```
*Modified version of simszha.src .
*
* This is a fiive variable SVAR.
*
OPEN DATA "D:\sampledata.txt"
CALENDAR(Q) 1997:4
DATA(FORMAT=PRN,ORG=COLUMNS,DATEFORM="d-m-y") 1997:04 2009:01 dgdp def_inf dm3 rate
gfce
***
compute nlags=2
compute nsteps=25
*
*
system(model=varmodel)
variables gfce dgdp def_inf dm3 rate
lags 1 to nlags
det constant
end(system)
*
estimate(noprint)
*
* Save for MC integration
*
compute fxx    =%decomp(%xx)
compute fwish  =%decomp(inv(%nobs*%sigma))
compute betaols=%modelgetcoeffs(varmodel)
compute wishdof=%nobs-%rows(fxx)
dec rect[series] impulses
dec vect[rect]   %%responses
*
compute ncoef=%nreg
```

```
compute nvar =%nvar
*
compute nfree=14
dec vect phi(nfree)
nonlin(parmset=svar) phi
*
dec frml[rect] afrml bfrml
frml afrml = ||1.0 ,0.0,phi(1) ,0.0 ,0.0|$
phi(2) ,1.0 ,0.0 ,phi(3) ,0.0|$
phi(4) ,phi(5) ,1.0 ,phi(6),0.0|$
0.0 ,phi(7),phi(8) ,1.0 ,0.0|$
0.0 ,phi(9),phi(10),phi(11),1.0||
frml bfrml = ||1.0 ,0.0,phi(12) ,0.0 ,0.0|$
0.0 ,1.0 ,0.0 ,phi(13) ,0.0|$
0.0 ,0.0 ,1.0 ,phi(14),0.0|$
0.0 ,0.0,0.0 ,1.0 ,0.0|$
0.0 ,0.0,0.0,0.0,1.0||
compute phi=%const(0.0)
*
* Compute the maximum of the log of the marginal posterior density for
* the A coefficients with a prior of the form |D|**(-delta). Delta
* should be at least (nvar+1)/2.0 to ensure an integrable posterior
* distribution.
*
compute delta=3
cvmodel(a=afrml,b=bfrml,parmset=svar,dfc=ncoef,pdf=delta,dmatrix=marginalized,$
   method=bfgs,factor=mlfactor) %sigma
compute nfree=%nreg
*
* Compute the impulse responses at the ML estimates of everything
*
```

```
compute shocklabels=||"y1","y2","y3","y4","y5"||
compute varlabels=||"y1","y2","y3","y4","y5"||
impulses(model=varmodel,steps=nsteps,results=baseirfs,factor=mlfactor,noprint)
*
* Start the chain at the maximizer
*
compute [vector] thetadraw=%parmspeek(svar)
compute logplast=%funcval
*
* This is the covariance matrix for the RW increment. We might have to
* change the scaling on this to achieve a reasonable acceptance
* probability.
*
compute f=%decomp(%xx)*.35
compute nudraw=5
*
compute nburn =10000
compute ndraws=25000
*
declare vect lambdadraw(nvar) vdiag(nvar)
*
dim %%responses(ndraws) impulses(nvar,nvar)
*
compute accept=0
infobox(action=define,progress,lower=-nburn,upper=ndraws) "AB type SVAR"
do draw=-nburn,ndraws
  *
  * Draw a new theta based at previous value
  *
  compute theta=thetadraw+%ranmvt(f,nudraw)
  *
```

```
  * Evaluate the model there
  *
  compute %parmspoke(svar,theta)

cvmodel(parmset=svar,dfc=ncoef,pdf=delta,dmatrix=marginalized,method=evaluate,a=afrml,b=bfrm
l) %sigma
  compute logptest=%funcval
  *
  * Compute the acceptance probability
  *
  compute alpha  =exp(logptest-logplast)
  if alpha>1.0.or.%uniform(0.0,1.0)<alpha
    compute thetadraw=theta,logplast=logptest,accept=accept+1
  *
  infobox(current=draw) %strval(100.0*accept/(draw+nburn+1),"##.#")
  if draw<=0
    next
  *
  * Conditioned on theta, make a draw for lambda. (We don't need to do
  * this until we're saving results, since theta is being drawn
  * unconditionally).
  *
  compute %parmspoke(svar,thetadraw)
  compute a=afrml(1)
  *********
  *********The following line has been added by me.
  *********
  compute b=bfrml(1)
  ********
  ********The "tr(inv(b)*a)" was inserted in the following line of code as per my understanding$
  ********from Amisano and Giannini (1997).
  ********Sims and Zha specified it as "tr(a)".
```

```
   ********
   compute vdiag  =%mqformdiag(%sigma,tr(inv(b)*a))
   ewise lambdadraw(i)=(%nobs/2.0)*vdiag(i)/%rangamma(.5*(%nobs-ncoef)+delta+1)
   ******
   ******Draw coefficients conditioned on the factor generated by theta and
   ******lambda. Again, we don't need to do this during the burn-in.
   ******"inv(inv(b)*a)" was provided by me instead of "inv(a)" in Simszha.
   ******
   compute fsigmad=inv(inv(b)*a)*%diag(%sqrt(lambdadraw))
   compute betau   =%ranmvkron(fsigmad,fxx)
   compute betadraw=betaols+betau
   *
   * Compute impulse responses given the factor and the new coefficients
   *
   compute %modelsetcoeffs(varmodel,betadraw)
   impulse(noprint,model=varmodel,factor=fsigmad,results=impulses,steps=nsteps)
   *
   * Save them
   *
   local vect ix
   dim %%responses(draw)(nvar*nvar,nsteps)
   ewise %%responses(draw)(i,j)=ix=%vec(%xt(impulses,j)),ix(i)
   infobox(current=draw)
end do draw
infobox(action=remove)
*
@mcgraphirf(model=varmodel,$
 shocklabels=shocklabels,varlabels=varlabels,$
 percent=||.025,.16,.84,.975||,center=input,impulses=baseirfs,$
 footer="Pointwise 68% and 95% Posterior Bands, Five Variable Model")
```