APPENDIX 3. The main controlling program of DECUMA, with each line of Fortran 95 code annotated.

Code or Module Call	Action or Activities
program Decuma	Initialize the program DECUMA
use Parameter_Values	Import into the program general parameter names (e.g., file names, constants)
implicit none	Require all the parameters to be explicitly declared, to avoid typographic errors
call Initialize_Values	Input parameter values that control program simulation (e.g., months modeled)
call Initialize_Metrics	Input parameter values that describe human energy needs, buy and sell prices, sheep forage units, human adult equivalents, and movement thresholds
call Initialize_Landscapes	Read landscape maps (e.g., household density) into the internal spatial database
call Initialize_Livestock	Input values for initial livestock age distributions and energy requirement
call Initialize_Houses	Initialize households, including their livestock holdings and all other attributes
call Initialize_Herds	Distribute each household's livestock species into age and sex cohorts
if (statev_flag == 2 .or. statev_flag == 3) then	If file from a spin-up is to be read-in (2) or read-in and written-out (3), then …
call Restore_HSID_Long_Term	Load the maps for each species representing long-term habitat suitability
call Restore_Agents	Load the attributes of all the households as they were at the end of the spin-up
end if	(continue)
call Livestock_Update_Summarize	Update total livestock in each herd for each species, plus condition indices
call Update_Agents	Update total livestock per person, plus running income for household
call Livestock_Distribute	Initially distribute livestock on landscape based upon suitability and access rules
do nmonth=1,months_modeled	For each month to be modeled, loop through …
call Monthly_Inputs	Input precipitation map, ensure that the ecosystem and household models sync
Write(*,*) 'Year: ',year,' Month: ',month	Echo the month and year being modeled to the screen
do week=1,4	For each week to be modeled, loop through ...
call Livestock_Distribute	Distribute livestock on the landscape based upon suitability and access rules
call Livestock_Energy_Acquired	Read from the ecosystem model the amount of energy livestock acquired
call Livestock_Update_Summarize	Update total livestock in each herd for each species, plus condition indices
end do	… end weekly loop
call Livestock_Energy_Used	Calculate energy used by livestock species, by age and sex cohorts in herds

```
    call Livestock_Weight_Change        Calculate weight change in livestock species, by age and sex cohorts in herds.  If
                                        energy acquired is less than used, weight loss, otherwise, weight gain
    call Livestock_Mortality            Simulate death of livestock, within age and sex cohorts.
    call Livestock_Age_Herds            Age animals, if the month is appropriate for them to be aged
    call Livestock_Give_Birth           Simulate reproduction in livestock species, if the month is appropriate

    call Update_Agents                  Update total livestock per person, plus running income for household
    call Harvest_Crops                  Harvest crops for each household.  Yields are related to local precipitation
    call Agent_Cash_Flows               Calculate incomes and expenses for each household
    call Agent_Cash_Needs               Estimate future needs for some select expenses, used to judge the need for
                                        livestock sales and purchases
    call Agent_Livestock_Trades         Allow those households that wish to sell animals to do so
    call Agent_Energy_Flows             Calculate the energy needed and available, and purchase energy if needed
    call Agent_Livestock_Buying         Allow those households that wish to purchase animals to do so
    call Agent_Livestock_Gifting        If households have lost all their animals, have a wealthy neighbor donate to them
    call Move_Herd_Camps                Allow households to consider whether to move their temporary camp to a new
                                        grazing location

    call Update_Agents                  Update total livestock per person, plus running income for household
    call Agent_Outputs                  Output population summary information for all households simulated
    call Livestock_Outputs              Output population summary information for the livestock species
    call Single_Herd_Out(10)            Output detailed information for herds of a single household, here household 10
    call Individual_Agent_Outputs       Output summary information for selected households, usually those with
                                        observed household survey data, for later comparison to simulated data

    call Spatial_Outputs                Output maps (e.g., habitat suitabilities, livestock densities, camp locations)
    call Monthly_Clean_Up               Reset monthly accumulators to zero
    if (month == 12) then               If the month simulated is December, then …
        call Smooth_HSID                Create a smooth habitat suitability map for each species, for use in camp
                                        movement modeling

        call Yearly_Clean_Up            Reset yearly accumulators to zero
    end if                              (continue)
    call Time_Stamp                     Write the month and year being simulated to a file, for use in a graphic interface
end do                                  … end monthly loop
```

```
    if (statev_flag == 1 .or. statev_flag == 3) then   If file from a spin-up is to written-out (1) or read-in and written-out (3), then …
        call Save_HSID_Long_Term                       Save the long-term habitat suitability maps for each species, used in movement
                                                       modeling
        call Save_Agents                               Save the state of all the agents at the end of the simulation into a single file
    end if                                             (continue)
    call Clean_Up                                      Close the open files
end program                                            End program DECUMA
```