Published in *Computing Science and Statistics*, **27**,
M. M. Meyer and J. L. Rosenberger, eds., pp. 202–211,
(Interface Foundation, Fairfax Station, VA 22039-7460, 1996)

A COMPUTATIONAL APPROACH TO BAYESIAN INFERENCE

K.M. Hanson and G.S. Cunningham

Los Alamos National Laboratory, MS P940
Los Alamos, New Mexico 87545, USA

ABSTRACT

Although the Bayesian approach provides a complete solution to model-based analysis, it is often difficult to obtain closed-form solutions for complex models. However, numerical solutions to Bayesian modeling problems are now becoming attractive because of the advent of powerful, low-cost computers and new algorithms. We describe a general-purpose implementation of the Bayesian methodology on workstations that can deal with complex nonlinear models in a very flexible way. The models are represented by a data-flow diagram that may be manipulated by the analyst through a graphical-programming environment that is based on a fully object-oriented design. Maximum a posteriori solutions are achieved using a general optimization algorithm. A new technique for estimating and visualizing the uncertainties in specific aspects of the model is incorporated.

Keywords: Bayesian analysis, MAP estimator, uncertainty estimation, object-oriented programming, adjoint differentiation, optimization

1. INTRODUCTION

We interpret physical reality in terms of models. Therefore, basic questions about building models from data are of fundamental importance: Which models are appropriate? What are the best values of the model parameters? As measurements can not be made exactly, all interpretations based on measurements can not be made with complete certainty. Bayesian analysis provides the methodology to quantitatively characterize the degree of uncertainty in the models that we build in light of the measurements. In the Bayesian approach the degree of certainty of any parameter is represented as a probability density function defined on that parameter. For example, a wide probability distribution on a parameter means that it is not known well. Conversely, a narrow one means the parameter is accurately known.

Historically, most Bayesian calculations, indeed the majority of signal recovery work, have been grounded on certain standard simplifying assumptions. For example, it is often assumed that there is a linear relationship between measurements and the model parameters that are to be determined. Furthermore, Gaussian probability density functions have been used extensively in Bayesian formulations (and other approaches to statistical analysis). These historical predilections may be understood by the fact that they result in a seductive simplicity in the solutions. Linear measurement models may be handled using the tools of algebra. Gaussian probabilities lead to quadratic expressions in the logarithm of the probabilities. Therefore, when differentiated to find the stationary point corresponding to the maximum a posteriori (MAP) solution, the result is a set of linear equations for the MAP solution for the parameters, which may be conveniently dealt with using the tools of linear algebra. The simplicity of such results accounts for the popularity of the historical assumptions.

However, many researchers have demonstrated the tremendous advantage of deviating from the traditional assumptions. First, the restriction to linearity between the measurements and the model parameters imposes very limiting constraints on the model for the measurement process, as well as on the model for the object. Such a restriction precludes direct use of important sources of experimental data that are represented by nonlinear transformations, for example, radiographic data [1]. Furthermore, many important models for objects imply a nonlinear relationship between measurements and the model parameters, e.g. deformable geometric models to describe the shape of objects [2, 3, 4, 5].

The second historical assumption, that all probability density functions have Gaussian form, has also proved beneficial to ignore. The use of an entropic prior in a Bayesian context has been a smashing success [6, 7]. Even the use of a simple modification to a Gaussian prior on image amplitudes such as a nonnegativity constraint can have a profound beneficial effect on the usefulness of tomographic reconstructions derived from a limited

Email addresses: kmh@lanl.gov, cunning@lanl.gov

number of projections [8]. NonGaussian likelihood functions are also clearly needed when the measurements are subject to a noise process that is not Gaussian. For example, when measurements are based on the number of discrete events, the proper likelihood function is a Poisson distribution.

The disadvantage of giving up the usual simplifying assumptions is that closed-form solutions are usually precluded. Answers often can only be obtained through the use of a computer. While this aspect may be disquieting to some, the power of these advanced models to provide useful results in situations not otherwise approachable has left little doubt that the computer results are worthwhile. Furthermore, the use of a computer actually frees one to employ numerical techniques to deal forthrightly with some messy issues in Bayesian analysis, such as integration of the posterior over unimportant parameters, a process called marginalization. Unfortunately, the new approaches to modeling are typically implemented with only one type of special model in mind and do not promote the use of alternative or compound models or allow more than one type of prior. These capabilities are implied by the Bayesian methodology and are a natural conclusion of hypothesis testing, a key component to model building.

The preceding discussion anticipates our goals in designing the computer application, which we call the Bayes Inference Engine (BIE). The BIE provides a computational approach to Bayesian inference incorporating many new features. The BIE is intended to allow great flexibility in modeling in order to facilitate complex models for the objects under study. It should be easy to develop hierarchical models by simply augmenting an existing model. The BIE allows one to construct models with both linear and nonlinear components with complete flexibility. The BIE will permit the use of a variety of types of distributions for the inherent probability density functions that describe the likelihoods and priors, such as Gaussian, Poisson, Cauchy, entropic, etc. We intend for the BIE to break the restrictive bonds of historic Bayesian analysis.

2. OVERVIEW OF THE BAYESIAN APPROACH

In this paper we present just the basics of the Bayesian approach under the assumption that the reader has some understanding of the field. Those readers who are unfamiliar with Bayesian analysis are urged to learn the fundamentals, presented in textbooks on the subject [9, 10, 11]. The papers by Gull and Skilling and their colleagues encapsulate the essential aspects of practical Bayesian data analysis [6, 7, 12]. Also, one of the authors has written several papers that treat general aspects of the methodology as applied to tomographic image reconstruction and medical image analysis [13, 14].

Bayesian methodology provides a rational and complete approach to inferring models and their parameters from noisy measurements. It transcends *ad hoc* approaches to modeling because it permits the use of quantified prior information concerning the chosen models and ultimately leads to results whose uncertainties can be fully characterized. In Bayesian analysis uncertainties in parameter values are represented by probability distributions on those parameters. A relatively large uncertainty in a parameter is represented by a broad distribution; a precisely-known parameter by a narrow distribution. Probability theory provides a quantitative and consistent basis for Bayesian analysis, which inherits its name from Bayes's fundamental law governing the updating of one probability distribution, called the prior, in the face of new data, called the likelihood, to obtain the resulting probability, called the posterior.

The essential action of Bayes law is captured in the theory of the propagation of experimental errors to which most scientists are exposed early in their careers. When accurate measurements are combined with less accurate ones, coming from prior experiments for example, the uncertainty in the combined result will be significantly reduced compared to that before the acquisition of the new data. By providing a much more thorough description of uncertainty in the form of a precise probability distribution, Bayesian analysis allows one to treat arbitrary probability distributions and address detailed issues, for example, optimal estimators, confidence intervals (or more properly called credible intervals in Bayesian analysis) in the estimates, and so forth.

Bayes law gives the posterior probability $p(\mathbf{x}|\mathbf{d})$ of a particular set of model parameters $\mathbf{x}$, given the observed data $\mathbf{d}$, in terms of the probability of the data given the parameters $p(\mathbf{d}|\mathbf{x})$ and a prior probability of the parameters $p(\mathbf{x})$ as

$$p(\mathbf{x}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{x})p(\mathbf{x}) \ . \tag{1}$$

The probability $p(\mathbf{d}|\mathbf{x})$, usually called the likelihood, comes from a comparison of the actual data to the data expected on the basis of the model of the object and its parameters. The expected data are generated using a model for how the measurements are related to the object, which we call the measurement model. The prior expresses what is known about the object before the measurements are taken and may represent knowledge acquired from previous measurements, specific information regarding the object itself, or simply general knowledge about the parameters, e.g. that they are nonnegative.

The full state of our knowledge about reality is summarized by the posterior probability, or simply, the posterior. We will use the symbol φ for minus the logarithm of the posterior $\varphi = -\log[p(\mathbf{x}|\mathbf{d})]$. Computations with this function are typically easier to do than with the posterior itself since the products of probabilities in Bayes law become sums.

The standard approach to obtaining a representative solution is to find the parameter values that maximize the posterior, or minimize φ, which is called the MAP solution. Although this single solution is often the goal for many investigators, the posterior can be more fully utilized to determine the degree and character of the uncertainty in the solution. We are developing a tool for exploring the posterior to provide an understanding of the degree of uncertainty in Bayesian solutions, which we describe in Sect. 8.

Bayesian analysis also provides the means to properly make subsequent decisions through the use of cost or utility functions, which specify the costs of making correct versus incorrect decisions. Examples of such kinds of decisions include, in the field of nondestructive testing, whether to accept or reject a precision part on the basis of a radiograph, or, in medicine, whether to follow up a positive outcome in a screening test with another test or with surgery.

3. THE BAYES INFERENCE ENGINE

We are developing the Bayes Inference Engine (BIE) to implement the Bayesian methodology on a computer workstation. Our goals for the BIE are that it should be easy to learn and to use and that it should provide a high degree of interactivity with good visualization of the inference process and the models. Additionally, we are building an application that provides the user with a great deal of flexibility in configuring object models and measurement models. We deem these features essential to the usefulness of the BIE.

From the form of Bayes law, we recognize that for a given object model the posterior can be evaluated by combining the likelihood, which requires the data values expected for that object model, and with a numerical value of the prior. This calculation is usually straightforward. It involves calculating the expected measurements for the given object model, which we refer to as the forward measurement calculation.

A typical nonBayesian approach to estimating model parameters from a given set of data is to attempt to apply the inverse of the forward measurement process to the data. Such an approach is fraught with problems when there are insufficient data to uniquely determine all aspects of the actual object. A common remedy for overcoming such problems is to invoke some sort of regularization to permit the inverse solution.

In the BIE we are avoiding the difficulties of direct inversion by basing the estimation on the forward measurement calculation, which evaluates φ. We also make use of the derivatives of φ with respect to the object parameters, which are calculated using the adjoint differentiation technique described in Sect. 5. The parameters for the object model are found using an algorithm to minimize φ with respect to those parameters. The use of priors provides the means of regularization in a probabilistic way that has a quantifiable basis, which may be potentially verified experimentally . We note that this numerical approach has many benefits, which were outlined in the Introduction. Thus with the computer we can obtain accurate Bayesian solutions to fairly complex problems that are intractable using analytic approaches. The computer also allows us to explore complex situations in real time employing data visualization to enhance understanding.

The BIE incorporates many innovative features, including:
1) a graphical programming tool programmed in an object-oriented language, which greatly enhances the flexibility of modeling objects and measurements,
2) adjoint differentiation to calculate the gradient of φ, with respect to all object parameters,
3) new approaches to solving the constrained optimization problem, which is required to find the MAP solution,
4) geometrical representations of physical objects, and
5) a new method to explore the reliability of the Bayesian solution.
We will describe each of these new developments in the following sections.

4. DATA FLOW DIAGRAM

The major mode of interaction with the BIE is through a graphical programming environment. Such an environment provides a very intuitive interface for building models because it mimics a data-flow diagram, with which most scientists and engineers have some experience. It has also been successfully used in a variety of computer applications, such as the Cantata environment found in the image-processing package KHOROS*.

We are programming the BIE using the object-oriented (OO) language Smalltalk in the version supplied by Par-

*Khoral Research, Inc., 6001 Indian School Rd. NE, Suite 200, Albuquerque, NM 87110

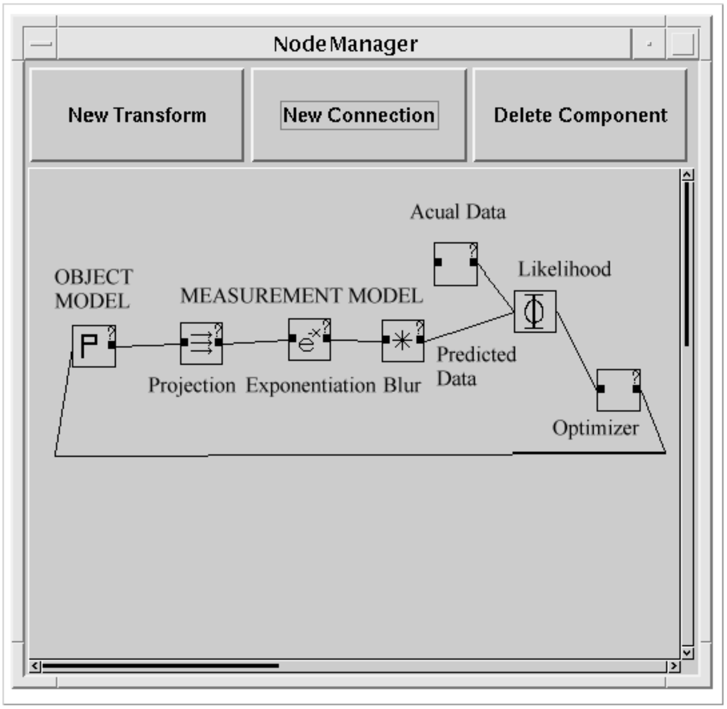

Figure 1: The canvas of the Bayes Inference Engine permits one to specify a data-flow diagram by connecting together Transforms.

cPlace Systems[†], which includes a complete class library for user-interface development. The main interface to the BIE is the graphical programming tool [15]. Figure 1 shows the canvas on which one can create a data-flow diagram. Buttons on the top of the canvas window allow one to add icons to, or delete icons from, the canvas. The square icons represent Transforms, which, together with the Connectors represented by lines drawn between them, describe a data-flow diagram. In this description the capitalized words are classes in the object-oriented language. Transforms act on input Data (actually called PhysicalTensors in the BIE) to calculate output Data. One specifies the flow of data by connecting one Transform to another using a Connector, which is represented by lines drawn between the two Transforms.

The Transforms are living objects with which one can interact. By clicking on the icon representing the Transform with the middle mouse button, a menu pops up that allows one to specify a request of the Transform. One can see a description of a Transform and change the parameters that define it. One can have the Transform display its output data structure. The fact that the Transform objects are alive and always present distinguishes this graphical programming tool from one that allows a user to construct and visualize a script that contains a sequence of actions to be executed (off line) in the prescribed order, as is the case with KHOROS.

Referring to the data-flow diagram in Fig. 1, the Parameters of the object model (the lefthand icon) provide input to the measurement model. The radiographic measurement model shown consists of the next three icons, which sequentially take the projection of the object, exponentiate the result, and perform a convolution with a point-spread function kernel to mimic radiographic blur. The output of the measurement model represent predicted Data, which are fed into a (minus) LogLikelihood function, designated by Φ, along with the actual data, the uppermost icon. A LogPrior, which operates on the model parameters, can also be specified. The output of the LogLikelihood is fed into the Optimizer, the lower righthand icon, whose task it is to find the values of the object-model parameters that result in a minimum value for Φ. One specifies the Parameters of the object model that are to be optimized by connecting their icons to the Optimizer. After optimization, the object model and its Parameter values represent the MAP solution.

An interesting aspect of the OO design of the data-

[†] ParcPlace Systems, Inc., 999 East Arques Ave., Sunnyvale, CA 94086, tel: 408-481-9090

flow diagram is that no supervisor of the sequence of calculations is needed. In Fig. 1 the Optimizer functions by requesting output from its inputs, the LogLikelihood in this case. In a rudimentary implementation, this request is propagated backwards through the Connectors of the data flow diagram to the beginning, namely to the Parameters (the box with a P indicating that they represent parameterizations of models). The forward calculation proceeds by each Transform using its inputs (and internal parameters) to calculate its output. Then the output is transferred upstream to the next transform, which does its calculation. Eventually the request by the Optimizer is answered with the value of φ.

The unnecessary recomputation of data structures is avoided with a slightly more sophisticated design in which each Transform keeps a status variable that indicates whether its output is current and valid. The correct status is maintained by giving each Transform the responsibility of transmitting a message forward through the data-flow diagram when its output status is switching from valid to invalid, in response to any change in its inputs. When a Transform receives such a message from downstream, it treats that message the same as if its input variables had changed. The Optimizer acts as the termination to this process by returning this charge-of-status message.

Another desirable aspect of the data-flow diagram that is easily accomplished is checking between Transforms that are connected together to see whether the data structures are consistent with what the Transforms can operate on.

The OO approach has provided more than just a programming environment; it has aided in the design of the overall application, as well as the numerical algorithms at a fundamental level. For example, the following aspects of the BIE have been elucidated by OO design: the adjoint differentiation technique, the accommodation of constraints in optimization, the automatic connection of the Optimizer to any parameter, and the appropriate role of Connectors and Transforms in the data-flow diagram.

5. ADJOINT DIFFERENTIATION

In our application we need to minimize the scalar function φ by varying the many (10^3 to 10^6 or more) variables that comprise the parameters of the object model. This optimization problem would be intractable without knowing the gradient of φ, or sensitivities, with respect to the many parameters on which it depends. We have uncovered a technique to calculate these crucial sensitivities, called adjoint differentiation [16], that is apparently little known. Using the adjoint differentiation technique the calculation of all these derivatives can be done in a computational time that is comparable to the forward calculation through the data-flow diagram. Our use of objects to represent transformations greatly aids the implementation of this adjoint calculation [17].

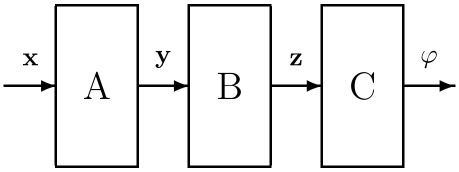

Figure 2: Data flow diagram showing a sequence of transformations, represented by the boxes A, B, and C, starting with the data structure $\mathbf{x}$ and ending with the scalar φ.

We assume that a calculation proceeds as a sequence of transformations such as that shown in Fig. 2. In the context of the BIE, the transformations are implemented by the Transform class. The independent variables in the data structures designated by the vector $\mathbf{x}$ are transformed by block A to produce the dependent variables $\mathbf{y}$. These are transformed by blocks B and C to produce the dependent data structure $\mathbf{z}$ and the final scalar φ, respectively. We call the sequence of transformations,

$$\mathbf{x} \xrightarrow{\text{A}} \mathbf{y} \xrightarrow{\text{B}} \mathbf{z} \xrightarrow{\text{C}} \varphi \, ,$$

the forward calculation and the flow of data from left to right in Fig. 2 is called the forward direction.

We assume that the transformations are general, with the only restriction being that they are differentiable. In accordance with the OO approach, each transformation is self-contained; it requires only its input variables to calculate its output variables, e.g. module B uses only its input $\mathbf{y}$ to calculate its output $\mathbf{z}$. Therefore, each transformation should require nothing more than its input to implement the derivative of its output variables with respect to its input variables.

The data structures are likewise general. They can consist of mixed types of data structures. Some of the data may be parameters that affect the transformations themselves, which could be viewed as separate inputs to the module affected. But there is no loss in generality if they are thought of as being carried along in the sequence of data structures up to the module at which they are used.

Keeping in mind that the data structures might rep-

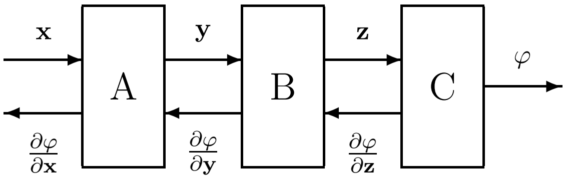
x
y
z
A
B
C
φ
$\frac{\partial\varphi}{\partial\mathbf{x}}$
$\frac{\partial\varphi}{\partial\mathbf{y}}$
$\frac{\partial\varphi}{\partial\mathbf{z}}$

Figure 3: As in Fig. 2 with the data flow for the adjoint derivatives indicated as going from right to left.

resent 2D images as a collection of their pixel values or objects in terms of complex geometric descriptions, we will base the following discussion on the assumption that the dimensionality of the data structures $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ could be very large, perhaps 10^6 or larger. We will further assume that the computing environment is such that the storage of these data structures is feasible. However, storage of the sensitivity matrices of the transformations, e.g. $\frac{\partial y_j}{\partial x_i}$ for all i and j, is likely to be impossible because that would require storing 10^{12} variables if $\mathbf{x}$ and $\mathbf{y}$ each possess 10^6 variables.

The chain rule allows us to calculate the derivatives of φ with respect to the ith component of $\mathbf{x}$,

$$\frac{\partial\varphi}{\partial x_i} = \sum_{jk} \frac{\partial\varphi}{\partial z_k}\frac{\partial z_k}{\partial y_j}\frac{\partial y_j}{\partial x_i} \ . \tag{2}$$

Even if the transformations are nonlinear, this expression amounts to a product of matrices, each element of which specifies the differential response of an output variable with respect to a differential change of an input variable.

The order of the summations can obviously be done in two different ways. If the sum over j is done before the sum over k, the calculation proceeds in the same direction as the forward model calculation. The sequence is:

$$\mathbf{I} \xrightarrow{\mathrm{A}'} \frac{\partial\mathbf{y}}{\partial\mathbf{x}} \xrightarrow{\mathrm{B}'} \frac{\partial\mathbf{z}}{\partial\mathbf{x}} \xrightarrow{\mathrm{C}'} \frac{\partial\varphi}{\partial\mathbf{x}} ,$$

where $\frac{\partial\mathbf{z}}{\partial\mathbf{x}}$ is shorthand for the matrix whose elements are $\frac{\partial z_j}{\partial x_i}$ and transformation B' propagates the derivative of the transformation B, i.e. by multiplication of the sensitivity matrix $\frac{\partial\mathbf{y}}{\partial\mathbf{x}}$ by the matrix $\frac{\partial\mathbf{z}}{\partial\mathbf{y}}$ to produce the sensitivity matrix $\frac{\partial\mathbf{z}}{\partial\mathbf{x}}$. The symbol $\mathbf{I}$ at the beginning of the sequence represents the identity structure, indicating that the sensitivity calculation begins with the first transformation A'. As the dimensions of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are assumed to be large, we see that this sequence results in very large intermediate matrices, which we would like to avoid.

On the other hand, if the sum on k is done before that on j, the sequence of calculations is

$$\mathbf{I} \xrightarrow{\mathrm{C}'^\dagger} \frac{\partial\varphi}{\partial\mathbf{z}}^\dagger \xrightarrow{\mathrm{B}'^\dagger} \frac{\partial\varphi}{\partial\mathbf{y}}^\dagger \xrightarrow{\mathrm{A}'^\dagger} \frac{\partial\varphi}{\partial\mathbf{x}}^\dagger ,$$

where, for example, $\mathrm{B}'^\dagger$ implements the adjoint of the matrix $\frac{\partial\mathbf{z}}{\partial\mathbf{y}}$. This sequence implies intermediate data structures (e.g. $\frac{\partial\varphi}{\partial\mathbf{y}}$) that mimic the normal data structures (e.g. $\mathbf{y}$). Thus the requirement for storing these data structures is merely double that required to store the structures for the forward calculation, which may be required for the sensitivity calculation if the transformations are nonlinear. This approach is called the adjoint differentiation technique.

The adjoint differentiation calculation is straightforward to program. Our experience in explaining this approach to others indicates that it is worth emphasizing that adjoint differentiation is not that same as calculating the inverse of the forward transformation, which might be an ill-posed problem. For example, if the transformation B is linear, the forward calculation amounts to multiplication by a matrix $\mathbf{B}$. The adjoint differentiation calculation would correspond to multiplication by the matrix $\mathbf{B}^\dagger$ (the same as the transpose of $\mathbf{B}$ when the matrix is real), which is quite different than multiplication by $\mathbf{B}^{-1}$.

The backward flow of the adjoint derivatives is depicted in Fig. 1. We emphasize that in the OO approach we are using, each Transform has the responsibility to propagate the adjoint derivative from its output side to its input side. The Transform "knows how" to do this because it knows how to accomplish the forward calculation. In reality, what this means is that when a new Transform is created by a programmer, the code for the adjoint derivative should be developed using the logic of the forward calculation to determine the derivatives of the output variables with respect to the input variables. We stress that this derivative matrix need not be explicitly calculated and stored. The effect of multiplying the adjoint gradient of φ with respect to the Transform output variables by the adjoint of this derivative matrix can often be achieved using computer code that is very similar to the code for the forward calculation.

A simple example of a nonlinear transformation is the exponentiation of each input element, which is one of the icons shown in Fig. 1. The output data elements are given by $y_j = \delta_{ij}\exp(x_i)$, where x_i are the input data elements and δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ when $i = j$ and 0 otherwise). Then the derivative is given by $\frac{\partial\varphi}{\partial x_i} = \sum_j \frac{\partial\varphi}{\partial y_j}\frac{\partial y_j}{\partial x_i} = \exp(x_i)\frac{\partial\varphi}{\partial y_i} = y_i\frac{\partial\varphi}{\partial y_i}$. Implementation of this expression in the adjoint context is clearly straightforward.

This example points out a few aspects of the calculation of the adjoint derivative. If, in the forward calculation, each output variable depends on a relatively limited number of input variables (one in this case), then the adjoint derivative with respect to an output variable will only contribute to the adjoint derivative with respect to those same relatively few input variables. Provided the logic of the forward calculation is not too intricate, the adjoint derivative calculation should involve an amount of computation comparable to the forward computation. A second point is that the adjoint calculation may need to have access to the data structures from the forward calculation. In this case, the value of either x_i or y_i is needed to compute $\frac{\partial \varphi}{\partial x_i}$ from $\frac{\partial \varphi}{\partial y_i}$. The implication is that the forward data structures may need to be stored for the time between which they are calculated and the time they are needed for the adjoint derivative calculation.

To further elucidate the adjoint differentiation technique, consider a sequence of linear transformations, $\mathbf{x} \xrightarrow{\mathrm{A}} \mathbf{y} \xrightarrow{\mathrm{B}} \mathbf{z}$, followed by the evaluation of the norm squared. Expressed in equations: $\mathbf{y} = \mathbf{A}\mathbf{x}$, $\mathbf{z} = \mathbf{B}\mathbf{y}$, $\varphi = \mathbf{z}^{\mathrm{T}}\mathbf{z}$, where $\mathbf{A}$ and $\mathbf{B}$ are matrices. The overall transformation is $\varphi = |\mathbf{B}\mathbf{A}\mathbf{x}|^2$. Then the forward calculation of the sensitivities amounts to evaluating $\frac{\partial \varphi}{\partial \mathbf{x}} = 2\mathbf{z}^\dagger\mathbf{B}\mathbf{A}$. The reverse (adjoint) calculation is $\frac{\partial \varphi}{\partial \mathbf{x}}^\dagger = \mathbf{A}^\dagger\mathbf{B}^\dagger(2\mathbf{z})$, where $\mathbf{A}^\dagger$ denotes the adjoint (or transpose) of matrix $\mathbf{A}$. It is clear that these two results are identical, the second equation being just the adjoint of the first. For linear transformations, the forward data structures need not be stored for the purpose of calculating the adjoint derivative.

Examples of linear transformations in image processing include convolution of an image by a fixed kernel (point-spread function) and the calculation of the projection (line integrals along along a set of parallel lines) of an image. The adjoint derivatives for these two examples are calculated by convolution with the adjoint of the kernel in the first case and backprojection, i.e. streaking the data from the projection domain back across the image along the lines of integration, in the second.

6. OPTIMIZATION

The MAP solution is found in the BIE by minimizing φ with respect to all the model parameters. Given the possibly large number of parameters, it is imperative to use the derivatives of φ with respect to all parameters, called the gradient. Fortunately, there is a technique to efficiently calculate them as described in the previous section. OO programming imposes certain conditions on how optimization can be implemented. We have found that these conditions may actually be strengths. For example, many of the models in the BIE impose constraints on parameters. Some constraints involve fixed limits on individual parameters, e.g. nonnegativity. Constraints can also exist between parameters. The approach to optimization should include these in an OO way. That is, the Optimizer should only request that the Parameters act on themselves. Examples of possible actions include a) add a specified vector to the present values of the parameters and b) satisfy constraints on the parameters.

In our present algorithm the Optimizer roughly follows these steps: 1) requests the value of φ from the last Transform, 2) requests the gradient from the Parameters it is attached to, 3) instructs the Parameters to increment themselves by a certain amount, 4) asks Parameters to apply constraints, 5) asks for φ from last Transform in the flow diagram and 6) repeats steps 3) to 5) to find the minimum in φ along the search direction. Then the Optimizer returns to step 2) to calculate a new search direction and continues until convergence to a minimum. The increment taken in the parameters in step 3) could be a parameter vector proportional to the gradient, but in the opposite direction (steepest descent), or might be determined by using the conjugate-gradient method. A major point to make is that the Optimizer functions in an abstract variable space. It does not really know what kind of Parameters are connected to it. It can only ask the Parameters to perform a limited set of tasks.

The general method that we employ to guarantee that the constraints on the parameters are met is by projection onto convex sets (POCS) [18]. Each Parameter checks whether constraints are violated. If they are, the Parameters are minimally changed to meet the constraint [19].

7. GEOMETRIC REPRESENTATION OF OBJECTS

Deformable models have been developed in a number of fields to describe objects geometrically, particularly in computer vision where the aim is to describe a scene (image) in terms of geometrical objects [2, 4, 20]. This approach is also being used in medical imaging to warp a template of the "average" brain to match a particular patient's actual brain [3].

We are pioneering the use of geometric models to improve tomographic reconstruction [5, 21]. This tack is quite different from the normal one of representing a 2D object in terms of its density, typically described by square pixels on an ordered grid. The use of a geometrical description recognizes the very important role that boundaries play in characterizing objects. The reconstruction process amounts to deforming an initial object geometry in a minimal way to match the data. In the Bayesian approach, one controls the geometric deformation by placing a prior on it. The net effect is to add to φ a deformation energy that penalizes greater deformations. This approach has proven to be a valuable means to achieve good reconstructions in situations where all other methods fail, for example when only two radiographs are available [21]. However, it must be emphasized that this approach can only be successful when the objects being reconstructed have a fairly simple morphology that is approximately known beforehand. Please refer to the above citations for more details and examples.

8. RELIABILITY EXPLORATION

A central aspect of the Bayesian approach is the characterization of the degree of certainty in the models given by the posterior. Bayesian modeling has been used to solve a variety of difficult imaging problems. However, surprisingly little has been done to make use of the full posterior of the models used to interpret data. Part of the reason reliability has not been approached by many lies in the difficulty of doing so, particularly in a large dimensional space.

We know of two ways to visualize the reliability of inferred models. The first, proposed by Skilling et al. [22], provides a stochastic look at the range of possible solutions. It involves the display of a sequence of solutions that are randomly chosen from the posterior probability distribution. This sequence, typically calculated off line, is presented as a video loop. By showing a representative range of alternative solutions, the degree of variability of this presentation provides the viewer with a visual impression of the degree of uncertainty in the inferred model.

Our new approach [23, 24] draws on an analogy between φ and a physical potential; then the gradient of φ is analogous to a force. An unconstrained MAP solution can be interpreted as the situation in which the forces on all the variables in the problem balance so that the net force on each variable is zero. Further, when a variable is perturbed from the MAP solution, the derivative of φ with respect to that variable is the force that drives it back towards the MAP solution. The phrase “force of the data” takes on real meaning in this context.

We exploit this physical analogy to facilitate the exploration of the reliability of a particular feature of a MAP solution, which the user specifies by directly interacting with the solution presented by the BIE. The uncertainty in the solution is explored by applying a constant force to the selected combination of parameters that characterize the feature of interest. All parameters are readjusted to minimize φ. The uncertainty in the parameters is indicated by the amount that they move away from their MAP values for a given applied external force. The correlations between parameters experiencing the external force and the others is demonstrated by how much and in what direction the parameters change. We have shown that this approach leads to a quantitative estimate for an appropriate part of the covariance matrix for problems in which the parameters are unconstrained [23, 24]. Ideally, these correlations could be seen through direct interaction with a rapidly-responding dynamical Bayesian system. Alternatively, they may be demonstrated as a video loop of results produced off line.

9. FUTURE DIRECTIONS

In the BIE we have an ideal tool with which to efficiently deal with Bayesian inference. It provides a framework that is well-suited to many desirable extensions, some of which we mention in this section.

The full and simple-to-use interactivity of the BIE provides the analyst with the capability to fully diagnose the models he creates. Feedback about what is needed from the model to match the data is provided by displaying the adjoint derivative, which shows the force of the data. The full interactivity with the object models makes it easy to augment the models to achieve a better fit to the data.

We intend to implement soon a means to generate random samples from the posterior [25]. This capability could be used to estimate the posterior mean (as an alternative to the posterior mode) and variance, which is one way to summarize the uncertainty in solutions. This technique permits marginalization with respect to any parameters that are not of interest. It also can provide a visualization of the uncertainty in solutions by displaying as a video loop the sequence of random samples [22].

Most of the models presently available in the BIE are two dimensional. We expect to develop 1D models soon. One-dimensional models will provide a very useful environment in which to develop demonstrations of how the Bayesian approach addresses many familiar problems such as deconvolution, or restoration, of blurred 1D signals, spectral estimation, interpolation, and line fitting. The advantage of 1D problems is that the calculations can be done very quickly, thereby providing excellent response time. This would be an ideal way to demonstrate the usefulness of the concepts put forward in Sect. 8. We also expect to develop three-dimensional modeling capabilities before long so that we can address the problem of tomographic reconstruction of 3D objects.

An interesting variation of the technique proposed in Sect. 8 to explore the reliability in Bayesian solutions presented is the possibility of decomposing the forces into various components. For example, the force derived from all the data (through the likelihood) may be compared to the force derived from the prior. This capability can provide the analyst with a better idea of how much the prior is affecting any aspect of the reconstruction. Other decompositions of the overall force on the solution are possible: it might be useful to see how much certain measurements contribute to the uncertainty of the solution.

We anticipate that in the future it may be possible to use the tools of virtual reality, coupled to supercomputation, to explore the reliability of a Bayesian solution through direct manipulation of the computer model. Force feedback would permit one to actually feel the stiffness of a model. Higher dimensional correlations might be felt through one's various senses.

In addition to estimation of the uncertainties in parameter values, the Bayesian approach also provides a methodology for inference from measurements about the choice of models appropriate to describe reality. Probabilities can ultimately be employed to compare and decide amongst different models. Our preference for simpler models over more complex ones can be incorporated through a prior on model complexity [12]. An often-used example of where model selection plays a role is in the choice of the number of terms to allow in a series expansion that is meant to describe the behavior of some data. A more interesting use of model selection is in the context of an object defined by its boundary, which is smooth by default. The boundary might be allowed to develop a kink, i.e. an abrupt change in slope, thereby negating the smoothness constraint at a particular place, if the data provide enough evidence for such a departure from the default model [26].

10. ACKNOWLEDGEMENTS

This work was supported by the United States Department of Energy under contract number W-7405-ENG-36. We would like to acknowledge the helpful conversations we have had with the following people: John Skilling, Stephen F. Gull, Kyle J. Myers, and Robert F. Wagner.

11. REFERENCES

[1] K. M. Hanson, "A Bayesian approach to nonlinear inversion: Abel inversion from x-ray data," in Transport Theory, Invariant Imbedding, and Integral Equations, *Lect. Notes in Pure and Appl. Math.* **115**, P. Nelson, V. Fabor, D. L. Seth, and A. B. White, Jr., eds., pp. 363–368, Marcel Dekker, New York, 1989.

[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Inter. J. Comp. Vision*, **1**, pp. 321–331, 1988.

[3] R. Bajcsy and S. Kovačič, "Multiresolution elastic matching," *Comput. Vision*, **46**, pp. 1–21, 1989.

[4] U. Grenander and M. Miller, "Representations of knowledge in complex systems," *Jour. Roy. Stat. Soc. B*, **56**, pp. 549–603, 1994.

[5] K. M. Hanson, "Bayesian reconstruction based on flexible prior models," *J. Opt. Soc. Amer. A*, **10**, pp. 997–1004, 1993.

[6] J. Skilling and R. K. Bryan, "Maximum entropy image reconstruction: general algorithm," *Mon. Not. R. Ast. Soc.*, **211**, pp. 111–124, 1984.

[7] S. F. Gull, "Developments in maximum-entropy data analysis," in *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., pp. 53–71, Kluwer Academic, Dordrecht, 1989.

[8] K. M. Hanson, "Method to evaluate image-recovery algorithms based on task performance," *J. Opt. Soc. Amer. A*, **7**, pp. 45–57, 1990.

[9] G. E. P. Box and G. C. Tiao, *Bayesian Inference in Statistical Analysis*, John Wiley & Sons, New York, 1973.

[10] S. J. Press, *Bayesian Statistics: Principles, Models, and Applications*, Wiley, New York, 1989.

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

[12] S. F. Gull, "Bayesian inductive inference and maximum entropy," in *Maximum Entropy and Bayesian Methods in Science and Engineering (Vol. 1)*, G. J. Erickson and C. R. Smith, eds., pp. 53–74, Kluwer Academic, Dordrecht, 1989.

[13] K. M. Hanson, “Bayesian and related methods in image reconstruction from incomplete data,” in *Image Recovery: Theory and Application*, H. Stark, ed., pp. 79–125, Academic, 1987.

[14] K. M. Hanson, “Introduction to Bayesian image analysis,” in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **1898**, pp. 716–731, 1993.

[15] G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, “An object-oriented implementation of a graphical-programming system,” in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **2167**, pp. 914–923, 1994.

[16] J. Skilling, “Automatic differentiation from an oceanographer’s perspective,” in *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, A. Griewank and G. F. Corliss, eds., pp. 191–201, SIAM, Philadelphia, 1991.

[17] G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, “An interactive tool for Bayesian inference,” in *Review of Progress in Quantitative Nondestructive Evaluation*, D. O. Thompson and D. E. Chimenti, eds., vol. 14A, pp. 747–754, Plenum, New York, 1995.

[18] D. C. Youla, “Mathematical theory of image restoration by the method of convex projections,” in *Image Recovery: Theory and Application*, H. Stark, ed., pp. 29–77, Academic, 1987.

[19] G. S. Cunningham, K. M. Hanson, G. R. Jennings, Jr., and D. R. Wolf, “An object-oriented optimization system,” in *Proc. IEEE Int. Conf. Image Processing, vol. III*, pp. 826–830, IEEE, 1994.

[20] R. Szeliski, “Probabilistic modeling of surfaces,” *Proc. SPIE*, **1570**, pp. 154–165, 1991.

[21] K. M. Hanson, G. S. Cunningham, G. R. Jennings, Jr., and D. R. Wolf, “Tomographic reconstruction based on flexible geometric models,” in *Proc. IEEE Int. Conf. Image Processing, vol. II*, pp. 145–147, IEEE, 1994.

[22] J. Skilling, D. R. T. Robinson, and S. F. Gull, “Probabilistic displays,” in *Maximum Entropy and Bayesian Methods*, W. T. Grandy, Jr. and L. H. Shick, eds., pp. 365–368, Kluwer Academic, Dordrecht, 1991.

[23] K. M. Hanson and G. S. Cunningham, “The hard truth,” in *Maximum Entropy and Bayesian Methods*, J. Skilling, ed., Kluwer Academic, Dordrecht, 1994 (to be published).

[24] K. M. Hanson and G. S. Cunningham, “Exploring the reliability of Bayesian reconstructions,” in *Image Processing*, M. H. Loew, ed., *Proc. SPIE*, **2434**, pp. 416–423, 1995.

[25] J. Besag, P. Green, D. Higdon, and K. Mengersen, “Bayesian computation and stochastic systems,” *Stat. Sci*, **10**, pp. 3–66, 1995.

[26] K. M. Hanson, R. L. Bilisoly, and G. S. Cunningham, “Kinky tomographic reconstruction,” to be published in *Image Processing*, M. H. Loew and K. M. Hanson, eds., *Proc. SPIE*, **2710**, 1996.