```
%EXPLORE.  Modified 12/02; designed to be friendlier to
%change than previous versions, and to explain what each term
%represents.
% Note that the percent sign (%) marks a comment; it can be used
%to add an axplanation to a line or a section of the program - or to remove
%a line of code from the execution of a run.

%constants; these values stay the same throughout a run, unless modified in
%the program.

updecompk=.05;    % maximum rate of SOM decomposition
CNcrit=20;        % critical C:N for decomposition
uplmtN=10;        % maximum amount of N uptake by plants
leff=50;          % minimum N use efficiency
allocCN=1;        % allocation of C, N to non-woody tissues
turnC=.25;        %annual turnover of non-woody C to litter
turnN=.25;        %annual turnover of non-woody N to litter
woodturn=.02;     % annual rate of turnover of live wood
addc=0;           % annual addition of C to soils (fertilization)
feedback=1;       % a switch for the presence of plant/soil feedback
orgloss=0;        % a switch for losses of dissolved organic N

%set the first element of each array; generally this represents the initial
%coditions of the model, the value at year 1 of a run.  These values
%generally would be updated annually.

soilC(1)=9500;        % Soil carbon pool
plantC(1)=1500;       % Plant C pool (non-woody tissues)
woodC(1)=0;           % Wood C pool
soilN(1)=475;         % Soil N pool
plantN(1)=30;         % Plant N pool
woodN(1)=0;           % Wood N pool
totalN(1)=515;        % Total N in the simulated system

litC(1)=500;          % Annual litter C
litN(1)=10;           % Annual litter N
woodlitC(1)=0;        % Annual wood litter C
woodlitN(1)=0;        % Annual wood litter N
decompk(1)=.05;       % Decomposition rate
decomp(1)=500;        % Annual C flux in decomposition
Nmin(1)=10;           % Annual N mineralization
upperlmtN(1)=10;      % Upper limit to N uptake
Nupt(1)=10;           % N uptake by plants
prodC(1)=500;         % Plant production, in C

plantCN(1)=50;        % Plant C:N ratio
litCN(1)=50;          % Litterfall C:N ratio
Neffic(1)=50;         % N use efficiency (g C produced/g N uptake)

Ninput(1)=.5;         % inorganic N inputs
Nadd(1)=0;            % N fertilizer, if any
orgN(1)=0;            % organic N losses
Nloss(1)=Ninput(1);   % Inorganic N loss

startminN(1)=10.5;    % Inorganic N pool in the system
% after inputs and net N mineralization, but before plant uptake
midminN(1)=.5;        % Inorganic N after uptake, before loss
```

```
finalminN(1)=0;       % Inorganic N after loss

adjusteff(1)=0;       % Adjustment to N use efficiency, which can occur in a run
adjustdecomp(1)=0;    % Adjustment to decomposition, which can occur in a run.
demandN(1)=10;        % Annual N demand by plants

year(1)=1;            % The initial year of simulation

%Here set the first element of the output matrix for the properties
%you want to graph, equal to the value in year 1 above. I use this
%when outputting to excel, when I'm going to use it within another package.

graph(1,1)=year(1);
graph(2,1)=demandN(1);
graph(3,1)=Nmin(1);
graph(4,1)=prodC(1);
graph(5,1)=Nloss(1);
graph(6,1)=Neffic(1);
graph(7,1)=decompk(1);
graph(8,1)=totalN(1);
graph(9,1)=orgN(1);

%  If you want to change conditions at some time during a run, set the time
%when you will make the change using the value of z here.  '1' means constant
%through the run; any other
%value is the year at which conditions change.  As it is set up now, you can
%only make a change once in a run; that would be easy to alter.
%There must be a value for z.
%I often change conditions (eg introducing a supply/demand imbalance) in
%year 100 of a simulation run.  In this case I comment out (block with a
%percent sign) the execution of z=1, so change takes place in year 100.  If
%I wanted no change during the run, I'd comment out z=100 and let z=1
%stand.

        %z=1;
        z=100;

%   Often I want to add N fertilization at some time during a run, to test
%N limits plant growth under the conditions simulated.  That can be
%done here, by setting zz to the year when fertilization is to begin. To
%avoid fertilization, set zz beyond the end of the run. Again, the way it's
%set, the change can only be made once.

        zz=1500;
        %zz=300;

%   I use zzzz to delay the onset of feedback, as would happen if genotype
%or species change is necessary before the feedback can get going.
%0 = no delay.

        zzzz=0;

%set up the time-step loop.  This starts the simulation with year 2, and
%runs to year 500.

for y = 2:500          %incrementing years
   x=y-1;
```

```
   year(y)=y;

%This next loop kicks in if there is a change in conditions at some
%point in the simulation.  Any portion of the changes here can be
%knocked out by commenting out the statement.

   if year(y)>z  %a conditional statement; has no effect if y <= z

      %The following two statements turn the plant-soil feedback on or off,
      %if it's off then decomposition and efficiency remain constant.

          %feedback=0;      %feedback off
          feedback=1;       %feedback on

      %These next terms give two alternative ways to create a supply-demand
      %imbalance - either by decreasing maximum uptake and the rate of
      %decomposition, the latter a little more than the former; or by adding
      %more C to the soil, thereby immobilizing N.

         %uplmtN=9;                    %reduce N uptake
         %updecompk=.04;               %reduce decomposition
         %if year(y)<=z+1              %put the change in the decomposition array
         %   decompk(x)=updecompk;
         %end
         addc=50;                      %alternatively, add labile C to the soil,
         %which has the effect of immobilizing N and so reducing the supply
         %of N.

      %This next statement allows for storage of C and N in a slowly
      %turning over pool, as wood - or in another application, an accumulating
      %layer of Sphagnum.

         %allocCN=.8;                  %allocate 1 minus allocCN to wood

   end

   if year(y)>zz                       %is N fertilization on?
       Nadd(x)=5;                      %if so add 5 units of N
   end


   Ninput(y)=Ninput(x);    %N input same as last year; or can set input to 0
   Nadd(y)=Nadd(x);        %fertilizer N, if any
   soilC(y)=soilC(x)+litC(x)+woodlitC(x)+addc;  %soil C this year is the pool at
                           %the end of last year, plus inputs from plant litter
   decompk(y)=decompk(x);  %decomposition coefficient
   decomp(y)=soilC(y)*decompk(y);  %decomp. coefficient times soil C pool
   soilC(y)=soilC(y)-decomp(y);    %subtract decomposed material from soil pool
   soilN(y)=soilN(x)+litN(x)+woodlitN(x);  %soil organic N pool,
   %calculated as for soil C
   Nmin(y)=soilN(y)-soilC(y)/CNcrit;    %mineralize any N in the soil below the
                           %critical C:N ratio
   if Nmin(y)<=0           %bookkeeping
      Nmin(y)=.00001;
   end

   soilN(y)=soilN(y)-Nmin(y);     %subtract mineralization from soil
```

```
   %orgnanic N pool
   startminN(y)=finalminN(x)+Nmin(y)+Ninput(y)+Nadd(y);  %inorganic N pool
                        %after mineralization, before plant uptake - includes
                        %carryover from last year, N inputs, N fertilizer
   if startminN(y)>=uplmtN        %if inorganic pool is greater than or equal to
      Nupt(y)=uplmtN;             %potential N uptake, plants take up their
                                  %potential
   else
      Nupt(y)=startminN(y);       %otherwise they take up what's available
   end

%   This next set of statements modifies plant N use efficiency depending
%on whether plants get their potential N uptake, or something less

   if feedback>0              %if the feedback is on
       %if y>=z+zzzz          %this applies only to a delayed feedback
          adjusteff(y)=((uplmtN-Nupt(y))/(uplmtN-(uplmtN/2)));  %if N uptake
                        %falls into the range between the potential and
                        %half the potential uptake
          Neffic(y)=leff+adjusteff(y)*((leff*1.8)-leff);        %then set N
                        %use efficiency between minimum N use efficiency
                        %and a maximum efficiency that's 1.8 times minimum
          if Nupt(y)>=uplmtN          %if potential N uptake is acheived
             Neffic(y)=leff;          %use the minimum N efficiency
             demandN(y)=uplmtN;       %and define demand for N as the potential
          elseif Nupt(y)<=(uplmtN/2)  %if less than half the potential N is
                        %taken up
             Neffic(y)=leff*1.8;      %set efficiency at the maximum level
             demandN(y)=uplmtN/2;     %and define demand as half the potential
                        %uptake
          else                        %otherwise
             demandN(y)=Nupt(y);      %define demand as N actually taken up
          end
          %else                       %delayed feedback terms
          %demandN(y)=uplmtN;
          %Neffic(y)=leff;
       %end
   end

   %In the absence of feedback, plant N demand and N use efficiency don't
   %change as a function of N availability.

   if feedback<1
       demandN(y)=uplmtN;
       Neffic(y)=leff;
   end

   %The next line can be used to explore components of the feedback.  It
   %resets nutrient use efficiency to the minimum value even if the
   %feedback is on; if you turn it on (by removing the percent sign in
   %front of it), you can have a change in decomposition without a change
   %in N use efficiency.

      %Neffic(y)=leff;

   %This next line is important for graphing supply/demand; it isn't used
   %in the flow of the model itself
```

```
   upperlmtN(y)=uplmtN;

%   The next set of lines calculates productivity and N accumulation
%in plants.

prodC(y)=Nupt(y)*Neffic(y);     %Production is efficiency times uptake
plantC(y)=plantC(x) + allocCN*prodC(y);  %if allocCN is less than 1,
plantN(y)=plantN(x) + allocCN*Nupt(y);   %only some of the C and N goes
                     %to non-woody parts of plants.  The remainder goes
                     %to wood, in the next set of lines.  Note that the
                     %C:N ratio in wood is the same as that in non-woody
                     %tissues; something that can be played with
plantCN(y)=plantC(y)/plantN(y);          %C:N of non-woody tissue
woodC(y)=woodC(x)+(1-allocCN)*prodC(y);  %wood C
woodN(y)=woodN(x)+(1-allocCN)*Nupt(y);   %and wood N
woodlitC(y)=woodC(y)*woodturn;           %litter production from wood
woodlitN(y)=woodN(y)*woodturn;
woodC(y)=woodC(y)-woodlitC(y);           %C and N remaining in wood
woodN(y)=woodN(y)-woodlitN(y);           %after litter production

%   The next set of lines introduce disturbances that eliminate plant
%C and N, in years 100 and 350 of the simulation.  These disturbances
%can be thought of as whole-tree longging or intense fire.  The model
%CNP, written for the element interactions rapid assessment (Hungate
%et al. in press), contains disturbances that mimic
%windthrow and grazing as well.

%if year(y)==100
%    plantC(y)=0.5;
%    plantN(y)=0.01;
%end
% if year(y)==350
%     plantC(y)=0.5;
%   plantN(y)=0.01;
%end

%calculate litter production, in terms of C and N

litC(y)=plantC(y)*turnC;     %turnover coefficient times biomass C
litN(y)=plantN(y)*turnN;     %turnover coefficient times N
litCN(y)=litC(y)/litN(y);    %C:N of non-woody litter

%  Adjust decomposition rate if the feedback is on.  Decomposition
%coefficient takes on the default value if N use efficiency (and
%so the C:N ratio of litter) is minimum; otherwise, decomposition
%is decreased in proportion to the change in C:N ratio

if feedback>0                %if the feedback is on
    %if y>=z+zzzz             %useful to explore delayed feedback
       adjustdecomp(y)=((litCN(y)-leff)/((1.8*leff)-leff));  %where does
                  %C:N ratio fall, along the spectrum from minimum to
                  %maximum N use efficiency?
       decompk(y)=updecompk-(.8*(adjustdecomp(y)*(updecompk-(updecompk/2))));
                  %calculate new decomposition constant, between the default
                  %value and half the default value. The .8 here is a
                  %coeffecient that can be used to reduce (or increase) the
```

```
                    %magnitude of the effect on decomposition.

      %else                       %useful for delayed feedback
         %decompk(y)=updecompk;
      %end

   else                           %if the feedback is off,
      decompk(y)=updecompk;      %keep the default decomposition rate
   end

   %The next line can be used to explore components of the feedback; as for
   %N use efficiency above, by removing percent sign you can turn off any
   %effect on decomposition while keeping an effect on N use efficiency.

   %decompk(y)=updecompk;

   %calculate plant biomass after litter fall

   plantC(y)=plantC(y)-litC(y);
   plantN(y)=plantN(y)-litN(y);

   midminN(y)=startminN(y)-Nupt(y);   %midminN is the inorganic N remaining
                %in the soil after plant uptake.  I assume it's lost via
                %leaching, although a fraction of it could be retained.
                %Losses can be set to 0 (carrying over the inorganic N
                %to next year) by making the Nloss = 0 in the next
                %statement.
   Nloss(y)=midminN(y);
   %You can turn on organic N losses here
   if orgloss>0       %if organic N losses are switched on (orgloss=1)
      orgN(y)=.001*soilN(y);     %.1% of total soil N is lost via leaching
      soilN(y)=soilN(y)-orgN(y); %Account for loss of organic N
   else               %otherwise loss of dissolved organic N is 0
      orgN(y)=0;
   end

   finalminN(y)=midminN(y)-Nloss(y);       %subtract inorganic N loss from
                %midminN to get the amount carried over to next year
                %(here 0)

   totalN(y)=plantN(y)+soilN(y)+litN(y)+woodN(y)+woodlitN(y)+finalminN(y);
                %sum up total N within the system

   %   This next set of lines continues to build the output matrix; the value
   %each pool or rate in this year is added to the matrix.  Any vector can be
   %summarized in this way.
   graph(1,y)=year(y);
   graph(2,y)=demandN(y);
   graph(3,y)=Nmin(y);
   graph(4,y)=prodC(y);
   graph(5,y)=Nloss(y);
   graph(6,y)=Neffic(y);
   graph(7,y)=decompk(y);
   graph(8,y)=totalN(y);
   graph(9,y)=orgN(y);
end                   %and on to the next time step.  Once the end of the run
       %is reached, then go on to output (below).
```

```
%   Outputs.  These next lines are used to get data into an output text file;
%I read that file into EXCEL, and then import it into to Sigmaplot
%for graphing. I also have some MATLAB graphs below; but I work with the
%Sigmaplot ones for slides/publication.  First I transpose the output matrix,

%then save the transposed file as an ascii file.
   outgraph=transpose(graph);
   save outs.txt outgraph -ascii -tabs

%   Below are MATLAB graphs.
figure
plot (year,demandN,'k', year,Nmin,'k:')
grid off
axis ([0 500 0 15])
ylabel ('Supply and Demand')
xlabel ('Year')
print SoilN -dpsc

figure
plot (year,prodC,'k')
grid off
axis ([0 500 0 550])
ylabel ('productivity')
xlabel ('Year')

figure
plot (year,totalN,'k')
grid off
axis ([0 500 0 800])
ylabel ('totalN')
xlabel ('Year')

%And that's it!
```