DYMEX v2

Perennial Plant Tutorial

1. An Introduction to Plant Modelling with DYMEX

Important
This tutorial set assumes that the user has no previous knowledge of the DYMEX package. It is designed to be followed sequentially. Do not skip any section, as vital information will be missed. The assumption is also made that the user's computer platform is Windows 2000.

1.1 The DYMEX Programs

DYMEX comes in two parts: the **Model Builder** and the **Simulator**. The Model Builder is used to build a model, while the Simulator applies the model's set of functions and parameters and simulates possible outcomes. This introduction will discuss the procedures involved in assembling such a model. DYMEX models can be extremely complex; however this introductory tutorial uses a greatly simplified hypothetical species so that the resulting model is reduced to the absolute minimum. This illustrates the basic procedures involved in operating DYMEX, and also shows how additional functionality can be added to this simple model in order to increase its realism. The complexity of any model created using the DYMEX Model Builder is determined by the user within the bounds of the Model Builder. The model's accuracy and reliability are measured by how well the model represents the real system. *Always remember that DYMEX is modelling a population, not an individual organism.*

1.2 Modelling a Hypothetical Perennial Plant

1.2.1 'Pseudo-wattle'

The characteristics of the hypothetical species used in this tutorial do not apply to any specific perennial plant; the name 'Pseudo-wattle' is only for convenience of reference. Pseudo-wattle will initially be described as a very simple organism but in each tutorial its characteristics will be made more complex. Four life cycle stages were considered the minimum number required to model perennial behaviour satisfactorily. In these initial tutorials, the number of seeds released annually will be set artificially low. This is necessary because at first the populations will have no limiting controls and total numbers would otherwise become too high, too quickly to allow trends to be displayed. To commence the model development, chronological age will be used to drive the population dynamics.

> Pseudo-wattle exists as seeds, seedlings, juvenile plants and adult plants. All seeds germinate to become seedlings after a dormancy interval of two months following their dispersal by the parent plant. The seedlings reach the juvenile plant stage after a further three months and remain as juvenile plants for 12 months. The adults then flower once every year in September to produce 30 seeds. Adult plants have a life-span of 8 years. For this initial tutorial, no lifestage of Pseudo-wattle is affected by temperature. It is assumed that soil nutrients and water are always sufficient for growth of the plants and that there are no predators/pathogens on either lifestage.

1.2.2 Model Attributes

Even with such a 'simple' life cycle, there are a number of attributes that need to be described in DYMEX. From the description of the life cycle, a DYMEX model will have to include the following:

The number of life cycle stages
The type of each life cycle stage (e.g., seed, adult)
The length of time an average individual spends in each life cycle stage
Conditions which affect the organism's transformation from one stage to another
The timing of reproduction
The number of potential offspring
The pattern of production of offspring (eg batches, continuous, etc.)
Mortality and when and how it occurs, and finally
Output of results from the model.

Throughout the following tutorial, the user should refer to this list as the model is developed.

1.3 Using DYMEX to Build the Model

1.3.1 Starting

Either open the 'Start' menu and select 'Model Builder' or if it is on the desktop, select the Model Builder icon (as shown here) in order to open the Model Builder program. From this point, keystrokes will be given as complete sequences after either explanations or discussions.

The main DYMEX Builder window contains a blank screen and a menu bar containing various options. The following procedure allows the user to create a new model.

1. From the menu bar, select **'File'** to produce a drop-down menu
2. Select **'New Model'** from the drop down menu.

Once this selection is made, the **'Model Details'** window appears (Figure 1.1). This window allows the user to insert details about the model. Information entered here is important for subsequent users of the model, and the information should be regularly updated as the model is modified. At this stage, the name of the new model, the name of the builder and the version number should be inserted; general details on the model's construction can be inserted immediately or added to or modified later as the model is developed. The **'Model Details'** window appears automatically when a new file is created or a previously saved file is opened, however the user can always open the **'Model Details'** window by selecting '**Model'** from the main menu bar and then **'Details'** from the drop-down menu. When finished, the **'OK'** button is selected from the window and the Model Window appears.

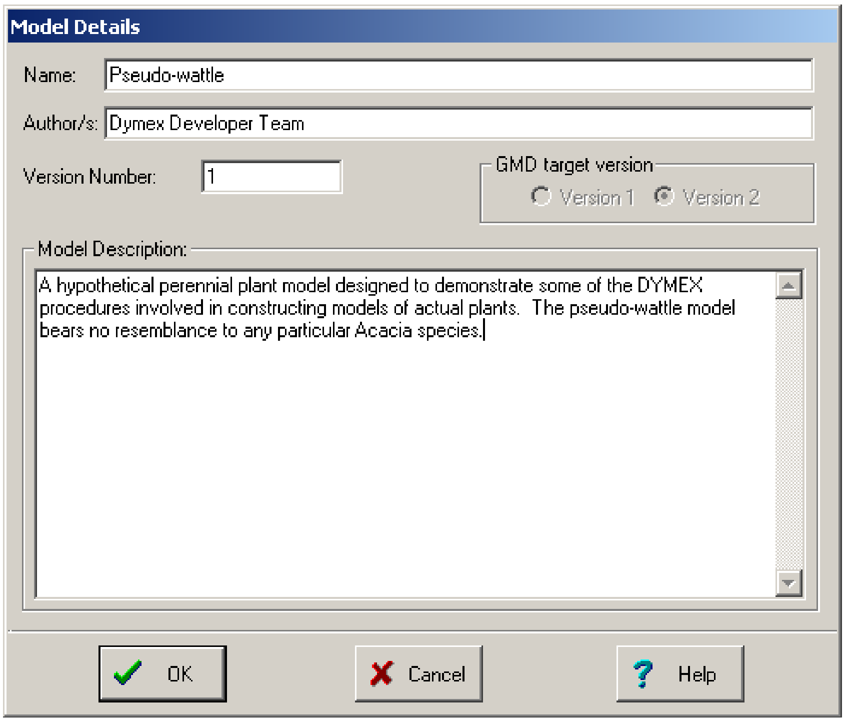


Figure 1.1 Model Details Window

3. Enter model details in the **'Model Details'** window and select **'OK'** on completion.

Once the above selection is made, the **'Model Component'** window appears (Figure 1.2). Currently, it will have no name other than the default, 'Document1', because the model has not yet been saved under any specific model name. The **'Model Component'** window lists the modules used in the model currently under construction. Since all models *must* have a **'Timer'** module, the **Builder** has inserted this module automatically.

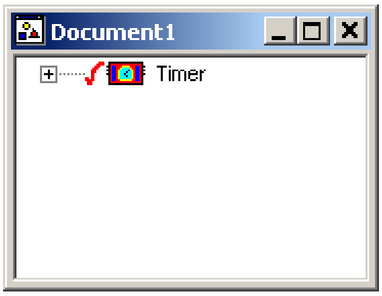


Figure 1.2 The DYMEX Model Window

The timer icon indicates that the line of information represents a module in the model while the red tick shows that the module has been specified sufficiently to allow the model to be run. Modules have Inputs, Outputs and Settings, and these can be examined by clicking on the small '+' button at the start of the line. This opens the Module to display its components as a 'tree diagram'. Further '+' buttons allow these components in turn to be examined in more (Figure 1.3). If the final text components of the tree diagram are "double clicked', their relevant windows can be opened for editing. A module can also be opened if the module name is 'double clicked' with the mouse and the module's functions, variables and parameters are then available for editing.

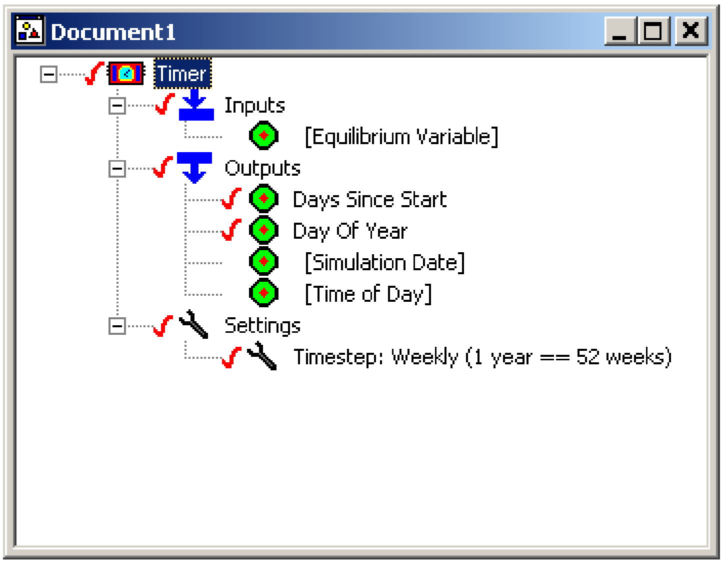


Figure 1.3 Timer module with 'Component Tree' opened. The tree shows the settings after the weekly timestep has been inserted

1.3.2 Building the Model

The Timer module is already in place, but it should be instructed to operate in weekly steps. A weekly time step is preferable for Pseudo-wattle because no processes act on smaller time-scales than this. Since Pseudo-wattle is such a long-lived organism, the use of a daily time-step would still allow the model to run, but excessively long times would be required to process the model.

1. 'Double click' on the 'Timer' text in the 'Model' window to open the 'Timer' module's dialogue box

The 'Timer' module's name cannot be changed and this is indicated by the fact that its module name is 'greyed out'.

2. Select the **'Settings'** button to obtain the **'Timer'** settings selection box and then examine the **'Model Timestep'** panel
3. Change the timestep from 1 day to 7 days by clicking on the **'7 (week)'** button
4. Check the "**Use exact years**" box
5. Select **OK'** and return to the **'Timer'** module's dialogue box
6. Select the **'Outputs'** button to open the **'Outputs (Timer)'** selection box
7. In a list box, **'Days Since Start'** should be highlighted - if it isn't, place the cursor on this choice and click once - it should then appear highlighted
8. Click once on the **'Select'** button - the symbol **' +> '** will appear in front of the text of **'Days Since Start'**
9. Similarly select **'Day of Year'**
10. Select **'OK'** and return to the **'Timer'** window
11. Select **'OK'** and return to the **'Model'** window.

Users may query the use of 'Days Since Start' rather than a choice such as 'Weeks Since Start'. DYMEX automatically increments the 'Days Since Start' output variable in 7-day steps for a weekly timestep model when it processes a simulation run or produces an output.

Once back in the **'Model'** window, open the **'Timer'** module tree. If all steps have been completed correctly, there should be red ticks now present in front of **'Days Since Start', 'Day of Year'** and **'Timestep: Weekly'** and the tree should be identical to the diagram of Figure 1.3. Small icons denote the function of each of the parts of a module.

The next requirement is a Lifecycle module.

1. From the menu bar, select "**Model**"
2. From the drop-down menu, select **"Add Module"**
3. From **"Create Module of Type ?"** list box, make sure that the Standard button is selected, and select **"Lifecycle"** from the list of modules
4. Select **"OK"**

The **'Lifecycle'** window will now be opened automatically with a **'Lifestage'** panel displayed (Figure 1.4). If the **'Model Component'** and **'Lifestage'** windows are kept open, the user can move back and forth between them using the standard Windows method of clicking on the exposed part of the window required. If either of two windows are set minimised, the required window can be obtained by using the menu bar command of the **'Window'** menu followed by selection of the required window from the drop down menu.

Important: When a model is loaded in the Model Builder, the **'Model Component'** window is always loaded by default. To obtain the **'Lifecycle'** window in this situation, double click on the line representing the lifecycle in the **'Model Component'** window. Once it is opened, the windows can be opened or changed as noted above. Any module in the **'Model Component'** window can be opened by double clicking on its text.

NOTE: ***To delete a module that has been accidentally created***, return to the 'Model Component' window and the unwanted module will be shown in the list of modules present. Click on the unwanted module so that it is highlighted. Next, select 'Model' from the main menu bar and obtain the drop-down menu; select 'Delete Module' and follow any required steps. When completed, the unwanted module will disappear from the listing.

At this point the lifecycle module can be given a name. Naming of the lifecycle module is not critical in this model, but DYMEX can be used to create models with multiple lifecycles and under those circumstances, the naming of individual lifecycles appropriately becomes very important:

1. With the Lifecycle window active, select "**Lifecycle**" from the menu bar
2. From the drop-down menu, select **"Properties"**
3. Type the name ***"Pseudo-wattle"*** into the **"Name"** edit box
4. Select **"OK"**

1.3.3 Constructing the Lifecycle

The **'Lifecycle'** window represents a life cycle and so far contains a single **'Lifestage'** panel (Figure 1.4).

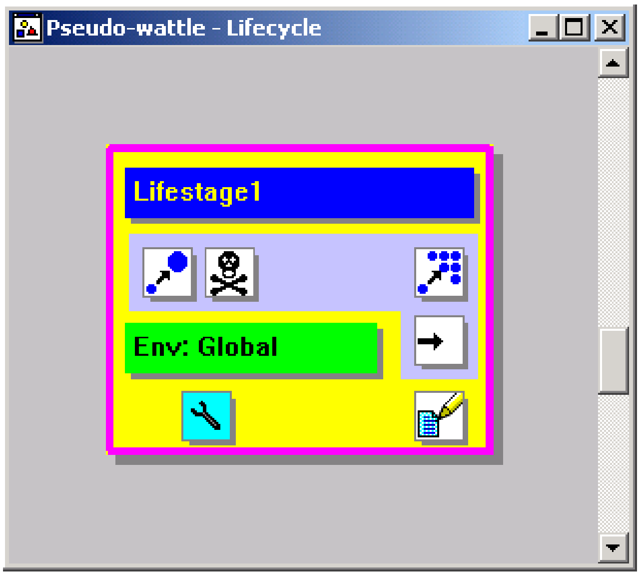


Figure 1.4 The Lifestage Panel

This panel represents one lifestage of the plant species being modelled. A number of environmental factors, (e.g., temperature, moisture, predators, nutrient availability, diseases, etc.) influence the species' rate of development, survival and reproduction. DYMEX can be set to simulate these processes by using the button icons in the Lifestage panel, each of which controls some aspect of the lifestage (The function of each button icon is described below). When an organism's life cycle is being developed, the Model Builder always shows which lifestage is currently selected by outlining it in magenta.

Each lifestage panel defines a particular stage (e.g., seed, seedling, juvenile plant, etc.) together with its environment and attributes. The total number of stages in the lifecycle is set by the user. For Pseudo-wattle, four stages are required (seed, seedling, juvenile plant and adult plant), however larger numbers of stages could be used depending upon the detail required by the model or the life cycle of the organism; for example, a model of an apple tree might include stages corresponding to seed, seedling, juvenile plant, adult plant and fruit; however it is conceivable that a user may wish to condense two of these three stages into one for a particular model and might end up with the three stages of seed, plant and fruit. Exactly how the model is constructed depends completely on the requirements and applications of the user.

The lifestage can now be given a name.

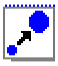
1. Select the **'Lifestage1'** button on the **'Lifestage'** panel to open the **'Lifestage Name'** edit box
2. Type **'Seed'** into the **'Name'** text entry box
3. Exit from the text entry box by selecting **'OK'**.

When choosing names for modules, components or variables, try to select descriptive names that allow easy recognition of the module, component or variable's application in the model.

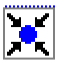
1.3.4 Lifestage Attribute Buttons

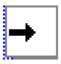
These buttons (located in the Lifestage panel) permit the user to open dialogue boxes in order to select variables or functions, name parameters and enter values for constants or variables. The functions represented by each button can also be accessed by choosing **'Lifestage'** from the main menu bar.

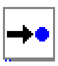
 **Lifestage Outputs**
The 'Lifestage Outputs' dialogue box opened with this button allows the user to select which variables will be used as outputs from that lifestage when the model is run in the 'Simulator'. Outputs may be tabular, graphical or written to a file.

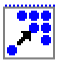
 **Development**
The 'Development' dialogue box is opened with this button. The user is allowed to select the functions and parameters controlling lifestage development and aging (i.e., the rate of accumulation of 'Chronological or Physiological Age').

 **Mortality**
The 'Mortality' dialogue box is opened with this button. The user is allowed to select the functions and parameters controlling the lifestage mortality rate.

 **User-defined Cohort Properties**
Within a model, certain cohort properties (such as chronological age, physiological age, fecundity, etc.) are pre-set. Situations can arise where the user needs to set new cohort properties for the particular organism being modelled: stress, plant size, sex ratio, etc. This button allows the user to define a set of functions which will control these properties. <u>It appears on the lifecycle icon only after at least one new cohort property has been correctly defined</u>.

 **Next Stage**
The **'Next Stage'** button adds a further lifestage. Once it has completed this operation, it changes and becomes a **'Stage Transfer'** button (see below) and consequently, only the last lifestage in a lifecycle will still have an operational **'Next Stage'** button. To remove an unwanted lifestage, select its panel, followed by **'Lifestage'** from the Main Menu bar of the window, and finally select **'Delete Stage'** from the drop-down menu.

 **Stage Transfer**
This button opens the 'Transfer Function' dialogue box in order to define/create or modify the transfer process which governs how an organism moves from one lifestage to the next. It is derived from the **'Next Stage'** button (see above).

 **Reproduction**

The 'Reproduction' dialogue box is opened with this button. Two processes, 'fecundity' and 'progeny production' (as well as their associated parameters) can be selected by the user.

 **Other Lifestage Properties**

This button opens the "Lifestage Properties" dialog, which gives access to various properties that may be used in more advanced models. One of these properties is the "Resource Variable", which is used as the divisor by the program when it is calculating density.

Environment Selection

This button allows the user to select the environment under which the model is to be run. By default a global environment is selected and this is all that is needed for this tutorial. The user is able to set up specific environments for particular models and then select them as required.

1.3.5 Completing the Lifecycle

Since Pseudo-wattle only occurs as seeds, seedlings, juvenile and adult plants, four lifestages are needed in the model.

1. Select the **'Next Stage'** button
2. Name the new lifestage **'Seedling'**
3. Now repeat this process to add two more lifestages and name them successively **'Juvenile'** and **'Adult'**.

Reproduction for the Adult Plant is modelled next.

1. Select the **'Reproduction'** button
2. In the **'Adult Reproduction'** window use the **'Destination Stage'** panel's scroll button to find and select **'Seed'**
3. Select **'OK'**.

A line ending in an arrow is now present which links the Adult Plant stage back to the Seed stage, and the whole structure should now resemble Figure 1.5. The arrows on the diagram define the direction of the flow of individuals within the life cycle.

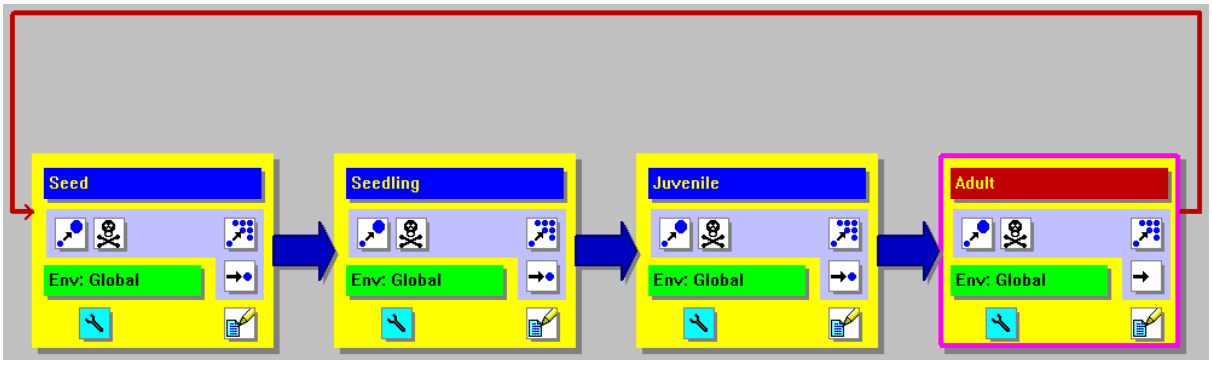


Figure 1.5 The completed life cycle structure for Pseudo-wattle

Notice that the **'Next Stage'** buttons for the Seed, Seedling and Juvenile Plant lifestages now have an extra blue dot. This indicates that they are now **'Stage Transfer'** buttons and are able to set the functional processes by which the next lifestage is reached.

1.3.6 Setting Lifestage Processes, Functions and Parameters

Once the structure of the life cycle has been defined, the physiological and ecological processes that define conditions under which individuals develop, die, and reproduce need to be specified. The lifestage buttons are used to select and define these relationships. Because (as presently defined) Pseudo-wattle has a very simple life cycle, not all of the buttons and their operations are needed in this first tutorial.

1.3.7 Completing the Seed Stage

In this initial form of the Pseudo-wattle model, seeds germinate after 8 weeks to become seedlings. Because of this simplicity, there are no requirements for development or mortality processes, so the **'Development'** and **'Mortality'** buttons are ignored for this stage. The **'Reproduction'** button is also not required because seeds do not reproduce. The **'Lifestage Outputs'** button is required to produce output from the stage and the **'Stage Transfer'** button is used to set the conditions under which the seeds become seedlings. The extreme simplicity of the model means there are very few output variables to consider.

1. Select the **Lifestage Outputs'** button to obtain the **'Seed Outputs'** list box
2. In the **'Seed Outputs'** list box, highlight **'Total Number'** then click on the **'Select'** button - once this is done, **'+>'** will appear beside the variable to indicate it is correctly selected
3. Select **'Rename'** button and type in a suitable name (e.g., 'Seeds: Total Number').
4. Select **'OK'** until back at the **'Lifecycle'** window.

The red tick that now appears on the 'Lifestage Outputs' button indicates that an output variable has been successfully selected for use by the model. **Always** give each variable a name that is easily recognisable and distinct from all others. DYMEX will suggest a name for the variable, but the user is advised to consider its suitability carefully and choose another if required. Good variable names greatly assist during the documentation of the model. Naming output variables by lifestage first, then variable type, groups conveniently the variables in scrolling lists. It is also useful to supply a description for the output variable in the **'Description'** panel so that a user of the model knows exactly what the variable represents. For this output, a description such as "The total number of Pseudo-aphid seeds in the seed-bank at any particular time" might be appropriate.

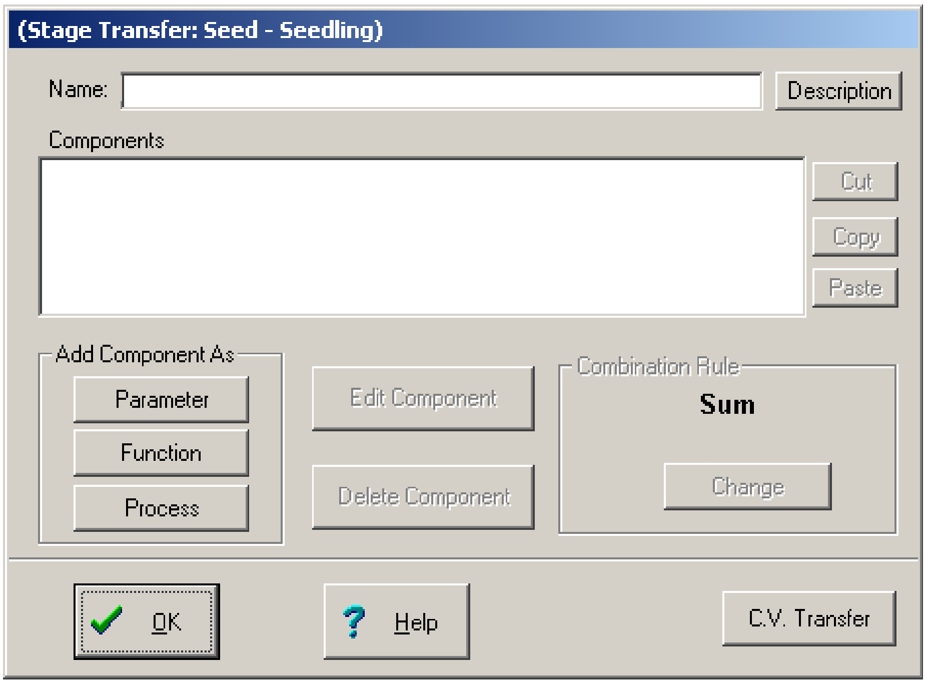


Figure 1.6 Seed - Transfer Selection Window

The **'Stage Transfer'** button is used to modify the process under which Pseudo-wattle changes from seeds into the seedling. DYMEX uses a standard selection window (Figure 1.6) for selection of variables, functions, sub-processes and parameters for lifestage processes. Processes consist of one or more components, each of which is evaluated to yield a process rate component. The rates are combined to give the total effect of this process. For example, the 'Seed-Transfer' rate evaluates, at each time-step, to the proportion of seeds that germinate to become seedlings during that time-step. Three buttons (**'Parameter', 'Function'** and **'Process'**) can be used to add process components, and the **'Edit Component'** button allows changes to be made to an existing process component. Components are deleted using the **'Delete Component'** button, whilst the **'Change'** button in the Combination Rule panel allows the user to select the method of component combination.

1. Select the **'Stage Transfer'** button in the Seed stage.

The **'Seed - Transfer'** window (Figure 1.6) is now used to select the functions and variables that define germination.

2. Type the name "Germination" into the **'Name'** edit field (since this process determines the germination rate).
3. Select the **'Function'** button in the **'Add Component As'** panel to obtain the **'Function'** selection window (Figure 1.7).

The **'Function'** selection window (Figure 1.7) is used to set the stage transfer function's independent variables and its associated parameters. DYMEX uses this window as a standard throughout the Model Builder. Under the large white panel at the top right is a scroll button which gives access to the library of mathematical functions contained in DYMEX. When a function is selected, its shape is illustrated on the screen. The **'Set Parameter'** button (in the Parameters panel) is used to set default and limiting values of the parameters for the function. The **'Set Function'** button allows a function parameter be replaced by another function. The **'Comments'** edit box should be used to record pertinent information such as a reference to the source of the data used to estimate parameter values or reasons for choosing this particular function shape. The **'?'** button provides a detailed description of the selected function, e.g., providing the equation for the function in mathematical notation and explaining the role of each parameter.

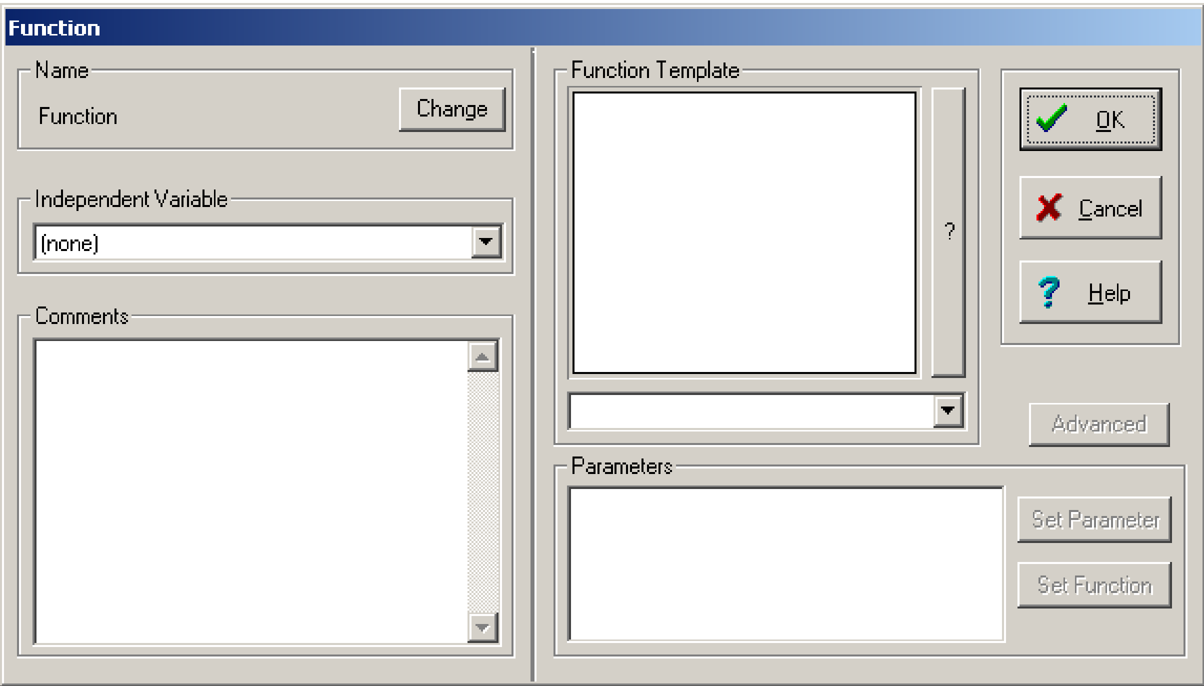


Figure 1.7 Function Selection Window

All seeds germinate to become seedlings 8 weeks (56 days) after dispersal, which implies the stage transfer variable is *Chronological Age* (a variable that reflects the age of individuals in days); and since all seeds germinate simultaneously, a step function is indicated.

4. Using the function scroll button, in the function template panel, select **'Step'**
5. Using the **'Independent Variable'** list box, select **'Chronological Age'**.

*A **'Step'** function will now be illustrated in the screen and **'Chronological Age'** will have appeared in the **'Name'** panel's box. The **'Step'** function which defines the **'Seed to Adult Plant Stage Transfer'** requires two parameters: (1) the chronological age(t_g) at which the seeds will germinate to become seedlings and (2) the proportion of those seeds that have reached the age t_g which actually germinate (become seedlings) during a time step. For the Pseudo-wattle model, the two parameters are the threshold (the chronological age in days) and the height of the step* *(the proportion of seeds becoming seedlings).*

6. Give the transfer function a suitable name by selecting the **'Change'** button (e.g., 'Germination function')
7. Select **'OK'**
8. Highlight the **'p1: Threshold'** line in the **'Parameters'** panel and click on the **'Edit Parameters'** button.

The **'Edit Parameters'** button opens the **'Set Parameter Properties'** dialogue box (Figure 1.8). The box shows the selected parameter, and allows its name ("User Name"), value and range to be set.

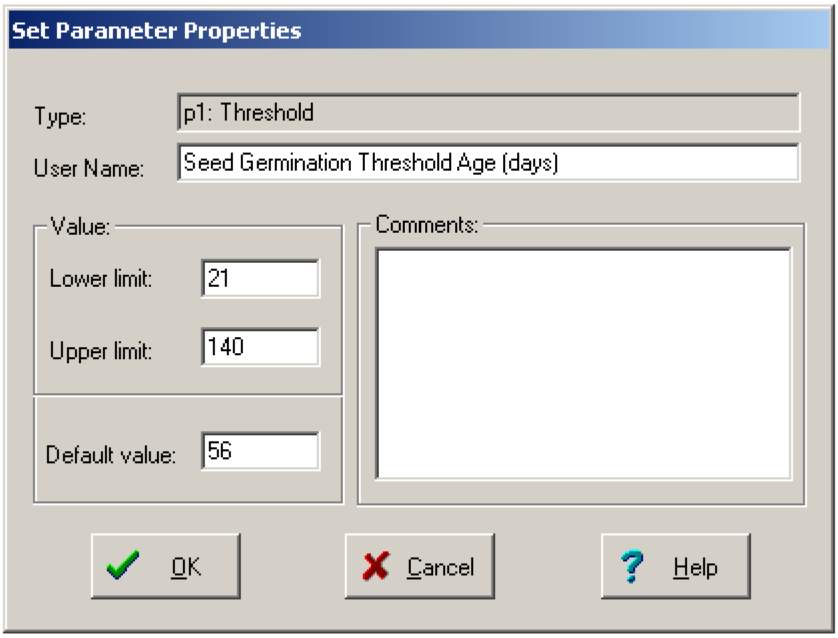


Figure 1.8 Set Parameter Properties Dialogue Box

The allowed range of the parameter and its default value are entered in three edit boxes titled 'Lower limit, Upper limit' and 'Default value'. For Pseudo-wattle, the default Threshold value is 56 (days); the lower and upper limits define the range over which the parameter can be varied while the model is in the DYMEX Simulator. If no limits are set, the Threshold value of 56 days can be varied to any value; if upper and lower limits are both set equal to 56, the parameter cannot be varied at all. With different limits (e.g., 24 and 140), the Threshold value is restricted to that range. An edit box allows a user-defined name to be inserted for the parameter and this should always be done as otherwise it will be easy to confuse parameters with the same (default) names in the DYMEX Simulator. A comment edit box is provided for explanatory remarks.

9. Select **'Lower limit'** edit box, type in the value 24
10. Select **'Upper limit'** edit box, type in the value 140
11. Select **'Default value'** edit box, type in the value 56
12. Select **'User-defined Name'** edit box, amend **'Threshold'** to a suitable name (e.g., 'Seed Germination Threshold Age (days)')
13. Type in comments if required by selecting **'Comments'**
14. Select **'OK'** to return to the **'Function Selection'** window.

The second parameter to be set in the 'Set Parameter Properties' dialogue box is the proportion of seeds that become adults at the 8-week point, and each week thereafter. DYMEX uses a decimal number to indicate this proportion; a value of 1 indicates ***all*** seeds become adult plants when they are 8 weeks old – i.e., all seeds produced in a particular week germinate to become adult plants simultaneously.

15. Highlight the **'p2: Step Height'** line in the **'Parameters'** panel and click on the **'Edit Parameters'** button to open the **'Set Parameter Properties'** window.
16. Select in turn each limit edit box and type in the value 1
17. Select **'Default value'** edit box, type in the value 1
18. Select **'User-defined Name'** edit box, delete **'Step Height'** and type in a suitable name (e.g., 'Proportion of seeds germinating')
19. Type in comments if required by selecting **'Comments'** Select **'OK'** in the dialogue boxes as necessary and return to the life cycle window.
20. Save the model by selecting **'File'** on the main menu bar, followed by **'Save'** from the drop down menu. (*For this initial occasion, **'Save'** will also open a sub-window allowing the user to set the name and location of the model file. Once a model file's name and location have been set, **'Save'** automatically saves the model file using those settings on all future occasions and does not reopen the file naming window.*)

The **'Stage Transfer'** button will now have a red tick to indicate its parameters are set and this completes the settings for the Seed lifestage.

1.3.8 Completing the Seedling and Juvenile Plant Lifestages

These intermediate lifestages now require completion. Since the procedures are almost identical to those already described in the Seed lifestage, only the outline is given here. Note that the values inserted for each lifestage refer to the time the organism spends in that lifestage, not the period since it was created as a seed. The mortality buttons are ignored for the moment, as it is assumed all individuals proceed to the next lifestage. Similarly, since only chronological age is determining an individual's progress through the lifecycle, the development buttons are unused.

a. To complete the Seedling lifestage:

1. Select the **'Lifestage Outputs'** button and ensure that **'Total Numbers'** is selected as an output - rename if required.
2. Select the **'Stage Transfer'** button and using a Step function with **'Chronological Age'** as the **'Independent Variable'**, set the threshold default value to 84 and lower and upper limits to 35 and 140. All Step Height values should be set to 1. Re-name the function and parameters appropriately.

b. To complete the Juvenile Plant lifestage:

1. Select the **'Lifestage Outputs'** button and ensure that **'Total Numbers'** is selected as an output - rename if required.

2. Select the **'Stage Transfer'** button and using a Step function with **'Chronological Age'** as the **'Independent Variable'**, set the threshold default value to 365 and the lower and upper limits to 210 and 490. All Step height values should be set to 1. Re-name the function and parameters appropriately.

1.4 Completing the Adult Plant Lifestage

The simplicity of the present Pseudo-wattle model means that the **'Development'** button can be ignored for the Adult Plant lifestage. Data output from the stage is obtained by selecting the **'Lifestage Outputs'** button. A red tick appears on each button after the dialogue and edit boxes have been set correctly.

1. Select the **'Lifestage Outputs'** button to obtain the **'Adult Plant Outputs'** selection window
2. In the **'Module Output Variables'** list box, highlight **'Total Number'** then click on the **'Select'** button
3. Select the **'Rename'** button and type in suitable name (e.g., 'Adults: Total Number'
4. Select **'OK'** until back at the **'Lifecycle'** window.

Because all Pseudo-wattle plants die eventually, the **'Mortality'** parameters need to be set. DYMEX offers three types of Mortality process: 'Continuous', 'Establishment' and 'Exit'. Continuous mortality operates throughout the duration of the lifestage; however certain species pass through life cycle stages where considerable mortality occurs when the organism tries to gain a 'foothold' in its new stage (e.g., orchid seeds released for germination - perhaps two out of a hundred thousand released will find a suitable substrate and form a new plant). Organisms such as this will require an 'Establishment' Mortality process in addition to, or instead of, the 'Continuous' Mortality process. In many cases it is not possible to ascertain how many deaths have occurred during a lifestage, until the end of the lifestage due for example to the cryptic nature of a lifestage. In such cases it may be necessary to employ an 'Exit' mortality process. The Pseudo-wattle adult requires only the 'Continuous' process. Since the adult plants all die at the end of a fixed time period (8 years), their mortality-inducing variable is *Chronological Age*, and once again a "Step" function provides the required characteristics (no mortality until a particular value of *Chronological Age*, and then all die). The threshold value will be 2922 (8 years x 365.25 days) and the constant value will be 1 (Note once again that the *Chronological Age* variable contains the length of time individuals have spent **within the stage**).

1. Select the **'Mortality'** button on the **'Adult Plant'** lifestage
2. From the **'Adult Plant Mortality'** selection box, select the **'Continuous'** button
3. In the **'Mortality (Adult)'** selection window, type **'Mortality'** into the **'Name'** panel
4. Select **'Function'** to obtain the **'Function'** selection window
5. Using the function scroll button, select **'Step'**
6. Using the **'Independent Variable'** list box, select **'Chronological Age'**
7. Select the **'Change'** button, suitably rename the function (e.g.,'Mortality due to Age') and then select **'OK'**

8. Highlight the **'p1: Threshold'** line in the **'Parameters'** panel and click on the **'Edit Parameters'** button to obtain the **'Set Parameter Properties'** dialogue box
9. Select the lower and upper limit edit boxes and type 2922 in each
10. Select **'Default value'** edit box, type in the value of 2922
11. Select **'User-defined Name'** edit box, delete 'Threshold' and type in a suitable name (e.g., 'Adult Lifetime')
12. Type in comments if required by selecting **'Comments'**
13. **Click** Select **'OK'** to return to the **'Function Selection'** window.
14. Highlight the **'p2: Step Height'** line in the **'Parameters'** panel and click on the **'Edit Parameters'** button to open the **'Set Parameter Properties'** window.
15. Select the lower and upper limit edit boxes and type 1 in each
16. Select **'Default value'** edit box, type in the value of 1
17. Select **'User-defined Name'** edit box, delete **'Step Height'** and type in a suitable name (e.g., 'Proportion of adults dying')
18. Type in comments if required by selecting **'Comments'**
19. Select **'OK'** as necessary to exit and return to the life cycle window.

Note again how a red tick will be displayed on the 'Mortality' button once the dialogue boxes are all closed, indicating that all functions and parameters are correctly set.

Reproductive parameters are set from the **'Reproduction'** dialogue box (Figure 1.9 and its three components are **Fecundity (E), Fecundity (R)** and **Progeny Production**. In DYMEX, **Fecundity** is a property that specifies the number of 'potential' seeds that are available to be produced (by the **Progeny Production** process) at any time. "**Fecundity (E)"** (where the E stands for 'establishment') is the process that that establishes a value for Fecundity when an individual enters the lifestage. By contrast, **Fecundity (R)** (R = 'recharge') can be used to "recharge" the Fecundity periodically. **Fecundity (E)** is most appropriate in models of insects, as adult insects can be considered to have a potential egg laying capacity (often dependent on larval nutrition) that is then exhausted through the adult stage. The same would also apply to annual plants. For most perennial plants, however, the fecundity will be determined annually based upon factors such as the size of the plant and the growing seasonal growing conditions prior to flowering. Hence, the **Fecundity (R)** process would be most appropriate for such models.

In the case of our very simplified Pseudo-aphid model, we will use the simpler **Fecundity (E)** process. Since an adult Pseudo-wattle plant produces 30 seeds per year for eight years, it has a lifetime fecundity of 240 seeds. If the value is set to 500, it leaves room for the user to experiment with some alternative values in seed production. Fecundity will usually vary with environmental factors, but in this simplistic model, this will not be considered.

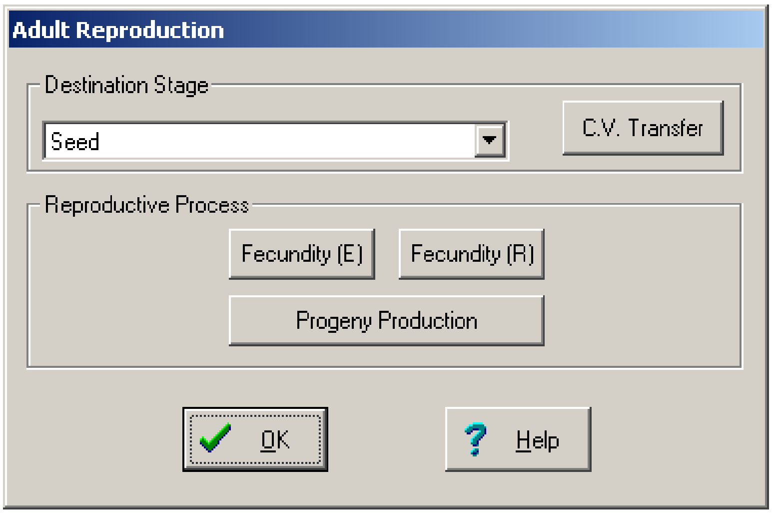


Figure 1.9 Reproduction Dialogue Box

'Progeny Production' defines the rate at which the seeds are produced. For example, some plant species produce all their seeds in a once only batch; others produce batches of offspring at regular/irregular intervals, while others may steadily increase production of seeds as the plant reaches full maturity and then decrease the production rate gradually to zero with senility. For Pseudo-wattle, the model will initially assume that all seeds are produced in a single batch, which only requires a step function.

1. In the **'Adult Plant'** lifestage, select the **'Reproduction'** button
2. Select the **'Fecundity (E)'** button to obtain the **'Adult Plant - Fecundity'** selection window
3. Name the process 'Fecundity'

The 'Fecundity' window has the standard format. Nothing in the Pseudo-wattle model affects fecundity (i.e., it is a constant at 500 for every adult plant). Hence we can set it as a parameter rather than a function.

4. Select the **'Parameter'** button in the **'Add Component As'** panel to obtain the **'Set Parameter Properties'** text entry window
5. Select the **'Default value'** edit box and enter 500, then set the lower and upper limit values to 100 and 10000 respectively
6. Select **'User-defined Name'** edit box and enter a suitable name (e.g., 'Fecundity')
7. Select **'OK'** until the **'Adult - Reproduction'** dialogue box is reached.

*Since each Pseudo-wattle plant produces a batch of 30 seeds each year in one week in mid-September (for example, day 258), a pulse function is indicated. This will effectively send an annual flowering 'trigger' to all adult plants of any age. The driving variable will be the **'Day of Year'**. For this pulse function, the threshold is the day seed production commences and the pulse height is the number of seeds produced per plant per annual season.*

8. Select **'Progeny Production'** to obtain the **'Progeny Production (Adult)'** selection window

9. Name the process 'Seed Production'
10. Select **'Function'**
11. Using the function scroll button, select **'Pulse'**
12. Using the '**Independent Variable'** list box, select **'Day of Year'**
13. Select **'Change'** and rename the function suitably (e.g., 'Seed production')
14. Highlight **"p1: Threshold"** and select the **'Edit Parameters'** button to obtain the 'Set Parameter Properties' dialogue box
15. Select **'Lower limit'** edit box, type in the value 258
16. Select **'Upper limit'** edit box, type in the value 258
17. Select **'Default value'** edit box, type in the value 258
18. Select **'User-defined Name'** edit box, amend 'Threshold' to a more suitable name (e.g., 'Day of Seed Production')
19. Type in comments if required
20. Click **"OK" to return to the "Function"** window
21. Highlight **"p2: Pulse Height"** and select the **'Edit Parameters'** button to obtain the 'Set Parameter Properties' dialogue box
22. Set the limits to 10 and 80 and the default to 30
23. Select **'User-defined Name'** edit box, delete **'Pulse Height'** and type in a suitable name (e.g., 'Seeds per plant per season')
24. Click **"OK" to return to the "Function"** window
25. Highlight **"p2: Pulse Width"** and select the **'Edit Parameters'** button to obtain the 'Set Parameter Properties' dialogue box
26. Set the limiting and default values to 7
27. Name the parameter suitably (e.g., 'Seed Production Period')
28. Type in comments if required
29. Select **'OK'** as required and exit to the life cycle window
30. Save the model, preferably into the C:\Program Files\Dymex2\Models directory with the name Pseudo-wattle1.gmd.

Assuming all has been correctly done, there will now be 9 red ticks on the life cycle diagram; one each on the 'Lifestage Output' and 'Lifestage Transfer' buttons of the Seed, Seedling and Juvenile lifestages, and one each on the 'Lifestage Output, Mortality' and 'Reproduction' buttons of the Adult stage.

31. Return to the **'Model'** window.

While in the **'Model'** window, it is worth seeing first how steps 1-5 have altered the model, and second, examining the model by expanding its 'tree diagram'. Try clicking once on the **'+'** for the **'Lifecycle'** module; then try opening each **'+'** button as it is reached. Eventually, the ends of each branch will be the parameter values that have been set during the procedures just covered. The values can be edited from the **'Model'** window by double clicking on the terminal text values of the tree diagram; the values are then altered from the resulting windows.

****

This completes the formation of the initial Pseudo-wattle model. You may now wish to use the Simulator to examine the model immediately. Before exiting from the Model Builder, read the first paragraph of the next part of this tutorial, which involves use of the Simulator.

1.5 Using DYMEX to Run the Model

1.5.1 Starting the Simulator

The DYMEX Simulator can be started from the desktop either by selecting its desktop icon, or from the **'Start'** button by selecting **Programs**, then **DYMEX 2** and finally choosing the **Simulator** for the options. If the Model Builder is open, DYMEX provides a short cut to the Simulator. After having saved the Pseudo-wattle file, select **'Model'** on the main menu bar and then select **'Run'** from the drop down menu. This procedure does not ***close*** the Model Builder, so that it will remain in memory.

The Simulator can provide the user with hints on operation and also has a button bar which allows a number of short cuts. The **'Hints'** dialogue box can be turned on or off during any operating session by selecting **'Preferences'** from the main menu bar followed by **'Show Hints'** in the drop down menu: a tick will appear beside **'Show Hints'** while the **'Hints'** dialogue box is present on the screen. The **'Hints'** dialogue box can be turned off permanently by setting the user preferences - see 1.5.2 below.

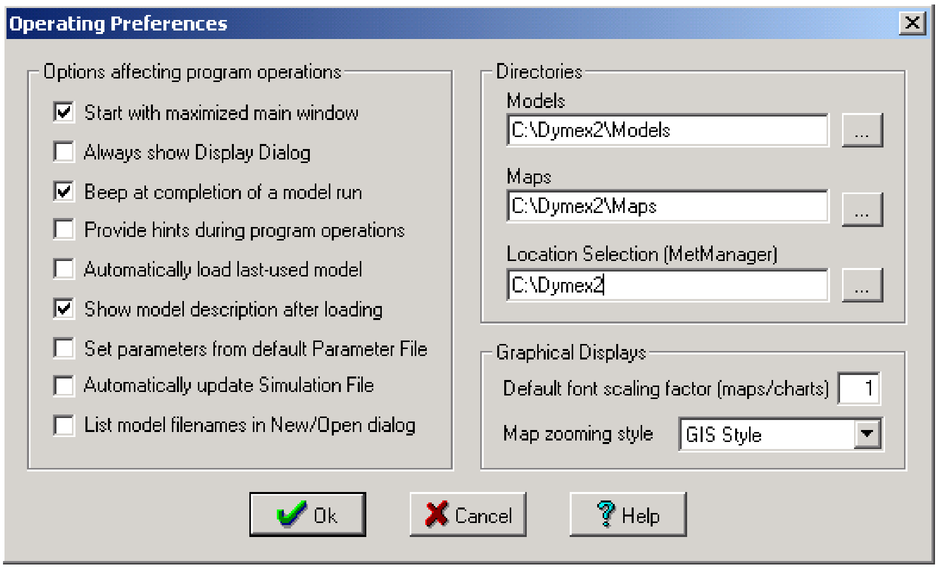


Figure 1.10 Simulator Operating Preferences Selection Box

1.5.2 User Preferences

The 'Operating Preferences' selection box allows personal preferences to be set for the operating conditions of the Simulator. Until the user is more familiar with DYMEX, the default settings (Figure 1.10) are likely to prove acceptable.

1. Select **'Preferences'** from the menu bar
2. Select **'Operating'** from the drop-down menu
3. In the **'Operating Preferences'** selection box check that the options are set to

the defaults shown in figure 1.10

4. Select **'OK'**.

1.5.3 Loading Files

The model that was saved in the Builder is a Model Description (gmd) file and contains a representation of the model's structure. The Simulator does not load these files directly. Instead it loads a ***Simulation File*** (.dxs or .ini), which contains settings for a simulation run using the model. Each model can have many Simulation Files (for example, if a particular model is needed to run at 5 locations, a separate Simulation File with the appropriate settings could be used for each location). Since this is a new model, no Simulation File exists for it yet. Therefore one must be created and this is done using the **New** option on the **File** menu (also accessible via the button bar's icon). This opens the **New Simulation** dialog (Figure 1.11). It lists all of the models in the model directories in the panel at the left.

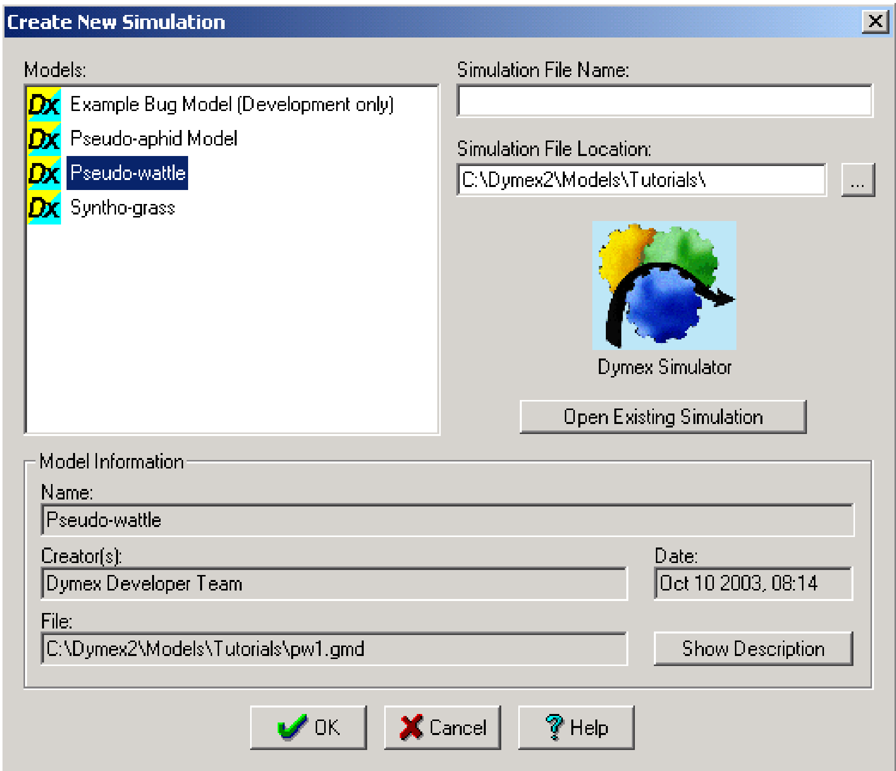


Figure 1.11 The "New Simulation" dialog

To create a new simulation file:

1. From the **'Models'** selection window at the left, select the "Pseudo-wattle" file by clicking on it. The name that appears in this list is the one you provided in the **"Model Details"** window in the DYMEX Builder. The **"File"** panel at the bottom of the dialogue will list the path where the pseudo-wattle file is stored.
2. In the **"Simulation File Name"** panel type in a suitable name for the new

simulation (e.g., "Pseudo-Wattle Tutorial").

3. Select the **'OK'** button on the **'New Simulation'** window

DYMEX now loads and checks the model 'gmd-file'. It will also inform you that it is creating a new parameter file for the model. Any problems found are reported as error messages. Finally, if no serious errors were found, the 'Model Description' window is displayed.

4. Select the **'OK'** button on the **'Model Description'** window.

While operating the Simulator, the user can alter parameter values within the ranges set by the default and limiting values that were incorporated into the model while it was being built in the Model Builder. The Simulator is prohibited from altering the master 'gmd-file' and so during the loading of a file, the program makes a working copy of the files' parameters, the 'gmp-file', which can then be altered as the user requires. The 'dxs-file' is a record of the user's personal settings for the Simulator. Normally, neither the 'gmp' nor the 'dxs' files require any direct user action, but it is useful to know that alterations to the working file in the Simulator do not mean that the original model values have been destroyed.

If, at any time the gmd-file is copied to another computer, or placed in a different directory, it is useful to copy the 'dxs' and 'gmp' files also. If they are not copied across, all user settings will have to be re-entered before the model will run in the new location.

From this point on, the tutorial assumes that the Pseudo-wattle model has been correctly built and is loaded in the Simulator.

Once a model file is successfully loaded, the Simulator window changes: additional items appear on the main menu, more buttons on the button bar are activated, the Simulator status is shown on the bottom window bar, and the 'Model Components' window appears (Figure 1.12).

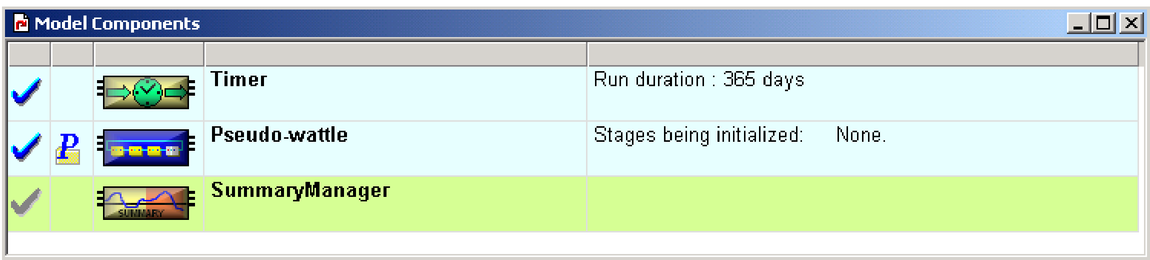


Figure 1.12 Fully initialised 'Model Components' window with the Timer set to one years (365 days) run duration.

The **"Model Components"** window indicates that a model is loaded in the Simulator and the window lists the number and type of modules present. A tick must be present before every module symbol before the model can be run. Any modules without such a tick need to be initialized. Figure 1.12 was produced by the Simulator after loading the pseudo-aphid model: it indicates that both the "Timer" and "Lifecycle" modules are correctly constructed and initialised, so that the model is ready to run. Note, however, that the Lifecycle module may require further initialization. If the model is run at it stands, the Total Numbers output for all lifestages will be zero at all times as no individuals have been specified to 'seed' the population. The Parameter icon (P) in front of the Lifecycle module symbol indicates that this module contains user-settable parameters. Note also the special "SummaryManager" module, which is present in every DYMEX model. It is not used in our very simple Pseudo-wattle model and can be ignored, but is important in more complex models. If the "Close" button ✕ at the top right corner is clicked, the model file will be closed and must be re-loaded if further work is required.

1.5.4 Module Initial Settings

Assuming all modules are shown as correctly constructed, initial settings will need to be entered or reset for each module. There are two ways of selecting the initialisation dialogue boxes: either from the menu bar by selecting **'Initialisation'** or from the **'Model Components'** window. Each method opens the same series of dialogue and edit boxes.

Each module icon within the 'Model Components' window acts as a button to a dialogue box and module settings can then be made. Since an annual flowers once a year, two years is the minimum time required to examine the progress of the model.

1.5.4.1 Timer Module

1. Click on the **'Timer'** module icon
2. From the drop-down menu, select **'Initialise Module'** to open the **'Simulation Duration'** edit box (Figure 1.13)
3. Set the simulation run to commence on 1/1/1980
4. Set the simulation run length to 3650 days (10 years)
5. Select **'OK'**.

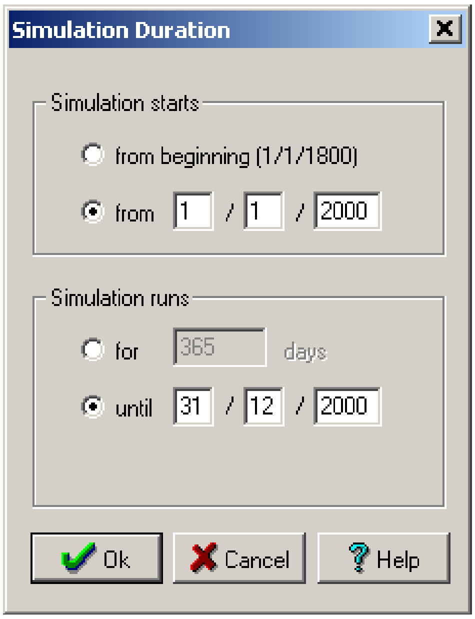


Figure 1.13 Simulation Duration Edit Box

1.5.4.2 The Lifecycle Module

1. Select the **'Lifecycle'** module icon
2. From the drop-down menu select **'Initialise Module'**

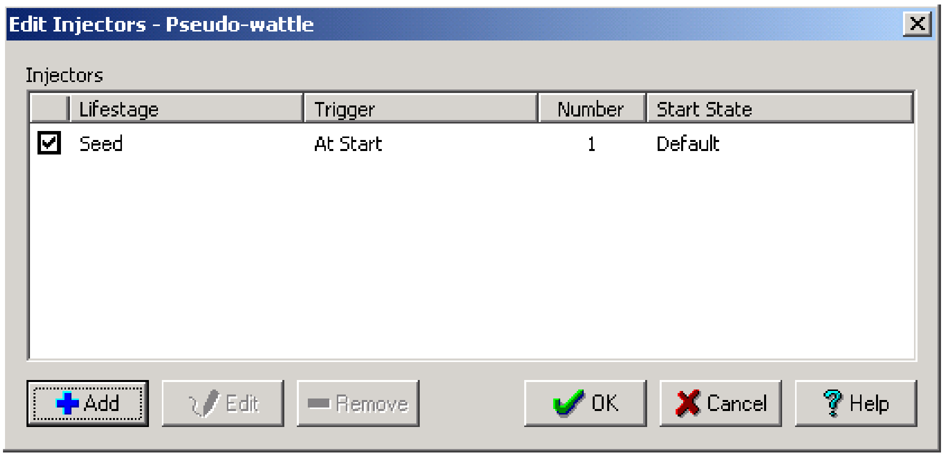


Figure 1.14 Initialise Lifestage Numbers Window after 'Seed' has been initialised with one individual

The 'Initialise Lifestage Numbers' window (Figure 1.14) allows the user to set the initial number of individuals present in each lifestage. No initialisation settings are present at this time. DYMEX will still run the Pseudo-wattle model and produce results for such quantities as development times, however it will not produce Pseudo-wattle population numbers because there are no individuals within the model. This is the situation that was implied by the discussion on the previous page. So that DYMEX can produce a useful output, a single individual will be added to the Seed lifestage to start the population.

1. Click on the 'Add' button in the Lifestage initialisation dialog
2. In the 'Edit Injector' dialogue, select the seed stage from the Lifestage selection box by clicking on the small button to the right and choosing **'Seed'** from the drop-down list.
3. Make sure that the **'AtStart'** Trigger is selected – it adds the individuals to the population at the start of a simulation.
4. Type the value 1 into the **'No of Individuals' edit box**
5. The default "Start State" should be selected – verify that this is so.
6. Select **'OK'**

This returns the user to the 'Initialise Lifestage Numbers' window, and if it has been correctly set, the list box will show that the Seed lifestage has been initialised with one individual at the start.

7. Select **'OK'**.

The current structure of the Pseudo-wattle lifecycle can be shown using the 'Lifecycle' module icon.

1. Select **'Lifecycle'** module icon

2. From the drop-down menu select **'Show Lifecycle Diagram'**

*A diagram will appear showing the lifecycle of the Pseudo-wattle. Like the 'Model Components' window (*Figure 1.12*), the lifestages of the lifecycle can be used as button icons.*

3. From the **"Lifecycle"** menu, select the "query mode" menu item. The mouse cursor will turn to a question mark when you hover over the lifecycle window.
4. Select the **'Seed'** lifestage box by clicking on it. The **"Lifestage - Seed"** window appears and the mouse cursor turns to a magnifying glass when it is over the **"Seed Germination"** graph.
5. Click on the **"Seed Germination"** graph to see the germination function in more detail. Note how you can change parameter settings by clicking on each of the parameters.
6. Close the **"Function Details: Germination function"** and **"Lifestage - Seed"** windows
7. Select the **'Adult Plant'** lifestage icon. Note the different colour-coding of the functions by type of process.
8. Close the Adult Lifestage and Lifecycle windows

1.5.5 Running the Model 

DYMEX allows two methods of starting a model run using either of the menu or the button bars. From the menu bar, **'Execution'** produces a drop-down menu containing **'Run'**, or the **'Run'** button (lightning flash) on the button bar produces the same result.

1. Select  from the button bar to run the simulation.

The model will now run and a 'Running Model' window (Figure 1.15) will appear briefly to indicate the progress of the Simulator.

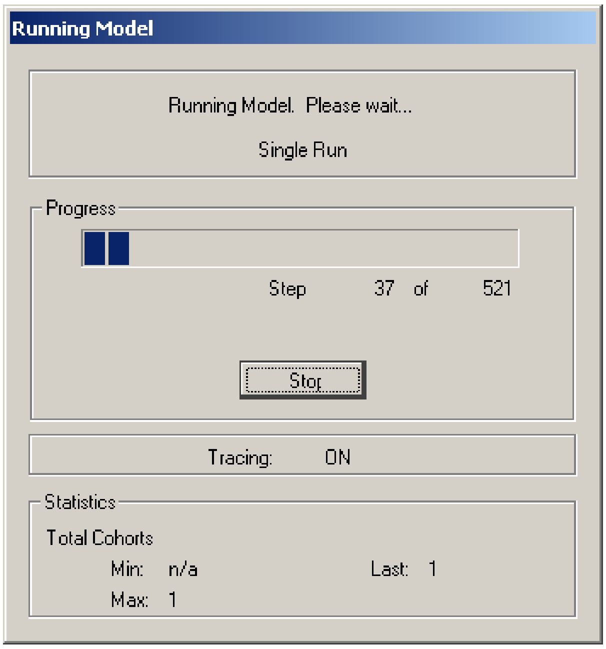


Figure 1.15 Running Model window

On completion, a 'Run' window will appear (Figure 1.16) which summarises information on the run. For Pseudo-wattle, this is very limited due to the simplicity of the model, however the 'Run' window summary becomes more complex in direct relation to the complexity of the model.

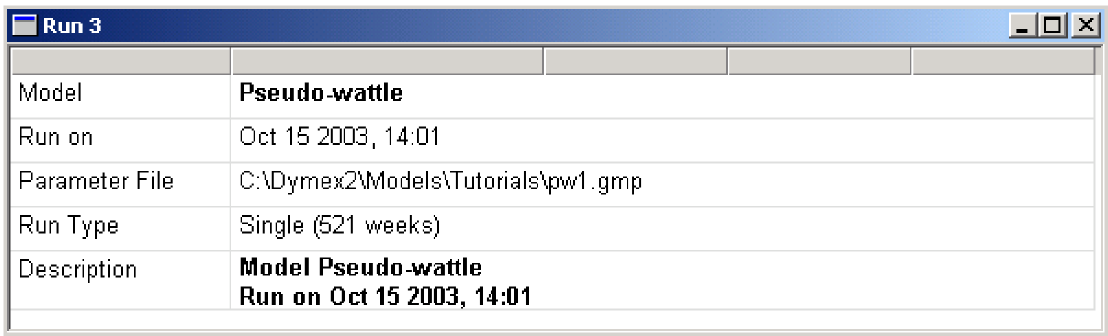
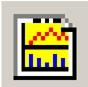


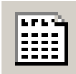
Figure 1.16 Run window

1.6 Producing Model Outputs

1.6.1 The Button Bar

On completion of a run, DYMEX outputs may be obtained using either the menu bar or the button bar. If the menu bar is used, select **'Results'** and then choose from the drop-down menu. The button bar offers the same processes with a single selection. Since a model has now been run, two more buttons on the button bar are activated which the model output to be presented in either chart or tabular form respectively. Using a series of dialogue and edit boxes the user can define the format of the presentation.

 - Charts output button.

- Tables output button.

1.6.2 Opening Table Displays

1. Select the  button

The **'Select Variables for Table'** dialogue box is now open (Figure 1.17) with seven variables presented in the **'Available Variables'** list box. **'Step'** is the time interval used in calculating the lifecycle and for the Pseudo-wattle model is measured in weeks. **'Days Since Start'** counts the number of days since the commencement of the run. The remaining variables were selected for the Pseudo-wattle model during its building in the Model Builder. Any combination of the variables can be selected for presentation in the output.

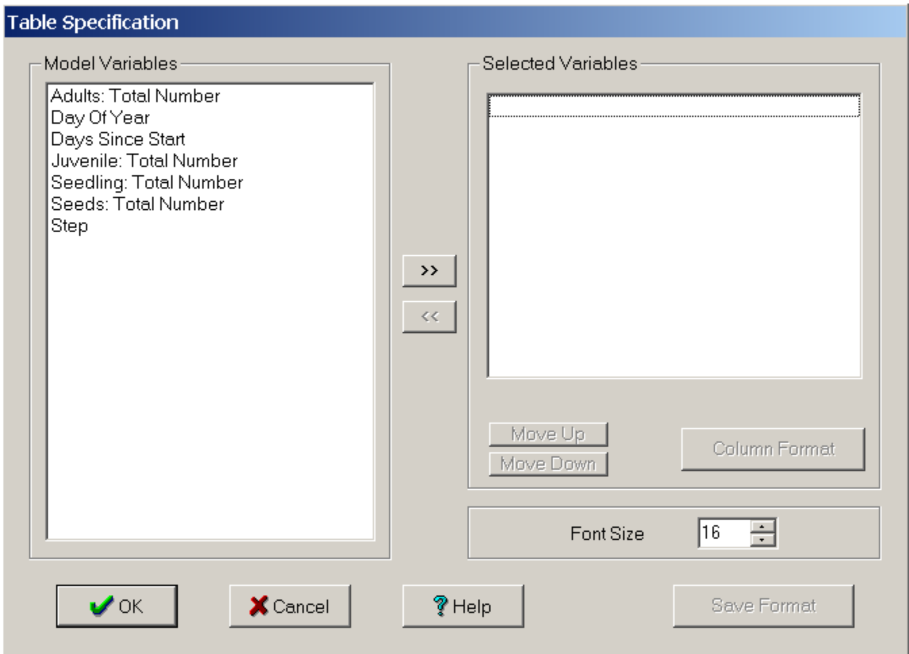


Figure 1.17 Select Variables for Table Selection Box

1. From the **'Model Variables'** list, select *'Step'* by clicking on it.
2. Clicking on the  button to move this variable into the **'Selected Variables'** list.
3. Repeat steps 1 & 2 above for the remaining variables.
4. Use the **Move Up** and **Move Down** buttons to place the selected variables into the following order: *Step*, *Day of Year*, *Days since Start*, *Seeds: Total Number*, *Seedling: Total Number*, *Juvenile: Total Number* and *Adult: Total*

Number.

The procedure used above has selected all variables for the output table. Variables can be removed from the **'Selected Variables'** list box by highlighting them and then selecting the << button.

DYMEX can format any selected variable. In figure 1.17 the **Format** button is greyed out, but after the selection of any of the available variables, the **Format** button becomes active and opens the **'Table Format'** window (Figure 1.18), which allows the table's data to be displayed in any suitable format. Each column can be selected individually and its required format set. The format options include whether: numerical data will be displayed as integers or floating point decimals, the number of decimal places to be used if the floating point option is used, the width for each data column, and whether or not a data column should be shaded for emphasis. Once a table format is fully defined, it can be saved using the **'Save Format'** option in the **'Select Variables for Table'** selection box (Figure 1.17). The **'Save Format'** option opens a standard dialogue box in which the name of the format can be entered for future use.

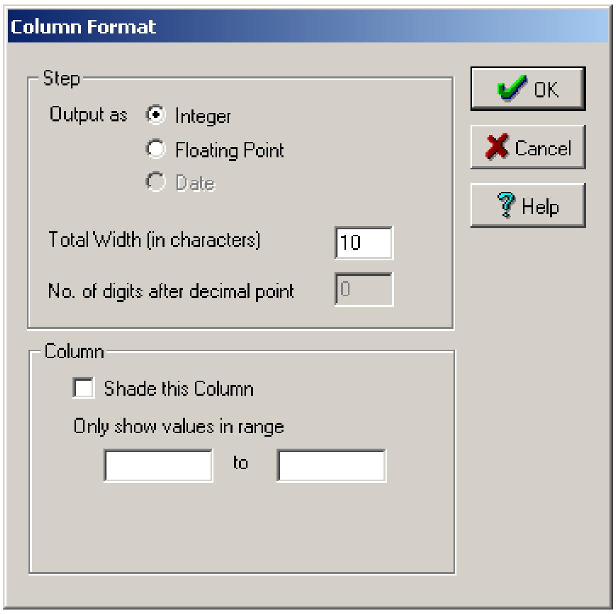


Figure 1.18 Table Format window

5. With *'Step'* highlighted, select the **'Format'** button and set the output to integers with the column width to 6 characters, then select **'OK'**
6. Repeat step 5 for *Day of Year* and *Days since Start*.
7. With *'Seeds: Total Number'* highlighted, select the **'Format'** button and set the output to floating point with the column width to 8 characters and 2 digits after the decimal point, then select **'OK'**
8. Repeat step 7 for *Adult: Total Number*, *Juvenile: Total Number* and *Seedling: Total Number*.
9. With *'Adult: Total Number'* highlighted, set its display to shaded format and then select **'OK'** to return to the **'Select Variables'** window
10. Finally, select **'OK'** to produce the output table (Figure 1.19).

DYMEX constructs condensed headings for each of the columns of the table. The data can be printed in full by selecting the **'Print'** button on the button bar. The **'Print Preview'** selection will show the table as it will look on the printed page before printing. Selected parts of the table can also be printed separately, by first marking the required area followed by **'Print Selection'** from a drop-down menu (see below).

Table - Table-1 [Run 1]

Step	Day Of Year	DaysSincSta	SeedTotaNumb	SeedTotaNumb	JuveTotaNumb	AdulTotaNumb
Model Pseudo-wattle Run on Dec 05 2003, 07:51						
1	1	1	1.00	0.00	0.00	0.00
2	8	8	1.00	0.00	0.00	0.00
3	15	15	1.00	0.00	0.00	0.00
4	22	22	1.00	0.00	0.00	0.00
5	29	29	1.00	0.00	0.00	0.00
6	36	36	1.00	0.00	0.00	0.00
7	43	43	1.00	0.00	0.00	0.00
8	50	50	1.00	0.00	0.00	0.00
9	58	58	0.00	1.00	0.00	0.00
10	65	65	0.00	1.00	0.00	0.00
11	72	72	0.00	1.00	0.00	0.00
12	79	79	0.00	1.00	0.00	0.00
13	86	86	0.00	1.00	0.00	0.00
14	93	93	0.00	1.00	0.00	0.00
15	100	100	0.00	1.00	0.00	0.00
16	107	107	0.00	1.00	0.00	0.00
17	114	114	0.00	1.00	0.00	0.00
18	121	121	0.00	1.00	0.00	0.00
19	128	128	0.00	1.00	0.00	0.00
20	135	135	0.00	1.00	0.00	0.00
21	142	142	0.00	0.00	1.00	0.00
22	149	149	0.00	0.00	1.00	0.00

Figure 1.19 Formatted output Table for Pseudo-wattle (first 22 steps of 3650 day run)

A **'Quick Graph'** procedure is available directly from the **'Tables'** display and it automatically uses the date (derived from the simulation starting date and the 'Days since Start' variable) for the X-axis. To commence the 'Quick Graph' procedure, place the cursor in the data column for which a 'Quick Graph' is required and 'double click' the left mouse button. A drop-down menu is produced which permits four options: saving of the table data in a separate file, examination of the variable description, changing the column format and production of a 'Quick Graph'.

1. Place the cursor in the **'Seeds: Total Number'** column ("SeedTotaNumb") and double click the left mouse button
2. From the drop-down menu, select **'Variable description'**
3. After inspecting the **'Details of Variable'** list box, select **'OK'**

The **'Details of Variable'** list box summarises all the information about the particular variable of the selected column and the procedure is available for any column.

1. Select the **'Seeds: Total Number'** column by double clicking
2. From the drop-down menu, select **'Quick Graph'**

The seed population (Figure 1.20) is not shown during the early years, as the numbers of seeds at that time are too small to register on the graph when the numbers of seeds at the end of the run are included. A better way of displaying these results is to use a logarithmic display and this will be covered later in the graph display section.

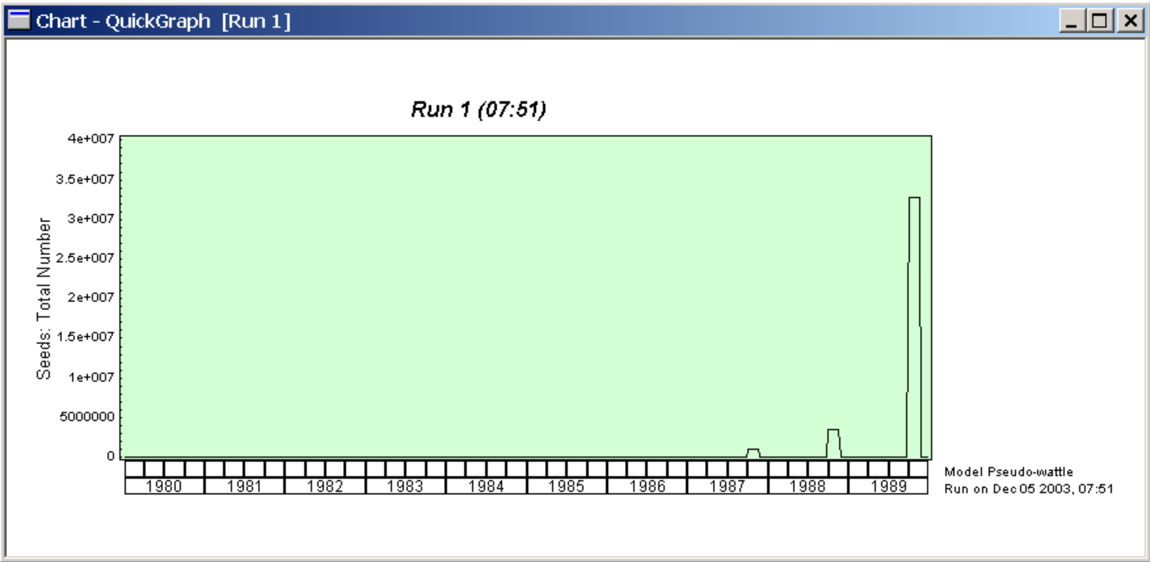


Figure 1.20'Quick Graph' for Total Numbers of Seeds

The populations of Seeds and Adult Plants have graphs that are shaped essentially the same, except that for each moment of time the sizes of the two populations will differ. The 'curve' is discontinuous and shows distinct 'gaps' where the seeds transfer to adult plants. The size of the gaps depends on the settings for the maturation period required for the seeds and the time of seed production. Over a longer period of time, the numbers show an extremely rapid increase.

While in the table mode, an 'area selection' mode is also available. If the cursor is 'clicked' on a column, it can be used to 'mark/select' an area of the table by the standard 'Windows' technique of holding the left mouse button down while dragging the cursor across the required area. When this is done the area is highlighted and the right hand mouse button can then be used to open a drop down menu while the cursor remains in the highlighted area.

The drop-down menu contains the following options, which are explained here for use as required:

Copy Selection	Places a copy of the selected area into the clipboard. The copy can then be accessed by the usual "Paste" command of the wordprocessor, spreadsheet or other program in use.
Page Setup	Allows setting of some properties for the printed table.
Print Selection	Sends the highlighted area to the printer. The dialog box that

appears refers **strictly** to the highlighted area.

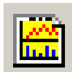
Print Table Prints the whole table.

This completes the introduction to the table displays and the user should now close the table by using the standard 'Windows' procedure:

1. Select top right 'Windows' X-button of the Table and 'click' once.

1.6.3 Opening Chart Displays

DYMEX provides two methods of presentation for chart outputs: the charts may be either separate or on a common X-axis and exemplars are shown on the panel (figure 1.25). The default selection is 'Common X-axis'.

1. With the Run window active, select the Chart button  to open the **'Chart Specification'** window (Figure 1.21)

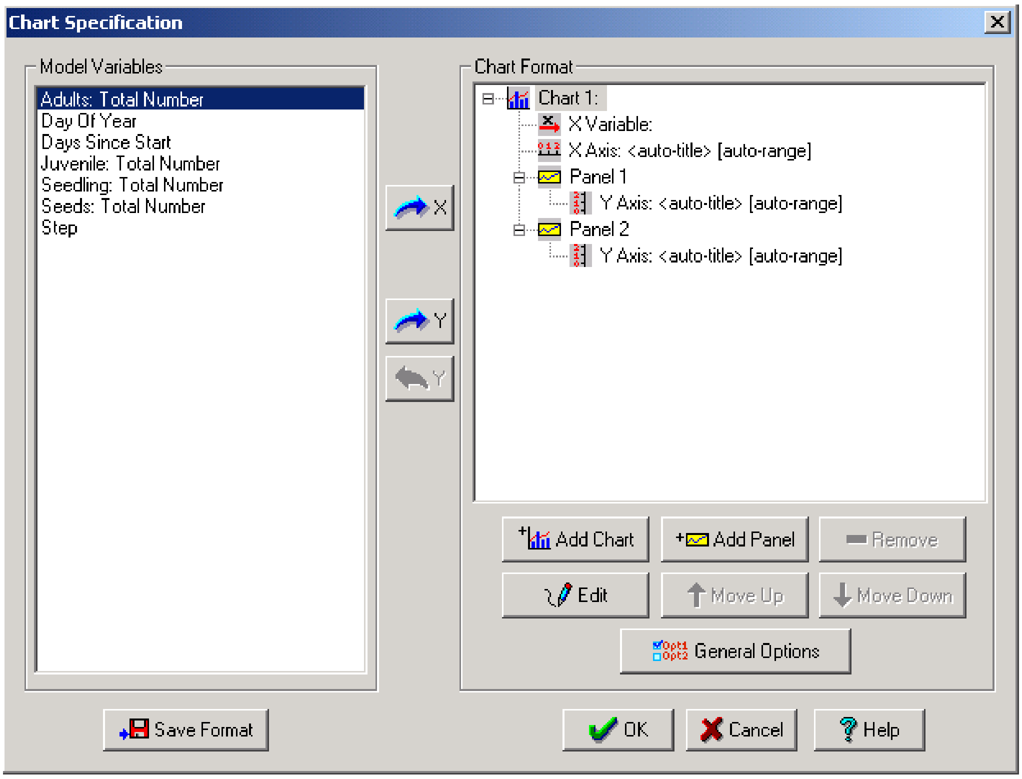


Figure 1.21 Chart Specification Panel

This window determines which variables will be used for chart formation, the format of the charts including their axis labels, whether the output values will be natural or logarithmic and provides a save option for frequently used formats. With five output variables to be displayed, single or combined chart outputs could be used.

Although for this initial display, a single chart with two panels on a common 'X'-axis will be used, the option of completely separate charts might be considered where different X-axis are required for separate panels. Insertion of user-defined axis labels should always be considered if more than one series is plotted in a panel, because by default DYMEX uses the name of the first series variable in the panel.

1. From the **'Model Variables'** panel area select *'Days Since Start'*, then select the  button to make it the X-variable
2. Select *'Adult: Total Number'*, and click the  button to make it the first Y-variable

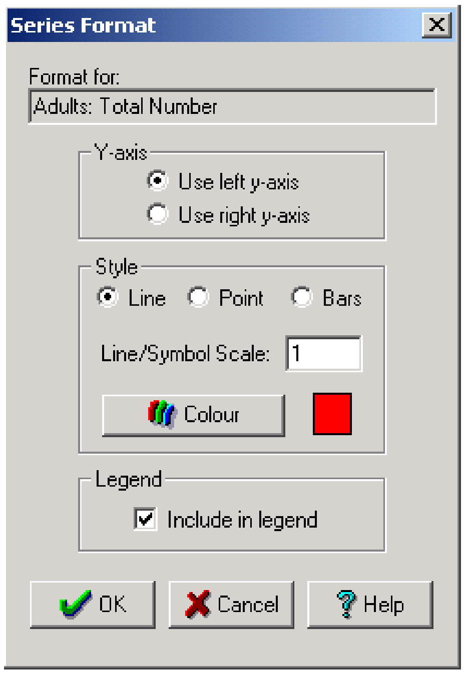


Figure 1.22 Graph Series Format Window

After selection of this button, the **'Adults: Total Number'** variable will appear in the **'Series Format'** window (Figure 1.22). This window enables the user to select whether the graph will use the left or right y-axis, whether it will be a point, line or bar display, and the colour (the default colour for the first variable in a panel is red). Click on **'Ok'** to accept the settings in the dialogue and return to the Chart Specification dialogue. A second 'reverse' transfer button () directly underneath the first transfer button will now be active and allow corrections if the wrong variable is selected.

By default, DYMEX displays two separate chart panels in the **'Chart Specification Panel'** window. This is done to draw the attention of new users to the possibility of drawing information on more than one panel. If the second panel is **not** required (i.e., no variables have been added to it) it will not be included in the chart. If we choose to use the second panel, then two separate chart panels will be drawn on the same X-axis. It is also possible to plot multiple variables within the same panel, sharing the same y-axis. For comparative purposes it is even possible to plot variables with different units, using the left Y-axis for one, and the right Y-axis for the other.

The Y-axis format can be edited to adjust the scale and labelling.

3. Select the **'Y-Axis'** icon under **'Panel 1'** and then click on the **'Edit'** button.

Examine the ***'Y-Axis Format'*** *window and the graph customisations that are available there.*

4. Click on **'Cancel'** to return to the Chart Format dialog

To add *'Juvenile: Total Number'* to the chart as a separate panel:

5. Ensure that *'Juvenile: Total Number'* is selected in the **'Model Variables'** list box
6. Select **'Panel 2'** in the **'Chart Format'** panel
7. Click on the  button to add *'Juvenile: Total Number'* to the second panel.
8. Set the line colour to blue in the **Series Format** dialog.
9. Select the **'Save Format'** button, save the display under a suitable name (e.g., 'Adults and Juveniles') and then return to the **Chart Specification** window
10. Select **'OK'** and the chart will now be displayed. It should be similar to Figure 1.23.

The chart very clearly shows how each chart can be used to display the output data. Frequently, as happens in Figure 1.23, totals are so large that only the final year's scores are displayed on natural number charts. Logarithmic charts are able to show the steady increase in population numbers much more clearly and the method of producing these is demonstrated in section 1.6.3.

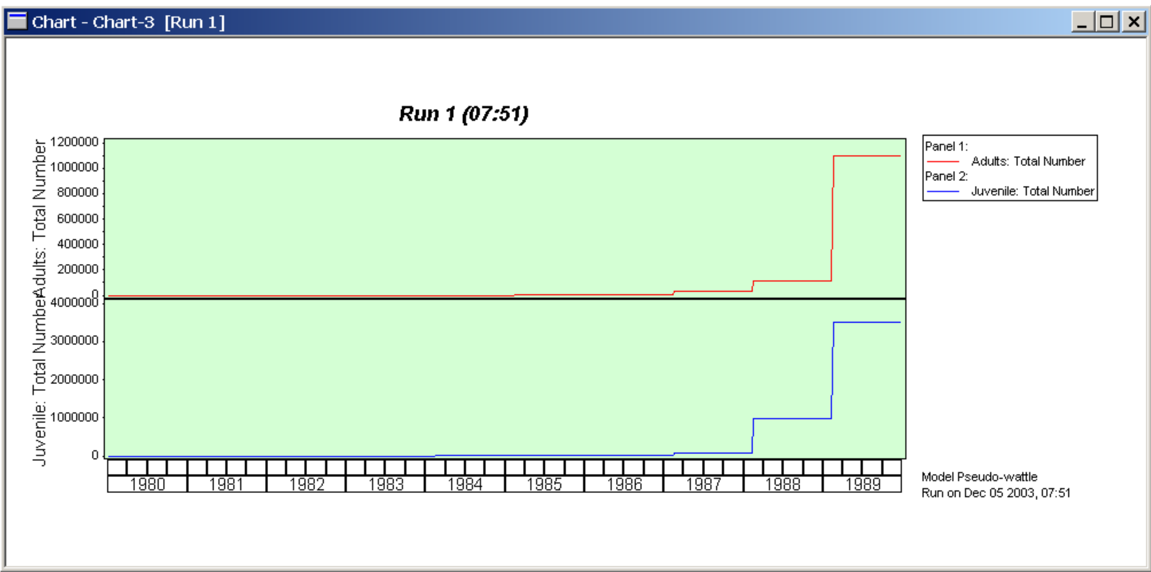


Figure 1.23 Pseudo-wattle Chart Output

To place both curves on the one panel, select the chart window. From the ‘Chart’ menu select the ‘Edit Format…’ item. In the ‘Chart Specifications’ window remove the second panel by selecting ‘Panel 2’, click on the ‘remove’ button and then confirm your intention. Select ‘Juvenile: Total Number’ and add it to ‘Panel 1’ using the  button, accepting the default display options. Each curve will appear as a different colour in the common panel.

To produce completely separate panels, select the chart window. From the ‘Chart’ menu select the ‘Edit Format…’ item. In the ‘Chart Specifications’ window remove ‘series 2: Juvenile Total Number’. Now add another chart using the ‘Add Chart’ button…then proceed as before with the selection of the panel variables using say ‘Day of Year’ as the X-axis variable.

Note that you can move the legends and annotations (the text at the lower right corner) to any position in the chart window by clicking on them and “dragging” them to the required position. The legends and annotations can also be removed using the **General Options** button in the **Chart Format** window and then un-checking the appropriate boxes in the resulting dialog.

1.6.3.1 Saving and Deleting Chart Formats

The above procedure demonstrated the method of saving a chart format that may be frequently required (step 9). To delete a saved format, the user must have a chart displayed in the current window. From the **‘Chart’** menu, select **‘Delete Format…’** and follows the steps required by DYMEX. To save the chart format of a currently-displayed chart, use the same procedure but choosing **‘Save Format…’**. Note that this menu option means that the user can save a chart format from the **‘Chart’** window as well as from the **‘Chart Specifications’** window.

1.6.3.2 Exporting Charts as Graphics Files

The **‘Chart’** menu bar option displayed while in chart display allows the user to export the displayed chart as one of several graphics formats. These format include Windows Bitmap (.bmp), JPEG (.jpg) and Windows Metafile (.emf). This is very useful for word-processing facilities and use in webpages.

1.6.4 Logarithmic Scaling

Without any controls, the current Pseudo-wattle model’s population increases exponentially. If the model is run for the 10 year period and the total number of plants is then charted using normal scaling, the resulting graph has little meaning as the numbers are so large that only the last years’ population numbers are discernable on the graph. Figure 1.24 displays the numbers of Adults after 10 years both normally and logarithmically. Users should always bear in mind that logarithmic displays may be more suitable in some instances.

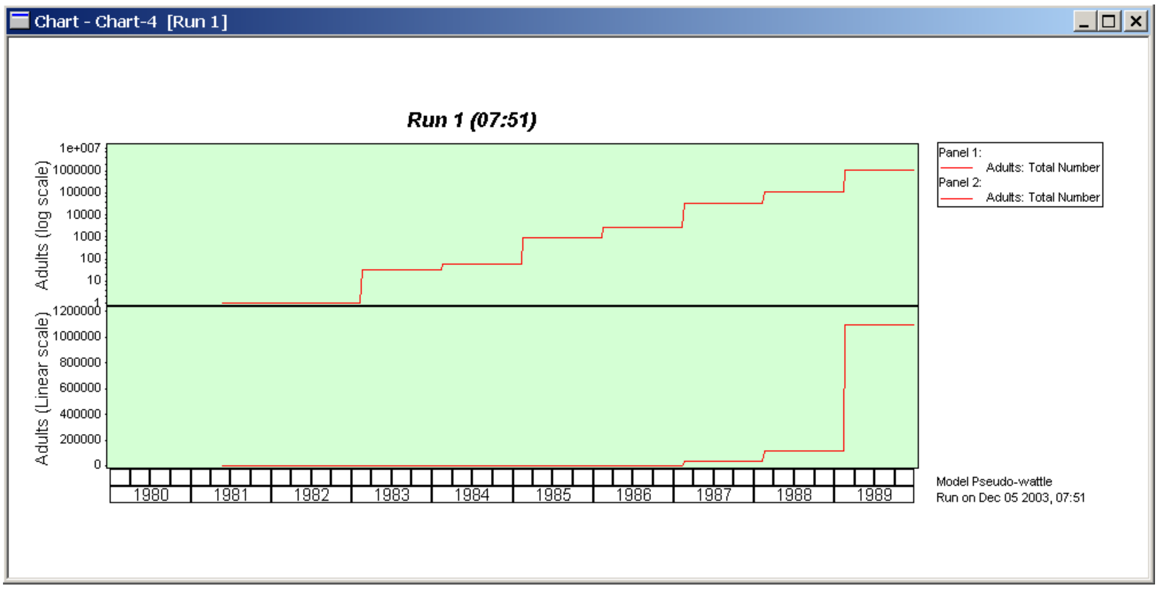


Figure 1.24 Yearly Populations of Pseudo-wattle Adultss for a 10 year period; logarithmic and normal scaling.

All the Pseudo-wattle charts show an uncontrolled increase in the population, especially if the model is run over several years. The model currently assumes that all seeds produced are viable, and pass automatically to the adult stage. In the field this is not the case; temperature and rainfall both play a part in establishing germination rates. Ways of introducing these aspects will be the subject of the next tutorial.

PW01.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

2 Seed Dormancy and Mortality

2.1 Introduction

The current Pseudo-wattle model links Adult Plant mortality solely to chronological age: this is artificial, but is adequate for the initial stages of model construction. Mortality in DYMEX may be either 'continuous', 'establishment' or 'exit', and each can be built into a lifestage and linked to any suitable user-selected variable(s). The effects of 'continuous mortality' reduce an organism's population over an interval of time, and they are caused by any of the environmental or genetic variables that influence the organism. Establishment and 'exit' mortalities are restricted to the situations where an organism is trying to pass from one lifestage to another and occurs through special factors which apply to that process. For example, the failure of a germinating seed to establish as a seedling (i.e., 'try, then die mortality') could be modelled using either an exit mortality from the seed lifestage, or an entry mortality for the seedling lifestage. The distinguishing factor between these mortalities and continuous mortality is that the mortalities are only applied once during the stage-transition. The choice of whether to choose an entry or exit mortality could be influenced by branching in complicated lifecycles. Establishment and exit mortalities will not be used for Pseudo-wattle in this tutorial.

During a perennial's reproduction, seed is destroyed before it can leave the parent plant. Insects attack the seeds during their formation and much lower quantities than those originally set actually mature. When mature, the dispersed seeds form a 'seed bank' in the environment from which losses then occur through the operation of various agents. Germination, ants or other insects, small mammals and birds consume or remove quantities of seed steadily over time, while rotting due to fungal or bacterial pathogens will reduce the remaining seed bank still further. Depending upon the rate of destruction, as much as 90% of all seed produced will be lost each year. Eventually, seeds become discoloured or buried (and therefore difficult to find), extremely hard or unpalatable, etc. and so seed that has survived for some time has a better chance of surviving until diminished seed viability ('old age') finally removes it from the 'seed bank'. Despite the seed losses described above, the seed bank numbers are usually more or less constant within seasonal fluctuations and seed bank losses are generally well tolerated by the species as otherwise it becomes locally extinct.

The continuous seed mortality described above suggests that a suitable model would be one in which the seed death rate is constant. This produces a seed survival function with respect to time that is exponential in form, and is easily modelled in DYMEX using a constant function within seed mortality. The exponential curve thus produced fits very well with the concept that although the seed bank's numbers decrease over time it never quite reaches zero, and so there is always a small residue of seeds to re-start the population.

2.2 Alterations to the Model

The Pseudo-wattle model will be changed as follows:

The constant mortality rate will assume that from a batch of 5000 seeds produced, there will still be 500 viable seeds remaining in 365 days (i.e., 90% of all seeds produced will be incapable of reproduction by the end of one year). The fecundity and reproduction functions of the model will also be changed to more accurately reflect the likely reproductive rates of a perennial in the field.

Annual reproduction per plant will become 1000 seeds and fecundity will be changed to 10000 seeds to accommodate the higher seed production.

2.3 The Continuous Mortality Model

The constant mortality rate is applied during each time-step so that the number of seeds in the current step differs from that of the previous step by the value of the mortality rate. For example, suppose the model starts with 100 seeds and the mortality rate is 0.1. After step 1, there will be 90 seeds remaining; after step 2, there will be 81 seeds left; after step 3, there will be 72.9 seeds left; and so on. Of course, fractional seeds are inapplicable to individual plants, however DYMEX is operating with average populations and these mathematical fractions are therefore valid.

The Pseudo-wattle model's mortality function assumes that after 52 weeks, 500 seeds are still viable from each batch of 5000 seeds produced. The constant (specifying the mortality within one 7-day timestep) required to obtain these results can be calculated once it is realised that the process is equivalent to an exponential decay curve for the numbers of viable seeds and that it has an equation of the form:

$$\mathbf{y = Ae^{-kT}}$$

For this equation, 'y' is the number of seeds surviving after one year; 'A' is the starting number of seeds; 'T' is the time over which the function is to operate (in this case 365 days, i.e., 52 7-day time steps); and 'k' is the decay constant - for this model, it is the mortality constant that will be applied to each day's seed survivors. If these values are substituted into the equation, we have:

$$\mathbf{500 = 5000\, e^{-k52}}$$

Dividing both sides by 5000 and then taking logarithms to both sides produces the result:

$$\mathbf{\ln 0.1 = -52k}$$

Which in turn produces the equation:

$$\mathbf{-52k = -2.303}$$

Therefore:

$$\mathbf{k = 0.0443}$$

2.4 Setting up the Model

1. Open the Model Builder and load the Pseudo-wattle file
2. Double click on **'Lifecycle'** to obtain the **'Lifecycle'** window
3. Select the **'Seed'** stage **'Mortality'** button to open the **'Seed Mortality'** selection box
4. Select the **'Continuous'** button to obtain the **'Mortality (Seed)'** dialogue box
5. Name the process '***Seed Mortality'***
6. Select the **'Parameter'** button from the **'Add component as'** panel to obtain

the **'Set Parameter Properties'** dialogue box

7. Change the name from **'Threshold'** to something more suitable (e.g., 'Seed Mortality rate')
8. Set the lower limit to 0, the upper to 1 and the default to 0.0443
9. Select **'OK'** as necessary to return to the **'Lifecycle'** window.
10. Select the **'Adult Plant'** lifestage
11. Select the **'Reproduction'** button
12. Select the **'Fecundity'** button and then change the parameter default to **'10000'** - set the upper and lower limits to **'5000'** and **'20000'**
13. Select **'OK'** as necessary to return to the **'Adult Plant Reproduction'** panel and then select **'Progeny Production'**
14. Select **'Edit Component'** and then select **'p2: Seeds per plant per season'**
15. Change the **'Seeds per plant per season'** default and upper and lower limits to **'1000'** and then select **'OK'** as necessary to return to the **'Lifecycle'** window
16. Save the model.

2.5 Running the Model

The model is loaded into DYMEX's Simulator exactly as previously described. Check that the initial population of seeds is **1**, and change the length of the model run to **1085** days.

A two panel chart output of Seed and Seedling populations will produce the results of Figure 2.1. This chart clearly displays the seed mortality rate producing the negative slope of seed numbers over time. (The first seed does not appear in the seed panel as it is too small to register on the graph.) It is worth examining the tabular outputs from this tutorial to see exactly how the seed mortality is affected by the function.

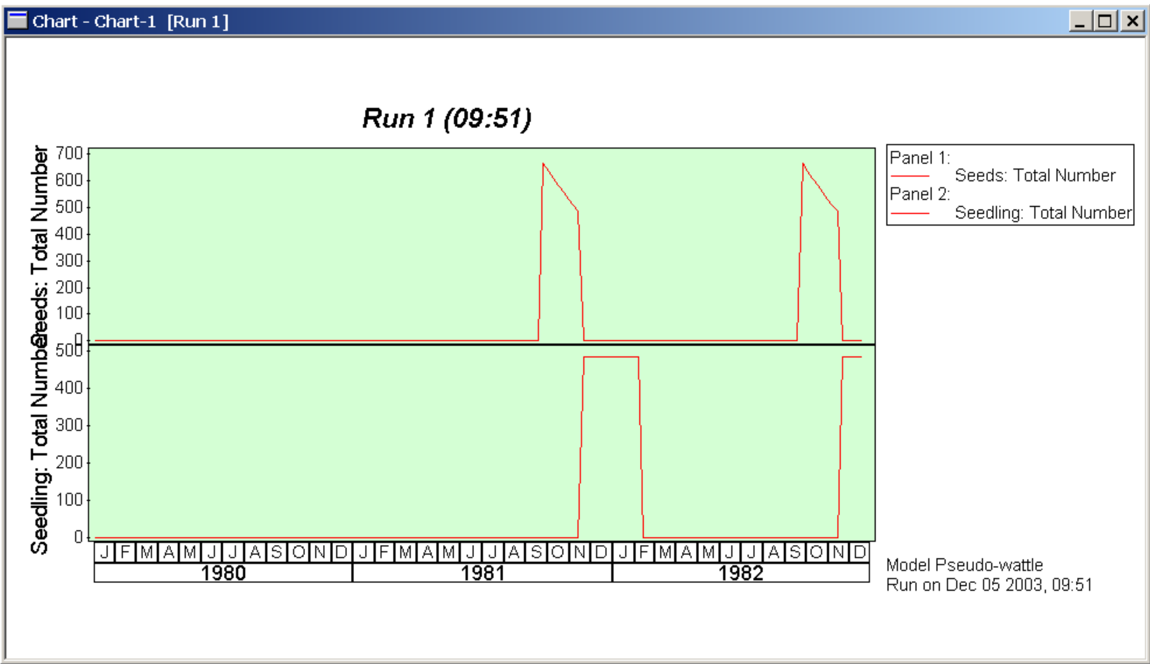


Figure 2.1 Seed and Seedling Populations for 1085 days

Users will appreciate that there will also be seedling mortality and this can be introduced into the model. Lastly, the current model forces all remaining seeds to break their dormancy simultaneously dependent only upon chronological age. In the field, there will be many factors which induce germination after the required dormancy period: suitable soil moisture, the presence of a fire for those seeds that require heat germination, or heavy rain to remove dormancy inducing 'hormones' from the seed coat itself. Some of these factors will be addressed in later tutorials.

PW02.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

3 Rainfall-Induced Germination

3.1 Introducing Rainfall

In the field, the germination of a perennial is largely triggered by soil moisture, provided the dormancy period is complete and the soil temperature is sufficiently high. Whilst this is a simplistic view, it is sufficient to permit the addition of the next stage of the model: rainfall controlled germination. In the present model, Pseudo-wattle seeds suffer continuous mortality but the 'seed bank' residue automatically germinates to become seedlings once the dormancy period of 8 weeks is complete. While it is essential that the dormancy period remains one of the germination conditions for Pseudo-wattle, the next step is to add rainfall germination conditions for the plant.

Ideally, soil moisture should be used as the germination trigger, not rainfall. Adequately modelling germination due to the presence of moisture requires more than linking germination to the quantity of rainfall in any one storm: other aspects such as humidity, soil moisture already present, soil water holding ability, evaporation rates, etc. must be considered. Soil moisture is, however, a derivative of rainfall and considerable changes have to be made to the model in order to introduce even rainfall-induced germination. To keep the increments to the model in each tutorial relatively small, the model will be changed so that a rainfall of 25 mm in any one step (week) will be sufficient to trigger germination of all mature seeds. More complex models would at least consider whether or not sufficient soil moisture remained in order for the seedlings to continue their growth and this in turn might interact with additional lifestages. The change to soil-moisture-regulated germination will be made in a later tutorial.

Once an input of rainfall is required, alterations to the model are needed. First, there must be a way of reading meteorological data into DYMEX so that weekly rainfall totals can be used to trigger germination. Next, the stage transfer must be altered so that it combines both the seed maturing time of 8 weeks and the rainfall requirement: this is done by using a combination function within DYMEX. Finally, the 'Timer' module must be altered so that it provides a start date that will allow it to operate with the real time dates in the meteorological data file.

3.2 Altering the Model

The model will assume that Pseudo-wattle seeds require a weekly total rainfall of 25mm in order to germinate.

Since the meteorological data we will use specifies actual dates, the 'Timer' module must be altered to produce an actual date output rather than a simple step

1. Start the DYMEX Model Builder program
2. Open the Pseudo-wattle file to obtain the **'Model'** window
3. Open the **'Timer'** module by double clicking the text
4. Select the **'Outputs'** button
5. From the **'Module Output Variables'** scroll list, ensure that the first three output variables are selected - if not, then complete step 6 - if the first three output variables are selected, go to step 7
6. Highlight any unselected output variables then click on the **'Select'**

button (the **'+>'** symbol will appear)

7. Select **'OK'** as necessary to return to the **'Model'** window.

The next procedure is to add a module to read the meteorological data for use in the model.

8. From the menu bar select **'Model'**
9. From the drop-down menu select **'Add Module'**
10. Making sure that '**Standard**' is selected, find **'*MetBase*'** in the list of modules and select it by clicking on it, then click **'OK'**
11. Re-name the **'MetBase'** module (e.g., 'Meteorological Data')

*(Note that in the **'Module Details'** panel of the window, each button has a description of its functions.)*

12. Select **'Inputs'** to produce the **'Inputs [Meteorological Database]**' link window (figure 3.1)

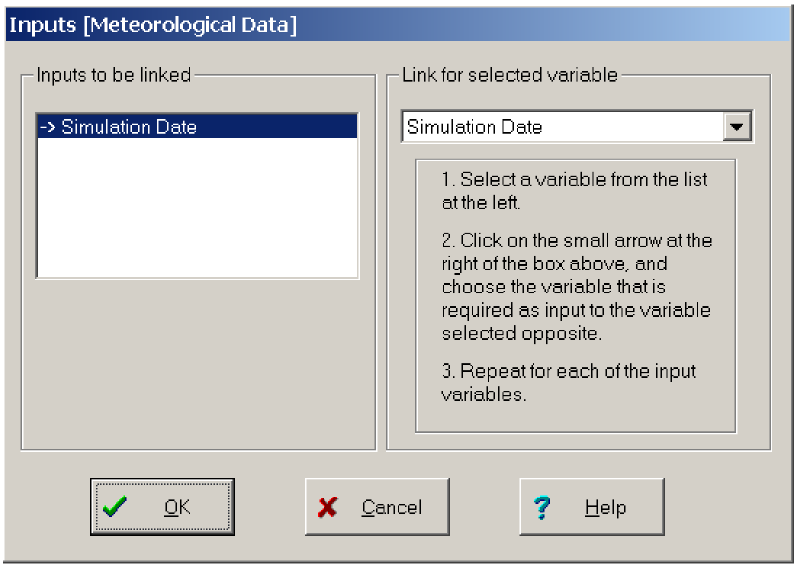


Figure 3.1 'Inputs [Meteorological Data]' Link Window

The 'Link for selected variable' panel will display '(none)'.

13. Using the **'Link for selected variable'** scroll box, select **'Simulation Date'**

Both boxes will now display 'Simulation Date' in highlighted form.

14. Select **'OK'** to return to the **'Meteorological Data'** module window
15. Select **'Outputs'** to open the **'Outputs [Meteorological Data]'** selection window (figure 3.2)
16. With **'Rainfall'** highlighted, click on the **'Select'** button

Since rainfall is to be read from a file, the user may feel that setting the minimum and maximum values is irrelevant however such settings can be useful as a check on incorrect or unusual values of data: if a value being read into the model falls outside the set range it will be reported as a possible error for user correction if necessary.

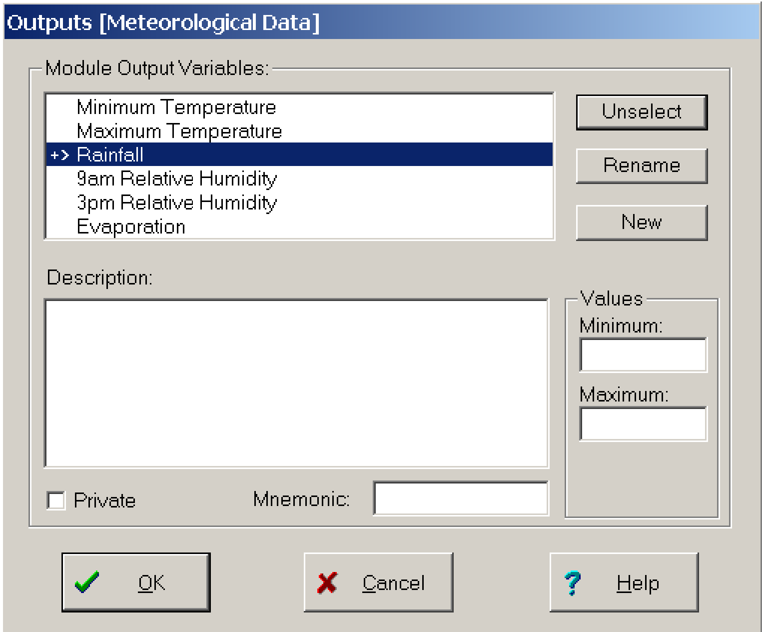


Figure 3.2 Outputs (Meteorological Data) Variables List Box

17. Set **'Minimum allowed value'** to 0
18. Set **'Maximum allowed value'** to 300
19. Enter a description (e.g., '*Total Weekly Rainfall - mm*')
20. Enter a short name (mnemonic) such as '*Rain*'.
21. Exit to the **'Model'** window.

The model window will now resemble figure 3.3 below, with three modules in the model window: Timer, Lifecycle and Meteorological Data.

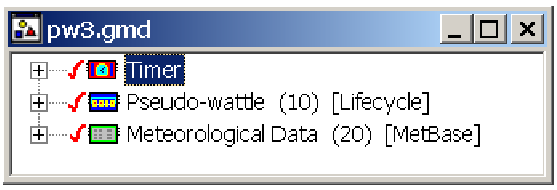


Figure 3.3 Partially completed Model window

The seed lifestage will now be altered to accept new information about the transfer from seed to seedling and to set the transfer function so that it is driven by the Meteorological Data module's *"Rainfall"* output. Since we assume that all Pseudo-wattle seeds germinate once the weekly rainfall totals 25mm, a step function is required.

Because rainfall is now an extra condition under which seeds germinate and move to the seedling lifestage, its effects must be combined with those of the 8-week seed maturation time. Multiplying the two functions achieves this and the result can be easily understood if it is remembered that the output of the seed maturation function is either zero or one. If the seed is not mature, the output will be zero and even if the weekly rainfall reaches 25mm within that 8-week period, the multiplication of zero will produce no germination. After the 8-week period, the output from the seed maturation function will be one and so germination and stage transfer is then solely determined by the rainfall function.

1. Open the **'Lifecycle'** window
2. Select the **'Stage Transfer'** button of the **'Seed'** lifestage to open the **'Seed - Transfer'** (Germination) dialogue box
3. Select the **'Function'** button to open the **'Function'** dialogue box in order to add a new transfer function
4. Rename the function **'Rainfall Germination Trigger'** and then select **'OK'**
5. Using the function scroll box select **'Step'**
6. Select **'Rainfall'** as the independent variable
7. Select the **'p1: Threshold'** parameter and click **'Edit Parameter'** button to open the **'Set Parameter Properties'** dialogue
8. Rename the parameter to (say) '*Germination Threshold'*
9. Set the lower limit to 0, the upper to 50 and the default to 25
10. Return to the **'Function'** dialogue
11. Select the **'p2: Step Height'** parameter and click **'Edit Parameter'** button

Since all seeds germinate simultaneously once the weekly rainfall total reaches 25mm, the step height is given a value of one (1).

12. Set the default and limits all to 1
13. Suitably re-name the variable (e.g., '*Proportion of seed germinating'*)
14. Select **'OK'** as necessary to return to the **'Process'** dialogue box.

There will now be available a previously 'greyed-out' button, **'Set Combination Rule'**. This button is used to set the 'rule' that will be used to combine the two components (functions) that make up this process. As has already been noted, the functions will be multiplied to give the transfer rate.

1. Select the **'Set Combination Rule'** button and open its list box
2. Select **'Product; R= (a x b x...)'**
3. Select **'OK'** as necessary to return to the Lifecycle window
4. Save the model.

3.3 Changing the Sort Order of a Module

You will have noticed numbers in brackets after each module (see Figure 3.3). These refer to the order in which the DYMEX program completes its calculations when running a simulation with the model. Obviously, the information in the Meteorological Data module is required by the Lifecycle module before it can complete its operations so that although the Meteorological Data module was added last, it must be placed above the Lifecycle module in the sort order. By default, the Timer module is always first in the list and has a sort order of '0'. To move the Meteorological Database module, it is only necessary to change its sort order to a value between 0 and 10.

1. Open the **'Meteorological Data'** module by double clicking on its text
2. On the right hand side of the window, select the Sort Order panel and change the text value from **'20'** to **'5'**
3. Select **'OK'** as necessary to return to the **'Model'** window - the **'Meteorological Database'** module will now have been shifted to a position in between the **'Timer'** and the **'Lifecycle'** modules.
4. Save the model.

This completes the model building procedure in the for this part of the tutorial.

3.4 Initialising a Model with Meteorological Data

3.4.1 Loading the Model

1. Open the DYMEX Simulator using your previously saved Simulation File.

Once the file is loaded into the DYMEX Simulator, a 'Model Components' window appears (figure 3.4). The Meteorological Database module is now shown in its new position above the Lifecycle module. Because there is no tick beside the 'Meteorological Database' module, it indicates that user initialisation of those module's variables is required before the Simulator will process the model. (The Timer module will also require re-initialisation to deal with the new start date of the metbase file.) The colour of the ticks indicates whether user initialisation may occur: a blue tick indicates that user alteration of the module initialisation settings may occur, a grey tick indicates that no user alteration is possible. The summary variable manager appears at the end of the list. As no summary variables have yet been defined, the module appears with a grey coloured tick.

Figure 3.4 also displays information about the current settings for the model: its run duration is presently set to 1085 days (from Tutorial 2) and the Lifecycle module is initialised with 1 seed.

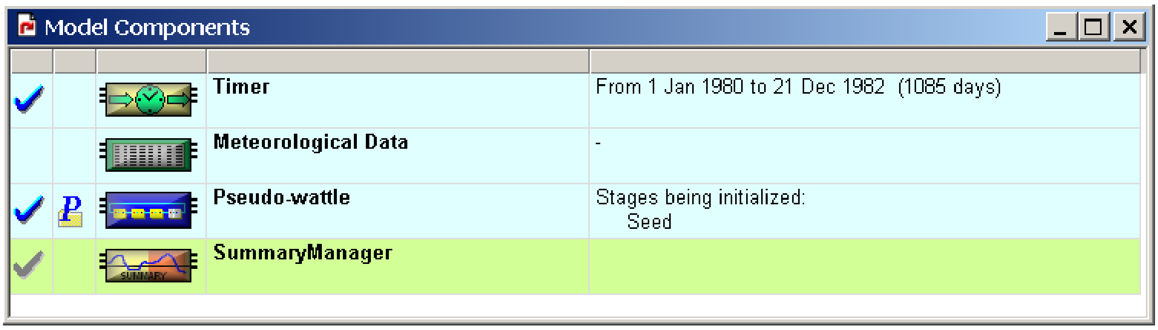


Figure 3.4 Model Components window

While a simulation is running, DYMEX extracts meteorological data from a previously constructed meteorological data file and uses it in the model. The Simulator can read meteorological information from almost any file format, but two provisos must be made:

1. **Apart from a header, each line of the file must have exactly the same format; and**
2. **The user must know the format of the file.**

A typical meteorological file contains information on temperature, rainfall, humidity, air pressure, evaporation, etc. and may be daily, weekly, monthly etc. The exact format depends upon the circumstances under which the file was built, and DYMEX must be told the precise format in order to find the necessary information.

3.4.2 The Amberley Meteorological File

The Amberley meteorological file supplied with DYMEX (Table 3.1) will be used in the Pseudo-wattle model. Only the first few lines are shown. Columns are counted from the left and blank columns must be included in the count. For the lines of data, the date is contained in columns 1-6 (columns 1 and 2 contain the day, columns 2-3 the month and columns 4-6 the year. The first two lines do not contain meteorological information – the first line gives location information while the second line labels some data columns.

```
A     Amberley Airport   Lat: -27.6  Long: 152.7  Elev:   25
       TMIN TMAX RAIN          RH9           RH3  DLNG EVAP
 1 165 18.0 30.0  0.  5.0 16.0 73. 17.3 17.0 46. 13.85  5.6 AMB
 2 165 17.0 32.0  0.  2.9 18.0 84. 21.1 17.0 41. 13.85  5.5 AMB
 3 165 19.0 33.0  0.  1.8 21.0 91. 22.2 18.0 41. 13.84  5.6 AMB
 4 165 19.0 31.0  0.  5.3 17.0 73. 19.2 17.0 43. 13.83  6.1 AMB
 5 165 19.0 31.0  0.  5.3 17.0 73. 19.2 17.0 43. 13.82  6.1 AMB
 6 165 19.0 32.0  0.  4.6 18.0 77. 21.1 17.0 41. 13.81  6.3 AMB
 7 165 21.0 29.0  0.  6.5 17.0 69. 16.4 16.0 45. 13.80  6.4 AMB
 8 165 17.0 29.0  0.  7.7 11.0 56. 20.8 10.0 31. 13.79  7.4 AMB
 9 165 15.0 29.0  0.  4.7 13.0 70. 20.8 10.0 31. 13.78  6.4 AMB
10 165 16.0 35.0  1.  5.6 15.0 70. 31.6 12.0 25. 13.77  8.1 AMB
11 165 21.0 39.0 11. 10.8 16.0 56. 44.9  7.0 14. 13.75 12.7 AMB
```

Table 3-1 Amberley Meteorological File

Structures in the file on which DYMEX will require information for this tutorial are:

- Lines 1 and 2 of the file is an information 'header'.
- The first 6 columns of the file are date information with format 'ddmmyy'. (Notice that some of these columns are blank at first but will be filled when either double digit days or months are reached.)
- Columns 17-22 contain the daily rainfalls.

The remainder of the file can be ignored for this tutorial. Of course, it would have been possible to have *only* rainfall data in the meteorological file; however this would restrict the use of the file and the program to the effects of rainfall only. Sooner or later, other meteorological variables will become necessary in the model and access to a complete file is preferable (e.g., columns 8-16 contain daily temperatures.).

3.4.3 Initialising the Meteorological Database Module

The first set of procedures is to open the required meteorological database file and set DYMEX so that it can read the necessary data from the file.

1. Select the **'Meteorological Database'** button in the **'Model Components'** window followed by **'Initialise Module'** from the drop-down menu

This opens the **'Data Files'** dialogue box which allows the user to find, open and format the meteorological file, 'Amberley.dat', required for this run. There are two common ways of data delimitation in data files: space-delimitation with set column widths or comma-separated variables where each datum is separated from the next by a comma. The Amberley file uses set column widths and so the default selection shown in the **Datafiles** selection panel can be left as it is.

2. Select **'Browse'** and scan the files/directories until **Amberley.dat'** is located (it is usually in a Dymex2 subdirectory called 'MetData'
3. Select **'Amberley.dat'** and then click on the **'Open'** button
4. Select the **'Format'** button to produce the **'File Format'** window (fig 3.5).

The 'Data File' window (Figure 3.5) allows the user to inform DYMEX of the format of the loaded meteorological data file so that the model can correctly interpret the data. For this tutorial the 'More Options' button may not required. If pressed, additional operations become available to the user. Once opened, the 'More Options' window remains open until the user closes the **File Format** dialogue.

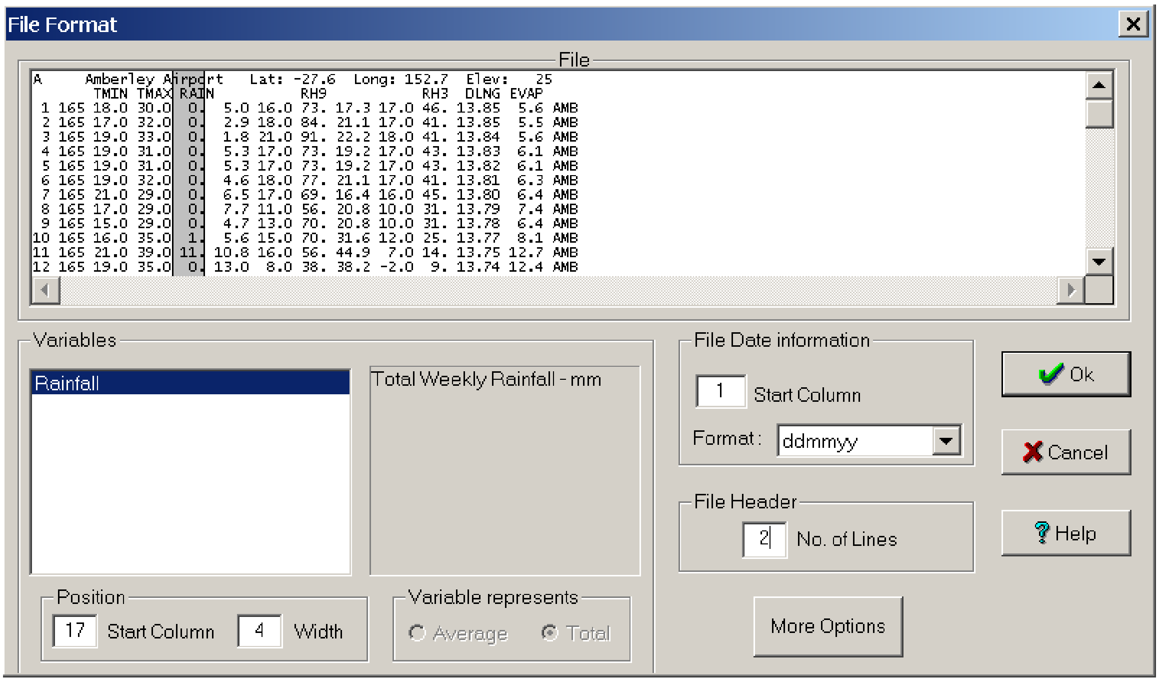


Figure 3.5 DataFile window with rainfall, file date format and file header settings shown.

3.4.3.1 Header Lines

The first step is to 'tell' DYMEX how many lines at the top of the file contain information other than meteorological data. Once this is set, DYMEX will ignore these lines while reading data. Inspection of the file data list box shows that two lines contains such material.

15. Select **'No. of Lines'** edit box
16. Type in the value 2.

3.4.3.2 Date Information

The second step is to 'tell' DYMEX the date format of the file. The Simulator allows a large number of formats: the user selects the one corresponding to the specified file. Since the file dates start in column 1, this is the first information to provide for the Simulator. Use the box area marked 'File Date Information'.

1. Select **'Start Column'** edit box
2. Ensure the value is set to 1
3. Select the scroll button on the **'Format'** list box
4. Scroll until the correct format of 'ddmmyy' is obtained and then highlight it to select the format.

3.4.3.3 Rainfall Information

The third step is to define the area of the file in which DYMEX will look for rainfall information. DYMEX's Simulator provides a very simple method of area definition in which the user does not even have to count where the columns begin and end. The method used is the standard 'Windows' mouse procedure of marking an area of text: place the cursor at the start of the desired area, hold the left hand button down, drag (slide) the mouse until the desired area is highlighted and then release the button. (Once this method is known, it can be used to 'track down' the column number of any column if the user is unsure of a count.)

1. With **'Rainfall'** highlighted in the **'Variables'** list box, place the cursor between the "A" and the "i" in "Airport"
2. With the left hand mouse button held down, slide the mouse until the column shown by shading in Figure 3.5 is highlighted
3. Release the mouse button - the selected area of the file will remain highlighted and will extend to the full height of each selected column
4. Inspect the 'Position' area of the window and the 'Start Column' edit box should now show 17 while the 'Width' box should now show 4; if it isn't repeat the procedure until it is correct

Note that the **Variable represents** panel near the centre bottom of the dialog shows "***Total***" selected, with no other choice available. This reflects the fact that we are using the MetBase module to read the data, which "knows" that its rainfall output represents totals over the simulation period. Even though the "Amberley.dat" file contains daily values of rainfall and our model has a weekly timestep, the Meteorological Data will correctly convert the daily rainfalls to weekly totals.

5. Select **'OK'** as necessary to return to the main window

The **'Meteorological Data'** icon will now have a tick beside it. To complete the initialisation procedures:

1. Select the **'Timer'** module followed by **'Initialise Module'**
2. Set the run default to **1085** days
3. Set the Simulation starts date to **1/1/1965**
4. Select **'OK'**.

This completes initialisation of the model for the Simulator and it can now be run.

3.4.4 Initialising the Lifecycle Module

Initialise the lifecycle module with 1 seed.

3.5 Running the Model

With the model now initialised, all that is required is to select 'Run'. The initial numbers of Pseudo-wattle should be 1 seed, but it is worth checking just to make sure. Run the model for a period of 1085 days and then produce a chart output with four panels containing rainfall (use bars instead of lines), total numbers of seeds, seedlings and juvenile plants. The result should resemble figure 3.6

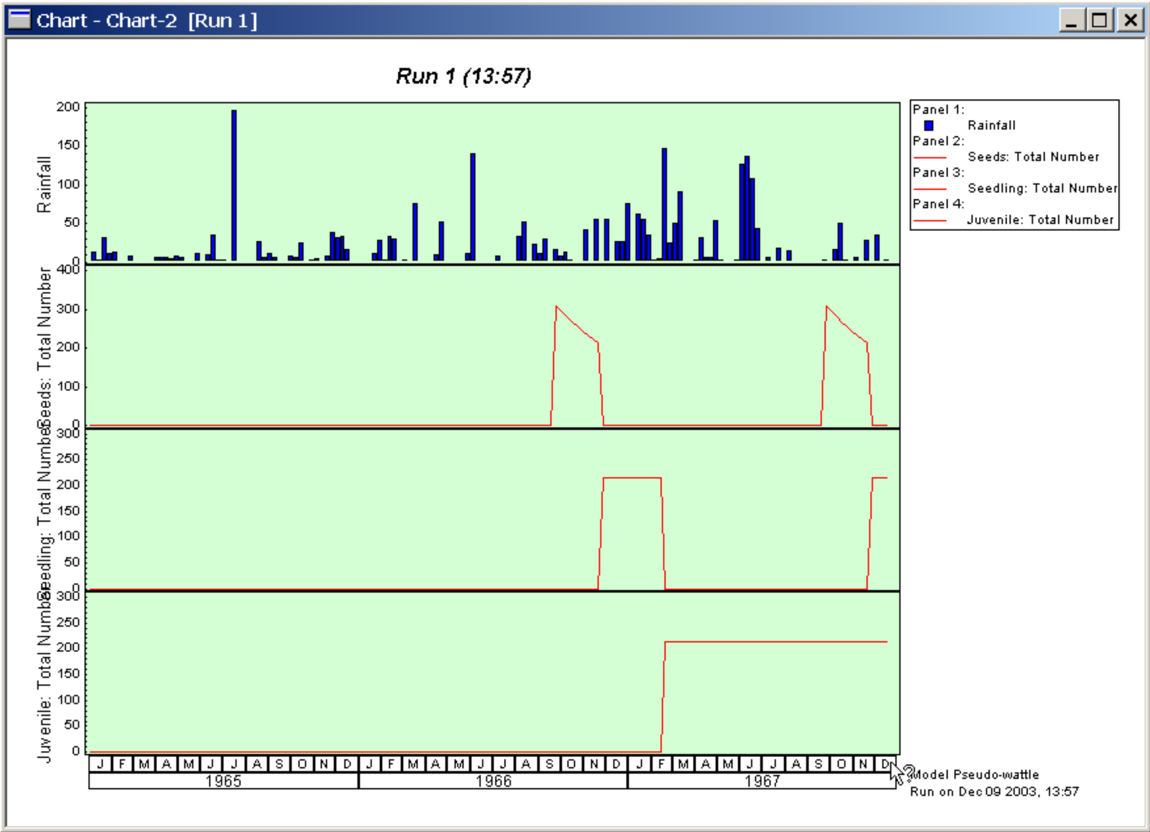


Figure 3.6 Pseudo-wattle with rainfall induced germination

It may be difficult to see the effects of rainfall induced germination clearly from the graphs, however the effects are very clearly seen if the model is run for the full 10 years and a tabular output of seeds, seedlings, step and simulation date is produced. If only the dormancy threshold operates, the seeds transfer to seedlings at the ninth step after dispersion, but if the rainfall threshold is added, the germination transfer to seedlings takes place anywhere from 8-13 weeks after dispersion and is tied to the appearance of the necessary weekly rainfall total.

PW03.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

4 Soil Moisture

4.1 Introduction

Tutorial 3 implied that rainfall by itself is generally too erratic for it to be the sole determinant of moisture levels in a model. Soil saturation deficiency may be used as an alternative but this also produces problems. Calculation of the amount of water held in the soil at any given time has proven to be the best method of introducing moisture levels into plant models, however when this technique is used, additional modules are needed.

Rainfall is usually the main source of water for soil moisture. Water losses are determined by evaporation (which is affected by relative humidity, season, hours of sunlight, intensity of sunlight, latitude, plant cover, etc.) and soil structure (ability to retain water, current soil moisture levels, etc.). These aspects require specialised DYMEX modules to simulate their effects.

4.2 Modelling Soil Moisture

4.2.1 Modules and Alterations Required

Four additional modules are required to model soil moisture: a soil moisture module to provide information on soil moisture levels to the lifecycle, an evaporation module to determine water loss from the soil, a daylength module to influence the rates of evaporation and a QueryUser module to allow the setting of the latitude. The Meteorological Data module must be edited so that it can provide the necessary inputs to all the new modules. Finally, the lifecycle module's seed lifestage must be altered to reflect the germination dependence upon soil moisture.

4.2.2 Soil Moisture Module

A 'Soil Moisture' module is required to provide the values for soil moisture to the 'Lifecycle' module. The module provides soil moisture values in the range 0-1 [permanent wilting point (fairly dry soil) to field capacity (completely saturated)] and it requires both internal settings and inputs from other modules.

The parameters required for the soil moisture module are: 'Soil Moisture Capacity', Evapotranspiration Coefficient' and 'Drainage Rate'. The Soil Moisture Capacity records the maximum water storage capacity of the soil *within the effective rooting zone of the plants being modelled*, and is normally between 50 and 200 mm. In practice, a setting of about 100-150 mm is typical of many soils. Sandy soils would have a low potential soil moisture store while clay soils would be quite high. The Evapotranspiration Coefficient sets the transpiration loss from plants and their land surface compared with a class A pan evaporation pond; a constant value of 0.8 is often adequate for coarse models. The rate of evapotranspiration however, tends to decrease as the water content of the soil drops toward the permanent wilting point. A simple function that works well for moderate-to-deep-rooted plants is a two-segment linear function (Johns & Smith 1975). The Drainage Rate sets the proportion of soil water content that is lost through drainage per time step. Normally, this variable is set to zero, which implies no loss of water through drainage.

The two main inputs to the 'Soil Moisture' module are values for 'Evaporation' and 'Rainfall'. Rainfall is obtained from the meteorological data file through the procedures already described for temperature but evaporation requires an additional module to be added.

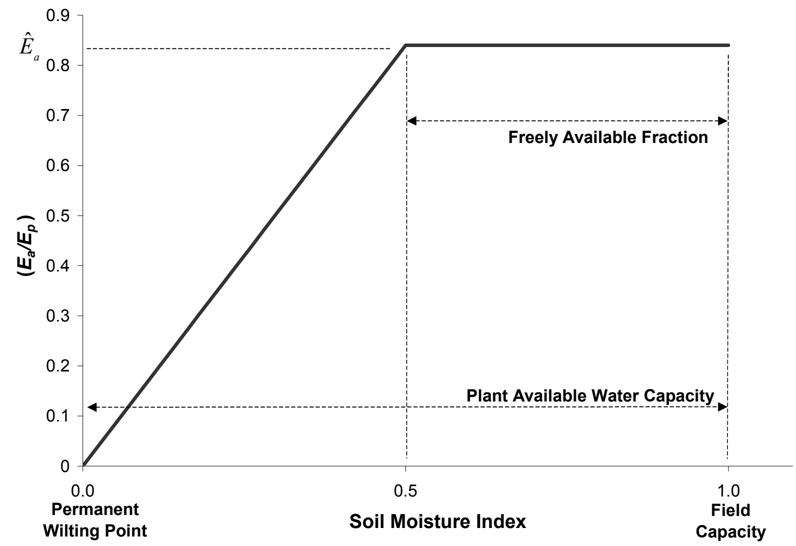


Figure 4.1 Evapotranspiration Coefficient as a function of Soil Moisture

4.2.3 Evaporation, Daylength and QueryUser Modules

The 'Evaporation' module used in DYMEX requires five climatic inputs: minimum temperature, maximum temperature, 9am relative humidity, 3pm relative humidity and daylength. The first four of these are obtained from the meteorological data file and only require that the relevant module be altered so that it is able to read the appropriate information from the file and provide it to the model. The daylength is supplied by a new module, the 'Daylength' module. This module requires two inputs: the day of the year (which is obtained from the 'Timer' module) and the latitude which is set by another new module: the 'QueryUser' module. The best way to envisage the process is to examine a schematic diagram (Figure 4.2).

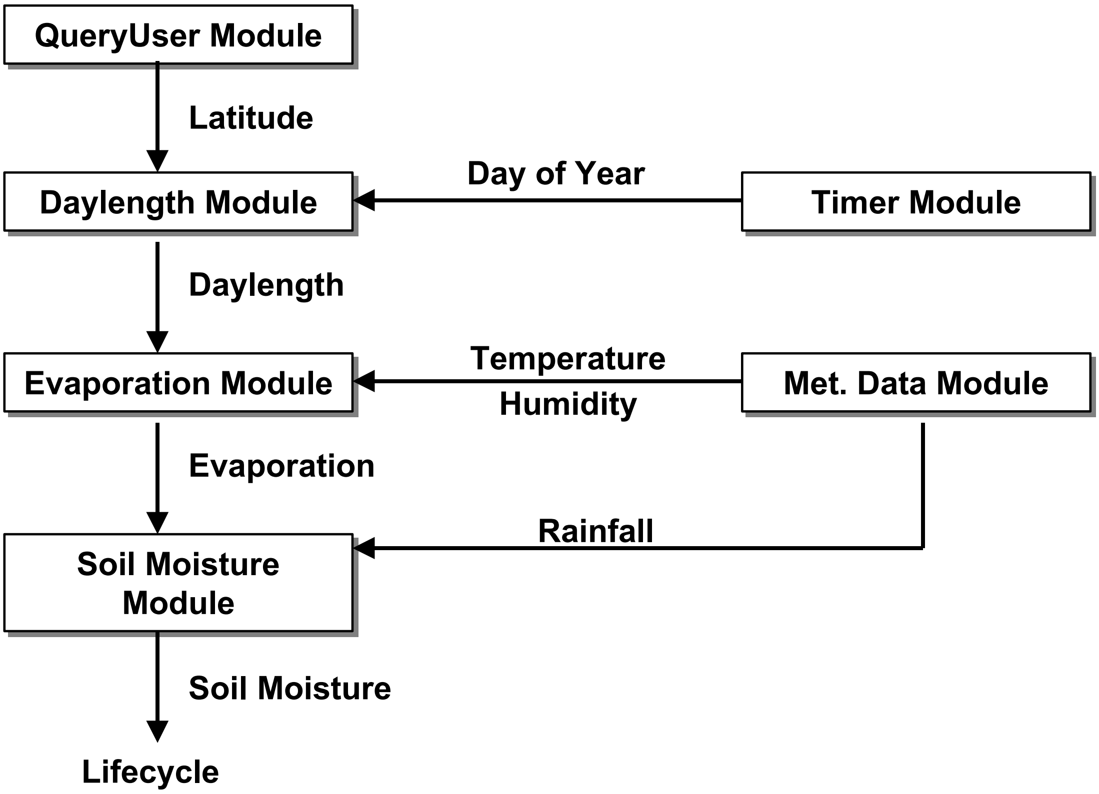


Figure 4.2 Modules for Soil Moisture

4.2.4 Changing the Seed Lifestage in the Lifecycle Module

Pseudo-wattle seeds currently germinate if two requirements are satisfied: the dormancy period of eight weeks must be complete and there must then be at least 25mm of rainfall on any one day. In the present model, germination of all seeds in the seed bank takes place, but this is usually not necessarily the case in the field. Germination rates often increase with soil moisture increase, and there may be a limiting value of soil moisture beyond which the soil contains so much water that germination becomes progressively inhibited and may eventually cease.
The following information relates to the soil moisture induced germination potential of pseudo-wattle:

> At least some pseudo-wattle seeds in the seed bank commence germination once soil moisture levels have reached 0.3. As soil moisture increases, the number of seeds germinating also increases until the value of soil moisture reaches 0.55. At this point, 80% of the seeds in the seed bank will germinate. This gives a slope of 3.2. Above 0.55, the rate of germination steadily decreases at more or less the same rate to give a slope of -3.2.

4.3 Building the Model

(Note: the user's knowledge of already covered DYMEX procedures is assumed. Individual keystrokes may be omitted for well-known procedures. Amberley's latitude is 27.6^{o}S.)

1. With the Model Builder open and the Pseudo-wattle file loaded:
   (a) select the **'Add Module'** procedure
   (b) select a **'QueryUser'** module
   (c) rename it ***'Latitude'***
   (d) select the **'Outputs'** button and obtain the **'Output (Latitude)'** dialogue box
   (e) select the **'New'** button and a variable named **'Latitude Variable 1'** will appear highlighted in the list box
   (f) click once on the **'Select'** button and **'+>'** will appear in front of the variable name - rename it **'Latitude'**
   (g) set the outputs to -27.6 for the default and -90 and 90 for the lower and upper limits respectively and click **Ok** to return to the **Module** dialogue.
   (h) set the sort order to '5'
   (i) select **'OK'** to return to the **'Model'** window.

(The latitude limit values of –90 to 90 allow the data from any world location to be used if desired.)

2. Select the **'Timer'** module for editing and ensure that all possible outputs except 'Time of Day' are selected (Days since Start, Day of Year, Simulation Date) and then return to the **'Model'** window
3. Add **'Daylength'** as a new module and name it ***Daylength***
4. With the **'Daylength'** module window open, select the **'Inputs'** button then select as inputs **'Latitude'** and **'Day of Year'** and link them to the variables of the same name (or type) from the right hand list box - then return to the editing window and set the sort order to '6'
5. Select the **'Outputs'** button, ensure that **'Daylength'** is selected **('+>')** and return to the model window
6. Select the **'Meteorological Data'** module for editing and ensure that **all** variables except evaporation {Temperature (max/min), Relative Humidity (9am/3pm) and Rainfall} are selected as outputs, then return to the 'Model' window
7. Add **'Evaporation'** as a new module and obtain its editing window
8. Rename the module to '***Evaporation***'
9. Select the '**Inputs**' button and link each of the inputs to the corresponding variable by clicking on the inputs in turn and then selecting the matching variable from the drop-down list at the right.
10. Return to the Module window, and go to 'Outputs", select the output variable and rename it to 'Pan Evaporation'.
11. Set the Sort Order so that the '**Evaporation**' module comes immediately after the Meteorological Database module in the model list.
12. Exit back to the 'Model' window and save the model

It is worth closing and re-opening the model at this stage to see the results of the sort order algorithm. Before the model is closed, some modules will have sort order values of 5, 6, etc, corresponding to the values specified in the procedure above. In the reopened model, all sort order values will now be multiples of '10' so that intermediate sort orders for new modules can again be added.

13. Create **'Soil Moisture (1-layer)'** as a new module, then edit as follows:
    a. Rename the module to ***Soil Moisture***
    b. Link the 'Rainfall' and 'Evaporation' inputs to 'Rainfall' and 'Pan Evaporation', respectively.
    c. Select for output only the 'Soil Moisture' variable
    d. Select the **'Factors'** button

There are three soil moisture factors. In this model the soil moisture capacity and drainage rate factors are constants. They are set by first selecting the factor name from the list box and then typing in the default and limiting values. The evapotranspiration rate is more complicated. Before proceeding to set the factors, it is worth re-considering the type of environment in which an annual would normally grow. Perennial plants are found under most Australian climatic conditions from the eastern coastline to the drier internal plains. These areas are usually not desert conditions although soils may range from sandy loams to heavy clay. Actual settings for the Soil Moisture (1-layer) module will depend on individual circumstances, however generalised settings can be used which can be modified for local conditions. Since pseudo-wattle is a semi-arid land species, the Initial Soil Moisture will be set at 0.05, the Soil Moisture Capacity will be set to 70 mm, the Evapotranspiration Coefficient will be a linear function with an intercept of 0, and a maximum value of 0.8 set using the advanced function button. The slope needs to be set so as to ensure that the line reaches the maximum value when the soil moisture level is 0.5. Using the formula for calculating a slope, $\Delta y/\Delta x$, $1.6=0.8\div0.5$. The Drainage Rate will remain at 0.

    e. Select **'Soil Moisture Capacity'** and then the **'Set Parameter'** button
    f. Set the default to 70 and the lower and upper limits to 50 and 200, respectively
    g. Select '**Evapotranspiration coefficient**' followed by the **'Set Function'** button
    h. Select **'Soil Moisture'** as the independent variable
    i. Set the function type to **'Linear'**, and set the **'Y-axis intercept'** to 0. This means that all evapotranspiration will cease when the plant available soil moisture store reaches permanent wilting point (soil moisture = 0).
    j. Set the default **'Slope'** to 1.6 and the lower and upper limits to 1.6 and 2. This will allow the slope to be altered to match any changes in the maximum value of the function in the range 0.8-1.
    k. Click on the **'Advanced'** button and set the **'High Limit'** value to 0.8.
    l. Return to the Factors dialogue and set all values of the **'Drainage Rate'** to 0
    m. Return to the module window and set its sort order so that the **'Soil Moisture'** module is directly after the **'Evaporation'** module
    n. Exit to the **'Model'** window (which will resemble figure 4.2) and then open the **Lifecycle** window.

There will now be 7 modules in the 'Model' window (figure 4.2). (The user is reminded that the 'plus' icons allow checks to be made of all module structures if they are selected. Where parameter values have been set, these are also displayed.)

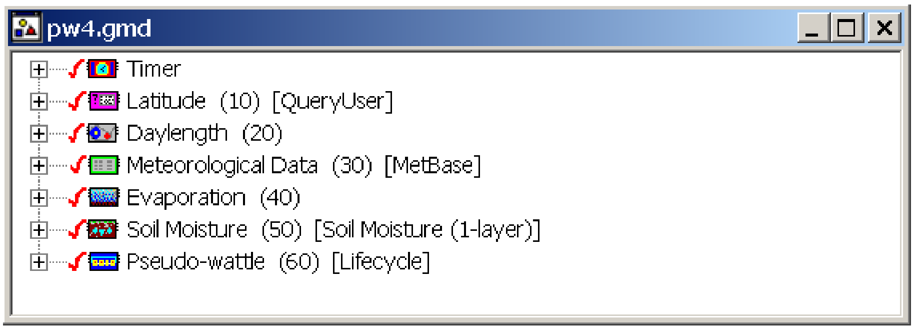


Figure 4.3 Modules present for the Pseudo-wattle model after saving and re-opening to obtain new sort order values.

The 'Lifecycle' seed module presently uses rainfall as its input variable to germination. This will be amended so that soil moisture levels control this function but the step function will be deleted and replaced.

1. Select the **'Seed'** lifestage **'Stage Transfer'** button
2. Highlight and select the **'Rainfall Germination Trigger'**
3. Select the **'Edit Component'** button
4. Change its name to **'Soil Moisture Induced Germination'**
5. Change the Independent Variable to **'Soil Moisture'**
6. Change the function to **'2-segment Linear'**
7. Specify the parameters as follows:
    a. for **'Line 1 X-intercept'** set the default to 0.3 and the lower and upper limits to 0 and 0.5 respectively
    b. for **'Line 1 Slope'** set the default to 3.2, the lower limit to 0 and the upper limit to 5
    c. for **'X-value at line intersection'** set the default to 0.55, the lower limit to 0.2 and the upper limit to 1
    d. for **'Line 2 Slope'** set the default to -3.2, the lower limit to -5 and the upper limit to 0
8. Select **'OK'** as necessary to return to the lifecycle window
9. Save the model.
10. In preparation for the next chapter, change the name of the file under '**Model**', '**Details**' to '**Pseudo-wattle5**' and save a copy of the model as '**Pseudo Wattle5.gmd**'

4.4 Running the Model

1. Load the Simulator and create a new Pseudo-wattle simulation file named 'Pseudowattle Tutorial 4.dxs'
2. The **'Model'** window should resemble Figure 4.5

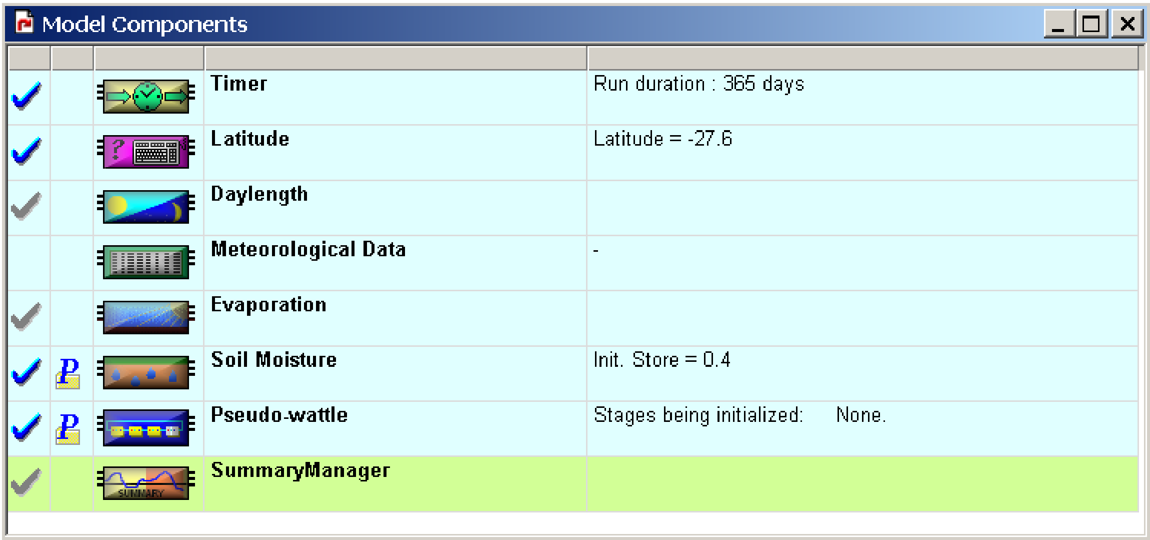


Figure 4.4 Model Components of Pseudo-wattle model

3. Make sure the latitude is set to -27.6 (as we will run the model for Amberley).
4. Initialise the **'Soil Moisture'** module by setting the current value to 0.05 to reflect the likely initial soil moisture levels found in the pseudo-wattle environment
5. Initialise the **'Meteorological Database'** module by setting:
   a. min. temp: column 7, width 5
   b. max. temp: column 12, width 5
   c. rainfall: column 17, width 4
   d. 9am Humidity: column 31, width 4
   e. 3pm Humidity: column 45, width 4.
6. In the **'Variable Represents'** panel, note that **'Average'** is selected for all the variables except **'Rainfall'** ***- see note at the end of this tutorial***
7. Set the 'No. of Lines' in the File Header to 2.
8. Check the Start Column and format for the Date.
9. Exit to the **'Model'** window.

All modules should now be 'ticked'.

10. Initialize the Pseudo-wattle module with 10 seeds at the start of simulation.
11. Ensure the **'Timer'** module is initialised to a start date of **'1/1/1965'**
12. Run the model for 800 days and create a chart to include soil moisture, total number of seeds and total number of seedlings; the results should be similar to Figure 4.6.

The Pseudo-wattle model is responding to the changes in soil moisture for germination. This is easily seen if the chart output is examined for seeds and seedlings. The non-linear trailing and leading edges of the curves show the effects of moisture on the seed germination rates as the individuals transfer from one lifestage to another.

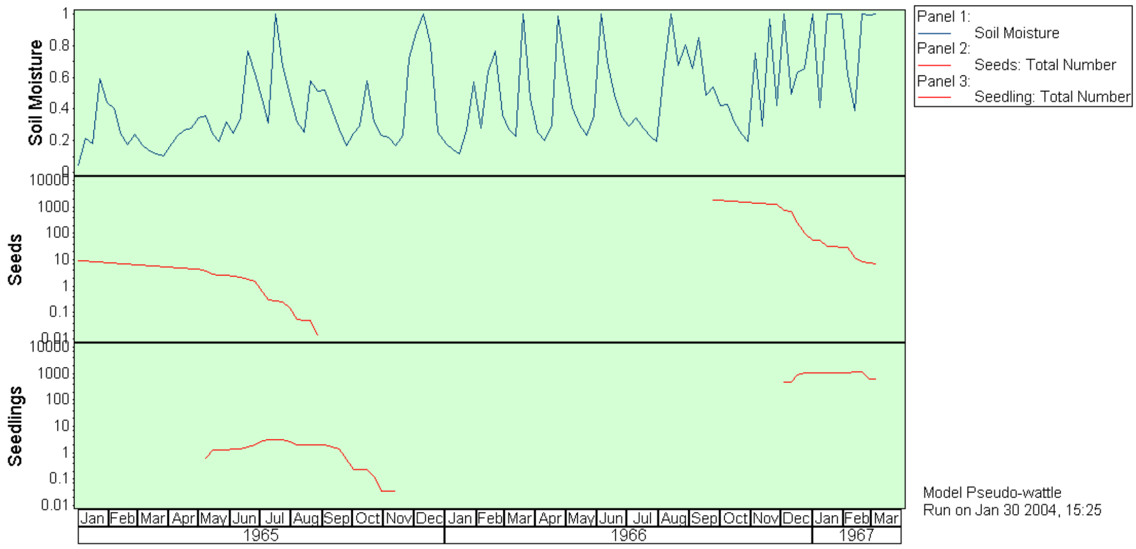


Figure 4.5 Pseudo-wattle model. Soil moisture included as a determinant for germination (Threshold 0.3). Run length 800 days.

It is also worth adjusting the value of the soil moisture germination threshold over its range to see the effects of altering this parameter.

Note: DYMEX is programmed to automatically set the variable type for the most common meteorological variables, e.g., **'Average'** is automatically selected for Humidity and Temperature in this model during formatting of the meteorological database. The model step is one week and for rainfall, the total for the week is required. For the remainder however, the average reading for each over the 7 days of the step is necessary. If, for example, **'Total'** was selected for humidity, the value would be the total of the readings for the week, which would result in a figure of several hundred percent humidity. The result on the model is that soil moisture would be driven permanently into saturation.

PW04.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

5 Soil Moisture and Temperature Effects on Seedling Development

5.1 Chronological and Physiological Age

The present Pseudo-wattle model uses chronological age to determine the timing of events in the entire lifecycle. Chronological age is unsatisfactory as the sole controlling influence on a plant's lifecycle because a plant's physiological development (and by implication its reproduction and mortality) can be independent of chronological age and may be largely controlled by temperature and moisture. This DYMEX tutorial demonstrates the process of modelling Pseudo-wattle seedling development based on physiological age and a temperature/soil moisture dependent rate of development.

Since physiological age now enters all further tutorial models, it is defined below:

> Physiological age measures the state of development of an individual with its units generally stated as a proportion of completed development. As an example, the development of Pseudo-wattle could be scaled from Germination (0) to arrival at adult reproduction (1). When plant development is temperature dependent, accumulation of physiological age is usually non-uniform with respect to time.

5.2 Pseudo-wattle Seedlings and Temperature

Since Pseudo-wattle's lifecycle is 'well known from published papers', the effects of temperature on seedling development are available and are presented in table form (Table 5.1).

Temperature (°C)	No. of weeks to develop from germination to juvenile plant
10	No development
15	48
20	18
23	14
25	12
30	12
40	No development

Table 5.1 Temperature effects on Pseudo-wattle adult plant development

The model developed in tutorial 4 remains essentially intact but the transition from germinated seedling to adult plant (currently determined by a chronological age of 12 weeks) becomes dependent upon physiological age and this in turn is dependent upon average daily temperature. All seedlings still become juvenile plants when they reach the required physiological age. Although the values of table 5.1 have been carefully 'chosen' to provide reasonably clear results, they demonstrate the principles on which DYMEX operates and the values have been selected with actual plant growth patterns under consideration.

The results of Table 5.1 can be amended to display rate of development per week. This is done by calculating the reciprocal of the number of weeks taken to develop to adult (which assumes that the value '1' represents the physiological age of an adult). For example, suppose a herb takes 50 weeks to develop from germinated seedling to adult plant; its rate of development per week would therefore be 0.02 (ie. 0.02 * 50 = 1). Table 5.2 shows the results for Pseudo-wattle.

Temperature (°C)	No. of weeks to develop to adult	Rate of development per week
10	No development	0
15	24	0.0208
20	18	0.0556
23	14	0.0714
25	12	0.0833
30	12	0.0833
40	No development	0

Table 5.2 Temperature controlled weekly development rates in Pseudo-wattle

These results can now be transposed into graphical format (Figure 5.1) which shows that Pseudo-wattle's growth pattern conforms very well to a three segment linear function which is readily modelled by DYMEX. An inspection of the graph shows that the AB segment has a threshold of 10^oC and a slope of approximately 0.0056; the section BC has a slope of 0.0 and the section CD has a slope of -0.0083.

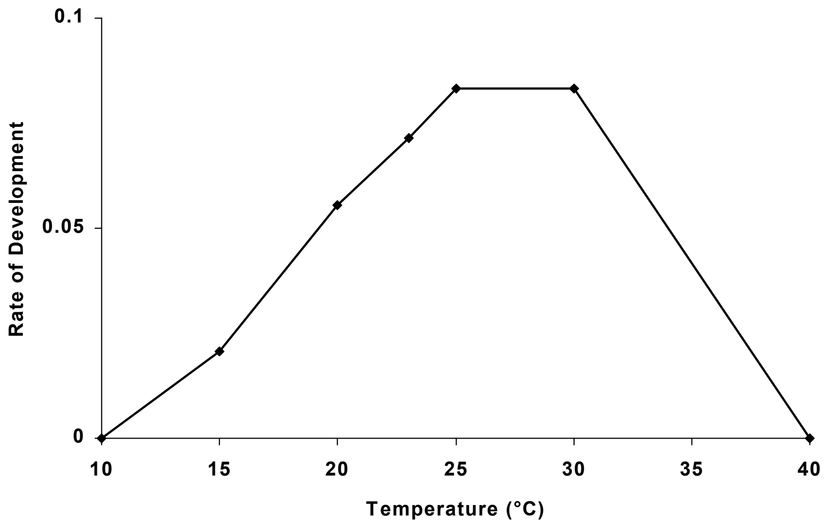


Figure 5.1 Rate of Temperature Controlled Physiological Development

5.3 The Degree-Day Concept

5.3.1 Calculating Degree Days Using Average Temperatures

Pseudo-wattle seedlings have a development threshold of 10°C, which implies that once the temperature rises above that threshold, development proceeds. Suppose that Pseudo-wattle seedlings were growing under ideal conditions where the temperature was maintained at a steady 27°C - in the mid-plateau area of the physiological development rate function. The result would be maximised development of the Pseudo-wattle seedlings limited only by other external climatic or physical factors. One way of considering the 'temperature-time bank account' for Pseudo-wattle's growth is shown in Figure 5.2.

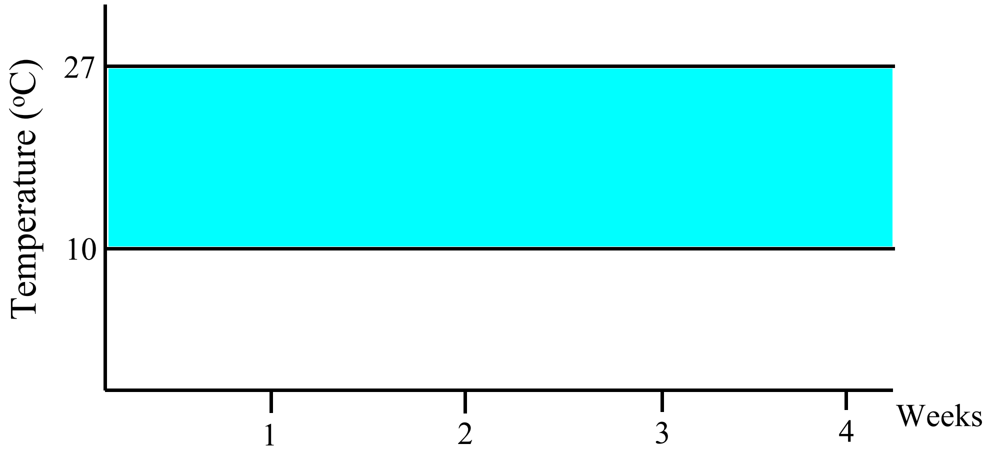


Figure 5.2 Degree Days for Pseudo-wattle (shaded area shows day degree accumulation)

The shaded area represents the available 'degree days' for each Pseudo-wattle plant over several weeks. Each day has a temperature difference of 17°C from the temperature threshold of 10°C. and in this case the accumulation of day degrees can be worked out by simple products (1 day = 17 degree days, 2 days = 34, etc.). Figure 5.2 resembles a greenhouse controlled environment in which the total degree days for the four week period would be 476 degree days. Suppose now the average temperature per week step is calculated and over a four week period is found to be 15°C, 6°C, 17°C and 23°C. Since for a simple average, the temperature is considered to be uniform throughout the weekly period, the result is a 'square wave' (Figure 5.3).

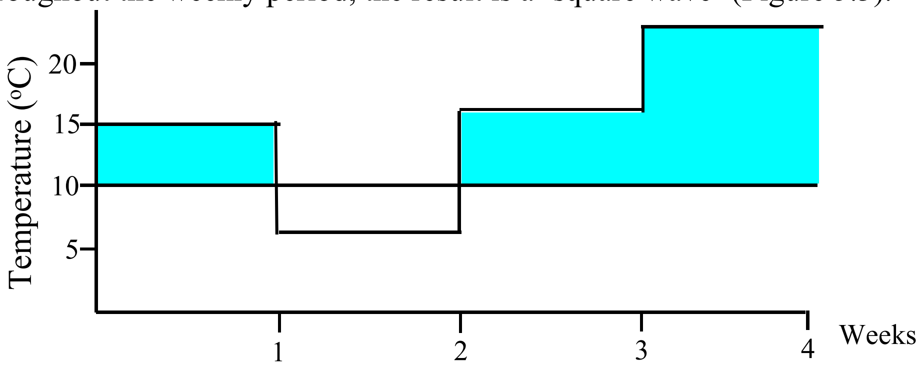


Figure 5.3 Degree days accumulation for Pseudo-wattle over four weeks using average temperature (shaded area represents degree day accumulation.)

When the temperature is above the threshold of 10°C, the Pseudo-wattle seedlings are able to develop and this is indicated by the shaded area; week 1 produces 35 degree days, none are produced in week 2, week 3 produces 49 and week 4 produces 91. During week 2, the temperature drops below the threshold for growth and the only effect on the Pseudo-wattle seedlings is that development temporarily ceases.

5.3.2 Calculating Degree Days Using the Circadian Cycle

Obviously, the 'square wave' of Figure 5.3 is also a poor approximation to the field situation because temperatures do not stay at the average value over the 24-hour period of a day. To compensate, DYMEX can apply a more or less sinusoidal wave shape for the 24-hour period. DYMEX uses the daily maximum and minimum temperatures of the meteorological database to fix the 'crest and trough' limits of a sinusoidal, circadian cycle of temperatures and interpolates for all the values it may require in between the two limits. Since the maxima and minima may fluctuate, DYMEX is able to smooth the 'circadian curve' so that it fits the daily fluctuations accurately (Figure 5.4). DYMEX can model several variations on the simple sine curve shape.

Once the temperature fluctuations are correctly modelled, DYMEX determines the degree days available for an organism's development by calculating the area under the circadian curve and above the threshold temperature for development. It does this by splitting the day into segments and calculating the area in each of the rectangular approximations thus formed. (The user is able to tell DYMEX how many segments are required, defining the precision of the estimated area under the curve, however in practice, 12 two-hour segments have been found to produce all the precision required for most situations.) In the following tutorial, the model will be modified so that circadian curve generated degree days will be incorporated. This will require the addition of a new Circadian module.

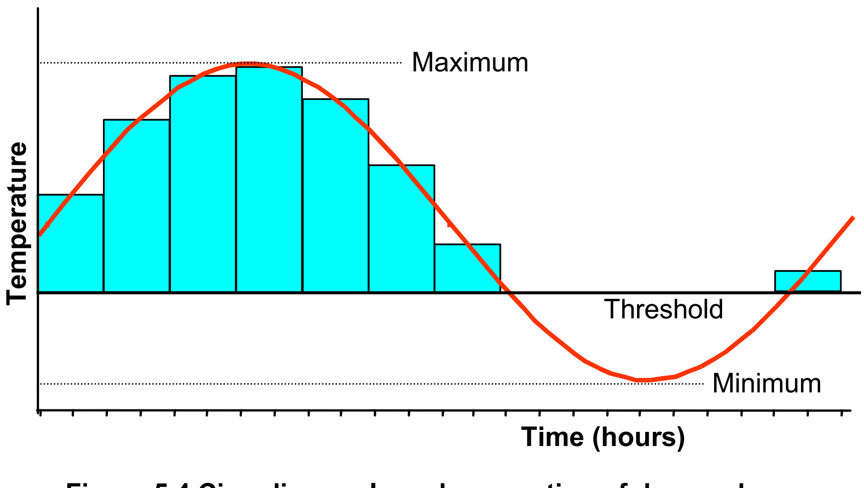


Figure 5.4 Circadian cycle and summation of degree days

5.4 Pseudo-wattle Seedlings and Soil Moisture

In the current model, seeds germinate on completion of the 8-week dormancy period and provided soil moisture is at least equal to 0.3. After germination, seedlings are very water dependent, and until a good root system is established they require ample soil moisture. The effects of soil moisture availability on growth rates are not as intense as those of temperature, but within limits, increased soil moisture does lead to increased seedling growth. Soil moisture levels will be revisited in the seedling lifestage when lack of soil moisture leads to seedling mortality. One suggested graphical model for the effects of soil moisture is displayed in figure 5.5.

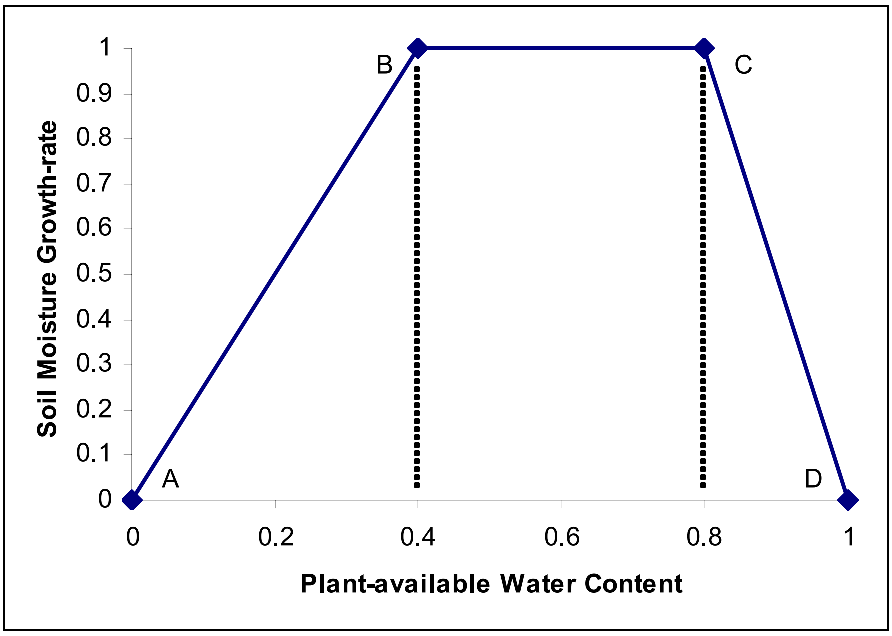


Figure 5.5 A suggested model for Soil Moisture Dependent Growth Rates for Pseudo-wattle seedlings

The graphical model of figure 5.5 implies that there are no moisture constraints on development if the soil moisture lies on or between the values of 0.4 and 0.8 but development decreases and eventually ceases if the soil moisture moves outside this range. This situation is readily modelled using a 3-segment linear function. Suppose that all dependent development ceases once soil moisture levels lower to 0 or if they increase to 1.0 (ie. Fully saturated soil). With these limits set, the slope of the line segment AB has a value of 2.5, BC has a value of 0 and CD has a value of -5.

The Temperature and Moisture Dependent Growth Rates will be multiplied together to give a final growth rate. This means that if either growth rate function results in maximum growth it produces a value of **'1'** which when multiplied allows the other function to control the overall growth rate.

5.5 Building the Model

Start the DYMEX Model Builder and open the Pseudo-wattle model. The 'Model' window will be displayed and all modules (currently seven) will be shown. The Circadian module is now added.

1. Start the Model Builder program and open the Pseudo-wattle model
2. From the menu bar, select **'Model'** followed by **'Add Module'** from the drop down menu

3. Select **‘Circadian’** followed by **‘OK’**
4. Rename the Circadian module **‘Daily Temperature Cycle’**
5. Select **‘Inputs’**
6. From **‘Inputs to be Linked’** list box, select **‘Daily Minimum Value’**
7. From **‘Link for Selected Variable’** list box, select **‘Minimum Temperature’**
8. Repeat steps 7 & 8 for **‘Daily Maximum Value’** and **‘Maximum Temperature’**
9. Select **‘OK’**
10. Check **‘Output’** is set to **‘Daily Cycle’** and then click on **‘Select’** to give **‘ +> ‘** beside the set variable
11. Rename the variable **‘Daily Temperature Cycle’**
12. Select the **‘Settings’** button and check that the default setting of **‘Sine’** is correctly shown - if not, correct this - and set the **‘Number of segments’** to **12**.
13. Return to the module window and set the sort order so that the **‘Daily Temperature Cycle (Circadian)’** module is placed underneath the **‘Meteorological Data’** module
14. Exit to the **‘Model’** window.

The next step is to alter the Seedling lifestage so that its development is controlled by both temperature and soil moisture. In the following steps, the function for temperature controlled development will be set up first, followed by soil moisture development.

1. Double click on **‘Pseudo-wattle’** line in the Model window to obtain the **‘Pseudo-Wattle Lifecycle’** window
2. Select the **‘Development’** button of the **‘Seedling’** lifestage
3. Name the module ‘Seedling Development’
4. Select **‘Function’** to obtain the ‘Function’ dialogue box
5. From the function scroll box select the **‘3-segment Linear’** function
6. Select **‘Independent Variable’** scroll button
7. From scroll list, select **‘Daily Temperature Cycle’**
8. From the **‘Name’** edit box, select the **‘Change’** button
9. In the resulting edit box, type in a suitable name (e.g., **‘Temperature controlled Development’**)
10. Select the **‘p1: Line 1 intercept’** parameter from the **‘Parameters’** list box, then click on the **‘Edit Parameter’** button to open the **‘Set Parameter Properties’** dialogue box

The user should set descriptive (but not excessively long) names for each of the parameters. Although this is not strictly necessary, it can help later when interpreting the meaning of a parameter or setting its value.

11. Rename the parameter with a descriptive name (e.g., ‘***Temperature Development Threshold***’)
12. Enter the values 0, 15 and 10 in the lower limit, upper limit and default boxes respectively
13. Click on **Ok**, select the **‘p2: Line 1 Slope’** parameter and click on the **‘Edit Parameter’** button

14. Set the lower and upper limit values to 0 and 1 respectively, set the default to 0.0056 and re-name the parameter (eg ***'Low temp development rate slope'***)
15. Click on **Ok**, select the **'p3: X value at intersection of lines 1,2'** parameter and click on the **'Edit Parameter'** button
16. Set the upper and lower limits to 20 and 30 and the default to 25
17. Re-name the parameter suitably (e.g., ***'Lower limit of optimal development'***)
18. Exit, then select the next parameter (**'p4: Line 2 Slope'**) for editing
19. Set all limits and the default to 0
20. Rename the parameter (eg ***'Development rate plateau'***)
21. Exit, then select the next parameter **('p5: X-value at intersection of lines 2,3'**) for editing
22. Set lower and upper limits to 25 and 35 respectively and the default to 30
23. Rename the parameter (eg ***'Upper limit of optimal development'***)
24. Exit, then select the next parameter **('p6: Line 3 Slope'**) for editing
25. Set the lower and upper limits to -1.0 and 0 respectively and the default value to -0.0083
26. Rename the parameter (e.g., ***'High temp development rate slope'***)
27. Select **'OK'** as necessary to return to the **'Seedling Development'** process dialogue.

The Seedling lifestage is now modified so that soil moisture also controls the lifestage development.

1. Select the **'Function'** button to open the **'Function'** dialogue
2. Select **'Soil Moisture'** as the **'Independent Variable'**
3. From the scroll list select **'3-segment linear'** as the function
4. Change the function name to **'Soil Moisture controlled development'**
5. Select the **'p1: Line 1 intercept'** parameter from the **'Parameters'** list box, then click on the **'Edit Parameter'** button to open the **'Set Parameter Properties'** dialogue box
6. Rename the parameter with a descriptive name (e.g., '***Moisture Development Threshold***')
7. Set the value to 0, the lower limit to 0 and the upper limit to 0.5
8. Click on **Ok**, select the **'p2: Line 1 Slope'** parameter and click on the **'Edit Parameter'** button
9. Rename the parameter (eg ***'Low moisture development rate slope'***) and set the default value to 2.5, the lower limit to 0 and the upper limit to 5
10. Click on **Ok**, select the **'p3: X value at intersection of lines 1,2'** parameter and click on the **'Edit Parameter'** button
11. Rename the parameter (e.g., to '***Lower limit of optimal development'***) and set the default value to 0.4, the lower limit to 0 and the upper limit to 1
12. Exit, then select the next parameter (**'p4: Line 2 Slope'**) for editing
13. Rename the parameter (eg ***'Development rate plateau'***) and set the default and both limits to 0
14. Exit, then select the next parameter **('p5: X-value at intersection of lines 2,3'**) for editing

15. Rename the parameter (eg ***'Upper limit of optimal development'***) and set the default to 0.8, the lower limit to 0 and the upper limit to 2
16. Exit, then select the next parameter **('p6: Line 3 Slope'**) for editing
17. Rename the parameter (e.g., ***'High moisture development rate slope'***) and set the default value to -5, the lower limit to -7 and the upper limit to 0
18. Select **'OK'** as necessary to return to the **'Seedling Development'** process dialogue.

Since both temperature and soil moisture now control physiological development, their effects must be combined to give a single output. This is done with a button which previously has been 'greyed out' as when there is only a single factor controlling a process, its operation is not applicable.

19. Click the **'Set Combination Rule'** button
20. Ensure that the combination rule is set to **'Product'** and then exit to the **'Lifecycle'** window

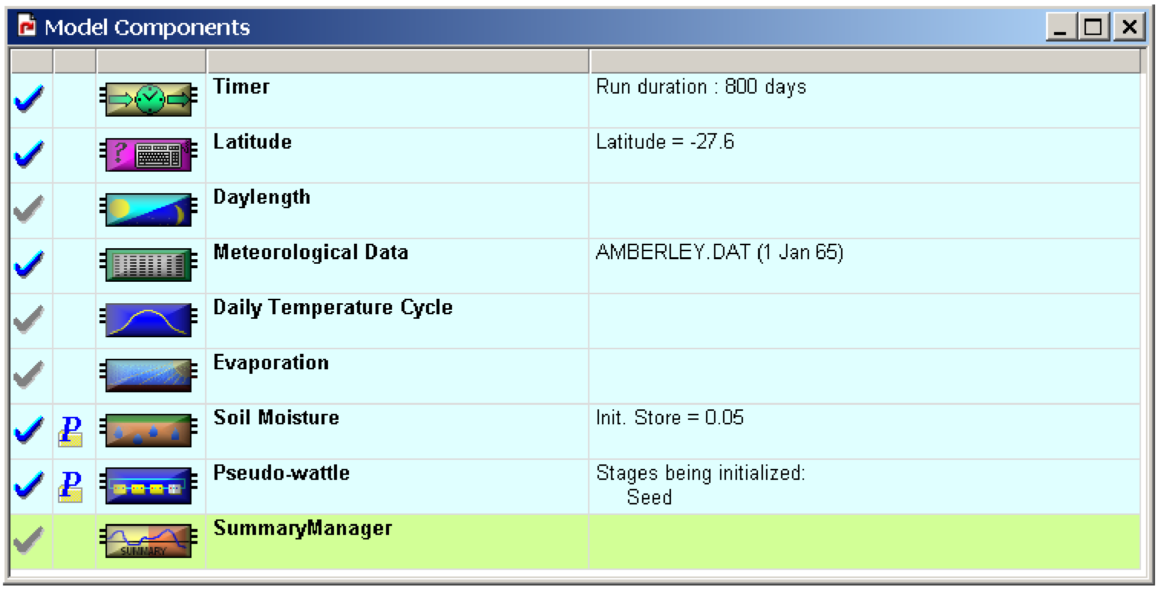
There will now be a red tick on the Seedling Development button. All that remains is to set the stage transfer of the seedlings to juvenile plants and select some additional outputs from the lifestage so that the results of the alterations can be observed.

21. Select the **'Stage Transfer'** button of the **'Seedling'** lifestage
22. With the process component selected (there should be just one), click on the **'Edit Component'** button.
23. Change the function's **'Independent Variable'** to **'Physiological Age'**
24. Click on the threshold parameter (***p1***) in the **Parameters** list box and change the default, lower limit and upper limit to 1.
25. Select **'Ok'** as necessary to return to the **Lifecycle** window
26. Select the **'Outputs'** button and from the scroll list select **'Development Time'** and **'Average Physiological Age'** – rename to the suggested names
27. Return to the **'Lifecycle'** window and save the model.

5.6 Running the Improved Model

Note: When the model is now reloaded, you may get messages that indicate missing or out-of-range parameter values. This is due to the fact that the parameter file (.gmp) created previously does not correspond to the new model anymore. While developing a model, it is useful to avoid using the parameter file altogether. This is done by checking the "**Use Parameter Defaults (not parameter file)**" box in the current model preferences dialog (available when the model is loaded). The model simulation should then be saved (File|Save) and the model closed and reloaded for the change to take effect.

Figure 5.6 Module Components Window



1. Open the model by clicking on the New option in the File menu, then create a new simulation file named ‘*Pseudowattle Tutorial 5*’
2. Set the latitude to -27.6 in the UserQuery module.
3. Initialise the **‘Soil Moisture’** module by setting the current value to 0.05 to reflect the likely initial soil moisture levels found in the pseudo-wattle environment
4. Initialise the **‘Meteorological Database’** module by setting:
    a. min. temp: column 7, width 5
    b. max. temp: column 12, width 5
    c. rainfall: column 17, width 4
    d. 9am Humidity: column 31, width 4
    e. 3pm Humidity: column 45, width 4.
5. In the **‘Variable Represents’** panel, note that **‘Average’** is selected for all the variables except **‘Rainfall’ - SEE NOTE AT END OF TUTORIAL**
6. Set the ‘No. of Lines’ in the File Header to 2.
7. Check the Start Column and format for the Date.
8. Exit to the **‘Model’** window.
9. Set the initial number of seeds in the lifecycle module to 10.

All modules should now be ‘ticked’ and the model components window in the Simulator should resemble figure 5.6. Run the model for a period of **500 days**. If the total numbers for seeds and seedlings, as well as the seedling development time is charted, the results will resemble figure 5.7. The effect of temperature and physiological development is very easily seen. The original model defined a 12 week threshold for the transfer of individuals from seedling to juvenile. Inspection of figure 5.7 shows that at least a 150-day period (21 weeks) is required for the seedlings to transfer to the adult stage. At the beginning of May 1965, when the first seedlings appear, the development graph shows that these seedlings will complete their development in about 240 days. This takes us to the end of December, and the “seedling” trace on the upper panel does show a decline in seedling number at that time, as they become juveniles. The decline in development time for seedlings germinating after June is probably mostly due to the fact that these seedlings spend a smaller proportion of their total development period in the winter months (when the rate of degree day accumulation will be lower) but the model also includes the effects of soil moisture and these will definitely control some aspects of seedling development.

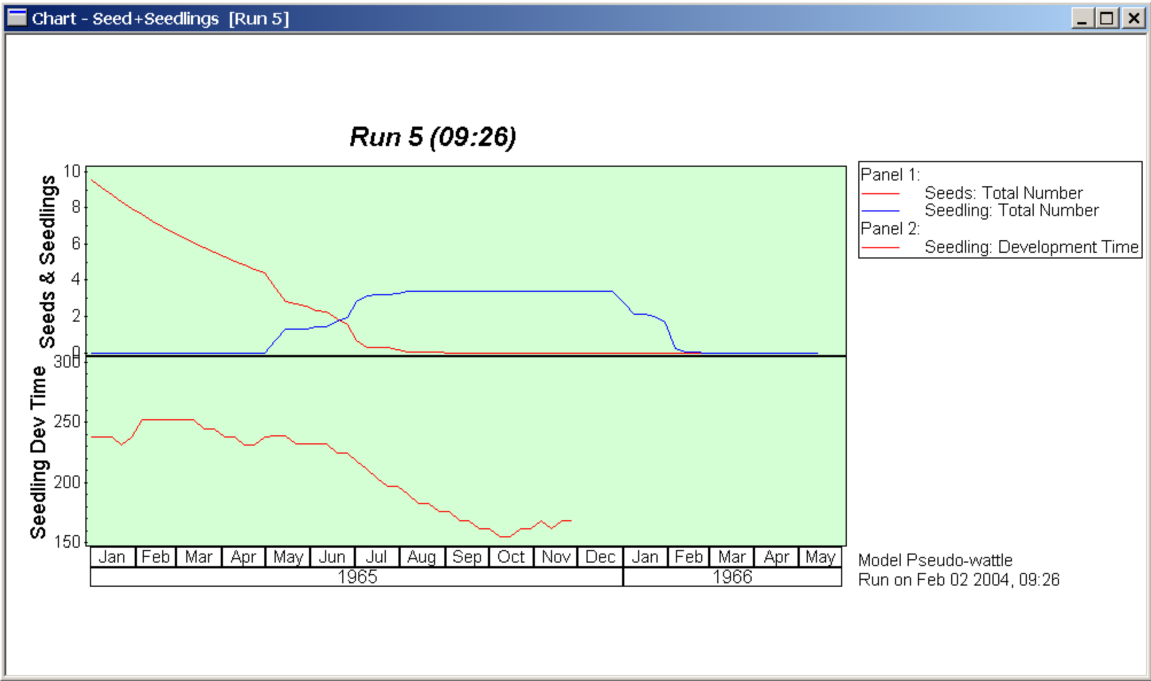


Figure 5.7 Pseudo-wattle population for a 500 day run. Physiological development of seedlings temperature/moisture controlled.

If the model is run over a three year period and all lifestage outputs are graphed, the results will be similar to figure 5.8. This the progression from seed to seedling, juvenile and then adult plants. Adult plants appear toward the end of year 2, and start producing seeds at the end of September (day 258), as specified by the Egg Production function.

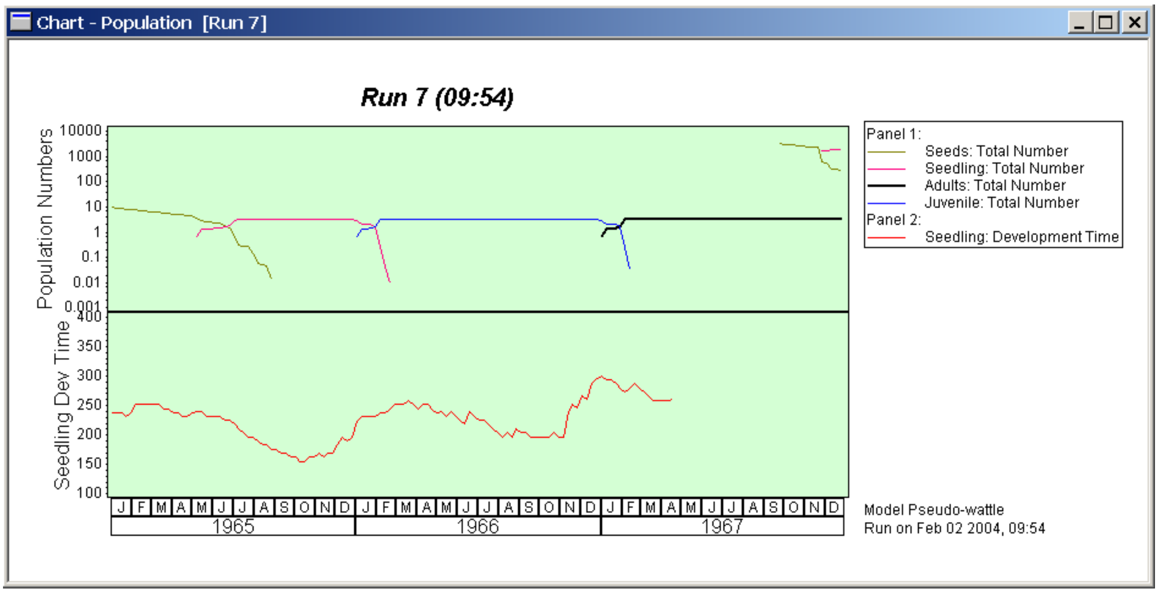


Figure 5.8 All lifestage outputs of Pseudo-wattle under temperature and soil moisture controlled physiological development. (log scale, 3-year run.)

Another interesting/useful operation is to chart the outputs for seedling development, soil moisture and daily temperature cycle over a longer period of time. Figure 5.9 demonstrates the results over a 2000 day interval. Without the effects of soil moisture, seedling development time would become more or less cyclical in step with the movements of the annual shifts of the daily temperature cycle; ie. development times would increase during the 'winter' period of May to October and decrease during the 'summer' months of November to April. With the effects of soil moisture added, the results are far more complex and the effects of saturated soil levels are very clearly seen during the initial months of 1967 when there was a prolonged period of rain. Seedling development times increase dramatically when temperatures are at their warmest during the year. (To be able to obtain this chart, the user must have selected seedling development time as an output for the seedling stage while in the Builder program.) Note that the Seddling Development Time trace finishes before the other model variables – this is because seedlings that germinate in the last months of the simulation do not have enough time to develop into juveniles before the simulation terminates and therefore DYMEX cannot calculate their development time.

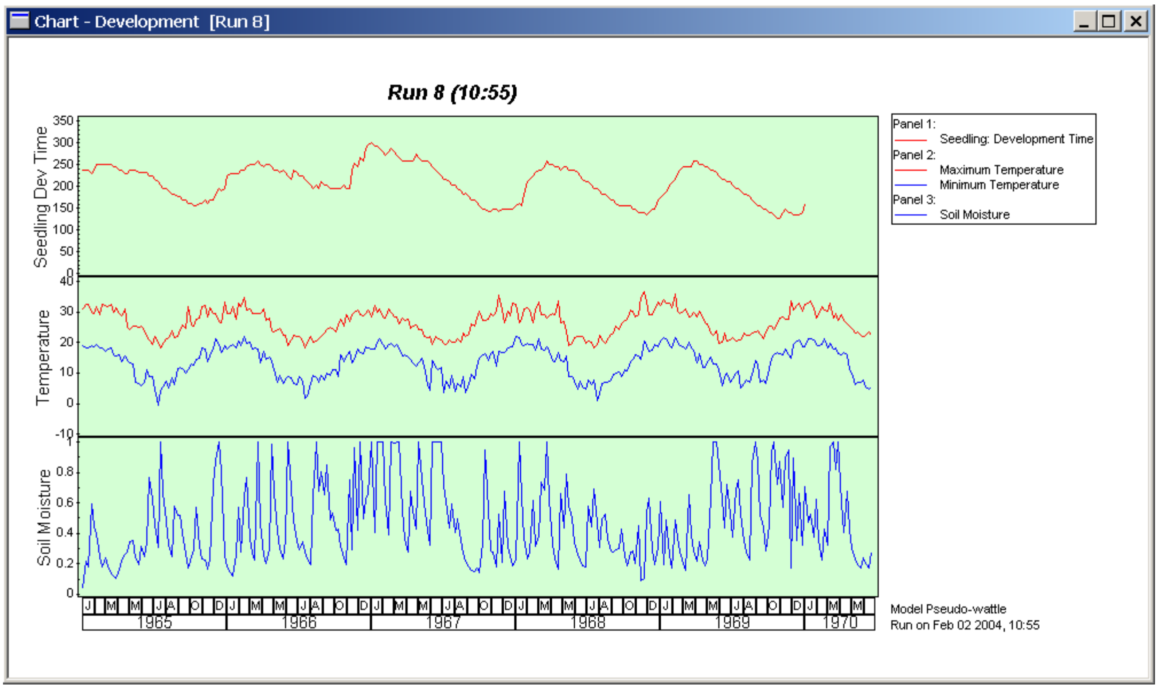


Figure 5.9 Seedling development time, soil moisture and daily temperature cycle for Pseudo-wattle under Amberley climatic conditions - 2000 days.

As a final display of the current model's results, figure 5.10 displays the pseudo-wattle population trends over a 10 year period in logarithmic format. This chart merely shows that the population is still unbounded and continues to increase more or less exponentially with time.

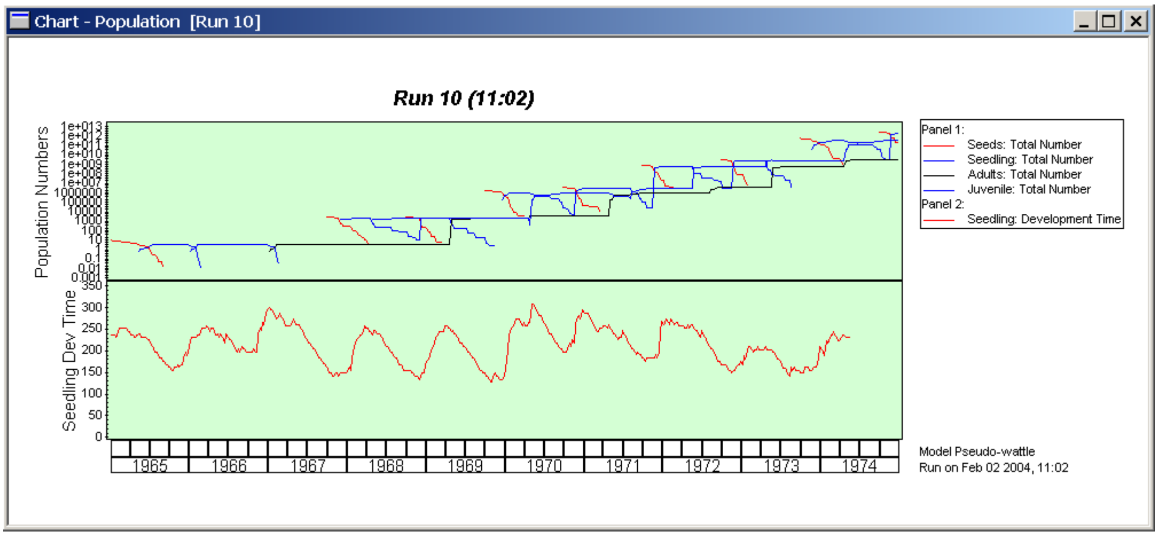


Figure 5.10 Pseudo-wattle populations; 10 year interval, logarithmic format

PW05.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

Completing Physiological Development in the Juvenile and Adult Lifestages

6.1 Introduction

In tutorial 5, the concepts of physiological age and its application under temperature and soil moisture controls were introduced to the seedling lifestage. All lifestages of the perennial are subject to physiological development as the driving variable rather than solely chronological age. In this tutorial, physiological age with appropriate parameters for temperature and soil moisture will be introduced to the juvenile and adult plant lifestages. Since the procedures follow the same logic as that already introduced in tutorial 5, there will be no further explanation of the requirements and only the parameter values will be given followed by the procedures required for their introduction into the model.

6.2 Required Model Parameters

6.2.1 The Juvenile Lifestage

The juvenile lifestage exhibits temperature dependent physiological development once a threshold of 12°C has been passed. The range of field temperatures encountered by the plant does not produce any limitation in growth rates unless increased beyond that normally encountered in its habitat range. Under these circumstances, a linear above threshold function is best used to model its behaviour and the slope can be given a value of 0.002.

Soil moisture effects on growth rates of the Pseudo-wattle juvenile lifestage mirror those encountered in the seedling lifestage although the juvenile is better able to withstand dry soil conditions. A 3-segment linear function will be used with a soil moisture 'best rate of development region' (slope = 0) determined by values of soil moisture from 0.2-0.8. Since the juvenile is considered able to withstand complete drought for limited periods, the initial 'x-value' will be 0 and the slope of the initial segment of the function is therefore equal to 5. The negative slope for the third linear segment will be set to the same value as for the seedling, -0.833.

Both the soil moisture and temperature functions will be combined by the product rule.

6.2.2 The Adult Lifestage

The adult lifestage exhibits both soil moisture and temperature dependent physiological development similar to that encountered in the Juvenile lifestage. All parameter values will therefore be identical and will also be combined by the product rule.

6.3 Building the Model

The procedures employed in the DYMEX Builder to complete the alterations given above are almost identical to those already described in tutorial 5. In addition, there are no new modules to be added. For these reasons, only the main points will be covered in the step description given below. The user is referred to the previous tutorial for additional information if it is needed.

1. Open the Pseudo-wattle model and then open the **'Lifecycle'** window

2. Select the **'Development'** button in the **'Juvenile'** lifestage and then complete the following operations:
   a. Set up a new function with **'Daily Temperature Cycle'** as its **'Independent Variable'**
   b. Select a **'Linear above Threshold'** function from the list
   c. Re-name the function suitably (e.g., *'Temperature-controlled Development'*)
   d. Set the parameters as follows:
      i. Set the threshold default to 12 and the lower and upper limits to 10 and 20 respectively
      ii. set the slope default to 0.002 with the lower and upper limits to 0 and 0.01 respectively.
      iii. give all parameters suitable names as required.
   e. Return to the **'Juvenile - Development'** process dialogue
   f. Name the process '*Juvenile Development'*
   g. Set up a new function with **'Soil Moisture'** as its **'Independent Variable'**
   h. Select a **'3-segment Linear with Plateau'** function from the list
   i. Set the parameters as follows:
      i. set the **'Line 1 slope'** default to 5 and the lower and upper limits to 2.5 and 7.5
      ii. set the **'X- value at intersection of lines 1,2'** default to 0.2 and the lower and upper limits to 0.1 and 0.5
      iii. set the **'X-value at intersection of lines 2,3'** default to 0.8 and the lower and upper limits to 0.5 and 1
      iv. set the **'line 3 slope'** default to -0.833, the lower limit to -1.0 and the upper limit to –0.5
      v. set the **'Line 2 Y-intercept'** to 1 for upper limit, lower limit and default fields.
      vi. give all parameters suitable names as necessary.
   j. Return to the **'Juvenile - Development'** process dialogue and set the combination rule to **'Product'**
3. Repeat the process of steps 1 and 2 for the **'Adult'** lifestage. Alternatively, the following "Copy/Paste" shortcut method can be used:
   a. In the Juvenile Development process dialogue, highlight the **'Temperature-controlled development'** component and click **'Copy'**.
   b. Click **OK**
   c. Open the Development process dialogue in the Adult lifestage
   d. Click **'Paste'** and name the process "*Adult Development*"
   e. Click **OK** and return to the Juvenile Development process.
   f. Highlight the Soil Moisture component of the process and click **'Copy'**
   g. Click **OK**
   h. Re-open the Development process in the Adult lifestage
   i. Click **'Paste'** and set the combination rule to **'Product'**
   j. Click **'OK'** to return to the lifecycle window.
4. Select the **'Juvenile to Adult' Transfer** button and edit the transfer function as follows:
   a. Change the Independent Variable to **'Physiological Age'**
   b. Change the **'Juvenile Age at Maturation'** values to default of 1 and the

lower and upper limits to 0 and 1 respectively. Rename the variable to **'Reproductive Maturation Threshold'**

c. Exit back to the **'Lifecycle'** window and save the model.

The **Adult** reproduction functions are now amended so that the plant's reproduction is a function of their physiological age . This is done by combining physiological maturity with the triggering function that currently controls reproduction. The combination under which the two functions will operate will be a product. The reasons for this are that until the plant reaches physiological maturity, the output of the physiological function will be 0, therefore even if the flowering/seedling trigger is applied to an adult plant, it cannot reproduce until the physiological function produces an output of 1.

5. Select the **'Reproduction'** button of the **'Adult'** lifestage followed by the **'Progeny Production'** button
6. Select the **'Function'** button to add a new function and open the relevant window - then complete the following steps:
   a. Select a **'Step'** function and **'Physiological Age'** as the **'Independent Variable'**
   b. Set the **'p1: Threshold'** default to 1 and the lower and upper limits to 0 and 1 respectively. Rename the parameter '*Threshold Development for Reproduction*'.
   c. Set all values of **'p2: Step Height'** to 1, and rename the parameter to '*Proportion reproducing at threshold*'
   d. Return to the **'Seed Production (Adult)'** process panel and set the **'Combination Rule'** to **'Product'**
   e. Exit to the **'Lifecycle'** window.
7. Select the **'Outputs'** buttons respectively on the **'Juvenile'** and **'Adult'** lifestages and make sure that **'Average Physiological Age'** and **'Development Time'** are selected for each and that each is renamed to ensure no name confusion in the outputs.
8. Save the model.

6.4 Running the Model

Run the model for a period of 10 years (3650 days) and then produce an output showing the total numbers in each lifestage, and the development times for each stage. Use logarithmic scaling for the population numbers. The results should be similar to figure 6.1.

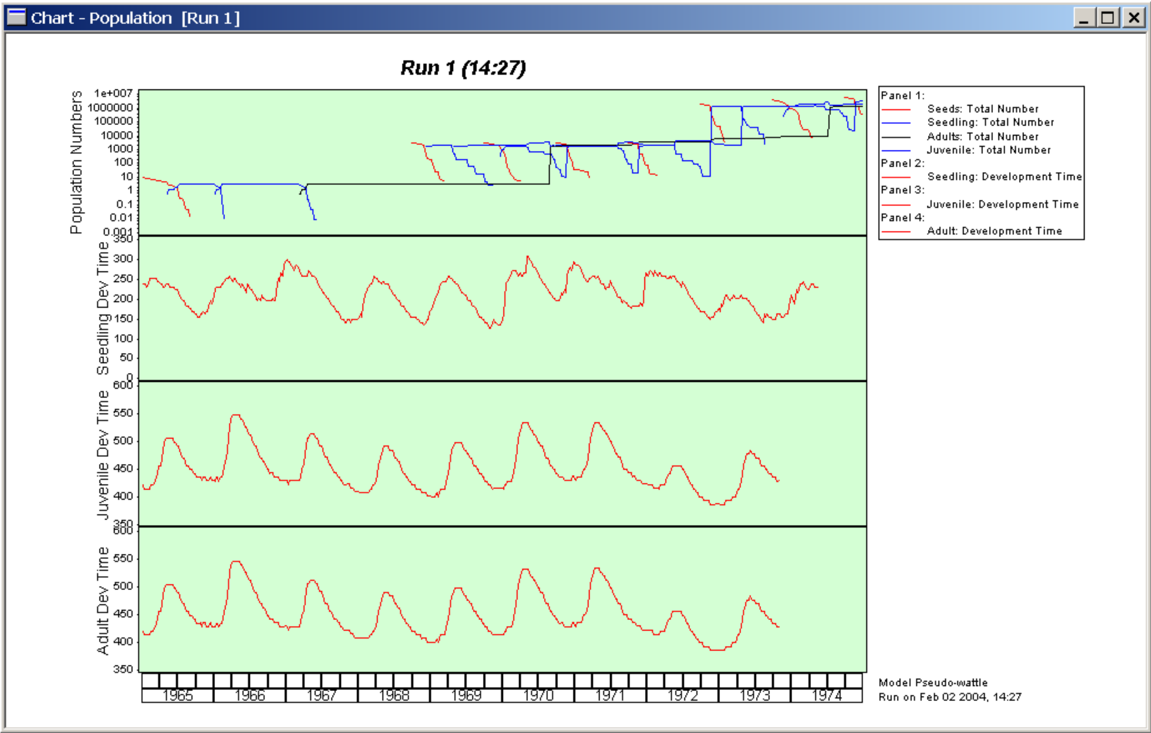


Figure 6.1 Pseudo-wattle populations over a 10 year period with temperature and soil moisture controlled physiological development applied to seedling, juvenile and adult lifestages. (logarithmic scaling in Population panel)

The model's increasing complexity is now easily seen with the seedling population numbers no longer forming 'discrete' groups in each year but prolonging their period in the seedling lifestage in accordance with the environmental parameters which define their physiological development.

In the current model, the population is allowed to increase unchecked. In the field this is impossible: various forms of mortality operate to remove individuals from the entire population and in addition the size of the resources (e.g. living space) will ultimately define how many individuals can exist. Mortality in its various forms will be applied to the model during the next few tutorials.

> ***PW06.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial***

7 Seedling and Juvenile Mortalities

7.1 Introduction

Populations passing through the seedling and the juvenile lifestages suffer some degree of mortality. Exactly what form of mortality suffered depends completely on the environmental circumstances of the population and its individuals at any given time - excluding for the moment any mortality that is produced by human activity. Some of the factors that might play a part include soil moisture, excessive temperature, grazing activity by animals, competition by more mature individuals of the population, etc. For this tutorial, the intention is only to give an indication of what factors might contribute to mortality in these two lifestages and therefore, to preserve simplicity, only one of the many factors - soil moisture - will be included in the model.

7.2 Soil Moisture Mortality

The effects of soil moisture on mortality of the seedling lifestage will be greater than on the juvenile lifestage because the juvenile has had more time to establish its root system and is therefore better able to withstand soil desiccation, whether from lack of rainfall or competition from other members of the entire plant population of its habitat. Additionally, it is possible for seedlings or juveniles to be killed if there is too much soil moisture. Before the consequences of varying soil moisture can be modelled, information on how soil moisture affects the mortality the two lifestages must be obtained. For example, not only will the effects of soil moisture be greater in the seedling lifestage, but the mortality effects should decrease as the seedlings age and establish better root systems (this more complex behaviour will not be included here). Seedlings should therefore be very sensitive to the lack of soil moisture at first but be much more resistant to desiccation as they near the juvenile lifestage.

For the purposes of the tutorial, the soil moisture effects on pseudo-wattle are as follows:

Seedling

Pseudo-wattle seedlings begin to suffer mortality due to soil moisture once it falls to 0.25. For simplicity, the assumption is made that 62.5% mortality of seedlings occurs during each week that the soil moisture falls to 0.00 and that the controlling function has the form of 'linear below threshold'. This will produce a slope that is equal to –2.5. The values given above allow for some revival from complete desiccation by the seedlings (see Figure 7.1). To compensate for mortality induced by too much soil moisture, a second function will be included. This will take the form of a linear above threshold function. Mortality effects due to soil moisture begin once soil moisture reaches a value of 0.9. In order to calculate a slope, an 'imaginary value of 2.5' for the soil moisture (see previous tutorial for explanatory notes) will be used and seedlings will be defined to suffer a weekly mortality rate of about 20% if this value is reached. This will produce a slope of 0.125. In practice, as has already been stated, the DYMEX models never have soil moisture levels greater than the saturation value of 1. The use of 'imaginary levels' greater than 1 suggest run-off and continuous flooding which can produce considerable plant mortality. The use of these imaginary values allows simple calculation of appropriate slopes.

Juvenile

Pseudo-wattle juvenile plants begin to suffer mortality due to soil moisture once it falls to 0.18, however the rate of mortality is much lower and the assumption is made that only 2% mortality per week is reached if the soil moisture temporarily becomes 0. The resulting slope of the linear below threshold function is -0.01. This simplistic view may not be adequate in the field as many plants, including *Acacia spp.*, have mechanisms which allow the plant to withstand extended periods of drought and merely reaching a soil moisture of 0 may not be sufficient to kill the plant - the absence of soil moisture may have to be maintained for an extended period of time.

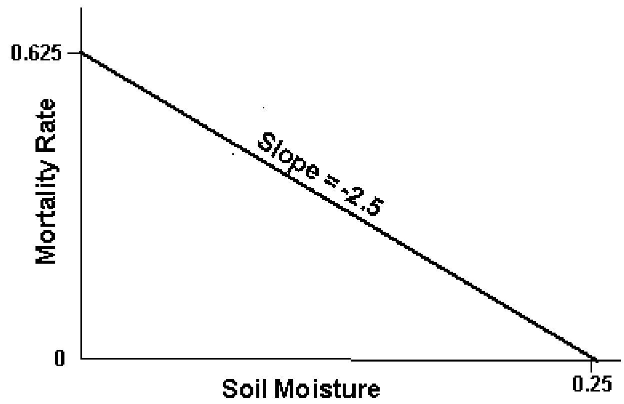


Figure 7.1 Seedling Soil Moisture Mortality function

7.3 Combining Mortality Functions with the Product Rule

(Note: DYMEX has two forms of the product rule which are available when combining functions. If a simple product is desired, choose the function labelled **'Product'**. If the special mortality product rule (as described below) is necessary, choose the button labelled **'Complement product'**.)

All mortality function rates are expressed as values from 0 to 1. If only one mortality function controls a population's survival, it cannot have a permanent value of 0 because the species would then be immortal; similarly, it cannot have a permanent value of 1 for the organism would then be extinct. If a mortality rate was 0.4, it would mean that when applied to the population, four tenths of it would die during each timestep.

In practice, several mortality functions will affect each lifestage and their effects must be combined to give a single mortality rate per time-step. For the purposes of Tutorial 7, the 'Complement product' combination rule will be used to set the total effect of the rates of all mortality functions in each lifestage. Suppose that several **M**ortality **F**unctions, MF_1, MF_2, MF_3 and MF_4 control mortality in a plant. The 'complement product' combination rule is then given by the equation:

$$\text{Combined Mortality Rate} = 1 - (1 - MF_1) \times (1 - MF_2) \times (1 - MF_3) \times (1 - MF_4)$$

Inspection of this equation will show that if any of the Mortality Functions has a value of '0', then the expression inside the bracket becomes: 1 - 0 which equals 1. When multiplied, this produces no change to the expression's value so the resultant effect is to make the population mortality dependent upon the remaining variables. If on the other hand, any of the Mortality Functions equals 1, then the result is: 1 - 1 which equals 0. This result reduces the value of the product series to zero and so nothing is subtracted from the initial 1 on the right hand of the equation. Therefore, if any mortality rate is equal to 1, the combined mortality rate also becomes

1 and the population becomes extinct.

7.4 Building the Model

Start the DYMEX model builder and load the Pseudo-wattle model. Then continue with the following steps:

1. Open the **'Lifecycle'** module for editing
2. Select the **'Seedling Mortality'** button and open its selection panel
3. Select the **'Continuous'** button
4. Name the process '*Continuous Seedling Mortality'*
5. Select the **'Function'** button
6. Set the Independent Variable to **'Soil Moisture'**
7. Set the function to **'Linear below Threshold'**
8. Change its name to **'Drought Mortality'**
9. Set the **'p1:Threshold'** parameter to default to 0.18, with lower and upper limits of 0 and 0.5, respectively, and name it '*Drought Mortality Threshold*'
10. Set the **'p2: Slope'** parameter to default to -3.0, with lower and upper limits of –6 and 0, respectively. Rename to *'Drought Mortality Slope'*
11. Exit to the **'Mortality (Seedling)'** process dialogue and re-select **'Function'**
12. Name the function **'Water-logging Mortality'**
13. Set the Independent Variable to **'Soil Moisture'** and select a **'Linear above Threshold'** function
14. Set the **'p1: Threshold'** parameter's default to 0.9, its lower limit to 0.5 and its upper limit to 1, and rename it to *'Wet MortalityThreshold'*

17. Set the **'p2: Slope'** parameter's default to 0.125, its lower limit to 0 and its upper limit to 0.5, and rename it to *'Wet Mortality Slope'*
18. Exit to the **'Mortality (Seedling)'** process dialogue and select the **'Set Combination Rule'** button
19. Ensure that **'Complement Product'** is selected
20. Exit to the **'Lifecycle'** window and save the model.

The next section completes the mortality function for the juvenile lifestage.

1. Select the **'Mortality'** button of the **'Juvenile'** lifestage
2. Select the **'Continuous'** button and open the **'Mortality (Juvenile)'** process dialogue
3. Name the process '*Continuous Juvenile Mortality'*
4. Click on the **'Function'** button to create a "function" component
5. Set the **'Independent Variable'** to **'Soil Moisture'**
6. Re-name function *'Drought Mortality'*
7. Select **'Linear Below Threshold'** as the function
8. Set the **'p1: Threshold'** default value to 0.18, the upper limit to 0.25 and the lower limit to 0
9. Set the **'p2: Slope'** default value to -0.01, the upper limit to 0 and the lower limit to -0.1
10. Exit to the **'Lifecycle'** window and save the model.

To be able to easily examine the effects of the new mortality processes, a new output will now be added to each of theses stages. This output (Proportion Stage Mortality) reports the proportion of individuals that have died during each timestep due to continuous or establishment mortality processes.

1. Select the **'Outputs'** button of the **'Seedling'** lifestage
2. Scroll down to and highlight the **'Proportion Stage Mortality'** output in the list, and click the **'Select'** button.
3. Rename the output to *'Seedling: Mortality'*
4. Note that the **'Value to use when no cohorts are present'** box can be left blank as the new variable will not be used as an input
5. Repeat steps 1-4 for the Juvenile stage, naming the variable *'Juvenile: Mortality'*
6. Exit to the **'Lifecycle'** window and save the model.

7.5 Running the Model

Increase the number of seeds present to 1000. Run the model for a period of 3650 days. The results of this simulation are shown in figure 7.2, where population numbers (logarithmic scale) and seedling and juvenile mortalities are plotted in separate panels.

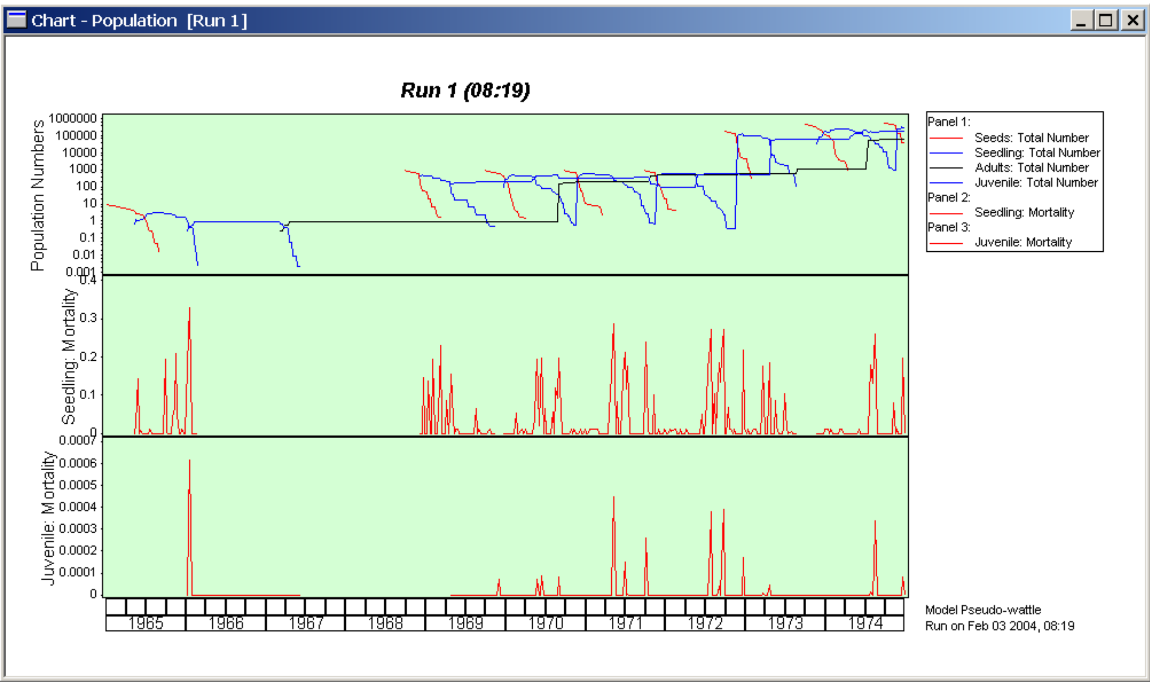


Figure 7.2 Total Lifestage Numbers for Pseudo-wattle over a 10 year period

The size of the adult Pseudo-wattle population after 10 years is now much less than that of the adult population of the previous tutorial even though the initialising seed numbers have been increased to 1000. The mortality effects have reduced the population but not controlled it: the graphs still show a steady increase in the population. This is readily demonstrated if the model is run over a 20 year period.

PW07.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

8 Setting up a New Cohort Property

8.1 Introduction to Cohorts

DYMEX has a default set of cohort properties (mortality, numbers, development, reproduction, etc. - see tutorial 1) that have all been used in the Pseudo-wattle model. There are, however, occasions when a cohort property is required which is not included in the default set, and DYMEX contains operational procedures, which allow the user to set up such properties.

This tutorial (and the next two) will develop the model so that it will finally examine the effects on the Pseudo-wattle population when the growth of its members limits the numbers of individuals that can exist in a particular area. To do this, a new cohort property (to be called 'Canopy Area') will be set up and since the model will examine the numbers of individuals in a specific area, the 'Resource' button's function will also eventually be introduced.

In DYMEX, the term 'cohort' has a specific meaning which is repeated here to ensure no confusion results during the application of this tutorial:

'A cohort is a population subgroup of a life stage whose individual organisms all begin that stage of their life cycle at the same time step interval.'

A cohort is the basic unit that is modelled in a DYMEX life cycle. Each cohort consists of a number of individuals, a single individual, or even fractions of an individual. (This last 'unusual' situation is caused by the attributes of a mathematical model: fractional individuals cannot occur in the field but they can appear in models which deal with average populations.) All the individuals of a cohort belong to the same lifestage, occupy the same spatial unit, and share the same properties in common, like the time (day) they entered a lifestage, or the rate of physiological growth. **All the individuals within a cohort are assumed to experience the same conditions during the course of a simulation**. An example of a cohort would be all the seeds germinating on a particular day during the simulation. At any one time during a simulation, each lifestage may contain many cohorts.

Individuals can leave a cohort in a variety of ways: death and migration (e.g. seeds attached to an animal or carried by wind) are two examples, and both will produce a net reduction in cohort numbers. It is possible for either or both of these factors to produce situations in which the number of individuals in the cohort falls to zero: the cohort is then removed from the simulation.

8.2 Multiple Cohorts in a Model

The easiest method of understanding how numbers of cohorts can be present in a very simple model is to consider seed germination in a comparatively short lived species of *Acacia* where the event of a low intensity fire produces maximum germination, for example *A. fimbriata*. Suppose at the end of flowering, a small shrub of *A. fimbriata* produces 5000 seeds of which only 12 seeds survive to germinate because of the effects of various seed mortality factors. Assume the simplest case where a single shrub is present at the January start of the model and: all seeds are produced at the end of a single flowering period during late winter (August), all seeds germinate simultaneously through heat stimulation from a brush fire that just happens to occur each year in the second week of November and the seedlings then mature under identical conditions(soil, climate, etc.); all plants begin seed production at the age of two years, die only of old age and have a life span from seed germination of 8 years. The cohort numbers for three years would resemble the following summary:

Year 1

Jan. ➔ Aug. 1 cohort with a single individual.
Sep.➔ Dec. 2 cohorts; (cohort #1 has a single adult plant; by December, cohort #2 has 12 seedlings).

Year 2

Jan. ➔ Aug. 2 cohorts.
Sep.➔ Dec. 3 cohorts; (cohort #1 has a single adult plant; by December, cohort #2 has 12 juvenile plants and cohort #3 has 12 seedlings.)

Year 3

Jan. ➔ Aug. 3 cohorts.
Sep.➔ Dec. 4 cohorts; (cohort #1 has a single adult plant; by December, cohort #2 has 12 adult plants; cohort #3 has 12 juvenile plants and cohort #4 has 48 seedlings, 12 from each of the four adult plants contained in cohorts #1 & #2).

This process would continue to increase the number of cohorts present until year 8 was reached when the individual of cohort #1 would die. It is left to the user to extrapolate the model to see what the numbers of cohorts will eventually become.

Suppose however, that the artificially contrived November fire that is assumed for the above example occurs on random dates. Further, assume that the fire intensity varies depending upon the amount of litter and local weather conditions and also that the surviving seeds may be more or less buried in the topsoil. Under these more natural conditions, the germination dates of *A. fimbriata* seeds will vary considerably. From any batch of seeds which will form a single cohort, subgroups of seedlings will appear at a variety of times depending upon all the above fire conditions. Each of these new groups will be separated from other groups by intervals of time and will enter the various lifestages at different times. Each group is therefore a cohort and under these conditions, the number of cohorts progressing through the model will be quite large.

Further complexity can be suggested by the fact that each plant cohort may be spread over some considerable area and therefore its individuals will experience a range of soil types (and hence nutrient resources), water availability, light intensity, etc. and so have a range of individual physiological development. In its current form, DYMEX cannot model differing rates of individual physiological development within a cohort lifestage - all individuals are assumed to develop at the same physiological rate. This simplifies procedures but places limits on DYMEX's ability to model complex situations in which cohorts are generated by differential physiological development. Instead, DYMEX can simulate this situation by using the stage transfer functions. If a simple step function is used in which all individuals from a single cohort cross to the next lifestage at a physiological age of 1, then a single cohort in the next lifestage will be produced. If a linear above threshold function is used instead, a series of cohorts will appear in the next lifestage, each of which will be separated from the others by time intervals and effectively simulating the varying rates of physiological development of the individuals in the previously single cohort.

8.3 Default Cohort Properties and Other Required Properties

A cohort property is a variable that each individual in the cohort shares. DYMEX has a number of default cohort properties which can be accessed and applied by the user. By now these properties will be quite familiar to the user and they include the:

- number of individuals in a cohort
- physiological age of individuals in a cohort
- chronological age of individuals in a cohort; and
- density of individuals in a cohort.

These properties are limited in application, and the user may wish to apply other properties to the cohort as it passes through a lifestage. Some additional properties that could be added to a particular model are:

- size of the individual (canopy area)
- sex ratio
- stress; and
- toxin build-up.

It is important to note that the scope of the newly created cohort property must also be considered: it can be local or global. DYMEX modelling defines a cohort property to be **local** if the property variable is reset as the cohort passes from one lifestage to the next; the cohort property is defined to be **global** if the variable is carried over unchanged from one lifestage to the next. As noted already, the user defined cohort property that will be created and applied will be Canopy Area. This property will have no application within the model to be produced in **this** tutorial which is intended only to show how a user defined cohort property is set up; however it will be required in the next tutorial where Canopy Area will be used to show how the Pseudo-wattle population is affected by resource competition from its own members. In addition, the effects will be made to vary according to the age of the cohort.

8.4 Model Parameters for the Canopy Area Cohort Property

Like all plants, Pseudo-wattle has an optimum range of field conditions for which the plant's canopy grows to its maximum. If a plant has all conditions suitable at germination, the growth of its canopy will follow a distinct pattern: slow increase in Canopy Area to begin with, followed by rapid increase during intermediate plant size, followed by slow increase as the mature plant size is approached. In the field this situation is highly modified: perennials do not occur on their own and they compete with each other for resources such as nutrients, water and light. Initially, this model will be simplified to the extent that it ignores such growth patterns and competition so that Canopy Area will be determined purely by chronological age. The rate of Canopy Area increase will be determined by a constant 'function' with a time step increment set by the time taken for a plant to change from a seedling to a flowering adult.

Several assumptions about Pseudo-wattle will now be made. All canopies will be defined as spherical for simplicity so that the area used by the individual is circular. The germinating seedling will be assumed to have a Canopy Area of 1 cm^2 (0.0001 m^2). The starting juvenile canopy will be assumed to be 10 cm in diameter or 0.03142 m^2. The adult Pseudo-wattle when first entering the adult lifestage will have a canopy area of 1.5 m^2 (about 1.4 m diameter), but at the point of flowering will have a 2 m diameter canopy to produce an area of 3.142 m^2. The time taken for a seedling to become a juvenile under ideal conditions is approximately 12 weeks and so the value of the constant area increment per time step will be 0.00261 m^2. [This constant increment value is obtained by subtracting the initial area of the seedling (0.0001 m^2) from the area of the juvenile plant (0.03142 m^2) and then dividing the result by 12.] Similarly, the time taken for a juvenile under ideal conditions to reach the adult size is approximately 48 weeks. This will give a constant area increment per time step of 0.0648 m^2. Assuming that the pseudo-wattle reaches its maximum adult size after two years as an adult, the maximum constant area increment per time step for the adult plant will be 0.0171 m^2. For the model, the consequence will be that the Canopy Area function will develop the size of each one of the Pseudo wattle lifestages, and when this is multiplied by the numbers in a cohort, the result will be the total canopy area produced by the cohort itself.

DYMEX also provides options on how the newly created cohort variable will change in response to its associated process (the **Update** method). Four update methods can be selected: direct and proportional, current value and current average, and these may be direct or inverted. Since Canopy Area is to be modelled, it will be used in the description that follows, however any variable name could be inserted. For a more detailed discussion of the Update methods, please refer to the Builder User's Guide.

Direct accumulation of Canopy Area is modelled by the equation:

$$\text{Canopy Area}_n = \text{Canopy Area}_{n-1} + r_n$$

(where 'r_n' is the value in timestep *n* of the Process associated with the cohort variable.)

This is the non-inverted mathematical model and obviously accumulates Canopy Area with each time step. It is also the default setting used by DYMEX. The user will also easily see that for this simple Pseudo-wattle model, direct accumulation is the obvious choice for calculating Canopy Area of the Pseudo-wattle cohorts.

If the inverted model is used, the equation becomes:

$$\text{Canopy Area}_n = \text{Canopy Area}_{n-1} - r_n$$

and in this case, Canopy Area is decreased by the value of 'r_n' at each time step.

Proportional accumulation of Canopy Area for a non-inverted situation is modelled by the equation:

$$\text{Canopy Area}_n = \text{Canopy Area}_{n-1} * (1 + r_n)$$

Again, with increasing values of 'r_n', Canopy Area increases. For the inverted and therefore decreasing Canopy Area situation, the equation becomes:

$$\text{Canopy Area}_n = \text{Canopy Area}_{n-1} * (1 - r_n)$$

8.5 Building the Model

Start the DYMEX model builder and load the Pseudo-wattle model. Continue with the following steps:

1. Open the **'Lifecycle'** module for editing
2. Select **'Lifecycle'** from the main menu bar and then select **'User Defined Cohort Variables....'** from the drop down menu
3. In the **'User Defined Cohort Properties'** dialogue box, select the **'Add'** button to obtain the **'Cohort Variable'** dialogue box (fig. 8.1)
4. Enter the name **'Canopy Area'** in the **'Name'** text entry box
5. Click on the **'Info'** button where you can describe the new variable in detail
6. For **'Scope'** select the **'Global'** button

Notes:
The Global radio button confirms that the variable is carried over into the next lifestage. The value of the variable however should be re-set to the default of zero as each cohort exits the Adult Plant lifestage; otherwise each new cohort would receive the Canopy Area previously accumulated and Canopy Area would simply increase without limit. The ***'Inverted'*** *and* ***'Use latest inputs'*** *buttons should be left unchecked. The program automatically selects* ***'Direct - Non-inverted'*** *as the default conditions (see previous section 8.4).*

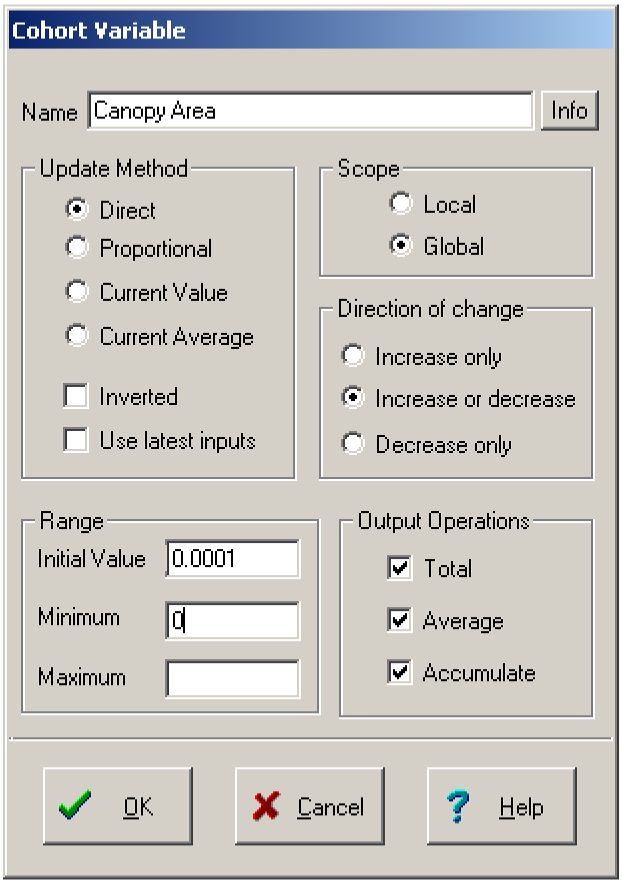


Figure 8.1 Cohort variable properties window

7. For the **'Range',** set the initial value to 0.0001 and the minimum value to 0; the maximum value can be left unset as it is not used in this model
8. **'Direction of Change'** is set to the default of **'Increase or Decrease'**
9. For **'Allowable Operations',** all three should be retained as outputs.

The **'Output Operations'** facility permits the user to decide which operations are best suited to the model. Any combination of the operations can be set and it will be completely dependent upon the requirements of the user as to which operations will finally provide the most useful output. **'Total'** provides the sum of the cohort property for all the cohorts in the particular lifestage that is being addressed at the time. For example, for the Adult Plant lifestage, **'Total'** will construct the total Canopy Area value produced by all the individuals in all the cohorts in the model that are currently present in the Adult Plant lifestage. In the same way, **'Average'** will produce the average Canopy Area for all the individuals in all the cohorts in the Adult Plant lifestage by dividing the total Canopy Area by all the individuals in all the cohorts in the Adult Plant lifestage. **'Accumulate'** has a different application. It displays the total of the cohort property *for each cohort* as it leaves the particular lifestage. **'Accumulate'** will display the total Canopy Area produced by each cohort as it leaves the adult plant lifestage. Since the linear function under which this model operates has an upper limit, each cohort will reach the same limiting area as it leaves the Adult Plant lifestage; also, since each cohort then dies, and the property is reset as the cohort leaves the Adult Plant lifestage, the value will be set back to zero.

The graph of **'Accumulate'** will display a disjoint series showing the appearance and disappearance of cohorts in the model.

10. Select **'OK'** as necessary to return to the **'Life cycle'** window.

As you do this you will be reminded that it is normal to reset a global variable after at least one lifestage.

11. Open the Adult Reproduction process and click on the **'C.V. Transfer**' (cohort variable transfer) button
12. Highlight the **'Canopy Area'** variable, and set the **'Reset Value'** radio button.

The next procedure is to define the processes for the **'Canopy Area'** property in the various lifestages. This can now be done as the user will see that a new button, the **'User-defined Cohort Properties'** button has now appeared on each lifestage icon (figure 8.2).

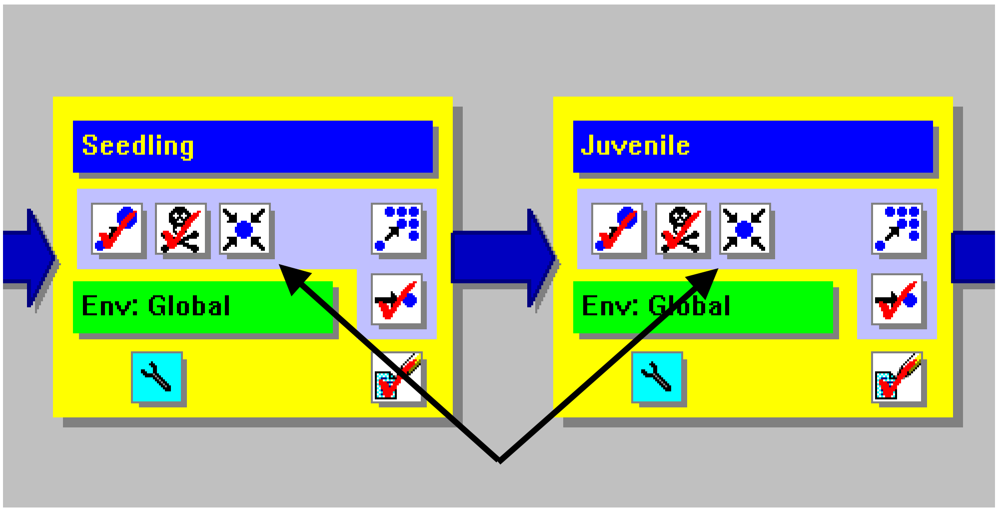


Figure 8.2 Lifestage icons with User-defined Cohort Properties buttons (marked with arrows).

13. Select the **'User-defined Cohort Properties'** button in the **Adult Plant** lifestage.
14. Select the **'Canopy Area'** process, and click on the **'Continuous'** process button to open the **'Canopy Area'** process window.
15. Name the process *'Canopy Growth'*
16. Select the **'Parameter'** button and open its edit window
17. Rename the constant suitably (e.g. *'Canopy Area Increment'*)
18. Set the default value for to 0.0171, the lower limit to 0 and the upper limit to 0.1
19. Repeat the above steps for the **'Juvenile'** and **'Seedling'** lifestages, setting all lower limit values to 0 and upper limit values to 0.1 - the default values are as follows:
    Seedling default value: 0.00261
    Juvenile default value: 0.0648
20. Ensure that all canopy area increments are suitably re-named
21. Exit back to the **'Life cycle'** window by selecting **'OK'** as necessary

22. Select the **'Adult Plant Lifestage Outputs'** button and ensure that **'Total Canopy Area', 'Accumulated Canopy Area'** and **'Average Canopy Area'** are all selected as outputs and renamed as *'Adult: Total Canopy Area'*, ...etc.
23. Repeat step 22 for the **'Juvenile'** and **'Seedling'** lifestages
24. Save the model.

8.6 Running the Model

Run the model for a period of 10 years. A logarithmic chart of the total Canopy Area for Seedling, Juvenile and Adult plants should resemble figure 8.3.

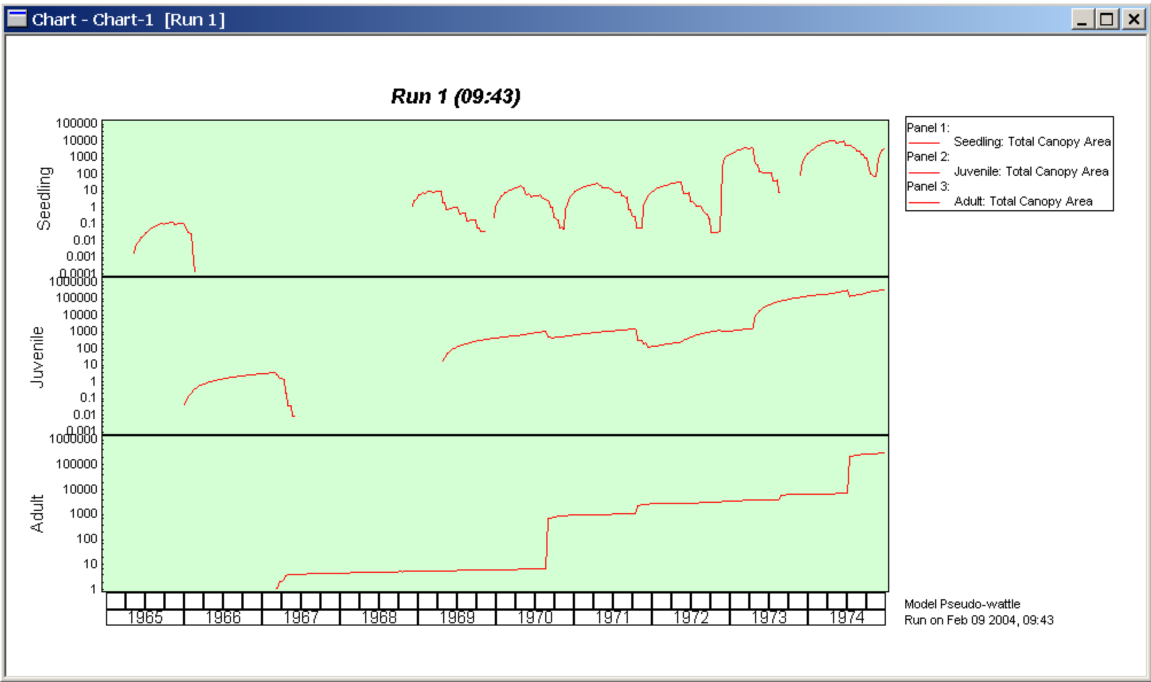


Figure 8.3 Total Canopy Areas for each of the lifestages of the Pseudo-wattle population over a 10 year period.

PW08.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

9 Resource Areas and Density-Dependent Mortality

9.1 Introduction

The user-defined cohort variable, Canopy Area, was set up in tutorial 8 to demonstrate how such a procedure is implemented. To calculate values of population density for the Pseudo-wattle, actual areas must be used and so a new module defining the area included in the simulation (the resource area) must be inserted in the model. Once inserted, the this area can be used to calculate population densities.

Resource Area
The area selected for this model is a 31.6 x 31.6 metre quadrat or 1000 square metres but the dimensions may be any that are suitable to the user. (*Note*: when inserting large numbers such as 10,000 neither the 'comma' nor spaces indicating the 'thousands' position should be entered.)

Adult Density Dependent Mortality
Once a resource area is defined, functions can be set up that restrict the whole population of the organism being modelled to the dimensions of that area. For example, the adult pseudo-wattle has a canopy diameter of 2 metres and an area of 3.142 m^2. Because the resource area is 1000 m^2, it forms a square of side 31.6 metres. This infers that if the plants were spaced in an array so that the circular canopies just touched, it would be possible to fit about 15 x 15 plants in the resource area or 225 plants. Suppose instead that the plants had 'flexible' canopies and fitted into the resource area so that it was all used; this results in about 318 plants. It is possible in the field that some overlapping of canopies can take place and under these conditions it might be possible that about 600-700 plants could 'squeeze' into the resource area, especially if some of the population was not fully grown. This situation implies a density dependent mortality function of linear above threshold form, in which the threshold is about 225 adult plants. Assuming that 100% mortality occurs when the number of adult plants reaches 600, then the slope is the reciprocal of the difference between 600 and 225 (375) or 0.003. Some flexibility must be employed when using this slope. Obviously, there would not be 100% mortality in the field when the plants reached this maximum value and it would be more reliable to suggest a slope of somewhere between 0.0005 and 0.003 and provide upper and lower limits to allow testing of various mortality models. This model will use a slope of 0.001 and upper and lower limits of 0 and 1.

Juvenile Establishment Mortality - Canopy/Resource Area Dependent
A second function can be introduced as a direct result of canopy establishment by adult plants. The increasingly closed canopy has a direct impact on seedling-juvenile establishment through competition for light and nutrients. Most, if not all, germinating seedlings will die and be unable to transfer to the juvenile lifestage. This is easily modelled in DYMEX by adding a mortality rate function which combines resource area with total adult canopy area.

As the Pseudo-wattle population number increases, the density of plants per square metre of the model also increases and as a consequence, the canopy area of the plants increases. A DYMEX module could be developed to provide an output designated 'Sum of All Canopy Areas' which would include all members of all the cohorts in all respective lifestages and sum their combined canopy areas. As this summed area increased, it would suppress the development of the seedlings and small juveniles.

This simpler model combines both the Adult Total Canopy Area and the Resource Area to give a value of an Adult Plant Population Dependent Mortality Rate which can then be inserted in the Juvenile establishment mortality function. The assumption will be made that as the Adult Total Canopy Area increases, the resulting competition for resources - particularly light and moisture - will produce considerable establishment mortality amongst the seedling population as it attempts to transfer to the juvenile lifestage. Although an apparently simple method, its underlying operations are actually quite complex as every lifestage function contributes to the value that is being fed back to the establishment mortality.

Although only the Adult Plant canopy area is being used here, the user will readily perceive that other and more complex methods could be used. For example, it would be possible to insert extra modules to sum the canopy areas of seedling, juvenile and adult lifestages or juvenile and adult stages and use the sums as inputs into Mortality Rate modules to control separate establishment mortalities for each lifestage. Because only the seedling to juvenile establishment mortality will be subjected to the alteration, it is emphasised that the model presented here is simplistic, and designed only to show the principles under which DYMEX can operate.

The underlying mathematical concept for the adult population dependent mortality rate is defined by the following equation:

$$\begin{array}{r}\text{Area Dependent}\\ \text{Mortality Rate}\end{array} = \frac{\text{Adult Total Canopy Area}}{\text{Sampling Area}}$$

Examination of this equation will show that at first, with a low value of Adult Canopy Area, (e.g. 20 square metres) and a Sampling Area value of 1000 square metres, the resultant value of the division will be less than 1 and so the value of the Area Dependent Mortality Rate will be quite small. As the adult population increases, the rate of mortality increases towards the value of 1, until the situation is reached where the adult canopy area is equal to the resource area; at this point, seedling establishment mortality will become equal to one and no seedlings will survive to become juveniles. This situation will remain until the members of an adult cohort die and free up resource area for use by the establishing juveniles. Since the adults will still be producing seeds, there will be (when the adults finally die) for a short time, a very large increase in the number of seedlings and juveniles until the canopy mortality re-establishes itself. Over time, the number of adult plants will remain more or less constant and the numbers of seeds, seedlings and juveniles will also fluctuate in between more or less fixed values.

To apply the above mortality rate equation, several steps are involved. First an expression module must be inserted which will provide a value for the quotient of the adult canopy area and the resource area. Next, this Area Dependent Mortality Rate must be inserted into an appropriate function which can be used to define this additional mortality in the Adult Plant lifestage. The simplest functional form which can apply the rate immediately to the establishment mortality procedures is the direct function however for this model the function will be a linear above threshold so that it will be possible to alter the rate of mortality.

9.2 Building the Model

Open the DYMEX Model Builder and load the Pseudo-wattle model; then complete the following steps:

1. In the **'Model'** window, select **'Model'** on the main menu bar
2. Add a new **'Query User'** module and call it **'Model Sampling Area'**
3. Select the **'Outputs'** button and open the **'Output Variables'** dialogue box
4. Select the **'New'** button: a newly created variable will appear in the list box
5. Select the **'Select'** button - the **'+>'** symbol appears beside the new variable
6. Re-name the variable suitably (e.g., 'Sampling Area')
7. Set the default to 1000, the lower limit to 100 and the upper limit to 10000
8. Return to the **'Module'** window and set the sort order so that the **'Model Sampling Area'** module is placed just below the **'Latitude'** QueryUser module
9. Save the model.

DYMEX must next be instructed to produce a plant density output. The model has a defined area in which to operate, but lacks instructions on what to do with it. Steps 10-15 provide this information.

10. Return to the **'Lifecycle'** window and in the **Adult Plant** lifestage complete the following steps
11. Select the **'Lifestage Properties'** button in the Adult stage to open the **'Adult – Lifestage Properties'** dialogue
12. In the **'Lifestage Resource Variable'** panel, click on the small button to the right of the edit box, scroll down the list and select **'Sampling Area'**
13. Close the dialogue and return to the **'Lifecycle'** window
14. Ensure that **'Average Density'** is selected as an output and renamed as **'Adult: Average Density'**
15. Repeat the procedure of steps 11-14 for **Seedlings** and **Juveniles**
16. Return to the **'Model'** window and save the model.

The next procedure will be to add an expression module which will combine the adult total canopy area with the resource/growth area of the model to provide the canopy cover proportion to be inserted into the juvenile plant lifestage establishment mortality:

1. Add an **'Expression'** module using the main menu bar and its drop down menu
2. Rename the module **'Area Dependent Mortality Rate'**
3. Select the **'Inputs'** button
4. Add two extra inputs by selecting the **'Add Extra Input'** button twice
5. Link the first variable to **'Adult Total Canopy Area'**
6. Link the second variable to **'Sampling Area'** and before exiting, **place a tick in the selection box marked 'Invert (1/x)'**

{Step 6 will ensure that the Resource Area value is actually dividing the Adult Lifestage Total Canopy Area when the two values are multiplied together. DYMEX uses this procedure to change a multiplication into a division.}

7. Select **'OK'** to return to the module's window and then select the **'Outputs'** button
8. Select the **'Select'** button to highlight the output variable and rename it suitably (e.g. 'Canopy Cover Proportion')
9. Return to the module's window and then select the **'Settings'** button
10. Set the function to **'Product'** and then return to the module's window
11. Change the module's sort order so that it is just above the **'Lifecycle'** module
12. Return to the main model window; the new module will be above the **'Lifecycle'** module and there will now be a tick beside the new expression module.

The previous operations have set up the first part of the process to produce population dependent mortality and the output of the last expression produces the independent variable for the linear above threshold function that will produce Population Dependent Mortality in the Adult Plants. The next procedure is to set up the linear above threshold function in the Juvenile Plant lifestage's establishment mortality.

1. Open the lifecycle window and select the Juvenile Plant's **Mortality** button
2. Select the **'Establishment'** button
3. Name the process **'Establishment Mortality'**
4. Select **'Function'** to set up a new function and then select **'Linear Above Threshold'** as the function
5. Set the Independent variable to **'Canopy Cover Proportion'**
6. Rename the function **'Density Dependence'**
7. Select the **'Parameters'** button and set:
   (a)Threshold default to 0, lower and upper limits to -1 and 1 respectively
   (b)Slope default to 1, lower and upper limits to 0 and 2 respectively
8. Rename the parameters suitably
9. Select **'OK'** and return to the **Lifecycle** window
10. Save the model.

The final procedure is to insert density dependent mortality in the Adult plant lifestage.

1. Select the **'Mortality'** button of the **Adult** lifestage
2. Select the **'Continuous'** button
3. Select the **'Function'** button to add a new function
4. Re-name it **'Density-dependence'**
5. Select **'Adult: Total Number'** as the Independent Variable
6. Select **'Linear Above Threshold'** as the function
7. Select the **'Parameters'** button
8. Set the threshold default value to 225, the upper limit to 500, the lower to 0
9. Set the slope default to 0.001, the upper limit to 0.01, the lower limit to 0
10. Rename the parameters suitably
11. During the exit to the model window check that the combination rule is set to complement-product

12. Save the model.

9.3 Running the Model

Run the model for 20 years. A chart output for the total numbers of the lifestages should be similar to figures 9.1 and 9.2.

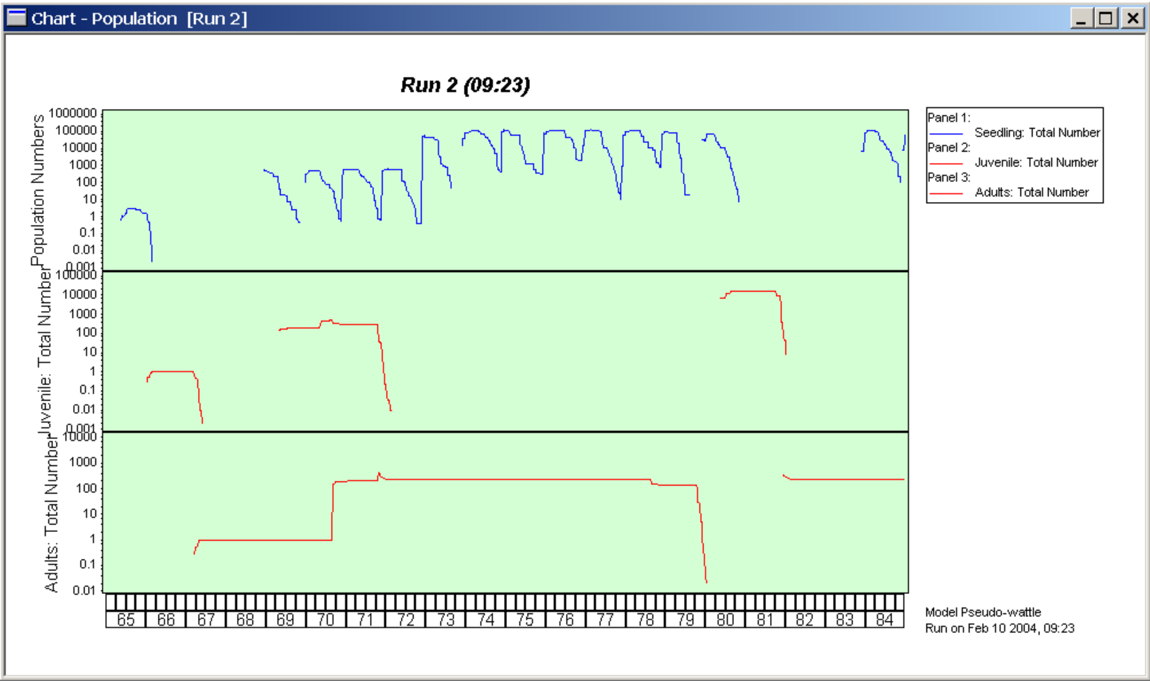


Figure 9.1 Total numbers for Pseudo-wattle lifestages 20 year run - logarithmic output.

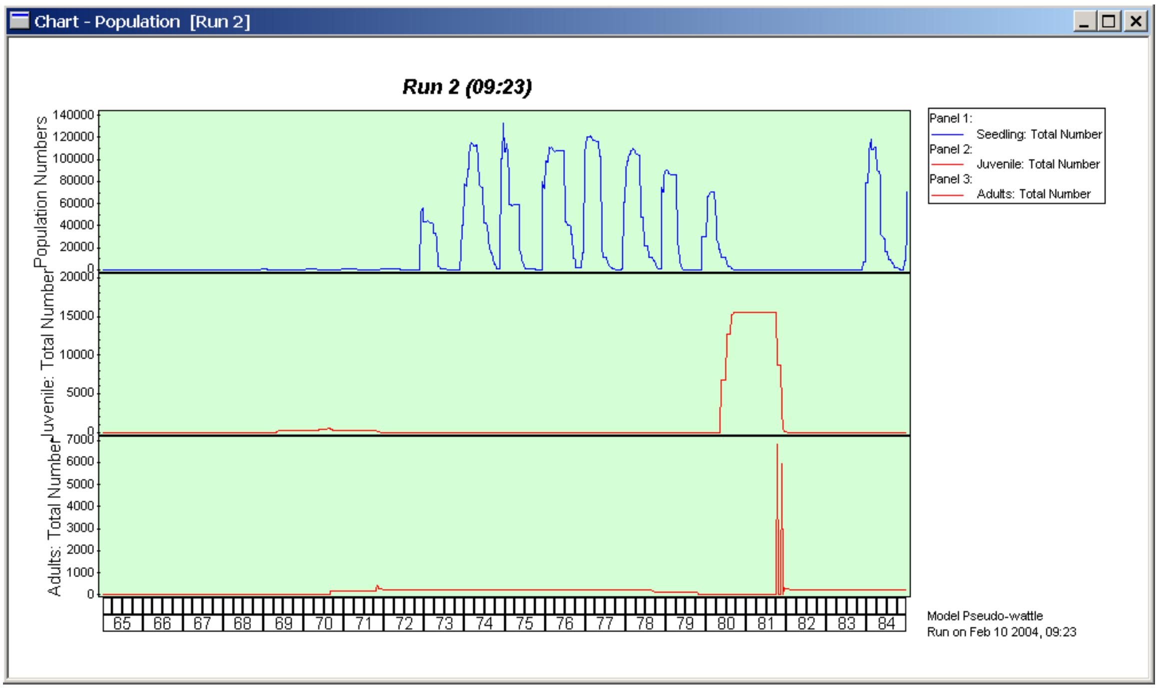


Figure 9.2 Total numbers for Pseudo-wattle lifestages 20 year run, natural numbers display.

These graphs indicate that the population of pseudo-wattle is now stable. Figure 9.1 clearly shows the suppression of transition of seedlings to juveniles until some of the adult population dies. In addition, the self suppression of the adult stage is seen by the flattening of the adult plant graph which reaches a maximum of between 200 and 400 plants during the years 1971-1979. The 'spikes' that appear in figure 9.2 are caused by the surge in juveniles that transfer to the adult stage before the establishment mortality function completely operates and the huge increase causes an equally large population dependency mortality to again restore the adult population to its stable value of about 220 individuals.

If canopy areas are charted, the results will resemble figure 9.3 .

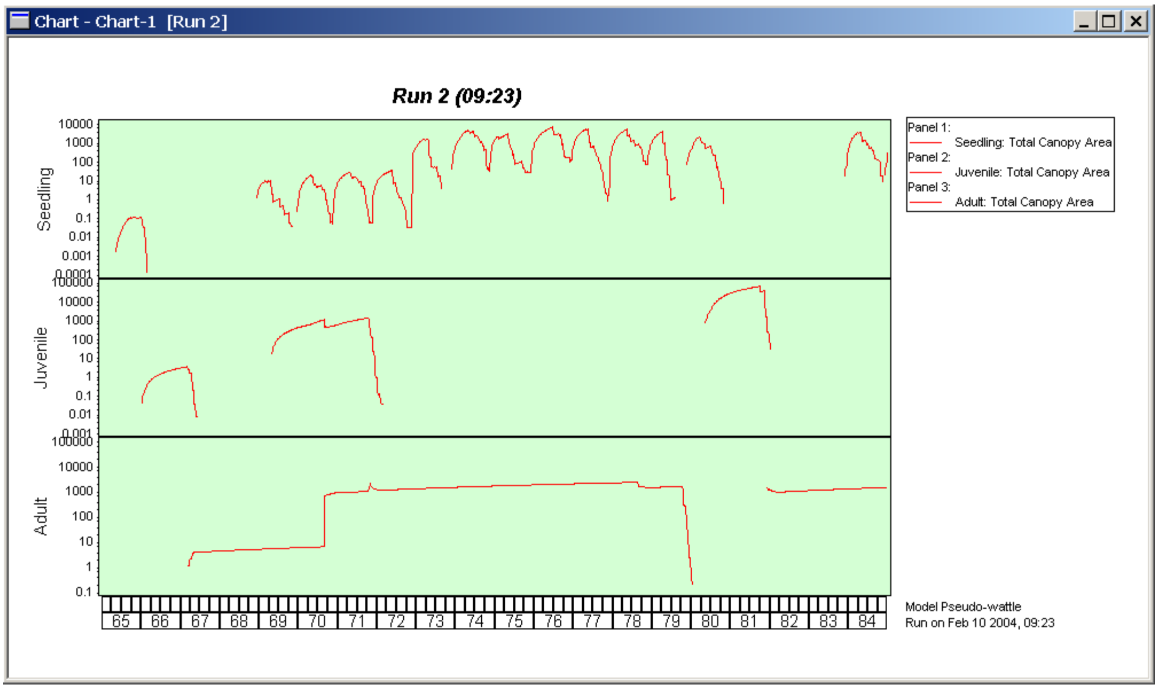


Figure 9.3 Canopy areas for 'above ground' lifestages of Pseudo-wattle 20 year run; (logarithmic output).

These results are very similar in shape to figure 9.1 as could be expected as the numbers are referenced to the sampling area.

> ***PW09.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial***

10 Flowering and Seed Dormancy

10.1 Introduction

In this and the next two tutorial we will add two major features to the model that will make it a much better approximation to the field situation. Currently, all our adult plants flower at a particular time of year and seeds are then produced immediately. Obviously this is far from realistic. Flowering tends to be a cyclic activity usually occuring in response to environmental cues such as daylength or temperature. This ensures that seeds are produced at a time of the year when environmental conditions are favourable. Thus the time of flowering will most likely be different at different places. A second that we have ignored so far is that that when flowering does occur, seeds need some time to develop on the parent plant, during which time they are wholly dependent on the parent. In other words, the immature seeds or seed pods are not independent stages. We shall use a DYMEX component termed an Endostage to model this. Then, in the next tutorial we will deal with more complex seed dynamics, including seed dormancy.

Figure 10.1 shows the lifecycle as it will look at the end of Tutorial 12. In the current tutorial, we will add the two stages representing the flowers (ovules) and immature seeds. Then, in Tutorial 11, we will change the current reproduction functions to more the more realistic ones outlined in the next paragraph. Finally, in Tutorial 12, the current Seed stage will be split into two stages to model seed dormancy.

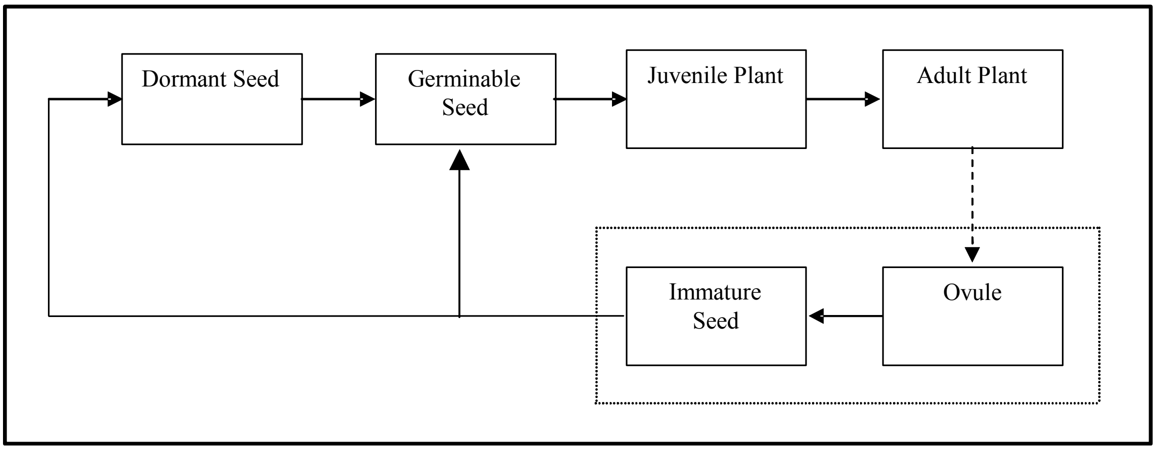


Figure 10.1 Diagram of plant lifecycle including flowering and dormancy cycling.

Flowering Phenology
The pseudo-wattle flowers in starts flowering in early spring, when the daylength exceeds 11.5 hours. Flowers are produced over a period of about 5 weeks. The total number of flowers (and hence seeds) produced is determined by the size of the adult plant (i.e., by its canopy area) in a quadratic relationship, i.e., seeds = k*(canopy area)^2. A maximum of about 100 000 seeds are produced by a plant with canopy area of 15 m^2 under ideal conditions, giving a value for k of around 450.

Flower and Seed Development

Flowers (ovules) become immature seeds after 2 weeks. The immature seeds then develop in response to temperatures above 15°C, taking 440 degree-days for full development to viable seeds. Also, we know that about 25 % of flowers do not produce seeds (due to factors such as lack of pollination).

10.2 Adding the Endostages

1. Open the model in the Builder, then open the Lifecycle window.
2. Use the **Reproduction** button on the **Adult** stage and select **New Stage** from the **Destination Stage** scrolling list (we don't want the reproductive process to link this lifestage to either of the existing stages).
3. Name the new stage **Ovule**.
4. Use the **Stage Transfer** button to create a new stage after the **Ovule** stage (call it **Immature Seed**).
5. To link the **Immature Seed** stage to the existing **Seed** stage, highlight the **Immature Seed** lifestage, then in the main Menu click on **Lifecycle** and select **Create new Stage Link…** Select **Seed** as the **Destination Lifestage**, and select **OK** to return to the Lifecycle window.

The **Ovule** and **Immature Seeds** lifestages have the special property that they are wholly dependent upon the adult plants for their continued survival, though the reverse is obviously not true. Lifestages with this special property are called ***Endostages***.

6. To define both the **Ovule** and **Immature Seed** stages as endostages, click on their **Lifestage Settings** buttons (bottom left hand corner of the lifestage panels), and select **Adult Plant** as the **Container Lifestage** at the bottom of the dialogue box that appears. Note that the name panels for these stages change from blue to grey, indicating that they are endostages.
7. Because both of these stages are endostages, it is no little sense to initialise a simulation run with either of these stages, as they cannot exist without an adult plant present. Thus, via **Lifestage Settings** buttons, click on the **User initialisation permitted** button to disable this function.
8. Save the model

Now we need to specify when ***Ovules*** become ***Immature Seeds***, and also add the loss of 25% of ovules as mortality.

9. Click on the **Stage Transfer** button in the Ovule stage and name the process *'Ovule Development'*.
10. Click on the Function button to add a function component.
11. Select ***Chronological Age*** as the independent variable and the ***Step*** function.
12. Set the "***p1: Threshold***" parameter to 14 (range 7-21) and the "***p2: Step***" parameter to 1.
13. Exit back to the Lifecycle window and click on the **Mortality** button in the Ovule Stage.
14. Select the '**Exit**' button, as we want the 25% mortality to happen only once,

when Ovules become Immature Seeds

15. Name the process, say, *'Exit Mortality*, and add a **Parameter** component
16. Name the parameter *'Ovule Mortality'*, giving it a value of .25 (range 0 to 0.5)
17. Return to the Lifecycle window
18. Click on the '**Development**' button in the 'Immature Seed' stage, and name the process *'Immature Seed Development'*
19. Select the Function button to add a function component
20. Select '**Daily Temperature Cycle**' as the independent variable and the **Linear above Threshold** as function template
21. Set the '***p1: Threshold***' parameter to 15 (with an allowed range of 10 to 20) and name it '*Development Threshold Temperature*'
22. Set the '***p2: Slope***' parameter to 7, the same as the model timestep (with no variation allowed in that parameter) and name it '*Degree-day accumulation Scale Factor*'
23. Return to the Lifecycle Window
24. Click on the **Stage Transfer** button in the Immature Seed stage and name the process *'Seed Maturity'*.
25. Click on the Function button to add a function component.
26. Select ***Physiological Age*** as the independent variable and the ***Step*** function.
27. Set the "***p1: Threshold***" parameter to 440 (range 400-500) and name it '*Degree-days for development'*
28. Set the "***p2: Step***" parameter to 1.0 (allow no variation)
29. Return to the Lifecycle window
30. Click on the **Output** icon in the Ovule stage
31. Select (and rename) '**Total Number**'. Do the same for the 'Immature Seed' stage
32. Return to the Lifecycle window and save the model

10.3 Running the Model

Initialize the model with a single adult plant and run it for 1000 days. Chart the total numbers of Ovules, Immature Seeds, Seeds and Juveniles in separate panels. Figure 10.2 shows what this chart should look like. The plant flowers for the first time in its second year (it has not accumulated enough development units until then).

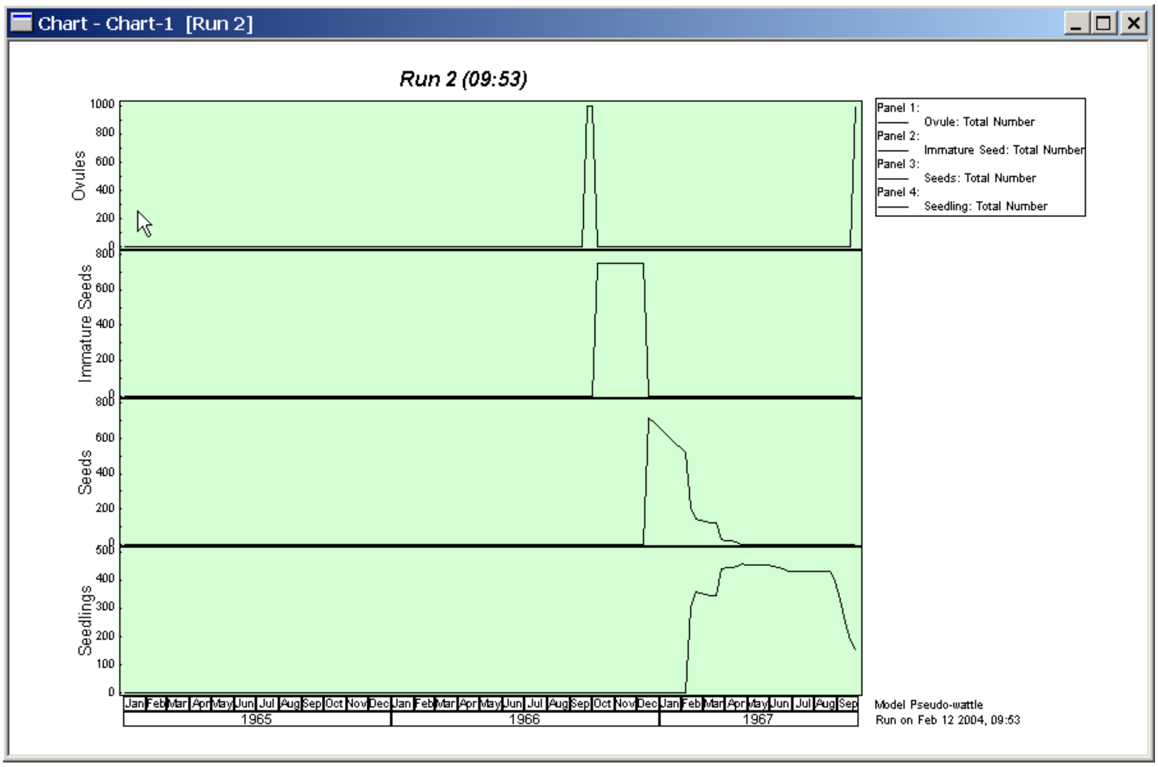


Figure 10.2 Total numbers of Ovules, Immature Seeds, Seeds and Seedlings.

> ***PW10.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial***

11 Flowering Phenology

11.1 Introduction

Having added the two endostages in the previous tutorial, the next step is to modify the reproduction processes to reflect the known plant characteristics described under the heading 'Flowering Phenology' in the previous tutorial. Currently we have set a lifetime fecundity of 10000 seeds, with flowering taking place at the same rate on a fixed time each year. In this updated version of the model, the fecundity will be determined each year by the size of the plant (its Canopy Area).

It is worthwhile reiterating the role of the three processes that determine reproduction. A pre-defined cohort property named '***Residual Fecundity***' determines how many individuals the cohort is currently capable of producing via the Progeny Production process. You may think of this as a store of potential new individuals (Figure 11.1). During any time step, the maximum number of individuals that can be produced (via the Progeny Production process) is the current value of the Residual Fecundity. The actual number of progeny produced then reduces the Residual Fecundity by that number.

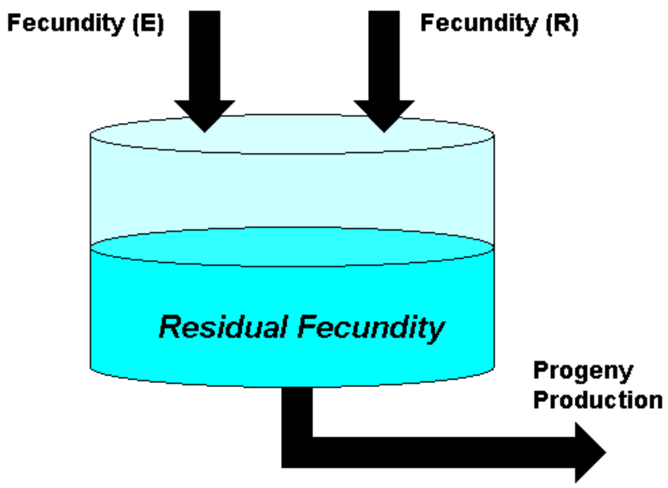


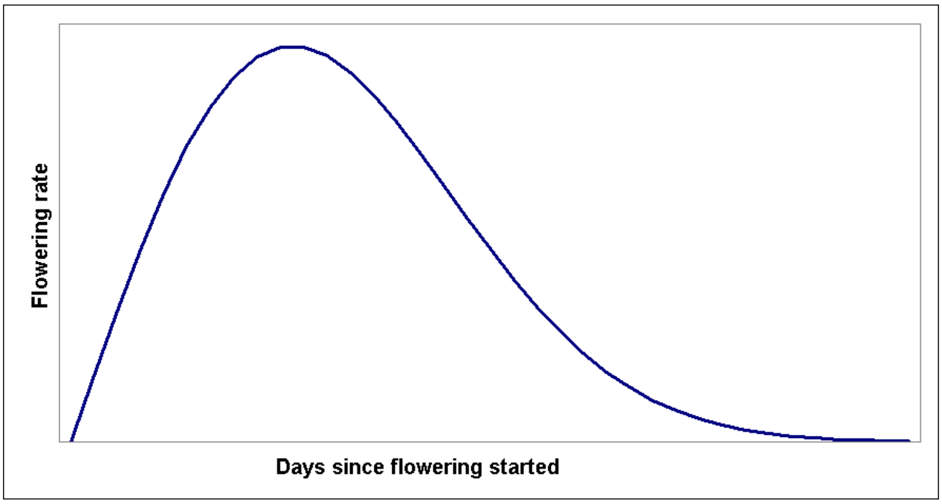
Figure 11.1 Fecundity, Residual Fecundity and Progeny Production.

The two Fecundity processes increase the Residual Fecundity. The **Fecundity (E)** process is applied just once, at the time when a new cohort of a reproductive stage is created (it is an "establishment" process). The '**Fecundity (R)**' (the R is for recharge) is applied at each timestep. Any or all of these three processes can be used to achieve the desired reproductive rates in the model. For example, in an insect model it is common to ignore **Fecundity (R)** completely, to use **Fecundity (E)** to set the maximum number of progeny that the insect can produce (often based on the size it achieves at the end of its immature phase) and to use **Progeny Production** to deplete the Residual Fecundity over the adult's lifetime. In a plant model, Fecundity (E) is usually not used; Fecundity (R) is generally used to reset the annual reproductive capacity (based on the current size of the plant), and Progeny Production determines the flowering phenology.

This is the approach that will be taken with the Pseudo-Wattle model.

One of the variables we will need to drive the reproduction is an indicator of daylength change. This is available from the **Daylength** module. We will then add a module whose output variable indicates the period of time when the day length exceeds 11.5 hours and is increasing. Let us also assume that over the 5-week flowering period, flowering rate to a maximum rate after about 2 weeks, and then decreases over the next 3 weeks, with giving a flowering rate relationship similar to that in Figure 11.2. Note that the y-axis here indicates the number of new flowers produced each week, so that the area bounded by the curve and the x-axis will be equal to F_t.

Figure 11.2 Fecundity, Residual Fecundity and Progeny Production.



To implement this relationship, we need a variable that represents the days since flowering started. Even though this variable could be created via a module outside the lifecycle, we will create this variable as a Cohort Property. This has the advantage that if we later find that the onset of flowering is in some way dependent on a characteristic of the adult plant (for example, its size), a Cohort Property would be able to handle that situation while a global variable could not.

From the description of flowering phenology in the last tutorial, the total number of flowers produced, F_t, is a quadratic function of Canopy Area, A.
I.e.,

$$F_t = kA^2,$$

where k is a constant with a value of 450.

1. Open the ***Daylength*** module dialogue and select the **Outputs** button.
2. Highlight *'Day Length Change'* by clicking on it and then select it as an output variable.
3. Add a new module of type **Equation** to the model, giving it the name '*Spring*'
4. Click on **Inputs** in the module dialogue and link ***Variable1*** to '*Day Length'* and ***Variable2*** to '*Day Length Change'*. Leave all other inputs unused.
5. Return to the module dialog, select the **Outputs** button and then select the

module output variable and rename it to '*Spring Indicator*'

6. Click **OK**, then select the **Settings** button
7. Type in the equation "***if([x1]>11.5 & [x2]>0, 1)***". This is DYMEX syntax for "if the first variable (x1 = Day Length) is greater than 11.5 AND the second variable (x2 = Day Length Change) is greater than 0, then set the output's value to 1, otherwise set it to 0"
8. Click **OK**, and in the module dialogue, set the **Sort Order** to 32, so that the module is placed immediately after the ***Daylength*** module

We shall use the first week in this "spring" period to capture the fecundity and then arrange for the flowering to occur over the following time steps. To help in this, we will create a variable that we can use to trigger the fecundity calculation and a second variable that specifies the extent of the flowering period.

9. Add a new module of type **Difference (t)** to the model, giving it the name '*Spring Transition*'
10. Click on **Inputs** in the module dialogue and link the input variable to '*Spring Indicator'*
11. Return to the module dialog, select the **Outputs** button and then select the module output variable and rename it to '*Spring Transition*'
12. Click **OK**, and set the **Sort Order** to 34, so that the module is placed immediately after the ***Spring*** module
13. Add a new module of type **Equation** to the model, giving it the name '*Flowering Period*'
14. Click on **Inputs** in the module dialogue and link ***Variable1*** to '*Spring Indicator'* and ***Variable2*** to '*Spring Transition*. Leave all other inputs unused.
15. Return to the module dialog, select the **Outputs** button and then select the module output variable and rename it to '*Flowering Period*'
16. Click **OK**, then select the **Settings** button
17. Type in the equation "***if([x1]>0&[x2]=0, 1)***"
18. Click **OK**, and in the module dialogue, set the **Sort Order** to 36, so that the module is placed immediately after the ***Spring Transition*** module
19. Return to the Model Component window and save the model.

At this point, you may want to run the model for a few years and plot out the new variables we have added to see that they are operating correctly (as in Figure 10.3). It is always a good idea to check any additions made to a model carefully before proceeding to the next step. Note how the Spring Transition variable becomes 1 for just one time step during the year, with the Flowering Period becoming 1 the time step after. The negative "spike" in the Spring Transition variable indicates the end of the "spring" period, but will not be used in this model.

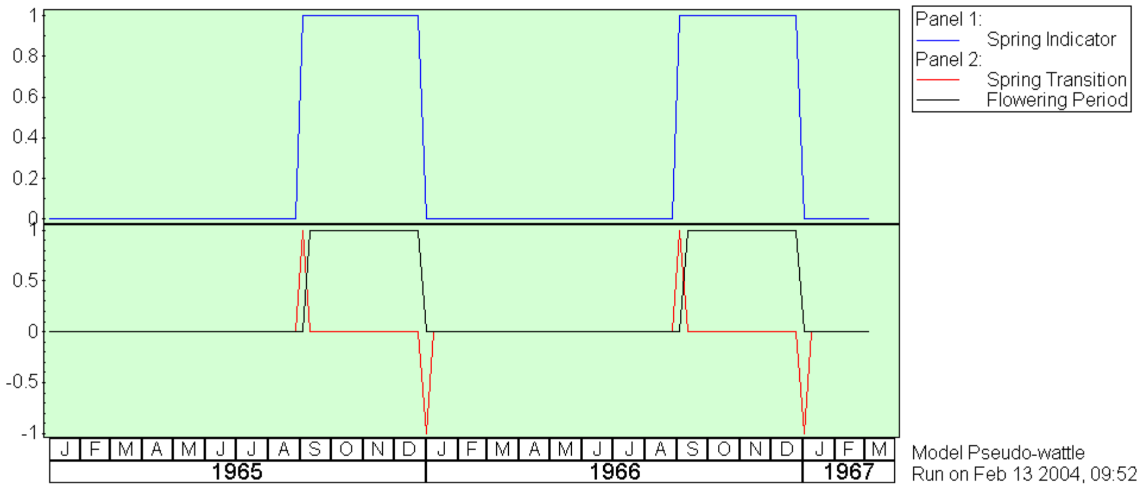


Figure 11.3 The variables indicating spring and flowering period in a 2-year simulation.

Now we can add a Cohort Property that counts the days since the beginning of spring to the lifecycle. Since this variable will set and used wholly within the Adult stage, it should be made Local. Note that the variable needs to be reset sometime before the next spring, when the count must start again from 0.

1. In the **Lifecycle** window, select the **Lifecycle** menu item and then **User-defined Cohort Variables...**
2. Select **Add** to create the new Cohort Variable
3. Name the variable '*Weeks since First Flower*'
4. Click on Info and add a suitable description that explains the function of this variable
5. Make sure **Direct**, **Local** and **Increase or Decrease** are selected, set the **Initial Value** and **Minimum** to 0, and select only '**Average**' as a valid output operation
6. Return to the Lifecycle window
7. Select the **User-defined Process** button in the Adult stage and highlight '*Weeks since First Flower*' in the list of variables
8. Click on the Continuous button to open the process dialogue, name the process '*Flowering Week Counter*'
9. Click on **Function** to add a function component, select '*Flowering Period'* as the independent variable and '***Direct***' as the Function Template
10. Return to the process dialogue, click on **Function** to add another function component, select '*Weeks since First Flower*' as the independent variable and '***Direct***' as the Function Template
11. Click on **OK**, this will issue a reminder to set the Combination Rule – click **OK** and set the Combination Rule to '***User-defined***', typing in the following as the rule to use: "***ife([r1]>0, 1, -[r2])***"
12. Exit to the Lifecycle window and save the model

Note that the Combination Rule specified is saying, "If it is during the flowering period (i.e., r1 is greater than 0), add 1 to the *Weeks since First Flower*; otherwise subtract its own value (r2 is the *Weeks since First Flower*)". The second part of the rule has the effect of forcing the variable to 0 outside the defined flowering period.

13. In the Lifecycle window, click on the **Reproduction** button on the **Adult** stage and then select the **Fecundity (E)** button.
14. In the Fecundity process dialogue, make sure the component is highlighted (there should be just one), and remove it by clicking on the **Delete Component** button.
15. Close the dialogue and then click on the **Fecundity (R)** button in the Reproduction dialogue.
16. Name the process **Fecundity Set**
17. Click on the **Function** button to add a Function process component
18. Select the **Quadratic above Threshold** function from the list of function types and set '*Canopy Area'* as the independent variable

The ***Quadratic above Threshold*** function has the equation $y = p_1 + p_3(x\text{-}p_2)^2$. If the parameters p_1 and p_2 are set to 0, this reduces to the quadratic relationship $y = p_3x^2$ that is required.

19. Select the parameter ***'p1: Constant'*** and set its default, minimum and maximum to 0.
20. Select the parameter ***'p2: Threshold'*** and set its default, minimum and maximum to 0.
21. Select the parameter ***'p3: Slope'*** and set its default, minimum and maximum to 450, 200 and 600, respectively. Rename the parameter to '*Fecundity vs Canopy Area Multiplier'*
22. Return to the process dialogue
23. Click on the **Function** button to add a new function component
24. In the Function dialogue, select *Spring Transition* as the independent variable and ***Direct*** as the function template
25. Return to the process dialogue and set the following **Combination Rule**: "***if([r2]>0, [r1])***". This will ensure that the Fecundity is set only once, just before the flowering period
26. Return to the Lifecycle window and save the model

Now all that remains to do is to spread the actual flowering across a 5 week period, in accord with the distribution shown in Figure 11.2. One way to specify the Progeny Production process rate, r_{pp}, is as follows:

$$r_{pp} = f(t) \times F_r$$

where F_r is the ***Residual Fecundity*** and $f(t)$ is a function of the number of time steps since flowering (t). This is just saying that at any time after flowering commences, the number of flowers produced per week is a proportion ($f(t)$) of the flowers remaining to be produced (F_r). We then need a function, f(t), that will give the curve shown in Figure 11.2 as the Progeny Production rate. It so happens that if we set $f(t) = kt$, where k is a slope constant, we will obtain the required relationship.

27. In the Lifecycle window, click on the **Reproduction** button on the **Adult** stage and then select the **Progeny Production** button.

28. Delete both process components by highlighting each in turn and then selecting **Delete Component**
29. Rename the process to '*Flowering*'
30. Click on the **Function** button to add a Function process component
31. Select the **Linear** function from the list of function types and set '*Weeks since First Flower'* as the independent variable
32. Select the parameter ***'p1: Threshold'*** and set its default, minimum and maximum to 0.
33. Select the parameter ***'p2: Slope'*** and set its default, minimum and maximum to 0.15, 0 and 0.5, respectively. Rename the parameter to '*Flowering Rate Slope'*
34. Return to the process dialogue and click on the **Function** button to add a second process component
35. Select the **Direct** function from the list of function types and set '*Residual Fecundity*' as the independent variable
36. Make sure that the Product Combination Rule is selected
37. Return to the Lifecycle window and save the model.

As a final step, let us add an output variable that will report the rate of flower production, so that we can examine and fine-tune this process.

33. Click on the **Output** icon in the Ovule stage
34. Select the output 'Recruitment via Stage Transfer' and rename it to '**Flower Production Rate**'.
35. Return to the Lifecycle window and save the model

11.2 Running the Model

Initialize the model with a single adult plant and run it for 2 years (730 days). Plot the Flower Production rate to give a chart similar to Figure 11.4. Note flower production is indeed giving a distribution quite similar to that in Figure 11.2. If more detailed data on flower production was available, the model could be run with a range of values for the 'Flowering Rate Slope' parameter to obtain the best fit to the data.

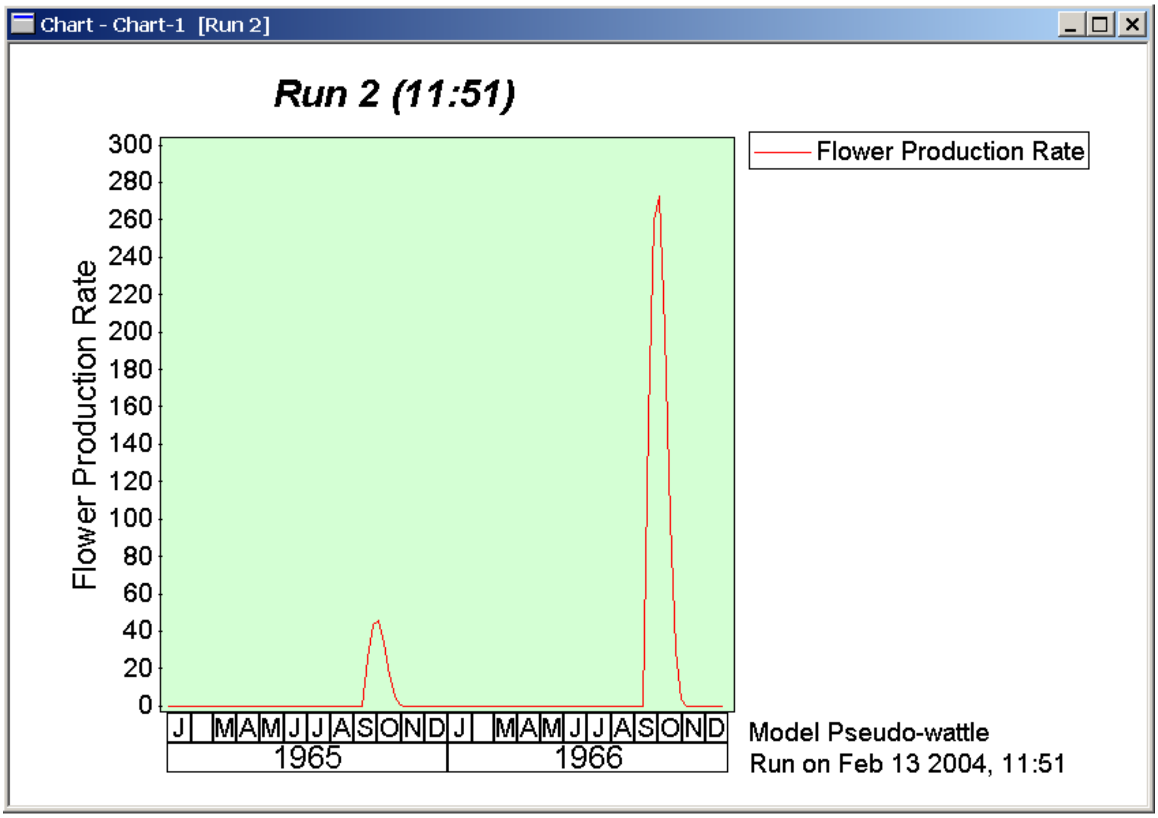


Figure 11.4 The variables indicating spring and flowering period in a 2-year simulation.

> ***PW11.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial***

12 Seed Dormancy and Mortality (continued)

12.1 Dormancy

In most plant species, seeds undergo a period of dormancy, during which they will not germinate even if the conditions that normally induce germination are present. This is termed seed dormancy. We treated the seed stage in Tutorials 1 and 2, and simulated dormancy by building in a fixed time interval during which seeds could not germinate even if the germination conditions are present. In this tutorial, we will deal with the dormancy issue in a more flexible way, by adding a dormant seed stage to our Pseudo-wattle model to simulate this phenomenon (see Figure 10.1). We will assume that a proportion (80%) of seeds are dormant when they become mature. These dormant seeds require an extended period of low temperature to break this dormancy. Consider the data in the table below:

Minimum Temperature	Dormancy terminated (weeks)
12	Not terminated
8	10
4	5
0	3.3

Lets say we place four sets of dormant seeds into an incubator in which the temperatures are set to fluctuate in a daily sinusoidal cycle. The first column in the table specifies the minimum daily temperatures set for each incubator, while the second column gives the observed time taken for the dormancy to be broken. In this (admittedly contrived) example, a plot of the <u>inverse</u> of the second column (which is just the weekly rate of dormancy breakage) against Minimum Temperature just happens to result in a linear relationship given by the equation

$$y = 0.3 - 0.025T$$

Assuming we can treat dormancy breakage in the same manner as development (i.e., by accumulating a weekly rate whose total indicates the current state of dormancy), we can use the Physiological Age as an indicator of dormancy state (when 0, no dormancy breakage has occurred, when 1 the seed has broken dormancy). Dormancy breakage itself is then specified as the Transfer process, which will consider individuals ready to move into the Germinable Seed stage when their Physiological Age is 1. We will assume that 80% of those seeds that have broken dormancy will become ready to germinate in any one week.

12.2 Mortality

Currently, we have a constant mortality rate in the seed stage that results in about 10% mortality in a year (see Section 2.3). We will keep this mortality for the Germinable Seed stage, but assume that the mortality rate in the Dormant Seed stage is <u>only half that value</u>.

1. Open the model's Lifecycle window
2. Select the ***Seed*** stage, click on the **Lifecycle** menu and select the **Add Stage After…** item
3. Make sure that "Seed -> NEW STAGE -> Seedling" is selected as the layout, click on OK to insert the new stage
4. Rename the current Seed stage to '*Dormant Seed*'

5. Rename the new stage to '*Germinable Seed*'
6. Select **Create new Stage Link** from the Lifecycle menu
7. In the **Create New Stagelink** dialogue, select '*Immature Seed*' as the **Source Lifestage** and '*Germinable Seed*' as the **Destination Lifestage**
8. Click OK to return to the Lifecycle window

The lifecycle should now appear as in Figure 12.1 (which corresponds to the diagram shown in Figure 10.1). There are now two "exits" from the Immature Seed stage, and the corresponding transfer processes must be specified. Seeds still mature on the tree in the same way as before, but now 80% of them start out dormant, with the rest being germinable. To put this into the model:

9. Click on the **Stage Transfer** button in the ***Immature Seed*** stage
10. From the "drop-down" menu, select the ***Dormant Seed*** stage
11. Rename the process to '*Seed Maturity (Dormant)*'
12. Add a new process component as a **Parameter**, call it '*Proportion Dormant*', and set its value to 0.8 (range 0.5 to 1.0)
13. Click **OK** to return to the **Process** dialogue; make sure the **Combination Rule** is set to ***Product***
14. Highlight the ***Physiological Age*** component in the list box and click the **Copy** button
15. Return to the Lifecycle window, click on the **Stage Transfer** button in the ***Immature Seed*** stage and select the ***Germinable Seed*** stage from the "drop-down" menu,
16. Name the process '*Seed Maturity (Germinable)*'
17. Select **Paste** to insert a copy of the ***Physiological Age*** dependent factor
18. Add new process component as a **Parameter**, call it '*Proportion Germinable'*, and set its value to 0.2 (range 0 to 0.5)
19. Click **OK** to return to the **Process** dialogue; make sure the **Combination Rule** is set to ***Product***

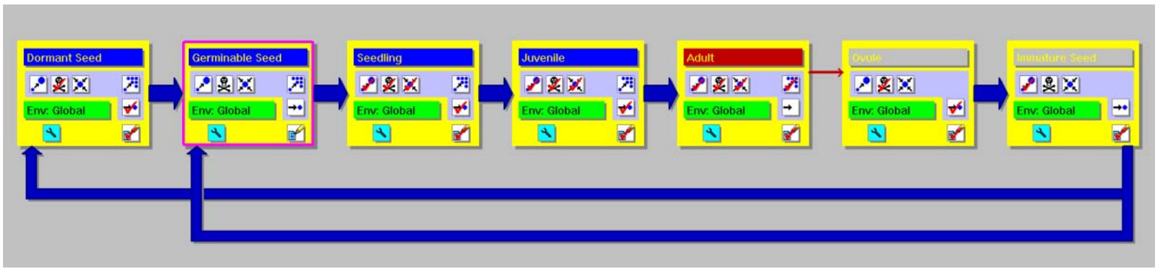


Figure 12.1 The lifecycle with dormant and germinable seed stages

The Immature Seed should now have the red 'tick' that indicates it is fully set. Note that the two parameters we have added (*Proportion Dormant* and *Proportion Germinable*) should add up to 1, so that the development of immature seeds proceeds as before. To ensure this, it would have been better to create this as a parameter (p) external to the lifecycle module (using perhaps a function module) and then use them as p and (1-p) in the two transfer processes. This is left as an exercise for the reader.

As we have added the new stage after the existing Seed stage of the previous version of the model, the germination process needs to be moved from the Dormant Seed to the Germinable Seed stage. This is most easily accomplished using the Cut and Paste buttons.

20. Click on the **Stage Transfer** button in the ***Dormant Seed*** stage
21. Highlight the first component (*Germination Function*) and select **Cut**
22. Click **OK**, then select the **Stage Transfer** button in the ***Germinable Seed*** stage and select **Paste**
23. Name the process '*Germination*'
24. Return to the ***Dormant Seed*** transfer process, highlight the remaining process component and select **Cut**
25. Click **OK**, then select the **Stage Transfer** button in the ***Germinable Seed*** stage and select **Paste**
26. Make sure the *Product* combination rule is selected and return to the Lifecycle window

Now we can implement the dormancy breakage relationships. As indicated in the introduction, we will use the Physiological Age variable (i.e., the development process) to implement the transition from dormant to germinable.

27. Click on the **Development** button in the ***Dormant Seed*** life stage
28. Call the process '*Dormancy Removal*'
29. Click on the **Function** button to add a function component
30. Select *Minimum Temperature* as the independent variable and ***Linear*** as the function shape
31. Highlight the first parameter (*p1: Y-axis intercept*), and set its default to 0.3 (range 0– 1)
32. Highlight the second parameter (*p2: Slope*), and set its default to –0.025 (range –0.1 – 0)
33. Return to the **Function** window, select the **Advanced** button and set the **Low Limit** to 0
34. Return to the Lifecycle window and select the Stage Transfer button
35. Rename the process to '*Dormancy Breaking*'
36. Add a component as a **Function**, select *Physiological Age* as its independent variable and ***Step*** as it function shape
37. Highlight the first parameter (*p1: Threshold*), and set its default, minimum and maximum values all to 1
38. Highlight the second parameter (*p2: Slope*), and set its default to 0.8 (range 0.5 – 1)
39. Return to the Lifecycle window

Now we will adjust the mortality rates for these two seed stages.

40. Click on the **Mortality** button in the ***Dormant Seed*** life stage and then select **Continuous**
41. Click on **Copy** to make a copy of the mortality component
42. Select Edit Component and change the default value to 0.022

43. Exit to the **Lifecycle** window, click on the **Mortality** button in the ***Germinable Seed*** stage, then Continuous and select **Paste** in the process dialogue
44. Name the process '*Seed Mortality*' and return to the Lifecycle window

All that remains now is to rename the "Total Number" output variable from the dormant seed stage and create such an output variable for germinable seeds.

45. Select the **Lifestage Outputs** button in the ***Dormant Seed*** stage
46. Making sure the '*Seeds: Total Number*' output variable is highlighted, click the **Rename** button and name the variable '*Dormant Seed: Total Number*' (also change the description if it is now not appropriate any more)
47. Click **OK**, then select the **Lifestage Outputs** button in the ***Germinable Seed*** stage
48. Highlight '***Total Number***', click on **Select** and rename the variable to '*Germinable Seed: Total Number*'
49. Return to the Lifecycle window and save the model.

12.3 Running the Model

Initialize the model with a single adult plant and run it for 10 years. Plot Adult, Juvenile and seed numbers in a chart such as that shown in Figure 12.2. Note the way that the dormant seeds become germinable towards the end of the winter, after they have spent the required amount of time at cool temperatures. The peaks in the Adult population are due to the density-dependent mortality that was added to the model in Tutorial 9. That function is not at all realistic, but making it more realistic is left as an exercise.

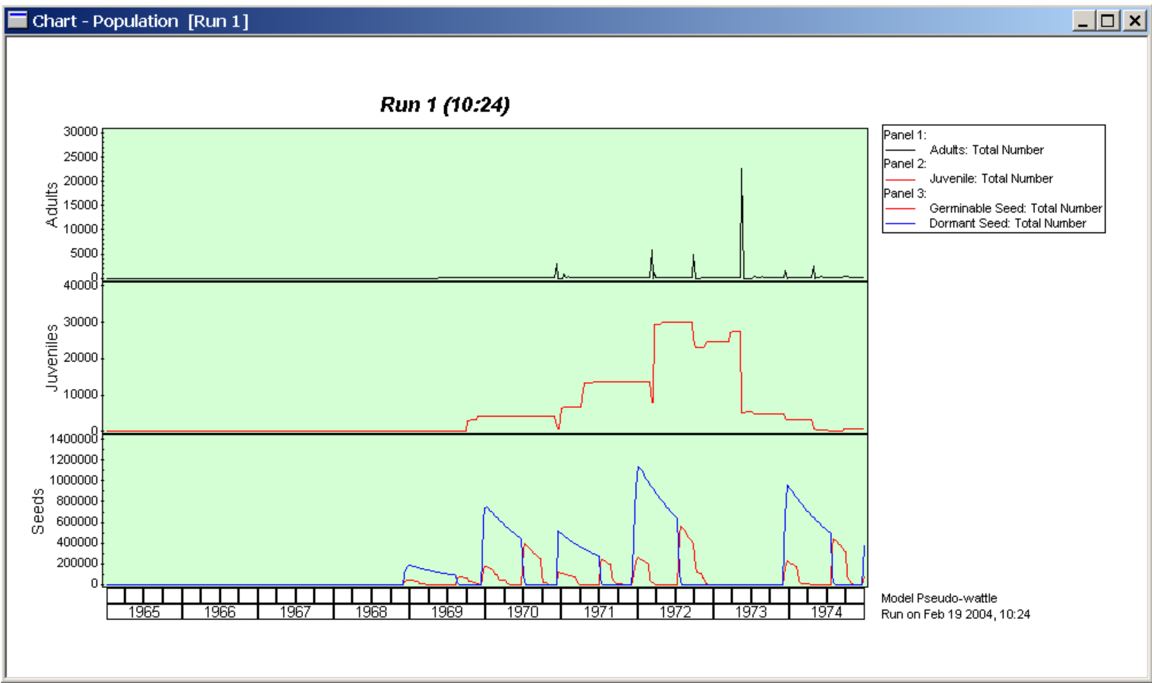


Figure 12.2 The lifecycle with dormant and germinable seed stages

PW12.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

13 Adding an 'Event' Module

13.1 Introduction

Mortality in the Pseudo-wattle population model is affected by several variables which are dependent upon either the age or the size of the population. Other aspects of mortality remain to be modelled amongst which are: the effects of too much or too little rainfall, herbivore destruction and parasite attack. Human induced mortality can also be added to the model. Assuming that Pseudo-wattle is an agricultural pest which competes efficiently for crop or pasturage space and additionally has little value as a fodder for stock, an agriculturalist's problem resolves itself into either reducing population numbers to acceptable levels or (preferably) eradicating the Pseudo-wattle population completely. To add this operation to the model, an 'Event' module is used.

An ***event*** is a particular occurrence which affects the lifecycle of the population and it can be a natural occurrence or human induced. Examples of events are the application of a spray, a fire, ploughing, heavy rainfall, or sudden loss of food. How DYMEX is used to model the event depends completely on how the user wishes to apply it.

13.2 Modelling an Event

13.2.1 The 'Event' Module

The Event module has several inputs, and for the Pseudo-wattle model there is a single output which is coupled to the Lifecycle module (Figure 13.1). The Timer module produces two of the inputs: *Day of Year* and *Simulation Date*. The *Day of Year* is a day count for the current year of the model run. For example, if the model was run over ten years, the day of year count would go through ten cycles of 1-365, not a single cycle of 0-3650 days. The Simulation Date is the calendar date of the particular day of the model run. The Threshold value is the 'trigger' for the Event module if it is desired to link the module to events determined by the changes produced in an actual run. The user has complete flexibility in choosing what will be the 'trigger'. For example, it could be rainfall, temperature, the number of individuals in the population, or the number of individuals present in a host population.

One simple plant control method is the use of fire. Plants vary markedly in their sensitivity to fire: some withstand intense fires or even 'fire storms' where the forest or woodland canopy ignites, while others are badly affected or even killed by very low intensity fires at ground level. The size of the plant also determines some degree of fire resistance - a ground dwelling herb is less likely to survive a low intensity fire than a 10 metre tree. Additionally, there are many species in which the application of fire is a necessity for successful reproduction. Generally, buried seeds are unaffected but for many species of the genus *Acacia*, the application of fire is a germination stimulant.

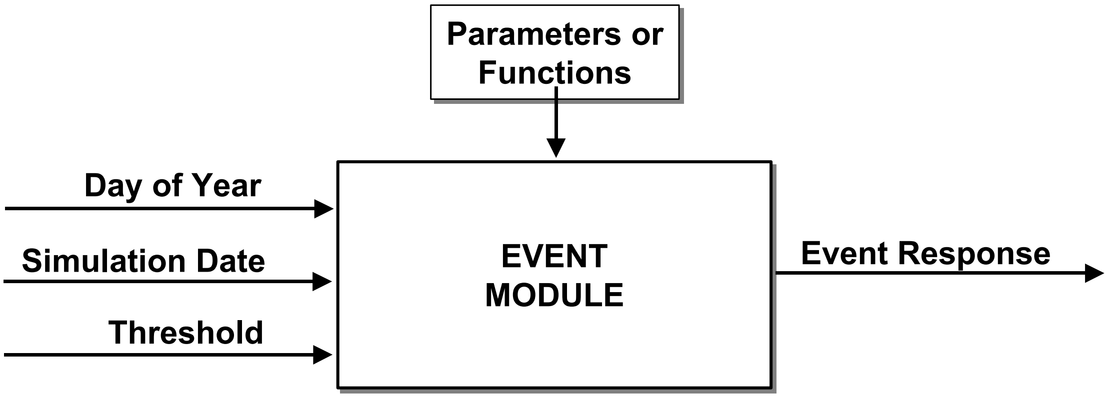


Figure 13.1 The 'Event' Module

Many plant species show increasing resistance to fire with maturity and so an accurate model of a Pseudo-wattle population subjected to a low intensity, ground level fire would have high mortality levels in population members with low physiological maturity but reduced levels of mortality in the physiologically mature population; seed mortality would be insignificant. A set of parameter values for Pseudo-wattle a fire dependent mortality event might then be:

Lifestage	Mortality Rate
Seed	1 %
Seedling	95.00%
Juvenile	80.00%
Adult	40.00%

The extremely low mortality in the seed lifestage is considered to be sufficient to cover those seeds that remain on the surface and whose seed coats are sufficiently thin to permit seed destruction in a 'cool' fire. Seedling mortality is likely to be very high. This is because a low intensity 'cool' fire travels slowly across the litter on the soil and while it does not harm mature or semi-mature plants, it is often very destructive on small, establishing plants. The tiny seedling population that survives is likely to be individuals which by random occurrence are on bare patches of ground and thereby shielded from the fire. (Other explanations will no doubt suggest themselves.) Juveniles contain all those individuals which are just larger than seedlings or just smaller than adult plants. Again, the survival rate is linked to the physiological development of the juvenile - older plants will survive better than younger plants. To adequately model this situation, the juvenile lifestage might be split into a number of lifestages, each of which has a separate level of fire sensitivity, however for this model a more simplistic system will be used. It will be assumed that on average, 80% of juveniles are killed in a fire. The adult population is much less likely to be affected by a low intensity fire and the 40% level suggested here indicates that fire is not a particularly good method of control for the adult plants.

Fire application is somewhat different from that of a spray in which the effects diminish over time. As suggested by the table of mortality rates above, in a fire the plant is either killed or it is not and therefore the simplest model of fire mortality is the step function. The 'threshold' will be the time of the event and the step height will be the percentage mortality

13.3 Forming the Model

13.3.1 Changing the Lifecycle Module

A number of alterations are required so that the Lifecycle module reacts to the Event module. First, an 'Event' module (figure 13.2) has to be built so that it can supply the event action variables to the Lifecycle module; and second, mortality functions that respond to the event output need to be set in each affected lifestage.

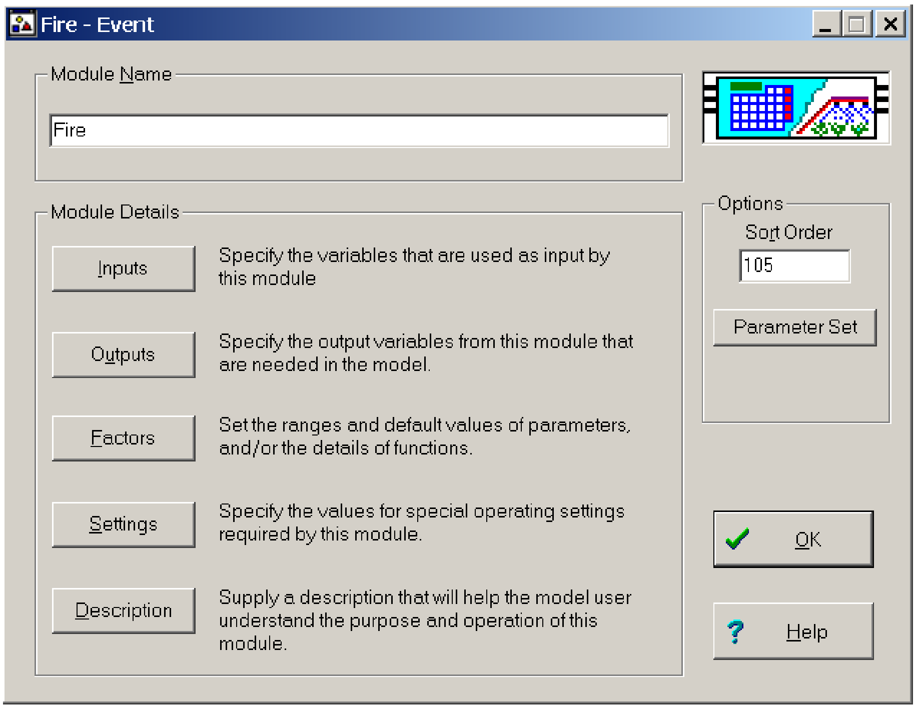


Figure 13.2 The 'Event' window - sort order has been set to place this module before the Lifecycle module in order of operation.

1. Start the Model Builder and open the Pseudo-wattle file
2. Select **'Model'** from the menu-bar and add an **'Event'** module
3. Open the **'Event'** module for editing (Figure 13.2)
4. Re-name the module to '*Fire*'
5. Select the **'Inputs'** button and obtain the link window (Figure 13.3)

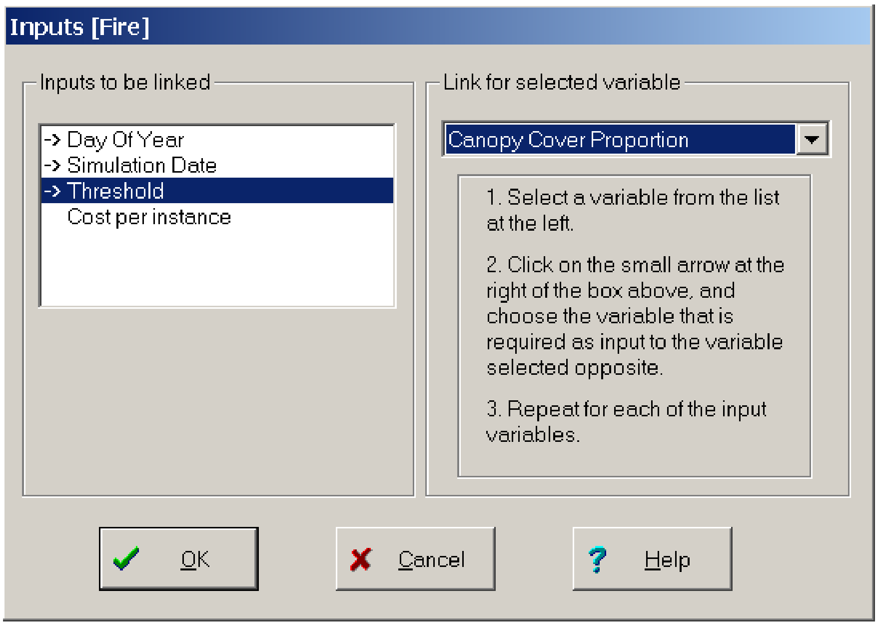


Figure 13.3 Event 'Link' window

6. Link the input *'Day of Year'* with the same name for the selected variable and then repeat this for *'Simulation Date'*
7. Link **'Threshold'** with *'Canopy Cover Proportion'*
8. Return to the **'Event'** module window
9. Select the **'Outputs'** button to obtain the **'Output Variables'** dialogue box
10. Highlight **'Event Variable',** select it and rename it *'Fire Effect'*
11. Return to the **'Event'** module window
12. Select the **'Factors'** button to open its window (Figure 13.4)

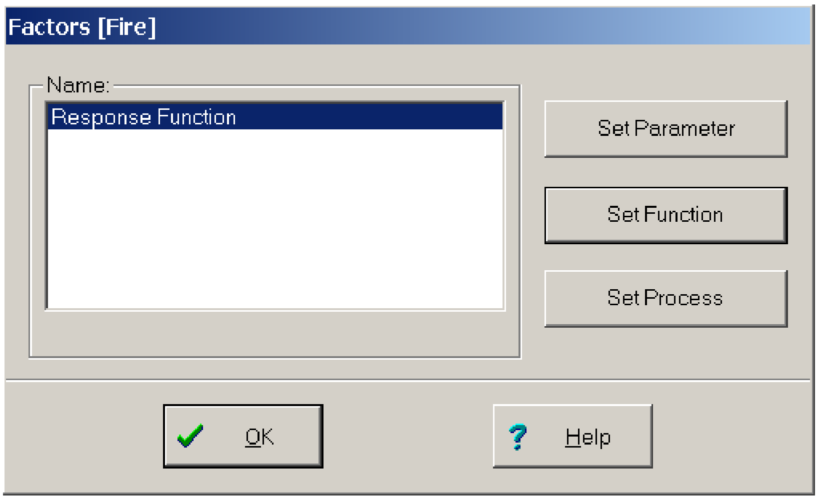


Figure 13.4 Factors window

13. Select the **'Set Parameter'** button to open its dialogue
14. Set the default to 1.0 and the lower and upper limits to 0 and 1.0, respectively
15. Return to the **'Event'** module window
16. Select the **'Settings'** button to obtain the Event Settings dialogue, shown in Figure 13.5

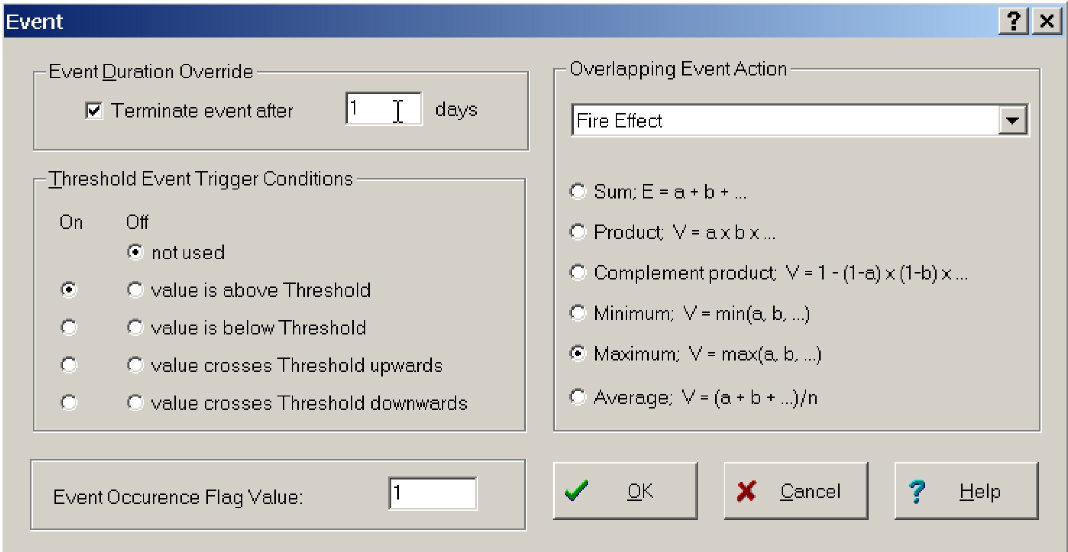


Figure 13.The Event Settings dialogue

17. Select the **'Terminate event after'** button and then insert **'1'** for the number of days
18. Ensure that the default setting of **'value is above Threshold'** is selected for the **On** threshold setting - this means that a fire can be triggered by the event threshold variable (*Canopy Cover Proportion*) exceeding a set limit
19. Nothing needs to be done for the **Overlapping Event Action**, as any fire will only last for one day
17. Exit to the **'Model'** window and save the model.

The Lifecycle module must now be altered to set mortalities in response to output from the Event module. A step function will be used in each 'above ground' lifestage to introduce the relevant mortality into the model.

1. Open the **'Lifecycle'** module for editing
2. Select the ***Dormant Seed*** lifestage's **'Mortality'** button, and then the **Continuous** mortality
3. Select the **'Function'** button to add a new process component
4. In the **'Function'** dialogue box, select the ***Direct*** function shape and the '*Fire Effect*' as independent variable
5. Return to the process dialogue and select the **Parameter** button to add a new process component
    a. Name the parameter '*Fire Susceptibility*'
    b. Set its default to 0.01 (allowed range 0 – 0.1)

6. Return to the process dialog and click on the **Change** button in the **Combination Rule** panel
7. Select User-defined and type in the following combination rule:

1-(1-[r1])*(1-[r2]*[r3])

Note that this is still a Complement-Product combination rule, but the second mortality is a product of factors [r2] (*Fire Effect*) and [r3] (*Fire Susceptibility*)

8. Repeat the sequence given in steps 2-7 for each of the Germinable Seed, Seedling Juvenile, and Adult lifestages, but set the default parameter values to 0.01, 0.95, 0.8 and 0.4, respectively (choosing appropriate allowed ranges for each of these parameters). Note, however, that the Seedling and Adult stages already have two mortality factors, hence the new factors end up as [r3] and [r4], giving a combination rule of **1-(1-[r1])*(1-[r2])*(1-[r3]*[r4])**
9. Save the model.

13.4 Running the Model

Low intensity, ground layer fires are very common in pastoral 'silvi-cultural' management where annual burning of open woodlands or pastures is used to stimulate new growth of grass. Each lifestage of the Pseudo-wattle population has a different reaction to fire mortality. Let us apply a fire in early spring, say September 3. Here we are using a 'trigger' for firing that depends on a date which may be determined on observations of plants in the field or my be determined by some other factor (such as availability of personnel or low risk of the fire escaping control). Another option for an event trigger is to start the event as soon as the farmer is visually 'conscious' that there is a problem. This type of trigger can be linked to the density of the weed plant.

To set a fire on a particular data or day of year, the Fire module needs to be initialized:

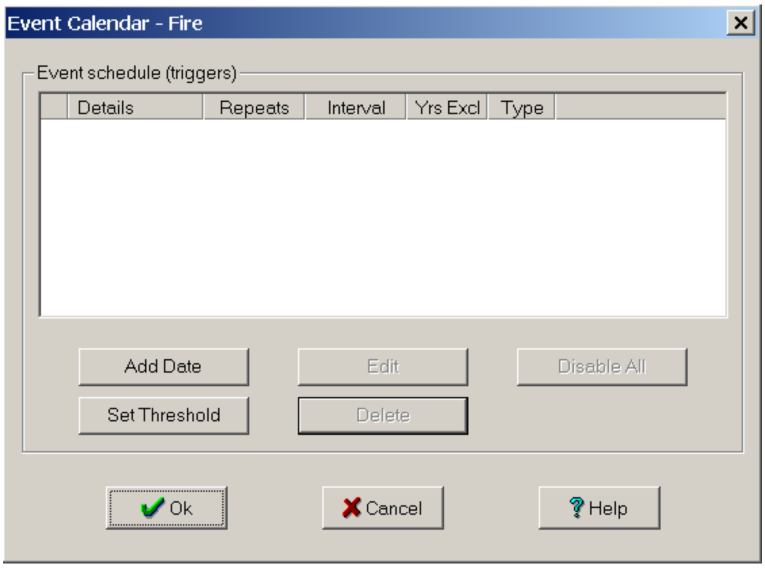


Figure 13.5 Event Calendar window

1. Start the Simulator and load the Pseudo-wattle model
2. Select the **'Fire'** module
3. Select **'Initialise Module'** from the drop-down menu and obtain the **'Event Calendar'** panel (figure 13.5)
4. Click on the **'Add Date'** button to open the **'Specify Event Date/s'** panel (figure 13.6)
5. Use the **'Week of Event'** scroll button to select **'Sep 03, (36)'** and then click **Ok**.
6. Ensure that the **Repetitions** are set to 0, that the event if of type **On** and that **'All Years'** is selected for the run
7. Click **Ok** to return to the **Event Calendar** dialogue
8. The **Event Calendar** now lists a fire event timed to occur on January 22 (each year). Make sure the checkbox in front of the data has a 'tick' in it.

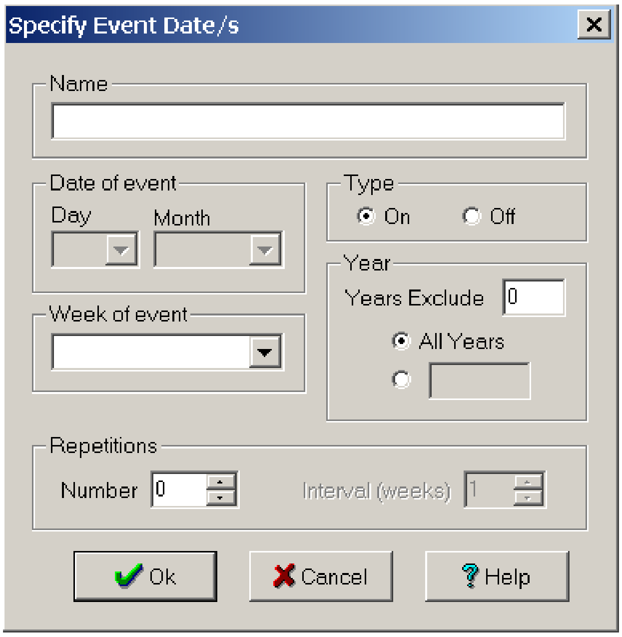


Figure 13.6 Specify Event Date/s Panel

9. Exit to the **'Model'** window and run the model for 10 years.

Once the model is run, a chart of population and canopy cover and should produce a result similar to figure 13.7. The lower panel plots the 'Fire Event' variable, indicating when a fire has occurred.

Figure 13.7 Results from the model with a fire applied on September 3 each year.

The Model Components window with the current settings is shown in Figure 13.8

Figure 13.8 Current Model Structure - Simulator Screen

PW13.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial

14 Finding the Best Time to Fire

The previous tutorial used a set date as the trigger for the fire event. This value was selected as the chart outputs indicated that January 23 was roughly the mid point of the lifestage length of the annual emergence of seedlings and this lifestage is most sensitive to fire. In practice, decisions on firing depend on local weather conditions, but modelling can provide very good indications of when such firing will be most effective. DYMEX can do this by setting up a 'run sequence'.

A 'Run Sequence' is a series of runs of the same event where the time variable is altered a prescribed amount each time. Since Pseudo-wattle populations rise and fall with the seasons, a simple way to find out where a fire would do most good would be to test the fire's effectiveness for each week of the year and calculate the resulting year's end Pseudo-wattle population. This would entail 52 runs of the program with the user then comparing the effects of each week's spray on the total population to find the greatest mortality. The operation could be done manually 52 times by the user, however DYMEX has this procedure built into its software. Because Pseudo-wattle is a perennial, the results of firing will be calculated for the same time in each year over a period of years.

14.1 Setting up a Summary Variable

Before we can get any useful results from running a sequence of simulations, we need a way of classifying the results of each simulation within the sequence. In other words, we need to be able to say that a fire on the 22nd January is better than a fire on the 5th of June. To do this, we create a variable (or set of variables) that summarizes some aspect of the simulation results into a number, which we can then use to compare the results of different simulations. Such variables are termed *Summary Variables*. In our case, we might judge the success of a fire by the reduction in the Canopy Cover of the wattle that we obtain. So we want to create a Summary Variable that gives the average Canopy Cover over the last year of the simulation.

1. Start the Builder and open the Pseudo-wattle file
2. Select **'Model'** from the menu bar followed by **'Summary Variables...'** to obtain the Summary Variables list window (figure 14.1)
3. Click the **Add** button to add a new Summary Variable
4. In the Edit Summary Variable dialog, use the scroll button on the **'Variable to summarize'** list, find '*Canopy Cover Proportion*' and select it
5. Name the Summary Variable '*Canopy Cover*'
6. Select the **Average** from the **Summary Variable Statistic** panel and then **1 year (365 days)** from the choice of **Summary Period**. This will cause the Summary variable to report the average Canopy Cover Proportion over the last year of the simulation.
7. Supply a **Description** and **Mnemonic** for this summary variable if desired.
8. Click **Ok**, and the new Summary Variable will be shown in the list box
9. Return to the model window and save the model

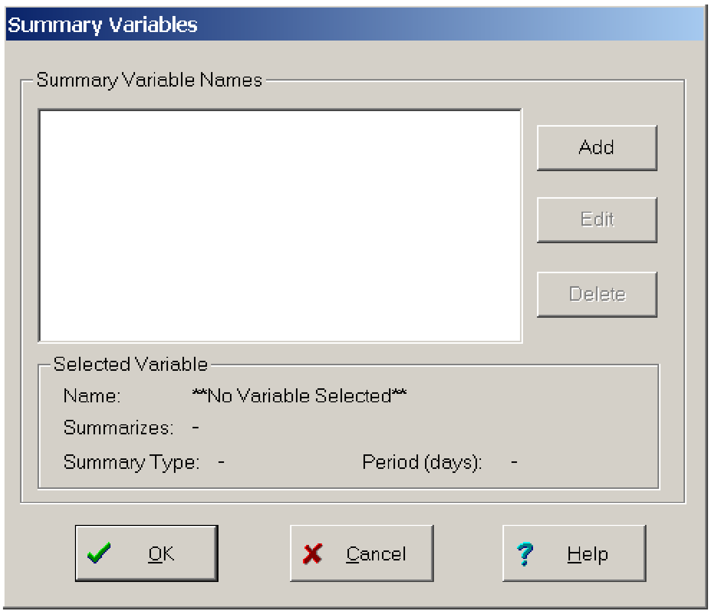


Figure 14.1 Summary Variables list Window

14.2 Setting up a Run Sequence

1. Start the **Simulator** and open the Pseudo-wattle model
2. Select **'Execution'** from the menu bar followed by **'Define Run Sequences...'** to open the **Run Sequence** list window
3. Select **New** to obtain a list of **Sequence Types** that can be created for this model. Highlight "*Fire*" and click **Ok**.

The Event Sequence definition dialogue (Fig 14.2) allows the user to specify a "base" event, consisting of a number of evenly-spaced event "actions" (in our case, fires) starting in a specified week. Once we have decided on this base event, the Sequence consists of repeating this base event for consecutive runs, with the starting week incremented by a given amount.

4. Name the sequence "*Annual Fire*"
5. Make sure that the **Vary Starting Date, fixed No. of Actions** is selected
6. In the **Base Event Definition** panel, make sure that all selections are set to 1
7. In the **Run Sequence Definition** panel, **Increment Starting Week** by should be 1 and **No. of runs in sequence** should be 52
8. Set **Initial Years to exclude** to 2 (this ensures that no fires occur in the first 2 years of any simulation, and gives the wattle time to establish)
9. Exit back to the **Model Components** window

The Fire module row in the Model Components window will now be coloured a light-brown to indicate that a sequence (utilizing this module) has been selected to run. Set the run for 10 years, and initialise the Pseudo-wattle lifecycle with 1 Adult plants

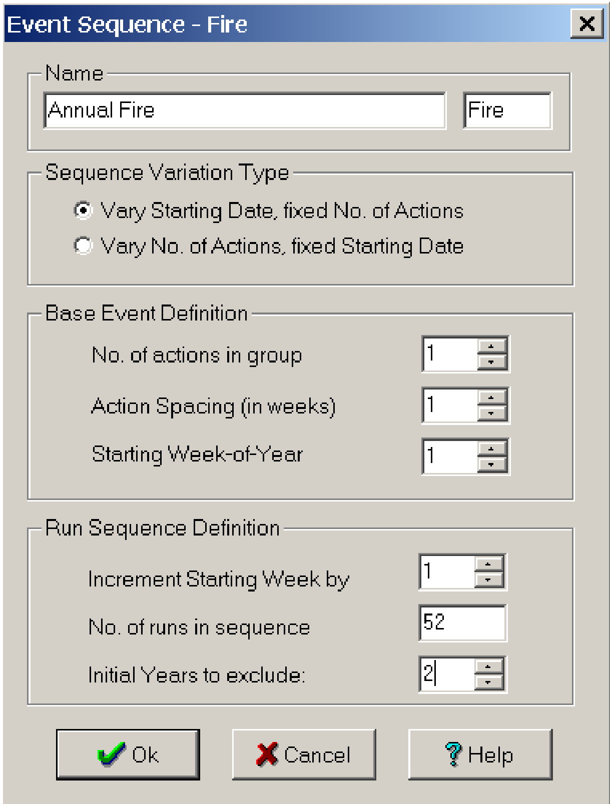


Figure 14.2 Run Sequence dialogue box

10. Select **'Run'** to start the run.

The run sequence now has to process 52 simulations over the ten years for a single model run. This will take some time - possibly several minutes depending on the speed of the computer.

On completion of the run, Charts and Tables can be displayed in the usual way. Notice, however, that the table and chart format dialogs only show a few variables for display. A special variable named "Run" is always present. This variable is a run index for runs within the Sequence. The only other variables present are Summary Variables. In our case, only a single one of these (Canopy Cover) has been defined

Double clicking on a row in the table pops up a menu, from which a detailed chart or table for the corresponding run in the Sequence can be selected. Note that DYMEX reruns the model for that selection, and thus these detailed results may take a moment to appear.

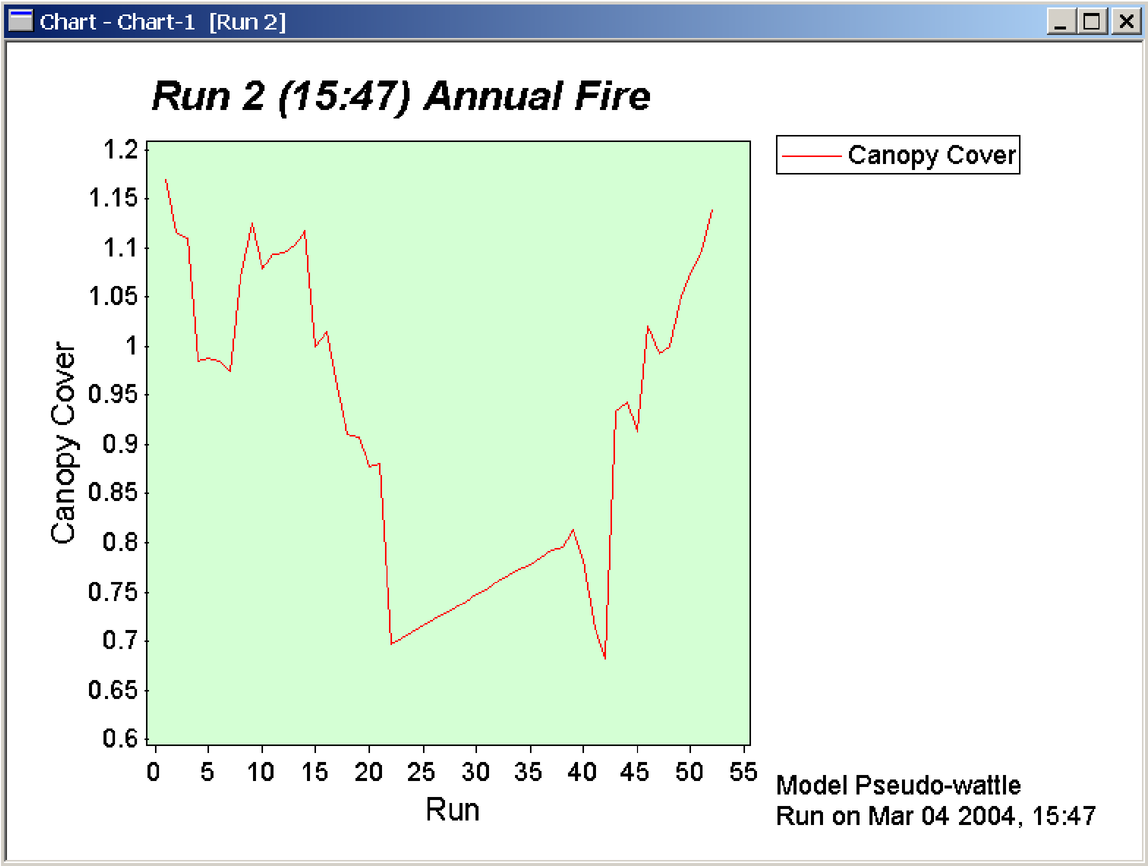
Figure 14.3 Part of Table with Summary Information

Table - Table-...

Fire1	Canopy Cover
Model Pseudo-wattle	
Run on Mar 04 2004, 15:47	
1	1.17
2	1.12
3	1.11
4	0.98
5	0.99
6	0.99
7	0.97
8	1.07
9	1.13
10	1.08
11	1.09
12	1.10
13	1.10
14	1.12
15	1.00
16	1.01
17	0.96
18	0.91
19	0.91
20	0.88
21	0.88
22	0.70
23	0.70
24	0.71
25	0.72
26	0.72
27	0.73
28	0.73
29	0.74
30	0.75

When the chart option is opened, the resultant graph should resemble figure 11.5

Figure 14.4 Summary run of Pseudo-wattle Firing over ten years, using 'Total Canopy Cover' as the summary variable.



The graph shows that the best long-term results from firing occur if it is administered in the middle of the year - the exact dates can be obtained from the tabulation outputs.

PW14.PDF contains the completed model structure, as printed out from the Builder program. Use it to check your settings if the results you obtain are not in agreement with those in the Tutorial