Using NIMBLE to implement Markov Chain Monte Carlo with Integrated Nested Laplace approximation

Kwaku Peprah Adjei[1,2], Robert B. O'Hara[1,2]

[1]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim Norway

[2]Center for Biodiversity Dynamics, Norwegian University of Science and Technology, Trondheim Norway

Summary

1. Bayesian inference using Markov Chain Monte Carlo (MCMC) and Integrated Nested Laplace approximation (INLA) are popular in applied statistics. MCMC can be slow to run for some models and INLA is also limited in the class of models it can fit. However, it is possible to combine INLA with MCMC to fit the class of models that INLA cannot fit. The MCMC with INLA approach involves dividing the set of parameters we aim at estimating into two: one that needs to sampled from MCMC before INLA can be used to fit a model given the sampled parameters.

2. In this study, we propose that NIMBLE be used to implement the MCMC with INLA methodology. We do this by proposing two approaches: a) using INLA-defined functions to write customized samplers in NIMBLE and b) writing a NIMBLE distribution function with the embedded INLA function. The approaches are applied to various class of models.

3. The posterior marginals of the model parameters from the MCMC with INLA using NIMBLE approaches were consistently similar to those from MCMC and/or INLA only.

keywords: Bayesian inference, Metropolis-Hastings algorithm, spatial occupancy model, binomial N-Mixture model

1 Introduction

Bayesian inference has gained popularity in applied statistics, with its applications spanning the breadth of multiple disciplines. Their popularity can be attributed to the flexibility it provides statisticians in fitting models (Blangiardo and Cameletti, 2015). Markov chain Monte Carlo (MCMC) method is one of the common approaches used for Bayesian inference (Blangiardo and Cameletti, 2015; Brooks et al., 2011; Gilks et al., 1995; Kéry and Royle, 2015), due to its easy implementation with open software like JAGS (Hornik et al., 2003), WinBUGS (Spiegelhalter et al., 2014), Stan

(Carpenter et al., 2017), NIMBLE (de Valpine et al., 2017). MCMC runs faster for simple models; however, using MCMC to fit complex models such as spatio-temporal state space models with large datasets can be very time consuming (Blangiardo and Cameletti, 2015; Kéry and Royle, 2020).

Alternatively, integrated nested Laplace approximation (INLA) was proposed by Rue et al. (2009) for Bayesian inference on hierarchical models that can be represented as latent Gaussian models. INLA provides a deterministic or numerical algorithm that approximates the marginal distribution of the latent states, which is the objective of Bayesian inference (Blangiardo and Cameletti, 2015). Models fitted with INLA take a fraction of time MCMC methods take (Blangiardo and Cameletti, 2015; Gómez-Rubio and Rue, 2018), but still provides accurate estimates of model parameters (Berild et al., 2022; Gómez-Rubio and Rue, 2018). In addition, INLA is available in an R-package **R-INLA** (Rue et al., 2009), providing users with the platform to fit complex hierarchical models in a matter of seconds. These reasons have made INLA a popular method to fit hierarchical models. Notwithstanding, there is a limitation in the class of models that can be fitted with **R-INLA**. For instance, models with missing covariates and mixture models cannot be fitted with **R-INLA** (Berild et al., 2022; Gómez-Rubio and Rue, 2018; Marin et al., 2005).

To widen the scope of models that can be fitted using INLA, attempts have been made at fitting conditional linear Gaussian models with **R-INLA** (Berild et al., 2022; Bivand et al., 2014; Gómez-Rubio and Rue, 2018; Li et al., 2012). These conditional models are developed by fixing some of the parameters in the full model. The values of the fixed parameters can be obtained from their maximum likelihood estimates (Li et al., 2012) or from their posterior distribution estimated with either MCMC (Bivand et al., 2014; Gómez-Rubio and Rue, 2018) or other Monte Carlo methods like (adaptive) importance sampling (Berild et al., 2022). The values of the fixed parameters can be plugged into the overall model, and the conditional model can then be fitted with **R-INLA**.

This study will focus on the approach by Gómez-Rubio and Rue (2018), where the marginal likelihood of the fitted conditional model is computed with **R-INLA**, and this marginal likelihood is used to calculate the acceptance probability in the Metropolis-Hastings algorithm for the MCMC method. Our study provides further advancement in the development of this INLA-MCMC methodology by proposing that NIMBLE (a system for programming statistical algorithms for general model structures within R; de Valpine et al. (2017)) can be used to implement this approach. The reasons for this proposal are discussed as follows.

Firstly, NIMBLE is designed to provide flexible model specifications by extending the BUGS language to include R-defined functions (de Valpine et al., 2023). NIMBLE also provides a programming language for algorithms and a balance between high-level programmability and execution efficiency by compiling models and functions via C++ (de Valpine et al., 2017, 2023). This means that the computational time of fitting models with the MCMC-INLA method can be considerably reduced.

In addition, NIMBLE has been implemented as an R-package, **nimble** (de Valpine et al.), making both NIMBLE and INLA readily accessible to users on the same platform. As mentioned already, NIMBLE provides a platform to integrate R-defined functions within its BUGS framework (de Valpine et al., 2023). This means the conditional model that we intend to fit using **R-INLA** can be written as a function in *R* (R Core Team, 2023) and integrated within the BUGS code to be run with **nimble**. This makes it possible to utilize the several sampling algorithms implemented in R-package **nimble** - without writing a new sampling algorithm - to obtain the posterior distribution of the fixed parameters, instead of just the Metropolis-Hasting algorithm used by Gómez-Rubio and Rue (2018). The implemented samplers in **nimble** include adaptive (block) random walk, slice sampler, conjugate ("Gibbs") samplers, binary sampler, cross level sampler, categorical sampler, elliptical slice sampler, automated factor slice sampler, Hamiltonian Monte Carlo sampler (de Valpine et al., 2023).

New sampling algorithms other than those implemented in **nimble** will be needed for the INLA-MCMC methodology in some instances. These instances include customizing the sampling algorithm to reduce the number of calls made to **R-INLA**, either by parallelizing the method by using importance sampling instead of MCMC (Berild et al., 2022), or specifying how many times **R-INLA** should be called by a sampler. This is possible with **nimble**, since NIMBLE provides a simple platform for users to easily implement a customized sampler of their choice (the reader is referred to the NIMBLE manual de Valpine et al. (2023) for further details on how to write customized samplers in NIMBLE).

This study proposes two alternative ways to implement the INLA-MCMC approach with NIMBLE. The first alternative involves writing an **R-INLA** function to return the marginal likelihood and samples from the latent states posterior distribution obtained from the fitted conditional model. A customised sampler is then defined to use this marginal likelihood in the sampler's decision process to generate samples of the fixed parameters. If the proposed samples of the fixed parameters are accepted, then the samples from the fitted conditional models are also saved; if the proposed samples are rejected, the posterior samples from the fitted conditional models are not saved. We write a customized random-walk block (RW) to implement this alternative.

The second alternative involves writing an **R-INLA** function to fit the conditional model, and then defining a NIMBLE distribution function that uses this defined **R-INLA** function. The distribution function is integrated into the BUGS code that will be used to fit the model with **nimble**. This alternative does not require new samplers to be written in **nimble**, but it utilizes the sampling algorithms already implemented in the R-package **nimble**.

We apply the two alternative approaches to fit various models with two simulated datasets and three real datasets. The bivariate regression model is fitted to the first simulated dataset and spatial occupancy model is fitted to the second simulated dataset. The proposed approaches are also used to fit the Bayesian lasso regression model to the *Hitters* dataset (Gareth et al., 2013), impute missing covariates with the *nhanes* dataset (accessed from Van Buuren and Groothuis-Oudshoorn, 2011), zero-inflated Poisson model with data on the number of fishes caught (data retrieved from Berild et al., 2022) and a binomial N-Mixture model for the mallard dataset (accessed from R-package **unmarked** Fiske and Chandler, 2011). The results from our proposed approaches are compared to those obtained from MCMC methods using **nimble** and maximum likelihood estimation from some R-packages. This study does not make comparison with the INLA with Metropolis-Hastings approach by Gómez-Rubio and Rue (2018), as we are only interested in implementing the INLA-MCMC methodology with **nimble**. Various summaries of the posterior marginal of the model parameters are presented, either numerically or graphically.

2 Integrated Nested Laplace Approxmation

Let $\mathbf{y} = \{y_1, \dots, y_n\}$ be observations from an exponential family, with each observation y_i for $i = 1, 2, \dots, n$ having mean μ_i. The mean μ_i is related to some linear predictor (η_i), defined as a combination of covariates and other functions using an additive structure and an appropriate link function g(.). The linear predictor can be defined as:

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m u_{mi} + \sum_{j=1}^{L} f_j(v_{ji}) + \epsilon_i, \tag{1}$$

where β_0 is the intercept, β_m is the linear effects of covariate u_m on the response, $f_l(.)$ defines the set of functions of covariate v_j which can be a smooth or non-linear effect, trend effect, etc. The unobserved latent effects η, β_0, $\{\beta_m\}_{m=1}^{M}$ are represented by the vector $\mathbf{x}$ in this study. Moreover, the conditional distribution of the latent field and the likelihood are assumed to depend on some hyperparameters, say $\theta_{\mathbb{1}}$ and θ_2 (which could be the residual variance) respectively. We use θ as an assemblage for both hyperparameters, i.e, $\theta = (\theta_{\mathbb{1}}, \theta_2)$.

It is clear from equation (1) that the observations are conditionally independent given the latent states and the hyperparameters. Therefore, the likelihood can be written as:

$$\pi(\mathbf{y}|\mathbf{x}, \theta) = \prod_{i \in I} \pi(y_i|x_i, \theta). \tag{2}$$

With an appropriate prior specified for the hyperparameters θ, the posterior distribution of the latent states and hyperparameters (which we aim to estimate) is obtained from equation (2) using the Bayes rule:

$$\pi(\mathbf{x}, \theta|\mathbf{y}) \propto \pi(\mathbf{x}|\theta)\pi(\theta) \prod_{i \in I} \pi(y_i|x_i, \theta), \tag{3}$$

where $\pi(\theta)$ is the prior distribution of θ. The joint posterior distribution defined in equation (3) is not always in closed form. Rue et al. (2009) proposed an approximation based on the Laplace approximation to estimate the marginal distribution of latent effects and hyperparameters. As such, the vector of latent effects is assumed to be Gaussian Markov random field (GMRF) with zero mean and precision matrix ($\mathbf{Q}(\theta)$).

With this assumption, the posterior distribution defined in equation (3) can be re-written as:

$$\pi(\mathbf{x}, \theta|\mathbf{y}) \propto \pi(\theta)|\mathbf{Q}(\theta)|^{1/2} exp\Big\{-\frac{1}{2}\mathbf{x}^T\mathbf{Q}(\theta)\mathbf{x} + \sum_{i \in I} \log(\pi(y_i|x_i, \theta))\Big\}. \tag{4}$$

The marginal distributions of the latent state and hyperparameters are estimated from the joint posterior distribution. INLA will aim at providing approximations to $\pi(\theta|\mathbf{y})$ and $\pi(\mathbf{x}|\mathbf{y})$. The posterior marginal for a hyperparameter θ_k is obtained as follows:

$$\pi(\theta_k|\mathbf{y}) = \int \pi(\theta|\mathbf{y}) d\theta_{-k}, \tag{5}$$

where θ_{-k} are the vector of hyperparameter without element θ_k. The joint posterior of the hyperparameters are approximated, as proposed by Rue et al. (2009), with:

$$\tilde{\pi}(\theta|\mathbf{y}) \propto \frac{\pi(\mathbf{x}, \theta, \mathbf{y})}{\tilde{\pi}_G(\mathbf{x}|\theta, \mathbf{y})}\Bigg|_{\mathbf{x}=\mathbf{x}^\star(\theta)}, \tag{6}$$

where $\tilde{\pi}_G(\mathbf{x}|\theta, \mathbf{y})$ is the Gaussian approximation to the latent effect full conditional distribution and $\mathbf{x}^\star(\theta)$ is the model of the full conditional for a given value of the vector of hyperparameters.

The posterior marginal for latent state x_l is defined as:

$$\pi(x_l|\mathbf{y}) = \int \pi(x_l, \theta|\mathbf{y})\pi(\theta|\mathbf{y}) d\theta \tag{7}$$

and this is approximated by integrating out the hyperparameters and marginalizing over the latent effects. INLA uses the following approximation for the latent state posterior:

$$\pi(x_l|\mathbf{y}) \approx \sum_{k=1}^{K} \tilde{\pi}(x_l|\theta^{(k)}, \mathbf{y})\tilde{\pi}(\theta^{(k)}|\mathbf{y})\Delta_k, \tag{8}$$

where $\theta^{(k)}$ represent the values of θ used for the numerical integration, each with an associated weight Δ_k.

3 NIMBLE with INLA

As discussed in the introduction, Gómez-Rubio and Rue (2018) proposed that MCMC can be combined with INLA to fit complex Bayesian models. Their proposed methodology assumed that the model could not be fitted with **R-INLA** unless some of the hyperparameters or latent effects are fixed. Let $\mathbf{z}_c$ be the full collection of latent effects and hyperparamters that needs to be fixed before

R-INLA can be used to fit the model, and $\mathbf{z}_{-c}$ be the full collection of latent effects and hyperparameters that will estimated using **R-INLA**. The full collection of latent effects and hyperparameters will therefore be $\mathbf{z} = \{\mathbf{z}_c, \mathbf{z}_{-c}\}$. The posterior distribution of $\mathbf{z}$ can be split as:

$$\pi(\mathbf{z}|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{z}_c, \mathbf{z}_{-c})\pi(\mathbf{z}_{-c}|\mathbf{z}_c)\pi(\mathbf{z}_c), \tag{9}$$

and integrating over $\mathbf{z}_{-c}$ conditional on $\mathbf{z}_c$, we obtain the posterior of $\mathbf{z}_c$:

$$\pi(\mathbf{z}_c|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{z}_c)\pi(\mathbf{z}_c). \tag{10}$$

This suggests that conditional models can be fitted with **R-INLA** and conditional marginal likelihood ($\pi(\mathbf{y}|\mathbf{z}_c)$) can also be computed.

To combine MCMC with INLA, Gómez-Rubio and Rue (2018) proposed a blocked Metropolis-Hasting algorithm for their multivariate parameter $\mathbf{z}_c$. At each time step, a new value is proposed for the ensemble $\mathbf{z}_c$, and they are all either accepted or rejected with probability α. This acceptance probability (α) depends on the conditional marginal likelihoods given $\mathbf{z}_c$ and $\mathbf{z}_{-c}$: $\pi(\mathbf{y}|\mathbf{z}_c)$ and $\pi(\mathbf{y}|\mathbf{z}_{-c})$ respectively (Gómez-Rubio and Rue, 2018). These conditional marginal likelihood $\pi(\mathbf{y}|\mathbf{z}_{-c})$ can be approximated with **R-INLA**.

In this study, we intend to use **nimble** to implement this INLA-MCMC approach by drawing samples from the posterior distribution of $\mathbf{z}$ by sampling $\mathbf{z}_c$ from MCMC and $\mathbf{z}_c$ from the conditional fitted model using **R-INLA**. We define two alternative approaches used to achieve this aim in sections 3.1 and 3.2.

3.1 Approach One: Using R-INLA functions in customised samplers

The random walk (block) sampler was chosen to implement the Metropolis-Hastings approach used by Gómez-Rubio and Rue (2018). The process of integrating the INLA function within the sampler used for the INLA-MCMC methodology is shown in Figure 1.

The random-walk block proposal is a Metropolis-Hastings algorithm, and following the discussion by Gómez-Rubio and Rue (2018), proposal distributions need to be chosen to propose new values of $\mathbf{z}_c$. This proposed value (say $\mathbf{z}_c^\star$) will be accepted or rejected with acceptance probability:

$$\alpha = \min\left\{1, \frac{\pi(\mathbf{y}|\mathbf{z}_c)\pi(\mathbf{z}_c^\star)q(\mathbf{z}_c^{(j)}|\mathbf{z}_c^\star)}{\pi(\mathbf{y}|\mathbf{z}_c^{(j)})\pi(\mathbf{z}_c^{(j)})q(\mathbf{z}_c^\star|\mathbf{z}_c^{(j)})}\right\}, \tag{11}$$

where $q(.|.)$ is the proposal distribution; $\pi(\mathbf{y}|\mathbf{z}_c^{(j)})$ and $\pi(\mathbf{y}|\mathbf{z}_c)$ are marginal distributions approximated with **R-INLA**; and $\pi(\mathbf{z}_c^\star)$ and $\pi(\mathbf{z}_c)$ are the prior distributions of $\mathbf{z}_c^\star$ and $\pi(\mathbf{z}_c)$ respectively.

For each step j of the MCMC, the conditional marginal distribution on $\mathbf{z}_c^{(j)}$ is approximated by

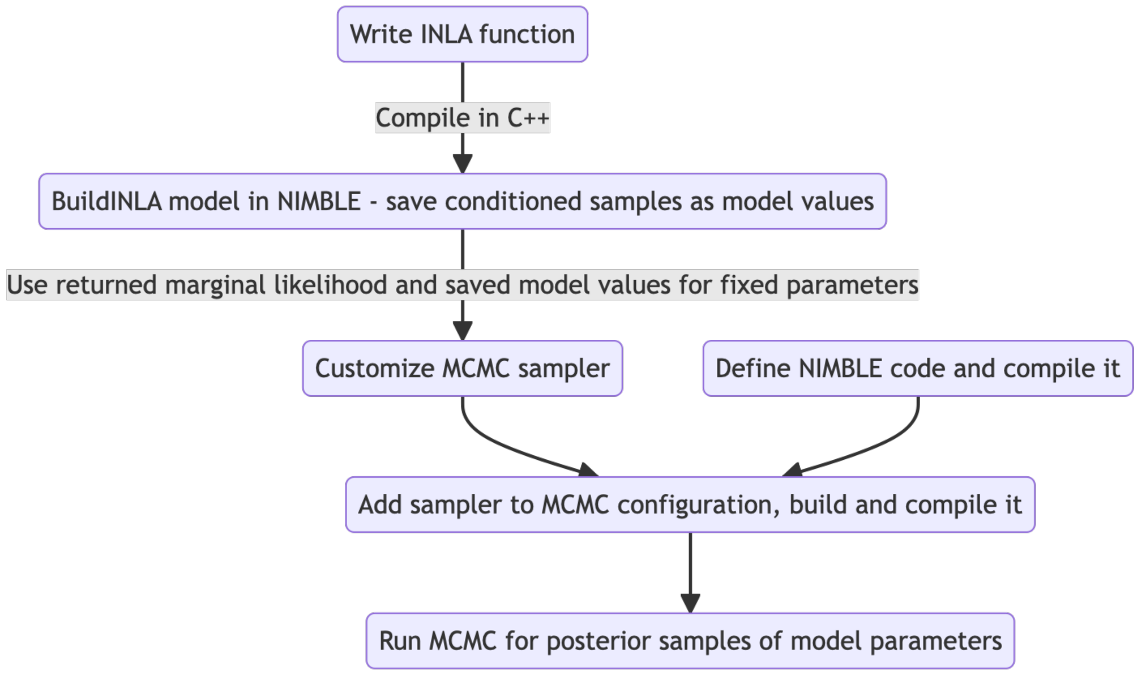

Figure 1: Flowchart showing how models can be fitted using INLA_MCMC framework implemented with NIMBLE. The process begins by writing and compiling INLA model written as an R-function. Then a NIMBLE model sampler is written to use the returned output from the INLA function. This customised sampler is assigned to a compiled NIMBLE code and MCMC is run to obtain posterior samples of all model parameters of interest.

integrating over $\mathbf{z}_c$:

$$\begin{aligned}\pi(z_{-c,k}|\mathbf{y}) &= \int \pi(z_{-c,k}|\mathbf{z}_c, \mathbf{y})\pi(\mathbf{z}_c|\mathbf{y})d\mathbf{z}_c \\ &= \frac{1}{N}\sum_{j=1}^{N} \pi(z_{-c,k}|\mathbf{z}_c^j, \mathbf{y}),\end{aligned} \tag{12}$$

where N is the number of samples of the posterior distribution of z_c. This implies that the marginal of $z_{-c,k}$ can be obtained via Bayesian model averaging (BMA). In **R-INLA**, posterior estimates can be obtained from functions *inla.emarginal* (for posterior mean) and *inla.zmarginal* (for several other summary statistics). In this study, however, we draw samples from the posterior distribution of $z_{-c,k}$ from the fitted conditional models using *inla.samples* function in **R-INLA** at each iteration.

Algorithm 1 illustrates how the random walk block sampler is customized to take into account the marginal likelihood and samples from the posterior distribution of $z_{-c,k}$ from **R-INLA**.

3.2 Approach two: Writing a distribution function with an embedded INLA function

The first approach requires the user to have some competency in writing algorithms in NIMBLE. Alternatively, we explored the option of writing a NIMBLE distribution function that includes the **R-INLA**-defined functions. The **R-INLA** function returns the conditional marginal ($\pi(y|\mathbf{z}_c)$) and

Algorithm 1 Metropolis Hastings with INLA using random walk block sampler

for i in $1 : n.iter$ **do**
  if i = 1 **then**
    $\mathbf{z}_c := \mathbf{0}$
  end if

  Propose $\mathbf{z}_c^\star \sim N(0, \Sigma)$.

  Fit conditional model with $\mathbf{z}_c^\star$ as fixed values using **R-INLA** and extract $\pi(\mathbf{y}|\mathbf{z}_c)$ and a sample from the posterior distribution $pi(z_{-c}^\star)$.

  Calculate acceptance probability with equation (11).

  Generate $u \sim Uniform(0, 1)$

  if $\alpha < u$ **then**
    Set $\mathbf{z}_c := \mathbf{z}_c^\star$, $\pi(\mathbf{y}|\mathbf{z}_c) := \pi(\mathbf{y}|\mathbf{z}_c^\star)$ and $\mathbf{z}_{-c} := \mathbf{z}_{-c}^\star$
  else
    set $\mathbf{z}_c := \mathbf{z}_c$, $\pi(\mathbf{y}|\mathbf{z}_c) := \pi(\mathbf{y}|\mathbf{z}_c^{(i-1)})$ and $\mathbf{z}_{-c} := \mathbf{z}_{-c}^{i-1}$
  end if
end for

samples of $\mathbf{z}_{-c}$ from its posterior distribution. The response variable $\mathbf{y}$ in the BUGS code is assigned this written NIMBLE distribution. The BUGS code can then be compiled, set up to run with MCMC configurations using the sampling algorithms implemented in **nimble**, as shown in Figure 2.

New distributions for use in BUGS code are set up for use in NIMBLE via the *registerDistributions* function in **nimble** (for further details on writing new distributions in NIMBLE, see Chapter 12.2 of de Valpine et al., 2023). Here is a simple example to illustrate how the NIMBLE distribution can be defined to integrate the **R-INLA** function.

```
# Write conditional model to be fitted with INLA as a function
fitInla <- function(beta, y, covariate){
  #fit inla model
  res <- inla(y ~ 1 + beta*covariate,
              ...)
  #return marginal likelihood estimate
  mld <- res$mld[1,1]
  return(mld)
}

# Convert the fitInla function to a NimbleFunction
nimbleINLA <- nimble::nimbleRcall(
  prototype = function(
 beta = double(0), #beta is a scalar
 y = double(1), #y is a vector
```

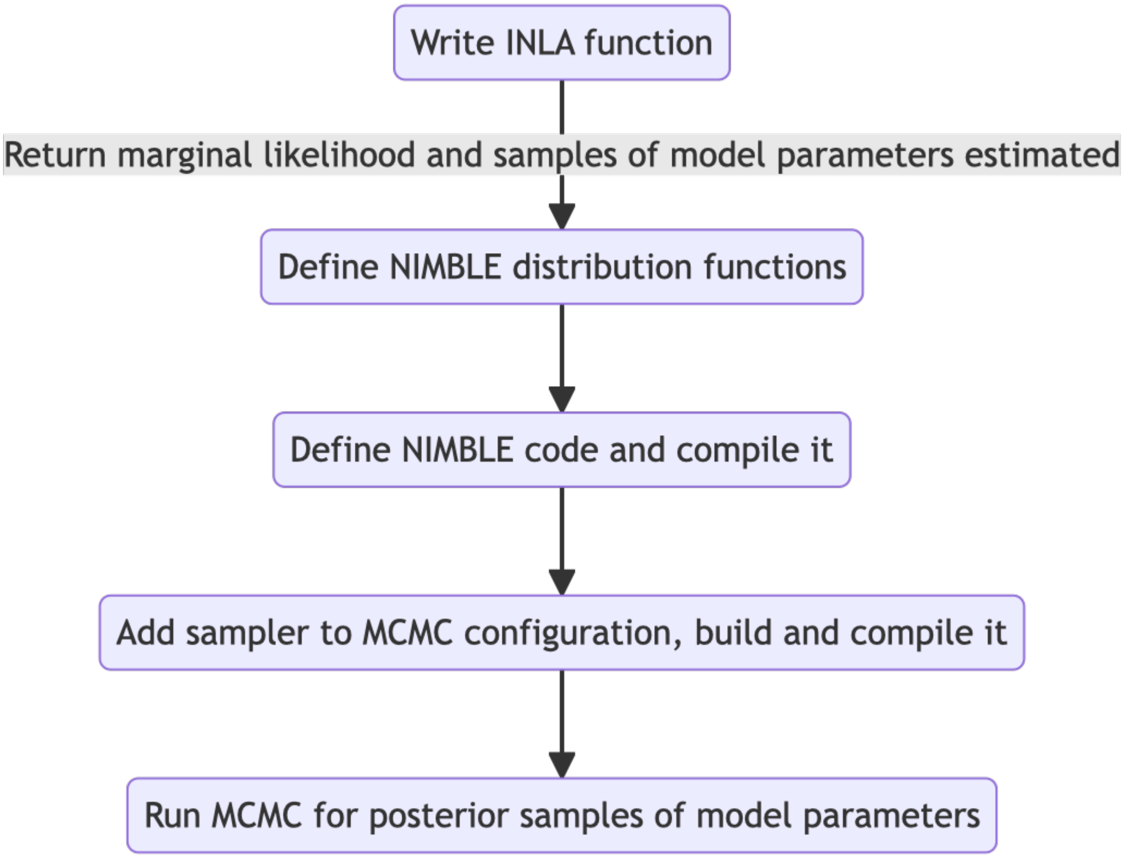

Figure 2: Flowchart showing how models can be fitted using INLA-MCMC methodology with alternative two. The process begins by writing and compiling an **R-INLA** function that fits the conditional model. New NIMBLE distributions that integrates the INLA function are defined for use in BUGS code, and ths BUGS code is configured to run in NIMBLE.

```
 x = double(1) #x is a vector
  ) {},
  returnType = double(0), # outcome is a scalar
  Rfun = 'fitInla'
)
# Define density function
dINLA <- nimbleFunction(
  run = function(y = double(1),
                 beta = double(0),
                 covariate = double(0),
                 log = logical(0, default = 0)) {
    returnType(double())
#run the INLA function and return marginal likelihood
    mld <- nimbleINLA(beta, y,covariate)
    if(log) return(mld)
    return(exp(mld))
  })

#Register the distributions
registerDistributions(list(
  dSpatial = list(
    BUGSdist = "dINLA(beta, covariate)",
    discrete = TRUE,
    range = c(-Inf, Inf),
    types = c('value = double(1)', 'covariate = double(1)', 'beta = double(0)')
  )))
```

4 Examples

We illustrate the proposed method with six examples. The first two examples are simulation studies on bivariate regression model from Gómez-Rubio and Rue (2018) and spatial occupancy model from Kéry and Royle (2020). The method was also applied to four models with datasets: Bayesian lasso regression with *Hitters* dataset (Gareth et al., 2013), imputation of missing covariates with *nhanes* dataset accessed from the R-package **mice** (Van Buuren and Groothuis-Oudshoorn, 2011), zero-inflated Poisson model and Binomial N-Mixture model with mallard data (Fiske and Chandler, 2011).

The various models are fitted with the proposed methodology described in section 3. Specifically. we fit approach one (described in section 3.1) with the customized RW-block samplers (referred to as `iNim-RW` in the rest of this paper). In addition, we fit approach two (described in section 3.2) with **nimble**'s implemented RW_block sampler assigned to $\mathbf{z}_c$ (we refer to this approach as `iNim2-RW` in the rest of this paper). We also fit bivariate regression model with **R-INLA**, the lasso regression with the R-package **glmnet** (Friedman et al., 2010), the zero-inflated Poisson regression with **pscl**(Jackman, 2020) and the Binomial N-mixture model with R-package **unmarked** (Fiske and Chandler, 2011) to compare with the results we obtain from the proposed framework. To assess the performance of the various approaches, the efficiency of each method is estimated. The efficiency is calculated as the effective sample size divided by the time used to generate the samples. The effective sample size was estimated with the R-package **mcmcse** (Flegal et al., 2021).

For all models fitted with NIMBLE, one chain is run for 100500 iterations and the first 50500 samples are discarded as burn-in samples. We keep every fifth of the remaining samples for inference.

4.1 Simulation Study

4.1.1 Bivariate regression model

We use the simulation study on Bivariate regression model in Gómez-Rubio and Rue (2018) as our first example. The aim of the simulation is to compare the marginal distribution of model parameters and joint posterior distribution of the covariate effect parameters estimated from the fitted model using INLA, MCMC and INLA-MCMC methods.

We simulate 100 observations with two covariates $\mathbf{u_1}$ and $\mathbf{u_2}$ as follows:

$$y_i = \alpha + \beta_1 u_{1i} + \beta_2 u_{2i} + \epsilon_i; \quad i = 1, \dots, 100 \tag{13}$$

where $\mathbf{u_1}$ and $\mathbf{u_2}$ are simulated from uniform distribution between 0 and 1, and ϵ_i is a Gaussian random term with mean zero and variance $1/\tau$. Moreover, we choose $\alpha = 2$, $\beta_1 = 3$, $\beta_2 = -3$ and $\tau = 1$.

As noted by both Gómez-Rubio and Rue (2018) and Berild et al. (2022), INLA can easily fit this model. We split the entire parameters in the model $\mathbf{z} = \{\alpha, \beta_1, \beta_2, \tau\}$ into two mutually exclusive sets: $\mathbf{z}_c = \{\beta_1, \beta_2\}$ and $\mathbf{z}_{-c} = \{\alpha, \tau\}$, similar to what was done by Gómez-Rubio and Rue (2018) and Berild et al. (2022). New samples of $\mathbf{z}_c$ are proposed from a multivariate normal distribution with zero mean and covariance matrix $\Sigma = 0.75^2 I$, where I is an identity matrix.

Figure 3 shows the marginal distributions of the four model parameters in the bivariate regression model defined in equation (13), with the joint distribution of β_1 and β_2 presented in Figure 4. The effective sample sizes and time taken for the model to be fitted with each study method is presented in Table 1. In all cases, the posterior marginals of the model parameters are very similar. Moreover,

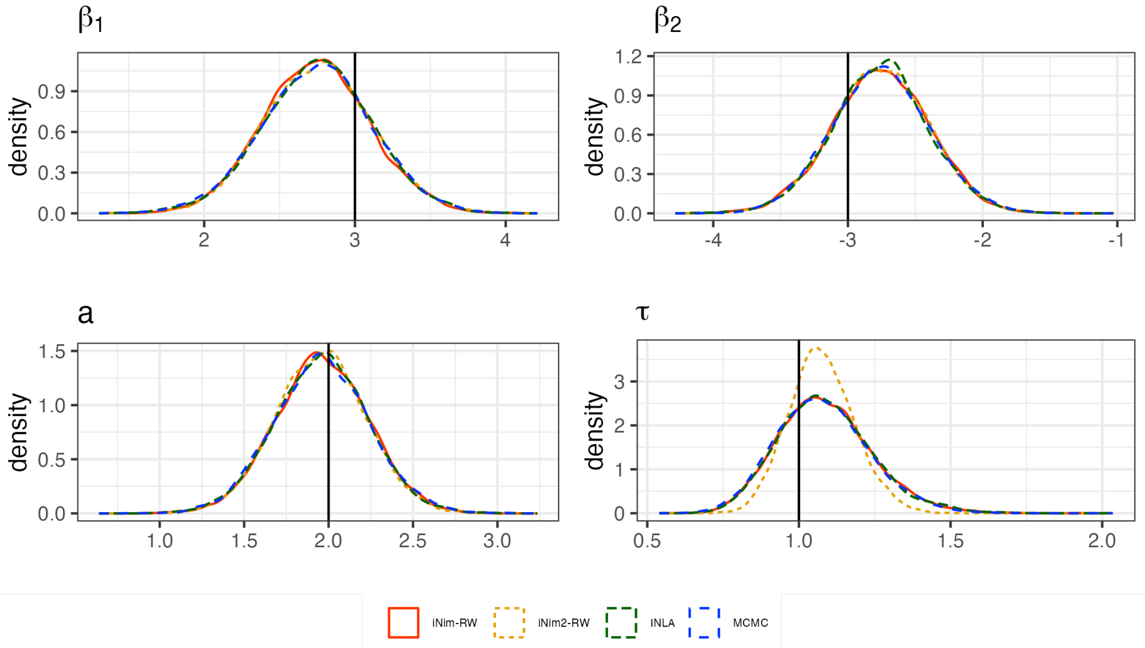

Figure 3: Posterior marginal distribution of the parameters in the bivariate regression model. The distributions are coloured by the method used to fit the model and the solid vertical line represents the true model parameter value used to simulate the data.

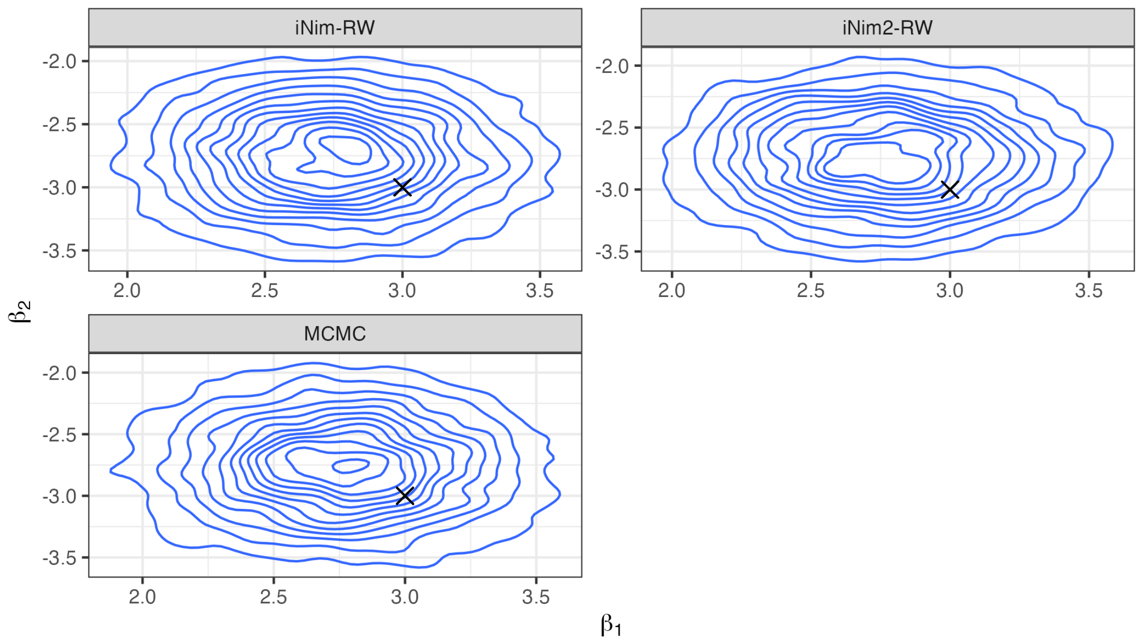

Figure 4: Joint posterior distribution of the covariate effects β_1 and β_2 from the bivariate regression model. The true value is indicated with the black cross.

the effective sample size is higher for the first approach (iNim-RW) than the second approach (iNim2-RW) and MCMC. This implies that to achieve the same number of independent observations from the posterior distribution, iNim-RW would require less iterations to do that.

Although iNim-RW, iNim2-RW and MCMC methods were compiled and run in C++, MCMC with **nimble** was the fastest in this example (2.87 seconds). Fitting the conditional model with **R-INLA** is very fast, however the implementation of the INLA-MCMC approaches with NIMBLE calls the **R-INLA** sequentially. As a result, iNim-RW and iNim2-RW took over 37 hours to run for 100500 iterations. This long computational time of the INLA-MCMC method has also been noted by Berild et al. (2022).

Table 1: Effective sample size (with efficiency provided in parenthesis) of each model parameter and time taken by each of the methods in the study. Efficiency is calculated as the effective sample size divided by the computational time.

Parameter	iNim-RW	iNim2-RW	MCMC
α	5094.29 (0.02)	4316.95 (0.02)	688.6 (313.42)
β_1	5143.61 (0.02)	4487.63 (0.02)	1140.77 (519.22)
β_2	5741.92 (0.02)	4787.93 (0.02)	975.87 (444.17)
τ	5970.26 (0.02)	7744.35 (0.04)	10000 (4551.5)

4.1.2 Spatial Occupancy model

Inference and predictions of species occupancy is an essential part of ecological and conservation studies. To make such inferences and predictions about species occupancy, species distribution models are fitted to biodiversity data with species presence and absence information whilst accounting for imperfect detection. These fitted species distribution models can include random effects that capture the spatial autocorrelation in the data. This spatial autocorrelation can be caused by either accidentally omitting spatial covariates in the model or because the data contains biotic processes such as dispersal and conspecific attraction (Kéry and Royle, 2020).

In ecological problems, these models are usually fitted with MCMC; however they can be very data-rich and computationally expensive to fit with MCMC (Kéry and Royle, 2020). Specialized R-packages for particular classes of models like **spBayes** (Finley et al., 2007) and **R-INLA** as well as custom written MCMC engines like NIMBLE, Stan (Carpenter et al., 2017) can be used for such complex problems (Kéry and Royle, 2015). The disadvantage of using **spBayes** and **R-INLA** is that it is not possible to incorporate imperfect detection and false positives in the species distribution models (Kéry and Royle, 2020). Here, we use the INLA-MCMC method to fit spatial occupancy model that accounts for imperfect detection to a simulated dataset. We do this by sampling the observation model parameters and the true species occupancy state from MCMC and the ecological process model with the R-package **inlabru** (Bachl et al., 2019), a wrapper around **R-INLA**.

We simulate data for a static occupancy model with residual spatial autocorrelation in occupancy probability using the function *simOCCSpatial* in the R-package **AHMbook** (Kéry et al., 2022). The function generates occupancy data with a negative exponential autocorrelation function for the Gaussian random field plus a linear and quadratic effect of elevation on the occupancy probability and negative effects of forest cover and wind-speed on detection probability. See Chapter 9.4 of Kéry and Royle (2020) for details of the simulation function. The data was simulated for 50 sites and 10 survey visits.

The model we aim to fit and the prior distributions assigned to the hyperparameters are as follows:

$$\begin{aligned}
logit(\psi_i) &= \beta_0 + \beta_1 elev_i + \beta_2 elev_i^2 + \eta_i, \quad \text{where } \eta_i \sim N(\mathbf{0}, \Sigma) \\
\zeta_i &\sim Bernoulli(\psi_i) \\
logit(p_{ij}) &= \alpha_0 + \alpha_1 forestCover_i + \alpha_2 wind_{ij} \\
y_{ij} &\sim Bernoulli(\zeta_i \times p_{ij})
\end{aligned} \tag{14}$$

Prior distributions:

$$\begin{aligned}
\alpha_0, \alpha_1, \alpha_2 &\sim N(0, 0.0001) \\
\beta_0, \beta_1, \beta_2 &\sim N(0, 0.0001)
\end{aligned}$$

where Σ is the variance covariance matrix of the site effect, defined in terms of a constant variance parameter σ^2 and an exponential decay parameter θ, ζ_i is the latent occupancy state at site i, ψ_i is the occupancy probability at site i and p_{ij} is the detection probability at site i during survey visit j.

The parameters and latent state ζ in the model $\mathbf{z} = \{\alpha_0, \alpha_1, \alpha_2, \theta, \sigma^2, \zeta_1, \dots, \zeta_{50}\}$ into two mutually exclusive sets: $\mathbf{z}_c = \{\alpha_0, \alpha_1, \alpha_2, \zeta_1, \dots, \zeta_{50}\}$ and $\mathbf{z}_{-c} = \{\theta, \sigma^2, \beta_1, \beta_2, \beta_3\}$. New samples of $\alpha_0, \alpha_1, \alpha_2$ are proposed from a multivariate normal distribution with zero mean and covariance matrix $\Sigma = I$, where I is an identity matrix, and the latent state parameter $\zeta_1, \dots, \zeta_{50}$ are sampled using **nimble**'s default binary sampling algorithm. Posterior samples of $\mathbf{z}_{-c}$ are obtained from fitting a spatial regression model with ζ as the response variable.

Table 2: Summary of spatial occupancy model parameters (with standard error in paranthesis). The parameter ψ_{fs} is the realised occupancy calculated as the average occupancy over all the study sites.

Parameter	Truth	iNim-RW	MCMC
α_1	-1	-0.7701 (0.2005)	-0.8026 (0.1775)
α_2	1	0.8842 (0.1845)	0.9704 (0.1784)
α_0	-0.45	-0.6537 (0.2079)	-0.4912 (0.1511)
β_1	2	0.0954 (0.9975)	1.3966 (2.9925)

(continued)

Parameter	Truth	iNim-RW	MCMC
β_2	-2	0.8111 (1.4912)	-0.6083 (2.181)
β_0	2.197	1.7398 (0.865)	0.655 (1.5259)
ψ_{fs}	0.4664	0.52 (0.1381)	0.52 (0.0245)

The true parameter values and summaries of their posterior distribution estimated from iNim-RW, iNim2-RW and MCMC are presented in Table 2. At the time of writing this manuscript, the MCMC chains had been run for 1000 iterations. The MCMC results are less biased than the INLA-MCMC results. Longer chains may produce comparable results.

4.2 Application to real datasets

4.2.1 Bayesian Lasso

The third example is taken from Gómez-Rubio and Rue (2018). The lasso performs variable selection and model fitting at the same time by providing estimates that are zero (Hastie et al., 2009). Lasso tries to estimate coefficients of a model with a Gaussian likelihood by minimizing:

$$\sum_{i=1}^{N}\left(y_i - \alpha - \sum_{j=1}^{n_\beta}\beta_j x_{ji}\right)^2 + \lambda\sum_{j=1}^{n_\beta}|\beta_j|, \tag{15}$$

where y_i is the response variable, x_{ji} are covariates with covariate effect β_j and n_β is the number of covariates. The shrinkage of the coefficients is controlled by the parameter λ, with higher values of λ shrinking the parameters to zero.

The Lasso regression can be used for Bayesian inference by fitting a standard regression model with Laplace priors on the model coefficients. The Laplace distribution is defined as:

$$f(\beta) = \frac{1}{2\sigma}exp\left(-\frac{|\beta-\mu|}{\sigma}\right) \tag{16}$$

where the location and scale parameter is μ and σ respectively. The scale parameter is related to the shrinkage parameter as $\sigma = 1/\tau$. Currently **R-INLA** does not have a Laplace prior distribution implemented as part of its latent field. Hence, we use **R-INLA** to fit a Lasso regression model conditioned on the parameters β.

The Bayesian lasso model is fitted to the *Hitters* dataset described in Gareth et al. (2013). The dataset contains statistics about players in the Major league baseball. We aim to predict the salary of players based on five covariates: number of times at bat, number of hits, number of home runs, number of runs and number of runs batted in. Player i's salary (y_i) is assumed to be Gaussian distributed with mean $\beta_0 + \sum_{i=1}^{p}\beta_p x_{ij}$ and precision τ, where β_p are covariate effects and β_0 is the intercept of the model.

We assumed a multivariate Gaussian with mean zero and covariance matrix $0.25(\mathbf{X}^T\mathbf{X})^{-1}$ as the proposal distribution on the covariate coefficients $= \{\beta_1, \beta_2, \dots, \beta_p\}$, where $\mathbf{X}$ is a matrix with covariates as its columns. Gómez-Rubio and Rue (2018) and Berild et al. (2022) noted this proposal distribution yields good acceptance rates for the random-walk samplers. Moreover, the prior on τ is assumed to be Gamma distributed with parameters 1 and $5e^{-5}$, and intercept $\beta_0 \propto 1$.

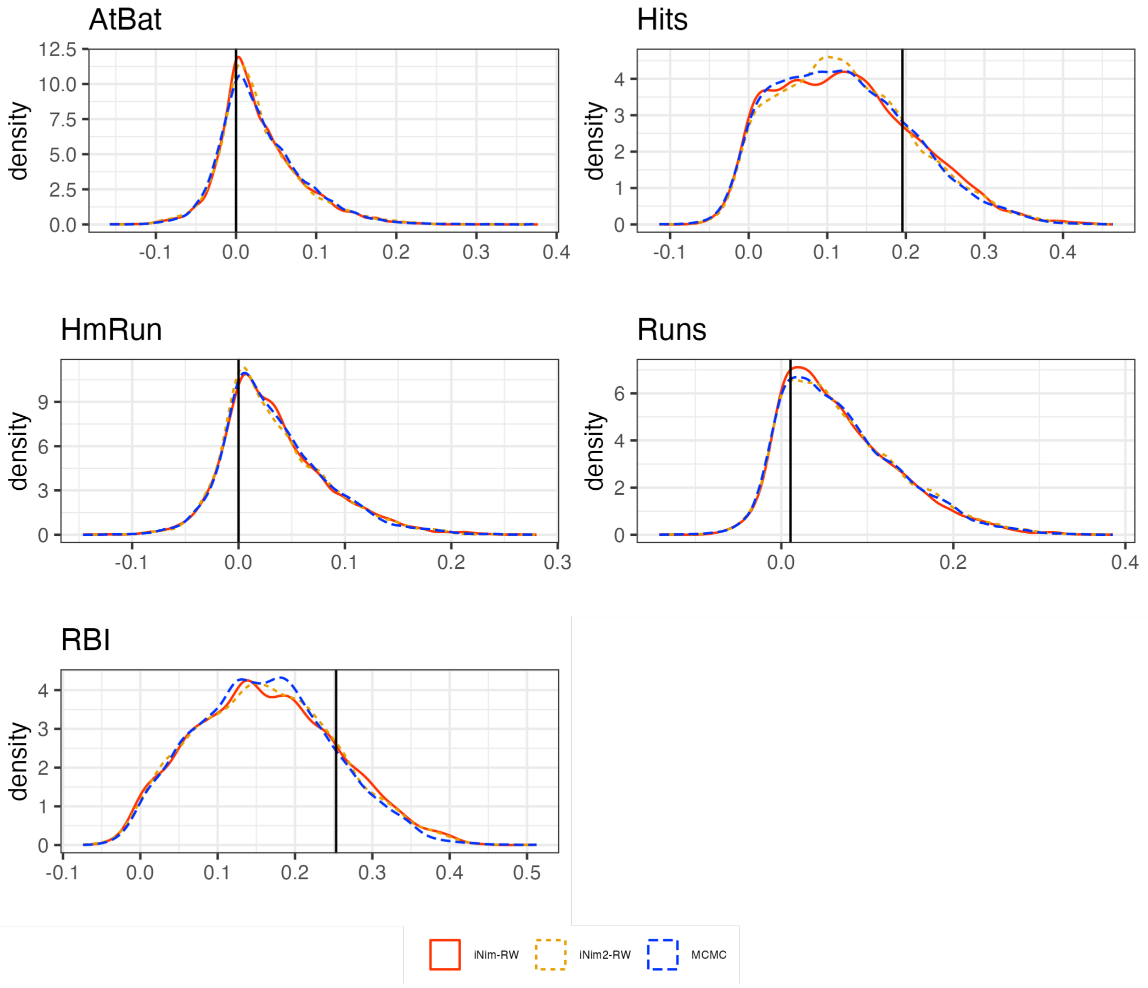

Figure 5: Posterior marginals of Lasso regression model parameters obtained from MCMC and the INLA-MCMC approaches. The solid vertical line indicates the maximum likelihood estimate of the model parameters estimated with the R-package **glmnet**.

Table 3: Summary estimates of Lasso estimated with the R-package **glmnet** and posterior mean from Bayesian Lasso (with standard deviation in paranthesis) using the MCMC and INLA-MCMC approaches.

Coefficient	iNim-RW	iNim2-RW	MCMC	Lasso
β_1	0.03 (0.05)	0.03 (0.05)	0.03 (0.03)	0.00
β_2	0.12 (0.09)	0.12 (0.08)	0.12 (0.12)	0.20
β_3	0.03 (0.05)	0.03 (0.05)	0.03 (0.03)	0.00
β_4	0.07 (0.07)	0.07 (0.07)	0.07 (0.07)	0.01

(continued)

Coefficient	iNim-RW	iNim2-RW	MCMC	Lasso
β_5	0.16 (0.09)	0.16 (0.09)	0.16 (0.16)	0.25

The posterior marginals of Lasso regression parameters are presented in Figure 5 and summary statistics of the posterior marginals are presented in Table 3. The posterior marginals of β estimated from MCMC are comparable to those from the INLA-MCMC methods. For the parameters with Lasso coefficients of 0, the posterior distributions were centered around 0.

4.2.2 Imputation of missing covariates

There are different multiple imputation methods that are available for missing covariates, with several of them implemented in the R-package **mice** (Van Buuren and Groothuis-Oudshoorn, 2011). This example uses the *nhanes* dataset in the **mice** package which contains information on age (`age`), body mass index (`bmi`), hypertension status (`hyp`) and cholesterol level (`chl`) of individuals from a study. The values of age are completely observed, but there are missing values in the body mass index and cholesterol level. Out of the nine missing values in the body mass index, six of them have missing cholesterol level values. The aim of this illustration, similar to that of Gómez-Rubio and Rue (2018), is to impute the missing values in body mass index and predict the cholesterol level through age and body mass index by fitting a multiple linear regression model defined in equation (17).

Until recently, **R-INLA** could not handle missing variables in covariates (Berild et al., 2022; Gómez-Rubio and Rue, 2018). Skarstein et al. (2023) has shown how to fit models with missing covariates in **R-INLA**. We proceed with this example assuming that the multivariate regression model cannot be fitted with **R-INLA** unless the missing covariates are imputed.

The imputation method employed in this example uses a Gaussian prior for the missing values of body mass index. The prior distribution is centered around the mean of the observed values (26.56) and its variance four times that of the observed values (71.07), similar to what Gómez-Rubio and Rue (2018) did.

The model we aim to fit is as follows:

$$
\begin{aligned}
chl_i &= \beta_0 + \beta_1 \text{bmi}_i + \beta_2 \text{age2}_i + \beta_3 \text{age3}_i \\
\beta_0 &\propto 1 \\
\beta_k &\propto N(0, 0.0001); \quad k = 1, 2, 3 \\
\epsilon &\sim N(0, \tau) \\
\tau &\sim Ga(1, 0.00005)
\end{aligned} \tag{17}
$$

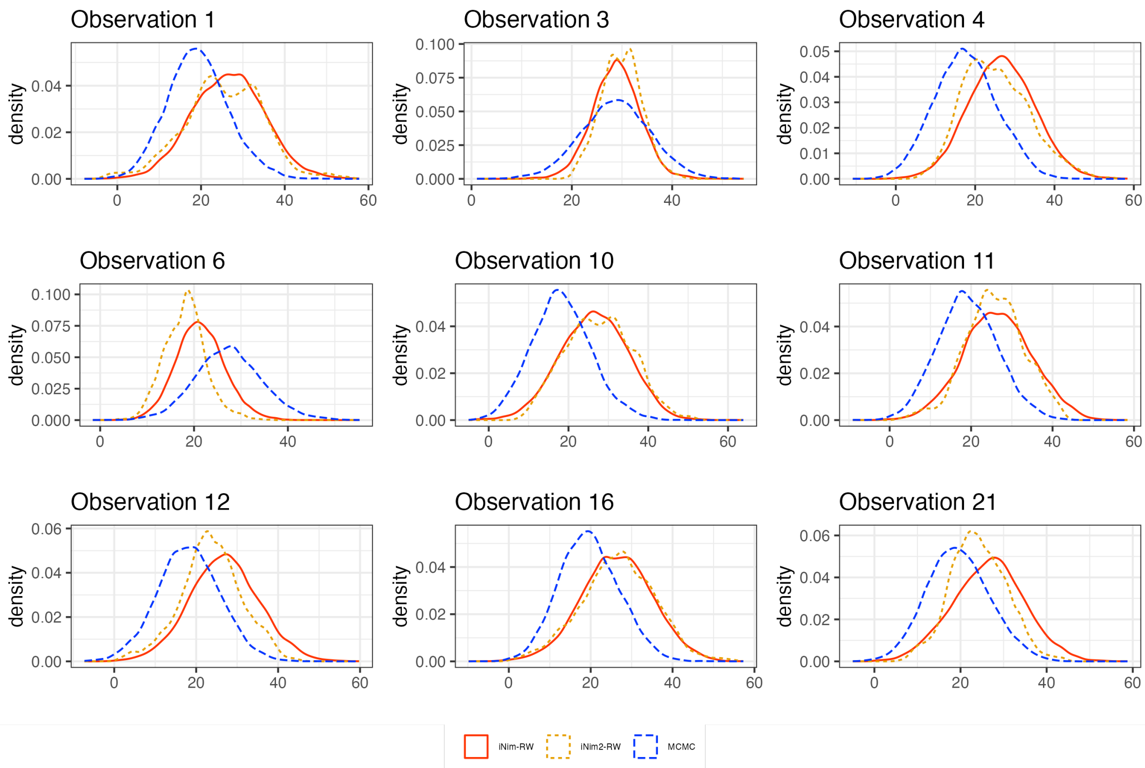

Figure 6: Posterior marginals of the imputed values of the body mass index.

Table 4: Summary of estimates of the parameters from the model with missing covariates (with standard error in paranthesis).

Parameter	iNim-RW	iNim-RW2	MCMC
α	9.02 (6.1)	3.78 (23.9)	-0.21 (-0.21)
β_1	34.13 (3.23)	42.97 (11.13)	14.16 (14.16)
β_2	57.1 (9.91)	77.14 (15.3)	21.98 (21.98)
β_3	6.03 (0.25)	6.02 (0.85)	5.58 (5.58)
τ	0 (0)	0 (0)	0 (0)

Figure 6 shows the posterior marginals of the imputed missing body mass index values and Table 4 shows the summary of posterior distribution of the model parameters. The posterior marginals obtained from the two INLA-MCMC approaches are similar, but both are different from those estimated using MCMC. The differences in the posterior marginals of the MCMC and INLA-MCMC approaches could be due to the differences in the posterior predictive distributions for missing values in the cholesterol from NIMBLe and INLA. This consequently affects how the estimates of the model parameters, which are different for all the three approaches (Table 4).

4.2.3 Zero - inflated Poisson

We also used the proposed INLA-MCMC approaches to model count data with excess zeroes using the zero-inflated Poisson regression model. The zero-inflated Poisson model assigns a probability p that a count is not zero, with this probability varying by a covariate.

Similar to the zero-inflated model in Berild et al. (2022), we aim to predict the number of fishes caught by 250 groups of people that went to a park based on the number of children (*child*) and whether or not the group brought a camper into the park (*camper*) for each group. The probability of getting counts greater than zero was modeled as a logistic regression on the number of people (*people*). The data used was accessed from the supplementary information in Berild et al. (2022).

The model we aim to fit and the prior distribution of model parameters are defined:

$$
\begin{aligned}
y_i|z_i &\sim ZIP(p_i, \mu_i), \quad i = 1, 2, ..., 250 \\
logit(p_i) &= \gamma_0 + \gamma_1 \text{people}_i \\
log(\mu_i) &= \beta_0 + \beta_1 \text{child}_i + \beta_2 \text{camper}_i \\
&\text{Prior distributions:} \\
\gamma_0, \gamma_1 &\sim N(0, 0.0001) \\
\beta_0, \beta_1, \beta_2 &\sim N(0, 0.0001)
\end{aligned}
\tag{18}
$$

where γ_0 and γ_1 are the intercept and covariate effect of *people* on the detection probability p, and γ_0, γ_1 and γ_2 are the intercept, covariate effect of number of children and whether the group brought a camper on the mean counts of fishes. We chose $\mathbf{z}_c = (\gamma_0, \gamma_1)$ and $\mathbf{z}_{-c} = (\beta_0, \beta_1, \beta_2)$. New values of γ_0 and γ_1 are proposed from normal distribution centered around the mean estimates and the standard deviation equal to three times the standard error estimates using the maximum likelihood approach. The maximum likelihood estimates were obtained from *zeroinfl()* from the R-package **pscl** (Jackman, 2020).

Table 5: Summary of estimates of the parameters from the zero-inflated Poisson model (with standard error in paranthesis). The maximum likelihood estimates (MLE) were obtained from fitting the model with the **pscl** package.

Parameter	MLE	iNim-RW	iNim2-RW	MCMC
β_0	1.6 (0.09)	0.76 (0.02)	0.77 (0.09)	0.21 (0.64)
β_1	-1.04 (0.1)	-1.03 (0.04)	-1.07 (0.07)	-1.05 (0.09)
β_2	0.83 (0.09)	0.83 (0.01)	0.83 (0.05)	1.12 (0.34)
γ_0	1.3 (0.37)	1.11 (0.32)	1.48 (0.25)	1.29 (0.41)
γ_1	-0.56 (0.16)	-0.49 (0.13)	-0.63 (0.11)	-0.6 (0.2)

The summary of the model parameters posterior distribution are presented in Table 5. The estimates from the INLA-MCMC approaches are similar to those estimated from MCMC and maximum likelihood estimates.

4.2.4 Binomial N-Mixture model with covariates in observation process

Modelling species abundance in space and time is a growing field in applied ecology. Binomial N-Mixture model, a hierarchical model that models latent abundance with a Poisson distribution and the observation process given the latent abundance as a Binomial distribution, has been widely used for such problems (Kéry and Royle (2017, 2020) and references therein). **R-INLA** can be used fit the Binomial N-mixture model, except for cases where there are observation level covariates for the detection process (Kéry and Royle, 2020; Meehan et al., 2017). In such instances, the observation level covariates are aggregated and used in modelling the detection process (Meehan et al., 2017).

We fit a Binomial N-Mixture model to mallard dataset that is publicly available through the R-package **unmarked** (Fiske and Chandler, 2011). The dataset contains counts of mallard ducks (*mallard.y*) collected from 239 sites in Switzerland during two to three sampling occasions in 2002. The dataset also contains abundance covariates called *mallard.site* which contains information on elevation (*elev*), forest cover (*forest*), length of transect (*length*); and detection covariates called *mallard.obs* which contains information on survey date (*data*) and survey intensity (*ivel*).

We are interested in predicting the total absolute abundance of mallard ducks whilst accounting for imperfect detection. The model we aim to fit and the prior distributions of the model parameters are as follows:

$$
\begin{aligned}
y_{ij} &\sim Binomial(N_i, p_{ij}) \\
N_i &\sim Poisson(\lambda_i) \\
log(\lambda_i) &= \beta_0 + \beta_1 elev_i + \beta_2 length_i + \beta_3 forest_i \\
logit(p_{ij}) &= \alpha_0 + \alpha_1 * ivel_{ij} + \alpha_2 * data_{ij} + \alpha_3 * data_{ij}^2
\end{aligned} \tag{19}
$$

Prior distributions:

$$
\begin{aligned}
\alpha_0, \alpha_1, \alpha_2, \alpha_3 &\sim N(0, 0.001) \\
\beta_0, \beta_1, \beta_2, \beta_3 &\sim N(0, 0.001)
\end{aligned}
$$

where β_0 is the intercept, β_1 is the effect of elevation, β_2 is the effect of length of transect and β_3 is the effect of forest cover on the duck abundance; α_0 is the intercept, α_1 is the effect of survey intensity, α_2 and α_3 are the effect of linear and quadratic effect of survey date respectively on the detection probability.

Here, we split the latent states into two sets: $\mathbf{z}_c = (N_1, N_2, \dots, N_{239}, \beta_0, \beta_1, \beta_2, \beta_3)$ and $\mathbf{z}_{-c} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$. We sample $\mathbf{z}_c$ from MCMC by using the default slice sampler for $N_1, N_2, \dots, N_{239}$ and the random walk block sampler for $\beta_0, \beta_1, \beta_2, \beta_3$; and $\mathbf{z}_{-c}$ is sampled from its posterior distribution from the conditional logistic regression model fitted with **R-INLA**. The N-Mixture model was also fitted with the R-package **unmarked** to obtain maximum likelihood estimates for the parameters; and with MCMC using **nimble** to obtain posterior distributions for the model parameters.

Table 6: Summary of estimates of the parameters from the Binomial N-Mixture model (with standard error in paranthesis). The maximum likelihood estimates (MLE) were obtained from the R-package **unmarked**.

Parameter	MLE	iNim-RW	MCMC
α_0	0.26 (0.22)	0.42 (0.06)	0.24 (0.22)
α_1	0.3 (0.18)	0.23 (0.06)	0.3 (0.17)
α_2	-0.37 (0.15)	-0.36 (0.02)	-0.33 (0.15)
α_3	0.01 (0.09)	0.07 (0.01)	0.04 (0.09)
β_0	-1.99 (0.24)	-1.93 (0.29)	-1.91 (0.28)
β_1	-0.41 (0.13)	-0.5 (0.14)	-0.49 (0.14)
β_2	-1.51 (0.25)	-1.36 (0.32)	-1.38 (0.31)
β_3	-0.71 (0.16)	-0.75 (0.18)	-0.76 (0.18)
Ntotal	86 (NA)	80 (2.58)	83 (4.86)

The summary of the posterior distribution of model parameters are presented in Table 6 and the posterior marginals of the model parameters presented in Figure 7. The results are comparable across all the approaches.

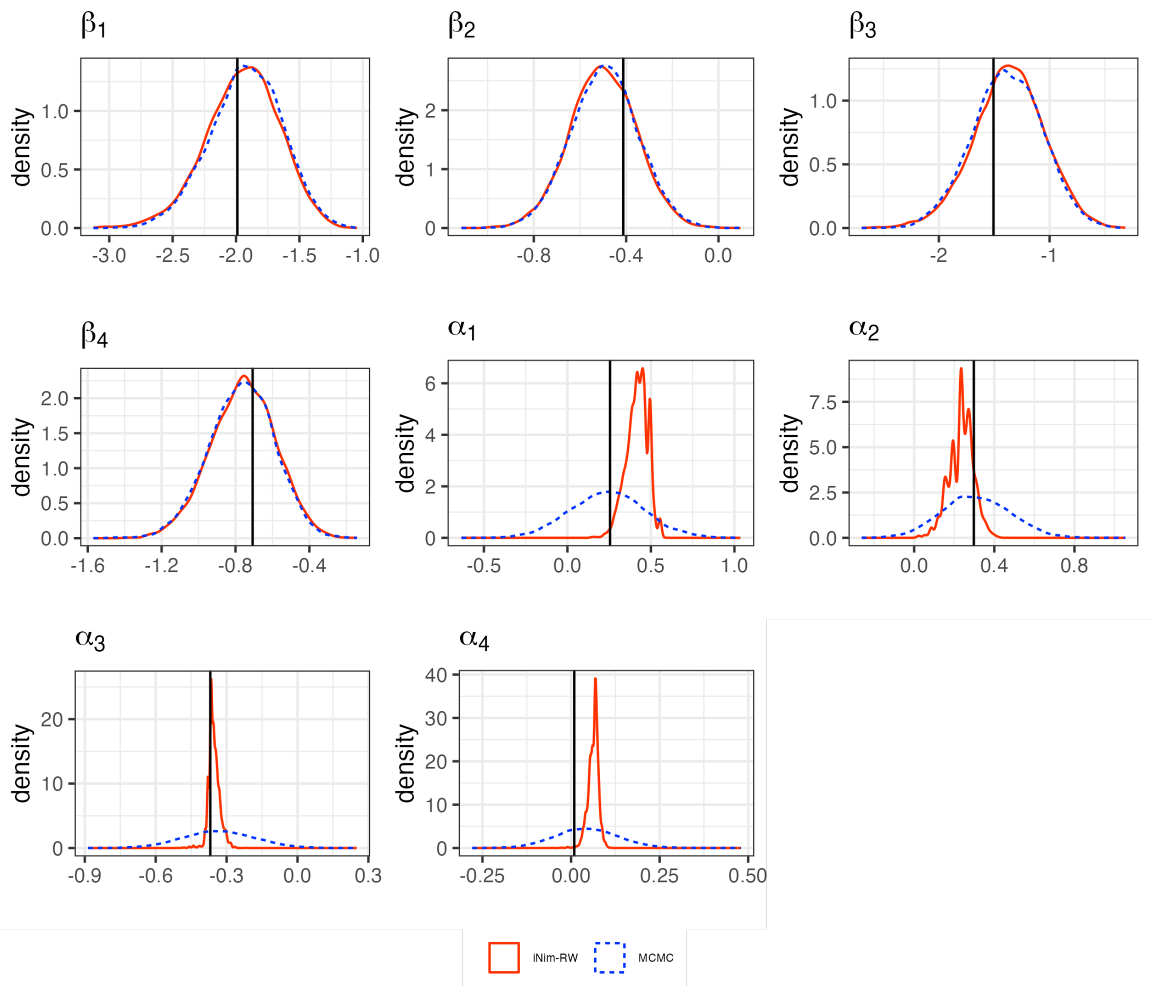

Figure 7: Posterior marginals of the binomial N-Mixture model parameters. The solid line indicates the maximum likelihood estimate of the parameter using the **unmarked** package.

5 Discussion

Using INLA-MCMC methodology for Bayesian inference is becoming an integral part of applied statistics. By using this methodology, the class of models fitted with **R-INLA** can be extended and the computational time of fitting models with MCMC can also be reduced. This methodology splits the model parameters into two mutually exclusive sets: one set that will be sampled from MCMC and the other set that will be sampled from the fitted conditional models using **R-INLA**.

Our study provided advancement in this methodology by showing how it can be implemented with NIMBLE. Two alternative approaches of this implementation are described in this study. The first approach is to use **R-INLA** defined functions to write customized samplers and the second approach is to use the **R-INLA** defined functions to write a nimble distribution model to be used in NIMBLE's BUGS framework. Our findings revealed that the marginal distribution of model parameters from both approaches are comparable to those from MCMC, INLA and maximum likelihood estimates.

Using NIMBLE for the INLA-MCMC methodology comes with its advantages. Firstly, NIMBLE is very efficient in running MCMC as compared to other competing software like JAGS and WinBUGS. NIMBLE also provides the platform to integrate R-defined functions into the BUGS code, compile them with C++ and efficiently generate samples using the numerous sampling algorithms implemented in **nimble**.

Although implementing the INLA-MCMC methodology with NIMBLE has some advantages, the longer computational times are inherited from the study in Gómez-Rubio and Rue (2018). As noted in our introduction, fitting models with **R-INLA** is very fast. However, integrating it with **nimble**'s sampling algorithm - that generates samples sequentially and makes calls to **R-INLA** for each iteration - can substantially increase the computational time of the INLA-MCMC methodology. This observation has already been noted in previous studies by Gómez-Rubio and Rue (2018) and Berild et al. (2022).

The number of calls made to **R-INLA** depends on the type of sampling algorithm. For instance, if INLA-MCMC methodology is run for M number of iterations, the random walk block sampling algorithm calls **R-INLA** M times during the running of the MCMC step (this ignores the number of calls made to **R-INLA** during the model building and compilation steps in **nimble**). If we choose the adaptive factor slice sampler (Tibbits et al., 2014) - which we also tried to implement in this study - then NIMBLE will call the **R-INLA** function for each of the univariate steps the sampling algorithm makes along the eigen vector to update the posterior distribution. If n_k steps are made at each of a multivariate k-dimensional model parameter space during iteration $i \in \{1, 2, \dots, M\}$, then the **R-INLA** call will be made a total of $n_k \times k \times M$ times.

Reducing the computational time can be achieved by using (adaptive) importance sampling for MCMC, as done in Berild et al. (2022). The concept of importance sampling has been implemented in the R-package **nimbleSMC**, and further studies can explore integrating **R-INLA** defined functions to fit models using the importance sampling with INLA methodology. This reduction in computational time is achieved due to the easy parallelization of the importance sampling algorithm, which will simultaneously call **R-INLA** functions multiple times at any given iteration. Moreover, the INLA framework can be integrated into the NIMBLE platform, since the Laplace approximation has been implemented in NIMBLE (de Valpine et al., 2023). This would be essential because the models that will be fitted can be compiled with C++, which will increase the efficiency of our sampling algorithm.

The usage of NIMBLE for the INLA-MCMC approach plays to a wide range of computational competency. Users who are new to the NIMBLE platform can use the second alternative described in section 3.2, where a **R-INLA** function is embedded in the BUGS code as a NIMBLE distribution function. The implemented samplers in **nimble** can be used to easily obtain posterior distribution of model parameters using the INLA-MCMC approach. However, there would be a limited control on the **R-INLA** function calls, since NIMBLE would call this function if the sampling of the model parameters depends on a random variable defined in the **R-INLA** function. To have much control of the number of calls, then the sampling algorithms have to be customized (as described in section 3.1). This approach comes with having expertise in writing codes in the NIMBLE platform.

This study processes the **R-INLA** output at each iteration and returns the marginal likelihood and samples from the posterior distribution of the parameters in the fitted conditional models. We do not save the output from **R-INLA** to perform any post-processing Bayesian averaging of the fitted models, as done in Berild et al. (2022) and Gómez-Rubio and Rue (2018). Further work can explore saving the **R-INLA** outputs as a NIMBLE list that can be processed after the INLA-MCMC methodology have been successfully run with **nimble**.

The proposed implementation presented in this study presents an opportunity to integrate various computational methods on the same platform. With the implementation of Laplace approximation in NIMBLE, the INLA-MCMC method can be completely implemented in NIMBLE, without external calls made to the **R-INLA**.

6 Code availability

All code and data used for this paper are on GitHub repository https://github.com/Peprah94/INLA_within_nimble.

7 Conflict of interest

The authors declare no conflict of interest.

8 Author contribution

KPA led the writing of the manuscript and the implementation of the methods. KPA and RBO were all involved in the idea conception.

References

Fabian E Bachl, Finn Lindgren, David L Borchers, and Janine B Illian. inlabru: an r package for bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, 10(6): 760–766, 2019.

Martin Outzen Berild, Sara Martino, Virgilio Gómez-Rubio, and Håvard Rue. Importance sampling with the integrated nested laplace approximation. *Journal of Computational and Graphical Statistics*, 31(4):1225–1237, 2022.

Roger S Bivand, Virgilio Gómez-Rubio, and Håvard Rue. Approximate bayesian inference for spatial econometrics models. *Spatial Statistics*, 9:146–165, 2014.

Marta Blangiardo and Michela Cameletti. *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons, 2015.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76, 2017.

Perry de Valpine, Christopher Paciorek, Daniel Turek, Nick Michaud, Cliff Anderson-Bergman, Fritz Obermeyer, Claudia Wehrhahn Cortes, Abel Rodrìguez, Duncan Temple Lang, and Sally Paganin. *NIMBLE: MCMC, Particle Filtering, and Programmable Hierarchical Modeling*.

Perry de Valpine, Daniel Turek, Christopher Paciorek, Cliff Anderson-Bergman, Duncan Temple Lang, and Ras Bodik. Programming with models: writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, 26:403–413, 2017. doi: 10.1080/10618600.2016.1172487.

Perry de Valpine, Christopher Paciorek, Daniel Turek, Nick Michaud, Cliff Anderson-Bergman, Fritz Obermeyer, Claudia Wehrhahn Cortes, Abel Rodrìguez, Duncan Temple Lang, and Sally Paganin. *NIMBLE User Manual*, 2023. URL https://r-nimble.org. R package manual version 1.0.1.

Andrew O. Finley, Sudipto Banerjee, and Bradley P. Carlin. spBayes: An R package for univariate and multivariate hierarchical point-referenced spatial models. *Journal of Statistical Software*, 19 (4):1–24, 2007. URL https://www.jstatsoft.org/article/view/v019i04.

Ian Fiske and Richard Chandler. unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43(10):1–23, 2011. URL https://www.jstatsoft.org/v43/i10/.

James M. Flegal, John Hughes, Dootika Vats, Ning Dai, Kushagra Gupta, and Uttiya Maji. *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA, and Kanpur, India, 2021. R package version 1.5-0.

Jerome Friedman, Robert Tibshirani, and Trevor Hastie. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01.

James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Spinger, 2013.

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice.* CRC press, 1995.

Virgilio Gómez-Rubio and Håvard Rue. Markov chain monte carlo with the integrated nested laplace approximation. *Statistics and Computing*, 28:1033–1051, 2018.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. An introduction to statistical learning. 2009.

Kurt Hornik, Friedrich Leisch, Achim Zeileis, and M Plummer. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of DSC*, volume 2, 2003.

Simon Jackman. *pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory.* United States Studies Centre, University of Sydney, Sydney, New South Wales, Australia, 2020. URL https://github.com/atahk/pscl/. R package version 1.5.5.1.

Marc Kéry and J Andrew Royle. Applied hierarchical modeling in ecology: Analysis of distribution, abundance and species richness in r and bugs. 2015.

Marc Kéry and J Andrew Royle. Applied hierarchical modeling in ecology: analysis of distribution, abundance and species richness in r and bugs. *(No Title)*, 2017.

Marc Kéry and J Andrew Royle. *Applied hierarchical modeling in ecology: Analysis of distribution, abundance and species richness in R and BUGS: Volume 2: Dynamic and advanced models.* Academic Press, 2020.

Marc Kéry, Andy Royle, and Mike Meredith. *AHMbook: Functions and Data for the Book 'Applied Hierarchical Modeling in Ecology' Vols 1 and 2*, 2022. URL https://CRAN.R-project.org/package=AHMbook. R package version 0.2.6.

Ye Li, Patrick Brown, Håvard Rue, Mustafa Al-Maini, and Paul Fortin. Spatial modelling of lupus incidence over 40 years with changes in census areas. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 61(1):99–115, 2012.

Jean-Michel Marin, Kerrie Mengersen, and Christian P Robert. Bayesian modelling and inference on mixtures of distributions. *Handbook of statistics*, 25:459–507, 2005.

Timothy D Meehan, Nicole L Michel, and Håvard Rue. Estimating animal abundance with n-mixture models using the r-inla package for r. *arXiv preprint arXiv:1705.01581*, 2017.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2023. URL https://www.R-project.org/.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

Emma Sofie Skarstein, Sara Martino, and Stefanie Muff. A joint bayesian framework for missing data and measurement error using integrated nested laplace approximations. *arXiv preprint arXiv:2303.15240*, 2023.

DJ Spiegelhalter, A Thomas, and NG Best. Winbugs version 1.4 user manual. mrc biostatistics unit, cambridge, united kingdom, 2014.

Matthew M Tibbits, Chris Groendyke, Murali Haran, and John C Liechty. Automated factor slice sampling. *Journal of Computational and Graphical Statistics*, 23(2):543–563, 2014.

Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.