Appendix A
Reference Material

A.1 Repast Batch Running

Using Repast Simphony in batch mode requires a complex list of Java class paths to be supplied in a command-line window. The sample script show in Code A.1 provides an example of how this might be done in a Windows command window. The reader will need to modify this for her particular system, but the adaptations to turn this into a Unix-style script file, for instance, are relatively straightforward. Note that the final `java` command line actually needs to be a single line, rather than broken over several as here, which has been done here purely for formatting purposes.

A.2 Some Common Rules of Differentiation and Integration

In this section we provide a short reference to some differentials and integrals of basic functions. We use the notation to $f(x)$ to represent a function of x and $f'(x)$ to represent the first derivative of $f(x)$, i.e., $\frac{df(x)}{dx}$.

The differential of a sum of two functions of x is the sum of the two differentials

$$\frac{d}{dx}(f(x)+g(x)) = f'(x)+g'(x) \tag{A.1}$$

whereas the differential of a product is a sum of products

$$\frac{d}{dx}(f(x)g(x)) = f'(x)g(x)+f(x)g'(x) \tag{A.2}$$

A.2.1 Common Differentials

Table A.1 illustrates the differentials of some common functions.

DOI 10.1007/978-1-84996-326-8, © Springer-Verlag London Limited 2010

```
@echo off
rem Unofficial configuration file for running Repast Simphony
rem in batch mode.
rem Author: David J. Barnes (d.j.barnes@kent.ac.uk)
rem         http://www.cs.kent.ac.uk/~djb/
rem Version: 2009.12.03
rem
rem Adjust Repast version and local paths below.
rem Windows users: prefer the form
rem                c:\progra~1 over c:\Program files
rem in order to avoid spaces in folder names.
rem
rem The version of Repast Simphony being used.
set VERSION=1.2.0
rem The installed path of Repast.
set REPAST=c:\progra~1\RepastSimphony-%VERSION%
rem The installed path of Eclipse.
set ECLIPSE=c:\progra~1\RepastSimphony-%VERSION%\eclipse
rem The plugins path of Eclipse.
set PLUGINS=%ECLIPSE%\plugins
rem The workspace containing the Repast model.
set WORKSPACE=%REPAST%\workspace
rem The name of the model. This might be case-sensitive.
set MODELNAME=MalariaModelBatch
rem The folder of the model. This might be case-sensitive.
set MODELFOLDER=%WORKSPACE%\%MODELNAME%
rem The file containing the batch parameters.
set BATCHPARAMS=%MODELFOLDER%\batch\batch_params.xml

rem Execute in batch mode.
rem NB: The following lines, up to ".rs" must be joined to be
rem     a single line with no spaces!
java -cp %PLUGINS%/repast.simphony.batch_%VERSION%/bin;
        %PLUGINS%/repast.simphony.runtime_%VERSION%/lib/*;
        %PLUGINS%/repast.simphony.core_%VERSION%/lib/*;
        %PLUGINS%/repast.simphony.core_%VERSION%/bin;
        %PLUGINS%/repast.simphony.bin_and_src_%VERSION%/*;
        %PLUGINS%/repast.simphony.score.runtime_%VERSION%/lib/*;
        %PLUGINS%/repast.simphony.data_%VERSION%/lib/*;
        %MODEL%/bin repast.simphony.batch.BatchMain
    -params %BATCHPARAMS% %MODELFOLDER%\%MODELNAME%.rs
```

Code A.1 Windows script file for running Repast in batch mode

A.2.2 Common Integrals

The notation $\int_a^b f(x)dx$ represents the *definite integral* of $f(x)$ over the interval $[a, b]$. The interval limits are omitted for indefinite integrals. Common integrals can usually be inferred from the corresponding differential (see Table A.1), with the addition of an integration constant, C. Table A.2 illustrates the integrals of some common functions.

Table A.1 Common simple differentials

$f(x)$	$f'(x)$
a	0
x^a	ax^{a-1}
a^x	$\ln(a)a^x$
e^x	e^x (from above)
$\log_a(x)$	$\frac{1}{x\ln(a)}$
$\ln(x)$	$\frac{1}{x}$ (from above)
$\sin(x)$	$\cos(x)$
$\cos(x)$	$-\sin(x)$
$\tan(x)$	$\frac{1}{\cos^2(x)}$

Table A.2 Common simple integrals

$f(x)$	$\int f(x)dx$
a	$ax+C$
a^x	$\frac{a^x}{\ln(a)}+C$
x^a	$\frac{x^{a+1}}{a+1}+C$
$\frac{1}{x}$	$\ln(\lvert x\rvert)+C$
$\log_a(x)$	$x\log_a(x)-\frac{x}{\ln(a)}+C$
$\sin(x)$	$-\cos(x)+C$
$\cos(x)$	$\sin(x)+C$
$\tan(x)$	$-\ln(\lvert\cos(x)\rvert)+C$

Table A.3 Commonly-used notation in Maxima

`%e`	Euler's e number
`%i`	Imaginary unit (i.e. $(\%i)2=-1$)
`%pi`	Pi
`log(x)`	Natural logarithm of x (i.e., $\ln(x)$ is *not* implemented in Maxima)
`inf`	Infinity
`realpart(c)`	Returns the real part of c
`imagpart(c)`	Returns the imaginary part of c
`binom(a,b)`	$\binom{a}{b}$

A.3 Maxima Notation

Full documentation on Maxima can be found in the documentation section at maxima.sourceforge.net. Table A.3 illustrates some commonly-used Maxima notation.

Table A.4 PRISM query summary

`P=?[A U[T1,T2] B]`	Probability that between time $t = T_1$ and $t = T_2$ first A is true and then B is true
`P=?[true U B{IC}]`	Starting from initial conditions IC, what is the probability that B will be true?
`P=?[F B{IC}]`	Equivalent to the above
`P>=0[G B]`	Returns true if B is always true (independent of random choices)
`S>=0[b=4]`	Returns true if b has a non-zero probability of taking a value of 4 in steady state
`R=?[S]`	Returns the reward in steady state
`R=?[C<=10]`	Returns the cumulative reward before time $T = 10$
`R=?[I=10]`	Returns the instantaneous reward

A.4 PRISM Notation Summary

Full documentation on the PRISM model checker may be found at www.prismmodelchecker.org. Table A.4 illustrates some of the most common notation used in model queries.

A.5 Some Mathematical Concepts

This section provides a brief introduction to some of the basic mathematics that has been assumed of the reader in this book. The exposition here cannot replace a proper textbook and is intended purely as an *aide memoire*. The reader who has not met these concepts in detail before is strongly advised to read a dedicated textbook.

A.5.1 Vectors and Matrices

A vector in n-dimensional space is an n-tuple of numbers. The following are examples:

$$\mathbf{v}_1 \doteq \begin{pmatrix} 1 \\ 3.42 \\ 12 \\ i \end{pmatrix} \qquad \mathbf{v}_2 \doteq (1, 3.42, 12, i)$$

v_1 and v_2 are column and row vectors, respectively. We will normally denote the variable names of vectors in boldface, in order to distinguish them from simple scalars. Our two example vectors contain the complex unit, i. Real vectors—that is vectors that contain only real numbers—can be interpreted graphically (Fig. A.1).

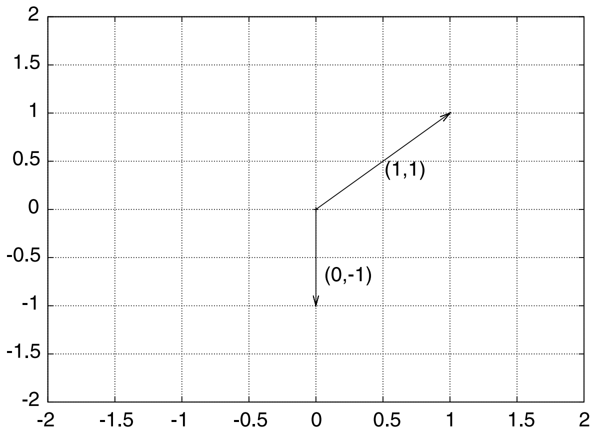

Fig. A.1 A graphical representation of the two vectors $(1, 1)$ and $(0, -1)$

There are a number of operations that can be performed on vectors. The most important is the dot-product. If $\mathbf{v}, \mathbf{w}$ are vectors with the elements $(v_1, v_2, \ldots)$ and $(w_1, w_2, \ldots)$, respectively, then $v \cdot w$ is defined as:

$$v \cdot w = \sum_i v_i \cdot w_i$$

An extension of the concept of a vector is that of a matrix. Matrices are arrays of vectors.

$$\mathbf{A} \doteq \begin{bmatrix} 29 & 52 & 42 \\ 70 & -13 & 18 \\ -32 & 82 & -59 \\ -1 & 72 & 12 \end{bmatrix} \qquad \mathbf{A^T} \doteq \begin{bmatrix} 29 & 70 & -32 & -1 \\ 52 & -13 & 82 & 72 \\ 42 & 18 & -59 & 12 \end{bmatrix}$$

Here $\mathbf{A^T}$ is the transposed matrix of $\mathbf{A}$, which means that the columns and rows are exchanged. More conveniently, one can represent a matrix in terms of its elements: A_{ij} is the element in row i and column j. The element A_{12} would be 52 in the example above. Given two matrices, $\mathbf{A}$ and $\mathbf{B}$, we can define matrix multiplication. The product matrix $(\mathbf{A} \cdot \mathbf{B})$ is the matrix of all products of the row vectors of $\mathbf{A}$ and column vectors of $\mathbf{B}$.

$$(\mathbf{A} \cdot \mathbf{B})_{ij} = \sum_k A_{ik} B_{kj}$$

A numerical example might be helpful. Let us define the matrix $\mathbf{B}$.

$$B \doteq \begin{bmatrix} -26 & -94 & -36 & -15 \\ -86 & -97 & -69 & 2 \\ 50 & -38 & 69 & -88 \end{bmatrix}$$

We can now explicitly calculate the product of $\mathbf{A} \cdot \mathbf{B}$, say.

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} -3126 & -9366 & -1734 & -4027 \\ 198 & -6003 & -381 & -2660 \\ -9170 & -2704 & -8577 & 5836 \\ -5566 & -7346 & -4104 & -897 \end{bmatrix}$$

In matrix multiplication, the order of the operands is significant. The number of rows in the left operand must be equal to the number of columns in the right operand. Therefore, with the current data, the product of $\mathbf{B} \cdot \mathbf{A}$ is not defined for matrix multiplication, because of the mismatch of the dimension of column and row vectors.

A.1 Confirm the results of the matrix products by explicitly calculating the products by hand.

A.2 Square matrices are matrices that have the same number of rows and columns. Define two square matrices $\mathbf{A}$ and $\mathbf{B}$ and calculate the products $\mathbf{A} \cdot \mathbf{B}$ and $\mathbf{B} \cdot \mathbf{A}$. Compare the results.

A vector can be seen as a special case of a matrix. If $\mathbf{v}$ is a row vector of dimension n and $\mathbf{A}$ is a matrix with n columns, then we can calculate the product $\mathbf{v}_2 = \mathbf{v} \cdot \mathbf{A}$ following the same rules as above. Similarly, the matrix product $\mathbf{A}^T \cdot \mathbf{v}^T$ is also well defined. On the other hand, given our definition of matrix product, constructs such as, $\mathbf{A} \cdot \mathbf{v}$ are *not* defined. Henceforth, we will always assume that the dimensions of matrices and vectors match to make sense of the products. It is easy to see that the product of a vector and a matrix is itself a vector. For example, $\mathbf{v}_2$ is again a row vector of dimension n.

Assume now a matrix, $\mathbf{A}$, given in terms of some elements A_{ij}, which we do not need to specify. We can ask whether there is a vector, $\mathbf{v}$, and a scalar value, λ, such that the product of the vector and the matrix equals the vector scaled up by a factor λ. This amounts to solving a linear equation:

$$\mathbf{v} \cdot \mathbf{A} = \lambda \mathbf{v}$$

The vector, $\mathbf{v}$, and the scalar, λ, that solve this equation for $\mathbf{A}$ are called the *eigenvector* and *eigenvalue* of $\mathbf{A}$, respectively. For any specific matrix, there may be more than one solution. Eigenvectors and eigenvalues are important in mathematics and there exists a large body of theory on this topic.[1]

[1]The interested reader is encouraged to read a textbook on *linear algebra* for a better idea of the theoretical underpinnings of this concept.

A.5.2 Probability

In order to talk about the probability of something (an "event"), one first needs to define the range of possible events under consideration. The standard example is the rolling of a die. There are six possible outcomes, which we can simply label 1, 2, 3, 4, 5 and 6. Before rolling the die, we do not know what the outcome will be, yet we can assign to each outcome a probability. If the die is fair then we can assume that each outcome is equally likely, occurring with a probability of $1/6$; or $P(x) = 1/6$ if $x \in \{1, 2, 3, 4, 5, 6\}$.

We can now also think about multiple experiments. If we consider two rolls of a single die, then the space of possible outcomes is extended in that there are now 36 possible outcomes instead of 6. They are the familiar pairings: $(1, 1)$, $(1, 2)$, $(1, 3)$, …, $(6, 6)$, where the first entry in the parenthesis refers to the first roll of the die (the result of the first experiment) and the second entry to the result of the second roll. Each pair of numbers is equally likely and, hence, the probability for each is $1/36$ (which is $\frac{1}{6} \cdot \frac{1}{6}$). Correspondingly, one can extend the space of events by introducing further rolls of the die.

If we perform n experiments, then we know that there are $1/6^n$ possible outcomes altogether—each equally likely. We might be interested in certain subsets of the possible outcomes. For example, we may wish to ask about the probability that the result of each trial is an even number. This can be formulated in a different way by asking for the probability that the first trial resulted in an even number *and* the second *and* the third … *and* the nth. In probability theory, whenever we ask about the probability of several events happening jointly, then we need to multiply the corresponding probabilities. Clearly, for each individual trial the probability of an even number is $1/2$, simply because there are equally many even numbers as odd numbers on the die. Hence, the probability of all outcomes of n trials being even is obtained by multiplication.

$$P(\text{all even}) = \prod_{\text{trials}} \frac{1}{2} = \frac{1}{2^n}$$

We might now ask for the probability that all rolls of the die yield the result 1. We could calculate this by simply multiplying the individual probabilities, as above. However, this is not really necessary. Since all possible outcomes of our die rolling experiment are equally likely, we already know that a result of only 1's has a probability of $1/6^n$. Having calculated this result, we can now ask for the probability that we *either* have only even outcomes *or* all outcomes are 1. Since these two possibilities exclude one another, we can calculate the desired answer by summing the separate probabilities.

$$P(\text{all even or all 1}) = \frac{1}{2^n} + \frac{1}{6^n}$$

We can also ask about *conditional probabilities*—the probability of an event given that another has occurred. For instance, the probability that we have an outcome of all 2's given that we know that all outcomes were even. Formally this is normally written using a vertical bar; so $P(\text{all } 2|\text{all even})$. In some contexts, conditional probabilities can be quite confusing yet, in essence, the concept is quite straightforward. Instead of considering the space of all possible outcomes, the conditional probability asks for the probability of an event based on a restricted subset of all possible outcomes. So, in the example above, we are already given the information that the outcome was even and, based on that, we ask for the probability that the outcome was 2. Another way to look at it is to ask for the probability of obtaining the outcome 2 when all outcomes with odd numbers are discarded from consideration. Formally, it can be calculated as follows:

$$P(\text{all } 2|\text{all even}) = \frac{P(\text{all } 2 \cap \text{all even})}{P(\text{all even})}$$

Here, the symbol $\cap$ means, essentially, that both events need to be true. Since "all 2" is a subset of "all even" the enumerator simply equals $P(\text{all } 2)$. Altogether we thus obtain the solution.

$$P(\text{all } 2|\text{all even}) = \frac{P(\text{all } 2)}{P(\text{all even})} = (2/6)^n = \frac{1}{3^n}$$

This result makes sense. If we assume a die consisting of 3 possible sides only (only even numbers) then the probability for each outcome of the n trials would be exactly $1/3^n$, just as we calculated above.

A.5.3 Probability Distributions

If we roll the die n times, what is the probability that we get an even number exactly $0 \leq k \leq n$ times? Here we do not care about whether the die gives us a 4 or a 6, and we thus reduce the outcomes to purely binary values. As it turns out, we do not need to worry about calculating this, as there is a pre-computed answer available: The *binomial distribution*. If we roll the die exactly n times then the probability that we get an even number exactly k times is given by:

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

The term $\binom{n}{k}$ is called the *binomial coefficient* and is defined as:

$$\binom{n}{k} \doteq \frac{n!}{k!(n-k)!}$$

The exclamation mark refers to the factorial function, i.e., $k! \doteq k \cdot (k-1) \cdot (k-2) \cdots 1$. The binomial coefficient has a specific meaning independent of the binomial distribution. It gives the number of different ways to choose k items out of a collection of n items. For example, assume we have a container with n distinct balls and we randomly pick k of them. The binomial coefficient then tells us the number of possible combinations of balls we end up with. For example, if each ball has a number on it, how many possible combinations of numbers are there when we randomly select k balls (assuming we disregard the order in which the balls are selected). This finds an immediate application in lotteries. If there are $n = 45$ balls in the container and each week $k = 6$ balls are drawn, then according to the binomial coefficient there are altogether 8145060 possible combinations; the probability of winning with a single ticket is the inverse of this number.

The binomial distribution is an example of a discrete probability distribution. It can be applied whenever there is a series of events, each of which has just two possible outcomes. An example is the flipping of a coin where the outcomes are "head" or "tail". In this case, both outcomes are equally likely and have a probability of $p = 0.5$. In general, the probabilities of the two outcomes do not need to be equal, although clearly they always need to sum to 1.

Our even/odd example of n rolls of the die is equivalent to n coin tosses. Each roll event has a probability of $p = 0.5$ of returning an even number. Hence, if we assume $n = 20$ and we want to know the probability that there will be exactly $k = 18$ outcomes with an even number, then we can calculate this as follows:

$$P(18) = \binom{20}{18}\left(\frac{1}{2}\right)^{18}\left(\frac{1}{2}\right)^{2} \approx 0.000181198$$

Another distribution that is very important in biological modeling is the *Poisson distribution* given by

$$P(k) = \frac{m^k \exp(-m)}{k!}.$$

Assume we have an event that happens m times per time unit, on average. The Poisson distribution describes the probability of *actually* observing exactly k events per time unit. For example, if somebody receives 10 emails per hour, on average, then we can use the Poisson distribution to calculate the probability that this person receives exactly 2 emails within the next hour by setting $m = 10$ and $k = 2$.

The best known probability distribution of all is the Gaussian distribution. Most readers will be familiar with it, so we will only write it down for reference.

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)$$

Here σ^2 is the variance of the distribution and m is the mean. This is one of the iconic and ubiquitous functions in science. The classical example of a Gaussian distributed variable is the size of children in a school class. One of the reasons for the omnipresence of the Gaussian is the *central limit theorem* which states that, under a wide range of conditions, random processes consisting of a large number of random variables will show Gaussian distributions.

A.5.4 Taylor Expansion

A technique that may be less well known to the reader, but is of high practical importance, is that of the *Taylor expansion* of a function at a particular point. The idea

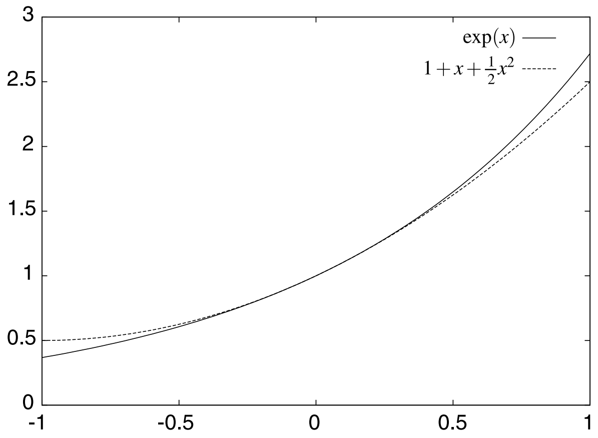

Fig. A.2 A Taylor expansion of the function $f(x) = \exp(x)$ to second order around $x_0 = 0$. The approximation is compared with the exact result

of a Taylor expansion is to approximate a function locally by a simpler (or more convenient, or computationally cheaper) function. The accuracy of the approximation depends on the function itself, as well as the order of the Taylor expansion. Assume a function $f(x)$. The Taylor expansion of this function is then given by:

$$f(x) = f(x_0) + f^{(1)}(x_0)(x - x_0) + \frac{1}{2!}f^{(2)}(x_0)(x - x_0)^2 + \frac{1}{3!}f^{(3)}(x_0)(x - x_0)^3 + \cdots$$

Here x_0 is a point in the neighborhood of x and $f^{(n)}$ is the nth derivative of f. Often x_0 is chosen to be zero, but this is not a requirement. The Taylor approximation becomes successively better for each additional term. However, in practice, the series is often truncated after the first order term, which sometimes still gives a good local approximation.

As an example consider the second-order approximation of the exponential function

$$\exp(x) \approx e^{x_0} + e^{x_0}(x - x_0) + 1/2e^{x_0}(x - x_0)^2$$

which is illustrated in Fig. A.2.

References

1. Bak, P.: How Nature Works. Oxford University Press, London (1997)
2. Barnes, D.J., Kölling, M.: Objects First with Java—A Practical Introduction Using BlueJ, 4th edn. Pearson Education, London (2008)
3. Bonabeau, E., Theraulaz, G., Dorigo, M.: Self-organization in social insects. Santa Fe Institute Working Paper 97-04-032 (1997)
4. Bratsun, D., Volfson, D., Tsimring, L.S., Hasty, J.: Delay-induced stochastic oscillations in gene regulation. Proceedings of the National Academy of Sciences of the United States of America **102**(41), 14,593–14,598 (2005)
5. Casti, J.: Reality Rules: I The Fundamentals. Wiley, New York (1992)
6. Casti, J.: Reality Rules: II The Frontier. Wiley, New York (1992)
7. Casti, J.: Would-Be Worlds. Wiley, New York (1997)
8. Cherry, J., Adler, F.: How to make a biological switch. Journal of Theoretical Biology **203**(2), 117–133 (2000). doi:10.1006/jtbi.2000.1068
9. Chu, D., Rowe, J.: Spread of vector borne diseases in a population with spatial structure. In: Proceedings of PPSN VIII—Eight International Conference on Parallel Problem Solving from Nature. Lecture Notes in Computer Science, vol. 3242, pp. 222–232. Springer, Birmingham (2004)
10. Chu, D., Zabet, N., Mitavskiy, B.: Models of transcription factor binding: sensitivity of activation functions to model assumptions. Journal of Theoretical Biology **257**(3), 419–429 (2009). doi:10.1016/j.jtbi.2008.11.026
11. Dawkins, R.: The Selfish Gene. University Press, Oxford (1989)
12. DeGroot, M.H.: A conversation with George box. Statistical Science **2**(3), 239–258 (1987). doi:10.1214/ss/1177013223
13. Devroye, L.: Non-uniform Random Variate Generation. Springer, New York (1986)
14. Dijkstra, E.W.: Notes on Structured Programming. Academic Press, London (1972). Chap. I
15. Foundation, F.S.: Gnu general public license (2007). URL http://www.gnu.org/licenses/gpl.html
16. Foundation, E.: Eclipse integrated development environment (2010). URL http://www.eclipse.org/
17. Frette, V., Christensen, K., Malthe-Sorenssen, A., Feder, J., Jssang, T., Meakin, P.: Avalanche dynamics in a pile of rice. Nature **379**, 49–52 (1996)
18. Gardiner, C.: Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences. Springer, Berlin (2008)
19. Gibson, M., Bruck, J.: Efficient exact stochastic simulation of chemical systems with many species and many channels. Journal of Physical Chemistry **104**, 1876–1889 (2000)
20. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics **22**(403) (1976)

21. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry **81**(25), 2340–2361 (1977)
22. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. Journal of Chemical Physics **115**(4), 1716–1733 (2001)
23. Gould, S.: The Structure of Evolutionary Theory. Belknap Press, Cambridge (2002)
24. Groovy: Groovy—an agile dynamic language for the java platform (2010). URL http://groovy.codehaus.org/
25. Gross, D., Strand, R.: Can agent-based models assist decisions on large-scale practical problems? A philosophical analysis. Complexity **5**(5), 26–33 (2000)
26. Huse, G., Giske, J.: Ecology in Mare Pentium: an individual based spatio-temporal model for fish with adapted behaviour. Fisheries Research **37**, 163–178 (1998)
27. Ierusalimschy, R., de Figueiredo, L.H., Celes, W.: Lua—an extensible extension language. Software: Practice & Experience **26**(6), 635–652 (1996). URL http://www.lua.org/
28. Kohler, T., Gumerman, G.: Dynamics of Human and Primate Societies. Oxford University Press, Oxford (1999)
29. Kwiatkowska, M., Norman, G., Parker, D.: PRISM: Probabilistic symbolic model checker. In: Kemper, P. (ed.) Proc. Tools Session of Aachen 2001 International Multiconference on Measurement, Moddeling and Evaluation of Computer-Communication Systems, pp. 7–12, September 2001
30. Murray, J.: Mathematical Biology. Springer, Berlin (2002)
31. Nowak, M.: Evolutionary Dynamics: Exploring the Equations of Life. Harvard University Press, Cambridge (2006)
32. Ramsey, S., Orrell, D., Bolouri, H.: Dizzy: stochastic simulation of large-scale genetic regulatory networks. Journal of Bioinformatics and Computational Biology **3**, 415–436 (2005)
33. Ramsey, S., Orrell, D., Bolouri, H.: Dizzy home page (2010). URL http://magnet.systemsbiology.net/software/Dizzy/
34. Ray, T.: An Approach to the Syntheses of Life. Oxford Readings in Philosophy, pp. 111–145. Oxford University Press, Oxford (1996)
35. RepastS: Repast agent simulation toolkit (2010). URL http://repast.sourceforge.net/
36. Ribeiro, A.S., Lloyd-Price, J.: Sgn sim, a stochastic genetic networks simulator. Bioinformatics **23**, 777–779 (2007). doi:10.1093/bioinformatics/btm004
37. Roussel, M.R., Zhu, R.: Validation of an algorithm for delay stochastic simulation of transcription and translation in prokaryotic gene expression. Physical Biology **3**, 274–284 (2006)
38. Slepoy, A., Thompson, A.P., Plimpton, S.J.: A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. Journal of Chemical Physics **128** (2008)
39. Sober, E., Wilson, D.: Unto Others, the Evolution and Psychology of Unselfish Behaviour. Harvard University Press, Cambridge (1998)
40. Tesfatsion, L.: Agent-based computational economics: growing economies from the bottom up. Artificial Life **8**(1), 55–82 (2002)
41. Traulsen, A., Nowak, M.: Evolution of cooperation by multilevel selection. Proceedings of the National Academy of Science of the United States of America **103**(29), 10,952–10,955 (2006). doi:10.1073/pnas.0602530103
42. Venables, M., Bilge, U.: Complex Adaptive Modelling at Sainsbury. Business Processes Resource Centre (1998)
43. Wilson, D.S.: A theory of group selection. Proceedings of the National Academy of Science of the United States of America **72**(1), 143–146 (1975)
44. Wilson, D.: Evolutionary biology: Struggling to escape exclusively individual selection. The Quarterly Review of Biology **76**(2), 199–205 (2001)
45. Wolfram, S.: Cellular Automata and Complexity. Addison–Wesley, Reading (1994)

Index