The need for flexibility in hierarchical spatial modelling

Hans J. Skaug[a,*], David A. Fournier[b]

[a] *Department of Mathematics, P. O. Box 7800, 5020 Bergen, Norway.*
[b] *Otter Research Ltd., Sidney, Canada*

Abstract

We discuss how generic interfaces to the Laplace approximation for fitting hierarchical spatial models can be constructed. The goal is to hide the technical details of computation, but leave the user with flexibility in model formulation. As an example we explain how the SPDE approach of Lindgren et al. (2011) can easily be implemented in the interface to the open source software AD Model Builder. Flexibility is needed not only in the model for spatial correlation, but also in the response distribution. Using a standard dataset we find that the introduction of a discrete mixture for the response improves the fit much more than does changing the parametric assumptions about the correlation function. This phenomenon can be expected to be common in practice, and underlines the need for flexibility in all aspects of model formulation. We contrast the user interface of AD Model Builder with the more familiar interface of R packages.

Keywords: Automatic differentiation, ADMB, empirical Bayes, Gaussian Markov random fields, hierarchical models, Laplace Approximation.

1. Introduction

Gaussian random fields are important building blocks in hierarchical models for spatial data (Banerjee et al., 2003). Their use facilitates both computation and interpretation. The assumption that the latent random variables are Gaussian allows the use of the Laplace approximation to evaluate

*Correspence to: Department of Mathematics, P.O. Box 7800, 5020 Bergen, Norway. Phone: (+47) 55584861, Fax: (+47) 55589672.

Email addresses: skaug@math.uib.no (Hans J. Skaug), davef@otter-rsch.com (David A. Fournier)

a marginal likelihood on which inference about the (hyper) parameters can be based (Skaug and Fournier, 2006). It was argued in Rue et al. (2009) that the Laplace approximation can offer substantial computational advantages over standard MCMC methods. However, implementing such computational schemes from scratch requires skills that most users of spatial statistics do not possess. It is thus important that generic implementations of the Laplace approximation exist, ones that hide the technical details of computation, but offer access to model fitting functions through a simplified interface. The nature and construction of such interfaces is the topic of the present paper.

The open source software R (R Development Core Team, 2012) contains many packages (functions) capable of fitting hierarchical spatial models. Among the most flexible and versatile of these is R-INLA (Rue et al., 2009) which is based on the Laplace approximation for evaluating the marginal likelihood. In R-INLA the interface between the user and the Laplace approximation is an R function. The model is formulated in a flexible generalized linear model framework, where the linear predictor can contain spatial components among other things. Although R-INLA gives access to large array of response distributions and correlation structures, it necessarily restricts the set of models that can be fitted. In the present paper we point out advantages and disadvantages of a restricted interface, and contrast it with more flexible interfaces based on a numerical technique known as automatic differentiation (Griewank and Walther, 2008). Automatic differentiation (AD) is gradually finding its way into statistical practice, partly through software packages such as AD Model Builder (Fournier et al., 2012), Stan (Stan Development Team, 2013) and Ceres Solver (Agarwal and Mierle, 2013).

For a fixed value of the parameters the Laplace approximation of the marginal likelihood involves the determinant of the Hessian matrix H of the log posterior distribution of the latent random field, appropriately discretized. Evaluating analytic expressions for these second order derivatives of the log posterior will often be an obstacle to the implementation of the Laplace approximation. It was shown by Skaug and Fournier (2006) that automatic differentiation, not to be confused with symbolic differentiation or 'finite differences', can make numerical evaluation of H transparent, given computer code for the joint likelihood of parameters and latent random variables. Skaug and Fournier (2006) made their argument more generally for models containing latent random variable, and in the present paper we point out the specifics that applies in the spatial setting.

Recently, Lindgren et al. (2011) derived the computationally efficient

SPDE approximation to Gaussian random fields with a Matern covariance function. The approximating model is Markov on a mesh which includes the points of observations as nodes. We show that very flexible hierarchical spatial models, with an SPDE based latent random field, can be fit easily in the open source software AD Model Builder (Fournier et al., 2012), which among other things implements the ideas from Skaug and Fournier (2006). The SPDE mesh, along with a set of matrices needed to build the precision matrix for the SPDE approximation, is imported from R-INLA. We use the Leukemia dataset studied by Lindgren et al. (2011) as an example to motivate the need for a flexible, but still fairly simple, interfaces to the underlying Laplace approximation.

Section 2 outlines the methodological components in our approach, including a description of the ADMB user interface. In Section 3 different models for spatial correlation and response distributions are fitted to the Leukemia data, and Section 4 contains a comparison of advantages and disadvantages of the ADMB user interface, relative to the more familiar interface of R packages.

2. Methodology

2.1. Hierarchical spatial models

We consider a vector of spatially referenced observations $\mathbf{y} = (y_1, \ldots, y_n)$, where each y_i is associated with the spatial location $s_i \in R^2,\ i = 1, \ldots, n$. Next, we postulate the existence of a zero mean Gaussian random field $X = \{x(s) : \ s \in R^2\}$ influencing the distribution of $\mathbf{y}$ via its values $\mathbf{x} = \{x(s_1), x(s_2), \ldots, x(s_n)\}$ at the locations of observation. Hence, $\mathbf{x}$ has a zero mean multivariate Gaussian distribution with covariance matrix Σ, say, which is induced by the correlation structure of the continuously indexed random field X. Beyond this fact, X does not play any particular role in present paper, although it does play a crucial role in the derivation of the underlying SPDE approach of Lindgren et al. (2011). To foresee the use of the SPDE approach we will augment $\mathbf{x}$ with a set of "support point" $x(s_{n+1}), x(s_{n+2}), \ldots, x(s_m)$, so that the total dimension of $\mathbf{x}$ is m.

It is often assumed that the y_i are conditionally independent given the x_i, i.e.

$$p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p_\theta(y_i|x_i) \tag{1}$$

and this will be the case in the example in Section 3 below. However, conditional independence is not needed for the general theory nor for the ADMB software, so we make no special provision in our notation. Both Σ and the conditional density of $\mathbf{y}$ will depend on parameters, and we shall write Σ_θ and $p_\theta(\mathbf{y}|\mathbf{x})$, respectively, where θ denotes a parameter vector. Typically, but not necessarily, Σ and p depend on different subsets of θ. In a typical application y_i, conditionally on x_i, is taken to have one of the standard distributions: Gaussian, binomial, Poisson, or Weibull as in the Leukemia example of Section 3.

The marginal likelihood,

$$L(\theta) = p_\theta(\mathbf{y}) = \int p_\theta(\mathbf{y}|\mathbf{x})\, N(\mathbf{x}; 0, \Sigma_\theta)\, d\mathbf{x}, \tag{2}$$

is the basis for maximum likelihood estimation of θ, where $N(\mathbf{x}; 0, \Sigma)$ denotes the zero mean multivariate Gaussian density with covariance matrix Σ. The computational challenge in all latent variable models, and in spatial statistics in particular where m typically is large, is the integration over $\mathbf{x} \in R^m$. In practically all but the Gaussian case this integral cannot be evaluated analytically and one is forced to resort to numerical evaluation. Among the non-sampling based approximations, the Laplace approximation is the one in most widespread use (references). It also underlies both software packages R-INLA and ADMB.

2.2. Laplace approximation and AD

The basis for the Laplace approximation is the log joint density of $\mathbf{x}$ and $\mathbf{y}$, given by

$$g(\mathbf{x}, \mathbf{y}; \theta) \;=\; \log\{p_\theta(\mathbf{y}|\mathbf{x})\} - \frac{m}{2}\log(2\pi) + \frac{1}{2}\log|\det(Q_\theta)| - \frac{1}{2}\mathbf{x}'Q_\theta\mathbf{x}, \tag{3}$$

where Q_θ denotes the $m \times m$ matrix inverse of Σ_θ Here, we focus on the computational aspects of the Laplace approximation. For other properties and its derivation see Butler (2007). The log likelihood approximation is given as

$$l^*(\theta) = -\frac{1}{2}\log|\det(H_\theta)| + \log\{p_\theta(\mathbf{y}|\widehat{\mathbf{x}}_\theta)\} + \frac{1}{2}\log|\det(Q_\theta)| - \frac{1}{2}(\widehat{\mathbf{x}}_\theta)'\, Q_\theta\widehat{\mathbf{x}}_\theta, \tag{4}$$

where

$$\widehat{\mathbf{x}}_\theta = \arg\max_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}; \theta) \tag{5}$$

and

$$H_\theta = \frac{\partial}{\partial \mathbf{x}^2}\, g(\mathbf{x}, \mathbf{y}; \theta)|_{\mathbf{x}=\widehat{\mathbf{x}}_\theta} = \frac{\partial}{\partial \mathbf{x}^2} \log\left\{p_\theta(\mathbf{y}|\mathbf{x})\right\}|_{\mathbf{x}=\widehat{\mathbf{x}}_\theta} - Q_\theta. \tag{6}$$

It was shown in Skaug and Fournier (2006) that the matrix of second order derivatives on the right hand side in (6) can be evaluated numerically by AD from specification of $\log\left\{p_\theta(\mathbf{y}|\mathbf{x})\right\}$. This is implemented in ADMB which, given C++ code $\log\left\{p_\theta(\mathbf{y}|\mathbf{x})\right\}$ and Q_θ as input, estimates θ by maximizing $l^*(\theta)$. The flexibility referred to in the title of this paper refers partly to the fact that AD hides the technical details of the Laplace approximation. ADMB also evaluates the gradient of $l^*(\theta)$ using third order AD and the implicit function theorem (Skaug and Fournier, 2006). The use of an exact gradient value makes ADMB a numerically stable software.

We next turn to the question of the cost of evaluating the Laplace approximation (4). At first, we shall disregard the cost of evaluating $p\left(\mathbf{y}|\mathbf{x}\right)$ and its derivatives. If the matrix Q (and hence H) is not sparse, the cost of evaluating (4) for large m is dominated by that of evaluating $\det\left(H\right)$ and $\det\left(Q\right)$, which both are $O\left(m^3\right)$. This becomes prohibitive on ordinary computers once m gets in the order of some thousands. If on the other hand Q and H are sparse matrices specialized numerical algorithms exists which can handle much larger values of m. As was shown in Lindgren et al. (2011) the SPDE approach yields a sparse Q and thus is computationally efficient.

Under the assumption (1) we have

$$\frac{\partial}{\partial \mathbf{x}^2} \log\left\{p_\theta(\mathbf{y}|\mathbf{x})\right\} = \sum_{i=1}^{n} \log\left\{p_\theta(y_i|x_i)\right\},$$

allowing AD to be applied separately to each term $\log\left\{p_\theta(y_i|x_i)\right\}$, which in principle involves only a single variable x_i. However, to get the gradient of $l^*(\theta)$ one still has to do the differentiation of each term jointly with respect to θ and x_i, but this nevertheless represents a great simplification relative to differentiating $\log\left\{p_\theta(\mathbf{y}|\mathbf{x})\right\}$ jointly with respect to θ and $\mathbf{x}$ (Skaug and Fournier, 2006).

2.3. The SPDE approach

The starting point for the SPDE approach is a stationary and isotropic Gaussian random field with a Matern correlation function. The Matern class of correlation functions has two parameters, but as pointed out by Lindgren

et al. (2011) one of the parameters (υ in their terminology) is often poorly identified in applications. We hence consider the one-parameter class of isotropic correlation functions,

$$C(s_1, s_2) = \kappa \|s_1 - s_2\| K_1 \{\kappa \|s_1 - s_2\|\}, \tag{7}$$

where s_1 and s_2 are arbitrary spatial locations, $\|s_1 - s_2\|$ denotes Eucledian distance, κ is a scaling parameter and K_1 is the modified Bessel function of the second kind of order 1. The precision matrix Q based on (7) is not sparse, and is hence not well suited for applications where n is large. The key insight from Lindgren et al. (2011) is that by augmenting the vector $\mathbf{x}$ by properly chosen elements a sparse approximation to Q can be obtained.

We here present the theory of Lindgren et al. (2011) from a computational perspective, and leave out details related to the finite element solution of the stochastic partial differential equation (SPDE) representation of the random field. In the terminology of Lindgren et al. (2011) we fix $\upsilon = 1$ and have dimension $d = 2$, such that the characterizing parameter becomes $\alpha = d/2 + \upsilon = 2$. The Matern field then has correlation function given by (7) and marginal variance $\sigma^2 = (2\pi\kappa^2)^{-1}$. Our starting point is that we seek an approximation of the precession matrix $Q^{(0)}$ for the vector $\mathbf{x}^{(0)} = \{x(s_1), x(s_2), \ldots, x(s_n)\}$ associated with locations where observations are made. This is done by obtaining a sparse approximation Q for an augmented vector $\mathbf{x} = \{x(s_1), x(s_2), \ldots, x(s_n), x(s_{n+1}) \ldots, x(s_m)\}$. As explained in Lindgren et al. (2011) the extra points $s_{n+1}, \ldots, s_m$ are part of a finite element mesh used to solve the SPDE. By definition of $\mathbf{x}$ the upper left $n \times n$ quadrant of Q^{-1} is an approximation to the covariance matrix of $\mathbf{x}^{(0)}$. In practice one never forms $Q^{(0)}$ or its inverse explicitly. The model is cast in terms of the augmented quantities Q and $\mathbf{x}$, but it is only the first n elements of $\mathbf{x}$ that enters into $p_\theta(\mathbf{y}|\mathbf{x})$.

The R-INLA function `inla.mesh.create` evaluates the finite element mesh, including determining the extra mesh locations $s_{n+1}, \ldots, s_m$. The function `inla.spde2.matern` returns three sparse matrices, M_0, M_1 and M_2, which do not depend on κ and which are used to build the sparse precision matrix

$$Q_\kappa = \kappa^4 M_0 + 2\kappa^2 M_1 + M_2. \tag{8}$$

The correlation function (7) is monotonically decreasing as a function of $\|s_1 - s_2\|$. Lindgren et al. (2011) also derives the approximation of Q for an

oscillating version of (7). The expression is given by

$$Q_{\kappa,\rho} = \kappa^4 M_0 + 2\cos(\rho\pi)\kappa^2 M_1 + M_2, \tag{9}$$

where $\rho \in [0, 1]$ is the parameter controlling the amount of oscillation.

2.4. The ADMB user interface

An example of an ADMB program is given in Figure 1. A preprocessor is used to convert the program to C++ code, which in turn is compiled by an ordinary C++ compiler and linked to libraries containing the routines for performing AD. The program consists of three sections corresponding to the following fundamental tasks: 1) reading data from file 2) specification of parameters to be estimated 3) specification of the joint log likelihood (3). Under 1) a variety of useful data structures, including ragged arrays, are available. An R package R2admb exists that allows the user to read and write the data files used by ADMB.

Under 2) (`PARAMETER_SECTION`) the vector $\mathbf{x}$ is specified as a `random_effects_vector`. This informs the preprocessor that $\mathbf{x}$ is the target from the Laplace approximation. The following line, `sparse_quadratic_prior Q(x)`, associates $\mathbf{x}$ with a sparse matrix Q which must be specified in a separate function (not shown in Figure 1). The `PARAMETER_SECTION` has two additional useful features. It allows the specification of bounds on the parameters in the numerical optimization of the likelihood, and it allows the user to activate parameters in different "phases" (not show in Figure 1). For models of the type considered in the present paper it is a good strategy to use a least two phases. In the first phase components of θ associated with the random field $\mathbf{x}$ are kept fixed at their initial values, while the likelihood is maximized with respect to the other elements of θ. During the first phase there is no reason to perform the expensive Laplace approximation, so $\widehat{\mathbf{x}} = 0$ can be inserted in (4) and the last two terms need not be evaluated. In the second phase all components of θ are estimated, starting out with the values obtained in the first phase, with the Laplace approximation now activated. The virtue of this approach is that the first phase, where no Laplace approximation is performed, is computationally much less expensive than the second. Depending on how complicated the response distribution $p(\mathbf{y}|\mathbf{x})$ is, one may choose to split the first phase into multiple phases with easily identified parameters being estimated first. The Laplace phase may also be split, where as an example, one could first estimate κ with $\rho = 0$ being fixed in (9).

Under 3) (`PROCEDURE_SECTION`) the user specifies $\log p_\theta(\mathbf{y}|\mathbf{x})$ and Q. Because the model shown in Figure 1 has the conditional independence structure (1) this can be done in a loop where the i'th call to the function `ll_weibull()` evaluates $\log p_\theta(y_i|x_i)$. It is through a call to the function of type `SEPARABLE_FUNCTION` that ADMB detects how to run the AD machinery only on the relevant parameters, x_i and θ, for each likelihood component $\log p_\theta(y_i|x_i)$. Recall that the need for differentiating with respect to θ as well as x_i stems from the fact we are also calculating the gradient of the likelihood approximation.

ADMB represents the sparse matrix $Q = \{q_{jk},\, j,k = 1,\ldots,m\}$ in triplet format, in which only non-zero elements are stored, and each non-zero element is stored as (q_{jk}, j, k). Because of symmetric only the upper diagonal of Q needs to be stored. The sparse matrices M_0, M_1 and M_2 occurring in (8) are conveniently output from R-INLA in sparse triplet form, so in the ADMB program one evaluates (8) by manipulating matrices in sparse triplet format (not shown in Figure 1). This task of explicitly writing code for (8) could easily be hidden from the user.

The flexibility of the ADMB user interface stems from the fact that the likelihood is specified in C++ code. There is hence no limitation to the set of response distributions the user may apply. For a given response distribution, one can always try a finite mixture of such distributions. If the mixture model gives a better fit one may be able to do more accurate inference also about the spatial part of the model. In the opposite case, i.e. the mixture does not improve the fit, the exercise serves as goodness of fit check of the original assumption about the response distribution. Code for a mixture model is provided in Figure 2. Further, it is not a requirement that each call to `ll_weibull()` uses only a single x_i. However, the presence of both x_j and x_k in a call to `ll_weibull()` will cause $Q_{jk} \neq 0$, so the sparsity of Q will decrease and likewise the benefit from using sparse matrix routines. Below some degree of sparsity it will be computationally advantageous to use ordinary numerical linear algebra.

A second aspect of flexibility of the user interface is the specification of Q in C++ code. The expression for Q can be specified freely in terms of as many parameters (subset of θ) as wished. In the anisotropic correlation function referred to above Q depends on covariates and associated regression parameters γ. Further, it is straight forward to write C++ functions for performing Kronecker products of matrices in triplet form, which simplifies formulation of separable space time models.

3. The Leukemia data

As an example we use the data set on spatial variation in leukemia survival in Northwest England which as been used by several authors, including Lindgren et al. (2011) as the first illustration of their SPDE approach. The dataset consists of $n = 1043$ observations ($m = 1749$ nodes elements in $\mathbf{x}$ in total), each of which involves the survival time and a censoring indicator for a leukemia patient, together with a spatial reference. We take as our starting point the same model as used in Lindgren et al. (2011). It is assumed that the (fully observed) survival time y of a patient follows a Weibull distribution with shape parameter α and scale parameter λ. If the observation is censored, i.e. the patient left the study for reasons other than death, the likelihood contribution is $\exp(-\lambda y^\alpha)$, while in the case that the time of death is observed (uncensored patient) the likelihood contribution is

$$\lambda \alpha y^{\alpha-1} \exp(-\lambda y^\alpha).$$

Covariates and the spatial effect enters the model via λ using a GLM framework with linear predictor

$$\log(\lambda) = [intercept + sex + age + wbc + tpis] + \tau^{-1} x(s), \qquad (10)$$

where the term in square brackets is expressed in Wilkinson–Rogers notation (McCullagh and Nelder, 1989, sec 3.4).

The ADMB program (slightly abbreviated) used to fit the model is shown in Figure 1. We found it necessary to make the design matrix associated with the covariate terms inside the square brackets in (10) orthogonal in order to improve the convergence of the numerical optimizer. The resulting regression parameters (β's) were subsequently back transformed to their original scale. Also, we found that the "inner optimization" (5) could be evaluated accurately using four Newton-Raphson steps starting from $\mathbf{x} = 0$. In situations where the (unscaled) log posterior (3) differs more a quadratic function, ADMB will by default use a quasi-Newton algorithm to evaluate (5).

Parameter estimates are given in Table 1 as the "Basic model". As a comparison empirical Bayes point estimates obtained from R-INLA (`int.strategy=''eb''`) are also given. The ADMB and R-INLA results should hence be directly comparable, except that R-INLA uses a non-informative prior on θ. This prior does not seem to affect the maximum likelihood estimate of θ much, though, so the discrepancies observed in the table are most likely due to differences

in the function optimizers used in the two packages. R-INLA shows a slight sensitivity to starting values for the optimization process, while ADMB is numerically stable to four digits. In an attempt to quantify the extent of spatial variation, a model without the spatial component was fitted. It is seen from Table 1 that maximum likelihood estimates based on $\log p_\theta(\mathbf{y}|\mathbf{x}=0)$ alone reduces the log likelihood by 11 units. Hence, the spatial components of the model is clearly preferred by the AIC criterion, despite having two extra parameters (κ and τ).

The different extensions of the Matern model described in Lindgren et al. (2011) were tried. The oscillating correlation model (9) gave a log likelihood increase of 1.44 at the cost of one extra parameter (ρ), and is barely preferred by the AIC criterion, but not by a standard likelihood ratio test. However, the estimated value $\hat{\rho} = 0.91$ has a rather dramatic effect on the shape of the estimated correlation function (Figure 3), and also increase the estimate of τ by a factor of three, which is rather drastic. Next, an anisotropic model was obtained by regressing τ on spatial coordinates, i.e. $\tau(s) = \tau \exp(\gamma_1 s^{(1)} + \gamma_2 s^{(2)})$, where $s = (s^{(1)}, s^{(2)})$ denotes the (normalized) coordinates of a spatial location. This model is just selected by the AIC criterion, and the inclusion of anisotropy has some effect on the estimate of κ. In total, oscillation and anisotropy are not very important elements of the model for this dataset.

Lindgren et al. (2011) used this dataset as an illustration of the computational aspects of their technique, and did not consider alternatives to the Weibull distribution. As our goal is to illustrate the importance of a flexible interface between the user and the computational machinery, we fit a mixture of two Weibull distributions to the data. Under the mixture model, the density of a fully observed survival time is assumed to be

$$p_{\text{mix}} \lambda \alpha y^{\alpha-1} \exp(-\lambda y^\alpha) + (1 - p_{\text{mix}}) \lambda_2 \alpha y^{\alpha-1} \exp(-\lambda_2 y^\alpha), \qquad (11)$$

where $\lambda_2 = c\lambda$, so that $c > 1$ and $p_{\text{mix}} \in (0, 1)$ are additional parameters to estimated. It is seen from Table 1 that the mixture model improves the fit by 41 units on a log-likelihood scale. Hence, the assumption about the observation mechanism is far more important than the spatial correlation structure. Also, allowing more flexibility in the response distribution affects the estimate of spatial variation. The marginal standard deviation of $x(s)/\tau$ is $\sigma = (2\pi\kappa^2\tau^2)^{-1/2}$ which is estimated to be $\hat{\sigma} = 0.39$ for the basis model (fit by Lindgren et al.) and $\hat{\sigma} = 0.25$ for the mixture model. Also, the correlation distance, as measured by κ^{-1}, increases by 18%.

Table 1 also give the computation time (in seconds) for each of the models. The computation time is seen to increase with model complexity as expected.

4. Discussion and conclusion

We have demonstrated that a flexible interface to the Laplace approximation for hierarchical spatial models can be based on AD (automatic differentiation). However, increased flexibility usually comes at the expense of increased complexity, and an important question is whether the user interface presented in Figure 1 is conceived as "complex" by users. As the `sparse_quadratic_prior` is new, and not part of the standard ADMB distribution (but available upon request from the authors) we do not have any actual data on this. However, the `random_effects_vector` statement have been part of ADMB for almost 10 years, so for that we have a much experience dealing with prospective ADMB users from various scientific fields. A clear piece of evidence that the code in Figure 1 is indeed conceived as being complicated is provided by the R package glmmADMB, which is an R interface to ADMB code for fitting zero inflated and over dispersed count models. Since its introduction in 2005 a large number of people has used glmmADMB, and from time to time, users have requested functionality, such as a particular link function, that is not part of glmmADMB. They have been encouraged to modify the underlying ADMB code, which consists only of 500 lines of code of the type shown in Figure 1, but very few of the users of glmmADMB have ever taken the step to modify the underlying ADMB code. This seems to indicate that the ADMB interface is found difficult by the general user of R. We postulate that it is the need build up a joint log-likelihood (g) from an hierarchical description of the model that is the prohibitive factor for users without a particularly strong background in statistics and probability. This view is supported by the fact that the Monte Carlo based software WinBUGS (Lunn et al., 2000), which also has a flexible user interface, has got widespread use in the wider scientific community. WinBUGS allows the user to formulate hierarchical models by assigning distributions to variables in a notation that is intuitive to users. Further comparison of the user interfaces of WinBUGS and ADMB can be found in Bolker et al. (2013).

The user with experience in programming, and with a training in statistics and probability, usually does not find the ADMB user interface difficult. Within this group one finds developers of R packages, and we expect that in the future people will develop mixed model software in R, which calls an

ADMB program that performs the actual model fitting and subsequently return the results to R. This is advantageous from the developer's perspective, as the infrastructure for doing function optimization and Laplace approximation is generically available in ADMB. This allows the number of code lines needed to implement an R package, such as glmmADMB, to be kept low, which is clearly desirable from a software maintenance perspective.

The above considerations also apply to spatial hierarchical models. R packages such as R-INLA, with a user interface that is easy to grasp for people with experience in R, will always attract a larger number of users than a program with a user interface as presented in Figure 1. However, in this paper we have tried to argue that flexibility is needed in statistical practice. Within the ADMB framework, the burden of building and exploring alternative response distributions, say, is put on the user, as ADMB itself only provides generic infrastructure to build and fit hierarchical models. Extra flexibility can of course always be added to R packages, but then the burden of extending the model is put on the developer of the R package, not on the user requesting the extra flexibility.

An important part of the "generic infrastructure" referred to above is automatic and efficient calculation of the gradient of the log likelihood. Most model fitting routines, in R and in other statistical software, that are based on maximizing/minimizing an objective function do not calculate the gradient. As it is well known that the presence of a gradient greatly facilitates the optimization process one must conclude that it is technical difficulties that prevent package developers from evaluating the gradient. In hierarchical models, where the Laplace approximation is utilized, evaluation of the gradient is especially complicated, as (4) and (5) constitute a nested optimization problem. One can hence expect even fewer package developers to take the burden of evaluating the gradient. Our notion of useful "generic infrastructure" is not limited to the gradient, but includes features such as functionality for fitting the model in stages ("phases") as discussed earlier.

We have presented the Laplace approximation, in combination with AD, as a generic way of evaluating the marginal likelihood (2) in hierarchical models with continuous latent random variables. It is well know that the Laplace approximation can be inaccurate in some situations, such as in models with binary responses, where the data often contain relatively little information about each individual latent random variable. It was argued by Skaug and Fournier (2006) that one fairly easily can modify the machinery underlying the Laplace approximation to perform (adaptive) importance sampling, and

still be able to obtain the gradient of the likelihood approximation by AD. Implementation of such methods are somewhat technical in nature, and are in our opinion best hidden behind a software interface.

Development of generic statistical software has a strong sociological component, in that the developer has to imagine how the user will conceive the user interface. WinBUGS is a success story in this respect, with the developers starting out from general principles, and with users in various field at a later stage discovering new ways of applying the software. The range of problems to which WinBUGS has been applied is so broad that the WinBUGS developers could not possible have seen its full extent early on.

There do exist instances in the literature where the Laplace approximation in combination with AD has been used to fit spatial hierarchical models (Kristensen, 2009; Bravington and Hedley, 2009). However, as these authors authors have primarily developed their software for personal use, they have not provided a well documented user interface. Otherwise, comparison to ADMB with respect to interface and computational speed would have been interesting.

Spatial models are playing an increasingly important role in statistical practice, a trend that is fueled by the development of better computational techniques. The discussion in this section has been rather general in nature, but applies strongly to spatial hierarchical models because they are computationally challenging. For instance, in the field of ecology people are indeed applying hierarchical spatial models for individual analyses, but the standard software packages used by most practitioners do not incorporate a fully hierarchical spatial component of the type described in the present paper. Examples include line transect analysis (Thomas et al., 2009) and capture-recapture analysis.

Our conclusion is that that the Laplace approximation, made available through a flexible and generic interface, will make spatial hierarchical models available to a much larger group of researchers. We have given one example of what this interface may look like, and we hope that the presence ADMB can inspire other researchers to develop alternative interfaces.

5. Acknowledgments

We are grateful to Håvard Rue for help with running R-INLA, and to Finn Lindgren for clarifying some aspects of the oscillating correlation function leading to (9).

Agarwal, S., Mierle, K., 2013. Ceres Solver: Tutorial & Reference. Google Inc.

Banerjee, S., Carlin, B., Gelfand, A., 2003. Hierarchical modeling and analysis for spatial data. volume 101. Chapman & Hall/CRC.

Bolker, B., Gardner, B., Maunder, M., Berg, C., Brooks, M., Comita, L., Crone, E., Cubaynes, S., Davies, T., Valpine, P., Ford, J., Gimenez, O., Kery, M., Kim, E., Lennert-Cody, C., Magnusson, A., Martell, S., Nash, J., Nielsen, A., Regetz, J., Skaug, H., Zipkin, E., 2013. Strategies for fitting nonlinear ecological models in R, AD Model Builder, and BUGS. Methods in Ecology and Evolution To appear.

Bravington, M., Hedley, S., 2009. Antarctic minke whale abundance estimates from the second and third circumpolar IDCR/SOWER surveys using the SPLINTR model. Technical Report. Paper SC/61/IA14 presented to the IWC Scientific Committee.

Butler, R., 2007. Saddlepoint Approximations with Applications. Cambridge University Press.

Fournier, D., Skaug, H., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M., Nielsen, A., Sibert, J., 2012. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. Optimization Methods and Software 27, 1–17. http://www.tandfonline.com/doi/pdf/10.1080/10556788.2011.597854.

Griewank, A., Walther, A., 2008. Evaluating derivatives: principles and techniques of algorithmic differentiation. Society for Industrial and Applied Mathematics (SIAM).

Kristensen, K., 2009. Statistical aspects of heterogeneous population dynamics. Ph.D. thesis. University of Copenhagen.

Lindgren, F., Rue, H., Lindström, J., 2011. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73, 423–498.

Lunn, D., Thomas, A., Best, N., Spiegelhalter, D., 2000. WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. Statistics and Computing 10, 325–337.

McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models, 2nd Ed. New York: Chapman & Hall.

R Development Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Rue, H., Martino, S., Chopin, N., 2009. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. Journal of the royal statistical society: Series B (statistical methodology) 71, 319–392.

Skaug, H., Fournier, D., 2006. Automatic approximation of the marginal likelihood in non-gaussian hierarchical models. Computational Statistics & Data Analysis 56, 699–709.

Stan Development Team, 2013. Stan: A C++ Library for Probability and Sampling, Version 1.1.

Thomas, L., Buckland, S., Rexstad, E., Laake, J., Strindberg, S., Hedley, S., Bishop, J., Marques, T., Burnham, K., 2009. Distance software: design and analysis of distance sampling surveys for estimating population size. Journal of Applied Ecology 47, 5–14.

Table 1: Parameter estimates for different models fit to the Leukemia example of Lindgren et al. (2011, sec. 2.4). Results for the basis model (fit by Lindgren et al.) are given for both ADMB and R-INLA (with option `int.strategy=''eb''`) for comparison. R-INLA is run with its empirical Bayes option (`int.strategy=''eb''`). Standard deviations (parentheses) are only shown for the basis model. Log-likelihood deviances for all ADMB fits are shown relative to the basis model. Run times for ADMB are given in seconds.

	Lindgren et al.		No. spat.	Oscilat.	Anisotr.	Mixture
	R-INLA	ADMB				
β_0	-5.5405	-5.6877 (0.2314)	-5.4204	-5.6648	-5.6917	-9.0451
β_{sex}	0.0702	0.0715 (0.0693)	0.0672	0.0665	0.0670	0.1331
β_{age}	0.0321	0.0326 (0.0023)	0.0300	0.0324	0.0328	0.041
β_{wbc}	0.003	0.0031 (0.0005)	0.0029	0.0031	0.0031	0.0052
β_{tpi}	0.0244	0.0249 (0.0099)	0.0251	0.0251	0.0242	0.041
τ	0.0956	0.0851 (0.0450)		0.3136	0.0997	0.1789
κ	10.5717	12.605 (6.162)		11.438	13.74	8.9815
α	0.5789	0.5956 (0.0163)	0.5753	0.5926	0.5969	0.7489
ρ				0.9067		
p_{mix}						0.1579
c_{mult}						9.4953
γ_1					0.3896	
γ_2					-1.7741	
Deviance		0.0	-10.86	1.44	2.12	41.19
Run time		31	0.12	68	59	46

Figure 1: ADMB program for fitting the Lindgren et al. (2011) model to the Leukemia data. The code has been slightly simplified, with a few larger code omissions denoted by "...". As in standard C++ a "//" marks the beginning of a comment extending to the end of the line. The number of code lines in the full program is 124 (excluding blank lines). In `DATA_SECTION` an `init_` defines an object to be read from file, while in `PARAMETER_SECTION` an `init_` indicates a parameter. Dimensions and parameter bounds are specified in parenthesis.

```
DATA_SECTION
  init_int n                    // Number of observations
  init_vector y(1,n)            // Response (survival time)
  init_ivector notcens(1,n)     // Cencoring indicator (values = 0,1)
  init_ivector meshidxloc(1,n) // Pointers into x (latent r.v.'s)
  init_int n_p                  // # columns in X
  init_matrix X(1,n,1,n_p)      // Desing matrix
  ...                           // Makes X orthonormal
  ...                           // Read M0, M1, M2.
PARAMETER_SECTION
  init_vector beta(1,n_p)                   // Regression parameters
Regression parameters
  init_bounded_number log_tau(-3.0,-1.0)    // log(tau)
  init_bounded_number log_kappa(2.0,3.0)    // log(kappa)
  init_bounded_number alpha(0.1,10.0)
  random_effects_vector x(1,n)              // GMRF vector
  sparse_quadratic_prior Q(x)               // Assciates Q with x
  objective_function_value g                // Negative log likelihood
PROCEDURE_SECTION                           // Where g is evaluated
  for (int i=1;i<=n;i++)                    // Loop over observations
    ll_weibull(i,beta,alpha,log_tau,x(meshidxloc(i)));
  ...                                       // Evaluates Q from M0, M1, M2.
SEPARABLE_FUNCTION void ll_weibull(i,...)  // Weibull log likelihood
  tau = exp(log_tau);
  eta = X(i)*beta + x(i)/tau;               // X(i) is i' row
  lambda = exp(eta);                        // Scale parameter
  y_alpha = exp(log(y(i))*alpha);           // Temporary variable
  S = mfexp(-lambda*y_alpha);               // Survival funciton
  f = lambda*alpha*y_alpha/y(i)*S;          // Probability density
  if(notcens(i))
    g -= log(f);                            // Adds ll contribution to g
  else
    g -= log(S);
```

Figure 2: Extension of Figure 1 with ADMB code needed to implement the two component Weibull mixture model (11). Only additions to Figure 1 are shown, and as before // marks the beginning of a comment (standard C++).

```
PARAMETER_SECTION
  init_bounded_number p_mix(.0001,.999)    // Mixture probability
  init_bounded_number mult(1.0,30.0)       // c in (11)
SEPARABLE_FUNCTION void ll_weibull(i,...)
  S1 =   exp(-lambda*t_alpha);             // Survival function 1
  S2 =   exp(-mult*lambda*t_alpha);        // Survival function 2
  S = p_mix*S1 + (1.0-p_mix)*S2;           // Marginal survival
  f1 = lambda*alpha*t_alpha/y(i)*S1;       // Density 1
  f2 = mult*lambda*alpha*y_alpha/y(i)*S2;  // Density 2
  f = p_mix*f1 + (1.0-p_mix)*f2;           // Marginal density
  if(notcens(i))
    g -= log(f);                           // Adds ll contribution to g
  else
    g -= log(S);
```

Figure 3: Histogram of observed pairwise distances in the Leukemia data overlaid by the fitted correlation functions for the basis model ($\rho = 0$) and the oscillating correlation function ($\hat{\rho} = 0.91$). The vertical axis is common to the histogram and the two correlation functions. Distances have been normalized so the horizontal axis does not have a unit.