Improving the Adaptive Gaussian Quadrature [*]

Authors: Freddy Hernández
– Escuela de Estadística, Universidad Nacional de Colombia,
Colombia (fhernanb@unal.edu.co)

Olga Usuga
– Departamento de Ingeniería Industrial, Universidad de Antioquia,
Colombia (olga.usuga@udea.edu.co)

Viviana Giampaoli
– Instituto de Matemática e Estatística, Universidade de São Paulo,
Brasil (vivig@ime.usp.br)

Abstract:

- The adaptive Gaussian quadrature technique is a method for obtaining multiple integrals by approximation and it is widely used in statistics, for instance, in estimation methods for mixed models; however, it can be prohibitive when the number of integrals or quadrature points is high. In this paper, we propose a method for improving the adaptive Gaussian quadrature approach, which involves eliminating those combinations of quadrature points that have little contribution to the approximation of the integral which decreases processing time without loss of accuracy. We used the proposed method to fit a Simplex mixed model related to a longitudinal ophthalmology study. With respect to the processing time required to fit the model, we found a reduction of at least 14%, and this reduction tends to increase as the number of quadrature points increases. We also conducted a simulation study with a Simplex mixed model with random intercept and slope, from this study we found that the time to estimate the model parameters with pruning approach was 7.94% lower than without pruning in all cases considered. This time reduction can be a considerable advantage when considering models with a large number of observations and multiple random effects.

Key-Words:

- *Adaptive Gaussian quadrature; Gaussian quadrature; Simplex mixed model*

AMS Subject Classification:

- 62-04.

[*] The opinions expressed in this text are those of the authors and do not necessarily reflect the views of any organization.

1. INTRODUCTION

Finding the solution of an integral in one or more dimensions is a task that arises frequently in various situations related to statistical analysis. When it is impossible to obtain the integral by analytical methods, an approximation of the integral is obtained with some methods as quadrature rules and Monte Carlo methods Kuonen (2003). For example, in mixed models, it is common to obtain likelihood functions that have integrals for which no analytic expressions are available; thus, numerical approximations are needed. In general, numerical approximations can be classified into those that are based on an approximation of the integrand, those based on an approximation of the data and those based on an approximation of the integral itself Molenberghs, Verbeke (2005). This paper addresses the approximation of an integral through the Gaussian quadrature method.

This paper is organized as follows. In the next section, we explain the Gaussian quadrature (GQ) and the adaptive Gaussian quadrature (AGQ) which are illustrated with two examples. In section 3, we present an approach for eliminating points that have a lower importance in AGQ. We present a table that shows the reduction in the number of points. We also present an application of AGQ with the pruning approach in the context of a Simplex mixed-effect model in section 4. In section 5 we present the results from a simulation study conducted to analyze the performance of the proposed methodology.

2. Gaussian Quadrature

The GQ is a very useful tool for approximating the integral of a function $g(x)$ over $\Re$ by a weighted sum, where the x variable is replaced by n quadrature points or nodes. Each quadrature point, denoted by p_i, is evaluated in the function, and the results are weighted by the quadrature weights w_i. The method can be summarized by the following expression:

$$\int_{\Re} g(x)dx \approx \sum_{i=1}^{n} g(p_i)\, \exp(p_i^2)\, w_i. \tag{2.1}$$

The set of n quadrature points $\boldsymbol{P} = \{p_1, p_2, \ldots, p_n\}$ corresponds to the roots of the Hermite polynomial given by

$$H_n(x) = (-1)^n e^{-x^2} \frac{d^n}{dx^n} e^{-x^2},$$

with its associated weights $\boldsymbol{W} = \{w_1, w_2, \ldots, w_n\}$ given by

$$w_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 \left[H_{n-1}(x_i)\right]^2}.$$

In R (R Core Team, 2016) we can find several functions for obtaining the quadrature points and the weights, some of which are `gauss.quad` from **statmod** package (Smyth et al., 2011), `ghq` from **glmmML** package Broström, Holmberg (2011) and `gaussHermiteData` from **fastGHQuad** package (Blocker, 2011). The quadrature order of the approximation is given by n, the number of quadrature points; the closer $g(x)$ is to $\exp(x^2)$, the better the precision of the GQ. We highlight some comments from Demidenko (2004) with respect to GQ. First, the farther the point is from zero, the lower the weight. Second, the points that are symmetric around zero have the same weight. Third, when the number of points n is greater than 13, it is unlikely that the precision can be improved because the weights rapidly approach zero with $n \to \infty$. For example, if we are interested in obtaining the quadrature points and weights for order $n = 5$, we could use the instruction `gauss.quad` from (Smyth et al., 2011) as `gauss.quad(n=5,kind="hermite")` to obtain the quadrature points $\boldsymbol{P} = \{-2.0202, -0.9586, 0, 0.9586, 2.0202\}$ and the corresponding weights $\boldsymbol{W} = \{0.0200, 0.3936, 0.9453, 0.3936, 0.0200\}$.

As a simple example, let us assume that our aim is to calculate a numerical approximation of the integral for the function $g_1(x) = \exp(-(x-1)^2)$ over $\Re$ using five quadrature points. Figure 1 shows the function $g_1(x)$ and the set of quadrature points is represented by black dots. Approximating the integral of $g_1(x)$ by the Equation (2.1) and using the quadrature points $\boldsymbol{P}$ and weights $\boldsymbol{W}$, we obtain the result 1.77134 while the exact value of the integral is 1.77245. In this example the approach generated a value close to the true value of the integral.

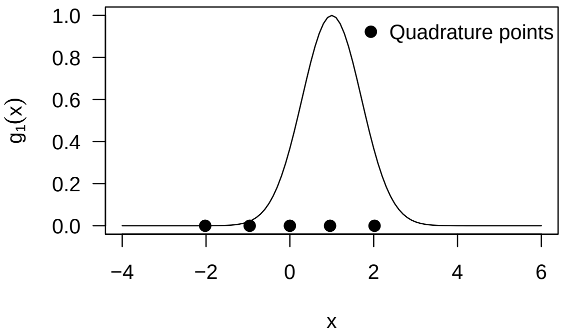

Figure 1: $g_1(x)$ function and quadrature points (represented by black dots).

Next, suppose that we are interested in obtaining a numerical approximation of the function $g_2(x) = \exp(-5(x-3)^2)$ over $\Re$ using five quadrature points. The left panel in Figure 2 shows the function $g_2(x)$ and the quadrature points p_i. By using the Equation (2.1), with $\boldsymbol{P}$ and $\boldsymbol{W}$, we find that the approximation to the integral is 0.00972 while the actual value is 0.79266. This inaccurate result arises because the quadrature points are far from the zone in which the function $g_2(x)$ has its highest density; therefore, this result leads to an imprecise approximation of the integral with the Equation (2.1). A simple solution is to increase the number of quadrature points in an attempt to cover the area in which the function has its highest density; however, this solution does not ensure a good approximation because the quadrature points away from zero have low weights and if these points are in the region of highest density for the function, they do not make a significant contribution because of their low importance. Furthermore, in some situations, an unnecessary increasing number of points could result in computational problems. To solve this situation, there is a version of the method called the adaptive Gauss-Hermite quadrature (AGQ) (Liu, Pierce, 1994; Pinheiro, Bates, 1995), which basically transforms the quadrature points, centering and spreading them around the maximum $\hat{x}$ of the function $\log(g(x))$. The transformation of the quadrature points p_i into the new points $p_i^\star$ is given by $p_i^\star = \sqrt{2}\hat{\sigma} p_i + \hat{x}$, where

$$\hat{\sigma}^2 = \left[-\frac{d^2}{dx^2} \log(g(x)) \bigg|_{x=\hat{x}} \right]^{-1}.$$

Thus, the approximation of the integral of $g(x)$ over $\Re$ is given by

$$\int_{\Re} g(x)dx \approx \sqrt{2}\hat{\sigma} \sum_{i=1}^{n} g(p_i^\star)\, \exp(p_i^2)\, w_i. \tag{2.2}$$

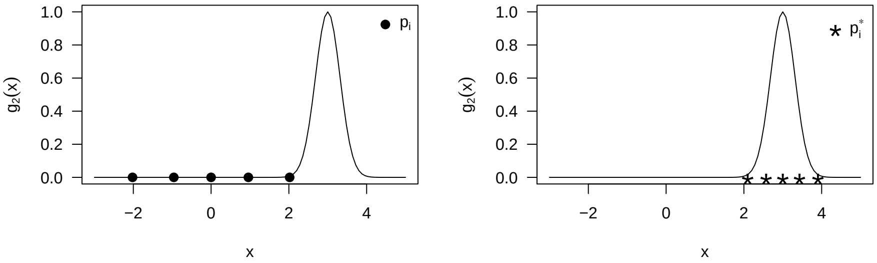

Figure 2: $g_2(x)$ function with quadrature points p_i and $p_i^\star$.

Note that in the Equation (2.2) the function $g(x)$ is evaluated at $p_i^\star$, while $\exp(\cdot)$ is evaluated at points p_i. For the function $g_2(x)$, we have $\hat{x} = 3$ and $\hat{\sigma} = 0.31623$; thus, the new set of quadrature points is $\boldsymbol{P}^\star = (2.0965, 2.5713, 3, 3.4287, 3.9035)$ with the same $\boldsymbol{W}$ weights. The right panel in Figure 2 presents the function $g_2(x)$ with the new quadrature points, and it can be observed that the five new points are centered around the area of the highest density of $g_2(x)$. Using the Equation (2.2) we find that the integral is 0.79266, which coincides with the exact value of the integral.

To facilitate the understanding of the CG and ACG methods in a multidimensional case, an example of bidimensional integration is presented in detail. Assume that we are interested in calculating the integral of $g(x_1, x_2)$ over $\Re^2$ using five quadrature points. Suppose that $g(x_1, x_2)$ is the density of a bivariate normal distribution with the mean vector $\boldsymbol{\mu} = (-2, -2)^\top$ and variance-covariance given by 1, 2 and -0.5. In this case, we must construct a two-dimensional GQ scheme, based on quadrature points $\boldsymbol{P} = \{p_1, p_2, \ldots, p_5\}$ that correspond to all combinations of elements $\boldsymbol{P}$. We denote the two-dimensional array of points by $\boldsymbol{Z}$, which, for this example, has 25 elements and is given by

$$\begin{aligned}\boldsymbol{Z} = \{&(p_1, p_1), (p_1, p_2), \ldots, (p_1, p_5),\\ &(p_2, p_1), (p_2, p_2), \ldots, (p_2, p_5),\\ &\qquad\vdots\\ &(p_5, p_1), (p_5, p_2), \ldots, (p_5, p_5)\}.\end{aligned}$$

The quadrature weights, denoted by $\boldsymbol{A}$, are given by the product of the associated weights of the elements of each pair from $\boldsymbol{Z}$ and is given by

$$\begin{aligned}\boldsymbol{A} = \{&w_1w_1,\ w_1w_2,\ \ldots,\ w_1w_5,\\ &w_2w_1,\ w_2w_2,\ \ldots,\ w_2w_5,\\ &\qquad\vdots\\ &w_5w_1,\ w_5w_2,\ \ldots,\ w_5w_5\}.\end{aligned}$$

Thus, the expressions for approximating the integral of $g(x_1, x_2)$ over $\Re^2$ by the GQ and AGQ methods are given by

$$\text{For GQ:} \qquad \int_{\Re^2} g(x_1, x_2) dx_1\, dx_2 \approx \sum_{i=1}^{25} g(\boldsymbol{z}_i)\, \exp(\boldsymbol{z}_i^\top \boldsymbol{z}_i)\, a_i, \tag{2.3}$$

$$\text{For AGQ:} \qquad \int_{\Re^2} g(x_1, x_2) dx_1\, dx_2 \approx 2\left|\hat{\boldsymbol{Q}}\right|^{1/2} \sum_{i=1}^{25} g(\boldsymbol{z}_i^\star)\, \exp(\boldsymbol{z}_i^\top \boldsymbol{z}_i)\, a_i, \tag{2.4}$$

where $\boldsymbol{z}_i$ and a_i correspond to the elements of $\boldsymbol{Z}$ and $\boldsymbol{A}$, respectively. The new quadrature points $\boldsymbol{z}_i^\star$ for AGQ are centered around the maximum $\hat{\boldsymbol{x}}$ of $\log(g(x_1, x_2))$ and are given by $\boldsymbol{z}_i^\star = \hat{\boldsymbol{x}} + \sqrt{2}\hat{\boldsymbol{Q}}^{1/2}\boldsymbol{z}_i$, where $\hat{\boldsymbol{Q}}^{1/2}$ corresponds to the Cholesky decomposition of the curvature matrix $\hat{\boldsymbol{Q}}$, which is given by

$$\hat{\boldsymbol{Q}} = \left[-\frac{d^2}{dx_1\, dx_2} \log(g(x_1, x_2))\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}}\right]^{-1}.$$

Figure 3 shows contour lines for $g(x_1, x_2)$; in the left panel, the 25 quadrature points $\boldsymbol{z}_i$ are shown, and in the right panel are the new transformed quadrature points $\boldsymbol{z}_i^\star$. Unlike the quadrature points $\boldsymbol{z}_i$, it can be observed that the transformed points $\boldsymbol{z}_i^\star$ are located in the area of greatest density for $g(x_1, x_2)$,

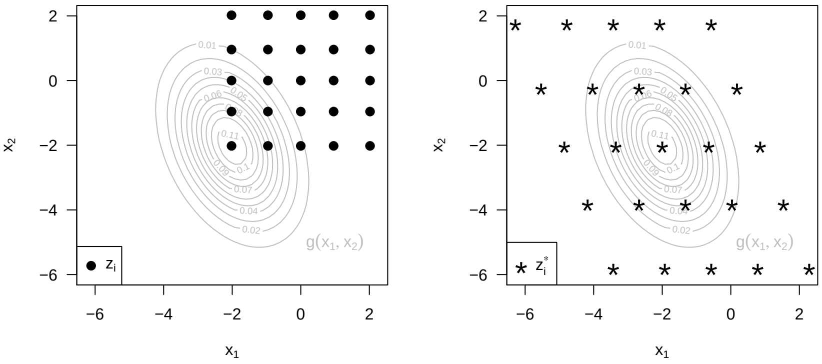

Figure 3: Contour lines of $g(x_1, x_2)$ with quadrature points $\boldsymbol{z}_i$ and $\boldsymbol{z}_i^\star$.

which is reflected in the approximation of the integral. Using the Equations (2.3) and (2.4), we find that the values for approximating the integral are 0.4719 with the GQ and 0.9999 with the AGQ.

Based on the expressions presented for the bivariate case, the GQ and AGQ methods can be extended for a q–dimensional integral of a function $g(\boldsymbol{x})$ over $\Re^q$. In this case, with n quadrature points, $\boldsymbol{Z}$ is based on the Cartesian product of $\boldsymbol{P}$, and the quadrature weights $\boldsymbol{A}$ are similarly based on the Kronecker product (denoted by $\otimes$) of the original weights $\boldsymbol{W}$, as given below:

$$\boldsymbol{Z} = \underbrace{\boldsymbol{P} \times \ldots \times \boldsymbol{P}}_{q \quad \text{times}} = \boldsymbol{P}^q,$$

$$\boldsymbol{A} = \underbrace{\boldsymbol{W} \otimes \ldots \otimes \boldsymbol{W}}_{q \quad \text{times}}.$$

Thus, the expressions for approximating the integral of $g(\boldsymbol{x})$ over $\Re^q$ by the GQ and AGQ methods are given by

$$\text{(2.5)} \qquad \text{For GQ:} \qquad \int_{\Re^q} g(\boldsymbol{x}) d\boldsymbol{x} \approx \sum_{i=1}^{n^q} g(\boldsymbol{z}_i)\, \exp(\boldsymbol{z}_i^\top \boldsymbol{z}_i)\, a_i,$$

$$\text{(2.6)} \qquad \text{For AGQ:} \qquad \int_{\Re^q} g(\boldsymbol{x}) d\boldsymbol{x} \approx \left|\hat{\boldsymbol{Q}}\right|^{1/2} 2^{q/2} \sum_{i=1}^{n^q} g(\boldsymbol{z}_i^\star)\, \exp(\boldsymbol{z}_i^\top \boldsymbol{z}_i)\, a_i,$$

where $\boldsymbol{z}_i$ and a_i correspond to the elements of $\boldsymbol{Z}$ and $\boldsymbol{A}$, respectively. The new quadrature points $\boldsymbol{z}_i^\star$ for the AGQ are centered around the maximum $\hat{\boldsymbol{x}}$ of $\log(g(\boldsymbol{x}))$ and are given by $\boldsymbol{z}_i^\star = \hat{\boldsymbol{x}} + \sqrt{2}\hat{\boldsymbol{Q}}^{1/2}\boldsymbol{z}_i$, where $\hat{\boldsymbol{Q}}^{1/2}$ corresponds to the Cholesky decomposition of the curvature matrix $\hat{\boldsymbol{Q}}$, which is given by:

$$\hat{\boldsymbol{Q}} = \left[-\frac{d^2}{d\boldsymbol{x}^2} \log(g(\boldsymbol{x})) \Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \right]^{-1}.$$

From the above examples, it is evident that the AGQ generates better results than the GQ; however, the computational cost of the AGQ approach is increased by the optimization required to transform the quadrature points. If we are interested in solving an integral once, then this additional computational cost is not significant, but when the AGQ is used in an iterative process to estimate model parameters from a mixed model, the method becomes computationally intensive. If the processing time is a key aspect in a mixed model, then Hartzel et al. (2001) recommends using the GQ with few quadrature points to obtain initial parameter values and then using the AGQ to improve the accuracy of the results.

3. Adaptive Gaussian quadrature with pruning

A key aspect in the implementation of GQ and AGQ is the choice of the number of quadrature points n. A large number of quadrature points allows for a better approximation but at a higher computational cost. For a problem with q integrals the computational cost increases because it is necessary to consider all combinations of n points for each integral, which generates a set of n^q total points. For example, to obtain the integral of a function with $q = 3$ variables using $n = 15$ quadrature points, it is necessary to construct an array with $15^3 = 3375$ combinations of points. For each of these combinations, the new weights are obtained as the initial product of the individual weights. However, some of the n^q combinations with small weights do not contribute to the approximation of the integral. Therefore, it is necessary to have a tool to identify and eliminate these possible combinations which do not contribute to the approximation process.

With this purpose in mind, in this paper we generalized the method presented by Jäckel, P. (2005) to eliminate points in the q-dimensional case. The idea of the generalization is as follows: those points from the new set of points $\boldsymbol{Z}$, for which the associated weights a_i are less than a reference value θ, are removed and are not used in the approximation because the importance a_i is not significant, and they do not contribute to the integral approximation. The reference value θ depends on n, q and on the minimum and median weights $\boldsymbol{W}$ and it is given by the expression

$$\theta = \frac{w_{[1]}\, w_{[\frac{n+1}{2}]}}{n^{q-1}},$$

where $w_{[1]}$ and $w_{[\frac{n+1}{2}]}$ correspond respectively, to the minimum and median of the original weights $\boldsymbol{W}$.

The proposed method works because the individual weights $w_i \in (0, 1)$ and

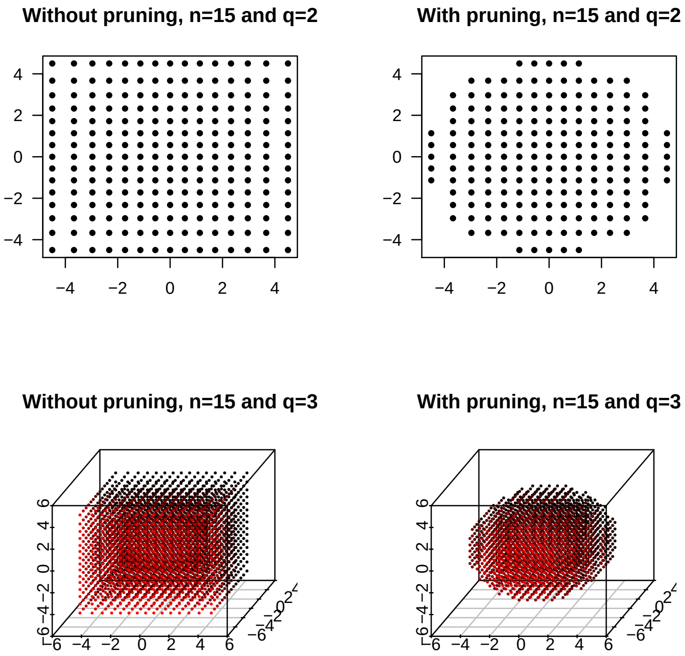

Figure 4: Quadrature points without and with pruning.

are symmetric around the central weight which has the greatest value, thus the weights product array $\boldsymbol{A}$ has some small components. The reference value θ helps to identify which elements of $\boldsymbol{Z}$ should be dropped due to its small importance $\boldsymbol{A}$ in the approximation.

Figure 4 compares the situation with and without pruning for the cases $q = 2$ and $q = 3$; we can observe from the figure that the eliminated combinations are located at the edges and that the remaining combinations are located in the central region which has the points with greatest weights. The advantage to conducting this elimination of points is the decrease in the number of times that the function of interest is evaluated to approximate an integral. This advantage is crucial in situations such as simulation and mixed models in which we must address several integrations within an iterative process to obtain estimated parameters.

As an illustration of the reduction on the number of points obtained with the pruning approach, Table 1 shows the number of total points with and without the elimination points for $q = 2, 3$ with $n = 3, 7, 15$ initial quadrature points. From the table, it can be observed that there is a decrease between 17% and 44% in the total number of points with pruning approach, this reduction could be important in a situation in which the interest is to approximate an integral a huge number of times or in a heavily iterative process.

	$q=2$			$q=3$		
n	Without pruning	With pruning	PR(%)	Without pruning	With pruning	PR(%)
3	9	5	44.44	27	19	29.63
7	49	37	24.49	343	251	26.82
15	225	185	17.77	3375	2199	34.84

Note: PR denotes the percentage of reduction in the number of points.

Table 1: Number of final points to solve $q = 2, 3$ integrals with $n = 3, 7, 15$ quadrature points.

Once the pruning is made we can use the Equation (2.5) or (2.6) with the remaining points of $\boldsymbol{Z}$, the sum will have fewer elements, resulting in a shorter processing time.

4. Application

We re-analyzed data from a longitudinal ophthalmology study described in Meyers et al. (1992) and Qui et al. (2008). The data set has information concerning 31 patients with respect to 4 variables. The response variable y_{ij} represents the observed proportion of the remaining gas volume relative to the initial volume of gas injected in the eye for patient i-th at the j-th follow-up day t_{ij}. Qui et al. (2008) considered the Simplex distribution $S^{-}(\mu, \sigma)$, which is useful for modeling variables between 0 and 1. The data set contains information about the initial gas concentration (at 15%, 20% and 25%) of C_3F_8 and information about the days on which the y_{ij} values were recorded. Figure 5 displays a longitudinal plot for the proportion of gas volume remaining. Each line in the figure represents the evolution of one patient.

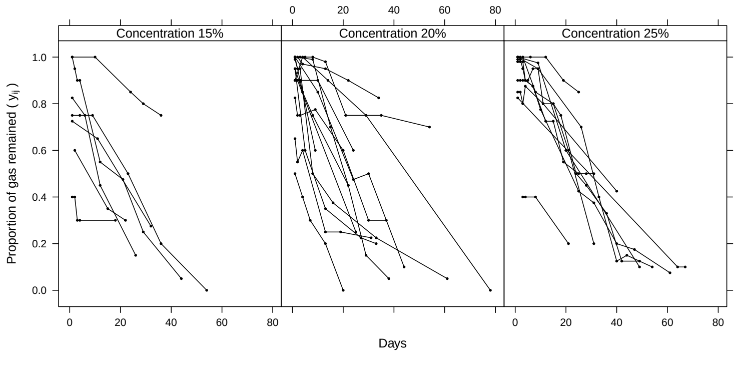

Figure 5: Longitudinal plot for the proportion of gas volume remaining.

In the Simplex distribution, the parameter $\mu \in (0,1)$ corresponds to the mean, and the parameter $\sigma^2 > 0$ corresponds to the dispersion parameter. We use the same approach as Qui et al. (2008) to model the parameter μ as a function of $\log(Days)$, $\log^2(Days)$ and x_{ij}, where x_{ij} is the covariate of standarized gas concentration levels, with the values -1 (15%), 0 (20%) and 1 (25%). The model for μ can be summarized by

$$g(\mu_{ij}) = \beta_0 + b_{0i} + \beta_1 \log(Days) + \beta_2 \log^2(Days) + (\beta_3 + b_{1i})x_{ij}, \tag{4.1}$$

where $g(x)$ corresponds to the *Cauchy* link function that maps from $(0,1)$ to $\Re$. The elements b_{0i} and b_{1i} are the random effects which were considered as $b_{0i} \sim N(0,\tau_1^2)$, $b_{1i} \sim N(0,\tau_2^2)$ and uncorrelated. The vector parameter for the model (4.1) is $\boldsymbol{\theta} = (\beta_0, \beta_1, \beta_2, \beta_3, \log(\sigma), \log(\tau_1), \log(\tau_2))^\top$. The log–likelihood function for this problem can be expressed as

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{31} \log \left[\int_{\Re^2} \prod_{j=1}^{n_i} f_Y(y_{ij} \mid b_{0i}, b_{1i}) \cdot f(b_{0i};\tau_1) \cdot f(b_{1i};\tau_2)\, db_{0i}\, db_{1i} \right], \tag{4.2}$$

where $f_Y(\cdot)$ corresponds to the probability density function of the Simplex distribution and $f(\cdot)$ corresponds to the normal density. The n_i value denotes the number of observations available for the i-th patient. Note that $l(\boldsymbol{\theta})$ in expression (4.2) has 31 integrals over $\Re^2$, none of which has an analytical solution; for this reason, we adopted the AGQ to solve each integral. The estimates for $\boldsymbol{\theta}$ were obtained by maximizing $l(\boldsymbol{\theta})$ using R Core Team (2016) on a Windows system with Intel Core i7 8Gb RAM without using another tool to speed up the process.

Parameter	Without pruning			With pruning		
	$n=3$	$n=5$	$n=7$	$n=3$	$n=5$	$n=7$
β_0	1.6297	1.6164	1.6174	1.6289	1.6135	1.6153
β_1	0.9108	0.9127	0.9128	0.9033	0.9153	0.9141
β_2	-0.4952	-0.4966	-0.4955	-0.4902	-0.4970	-0.4953
β_3	0.9715	0.9563	0.9582	0.9644	0.9538	0.9566
$\log(\sigma)$	1.3303	1.3282	1.3292	1.3370	1.3285	1.3293
$\log(\tau_1)$	-1.5739	-1.6168	-1.6285	-1.5739	-1.6248	-1.6331
$\log(\tau_2)$	-1.7127	-1.7265	-1.7137	-1.7032	-1.7193	-1.7093
Time (sec)	626	816	1479	549	612	1065

Table 2: Values of $\boldsymbol{\theta}$ estimates for different n values with and without pruning and the processing time in seconds.

The estimated values for $\boldsymbol{\theta}$ without and with pruning are shown in Table 2. From this table we note that all $\hat{\boldsymbol{\theta}}$ values (columns) for each setting of n, with and without pruning, are close. The last row shows the processing time (in seconds) required to obtain the estimates $\hat{\boldsymbol{\theta}}$. As expected, time increases as the number of quadrature points n increases. We can observe that there is a decrease in the time when comparing the results without against with pruning; the percent decrease in time is 14%, 33% and 39% for $n = 3$, $n = 5$ and $n = 7$, respectively; this decrease in time increases as there are more quadrature points. In this application we can see that the required time to fit the model with the proposed approach is less than the usual approach without pruning, we clarify that this difference in time can be significant for larger data sets or in simulation studies.

5. Simulation study

In this section we present the results from a simulation study conducted to analyze the performance of the proposed methodology. The background for the simulation study was the application presented in the previous section. We considered the response variable y_{ij} (the response of the i-th patient at the j-th follow-up day t_{ij}) distributed as Simplex distribution with σ parameter fixed at value 3.78 (obtained from $e^{1.33}$). For the μ parameter was considered the following model with known fixed parameters

$$g(\mu_{ij}) = 1.61 + b_{0i} + 0.91\log(Days) - 0.50\log^2(Days) + (0.96 + b_{1i})x_{ij} \tag{5.1}$$

We assumed that all patients were examined at 0, 10, 20, 40 and 60 days. The random effects b_{0i} and b_{1i} were considered as $b_{0i} \sim N(0, 0.19^2)$, $b_{1i} \sim N(0, 0.18^2)$ and uncorrelated. The covariate of standarized gas concentration levels x_{ij} assumed values of -1 (15%), 0 (20%) and 1 (25%). All fixed values considered in the simulation study were obtained from results showed in Table 2. With this choice the true parameter vector was $\boldsymbol{\theta} = (1.61, 0.91, -0.50, 0.96, \log(3.78), \log(0.19), \log(0.18))^\top$. For N = 15, 21, 30, 60, 90 and 120 patients were simulated 500 data sets and then were estimated the vector parameter $\hat{\boldsymbol{\theta}}$ for model 5.1. We used the processing time and the relative distance $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|/\|\boldsymbol{\theta}\|$ to quantify the quality estimators.

In Tables 3 to 8 and Figures 6 to 7 are the results from the simulation study. From these tables and figures we highlight that:

- the mean relative distance for each estimated parameter $(\hat{\beta}_0, \ldots, \log(\hat{\tau}_2))$ and for the vector $\hat{\boldsymbol{\theta}}$ are similar with and without pruning, this means that the quality of the estimates are the same despite of points eliminating in the approximation.
- the mean relative distance for $\hat{\boldsymbol{\theta}}$ decreases as the number of patients increases, this is possibly because we have more information or data to estimate the parameter vector,
- from Figures 6 and 7 we note that mean timelines with pruning tend to be below the mean timelines without pruning, the time reduction for all cases was around 7.94%.

- the greatest difference in mean time to find $\hat{\boldsymbol{\theta}}$ was 21.57% observed in Figure 6 for the case of $N = 30$ and $n = 21$.

These findings support the proposed pruning choice because the quality of the estimates are similar with and without pruning, and the most important feature obtained from the simulation study is the decrease in processing time around 7.94%, this reduction could be important for researchers and users in simulation studies or in applications with big data sets.

Parameter	Without pruning			With pruning		
	$n=3$	$n=11$	$n=21$	$n=3$	$n=11$	$n=21$
β_0	0.29	0.29	0.37	0.29	0.30	0.32
β_1	0.76	0.54	1.26	0.73	0.76	0.77
β_2	2.96	3.18	3.18	2.93	2.99	2.70
β_3	0.47	0.56	0.48	0.46	0.44	0.36
$\log(\sigma)$	0.12	0.09	0.07	0.11	0.09	0.10
$\log(\tau_1)$	0.42	0.41	0.39	0.41	0.43	0.39
$\log(\tau_2)$	0.19	0.17	0.14	0.20	0.18	0.18
Mean relative distance	0.59	0.60	0.69	0.58	0.60	0.57
Mean time (sec)	70.38	97.00	250.58	66.33	85.68	213.47

Table 3: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 15$.

Parameter	Without pruning			With pruning		
	$n=3$	$n=11$	$n=21$	$n=3$	$n=11$	$n=21$
β_0	0.25	0.21	0.23	0.25	0.38	0.13
β_1	0.62	0.74	0.59	0.65	0.93	0.51
β_2	2.98	3.00	2.99	2.99	2.93	2.94
β_3	0.44	0.35	0.45	0.43	0.52	0.37
$\log(\sigma)$	0.12	0.09	0.09	0.11	0.10	0.09
$\log(\tau_1)$	0.43	0.43	0.42	0.41	0.45	0.42
$\log(\tau_2)$	0.19	0.17	0.17	0.20	0.19	0.20
Mean relative distance	0.57	0.56	0.55	0.57	0.63	0.54
Mean time (sec)	106.00	148.19	398.30	96.96	126.94	331.33

Table 4: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 21$.

6. Concluding

In this paper we proposed a pruning approach to eliminate points that have lower importance to approximate integrals with the Gaussian quadrature

Parameter	Without pruning			With pruning		
	$n = 3$	$n = 11$	$n = 21$	$n = 3$	$n = 11$	$n = 21$
β_0	0.20	0.22	0.30	0.22	0.24	0.28
β_1	0.48	0.44	0.67	0.49	0.68	0.59
β_2	2.98	2.92	2.94	2.98	3.01	3.04
β_3	0.38	0.30	0.38	0.37	0.45	0.40
$\log(\sigma)$	0.13	0.09	0.09	0.12	0.12	0.09
$\log(\tau_1)$	0.44	0.42	0.45	0.43	0.47	0.45
$\log(\tau_2)$	0.20	0.18	0.22	0.21	0.20	0.19
Mean relative distance	0.55	0.52	0.58	0.55	0.58	0.57
Mean time (sec)	156.33	245.71	623.75	147.65	243.29	489.17

Table 5: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 30$.

Parameter	Without pruning			With pruning		
	$n = 3$	$n = 11$	$n = 21$	$n = 3$	$n = 11$	$n = 21$
β_0	0.18	0.13	0.16	0.16	0.17	0.16
β_1	0.30	0.33	0.37	0.33	0.35	0.41
β_2	2.98	3.05	3.10	3.02	3.08	3.01
β_3	0.37	0.29	0.30	0.37	0.30	0.28
$\log(\sigma)$	0.14	0.10	0.10	0.14	0.12	0.11
$\log(\tau_1)$	0.44	0.43	0.43	0.42	0.45	0.43
$\log(\tau_2)$	0.20	0.19	0.20	0.22	0.22	0.20
Mean relative distance	0.53	0.52	0.54	0.53	0.54	0.53
Mean time (sec)	244.13	485.67	1377.97	230.98	442.91	1149.85

Table 6: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 60$.

method. The pruning step ensures that the removed points are outside of the original cloud of points and that they do not contribute to the integral approximation. The proposed method is useful because it reduces the computational effort required, which could be significant in the case of large datasets, large sample sizes or high-dimensional integration, while maintaining the quality of the integral approximations. These findings were showed by a simulation study with a Simplex mixed model, we found a 7.94% reduction time to estimate the parameter vector with a similar mean relative distance with and without pruning.

References

Blocker Alexander W. **fastGHQuad**: Fast Rcpp Implementation of Gauss-Hermite Quadrature. 2011. R package version 0.1-1.

Parameter	Without pruning			With pruning		
	$n = 3$	$n = 11$	$n = 21$	$n = 3$	$n = 11$	$n = 21$
β_0	0.17	0.18	0.12	0.14	0.13	0.15
β_1	0.27	0.28	0.29	0.24	0.31	0.29
β_2	2.97	2.96	3.04	2.97	3.01	3.03
β_3	0.36	0.32	0.28	0.33	0.28	0.33
$\log(\sigma)$	0.14	0.10	0.11	0.12	0.11	0.11
$\log(\tau_1)$	0.43	0.44	0.44	0.44	0.43	0.44
$\log(\tau_2)$	0.21	0.19	0.20	0.21	0.19	0.21
Mean relative distance	0.52	0.52	0.52	0.51	0.51	0.52
Mean time (sec)	362.67	840.99	2443.29	342.45	782.48	2025.75

Table 7: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 90$.

Parameter	Without pruning			With pruning		
	$n = 3$	$n = 11$	$n = 21$	$n = 3$	$n = 11$	$n = 21$
β_0	0.14	0.12	0.14	0.12	0.15	0.15
β_1	0.24	0.27	0.23	0.26	0.27	0.25
β_2	3.00	3.01	3.03	2.94	2.97	2.95
β_3	0.37	0.28	0.33	0.37	0.32	0.37
$\log(\sigma)$	0.14	0.11	0.10	0.13	0.12	0.11
$\log(\tau_1)$	0.44	0.44	0.43	0.43	0.45	0.44
$\log(\tau_2)$	0.21	0.20	0.20	0.21	0.20	0.19
Mean relative distance	0.52	0.52	0.52	0.51	0.52	0.51
Mean time (sec)	484.90	1058.34	2818.33	537.52	980.55	2574.51

Table 8: Mean relative distance for each estimated parameter and for $\hat{\boldsymbol{\theta}}$ vector when the number of patients is $N = 120$.

Broström G., Holmberg H. **glmmML**: Generalized Linear Models with Clustering. 2011. R package version 0.82-1.

Demidenko E. Mixed Model, Theory and Applications. New Jersey: John Wiley & Sons, Inc, 2004.

Hartzel H., Agresti A., Caffo B. Multinomial Logit Random Effects Models // Statistical Modelling. 2001. 1. 81–102.

Jäckel, P. . A Note on Multivariate Gauss-Hermite Quadrature. 2005.

Kuonen Diego. Numerical Integration in S-PLUS or R: A Survey // Journal of Statistical Software. 7 2003. 8, 13. 1–14.

Liu Q., Pierce D. A. A Note on Gauss–Hermite Quadrature // Biometrika. 1994. 81, 3. 624–629.

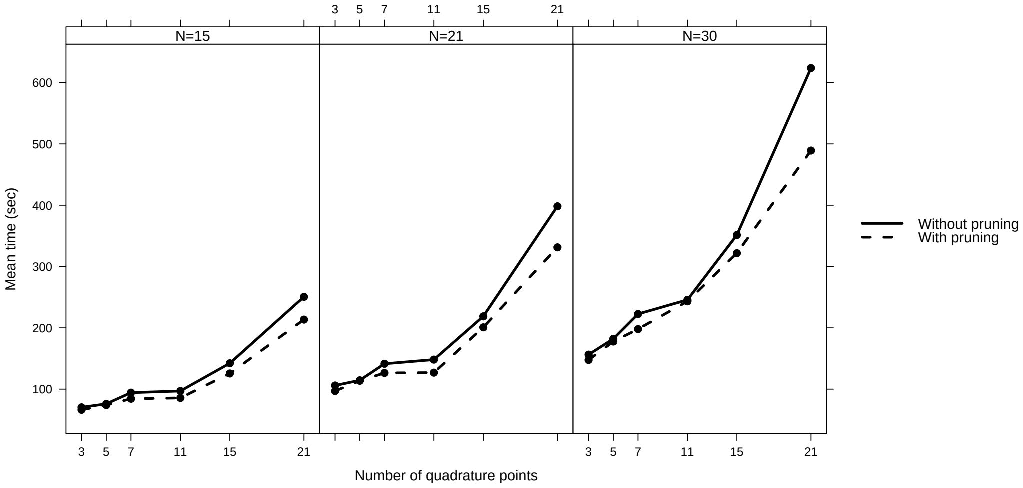

Figure 6: Mean times to obtain the estimations against quadrature points number (n) for $N = 15, 21, 30$ patients.

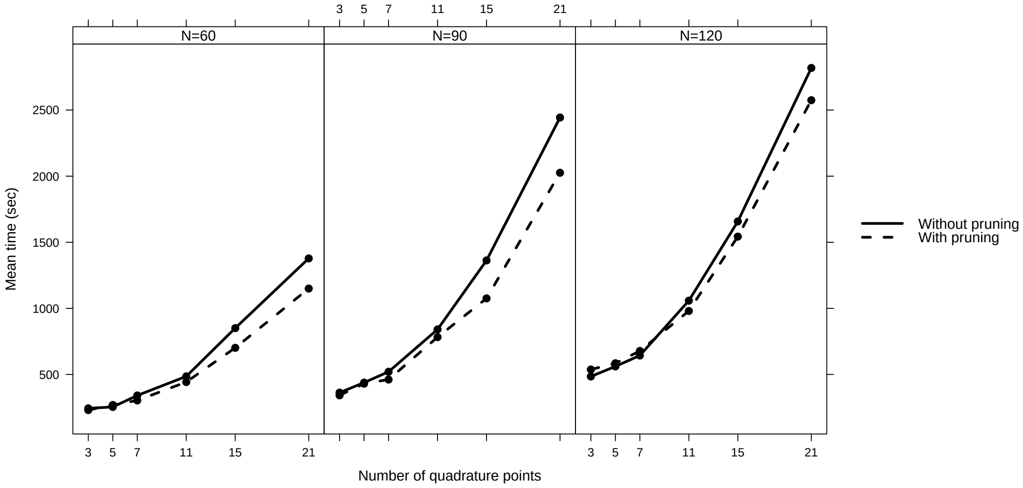

Figure 7: Mean times to obtain the estimations against quadrature points number (n) for $N = 60, 90, 120$ patients.

Meyers S. M., Ambler J. S., Tan M., Werner J. C., Huang S. S. Variation of Perfluorpropane Disappearance After Vitrectomy // Retina. 1992. 12. 359–363.

Molenberghs G., Verbeke G. Models for Discrete Longitudinal Data. New York: Springer, 2005. First.

Pinheiro J. C., Bates D. M. Approximations to the Log–Likelihood Function in the Nonlinear Mixed–Effects Model // Journal of Computational and Graphical Statistics. 1995. 4, 1. 12–35.

Qui A., Song P. X., Tan M. Simplex Mixed-Effects Models for Longitudinal Proportional Data // Scandinavian Journal of Statistics. 2008. 35. 577–596.

R Core Team . R: A Language and Environment for Statistical Computing. Vienna, Austria, 2016.

Smyth G., Hu Y., Dunn P., Phipson B. **statmod**: Statistical Modeling. 2011. R package version 1.4.14.