Solution to Kolmogorov's equations for some common Markov models

Nicky J. Welton
MRC Health Services Research Collaboration
Department of Social Medicine, University of Bristol
Nicky.Welton@bristol.ac.uk

February 20, 2007

1 Introduction

This material is intended as a supplement to Welton, N.J. and Ades, A.E. 'Estimation of Markov Chain Transition Probabilities and Rates and the Propagation of Uncertainty' submitted to Medical Decision Making, in which we show how Kolmogorov's forward equations can be exploited to recover underlying transition rates from data where observations are made in discrete time. In the manuscript we give the solution to Kolmogorov's equations for the simple 2-state model and for the 3-state model with forward transitions only. Here, we extend these results for more complex Markov Chain models. These solutions can be used within a Markov chain Monte Carlo simulation to provide estimates for both transition rates and transition probabilities, based on transition probability data over fixed time intervals. We outline an approach for the practical estimation of transition rates and probabilities for general homogenous Markov models. Note that models with both forward and backward transitions will generally not be identifiable in the Case 3 situation unless observations are available at more than one point in time after the start of the study.

2 Kolmogorov's Equations

We seek the solution to Kolmogorov's forward equations (e.g. [1]):

$$\begin{aligned} P'(t) &= P(t)G \\ \text{subject to} \quad P(0) &= I, \end{aligned} \tag{1}$$

where G is the matrix of transition rates with elements $\gamma_{i,j}$, and $P(t)$ is the matrix of transition probabilities with elements $\pi_{i,j}(t)$, the probability of a transition from state i to state j over time cycle t.

Under certain conditions (see [1]), Equations (1) have solution:

$$P(t) = \exp(tG) = \sum_{n=0}^{\infty} \frac{t^n}{n!} G^n. \tag{2}$$

3 Examples

Here we evaluate Equation (2) for various different Markov models. It is useful to define the sum transition rate from state i:

$$\lambda_i = \sum_{j \neq i} \gamma_{i,j}.$$

3.1 Two-State Models

In the simplest case there are just two states, and only forward transitions are possible (Figure 1a). Here the solution to Equation (2) is:

$$P(t) = \begin{pmatrix} \pi_{1,1}(t) & \pi_{1,2}(t) \\ \pi_{2,1}(t) & \pi_{2,2}(t) \end{pmatrix} = \begin{pmatrix} e^{-\gamma_{1,2}t} & 1 - e^{-\gamma_{1,2}t} \\ 0 & 1 \end{pmatrix}$$

If we add backwards transitions to the two-state model (Figure 1b) then the solution becomes:

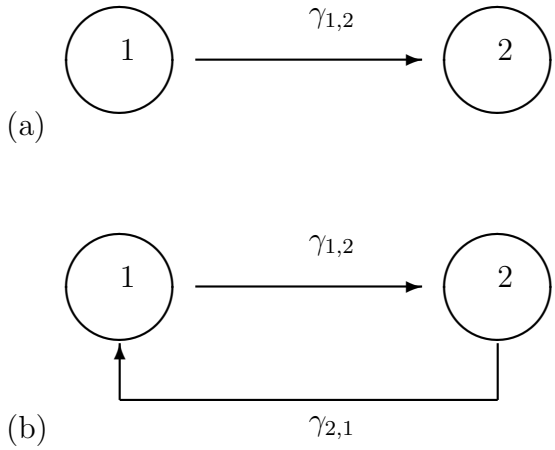

Figure 1: Two-state model with (a) forward transitions only; and (b) forward and backward transitions

$$\begin{aligned}
\pi_{1,1}(t) &= \frac{\gamma_{2,1} + \gamma_{1,2}e^{-(\gamma_{1,2}+\gamma_{2,1})t}}{(\gamma_{1,2} + \gamma_{2,1})} \\
\pi_{1,2}(t) &= 1 - \pi_{1,1}(t) \\
\pi_{2,1}(t) &= 1 - \pi_{2,2}(t) \\
\pi_{2,2}(t) &= \frac{\gamma_{1,2} + \gamma_{2,1}e^{-(\gamma_{1,2}+\gamma_{2,1})t}}{(\gamma_{1,1} + \gamma_{2,1})}
\end{aligned}$$

3.2 Three-State Models

In the three-state model with forward transitions only (Figure 2a), the solution to Equation (2) is

$$\begin{aligned}
\pi_{1,1}(t) &= e^{-\lambda_1 t} \\
\pi_{1,2}(t) &= \frac{\gamma_{1,2}\left(e^{-\lambda_2 t} - e^{-\lambda_1 t}\right)}{(\lambda_1 - \lambda_2)} \\
\pi_{1,3}(t) &= (1 - \pi_{1,1}(t) - \pi_{1,2}(t)) \\
\pi_{2,1}(t) &= 0 \\
\pi_{2,2}(t) &= e^{-\lambda_2 t}
\end{aligned}$$

$$\begin{aligned} \pi_{2,3}(t) &= (1 - \pi_{2,1}(t) - \pi_{2,2}(t)) \\ \pi_{3,1}(t) &= \pi_{3,2}(t) = 0 \qquad \pi_{3,3}(t) = 1 \end{aligned}$$

and this also gives the solution for the forward transition tunnel model (Figure 2b), by setting $\gamma_{1,3} = 0$.

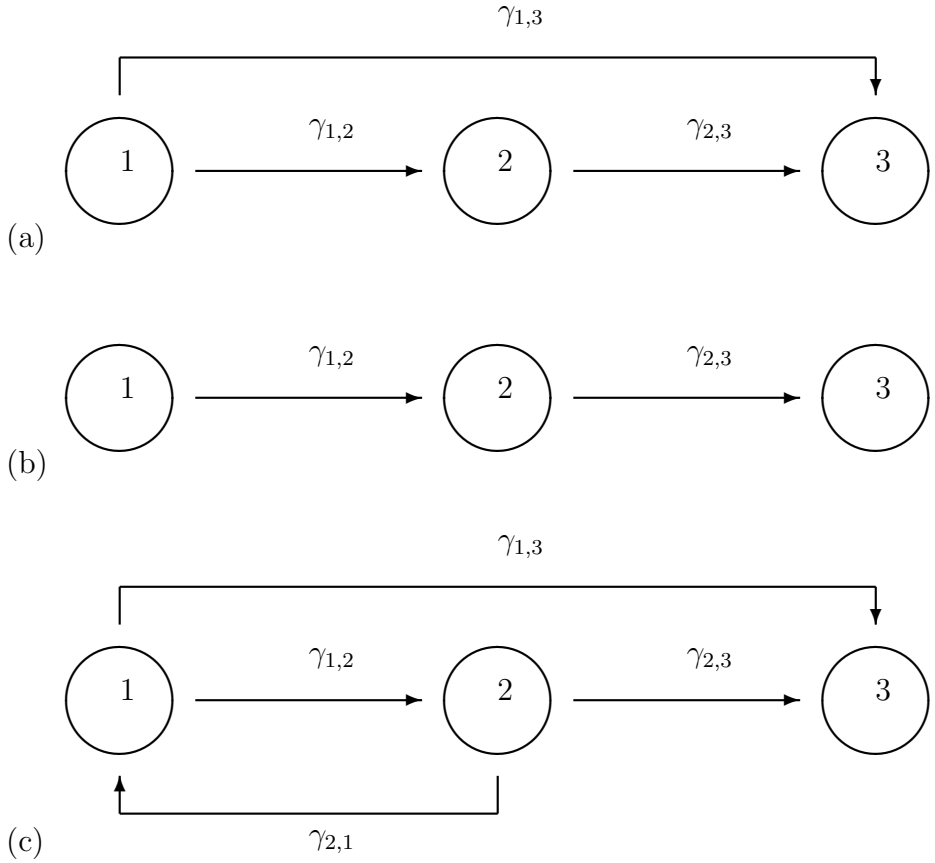

Figure 2: Three-state model with (a) forward transitions only; (b) forward transitions through a tunnel model; (c) forward and backward transitions

The solution for the three-state model, with both forward and backward transitions (Figure 2c) is more complex, but can be found in analytical form. Let

$$h = \sqrt{(\lambda_1 - \lambda_2)^2 + 4\gamma_{1,2}\gamma_{2,1}}.$$

Then,

$$\pi_{1,1}(t) = \frac{(-\lambda_1 + \lambda_2 + h)e^{-\frac{1}{2}(\lambda_1+\lambda_2-h)t} + (\lambda_1 - \lambda_2 + h)e^{-\frac{1}{2}(\lambda_1+\lambda_2+h)t}}{2h}$$

$$
\begin{aligned}
\pi_{1,2}(t) &= \frac{(-\lambda_1+\lambda_2+h)(\lambda_1-\lambda_2+h)\left(e^{-\frac{1}{2}(\lambda_1+\lambda_2-h)t}-e^{-\frac{1}{2}(\lambda_1+\lambda_2+h)t}\right)}{4h\gamma_{2,1}} \\
\pi_{1,3}(t) &= 1-\pi_{1,1}(t)-\pi_{1,2}(t) \\
\pi_{2,1}(t) &= \frac{\gamma_{2,1}\left(e^{-\frac{1}{2}(\lambda_1+\lambda_2-h)t}-e^{-\frac{1}{2}(\lambda_1+\lambda_2+h)t}\right)}{h} \\
\pi_{2,2}(t) &= \frac{(\lambda_1-\lambda_2+h)e^{-\frac{1}{2}(\lambda_1+\lambda_2-h)t}+(-\lambda_1+\lambda_2+h)e^{-\frac{1}{2}(\lambda_1+\lambda_2+h)t}}{2h} \\
\pi_{2,3}(t) &= 1-\pi_{2,1}(t)-\pi_{2,2}(t) \\
\pi_{3,1}(t) &= \pi_{3,2}=0, \qquad \pi_{3,3}=1
\end{aligned}
$$

3.3 Four-State Models

In the four-state model with forward transitions only (Figure 3a), the solution to Equation (2) is

$$
\begin{aligned}
\pi_{1,1}(t) &= e^{-\lambda_1 t} \\
\pi_{1,2}(t) &= \frac{\gamma_{1,2}\left(e^{-\lambda_2 t}-e^{-\lambda_1 t}\right)}{(\lambda_1-\lambda_2)} \\
\pi_{1,3}(t) &= \frac{\gamma_{1,3}\left(e^{-\lambda_3 t}-e^{-\lambda_1 t}\right)}{(\lambda_1-\lambda_3)}+\frac{\gamma_{1,2}\gamma_{2,3}\left((\lambda_1-\lambda_2)e^{-\lambda_3 t}-(\lambda_1-\lambda_3)e^{-\lambda_2 t}+(\lambda_2-\lambda_3)e^{-\lambda_1 t}\right)}{(\lambda_1-\lambda_2)(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)} \\
\pi_{1,4}(t) &= 1-\pi_{1,1}(t)-\pi_{1,2}(t)-\pi_{1,3}(t) \\
\pi_{2,1}(t) &= 0 \\
\pi_{2,2}(t) &= e^{-\lambda_2 t} \\
\pi_{2,3}(t) &= \frac{\gamma_{2,3}\left(e^{-\lambda_3 t}-e^{-\lambda_2 t}\right)}{(\lambda_2-\lambda_3)} \\
\pi_{2,4}(t) &= 1-\pi_{2,1}(t)-\pi_{2,2}(t)-\pi_{2,3}(t) \\
\pi_{3,1}(t) &= \pi_{3,2}(t)=0 \\
\pi_{3,3}(t) &= e^{-\lambda_3 t} \\
\pi_{3,4}(t) &= 1-\pi_{3,1}(t)-\pi_{3,2}(t)-\pi_{3,3}(t) \\
\pi_{4,1}(t) &= \pi_{4,2}(t)=\pi_{4,3}(t)=0 \qquad \pi_{4,4}(t)=1
\end{aligned}
$$

and this also gives the solution for the forward transition tunnel model (Figure 3b), by setting $\gamma_{1,3}=\gamma_{1,4}=\gamma_{2,4}=0$.

Although a pattern is beginning to emerge, the solution to Equations (1) quickly become unwieldy. In the following section we give WinBUGS1.4 code to estimate transition rates and probabilities for the models described above, and outline how to extend to more complex Markov models by solving the differential equations numerically, avoiding the necessity to write down the solution to Equation (2) algebraically.

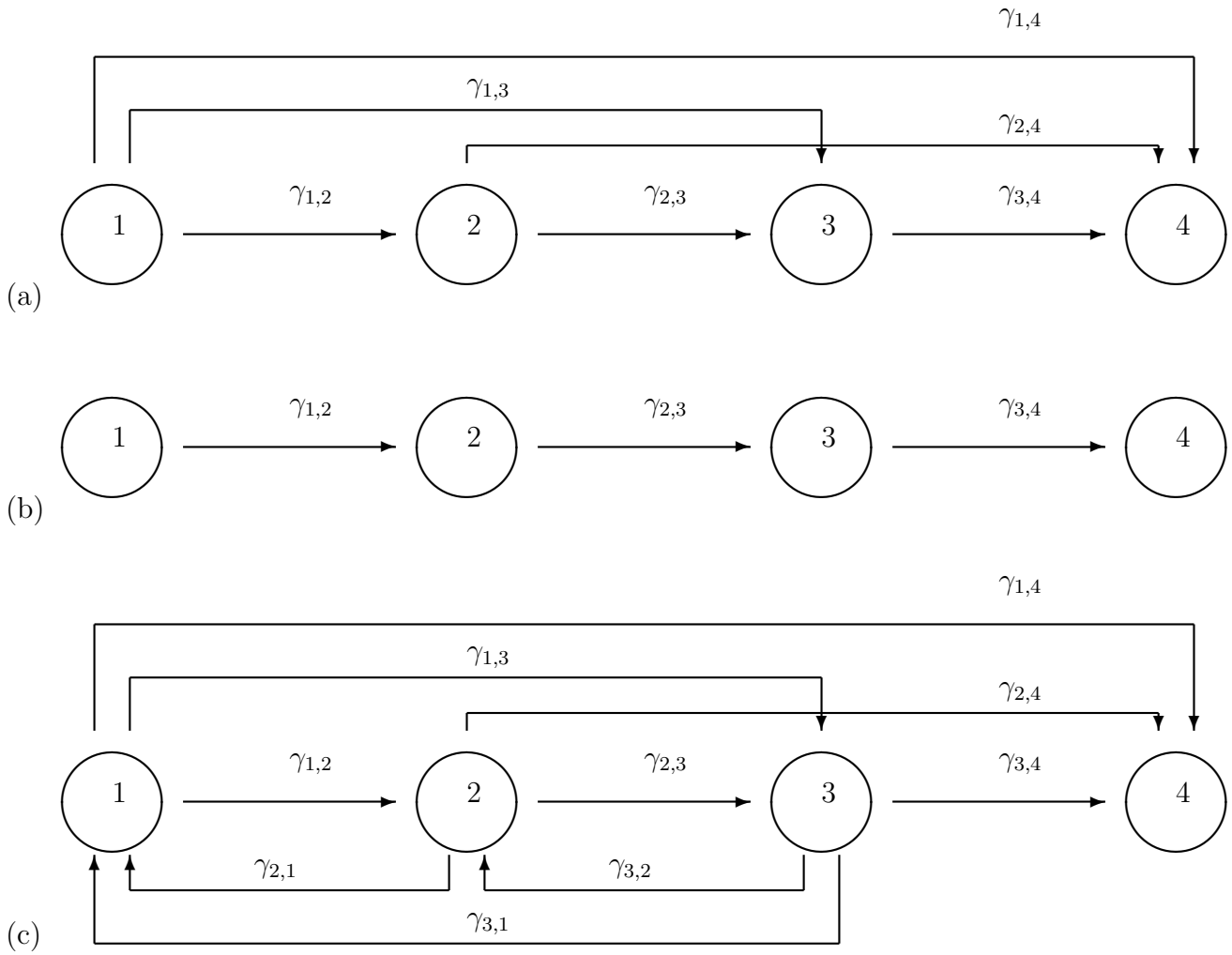

Figure 3: Four-state model with (a) forward transitions only; (b) forward transitions through a tunnel model; (c) forward and backward transitions

4 Example WinBUGS1.4 code

For a description of the data and model see Welton, N.J. and Ades, A.E. 'Estimation of Markov Chain Transition Probabilities and Rates and the Propagation of Uncertainty' submitted to Medical Decision Making. Here we list example WinBUGS1.4 programs not included in the manuscript (the .odc file can be downloaded from the web-site). For details of the freely available WinBUGS1.4 software, see

http://www.mrc-bsu.cam.ac.uk/bugs/welcome.shtml

4.1 Three-state forwards model: evidence consistency

```
model{ #Multinomial likelihood for observed data for (i in 1:2){
r[i,1:3] ~ dmulti(pi[i,1:3],n[i])
    for(j in 1:3){
        pi[i,j]<- P[i,j,1]
#Predicted transitions
        r.hat[i,j]<-pi[i,j]*n[i] + .0001*equals(pi[i,j],0) -.0001*equals(pi[i,j],1)
        }
#Calculate deviance for multinomial data, (comparing observed and
predicted values)
        for (j in i:3){dev[i,j]<-2*r[i,j]*log(r[i,j]/r.hat[i,j])}
}
#Deviance from given initial state
        dev.ini[1]<-sum(dev[1,])
        dev.ini[2]<-dev[2,2]+dev[2,3]
#Total deviance
    dev.tot<-sum(dev.ini[])
#Prob. that the deviance exceeds the 95% critical value from the chi-square 3 distn
    p.dev<-step(dev.tot - 7.815)

#Find transition probabilities (for given time) in terms of rates
using Fig. 3 s<-G[1,2]/(G[1,2]+G[1,3]-G[2,3])

for (t in 1:2){
    time[t]<- 2*equals(t,1) + .25*equals(t,2)
```

```
    P[1,1,t]<- exp(-(G[1,2] + G[1,3])*time[t])
    P[1,2,t]<- s*exp(-G[2,3]*time[t])*(1 - exp(-(G[1,2]+G[1,3]-G[2,3])*time[t]))
    P[1,3,t]<-1 - P[1,1,t] - P[1,2,t]

    P[2,1,t]<-0
    P[2,2,t]<-exp(-G[2,3]*time[t])
    P[2,3,t]<-1 - P[2,2,t]
}

for (j in 2:3){G[1,j] ~ dexp(.001)} G[2,3] ~ dexp(.001)
}

#Data
list(r=structure(.Data=c(20,6,4,0,5,5),.Dim=c(2,3)),n=c(30,10))

#Initial Values
list(G=structure(.Data=c(NA,4,NA,NA,NA,9),.Dim=c(2,3)))
```

The 2-parameter tunnel model can be fitted by adjusting the above code, so that G[1,3]=0.

4.2 Three-state general model

```
model{

#Multinomial likelihood for observed data
    for (i in 1:2){ r[i,1:3] ~ dmulti(P[i,1:3],n[i])    }

#Find transition probabilities (for given time) in terms of rates
    h<- sqrt(pow(lambda[1]-lambda[2], 2) + 4*G[1,2]*G[2,1])
    e1<-exp(-.5*(lambda[1] + lambda[2] - h)*t.obs)
    e2<-exp(-.5*(lambda[1] + lambda[2] + h)*t.obs)

    P[1,1]<-((-lambda[1]+lambda[2]+h)*e1
                + (lambda[1]-lambda[2]+h)*e2)/(2*h)
```

```
    P[1,2]<-((-lambda[1]+lambda[2]+h)
            *(lambda[1]-lambda[2]+h)*(e1-e2))/(4*h*G[2,1])
    P[1,3]<-1 - P[1,1] - P[1,2]

    P[2,1]<- G[2,1]*(e1-e2)/h
    P[2,2]<- ((lambda[1]-lambda[2]+h)*e1
            + (-lambda[1] + lambda[2] + h)*e2)/(2*h)
    P[2,3]<-1 - P[2,1] - P[2,2]

#Give exponential priors for unknown transition rate parameters
    for (i in 1:2){
        for (j in (i+1):3){G[i,j] ~ dexp(.001)}
    }
    for (i in 2:2){
    for (j in 1:(i-1)){G[i,j] ~ dexp(.001)}
    }

    lambda[1]<- G[1,2] + G[1,3]
    lambda[2]<- G[2,1] + G[2,3]


#Find P(t.new) for given new time of interest
    e1.new<-exp(-.5*(lambda[1] + lambda[2] - h)*t.new)
    e2.new<-exp(-.5*(lambda[1] + lambda[2] + h)*t.new)

    Pt[1,1]<-((-lambda[1]+lambda[2]+h)*e1.new
                + (lambda[1]-lambda[2]+h)*e2.new)/(2*h)
    Pt[1,2]<-((-lambda[1]+lambda[2]+h)
                *(lambda[1]-lambda[2]+h)*(e1.new-e2.new))/(4*h*G[2,1])
    Pt[1,3]<-1 - P[1,1] - P[1,2]

    Pt[2,1]<- G[2,1]*(e1.new-e2.new)/h
    Pt[2,2]<- ((lambda[1]-lamba[2]+h)*e1.new
                + (-lambda[1] + lambda[2] + h)*e2.new)/(2*h)
    Pt[2,3]<-1 - P[2,1] - P[2,2]

}

#Data
list(r=structure(.Data=c(17,6,7,1,4,5),.Dim=c(2,3)),n=c(30,10),t.obs=2,t.new=.25)
```

```
#Initial Values
list(G=structure(.Data=c(NA,.1,.1,.1,NA,.1),.Dim=c(2,3)))
```

4.3 Four-state forwards model

```
model{

#Multinomial likelihood for observed data
    for (i in 1:3){ r[i,1:4] ~ dmulti(P[i,1:4],n[i])    }

#Find transition probabilities (for given time) in terms of rates.
Sum transition rate from #state i, lambda[i] denoted s[i] for
brevity.
    for (i in 1:3){
        s[i]<- sum(G[i,(i+1):4])
        e[i]<- exp(-s[i]*t.obs)
        P[i,4]<-  1 - sum(P[i,1:3])

#Give exponential priors for unknown transition rate parameters
    for (j in (i+1):4){G[i,j] ~ dexp(.001)}
    G[i,i]<- -lambda[i]
    }

    P[1,1]<-e[1]
    P[1,2]<- G[1,2]*(e[2] - e[1])/(s[1] - s[2])
    P[1,3]<- G[1,3]*(e[3] - e[1])/(s[1] - s[3]) + G[1,2]*G[2,3]*((s[1]-s[2])*e[3]
        - (s[1]-s[3])*e[2] + (s[2]-s[3])*e[1])/((s[1]-s[2])*(s[1]-s[3])*(s[2]-s[3]))

    P[2,2]<-e[2]
    P[2,3]<- G[2,3]*(e[3] - e[2])/(s[2] - s[3])

    P[3,3]<-e[3]

    P[2,1]<-0
    P[3,1]<-0
    P[3,2]<-0
```

```
#Find P(t) where t = t.new
    for (i in 1:3){
        e.t[i]<- exp(-s[i]*t.new)
        P.t[i,4]<-  1 - sum(P.t[i,i:3])
    }

    P.t[1,1]<-e.t[1]
    P.t[1,2]<- G[1,2]*(e.t[2] - e.t[1])/(s[1] - s[2])
    P.t[1,3]<- G[1,3]*(e.t[3] - e.t[1])/(s[1] - s[3])
        + G[1,2]*G[2,3]*((s[1]-s[2])*e.t[3] - (s[1]-s[3])*e.t[2]
        + (s[2]-s[3])*e.t[1])/((s[1]-s[2])*(s[1]-s[3])*(s[2]-s[3]))

    P.t[2,2]<-e.t[2]
    P.t[2,3]<- G[2,3]*(e.t[3] - e.t[2])/(s[2] - s[3])

    P.t[3,3]<-e.t[3]

}

#Data
list( r=structure(.Data=c(20,6,4,2,
                          0,5,5,3,
                          0,0,3,7), .Dim=c(3,4)),
                          n=c(32,13,10), t.obs=2, t.new=.25)

#Initial Values
                    list(G=structure(.Data=c(NA,1,1,1,
                                             NA,NA,1,1,
                                             NA,NA,NA,1
                                             ), .Dim=c(3,4)))
```

5 More complex Markov models

Rather than writing down the algebraic solution to Equation (2), a more promising approach from a practical perspective is to use software designed to specify complex functions of parameters via arbitrary systems of ordinary

differential equations, such as the WinBUGS Differential Interface (WBDiff). This requires the user to specify the system of differential equations (Equations 1). WBDiff is currently at the beta-testing stage, but is due for official release in March 2004. For details, see

`http://homepages.tesco.net/~creeping_death/`

Example programs using the WBDiff interface can be obtained from Nicky Welton on request (Nicky.Welton@bristol.ac.uk).

References

[1] G.D.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, 2nd edition, 1992.