield)
279 RBBBP    = S&P BBB corporate bond rate, risk/term premium
280 RCAR     = New car loan rate at finance companies
281 RCCD     = Cost of capital for consumer durables
282 RCCH     = Cost of capital for residential investment
283 RCGAIN   = Rate of capital gain on the non-equity portion of household wealth
284 REQ      = Real expected rate of return on equity
285 REQP     = Real expected rate of return on equity, premium component
286 RFF      = Federal funds rate
287 RFFALT   = Value of eff. federal funds rate given by estimated policy rule
288 RFFE     = Federal funds rate (effective ann. yield)
289 RFFFIX   = Federal funds rate given by fixed, pre-determined funds rate path
290 RFFGEN   = Value of eff. federal funds rate given by the generalized reaction function
291 RFFINTAY = Value of eff. federal funds rate given by the inertial Taylor rule
292 RFFMIN   = Minimum nominal funds rate (set at 0 to impose zero lower bound)
293 RFFRULE  = Federal funds rate (effective ann. yield)
294 RFFTAY   = Value of eff. federal funds rate given by the Taylor rule with output gap
295 RFFTLR   = Value of eff. federal funds rate given by the Taylor rule with unemployment gap
296 RFNICT   = Residual in FNICN equation
297 RFRS10   = Real foreign short-term interest rate
```

```
298 RFYNIC   = Average yield earned on gross claims of US residents on the rest of th
299 RFYNIL   = Average yield earned on liabilities of US residents on the rest of the
300 RG10     = 10-year Treasury bond rate
301 RG10E    = 10-year Treasury bond rate (effective ann. yield)
302 RG10P    = 10-year Treasury bond rate, term premium
303 RG30     = 30-year Treasury bond rate
304 RG30E    = 30-year Treasury bond rate (effective ann. yield)
305 RG30P    = 30-year Treasury bond rate, term premium
306 RG5      = 5-year Treasury note rate
307 RG5E     = 5-year Treasury note rate (effective ann. yield)
308 RG5P     = 5-year Treasury note rate. term premium
309 RGFINT   = Average rate of interest on existing federal debt
310 RGW      = Approximate average rate of interest on new federal debt
311 RME      = Interest rate on conventional mortgages (effective ann. yield)
312 RPD      = After-tax real financial cost of capital for business investment
313 RRFFE    = Real federal funds rate (effective ann. yield)
314 RRFIX    = Real federal funds rate given by fixed, pre-determined real funds rate
315 RRMET    = Real mortgage rate, trend
316 RRTR     = Expected long-run real federal funds rate
317 RSPNIA   = Personal saving rate
318 RSTAR    = Equilibrium real federal funds rate (for monetary policy reaction func
319 RTB      = 3-month Treasury bill rate
320 RTBE     = 3-month Treasury bill rate (effective ann. yield)
321 RTINV    = User cost of capital for inventories
322 RTPD     = User cost of capital for equipment
323 RTPI     = User cost of capital for intellectual property
324 RTPS     = User cost of capital for nonresidential structures
325 RTR      = Expected federal funds rate in the long run (Blue Chip)
326 T47      = Time trend, begins in 1947q1 (0 before)
327 TAPDAD   = Proportion of investment in equipment using accelerated depreciation
328 TAPDD    = Present value of depreciation allowances for equipment
329 TAPDDP   = Proportion of investment tax credit deducted from depr. base
330 TAPDS    = Tax service life of equipment
331 TAPDT    = Investment tax credit rate for equipment
332 TAPSAD   = Proportion of investment in nonresidential structures using accelerate
333 TAPSDA   = Present value of depreciation allowances for nonresidential structures
334 TAPSSL   = Tax service life of nonresidential structures
335 TFCIN    = Federal corporate income tax accruals, current $
336 TFDIV    = Federal income receipts on assets, dividends, current $
337 TFIBN    = Federal indirect business tax receipts, current $
338 TFPN     = Federal personal income tax and nontax receipts, current $
339 TFSIN    = Federal social insurance tax receipts
340 TRFCI    = Average federal corporate income tax rate
341 TRFCIM   = Marginal federal corporate income tax rate
342 TRFIB    = Average federal indirect business tax rate
343 TRFP     = Average federal tax rate for personal income tax and nontax receipts
```

```
344 TRFPM    = Marginal federal personal income tax rate (at twice median family income)
345 TRFPT    = Average federal tax rate for personal income tax, trend
346 TRFPTX   = Average federal tax rate for personal income tax, trend, policy setting
347 TRFSI    = Average federal social insurance tax rate
348 TRSCI    = Average S&L corporate income tax rate
349 TRSCIT   = Average S&L corporate income tax rate, trend
350 TRSIB    = Average S&L indirect business tax rate
351 TRSIBT   = Average S&L indirect business tax rate, trend
352 TRSP     = Average S&L tax rate for personal income tax and nontax receipts
353 TRSPP    = Marginal S&L tax rate on personal property
354 TRSPT    = Trend S&L personal income tax rate
355 TRSPTX   = Average state and local tax rate for personal income, trend
356 TRSSI    = Average S&L social insurance tax rate
357 TRSSIT   = Average S&L social insurance tax rate, trend
358 TRYH     = Average tax rate on household income
359 TSCIN    = S&L corporate income tax accruals, current $
360 TSIBN    = S&L indirect business tax receipts, current $
361 TSPN     = S&L personal income tax and nontax receipts, current $
362 TSSIN    = S&L social insurance tax receipts, current $
363 UCES     = Energy share of nominal consumption expenditures
364 UCFS     = Food share of nominal consumption expenditures
365 UEMOT    = Trend in ratio of EMON to XGDEN
366 UEMP     = Multiplicative factor in EMP identity
367 UFCBR    = Multiplicative factor in FCBRN identity
368 UFNIR    = Multiplicative factor in FNIRN identity
369 UFPCM    = Multiplicative factor in FPCM identity
370 UFPXM    = Multiplicative factor in FPXM identity
371 UFTCIN   = Multiplicative factor in FTCIN identity
372 UGFDBT   = Multiplicative factor in GFDBTN identity
373 UGSDBT   = Multiplicative factor in GSDBTN identity
374 UGSINT   = Multiplicative factor in GSINTN identity
375 UGSSUB   = Multiplicative factor in GSSUB identity
376 UJCCA    = Multiplicative factor in JCCAN identity
377 UJCCAC   = Multiplicative factor in JCCACN identity
378 UJYGFE   = Multiplicative factor in JYGFEN identity
379 UJYGFG   = Multiplicative factor in JYGFGN identity
380 UJYGSE   = Multiplicative factor in JYGSEN identity
381 UJYGSG   = Multiplicative factor in JYGSGN identity
382 ULEF     = Multiplicative factor in LEF identity
383 ULES     = Multiplicative factor in LES identity
384 UPCPI    = Multiplicative factor in PCPI identity
385 UPCPIX   = Multiplicative factor in PCPIX identity
386 UPGFL    = Multiplicative factor in PGFL identity
387 UPGSL    = Multiplicative factor in PGSL identity
388 UPKPD    = Multiplicative factor in PKPDR identity
389 UPMP     = Multiplicative factor in PMP identity
```

```
390 UPXB     = Multiplicative factor in PXB   identity
391 UQPCT    = Stochastic component of trend ratio of PCNIA to PXP
392 UVEOA    = Multiplicative factor in VEOA identity
393 UVPD     = Multiplicative factor in VPD identity
394 UVPI     = Multiplicative factor in VPI identity
395 UVPS     = Multiplicative factor in VPS identity
396 UXBT     = Stochastic component of trend ratio of XGDPT to XBT
397 UXENG    = Multiplicative factor in XENG identity
398 UYD      = Multiplicative factor in YDN identity
399 UYHI     = Multiplicative factor in YHIN identity
400 UYHLN    = Multiplicative factor in YHLN identity
401 UYHPTN   = Multiplicative factor in YHPTN identity
402 UYHSN    = Multiplicative factor in personal saving identity (accounts for transf
403 UYHTN    = Multiplicative factor in YHTN identity
404 UYL      = Multiplicative factor in YLN identity
405 UYNI     = Multiplicative factor in YNIN identity
406 UYNICP   = Multiplicative factor in YNICPN identity
407 UYP      = Multiplicative factor in YPN identity
408 UYSEN    = Multiplicative factor in YSEN identity
409 VEO      = Desired energy-output ratio
410 VEOA     = Average energy-output ratio of existing capital stock
411 VPD      = Desired equipment-output ratio
412 VPI      = Desired intellectual property-output ratio
413 VPS      = Desired structures-output ratio
414 WDNFCN   = Net financial liabilities, nonfinancial nonfarm corporations
415 WPO      = Household property wealth ex. stock market, real
416 WPON     = Household property wealth ex. stock market, current $
417 WPS      = Household stock market wealth, real
418 WPSN     = Household stock market wealth, current $
419 XB       = Business output (BEA definition), cw 2009$
420 XBN      = Business output (BEA definition), current $
421 XBO      = Business output, adjusted for measurement error, cw  2009$
422 XBT      = Potential business output, cw 2009$
423 XENG     = Crude energy production, cw 2009$
424 XFS      = Final sales of gross domestic product, cw 2009$
425 XFSN     = Final sales of gross domestic product, current $
426 XG       = Output of  business sector plus oil imports, cw 2009$
427 XGAP     = Output gap for  business plus oil imports  (100*log(actual/potential)
428 XGAP2    = Output gap for GDP (100*log(actual/potential)
429 XGDE     = Domestic absorption, cw 2009$
430 XGDEN    = Nominal Absorption, current $
431 XGDI     = Gross domestic income, cw 2009$
432 XGDIN    = Gross domestic income, current $
433 XGDO     = Gross domestic product, adjusted for measurement error, cw 2009$
434 XGDP     = GDP, cw 2009$
435 XGDPN    = GDP, current $
```

```
436 XGDPT    = Potential GDP, cw 2009$
437 XGDPTN   = Potential GDP, current $
438 XGN      = Output of  business sector plus oil imports, current $
439 XGO      = Output of business sector plus oil imports, adjusted for measurement error, cw 2
440 XGPOT    = Potential output of  business sector plus oil imports,  cw 2009$
441 XP       = Final sales plus imports less government labor, cw 2009$
442 XPN      = Final sales plus imports less government labor, current $
443 YCSN     = Net corporate cash flow with IVA and CCA
444 YDN      = Disposable income
445 YGFSN    = Federal government saving
446 YGSSN    = State and Local government saving
447 YH       = Income, household, total (real after-tax)
448 YHGAP    = Income, household, total, ratio to XGDP, cyclical component (real after-tax)
449 YHIBN    = Consumer interest payments to business
450 YHIN     = Income, household, net interest and rent
451 YHL      = Income, household, labor compensation (real after-tax)
452 YHLN     = Income, household, labor compensation
453 YHP      = Income, household, property (real after-tax)
454 YHPCD    = Imputed income of the stock of consumer durables, 2009$
455 YHPGAP   = Income, household, property, ratio to YH, cyclical component (real after-tax)
456 YHPNTN   = Income, household, property, non-taxable component
457 YHPSHR   = Income, household, property, ratio to YH (real after-tax)
458 YHPTN    = Income, household, property, taxable component
459 YHSHR    = Income, household, total, ratio to XGDP (real after-tax)
460 YHSN     = Personal saving
461 YHT      = Income, household, transfer (real after-tax), net basis
462 YHTGAP   = Income, household, transfer, ratio to YH, cyclical component (real after-tax)
463 YHTN     = Income, household, transfer payments. net basis
464 YHTSHR   = Income, household, transfer, ratio to YH (real after-tax)
465 YKIN     = Income from stock of inventories
466 YKPDN    = Income from stock of equipment
467 YKPSN    = Income from stock of nonresidential structures
468 YMSDN    = Microsoft one-time dividend payout in 2004Q4
469 YNICPN   = Corporate profits (national income component)
470 YNIDN    = Dividends (national income component)
471 YNIIN    = Net interest and rental income (national income component)
472 YNILN    = Labor income (national income component)
473 YNIN     = National income
474 YNISEN   = Propprietors' income (national income component)
475 YPN      = Personal income
476 ZDIVGR   = Expected growth rate of real dividends, for WPSN eq. (VAR exp.)
477 ZECD     = Expected growth rate of target durable consumption, for ECD eq. (VAR exp.)
478 ZECO     = Expected growth rate of target nondurables and nonhousing services, for ECO eq (
479 ZEH      = Expected growth rate of target residential investment, for EH eq. (VAR exp.)
480 ZGAP05   = Expected output gap, for RG5E eq. (VAR exp.)
481 ZGAP10   = Expected output gap, for RG10E eq. (VAR exp.)
```

```
482 ZGAP30   = Expected output gap, for RG30E eq. (VAR exp.)
483 ZGAPC2   = Expected output gap, for ECD eq. (VAR exp.)
484 ZLHP     = Expected growth rate of target aggregate hours (VAR exp.)
485 ZPI10    = Expected cons. price infl., for RCCH, RRMET, and YHPNTN eqs. (10-yr ma
486 ZPI10F   = Expected cons. price infl., for FPXR eq. (10-yr mat.) (VAR exp.)
487 ZPI5     = Expected cons. price infl., for RCCD eq. (5-yr mat.) (VAR exp.)
488 ZPIB5    = Expected output price infl., for RPD eq. (5-yr mat.) (VAR exp.)
489 ZPIC30   = Expected cons. price infl., for REQ eq. (30-yr mat.) (VAR exp.)
490 ZPIC58   = Expected 4-qtr consumer price inflation (8 qtrs. in the future) (VAR e
491 ZPICXFE  = Expected value of picxfe in the next quarter (VAR exp.)
492 ZPIECI   = Expected value of pieci in the next quarter (VAR exp.)
493 ZRFF10   = Expected federal funds rate, for RG10E eq. (10-yr mat.) (VAR exp.)
494 ZRFF30   = Expected federal funds rate, for RG30E eq. (30-yr mat.) (VAR exp.)
495 ZRFF5    = Expected federal funds rate, for RG5E eq. (5-yr mat.) (VAR exp.)
496 ZVPD     = Expected growth rate of capital-output ratio, for EPD (VAR exp.)
497 ZVPI     = Expected growth rate of capital-output ratio, for EPI (VAR exp.)
498 ZVPS     = Expected growth rate of des. capital-output ratio, for EPS eq. (VAR ex
499 ZXBD     = Expected growth rate of buisiness output for EPD (VAR exp.)
500 ZXBI     = Expected growth rate of business output, for EPI (VAR exp.)
501 ZXBS     = Expected growth rate of business output, for EPS (VAR exp.)
502 ZYH      = Expected level of real after-tax household income, for QEC eq. (VAR ex
503 ZYHP     = Expected level of real after-tax property income, for QEC eq. (VAR exp
504 ZYHPST   = Expected trend share of property income in household income
505 ZYHST    = Expected trend ratio of household income to GDP
506 ZYHT     = Expected level of real transfer income, for QEC eq. (VAR exp.)
507 ZYHTST   = Expected trend share of transfer income in household income
508 ZYNID    = Expected rate of growth of target real dividends, for YNIDN eq. (VAR e
509 ZZZBLANK = empty slot
510 ZZZBLANK = empty slot
511 ZZZBLANK = empty slot
512 ZZZBLANK = empty slot
513 ZZZBLANK = empty slot
514 ZZZBLANK = empty slot
515 ZZZBLANK = empty slot
516 ZZZBLANK = empty slot
517 ZZZBLANK = empty slot
518 ZZZBLANK = empty slot
519 ZZZBLANK = empty slot
520 ZZZBLANK = empty slot
521 ZZZBLANK = empty slot
522 ZZZBLANK = empty slot
523 ZZZBLANK = empty slot
524 ZZZBLANK = empty slot
525 ZZZBLANK = empty slot
526 ZZZBLANK = empty slot
527 ZZZBLANK = empty slot
```

```
528 ZZZBLANK = empty slot
529 ZZZBLANK = empty slot
530 ZZZBLANK = empty slot
531 ZZZBLANK = empty slot
532 ZZZBLANK = empty slot
533 ZZZBLANK = empty slot
534 ZZZBLANK = empty slot
535 ZZZBLANK = empty slot
536 ZZZBLANK = empty slot
537 ZZZBLANK = empty slot
538 ZZZBLANK = empty slot
539 ZZZBLANK = empty slot
540 ZZZBLANK = empty slot
541 ZZZBLANK = empty slot
542 ZZZBLANK = empty slot
543 ZZZBLANK = empty slot
544 ZZZBLANK = empty slot
545 ZZZBLANK = empty slot
546 ZZZBLANK = empty slot
547 ZZZBLANK = empty slot
548 ZZZBLANK = empty slot
549 ZZZBLANK = empty slot
550 ZZZBLANK = empty slot
551 ZZZBLANK = empty slot
552 ZZZBLANK = empty slot
553 ZZZBLANK = empty slot
554 ZZZBLANK = empty slot
555 ZZZBLANK = empty slot
556 ZZZBLANK = empty slot
557 ZZZBLANK = empty slot
558 ZZZBLANK = empty slot
559 ZZZBLANK = empty slot
560 ZZZBLANK = empty slot
561 ZZZBLANK = empty slot
562 ZZZBLANK = empty slot
563 ZZZBLANK = empty slot
564 ZZZBLANK = empty slot
565 ZZZBLANK = empty slot
566 ZZZBLANK = empty slot
567 ZZZBLANK = empty slot
568 ZZZBLANK = empty slot
569 ZZZBLANK = empty slot
570 ZZZBLANK = empty slot
571 ZZZBLANK = empty slot
572 ZZZBLANK = empty slot
573 ZZZBLANK = empty slot
```

```
574 ZZZBLANK = empty slot
575 ZZZBLANK = empty slot
576 ZZZBLANK = empty slot
577 ZZZBLANK = empty slot
578 ZZZBLANK = empty slot
579 ZZZBLANK = empty slot
580 ZZZBLANK = empty slot
581 ZZZBLANK = empty slot
582 ZZZBLANK = empty slot
583 ZZZBLANK = empty slot
584 ZZZBLANK = empty slot
585 ZZZBLANK = empty slot
586 ZZZBLANK = empty slot
587 ZZZBLANK = empty slot
588 ZZZBLANK = empty slot
589 ZZZBLANK = empty slot
590 ZZZBLANK = empty slot
591 ZZZBLANK = empty slot
592 ZZZBLANK = empty slot
593 ZZZBLANK = empty slot
594 ZZZBLANK = empty slot
595 ZZZBLANK = empty slot
596 ZZZBLANK = empty slot
597 ZZZBLANK = empty slot
598 ZZZBLANK = empty slot
599 ZZZBLANK = empty slot
600 ZZZBLANK = empty slot
601 ZZZBLANK = empty slot
602 ZZZBLANK = empty slot
603 ZZZBLANK = empty slot
604 ZZZBLANK = empty slot
605 ZZZBLANK = empty slot
606 ZZZBLANK = empty slot
607 ZZZBLANK = empty slot
608 ZZZBLANK = empty slot
609 ZZZBLANK = empty slot
610 ZZZBLANK = empty slot
611 ZZZBLANK = empty slot
612 ZZZBLANK = empty slot
613 ZZZBLANK = empty slot
614 ZZZBLANK = empty slot
615 ZZZBLANK = empty slot
616 ZZZBLANK = empty slot
617 ZZZBLANK = empty slot
618 ZZZBLANK = empty slot
619 ZZZBLANK = empty slot
```

```
620 ZZZBLANK = empty slot
621 ZZZBLANK = empty slot
622 ZZZBLANK = empty slot
623 ZZZBLANK = empty slot
624 ZZZBLANK = empty slot
625 ZZZBLANK = empty slot
626 ZZZBLANK = empty slot
627 ZZZBLANK = empty slot
628 ZZZBLANK = empty slot
629 ZZZBLANK = empty slot
630 ZZZBLANK = empty slot
631 ZZZBLANK = empty slot
632 ZZZBLANK = empty slot
633 ZZZBLANK = empty slot
634 ZZZBLANK = empty slot
635 ZZZBLANK = empty slot
636 ZZZBLANK = empty slot
637 ZZZBLANK = empty slot
638 ZZZBLANK = empty slot
639 ZZZBLANK = empty slot
640 ZZZBLANK = empty slot
641 ZZZBLANK = empty slot
642 ZZZBLANK = empty slot
643 ZZZBLANK = empty slot
644 ZZZBLANK = empty slot
645 ZZZBLANK = empty slot
646 ZZZBLANK = empty slot
647 ZZZBLANK = empty slot
648 ZZZBLANK = empty slot
649 ZZZBLANK = empty slot
650 ZZZBLANK = empty slot
651 ZZZBLANK = empty slot
652 ZZZBLANK = empty slot
653 ZZZBLANK = empty slot
654 ZZZBLANK = empty slot
655 ZZZBLANK = empty slot
656 ZZZBLANK = empty slot
657 ZZZBLANK = empty slot
658 ZZZBLANK = empty slot
659 ZZZBLANK = empty slot
660 ZZZBLANK = empty slot
661 ZZZBLANK = empty slot
662 ZZZBLANK = empty slot
663 ZZZBLANK = empty slot
664 ZZZBLANK = empty slot
665 ZZZBLANK = empty slot
```

```
666 ZZZBLANK = empty slot
667 ZZZBLANK = empty slot
668 ZZZBLANK = empty slot
669 ZZZBLANK = empty slot
670 ZZZBLANK = empty slot
671 ZZZBLANK = empty slot
672 ZZZBLANK = empty slot
673 ZZZBLANK = empty slot
674 ZZZBLANK = empty slot
675 ZZZBLANK = empty slot
676 ZZZBLANK = empty slot
677 ZZZBLANK = empty slot
678 ZZZBLANK = empty slot
679 ZZZBLANK = empty slot
680 ZZZBLANK = empty slot
681 ZZZBLANK = empty slot
682 ZZZBLANK = empty slot
683 ZZZBLANK = empty slot
684 ZZZBLANK = empty slot
685 ZZZBLANK = empty slot
686 ZZZBLANK = empty slot
687 ZZZBLANK = empty slot
688 ZZZBLANK = empty slot
689 ZZZBLANK = empty slot
690 ZZZBLANK = empty slot
691 ZZZBLANK = empty slot
692 ZZZBLANK = empty slot
693 ZZZBLANK = empty slot
694 ZZZBLANK = empty slot
695 ZZZBLANK = empty slot
696 ZZZBLANK = empty slot
697 ZZZBLANK = empty slot
698 ZZZBLANK = empty slot
699 ZZZBLANK = empty slot
700 ZZZBLANK = empty slot
701 ZZZBLANK = empty slot
702 ZZZBLANK = empty slot
703 ZZZBLANK = empty slot
704 ZZZBLANK = empty slot
705 ZZZBLANK = empty slot
706 ZZZBLANK = empty slot
707 ZZZBLANK = empty slot
708 ZZZBLANK = empty slot
709 ZZZBLANK = empty slot
710 ZZZBLANK = empty slot
711 ZZZBLANK = empty slot
```

```
712 ZZZBLANK = empty slot
713 ZZZBLANK = empty slot
714 ZZZBLANK = empty slot
715 ZZZBLANK = empty slot
716 ZZZBLANK = empty slot
717 ZZZBLANK = empty slot
718 ZZZBLANK = empty slot
719 ZZZBLANK = empty slot
720 ZZZBLANK = empty slot
721 ZZZBLANK = empty slot
722 ZZZBLANK = empty slot
723 ZZZBLANK = empty slot
724 ZZZBLANK = empty slot
725 ZZZBLANK = empty slot
726 ZZZBLANK = empty slot
727 ZZZBLANK = empty slot
728 ZZZBLANK = empty slot
729 ZZZBLANK = empty slot
730 ZZZBLANK = empty slot
731 ZZZBLANK = empty slot
732 ZZZBLANK = empty slot
733 ZZZBLANK = empty slot
734 ZZZBLANK = empty slot
735 ZZZBLANK = empty slot
736 ZZZBLANK = empty slot
737 ZZZBLANK = empty slot
738 ZZZBLANK = empty slot
739 ZZZBLANK = empty slot
740 ZZZBLANK = empty slot
741 ZZZBLANK = empty slot
742 ZZZBLANK = empty slot
743 ZZZBLANK = empty slot
744 ZZZBLANK = empty slot
745 ZZZBLANK = empty slot
746 ZZZBLANK = empty slot
747 ZZZBLANK = empty slot
748 ZZZBLANK = empty slot
749 ZZZBLANK = empty slot
750 ZZZBLANK = empty slot
751 ZZZBLANK = empty slot
752 ZZZBLANK = empty slot
753 ZZZBLANK = empty slot
754 ZZZBLANK = empty slot
755 ZZZBLANK = empty slot
756 ZZZBLANK = empty slot
757 ZZZBLANK = empty slot
```

```
758 ZZZBLANK = empty slot
759 ZZZBLANK = empty slot
760 ZZZBLANK = empty slot
761 ZZZBLANK = empty slot
762 ZZZBLANK = empty slot
763 ZZZBLANK = empty slot
764 ZZZBLANK = empty slot
765 ZZZBLANK = empty slot
766 ZZZBLANK = empty slot
767 ZZZBLANK = empty slot
768 ZZZBLANK = empty slot
769 ZZZBLANK = empty slot
770 ZZZBLANK = empty slot
771 ZZZBLANK = empty slot
772 ZZZBLANK = empty slot
773 ZZZBLANK = empty slot
774 ZZZBLANK = empty slot
775 ZZZBLANK = empty slot
776 ZZZBLANK = empty slot
777 ZZZBLANK = empty slot
778 ZZZBLANK = empty slot
779 ZZZBLANK = empty slot
780 ZZZBLANK = empty slot
781 ZZZBLANK = empty slot
782 ZZZBLANK = empty slot
783 ZZZBLANK = empty slot
784 ZZZBLANK = empty slot
785 ZZZBLANK = empty slot
786 ZZZBLANK = empty slot
787 ZZZBLANK = empty slot
788 ZZZBLANK = empty slot
789 ZZZBLANK = empty slot
790 ZZZBLANK = empty slot
791 ZZZBLANK = empty slot
792 ZZZBLANK = empty slot
793 ZZZBLANK = empty slot
794 ZZZBLANK = empty slot
795 ZZZBLANK = empty slot
796 ZZZBLANK = empty slot
797 ZZZBLANK = empty slot
798 ZZZBLANK = empty slot
799 ZZZBLANK = empty slot
800 ZZZBLANK = empty slot
801 ZZZBLANK = empty slot
802 ZZZBLANK = empty slot
803 ZZZBLANK = empty slot
```

```
804 ZZZBLANK = empty slot
805 ZZZBLANK = empty slot
806 ZZZBLANK = empty slot
807 ZZZBLANK = empty slot
808 ZZZBLANK = empty slot
809 ZZZBLANK = empty slot
810 ZZZBLANK = empty slot
811 ZZZBLANK = empty slot
812 ZZZBLANK = empty slot
813 ZZZBLANK = empty slot
814 ZZZBLANK = empty slot
815 ZZZBLANK = empty slot
816 ZZZBLANK = empty slot
817 ZZZBLANK = empty slot
818 ZZZBLANK = empty slot
819 ZZZBLANK = empty slot
820 ZZZBLANK = empty slot
821 ZZZBLANK = empty slot
822 ZZZBLANK = empty slot
823 ZZZBLANK = empty slot
824 ZZZBLANK = empty slot
825 ZZZBLANK = empty slot
826 ZZZBLANK = empty slot
827 ZZZBLANK = empty slot
828 ZZZBLANK = empty slot
829 ZZZBLANK = empty slot
830 ZZZBLANK = empty slot
831 ZZZBLANK = empty slot
832 ZZZBLANK = empty slot
833 ZZZBLANK = empty slot
834 ZZZBLANK = empty slot
835 ZZZBLANK = empty slot
836 ZZZBLANK = empty slot
837 ZZZBLANK = empty slot
838 ZZZBLANK = empty slot
839 ZZZBLANK = empty slot
840 ZZZBLANK = empty slot
841 ZZZBLANK = empty slot
842 ZZZBLANK = empty slot
843 ZZZBLANK = empty slot
844 ZZZBLANK = empty slot
845 ZZZBLANK = empty slot
846 ZZZBLANK = empty slot
847 ZZZBLANK = empty slot
848 ZZZBLANK = empty slot
849 ZZZBLANK = empty slot
```

```
850 ZZZBLANK = empty slot
851 ZZZBLANK = empty slot
852 ZZZBLANK = empty slot
853 ZZZBLANK = empty slot
854 ZZZBLANK = empty slot
855 ZZZBLANK = empty slot
856 ZZZBLANK = empty slot
857 ZZZBLANK = empty slot
858 ZZZBLANK = empty slot
859 ZZZBLANK = empty slot
860 ZZZBLANK = empty slot
861 ZZZBLANK = empty slot
862 ZZZBLANK = empty slot
863 ZZZBLANK = empty slot
864 ZZZBLANK = empty slot
865 ZZZBLANK = empty slot
866 ZZZBLANK = empty slot
867 ZZZBLANK = empty slot
868 ZZZBLANK = empty slot
869 ZZZBLANK = empty slot
870 ZZZBLANK = empty slot
871 ZZZBLANK = empty slot
872 ZZZBLANK = empty slot
873 ZZZBLANK = empty slot
874 ZZZBLANK = empty slot
875 ZZZBLANK = empty slot
876 ZZZBLANK = empty slot
877 ZZZBLANK = empty slot
878 ZZZBLANK = empty slot
879 ZZZBLANK = empty slot
880 ZZZBLANK = empty slot
881 ZZZBLANK = empty slot
882 ZZZBLANK = empty slot
883 ZZZBLANK = empty slot
884 ZZZBLANK = empty slot
885 ZZZBLANK = empty slot
886 ZZZBLANK = empty slot
887 ZZZBLANK = empty slot
888 ZZZBLANK = empty slot
889 ZZZBLANK = empty slot
890 ZZZBLANK = empty slot
891 ZZZBLANK = empty slot
892 ZZZBLANK = empty slot
893 ZZZBLANK = empty slot
894 ZZZBLANK = empty slot
895 ZZZBLANK = empty slot
```

```
896 ZZZBLANK = empty slot
897 ZZZBLANK = empty slot
898 ZZZBLANK = empty slot
899 ZZZBLANK = empty slot
900 ZZZBLANK = empty slot
```

This code is written to file `stdver.varinfo`.

A.2 Standard Version Equations File

30 ⟨*stdver.eqs.txt* 30⟩≡

```
ceng: d( log(ceng), 0, 1 ) - ceng_aerr = _
                      y_ceng(1) * (log(ceng(-1))-log(xg(-1)*veoa(-1))) _
                    + y_ceng(2) * d( log(xg), 0, 1 ) _
                    + y_ceng(3) * d( log(xg(-1)), 0, 1 ) _
                    + y_ceng(4) * d( log(ceng(-1)), 0, 1 ) _
                    + y_ceng(5) * d( log(veoa(-1)), 0, 1 ) _
                    + y_ceng(6) * hgx(-1)/400


delrff: delrff - delrff_aerr = rff - rff(-1)


dmptlur: dmptlur - dmptlur_aerr = 1/(1+exp(y_dmptlur(1)*(lur-lurtrsh)))


dmptmax: dmptmax - dmptmax_aerr = (@recode((dmptlur)>(dmptpi),dmptlur,dmptpi))


dmptpi: dmptpi - dmptpi_aerr = 1/(1+exp(y_dmptpi(1)*(zpic58-pitrsh)))


dmptr: dmptr - dmptr_aerr = (@recode((dmptmax)>(dmptr(-1)),dmptmax,dmptr(-1)))


dpadj: dpadj - dpadj_aerr - dpadj(-1) = y_dpadj(1) * dpgap(-1)


dpgap: dpgap - dpgap_aerr =  pipxnc/400 - ( _
       .5 * (ehn/(xpn - ecnian)+ ehn(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(phr*pxp), 0, 1) _
     + .5 * (epdn/(xpn - ecnian)+ epdn(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(ppdr*pxp), 0, 1) _
     + .5 * (epin/(xpn - ecnian)+ epin(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(ppir*pxp), 0, 1) _
     + .5 * (epsn/(xpn - ecnian)+ epsn(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(ppsr*pxp), 0, 1) _
     + .5 * (egfon/(xpn - ecnian)+ egfon(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(pgfor*pxp), 0, 1) _
     + .5 * (egfin/(xpn - ecnian)+ egfin(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(pgfir*pxp), 0, 1) _
     + .5 * (egson/(xpn - ecnian)+ egson(-1)/(xpn(-1) - ecnian(-1))) _
              * d(log(pgsor*pxp), 0, 1) _
     + .5 * (egsin/(xpn - ecnian)+ egsin(-1)/(xpn(-1) - ecnian(-1))) _
```

```
                * d(log(pgsir*pxp), 0, 1) _
     + .5 * (exn/(xpn - ecnian)+ exn(-1)/(xpn(-1) - ecnian(-1))) _
                * d(log(pxr*pxp), 0, 1))


ec: log(ec) - ec_aerr = log(ec(-1)) +  _
      .5 * (pcor*pcnia*eco/(ec*pcnia)  _
         + pcor(-1)*pcnia(-1)*eco(-1)/(ec(-1)*pcnia(-1)))  _
           * d(log(eco), 0, 1) _
    + .5 * (pchr*pcnia*ech/(ec*pcnia)  _
         + pchr(-1)*pcnia(-1)*ech(-1)/(ec(-1)*pcnia(-1)))  _
           * d(log(ech), 0, 1) _
    + .5 * ((pcdr*pcnia*yhpcd+pcdr*pcnia*jkcd)/(ec*pcnia)  _
         + (pcdr(-1)*pcnia(-1)*yhpcd(-1)+pcdr(-1)*pcnia(-1)*jkcd(-1))/(ec(-1)*pcnia(-1)))  _
           * d(log(yhpcd+jkcd), 0, 1)


ecd: d( log(ecd), 0, 1) - ecd_aerr  _
                = y_ecd(1) * log(qecd(-1)/ecd(-1)) _
                + y_ecd(2) * d( log(ecd(-1)), 0, 1) _
                + y_ecd(3) * zecd _
                + y_ecd(4) * zgapc2 / 400


ech: d( (ech)/kh(-1), 0, 1 ) - ech_aerr _
                = y_ech(1)  _
                + y_ech(2) * ech(-1)/kh(-2) _
                + y_ech(3) * d( ech(-1)/kh(-2), 0, 1 ) _
                + y_ech(4) * rrmet/100


ecnia: log(ecnia) - ecnia_aerr = log(ecnia(-1)) +  _
      .5 * .01 * (pcor*pcnia*eco/ecnian  _
         + pcor(-1)*pcnia(-1)*eco(-1)/ecnian(-1))  _
           * d(log(eco), 0, 1) _
    + .5 * .01 * (pcdr*pcnia*ecd/ecnian  _
         + pcdr(-1)*pcnia(-1)*ecd(-1)/ecnian(-1))  _
           * d(log(ecd), 0, 1) _
    + .5 * .01 * (pchr*pcnia*ech/ecnian  _
         + pchr(-1)*pcnia(-1)*ech(-1)/ecnian(-1))  _
           * d(log(ech), 0, 1)


ecnian: ecnian - ecnian_aerr = .01*pcnia*ecnia
```

```
eco: d( log(eco), 0, 1) - eco_aerr  _
                = (y_eco(1) * log(qeco(-1)/eco(-1)) _
                + y_eco(2) * d(log(eco(-1)), 0, 1) _
                + y_eco(3) * zeco) * (1-y_eco(4)) _
                + y_eco(4) * (d(log(yhl+yht), 0, 1))


egf: log(egf) - egf_aerr = log(egf(-1)) _
     + .5 * (egfon/egfn + egfon(-1)/egfn(-1)) * d(log(egfo), 0, 1) _
     + .5 * (egfin/egfn + egfin(-1)/egfn(-1)) * d(log(egfi), 0, 1) _
     + .5 * (egfln/egfn + egfln(-1)/egfn(-1)) * d(log(egfl), 0, 1)


egfi: d( log(egfi), 0, 1 ) - egfi_aerr  _
                   = y_egfi(1) _
                   + y_egfi(2) * log(egfi(-1)/egfit(-1)) _
                   +  ( y_egfi(3) *  d( log(egfi(-1)), 0, 1 ) +  y_egfi(4) *  d( log
                   + y_egfi(5) * d( log(egfit), 0, 1 ) _
                   +  ( y_egfi(6) *  xgap2 +  y_egfi(7) *  xgap2(-1))


egfin: egfin - egfin_aerr = .01 * pxp * pgfir * egfi


egfit: d( log(egfit), 0, 1 ) - egfit_aerr  _
                    = y_egfit(1) _
                    + y_egfit(2) * log(.01*pgfir(-1)*pxp(-1)*egfit(-1)/xgdptn(-1)) _
                    + y_egfit(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


egfl: d( log(egfl), 0, 1 ) - egfl_aerr  _
                   = y_egfl(1) _
                   + y_egfl(2) * log(egfl(-1)/egflt(-1)) _
                   +  ( y_egfl(3) *  d( log(egfl(-1)), 0, 1 ) +  y_egfl(4) *  d( log
                   + y_egfl(5) * d( log(egflt), 0, 1 ) _
                   +  ( y_egfl(6) *  xgap2 +  y_egfl(7) *  xgap2(-1))


egfln: egfln - egfln_aerr = .01 * pgfl * egfl


egflt: d( log(egflt), 0, 1 ) - egflt_aerr  _
                    = y_egflt(1) _
                    + y_egflt(2) * log(.01*pgfl(-1)*egflt(-1)/xgdptn(-1)) _
                    + y_egflt(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600
```

```
egfn: egfn - egfn_aerr = egfln + egfin + egfon


egfo: d( log(egfo), 0, 1 ) - egfo_aerr _
                   = y_egfo(1) _
                   + y_egfo(2) * log(egfo(-1)/egfot(-1)) _
                   +  ( y_egfo(3) *  d( log(egfo(-1)), 0, 1 ) +  y_egfo(4) *  d( log(egfo(-2)),
                   + y_egfo(5) * d( log(egfot), 0, 1 ) _
                   +  ( y_egfo(6) *  xgap2 +  y_egfo(7) *  xgap2(-1))


egfon: egfon - egfon_aerr = .01 * pxp * pgfor * egfo


egfot: d( log(egfot), 0, 1 ) - egfot_aerr  _
                    = y_egfot(1) _
                    + y_egfot(2) * log(.01*pgfor(-1)*pxp(-1)*egfot(-1)/xgdptn(-1)) _
                    + y_egfot(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


egpdin: egpdin - egpdin_aerr = epdn + epsn + epin + ehn + ein


egs: log(egs) - egs_aerr = log(egs(-1)) _
     + .5 * (egson/egsn + egson(-1)/egsn(-1)) * d(log(egso), 0, 1) _
     + .5 * (egsin/egsn + egsin(-1)/egsn(-1)) * d(log(egsi), 0, 1) _
     + .5 * (egsln/egsn + egsln(-1)/egsn(-1)) * d(log(egsl), 0, 1)


egsi: d( log(egsi), 0, 1 ) - egsi_aerr _
                   = y_egsi(1) _
                   + y_egsi(2) * log(egsi(-1)/egsit(-1)) _
                   +  ( y_egsi(3) *  d( log(egsi(-1)), 0, 1 ) +  y_egsi(4) *  d( log(egsi(-2)),
                   + y_egsi(5) * d( log(egsit), 0, 1 ) _
                   +  ( y_egsi(6) *  xgap2 +  y_egsi(7) *  xgap2(-1))


egsin: egsin - egsin_aerr = .01 * pxp * pgsir * egsi


egsit: d( log(egsit), 0, 1 ) - egsit_aerr  _
                    = y_egsit(1) _
                    + y_egsit(2) * log(.01*pgsir(-1)*pxp(-1)*egsit(-1)/xgdptn(-1)) _
                    + y_egsit(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600
```

```
egsl: d( log(egsl), 0, 1 ) - egsl_aerr _
                   = y_egsl(1) _
                   + y_egsl(2) * log(egsl(-1)/egslt(-1)) _
                   +  ( y_egsl(3) *  d( log(egsl(-1)), 0, 1 ) +  y_egsl(4) *  d( log(
                   + y_egsl(5) * d( log(egslt), 0, 1 ) _
                   +  ( y_egsl(6) *  xgap2 +  y_egsl(7) *  xgap2(-1))


egsln: egsln - egsln_aerr = .01 * pgsl * egsl


egslt: d( log(egslt), 0, 1 ) - egslt_aerr  _
                    = y_egslt(1) _
                    + y_egslt(2) * log(.01*pgsl(-1)*egslt(-1)/xgdptn(-1)) _
                    + y_egslt(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


egsn: egsn - egsn_aerr = egsln + egsin + egson


egso: d( log(egso), 0, 1 ) - egso_aerr _
                   = y_egso(1) _
                   + y_egso(2) * log(egso(-1)/egsot(-1)) _
                   +  ( y_egso(3) *  d( log(egso(-1)), 0, 1 ) +  y_egso(4) *  d( log(
                   + y_egso(5) * d( log(egsot), 0, 1 ) _
                   +  ( y_egso(6) *  xgap2 +  y_egso(7) *  xgap2(-1))


egson: egson - egson_aerr = .01 * pxp * pgsor * egso


egsot: d( log(egsot), 0, 1 ) - egsot_aerr  _
                    = y_egsot(1) _
                    + y_egsot(2) * log(.01*pgsor(-1)*pxp(-1)*egsot(-1)/xgdptn(-1)) _
                    + y_egsot(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


eh: d( log(eh), 0, 1 ) - eh_aerr _
                  = y_eh(1) * log(qeh(-1)/eh(-1)) _
                  + y_eh(2) * d( log(eh(-1)), 0, 1 ) _
                  + y_eh(3) * d( log(eh(-2)), 0, 1 ) _
                  + y_eh(4) * zeh _
                  + y_eh(5) * d( rme(-1), 0, 1 ) _
                  + y_eh(6) * d83 * d( rme(-1), 0, 1 )
```

```
ehn: ehn - ehn_aerr = .01 * phr * pxp * eh


ei: ei - ei_aerr = 4*d( ki, 0, 1 )


ein: ein - ein_aerr = .01*pxp*pkir*ei


em: log(em) - em_aerr = log(em(-1))   _
                  + .5 * (emon/emn + emon(-1)/emn(-1)) * d(log(emo), 0, 1) _
                  + .5 * (empn/emn + empn(-1)/emn(-1)) * d(log(emp), 0, 1)


emn: emn - emn_aerr = emon + empn


emo: d( log(emo), 0, 1 ) - emo_aerr _
                  =  y_emo(1) _
                   + y_emo(2) * log(emo(-1)*(pmo(-1)/100)/(uemot(-1)*xgden(-1))) _
                   + y_emo(3) * (xgap2-xgap2(-1))/100 _
                   + y_emo(4) * (xgap2(-1)-xgap2(-2))/100 _
                   + y_emo(5) * log(ddockm) _
                   + y_emo(6) * log(ddockm/ddockm(-1))


emon: emon - emon_aerr = .01 * pmo * emo


emp: emp - emp_aerr = uemp*(ceng-xeng)


empn: empn - empn_aerr = .01*pmp*emp


empt: d( log(empt), 0, 1) - empt_aerr  _
                  = y_empt(1) * log(emp(-1)/empt(-1)) _
                  + y_empt(2) * hgx/400


epd: d( log(epd), 0, 1 ) - epd_aerr = _
 ( y_epd(1)*(log(qepd(-2)/epd(-2)))  _
 +  ( y_epd(2) *  d( log(epd(-1)), 0, 1 ) +  y_epd(3) *  d( log(epd(-2)), 0, 1 ))  _
 + zxbd(-1)  _
 + zvpd(-1) )*(1-y_epd(4))  _
```

```
 + y_epd(4) * (d( log(xbo(-1)), 0, 1 ) + hgvpd(-1))


epdn: epdn - epdn_aerr = 0.01*ppdr*pxp*epd


epi: d( log(epi), 0, 1 ) - epi_aerr = _
 ( y_epi(1)*(log(qepi(-2)/epi(-2)))  _
 +  ( y_epi(2) *  d( log(epi(-1)), 0, 1 ) +  y_epi(3) *  d( log(epi(-2)), 0, 1 ))  _
 + zxbi(-1)  _
 + zvpi(-1) )*(1-y_epi(4))  _
 + y_epi(4) * d( log(xbo(-1)), 0, 1 )


epin: epin - epin_aerr = 0.01*ppir*pxp*epi


eps: d( log(eps), 0, 1 ) - eps_aerr = _
                   (y_eps(1) * log(qeps(-2)/eps(-2)) _
                  +  ( y_eps(2) *  d( log(eps(-1)), 0, 1 ) +  y_eps(3) *  d( log(eps
                  + zxbs(-1) _
                  + zvps(-1)) * (1-y_eps(4)) _
                  + y_eps(4) * (d( log(xbo(-1)), 0, 1 )) _
                  + y_eps(5) * d01q4


epsn: epsn - epsn_aerr = .01 * ppsr * pxp * eps


ex: d( log(ex), 0, 1 ) - ex_aerr _
                 = y_ex(1)  _
                 + y_ex(2) * log(ex(-1)*(pxr(-1)*pxp(-1)*fpx(-1))/(fgdp(-1)*fpc(-1)))
                 + y_ex(3) * (fxgap - fxgap(-1))/100 _
                 + y_ex(4) * (fxgap(-1) - fxgap(-2))/100 _
                 + y_ex(5) * ddockx


exn: exn - exn_aerr = .01*pxp*pxr*ex


fcbn: fcbn - fcbn_aerr = exn - emn + fynin + fcbrn


fcbrn: fcbrn - fcbrn_aerr = ufcbr*pxg*xgpot/100
```

```
fgdp: fgdp - fgdp_aerr = fgdpt*exp(fxgap/100)


fgdpt: d( log(fgdpt), 0, 1 ) - fgdpt_aerr  _
                    = y_fgdpt(1) _
                    + y_fgdpt(2) * log(fgdpt(-1)/xgdpt(-1)) _
                    + y_fgdpt(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


fnicn: d(fnicn, 0, 1)/xgdptn - fnicn_aerr = .54 * d( log(fpc), 0, 1)*fnicn(-1)/xgdptn _
                                  - .67 * d( log(fpx), 0, 1)*fnicn(-1)/xgdptn  _
                                  + rfnict


fniln: fniln - fniln_aerr = fnicn - fnin


fnin: d( fnin, 0, 1 ) - fnin_aerr = .25*fcbn  _
                         + .54 * (d( log(fpc), 0, 1) * fnicn(-1)) _
                         - .32 * (d( log(pgdp), 0, 1) * fniln(-1))   _
                         - .67 * (d( log(fpx), 0, 1) * fnicn(-1)) _
                         + .06 * (d( log(fpx), 0, 1) * fniln(-1))  _
                         + fnirn


fnirn: fnirn - fnirn_aerr = ufnir * xgdpn


fpc: fpc - fpc_aerr = fpc(-1)*exp(fpic/400)


fpcm: fpcm - fpcm_aerr = ufpcm*fpc


fpi10: fpi10-fpi10_aerr = y_fpi10(1) * ( ( fpi10(-1) +  fpi10(-2) +  fpi10(-3) +  fpi10(-4))  /
                 + y_fpi10(2) * fpitrg _
                 + y_fpi10(3) * fxgap(-1) _
                 +  ( y_fpi10(4) * d( log(poilr), 0, 1 ) +  y_fpi10(5) * d( log(poilr(-1)), 0,


fpi10t: fpi10t-fpi10t_aerr = y_fpi10t(1) * fpi10t(-1) _
                   + y_fpi10t(2) * fpi10


fpic: fpic-fpic_aerr = y_fpic(1) _
               + y_fpic(2) * fpi10 _
```

```
               + y_fpic(3) * fpic(-1)


fpx: fpx - fpx_aerr = fpxr*fpc/pcpi


fpxm: fpxm - fpxm_aerr = ufpxm*fpx*fpcm/fpc


fpxr: log(fpxr) - fpxr_aerr - log(fpxrr) = _
                        y_fpxr(1)*(rg10e-zpi10f-frl10+fpi10t) _
                      + y_fpxr(2)*(fnin/xgdpn)


fpxrr: d( log(fpxrr), 0, 1 ) - fpxrr_aerr _
                    = y_fpxrr(1) * log(fpxrrt(-1)/fpxrr(-1)) _
                    + y_fpxrr(2) * d( log(fpxrr(-1)), 0, 1 ) _
                    + (1-y_fpxrr(2)) * d( log(fpxrrt), 0, 1 )


frl10: frl10 - frl10(-1) - frl10_aerr = y_frl10(1) _
                              + y_frl10(2) * (frl10(-1) - frs10(-1)) _
                              + y_frl10(3) * (frl10(-1) - frl10(-2)) _
                              + y_frl10(4) * (frs10 - frs10(-1)) _
                              + y_frl10(5) * (fxgap - fxgap(-1))


frs10: frs10 - frs10_aerr = dfmprr * (y_frs10(1) _
                    + y_frs10(2) * frstar(-1) _
                    + y_frs10(3) * ( ( fpi10 +  fpi10(-1) +  fpi10(-2) +  fpi10(-3))
                    + y_frs10(4) * ( ( fpi10 +  fpi10(-1) +  fpi10(-2) +  fpi10(-3))
                    + y_frs10(5) * fxgap) _
                    + (1-dfmprr) * (rfrs10 +  ( fpi10 +  fpi10(-1) +  fpi10(-2) +  fpi


frstar: frstar - frstar_aerr = y_frstar(1) * frstar(-1) _
                      + y_frstar(2) * (frs10 -  ( fpi10 +  fpi10(-1) +  fpi10(-2) +  f


ftcin: ftcin - ftcin_aerr = uftcin * ynicpn


fxgap: fxgap - fxgap_aerr =  _
                    + y_fxgap(1) * fxgap(-1) _
                    + y_fxgap(2) * fxgap(-2) _
                    + y_fxgap(3) * ( ( frs10(-1) _
```

```
                      -(fpi10(-1)+fpi10(-2)+fpi10(-3)+fpi10(-4))/4 +  frs10(-2) _
                      -(fpi10(-2)+fpi10(-3)+fpi10(-4)+fpi10(-5))/4 +  frs10(-3) _
                      -(fpi10(-3)+fpi10(-4)+fpi10(-5)+fpi10(-6))/4) /3-frstar) _
                    + y_fxgap(4) * xgap2(-1)


fynicn:  fynicn - fynicn_aerr = .01*rfynic*fnicn(-1)


fyniln:  fyniln - fyniln_aerr = .01*rfynil*fniln(-1)


fynin: fynin - fynin_aerr = fynicn - fyniln


gfdbtn: gfdbtn - gfdbtn_aerr = ugfdbt*(gfdbtn(-1) - .25*gfsrpn + .25*egfin _
                               - .25*jygfgn - .25*jygfen)


gfintn: gfintn - gfintn_aerr = rgfint*gfdbtn(-1)


gfs: d( log(gfs), 0, 1 ) - gfs_aerr  _
                   = y_gfs(1) _
                   + y_gfs(2) * log(gfsn(-1)/xgdptn(-1)) _
                   + y_gfs(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


gfsn: gfsn - gfsn_aerr = .01*pgdp*gfs


gfsrpn: gfsrpn - gfsrpn_aerr = tfpn + tfcin + tfibn + tfsin + tfdiv _
                     - egfln - egfon - gftn - gfintn _
                     - gfsubn - gfsn


gfsub: d( log(gfsub), 0, 1 ) - gfsub_aerr  _
                     = y_gfsub(1) _
                     + y_gfsub(2) * log(gfsubn(-1)/xgdptn(-1)) _
                     + y_gfsub(3) * (hggdpt+hggdpt(-1)+hggdpt(-2)+hggdpt(-3)) / 1600


gfsubn: gfsubn - gfsubn_aerr = .01*pgdp*gfsub


gft: gft - gft_aerr = (gftrd+gftrt)*xgdpt
```

```
gftn: gftn - gftn_aerr = .01*pgdp*gft


gftrd: gftrd - gftrd_aerr = y_gftrd(1) _
                  + y_gftrd(2) * gftrd(-1) _
                  + y_gftrd(3) * xgap2


gsdbtn: gsdbtn - gsdbtn_aerr = ugsdbt*(gsdbtn(-1) - .25*gssrpn + .25 * egsin _
                              - .25*jygsgn - .25*jygsen)


gsintn: gsintn - gsintn_aerr = rgfint*gsdbtn(-1) + ugsint*xbn


gssrpn: gssrpn - gssrpn_aerr = tspn + tscin + tsibn + tssin + gfsn  _
                     - egsln - egson - gstn - gsintn - gssubn


gssub: gssub - gssub_aerr = ugssub*xgdpt


gssubn: gssubn - gssubn_aerr = .01*pgdp*gssub


gst: gst - gst_aerr = (gstrd+gstrt)*xgdpt


gstn: gstn - gstn_aerr = .01*pgdp*gst


gstrd: gstrd - gstrd_aerr = y_gstrd(1) _
                  + y_gstrd(2) * gstrd(-1) _
                  + y_gstrd(3) * xgap2


hgemp: hgemp - hgemp_aerr = y_hgemp(1) * hgemp(-1) _
                  + y_hgemp(2) * 400*log(emp/emp(-1))


hggdp: hggdp - hggdp_aerr = 400*d( log(xgdp), 0, 1 )


hggdpt: hggdpt - hggdpt_aerr = hxbt + huxb
```

```
hgpdr: hgpdr - hgpdr_aerr = y_hgpdr(1) * hgpdr(-1) _
                     + y_hgpdr(2) * 400*log(ppdr/ppdr(-1))


hgpir: hgpir - hgpir_aerr = y_hgpir(1) * hgpir(-1) _
                     + y_hgpir(2) * 400*log(ppir/ppir(-1))


hgpkir: hgpkir - hgpkir_aerr = y_hgpkir(1) * hgpkir(-1) _
                     + y_hgpkir(2) * 400*log(pkir/pkir(-1))


hgppsr: hgppsr - hgppsr_aerr = y_hgppsr(1) * hgppsr(-1) _
                     + y_hgppsr(2) * 400*log(ppsr/ppsr(-1))


hgvpd: hgvpd - hgvpd_aerr = y_hgvpd(1) * hgvpd(-1) _
                     + y_hgvpd(2) * log(vpd/vpd(-1))


hgvpi: hgvpi - hgvpi_aerr = y_hgvpi(1) * hgvpi(-1) _
                     + y_hgvpi(2) * log(vpi/vpi(-1))


hgvps: hgvps - hgvps_aerr = y_hgvps(1) * hgvps(-1) _
                     + y_hgvps(2) * log(vps/vps(-1))


hgx: hgx - hgx_aerr = (.7*(hlept + hqlww + 400*d( log(lqualt), 0, 1 )) + .265*hks _
               + .035*400*d( log(veoa), 0, 1 ) + hmfpt)/.965


hgynid: hgynid - hgynid_aerr = 400*d( log((ynicpn-tfcin-tscin)*.5/pxg), 0, 1 )


hks: hks - hks_aerr = 400 * (ykpdn * d( log(kpd), 0, 1 ) _
                     +  ykpsn * d( log(kps), 0, 1 ) + ykin * d( log(ki), 0, 1 )) / _
                     (ykpdn + ykpsn + ykin) + hksr


hlept: hlept - hlept_aerr = (1-dmpstb) * 400 *  _
        (hqlfpr * n16 * (1-.01*lurnat-qleor) _
      + d( n16, 0, 1) * qlfpr * (1-.01*lurnat-qleor) _
      - d( left, 0, 1) _
```

```
      - d( lest, 0, 1) ) _
      /  ( leppot/2 +  leppot(-1)/2)  _
      + dmpstb * 400 * d( log(n16), 0, 1 )


hlprdt: hlprdt - hlprdt_aerr = hgx - hlept - hqlww


hmfpt: hmfpt - hmfpt_aerr = y_hmfpt(1) + y_hmfpt(2)*hmfpt(-1)


hqlfpr: hqlfpr - hqlfpr_aerr = y_hqlfpr(1) + y_hqlfpr(2)*hqlfpr(-1)


hqlww: hqlww - hqlww_aerr = y_hqlww(1) * hqlww(-1) + (1-y_hqlww(1)) * y_hqlww(2)


huqpct: huqpct - huqpct_aerr = y_huqpct(1) + y_huqpct(2)*huqpct(-1)


huxb: huxb - huxb_aerr = (1-dglprd) *(y_huxb(1) + y_huxb(2)*huxb(-1))


hxbt: hxbt - hxbt_aerr = ( hgx  _
  - .5 *(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1))) * 400*d
  (1 -  .5 *(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1))))


jccacn: jccacn - jccacn_aerr = ujccac*(jccan - jygfgn - jygfen - jygsgn - jygsen _
                              -.01*jrh*phr(-1)*pxp(-1)*kh(-1))


jccan: jccan - jccan_aerr = jygfgn + jygfen + jygsgn + jygsen + .01*ujcca*pxp(-1) _
                   * (phr(-1)*kh(-1)*jrh + ppsr(-1)*kps(-1)*jrps _
                     + pkpdr(-1)*kpd(-1)*jrpd)


jkcd: jkcd - jkcd_aerr = jrcd * kcd(-1)


jygfen: jygfen - jygfen_aerr = ujygfe *  (.01 * pgdp * xgdpt)


jygfgn: jygfgn - jygfgn_aerr = ujygfg *  (.01 * pgdp * xgdpt)
```

```
jygsen: jygsen - jygsen_aerr = ujygse *  (.01 * pgdp * xgdpt)


jygsgn: jygsgn - jygsgn_aerr = ujygsg *  (.01 * pgdp * xgdpt)


jyncn: jyncn - jyncn_aerr = jccan - jccacn - jygfgn - jygfen - jygsgn - jygsen


kcd: kcd - kcd_aerr = .25*ecd + (1-jrcd/4)*kcd(-1)


kh: kh - kh_aerr = .25*eh + (1-jrh/4)*kh(-1)


ki: d( log(ki), 0, 1 ) - ki_aerr _
                 = y_ki(5) _
                 + y_ki(1) * (log(qkir) - log(ki(-1)/xfs(-1))) _
                 + y_ki(2) * (d( log(ki(-1)), 0, 1 ) - y_ki(5)) _
                 + y_ki(3) * d( log(xfs(-1)), 0, 1 ) _
                 + y_ki(4) * d( log(xfs(-2)), 0, 1 )


kpd: kpd - kpd_aerr = 0.25 * epd + (1-jrpd/4) * kpd(-1)


kpi: kpi - kpi_aerr = 0.25 * epi + (1-jrpi/4) * kpi(-1)


kps: kps - kps_aerr = 0.25 * eps + (1-jrps/4) * kps(-1)


ks: log(ks) - ks_aerr = log(ks(-1)) + hks/400


lef: d( log(lef), 0, 1 ) - lef_aerr = d( log(ulef), 0, 1 )  _
                       + d( log(egfl), 0, 1 )  _
                - dglprd*(d( log(lprdt), 0, 1 ))


left: left - left_aerr = y_left(1) * left(-1) * (hqlfpr+n16/n16(-1))  _
                + y_left(2) * lef


leh: leh - leh_aerr = lep + leo + les + lef
```

```
leo: log(leo) - leo_aerr = log(qleor*qlf) + y_leo(1)*log(leo(-1)/(qleor(-1)*qlf(-1)))
                                          + y_leo(2)*xgap2(-1)


lep: lep - lep_aerr = lhp / lww


leppot: leppot - leppot_aerr = qlf*(1-.01*lurnat - qleor) - left - lest


les: d( log(les), 0, 1 ) - les_aerr = d( log(ules), 0, 1 )  _
                        + d( log(egsl), 0, 1 )  _
                  - dglprd*(d( log(lprdt), 0, 1 ))


lest: lest - lest_aerr = y_lest(1) * lest(-1) * (hqlfpr+n16/n16(-1))  _
                 + y_lest(2) * les


lf: lf - lf_aerr = lfpr * n16


lfpr: d( lfpr, 0, 1)  - lfpr_aerr = hqlfpr _
                          + y_lfpr(1) * (qlfpr(-1) - lfpr(-1)) _
                          + y_lfpr(2) * (lur(-1) - lurnat(-1))


lhp: d( log(lhp), 0, 1 ) - lhp_aerr = _
                    y_lhp(1) * (log(qlhp(-1)/lhp(-1))-d( log(mfpt), 0, 1 )/.965) _
                  + y_lhp(2) * d( log(lhp(-1)), 0, 1 ) _
                  + y_lhp(3) * zlhp _
                  + y_lhp(4) * (d( log(xgo), 0, 1 ) - hlprdt(-1)/400 - d( hmfpt, 0, 1
                  + y_lhp(5) * (d( log(xgo(-1)), 0, 1 ) - hlprdt(-2)/400 - d( hmfpt(-


lprdt: log(lprdt) - lprdt_aerr  = log(xgpot) - log(leppot) - log(qlww)


lur: lur - lur_aerr = 100*(1 - leh/lf)


lurbls: lurbls - lurbls_aerr = lur


lurnat: lurnat - lurnat_aerr = lurnat(-1)
```

```
lww: d( log(lww), 0, 1 ) - lww_aerr _
                  = hqlww/400 _
                  + y_lww(1) * log(qlww(-1)/lww(-1)) _
                      + y_lww(2) * (d( log(lhp), 0, 1 ) - (hlept + hqlww)/400)


mei: log(mei) - mei_aerr = y_mei(1) * log(mei(-1))


mep: log(mep) - mep_aerr = y_mep(1) * log(mep(-1))


mfpt: log(mfpt) - mfpt_aerr = y_mfpt(1) + log(mfpt(-1)) + hmfpt/400


pcdr: d(log(pcdr), 0, 1) - pcdr_aerr = y_pcdr(1) _
                             + y_pcdr(2)*d(log(pcdr(-1)), 0, 1)


pceng: pceng - pceng_aerr = pcengr*pxb


pcengr: d( log(pcengr), 0, 1 ) - pcengr_aerr _
                    = y_pcengr(1) _
                    + y_pcengr(2) * d( log(pcengr(-1)), 0, 1 ) _
                    + y_pcengr(3) * log(pcengr(-1)) _
                    + y_pcengr(4) * log(poilr(-1)) _
                    + y_pcengr(5) * d( log(poilr), 0, 1 )


pcer: d( log(pcer), 0, 1 ) - pcer_aerr _
                  = y_pcer(1) * log((y_pcer(2) *pceng(-1) + (1-y_pcer(2))*pcxfe(-1))/(pcer(-1)
                  + y_pcer(3) * d( log((y_pcer(2) *pceng + (1-y_pcer(2))*pcxfe)/pcxfe), 0, 1 )
                  + y_pcer(4) * d( log((y_pcer(2) *pceng(-1) + (1-y_pcer(2))*pcxfe(-1))/pcxfe(


pcfr: d( log(pcfr), 0, 1 ) - pcfr_aerr _
                  = y_pcfr(1) * log(pcfr(-1)/pcfrt(-1)) _
                  + y_pcfr(2) _
                  +  ( y_pcfr(3) *  d( log(pcfr(-1)), 0, 1 ) +  y_pcfr(4) *  d( log(pcfr(-2)),
                  + y_pcfr(6) * d( log(pcfrt), 0, 1 )


pchr: d(log(pchr), 0, 1) - pchr_aerr = y_pchr(1) _
```

```
                              + y_pchr(2)*d(log(pchr(-1)), 0, 1)


pcnia: d( log(pcnia), 0, 1 ) - pcnia_aerr = picnia / 400


pcor: log(pcor) - log(pcor(-1)) - pcor_aerr =   _
           (- .5 * .01 * (pcdr*pcnia*ecd/ecnian  _
         + pcdr(-1)*pcnia(-1)*ecd(-1)/ecnian(-1)))  _
         / (.5 * .01 * (pcor*pcnia*eco/ecnian  _
         + pcor(-1)*pcnia(-1)*eco(-1)/ecnian(-1))) _
           * d(log(pcdr), 0, 1) _
    - .5 * .01 * (pchr*pcnia*ech/ecnian  _
         + pchr(-1)*pcnia(-1)*ech(-1)/ecnian(-1))  _
           * d(log(pchr), 0, 1) _
    / (.5 * .01 * (pcor*pcnia*eco/ecnian  _
         + pcor(-1)*pcnia(-1)*eco(-1)/ecnian(-1)))


pcpi: pcpi - pcpi_aerr = upcpi * exp(.025*log(pcer)) * pcnia


pcpix: pcpix - pcpix_aerr = upcpix * pcxfe


pcxfe: d(log(pcxfe), 0, 1) - pcxfe_aerr = picxfe/400


pgdp: pgdp - pgdp_aerr = 100*xgdpn/xgdp


pgfir: log(pgfir) - pgfir_aerr - log(pgfir(-1)) = y_pgfir(1) + pipxnc/400 + dpadj - c


pgfl: d( log(pgfl), 0, 1 ) - pgfl_aerr = d( log(upgfl), 0, 1 )  _
                        + d( log(pl), 0, 1 )  _
                  - dglprd*(d( log(lprdt), 0, 1 ))


pgfor: log(pgfor) - pgfor_aerr - log(pgfor(-1)) = y_pgfor(1) + pipxnc/400 + dpadj - c


pgsir: log(pgsir) - pgsir_aerr - log(pgsir(-1)) = y_pgsir(1) + pipxnc/400 + dpadj - c


pgsl: d( log(pgsl), 0, 1 ) - pgsl_aerr = d( log(upgsl), 0, 1 )  _
```

```
                        + d( log(pl), 0, 1 )  _
                    - dglprd*(d( log(lprdt), 0, 1 ))


  pgsor: log(pgsor) - pgsor_aerr - log(pgsor(-1)) = y_pgsor(1) + pipxnc/400 + dpadj - d(log(pxp),


  phouse: d( log(phouse), 0, 1) - phouse_aerr = y_phouse(1) + y_phouse(2) * d( log(phouse(-1)), 0
                        + y_phouse(3) * log(phouse(-1)/(pchr(-1)*pcnia(-1)))


  phr: log(phr) - phr_aerr - log(phr(-1)) = y_phr(1) + pipxnc/400 + dpadj - d(log(pxp), 0, 1)


  pic4:             pic4 - pic4_aerr = 100*(pcnia/pcnia(-4) - 1)


  picngr: picngr - picngr_aerr = (d( log(pceng/pxp(-1)), 0, 1 ) *  _
                        ( pceng*ceng/(pxp*xp) +  pceng(-1)*ceng(-1)/(pxp(-1)*xp(-1)))  / 2 )


  picnia: picnia - picnia_aerr = picxfe _
                      + ( ( ucfs +  ucfs(-1)) /2) * 400 * d(log(pcfr), 0, 1) _
                      + ( ( uces +  uces(-1)) /2) * 400 * d(log(pcer), 0, 1)


  picx4:             picx4 - picx4_aerr = 100*(pcxfe/pcxfe(-4) - 1)


  picxfe: picxfe - picxfe_aerr = (y_picxfe(1)*picxfe(-1) _
                      + y_picxfe(3)*zpicxfe _
                      + (1-y_picxfe(3))*(1-y_picxfe(1))*ptr(-1) _
                      + y_picxfe(2)*400*log(qpcnia(-1)/pcnia(-1))) / (1+y_picxfe(1)*y_picxfe(3))


  pieci: pieci - pieci_aerr = (.25*y_pieci(1)*((1-y_pieci(4))*(pieci(-1)+pieci(-2)+pieci(-3)) + p
                      + y_pieci(4)*zpieci _
                      + (1-y_pieci(4))*(1-y_pieci(1))*(ptr(-1) + hlprdt(-1) - 400*huqpct(-1)) _
                      + y_pieci(2)*(lur(-1)-lurnat(-1)) _
                      + y_pieci(3)*400*log(qpl(-1)/pl(-1))) / (1+.25*y_pieci(1)*y_pieci(4))


  pigdp: pigdp - pigdp_aerr = 400*d( log(pgdp), 0, 1 )


  pipl: pipl - pipl_aerr = pieci
```

```
pipxnc: pipxnc - pipxnc_aerr = picnia - 1.99 * 400 * huqpct _
                     + y_pipxnc(1) * (pipxnc(-1) - picnia(-1) + 1.99 * 400 * huqpct(-
                     + y_pipxnc(2) * (pipxnc(-2) - picnia(-2) + 1.99 * 400 * huqpct(-
                     + y_pipxnc(3) * .5 * ( ( (emon/xpn) +  (emon(-1)/xpn(-1))) ) * 4


pkpdr: pkpdr - pkpdr_aerr = upkpd * ppdr


pl: log(pl) - pl_aerr = log(pl(-1)) + pipl/400


plmin: plmin - plmin_aerr = plminr*.01*pl


pmo: d( log(pmo), 0, 1 ) - pmo_aerr = y_pmo(1) _
                   + y_pmo(2) * (log(qpmo) + .64*log(fpcm(-1)/fpxm(-1)) + .36*log(pxb(
                            - log(pmo(-1))) _
                   + y_pmo(3) * d( log(fpcm/fpxm), 0, 1 ) _
                   + y_pmo(4) * d( log(pxb), 0, 1 )


pmp: pmp - pmp_aerr = upmp*poil


poil: poil - poil_aerr = poilr*pxb


poilr: d( log(poilr), 0, 1 ) - poilr_aerr _
                     = y_poilr(1) * log(poilr(-1)/poilrt(-1)) _
                     + y_poilr(2) _
                     + y_poilr(3) * d( log(poilr(-1)), 0, 1 ) _
                     + y_poilr(4) * d( log(poilrt), 0, 1 )


ppdr: log(ppdr) - ppdr_aerr - log(ppdr(-1)) = y_ppdr(1) + pipxnc/400 + dpadj - d(log(


ppir: log(ppir) - ppir_aerr - log(ppir(-1))  = pipxnc/400 + dpadj - d(log(pxp), 0, 1)


ppsr: log(ppsr) - ppsr_aerr - log(ppsr(-1)) = y_ppsr(1) + pipxnc/400 + dpadj - d(log(
```

```
ptr: ptr - ptr_aerr = y_ptr(1)*ptr(-1) + y_ptr(2)*picxfe(-1)+ y_ptr(3)*pitarg(-1)


pwstar: pwstar - pwstar_aerr = y_pwstar(1) + y_pwstar(2)*pwstar(-1)


pxb: pxb - pxb_aerr = upxb*pgdp


pxg: pxg - pxg_aerr = 100*xgn/xg


pxnc: d( log(pxnc), 0, 1 ) - pxnc_aerr = pipxnc/400


pxp: d( log(pxp), 0, 1 ) - pxp_aerr =  _
       .5*( ecnian/xpn + ecnian(-1)/xpn(-1)) * d( log(pcnia), 0, 1) _
     + .5*( (xpn-ecnian)/xpn + (xpn(-1)-ecnian(-1))/xpn(-1)) * d( log(pxnc), 0, 1)


pxr: log(pxr) - pxr_aerr - log(pxr(-1)) = y_pxr(1) + pipxnc/400 + dpadj - d(log(pxp), 0, 1)


qec: qec - qec_aerr = y_qec(1) * zyh  _
               + y_qec(2) * (dcon*(zyh-zyht)) _
               + y_qec(3) * zyht _
               + y_qec(4) * zyhp _
               + y_qec(5) * (wps+wpo)


qecd: qecd - qecd_aerr = qec _
                 * (jrcd/4 + hggdpt/400 + y_qecd(1)*hgpcdr/400) _
                 * exp(y_qecd(2) + y_qecd(3)*log(pcdr*rccd))


qeco: log(qeco) - qeco_aerr = log(qec) - log(pcor) + y_qeco(1)


qeh: qeh - qeh_aerr = qec _
               * (jrh/4 + hggdpt/400) _
               * exp(y_qeh(1) - log(phr*pxp/pcnia) + y_qeh(2)*log(rcch))


qepd: log(qepd) - qepd_aerr = y_qepd(1)  _
                        + y_qepd(2) * log(xbo) _
                        + y_qepd(3) * log(vpd) _
```

```
                          + y_qepd(4) * log(hgx/100 + jrpd )


qepi: log(qepi) - qepi_aerr = y_qepi(1)  _
                          + y_qepi(2) * log(xbo) _
                          + y_qepi(3) * log(vpi) _
                          + y_qepi(4) * log(hgx/100 + jrpi )


qeps: log(qeps) - qeps_aerr = y_qeps(1)  _
                      + y_qeps(2) * log(xbo) _
                      + y_qeps(3) * log(vps) _
                      + y_qeps(4) * log(hgx/100 + jrps )


qkir: log(qkir) - qkir_aerr = (1-dglprd)*y_qkir(1) + log(qkir(-1))


qlep: qlep - qlep_aerr = lhp / qlww


qlf: qlf - qlf_aerr = qlfpr * n16


qlfpr: qlfpr - qlfpr_aerr = qlfpr(-1) + hqlfpr


qlhp: qlhp - qlhp_aerr = xgo/lprdt


qlww: log(qlww) - qlww_aerr = log(qlww(-1)) + hqlww(-1)/400


qpcnia: log(qpcnia) - qpcnia_aerr = log(qpxp) + log(uqpct)


qpl: log(qpl) - qpl_aerr = log(pl) + y_qpl(1) * log(pxg/qpxg)


qpmo: log(qpmo) - qpmo_aerr = log(qpmo(-1)) + y_qpmo(1)


qpxg: log(qpxg) - qpxg_aerr = log(pwstar) + y_qpxg(1) + y_qpxg(2)*log(pl/lprdt)


qpxnc: log(qpxnc) - qpxnc_aerr = log(pxnc)  _
```

```
                        + y_qpxnc(1) * log(qpxp/pxp)  _
                        + y_qpxnc(2) * log(qpcnia/pcnia)


  qpxp: qpxp - qpxp_aerr = 100*(xpn + (.01*qpxg*xg-xgn))/xp


  qynidn: log(qynidn) - qynidn_aerr = y_qynidn(1)  _
                           + y_qynidn(2)*d79a  _
                           + y_qynidn(3)*log((@recode((ynicpn-tfcin-tscin)>(.01),ynicpn-tfcin-ts


  rbbb: rbbb - rbbb_aerr = ( ( (0.01*rbbbe + 1)^.5 - 1 ) * 200 )


  rbbbe: rbbbe - rbbbe_aerr = rbbbp + rg10e


  rbbbp: rbbbp - rbbbp_aerr = y_rbbbp(1) _
                    + y_rbbbp(2) * zgap10 _
                    + y_rbbbp(3) * (rbbbp(-1) - y_rbbbp(4) - y_rbbbp(5)*zgap10(-1))


  rcar: rcar - rcar_aerr = y_rcar(1) _
                  + y_rcar(2) * d79a _
                  + y_rcar(3) * ((1-d79a)*t47) _
                  + y_rcar(4) * rcar(-1) _
                  +  ( y_rcar(5) *  rg5 +  y_rcar(6) *  rg5(-1))


  rccd: rccd - rccd_aerr = (@recode((100*jrcd + rcar - zpi5)>( .01),100*jrcd + rcar - zpi5, .01))


  rcch: rcch - rcch_aerr = (@recode((100*jrh + (1-trfpm/100)*(rme+100*trspp) - zpi10)>( .1),100*j


  rcgain: rcgain - rcgain_aerr = picx4 + y_rcgain(1) _
                      + y_rcgain(2) * xgap2 _
                      + y_rcgain(3) * (rcgain(-1) - picx4(-1) - y_rcgain(4)  _
                      - y_rcgain(5) * xgap2(-1) )


  req: req - req_aerr = rg30e - zpic30 + reqp


  reqp: reqp - reqp_aerr = y_reqp(1) + y_reqp(2) * rbbbp _
```

```
                        + y_reqp(3) * (reqp(-1) - y_reqp(4) - y_reqp(5)*rbbbp(-1))


rff: rff - rff_aerr = 36000*( (1+.01*rffe)^(1/365) - 1 )


rffalt: rffalt - rffalt_aerr = y_rffalt(1) _
                     + y_rffalt(2) * rff(-1) _
                     + y_rffalt(3) * rff(-2) _
                     + y_rffalt(4) * xgap2 _
                     + y_rffalt(5) * xgap2(-1) _
                     + y_rffalt(6) * ( ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe


rffe: rffe - rffe_aerr = (1-dmptrsh) * (@recode((rffrule)>( rffmin),rffrule, rffmin))
                   + dmptrsh * (@recode(((dmptr(-1)*rffrule + (1-dmptr(-1))*rffmin))>


rffgen: rffgen - rffgen_aerr = y_rffgen(1) _
                     +  ( y_rffgen(2) *  rffe(-1) +  y_rffgen(3) *  rffe(-2) +  y_rff
                     +  ( y_rffgen(6) *  picnia +  y_rffgen(7) *  picnia(-1) +  y_rff
                     +  ( y_rffgen(11) *  xgap2 +  y_rffgen(12) *  xgap2(-1) +  y_rff
                     +  ( y_rffgen(16) *  lur +  y_rffgen(17) *  lur(-1) +  y_rffgen(
                     +  ( y_rffgen(21) *  pcnia +  y_rffgen(22) *  pcnia(-1) +  y_rff
                     +  ( y_rffgen(26) *  rstar +  y_rffgen(27) *  rstar(-1) +  y_rff
                     +  ( y_rffgen(31) *  pitarg +  y_rffgen(32) *  pitarg(-1) +  y_r
                     +  ( y_rffgen(36) *  lurnat +  y_rffgen(37) *  lurnat(-1) +  y_r
                     +  ( y_rffgen(41) *  pcstar +  y_rffgen(42) *  pcstar(-1) +  y_r
                     +  ( y_rffgen(46) *  picxfe +  y_rffgen(47) *  picxfe(-1) +  y_r


rffintay: rffintay - rffintay_aerr = y_rffintay(3) * rffe(-1) _
                     + (1-y_rffintay(3)) * (rstar _
                     +  ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe(-3))  / 4  _
                     + y_rffintay(1) * ( ( picxfe +  picxfe(-1) +  picxfe(-2) +  picx
                     + y_rffintay(2) * xgap2)


rffrule: rffrule - rffrule_aerr =  (@recode((dmpex * 100 * ((1+rfffix/36000)^365-1) _
                       + dmprr  * (rrfix +  ( picxfe +  picxfe(-1) +  picxfe(-2) +  p
                       + dmptay * rfftay  _
                       + dmptlr * rfftlr  _
                       + dmpintay * rffintay  _
                       + dmpalt * 100*((1+rffalt/36000)^365-1) _
                       + dmpgen * rffgen)>(rffmin),dmpex * 100 * ((1+rfffix/36000)^36
                       + dmprr  * (rrfix +  ( picxfe +  picxfe(-1) +  picxfe(-2) +  p
```

```
                        + dmptay * rfftay  _
                        + dmptlr * rfftlr  _
                        + dmpintay * rffintay  _
                        + dmpalt * 100*((1+rffalt/36000)^365-1) _
                        + dmpgen * rffgen,rffmin))


rfftay: rfftay - rfftay_aerr = rstar _
                      +  ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe(-3))  / 4  _
                      + y_rfftay(1) * ( ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe(-3))  / 4
                      + y_rfftay(2) * xgap2


rfftlr: rfftlr - rfftlr_aerr = rstar  _
                        + y_rfftlr(1) * pitarg _
                        + y_rfftlr(2) * ( ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe(-3)) )
                        + y_rfftlr(3) * (lurnat + deuc * leuc - lur)


rfynic: d( rfynic, 0, 1 ) - rfynic_aerr = y_rfynic(1) _
                + y_rfynic(2) * (rfynic(-1)-rfynil(-1)) _
                + y_rfynic(3) * d( rfynic(-1), 0, 1 ) _
                + y_rfynic(4) * d( rfynil, 0, 1 )


rfynil: d( rfynil, 0, 1 ) - rfynil_aerr = y_rfynil(1) _
                + y_rfynil(2) * rfynil(-1) _
                + y_rfynil(3) * rg10(-1) _
                + y_rfynil(4) * rtb(-1) _
                + y_rfynil(5) * reqp(-1) _
                + y_rfynil(6) * d( rfynil(-1), 0, 1 ) _
                + y_rfynil(7) * d( rg10, 0, 1 ) _
                + y_rfynil(8) * d( rtb, 0, 1 ) _
                + y_rfynil(9) * d( reqp, 0, 1 )


rg10: rg10 - rg10_aerr = (( (.01*rg10e + 1)^.5 - 1) * 200)


rg10e: rg10e - rg10e_aerr = zrff10 + rg10p


rg10p: rg10p - rg10p_aerr = y_rg10p(1) _
                    + y_rg10p(2) * zgap10 _
                    + y_rg10p(3) * d8095 _
                    + y_rg10p(4) * (rg10p(-1) - y_rg10p(1)  - y_rg10p(2)*zgap10(-1) - y_rg10p(3)
```

```
rg30: rg30 - rg30_aerr = (( (.01*rg30e + 1)^.5 - 1) * 200)


rg30e: rg30e - rg30e_aerr = zrff30 + rg30p


rg30p: rg30p - rg30p_aerr = y_rg30p(1) _
                   + y_rg30p(2) * zgap30 _
                   + y_rg30p(3) * d8095 _
                   + y_rg30p(4) * (rg30p(-1) - y_rg30p(1) - y_rg30p(2)*zgap30(-1) - y


rg5: rg5 - rg5_aerr = (( (.01*rg5e + 1)^.5 - 1) * 200)


rg5e: rg5e - rg5e_aerr = zrff5 + rg5p


rg5p: rg5p - rg5p_aerr = y_rg5p(1) _
                 + y_rg5p(2) * zgap05 _
                 + y_rg5p(3) * (rg5p(-1) - y_rg5p(1) - y_rg5p(2)*zgap05(-1))


rgfint: rgfint - rgfint_aerr  _
                = (y_rgfint(1) * rgfint(-1) + (1-y_rgfint(1))*rgw(-1))*(gfdbtn(-2)/gf
                + rgw(-1)*(1-gfdbtn(-2)/gfdbtn(-1)) + y_rgfint(2)


rgw: rgw - rgw_aerr = y_rgw(1) * rtb _
               + y_rgw(2) * rg5 _
               + y_rgw(3) * rg10 _
               + y_rgw(4) * rg30


rme: d( rme, 0, 1 ) - rme_aerr = y_rme(1) _
               + y_rme(2) * d( rg10e, 0, 1) _
               + y_rme(3) * d87 * d( rg10e, 0, 1) _
               + y_rme(4) * (rg10e(-1)-rme(-1)) _
               + y_rme(5) * d87 * (rg10e(-1)-rme(-1))


rpd: rpd - rpd_aerr = 0.5*(7.2 + (1-trfcim)*(rg5e + rbbbe- rg10e) - zpib5) + 0.5*req
```

```
rrffe: rrffe - rrffe_aerr = rffe -  ( picxfe +  picxfe(-1) +  picxfe(-2) +  picxfe(-3))  / 4


rrmet: rrmet - rrmet_aerr = y_rrmet(1) * rrmet(-1) _
                  + y_rrmet(2) * (rme-zpi10)


rrtr: rrtr - rrtr_aerr = y_rrtr(1) * rrtr(-1) _
                + y_rrtr(2) * rrffe


rspnia: rspnia - rspnia_aerr =  100 * yhsn / ydn


rstar: rstar - rstar_aerr = rstar(-1) _
                  + y_rstar(1) * ((rrffe-rstar(-1))*drstar)


rtb: rtb - rtb_aerr = 36000/90 * (1-(.01*rtbe+1)^(-90/365))


rtbe: rtbe - rtbe_aerr = y_rtbe(1) _
                +  ( y_rtbe(2) *  rtbe(-1) +  y_rtbe(3) *  rtbe(-2))  _
                +  ( y_rtbe(4) *  rffe +  y_rtbe(5) *  rffe(-1))


rtinv: rtinv - rtinv_aerr = (.01*rpd - .01*hgpkir) _
                  * ( ( pxp*pkir +  pxp(-1)*pkir(-1)) /2)/pxb


rtpd: rtpd - rtpd_aerr = (.01*rpd + jrpd - .01*hgpdr) _
                  * ((1-.01*tapdt-trfcim*(1-tapddp*.01*tapdt)*tapdd)/(1-trfcim)) _
                  * ( ( pxp*pkpdr +  pxp(-1)*pkpdr(-1)) /2)/pxb


rtpi: rtpi - rtpi_aerr = (.01*rpd + jrpi - .01*hgpir) _
                  * ( ( pxp*ppir +  pxp(-1)*ppir(-1)) /2)/pxb


rtps: rtps - rtps_aerr = (@recode(((.01*rpd + jrps - .01*hgppsr)  _
                * ((1-trfcim*tapsda)/(1-trfcim)) _
                * ( ( pxp*ppsr +  pxp(-1)*ppsr(-1)) /2)/pxb)>( .02),(.01*rpd + jrps - .01*hgpp
                * ((1-trfcim*tapsda)/(1-trfcim)) _
                * ( ( pxp*ppsr +  pxp(-1)*ppsr(-1)) /2)/pxb, .02))
```

```
rtr: rtr - rtr_aerr = rrtr + ptr


tapdd: tapdd - tapdd_aerr = .5 * d2003 + .5 * d2003 * (2.0 / (2.0 + .01 * tapds * (rp
                      + .3 * d2002 + .7 * d2002 * (2.0 / (2.0 + .01 * tapds * (rpd + z
                      + (d87 - d2002 - d2003) * (2.0 / (2.0 + .01 * tapds * (rpd + zpi
                      + (d81-d87) * (1.5 / (1.5 + .01 * tapds * (rpd + zpib5))) _
                      + (1-d81) _
                            * (((1-tapdad)*(1-exp(-(.01*tapds*(rpd+zpib5)))) _
                                      /(.01*tapds*(rpd+zpib5))) _
                                + tapdad *2*(1-(1-exp(-(.01*tapds*(rpd+zpib5)))) _
                                     /(.01*tapds*(rpd+zpib5))) _
                                     /(.01 * tapds * (rpd + zpib5)))


tapsda: tapsda - tapsda_aerr = (1-tapsad)*(1-exp(-0.01*(rpd+zpib5)*tapssl))/ _
                      (0.01*(rpd+zpib5)*tapssl) + _
                      tapsad*(1-d69) * 2 *  _
                      (1 - (1-exp(-0.01*(rpd+zpib5)*tapssl))/ _
                      (0.01*(rpd+zpib5)*tapssl)) / (0.01*(rpd+zpib5)*tapssl) _
                      + tapsad*(d69-d81) *(  (1.5 /  _
                      (1.5 + .01 * tapssl * (rpd + zpib5))) *  _
                      (1 - exp(-0.5-0.33*(0.01*(rpd+zpib5)*tapssl))) + _
                      (exp(-0.5)/(0.67*(0.01*(rpd+zpib5)*tapssl)))* _
                      (exp(-0.33*(0.01*(rpd+zpib5)*tapssl)) -  _
                      exp(-(0.01*(rpd+zpib5)*tapssl))) _
                      + tapsad * (d81-d86) *(  (1.75 /  _
                      (1.75 + .01 * tapssl * (rpd + zpib5))) *  _
                      (1 - exp(-0.75-0.428*(0.01*(rpd+zpib5)*tapssl))) + _
                      (exp(-0.75)/(0.572*(0.01*(rpd+zpib5)*tapssl)))* _
                      (exp(-0.428*(0.01*(rpd+zpib5)*tapssl)) -  _
                      exp(-(0.01*(rpd+zpib5)*tapssl))) _
                      + tapsad * d86 * (1-exp(-0.01*(rpd+zpib5)*tapssl))/ _
                      (0.01*(rpd+zpib5)*tapssl)


tfcin: tfcin - tfcin_aerr = trfci * ynicpn


tfibn: tfibn - tfibn_aerr = trfib * ecnian


tfpn: tfpn - tfpn_aerr = trfp * (ypn - gftn - gstn)


tfsin: tfsin - tfsin_aerr = trfsi * yniln
```

```
trfci: trfci - trfci_aerr = y_trfci(1) _
                  + y_trfci(2) * trfci(-1) _
                  + y_trfci(3) * trfcim _
                  + y_trfci(4) * .01*pxp*epd*ppdr*.01*tapdt/ynicpn _
                  + y_trfci(5) * xgap2 _
                  + y_trfci(6) * picnia


trfp: trfp - trfp_aerr = y_trfp(1) * trfpt _
                +  ( y_trfp(2) *  (trfp(-1)-trfpt(-1)) +  y_trfp(3) *  (trfp(-2)-trfpt(-2)))
                + y_trfp(4) * xgap2(-1)


trfpt: trfpt - trfpt_aerr = dfpex * trfptx _
                  + dfpdbt * ( trfpt(-1) _
                       + y_trfpt(1) * (gfdbtn(-1)/xgdpn(-1) - gfdrt(-1))  _
                       + y_trfpt(2) * d( gfdbtn(-1)/xgdpn(-1) - gfdrt(-1), 0, 1 ) ) _
                  + dfpsrp * ( trfpt(-1) _
                       + y_trfpt(3) * ((gfsrpn(-1) - egfin(-1) + jygfgn(-1) _
                             + jygfen(-1))/xgdpn(-1) - gfsrt(-1)))


trsci: trsci - trsci_aerr = y_trsci(1) * trsci(-1) _
                  +  ( y_trsci(2) *  trscit +  y_trsci(3) *  trscit(-1))  _
                  +  ( y_trsci(4) *  xgap2 +  y_trsci(5) *  xgap2(-1))  _
                  + y_trsci(6) * d( trfci, 0, 1 )


trsib: trsib - trsib_aerr = y_trsib(1) * trsib(-1) _
                  +  ( y_trsib(2) *  trsibt +  y_trsib(3) *  trsibt(-1))  _
                  + y_trsib(4) * xgap2


trsp: trsp - trsp_aerr = y_trsp(1) * trsp(-1) _
                +  ( y_trsp(2) *  trspt +  y_trsp(3) *  trspt(-1))  _
                + y_trsp(4) * xgap2(-1) _
                + y_trsp(5) * d( trfp, 0, 1 )


trspt: trspt - trspt_aerr = dfpex * trsptx _
                  + dfpdbt * ( trspt(-1) _
                       + y_trspt(1) * (gsdbtn(-1)/xgdpn(-1) - gsdrt(-1))  _
                       + y_trspt(2) * d( gsdbtn(-1)/xgdpn(-1) - gsdrt(-1), 0, 1 ) ) _
                  + dfpsrp * ( trspt(-1) _
```

```
                        + y_trspt(3) * ((gssrpn(-1) - egsin(-1) + jygsgn(-1) _
                              + jygsen(-1))/xgdpn(-1) - gssrt(-1)))


trssi: trssi - trssi_aerr =  ( y_trssi(1) *  trssi(-1) +  y_trssi(2) *  trssi(-2))  _
                  +  ( y_trssi(3) *  trssit +  y_trssi(4) *  trssit(-1))  _
                  + y_trssi(5) * xgap2


tryh: tryh - tryh_aerr = (tfpn+tspn)/(yhln+yhptn)


tscin: tscin - tscin_aerr = trsci * ynicpn


tsibn: tsibn - tsibn_aerr = trsib * ecnian


tspn: tspn - tspn_aerr = trsp * (ypn - gftn - gstn)


tssin: tssin - tssin_aerr = trssi * yniln


uces: d( log(uces), 0, 1 ) - uces_aerr _
                  = y_uces(1) * log(uces(-1)) _
                  + y_uces(2) * log(pcer(-1)) _
                  + y_uces(3) * log(ceng(-1)/xg(-1)) _
                  + y_uces(4) * t47 _
                  + y_uces(5) _
                  + y_uces(6) * d( log(uces(-1)), 0, 1 ) _
                  + y_uces(7) * d( log(pcer), 0, 1 ) _
                  + y_uces(8) * d( log(ceng/xg), 0, 1 )


ucfs: d( log(ucfs), 0, 1 ) - ucfs_aerr  _
                  = y_ucfs(1) * log(ucfs(-1)) _
                  + y_ucfs(2) * log(pcfr(-1)) _
                  + y_ucfs(3) * t47 _
                  + y_ucfs(4) _
                  + y_ucfs(5) * d( log(ucfs(-1)), 0, 1 ) _
                  + y_ucfs(6) * d( log(pcfrt), 0, 1 ) _
                  + y_ucfs(7) * d( log(pcfr/pcfrt), 0, 1 )


uqpct: log(uqpct) - uqpct_aerr = y_uqpct(1) + log(uqpct(-1)) + huqpct
```

```
uxbt: log(uxbt) - uxbt_aerr = y_uxbt(1) + log(uxbt(-1)) + .0025*huxb


veo: log(veo) - veo_aerr = log(pxb/pceng)


veoa: log(veoa) - veoa_aerr = y_veoa(1) * log(veoa(-1)) _
                      + y_veoa(2) * log(veo(-1)) _
                      + uveoa


vpd: vpd - vpd_aerr = uvpd*(pkpdr/ppdr)/rtpd


vpi: vpi - vpi_aerr = uvpi/rtpi


vps: vps - vps_aerr = uvps/rtps


wdnfcn: d( log(wdnfcn), 0, 1) - wdnfcn_aerr  _
                = y_wdnfcn(1) * log(wdnfcn(-1)/(ynin(-1)-yniln(-1))) _
                + y_wdnfcn(2)  _
                + y_wdnfcn(3) * d( log(wdnfcn(-1)), 0, 1) _
                + y_wdnfcn(4) * d( log(wdnfcn(-2)), 0, 1) _
                + y_wdnfcn(5) * xgap2


wpo: wpo - wpo_aerr = wpon/(.01*pcnia)


wpon: wpon - wpon_aerr = wpon(-1)*exp( (1-((phouse(-1)*kh(-1)/116)/wpon(-1)))*rcgain/400  _
+ ((phouse(-1)*kh(-1)/116)/wpon(-1))*d( log(phouse), 0, 1) ) _
                 + .25 * (ydn-ecnian-yhibn) _
                 + .25 * (.01*pcdr*pcnia*(ecd-jkcd))


wps: wps - wps_aerr = wpsn/(.01*pcnia)


wpsn: log(wpsn) - wpsn_aerr = log((ynicpn-tfcin-tscin)*.5)  _
                      - .25 * (req-zdivgr) _
                      + log(25) + 1
```

```
xb: xb - xb_aerr =  xbn/ (pxb/100)


xbn: xbn - xbn_aerr =  pxb/100*xbo + xgdpn -xgdo*pgdp/100


xbo: log(xbo) - xbo_aerr =  log(xbt) + y_xbo(1) * xgap2/100


xbt: log(xbt) - xbt_aerr = log(xb) + (log(xgpot/xg)  _
   - .5 *(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1))) * log
  (1 -  .5 *(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1))))


xeng: xeng - xeng_aerr = uxeng * xgpot


xfs: log(xfs) - xfs_aerr = log(xfs(-1)) _
  + .5*( (ecnian/xfsn + ecnian(-1)/xfsn(-1)) * d(log(ecnia), 0, 1) _
  + (ehn/xfsn + ehn(-1)/xfsn(-1)) * d(log(eh), 0, 1) _
  + (epdn/xfsn + epdn(-1)/xfsn(-1)) * d(log(epd), 0, 1) _
  + (epsn/xfsn + epsn(-1)/xfsn(-1)) * d(log(eps), 0, 1) _
  + (epin/xfsn + epin(-1)/xfsn(-1)) * d(log(epi), 0, 1) _
  + (egfon/xfsn + egfon(-1)/xfsn(-1)) * d(log(egfo), 0, 1) _
  + (egfin/xfsn + egfin(-1)/xfsn(-1)) * d(log(egfi), 0, 1) _
  + (egfln/xfsn + egfln(-1)/xfsn(-1)) * d(log(egfl), 0, 1) _
  + (egson/xfsn + egson(-1)/xfsn(-1)) * d(log(egso), 0, 1) _
  + (egsin/xfsn + egsin(-1)/xfsn(-1)) * d(log(egsi), 0, 1) _
  + (egsln/xfsn + egsln(-1)/xfsn(-1)) * d(log(egsl), 0, 1) _
  + (exn/xfsn + exn(-1)/xfsn(-1)) * d(log(ex), 0, 1) _
  - (emon/xfsn + emon(-1)/xfsn(-1)) * d(log(emo), 0, 1) _
  - (empn/xfsn + empn(-1)/xfsn(-1)) * d(log(emp), 0, 1))


xfsn: xfsn - xfsn_aerr = xgdpn - ein


xg: log(xg) - xg_aerr = log(xg(-1))   _
  + (1 - .5*(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1)))) *
  +  .5*(.035*empn/(.01*pceng*ceng) + .035*empn(-1)/(.01*pceng(-1)*ceng(-1))) * d(log


xgap: xgap - xgap_aerr = 100*log(xgo/xgpot)
```

```
xgap2: xgap2 - xgap2_aerr = 100 * log(xgdo/xgdpt)


xgde: log(xgde) - xgde_aerr = log(xgde(-1)) _
  + .5*( (xgdpn/xgden + xgdpn(-1)/xgden(-1)) * d(log(xgdp), 0, 1) _
   - (exn/xgden + exn(-1)/xgden(-1)) * d(log(ex), 0, 1) _
  + (emon/xgden + emon(-1)/xgden(-1)) * d(log(emo), 0, 1) _
  + (empn/xgden + empn(-1)/xgden(-1)) * d(log(emp), 0, 1))


xgden: xgden - xgden_aerr = xgdpn + emn - exn


xgdi: xgdi - xgdi_aerr = xgdo*mei


xgdin: xgdin - xgdin_aerr = xgdi *(pgdp/100)


xgdo: xgdo - xgdo_aerr = xgdp/mep


xgdp: xgdp - xgdp_aerr = xgdp(-1) * @sqrt( _
          ( (xfsn(-1)/xgdpn(-1)) * (xfs/xfs(-1)) _
           + (.01 * ei(-1)*pkir(-1)*pxp(-1) / xgdpn(-1)) * (ei/ei(-1))) _
          * 1/ _
          ((xfsn/xgdpn) * (xfs(-1)/xfs) _
           + (.01 * ei*pkir*pxp / xgdpn) * (ei(-1)/ei)))


xgdpn: xgdpn - xgdpn_aerr = xpn + ein - emn + egfln + egsln


xgdpt: log(xgdpt) - xgdpt_aerr = log(xbt) + log(uxbt)


xgdptn: xgdptn - xgdptn_aerr = .01*pgdp*xgdpt


xgn: xgn - xgn_aerr = xbn + empn


xgo: log(xgo) - xgo_aerr =  log(xgpot) + y_xgo(1) * xgap2/100


xgpot: log(xgpot) - xgpot_aerr = (y_xgpot(1) * (log(leppot) + log(qlww) + log(lqualt)) _
```

```
                        + y_xgpot(2) * log(ks) _
                        + y_xgpot(3) * log(veoa) _
                        + log(mfpt)) / (1-y_xgpot(4))


xp: log(xp) - xp_aerr = log(xp(-1)) _
  + .5 * (ecnian/xpn + ecnian(-1)/xpn(-1)) * d(log(ecnia), 0, 1) _
  + .5 * (ehn/xpn + ehn(-1)/xpn(-1))       * d(log(eh), 0, 1) _
  + .5 * (epdn/xpn + epdn(-1)/xpn(-1))     * d(log(epd), 0, 1) _
  + .5 * (epin/xpn + epin(-1)/xpn(-1))     * d(log(epi), 0, 1) _
  + .5 * (epsn/xpn + epsn(-1)/xpn(-1))     * d(log(eps), 0, 1) _
  + .5 * (egfon/xpn + egfon(-1)/xpn(-1))   * d(log(egfo), 0, 1) _
  + .5 * (egfin/xpn + egfin(-1)/xpn(-1))   * d(log(egfi), 0, 1) _
  + .5 * (egson/xpn + egson(-1)/xpn(-1))   * d(log(egso), 0, 1) _
  + .5 * (egsin/xpn + egsin(-1)/xpn(-1))   * d(log(egsi), 0, 1) _
  + .5 * (exn/xpn + exn(-1)/xpn(-1))       * d(log(ex), 0, 1)


xpn: xpn - xpn_aerr = .01 * pxp * xp


ycsn: ycsn - ycsn_aerr = ynicpn - tfcin - tscin - ftcin - ynidn + jccacn


ydn: ydn - ydn_aerr = uyd * (ypn - tfpn - tspn)


ygfsn: ygfsn - ygfsn_aerr = gfsrpn + jygfgn + jygfen


ygssn: ygssn - ygssn_aerr = gssrpn + jygsgn + jygsen


yh: yh - yh_aerr = yhl + yht + yhp


yhgap: yhgap - yhgap_aerr = 100*(yhshr/zyhst-1)


yhibn: d( log(yhibn), 0, 1 ) - yhibn_aerr _
                         = y_yhibn(1) *  ( picxfe/1600 +  picxfe(-1)/1600 +  picxfe
                         + y_yhibn(2)  _
                         + y_yhibn(3) * log(ecnian(-1)/yhibn(-1)) _
                         + y_yhibn(4) * (d( log(yhibn(-1)), 0, 1 ) -  ( picxfe(-1)/1
                         + y_yhibn(5) * d79a _
                         + y_yhibn(6) * rcar(-1) _
```

```
                          + y_yhibn(7) * log(.01*pcdr(-1)*pcnia(-1)*ecd(-1)/ecnian(-1)) _
                          + y_yhibn(8) * d( rffe, 0, 1 )


  yhin: yhin - yhin_aerr = uyhi * (yniin + gfintn + gsintn + yhibn)


  yhl: yhl - yhl_aerr = (1-tryh)*yhln/(.01*pcnia)


  yhln: yhln - yhln_aerr = uyhln * (yniln - tfsin - tssin)


  yhp: yhp - yhp_aerr = ((1-tryh)*yhptn+yhpntn)/(.01*pcnia)


  yhpcd: log(yhpcd) - yhpcd_aerr = log(y_yhpcd(1)) + log(kcd(-1))


  yhpgap: yhpgap - yhpgap_aerr = 100*(yhpshr/zyhpst-1)


  yhpntn: yhpntn - yhpntn_aerr = .01*pcnia*pcdr*yhpcd  _
                       - yhibn + ynicpn - tfcin - tscin - ynidn _
                       - .01 * zpi10 *(gfdbtn+gsdbtn)


  yhpshr: yhpshr - yhpshr_aerr = yhp/yh


  yhptn: yhptn - yhptn_aerr = uyhptn*(ynisen+yhin+ynidn)


  yhshr: yhshr - yhshr_aerr = yh/xgdp


  yhsn: yhsn - yhsn_aerr = yhln + yhtn + yhptn - tfpn - tspn - ecnian - yhibn _
                   + uyhsn * xgdptn


  yht: yht - yht_aerr = yhtn/(.01*pcnia)


  yhtgap: yhtgap - yhtgap_aerr = 100*(yhtshr/zyhtst-1)
```

```
yhtn: yhtn - yhtn_aerr = uyhtn*(gftn+gstn)


yhtshr: yhtshr - yhtshr_aerr = yht/yh


ykin: ykin - ykin_aerr = .01*rtinv*pxb* (ki + ki(-1)) /2


ykpdn: ykpdn - ykpdn_aerr = .01*rtpd*pxb* ( kpd +  kpd(-1)) /2


ykpsn: ykpsn - ykpsn_aerr = .01*rtps*pxb* ( kps +  kps(-1)) /2


ynicpn: ynicpn - ynicpn_aerr = uynicp * (@recode((ynin-yniln-yniin-ynisen-tfibn-tsibn


ynidn: d( log((ynidn-ymsdn)/pxb), 0, 1 ) - ynidn_aerr =  _
                          y_ynidn(1) * log(qynidn(-1)/(ynidn(-1)-ymsdn(-1))) _
                        + y_ynidn(2) * d( log((ynidn(-1)-ymsdn(-1))/pxb(-1)), 0, 1 )
                        + y_ynidn(3) * zynid


yniin: yniin/(ynin(-1)-yniln(-1)) - yniin_aerr  _
                = y_yniin(1) _
                + y_yniin(2) * (yniin(-1)/(ynin(-2)-yniln(-2))) _
                + y_yniin(3) * (.01*rrmet*.01*phr(-1)*pxp(-1)*kh(-1)/(ynin(-1)-yniln
                + y_yniin(4) * ((.01*rbbbe)*(wdnfcn(-1)/(ynin(-1)-yniln(-1)))) _
                + y_yniin(5) * (.01*d( rbbbe*(wdnfcn(-1)/(ynin(-1)-yniln(-1))), 0, 1
                + y_yniin(6) * (.01*fnin(-1)/(ynin(-1)-yniln(-1)))


yniln: yniln - yniln_aerr = 0.01 *  uyl * (pl*lhp + pgfl*egfl + pgsl*egsl)


ynin: ynin - ynin_aerr = uyni*(xgdin+fynin-jccan)


ynisen: ynisen - ynisen_aerr = uysen*xbn


ypn: ypn - ypn_aerr = uyp * (yhln + yhtn + yhptn)


zdivgr: zdivgr-zdivgr_aerr = y_zdivgr(1) _
```

```
          +  ( y_zdivgr(2) *  picnia +  y_zdivgr(3) *  picnia(-1) +  y_zdivgr(4) *  picnia(-2)
          +  ( y_zdivgr(6) *  rffe +  y_zdivgr(7) *  rffe(-1) +  y_zdivgr(8) *  rffe(-2) +  y_z
          + y_zdivgr(10) * rtr _
          + y_zdivgr(11) * ptr _
          +  ( y_zdivgr(12) *  xgap +  y_zdivgr(13) *  xgap(-1) +  y_zdivgr(14) *  xgap(-2) +
          +  ( y_zdivgr(16) *  (400*d( log((ynicpn-tfcin-tscin)*.5/(.01*pxg)), 0, 1 )) +  y_zdi
          + y_zdivgr(20) * hgx


zecd: zecd-zecd_aerr =  ( y_zecd(1) *  picnia(-1) +  y_zecd(2) *  picnia(-2) +  y_zecd(3) *  pi
               +  ( y_zecd(5) *  rffe(-1) +  y_zecd(6) *  rffe(-2) +  y_zecd(7) *  rffe(-3) +
               +  ( y_zecd(9) *  xgap2(-1) +  y_zecd(10) *  xgap2(-2) +  y_zecd(11) *  xgap2(-3
               + y_zecd(13) * ptr(-1) _
               + y_zecd(14) * rtr(-1) _
               +  ( y_zecd(15) *  yhgap(-1) +  y_zecd(16) *  yhgap(-2) +  y_zecd(17) *  yhgap(-
               +  ( y_zecd(19) *  yhtgap(-1) +  y_zecd(20) *  yhtgap(-2) +  y_zecd(21) *  yhtga
               +  ( y_zecd(23) *  yhpgap(-1) +  y_zecd(24) *  yhpgap(-2) +  y_zecd(25) *  yhpga
               + y_zecd(27)* (hggdpt(-1)/400) _
               + y_zecd(28)* (hgpcdr(-1)/400) _
               +  ( y_zecd(29) *  d( log(qecd(-1)), 0, 1 ) +  y_zecd(30) *  d( log(qecd(-2)), 0


zeco: zeco-zeco_aerr =  _
                  ( y_zeco(1) *  picnia(-1) +  y_zeco(2) *  picnia(-2) +  y_zeco(3) *  picnia(-3)
             +  ( y_zeco(5) *  rffe(-1) +  y_zeco(6) *  rffe(-2) +  y_zeco(7) *  rffe(-3) +  y_
             +  ( y_zeco(9) *  xgap2(-1) +  y_zeco(10) *  xgap2(-2) +  y_zeco(11) *  xgap2(-3)
             + y_zeco(13) * ptr(-1) _
             + y_zeco(14) * rtr(-1) _
             +  ( y_zeco(15) *  yhgap(-1) +  y_zeco(16) *  yhgap(-2) +  y_zeco(17) *  yhgap(-3)
             +  ( y_zeco(19) *  yhtgap(-1) +  y_zeco(20) *  yhtgap(-2) +  y_zeco(21) *  yhtgap(
             +  ( y_zeco(23) *  yhpgap(-1) +  y_zeco(24) *  yhpgap(-2) +  y_zeco(25) *  yhpgap(
             + y_zeco(27)* ((hggdpt(-1)/400)) _
             +  ( y_zeco(28)  _
             * (d( log(qeco(-1)), 0, 1 )) +  y_zeco(29)  _
             * (d( log(qeco(-2)), 0, 1 )) +  y_zeco(30)  _
             * (d( log(qeco(-3)), 0, 1 )) +  y_zeco(31)  _
             * (d( log(qeco(-4)), 0, 1 )))


zeh: zeh-zeh_aerr =  _
                  ( y_zeh(1) *  picnia(-1) +  y_zeh(2) *  picnia(-2) +  y_zeh(3) *  picnia(-3) +
             +  ( y_zeh(5) *  rffe(-1) +  y_zeh(6) *  rffe(-2) +  y_zeh(7) *  rffe(-3) +  y_zeh
             +  ( y_zeh(9) *  xgap2(-1) +  y_zeh(10) *  xgap2(-2) +  y_zeh(11) *  xgap2(-3) +
             + y_zeh(13) * ptr(-1) _
             + y_zeh(14) * rtr(-1) _
             +  ( y_zeh(15) *  yhgap(-1) +  y_zeh(16) *  yhgap(-2) +  y_zeh(17) *  yhgap(-3) +
```

```
            +  ( y_zeh(19) *  yhtgap(-1) +  y_zeh(20) *  yhtgap(-2) +  y_zeh(21) *
            +  ( y_zeh(23) *  yhpgap(-1) +  y_zeh(24) *  yhpgap(-2) +  y_zeh(25) *
            + y_zeh(27)* (hggdpt(-1)/400) _
            +  ( y_zeh(28) *  d( log(qeh(-1)), 0, 1 ) +  y_zeh(29) *  d( log(qeh(-2)


zgap05:  zgap05-zgap05_aerr = y_zgap05(1) _
                  +  ( y_zgap05(2) *  picnia +  y_zgap05(3) *  picnia(-1) +  y_zgap
                  +  ( y_zgap05(6) *  rffe +  y_zgap05(7) *  rffe(-1) +  y_zgap05(8
                  + y_zgap05(10) * rtr _
                  + y_zgap05(11) * ptr _
                  +  ( y_zgap05(12) *  xgap +  y_zgap05(13) *  xgap(-1) +  y_zgap05


zgap10:  zgap10-zgap10_aerr = y_zgap10(1) _
                  +  ( y_zgap10(2) *  picnia +  y_zgap10(3) *  picnia(-1) +  y_zgap
                  +  ( y_zgap10(6) *  rffe +  y_zgap10(7) *  rffe(-1) +  y_zgap10(8
                  + y_zgap10(10) * rtr _
                  + y_zgap10(11) * ptr _
                  +  ( y_zgap10(12) *  xgap +  y_zgap10(13) *  xgap(-1) +  y_zgap10


zgap30:  zgap30-zgap30_aerr = y_zgap30(1) _
                  +  ( y_zgap30(2) *  picnia +  y_zgap30(3) *  picnia(-1) +  y_zgap
                  +  ( y_zgap30(6) *  rffe +  y_zgap30(7) *  rffe(-1) +  y_zgap30(8
                  + y_zgap30(10) * rtr _
                  + y_zgap30(11) * ptr _
                  +  ( y_zgap30(12) *  xgap +  y_zgap30(13) *  xgap(-1) +  y_zgap30


zgapc2:  zgapc2-zgapc2_aerr =  ( y_zgapc2(1) *  picnia(-1) +  y_zgapc2(2) *  picnia(-
               +  ( y_zgapc2(5) *  rffe(-1) +  y_zgapc2(6) *  rffe(-2) +  y_zgapc2(7)
               +  ( y_zgapc2(9) *  xgap2(-1) +  y_zgapc2(10) *  xgap2(-2) +  y_zgapc2
               + y_zgapc2(13) * ptr(-1) _
               + y_zgapc2(14) * rtr(-1)


zlhp:  zlhp-zlhp_aerr =  ( y_zlhp(1) *  picnia(-1) +  y_zlhp(2) *  picnia(-2) +  y_zl
                +  ( y_zlhp(5) *  rffe(-1) +  y_zlhp(6) *  rffe(-2) +  y_zlhp(7) *  r
                + y_zlhp(9) * rtr(-1) _
                + y_zlhp(10) * ptr(-1) _
                +  ( y_zlhp(11) *  xgap(-1) +  y_zlhp(12) *  xgap(-2) +  y_zlhp(13) *
                + y_zlhp(15) * (d( log(xgo(-1)), 0, 1 ) - (d( log(lprdt(-1)), 0, 1 ))
                + y_zlhp(16) * ((hlept(-1) - hqlww(-1))/400)
```

```
zpi10:  zpi10-zpi10_aerr =  ( y_zpi10(1) *  picnia(-1) +  y_zpi10(2) *  picnia(-2) +  y_zpi10(3
                 +  ( y_zpi10(5) *  rffe(-1) +  y_zpi10(6) *  rffe(-2) +  y_zpi10(7) *  rffe(-
                 + y_zpi10(9) * rtr(-1) _
                 + y_zpi10(10) * ptr(-1) _
                 +  ( y_zpi10(11) *  xgap(-1) +  y_zpi10(12) *  xgap(-2) +  y_zpi10(13) *  xga


zpi10f:  zpi10f-zpi10f_aerr = zpi10


zpi5:  zpi5-zpi5_aerr =  ( y_zpi5(1) *  picnia(-1) +  y_zpi5(2) *  picnia(-2) +  y_zpi5(3) *  p
               +  ( y_zpi5(5) *  rffe(-1) +  y_zpi5(6) *  rffe(-2) +  y_zpi5(7) *  rffe(-3) +
               + y_zpi5(9) * rtr(-1) _
               + y_zpi5(10) * ptr(-1) _
               +  ( y_zpi5(11) *  xgap(-1) +  y_zpi5(12) *  xgap(-2) +  y_zpi5(13) *  xgap(-3)


zpib5:  zpib5-zpib5_aerr = y_zpib5(1) _
         +  ( y_zpib5(2) *  picnia(-1) +  y_zpib5(3) *  picnia(-2) +  y_zpib5(4) *  picnia(-3)
         +  ( y_zpib5(6) *  rffe(-1) +  y_zpib5(7) *  rffe(-2) +  y_zpib5(8) *  rffe(-3) +  y_
         + y_zpib5(10) * rtr(-1) _
         + y_zpib5(11) * ptr(-1) _
         +  ( y_zpib5(12) *  xgap(-1) +  y_zpib5(13) *  xgap(-2) +  y_zpib5(14) *  xgap(-3) +
         +  ( y_zpib5(16) *  (400*d( log(pxb(-1)), 0, 1 )) +  y_zpib5(17) *  (400*d( log(pxb(-


zpic30:  zpic30-zpic30_aerr = y_zpic30(1) _
         +  ( y_zpic30(2) *  picnia +  y_zpic30(3) *  picnia(-1) +  y_zpic30(4) *  picnia(-2)
         +  ( y_zpic30(6) *  rffe +  y_zpic30(7) *  rffe(-1) +  y_zpic30(8) *  rffe(-2) +  y_z
         + y_zpic30(10) * rtr _
         + y_zpic30(11) * ptr _
         +  ( y_zpic30(12) *  xgap +  y_zpic30(13) *  xgap(-1) +  y_zpic30(14) *  xgap(-2) +


zpic58:  zpic58-zpic58_aerr =  ( y_zpic58(1) *  picnia +  y_zpic58(2) *  picnia(-1) +  y_zpic58
                +  ( y_zpic58(5) *  rffe +  y_zpic58(6) *  rffe(-1) +  y_zpic58(7) *  rffe(-2
                + y_zpic58(9) * rtr _
                + y_zpic58(10) * ptr _
                +  ( y_zpic58(11) *  xgap +  y_zpic58(12) *  xgap(-1) +  y_zpic58(13) *  xgap


zpicxfe:   zpicxfe-zpicxfe_aerr =  ( y_zpicxfe(1) *  picxfe(-1) +  y_zpicxfe(2) *  picxfe(-2) +
              +  ( y_zpicxfe(5) *  pieci(-1) +  y_zpicxfe(6) *  pieci(-2) +  y_zpicxfe(7) *
              +  ( y_zpicxfe(9) *  rffe(-1) +  y_zpicxfe(10) *  rffe(-2) +  y_zpicxfe(11) *
              +  ( y_zpicxfe(13) *  xgap2(-1) +  y_zpicxfe(14) *  xgap2(-2) +  y_zpicxfe(15)
              + y_zpicxfe(17) * rtr(-1) _
```

```
                + y_zpicxfe(18) * ptr(-1) _
                + y_zpicxfe(19) * log(qpcnia(-1)/pcnia(-1)) _
                + y_zpicxfe(20) * log(qpl(-1)/pl(-1)) _
                + y_zpicxfe(21) * (hlprdt(-1) - 400*huqpct(-1)) _
                +  ( y_zpicxfe(22) * (lur(-1) - lurnat(-1)) +  y_zpicxfe(23) * (lur


zpieci:   zpieci-zpieci_aerr =  ( y_zpieci(1) *  picxfe(-1) +  y_zpieci(2) *  picxfe
                +  ( y_zpieci(5) *  pieci(-1) +  y_zpieci(6) *  pieci(-2) +  y_zpiec
                +  ( y_zpieci(9) *  rffe(-1) +  y_zpieci(10) *  rffe(-2) +  y_zpieci
                +  ( y_zpieci(13) *  xgap2(-1) +  y_zpieci(14) *  xgap2(-2) +  y_zpi
                + y_zpieci(17) * rtr(-1) _
                + y_zpieci(18) * ptr(-1) _
                + y_zpieci(19) * log(qpcnia(-1)/pcnia(-1)) _
                + y_zpieci(20) * log(qpl(-1)/pl(-1)) _
                + y_zpieci(21) * (hlprdt(-1) - 400*huqpct(-1)) _
                +  ( y_zpieci(22) * (lur(-1) - lurnat(-1)) +  y_zpieci(23) * (lur(-2


zrff10:  zrff10-zrff10_aerr = y_zrff10(1) _
                    +  ( y_zrff10(2) *  picnia +  y_zrff10(3) *  picnia(-1) +  y_zrf
                    +  ( y_zrff10(6) *  rffe +  y_zrff10(7) *  rffe(-1) +  y_zrff10(8
                    + y_zrff10(10) * rtr _
                    + y_zrff10(11) * ptr _
                    +  ( y_zrff10(12) *  xgap +  y_zrff10(13) *  xgap(-1) +  y_zrff10


zrff30:  zrff30-zrff30_aerr = y_zrff30(1) _
                    +  ( y_zrff30(2) *  picnia +  y_zrff30(3) *  picnia(-1) +  y_zrf
                    +  ( y_zrff30(6) *  rffe +  y_zrff30(7) *  rffe(-1) +  y_zrff30(8
                    + y_zrff30(10) * rtr _
                    + y_zrff30(11) * ptr _
                    +  ( y_zrff30(12) *  xgap +  y_zrff30(13) *  xgap(-1) +  y_zrff30


zrff5:  zrff5-zrff5_aerr = y_zrff5(1) _
                   +  ( y_zrff5(2) *  picnia +  y_zrff5(3) *  picnia(-1) +  y_zrff5(4)
                   +  ( y_zrff5(6) *  rffe +  y_zrff5(7) *  rffe(-1) +  y_zrff5(8) *
                   + y_zrff5(10) * rtr _
                   + y_zrff5(11) * ptr _
                   +  ( y_zrff5(12) *  xgap +  y_zrff5(13) *  xgap(-1) +  y_zrff5(14)


zvpd:  zvpd-zvpd_aerr = y_zvpd(1) _
                  +  ( y_zvpd(2) *  picnia(-1) +  y_zvpd(3) *  picnia(-2) +  y_zvpd(4)
                  +  ( y_zvpd(6) *  rffe(-1) +  y_zvpd(7) *  rffe(-2) +  y_zvpd(8) *  r
```

```
                + y_zvpd(10) * rtr(-1) _
                + y_zvpd(11) * ptr(-1) _
                +  ( y_zvpd(12) *  xgap(-1) +  y_zvpd(13) *  xgap(-2) +  y_zvpd(14) *  xgap(-3)
                +  ( y_zvpd(16) *  d( log(xbo(-1)), 0, 1 ) +  y_zvpd(17) *  d( log(xbo(-2)), 0,
                +  ( y_zvpd(20) *  d( log(vpd(-1)), 0, 1 ) +  y_zvpd(21) *  d( log(vpd(-2)), 0,
                + y_zvpd(24) * hgvpd(-1)


zvpi:  zvpi-zvpi_aerr =  ( y_zvpi(1) *  picnia(-1) +  y_zvpi(2) *  picnia(-2) +  y_zvpi(3) *  p
                +  ( y_zvpi(5) *  rffe(-1) +  y_zvpi(6) *  rffe(-2) +  y_zvpi(7) *  rffe(-3) +
                + y_zvpi(9) * rtr(-1) _
                + y_zvpi(10) * ptr(-1) _
                +  ( y_zvpi(11) *  xgap(-1) +  y_zvpi(12) *  xgap(-2) +  y_zvpi(13) *  xgap(-3)
                +  ( y_zvpi(15) *  d( log(xbo(-1)), 0, 1 ) +  y_zvpi(16) *  d( log(xbo(-2)), 0,
                +  ( y_zvpi(19) *  d( log(vpi(-1)), 0, 1 ) +  y_zvpi(20) *  d( log(vpi(-2)), 0,
                + y_zvpi(23) * hgvpi(-1)


zvps:  zvps-zvps_aerr =  ( y_zvps(1) *  picnia(-1) +  y_zvps(2) *  picnia(-2) +  y_zvps(3) *  p
                +  ( y_zvps(5) *  rffe(-1) +  y_zvps(6) *  rffe(-2) +  y_zvps(7) *  rffe(-3) +
                + y_zvps(9) * rtr(-1) _
                + y_zvps(10) * ptr(-1) _
                +  ( y_zvps(11) *  xgap(-1) +  y_zvps(12) *  xgap(-2) +  y_zvps(13) *  xgap(-3)
                +  ( y_zvps(15) *  d( log(xbo(-1)), 0, 1 ) +  y_zvps(16) *  d( log(xbo(-2)), 0,
                +  ( y_zvps(19) *  d( log(vps(-1)), 0, 1 ) +  y_zvps(20) *  d( log(vps(-2)), 0,
                + y_zvps(23) * hgvps(-1)


zxbd:  zxbd-zxbd_aerr = y_zxbd(1) _
              +  ( y_zxbd(2) *  picnia(-1) +  y_zxbd(3) *  picnia(-2) +  y_zxbd(4) *  picnia(-3
              +  ( y_zxbd(6) *  rffe(-1) +  y_zxbd(7) *  rffe(-2) +  y_zxbd(8) *  rffe(-3) +  y
              + y_zxbd(10) * rtr(-1) _
              + y_zxbd(11) * ptr(-1) _
              +  ( y_zxbd(12) *  xgap(-1) +  y_zxbd(13) *  xgap(-2) +  y_zxbd(14) *  xgap(-3) +
              +  ( y_zxbd(16) *  d( log(xbo(-1)), 0, 1 ) +  y_zxbd(17) *  d( log(xbo(-2)), 0, 1
              +  ( y_zxbd(20) *  d( log(vpd(-1)), 0, 1 ) +  y_zxbd(21) *  d( log(vpd(-2)), 0, 1
              + y_zxbd(24) * hgx(-1)/400


zxbi:  zxbi-zxbi_aerr =  _
                 ( y_zxbi(1) *  picnia(-1) +  y_zxbi(2) *  picnia(-2) +  y_zxbi(3) *  picnia(-3
              +  ( y_zxbi(5) *  rffe(-1) +  y_zxbi(6) *  rffe(-2) +  y_zxbi(7) *  rffe(-3) +  y
              + y_zxbi(9) * rtr(-1) _
              + y_zxbi(10) * ptr(-1) _
              +  ( y_zxbi(11) *  xgap(-1) +  y_zxbi(12) *  xgap(-2) +  y_zxbi(13) *  xgap(-3) +
              +  ( y_zxbi(15) *  d( log(xbo(-1)), 0, 1 ) +  y_zxbi(16) *  d( log(xbo(-2)), 0, 1
```

```
              +  ( y_zxbi(19) *  d( log(vpi(-1)), 0, 1 ) +  y_zxbi(20) *  d( log(vpi
              + y_zxbi(23) * hgx(-1)/400


zxbs:  zxbs-zxbs_aerr =  _
                  ( y_zxbs(1) *  picnia(-1) +  y_zxbs(2) *  picnia(-2) +  y_zxbs(3) *
              +  ( y_zxbs(5) *  rffe(-1) +  y_zxbs(6) *  rffe(-2) +  y_zxbs(7) *  rf
              + y_zxbs(9) * rtr(-1) _
              + y_zxbs(10) * ptr(-1) _
              +  ( y_zxbs(11) *  xgap(-1) +  y_zxbs(12) *  xgap(-2) +  y_zxbs(13) *
              +  ( y_zxbs(15) *  d( log(xbo(-1)), 0, 1 ) +  y_zxbs(16) *  d( log(xbo
              +  ( y_zxbs(19) *  d( log(vps(-1)), 0, 1 ) +  y_zxbs(20) *  d( log(vps
              + y_zxbs(23) * hgx(-1)/400


zyh: log(zyh) - zyh_aerr =  ( y_zyh(1) *  picnia +  y_zyh(2) *  picnia(-1) +  y_zyh(3
                    +  ( y_zyh(5) *  rffe +  y_zyh(6) *  rffe(-1) +  y_zyh(7) *  rffe
                    +  ( y_zyh(9) *  xgap2 +  y_zyh(10) *  xgap2(-1) +  y_zyh(11) *
                    + y_zyh(13) * ptr _
                    + y_zyh(14) * rtr _
                    +  ( y_zyh(15) *  yhgap +  y_zyh(16) *  yhgap(-1) +  y_zyh(17) *
                    + log(zyhst*xgdpt)


zyhp: log(zyhp) - zyhp_aerr =  ( y_zyhp(1) *  picnia +  y_zyhp(2) *  picnia(-1) +  y_
                      +  ( y_zyhp(5) *  rffe +  y_zyhp(6) *  rffe(-1) +  y_zyhp(7) *
                      +  ( y_zyhp(9) *  xgap2 +  y_zyhp(10) *  xgap2(-1) +  y_zyhp(11
                      + y_zyhp(13) * ptr _
                      + y_zyhp(14) * rtr _
                      +  ( y_zyhp(15) *  yhgap +  y_zyhp(16) *  yhgap(-1) +  y_zyhp(1
                      +  ( y_zyhp(19) *  yhpgap +  y_zyhp(20) *  yhpgap(-1) +  y_zyhp
                      + log(zyhpst*zyhst*xgdpt)


zyhpst: zyhpst-zyhpst_aerr = zyhpst(-1) + y_zyhpst(1)*(yhpshr-zyhpst(-1))


zyhst: zyhst-zyhst_aerr = zyhst(-1) + y_zyhst(1)*(yhshr-zyhst(-1))


zyht: log(zyht) - zyht_aerr  =  ( y_zyht(1) *  picnia +  y_zyht(2) *  picnia(-1) +  y
                       +  ( y_zyht(5) *  rffe +  y_zyht(6) *  rffe(-1) +  y_zyht(7) *
                       +  ( y_zyht(9) *  xgap2 +  y_zyht(10) *  xgap2(-1) +  y_zyht(1
                       + y_zyht(13) * ptr _
                       + y_zyht(14) * rtr _
                       +  ( y_zyht(15) *  yhgap +  y_zyht(16) *  yhgap(-1) +  y_zyht(
```

```
                        +  ( y_zyht(19) *  yhtgap +  y_zyht(20) *  yhtgap(-1) +  y_zyht(21) *  y
                        + log(zyhtst*zyhst*xgdpt)


  zyhtst: zyhtst - zyhtst_aerr = zyhtst(-1) + y_zyhtst(1)*(yhtshr-zyhtst(-1))


  zynid: zynid - zynid_aerr = y_zynid(1) _
                      +  ( y_zynid(2) *  picnia(-1) +  y_zynid(3) *  picnia(-2) +  y_zynid(4) *  p
                      +  ( y_zynid(6) *  rffe(-1) +  y_zynid(7) *  rffe(-2) +  y_zynid(8) *  rffe(
                      + y_zynid(10) * rtr(-1) _
                      + y_zynid(11) * ptr(-1) _
                      +  ( y_zynid(12) *  xgap(-1) +  y_zynid(13) *  xgap(-2) +  y_zynid(14) *  xg
                      +  ( y_zynid(16) *  d( log(qynidn(-1)/pxb(-1)), 0, 1 ) +  y_zynid(17) *  d(
                      + y_zynid(20) * (hggdpt(-1)/400)


  theend
```

This code is written to file `stdver.eqs.txt`.

A.3 Standard Version Coefficients File

72 ⟨*stdver.coeffs.txt* 72⟩≡

```
y_ceng  6       -0.1483451935619194,0.475653118183134,0.5437644321944857,-0.23015987
y_dmptlur       1       25
y_dmptpi        1       -25
y_dpadj 1       1.0000
y_ecd   4       0.1553557918476032,-0.05860156240430123,1,9.039065475739223
y_ech   4       0.002890569762594884,-0.02415873224871467,0.5006794105950545,0.001736
y_eco   4       0.1088704831212408,0.4609714707829828,1,0.252176379778204
y_egfi  7       -0.001620944144695763,-0.1243761665741676,-0.1946254304372423,-0.1028
y_egfit 3       -.4027,-.1,1.0
y_egfl  7       -6.057249900438316e-05,-0.06931736294593471,0.3048866347485139,-0.049
y_egflt 3       -.375978,-.1,1.0
y_egfo  7       -0.00272437480660757,-0.165188738562342,-0.2655033775214354,-0.138133
y_egfot 3       -.342813,-.1,1.0
y_egsi  7       -1.405740361028989e-05,-0.2020609033108234,0.05134522874864941,-0.080
y_egsit 3       -.379944,-.1,1.0
y_egsl  7       0.000432632357275569,-0.1411968485071547,0.173955823870621,0.03758904
y_egslt 3       -.259779,-.1,1.0
y_egso  7       -0.0002007505801469657,-0.09372198933526569,0.5475507872556951,0.1640
y_egsot 3       -.382643,-.1,1.0
y_eh    6       0.01184830003855771,0.3575993755366778,0.2161402157869259,1,-0.051357
y_emo   6       0.01701497186817749,-0.1984753225812535,1.352328263830308,1.67397668
y_empt  2       0.10000000000000000E+00,1.00000000000000000E+00
y_epd   4       0.1639648722427122,0.4446158979500308,0.3699597791648127,0.5
y_epi   4       0.01211724517486588,0.6819035622357826,0.1766782129232528,0.21229452
y_eps   5       0.06660965676110558,0.5425646472109228,0.3261733908091358,0.5,-0.0969
y_ex    5       0.8118629319610274,-0.1074807087618527,1.38575824141273,1.09285611828
y_fgdpt 3       -.458264,-.1,1.0
y_fpi10 5       0.7045829169372979,0.2954170830627021,0.2531839520282475,5.3242127898
y_fpi10t        2       9.500000000000000000e-01,5.000000000000000000e-02
y_fpic  3       2.174669585864584,0.6994194241702426,0.3005805758297574
y_fpxr  2       0.048,0.5
y_fpxrr 2       0.03011994048459088,0.2026244928161041
y_frl10 5       0.03993364460261257,-0.07293669623744157,0.08403561227292196,0.363792
y_frs10 5       0.0,1.0,1.0,0.5,1.0
y_frstar        2       .95,.05
y_fxgap 4       1.284002584226955,-0.4544105287732581,-0.05,0.02742233318740996
y_gfs   3       -.361185,-.1,1.0
y_gfsub 3       -.550087,-.1,1.0
y_gftrd 3       -3.598159243340642e-05,0.6589196196672864,-0.0002408286743628969
y_gstrd 3       -1.235658095172135e-05,0.7366990097980338,-4.483509762335216e-05
y_hgemp 2       .9,.1
y_hgpdr 2       .9,.1
```

```
y_hgpir 2       .9,.1
y_hgpkir        2       .9,.1
y_hgppsr        2       .9,.1
y_hgvpd 2       0.97,0.03
y_hgvpi 2       0.97,0.03
y_hgvps 2       0.97,0.03
y_hmfpt 2       0.055,0.95
y_hqlfpr        2       0.00,0.95
y_hqlww 2       .95,-0.3129029344874886
y_huqpct        2       0.00,0.95
y_huxb  2       -0.01817091647656927,0.95
y_ki    5       0.01679108530917215,0.451650730999944,0.2617948535758293,0.2865544154242267,-0.
y_left  2       0.90000000000000000E+00,0.10000000000000000E+00
y_leo   2       0.6995814979956745,-0.01620869768699893
y_lest  2       0.90000000000000000E+00,0.10000000000000000E+00
y_lfpr  2       0.5580285205989896,-0.0008755566736369085
y_lhp   5       0.255040531063274,0.1491232069118806,0.3902648422452434,0.6097351577547565,-0.0
y_lww   2       0.1984470411422383,0.3128887644653584
y_mei   1       0.86
y_mep   1       0.86
y_mfpt  1       0.0
y_pcdr  2       -0.003205436686618677,0.5065758198036935
y_pcengr        5       0.04621048926220116,-0.01053548206463643,-0.09617350148754544,0.0821372
y_pcer  4       0.1050137345817281,0.5632388610140522,0.6858569548199248,0.04030768373454912
y_pcfr  6       -0.1757649679968763,-7.899990101672884e-05,0.3777936884215714,0.023492922129461
y_pchr  2       0.0005315862255843622,0.5948038682986249
y_pgfir 1       0.0
y_pgfor 1       0.0
y_pgsir 1       0.0
y_pgsor 1       0.0
y_phouse        3       0.004817103239693556,0.8898461413782496,-0.01120829645070205
y_phr   1       0.0
y_picxfe        3       0.644974342322,0.00373609153735,0.98
y_pieci 4       0.811777544324,-0.0148780773818,0.00186804576867,0.98
y_pipxnc        3       .462801,.229745,-.284477
y_pmo   4       -0.003166815111887241,0.4492916534287926,0.2944651755345454,0.7055348244654546
y_poilr 4       -0.2386347615324657,-0.003817963307816998,0.3988973185364578,0.2246596594065311
y_ppdr  1       0.0
y_ppsr  1       0.0
y_ptr   3       0.9,0.05,0.05
y_pwstar        2       0.00,1.00
y_pxr   1       0.0
y_qec   5       0.7592609842874721,0.002578773939057793,0.2407390157125279,-0.2514158240890368,
y_qecd  3       -0.6165972226120303,2.557266037164673,-0.6165972226120303
y_qeco  1       -0.3372292498223053
y_qeh   2       1.935026993649364,-0.1570195518635583
```

```
y_qepd  4       0,1.000000000000000e+00,1.000000000000000e+00,1.00000000000000
y_qepi  4       0,1.000000000000000e+00,1.000000000000000e+00,1.00000000000000
y_qeps  4       0,1.000000000000000e+00,1.000000000000000e+00,1.00000000000000
y_qkir  1       -0.001885366737710053
y_qpl   1       1.0
y_qpmo  1       -.003347
y_qpxg  2       0.0,1
y_qpxnc 2       2.98507462687,-1.98507462687
y_qynidn        3       -0.9889159016018153,0.3614481909275686,1
y_rbbbp 5       1.663544231588651,-0.1493888609930089,0.8866986585299741,1.663544231
y_rcar  6       2.100170296931854,-1.167642954704071,-0.008386800063101975,0.693768
y_rcgain        5       0.1522590051966577,0.2987109747902424,0.2513416212164487,0.15
y_reqp  5       2.882980324228344,0.6395674906531285,0.8185047577678474,2.882980324
y_rffalt        6       .0551,1.2,-.39,.6954,-.5168,.3287
y_rffgen        50      0.000000000000000000e+00,0.000000000000000000e+00,0.00000000
y_rffintay      3       0.5,1.0,.85
y_rfftay        2       0.5,1.0
y_rfftlr        3       -0.5,.375,1.1
y_rfynic        4       0.2599432734430575,-0.1468767116652314,0.1482396937168886,0.6
y_rfynil        9       0.1878356791714486,-0.2435367622231839,0.07902780819914431,0.
y_rg10p 4       0.9985065593208419,-0.4718548432007495,0.7314217770878953,0.895933637
y_rg30p 4       1.337544689343979,-0.5892843861420656,0.8365523842356651,0.904558899
y_rg5p  3       0.7478923780795074,-0.3984697511015516,0.9119509672669279
y_rgfint        2       0.86,0.005417428040208504
y_rgw   4       .00495,.00271,.00129,.00105
y_rme   5       0.4927100798849811,0.6776016328060693,0.2424386344238626,0.230503798
y_rrmet 2       .9048,.0952
y_rrtr  2       .97,.03
y_rstar 1       .05
y_rtbe  5       -0.06677368009690213,0.7720707564737897,0.1224099968713681,0.785095
y_trfci 6       0.00133892767133083,0.8130157141532537,0.1085501838146501,-0.219188
y_trfp  4       1,0.6249369098272274,0.2896464773374296,0.0003722869429144596
y_trfpt 3       0.05000000000000000E+00,0.50000000000000000E+00,-0.10000000000000000
y_trsci 6       0.791150698521011,0.9058859419794156,-0.6970366405004266,-0.000768122
y_trsib 4       0.9134383490112551,1.33647889726315,-1.249917246274406,-3.353806684
y_trsp  5       0.632946369509944,0.882450152119161,-0.515396521629105,2.41466405329
y_trspt 3       0.05000000000000000E+00,0.50000000000000000E+00,-0.25000000000000000E+
y_trssi 5       1.18174981903228,-0.2318024453193926,1.575674530080275,-1.525621903
y_uces  8       -0.1834529206587357,0.1554187181683198,0.08000391518229149,-0.000441
y_ucfs  7       -0.03523462021069426,0.0453107908363,-0.0001497160154925362,-0.05646
y_uqpct 1       0.0
y_uxbt  1       0.0
y_veoa  2       0.988,0.012
y_wdnfcn        5       -0.02207644135378071,0.01442097831747879,0.2375257265379373,
y_xbo   1       1.338129148984226
y_xgo   1       1.313096
```

```
y_xgpot 4       .7000,.265,.035,.035
y_yhibn 8       1,-0.1336307554530098,0.06545518537060361,0.2942182559897778,0.023569042697273
y_yhpcd 1       0.053750000000000000E+00
y_ynidn 3       0.0903554997290158,-0.1364018197288298,1
y_yniin 6       0.01335460515030035,0.8715712577633621,0.03107757397810296,0.1284287422366379,0
y_zdivgr        20      1.511071172206618e-15,-0.009111480239164081,0.03183741780107196,0.02833
y_zecd  32      -0.0005835440697737298,-0.0004890487384829661,-0.0003178601486946526,-1.1527354
y_zeco  31      -8.302302840394758e-05,-8.481341005195437e-05,-1.070919356458063e-05,9.3814944
y_zeh   31      -0.0001475636416872941,-3.032365273125124e-05,-4.473855969321594e-06,1.8401597
y_zgap05        15      2.257007909357927e-15,-0.1597149595303493,-0.02714596421531133,-0.0564
y_zgap10        15      1.913550184020851e-15,-0.08856716084344839,-0.01514793353340919,-0.0306
y_zgap30        15      9.185040883300084e-15,-0.04699887854311754,-0.008064404203305675,-0.016
y_zgapc2        14      -0.01642348362157579,-0.003669559326500591,-0.008031103190068016,0.0041
y_zlhp  16      -0.0002522439372141123,-5.098270125007645e-05,-0.0002552621374828649,3.9816074
y_zpi10 14      0.03879756717884661,0.01310655690781879,0.01249073978840772,0.0020136444470026
y_zpi5  14      0.06758353158403318,0.02161485431596137,0.01782456814136856,0.0029645246982114
y_zpib5 19      2.014761562942157e-14,0.08381220448829916,0.03966837250165698,0.029682684899265
y_zpic30        15      9.998348776898279e-14,0.03772442939281018,0.00691792724638696,0.006611
y_zpic58        14      0.3419924857225884,0.05029077146057983,0.04280461383060537,-0.017196351
y_zpicxfe       23      0.323685055125,-0.00320254773354,0.000957688783119,0.0104690425827,0.07
y_zpieci        23      -0.0173696976108,-0.00564002523431,0.000750046022225,0.0186445167159,0.
y_zrff10        15      -1.225928191740291e-13,-0.02771619956382117,-0.01188080871189547,-0.032
y_zrff30        15      -6.431098710768743e-14,-0.01469452480129645,-0.006366611548946281,-0.01
y_zrff5 15      -2.893994419845934e-13,-0.03329615692337154,-0.01651377444295286,-0.0523236718
y_zvpd  24      -3.503545878896081e-16,-0.0002563318120287816,-0.0003053817493858787,0.00027546
y_zvpi  23      3.869791235963136e-05,3.80256114092935e-06,2.612181181174604e-05,2.05719790994
y_zvps  23      0.0007852314783589356,-0.0001325502846474631,0.0001060837434901873,0.0001039260
y_zxbd  24      -2.515799209424174e-16,-0.0001835522663957102,-9.20694428089123e-05,-0.0001690
y_zxbi  23      -3.907288119414607e-05,-1.536565753314579e-05,-1.048653204032815e-05,1.1110647
y_zxbs  23      -0.0001994456999380124,-7.214041996312615e-05,-7.99329702758048e-05,2.21136030
y_zyh   18      -0.0002301813961999326,0.0007292083597749006,0.0006217471253906824,0.000593471
y_zyhp  22      0.0003844677024979630,0.001205361597423436,0.0009620980096161766,0.000696883450
y_zyhpst        1       0.050000000000000000E+00
y_zyhst 1       0.050000000000000000E+00
y_zyht  22      -0.0005375756842287296,0.0004256398977551294,0.000429593178783961,0.0003414271
y_zyhtst        1       0.050000000000000000E+00
y_zynid 20      -5.177745029596233e-16,3.507527558415562e-05,0.0004354171509883335,0.0003765833
theend
```

This code is written to file `stdver.coeffs.txt`.

Appendix B

Notes, Bibliography and Indexes

B.1 Chunks

⟨*stdver.coeffs.txt* 72⟩
⟨*stdver.eqs.txt* 30⟩
⟨*stdver.varinfo* 9⟩

B.2 Index