Department of Mathematics and Statistics
WinBUGS and OpenBUGS applications
Exam 24.3.2011

(5 questions, 30 points total. You may write in English/Finnish)

1. Explain the meaning of the following terms (1 point each):
(A) Jeffreys' prior, (B) alternating sampling, (C) burn-in period,
(D) cut-function, (E) I-function, (F) logit-link.

(A) For a scalar parameter r, Jeffrey's prior distribution is proportional to the square root of Fisher Information for r. Hence, $\pi(r) \propto J(r)^{1/2}$, where $J(r) = E\Big[\Big(\frac{\mathbf{d} \log \pi(X|r)}{\mathbf{d}r}\Big)^2 \mid r\Big] = -E\Big[\frac{\mathbf{d}^2 \log \pi(X|r)}{\mathbf{d}r^2} \mid r\Big]$. This prior is invariant under parameter transformations, because $J(\psi)^{1/2} = J(r)^{1/2} \mid \frac{\mathbf{d}r}{\mathbf{d}\psi} \mid$. With binomial parameter r, the prior is Beta(1/2,1/2).
(B) MCMC sampling method (Gibbs sampling) where one parameter at at time is sampled from the full conditional distribution, given the current values of other parameters. E.g. start with initial values θ_1^0, θ_2^0 and sample θ_1^1 from $\pi(\theta_1 \mid \theta_2^0, \text{data})$ and θ_2^1 from $\pi(\theta_2 \mid \theta_1^1, \text{data})$, then θ_1^2 from $\pi(\theta_1 \mid \theta_2^1, \text{data})$, etc. The full conditionals need to be solved from the full joint distribution.
(C) initial phase of MCMC sampling which does not yet represent the target distribution because starting values were too far from the distribution and/or because of slowly mixing chain.
(D) cut-function in BUGS can be used to cut the feedback of information upwards from descendant nodes but the uncertainties are still propagated downwards. Hence, data about child nodes would not affect our inferences about their parent nodes, but uncertainties about parent nodes would be propagated to child nodes.
(E) I-function is used to incorporate correct likelihood contribution from censored observations. That is, to encode probabilities of the type $P(X_i > a_i \mid \theta)$, $P(X_i < b_i \mid \theta)$ or $P(a_i < X_i < b_i \mid \theta)$ for measurements X_i with censored values. Censoring can be from left, right, or interval censoring. For some density in WinBUGS we can write `x[i]` $\sim$ `ddensity(theta)I(,)` with `I(a[i],)` or `I(,b[i])` or `I(a[i],b[i])`, and with `x[i]` as unknown 'NA' data point.
(F) logit-link is used for mapping real valued expressions of explanatory variables to probabilities between zero and one. For example: logit(p_i)= $\log(p_i/(1-p_i)) = \alpha_0 + \alpha_1 X_{i,1} + ... + \alpha_p X_{i,p}$ with linear predictor.

2. (6 points):
Explain 'pooled estimates', 'separate estimates' and how hierarchical models are related to these.

When analyzing data from from groups, we could assume a common parameter over all groups and therefore all group observations would contribute equally to the common parameter estimate. This pooled estimate does not account for possible important differences between groups. In contrast, separate estimates would be computed separately for each group. Separate estimates do not account for the general information drawn from other groups which may be useful, especially when dealing with groups that have very limited (or even missing) data. Hierarchical models are used to draw information over groups, but acknowledging group structures, balancing between pure separate and pure pooled estimates.

3. (6 points):
Explain the model(s) behind the following BUGS implementation(s):

```
  model{
  mu[1] ~ dnorm(0,0.001); mu[2] ~ dnorm(0,0.001)
  T[1:2,1:2] <- inverse(C[1:2,1:2])
  C[1,1] <- s12 # sigma1 squared
  s12 ~ dgamma(0.01,0.01)
  C[2,2] <- s22 # sigma2 squared
  s22 ~ dgamma(0.01,0.01)
  C[1,2] <- C[2,1]
  C[2,1] <- rho*sqrt(C[1,1]*C[2,2])
  rho ~ dunif(-1,1)
  # alternative prior for T:
  #T[1:2,1:2] ~ dwish(R[1:2,1:2],2)
  # the R in BUGS-Wishart needs to be a precision matrix, not covariance matrix
  #R[1,1] <- 0.001
  #R[1,2] <- 0
  #R[2,1] <- 0
  #R[2,2] <- 0.001
  for(i in 1:N){
  logtimetemp[i,1] <- log(timetemp[i,1])
  logtimetemp[i,2] <- log(timetemp[i,2])
  logtimetemp[i,1:2] ~ dmnorm(mu[1:2],T[1:2,1:2]) }
  logttpred[1:2] ~ dmnorm(mu[1:2],T[1:2,1:2])
  ttpred[1]<-exp(logttpred[1])
  ttpred[2]<-exp(logttpred[2])
  prisk[1] <- step(ttpred[1]-50)*step(ttpred[2]-175)
  prisk[2] <- step(ttpred[1]-50)*step(ttpred[2]-132)*step(175-ttpred[2])
  prisk[3] <- step(ttpred[1]-50)*step(132-ttpred[2])
  prisk[4] <- step(50-ttpred[1])
  }
 list(N=19, timetemp=structure(.Data=c(
 43, 179,
 44, 217,
 47, 206,
 ... ... ...
```

The model is a 2D-model for measurements $\log(X)$ ('logtimetemp') with 2D-normal distribution. This distribution has unknown mean vector and unknown covariance matrix to be estimated. An uninformative normal distribution is placed for the two mean parameters. An uninformative Wishart prior is constructed for the precision matrix (alternative), or the precision matrix is constructed from its elements, placing priors on variances and correlation-parameter. Finally, the model is used to compute predictive probabilities that the 2D observations (in original scale X) would be in different ranges, specified by step-functions. ($X_1 > 50$ & $X_2 > 175$ etc...)

4. (6 points):
Write a BUGS model code for the following posterior distribution, where the data are measurements x_i and group indicators z_i (=M/F), $i = 1, \ldots, n$. The model for x_i is normal distribution.

$$\pi(p, \mu_{\mathrm{M}}, \mu_{\mathrm{F}}, \sigma_{\mathrm{M}}, \sigma_{\mathrm{F}} \mid x_1, \ldots, x_n, z_1, \ldots, z_n) \propto$$

$$\pi(p)\pi(\mu_{\mathrm{M}}, \mu_{\mathrm{F}})\pi(\sigma_{\mathrm{M}}, \sigma_{\mathrm{F}}) \prod_{i=1}^{n} \pi(z_i \mid p)\pi(x_i \mid \mu_{z_i}, \sigma_{z_i})$$

including simulation of prediction from $\pi(x^*, z^* \mid \text{data})$. Use uninformative priors.

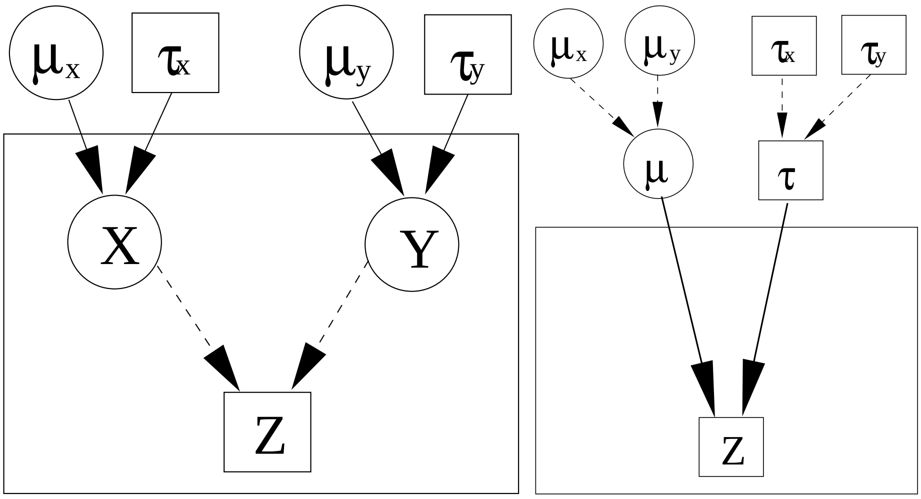
```
model{
p[1] ~ dunif(0,1); p[2]<- 1-p[1]
for(i in 1:20){
x[i,2] ~ dcat(p[1:2]);
# could also make use of dbern, e.g. in this way:
# x[i,2] <- xb[i]+1; xb[i] ~ dbern(p[1])
x[i,1] ~ dnorm(mu[x[i,2]],tau[x[i,2]]) }
mu[1] ~ dnorm(0,0.0000001)
mu[2] ~ dnorm(0,0.0000001)
s[1] ~ dunif(0,1000); tau[1] <-1/(s[1]*s[1])
s[2] ~ dunif(0,1000); tau[2] <-1/(s[2]*s[2])
}
```

5. (6 points):
What is the problem in the following BUGS code and how to fix it? Write the corrected code. Draw the original model and your fixed model as a DAG. Could the original version be implemented in some other way of MCMC than with BUGS?

```
model{
for (i in 1:n) {
z[i] <- log(x[i]+y[i])
x[i] ~ dnorm(mux,taux)
y[i] ~ dnorm(muy,tauy)
}
mux ~ dnorm(0,0.001) # uninformative prior for mux
muy ~ dnorm(1,2) # informative prior for muy is assumed
taux <- 1; tauy <- 1 # precisions assumed known
}
list(z=c(0.5,0.9,1.6,2.1,0.4,0.1,0.4,0.2))
```

Literally taken, it is not possible to infer exactly what model has been *attempted*, based on an erroneous code only. However, the basic problem in this case is the following: This code exemplifies the syntax error of trying to assign data value to a variable that is a deterministic function of two other variables. Data should always be assigned to stochastic variables. Therefore, we could solve what the conditional distribution for $\exp(z)$ is. For this, we recall the calculus for the sum of two normally distributed variables. (Mean is the sum of means, variance is the sum of variances). Note that there is an informative prior involved about μ_y so that we cannot eliminate this. Without additional information, the parameters would be unidentifiable from data z which alone is not sufficient to separate μ_x from μ_y. Also, we could note that the original construction would be possible to implement by general Metropolis-Hastings algorithm if implemented by a suitable proposal distribution that is guaranteed to propose such combinations of x and y that always satisfy the constraint due to observation z. Proposing x and y independently would violate the constraint immediately, because once a value for x is proposed, then y is already determined, and vice versa. This does not follow the simple single-site updating scheme in BUGS where we could generate x from a distribution, given y, and vice versa. Hence, data values are allowed to be assigned for stochastic nodes.

```
model{
mu <- mux+muy;  tau <- 1/(1/taux+1/tauy)
mux ~ dnorm(0,0.001); muy ~ dnorm(0,2)
taux <- 1/varx; varx <- 1 ; tauy <- 1/vary; vary <- 1
for (i in 1:n) {
ez[i] ~ dnorm(mu,tau);
ez[i] <- exp(z[i])
}}
```


μ_x
τ_x
μ_y
τ_y
X
Y
Z
μ_x
μ_y
τ_x
τ_y
μ
τ
Z