United States

Department of the Interior

Geological Survey

---

# Computer Program for a Generic Western Coal Region Simulation Model Developed to Investigate Potential Applications of System Dynamics Modeling to the EIS Process

By Robert K. Mark, David S. Harwood, Richard R. Doell, and Evelyn B. Newman

---

Open File Report 78-321

This report is preliminary and has not been edited or reviewed for conformity with Geological Survey standards and nomenclature.

Menlo Park, California
1978

This generic western coal region model (WCR) was prepared to investigate possible applications of system dynamics simulation modeling (Forrester, 1961; 1968) to the regional EIS process. It is intended as a tool for research and perhaps a starting point for further work. It is not intended for actual application in its present form.

The model is driven by user selected coal development scenarios and impacts are simulated in the mining, land, water, demographic, political, and environmental sectors. Interaction are based upon hypothesized causal relations. Provision is made for system delays and feedback effects among the sectors. All variables (such as electric generation capacity, industrial water use, number of miners, sulfur dioxide emission) are summed or averaged over the region. The model has a time horizon of 30 years (1970-2000).

Members of the Northern Powder River Basin EIS Task Force reviewed this model during our research effort. We believe that the task force benefiteb from exposure to this system dynamics model perspective, but the preliminary status of the modeling effort precluded formal application of it to this EIS. The EIS task force was not able to commit extensive staff time to this experimental effort. The time required for refinement and adjustment of a model for the Northern Powder River region extended past the time by which the task force was required to complete its assembly of analytical tools and information and proceed with its analysis and writing of the statement. Prospects for future use of system dynamics modeling in support of EIS preparation by the Geological Survey are uncertain. The collaborative effort between the modeling research team and those concerned with EIS preparation lasted one and a half years. No further exploratory efforts are active at this time, although they may be reinstituted after the professional community has had the opportunity to evaluate and critique the modeling effort.

The five county region of southeastern Montana (Bighorn, Custer, Powder River, Rosebud and Treasure Counties) was used as a source for much of the data. In many cases where data was not readily available, other sources or estimates were used. In addition, where specific information was not available we chose to hypothesize the nature of some causal relations rather than omit them from this experimental model.

The model source code was written for the Honeywell Multics system. The main program and most of the subroutines are in "old fortran"; one (wcrcat) is in PL/1. The overall structure of the model is shown in the figure on the following page.

References Cited

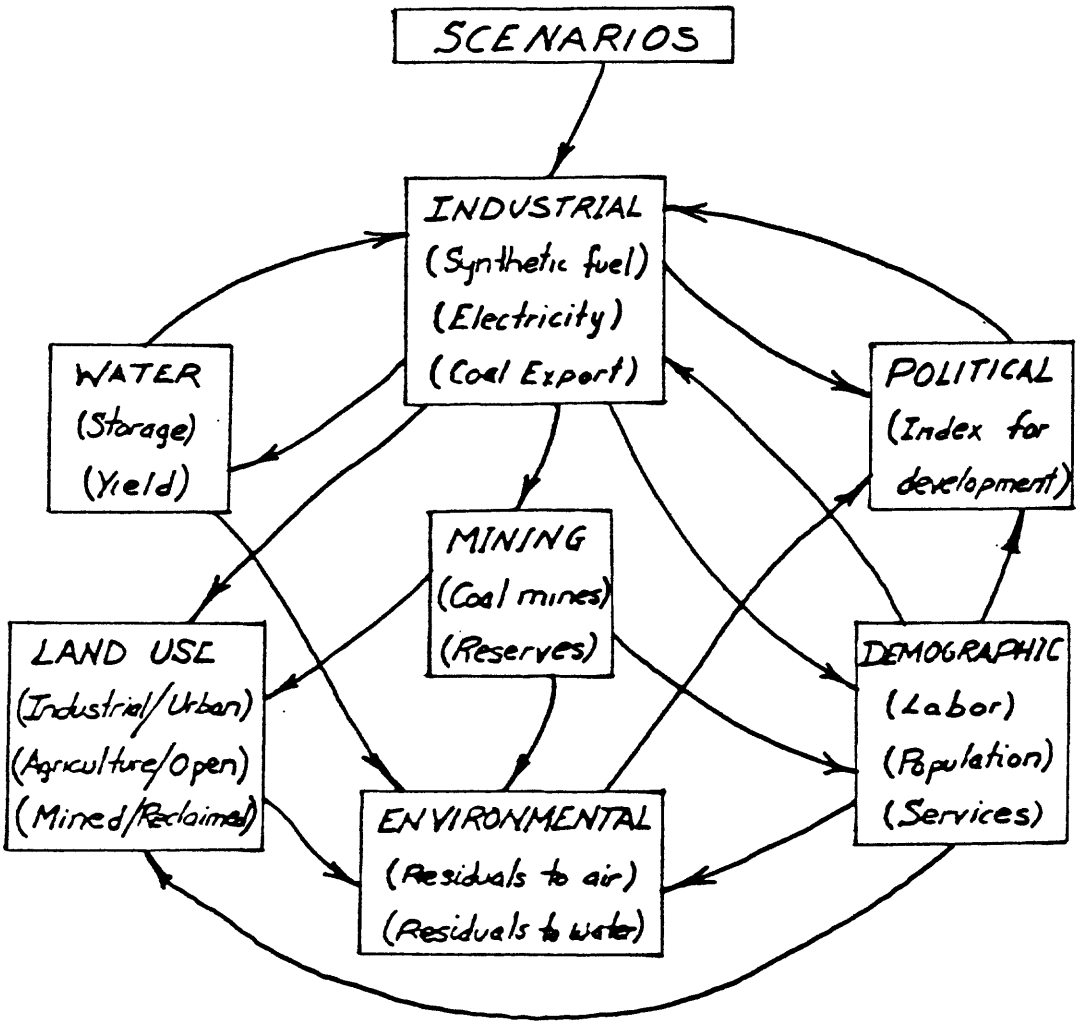
Ford, Andrew, 1976, User's guide to the BOOM1 model: Los Alamos Scientific Lab. Informal Report LA-6396-MS, 54 p.

Forrester, J. W., 1961, Industrial Dynamics: Cambridge, Mass., Massachusetts Inst. Technology Press, 464 p.

_____1968, Principles of Systems: Cambridge, Mass., Wright-Allen Press, 392 p.

Gilmore, J. S., and Duff, M. K., 1975, Boom town growth management: a case study of Rock Springs - Green River, Wyoming: Westview Press, 177 p.

Riggs, H. C., and Hardison, C. H., 1973, Storage analyses for water supply: Chap B2, Book 4 of Techniques of water-resources investigations of the United States Geological Survey, 20 p.



```
c       ************ Western Coal Region (08-26-77)************
c
c   This is the  "main" program for the western coal  regions  model.
c   Its major functions are to call the various sector subroutines as
c   needed, store the  proper calculation  for  the  output,  and  to
c   construct  the  output tables and graphs.  It also includes a few
c   data  statements,  but  like  all other constants, these  can  be
c   altered by the set statements.
c
c   A  detailed understanding of this master program is not needed in
c   order to run the model.
c
c To run the model:  Type  "ec wcr" (terminate all lines with
c a "return")
c program wcrcrd (subroutine output) prompts "n:".
c Type "0" to set new plot variables
c       "9" to retain plot variables from previous run (if any)
c  or "99" to stop
c
c  wcrcrd (subroutine wcrcat) then prompts "type show, set, or run"
c "show" is used to examine constants and is terminated by ";"
c for example:
c         type show, set, or run:  show
c              show:    c(10)
c         c(10) = 5.0
c              show:    ;
c         type show, set, or run:
c
c "set" is used to change constants.  Changes remain in effect
c until reset or the run is terminated.  "set" is also
c terminated by ";" .
c for example:
c           type show, set, or run:  set
c                set: c(10)=7.0,tc(6)=3.5,switch(99)=1;
c           type show, set, or run:
c note, switch(99)=1 turns on the table printout
c
c "run" is used to begin execution
c  .
c *************** start program statements ************************
      common time
      common/dt/dt
      common /t/ tnam(35),tmxx(35),ttim(35),tcnt(35)
      common/graph/put,max,min,flo,fup,opsym,ii,nput
      common/system/v(1000),c(1000),tc(1000),t1
      common/energy/ctime
      common/switch/switch(100)
      integer switch
      integer indgo, lndgo, wtrgo, demgo, polgo, envgo
      integer opsym(10),ii(10)
      real flo(10),fup(10)
      real put(10),max(10),min(10)
```

```
c model termination date, iteration time interval, and
c time between ordering decisions.
      tstop=2000.0
      dt=0.1
      t1=1.0
c time after which development scenario is held constant
      ctime=0.0
c data needed for plot routine
      nprint=10
      nput=6
c default value for coal development scenario being simulated
      ncdp=6
c
c -------------------------------------------------------
c switch array  (0 is off, 1 is on)
c
      do 5 i=1,100
      switch(i)=0
 5    continue
c
c subroutine switches (0 is off, 1 is on)
c
      indgo=1
      wtrgo=1
      lndgo=1
      demgo=1
      polgo=1
      envgo=1
      mingo=1
c
      data mingo/1/
c output arrays
c
      do 6 i=1,10
      flo(i)=0.
      fup(i)=0.
 6    continue
c
c set output symbols
c
      data opsym/0,'1','2','3','4','5',4*' '/
c -------------------------------------------------------
c initialize constants
c
      call wcrco
c
10    continue
c prompts for 5 variables to be plotted
      call output(ii)
c
c initialize dictionaries
      call indic
      do 305 i=1,35
```

```
305    tnam(i)=0.0
       do 315 i=1,10
       max(i)=-1.0e10
       min(i)=+1.0e10
315    continue
c
c initialize variable array
c
       do 101 i=1,1000
       v(i)=0.0
101    continue
c initialize levels
c
       call wcrin
       time=1970.0
c
c following statement permits the use of set statements to
c alter constants, initial values, default values, etc.
c for any given simulation. (see instructions for making
c simulations).  Subroutine wcrcat is written in pl1.
c
       call wcrcat (v,c,tc,switch,ncdp,flo,fup,tstop,ctime)
c
c tpy is an abbreviation for tons per year
c initial synthetic fuel capacity   million tpy
       v(14)=energy(3,ncdp,1970.0)/c(3)
c initial electrical generation nominal capacity   tpy
       v(55)=energy(1,ncdp,1970.)/c(38)
c initial coal export capacity  million tpy
       v(80)=energy(2,ncdp,1970.0)
c initial coal mine capacity   million tpy
       v(825)=totcol(ncdp,1970.0)
c -----------------------------------------------------
       krate=1
       go to 500
c
20     continue
       kput = 1
       go to 600
c
40     continue
       if(time.ge.tstop) goto 95
       krate = 2
       kput = 2
50     continue
       kpr=0
55     time=time+dt
       go to 500
c
70     continue
       kpr=kpr+1
       if(kpr .le. nprint-1) goto 55
       go to 600
```

```
c
90     continue
       if(time .lt. tstop-dt) go to 50
c ----------------------------------------------------------
95     continue
       if (switch(99).eq.1) write(21,98)
98     format(' 014')
       rewind 20
c
       call graph
c
       go to 10
c ------------------------------------------------------------------
c preliminary output subroutine
c
600    continue
c
c put(*)=output var
c
       put(1)=time
       do 700 i=1,nput-1
       put(i+1)=v(ii(i))
700    continue
       do 620 i=2,nput
       if(max(i).lt.put(i)) max(i)=put(i)
       if(min(i).gt.put(i)) min(i)=put(i)
620    continue
       write(20) (put(i),i=1,nput)
c
c for output table, set switch(99)=1
       if(switch(99).eq.1)write(6,99)(put(i),i=1,nput)
       if(switch(99).eq.1)write(21,99)(put(i),i=1,nput)
99     format(1h ,6f10.2)
c
       go to (40,90), kput
c
c --------------------------------------------------------------
c
500    continue
c
c sector subroutines
c
       call wcrind(ncdp,indgo)
       call wcrmin(ncdp,mingo)
       call wcrwtr(wtrgo)
       call wcrdem(demgo)
       call wcrlnd(lndgo)
       call wcrenv(envgo)
       call wcrpol(polgo,ncdp)
c
       go to (20,70), krate
       end
```

```
c     *************** Industry Sector (08-26-77) ***************
c
c This sector simulates the growth of the coal  industries  in  the
c region,  so as to provide data for impacts of these industries in
c other model sectors, and to accept feedback  which may  influence
c the  growth  of the industries.  The sector is divided into three
c subsectors:  synthetic fuel production,  electrical  generation,
c and  coal  export.  There are three principle exogenous inputs to
c the sector: these are the assumed scenarios for synthetic  fuels,
c electricity,  and  for coal export capacity in the region.  These
c exogenous variables are the primary "driving forces" of the model
c and  thus  determine  the  relative  magnitude  of   development,
c impacts, etc.
c
c The decisive process for each of the subsectors is  the  ordering
c and construction of new  facilities.   It  is  assumed  that  the
c scenarios    represent     economically     and/or    politically
c "desirable"  developments which  utility  management will try  to
c attain.   However,  the  desired rates of accuiring  new capacity
c may   be   modified   by   several   other   factors,   such   as
c "political climate" for industrial growth, labor shortages, water
c shortages, etc.
c
c The  basic  unit  of  energy is one million tons of coal (at 9000
c btu/lb).  In addition, for the  electrical  generation  subsector
c the model also calculates the  number  of "standard 1000 megawatt
c electrical generation (mwe) plants", and  for the  synthetic fuel
c subsector the number of "standard 250 million standard cubic feet
c per day (scfpd) plants".

c The  following  text  generally  describes all three  subsectors.
c The program begins in  each  subsector  by  calculating  scenario
c planned capacities v(3),v(38),v(73) for each of the industries by
c calling  the  function  "energy"  which contains the data for the
c scenarios being modeled.  Note that these three calculations  are
c made  for  times tc(3), tc(38), and tc(73) into the future; these
c planning horizon time  constants  are  the  perceived  times  for
c construction  of  the  three  different types of facilities.  For
c synthetic fuels and electrical generation the scenarios  must  be
c divided  by  the  nominal  capacity  utilizations  of  these  two
c industries, c(3) and c(38), since the scenarios  are  for  energy
c output   and   the   model   uses  plant  capacity.   The   model
c assumes,   for   planning   purposes,   full   use   of   export
c capacity.  By setting switch (02) to a value of 1, the model will
c transfer  all  demand  for  energy to the export sector.
c
c The  next  actions  are  to determine construction, planning, and
c regulatory delays that may be influenced by labor  shortages  and
c political  climate  for  industrial  growth, and to set the delay
c variables  in  the  subsectors.   Following  this,  the   model
c determines  plant and capacity depreciation rates. In the case of
c electrical  generation  the  plants  are  derated   rather   than
c depreciated  since  older plants are generally used less and less
```

```
c for base load as they age.
c
c The variables which initiate ordering are the capacity initiation
c rates v(5),v(40),v(75).  They are  calculated  from  the  deficit
c between scenario planned capacities and what is expected to be on
c line at the planning horizon times (present capacities plus those
c in  planning and construction less those that will have also been
c depreciated).  These rates can also  be  influenced  by  possible
c coal  shortages  and  may  be  influenced  by water shortages, if
c desired (by setting switch (01) to 1).   Lastly,  the  subsectors
c calculate  future coal needs to be used for coal mine planning in
c the mine sector.
c
c applicable switches in the sector    (default value is 0 = off)
c
c       switch number               function
c          01               hypothetical water shortage feedback
c          02               total export option
c          40               political climate feedback
c          04               construction labor shortage feedback
c
c
c     **************** start program statements **************
c
      subroutine wcrind(ncdp,indgo)
      integer indgo
c
c common statements
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c
c
c switch statements
      common/switch/switch(100)
      integer switch
c if indgo=0, subroutine is not executed.  if indgo=1 (de-
c fault value), subroutine is executed for all iterations.
c if indgo=-n, subroutine is executed for n iterations only.
      if(indgo.eq.0) return
      if(indgo.gt.0) go to 10
      if(indgo.lt.0) indgo=indgo+1
10    continue
c
c coal mine planning delay plus regulatory delay   years
c (minimum planning time plus maximum regulatory delay
c times Fermi function of political climate index.
c Fermi has a value of 1/2 at an index of -0.1)
      pci=c(420)
      if(time.gt.1970.0) pci=v(420)
      v(819)=tc(819)+tc(818)/(1.0+exp((pci+0.1)/0.15))
      tc(807)=v(819)
c coal mine construction delay time  (nominal time divided
```

```
c by one less the fractional constructional labor
c shortage)  years
c
      v(826)=tc(826)
      if(v(232).lt.0.0) v(232)=0.0
      if(switch(04).eq.1) v(826)=tc(826)/(1.0-v(232))
c
c ------------------ synthetic fuel subsector -------------------
c
c tpy is an abbreviation for tons per year
c synthetic fuel scenario (output)   million tpy
      v(1)=energy(3,ncdp,time)
c syn fuel scenario planned capacity   million tpy   (tcsf is the
c planning horizon time, tc(3), shortened by the Fermi function
c for times near the model starting time)
      tcsf=tc(3)*(1.0-(1.0/(1.0+exp((time-1970.0)/1.7))))
      v(3)=c(2)*energy(3,ncdp,time+tcsf)/c(3)
c total export option
      if(switch(02).eq.1) v(3)=0
c synthetic fuel planning delay plus regulatory delay   years
c (see also analagous note for coal mine planning delay)
      v(22)=tc(22)+tc(21)/(1.0+exp((pci+0.1)/0.15))
      tc(6)=v(22)
c synthetic fuel construction delay time (nominal time divided
c by one less the fractional construction labor shorage)   years
c (the total delay is apportioned into the three delay macro
c functions in a manner to better model the construction
c process)
      v(26)=tc(26)
      if(switch(04).eq.1) v(26)=tc(26)/(1.0-v(232))
c synthetic fuel construction time constants   years
      tc(8)=0.1875*v(26)
      tc(10)=0.1875*v(26)
      tc(12)=0.6250*v(26)
c synthetic fuel water shortage multiplier (hypothetical)
c (modeled as the sine squared of the fractional water to industry)
      v(23)=1.0
      if(switch(01).eq.1) v(23)=(sin(v(143)*3.14159/2.0))**2.0
c initialize synthetic fuel capacity in planning and in construc-
c tion to steady state in 1970  (for syn fuel, nothing is assumed
c to be in planning or in construction in 1970)
      if(time.gt.1970.0) go to 20
      vtemp1=0.0
      v(6)=delay3('v5',vtemp1,tc(6),v(7))
      v(8)=delay3('v6',vtemp1,tc(8),v(9))
      v(10)=delay3('v8',vtemp1,tc(10),v(11))
      v(12)=delay3('v10',vtemp1,tc(12),v(13))
20    continue
c initialize coal shortage allocation function(assume no shortage in 1970)
      if(time.eq.1970.0) v(823)=1.0
c synthetic fuel plant average lifetime   years   (initialize as
c if all previous plants were new in 1970)
      tc(16)=tc(15)/2.0
```

```
      tc(18)=tc(15)/2.0
      if(time.eq.1970.0) v(16)=dlinf3('v12',0.0,tc(16))
      if(time.gt.1970.0) v(16)=dlinf3('v12',v(12),tc(16))
      v(18)=dlinf3('v16',v(16),tc(18))
c synthetic fuel capacity depreciation rate   million tpy/year
      v(15)=v(18)
c synthetic fuel capacity in construction   million tpy
c (v(9), v(11), and v(13) are the capacities within the delays)
      v(25)=v(9)+v(11)+v(13)
c synthetic fuel future capacity deficit   million tpy
c (if an excess is evident, the deficit is taken as 0)
      v(4)=amax1(v(3)-(v(14)+v(25)+v(7)-v(15)*tc(3)),0.0)
c synthetic fuel capacity initiation rate   million tpy/year
c (the functional relationships are to prevent unreal
c ordering rates even when deficits are very large; c(5) is max rate)
      v(5)=v(823)*v(23)*(c(5)*sin(1.11*v(4)/c(5))/t1)
      if(v(4).gt.1.42*c(5)) v(5)=c(5)
c synthetic fuel construction initiation rate   million tpy/year
      v(6)=delay3('v5',v(5),tc(6),v(7))
c synthetic fuel capacity coming on line   million tpy/year
c (that is,  the amount coming out of the last delay macro, v(12))
      v(8)=delay3('v6',v(6),tc(8),v(9))
      v(10)=delay3('v8',v(8),tc(10),v(11))
      v(12)=delay3('v10',v(10),tc(12),v(13))
c synthetic fuel standard plants (number)
      v(20)=v(14)/c(20)
c synthetic fuel standard plants under construction (number)
      v(28)=v(25)/c(20)
c synthetic fuel future coal capacity   million tpy
c (present capacity + current mine planning and construction
c delay times multiplied by the difference between current
c synfuel plant completion and depreciation rates times the
c nominal capacity utilization factor.  This number is used
c for planning new coal mine development)
      v(24)=(v(14)+(tc(807)+v(826))*(v(12)-v(15)))*c(3)
c synthetic fuel current coal needs   million tpy
      v(27)=v(14)*c(3)
c
c synthetic fuel capacity level equation   million tpy
      v(14)=v(14)+dt*(v(12)-v(15))
c
c -------------- electrical generation subsector ----------------
c
c electric generation scenario (coal input)    million tpy
      v(37)=energy(1,ncdp,time)
c electrical generation scenario planned capacity   million tpy
c (see also the note for analogous synfuel statement)
      tceg=tc(38)*(1.0-(1.0/(1.0+exp((time-1970.0)/1.7))))
      v(38)=c(36)*energy(1,ncdp,time+tceg)/c(38)
c total export option
      if(switch(02).eq.1) v(38)=0.0
c electrical generation planning plus regulatory delay   years
c (see also the note above for analogous coal mine delays)
```

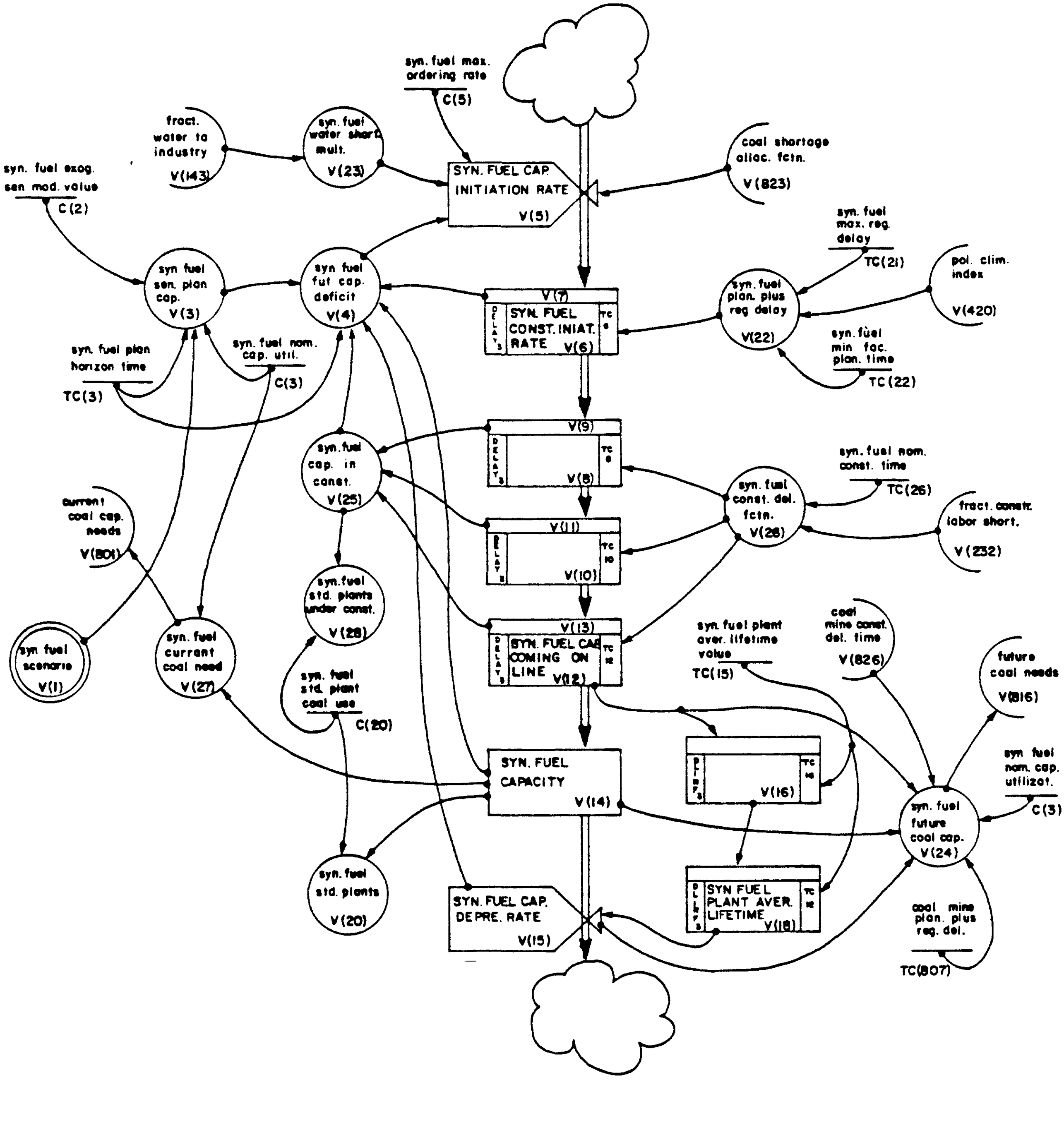
```
      v(56)=tc(56)+tc(55)/(1.0+exp((pci+0.1)/0.15))
      tc(41)=v(56)
c electrical generation construction delay function (including
c labor shortage effects   years   (see also the note for
c analogous synfuel statement)
      v(61)=tc(61)
      if(switch(04).eq.1) v(61)=tc(61)/(1.0-v(232))
c electrical generation construction time constants   years
      tc(43)=0.1875*v(61)
      tc(45)=0.1875*v(61)
      tc(47)=0.6250*v(61)
c electrical generation water shortage multiplier (hypothetical)
c (see also note for analogous synfuel statement)
      v(57)=1.0
      if(switch(01).eq.1) v(57)=(sin(v(143)*3.14159/2.0))**2.0
c initialize electrical generation capacity in planning and in
c construction to steady state in 1970
c (difference in 1970 between future planned capacity and
c present existing capacity divided by 1 + the total time
c through the planning and construction processes)
      if(time.gt.1970.0) go to 30
      vtemp2=(v(38)-v(55))/(1.0+v(61)+tc(41))
      v(41)=delay3('v40',vtemp2,tc(41),v(42))
      v(43)=delay3('v41',vtemp2,tc(43),v(44))
      v(45)=delay3('v43',vtemp2,tc(45),v(46))
      v(47)=delay3('v45',vtemp2,tc(47),v(48))
30    continue
c electrical generation plants mean age   years
c (in the model the electrical plants are not physically
c depreciated because it is assumed that the modeling time
c does not go out far enough.  However, their utilization
c factors are degraded with time.  A mean age
c is needed for this)
      if(time.eq.1970.0) v(50)=v(55)*time
      v(51)=0.0
      if(v(55).gt.0.0) v(51)=time-v(50)/v(55)
c electrical generation load factor
c (new plants are assumed to operate at a load factor of
c 80% for the first 9 years)
      v(52)=0.8-0.4*(v(51)-9.0)/28.0
      if(v(52).lt.0.0) v(52)=0.0
      if(v(51).lt.9.0) v(52)=0.8
c electrical generation normal utilized capacity   million tpy
      v(53)=v(52)*v(55)
c (statements for v(62) through v(64) are used to determine
c how much exisiting capacity will be degraded during the
c planning horizon time interval for electrical plants)
c elec. gen. capacity depreciation planning age   years
      v(62)=v(51)+tc(38)
c elec. gen. capacity depreciation planning fraction   per year
      if(v(62).lt.9.0) v(63)=0.0
      if(v(62).gt.9.0.and.v(62).le.9.0+tc(38)) v(63)=((v(62)-9.0)/tc(38)
     1)*0.0143
```

```
      if(v(62).gt.9.0+tc(38)) v(63)=0.0143
c elec. gen. planned capacity depreciation rate
c million tpy per year
      v(64)=v(63)*v(55)
c (as for planning electrical plants, statements for v(65)
c through v(67) degrade the utilization over the planning
c horizon time interval for coal mines)
c elec. gen. coal use depreciation planning age   years
      v(65)=v(51)+tc(807)+v(826)
c elec. gen. coal use depreciation planning fraction   per year
      if(v(65).le.9.0) v(66)=0.0
      if(v(65).gt.9.0.and.v(65).le.9.0+tc(807)+v(826)) v(66)=(((v(65)-9.
     1 0)/(tc(807)+v(826))))*0.0143
      if(v(65).gt.9.0+tc(807)+v(826)) v(66)=0.0143
c elec. gen. planned coal use depreciation   million tpy/year
      v(67)=v(66)*v(55)
c electrical generation capacity in construction   million tpy
      v(60)=v(44)+v(46)+v(48)
c electrical generation future capacity deficit   million tpy
      v(39)=amax1(v(38)-(v(53)/c(38)+v(60)+v(42)-v(64)*tc(38)),0.0)
c electrical generation capacity initiation rate   million tpy/yr
c (the functions prevent unduly large ordering rates)
      v(40)=v(823)*v(57)*(c(40)*sin(1.11*v(39)/c(40))/t1)
      if(v(39).gt.1.42*c(40)) v(40)=c(40)
c electrical generation construction initiation rate   million
c tpy/year
      v(41)=delay3('v40',v(40),tc(41),v(42))
c electrical generation capacity coming on line   million tpy/yr
c (see also note for analogous synfuel statements)
      v(43)=delay3('v41',v(41),tc(43),v(44))
      v(45)=delay3('v43',v(43),tc(45),v(46))
      v(47)=delay3('v45',v(45),tc(47),v(48))
c electrical generation standard 1000 mwe plants (number)
      v(54)=v(55)/c(54)
c electrical generation effective standard 1000 mwe plants
      v(59)=v(53)/c(54)
c electrical generation standard plants under construction (number)
      v(68)=v(60)/c(54)
c electrical generation future coal capacity   million tpy
c (present utilized capacity + rates added - rates depreciated
c taken over the total mine planning + construction times)
      v(58)=v(53)+(v(47)-v(67))*c(38)*(tc(807)+v(826))
c time weighted electrical generation growth
      v(49)=v(47)*time
c
c electrical generation level equations
c electrical generation nominal capacity   million tpy
      v(55)=v(55)+dt*v(47)
c time weighted electrical generation capacity
      v(50)=v(50)+dt*v(49)
c
c --------------- coal export subsector -------------------------
c
```

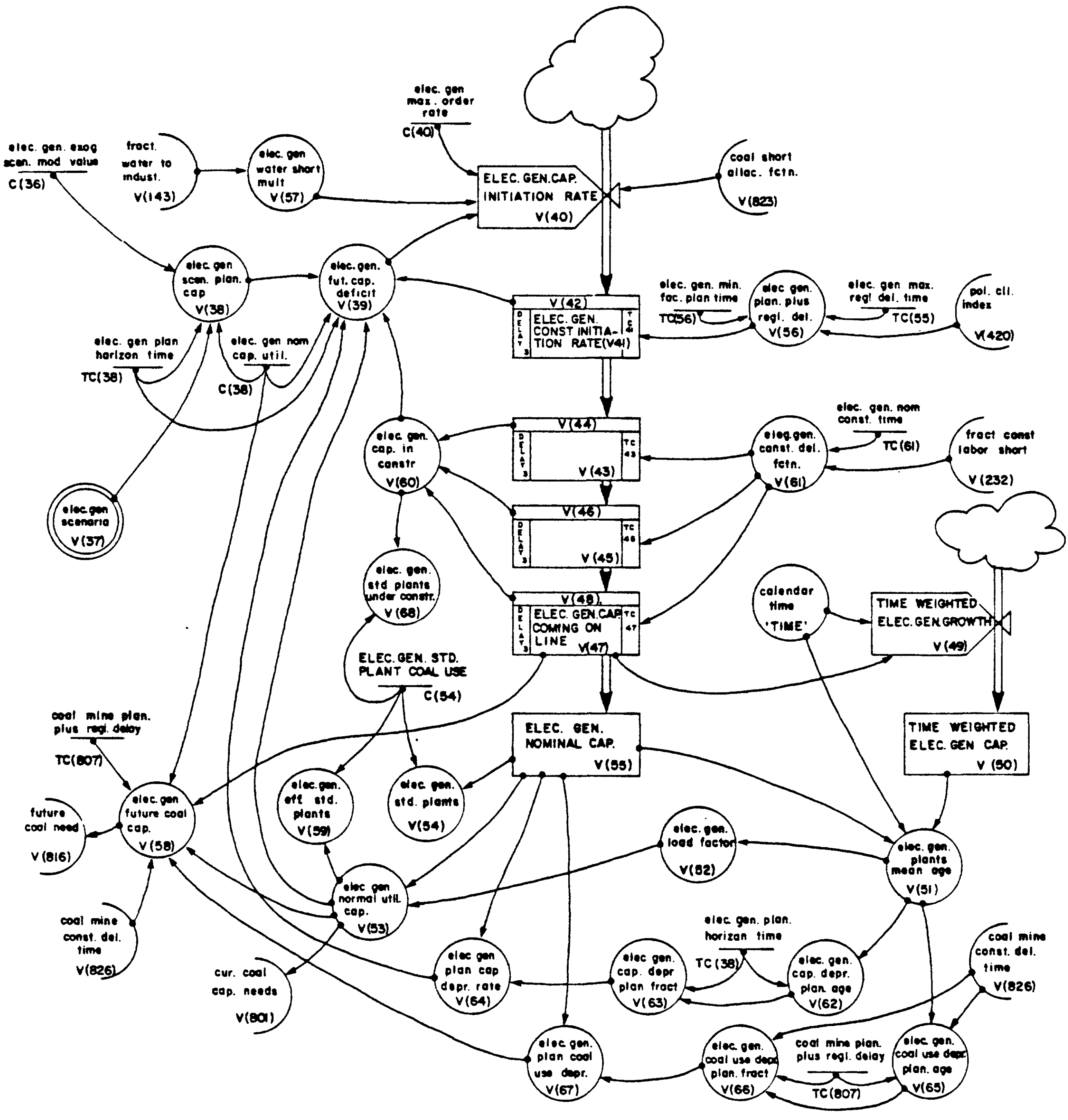
```
c coal export scenario   million tpy
      v(71)=energy(2,ncdp,time)
c coal export senario planned capacity   million tpy
c (see also note for analogous synfuel statements)
      tcce=tc(73)*(1.0-(1.0/(1.0+exp((time-1970.0)/1.7))))
      v(73)=c(72)*energy(2,ncdp,time+tcce)
c total export option
      if(switch(02).eq.1) v(73)=v(72)*totcol(ncdp,time+tc(73))
c coal export planning plus regulatory delay   years
c (see also note for analogous statements concerning coal
c mine planning at the beginning of this sector description)
      v(87)=tc(87)+tc(86)/(1.0+exp((pci+0.1)/0.15))
      tc(76)=v(87)
c coal export construction delay function (including labor short-
c age effects   years   (see also note for analogous synfuel)
      v(91)=tc(91)
      if(switch(04).eq.1) v(91)=tc(91)/(1.0-v(232))
      tc(78)=0.1875*v(91)
      tc(92)=0.1875*v(91)
      tc(94)=0.6250*v(91)
c fraction exported by slurry pipeline
      v(89)=c(89)
c coal export water shortage multiplier (hypothetical)
      v(88)=1.0
      if(switch(01).eq.1) v(88)=1.0-v(89)*(sin(v(143)*3.14159/2.0))**0.5
c initialize coal export capacity in planning and construction to
c steady state in 1970.  (see also note for analogous statements
c in electrical generation subsector)
      if(time.gt.1970.0) go to 40
      vtemp3=(v(73)-v(80))/(1.0+tc(76)+v(91))
      v(76)=delay3('v75',vtemp3,tc(76),v(77))
      v(78)=delay3('v76',vtemp3,tc(78),v(79))
      v(92)=delay3('v78',vtemp3,tc(92),v(93))
      v(94)=delay3('v92',vtemp3,tc(94),v(95))
40    continue
c coal export capacity average lifetime   years   (initialize as
c if all capacity were new in 1970)
      tc(82)=tc(81)/2.0
      tc(84)=tc(81)/2.0
      if(time.eq.1970.0) v(82)=dlinf3('v78',0.0,tc(82))
      if(time.gt.1970.0) v(82)=dlinf3('v94',v(94),tc(82))
      v(84)=dlinf3('v82',v(82),tc(84))
c coal export capacity depreciation rate   million tpy/year
      v(81)=v(84)
c coal export capacity under construction   million tpy
      v(86)=v(79)+v(93)+v(95)
c coal export future capacity deficit   million tpy
      v(74)=amax1(v(73)-(v(80)+v(86)+v(77)-v(81)*tc(73)),0.0)
c coal export capacity initiation rate   million tpy/year
c (see also note for analogous synfuel statements)
      v(75)=v(223)*v(98)*(c(75)*sin(1.11*v(74)/c(75))/t1)
      if(v(74).gt.1.42*c(75)) v(75)=c(75)
c coal export capacity construction initiation rate   million
```

```
c tpy/year
      v(76)=delay3('v75',v(75),tc(76),v(77))
c coal export capacity coming on line   million tpy/year
      v(78)=delay3('v76',v(76),tc(78),v(79))
      v(92)=delay3('v78',v(78),tc(92),v(93))
      v(94)=delay3('v92',v(92),tc(94),v(95))
c coal export future coal capacity   million tpy
      v(90)=v(80)+(tc(807)+v(826))*(v(94)-v(81))
c
c coal export capacity level equation   million tpy
      v(80)=v(80)+dt*(v(94)-v(81))
c
c
      return
      end
```

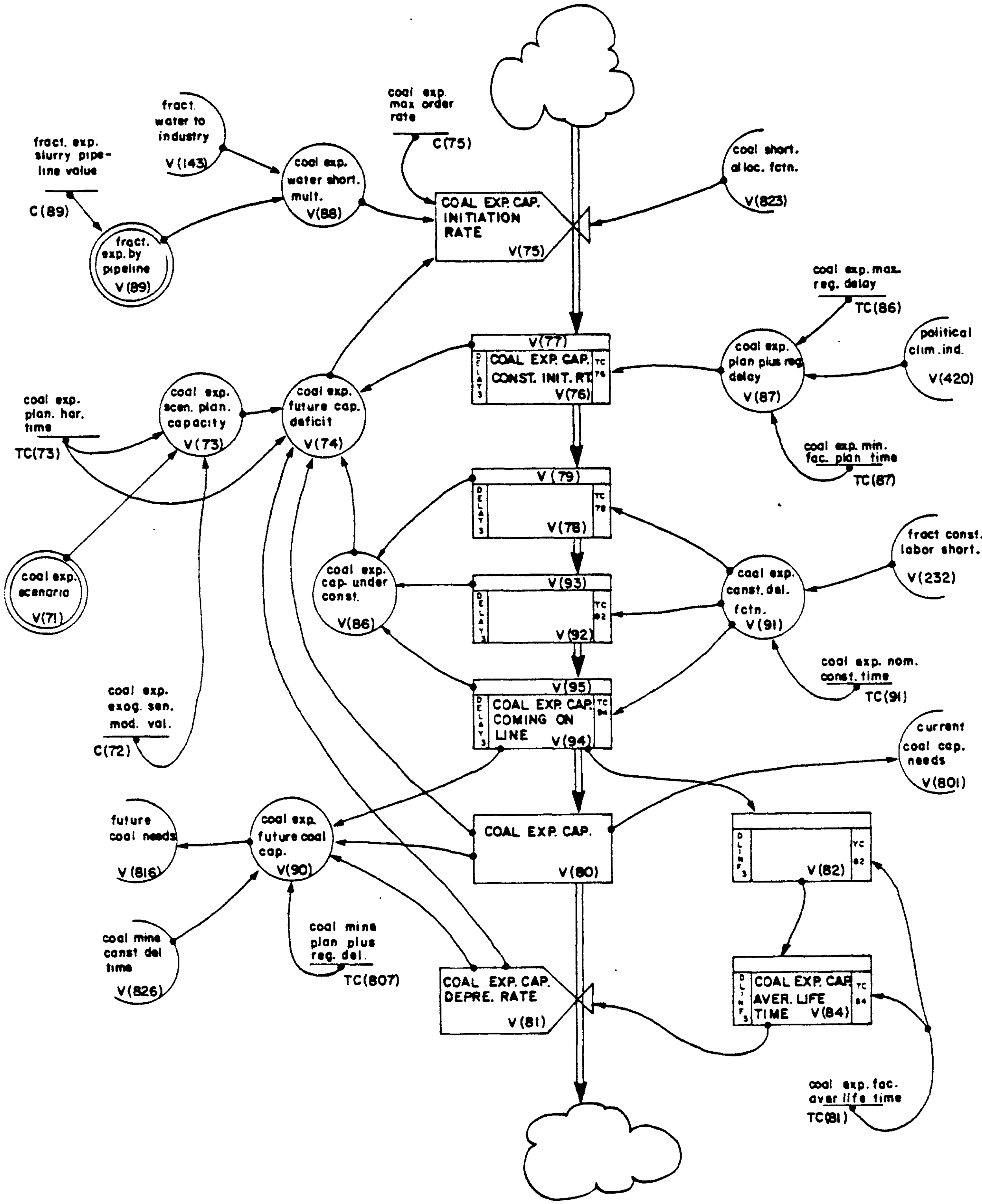
## SYNTHETIC FUEL SUBSECTOR
## WCRIND



# ELECTRICAL GENERATION SUBSECTOR WCRIND



# COAL EXPORT SUBSECTOR
# WCRIND

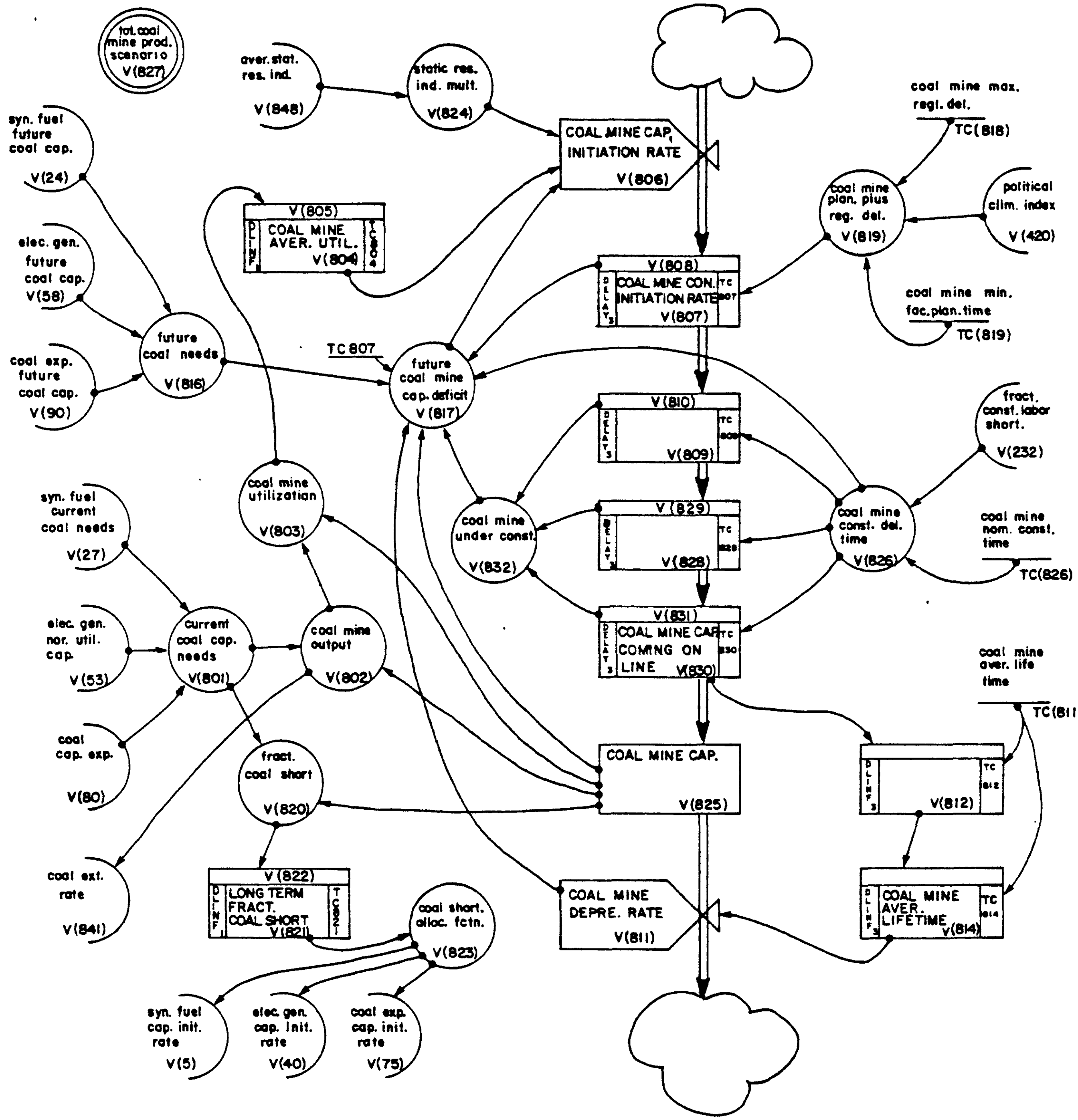


```
c *************** Mining Sector (08-26-77)***********************
c
c This sector models the growth of coal mines in  response  to  the
c user  industries  and  keeps  an  accounting of coal reserves and
c resources.
c
c The coal mining subsector structure is very similar to the  user
c industries,  and  the  annotation  for  wcrind  will  explain the
c modeling rational.
c
c The reserves subsector removes coal from  reserves  according  to
c mine  output  and initiates exploration to convert resources into
c reserves  when  the  static  reserve  index  gets  too  low.   As
c resources  are  depleted,  it  becomes more difficult to prove up
c reserves, and more land must be stripped per unit of  production.
c
c    applicable switches (set to 1 to activate, default value is 0)
c
c      switch number               function
c      03                 political climate feedback
c      04                 construction labor shortage feedback
c
c
c ****************start program statements**********************
c
c mining sector file name
      subroutine wcrmin(ncdp,mingo)
      integer mingo
c
c common statements
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c
c switch link statements
      common/switch/switch(100)
      integer switch
c see industry sector description concerning analogous indgo
      if(mingo.eq.0) return
      if(mingo.gt.0) go to 10
      if(mingo.lt.0) mingo=mingo+1
10    continue
c
c ---------------- coal mining subsector ----------------------
c
c coal mine planning delay plus regulatory delay   years
c calculated in wcrind
c coal mine construction delay time (including labor
c shortage effects)   years   calculated in wcrind
c coal mine construction time constants   years
      tc(809)=0.1875*v(826)
      tc(828)=0.1875*v(826)
      tc(830)=0.6250*v(826)
```

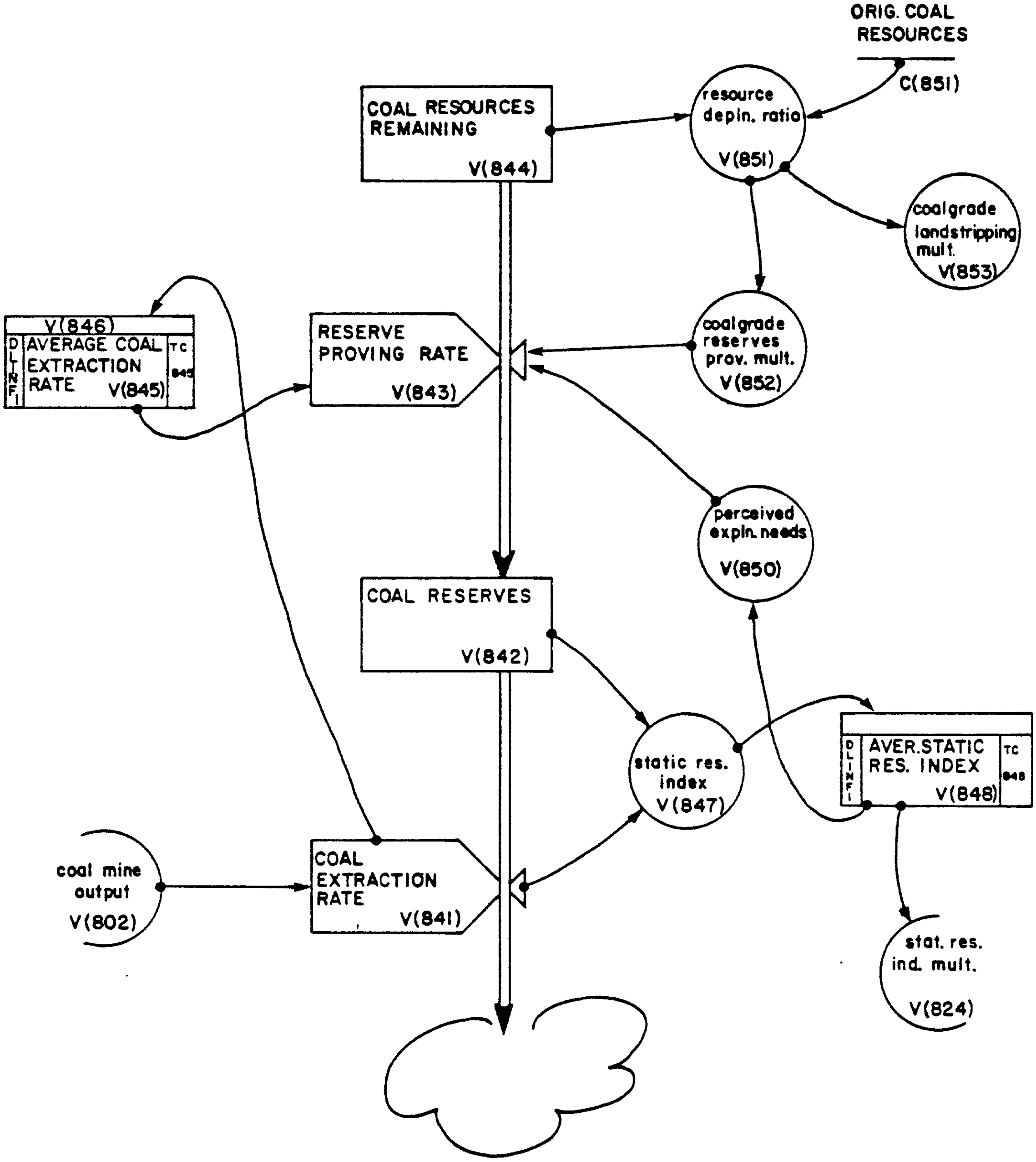
```
c tpy is an abbreviation for tons per year
c future coal needs   million tpy
      v(816)=v(24)+v(58)+v(90)
c initialize coal mine capacity in planning and construction to
c steady state in 1970
      if(time.gt.1970.0) go to 20
      vtemp4=(v(816)-v(825))/(1.0+tc(807)+v(826))
      v(807)=delay3('v806',vtemp4,tc(807),v(808))
      v(809)=delay3('v807',vtemp4,tc(809),v(810))
      v(828)=delay3('v809',vtemp4,tc(828),v(829))
      v(830)=delay3('v828',vtemp4,tc(830),v(831))
20    continue
c coal mine average lifetime   years   (initialize as if all
c mines are new in 1970)
      tc(812)=tc(811)/2.0
      tc(814)=tc(811)/2.0
      if(time.eq.1970.0) v(812)=dlinf3('v830',0.0,tc(812))
      if(time.gt.1970.0) v(812)=dlinf3('v830',v(830),tc(812))
      v(814)=dlinf3('v812',v(812),tc(814))
c coal mine depreciation rate   million tpy/year
      v(811)=v(814)
c coal mines under construction   million tpy
      v(832)=v(810)+v(829)+v(831)
c future coal mine capacity deficit   million tpy
      v(817)=amax1(v(816)-(v(825)+v(832)+v(808)-v(811)*(tc(807)+v(826)))
     1,0.0)
c current coal capacity needs   million tpy
      v(801)=v(53)+v(27)+v(80)
c fractional coal shortage
      v(820)=0.0
      if(time.eq.1970.0) v(801)=v(825)
      if(v(801).gt.v(825)) v(820)=(v(801)-v(825))/v(825)
c long term (averaged) fractional coal shortage
      v(821)=dlinf1('v820',v(820),tc(821))
c coal shortage allocation function
      v(823)=1.0-v(821)
c coal mine output   million tpy
      v(802)=amin1(v(801),v(825))
c coal mine utilization
      v(803)=1.0
      if(v(825).gt.0.0) v(803)=v(802)/v(825)
c coal mine average utilization
      v(804)=dlinf1('v803',v(803),tc(804))
c static reserve index multiplier
      v(824)=1.0
      if(v(848).lt.10.0) v(824)=(sin((v(848)/10.0)*(3.14159/2.0)))**0.3
      if(time.eq.1970.0) v(824)=1.0
c coal mine capacity initiation rate   million tpy/year
      v(806)=v(824)*v(804)*amax1(v(817)/t1,0.0)
c coal mine construction initiation rate   million tpy/year
      v(807)=delay3('v806',v(806),tc(807),v(808))
c coal mine capacity coming on line   million tpy/year
      v(809)=delay3('v807',v(807),tc(809),v(810))
```

```
      v(828)=delay3('v809',v(809),tc(828),v(829))
      v(830)=delay3('v828',v(828),tc(830),v(831))
c
c coal mine capacity level equation   million tpy
      v(825)=v(825)+dt*(v(830)-v(811))
c
c total coal mine production scenario   million tpy
      v(827)=totcol(ncdp,time)
c
c ---------------- coal reserves subsector -------------------------
c
c coal extraction rate   million tpy
      v(841)=v(802)
c average coal extraction rate   million tpy
      v(845)=dlinf1('v841',v(841),tc(845))
c static reserve index   years
      v(847)=11.0
      if(v(841).gt.0.0) v(847)=v(842)/v(841)
c average static reserve index   years
      v(848)=dlinf1('v847',v(847),tc(848))
c perceived exploration needs
      v(850)=0.0
      if(v(848).lt.20.0) v(850)=10.0*(1.0-(sin((v(848)/20.0)*(3.14159/2.
     10)))**0.3)
c resource depletion ratio
      v(851)=v(844)/(c(851)-v(844))
c coal grade reserves proving multiplier
      v(852)=1.0
      if(v(851).lt.7.0) v(852)=(sin((v(851)/7.0)*(3.14159/2.0)))**2.0
c reserve proving rate  million tpy
      v(843)=v(845)*v(850)*v(852)
c coal grade land stripping multiplier
      v(853)=1.0
      if(v(851).lt.7.0) v(853)=(sin((v(851)/7.0)*(3.14159/2.0)))**(-2.0)
c
c coal reserves level equations
c
c coal reserves   tons
      v(842)=v(842)+dt*(v(843)-v(841))
c coal resources remaining   millions tons
      v(844)=v(844)-dt*v(843)
c
      return
      end
```

# COAL MINING SUBSECTOR
## WCRMIN



# COAL RESERVES SUBSECTOR
## WCRMIN

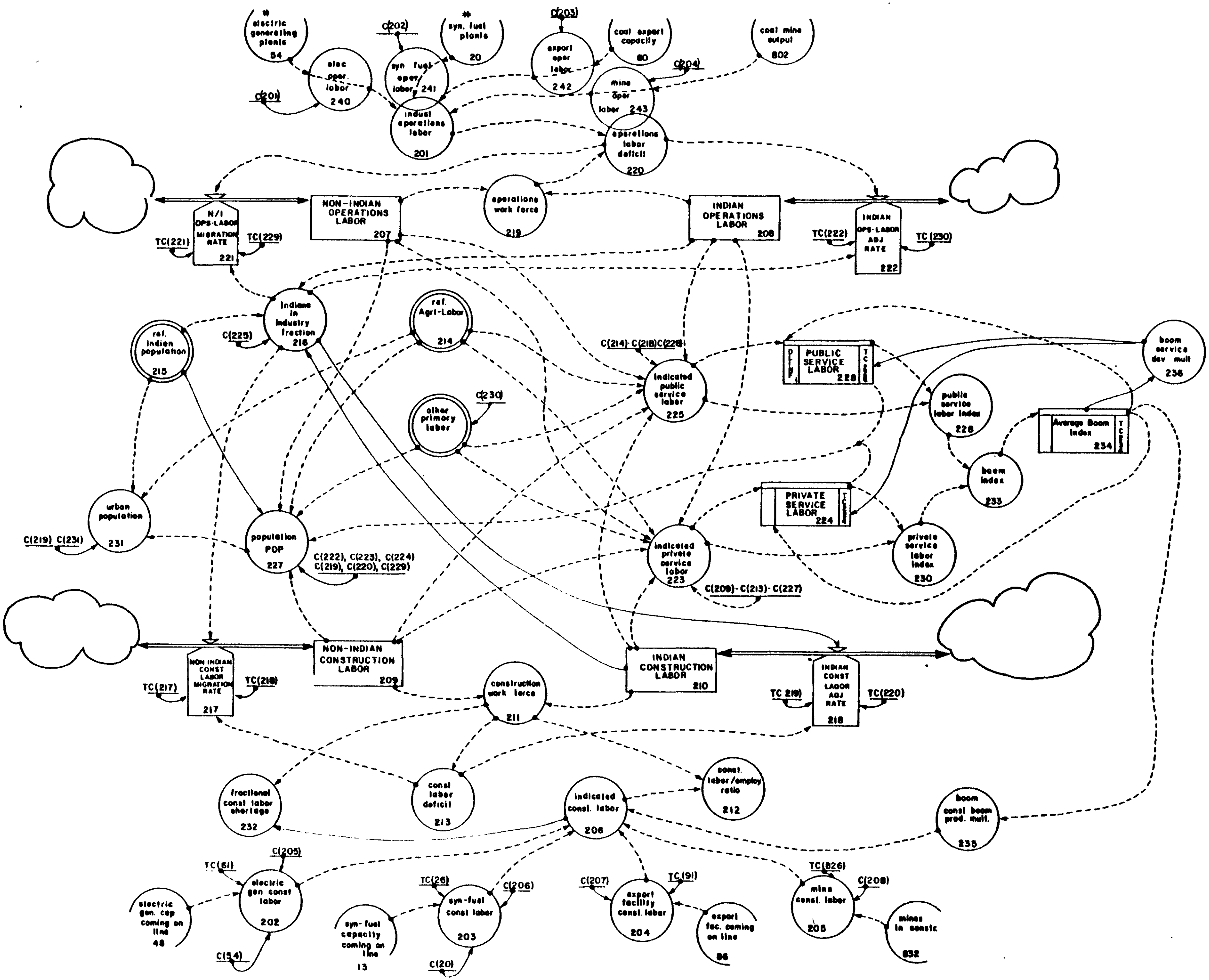


```
c ********************* Demographic Sector (08-26-77) *******************
c
c       The demographic sector calculates the number of laborers
c  required to construct and operate the mines, conversion plants
c  and export facilities specified by the industrial sector of the
c  model.   This  industrial  labor  force  together  with  an
c  exogenously determined agricultural labor force v(214) and other
c  primary  labor  v(229)  (primary manufacturing, recreation, etc.)
c  work through multipliers to determine the needed public v(225)
c  and private v(223) service labor.  The available service labor
c  v(226,224) is a time delayed function of the indicated service
c  labor categories.  Public and private service indices v(228,230)
c  are indications of how well the need for service labor is being
c  met.
c     Total population v(227) is determined by applying specific
c  family  multipliers  to  the  various  labor  forces  plus  the
c  exogenously determined indian population v(215).  Indians in the
c  industrial  labor  force  are  accounted  for  separately
c  v(208,210,216).  This allows some flexibility to accomodate
c  possible large and abrupt increases in indian employment through
c  specific indian employment clauses in future leases for coal on
c  the reservations.
c    A boom index  ( A. Ford,1976 )  is calculated v(233) as a
c  function of service shortages.  Boom conditions cause a decrease
c  in  construction  worker productivity  v(235)  and an increase
c  in  service  delay  times  v(236),  thus  aggravating  the  boom
c  conditions.
c
c  Note that there is no way of analyzing local "boom and bust"
c  cycles related to local development.  It is quite possible that
c  specific towns within the region will experience a relatively
c  large influx of construction labor and related service personnel
c  and, after the construction phase is completed, the construction
c  labor will move to a new site still within the region but far
c  enough away from the first site so that town experiences a "bust"
c  in population.  Such an effect could not be recorded in this
c  aggregated model of the region.
c
c  Demographic variables are aggregated over the entire region.
c
c      switches (default value = 0)
c
c          set switch(21)=1 to exclude indian reservations population
c          set switch(22)=1 to stabilize agricultural labor at 1975 level
c
c *************** start program statements *************************
      subroutine wcrdem(demgo)
      integer demgo
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c
c switch array
```

```
c
      common/switch/switch(100)
      integer switch
c
c
      if (demgo.eq.0) return
      if (demgo.gt.0) goto 10
      if (demgo.lt.0) demgo=demgo+1
10    continue
c ---------------------------------------------------------------------
c operations labor    (number of workers)
c
c electrical operations labor
       v(240)=c(201)*v(54)
c synfuel operations labor
       v(241)=c(202)*v(20)
c export operations labor
       v(242)=c(203)*v(80)
c mine operations labor
       v(243)=c(204)*v(802)
c industrial operations labor
       v(201)=v(240)+v(241)+v(242)+v(243)
c ---------------------------------------------------------------------
c construction labor   (number of workers)
c
c electrical generation construction labor
      v(202)=c(205)*v(48)/(tc(61)*0.6250*c(54))
c synthetic fuel construction labor
      v(203)=c(206)*v(13)/(tc(26)*0.6250*c(20))
c export construction labor
      v(204)=c(207)*v(86)/tc(91)
c mining construction labor
      v(205)=c(208)*v(832)/tc(826)
c indicated industrial construction labor
      v(206)=v(202)+v(203)+v(204)+v(205)
c boom construction productivity multiplier
      v(235)=0.5+0.5/(1.0+exp((v(234)-0.21)/0.055))
c industrial construction labor
      if(switch(25).eq.0)v(206)=v(206)/v(235)
c construction work force
      v(211)=v(209)+v(210)
c construction labor/employment ratio
      if(v(211).eq.0.0)v(212)=1.0
      if(v(211).gt.0.0)v(212)=v(206)/v(211)
c construction labor deficit
      v(213)=v(206)-v(211)
c ---------------------------------------------------------------------
c
c reference  agricultural labor
      v(214)=0.63*4292.0*exp(-0.0239*(time-1970.0))
      if(switch(22).eq.1.and.time.ge.1975.0) v(214)=2400.0
c reference indian population
      v(215)=4125.0*exp(0.0164*(time-1970.0))+2674*exp(0.0301*(time-1970
```

```
     1.0))
      if(switch(21).eq.1) v(215)=0.0
c indian industrial work force fraction
      v(216)=0.0
      if(v(215).gt.0.0) v(216)=c(225)/(1.0+exp((((v(208)+v(210))/v(215))
     1-0.13)/0.025))
c nonindian construction labor transfer rate
      if(v(213).ge.0.0)v(217)=(1.0-v(216))*v(213)/tc(217)
      if(v(213).lt.0.0)v(217)=(1.0-v(216))*v(213)/tc(218)
c indian construction labor transfer rate
      if(v(213).ge.0.0)v(218)=v(216)*v(213)/tc(219)
      if(v(213).lt.0.0)v(218)=v(216)*v(213)/tc(220)
c operations work force
      v(219)=v(207)+v(208)
c operations labor deficit
      v(220)=v(201)-v(219)
c nonindian operation labor transfer rate
      if(v(220).ge.0.0)v(221)=(1.0-v(216))*v(220)/tc(221)
      if(v(220).lt.0.0)v(221)=(1.0-v(216))*v(220)/tc(229)
c indian operation labor transfer rate
      if(v(220).ge.0.0)v(222)=v(216)*v(220)/tc(222)
      if(v(220).lt.0.0)v(222)=v(216)*v(220)/tc(230)
c other primary labor
      v(229)=c(230)
c indicated private service labor
      v(223)=c(209)*v(214)+c(210)*v(208)+c(211)*v(207)+c(212)*v(209)+c(2
     113)*v(210)+c(227)*v(229)
c boom service development multiplier
      v(236)=0.5+0.5/(1.0+exp((v(234)-0.165)/0.04))
c
c private service labor
      v(237)=tc(224)
      if(switch(26).eq.0)v(237)=v(237)/v(236)
      v(224)=dlinf1('v224',v(223),v(237))
c indicated public service labor
      v(225)=c(214)*v(214)+c(215)*v(208)+c(216)*v(207)+c(217)*v(209)+c(2
     118)*v(210)+c(228)*v(229)
c
c public service labor
      v(238)=tc(226)
      if(switch(26).eq.0)v(238)=v(238)/v(236)
      v(226)=dlinf1('v226',v(225),v(238))
c
c total population
c
      v(227)=c(219)*v(214)+c(220)*v(207)+v(215)+c(222)*v(226)+c(223)*v(2
     124)+c(224)*v(209)+c(229)*v(229)
      if (switch(25).eq.0)v(227)=c(219)*v(214)+c(220)*v(207)+v(215)+c(22
     12)*v(226)+c(223)*v(224)+(c(224)*(.5+.5/(1.+exp((v(234)-.33)/.08)))
     2*v(209))+c(229)*v(229)
c public service labor index
      v(228)=v(226)/v(225)
c private service labor index
```

```
      v(230)=v(224)/v(223)
c boom index
      v(233)=((0.5/v(228))+(0.5/v(230)))-1.0
c average boom index
      v(234)=dlinf1('v234',v(233),tc(234))
c urban population
      v(231)=c(231)*(v(227)-v(215)+v(214)*c(219))
c fractional construction labor shortage
      v(232)=0.0
      if(v(206).ne.0.0) v(232)=(v(206)-v(211))/v(206)
c
c level equations
c
c nonindian operations labor
      v(207)=v(207)+dt*v(221)
c indian operations labor
      v(208)=v(208)+dt*v(222)
c nonindian construction labor
      v(209)=v(209)+dt*v(217)
c indian construction labor
      v(210)=v(210)+dt*v(218)
      return
      end
```



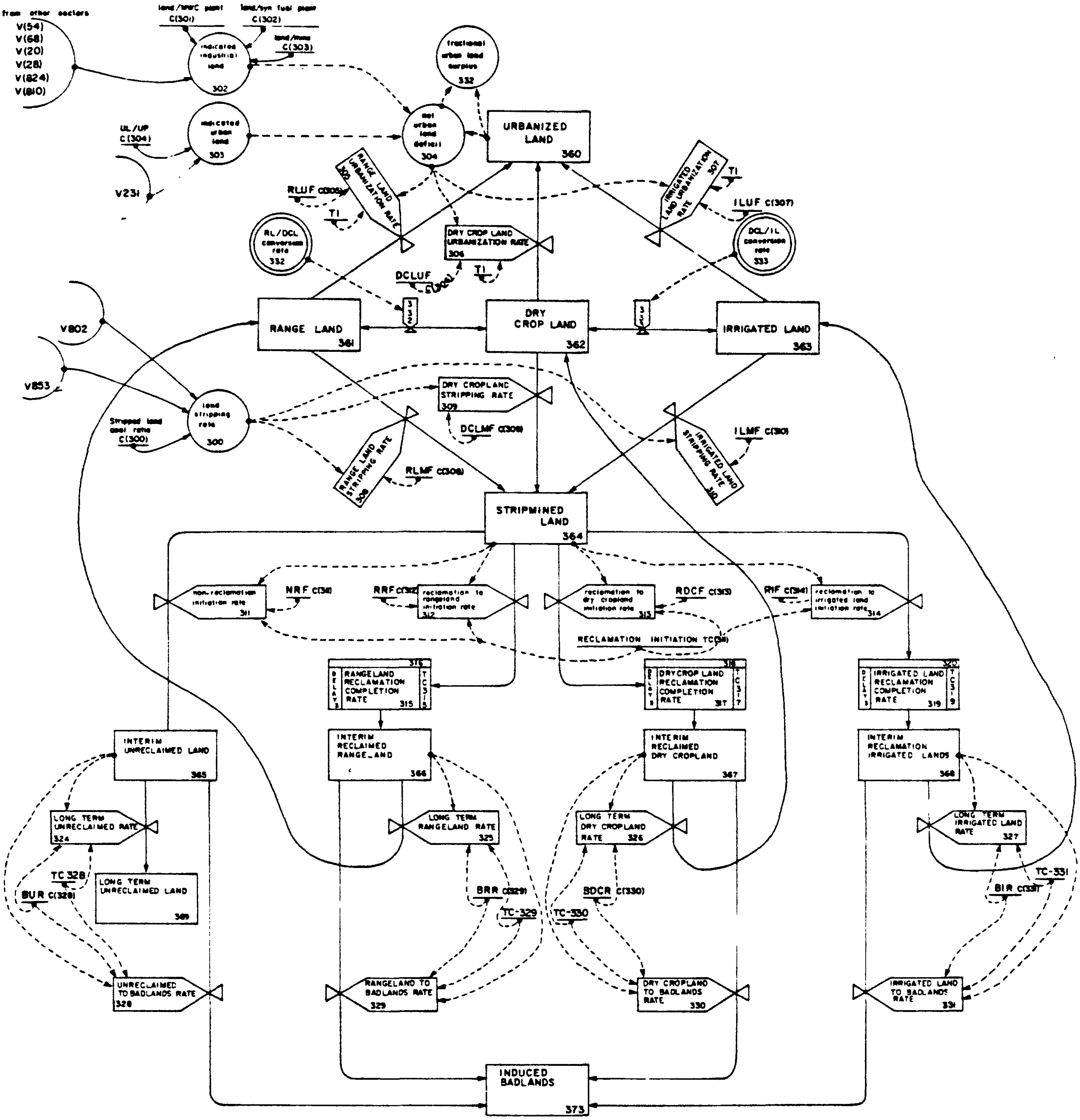

# DEMOGRAPHIC SECTOR

WCRDEM

```
c **************** Land Sector (08-26-77) ***********************
c
c The land sector keeps track of land  used  in  agricultural/open-
c space,  urban/industrial,  and  mine/reclamation  catagories. The
c agricultural/open-space catagories  are  range  and  forest  land
c v(361),   dry  crop  land  v(362),  and  irrigated  land  v(363).
c Provision  is  made  for  exogenous  transitions  between   these
c catagories  v(332,333).  Land  is  converted from agricultural to
c urban/industrial v(360) in  responce  to  indicated  requirements
c v(302,303)  calculated  from  industrial  development  and  urban
c population. Conversion from agriculture/open-space to  stripmined
c land  v(364) is governed by the land stripping rate v(300). Mined
c land, after a mining and reclamation delay ( tc315,tc317,tc319  )
c is  converted to interim reclaimed (or non-reclaimed) catagories.
c Land undergoing reclamation is in one of three categories, v(316)
c (to range land),  v(318)  (to  dry  crop  land),  or  v(320)  (to
c irrigated land). From the interim catagories, land either returns
c to  the  agricultural/open-space  catagories  or  deteriorates to
c induced badlands v(373).
c
c        units are 10e3 acres.
c
c *************** start program statements ************************
      subroutine wcrlnd(lndgo)
      integer lndgo
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
      if (lndgo.eq.0) return
      if (lndgo.gt.0) goto 10
      if (lndgo.lt.0) lndgo=lndgo+1
10    continue
c
c land stripping rate
      v(300)=v(802)*v(853)*c(300)
c indicated industrial land
      v(302)=(v(54)+v(68))*c(301)+(v(20)+v(28))*c(302)+(v(824)+v(8
     110))*c(303)
c indicated urban land
      v(303)= c(304)*v(231)
c net urban land deficit
      v(304)=v(302)+v(303)-v(360)
c fractional urban land surplus
      v(332)=amax1(-v(304),0.0)/v(360)
c range land urbanization rate
      v(305)=amax1(v(304),0.0)*c(305)/t1
c dry croo land urbanization rate
      v(306)=amax1(v(304),0.0)*c(306)/t1
c irrigated land urbanization rate
      v(307)=amax1(v(304),0.0)*c(307)/t1
c range land/dry crop land conversion rate
      v(332)=0.0
c dry crop land/irrigated land conversion rate
```

```
      v(333)=0.0
c range land stripping rate
      v(308)=v(300)*c(308)
c dry crop land stripping rate
      v(309)=v(300)*c(309)
c irrigated land stripping rate
      v(310)=v(300)*c(310)
c non-reclamation initiation rate
      v(311)=v(364)*c(311)/tc(311)
c reclamation to range land initiation rate
      v(312)=v(364)*c(312)/tc(311)
c reclamation to dry crop land initiation rate
      v(313)=v(364)*c(313)/tc(311)
c reclamation to irrigated land initiation rate
      v(314)=v(364)*c(314)/tc(311)
c range land reclamation completion rate
      v(315)=delay3('v312',v(312),tc(315),v(316))
c dry crop land reclamation completion rate
      v(317)=delay3('v313',v(313),tc(317),v(318))
c irrigated land reclamation completion rate
      v(319)=delay3('v314',v(314),tc(319),v(320))
c long term unreclaimed rate
      v(324)=v(365)*(1.0-c(328))/tc(328)
c unreclaimed to badlands rate
      v(328)=v(365)*c(328)/tc(328)
c long term range land rate
      v(325)=v(366)*(1.0-c(329))/tc(329)
c range land to badlands rate
      v(329)=v(366)*c(329)/tc(329)
c long term dry crop land rate
      v(326)=v(367)*(1.0-c(330))/tc(330)
c dry crop land to badlands rate
      v(330)=v(367)*c(330)/tc(330)
c long term irrigated land rate
      v(327)=v(368)*(1.0-c(331))/tc(331)
c irrigated lands to badlands rate
      v(331)=v(368)*c(331)/tc(331)
c total land
      v(399)=v(360)+v(361)+v(362)+v(363)+v(364)+v(316)+v(318)+v(320)+v(3
     165)+v(366)+v(367)+v(368)+v(369)+v(373)
c
c level equations
c
c urbanized land
      v(360)=v(360)+dt*(v(305)+v(306)+v(307))
c range land
      v(361)=v(361)-dt*(v(305)+v(308)+v(332)-v(325))
c dry crop land
      v(362)=v(362)-dt*(v(306)+v(309)+v(333)-v(332)-v(326))
c irrigated land
      v(363)=v(363)-dt*(v(307)+v(310)-v(333)-v(327))
c stripmined land
      v(364)=v(364)+dt*(v(308)+v(309)+v(310)-v(311)-v(312)-v(313)-v(314)
```

```
     1)
c interim unreclaimed land
      v(365)=v(365)+dt*(v(311)-v(324)-v(328))
c interim reclaimed range land
      v(366)=v(366)+dt*(v(315)-v(325)-v(329))
c interim reclaimed dry crop land
      v(367)=v(367)+dt*(v(317)-v(326)-v(330))
c interim reclaimed irrigated land
      v(368)=v(368)+dt*(v(319)-v(327)-v(331))
c long term unreclaimed land
      v(369)=v(369)+dt*v(324)
c induced badlands
      v(373)=v(373)+dt*(v(328)+v(329)+v(330)+v(331))
c cumulative stripped land
      v(374)=v(374)+dt*v(300)
c total mining disturbed land
      v(375)=v(399)-v(361)-v(362)-v(363)-v(360)
c land undergoing reclamation
      v(376)=v(316)+v(318)+v(320)
      return
      end
```




# LAND SECTOR

**WCRLND**

```
c ************ Water Sector (08-26-77)****************************
c
c The water sector is based upon a functional relationship between
c expected water yield at some percent deficiency (eg. a yield that
c is arbitrarily expected to be met in all but 2%  of the years)
c and the active storage (reservoir) capacity v(111,116).  Both
c yield and storage capacity are normalized to mean annual flow
c v(108,114).   Net yield  v(112,117)  is  calculated  by
c subtracting reservior evaporation v(110,115) from gross yield.
c The yield thus peaks at some finite value of the storage index
c (storage capacity/ mean annual flow).  The  yield  function
c depends upon the hydrologic statistics of the stream flow (Riggs
c and Henderson, 1973).
c
c Since in-stream flow reservation schedules have been proposed for
c fish and wildlife protection (ranging from about 50%  to  80%  of
c the mean annual flow), an "effective flow" v(109) is calculated
c as the remaining flow after reservation, and is used as a
c parameter in yield-storage function. Provision is made for
c "political feedback" to weaken fish and wildlife flow
c requirements in the event of severe shortage v(134,150,135,136).
c The fish and game index v(137) measures the degree to which the
c instream flow reservation is met. A subjective "free flowing
c river index" v(107) is also calculated as a function of the
c cumulative storage index v(106).
c
c The water sector represents the entire drainage basin.  The
c industrial water requirements are taken to be v(170) times the
c industrial requirements in the region.
c
c Water requirements and projections are calculated from the
c industrial sector. The projected water deficit v(146) is divided
c by the marginal yield v(118) to calculate the projected storage
c deficit  v(119).  The decision to build additional storage
c capacity depends upon a "benefit/cost" analysis v(163,122).
c "Cost" v(121) depends upon the annualized cost of storage
c capacity and the marginal yield (incremental acre-feet/year yield
c per acre-foot storage). "Benefit" v(161) is calculated as a
c (linear) function of the projected fractional deficit v(160).
c Available water is allocated by a "water allocation function"
c v(139).
c
c The aggregated drainage basin model tends to overestimate yield
c and underestimate resource development because it assumes that
c water available at any point within the basin is available at
c every point, and that reservoirs are of optimal efficiency. That
c is, storage is generally assumed to be on the lower, main stem,
c portion of the basin's river system. For these reasons it can be
c assumed that water resource impacts would exceed those simulated.
c
c          units are thousand acre-feet    (1.0e3af)
c
c**************** start program statements ********************
```

```
      subroutine wcrwtr(wtrgo)
      integer wtrgo
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c switch array  (0 is off, 1 is on)
      common/switch/switch(100)
      integer switch
      real ifrf,myf,fgf
c (see note for analogous indgo in industry sector description)
      if (wtrgo.eq.0) return
      if (wtrgo.gt.0) go to 10
      if (wtrgo.lt.0) wtrgo=wtrgo+1
10    continue
c
c indicated irrigation water to agriculture in basin
c (increase in consumptive use since 1970)
      v(126)=c(126) +(time-1970.)*c(125)/30.
c miscellaneous water requirements     (in basin)
      v(127)=c(123) +(time-1970.)*c(124)/30.
c indicated industrial water requirements     (in region)
      v(128)=v(20)*c(132)+v(59)*c(131)+v(802)*c(133)+v(80)*c(134)*v(89)
c developing industrial water requirements
      v(129)=v(128)+v(28)*c(132)+v(68)*c(131)+v(810)*c(133)+v(79)*
     1c(134)*v(89)
c diversion point index
c (fraction of industrial water from surface sources within basin)
      v(130)=c(130)
c basin/region water to industry multiplier
      v(170)=(c(6)+c(7))/c(6)
c effective indicated industrial water requirements (in basin)
      v(131)=v(130)*v(128)*v(170)
c effective developing industrial water requirements
      v(132)=v(130)*v(129)*v(170)
c indicated water requirements    (in basin)
      v(133)=v(131)+v(127)+v(126)
c cumulative storage index
c (storage/mean annual flow   years)
      v(106)=(v(105)+v(165))/c(106)
c free flowing river index    (Fermi function of storage index)
      v(107)=1.01/(1.0+exp((v(106)-0.5)/0.1))
c instream flow reservation index
c (political accommodation to shortage;
c Fermi function of fractional deficit)
      v(135)=ifrf(v(150))
c mean instream flow reservation
      v(136)=c(136)*v(135)
c mean effective stream flow
c (initial instream flow - instream flow reservation)
      v(109)=dlinf1('esf',c(106)-v(136),tc(109))
c effective storage index
      v(108)=v(105)/v(109)
c evaporation
```

```
      v(110)=v(105)*c(110)
c gross yield
      v(111)=v(109)*gyf(v(108),v(109)/c(106))
c net yield
      v(112)=v(111)+v(172)-v(110)
c yield defecit
      v(113)=v(133)-v(112)
c fractional yield deficit
      v(134)=amax1(0.0,v(113)/v(133))
c perceived fractional yield deficit  (perception delay)
      v(150)=dlinf1('v134',v(134),tc(120))
c yield surplus   (returned to instream flow)
      v(151)=-amin1(v(113),0.0)
c mean instream flow
      v(152)=v(136)+v(151)
c fish & game index   (fraction of reservation)
      v(137)=fgf(v(152))
c water allocation function
      call waf(v(112),v(131),v(127),v(126),v(140),v(141),v(142))
c v(140) water to industry
c v(141) water to miscellaneous
c v(142) water to agriculture
c fractional water to industry
      v(143)=1.0
      if(v(131).gt.0.0) v(143)=1.0-v(130)*(v(131)-v(140))/v(131)
c fractional water to miscellaneous
      v(144)=1.0
      if(v(127).gt.0.0) v(144)=v(141)/v(127)
c fractional water to agriculture
      v(145)=1.0
      if(v(126).gt.0.0) v(145)=v(142)/v(126)
c developing effective storage index
c (includes storage under construction, v(103))
      v(114)=(v(105)+v(103))/v(109)
c developing evaporation
      v(115)=(v(105)+v(103))*c(110)
c developing gross yield
      v(116)=v(109)*gyf(v(114),v(109)/c(106))
c developing net yield
      v(117)=v(116)+v(172)-v(115)
c marginal yield  (acre-feet yield per year per acre-foot stoage
c decreasing function of storage index)
      v(118)=myf(v(114),v(109)/c(106))-c(110)
      v(118)=amax1(v(118),0.0)
c weighted planning water deficit
      v(146)=c(146)*v(132)+c(147)*v(127)+c(148)*v(126)-v(117)
c fractional planning water deficit
      v(160)=v(146)/(v(146)+v(117))
c indicated storage capacity deficit
c (water deficit/marginal yield)
      if(v(118).gt.0.0) v(119)=v(146)/v(118)
c perceived storage capacity deficit  (perception delay)
      v(120)=dlinf1('v119',v(119),tc(120))
```

```
c marginal cost
c (annualized cost of storage/marginal yield)
      v(121)=10000.0
      if (v(118).gt.0.0) v(121)=c(121)/v(118)
c marginal value
c (linear function of fractional deficit)
c 'demand curve' for industrial water
      v(161)=c(161)+v(160)*(c(162)-c(161))
c benefit/cost ratio
      v(163)=amax1(v(161)/v(121),0.0)
c reservior utility multiplier
c (1-Fermi function of 'b/c' ratio)
      v(122)=1.0
      if(v(163).lt.8.0) v(122)=1.0-1.0/(1.0+exp((v(163)-c(127))/c(128)))
      if (v(118).le.0.0) v(122)=0.0
c water storage capacity growth
c (construction initiation rate)
      v(101)=amax1(0.0,v(120)*v(122))/tc(101)
c completed storage capacity
c (consrtuction completion rate)
      v(102)=delay3('v101',v(101),tc(102),v(103))
c storage capacity depreciation rate
c (sediment filling rate)
      v(104)=v(105)/tc(104)
c
c level equations
c
c storage capacity
      v(105)=v(105)+dt*(v(102)-v(104))
c sediment-filled storage
      v(165)=v(165)+dt*v(104)
c
100   continue
      return
      end
```

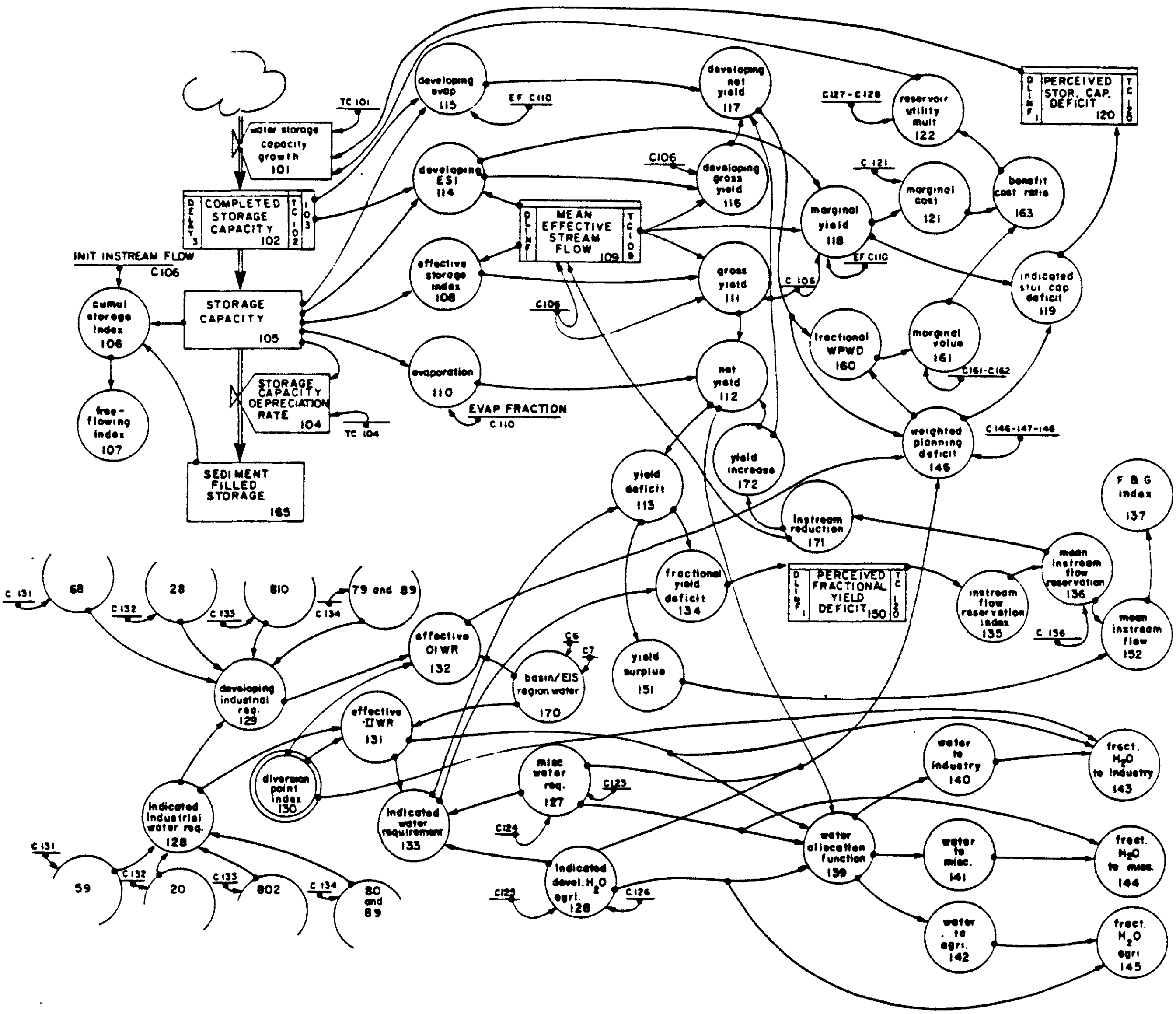
```
c      ****************real function ifrf**************
c
c instream flow reservation index (ifrf) represents a political
c accommodation to water shortage by relaxing the instream flow
c reservation.
c ifrf is a Fermi function of fractional water deficit (f).
c
      real function ifrf(f)
      if(f.gt.1.0) pause'ifrf'
      ifrf=0.1+0.9/(1.0+exp((f-0.6)*8.0))
      return
      end


c ********** gross yield function (05-11-76) **********
c
c The gross yield function (gyf) is the fraction of mean effective
c stream flow that could be expected to be available in 99% of the
c years.  It is crudely estimated for the Yellowstone River Basin.
c Effective stream flow is initial instream flow less instream flow
c reservation.
c x is the effective storage index (effective storage/mean effective
c stream flow).
c m is mean effective stream flow/initial instream flow.
c
      real function gyf(x,m)
      real x,m
      a=.250+1.41*m+0.20*m*m
      b=0.001+0.215*m-0.237*m*m+0.283*m*m*m
      c=0.366+0.878*m-0.476*m*m+0.356*m*m*m
      if(m.lt.0.0) pause'gyf m<0'
      if(m.gt.1.0) pause'gyf m>1'
      gyf=(b+(c-a+a*b)*x-1.0)*exp(-a*x)+1.0
      return
      end


c      ****************real function fgf*************
c
c Fish and game function (fgf) is a 'measure' of the mean instream flow
c reservation (x) against an instream flow schedule.
c
      real function fgf(x)
      xx=x/1000.0
      fgf=.011+.123*xx+.00549*xx*xx
      if(x.gt.6187.0) fgf=1.0
      if(x.lt.0.0) pause'fgf'
      return
      end
```

```
c ******************water allocation function (04-13-76)**********
c
c Allocates water net yield (ny), based upon indicated requirements
c for industry (iw), misc. (mw), and agriculture (aw).  Allocations
c are wi, wm, and wa.
c waf can be modified to reflect water rights.
c
      subroutine waf(ny,iw,mw,aw,wi,wm,wa)
      real ny,iw,mw,aw,wi,wm,wa
      f=ny/(iw+mw+aw)
      if(f.gt.1.0) f=1.0
      wi=iw*f
      wm=mw*f
      wa=aw*f
      return
      end

c ******************marginal yield function (05-06-76)***************
c
c The marginal yield function is the derivative of the gross yield
c function (gyf) with respect to the effective storage index (x).
c See gross yield function.
c
      real function myf(x,m)
      real x,m
      a=.250+1.41*m+0.20*m*m
      b=0.001+0.215*m-0.237*m*m+0.283*m*m*m
      c=0.366+0.878*m-0.476*m*m+0.356*m*m*m
      if(m.lt.0.0) pause'myf m<0'
      if(m.gt.1.0) pause'myf m>1'
      myf=(-a*((c-a+a*b)*x-1.0)+(c-a+a*b))*exp(-a*x)
      return
      end
```



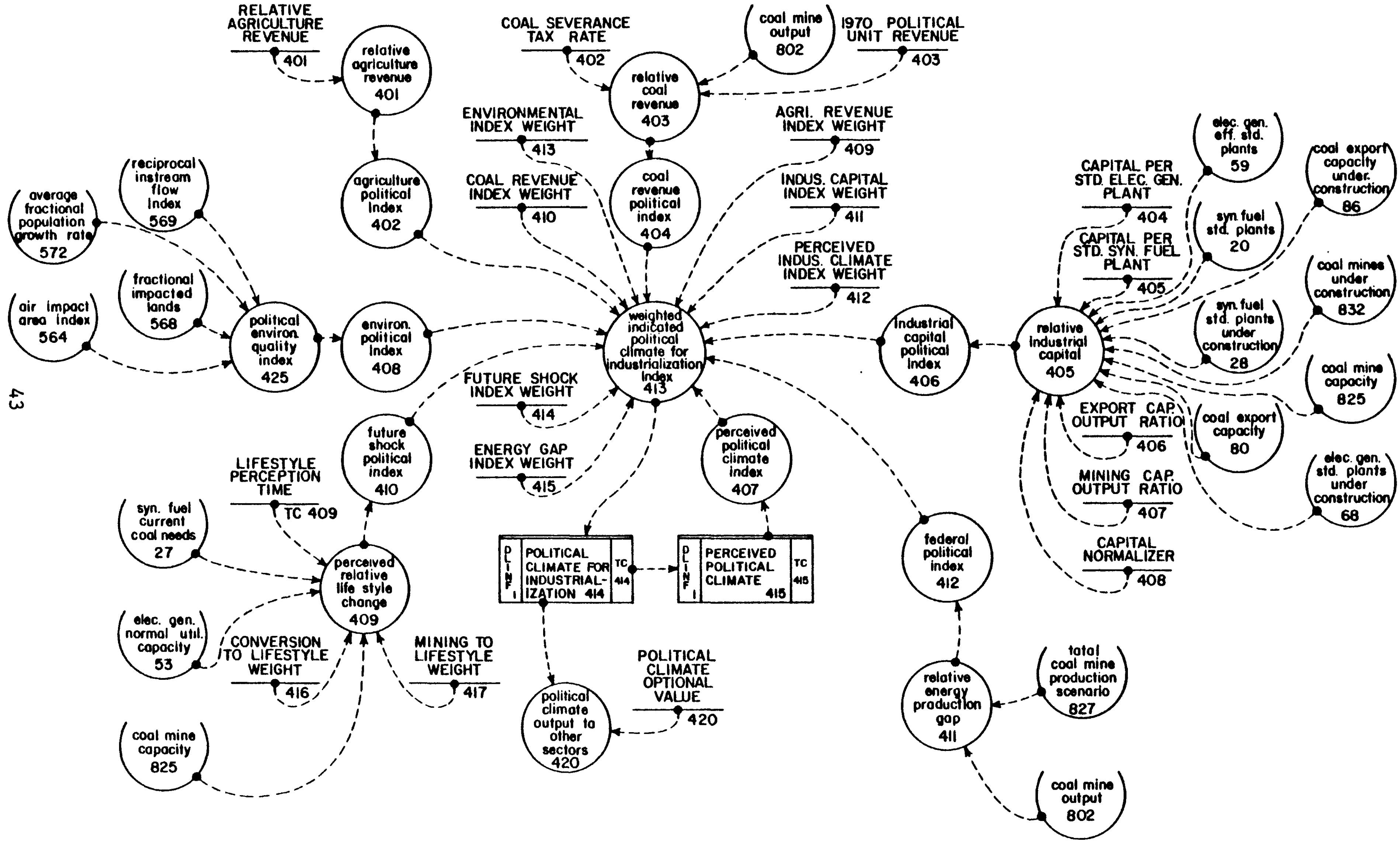
# WATER SECTOR

**WCRWTR**

```
c **************** Political Sector (7-22-77) *****************
c
c This sector calculates a "political climate for industrialization
c index," v(414), which is a time average of a weighted sum of
c several other indicies. These are: (1) agricultural political
c index, v(402), based on an exogenously supplied estimate; (2)
c coal revenue political index, v(404), based on a ratio between a
c severance tax and all other revenue to the political unit; (3)
c industrial capital political index, v(406), based on the amount
c of industrial capital invested in the region; (4) perceived
c political climate for industrialization, v(407), based on some
c perception delay of political climate; (5) environmental
c political index, v(408), based on a perception of environmental
c quality; (6) future shock political index, v(410), which
c attempts to model resistance to rapid change; and (7) federal
c political index, v(412), which models pressure from the national
c level for development when there is a gap between actual and
c expected (scenario) capacities.
c
c The hypothetical indicies are calculated in various ways, but all
c have the property of having values between -1 (complete
c opposition to development) to +1 (thorough support for develop-
c ment). Values of 0 are neutral.
c
c
c switch default value =0, set to 1 for feedback
c
c switch number                    function
c      40              political climate index feedback
c
c *************** start program statements ******************
c
        subroutine wcrpol(polgo,ncdp)
        integer polgo,ncdp
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c
c switch link statements
c
      common/switch/switch(100)
      integer switch
c
c (see note for analogous indgo in industrial sector description)
c
      if(polgo.eq.0) return
      if(polgo.gt.0) go to 10
      if(polgo.lt.0) polgo=polgo+1
10    continue
c
c relative agricultural revenue
        v(401)=c(401)
c agricultural political index (relative agricultural revenue)
```

```
      v(402)=2.0/(1.0+exp((v(401)-1.0)/0.3))-1.0
c relative coal revenue
      v(403)=v(802)*c(402)/c(403)
c coal revenue political index (relative coal revenue)
      v(404)=1.0-1.0/(1.0+exp((v(403)-0.1)/0.04))
c relative industrial capital
      v(405)=((v(59)+v(68))*c(404)+(v(20)+v(28)
     1 )*c(405)+(v(80)+v(86))*c(406)+(v(832)+v(825))*c(407))/c(408)
c industrial capital political index (relative industrial capital)
      v(406)=1.0-1.0/(1.0+exp((v(405)-5.0)/1.5))
c perceived political climate index (perceived political climate)
      v(407)=2.0*sin(1.5708*(v(415)+1.0)/2.0)-1.0
c political environmental quality index
      v(425)=((v(569)**2+(abs(v(572))/0.075)**2+v(564)**2+
     1 (v(568)/0.1)**2)/4.)**.5
c environmental political index (peqi)
      v(408)=0.7*exp(-(v(425)-1.0)**2)-1.0
c perceived relative lifestyle change
      v(409)=((energy(1,ncdp,time+tc(409))+energy(2,ncdp,time+tc(409))
     1 -v(27)-v(53))*c(416)+(totcol(ncdp,time+tc(409))-v(825))*c(417))/
     1 (c(416)*(v(27)+v(53))+c(417)*v(825))
c future shock political index (relative lifestyle change)
      v(410)=1.0/(1.0+exp(v(409)-3.5)/0.1)-1.0
c relative energy production gap
      v(411)=(v(827)-v(802))/v(827)
c federal political index (gap)
      v(412)=1.0-1.0/(1.0+exp((v(411)-0.25)/0.1))
c weighted indicated political climate for industrialization index
      v(413)=(v(402)*c(409)+v(404)*c(410)+v(406)*c(411)+v(407)*c(412)+
     1 v(408)*c(413)+v(410)*c(414)+v(412)*c(415))/(c(409)+c(410)+
     1 c(411)+c(412)+c(413)+c(414)+c(415))
c political climate for industrialization
      v(414)=dlinf1('v413',v(413),tc(414))
c political climate output to other sectors
      v(420)=v(414)
      if(switch(40).eq.0) v(420)=c(420)
c perceived political climate
      v(415)=dlinf1('v414',v(414),tc(415))
      return
      end
```

# POLITICAL SECTOR

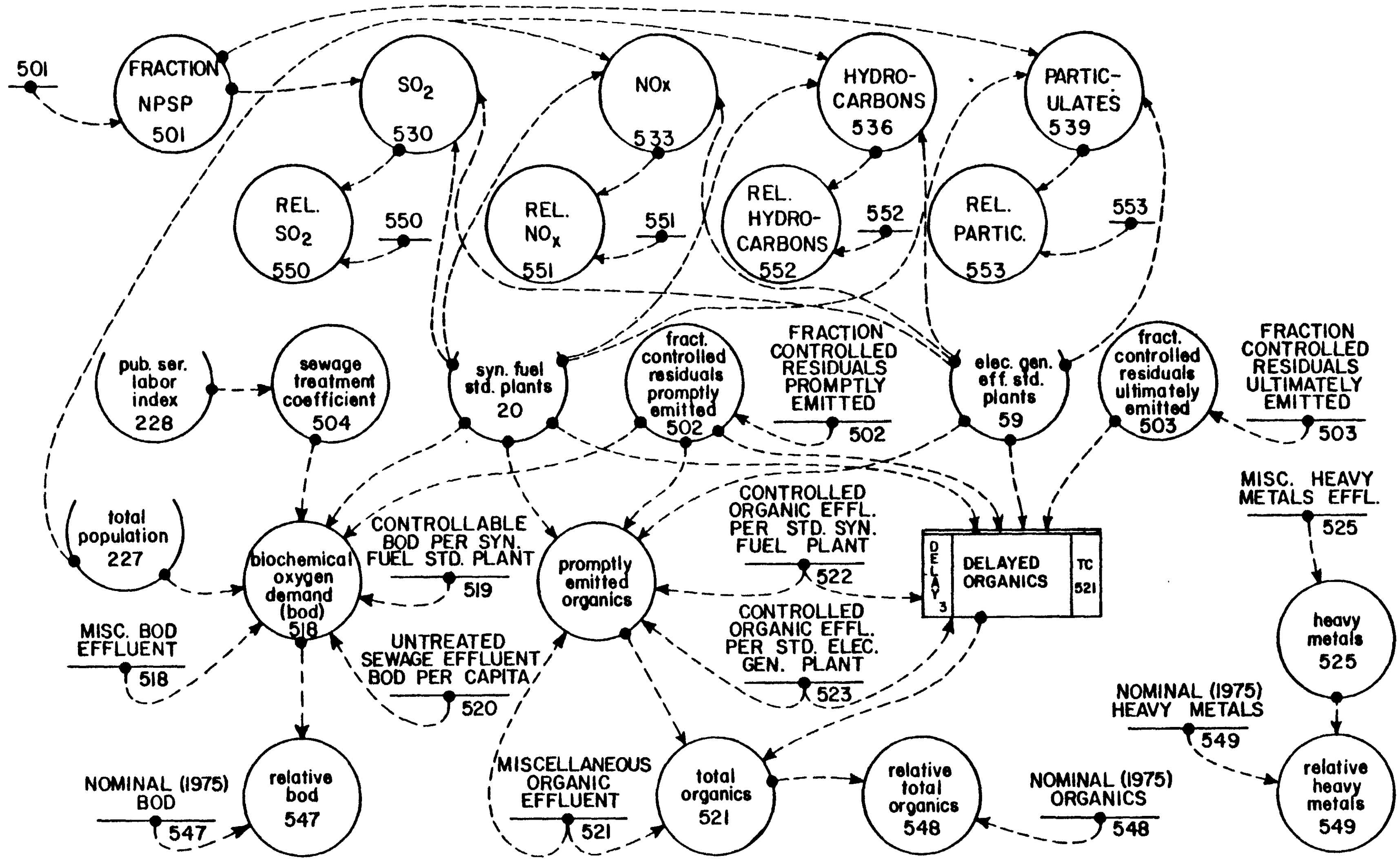


```
c ********** Environmental Indicators Sector (10-05-77) **********
c
c The environmental sector of the model estimates relevant effluent
c residuals as linear functions of the various industrial
c construction and operation activities and as functions of total
c population and land use.
c
c Provision has been made to model a partial control of waterborne
c effluents and their delayed, partial release to the environment.
c In this way the model simulates short-term control of waterborne
c effluents and also a gradual leaching of these pollutants into
c the hydrologic system. Several of the air pollutants have been
c taken as specified by the New Source Performance Standards (NSPS)
c but provision has also been made to model additional control to
c hold the effluents below the NSPS. The degree of urban sewage
c treatment (none, primary, secondary) is considered to be a
c function of the public service index calculated in the
c demographic sector.
c
c For ease of presentation, the effluent residuals are normalized
c to nominal values, which, it should be noted, are not meant to be
c viewed as acceptable or unacceptable values. These normalized
c values are also not equivalent to relative ambient concentrations
c and are no substitute for analyses of site specific ambient
c concentrations. Moreover, these normalized and averaged values
c underestimate impacts primarily because they do not recognize the
c nonlinearities, synergistic effects, or the critical near-source
c "hot spots."
c
c This sector also calculates an average population growth rate
c which may be considered as a measure of local social and
c political disruption. Gilmore and Duff (1976) suggest that a 5
c percent annual growth rate is about as much as a small community
c can manage without experiencing the ill effects of boom-town
c growth. The model also calculates an urban/industrial land
c impact parameter proportional to the land area in urban,
c industrial, and mining use. This parameter indicates roughly the
c degree of change in traditional land use.
c
c In addition to the population growth impact and the land impact
c parameters, two other indices are calculated for use in the
c political sector for the calculation of a political environmental
c quality index. One is a coal conversion air impact area index,
c which is a measure of the portion of the area affected by stack
c emissions, and the other is a reciprocal instream flow index.
c The latter is a measure of how well the desired instream flows
c are being met and perhaps a measure of stream assimilative
c capacity.
c
c It would be very desirable to derive ambient residual
c concentrations from the residuals emission data and calculate
c direct impacts that could then be incorporated into political
c environmental quality index. However, until this is possible,
```

```
c the residuals are simply left in the form of separate  normalized
c values.
c
c ******************** start program statements ********************
c
      subroutine wcrenv(envgo)
      integer envgo
      common/system/v(1000),c(1000),tc(1000),t1
      common time
      common/dt/dt
c (see note concerning analogous indgo in industry sector description)
      if(envgo.eq.0) return
      if(envgo.gt.0) go to 10
      if(envgo.lt.0) envgo=envgo+1
10    continue
c
c fraction of new source performance standards (nsps) emitted
      v(501)=c(501)
c fraction controllable water-borne residuals promptly emitted
      v(502)=c(502)
c fraction controlled water-borne residuals ultimately emitted
      v(503)=c(503)
c sewage treatment coefficient
      v(504)=0.3+0.7/(1.0+exp((v(228)-0.8)/0.1))
c ---------------------------------------------
c
c residuals (water)            tons/yr
c
c sediment runoff from construction
      v(505)=v(28)*c(505)+v(68)*c(506)+v(86)*c(507)+v(832)*c(508)
c sediment runoff from operations
      v(506)=v(802)*c(509)+(v(20)*c(510)+v(59)*c(511))*v(502)
c delayed sediment runoff from operations
      v(507)=delay3('dsro',(v(20)*c(510)+v(59)*c(511))*(1.0-v(502))*v(50
     13),tc(507),v(508))
c sediment from mined/reclaimed lands
      v(509)=(v(316)+v(318)+v(320))*c(512)+(v(365)+v(369))*c(513)+(v(366
     1)+v(367)+v(368))*c(514)+v(373)*c(515)
c total sediment runoff
      v(510)=c(570)+v(505)+v(506)+v(507)+v(509)
c ---------------------------------------------
c dissolved solids
c current operations effluent
      v(511)=v(802)*c(571)+v(59)*(c(572)+c(573)*v(502))+v(20)*c(574)*v(5
     102)
c delayed operations effluent
      v(512)=delay3('doe',(v(59)*c(573)+v(20)*c(574))*(1.0-v(502))*v(503
     1),tc(512),v(513))
c sewage effluent
      v(514)=v(504)*v(227)*c(575)
c mined lands leaching
      v(515)=v(374)*c(516)
c total dissolved solids effluent
```

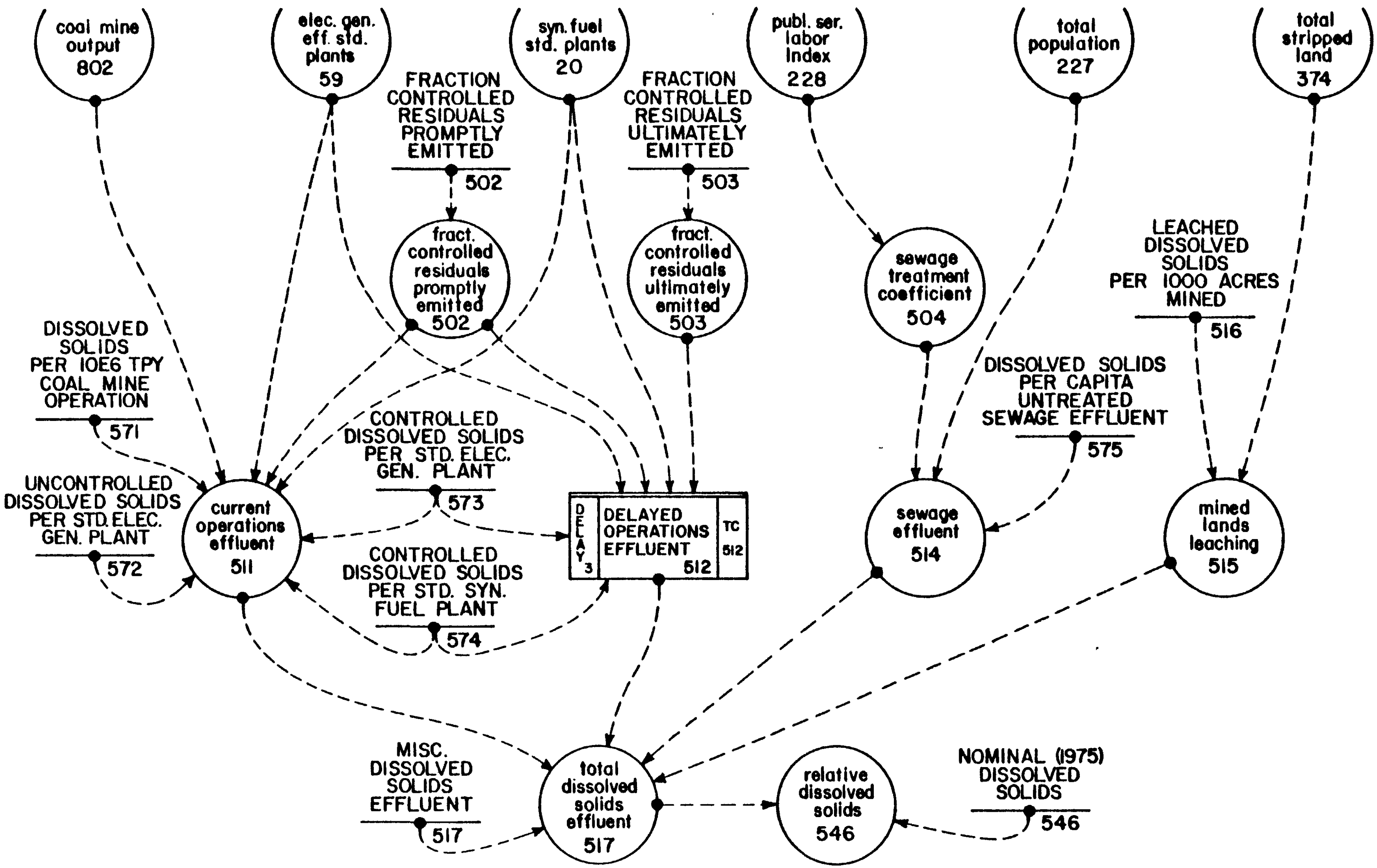
```
      v(517)=c(517)+v(511)+v(512)+v(514)+v(515)
c ----------------------------------------------
c biochemical oxygen demand (bod)
      v(518)=c(518)+v(20)*v(502)*c(519)+v(227)*v(504)*c(520)
c organics
      v(521)=c(521)+(v(20)*c(522)+v(59)*c(523))*v(502)+delay3('org',(v(2
     10)*c(522)+v(59)*c(523))*(1.0-v(502))*v(503),tc(521),v(522))
c heavy metals
      v(525)=c(525)
c ----------------------------------------------
c residuals (air)                tons/yr
c
c so2
      v(530)=c(530)+(v(20)*c(531)+v(59)*c(532))*v(501)
c nox
      v(533)=c(533)+v(20)*c(534)+v(59)*c(535)+c(542)*v(227)
c hydrocarbons
      v(536)=c(536)+(v(20)*c(537)+v(59))*c(538)+c(543)*v(227)
c particulates
      v(539)=c(539)+(v(20)*c(540)+v(59)*c(541))*v(501)
c
c relative residuals normalized to nominal (1975) values
c
c sediment
      v(545)=v(510)/c(545)
c fractional sediment
      v(554)=v(545)-1.0
c dissolved solids
      v(546)=v(517)/c(546)
c bod
      v(547)=v(518)/c(547)
c fractional dissolved solids
      v(555)=v(546)-1.0
c organics
      v(548)=v(521)/c(548)
c heavy metals
      v(549)=v(525)/c(549)
c so2
      v(550)=v(530)/c(550)
c nox
      v(551)=v(533)/c(551)
c hydrocarbons
      v(552)=v(536)/c(552)
c particulates
      v(553)=v(539)/c(553)
c ----------------------------------------------
c relative or fractional indices
c
c relative instream flow
      v(560)=v(152)/c(106)
c reciprocal instream flow index
      if(v(137).ne.0.0) v(569)=1.0/v(137)-1.0
c fractional sediment/water index
```

```
      v(561)=1.0
      if(v(560).ne.0.0) v(561)=(v(545)-1.0)/v(560)
c coal conversion air impact area index
      v(564)=((v(59)*c(564)+v(20)*c(565))*v(501)**2)/(v(399)/0.64)
c
c urban-industrial impacted lands     thousand acres
      v(566)=(v(360)+v(364)+v(365)+v(369)+v(373)+v(316)+v(318)+v(320))*c
     1(566)
c relative impacted land index, normalized to nominal (1975) value
      v(567)=v(566)/c(567)
c fractional impacted lands
      v(568)=v(566)/v(399)
c
c population rate of change calculation
      if(v(570).eq.0.0) v(570)=v(227)
c instantaneous fractional growth rate      people per year
      v(571)=(v(227)-v(570))/(v(227)*dt)
c average fractional pop. growth rate      people per year
      v(572)=smooth('v571',v(571),tc(572))
c 'previous population'
      v(570)=v(227)
c
      return
      end
```

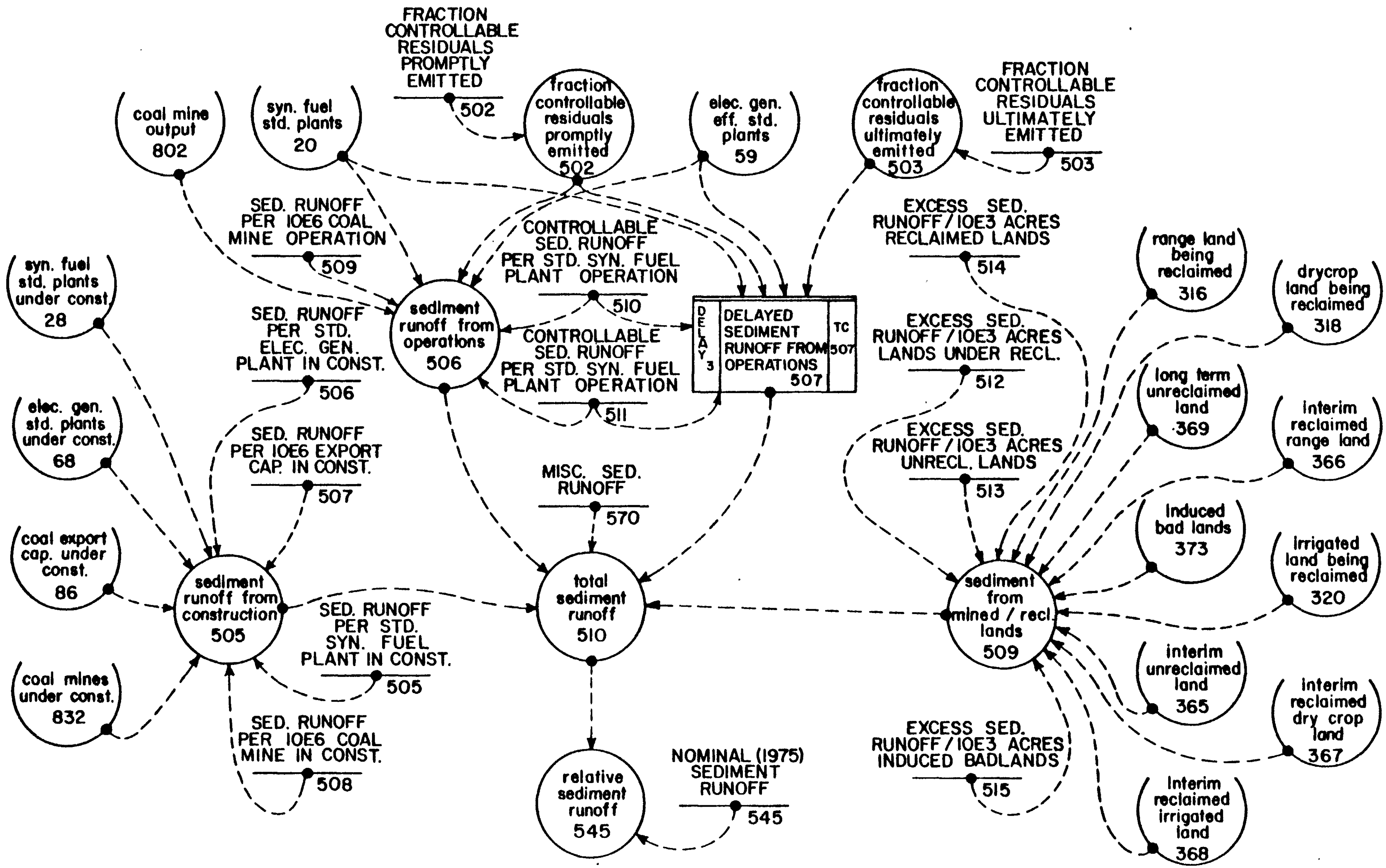



AIR AND WATER RESIDUALS SUBSECTOR

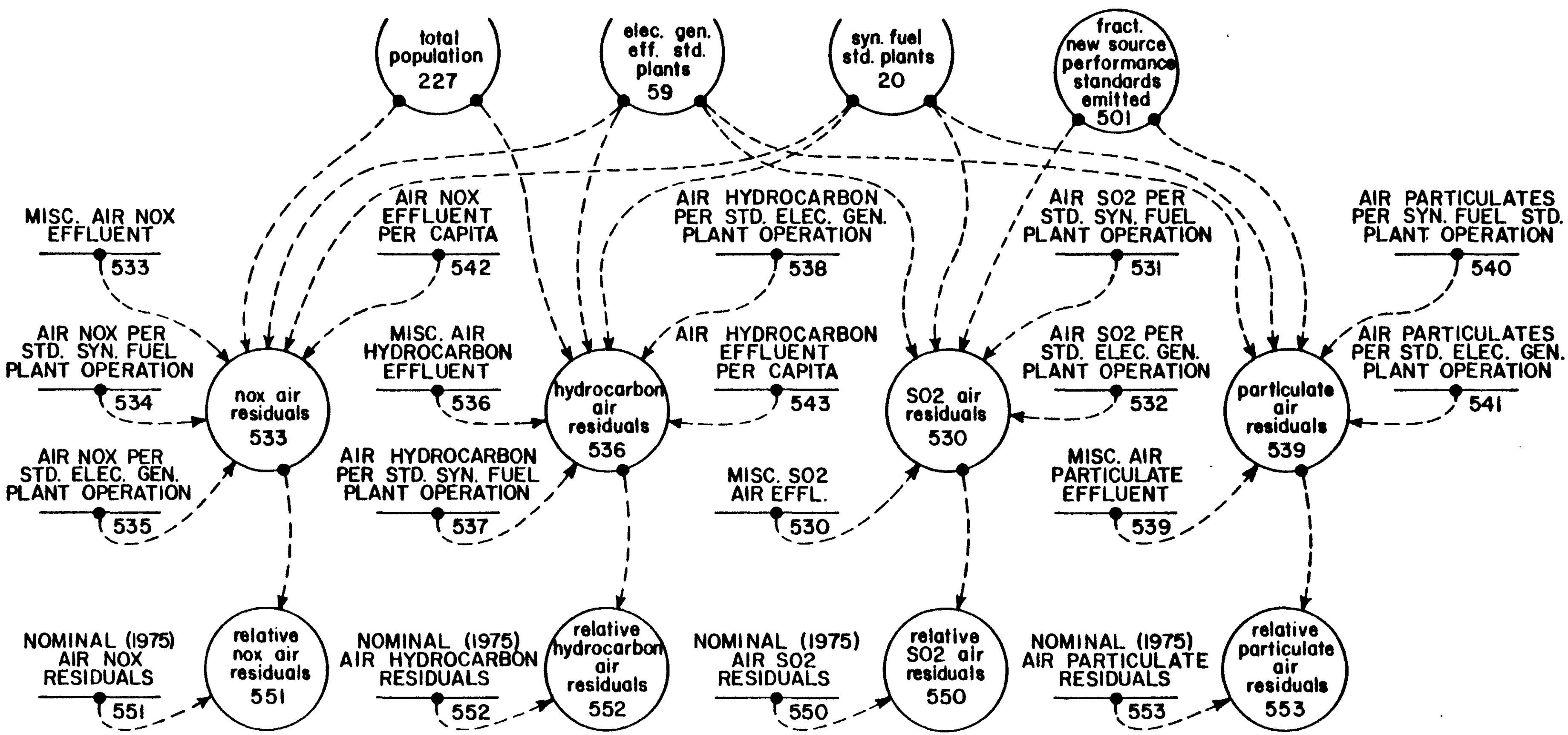
# DISSOLVED SOLIDS EFFLUENT SUBSECTOR



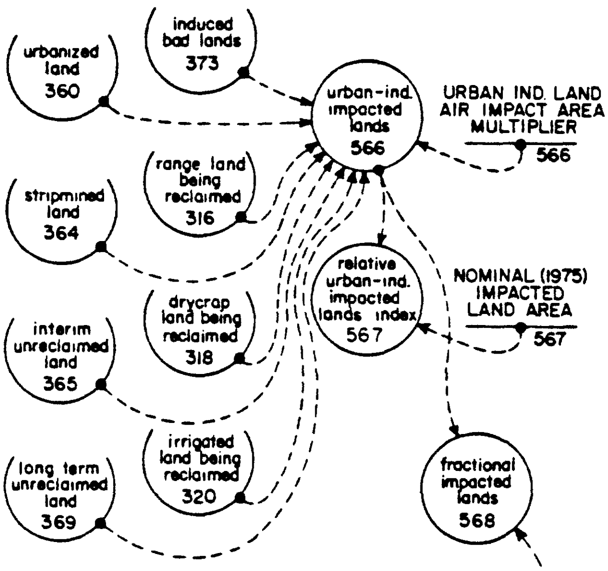
# SEDIMENT RUNOFF RESIDUALS SUB-SECTOR



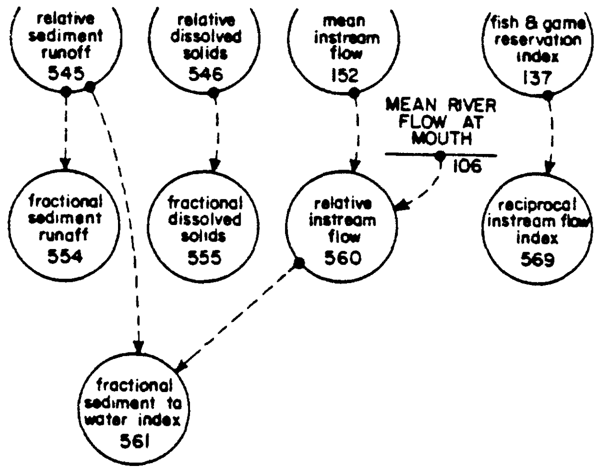
# AIR RESIDUALS SUB-SECTOR



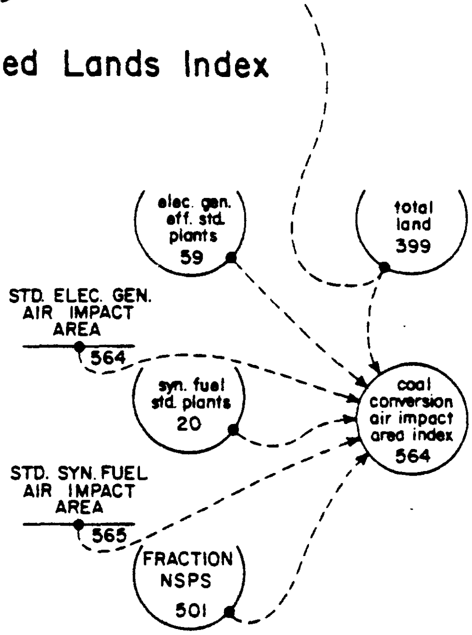
# ENVIRONMENTAL IMPACT INDEX SUBSECTOR



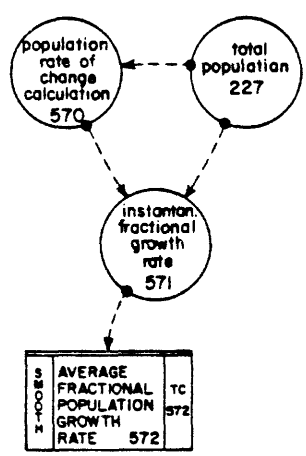

Impacted Lands Index




Water Impact Indices




Coal Conversion Air Impact Area Index




Population Change Index

```
c      ***************** Constants List (08-26-77)*************************
c
c Some of these constants are based upon little or no data.
c
c all time constants (tc) in years.
c
      subroutine wcrco
      common/system/v(1000),c(1000),tc(1000),t1
c
c rate at which plant ordering decisions are made  years
      t1=1.0
c
c            abbreviations list
c
c 1.0e6      million
c syn.       synthetic
c tpy        tons per year  .
c mscfpd     million standard cubic feet per day
c btu        british thermal unit
c lb         pound
c elec.      electrical
c gen.       generation
c mwe        megawatt electric
c 1.0e3      thousand
c af         acre-feet
c std.       standard
c misc.      miscellaneous
c ac         acres
c sed.       sediment
c ds         dissolved solids
c eis        environmental impact statement
c bod        biochemical oxygen demand
c org.       organics
c so2        sulpher dioxide
c nox        nitrous oxides
c part.      particulates
c nsps       new source performance standards
c agri.      agriculture
c
c -----------------------------------------------------
c industrial constants (energy in million tons coal)
c
c syn. fuel exogenous scenario modification value
      c(2)=1.0
c syn. fuel nominal capacity utilization
      c(3)=0.7
c synthetic fuel maximum ordering rate   million tpy/year
      c(5)=35.0
c syn. fuel standard plant coal use
c million tpy/250 mscfpd std. plant at 9000 btu/lb
      c(20)=8.0
c syn. fuel planning horizon time   years
      tc(3)=8.0
```

```
c syn. fuel plant average lifetime   years
      tc(15)=25.0
c syn. fuel maximum regulatory delay   years
      tc(21)=6.0
c syn. fuel minimum facility planning time   years
      tc(22)=1.0
c syn. fuel nominal construction time   years
      tc(26)=3.0
c ---------------------------------------------------
c elec. gen. exogenous scenario modification value
      c(36)=1.0
c elec. gen. nominal capacity utalization (new plants)
      c(38)=0.8
c electrical generation maximum ordering rate   million tpy/year
      c(40)=4.2
c elec. gen. standard plant coal use
c million tpy/1000 mwe std. plant at 9000 btu/lb
      c(54)=4.0
c elec. gen. planning horizon time   years
      tc(38)=8.0
c elec. gen. maximum regulatory delay time   years
      tc(55)=6.0
c elec gen. minimum facility planning time   years
      tc(56)=1.0
c elec. gen. nominal construction time   years
      tc(61)=3.0
c ---------------------------------------------------
c coal export exogenous scenario modification value
      c(72)=1.0
c coal export maximum ordering rate   million tpy/year
      c(75)=30.0
c fraction exported by slurry pipeline
      c(89)=0.0
c coal export facilities average life time   years
      tc(81)=40.0
c coal export planning horizon time   years
      tc(73)=7.0
c coal export maximum regulatory delay   years
      tc(86)=6.0
c coal export minimum facility planning time   years
      tc(87)=1.0
c coal export nominal construction time   years
      tc(91)=2.0
c ---------------------------------------------------
c original coal resources   million tons
      c(851)=60000.6
c average mine utilization time constant   years
      tc(804)=3.0
c coal mine average lifetime   years
      tc(811)=40.0
c coal mine maximum regulatory delay   years
      tc(818)=4.0
c coal mine minimum facility planning time   years
```

```
      tc(819)=1.0
c coal mine nominal construction time   years
      tc(826)=1.0
c coal production shortage averaging time   years
      tc(821)=5.0
c coal extraction rate averaging time   years
      tc(845)=5.0
c static researve index averaging time   years
      tc(848)=5.0
c -----------------------------------------------------
c eis region fraction of coal development profile
      c(6)=0.39
c river basin fraction of cdp
      c(7)=0.44
c ------------------------------------------------------
c water sector constants (water in 1.0e3 acre-feet)
c
c storage construction initiation
      tc(101)=1.0
c storage construction
      tc(102)=5.0
c sediment filling
      tc(104)=100.0
c effective stream flow averaging time
      tc(109)=3.0
c yield deficit perception
      tc(120)=5.0
c approx. mean river flow at mouth (at 1970 diversion level)
      c(106)=8800.0
c evaporation/storage ratio
      c(110)=0.03
c mean annualized storage cost [$/af/yr] at 7.35 discount rate
      c(121)=13.5
c misc. shortage in 1970
c in Yellowstone Basin
      c(123)=0.
c misc. consumptive use increase (1970-2000) in drainage basin
      c(124)=140.
c consumptive use shortage for irrigation (in 1970)
c in drainage basin
      c(126)=0.
c increase in consumptive use (1970-2000) for irrigation
c in drainage basin
      c(125)=540.
c Fermi function "b" for reservior utility multiplier
      c(127)=1.5
c Fermi function "a"
      c(128)=0.1
c fraction industrial water diverted within drainage basin
      c(130)=1.0
c water (1.0e3 af/yr)/1000 mwe plant
      c(131)=15.0
c water (1.0e3 af/yr)/250 mscfpd syn. fuel plant
```

```
      c(132)=10.0
c water/mining  (1.0e3 af/1.0e6 tons coal)
      c(133)=0.05
c water/export by slurry pipeline   (1.0e3 af/1.0e6 tons coal)
      c(134)=0.7
c nominal mean annual instream flow reservation
      c(136)=4000.0
c planning weight for industry
      c(146)=1.0
c planning weight for misc.
      c(147)=1.0
c planning weight for agriculture
      c(148)=1.0
c 'marginal value' of water at 0% deficit ($/af)
      c(161)=5.0
c 'marginal value' of water at 100% deficit
      c(162)=300.0
c -------------------------------------------------------
c demographic sector constants   (workers/1.0e6 tons coal/year)
c
c operations workers/1000 mwe plant
      c(201)=100.0
c operations workers/250 mscfpd syn. fuel plant
      c(202)=600.0
c operations workers/1.0e6 tpy coal exported
      c(203)=3.8
c operations workers/1.0e6 tpy mined
      c(204)=25.0
c construction worker-years/1000 mwe plant
      c(205)=3500.0
c construction worker-years/250 mscfpd syn. fuel plant
      c(206)=5400.0
c construction worker-years/1.0e6 tpy coal export capacity
      c(207)=40.0
c construction worker-years/1.0e6 tpy coal mine capacity
      c(208)=20.0
c private service/agri. labor
      c(209)=0.99
c private service indian industrial operations labor
      c(210)=0.99
c private service/nonindian industrial operations labor
      c(211)=0.99
c private service/nonindian industrial construction labor
      c(212)=0.99
c private service/indian industrial construction labor
      c(213)=0.99
c public service/agri. labor
      c(214)=0.38
c public service/indian industrial operations labor
      c(215)=0.38
c public service/nonindian industrial operations labor
      c(216)=0.38
c public service/nonindian industrial construction labor
```

```
      c(217)=0.38
c public service/indian industrial construction labor
      c(218)=0.38
c agri. labor family multiplier
      c(219)=2.77
c nonindian operations labor family multiplier
      c(220)=2.0
c public service family multiplier
      c(222)=2.77
c private service family multiplier
      c(223)=2.77
c nonindian construction labor family multiplier
      c(224)=2.3
c indians in industry
      c(225)=0.1
c private service/other primary labor
      c(227)=0.99
c public service/other primary labor
      c(228)=0.38
c other primary labor family multiplier
      c(229)=2.0
c other primary labor-indiginous
      c(230)=1000.0
c urban fraction of non-indian & non-agri.
      c(231)=.66
c nonindian construction labor in-migration time
      tc(217)=1.0
c nonindian construction labor out-migration time
      tc(218)=1.0
c indian construction labor in-transfer time
      tc(219)=1.0
c indian construction labor out-transfer time
      tc(220)=1.0
c nonindian operation labor in-migration time
      tc(221)=1.0
c nonindian operation labor out-migration time
      tc(229)=1.0
c indian operation labor in-transfer time
      tc(222)=1.0
c indian operation labor out-transfer time
      tc(230)=1.0
c private service labor time constant
      tc(224)=1.0
c public service labor time constant
      tc(226)=3.0
c boom index smoothing time
      tc(234)=.5
c ---------------------------------------------------
c land sector constants (land in 1.0e3 acres)
c
c stripped land/coal ratio (1.0e3 acres/1.0e6 tons coal)
      c(300)=0.024
c land (1.0e3 ac)/1000 mwe plant
```

```
      c(301)=0.5
c land (1.0e3 ac)/250 mscfpd syn. fuel plant
      c(302)=1.0
c land (1.0e3 ac)/1.0e6 tpy mining capacity
      c(303)=0.15
c urban land/urban population (1.0e3 ac/capita)
      c(304)=0.0003
c rangeland urbanization fraction
      c(305)=0.91
c dry crop land urbanization fraction
      c(306)=0.08
c irrigated land urbanization fraction
      c(307)=0.01
c rangeland mining fraction
      c(308)=0.91
c dry crop land mining fraction
      c(309)=0.08
c irrigated land mining fraction
      c(310)=0.01
c non-reclamation fraction
      c(311)=0.20
c reclamation to rangeland fraction
      c(312)=0.70
c reclamation to dry crop land fraction
      c(313)=0.05
c reclamation to irrigated land fraction
      c(314)=0.05
c unreclaimed to badlands fraction
      c(328)=0.75
c 'reclaimed rangeland' to badlands fraction
      c(329)=0.25
c 'reclaimed dry crop land' to badlands fraction
      c(330)=0.10
c 'reclaimed irrigated land' to badlands fraction
      c(331)=0.05
c
c time constants
c
c reclamation initiation
      tc(311)=1.0
c rangeland reclamation
      tc(315)=5.0
c dry crop land reclamation
      tc(317)=5.0
c irrigated land reclamation
      tc(319)=5.0
c unreclaimed/badlands
      tc(328)=10.0
c 'reclaimed rangeland'/badlands
      tc(329)=25.0
c 'reclaimed dry crop land'/badlands
      tc(330)=25.0
c 'reclaimed irrigated land'/badlands
```

```
      tc(331)=30.0
c -----------------------------------------------------
c political sector constants  (1.0e6 dollars/1.0e6 tons coal)
c
c relative agri. revenue
      c(401)=2.0
c coal severance tax rate
      c(402)=0.3
c 1970 political unit revenue
      c(403)=500.0
c capital ($1.0e6)/1000 mwe plant
      c(404)=250.0
c capital ($1.0e6)/250 mscfpd synthetic fuel plant
      c(405)=450.0
c export capital/output ratio  ($1.0e6/1.0e6 tpy coal)
      c(406)=0.0
c mining capital/output ratio  ($1.0e6/1.0e6 tpy coal)
      c(407)=0.7
c capital normalizer
      c(408)=1000.0
c conversion / lifestyle weight
      c(416)=0.8
c mining / lifestyle weight
      c(417)=0.2
c lifestyle perception time
      tc(409)=15.0
c agri. revenue index weight
      c(409)=7.0
c coal revenue index weight
      c(410)=10.0
c industrial capital index weight
      c(411)=10.0
c perceived industrial climate index weight
      c(412)=5.0
c environmental index weight
      c(413)=5.0
c future shock index weight
      c(414)=7.0
c energy gap index weight
      c(415)=8.0
c political adjustment time
      tc(414)=3.0
c political perception time
      tc(415)=2.0
c political climate optional constant
      c(420)=0.0
c
c ------------------------------------------------
c
c environmental sector constants  (tons/year)
c
c elec. gen. std. plant: 1000 mwe at 80% load factor
c syn. fuel std. plant: 250 mmcfd at 70% load factor
```

```
c
c fraction of new source performance stds. (nsps) emitted
      c(501)=1.0
c fraction controllable water-borne residuals promptly emitted
      c(502)=0.1
c fraction controlled water-borne residuals ultimately emitted
      c(503)=0.5
c
c ---------------------------------------------------
c sediment runoff          tons/year
c sed. per syn. fuel std. plant under construction
      c(505)=0
c sed. per elec. gen. std. plant under construction
      c(506)=0
c sed. per 1.0e6 tpy transportation facility under construction
      c(507)=0
c sed. per 1.0e6 tpy coal mines under construction
      c(508)=0
c sed. per syn. fuel std. plant operation ('controllable')
      c(510)=57.
c sed. per elec. gen. std. plant operation ('controllable')
      c(511)=438.
c sed. per 1.0e6 tpy coal mine operation
      c(509)=950.
c delay time for 'controlled' sediment    years
      tc(507)=20.
c excess sed. per 1.0e3 acres mined/reclaimed lands
c lands undergoing reclamation
      c(512)=5.e3
c unrecaimed lands
      c(513)=6.e3
c reclaimed lands
      c(514)=0.e3
c induced badlands
      c(515)=10.e3
c misc. sediment runoff (eis region)
      c(570)=8.6e6
c
c --------------------------------------------------
c dissolved solids (ds)      tons/year
c ds per 1.0e6 tpy coal mine operation
      c(571)=620.
c ds per elec. gen. std. plant operation ('uncontrollable')
      c(572)=252.
c ds per elec. gen. std. plant operation ('controllable')
      c(573)=3758.
c ds per syn. fuel std. plant operation ('controllable')
      c(574)=2722.
c delay time for 'controlled' dissolved solids   years
      tc(512)=2.
c ds per capita, untreated sewage effluent
      c(575)=0.05
c ds per 1.0e3 acres mined lands leaching
```

```
      c(516)=0
c misc. ds effluent from eis region
      c(517)=6.6e6
c
c ---------------------------------------------
c biochemical oxygen demand (bod)     tons/year
c misc. bod effluent (eis region)
      c(518)=0
c bod per syn. fuel std. plant operation ('controllable')
      c(519)=1.8
c bod per capita, untreated sewage effluent
      c(520)=0.03
c
c ---------------------------------------------
c organics          tons/year
c misc. organic effluent (eis region)
      c(521)=0
c org. per syn. fuel std. plant operation ('controllable')
      c(522)=27.
c org. per elec. gen. std. plant operation ('controllable')
      c(523)=58.
c delay time for 'controlled' organics    years
      tc(521)=20.
c
c ---------------------------------------------
c heavy metals        tons/year
c misc. heavy metals effluent (eis region)
      c(525)=0
c
c ---------------------------------------------
c so2 effluent       tons/year
c misc. so2 effluent (eis region)
      c(530)=0
c so2 per syn. fuel std plant operation
      c(531)=1800
c so2 per elec. gen. std. plant operation
      c(532)=45600.
c
c ---------------------------------------------
c nox effluent       tons/year
c misc. nox. effluent (eis region)
      c(533)=0
c nox per syn. fuel std. plant operation
      c(534)=3900.
c nox per elec. gen. std. plant operation
      c(535)=33600.
c nox per capita
      c(542)=0
c
c ---------------------------------------------
c hydrocarbons        tons/year
c misc. hydrocarbon effluent (eis region)
c c(536)=0
```

```
c hydrocarbons per syn. fuel std. plant operation
      c(537)=132.
c hydrocarbons per elec. gen. std. plant operation
      c(538)=528.
c hydrocarbons per capita
      c(543)=0
c
c ----------------------------------------
c particulates       tons/year
c misc. particulate effluent (eis region)
      c(539)=0
c part. per syn. fuel std. plant operation
      c(540)=412.
c part. per elec. gen. std. plant operation
      c(541)=20000.
c
c ----------------------------------------
c nominal (1975) values for residual normalization    tons/year
c sediment
      c(545)=.86e7
c dissolved solids
      c(546)=.66e7
c bod
      c(547)=.46e3
c organics
      c(548)=7.1
c heavy metals
      c(549)=1
c so2
      c(550)=.14e5
c nox
      c(551)=.11e5
c hydrocarbons
      c(552)=.17e3
c particulates
      c(553)=.63e4
c
c ---------------------------------------------
c impact area for air pollution    square miles
c std. syn. fuel plant
      c(565)=3600.
c std. elec. gen. plant ( at nsps)
      c(564)=3600.
c
c impact area multiplier for urban-industrial lands  acre/acre
      c(566)=1.5
c nominal (1975) impacted land area    1.0e3 acres
      c(567)=.11e2
c
c population growth rate smoothing time constant
      tc(572)=5.
      return
      end
```

```
c **************** Initial Levels (08-26-77) ****************
c
      subroutine wcrin
c
      common/system/v(1000),c(1000),tc(1000),t1
c
c     industrial sector  (million tons coal)
c
c coal reserves
      v(842)=26600.0
c coal resources remaining
      v(844)=34000.0
c initial time weighted electrical generation capacity
      v(50)=0.0
c
c     demographic sector  (number of workers)
c
c nonindian operations labor
      v(207)=300.0
c nonindian construction labor
      v(209)=75.0
c
c     water sector  (thousand acre-feet)
c
c mean instream flow
      v(102)=delay3('v101',0.0,tc(102),v(103))
c storage capacity
      v(105)=0.0
c mean instream flow reservation
      v(136)=c(136)
c sediment filled storage
      v(165)=0.0
c
c     land sector (thousand acres)
c
c urbanized land
      v(360)=6.5
c (indicated urban land)
      v(303)=v(360)
c range land
      v(361)=9389.0
c dry crop land
      v(362)=1980.0
c irrigated land
      v(363)=184.0
c stripmined land
      v(364)=0.0
c reclamation land
      v(365)=0.0
      v(366)=0.0
      v(367)=0.0
      v(368)=0.0
      v(369)=0.0
```

```
c induced badlands
      v(373)=0.0
c
      return
      end
```

```
c ****************Real Function Energy (06-28-77)*************
c
c Values of ncdp less than 50 are routed to polynomial
c calculation.  Values of ncdp greater than 50 are
c routed to the table look-up calculation
c
      real function energy(ntyp,ncdp,etime)
      common/system/v(1000),c(1000),tc(1000),t1
      common rtime
      common/energy/ctime
c
c
c rtime is real time, ctime is clip time, etime is energy planning
c time, and time is a local variable.
      real syn1(11),syn2(11),syn3(11),syn4(11)
      real ele1(11),ele2(11),ele3(11),ele4(11)
      real exp1(11),exp2(11),exp3(11),exp4(11),exp5(18)
c
      data syn1/'syn1',5.0,1970.0,2000.0,0.0,0.0,0.0,0.0,2.5,5.0,7
     1.6/
      data syn2/'syn2',5.0,1970.0,2000.0,0.0,0.0,0.0,23.0,30.3,37.
     17,45.0/
      data syn3/'syn3',5.0,1970.0,2000.0,0.0,0.0,0.0,61.0,67.8,74.
     16,81.4/
      data syn4/'syn4',5.0,1970.0,2000.0,0.0,0.0,0.0,61.0,78.7,96.
     13,114.0/
      data ele1/'ele1',5.0,1970.0,2000.0,0.8,1.2,3.8,3.8,3.8,3.8,3
     1.8/
      data ele2/'ele2',5.0,1970.0,2000.0,0.8,1.2,3.8,5.4,9.6,13.9,
     118.1/
      data ele3/'ele3',5.0,1970.0,2000.0,0.8,1.2,9.4,9.4,11.3,13.1
     1,15.0/
      data ele4/'ele4',5.0,1970.0,2000.0,0.8,1.2,9.4,11.0,15.2,19.
     15,23.7/
      data exp1/'exp1',5.0,1970.0,2000.0,6.1,19.6,41.1,41.4,41.4,4
     11.4,41.4/
      data exp2/'exp2',5.0,1970.0,2000.0,6.1,19.6,41.4,49.0,55.3,6
     11.7,68.0/
      data exp3/'exp3',5.0,1970.0,2000.0,6.1,19.6,41.4,49.0,88.5,1
     128.0,167.5/
      data exp4/'exp4',5.0,1970.0,2000.0,6.1,19.6,41.4,89.0,145.7,
     1202.3,259.0/
      data exp5/'exp5',1.0,1970.0,1983.0,3.5,7.0,10.5,14.0,17.5,21.0,
     124.0,27.0,28.0,29.0,61.0,61.0,96.0,96.0/
c
c if ctime > 1970 then scenarios are clipped after rtime > ctime
c to scenario value at ctime.
c
      time=etime
      if(ctime.gt.1970.0.and.rtime.gt.ctime) time=amin1(etime,ctime)
c
      if(ncdp.gt.50) go to 1000
c
```

```
c polynomial calculations ------------------------
c
      e=0.0
      goto(100,200,300),ntyp
c synthetic fuel
100   goto(110,120,130,140,150,160),ncdp
110   a=.727161
      b=.851666
      cc=-3.54572e-2
      d=5.19651e-4
      goto 4000
120   a=1.26895
      b=.314412
      cc=4.86387e-2
      d=-1.17682e-3
      goto 4000
130   a=3.39982
      b=-1.62899
      cc=.32283
      d=-6.00905e-3
      goto 4000
140   a=6.24355
      b=1.49346
      cc=0.328828
      d=-1.87009e-2
      e=3.04421e-4
      goto 4000
150   goto 160
160   a=6.04196
      b=1.63108
      cc=0.324294
      d=-2.08284e-2
      e=4.29213e-4
      goto 4000
c electric generation
200   goto(210,220,230,240,250,260),ncdo
210   a=.397743
      b=.307076
      cc=-4.89236e-3
      d=0.0
      goto 4000
220   a=.129894
      b=.561536
      cc=-.040981
      d=9.70258e-4
      goto 4000
230   a=.622447
      b=.253134
      cc=-2.11733e-2
      d=6.42339e-4
      goto 4000
240   a=1.57729
      b=9.31602
```

```
      cc=-0.617522
      d=1.76564e-2
      e=-1.80021e-4
      goto 4000
250   a=3.50975
      b=5.98806
      cc=0.167302
      d=-1.92489e-2
      e=3.66135e-4
      goto 4000
260   if(time.lt.1980.12)goto 250
      a=-78.7665
      b=3.28419
      cc=1.5848
      d=-5.79683e-2
      e=7.96009e-4
      goto 4000
c export
300   goto(310,320,330,340,350,360),ncdp
310   a=0.0
      b=0.0
      cc=0.0
      d=0.0
      goto 4000
320   a=-9.30001
      b=1.12333
      cc=-1.93334e-2
      d=0.0
      goto 4000
330   a=-2.37182
      b=.574696
      cc=-8.79558e-3
      d=0.0
      goto 4000
340   a=0.0
      b=0.0
      cc=0.0
      d=0.0
      e=0.0
      goto 4000
350   a=-201.346
      b=28.3359
      cc=-0.934234
      d=1.14103e-2
      e=0.0
      goto 4000
360   a=-38.9533
      b=3.31451
      cc=0.882511
      d=-2.04806e-2
      e=0.0
      goto 4000
c
```

```
c ---------- table look-up calculation -----------
c
1000  continue
c
c
      new=ncdp-50
      go to (1200,1300,1100),ntyp
c synthetic fuel
1100  go to (1110,1120,1130,1140,1150,1160),new
1110  energy=tlu(syn1,time)
      return
1120  energy=tlu(syn2,time)
      return
1130  energy=tlu(syn3,time)
      return
1140  energy=tlu(syn4,time)
      return
1150  energy=0.0
      return
1160  energy=0.0
      return
c electric generation
1200  go to (1210,1220,1230,1240,1250,1260),new
1210  energy=tlu(ele1,time)
      return
1220  energy=tlu(ele2,time)
      return
1230  energy=tlu(ele3,time)
      return
1240  energy=tlu(ele4,time)
      return
1250  energy=0.8
      return
1260  energy=0.8
      if(time.ge.1975.0) energy=4.3
      if(time.ge.1980.0) energy=10.8
      return
c export
1300  go to (1310,1320,1330,1340,1350,1360),new
1310  energy=tlu(exp1,time)
      return
1320  energy=tlu(exp2,time)
      return
1330  energy=tlu(exp3,time)
      return
1340  energy=tlu(exp4,time)
      return
1350  energy=tlu(exp5,time)
      return
1360  energy=tlu(exp5,time)
      return
c
c
```

```
4000   t=time-1970.0
       if(t.gt.30.0)goto 5000
       y=a+b*t+cc*t*t+d*t*t*t+e*t*t*t*t
6000   if(ncdp.ge.4.and.ncdp.le.6)y=0.1*y
       energy=c(6)*10.0*amax1(y,0.0)
       return
5000   tt=a+b*30.0+cc*900.0+d*27000.0+e*810000
       dd=b+cc*60.0+d*2700.0+e*108000
       y=tt+dd*(t-30.0)
       goto 6000
       end



c ****************Real Function Totcol (04-08-76)**********
c
       real function totcol (ncdp,etime)
       totcol=energy(1,ncdp,etime)+energy(2,ncdp,etime)
      1+energy(3,ncdp,etime)
       return
       end
```

```
c ********** Table Look Up (real function tlu, 9-25-75)**********
c
c tlu(t,x) should be in the form tlu('name',t(2),t(3),t(4),
c t(5),...,x)  where t(2) is the abscissa interval between
c successive data points, t(3) and t(4) are the lower and
c upper ranges of the abscissa, and t(5),... are the
c ordinate values starting at t(3).  x is the input variable
c for which the value of tlu is desired.
c
      real function tlu(t,x)
      real x,t(1),a
      common /t/ tnam(35),tmxx(35),ttim(35),tcnt(35)
      common time
      if(x-t(3)) 15,10,20
10    tlu=t(5)
      return
15    write(6,16) x,t(1)
16    format('  x=',e10.4,' below range of table ',a4)
      pause 'tlu'
      return
20    if(x-t(4)) 55,50,25
25    do 40 i=1,35
      if(tnam(i)) 35,30,35
30    tnam(i)=t(1)
      tmxx(i)=x
      ttim(i)=time
      tcnt(i)=1
      goto 50
35    if(t(1) .eq. tnam(i)) goto 45
40    continue
      write(6,41)
41    format(' capacity of tlu error tables exceeded')
      pause 'tlu'
      goto 50
45    tcnt(i)=tcnt(i)+1
      if(x .gt. tmxx(i)) tmxx(i)=x
50    i=(t(4)-t(3))/t(2)+1.0e-2
      i=i-1
      goto 60
55    i=(x-t(3))/t(2)
60    a=i*t(2)+t(3)
      i=i+5
      tlu=t(i)+(t(i+1)-t(i))*(x-a)/t(2)
      return
      end
```

```
c ************ Subroutine Graph (06-07-77) ************
c
c It scales the variables and plots them.  It is a modified
c version of a routine written by Patrick C. Doherty of the
c Computer Center Division
c
      subroutine graph
      common/graph/put(10),max(10),min(10),flo(10),fup(10),opsym(10),
     1ii(10),nput
      common /t/ tnam(35),tmxx(35),ttim(35),tcnt(35)
      implicit real (i,m)
      integer i,iplot,itemp,ii,opsym
      character*24 string
      double precision time
      real put,max,min,lolim(10),uplim(10)
      integer pline(61)
      real atem(5)
      write(6,99)
      write(21,99)
      opsym(2)="1"
      opsym(3)="2"
      opsym(4)="3"
      opsym(5)="4"
      opsym(6)="5"
98    format(1x,5i4/)
99    format(///)
      write(6,98) ii(1),ii(2),ii(3),ii(4),ii(5)
      write(21,98) ii(1),ii(2),ii(3),ii(4),ii(5)
      call clock_ (time)
      call date_time_ (time,string)
      write(6,122) string
      write(21,122) string
122   format (3x,a24/)
800   format(1x,2a4,2x,2a4)
      do 120 i=2,nput
      if(min(i).gt.0.0) min(i)=0.0
      if(fup(i).eq.0.0) goto 101
      max(i)=fup(i)
101   if(flo(i).eq.0.0) go to 102
      min(i)=flo(i)
102   a=(max(i)-min(i))/4.0
      if(a.le.0.0) a=1.0
      c=alog10(a)
      b=aint(c)
      if(b.gt.c) b=b-1.0
      a=1.0+aint(a*10.0**(-b))
      c=a
      if(c.gt.2.0) a=5.0
      if(c.gt.5.0) a=10.0
      if(min(i).ge.0.0) goto 103
      c=a*10.0**b
      d=aint(abs(max(i))/c)
      e=aint(abs(min(i))/c)
```

```
      f=d+e
      if(f.gt.2.0) a=a+a
103   a=a*10.0**b
c
      j=0
      if(min(i)) 104,110,106
104   if(j*a.le.min(i)) goto 110
      j=j-1
      goto 104
106   if(j*a.gt.min(i)) goto 108
      j=j+1
      goto 106
108   j=j-1
110   continue
      do 112 k=1,5
      atem(k)=j*a
112   j=j+1
      lolim(i)=atem(1)
      uplim(i)=atem(5)
      write(6,114) opsym(i),atem
      write(21,114) opsym(i),atem
114   format(1x,a1,2x,e8.2,4(7x,e8.2))
120   continue
130   do 200 iplot = 1,10
      read (20,end=205) (put(i),i=1,nput)
      if (iplot .eq. 1) go to 150
      do 140 i=1,61
140   pline(i) = " "
      pline(1) = "."
      pline(16) = "."
      pline(31) = "."
      pline(46) = "."
      pline(61) = "."
      go to 160
150   do 155 i=1,61
155   pline(i) = "."
160   continue
      itemp = put(1)+1.0e-2
      do 170 i=2,nput
      l=60.0*(put(i)-lolim(i))/(uplim(i)-lolim(i))
      if (l .gt. 60) l=60
170   pline(l+1) = opsym(i)
      if (iplot .eq. 1) go to 190
      write(6,180) (pline(i),i=1,61)
      write(21,180) (pline(i),i=1,61)
180   format(1x,6x,61a1)
      go to 200
190   write(6,195) itemp,(pline(i),i=1,61)
      write(21,195) itemp,(pline(i),i=1,61)
195   format(1x,i5,1x,61a1)
200   continue
      go to 130
205   continue
```

```
      go to 240
240   write(6,245)
      write(21,245)
245   format(////)
c     rewind 20
      end file 20
      do 340 i=1,35
      if(tnam(i) .eq. 0.0) goto 350
      write(6,330) tnam(i),tmxx(i),ttim(i),tcnt(i)
330   format(" exceeded table ",a4,"  max x=",e10.4,"  1st time=",
     1f6.0,"  count=",f5.0)
340   continue
350   continue
      return
      end
```

```
c *************** Subroutine Indic (10-14-77) ***************
c
c initializes dictionary for delay functions
c
      subroutine indic
      common/sm/ansm(35),arsm(35),alsm(35)
      common/dl1/andl1(35),aldl1(35)
      common/dl3/andl3(35),al1dl3(35),al2dl3(35),al3dl3(35),ar1dl3(35),
     1ar2dl3(35),ar3dl3(35)
      common/de3/ande3(35),al1de3(35),al2de3(35),al3de3(35),ar1de3(35),
     1ar2de3(35),ar3de3(35)
      do 10 i=1,35
      ansm(i)=0.0
      andl1(i)=0.0
      andl3(i)=0.0
      ande3(i)=0.0
10    continue
      return
      end


c ********************Subroutine Output (05-02-77)****************
c
c sets output variables for plotting
c uses system command read_list_$prompt
c requires link >sss>read_list_ prompt
c use n=0 for prompting, n=9 for repeat
c other values of n can be preset
c use n=99 to stop
c
      subroutine output(ii)
c
      integer ii(10)
c
      call prompt("n: ",n)
      if(n.eq.99) stop
      if(n.eq.9) return
c
      call prompt("ii(1): ",ii(1),"ii(2): ",ii(2),"ii(3): ",ii(3),
     1"ii(4): ",ii(4),"ii(5): ",ii(5))
c
      return
      end
```

```
 /* ******* Procedures wcrcat and prbcat ******* */
 /*                                              */
wcrcat:prbcat: proc (v, c, tc, switch, ncdp, flo, fup, tstop, ctime);
dcl (v (1000), c (1000), tc (1000), flo (10), fup (10), tstop, ctime) float;
dcl (switch (100), ncdp) fixed;
dcl (in1, in2) fixed, (string, sub1, sub2) char (12) var;
dcl (name, conversion) cond;
          on name (sysin) go to error;
          on conversion begin;
               put skip list ("error,try again");
               go to enter;
          end;
dcl (sysin, sysprint, file21) file;
dcl  a char (4);
          put file (file21) edit (" 014")(a);
          put skip file (file21);
          put skip file (file21);
enter:    put skip list ("ENTER show,set,OR run.");
          get list (a);
          if a = "show" then go to show;
          if a = "set " then go to set;
          if a = "run " then go to run;
          go to enter;
show:     put skip list ("show:");
          get list (string);
          if string = ";" then go to enter;
          if string = "ncdp" then put list (string, "=", ncdp);
          if string = "ncdp" then go to show;
          if string = "tstop" then put list (string, "=", tstop);
          if string = "ctime" then put list (string, "=", ctime);
          if string = "ctime" then go to show;
          if string = "tstop" then go to show;
          in1 = index (string, "(");
          in2 = index (string, ")");
          if in1*in2 = 0 then go to error2;
          sub1 = substr (string, 1, in1-1);
          sub2 = substr (string, in1+1, in2-in1-1);
          if sub1 = "v" then do;
               put list (string, "=", v (sub2));
               go to show;
          end;
          if sub1 = "c" then do;
               put list (string, "=", c (sub2));
               go to show;
          end;
          if sub1 = "tc" then do;
               put list (string, "=", tc (sub2));
               go to show;
          end;
          if sub1 = "switch" then do;
               put list (string, "=", switch (sub2));
               go to show;
          end;
```

```
          if sub1 = "flo" then do;
               put list (string, "=", flo (sub2));
               go to show;
          end;
          if sub1 = "fup" then do;
               put list (string, "=", fup (sub2));
               go to show;
          end;
error2:   put skip list ("error,try again.");
          go to show;
error:    put skip list ("error,try again.");
set:      put list ("set:");
          get data;
          if tstop <1970.0
          then go to error3;
          else if tstop >2050.0
          then go to error3;
          put skip;
          go to enter;
error3:   put skip list ("tstop out of bounds");
          go to set;
run:      put skip file (file21) data (ncdp);
          put skip file (file21);
          return;
     end;
```

```
c *************Real Function Delay3 (02-27-76)*************
c
c third order exponential material (conservative) delay
c
      real function delay3(qn,q,del,ql)
      common/dt/dt
      common/de3/an(35),al1(35),al2(35),al3(35),ar1(35),ar2(35),ar3(35)
      do 10 i=1,35
      if(an(i).eq.0.0)goto 20
      if(an(i).eq.qn)goto 30
10    continue
      write(6,15)
15    format(//' dictionary capacity exceeded in function delay3')
      pause 'delay3'
      ql=q*del
      delay3=q
      return
20    an(i)=qn
      ar1(i)=q
      ar2(i)=q
      ar3(i)=q
      al1(i)=q*del/3
      al2(i)=q*del/3
      al3(i)=q*del/3
      delay3=q
      ql=q*del
      return
30    ar1(i)=3.0*al1(i)/del
      ar2(i)=3.0*al2(i)/del
      ar3(i)=3.0*al3(i)/del
      al1(i)=al1(i)+dt*(q-ar1(i))
      al2(i)=al2(i)+dt*(ar1(i)-ar2(i))
      al3(i)=al3(i)+dt*(ar2(i)-ar3(i))
      delay3=ar3(i)
      ql=al1(i)+al2(i)+al3(i)
      return
      end
```

```
c ************ Real Function Dlinf1 (3-31-77) ************
c
c first order exponential information delay
c
      real function dlinf1(xn,x,t)
      common/dt/dt
      common/dl1/an(35),al(35)
      do 10 i=1,35
      if(an(i).eq.0.0)goto 20
      if(an(i).eq.xn)goto 30
10    continue
      write(6,15)
15    format(//' dictionary capacity exceeded in function dlinf1')
      pause 'dlinf1'
      dlinf1=x
      return
20    an(i)=xn
      al(i)=x
      dlinf1=x
      return
30    al(i)=al(i)+dt*(x-al(i))/t
      dlinf1=al(i)
      return
      end
```

```
c **********************Real Function Smooth***********************
c
c first order exponential information delay
c
      real function smooth(xn,x,t)
      common/dt/dt
      common/sm/an(35),ar(35),al(35)
      tt=t
      do 10 i=1,35
      if(an(i).eq.0.0)goto 20
      if(an(i).eq.xn)goto 30
10    continue
      write(6,15)
15    format(//' dictionary capacity exceeded in function smooth')
      pause 'smooth'
      smooth=x
      return
20    an(i)=xn
      ar(i)=x
      al(i)=x*tt
      smooth=x
      return
30    al(i)=al(i)+dt*(x-ar(i))
      ar(i)=al(i)/tt
      smooth=ar(i)
      return
      end
```

```
&           wcr.ec
&
& This is a series of multics commands which are invoked
& by typing "ec wcr". If the model user wants to
& store simulation runs in a file (segment named record)
& which can be renamed, saved, printed, etc. then type
& "ec wcr record".
&
&command_line off
&if [exists file file20]
&then delete file20
io attach file20 vfile_ file20
&if [equal &1. record.]
&then &goto record
&else &goto no_record
&label record
io attach file21 vfile_ record -append
wcrcrd
close_file -all
&quit
&label no_record
io attach file21 vfile_ discard
wcrcrd
close_file -all
&if [exists file discard] &then delete discard
&quit
```

The last two pages contain an example of
one of the models simulation runs.