Geophys. J. Int. (1999) **138,** 727–746

Geophysical inversion with a neighbourhood algorithm—II. Appraising the ensemble

Malcolm Sambridge

Research School of Earth Sciences, Institute of Advanced studies, Australian National University, Canberra, ACT 0200, *Australia.*
E-mail: malcolm@rses.anu.edu.au

Accepted 1999 April 12. Received 1999 April 8; in original form 1998 December 14

SUMMARY

Monte Carlo direct search methods, such as genetic algorithms, simulated annealing, etc., are often used to explore a finite-dimensional parameter space. They require the solving of the forward problem many times, that is, making predictions of observables from an earth model. The resulting ensemble of earth models represents all 'information' collected in the search process. Search techniques have been the subject of much study in geophysics; less attention is given to the appraisal of the ensemble. Often inferences are based on only a small subset of the ensemble, and sometimes a single member.

This paper presents a new approach to the appraisal problem. To our knowledge this is the first time the general case has been addressed, that is, how to infer information from a complete ensemble, previously generated by any search method. The essence of the new approach is to use the information in the available ensemble to guide a resampling of the parameter space. This requires no further solving of the forward problem, but from the new 'resampled' ensemble we are able to obtain measures of resolution and trade-off in the model parameters, or any combinations of them.

The new ensemble inference algorithm is illustrated on a highly non-linear waveform inversion problem. It is shown how the computation time and memory requirements scale with the dimension of the parameter space and size of the ensemble. The method is highly parallel, and may easily be distributed across several computers. Since little is assumed about the initial ensemble of earth models, the technique is applicable to a wide variety of situations. For example, it may be applied to perform 'error analysis' using the ensemble generated by a genetic algorithm, or any other direct search method.

Key words: numerical techniques, receiver functions, waveform inversion.

1 INTRODUCTION

Inversion techniques are often used in the Earth Sciences to provide constraints on Earth structure, or processes, from indirect observations at the surface. If the quantities of interest have been discretized into a finite (usually small) number of unknowns, and the relationship between observation and unknown is highly non-linear, then Monte Carlo direct search methods become useful (see Sen & Stoffa 1995 for a review). Usually, their role is to explore a multidimensional parameter space and collect models (i.e. sets of unknowns) which fit the observed data to some satisfactory level. Many examples exist in the literature. Popular methods have been uniform search (Keilis-Borok & Yanovskaya 1967; Press 1968; Wiggins 1969), simulated annealing (Rothman 1985, 1986) and genetic algorithms (Stoffa & Sen 1991; Sambridge & Drijkoningen 1992; Scales *et al.* 1992).

With these derivative-free search methods one is forced to solve the forward problem many times; that is, to calculate predictions based on an earth model, and compare them to the observations. This results in a large ensemble of models for which the fit to the data has been determined. The ensemble will often contain models with a wide range of data fits, and, one hopes, some at an acceptable level, given the noise in the data.

Once the ensemble has been collected, the next stage of the inverse problem is to draw inferences from the ensemble. Although much attention has been devoted to developing methods which efficiently search a parameter space, much less effort has been devoted to the problem of analysing the resulting ensemble in a quantitative manner. In many cases the appraisal of the results is based on only a subset of the ensemble, with the rest discarded. For example, this is the case if the objective is to locate a single optimal model (in some sense), and also with statistical methods such as importance sampling (e.g. Smith & Roberts 1993; Mosegaard & Tarantola 1995), which use only a statistically independent subset of the ensemble. In principle, however, the entire ensemble may provide useful information from which to draw inferences. In some cases, models which fit the data poorly may tell us just as much as those which fit the data well.

Several authors have proposed methods for analysing an ensemble of data-acceptable models, primarily using cluster analysis techniques (Kennett 1978; Kennett & Nolet 1978; Vasco *et al.* 1993; Lomax & Snieder 1995). However, use of the entire ensemble has largely been restricted to purely graphical methods (e.g. Nolte & Frazer 1994; Shibutani *et al.* 1996; Kennett 1998). Techniques have also been developed for characterizing an infinite set of acceptable models by determining the properties which they all share (Parker 1977; Constable *et al.* 1987). This approach involves seeking a solution with extremal properties, or testing a hypothesis on the observed data, and has mainly been applied to linear or weakly non-linear problems.

In this paper we study the appraisal stage of the inverse problem; that is, how to make quantitative inferences from the entire ensemble produced by a direct search method. We make no assumption as to how that ensemble may have been generated, only that it is available and that the forward problem has been solved for all models within it. Any randomized, or deterministic, algorithm may be used to generate the ensemble, for example genetic algorithms, simulated annealing, evolutionary programming or even guesswork! We present a new approach for extracting information from the ensemble, which requires no further solving of the forward problem, but allows measures of resolution and trade-off to be determined, within a Bayesian framework. The algorithm presented here is based on some simple geometrical concepts, which are also used in a related paper (Sambridge 1999) (hereafter referred to as Paper I) as the basis of a new direct search method.

In the next section we briefly outline the Bayesian approach, and define the measures of constraint, resolution and trade-off that are commonly used to appraise the information in the data. All of these take the form of integrals over the multidimensional model space, and may be evaluated using Monte Carlo (MC) integration techniques. This involves sampling the parameter space according to a prescribed distribution, and evaluating ensemble averages of various quantities. It is shown that, in general, the complete ensembles, generated by techniques commonly used for the search stage of the inverse problem, follow unknown distributions and, therefore, cannot be used directly for MC integration.

Section 3 contains the details of the new approach proposed here. It is shown how the input ensemble may be used to construct a multidimensional interpolant of the data fit measure, or more generally the posterior probability density function (PPD). This interpolant is based on Voronoi cells (nearest neighbour regions) and we use it to represent all information contained in the input ensemble. The key idea in the paper is to replace the real PPD with this approximate PPD, and then evaluate any Bayesian integral through MC integration. This requires a second ensemble to be generated with a distribution that follows the shape of the approximate PPD, but no further solving of the forward problem. All integrals then become simple averages over this 'resampled' ensemble and are trivial to evaluate.

The main computational task of the new algorithm is the generation of the resampled ensemble. It is shown how one may importance sample the approximate PPD using a standard statistical technique known as Gibbs sampler (Geman & Geman 1984). In order to apply the technique in this case several geometrical problems need to be solved concerning multidimensional Voronoi cells, and these are discussed in detail. The computational costs and memory requirements of the method are carefully analysed. It is shown that the resulting numerical algorithm lends itself easily to a parallel implementation.

In Section 4 the new resampling algorithm is illustrated with a numerical example. The problem is one of inversion of receiver functions for crustal seismic structure, which is known to be highly non-linear (Ammon *et al.* 1990). The resampling algorithm is used to calculate Bayesian measures of resolution and trade-off from two separate ensembles generated with different search methods. The results show that useful constraints and 'error' information can be obtained if the information is contained in the input ensemble. The technique presented here is only one route to addressing the appraisal problem, although at present we know of no comparable alternative.

2 ENSEMBLE INFERENCE AND BAYESIAN INTEGRALS

The starting point for our study is an ensemble of models ($\mathbf{p}_j$; $j = 1, \ldots, N_e$) with their corresponding fits to the data. For example, the ensemble may represent a collection of seismic velocity profiles with depth, and the data a set of surface wave dispersion measurements, as in Lomax & Snieder (1995). The ensemble is generated by the search stage of the non-linear inverse problem. The objective of the appraisal stage is to infer information (on the earth) from the finite irregularly distributed ensemble. We do not expect there to be a simple or unique solution. Two factors limit the information that can be obtained from the ensemble. The first is the degree of constraint provided by the observed data, and the second is the distribution of the given ensemble; that is, how well it samples the 'important' (good data fitting) regions of parameter space. Both of these are difficult to quantify.

In geophysical problems one often finds that the data/model relationship is non-linear, sometimes highly so, leading to multiple minima in the data misfit function. An example is the inversion of high-frequency body waveforms for seismic structure (e.g. Cary & Chapman 1988; Koren *et al.* 1991; Gouveia & Scales 1998). This can make it very difficult to identify the acceptable regions of parameter space and generate a 'good' ensemble. Also, the constraints provided by the data may result in there being none, one, or an infinite class of models which fit the data satisfactorily (even if the allowable earth models are restricted to a finite-dimensional parameter space). Therefore, we must accept that the ensemble will always be inadequate, and the information it contains limited, regardless of how sophisticated a search method may have been used to collect it.

2.1 Bayesian integrals

To address the appraisal problem we choose the framework of Bayesian inference. This has been presented many times in the geophysical literature and we do not attempt to repeat that material here. For summaries and tutorials within a geophysical context the reader is referred to Tarantola (1987), Duijndam (1988a,b), Cary & Chapman (1988), Mosegaard & Tarantola (1995) and Gouveia & Scales (1998). Useful books on posterior simulation are Gelman *et al.* (1995) and Tanner (1996), and summary papers are by Smith (1991) and Smith & Roberts (1993). From the Bayesian viewpoint, the solution to the inverse problem is the posterior probability density function (PPD). This quantity is used to represent all information available on the model. Its calculation depends upon the data, any prior information, and the statistics of all noise present, which must be assumed known. At any point, $\mathbf{m}$, in model space, $\mathcal{M}$, the PPD is given by

$$P(\mathbf{m}|\mathbf{d}_o)=k\rho(\mathbf{m})L(\mathbf{m}|\mathbf{d}_o)\,, \tag{1}$$

where $\rho(\mathbf{m})$ is the prior probability distribution which we shall refer to simply as 'the prior', $L(\mathbf{m}|\mathbf{d}_o)$ is a likelihood function which represents the fit to the observations, and k is a normalizing constant. (Note that the likelihood function and hence the PPD are conditional on the vector of observed data, $\mathbf{d}_o$; however, for notational convenience we will drop the $|\mathbf{d}_o$ terms from here on.) For Gaussian error statistics we have the familiar expression

$$L(\mathbf{m})=k\exp\left(-\frac{1}{2}\,(\mathbf{d}_o-\mathbf{g}(\mathbf{m}))^{\mathrm{T}}C_{\mathrm{D}}^{-1}(\mathbf{d}_o-\mathbf{g}(\mathbf{m}))\right), \tag{2}$$

where $\mathbf{g}(\mathbf{m})$ are the predictions from the model, and C_{D} is the data covariance matrix describing noise statistics. Since the PPD is a multidimensional function, it is usually characterized in terms of its properties in model space (often moments of the distribution). The model which maximizes the PPD is one property of interest, and in the absence of prior information would correspond to the best data fit model. The posterior mean model for the ith parameter, m_i, is given by the integral

$$\langle m_i\rangle=\int_{\mathcal{M}} m_iP(\mathbf{m})\,d\mathbf{m}\,. \tag{3}$$

Note that if the PPD were Gaussian then the mean would be equal to the maximum PPD model. Another quantity of particular interest is the posterior model covariance matrix,

$$C^{\mathrm{M}}_{i,j}=\int_{\mathcal{M}} m_im_jP(\mathbf{m})\,d\mathbf{m}-\langle m_i\rangle\langle m_j\rangle\,. \tag{4}$$

The diagonals of the posterior model covariance matrix are the posterior variances of the model parameters, and the off-diagonal terms contain information on trade-off between the model parameters. From this a resolution matrix can be determined using

$$R=I-C^{-1}_{\mathrm{M,prior}}C^{\mathrm{M}}\,, \tag{5}$$

where $C^{-1}_{\mathrm{M,prior}}$ is the inverse prior model covariance matrix determined from $\rho(\mathbf{m})$, and I is the identity matrix [see Tarantola (1987) for a proof of (5)]. The columns of R give a discrete approximation of the resolution kernel, which indicates how the real Earth is resolved by the model parameters. (Note that the posterior model covariance matrix and the resolution kernels are essentially linearized concepts. They are most useful if the PPD has a single dominant peak, and become less useful if multiple 'significant' maxima are present.) Another type of PPD property that may be useful, even when multiple maxima are present, is the marginal PPD. This is a function of one or more variables and is formed from an integral of the PPD over the remaining dimensions of the parameter space. For example, the marginal distribution of variable m_i is given by

$$M(m_i)=\int\ldots\int P(\mathbf{m})\prod_{\substack{k=1\\k\neq i}}^{d}dm_k\,. \tag{6}$$

Joint marginals between any pair of variables, $M(m_i, m_j)$, can be defined in a similar manner to the 1-D case. The marginals are a useful way of looking at the information provided on a single variable, or pair of variables, with all possible variations of other parameters taken into account.

2.1.1 *Monte Carlo integration*

In each case the integrals in eqs (3)–(6) take the form

$$J=\int_{\mathcal{M}} g(\mathbf{m})P(\mathbf{m})\,d\mathbf{m}\,, \tag{7}$$

where the function $g(\mathbf{m})$ is used to define each integrand. A numerical estimate can be obtained using multidimensional Monte Carlo integration over $\mathcal{M}$. We have

$$\hat{J}=\frac{1}{N}\sum_{k=1}^{N}\frac{g(\mathbf{m}_k)P(\mathbf{m}_k)}{h(\mathbf{m}_k)}\,, \tag{8}$$

where N is the number of discrete samples in the MC integration, $\mathbf{m}_k$ is the kth model sample and $h(\mathbf{m})$ is the density distribution of the samples. (We use a hat to denote an MC estimate of a variable.) The sampling density is assumed to be normalized, so we have

$$\int h(\mathbf{m})\,d\mathbf{m}=1\,. \tag{9}$$

Methods of multidimensional integration is an active area of statistical research (see Flournay & Tsutakawa 1989, Gelfand & Smith 1990 and Smith 1991 for reviews). Eq. (8) is just a weighted average of the $g(\mathbf{m}_k)$ over the ensemble,

$$\hat{J}=\frac{1}{N}\sum_{k=1}^{N}g(\mathbf{m}_k)w_k\equiv\bar{g}\,, \tag{10}$$

where the weights are

$$w_k=\frac{P(\mathbf{m}_k)}{h(\mathbf{m}_k)}\,, \tag{11}$$

and are usually called importance ratios.

The error in the numerical integration for $\hat{J}$ depends on the variance of $w_kg(\mathbf{m}_k)$ over the ensemble and is given by

$$\epsilon_{\hat{J}}=\frac{1}{\sqrt{N}}\,[\overline{g^2}-\bar{g}^2]^{1/2}\,. \tag{12}$$

By evaluating this simultaneously with (8), the standard error in the MC integral can be monitored, and the integration stopped when the error has been reduced to an acceptable level.

The rate at which the standard error in the Monte Carlo estimate decreases depends heavily upon the choice of sampling density, $h(\mathbf{m})$. The integration will be most efficient when the ensemble 'importance samples' the integrand, that is, so that $h(\mathbf{m})$ is similar in shape to each integrand in eq. (7). Since the PPD is the common factor in all integrals, then ideally one would have $h(\mathbf{m}) \approx P(\mathbf{m})$, and evaluate all integrals from the resulting ensemble. However, the distribution of the input ensemble is determined by whichever technique was used in the search stage of the inverse problem, and hence we have no control over it. Therefore, it is worthwhile knowing what type of distributions are produced by search algorithms used in geophysics. In the next section we briefly consider the three most common types of method, uniform Monte Carlo sampling (UMC), genetic algorithms (GA) and simulated annealing (SA).

2.1.2 The sampling densities of common search methods

For uniform sampling, by definition, the distribution of the ensemble tends to a constant, as the number of samples becomes large. [However, different methods, e.g. pseudo- or quasi-random sampling, differ in how quickly they converge. See Press *et al.* (1992) for a comparison.] In principle it is quite straightforward to perform MC integration when the samples are uniform; however, it is well known that the error in $\hat{J}$ will decrease very slowly, especially when the dimension of the space is high and the integrand complex. In the case of a genetic algorithm, it is not known what type of distribution the samples follow. Indeed, since the details of a GA can vary significantly between applications, it seems likely that no single sampling density will exist.

In simulated annealing a statistical importance sampling method is used to generate samples which follow a 'rescaled' posterior probability density function. The rescaled function takes the form

$$h(\mathbf{m}) = \exp\left(-\frac{\phi(\mathbf{m})}{T}\right), \tag{13}$$

where T is the scaling parameter (called temperature), and $\phi(\mathbf{m})$ represents the negative logarithm of the PPD. (Note that for $T=1$, the sampling density, $h(\mathbf{m})$, becomes equivalent to the PPD.) For each fixed temperature the SA algorithm uses an importance sampling method such as the Metropolis–Hastings method (Metropolis *et al.* 1953; Hastings 1970) to generate samples whose distribution tends towards the target $h(\mathbf{m})$ in (13). As the algorithm proceeds the value of T is gradually decreased towards $T=1$, and so the final ensemble contains a subset which is distributed according to the PPD. We see then that by using an importance sampling algorithm on a gradually changing target distribution, SA effectively combines the search and appraisal stages of the inverse problem.

The ratio of the number of models for which the forward problem has been solved to that in the subset which is drawn from the PPD has been called the 'loss factor' by Sambridge (1998). Its value is determined by the number of temperature steps required for the algorithm to sample the true PPD efficiently. Usually the loss factor is quite large because for each fixed temperature it is only the statistically independent models which tend towards the target distribution $h(\mathbf{m})$, and again these are usually only a small subset of the total. For all intervening samples the target distribution $h(\mathbf{m})$ must also be evaluated, and hence the forward problem solved (see Mosegaard & Tarantola 1995 and Sambridge 1998 for a discussion). For every independent model in the final 'PPD subset ensemble' it is common for the forward problem to be solved between 100 and 1000 times (Mosegaard & Tarantola 1995). This type of loss factor occurs in all importance sampling methods, e.g. Vasco *et al.* (1993).

In summary, both GA and SA produce an ensemble which preferentially samples the model space where the PPD is high, but in neither case does the complete ensemble follow a known sampling distribution. Therefore, to make use of all sampling produced by these or any other method, one needs to deal with the general case, that is, to construct estimates of the Bayesian integrals (3)–(6), from an ensemble with unknown distribution. In the next section we propose a solution.

3 INFERENCE FROM AN IRREGULARLY DISTRIBUTED ENSEMBLE

The central idea in this paper is to construct an approximation of the PPD everywhere in model space directly from the input ensemble, and then use this approximate PPD for Monte Carlo evaluations of the Bayesian integrals (7). In the next section we describe our choice of constructing an approximate PPD. In the subsequent section we show how the Bayesian integrals may be evaluated by generating a second ensemble of integration points which importance sample the approximate PPD. Since we only apply importance sampling to the approximate PPD, this requires no solving of the forward problem and hence the computational inefficiency associated with the loss factor, referred to above, is avoided.

3.1 The neighbourhood approximation of the PPD

The reconstruction of the PPD from a finite ensemble of samples is effectively an interpolation problem in a multidimensional space. The interpolation of scattered data is a much studied problem in two and three dimensions (see Watson 1992). However, for higher dimensions most methods become computationally unwieldy. Here we construct a multidimensional interpolant using Voronoi cells (Voronoi 1908). These geometrical constructs have been used in many areas of the physical sciences (see Okabe *et al.* 1992 for a review) and more recently in geophysics (Sambridge *et al.* 1995; Sambridge & Gudmundsson 1998; Gudmundsson & Sambridge 1998). Voronoi cells are simply the nearest-neighbour regions about each point in model space, as defined by a particular distance norm. Fig. 1 shows an example in two dimensions using the L_2-norm.

Voronoi cells have some useful properties which make them ideal for the basis of our multidimensional interpolant. For any distribution of irregular points in any number of dimensions, they are unique, space-filling, convex polyhedra, whose size and shape are automatically adapted to the distribution of the point set. Note that the size (volume) of each cell is inversely proportional to the density of the points. Voronoi cells are also used in Paper I as the basis of a new direct search algorithm. In that case they are repeatedly updated as more models are generated. Here we use them to construct an approximate PPD from a fixed ensemble. This is done by simply setting the known PPD of each model to be constant inside its Voronoi cell. In this way the scale lengths of variation in the interpolated

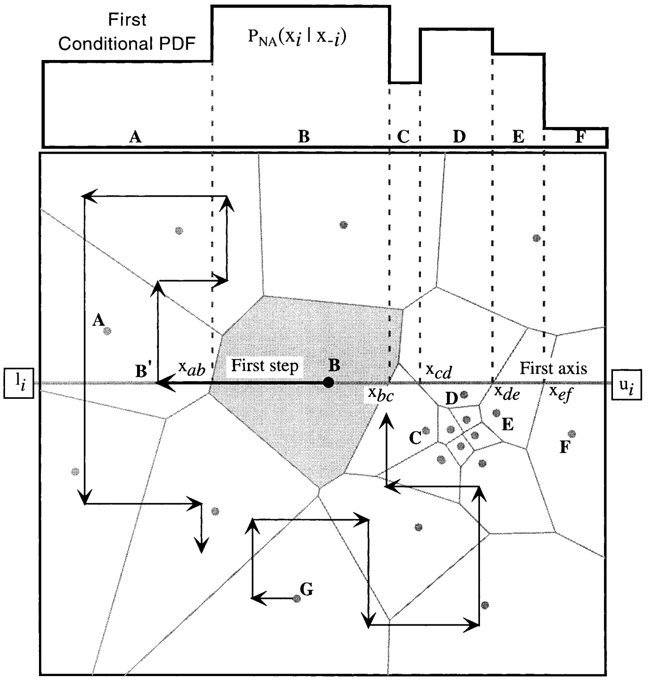

Figure 1. Two independent random walks through the neighbourhood approximation of the PPD. The Gibbs sampler is used. For the first step (x-direction) of the walk starting at cell B (shaded) the shape of the conditional, $P_{NA}(x_i|x_{-i})$, is shown above the figure. After many steps the density distribution of the random walk will asymptotically tend to the approximate PPD, $P_{NA}(\mathbf{x})$.

function are directly tied to the spacing of the samples in the ensemble. In effect, each Voronoi cell acts as a 'neighbourhood of influence' about the corresponding model in the ensemble. We call this the neighbourhood approximation to the PPD, and write it as $P_{NA}(\mathbf{m})$. Specifically, we have

$$P_{NA}(\mathbf{m}) = P(\mathbf{p}_i), \tag{14}$$

where $\mathbf{p}_i$ is the model in the input ensemble which is closest to the point $\mathbf{m}$.

It is interesting to note that the approximation $P_{NA}(\mathbf{m})$ is related to the bootstrap method (Efron 1982; Efron & Tibshirani 1986), used for determining measures of statistical accuracy. The philosophy behind the bootstrap is similar to that here; that is, to reconstruct a probability distribution from a finite set of realizations. In the bootstrap it is achieved as a sum of Dirac delta functions centred on the members of the ensemble. Therefore, resampling with the bootstrap always produces copies of the original samples. In contrast, the neighbourhood approximation, $P_{NA}(\mathbf{m})$, has a uniform probability inside each Voronoi cell. Therefore, with $P_{NA}(\mathbf{m})$ the influence of each model is spread uniformly across each cell, rather than concentrated at a point.

Voronoi cells are defined in terms of a distance norm in model space, which must be chosen *a priori*. For the L_2-norm the distance between points $\mathbf{x}_a$ and $\mathbf{x}_b$ in model space is defined as

$$\|\mathbf{x}_a - \mathbf{x}_b\| = ((\mathbf{x}_a - \mathbf{x}_b)^{\mathrm{T}} C_{\mathrm{M}}^{-1} (\mathbf{x}_a - \mathbf{x}_b))^{1/2}, \tag{15}$$

where C_{M}^{-1} is a matrix that removes the dimensionality of the variables. In effect, it controls the influence of different variables on the shape of Voronoi cells, and is particularly important when the variables have different physical dimension. An appropriate, although not necessary, choice is to use the prior model covariance matrix (see Menke 1989).

3.2 Monte Carlo integration of the neighbourhood approximation

In this work we use the neighbourhood approximation to represent all information contained in the input ensemble of models. Since this is the only information we have on the PPD, we use our approximate PPD in place of the real PPD in all Bayesian integrals, i.e. we have

$$P_{NA}(\mathbf{m}) \approx P(\mathbf{m}). \tag{16}$$

This approximation is at the heart of the algorithm presented in this paper and is discussed further below. The Bayesian integrals can then be evaluated by generating a new set of MC integration points in model space, $\mathbf{s}_k (k = 1, \ldots, N_{\mathrm{r}})$, whose distribution asymptotically tends towards $P_{NA}(\mathbf{m})$. We call this the 'resampled ensemble'. In other words, the new points are designed to importance sample the neighbourhood approximation to the PPD. Therefore, if the sampling is performed correctly the sampling density, $h_R(\mathbf{m})$, should satisfy

$$h_R(\mathbf{m}) \approx P_{NA}(\mathbf{m}). \tag{17}$$

Combining (16) and (17) in (11) gives

$$w_k \approx 1 \quad (k = 1, \ldots, N_{\mathrm{r}}), \tag{18}$$

so the Bayesian integrals (7) become

$$\hat{\mathcal{J}}_{NA} = \frac{1}{N_{\mathrm{r}}} \sum_{k=1}^{N_{\mathrm{r}}} g(\mathbf{s}_k), \tag{19}$$

where we use the subscript NA to indicate that the approximate PPD has been used. Therefore, only simple averages over the resampled ensemble have to be calculated. Note that by replacing the true PPD with $P_{NA}(\mathbf{m})$ in (7) we have only approximated part of each integrand. The spatial variability of the remaining term, $g(\mathbf{m})$, is fully taken into account in estimating the numerical integral because $g(\mathbf{m})$ is evaluated at the resampled points $\mathbf{s}_k (k = 1, \ldots, N_{\mathrm{r}})$ in (19). Note also that the approximate PPD no longer appears in eq. (19) directly, but instead controls the distribution of the resampled ensemble (17). Therefore, to proceed we need only to be able to generate the new ensemble and then evaluate simple ensemble averages using (19).

3.3 Importance sampling the neighbourhood approximation of the PPD

The resampled ensemble can be generated with a standard approach known as a Gibbs sampler (Geman & Geman 1984; Smith & Roberts 1993). With this method one can generate a random walk in model space, whose distribution asymptotically tends towards any given distribution (see Geman & Geman 1984, Gelman *et al.* 1995 and Tanner 1996 for proofs of convergence). Here we use it to generate samples distributed according to the approximate PPD, $P_{NA}(\mathbf{m})$. Fig. 1 illustrates the procedure in two dimensions. The random walk starts at point B, which we write as $\mathbf{m}_B$. (This can be a model from the input ensemble.) From this point it takes a series of steps along each parameter axis in turn. A step is performed by drawing a random deviate from the conditional probability density function of $P_{\mathrm{NA}}(\mathbf{m})$ along the ith axis. We write this as $P_{\mathrm{NA}}(x_i|x_{-i})$, where x_i is a position variable along the ith axis and x_{-i} denotes the fixed values of all other components of the vector $\mathbf{m}_B$. It is clear that the conditional $P_{\mathrm{NA}}(x_i|x_{-i})$ is just the function $P_{\mathrm{NA}}(\mathbf{m})$ sampled along the ith axis which passes through $\mathbf{m}_B$. Fig. 1 shows the conditional for the first step. Since $P_{\mathrm{NA}}(\mathbf{m})$ is constant inside each Voronoi cell, the conditional is built from the PPD values inside each Voronoi cell intersected by the axis. Note that it is possible for the random walk to move into any of these Voronoi cells, with probability determined by the product of the PPD value and the width of the intersection.

The Gibbs sampler continues by generating the next step along the $(i+1)$th axis through the new point, B′, and so on, cycling through each parameter axis in turn. At each step one element of $\mathbf{m}_B$ is updated. An iteration is completed when all dimensions have been cycled through once, and a complete new model space vector has been generated. It can be shown that after many iterations this random walk produces model space samples with a distribution that asymptotically tends towards the target distribution, i.e. $P_{\mathrm{NA}}(\mathbf{m})$ (see Gelman *et al.* 1995 for a proof). Note that overall the random walk is influenced by all Voronoi cells that are intersected by the axes, not just the cells that the walk passes through. Fig. 1 shows four iterations of a walk starting from point B, and four more from an independent walk starting from point G.

In this way the Gibbs sampler can be used to generate the resampled ensemble of any chosen size, N_{r}. (One would expect $N_{\mathrm{r}} \gg N_{\mathrm{e}}$, i.e. the size of the input ensemble.) Note that one does not have to collect the ensemble from just a single random walk. It is preferable to use multiple independent random walks, each starting from a different point in model space. This will significantly reduce computation time, because calculations can be performed simultaneously, and also improves the sampling of the parameter space, since each walk starts in a different place. A useful choice might be to select the starting points from the positions of the better data fitting models in the input ensemble.

After all random walks have been performed, the results can be combined and the ensemble averages in (19) evaluated. For N_{w} walks we have

$$\hat{J}_{\mathrm{NA}} = \frac{1}{N_{\mathrm{r}}} \sum_{j=1}^{N_{\mathrm{w}}} n_j \bar{g}_j \,, \tag{20}$$

where n_j is the number of samples generated in the jth walk, N_{r} is the total number of samples, given by

$$N_{\mathrm{r}} = \sum_{j=1}^{N_{\mathrm{w}}} n_j \,, \tag{21}$$

and $\bar{g}_j$ is the average (of the variable) from the jth walk, i.e.

$$\bar{g}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} g_{ij} \,, \tag{22}$$

where g_{ij} is the ith sample from the jth walk.

In practice these expressions will be trivial to evaluate. In the next section we describe how to determine the 1-D conditional, $P_{\mathrm{NA}}(x_i|x_{-i})$, for each axis, and then show how to calculate the steps of the random walk.

3.3.1 Calculating the conditional

Fig. 1 shows that the conditional, $P_{\mathrm{NA}}(x_i|x_{-i})$, is simply a set of step functions with abrupt changes at the points where the axis passes into a new Voronoi cell. These intersection points must be found for each new axis produced during the random walk. This ‘intersection’ problem is the main computational task involved in the resampling algorithm. A method for efficiently calculating the intersection points of a single multidimensional Voronoi cell and any 1-D axis is given in Paper I. We do not repeat the description here. Using this ‘single-cell method’ we obtain the intersection points of a given Voronoi cell with the axis, as well as the indices of the two neighbouring Voronoi cells; that is, starting from cell B in Fig. 1 the single-cell algorithm would give us the intersection points (x_{ab}, x_{bc}) and identify the neighbouring cells A and C.

To find the points where the remaining Voronoi cells intersect with the axis, one simply repeats the procedure in both directions until the boundaries of the parameter space are reached. In this way we find all Voronoi cells intersected by the current axis, between (l_i, u_i), and their intersection points. The computational cost of solving the intersection problem is crucial to the overall efficiency of the resampling algorithm. This aspect is discussed in detail below. It turns out that even in high dimensional spaces the number of Voronoi cells intersected by any axis is usually quite small, and so one only has to apply the single-cell method a few times. In the numerical example presented in Section 4, there are 10 000 Voronoi cells in a 24-D space, and these produced an average of less than four intersections per axis. [The reason that this number is small is because an L_2-norm (eq. 15) is used to define Voronoi cells. We recall that in order to move from one Voronoi cell to another, the point along the axis must be closer to a new model in the input ensemble (see eq. 14). However, only one variable in the n-D distance norm changes as we move along a coordinate axis and so this only has a relatively small effect on the overall distance. Even with a large number of points in the input ensemble, the nearest neighbour will only change occasionally as we move along the axis. Hence each axis will only pass through a few Voronoi cells.]

3.3.2 Generating a random step

Once the intersection points $(x_{ab}, \ldots, x_{ef})$ and cells (A, ..., F) are known, then the conditional is completely specified and standard techniques may be used to generate a random deviate from this 1-D probability distribution. Here we use a ‘rejection’ method (see Press *et al.* 1992 for full details). In this approach a proposed step, x_i^p, is generated as a uniform random deviate between the endpoints of the axis, that is, in the interval (l_i, u_i) in Fig. 1. This proposed step is accepted if a second random deviate, r, generated on the unit interval (0, 1), satisfies

$$r \le \frac{P_{\mathrm{NA}}(x_i^p|x_{-i})}{P_{\mathrm{NA}}(x_i^{\max}|x_{-i})} \,, \tag{23}$$

where $P_{\mathrm{NA}}(x_i^{\max}|x_{-i})$ is the maximum value of the conditional along the axis. If the proposed step is rejected then the whole procedure is repeated until an accepted step is produced. It is simple to show that the density distribution of the accepted steps is equal to the 1-D conditional PDF, as required (for example, see pp. 281–282 of Press *et al.* 1992).

A salient feature of the rejection method is that only the ratio of two PPD values appears in eq. (23). Therefore, the PPD need only be known up to a multiplicative constant, that is, its normalization can be ignored. Furthermore, by taking logs of both sides, the condition (23) becomes

$$\log(r) \le \log(P_{\mathrm{NA}}(x_i^p|x_{-i})) - \log(P_{\mathrm{NA}}(x_i^{\max}|x_{-i}))\,. \tag{24}$$

Therefore, to implement the rejection method one need only evaluate the difference between logs of the PPD, and never the actual PPD itself. In practice, this becomes important because in many problems the ratio of the PPD values can become infinitesimally small and cause numerical 'underflow' problems when implemented. (An example occurs in the numerical example below.) Since (24) uses only the log of the PPD no such problems arise. Note that the other common method for generating a random deviate from a 1-D conditional is the transformation method (see Press *et al.* 1992), which is simpler to implement than the rejection method, but requires explicit evaluation of the PPD, and hence will be prone to this type of numerical problem.

This completes the description of the resampling algorithm. Note that even though a random walk is generated through the multidimensional function $P_{\mathrm{NA}}(\mathbf{m})$, the only calculation involving Voronoi cells is to determine their intersections with a known 1-D axis. This problem must be solved many times over, but one does not have to determine any data structure defining the Voronoi cell itself, for example the vertices of each cell (see Fig. 1). It turns out that the number of vertices of each Voronoi cell grows extremely rapidly with the dimension of the space (Okabe *et al.* 1992; Sambridge 1998), so the problem would become computationally intractable if the full multidimensional Voronoi cells had to be determined. With the resampling algorithm we are able to take advantage of the properties of Voronoi cells in a multidimensional space, without having actually to calculate them.

3.4 Computational issues

The computation time and memory requirements of the resampling algorithm are the factors which determine its practicality for many applications. It is therefore important to know how these quantities scale with the size of the input ensemble, resampled ensemble, and dimension of the parameter space.

3.4.1 Memory requirements

The storage required by the algorithm is controlled by the input ensemble only; that is, the total memory, M, scales as

$$M \propto N_e d\,. \tag{25}$$

The point to note here is that the resampled ensemble need not be stored in memory, hence N_r does not appear in (25). This is because all calculations can be performed with only a 'single loop' over the resampled ensemble; that is, once each model of the resampled ensemble is determined it contributes to the appropriate ensemble averages and is discarded. (This is true of many MC integration techniques.) In the Appendix it is shown how the MC estimates of all Bayesian integrals in eqs (3)–(4) and their error estimates (12) can be arranged as 'single-loop calculations', which require only ensemble averages to be determined.

3.4.2 Computation time

Two factors influence the computational time of the algorithm. These are the generation of the resampled ensemble and the evaluation of the ensemble averages. The latter clearly depends linearly on N_r, and in practice is quite trivial. The time taken to generate the resampled ensemble follows

$$T \propto N_r N_e d\,. \tag{26}$$

The linear dependence on N_r and d is clearly optimal because we generate N_r samples, each of which is a d-dimensional vector. It is unknown whether the linear dependence on the size of the input ensemble, N_e, can be improved upon. This factor comes from solving the intersection problem along a single axis (described in Paper I). The intersection algorithm requires extra overhead calculations to be performed which are linearly dependent on d, but the solution for a single axis becomes independent of d. Therefore, the overall computation time in (26) remains linearly dependent on d.

It seems likely that, in practice, the most effective way of further reducing computational time, for any application, would be by exploiting the parallel nature of the resampling algorithm. Note that we effectively have perfect parallelization; that is, if the walks are distributed across n processors of the same CPU speed, the overall time taken will be reduced by a factor of n. This is because no communication is required between processors until the very end, when the ensemble averages are combined. Typical CPU times are given for a numerical example in the next section.

4 APPLICATION OF THE RESAMPLING ALGORITHM TO RECEIVER FUNCTION INVERSION

To illustrate the resampling technique we apply it to the inversion of receiver functions for crustal seismic structure. This is a highly non-linear waveform fitting problem (Ammon *et al.* 1990), which serves as an example of the difficulties present in many studies of seismogram inversion.

The crustal structure is parametrized using 24 variables. The S-velocity depth profile is constructed from six horizontal layers, with four parameters in each layer, representing the thickness of the layer (km), the S velocity at the topmost point in the layer (km s^{-1}), the S velocity at bottommost point in the layer (km s^{-1}) and the ratio of P to S velocity in the layer. The velocity profile is completed by imposing a linear gradient between the two velocities in each layer. Each parameter is identified by an index. See Table 1 for the list. This table also includes indices for 12 additional variables, which are combinations of the inversion parameters. The resampling algorithm can be used to evaluate Bayesian indicators involving these, or any other, transformed parameters in an identical manner to the original variables. The parametrization used here is the same as has previously been used for the inversion of receiver functions recorded in Eastern Australia (Shibutani *et al.* 1996). Table 1 contains the parameter space bounds for each parameter, and Paper I contains figures of typical velocity profiles produced with this parametrization.

Since the Bayesian indicators reflect the information in sampling of the ensemble as well as the data, we need to distinguish between the two in assessing the resampling algorithm. We therefore choose a synthetic data problem so

Table 1. Parameter space bound used in the receiver function inversion. Brackets show indices. All values above 24 are combinations of the model parameters. H denotes a layer thickness (km), V_{s1} the S velocity at the top of a layer (km s^{-1}), V_{s2} that at the bottom of a layer (km s^{-1}), V_p/V_s the velocity ratio in a layer, and Z the depth of the bottom of the layer (km). The Moho depth, Z_{moho}, is represented by parameters 26 and 36. The variable 25 is missing from the table and represents the velocity jump across the Moho, ΔS_{moho}.

Layer	H	V_{s1}	V_{s2}	V_p/V_s	$\partial V_s/\partial H$	Z
Sediment	0–2 (1)	1.75–3.0 (7)	1.75–3.0 (13)	2.0–3.0 (19)	−12.5–12.5 (27)	0–2 (32)
Basement	0–3 (2)	1.5–3.5 (8)	1.5–3.5 (14)	1.65–2.0 (20)	−20.0–20.0 (28)	0–5 (33)
Uppercrust	1–15 (3)	2.6–3.6 (9)	2.8–4.0 (15)	1.65–1.8 (21)	−0.8–1.4 (29)	1–20 (34)
Middlecrust	5–20 (4)	3.2–4.5 (10)	3.2–4.5 (16)	1.65–1.8 (22)	−0.26–0.26 (30)	6–40 (35)
Lowercrust	5–20 (5)	3.2–4.5 (11)	3.2–4.5 (17)	1.65–1.8 (23)	−0.26–0.26 (31)	11–60 (36)
Mantle	5–30 (6)	4.0–5.0 (12)	4.0–5.0 (18)	1.70–1.9 (24)		

that the results of Bayesian integrals (using either ensemble) can be compared to a known 'true earth' model. To illustrate the resampling algorithm we calculate a series of Bayesian integrals for two separate ensembles of earth models and compare results. Both were produced with direct search methods. The first was generated with the neighbourhood algorithm described in Paper I (we call this the 'NA ensemble') and the second with a genetic algorithm (we call this the 'GA ensemble'). Both ensembles contain approximately 10^4 crustal S-wave velocity profiles. Details of both algorithms and a comparison of the two ensembles can be found in Paper I. (See Figs 5b and d of Paper I for a plot of the GA ensemble and receiver function of the best-fitting model, and Figs 6a and c of Paper I for the NA ensemble.)

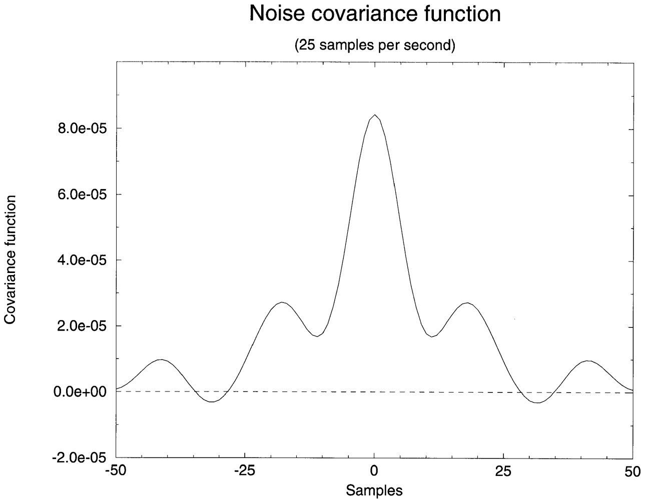

Figure 2. The diagonals of the data covariance matrix describing the statistics of the noise in the receiver functions. The central diagonal is plotted as sample 0. The first 50 upper and lower diagonals of the matrix are non-zero, corresponding to a covariance function of width ±2 s.

4.1 The data covariance and prior PDF

A prior PDF is required for the Bayesian method. Here we simply set it to be uniform within the parameter space boundaries. These are chosen to allow a wide class of potential earth models (see Table 1), and hence impose relatively weak prior information on all variables. In many situations one might have more complex prior information than is assumed here; however, this is sufficient to illustrate the algorithm.

To calculate the posterior we need to determine the inverse data covariance matrix, C_D^{-1}, describing the statistics of the noise added to the synthetic data (see Paper I). We treat the synthetic data as if it were observed data and calculate C_D using the method described by Gouveia & Scales (1998). This involves obtaining realizations of noise receiver functions, $\mathbf{r}_i$ ($i=1, \ldots, N_d$), and calculating their covariance matrix using

$$C_D = \frac{1}{N_d} \sum_{i=1}^{N_d} \mathbf{r}_i \mathbf{r}_i^T . \tag{27}$$

Since our noise is synthetically generated we have the luxury of choosing a large number of realizations. Here we set $N_d = 500$. The matrix C_D was calculated using eq. (27) and then smoothed by replacing each element with the average of its diagonal. (This removes some numerical artefacts and assumes stationarity of the noise.) A plot of the resulting noise covariance function (i.e. the cross-diagonal terms) is shown in Fig. 2. The sampling frequency is 25 Hz and the length of each trace is 30 s, giving 876 samples, and hence C_D is a matrix of size 876 × 876. The covariance function shows how the noise is temporally correlated in the receiver function. We choose to use only the first 50 diagonals either side of the main diagonal, since the amplitude of the covariance function falls away rapidly beyond ±2 s.

In theory, C_D should have full rank, but in this case we found that contamination by numerical noise caused it to be singular. We obtained an inverse using singular value decomposition (Lanczos 1961). After some experimentation, we chose to construct an inverse from the largest 140 eigenvalues of C_D, which also produced an excellent recovery of the original matrix.

4.2 Data fit of the two ensembles

Fig. 3 shows histograms of the χ_ν^2 values of data fit for all models in both ensembles, where

$$\chi_\nu^2(\mathbf{m}) = \frac{1}{\nu} (\mathbf{d}_o - \mathbf{g}(\mathbf{m}))^T C_D^{-1} (\mathbf{d}_o - \mathbf{g}(\mathbf{m})) \tag{28}$$

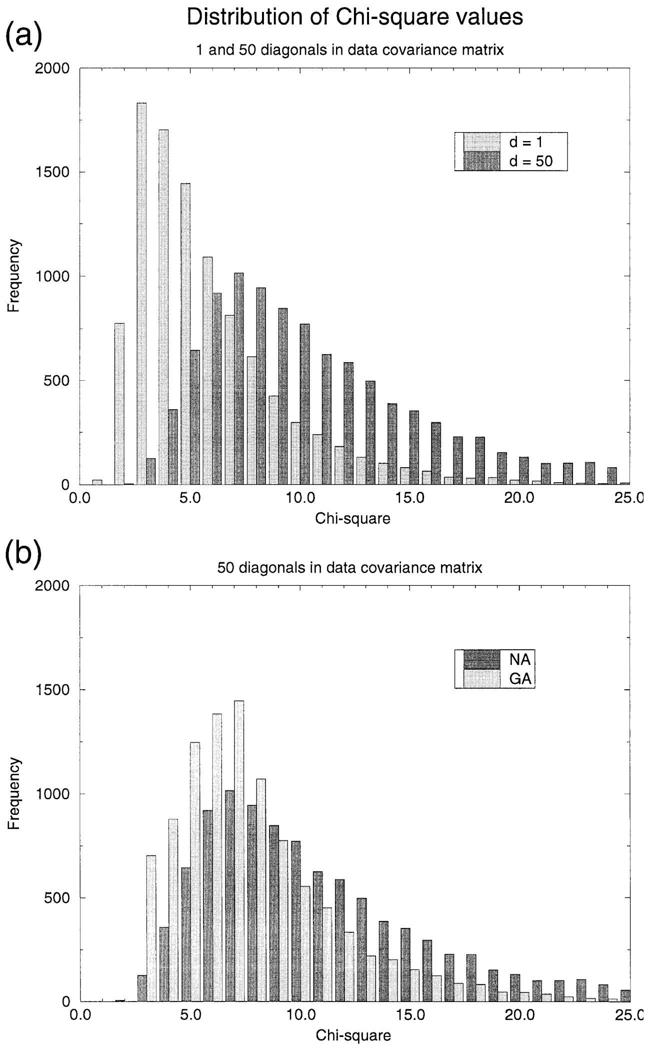

Figure 3. (a) Histograms of the χ^2_ν values of 10 000 models in the 'NA ensemble', determined with two alternative inverse data covariance matrices. The first is constructed using only the main diagonal of C_D (lighter) and the second with ± 50 diagonals (darker). The second histogram takes account of temporal correlation in the receiver function noise. (b) Same as (a) except first histogram (lighter) is of χ^2_ν values of models generated by a genetic algorithm ('GA ensemble'). This ensemble contains many duplicate models with a higher data fit giving a more skewed distribution.

and ν is the number of degrees of freedom (number of data minus the number of independently constrained model parameters). Its value is taken as 116 in evaluating (28) (i.e. $140-24$). Note that this is an approximation because we do not expect all model parameters to be well constrained. The two histograms in Fig. 3(a) were determined from the NA ensemble using the data covariance matrix restricted to the main diagonal (light) and using all ± 50 diagonals (dark). As the number of diagonals is increased, the χ^2_ν values tend to decrease because the temporal correlation of the receiver function noise is taken into account. This trend is also reflected in the best-fit models. The smallest χ^2_ν in the GA ensemble are 1.69 and 3.03 for C_D with 1 and 50 diagonals, respectively. Similarly, in the NA ensemble the best-fit value changes from 1.42 to 1.87. We take the ± 50 diagonal data covariance matrix to be more representative of the noise in the data. (In Paper I the temporal correlation in the noise was ignored and a single diagonal was used to construct C_D^{-1}.)

In Fig. 3(b) we compare the histograms of χ^2_ν, calculated with the full C_D^{-1}, for both the NA and the GA ensembles. This shows that the GA ensemble apparently contains a higher proportion of better data fitting models; however, the figure is rather misleading because the GA ensemble contains multiple copies of models, which tends to skew the histogram towards the lower χ^2_ν values. (An examination showed that of the 100 models with the lowest χ^2_ν values, 50 were complete copies of other models, and over 30 per cent of the entire GA ensemble was at least partial copies of other models.) The NA ensemble contained no copies and no pair of identical χ^2_ν values.

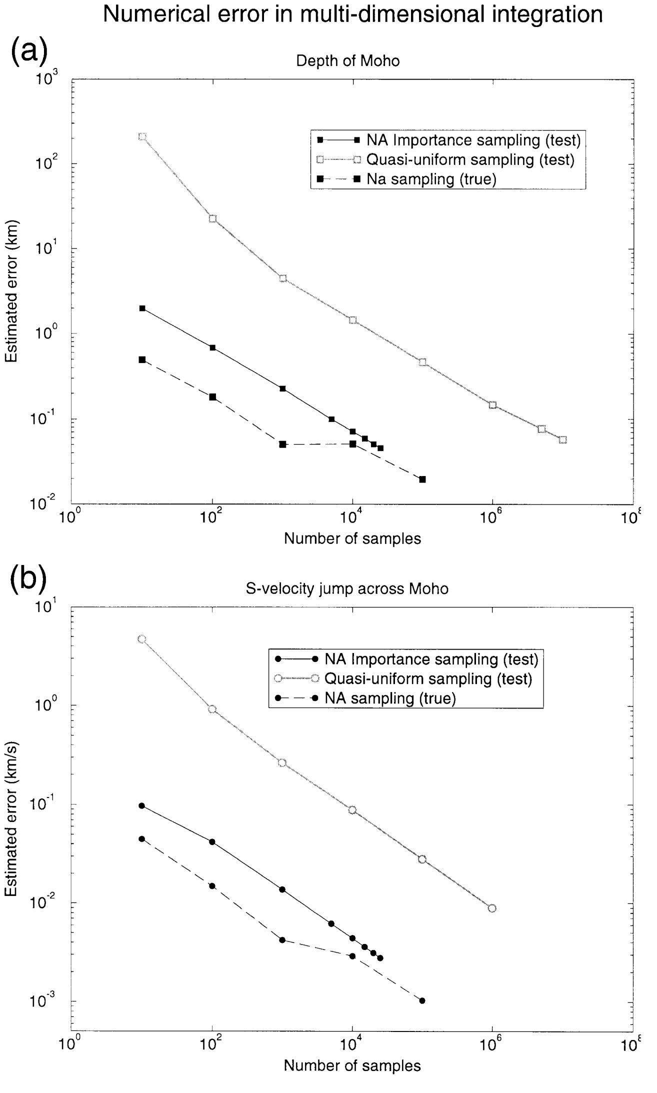

Figure 4. Estimated error in numerical integration for (a) mean Moho depth and (b) mean *S*-velocity jump across the Moho, as a function of the number of samples in the numerical integration. In both figures the open symbols represent a uniform Monte Carlo integration on a test function (see text); the solid lines (filled symbols) are the corresponding curves for the new resampling algorithm. The dashed curves are for the resampling algorithm applied to the neighbourhood approximation of the PPD. The resampling scheme has a much faster error reduction than uniform integration, suggesting that it is able to importance sample the PPD.

4.3 Estimating Bayesian integrals

With the data covariance matrix and prior information established, the PPD can be written as follows:

$$P(\mathbf{m}) = k \exp\left(-\frac{v}{2}\chi_v^2(\mathbf{m})\right); \tag{29}$$

compare eqs (1), (2) and (28). The PPD of the 10 000 models in the two ensembles is therefore known, although as stated above we only need their logs and hence the term in the brackets. The task of the resampling algorithm is then to calculate the Bayesian integrals in eqs (3)–(6) using either of the available ensembles.

4.3.1 Verifying importance sampling

To determine whether the new approach is adequately importance sampling the multidimensional approximation to the PPD, we evaluate the mean integrals in eq. (3) for two transformed parameters, the Moho depth (Z_{moho}) (defined as the sum of the first five layer thicknesses) and the *S*-velocity jump across the Moho (ΔS_{moho}) (defined as the difference between the *S* velocity at the top of layer six and that at the bottom of layer five.) The importance sampling can be assessed by comparing the rate at which the integration error reduces. We evaluate these integrals using both the resampling algorithm and a simple uniform Monte Carlo integration (UMC). In both cases the approximate PPD is built from the NA ensemble. By definition, the UMC estimate is guaranteed to produce an accurate result, albeit with extremely slow convergence.

It turns out that it was not possible to evaluate these, or any other, integrals using UMC. This is because UMC requires direct evaluation of the PPD, which resulted in an arithmetic underflow in the computation. One can see from Fig. 3 that the largest χ_v^2 values in both ensembles is greater than 25, which means that the actual PPD values vary by many orders of magnitude. This effect is likely to occur in many applications, and shows the merit of avoiding direct evaluation of the PPD. In order to make a comparison with the resampling algorithm we devised a test function to replace the PPD in the integrals. This was produced by using eq. (13) to rescale the PPD values so that the range of χ_v^2 values was reduced.

In Fig. 4, three error curves are plotted for both Z_{moho} and ΔS_{moho}. The first two (indicated by open and filled squares) show the integration errors for UMC and the resampling method using the test function, and the third (dashed line) that for the resampling algorithm using the full $P_{\text{NA}}(\mathbf{m})$.

One can see from Figs 4(a) and (b) that the integration error in the means reduces much more rapidly using the resampling algorithm on the test functions than with uniform sampling. For example, after 10^7 samples UMC has a higher integration error than the resampling algorithm after only 2.5×10^4 samples. The curves for the real PPD decrease slightly faster than using the test function, so we assume that a similar level of importance sampling occurs here. These results were found to be characteristic of all other integrals that we evaluated. We conclude that the NA resampling algorithm is able to importance sample the approximate PPD.

4.3.2 The posterior model covariance matrix

Each element of the 24 × 24 posterior model covariance matrix, $C_{i,j}$, was determined using the resampling algorithm. In this case we generated 10^5 samples from 100 independent random walks starting at the 100 best data fitting models in the ensemble. We included the 12 transformed parameters shown in Table 1. The diagonals of the covariance matrix can be taken as 'standard errors' in the parameters. For the off-diagonal elements we can plot the matrix. Since the variables differ in type and dimension it is difficult to display the covariance

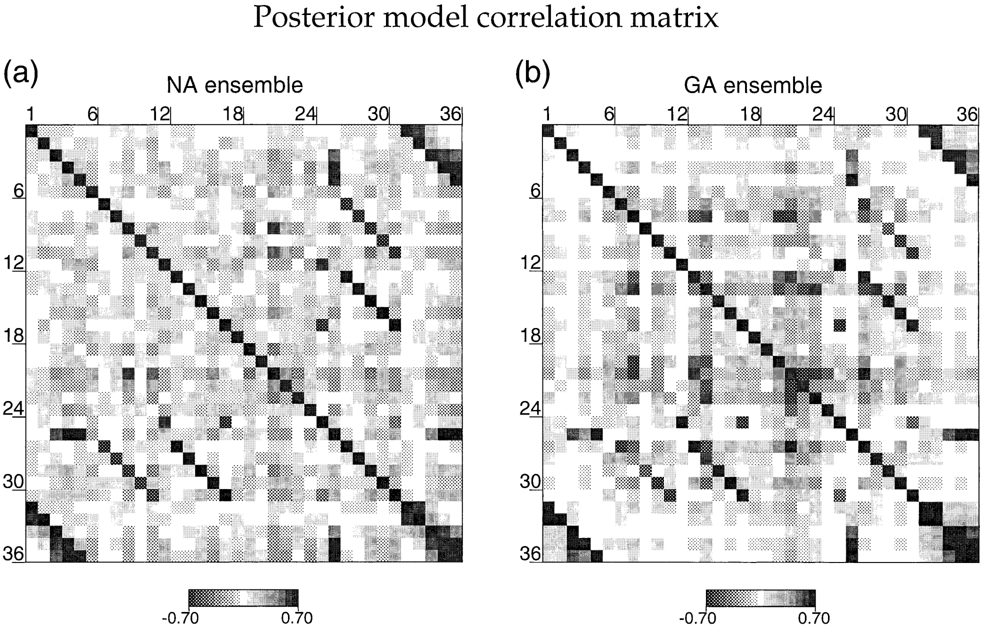

Figure 5. Posterior model correlation matrix from (a) the NA ensemble and (b) the GA ensemble. Each element of this matrix is the result of a multidimensional integration with the resampling algorithm. In all cases 10^5 resamples were used. The elements in the first 24 rows and columns are the original model parameters and the last 12 are determined from combinations of these parameters (see Table 1). Several patterns and trade-off features are observed.

matrix directly. Instead, we calculate the correlation matrix, defined by

$$\Gamma_{i,j} = \frac{C_{i,j}}{\sqrt{C_{i,i}C_{j,j}}}, \tag{30}$$

which conveys similar information in the off-diagonal elements.

Fig. 5(a) shows the correlation matrix calculated from the NA ensemble for all 36 parameters. The strongest patterns in the off-diagonal elements are simply due to the dependence of the transformed parameters on the original parameters; however, many other more subtle patterns are present, too numerous to discuss in detail. (Here we display the entire matrix, although for real problems one would probably need to plot submatrices of selected parameters to examine the trade-offs in detail.) Some simple observations can be made from Fig. 5. The covariances of the layer thickness parameters (upper left corner) show a negative correlation between the first and deeper layers, the strength of which decreases with depth. The negative correlation between the first two layers is reasonable because these layers are largely responsible for the high-amplitude reverberation in 0–3 s of the receiver functions (for examples see Paper I). If one layer is thick then a similar alignment of the high-amplitude phase can be obtained by making the other layer thinner. We note also that the velocity gradients in the first two layers (parameters 27 and 28) are also negatively correlated. On average the V_p/V_s ratio parameters have higher-amplitude correlations with each other and with other parameter types, suggesting that they may be more poorly constrained.

The equivalent calculation for the GA ensemble (shown in Fig. 5b) leads to a covariance matrix with a broadly similar pattern to that from the NA ensemble, although the amplitudes differ in some parts of the matrix. In particular, all cross-correlation coefficients involving the layer thickness parameters seem to have relatively small amplitude, which apparently indicates that these parameters are independently resolved, when in fact we know them not to be. This may be an indication that the GA ensemble contains less information on layer thicknesses. The fact that we obtain similar results based on the two different ensembles is encouraging, and suggests that the resampling algorithm is estimating the true posterior covariance matrix.

4.3.3 1-D marginal distributions

The next indicators we examine are 1-D marginal distributions. By comparing eqs (6), (8) and (17) one also sees that the marginals are equal to the distribution of the resampled ensemble projected onto the corresponding axes. For example, the 1-D marginal for the *i*th parameter is given by a histogram of the *i*th variable of the models in the resampled ensemble.

Figs 6(a) and (b) show the prior and posterior marginals for Z_{moho} and ΔS_{moho}. (Note that the prior distributions are not uniform because they are transformations of the original variables.) With both the GA and the NA ensembles, the posterior distribution is narrower than the prior, due to the information provided by the data. Note that the peak of the marginal is at the true value of the Moho depth. This is very encouraging, and strongly suggests that the resampling algorithm has been able to extract a realistic marginal from both distributions. We note that the GA marginal does not contain the higher probabilities at shallower depths seen in the NA ensemble.

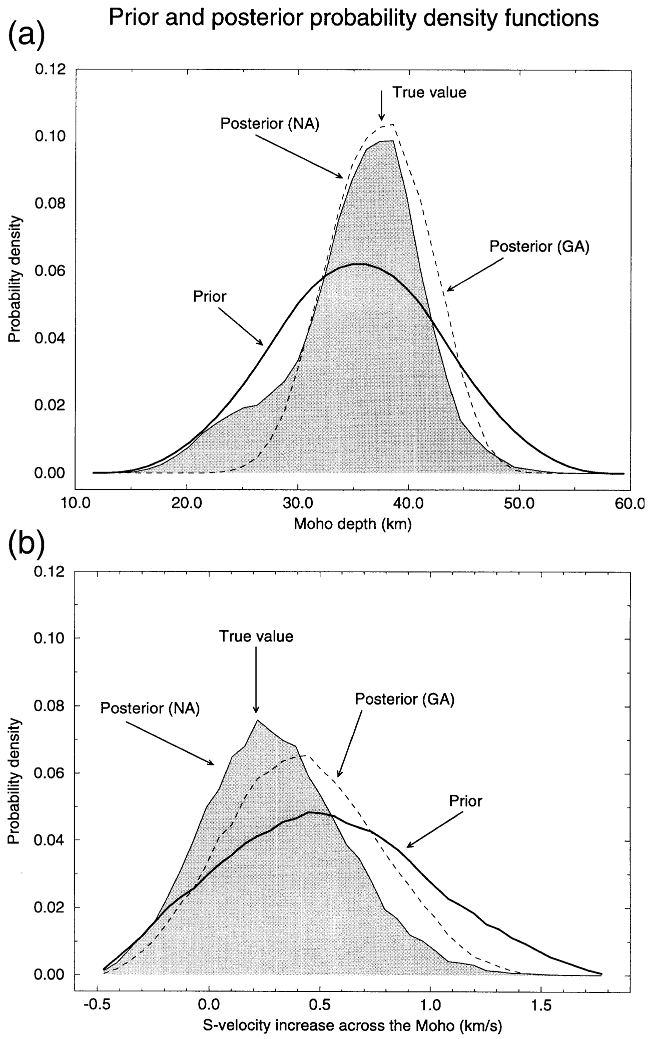

Figure 6. (a) The prior and posterior marginal PDFs for the Moho depth determined from the NA and GA ensembles. The posteriors are determined using the resampling algorithm. In both cases the posterior has a narrower width than the prior, with the peak at the true value. The NA ensemble also gives an increased likelihood at lower depths relative to the GA ensemble. (b) Same as in (a) but for the *S*-velocity jump across the Moho. The areas under all curves are the same.

For the ΔS_{moho} marginals determined from the NA ensemble (Fig. 6b), we again see a narrower posterior than the prior, with a peak centred on the true value. In this case the GA ensemble lies midway between the prior and the NA posterior. The difference between these curves cannot be attributed to a lack of information in the data (which is common to both) but must reflect the information contained in the ensembles themselves. The GA ensemble appears to be inferior to the NA ensemble in sampling this parameter.

These results suggest that the resampling algorithm is able to construct the posterior 1-D marginals accurately for these two parameters, if enough information exists in the input ensemble. Also, they indicate that, in this case, the posterior marginals provide useful information on the true values of the model parameters. Note that the width of the posterior marginals is another indicator of the degree of constraint that may be placed on each parameter, together with the model variances, $\sigma_{\rm moho}$ and $\sigma_{\Delta S}$, obtained from the posterior model covariance matrix.

Fig. 7 shows similar marginals calculated using the NA ensemble for all 24 original parameters, together with their true values. In all cases the prior distributions are uniform. In this figure the plots are individually normalized to a maximum amplitude, so it is not possible to compare densities between variables directly. We see that the shape and spread of the marginals varies significantly. On average the peaks of the marginals (when they exist) appear reasonably well correlated with the true values. The correlation appears to be strongest with the velocity parameters at the base of each layer (13–18), and weakest with the V_p/V_s ratio parameters, which appear to be poorly resolved. The first layer (defined by parameters 1, 7, 13 and 19) and the fourth layer (4, 10, 16 and 22) seem to be the best resolved layers, since all marginals are peaked close to the true values.

The 1-D marginals also provide a useful way of testing whether the resampling algorithm is needed in the first place. We recall that resampling could be avoided if the input ensemble were already distributed according to the PPD. We ask what difference it makes if we simply assume that the input ensemble is distributed according to the PPD. This is equivalent to setting

$$h(\mathbf{m}) = P(\mathbf{m}) \qquad (31)$$

in eq. (8). In this case there is no need to generate a resampled ensemble and all Bayesian integrals reduce to simple averages over the input ensemble. Also, the marginals (in eq. 6) simply become the distributions of the input ensemble projected onto the appropriate axes (we call these the 'raw' marginals). Fig. 8 shows the raw and resampled marginals for the NA ensemble, plotted with the same area normalization.

Significant differences can be seen between the two sets of curves. The 'raw' 1-D marginals all tend to have higher amplitudes and narrower peaks than the resampled marginals; that is, they imply greater constraint on the model parameters than is actually present. Note that there are cases where the shape of the resampled marginal differs significantly from the raw marginal, and has a peak closer to the true value (e.g. parameter 16). Also, there are cases where a double peak in the raw marginal has been removed (parameter 9), and where one has been added (parameter 12). The assumption that the input ensemble is distributed according to the PPD seems to lead to poorly determined marginals, which give a false impression of accuracy, and potentially a completely false shape as well. Note that the roughness of the raw marginals is partially due to the smaller size of the ensemble (10^4 compared to 10^5 in the resampled marginals).

Fig. 9 shows a similar plot for the GA ensemble. The most striking feature here is the difference between the GA raw marginals and those in Fig. 8 for the NA raw ensemble. This difference reflects the underlying differences in the two ensembles. The GA ensemble produces marginals with

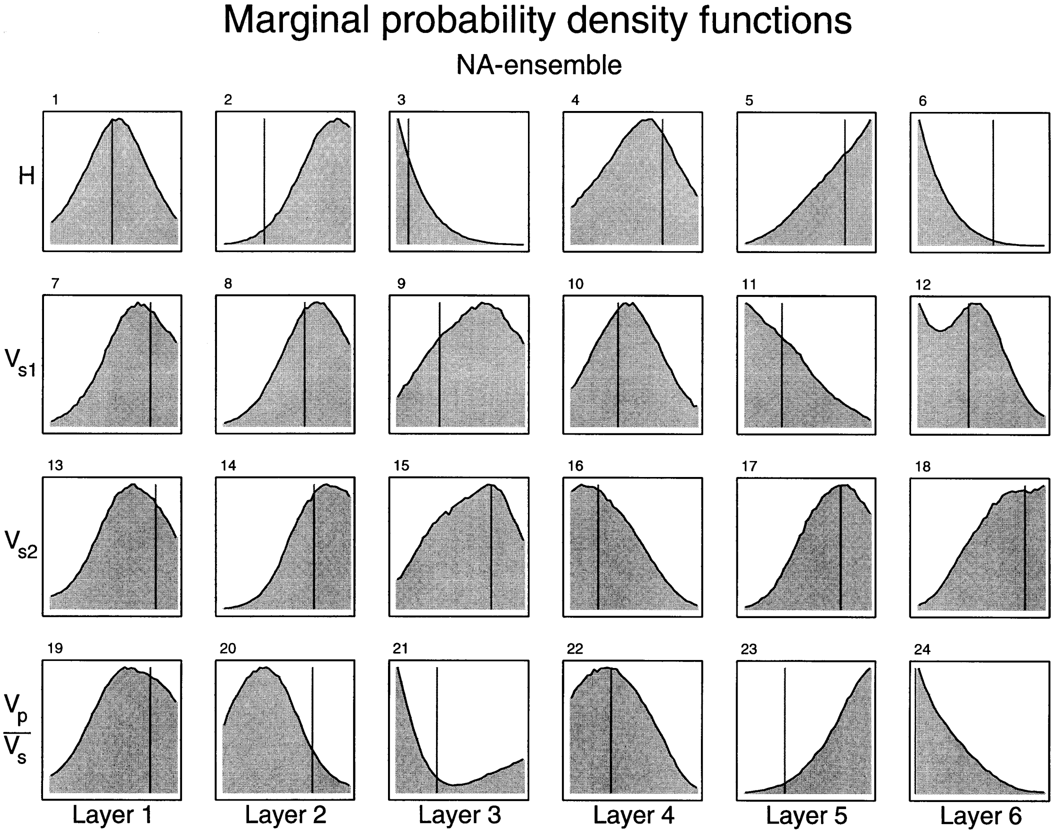

Figure 7. 1-D marginal PDFs for all 24 model parameters obtained by the resampling algorithm (using the NA ensemble). The prior distributions for each are uniform. The panels are arranged so that each row represents a different parameter type and each column a different layer in the velocity model. The x-axis of each panel is over the complete range of the parameter, found in Table 1. The solid lines show the true value of each model parameter. Each curve is scaled to the same maximum height, not the same area.

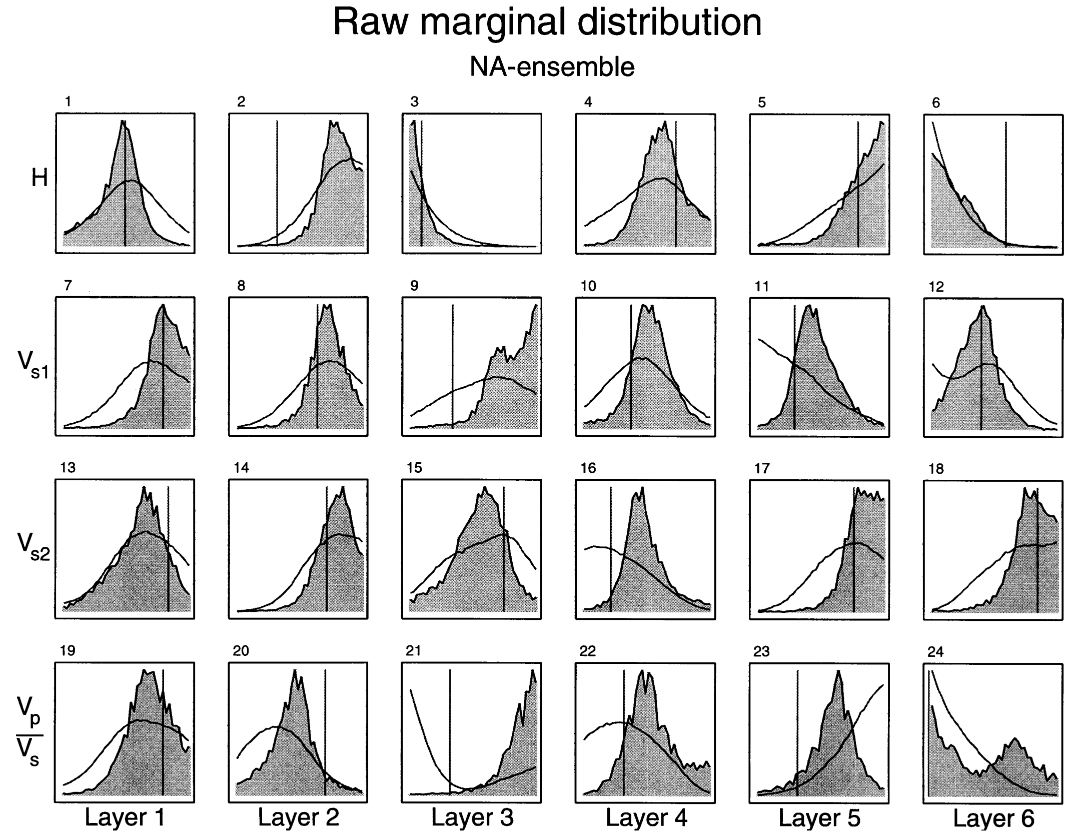

Figure 8. Marginal distributions, from the 'NA ensemble', assuming that it is distributed according to the PPD (shaded), compared to the corresponding marginals determined by the resampling algorithm, i.e. same as in Fig. 7 (unshaded). The areas under the two curves are equal in each panel. The two sets differ quite markedly, indicating that the input ensemble is not distributed according to the PPD.

multiple spikes, which are a result of the crude discretization of the parameter space used by the genetic algorithm which generated it (see Paper I). The amplitudes of the spikes are very large and dominate over the resampled marginals, which appear quite small on the plot (due to the area normalization). Nevertheless, we again see that the resampled marginals are not dominated by the irregular distribution of the underlying ensemble, but instead are more distributed and smooth. In this case the peaks are not as well correlated with true values as those from the NA ensemble, again indicating that the GA ensemble contains less information.

4.3.4 2-D marginal distributions

Figs 10, 11 and 12 show 2-D posterior marginals calculated with the resampling algorithm for selected pairs of parameters using the NA ensemble. Fig. 10 shows the prior and posterior for the (Z_{moho}, ΔS_{moho}) pair. The posterior is clearly more highly peaked than the prior (note the greyscales) and its peak is shifted close to the position of the true values (triangle). In this case accurate information has been extracted from the ensemble, and the posterior itself seems to indicate reasonable resolution in the data for these parameters.

Fig. 11 is a similar plot for the S velocity just above and just below the bottom of the Moho. (Note that the difference in these parameters is ΔS_{moho}.) The prior in this case is uniform, with a value falling in the second lowest contour interval. In this case we again see a well-defined posterior with a peak close to the true values (triangle). There is also a suggestion of a secondary maximum in the posterior just beyond the bounds of the parameter space.

To get an impression of how representative these results are across a range of parameters, we plot a selection of 2-D prior and posterior marginals in Fig. 12. The first set of five plots show the velocity gradient in each of the first five layers (x-axis) against the S velocity at the base of the layer (y-axis). The next five are various combinations of parameters. The correlation between the position of the peak of each marginal and the true values (triangles) is quite high. However, in a few cases it is very poor (e.g. for variables 2 and 8, and 2 and 14). Note that a poor correlation does not necessarily mean that the posterior has not been recovered well, but could indicate that the data contain little information on these variables. Overall, the shape of the marginals varies significantly between plots, which we interpret as reflecting the relative constraint imposed by the data. Many of the 2-D marginals show a clear correlation between the true values and their peaks, suggesting that the resampling algorithm has recovered the posterior reasonably well.

4.3.5 Resolution kernels

In cases where different parameter types are involved (e.g. layer thicknesses and seismic velocities), the non-diagonal elements of the resolution matrix (5) become dimensionally dependent. This can be seen by close inspection of eq. (5), but is also made

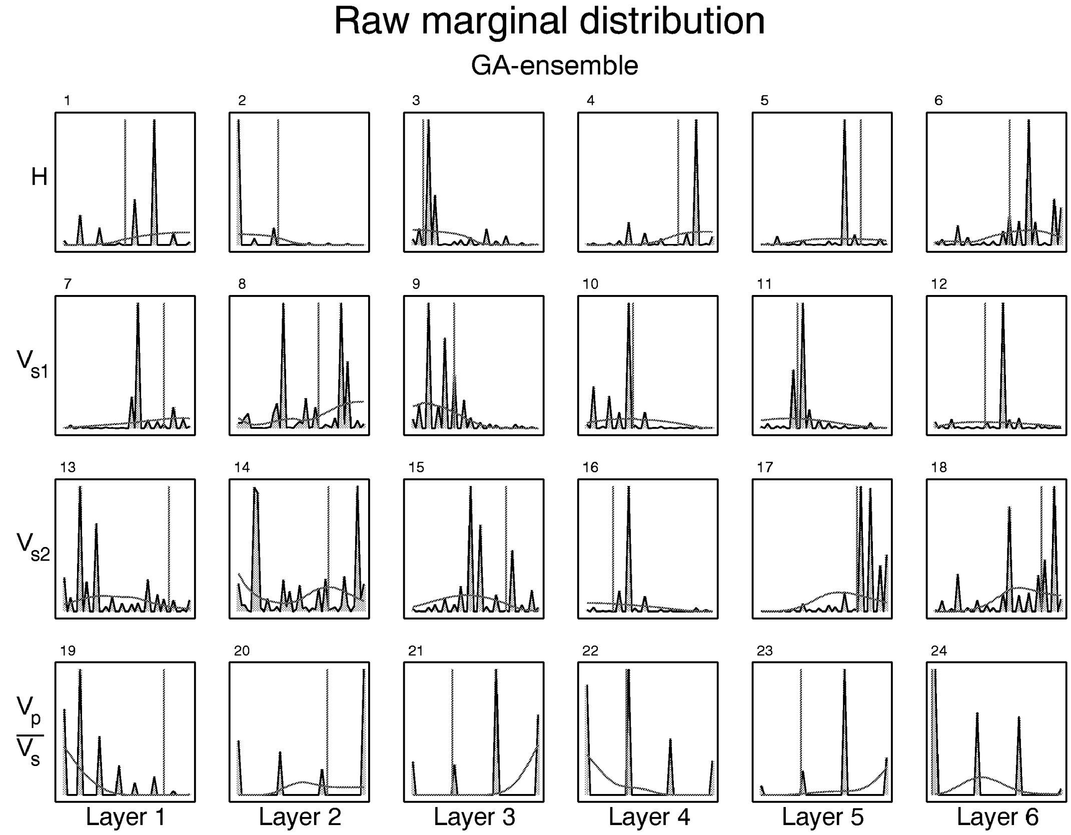

Figure 9. Same as Fig. 8 but for the GA ensemble. Note that the GA ensemble results in many tall spikes which arise from the discretized nature of the parameter space used by the GA. The resampling algorithm smoothes these out. The NA and GA ensembles have very different sampling densities.

clear when we recall the definition of the resolution matrix for a discrete linear inverse problem (see Menke 1989),

$$\mathbf{m}_{\rm est} = R\mathbf{m}_{\rm true} \,, \tag{32}$$

where $\mathbf{m}_{\rm true}$ are the true earth values of the parameters and $\mathbf{m}_{\rm est}$ are the estimated values. Clearly, the ijth element of the resolution matrix $R_{i,j}$ will have the dimension of parameter i divided by that of parameter j. The sizes of the off-diagonal elements will therefore be influenced by the relative scale of the different parameter types and do not lend themselves easily to plotting. Rather than plotting the rows or columns of R, we define a 'non-dimensional' resolution matrix, $R'_{i,j}$, by multiplying each element by σ_j/σ_i, where σ_i is a representative scale length for parameter i. Here we choose the square roots of the diagonals of the prior covariance matrix, $C_{\rm M,prior}$, so

$$R'_{i,j} = R_{i,j}\,\frac{\sigma_j}{\sigma_i} \,. \tag{33}$$

Fig. 13 shows an example of 'non-dimensional' resolution kernels (i.e. the rows of R') determined by applying the resampling algorithm to the NA ensemble. The prior model covariance is calculated using the formulae in the Appendix, and the resolution matrix is determined using (5). We see from (32) that each column of the resolution kernel gives an indication of how the 'true earth' model parameters influence the estimated values. Conversely, the rows of the resolution matrix show how well each parameter can be independently resolved. Strictly speaking, these concepts apply to linearized inverse problems, but they nevertheless give an impression of resolution for the non-linear case. [Snieder (1991), Snieder (1998) presents an extension of resolution kernels for the non-linear case.]

Fig. 13 shows that the level of contamination in the receiver function problem varies significantly between parameters. The thicknesses and velocities in the first two layers are the best-resolved parameters, whilst the V_p/V_s parameters are the worst resolved. Note that when 'leakage' occurs it does so across all parameter types, not just between those in the same class. It is particularly severe between the V_p/V_s parameters. This is consistent with the previously calculated Bayesian indicators, and also our prior expectation that the V_p/V_s parameters would be poorly resolved.

4.3.6 *Convergence of the Gibbs sampler*

All of the results presented here rely on the convergence of the Gibbs sampler, that is, the assumption that eq. (17) has been satisfied, and hence the resampled ensemble is distributed according to the approximate PPD, $P_{\rm NA}(\mathbf{m})$. Convergence to the required distribution can be monitored using standard statistical techniques. We recall that the resampled ensemble is generated from $N_{\rm W}$ independent random walks. In this case the convergence of the Gibbs sampler can be monitored by calculating the 'potential scale reduction' (PSR) factor for all

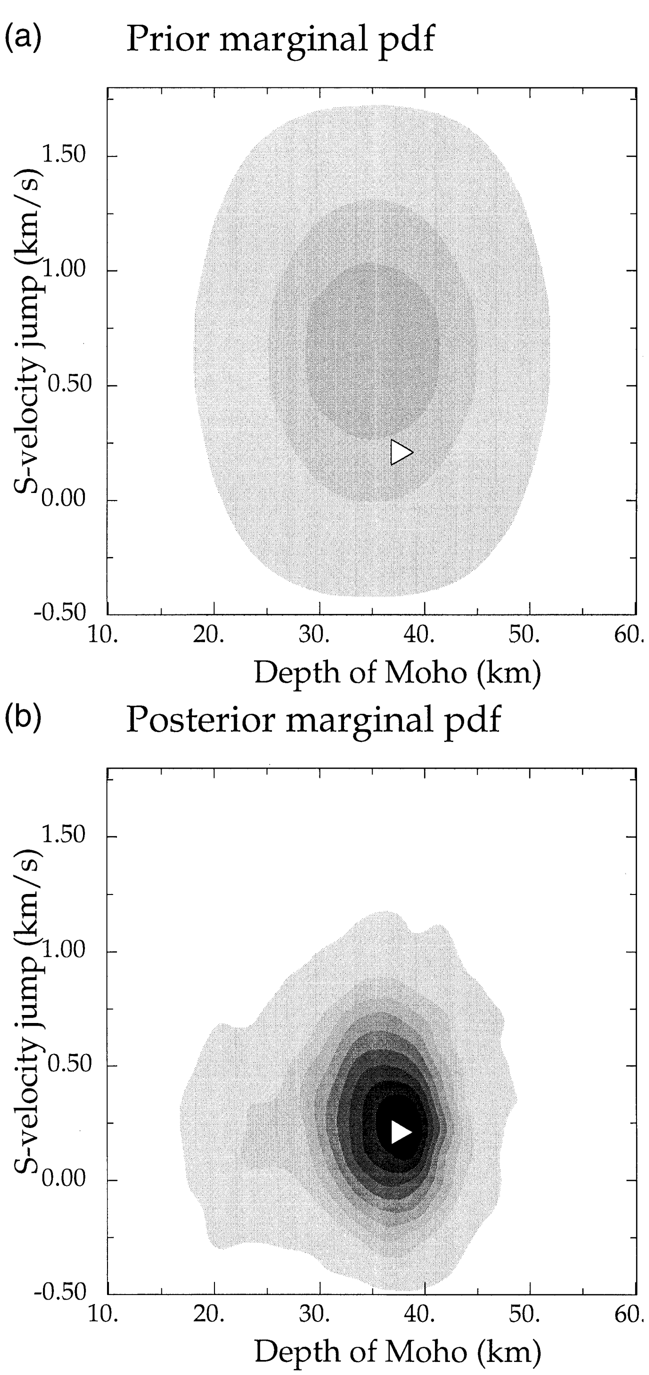

Figure 10. (a) prior and (b) posterior 2-D marginal PDFs for Moho depth and *S*-velocity jump across the Moho calculated with the resampling algorithm from the NA ensemble. The triangle represents the position of the true values and the shading scale indicates the value of the probability density function. Note that the posterior is more concentrated than the prior and its peak is close to the true values.

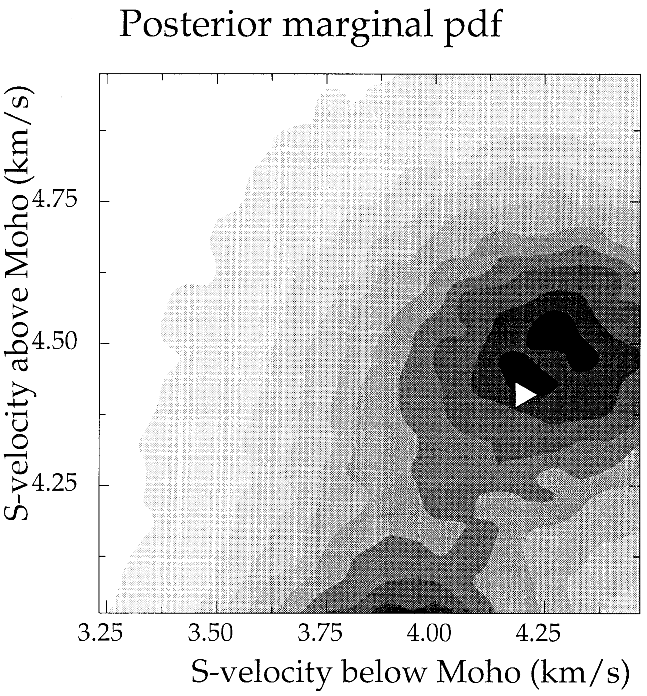

Figure 11. Posterior 2-D marginal PDF for the parameters representing the *S* velocity above and below the Moho. In this case the prior is uniform in both parameters and falls in the second lowest (lightest) shading interval. The posterior has a well-defined peak close to the true values (triangle). Note that there is a suggestion of a secondary peak in the marginal beyond the range of the parameter space.

estimands of interest (see Gelman *et al.* 1995 and Tanner 1996 for full details). [Estimands include all parameters and other quantities of interest, i.e. that represented by $g(\mathbf{m})$ in eq. (7).] The PSR factor is a scalar which measures the difference between the 'within-walk' and 'between-walk' variances for any estimand. Let g_{ij} be the ith estimand from the jth walk (see eqs 20–22), and let each walk generate n samples, then we write W for the average of the within-walk variances for g_{ij},

$$W=\frac{1}{N_{\mathrm{W}}}\sum_{j=1}^{N_{\mathrm{W}}} s_j^2\,, \tag{34}$$

where

$$s_j^2=\frac{1}{n-1}\sum_{i=1}^{n}(g_{ij}-\overline{g}_j)^2\,. \tag{35}$$

The between-walk variance, B, is given by

$$B=\frac{n}{N_{\mathrm{W}}-1}\sum_{j=1}^{N_{\mathrm{W}}}(\overline{g}_j-\overline{g})^2\,, \tag{36}$$

where $\overline{g}_j$ is given by (22), and

$$\overline{g}=\frac{1}{N_{\mathrm{W}}}\sum_{j=1}^{N_{\mathrm{W}}}\overline{g}_j\,. \tag{37}$$

The potential scale reduction factor, $\sqrt{\hat{R}}$, is then given by

$$\sqrt{\hat{R}}=\left[\frac{1}{W}\left(\frac{n-1}{n}\,W+\frac{1}{n}\,B\right)\right]^{1/2}. \tag{38}$$

The PSR factor decreases to 1 as $n\rightarrow\infty$. If the value is high then the variance within the walks is small compared to that between the walks and there is reason to believe that longer walks are needed to achieve convergence. Usually a Gibbs simulation is considered acceptable if all values of $\sqrt{\hat{R}}$ (for all variables) are less than 1.2 (Gelman *et al.* 1995). In the calculations performed here ($N_{\mathrm{W}}=100$, $n=1000$), PSR factors

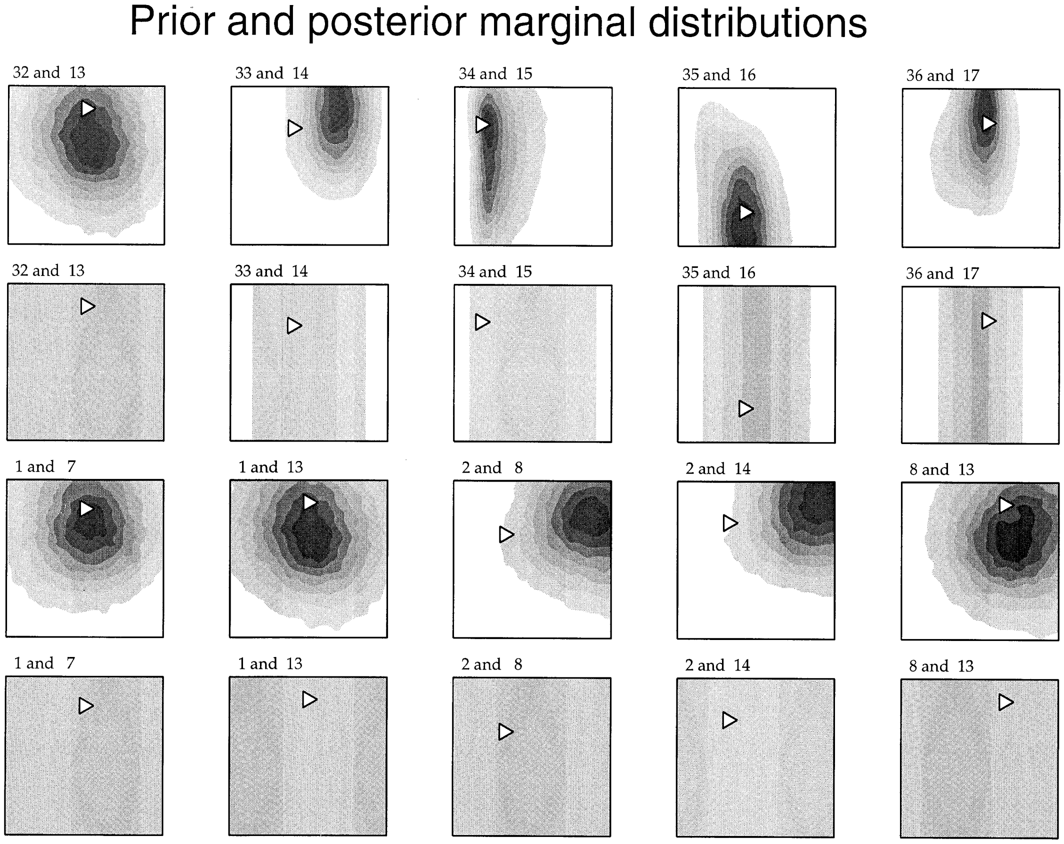

Figure 12. 10 selected 2-D marginals calculated from the NA ensemble, together with their prior distributions below. Each pair of parameter indices is labelled above the panel. The first parameter is plotted on the x-axis and the triangle represents the true values. The shading scale is normalized for each pair of prior and posterior plots, so a comparison of relative heights (greyscales) is only meaningful between prior and posterior pairs. Many, but not all, show a peak near the true value.

were calculated for all 36 parameters in Table 1. For the NA ensemble the maximum PSR factor was 1.18 and the median was 1.09; for the GA ensemble the maximum was 1.19 and the median was 1.01. Therefore, PSR factors for all posterior means suggest that the Gibbs sampler has converged reasonably well.

Another test for convergence is to examine the dependence of the results on the starting points for each random walk. Most of the results were generated with 100 independent random walks, starting from the 100 best data fitting models in the input ensemble. In each case we repeated the calculations starting from randomly chosen starting models. In all cases the results were virtually identical to those shown in Figs 5–13. A final test was performed to determine the influence of the starting point on each individual walk. This consisted of increasing the length of each walk to twice the number of samples (i.e. 2000), but collecting the results from the second halves of each walk. (Note that this means that the same number of samples is used to evaluate the Bayesian integrals and marginals, only they are generated further down each chain.) We did this for both the GA and the NA ensembles and again all results were indistinguishable from those presented above.

From these tests we can conclude that the Gibbs sampler has converged. The resamples may therefore be treated as being drawn from the approximate PPD, $P_{\mathrm{NA}}(\mathbf{m})$ (hence 17 is satisfied). If this had not been the case and the Gibbs sampler had exhibited excessively slow convergence, then it is possible to try and speed up convergence. It is well known that slow convergence of a Gibbs sampler can occur if the parameters are highly dependent (Gelman *et al.* 1995). Here this might occur if a Voronoi cell containing a relatively high PPD value were elongated and inclined at $\pi/4$ to several axes. The most commonly used technique to speed up convergence is to rotate the parameter space so that the axes become aligned with independent parameters. These could be detected from the correlation matrix built from some trial random walkers using the original parameter axes (see Smith 1991, Gelman *et al.* 1995 and Tanner 1996 for further details). This remains an option for the NA resampling algorithm, although it has not been necessary in the example given here.

4.3.7 *Computational costs*

All of the Bayesian integrals (with the exception of those in Fig. 4) were calculated with 10^5 samples generated from 100 independent random walks. Since the parameter space has 24 dimensions, then the 10^5 samples required the Voronoi intersection problem to be solved 2.4 million times, and this resulted in approximately 10.5 million intersections, with an average of just over four Voronoi cells being intersected for each axis. Overall, 93 per cent of the Voronoi cells were intersected by the axes.

The calculations were performed on four separate SUN Ultra 5 workstations simultaneously, and the total amount of CPU time taken was 3.9×10^4 s ($\approx10^4$ s on each machine). This represents almost perfect parallelization because the CPU

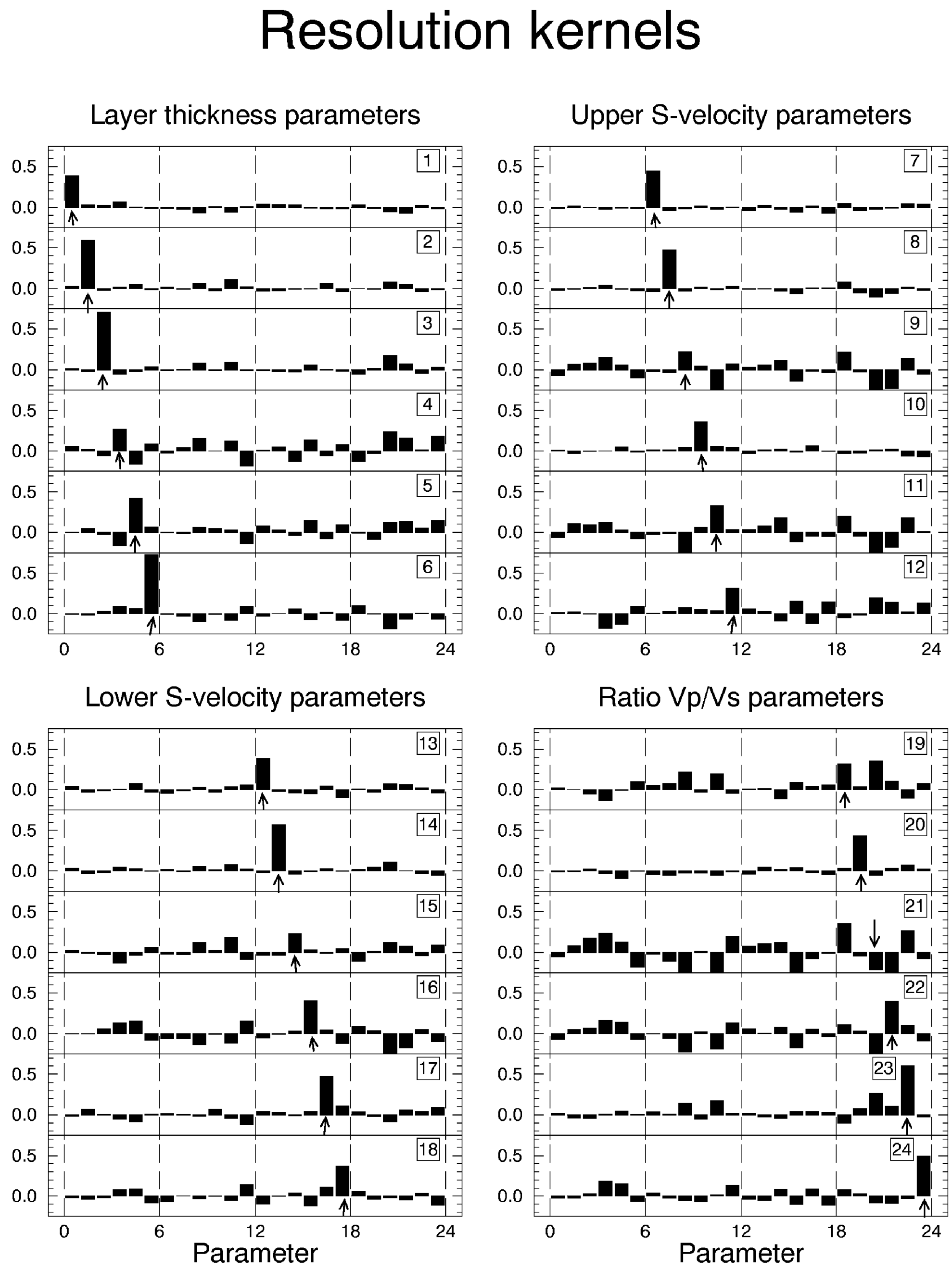

Figure 13. Non-dimensional resolution kernels, i.e. rows of the estimated 'non-dimensional' resolution matrix [see eq. (33) for each of the 24 model parameters]. Each row shows how the particular model parameter (indicated by the arrow) is contaminated by estimates of the other model parameters. The relatively poor resolution in V_p/V_s parameters (19–24) is evident. The shallow layer thickness parameters (1–3) are better resolved than the deeper ones (4–6). Numbers in boxes indicate the parameter index.

time spent calculating and combining the ensemble averages was just 0.2 per cent of the total. These figures suggest that the algorithm presented here is reasonably efficient for the 24-D problem and may be practical in much higher dimensions, especially if parallelization can be exploited.

5 DISCUSSION

The key idea in this paper is to extract information from an ensemble of forward solutions by constructing a multidimensional interpolant in model space. The resampling algorithm consists of drawing random deviates from the neighbourhood approximation to the PPD, and using these as the basis of Monte Carlo integration. In this way any Bayesian integral can be evaluated. However, the accuracy of the approximation and hence the numerical integrals will, necessarily, depend on how well the input ensemble samples the regions of high data fit. This will be reflected in the accuracy of the approximation in eq. (16). There seems to be no comprehensive way of assessing how well this approximation is satisfied, without extensive further solving of the forward problem. However, a simple way to test the 'quality' of the input ensemble would be to experiment with different subsets and examine the variability in the results. Clearly, if little information is contained in the ensemble then the results will be poor. The objective of the algorithm presented in this paper is to extract what information exists in an input ensemble of any size, generated with any method, so the quality of the input ensemble will always be an issue.

In the receiver function example presented here, the resampling algorithm appears to have worked well. We have also shown that it is reasonably efficient computationally and demonstrated its parallel nature. The Bayesian indicators recovered with the resampling algorithm collectively provide information on the degree of constraint, resolution and trade-off between different parameters. Any transformation of the variables can be treated in the same way. Not all of these results are simple to interpret. In the synthetic example presented here the 1- and 2-D marginal distributions were the most useful in assessing the information content of the data, and distinguishing between the 'quality' of the two input ensembles. All Bayesian integrals and their error estimates can be evaluated by collecting simple averages over the resampled ensemble.

A powerful feature of the resampling algorithm is that nothing is assumed about the distribution of the initial ensemble of earth models. This means that it may be used in a variety of situations, for example, the quantitative appraisal of the ensemble produced by a genetic algorithm, which was previously only possible in a qualitative manner with graphical methods. It may be used in conjunction with the new direct search algorithm described in Paper I, or even as a 'correction' procedure, in cases where it was previously assumed that an ensemble was already distributed according to the PPD. One could equally well apply it to the ensemble produced from the combination of several different search methods.

A prerequisite of any Bayesian method is a suitable definition of a prior and a likelihood function (eqs 1 and 2). The accuracy of all results will be dependent on these terms. The NA resampling method is no different from any other Bayesian approach in this respect.

The main philosophy behind this paper has been that all solutions to the forward problem should, in principle, contain information. We have presented one particular method to extract that information from an ensemble of solutions. Although the method has its limitations, it provides a way of incorporating all models for which the forward problem has been solved into the appraisal stage of the inverse problem, which is preferable to the alternative of being forced to throw most of them away. The author's computer programs associated with this work will be made available. See http://rses.anu.edu.au/~malcolm/na/na.html for details.

ACKNOWLEDGMENTS

I am grateful for many stimulating discussions and helpful suggestions on this work from Jean Braun, Kerry Gallagher, Oli Gudmundsson and Herb McQueen. I am also indebted to Brian Kennett, Don Vasco and Luis Tenorio, who suggested numerous improvements to the manuscript.

REFERENCES

Ammon, C.J., Randall, G.E. & Zandt, G., 1990. On the non-uniqueness of receiver function inversions, *J. geophys. Res.,* **95,** 15 303–15 318.

Cary, P.W. & Chapman, C.H., 1988. Automatic 1-D waveform inversion of marine seismic refraction data, *Geophys. J.,* **93,** 527–546.

Constable, S.C., Parker, R.L. & Constable, C.G., 1987. Occam's inversion: a practical algorithm for generating smooth models from electromagnetic sounding data, *Geophysics,* **52,** 289–300.

Duijndam, A.J.W., 1988a. Bayesian estimation in seismic inversion part I: principles, *Geophys. Prospect.,* **36,** 878–898.

Duijndam, A.J.W., 1988b. Bayesian estimation in seismic inversion part II: uncertainty analysis, *Geophys. Prospect.,* **36,** 899–918.

Efron, B., 1982. *The Jackknife, the Bootstrap, and other Resampling plans,* Society Industrial and Applied Math, CBMS-Natl. Sci. Found. Monogr., Philadelphia, PA.

Efron, B. & Tibshirani, R., 1986. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy, *Stat. Sci.,* **1,** 54–77.

Flournay, N. & Tsutakawa, R.K., eds., 1989. Statistical multiple integration, *Proc. AMS-IMS-SIAM Summer Research Conference on Statistical Multiple Integration,* Am. Math. Soc., Providence, RI.

Gelfand, A.E. & Smith, A.F.M., 1990. Sampling based approaches to calculating marginal densities, *J. Am. stat. Assoc.,* **85,** 398–409.

Gelman, A., Carlin, J.B., Stern, H.S. & Rubin, D.B., 1995. *Bayesian Data Analysis,* Chapman & Hall, London.

Geman, S. & Geman, D., 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. Patt. Analysis Mach. Int.,* **6,** 721–741.

Gouveia, W.P. & Scales, J.A., 1998. Bayesian seismic waveform inversion: parameter estimation and uncertainty analysis, *J. geophys. Res.,* **103,** B2, 2759–2779.

Gudmundsson, O. & Sambridge, M., 1998. A regionalized upper mantle (RUM) seismic model, *J. geophys. Res.,* **103,** B4, 7121–7136.

Hastings, W.K., 1970. Monte Carlo sampling methods using Markov Chain and their applications, *Biometrika,* **57,** 97–109.

Keilis-Borok, V.I. & Yanovskaya, T.B., 1967. Inverse problems of seismology, *Geophys. J.,* **13,** 223–234.

Kennett, B.L.N., 1978. Some aspects of non-linearity in inversion, *Geophys. J. R. astr. Soc.,* **55,** 373–391.

Kennett, B.L.N., 1998. On the density distribution within the Earth, *Geophys. J. Int.,* **132,** 374–382.

Kennett, B.L.N. & Nolet, G., 1978. Resolution analysis for discrete systems, *Geophys. J. R. astr. Soc.,* **53,** 413–425.

Koren, Z., Mosegaard, K., Landa, E., Thore, P. & Tarantola, A., 1991. Monte Carlo estimation and resolution analysis of seismic background velocities, *J. geophys. Res.,* **96,** B12, 20 289–20 299.

Lanczos, C., 1961. *Linear Differential Operators,* Van Nostrand, New York.

Lomax, A. & Snieder, R., 1995. Identifying sets of acceptable solutions to non-linear geophysical inverse problems which have complicated misfit functions, *Nonlinear Proc. Geophys.,* **2,** 222–227.

Menke, W. 1989. *Geophysical Data Analysis: discrete Inverse Theory,* rev. edn, Academic Press, San Diego.

Metropolis, N., Rosenbluth, M.N., Rosenbluth, AW., Teller, A.H. & Teller, E., 1953. Equation of state calculations by fast computing machines, *J. Chem. Phys.,* **21,** 1087–1092.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. geophys. Res.,* **100,** B7, 12 431–12 447.

Nolte, B. & Frazer, L.N., 1994. Vertical seismic profile inversion with genetic algorithms, *Geophys. J. Int.,* **117,** 162–179.

Okabe, A., Boots, B. & Sugihara, K., 1992. *Spatial Tessellations Concepts and Applications of Voronoi Diagrams,* John Wiley & Sons, New York.

Parker, R.L., 1977. Understanding inverse theory, *Ann. Rev. Earth planet. Sci.,* **5,** 35–64.

Press, F., 1968. Earth models obtained by Monte Carlo inversion, *J. geophys. Res.,* **73,** 5223–5234.

Press, W.H., Flannery, B.P., Saul, A.T. & Vetterling, W.T., 1992. *Numerical Recipes,* 2nd edn, Cambridge University Press, Cambridge.

Rothman, D.H., 1985. Nonlinear inversion statistical mechanics, and residual statics corrections, *Geophysics,* **50,** 2784–2796.

Rothman, D.H., 1986. Automatic estimation of large residual statics corrections, *Geophysics,* **51,** 332–346.

Sambridge, M., 1998. Exploring multi-dimensional landscapes without a map, *Inverse Problems,* **14,** 427–440.

Sambridge, M., 1999. Geophysical inversion with a neighbourhood algorithm—I. Searching a parameter space, *Geophys. J. Int.,* **138,** 479–494.

Sambridge, M. & Drijkoningen, G.G., 1992. Genetic algorithms in seismic waveform inversion, *Geophys. J. Int.,* **109,** 323–342.

Sambridge, M. & Gudmundsson, O., 1998. Tomography with irregular cells, *J. geophys. Res.,* **103,** B1, 773–781.

Sambridge, M., Braun, J. & McQueen, H., 1995. Geophysical parametrization and interpolation of irregular data using natural neighbours, *Geophys. J. Int.,* **122,** 837–857.

Scales, J.A., Smith, M.L. & Fischer, T.L., 1992. Global optimization methods for multi-model inverse problems, *J. Comp. Phys.,* **103,** 258–268.

Sen, M. & Stoffa, P.L., 1995. *Global Optimization Methods in Geophysical Inversion,* Advances in Exploration Geophysics 4, Elsevier, Amsterdam.

Shibutani, T., Sambridge, M. & Kennett, B., 1996. Genetic algorithm inversion for receiver functions with application to crust and uppermost mantle structure beneath Eastern Australia, *Geophys. Res. Lett.,* **23,** 1829–1832.

Smith, A.F.M., 1991. Bayesian computational methods, *Phil. Trans. R. Soc. Lond.,* A, **337,** 369–386.

Smith, A.F.M. & Roberts, G.O., 1993. Bayesian computation via the Gibbs Sampler and related Markov chain Monte Carlo methods, *J. R. stat. Soc.,* B., **55,** 3–23.

Snieder, R., 1991. An extension of Backus-Gilbert theory to nonlinear inverse problems, *Inverse Problems,* **7,** 409–433.

Snieder, R., 1998. The role of nonlinearity in inverse problems, *Inverse Problems,* **14,** 387–404.

Stoffa, P.L. & Sen, M.K., 1991. Nonlinear multiparameter optimization using genetic algorithms: inversion of plane wave seismograms, *Geophysics,* **56,** 1794–1810.

Tanner, M.A., 1996. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions,* 3rd edn, Springer-Verlag, New York.

Tarantola, A., 1987. *Inverse Problem Theory,* Elsevier, Amsterdam.

Vasco, D.W., Johnson, L.R. & Majer, E.L., 1993. Ensemble inference in geophysical inverse problems, *Geophys. J. Int.,* **117,** 711–728.

Voronoi, M.G., 1908, Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *J. reine Angew. Math.,* **134,** 198–287.

Watson, D.F., 1992. *Contouring: A Guide to the Analysis and Display of Spatial Data,* Pergamon, Oxford.

Wiggins, R.A., 1969. Monte Carlo inversion of body wave observations, *J. geophys. Res.,* **74,** 3171–3181.

APPENDIX A: MONTE CARLO ESTIMATES OF BAYESIAN INTEGRALS AND THEIR NUMERICAL ERROR

The general formula for estimating the numerical error in Monte Carlo integrals is given by eq. (12). Applying this to the estimated mean of variable m_i in eq. (3) and using (17) gives immediately

$$\epsilon_{\langle m_i \rangle} = \left[\frac{\langle m_i^2 \rangle - \langle m_i \rangle^2}{N_\mathrm{r}} \right]^{1/2} , \tag{A1}$$

thus two ensemble averages need to be calculated to obtain the numerical error in the mean of any variable. (Note that this is just the well-known expression for the error of a sample mean.)

To find the error in each element of the covariance matrix we first need to rewrite eq. (4) in the form

$$C^{\mathrm{M}}_{i,j} = \frac{1}{v} \int_{\mathcal{M}} (m_i - \langle m_i \rangle)(m_j - \langle m_j \rangle) P(\mathbf{m})\, d\mathbf{m} . \tag{A2}$$

This is the original definition of the ijth element of the covariance matrix. Using eq. (A2) and eqs (17) and (8), we obtain the Monte Carlo estimate of $C^{\mathrm{M}}_{i,j}$,

$$\hat{C}_{i,j} = \frac{1}{N_\mathrm{r}} \sum_{k=1}^{N_\mathrm{r}} (m_i^k - \overline{m_i})(m_j^k - \overline{m_j}) , \tag{A3}$$

where m_i^k denotes the value of variable m_i for the kth member of the ensemble, and

$$\overline{m_i} = \sum_{l=1}^{N_\mathrm{r}} m_i^l . \tag{A4}$$

Note that we have dropped the superscript M for convenience. By expanding the double summation terms we get

$$\hat{C}_{i,j} = \overline{m_i m_j} - \overline{m_i}\, \overline{m_j} , \tag{A5}$$

so each covariance element requires one extra ensemble average to be evaluated, i.e. the cross-term $\overline{m_i m_j}$. Eq. (A5) shows that each element of the covariance matrix can be estimated with a single loop over the ensemble. From eq. (A3), we see that each element of the covariance matrix is itself an ensemble average of the variable $C^k_{i,j}$, where

$$C^k_{i,j} = (m_i^k - \overline{m_i})(m_j^k - \overline{m_j}) . \tag{A6}$$

The numerical error of the ijth covariance element can be found from the variance of $C^k_{i,j}$ over the ensemble, i.e. we apply eq. (12) to get

$$\epsilon_{C_{i,j}} = \frac{1}{\sqrt{N_\mathrm{r}}} [\overline{C^2_{i,j}} - \overline{C_{i,j}}^2]^{1/2} . \tag{A7}$$

By substituting eq. (A6) into eq. (A7) and expanding the summation terms, we obtain

$$\begin{aligned} \epsilon_{C_{i,j}} = \frac{1}{\sqrt{N_\mathrm{r}}} [&\overline{m_i^2 m_j^2} + \overline{m_i^2}\, \overline{m_j}^2 + \overline{m_i}^2\, \overline{m_j^2} - 2\, \overline{m_i^2 m_j}\, \overline{m_j} \\ &- 2\, \overline{m_i}\, \overline{m_i m_j^2} - 4\, \overline{m_i}^2\, \overline{m_j}^2 + 6\, \overline{m_i m_j}\, \overline{m_i}\, \overline{m_j} - \overline{m_i m_j}]^{1/2} , \end{aligned} \tag{A8}$$

which again only requires ensemble averages to be determined, and allows the error estimates to be evaluated with a 'single loop' over the ensemble. The same expressions may be used to determine MC estimates of the means, covariances and their numerical errors for any transformed variable.

A1 Variance estimates of the prior

The prior probability density distribution used in this paper is simply a constant over the parameter space. In this case the prior is separable and the mean of the ith variable according to the prior is given by

$$\langle m_i \rangle = \frac{1}{\Delta m_i} \int_{l_i}^{u_i} m_i \, dm_i , \tag{A9}$$

where $\Delta m_i \equiv u_i - l_i$. This gives

$$\langle m_i \rangle = \frac{1}{2}(l_i + u_i), \tag{A10}$$

as one would expect. The elements of the prior model covariance matrix, C^{prior}, can be determined by replacing $P(\mathbf{m})$ with the prior in eq. (A2). For the *ij*th element we get

$$C_{i,j}^{\text{prior}} = \frac{1}{\Delta m_i \Delta m_j} \int_{l_i}^{u_i} \int_{l_j}^{u_j} (m_i - \langle m_i \rangle)(m_j - \langle m_j \rangle)\, dm_j dm_i, \tag{A11}$$

which gives

$$C_{i,j}^{\text{prior}} = \begin{cases} \frac{1}{\sqrt{12}}(u_i - l_i)^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}. \tag{A12}$$

With these expressions the prior model covariance matrix can be calculated and used in eq. (5) to determine the resolution matrix. If the prior does not take a simple analytical form then it will be necessary to evaluate the above integrals numerically. This can be done by generating an ensemble with a distribution that follows the prior, and using eq. (8). If the prior can be evaluated (up to a multiplicative constant) for any point in model space, then a standard Gibbs sampling algorithm can be used to generate integrals over the prior, that is, by replacing $P_{\text{NA}}(\mathbf{m})$ with the prior distribution in eq. (24). In cases where the prior cannot be directly evaluated then it may still be possible to generate an ensemble distributed according to the 'prior'. Recently, several authors have presented examples of generating an ensemble according to complex priors (Mosegaard & Tarantola 1995; Gouveia & Scales 1998). These authors have also stressed the importance of using a realistic prior in any Bayesian treatment of non-linear inverse problems.