

Paper 4911-2020

## **A Level-Headed Approach to the METHOD = FASTQUAD Option in the GLIMMIX Procedure**

Sidney J. Hann, Grand Valley State University and Spectrum Health Office of Research and Education

### ABSTRACT

The Gaussian quadrature algorithm approximates the likelihood of a generalized linear model with multiple nested levels using the Gauss-Hermite quadrature method of integration. While the approximations are close to exact, the calculation of these approximations in SAS are computationally demanding both in memory and time. To address these intense requirements, the SAS/STAT 14.1 update added the multilevel adaptive Gaussian quadrature algorithm of Pinheiro and Chao (2006) to PROC GLIMMIX using the METHOD = FASTQUAD option. Pinheiro and Chao's algorithm reduces the number of integrations required while SAS processes the estimation of the conditional log-likelihoods required for this modelling technique. The reduction of integrations reduces computation memory and time. Using a public dataset, this paper will examine the process of creating a multilevel model using adaptive Gaussian quadrature while comparing the run times of single level and multilevel adaptive Gaussian quadrature.

### INTRODUCTION

Adaptive Gaussian Quadrature algorithms are based on the Gauss-Hermite Quadrature rule defined in the Handbook of Mathematical Functions of Abramowitz and Stegun (1964). The single level Adaptive Gaussian Quadrature algorithm of Pinheiro and Bates (1995) and multilevel Adaptive Gaussian Quadrature algorithm of Pinheiro and Chao (2006) both use importance sampling to approximate the log-likelihoods in Generalized Linear Mixed Models (GLMMs). Using the approximations of the log-likelihoods in GLMMs the maximum likelihood estimator for the parameters in the model may be derived. Multilevel GLMMs, such as students being nested within teachers, require exponentially more integrations as the number of levels increases. The increase in integrations can quickly cause the statistical software and the computers processing programs to quickly run out of computational memory.

With the release of SAS/STAT® 14.1 the GLIMMIX procedure may now use the multilevel Adaptive Gaussian Quadrature algorithm. To use the multilevel algorithm instead of the single level algorithm, the FASTQUAD sub-option must be specified alongside METHOD=QUAD. With the FASTQUAD sub-option SAS is now able to process multilevel GLMMs more efficiently through integration reduction. The reduction in number of integrations leads to faster processing speeds and lower memory requirements.

The largest advantage to the addition of the FASTQUAD sub-option is reduced computation memory and time. To adduce this advantage, we compared computation memory and run time between single level and multilevel Adaptive Gaussian Quadrature using a created dataset based on a SAS Support simulation (SAS Support, 2017). Due to a lack of memory the application of single level Adaptive Gaussian Quadrature to a multilevel model was unable to process. In place of the single level model, the number of quadrature points used in multilevel Adaptive Gaussian Quadrature were compared at levels of 3, 5, and 7. The increase in quadrature points leads to an increase in integrations required when approximating the log-likelihoods. This paper will discuss the theoretical differences

between single level and multilevel Adaptive Gaussian Quadrature, the implementation of the FASTQUAD sub-option in the GLIMMIX procedure, and compare computational memory and run time requirements for using 3, 5 and 7 quadrature points for approximation.

## ADAPTIVE GAUSSIAN QUADRATURE

### WHAT IS ADAPTIVE GAUSSIAN QUADRATURE

To understand the differences between single level and multilevel Adaptive Gaussian Quadrature we must start at its roots. Let  $p(x)$  be a probability density function (pdf) and  $f(x)$  be a function which the pdf is integrated against. Gauss-Hermite quadrature is appropriate when the density of  $p(x)$  has kernel  $e^{-x^2}$  and the limits of integration span all real numbers. From formula 25.4.46 of Abramowitz and Stegun (1964), the Gauss-Hermite quadrature rule from which Adaptive Gaussian Quadrature is based follows the form

$$\int_{-\infty}^{\infty} f(x)p(x)dx = \int_{-\infty}^{\infty} e^{-x^2} f(x)dx \approx \sum_{i=1}^N w_i f(x_i)$$

Where  $N$  represents the number of quadrature points,  $w_i$  and  $x_i$  represent the weights and abscissas. These weights and abscissas are based on areas of high densities in the kernel function. Weights and densities for common kernels, such as the normal distribution, are calculated following the methods described in Hammer, P. C., & Wymore, A. W. (1957).

In single level Generalized Linear Mixed Models (GLMMs) the marginal distribution of the random effects follows a  $\sim N(0, \sigma^2)$  distribution. Using the structure of the integrand in GLMMs, Adaptive Gaussian Quadrature transforms the calculation of multiple Gaussian quadrature rules into repetitive applications of one-dimensional Gaussian quadrature rules. In this case, the Gaussian quadrature rule is a deterministic version of a Monte Carlo integration algorithm in which random samples of the random effects,  $b_i$ , are generated from the  $\sim N(0, \sigma^2)$  distribution; and the samples and weights are fixed beforehand instead of randomly chosen. (Pinheiro and Chao 2006). To obtain increased efficiency, an importance sampling version of Gaussian quadrature is utilized. This importance sampling version, defined by Pinheiro and Bates (1995), is known as *Adaptive Gaussian Quadrature*.

### SINGLE LEVEL ADAPTIVE GAUSSIAN QUADRATURE

In single level Adaptive Gaussian Quadrature for GLMMs the likelihood estimation, for a given group  $i$ , relies on the marginal density of the response vector  $y_i$ . The marginal density of  $y_i$  is obtained by integrating the joint likelihood function of  $(y_i, b_i)$  with respect to the random effects vector  $b_i$ . For the purposes of this paper, let  $\beta$  represent the vector of fixed effects,  $X_i$  is the fixed effects regression matrix,  $Z_i$  is the random effects matrix, and  $h = g^{-1}$  is the inverse link function. The joint density of  $(y_i, b_i)$  when assuming an exponential link function in canonical form for appropriate functions  $a(\cdot)$ ,  $c(\cdot)$ , and  $d(\cdot)$  is

$$p(y_i, b_i) = \frac{e^{\left[ \frac{y_i'(X_i\beta + Z_i b_i) - d(X_i\beta + Z_i b_i)'1}{a(\theta)} + c(y_i, \theta)' - \frac{b_i'(\sigma^2)^{-1}b_i}{2} \right]}}{\left[ (2\pi)^{\frac{q}{2}} |(\sigma^2)|^{\frac{1}{2}} \right]}$$

where  $q$  is equal to the number of random effects.

As previously stated, the marginal density of  $y_i$ , which the likelihood estimation relies upon, is obtained by integrating the joint likelihood function of  $(y_i, \mathbf{b}_i)$  with respect to the random effects vector  $\mathbf{b}_i$ . The marginal GLMM likelihood in the single level case can be written as

$$p(y_i) = \int p(y_i, \mathbf{b}_i) d\mathbf{b}_i = (2\pi)^{-\frac{q}{2}} |\sigma^2|^{-\frac{1}{2}} \int e^{g(\beta, \sigma^2, \emptyset, y_i, \mathbf{b}_i)} d\mathbf{b}_i$$

and defining  $g(\beta, \sigma^2, \emptyset, y_i, \mathbf{b}_i)$  as

$$g(\beta, \sigma^2, \emptyset, y_i, \mathbf{b}_i) = \frac{y_i'(X_i\beta + Z_i\mathbf{b}_i) - d(X_i\beta + Z_i\mathbf{b}_i)'1}{a(\emptyset)} + c(y_i, \emptyset)' - \frac{\mathbf{b}_i'(\sigma^2)^{-1}\mathbf{b}_i}{2}$$

Through a second-order Taylor expansion  $g(\beta, \sigma^2, \emptyset, y_i, \mathbf{b}_i)$  is approximated around the value of  $\mathbf{b}_i$  which maximizes  $g(\cdot, \mathbf{b}_i)$ . The second-order derivative of  $g(\cdot, \mathbf{b}_i)$  is negative definite resulting in  $g(\cdot, \mathbf{b}_i)$  being a strictly concave function of  $\mathbf{b}_i$ . This strictly concave function allows the estimation of a unique point of maximum,  $\hat{\mathbf{b}}_i$ , corresponding to the value of the second-order derivative of  $g(\cdot, \mathbf{b}_i)$  is equal to zero. The calculation of convergence on  $\mathbf{b}_i$  can be transferred to a least squares problem from which  $\mathbf{R}_i$  is defined as part of the Q-R decomposition in Thisted (1988).

We can now approximate the importance distribution the Adaptive Gaussian Quadrature rule utilizes. In a single level GLMM the integrand is proportional to  $e^{g(\beta, \sigma^2, \emptyset, y_i, \mathbf{b}_i)}$  which is approximated by a  $\sim N(\hat{\mathbf{b}}_i, \mathbf{R}_i^{-1}\mathbf{R}_i^{-T})$  density. The use of this importance distribution centers the grid of abscissas in the  $\mathbf{b}_i$  scale around the conditional nodes  $\hat{\mathbf{b}}_i$  and uses  $\mathbf{R}_i$  for scaling. To put everything together, let  $\mathbf{z}_j, w_j$  for  $j = 1, \dots, N_{GQ}$  denote the abscissas and weights for the one-dimensional Gaussian Quadrature rule with  $N_{GQ}$  points based on the  $\sim N(0, 1)$  kernel. Define the following:

- $\mathbf{z}_j = (z_{j1}, \dots, z_{jq})^T$
- $\tilde{\mathbf{b}}_{ij} = \hat{\mathbf{b}}_i + \mathbf{R}_i^{-1}\mathbf{z}_j$
- $W_j = e^{\|\mathbf{z}_j\|^2} \prod_{k=1}^q w_{jk}$

The single level Adaptive Gaussian Quadrature rule is then

$$\int e^{g(\beta, \sigma^2, \emptyset, y_i, \hat{\mathbf{b}}_i + \mathbf{R}_i^{-1}\mathbf{z}) + \frac{\|\mathbf{z}\|^2}{2}} e^{-\frac{\|\mathbf{z}\|^2}{2}} d\mathbf{z} \cong (2\pi)^{\frac{q}{2}} |\mathbf{R}_i|^{-1} \sum_{j_1=1}^{N_{GQ}} \dots \sum_{j_q=1}^{N_{GQ}} [e^{g(\beta, \sigma^2, \emptyset, y_i, \tilde{\mathbf{b}}_{ij})} W_j]$$

and the single level GLMM Adaptive Gaussian Quadrature approximation of the log likelihood function is

$$l_{AGQ}(\beta, \sigma^2, \emptyset | \mathbf{y}) = M \log |\Delta| + \sum_{i=1}^M \left\{ -\log |\mathbf{R}_i| + \log \left[ \sum_j^{N_{GQ}} e^{[g(\beta, \sigma^2, \emptyset, y_i, \tilde{\mathbf{b}}_{ij})] W_j} \right] \right\}$$

Where M is the number of groups and  $\Delta$  is a precision matrix from the least squares transformation. [3]

## MULTILEVEL ADAPTIVE GAUSSIAN QUADRATURE

Multilevel Adaptive Gaussian Quadrature for GLMMs allows computational feasibility of approximating the log-likelihood function from data with a nested model structure. Extending the structure of multilevel GLMMs to the Adaptive Gaussian Quadrature, the approximation of the log-likelihood for the  $i^{th}$  group in a two-level GLMM becomes

$$\begin{aligned}
 l_{AGQ2}(\beta, \sigma_1^2, \sigma_2^2, \emptyset | y_i) &= \log|\Delta_1| - \log|R_{11(i)}| + M_i \log|\Delta_2| - \sum_{j=1}^{M_i} \log|R_{22(ij)}| \\
 &+ \log \sum_{k=1}^{NGQ} \left\{ \left[ e^{g_{21}(\beta, \sigma_1^2, \emptyset, y_i, \bar{b}_i^{(1)}(k))} W_k^{(1)} \right] \times \prod_{j=1}^{M_i} \sum_l^{NGQ} \left[ e^{g_{22}(\beta, \sigma_2^2, \emptyset, y_{ij}, \bar{b}_i^{(1)}(k), \bar{b}_{ij}^{(2)}(l,k))} W_l^{(2)} \right] \right\}
 \end{aligned}$$

Where the full data approximation is

$$l_{AGQ2}(\beta, \sigma_1^2, \sigma_2^2, \emptyset | y) = \sum_{i=1}^{M_i} l_{AGQ2}(\beta, \sigma_1^2, \sigma_2^2, \emptyset | y_i)$$

The approximation requires  $N_{GQ}^{q_1}$  evaluations of  $g_{21}(\cdot)$  and  $M_i N_{GQ}^{q_1+q_2}$  evaluations of  $g_{22}(\cdot)$  which increases linearly with the number of level two groups,  $M_i$ . If single level Adaptive Gaussian Quadrature were to be directly applied to a two-level model the approximation would require  $N_{GQ}^{q_1+M_i q_2}$  evaluations of the  $g(\cdot)$  function. The increase in computational speed and memory efficiency comes from using  $M_i N_{GQ}^{q_1+q_2}$  evaluations instead of  $N_{GQ}^{q_1+M_i q_2}$  evaluations of the respective  $g(\cdot)$  functions. This difference in evaluations is a product of theoretical change in the Q-R decompositions for the level 2 arrays in the data. The multilevel Adaptive Gaussian Quadrature algorithm reduces the number of conditional log-likelihood evaluations by inverting the least squares upper-triangular matrices; and the log-determinants used in the computations become the sum of the logarithms of the absolute diagonal values of the corresponding upper-triangular matrices. Pinheiro and Chao (2006) documents the application of inverting the upper-triangular matrices to the Adaptive Gaussian Quadrature rule in detail. [4]

## PROC GLIMMIX

The dataset used in the following section was simulated using SAS code which can be found in the appendix. The simulation creates a multilevel dataset with 20 schools, 30 teachers at each school, and between 16 and 30 students for each teacher. Each student has a value for the fixed effect x1. The response variable in the model follows a binomial response tracking whether a student had the flu. The following code applies a direct application of the single level Adaptive Gaussian Quadrature approximation to the multilevel simulated dataset using 3 quadrature points:

```

proc glimmix data=sim method=quad(qpoints=3);
  class school teacher;
  model Flu = x1 /dist=binomial link=logit;
  random int / subject=school;
  random int / subject=teacher(school);
run;

```

Unless the dataset used is small, SAS will return the following error message:

```
ERROR: Insufficient resources to perform adaptive quadrature with 3
quadrature points. METHOD=LAPLACE, corresponding to a single point, might
provide a computationally less intensive possibility.
```

The reason this error message displays is because the number of integrations the single level Adaptive Gaussian Quadrature approximation requires SAS to perform. In the context of the simulated data, this procedure asks SAS to perform  $N_{GQ}^{q_1+M_iq_2} = (3)^{1+20*1} = 10,460,353,203$  evaluations of the conditional log-likelihoods.

To use the multilevel Adaptive Gaussian Quadrature approximation the METHOD=QUAD option with the FASTQUAD sub-option must be used. The only difference in the code is the addition of the FASTQUAD sub-option shown below:

```
proc glimmix data=sim method=quad(fastquad qpoints=3);
  class school teacher;
  model Flu = x1 /dist=binomial link=logit;
  random int / subject=school;
  random int / subject=teacher(school);
run;
```

This time, we receive no error message because using the FASTQUAD sub-option asks SAS to perform  $N_{GQ}^{q_1+q_2} = (3)^{1+1} = 9$  evaluations of the conditional likelihoods. The output for the GLIMMIX procedure can be found in the appendix.

## MEMORY AND RUNTIME IMPROVEMENT

Since the single level Adaptive Gaussian Quadrature approximation may not be successfully run when applied to a multilevel model, the memory usage and runtimes of multilevel models with quadrature points of 3, 5 and 7 will be compared. The computer which processed each GLIMMIX procedure is a Lenovo™ Thinkpad® L480. The L480 uses an Inter® Core™ i5-8350U CPU with an internal clock of 1.70GHz, 8.00 GB of RAM, and a 64-bit Windows® 10 Operating System. SAS Enterprise Guide was used to run the SAS programs.

There was a total of 10 trials created following the format of the simulated dataset mentioned in the previous section. For each trial a GLIMMIX procedure with quadrature points of 3, 5, and 7 were ran one at a time. The runtime and memory usage statistics were recorded from the log after running the code:

```
options fullstimer;
```

After which the log information will now display:

```
NOTE: Convergence criterion (GCONV=1E-8) satisfied.
NOTE: At least one element of the gradient is greater than 1e-3.
NOTE: PROCEDURE GLIMMIX used (Total process time):
      real time           12.42 seconds
      user cpu time       11.98 seconds
      system cpu time     0.42 seconds
      memory              5897.03k
      OS Memory           23640.00k
      Timestamp           02/16/2020 07:27:51 PM
```

From which the `real time` and the `memory` was recorded for each of the procedures ran. A means procedure was used to summarize the results on the following page:

Quadrature points	Measure	Mean	Standard Deviation	Minimum	Median	Maximum
3	Runtime (s)	12.35	0.90	11.07	12.30	13.74
	Memory (k)	5619.55	8.33	5611.28	5618.05	5641.42
5	Runtime (s)	23.05	1.90	20.53	23.56	26.15
	Memory (k)	5617.71	11.97	5585.34	5619.70	5630.19
7	Runtime (s)	39.61	2.74	35.92	40.30	43.53
	Memory (k)	5620.87	4.25	5613.60	5619.67	5627.67

Table 1. MEANS Procedure: Runtime and Memory Statistics for FASTQUAD

The mean runtime of the GLIMMIX procedure in seconds increases with the number of quadrature points used. This can be explained by the increase in the number evaluations of the conditional log-likelihoods:

- 3 quadrature points:  $N_{GQ}^{q_1+q_2} = (3)^{1+1} = 9$  evaluations of the conditional log-likelihoods.
- 5 quadrature points:  $N_{GQ}^{q_1+q_2} = (5)^{1+1} = 25$  evaluations of the conditional log-likelihoods.
- 7 quadrature points:  $N_{GQ}^{q_1+q_2} = (7)^{1+1} = 49$  evaluations of the conditional log-likelihoods.

As expected, when SAS is asked to perform more evaluations of the conditional log-likelihoods the program takes longer to run.

The standard deviations for the memory requirements increased from 3 points to 5 but decreased from 5 to 7. While this result was not expected, it also is not surprising. This is due to the limitation of running multiple procedures using the same dataset. After a couple procedures are completed, results may be stored in either the operating system, within SAS, or another process on the computer. The caching of these results likely lead to subsequent procedures having lower variances.

## CONCLUSION

The addition of the FASTQUAD sub-option in the GLIMMIX procedure for SAS/STAT 14.1 allows the use of the multilevel Adaptive Gaussian Quadrature algorithm of Pinheiro and Chao (2006). The multilevel algorithm reduces the number of evaluations of the conditional log-likelihoods by inverting the least squares upper-triangular matrices. By decreasing the number of integrations of the conditional log-likelihoods SAS can process multilevel models faster and more efficiently. The reduction in integrations allows users to use additional quadrature points to increase accuracy of the evaluations performed **to derive the model's** parameter estimates.

## REFERENCES

1. Abramowitz, M., & Stegun, I. (1964). Handbook of mathematical functions with formulas, graphs, and mathematical tables. NEW YORK: DOVER PUBLICATIONS.
2. Hammer, P. C., & Wymore, A. W. (1957). Numerical evaluation of multiple integrals. I. *Mathematics of Computation*, 11(58), 59–67. doi: 10.1090/s0025-5718-1957-0087220-6
3. Pinheiro, J., & Bates, D. (1995). Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. *Journal of Computational and Graphical Statistics*, 4(1), 12-35. doi: 10.2307/1390625
4. Pinheiro, J., & Chao, E. (2006). Efficient Laplacian and Adaptive Gaussian Quadrature Algorithms for Multilevel Generalized Linear Mixed Models. *Journal of Computational and Graphical Statistics*, 15(1), 58-81. Retrieved from [www.jstor.org/stable/27594165](http://www.jstor.org/stable/27594165)
5. SAS Support (2017). Using the FASTQUAD suboption in PROC GLIMMIX to overcome long computing times or insufficient memory problems. Retrieved from <http://support.sas.com/kb/60/666.html>
6. Thisted, R. (1988). Elements of Statistical Computing. New York: Routledge, <https://doi.org/10.1201/9780203758212>

## ACKNOWLEDGMENTS

The entire Scholarly Activity and Scientific Support Team at Spectrum Health Office of Research and Education for their feedback on my paper, and the encouragement to submit to SAS Global. A special thanks to Jessi Parker for her mentorship and guidance on the paper and submission

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

**Sidney Hann**  
**Grand Valley State University**  
**Spectrum Health Office of Research and Education**  
[hann.sidney@gmail.com](mailto:hann.sidney@gmail.com)

## APPENDIX

### CODE

```
/* Create the simulation dataset.*/
data sim;
  do trial = 1 to 10; /* Create 10 trials w/ random seeds*/
    call streaminit(rand('uniform', 10000, 10000000));
    do school = 1 to 20; /* Create 20 schools */
      rs = rand('normal')*2;
      do teacher = 1 to 30; /*Create 20 teachers*/
        rt = rand('normal');
        do student = 1 to ceil(rand('uniform')*14)+15;
          x1 = rand('normal');
          /*Create linp relating the random effects*/
          linp = -3 + x1 + rs + rt;
          pi = 1/(1+exp(-linp));
          /* Create a Flu Response based on linp*/
          Flu = ranbin(0,1,pi);
          output;
        end;
      end;
    end;
  end;
run;

*Simulation code;

/* Creating a call function to quickly change trial number*/
%let trialnum = 1;

/* Telling SAS to print the full usage statistics in the log*/
options fullstimer;

/*Single level Adaptive Gaussian Quadrature applied to multilevel model*/
proc glimmix data=sim method=quad;
  where trial=&trialnum;
  class school teacher;
  model Flu = x1 /dist=binomial link=logit;
  random int / subject=school;
  random int / subject=teacher(school);
run;

/*Multilevel model with 3 quadrature points*/
proc glimmix data=sim method=quad(fastquad qpoints=3);
  where trial=&trialnum;
  class school teacher;
  model Flu = x1 /dist=binomial link=logit;
  random int / subject=school;
  random int / subject=teacher(school);
run;

/*Multilevel model with 5 quadrature points*/
proc glimmix data=sim method=quad(fastquad qpoints=5);
  where trial=&trialnum;
  class school teacher;
```



```

model Flu = x1 /dist=binomial link=logit;
random int / subject=school;
random int / subject=teacher(school);
run;

/*Multilevel model with 7 quadrature points*/
proc glimmix data=sim method=quad(fastquad qpoints=7);
  where trial=&trialnum;
  class school teacher;
  model Flu = x1 /dist=binomial link=logit;
  random int / subject=school;
  random int / subject=teacher(school);
run;

/*Importing dataset with the 10 trial's runtime and memory usage*/
proc import datafile='...\SimulationData.xlsx' replace out=simres;
run;

/*Creating a dataset, sorting and printing the means with the statistics
desired*/
data results;
  set simres(drop=trial);
run;
proc sort data=results;
  by Qpoints;
run;
proc means data=results nonobs min mean std median max maxdec=2;
  by Qpoints;
  var Runtime 'Memory(k)'\n;
run;

```

## OUTPUT

When the GLIMMIX procedure completes, the following output is given by default:

Model Information	
Data Set	WORK.SIM
Response Variable	Flu
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix Blocked By	school
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Multilevel Gauss-Hermite Quadrature
Degrees of Freedom Method	Containment

Output 1. Model Information

Class Level Information	
Class	Levels Values
school	20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
teacher	30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Output 2. Class Level Information

<b>Number of Observations Read</b>	13395
<b>Number of Observations Used</b>	13395

Output 3. Observations Read/Used

Dimensions	
G-side Cov. Parameters	2
Columns in X	2
Columns in Z per Subject	31
Subjects (Blocks in V)	20
Max Obs per Subject	699

Output 4. Matrix Dimensions

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	4
Lower Boundaries	2
Upper Boundaries	0
Fixed Effects	Not Profiled
Starting From	GLM estimates
Quadrature Points	3

Output 5. Optimization Information

Iteration History				
Iteration	Restarts	Evaluations	Objective Function	Change
0	0	12	7413.2197697	.
1	0	6	7343.8905837	69.32918608
2	0	6	7277.3244292	66.56615445
3	0	6	7261.5954038	15.72902546
4	0	6	7257.4708914	4.12451238
5	0	6	7255.3648882	2.10600315
6	0	6	7253.6717567	1.69313152
7	0	7	7252.986472	0.68528471
8	0	7	7252.9101116	0.07636042
9	0	12	7252.5764539	0.33365764
10	0	7	7252.4142731	0.16218086
11	0	7	7252.3886211	0.02565198
12	0	7	7252.387723	0.00089805
13	0	7	7252.3877108	0.00001219

Convergence criterion (GCONV=1E-8) satisfied.

Output 6. Iteration History

Fit Statistics	
-2 Log Likelihood	7252.39
AIC (smaller is better)	7260.39
AICC (smaller is better)	7260.39
BIC (smaller is better)	7264.37
CAIC (smaller is better)	7268.37
HQIC (smaller is better)	7261.17

Output 7. Model Fit Statistics for the Multilevel Model

Fit Statistics for Conditional Distribution	
-2 log L(Flu   r. effects)	6326.19
Pearson Chi-Square	8930.74
Pearson Chi-Square / DF	0.67

Output 8. Model Fit Statistics for the Conditional Distribution

Covariance Parameter Estimates			
Cov Parm	Subject	Estimate	Standard Error
Intercept	school	3.9210	1.2844
Intercept	teacher(school)	1.1050	0.1248

Output 9. Covariance Parameter Estimates

Type III Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
x1	1	12794	719.33	<.0001

Output 10. Type III Tests of Fixed Effects