Stat 8931 Spin Glass Homework Solution, Part 1

Charles J. Geyer

December 14, 2005

1 Setup

Read betas.

```
> foo <- try(scan("betas.txt"))
> if (inherits(foo, "try-error")) {
+     write(c(br, bd), file = "betas.txt")
+     foo <- scan("betas.txt")
+ }
> n <- sqrt(length(foo)/2)
> br <- matrix(foo[1:n^2], n, n)
> bd <- matrix(foo[n^2 + 1:n^2], n, n)
```

Set tau.

```
> tau <- 0.2
```

Setup.

```
> i <- matrix(seq(1, n^2), n, n)
> ir <- cbind(i[, -1], i[, 1])
> il <- cbind(i[, n], i[, -n])
> id <- rbind(i[-1, ], i[1, ])
> iu <- rbind(i[n, ], i[-n, ])
> x <- matrix(1, n, n)
```

```
> co <- (i + (1 - n%%2) * col(i))%%2
> ico0 <- i[co == 0]
> ico1 <- i[co == 1]
```

Define update.

```
> block.gibbs <- function(x, tau) {
+     foo <- x[ir] * br + x[id] * bd + x[il] * br[il] + x[iu] *
+         br[iu]
+     foo <- foo/tau
+     p <- 1/(1 + exp(-2 * foo))
+     x[ico0] <- as.numeric(runif(n^2/2) < p[ico0]) * 2 - 1
+     foo <- x[ir] * br + x[id] * bd + x[il] * br[il] + x[iu] *
+         br[iu]
+     foo <- foo/tau
+     p <- 1/(1 + exp(-2 * foo))
+     x[ico1] <- as.numeric(runif(n^2/2) < p[ico1]) * 2 - 1
+     return(x)
+ }
```

Set random seed (if we always want the same results, otherwise omit).

```
> set.seed(42)
```

2 Run One

```
> nbatch <- 1000
> blen <- 1000
```

```
> batch <- matrix(NA, nbatch, 3 * n^2)
> for (ibatch in 1:nbatch) {
+     xbatch <- rep(0, 3 * n^2)
+     for (iiter in 1:blen) {
+         x <- block.gibbs(x, tau)
+         xbatch <- xbatch + as.numeric(c(x, x * x[ir], x * x[id]))
+     }
+     batch[ibatch, ] <- xbatch/blen
+ }
> mu <- apply(batch, 2, mean)
> mcse <- apply(batch, 2, sd)/sqrt(nbatch)
> xmu <- matrix(mu[1:n^2], n, n)
> xmcse <- matrix(mcse[1:n^2], n, n)
> xxrmu <- matrix(mu[n^2 + 1:n^2], n, n)
> xxrmcse <- matrix(mcse[n^2 + 1:n^2], n, n)
> xxdmu <- matrix(mu[2 * n^2 + 1:n^2], n, n)
> xxdmcse <- matrix(mcse[2 * n^2 + 1:n^2], n, n)
> round(xmu, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] -0.854 -0.854 -0.841  0.854 -0.854 -0.861
[2,] -0.906 -0.834  0.841 -0.854 -0.854 -0.972
[3,] -0.682  0.396  0.396 -0.856  0.943 -0.974
[4,]  0.681  0.396  0.007 -0.977  0.977 -0.977
[5,] -0.915 -0.873 -0.864 -0.853 -0.977 -0.988
[6,] -0.989 -0.854 -0.854  0.854 -0.989  0.989

> round(xmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.016 0.016 0.016 0.016 0.016 0.016
[2,] 0.009 0.016 0.016 0.016 0.016 0.006
[3,] 0.006 0.012 0.012 0.016 0.006 0.006
[4,] 0.006 0.012 0.008 0.006 0.006 0.006
[5,] 0.008 0.014 0.015 0.016 0.006 0.005
[6,] 0.005 0.016 0.016 0.016 0.004 0.005

> round(xxrmu, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  1.000  0.984 -0.985 -1.000  0.993  0.993
[2,]  0.902 -0.966 -0.985  1.000  0.839  0.911
[3,] -0.628  0.938 -0.521 -0.880 -0.953  0.687
[4,]  0.694 -0.318  0.016 -1.000 -1.000 -0.683
[5,]  0.932  0.984  0.984  0.830  0.965  0.927
[6,]  0.864  1.000 -1.000 -0.862 -0.998 -1.000

> round(xxrmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.000 0.001 0.001 0.000 0.001 0.001
[2,] 0.008 0.001 0.001 0.000 0.017 0.009
[3,] 0.006 0.000 0.005 0.012 0.005 0.006
[4,] 0.006 0.010 0.008 0.000 0.000 0.006
[5,] 0.006 0.001 0.002 0.017 0.008 0.006
[6,] 0.016 0.000 0.000 0.016 0.001 0.000

> round(xxdmu, 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  0.923  0.980 -1.000 -1.000  1.000  0.845
[2,]  0.594 -0.541  0.514  0.998 -0.878  0.995
[3,] -0.394  0.938 -0.318  0.833  0.953  0.996
[4,] -0.705 -0.498 -0.142  0.830 -1.000  0.965
[5,]  0.926  0.969  0.985 -0.999  0.969 -0.998
[6,]  0.864  1.000  0.984  1.000  0.863 -0.871
```

```
> round(xxdmcse, 3)
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.008 0.000 0.000 0.000 0.000 0.016
[2,] 0.015 0.005 0.005 0.000 0.013 0.001
[3,] 0.012 0.000 0.010 0.017 0.005 0.000
[4,] 0.004 0.004 0.008 0.017 0.000 0.008
[5,] 0.007 0.003 0.002 0.000 0.007 0.000
[6,] 0.016 0.000 0.001 0.000 0.016 0.015
```

The maximum of all the MCSE is 0.0175, so (assuming the MCSE were correct, which they are not) we only need to run about 17.5^2 times longer, that is `nbatch * blen` equal to

```
> ntodo <- nbatch * blen * (max(xmcse, xxrmcse, xxdmcse)/0.001)^2
> ntodo
```

```
[1] 304807113
```

```
> ntodo.exp <- floor(log10(ntodo))
> ntodo.frac <- round(ntodo/10^ntodo.exp, 1)
```

`nbatch * blen` equal to 3×10^8, divided in any way so the batches are long enough.

Examination of the autocorrelation plots for these shows that `blen` should be at least 100 times longer.

Moreover, the means of the X_i are completely wrong. So none of the MCSE are worth anything and all of the numbers in the summary above are wrong. We must run longer.

3 Run Two

```
> blen <- 100 * blen
> nbatch <- nbatch/5
> batch <- matrix(NA, nbatch, 3 * n^2)
> for (ibatch in 1:nbatch) {
+     xbatch <- rep(0, 3 * n^2)
+     for (iiter in 1:blen) {
+         x <- block.gibbs(x, tau)
+         xbatch <- xbatch + as.numeric(c(x, x * x[ir], x * x[id]))
+     }
+     batch[ibatch, ] <- xbatch/blen
+ }
> mu <- apply(batch, 2, mean)
> mcse <- apply(batch, 2, sd)/sqrt(nbatch)
> xmu <- matrix(mu[1:n^2], n, n)
> xmcse <- matrix(mcse[1:n^2], n, n)
> xxrmu <- matrix(mu[n^2 + 1:n^2], n, n)
> xxrmcse <- matrix(mcse[n^2 + 1:n^2], n, n)
> xxdmu <- matrix(mu[2 * n^2 + 1:n^2], n, n)
> xxdmcse <- matrix(mcse[2 * n^2 + 1:n^2], n, n)
> round(xmu, 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,] -0.119 -0.119 -0.117  0.119 -0.119 -0.117
[2,] -0.093 -0.117  0.117 -0.119 -0.119 -0.063
[3,] -0.065  0.070  0.070 -0.119  0.077 -0.061
[4,]  0.065  0.070  0.030 -0.060  0.060 -0.060
[5,] -0.084 -0.108 -0.113 -0.119 -0.060 -0.062
[6,] -0.061 -0.119 -0.119  0.119 -0.061  0.061
```

```
> round(xmcse, 3)
```

```
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.068 0.068 0.067 0.068 0.068 0.067
[2,] 0.062 0.066 0.067 0.068 0.068 0.065
[3,] 0.046 0.037 0.037 0.068 0.062 0.065
[4,] 0.046 0.037 0.020 0.065 0.065 0.065
[5,] 0.062 0.065 0.066 0.068 0.065 0.066
[6,] 0.067 0.068 0.068 0.068 0.067 0.067
```

```
> round(xxrmu, 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  0.999  0.984 -0.984 -1.000  0.986  0.986
```

```
[2,]  0.838 -0.967 -0.985  1.000  0.696  0.835
[3,] -0.580  0.940 -0.543 -0.775 -0.910  0.641
[4,]  0.645 -0.239  0.081 -1.000 -1.000 -0.632
[5,]  0.871  0.970  0.969  0.681  0.968  0.863
[6,]  0.706  1.000 -1.000 -0.705 -0.998 -1.000
```

```
> round(xxrmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.000 0.000 0.000 0.000 0.002 0.002
[2,] 0.018 0.000 0.000 0.000 0.040 0.022
[3,] 0.013 0.001 0.007 0.030 0.012 0.013
[4,] 0.014 0.023 0.019 0.000 0.000 0.014
[5,] 0.016 0.004 0.004 0.042 0.003 0.017
[6,] 0.042 0.000 0.000 0.042 0.000 0.000
```

```
> round(xxdmu, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  0.858  0.981 -1.000 -1.000  1.000  0.710
[2,]  0.466 -0.562  0.536  0.996 -0.771  0.990
[3,] -0.298  0.940 -0.239  0.686  0.911  0.992
[4,] -0.693 -0.495 -0.212  0.681 -1.000  0.968
[5,]  0.860  0.940  0.970 -0.999  0.973 -0.997
[6,]  0.707  1.000  0.984  1.000  0.706 -0.721
```

```
> round(xxdmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.019 0.000 0.000 0.000 0.000 0.038
[2,] 0.036 0.007 0.007 0.001 0.030 0.001
[3,] 0.027 0.001 0.023 0.042 0.012 0.001
[4,] 0.003 0.001 0.019 0.042 0.000 0.003
[5,] 0.018 0.008 0.004 0.000 0.002 0.000
[6,] 0.042 0.000 0.000 0.000 0.042 0.040
```

The maximum of all the MCSE is 0.0678, so we only need to run about 67.8^2 times longer, that is `nbatch * blen` equal to

```
> ntodo <- nbatch * blen * (max(xmcse, xxrmcse, xxdmcse)/0.001)^2
> ntodo
```

```
[1] 92014970274
```

```
> ntodo.exp <- floor(log10(ntodo))
> ntodo.frac <- round(ntodo/10^ntodo.exp, 1)
```

`nbatch * blen` equal to 9.2×10^{10}, divided in any way so the batches are long enough.

Examination of the autocorrelation plots for these shows that `blen` should be at least 5 times longer.

Moreover, the means of the X_i are completely wrong. So none of the MCSE are worth anything and all of the numbers in the summary above are wrong. We must run longer.

4 Run Three

```
> blen <- 10 * blen
> batch <- matrix(NA, nbatch, 3 * n^2)
> for (ibatch in 1:nbatch) {
+     xbatch <- rep(0, 3 * n^2)
+     for (iiter in 1:blen) {
+         x <- block.gibbs(x, tau)
+         xbatch <- xbatch + as.numeric(c(x, x * x[ir], x * x[id]))
+     }
+     batch[ibatch, ] <- xbatch/blen
+ }
> mu <- apply(batch, 2, mean)
> mcse <- apply(batch, 2, sd)/sqrt(nbatch)
> xmu <- matrix(mu[1:n^2], n, n)
> xmcse <- matrix(mcse[1:n^2], n, n)
> xxrmu <- matrix(mu[n^2 + 1:n^2], n, n)
> xxrmcse <- matrix(mcse[n^2 + 1:n^2], n, n)
> xxdmu <- matrix(mu[2 * n^2 + 1:n^2], n, n)
> xxdmcse <- matrix(mcse[2 * n^2 + 1:n^2], n, n)
> round(xmu, 3)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]  [,6]
[1,]  0.036  0.036  0.035 -0.036  0.036 0.037
[2,]  0.050  0.035 -0.035  0.036  0.036 0.067
[3,]  0.040 -0.012 -0.012  0.036 -0.059 0.068
[4,] -0.040 -0.012  0.010  0.068 -0.068 0.068
```

```
[5,]  0.054  0.042  0.039  0.036  0.068  0.069
[6,]  0.070  0.036  0.036 -0.036  0.070 -0.070

> round(xmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.049 0.049 0.048 0.049 0.049 0.049
[2,] 0.046 0.048 0.048 0.049 0.049 0.046
[3,] 0.034 0.026 0.026 0.049 0.046 0.046
[4,] 0.034 0.026 0.010 0.046 0.046 0.046
[5,] 0.046 0.048 0.048 0.049 0.046 0.047
[6,] 0.048 0.049 0.049 0.049 0.047 0.048

> round(xxrmu, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  0.999  0.984 -0.985 -1.000  0.986  0.986
[2,]  0.839 -0.967 -0.985  1.000  0.700  0.837
[3,] -0.581  0.940 -0.543 -0.778 -0.911  0.642
[4,]  0.646 -0.241  0.079 -1.000 -1.000 -0.633
[5,]  0.873  0.971  0.969  0.685  0.968  0.865
[6,]  0.712  1.000 -1.000 -0.711 -0.998 -1.000

> round(xxrmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.000 0.000 0.000 0.000 0.001 0.001
[2,] 0.009 0.000 0.000 0.000 0.021 0.011
[3,] 0.007 0.000 0.004 0.015 0.006 0.007
[4,] 0.007 0.012 0.010 0.000 0.000 0.007
[5,] 0.008 0.002 0.002 0.022 0.001 0.009
[6,] 0.022 0.000 0.000 0.022 0.000 0.000

> round(xxdmu, 3)

       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
[1,]  0.860  0.981 -1.000 -1.000  1.000  0.714
[2,]  0.470 -0.562  0.535  0.996 -0.773  0.990
[3,] -0.301  0.940 -0.241  0.690  0.913  0.993
[4,] -0.694 -0.496 -0.210  0.685 -1.000  0.968
[5,]  0.863  0.941  0.970 -0.999  0.973 -0.997
[6,]  0.712  1.000  0.984  1.000  0.711 -0.726
```

```
> round(xxdmcse, 3)

      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 0.010 0.000 0.000 0.000 0.000 0.020
[2,] 0.018 0.004 0.004 0.000 0.016 0.001
[3,] 0.014 0.000 0.012 0.021 0.006 0.001
[4,] 0.001 0.000 0.010 0.022 0.000 0.001
[5,] 0.009 0.004 0.002 0.000 0.001 0.000
[6,] 0.022 0.000 0.000 0.000 0.022 0.021
```

The maximum of all the MCSE is 0.0491, so we only need to run about 49.1^2 times longer, that is `nbatch * blen` equal to

```
> ntodo <- nbatch * blen * (max(xmcse, xxrmcse, xxdmcse)/0.001)^2
> ntodo

[1] 482904203493

> ntodo.exp <- floor(log10(ntodo))
> ntodo.frac <- round(ntodo/10^ntodo.exp, 1)
```

`nbatch * blen` equal to 4.8×10^{11}, divided in any way so the batches are long enough.