# Lab 1: Introduction to forest-level simulation[1]

## Learning Objectives

Participants will understand:

- how simulation models are used in forest level analysis
- how the elements of forest-level analysis are represented in a model (yield tables, inventory, management and other assumptions)
- the mechanics of a simulation model (growth, cutting, treatment)
- the difference between forest-level and stand-level perspectives
- that the definition of sustainability depends on the context in which it is used.

## Key Concepts

- simulation model
- forest-level perspective
- even-aged management
- modeling approach should fit the problem.

---

[1] This material was adapted from Buongiorno, J. and K. Gilless, 1986, Forest management and economics, Macmillan. Pages 203-214. Changes have been made to the computer program (Even), some of the text, and the problems.

## Introduction to forest-level modelling using *Even*

This lab exercise uses a simulation model called *Even* to investigate harvest scheduling in even-aged forests. The model can be used to:

1. describe the evolution of the forest over time when it is managed according to a variant of area control

2. predict the effects of changing the allowable cut on the value of the forest.

The *Even* model is based on the following assumptions:

- the state of the forest at a specific point in time is described by the area in each age class
- the silvicultural system consists of clearcutting followed by immediate artificial regeneration
- the volume of timber in a particular age class is strictly a function of the age of the stand
- the management policy is a variant of area control—specifically, the manager fixes the area to be cut during each time interval
- oldest stands are always cut first; this implies a rotation that is equal to the total area of the forest divided by the average area cut every year.

The code, written in BASIC, is listed in Figure 1. *Even* consists of one main program and seven subroutines, described in the following sections. The variables used in the program are defined in Table 1.

### *MainProgram*

The program *MainProgram* directs the flow of information between subroutines and updates the simulation clock. *MainProgram* first reads the initial condition of the forest and the management data. Then, for each decade, it displays the current status of the forest, and the predicted growth of the remaining forest during the next decade. The program then increases the time by 10 years. This cycle is repeated to the end of the specified simulation period.

### Subroutine *InitializeSimulation*

Subroutine *InitializeSimulation* sets the initial time and the initial present value of the harvests to zero. It then reads the initial area in each of the 10 age classes. The ages of trees within an age class span a decade. For example, in Figure 1 there are 250 ha in age class 1, 150 ha in age class 2, and so on, up to 120 ha in age class 10. The total area of the forest is 2280 ha. Any stand older than 100 years is included in the oldest age class. The number of age classes remains the same throughout the projection period, although the area in each age class changes as area is cut and ages over time.

Subroutine *InitializeSimulation* then reads the volume of timber as a function of age, in cubic metres per hectare. For example, in Figure 1, stands of timber 50 years old carry some 289 $m^3$ / ha (data are for lodgepole pine in the BC Interior, site index 25).

### Subroutine *ManagementInformation*

Subroutine *ManagementInformation* reads the management data, beginning with the desired length of the simulation—set at 150 years (Figure 1). The subroutine then reads the price of timber ($10 / $m^3$), the interest rate (4% / yr), and the reforestation cost ($400 / ha). The last information read by subroutine *ManagementInformation* is the maximum allowable cut (380 ha). This is the area that would be cut every decade if the forest (2280 ha) was regulated at a rotation age of 60 years (2280 / 60 x 10 = 380 ha / decade).

### Subroutine *ShowGrowingStock*

Subroutine *ShowGrowingStock* displays on the computer screen the current year and, under it, the area in each age class at the beginning of that year (screen output shown in Figure 2).

### Subroutine *Cutting*

Subroutine *Cutting* first initializes the total area and the total volume cut in the next 10 years to zero. Then it cuts the allowed area in the following steps, starting with the oldest age class:

1. The area to be cut in the current age class is set equal to the allowable cut minus the total area already cut in the current decade.
2. If the area to be cut is greater than the area that is available in the current age class, then the cut can only be equal to the area available.
3. The area left in the current age class after the cut is determined.
4. The total area and the total volume that have been cut in the current decade are computed.

These steps are repeated for younger and younger age classes, until the total area cut in the decade equals the allowable cut.

### Subroutine *PresentValue*

Subroutine *PresentValue* computes the cumulative net present value of all harvests up to the last one. The net returns in the current decade are the stumpage value of the volume cut, minus the reforestation cost. These net returns are discounted to the present and added to the cumulative present value up to that year. Note that harvest is assumed to proceed continuously throughout the current decade. For simplicity, however, all revenues and costs during a decade are accounted for in the middle of the decade. Therefore, the time used in the discount formula is the current year plus five.

## Subroutine *ShowCut*

Subroutine *ShowCut* displays the area cut in each age class during the coming decade, as well as the volume cut and the cumulative net present value of the returns up to and including those generated during the coming decade (output shown in Figure 2).

## Subroutine *Growth*

Subroutine *Growth* simulates the growth of the part of the forest that will not be cut during the coming decade. The area cut during the coming decade will constitute age class 1 at the end of the decade. Simultaneously, the areas in age class 1 to 8 grow into age classes 2 to 9. The new age class 10 at the end of the decade consists of what is left of the old age class 10 after the cut, plus what was in age class 9 at the beginning of the decade.

**Figure 1 EVEN: Simulation of even-aged forest management under area control.** (Source: Buongiorno, J. and K. Gilless 1986. Forest management and economics. Macmillan p. 203-214.)

```
'______________________________________________
' MainProgram:   Directs flow of information to subroutines
'______________________________________________
mainProgram:
  CLS
  GOSUB initializeSimulation              'initiate forest
  GOSUB managementInformation             'define management
  WHILE year <= endingYear                'until max time is reached
    GOSUB showGrowingStock                'display current growing stock
    GOSUB cutting                         'cut forest
    GOSUB presentValue                    'determine present value
    GOSUB showCut                         'display cut data
    GOSUB growth                          'predict growth
    PRINT
    INPUT "Hit Enter to continue with next iteration", CR
    year = year + 10                      'increase time
  WEND                                    'go back to WHILE statement
END                                       'end of main program

'______________________________________________
'InitializeSimulation:   Read initial forest condition
'______________________________________________
initializeSimulation:
  year = 0                                'initial time
  presentValue = 0                        'initial present value
  FOR ageClass = 1 TO 10
    READ areaBeginning(ageClass)          'initial area (hectares)
  NEXT ageClass
  DATA 250,150,320,400,0,100,300,110,530,120
  FOR ageClass = 1 TO 10
    READ VolumePerHectare(ageClass)       '(m3/ha)
  NEXT ageClass
  DATA 0.0,2.0,87,192,289,364,421,466,470,470
RETURN                                    'return to MainProgram
```

```
'____________________________________________
'ManagementInformation:  Define management regime
'____________________________________________
managementInformation:
  READ endingYear                              'max simulated time (years)
  DATA 150
  READ price, discountRate, regenCost
  DATA 10,0.04,400
  READ allowableCutArea                        '(hectares/decade)
  DATA 380
RETURN '                                       return to MainProgram

'____________________________________________
'ShowGrowingStock:  Display current growing stock
'____________________________________________
showGrowingStock:
  PRINT "year"; year
  PRINT " stock(ha)";
  FOR ageClass = 1 TO 10
    PRINT USING "####."; areaBeginning(ageClass);
  NEXT ageClass
  PRINT
RETURN '                                       return to MainProgram

'____________________________________________
'Cutting:  Cut forest
'____________________________________________
cutting:
  totalAreaCut = 0                             'total area cut during decade
  totalVolumeCut = 0                           'total volume cut during decade
  FOR ageClass = 1 TO 10
    areaCut(ageClass) = 0                      'area cut in age class during decade
  NEXT ageClass
  ageClass = 10                                'start with oldest age class
  WHILE totalAreaCut < allowableCutArea
    areaCut(ageClass) = allowableCutArea - totalAreaCut
    IF areaCut(ageClass) > areaBeginning(ageClass)
      THEN areaCut(ageClass) = areaBeginning(ageClass)
    areaBeginning(ageClass) = areaBeginning(ageClass)-
      areaCut(ageClass)                        'area left after cut
    totalAreaCut = totalAreaCut + areaCut(ageClass)
    totalVolumeCut = totalVolumeCut +
      areaCut(ageClass) * volumePerHectare(ageClass)
    ageClass = ageClass-1
  WEND
RETURN                                         'return to MainProgram

'____________________________________________
'PresentValue:  Compute cumulative net present value
'____________________________________________
presentValue:
  discountFactor = (1 + discountRate)^(year + 5)
  presentValue = presentValue + (totalVolumeCut * price –
totalAreaCut * regenCost) / discountFactor
RETURN                                         'return to MainProgram
```

```
'_______________________________________________
'Showcut:  Display area, volume of cut and net present value
'_______________________________________________
showCut:
  PRINT
  PRINT " cut (ha) ";
  FOR ageClass = 1 TO 10
    PRINT USING "####."; areaCut(ageClass);
  NEXT ageClass
  PRINT
  PRINT USING " Volume cut ######,_ Present value_    $########,";
      totalVolumeCut; presentValue
RETURN                                      'return to MainProgram

'_______________________________________________
'Growth:  Predict growth of forest during next decade
'_______________________________________________
growth:
  areaEnding(1) = totalAreaCut              'area reforested
  FOR ageClass = 2 TO 9
    areaEnding(ageClass) = areaBeginning(ageClass-1)
                                            'moves up one age class
  NEXT ageClass
  areaEnding(10) = areaBeginning(10) + areaBeginning(9)
                                            'area in oldest age class

'_______________________________________________
'Area at beginning of next decade is equal to area at end of current decade
'_______________________________________________
  FOR ageClass = 1 TO 10
    areaBeginning(ageClass) = areaEnding(ageClass)
  NEXT ageClass
RETURN                                      'return to MainProgram
```

**Table 1 Definition of variables in program *Even*.**

| Variable | Definition |
|---|---|
| *ageClass* | Age-class index, varies from 1 to 10. Class width is 10 years. |
| *allowableCutArea* | Area to be cut per decade (ha) |
| *areaBeginning (ageClass)* | Area in an age class at the beginning of a decade (ha) |
| *areaCut (ageClass)* | Area cut in an age class during a decade (ha) |
| *areaEnding (ageClass)* | Area in an age class at the end of a decade (ha) |
| *discountRate* | Discount rate (% per year) |
| *endingYear* | Maximum simulated time (years) |
| *presentValue* | Cumulative net present value ($) |
| *price* | Price of logs, net logging costs ($/m$^3$) |
| *regenCost* | Regeneration cost ($/ha) |
| *totalAreaCut* | Total area cut in a decade (ha) |
| *totalVolumeCut* | Total volume cut in a decade (m$^3$) |
| *volumePerHectare* | Volume per hectare (m$^3$) |
| *year* | Simulation time, increased by 10-year intervals |

# Output and Applications of *Even*

## Forest Dynamics

Part of the results of a simulation run with *Even* appear in Figure 2. The input data are those from subroutines *InitializeSimulation* and *ManagementInformation* (Figure 1). Every decade, 380 ha are cut from the forest, as required by the area-control strategy. This corresponds to a rotation of 60 years. The oldest age class is always cut first, and the number of age classes decreases over time to six. The presence of old timber on the initial forest results in more volume being cut during the first decades than later on. Sixty years after the beginning of the simulation, the forest is regulated. Thereafter, it remains in a steady state, producing always the same timber volume and retaining the same growing stock.

## Forest Value

The data in Figure 2 show that a large part of the net present value of the returns from this forest is generated during the first decades. By the time the forest has been regulated, 60 years after the management is initiated, the net present value of the forest increases very slowly. In fact, continuing the simulation shows that the net present value gradually approaches a maximum of about $4 million. This is the value of the forest, inclusive of land and growing stock, given the specified management regime and all other assumptions. Given these assumptions, a buyer could pay $4 million for the forest and be ensured a real rate of return on the investment of 4% per year.

**Figure 2 Output of program *Even.***
(volumes are in $m^3$; stock and cut are in hectares)

```
year 0
  stock (ha)250. 150. 320. 400.   0. 100. 300. 110. 530. 120.
  cut (ha)    0.   0.   0.   0.   0.   0.   0.   0. 260. 120.
  Volume cut 178,600 cu.m.    Present value $1,343,029

Hit Enter to continue with the next iteration

year 10
  stock (ha)380. 250. 150. 320. 400.   0. 100. 300. 110. 270.
  cut (ha)    0.   0.   0.   0.   0.   0.   0.   0. 110. 270.
  Volume cut 178,600 cu.m.    Present value $2,250,331

Hit Enter to continue with the next iteration

year 20
  stock (ha)380. 380. 250. 150. 320. 400.   0. 100. 300.   0.
  cut (ha)    0.   0.   0.   0.   0.   0.   0.  80. 300.   0.
  Volume cut 178,280 cu.m.    Present value $2,862,072

                                ...

Hit Enter to continue with the next iteration
year 60
  stock (ha)380. 380. 380. 380. 380. 380.   0.   0.   0.   0.
  cut (ha)    0.   0.   0.   0.   0. 380.   0.   0.   0.   0.
  Volume cut 138,320 cu.m.    Present value $3,727,603

Hit Enter to continue with the next iteration
```

```
year 70
  stock (ha)380. 380. 380. 380. 380. 380.   0.   0.   0.   0.
  cut (ha)    0.   0.   0.   0.   0. 380.   0.   0.   0.   0.
  Volume cut 138,320 cu.m.    Present value $3,792,591
```

## Best Rotation

An interesting application of *Even* is to determine the rotation that is optimum given a particular management strategy. In this case, what rotation maximizes the monetary value of the forest? The forest owner could use this information to choose the management regime that maximizes the present value of his property. A potential buyer of the forest would want to know this to determine the most he should pay for the forest.

*Even* can be used to address this question by changing the values of the allowable cut variable, *allowableCutArea*, in subroutine *ManagementInformation* (see Figure 1).

For example, *allowableCutArea* = 2280 ha implies a rotation age of 10 years—i.e., that all the forest is cut and regenerated during one decade. Similarly, *allowableCutArea* = 1140 ha implies a rotation of 20 years, so that half of the forest is cut each decade. Finally, *allowableCutArea* = 228 ha implies a rotation of 100 years, with one-tenth of the forest being cut and replanted every decade.

Table 2 shows the results of several simulation runs with *Even*, leaving everything but the allowable cut area constant. In each case, the forest value shown is the present value of the forest **at the end of the planning horizon** (150 years), using the corresponding allowable cut. The rotation that maximizes forest monetary value, about $4.9 million, is 30 years (*allowableCutArea* = 760 ha).

**Table 2 Forest value for different rotation ages.**

| Rotation (years) | Allowable cut (hectares) | Forest value ($ millions) |
|---|---|---|
| 10 | 2280 | 2.8 |
| 20 | 1140 | 4.2 |
| 30 | 760 | 4.9 |
| 40 | 570 | 4.8 |
| 50 | 456 | 4.4 |
| 60 | 380 | 3.9 |

In this example, a forest-level perspective was taken for determining the best management regime (rotation age), and the “goodness” of the management regime was measured in terms of monetary value.

Contrast this result with the best rotation age from a stand-level perspective, determined as the rotation that produces the highest soil expectation value (SEV).

Soil expectation value (SEV) is the value of bare land, calculated as the net present value of the series of future crops of timber that could be grown on it. The following formula can be used to calculate SEV:

SEV = [(p x v(t) - c1) / ((1+r)^t)-1]- c2

where,

p = unit value of standing timber ($ / $m^3$)

v(t) = standing volume at age t ($m^3$ / ha)

c1 = cost of regeneration after harvest

r = discount rate

t = rotation age (years)

c2 = cost of establishing the initial stand ($ / ha)

p x v(t) is the harvest revenue in $ / ha, and c1 is the regeneration cost ($ / ha): therefore (p x v(t)-c1) is the net revenue from cutting and regenerating one hectare. The term ((1+r)^t)-1 in the divisor discounts (to the present) the infinite series of these periodic future harvests occurring t years (one rotation) apart. Finally the cost of establishing the initial stand (c2) is deducted (we start with bare land).

Table 3 shows that a rotation age of 50 years produces the highest SEV.

**Table 3 Determining economic rotation by highest soil expectation value.**

| Rotation (years) | Vol/ha ($m^3$/ha) | Revenue ($/ha) | SEV ($/ha) |
|---|---|---|---|
| 10 | 0 | 0 | -1233 |
| 20 | 2 | 20 | -719 |
| 30 | 87 | 870 | -190 |
| 40 | 192 | 1920 | 0 |
| 50 | 289 | 2890 | 8 |
| 60 | 364 | 3640 | -60 |
| 70 | 421 | 4210 | -139 |

In a sense, the difference in the best rotation ages indicated by the two different techniques reflects the difference between taking a stand-level perspective (SEV of the stand), and taking a forest-level perspective (simulation with *Even*).

The best rotation (30 years) found with *Even* is shorter than that found using SEV (50 years). Since *Even*'s area-control approach cuts the same number of hectares each year, and there is a surplus of old forest initially, in the first few decades larger volumes are cut from these hectares than would be otherwise produced at the specified rotation age. The SEV approach on the other hand, is based on cutting only the yield projected to be available at the specified rotation age.

In defining an optimum, whether for rotation or any other management parameter, it is important to define precisely the context in which the optimization is to be done. The context includes the status of the forest, the management regime that is to be applied, and the economic and social environment in which management takes place.

This last point is worth stressing, since area control implies liquidation of all the old-growth timber in one rotation (30 years). For many reasons, this may not be acceptable. In making the allowable-cut decision, a manager would want to carefully weigh the economic gain resulting from shorter rotations against the loss of other values that, although more difficult to quantify, are not less real.

## Conclusions

The simulation model *Even* offers a straightforward approach to evaluating some management strategies for even-aged forest management. The examples shown here illustrate the importance of context and perspective in choosing management parameters like rotation age.

Although this model deals only with area-control management, it is not hard to see how it could be adapted to volume control by simply changing the allowable cut from a specified area to a volume (see Assignment problem 4). Other features could also be added to the model. While there is a natural tendency to want to add refinements to make a model as realistic as possible, it should be resisted. Increasing the realism of a model does not necessarily improve it—added features may make a model more difficult to understand and more prone to errors. The model should be designed at a level of abstraction and simplicity that helps one focus on the key characteristics of the problem.

## EVEN Lab Assignment

1. Read the section of this handout entitled **Best Rotation**. Using the model as explained in this section, replicate the results shown in Table 2.

2. Change the allowable cut (*allowableCutArea*) to 0 ha—i.e., no harvest—what happens?

3. The program *Even* simulated the growth of a forest that had many hectares of old timber.

   Consider instead a forest of the same size and productivity that has been recently cut over. It has 1900 ha, 300 ha, and 80 ha in age classes 1, 2, and 3, respectively, and no area in older age classes. Change the data of *Even* to reflect this new initial condition. Set the allowable cut back to 380 ha.

   Run this modified version of *Even*. Determine the value of the forest at the end of the planning horizon (e.g., 150 years) when the forest is managed on rotations of 10, 20, 30, 40, 50, and 60 years.

   Compare these results to those in Table 2. Is the best rotation still 30 years? Why? How important is initial forest structure in determining the value of the forest?

4. After completing problem 3 restore the original age-class distribution from Figure 1.

   Consider a forest of same characteristics as the one described in Figure 1, but growing on a better site. On this site, the yield of each age class is:

| **Age class** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| $m^3$/ha | 0 | 10 | 118 | 242 | 343 | 421 | 479 | 510 | 540 | 560 |

   Modify the data in *Even* to reflect this higher yield. Run this modified version of *Even*. Determine the value of this new forest when managed on different rotations. Compare your results to those in Table 2. What is the new best economic rotation?

5. Ensure that you have restored the original age-class distribution from Figure 1 after completing problem 3, and that you restore the original yields from Figure 1 after completing problem 4.

   The version of *Even* shown in Figure 1 can easily be modified to allow for volume control rather than area control. First, change the third READ statement in subroutine *ManagementInformation* to:

```
READ allowableCutVolume
```

Then change the WHILE statement in subroutine *Cutting* to:

```
WHILE totalVolumeCut < allowableCutVolume AND ageClass>1
```

Last, change the next statement in subroutine *Cutting* to:

```
areaCut(ageClass) = (allowableCutVolume - totalVolumeCut)/
volumePerHectare(ageClass)
```

*Even*, thus modified, recognizes the value in the third DATA statement in subroutine *ManagementInformation* as the allowable cut, in cubic metres per decade.

Run this modified version of *Even*. Find the “best” allowable cut level by repeating a procedure of running the model, inspecting the simulation results after each run, and adjusting the allowable cut volume as needed. Find the rotation ages resulting from different levels of cut, say between 100 000 $m^3$/ decade and 150 000 $m^3$/ decade.

6. Write and submit a short report in which you simply and concisely explain:
   - how forest structure is defined in *Even* (this is **not** a trick question—the answer is very simple)
   - the term *initial conditions*, and where initial conditions are defined in Even
   - the term *rules of change*, and the rules of change used in Even
   - each line in the subroutine Cut.