In *Maximum Entropy and Bayesian Methods*,
G. J. Erickson, J. T. Rychert, and C. R. Smith, eds., pp. 121-135,
Kluwer Academic, Dordrecht, 1998.

INVERSION BASED ON COMPUTATIONAL SIMULATIONS

K.M. Hanson, G.S. Cunningham, and S. S. Saquib*
Los Alamos National Laboratory, MS P940
Los Alamos, New Mexico 87545 USA

Abstract. A standard approach to solving inversion problems that involve many parameters uses gradient-based optimization to find the parameters that best match the data. We will discuss enabling techniques that facilitate application of this approach to large-scale computational simulations, which are the only way to investigate many complex physical phenomena. Such simulations may not seem to lend themselves to calculation of the gradient with respect to numerous parameters. However, adjoint differentiation allows one to efficiently compute the gradient of an objective function with respect to all the variables of a simulation. When combined with advanced gradient-based optimization algorithms, adjoint differentiation permits one to solve very large problems of optimization or parameter estimation. These techniques will be illustrated through the simulation of the time-dependent diffusion of infrared light through tissue, which has been used to perform optical tomography. The techniques discussed have a wide range of applicability to modeling including the optimization of models to achieve a desired design goal.

Key words: simulation, inversion, reconstruction, adjoint differentiation, sensitivity analysis, optimization, model validation

1. INTRODUCTION TO THE GENERAL PROBLEM

Frequently a physical situation can only be described fully by a computational model. We wish to address the general problem of finding the values of the parameters in such a model that best match a given set of data. This problem in often referred to as that of inversion. In data matching the objective function to be minimized is often the negative logarithm of the likelihood of the data given their predicted values, which yields the maximum likelihood (ML) solution. Alternative approaches include regularized versions of maximum likelihood and Bayesian methods, in which the objective function is the minus-log-posterior, yielding the maximum a posteriori (MAP) estimate.

We confine ourselves to objective functions that depend on the parameters in a continuous and differentiable fashion. We do not necessarily avoid problems for which the objective function possesses multiple minima. However, because the techniques that we present make use of gradients in the optimization process, they will work effectively only when one can easily find the basin of attraction for the global minimum, for example, by multiscale or multiresolution optimization.

*Email: `kmh@lanl.gov, cunning@lanl.gov, saquib@polaroid.com`
WWW: `http://www.lanl.gov/home/kmh/`
Saquib's current address: Polaroid Corporation, 750 Main St., Cambridge, MA 02139.

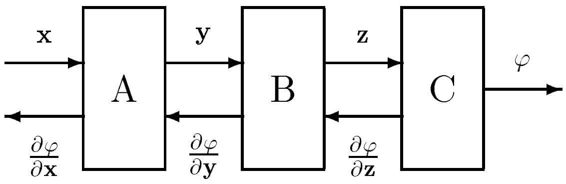


Figure 1: Data flow diagram showing a sequence of transformations, represented by the boxes A, B, and C connected by the arrows pointing to the right, starting with the data structure $\mathbf{x}$ and ending with the scalar φ. The data flow for the adjoint derivatives is indicated by the arrows pointed left.

The proposed method for solving the inverse problem is applicable to a wide variety of problems in which the measurements for the process in question are adequately described by a predictive forward computational model. We demonstrate the method in the context of time-dependent diffusion. We demonstrate the feasibility of reconstructing the diffusion constants within a region from time-dependent light intensities measured for infrared light that diffuses through the region [1]. Adjoint differentiation is already being used in ocean modeling [2, 3]. We believe it may be useful in dealing with a host of other advanced modeling situations, including geophysical interpretation of seismic data and modeling of the atmosphere, fluid flow, and shock-wave phenomena. It should be applicable to optimization of engineering designs in complex situations, such as streamlining of airplane foils and automobile bodies to reduce drag.

A number of other enabling techniques will be mentioned, including multiscale analysis to constrain and accelerate the optimization process, deformable geometric models for describing the boundaries of objects, and the Markov Chain Monte Carlo method of sampling the uncertainty distribution of the estimated parameters.

2. ADJOINT DIFFERENTIATION

We wish to address problems that require minimizing a 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 parameters on which it depends. There exists a technique to calculate these crucial sensitivities, called adjoint differentiation [4], that is apparently relatively unappreciated. Using the adjoint differentiation technique, the calculation of all these derivatives can be done in a computational time comparable to the forward calculation.

Suppose that a calculation proceeds as a sequence of transformations as shown in Fig. 1. The independent variables in the data structures designated by the vector $\mathbf{x}$ are transformed in block A to produce the dependent variables $\mathbf{y}$. These are transformed in 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. We assume that the transformations are general, with the only restriction being that they are differentiable. 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.

The derivatives of φ with respect to the ith component of $\mathbf{x}$ are obtained using the chain rule,

$$\frac{\partial \varphi}{\partial x_i} = \sum_{jk} \frac{\partial \varphi}{\partial z_k} \frac{\partial z_k}{\partial y_j} \frac{\partial y_j}{\partial x_i}\,. \tag{1}$$

Even if the transformations are nonlinear, this expression amounts to a product of matrices. 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. As the dimensions of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are assumed to be large, 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

$$\mathbf{I} \xrightarrow{\text{C}'^{\text{T}}} \frac{\partial \varphi}{\partial \mathbf{z}}^{\text{T}} \xrightarrow{\text{B}'^{\text{T}}} \frac{\partial \varphi}{\partial \mathbf{y}}^{\text{T}} \xrightarrow{\text{A}'^{\text{T}}} \frac{\partial \varphi}{\partial \mathbf{x}}^{\text{T}}\,,$$

where, for example, B'^{T} effectively multiplies $\frac{\partial \varphi}{\partial \mathbf{z}}$ by the transpose of the matrix $\frac{\partial \mathbf{z}}{\partial \mathbf{y}}$. The symbol $\mathbf{I}$ at the beginning of the sequence represents the identity structure, indicating that the sensitivity calculation begins with the transpose of the derivative of transformation C, that is, C'^{T}. This sequence implies intermediate data structures (e.g., $\frac{\partial \varphi}{\partial \mathbf{y}}$) that mimic the normal data structures (e.g., $\mathbf{y}$) implying storage requirements identical to the forward calculation. If the forward transformations are nonlinear, the forward data may be required for the adjoint calculation. The backward flow of the adjoint derivatives is depicted in Fig. 1.

The significant conclusion is that with adjoint differentiation the derivatives of a scalar quantity, φ in the above example, with respect to all the variables in the model can be computed in a time comparable to the forward calculation.

We have coined the acronym Adjoint Differentiation In Code Technique (ADICT) [5] to describe a particular approach to adjoint differentiation. The distinctive feature of ADICT is that a separate computer code is used to compute the adjoint derivatives and that code is based on the simulation code with the explicit intent to "differentiate" the forward calculation or numerical algorithm. For optimization of a functional based on computation, it is desirable to have the gradient of the computation, not of the physics equations that the computation is supposed to approximate [6, 7]. The ADICT principle can produce more accurate derivatives [8].

For the Bayes Inference Engine (BIE) [5], we manually coded the adjoint code for each transformation based on its forward code, or the specific algorithm it employs. This task can sometimes be daunting, but one that can be learned. For example, such manual coding was employed to obtain the adjoint differentiation of a transformation from a surface, described in terms of triangles, to a 3D voxelated grid, which involves some fairly complex calculations [9]. In some cases, it may be helpful to make use of known derivatives of functions of matrices [10]. Object-oriented design and programming is tremendously advantageous for linking calculations together to form a data-flow diagram made of autonomous transformations, as in the BIE.

There are several compilers of FORTRAN code that "automatically" produce differentiation code, including the well-known ADIFOR [11] and GRESS [12]. These compilers are based on computing the derivatives of the result of each line of the simulation code during the forward calculation. In ADIFOR these derivatives are propagated along with the forward calculation, which limits the number of parameters for which it can compute derivatives in large problems. On the other hand, GRESS stores these line-by-line derivatives for a subsequent adjoint calculation, which makes it more useful for computing derivatives with respect to all the parameters. However, this approach imposes heavy memory requirements that make it unsuitable for lengthy simulations.

More promising for application to large simulation codes is a code-based approach created by Ralf Giering [13]. His compiler, dubbed TAMC for Tangent-linear and Adjoint Model Compiler, effectively implements the underlying concepts of ADICT to convert an input FORTRAN code into an enhanced forward code and adjoint code. These codes are to be run consecutively to obtain the result of the forward calculation and the derivatives with respect to the desired variables. Its significant difference from GRESS is that the adjoint code effectively reverses the forward calculations accumulating the derivatives as it goes, instead of calculating and storing the same derivatives during the forward calculation, as GRESS does. This approach allows TAMC to avoid storing a vast number of derivatives and thus opens up the possibility of handling really huge problems. Another issue that TAMC handles is the storage of the state of the forward calculation, which is typically needed for the adjoint derivative calculation. To keep these storage requirements from becoming overwhelming, TAMC allows one to insert check points in the forward calculation at which the state of calculation is stored. Then, during the adjoint calculation, the forward simulation is automatically redone to reestablish the intervening parameters that were not stored.

TAMC has successfully been used to generate adjoint differentiation code for models that simulate ocean dynamics [2, 3], which is a really large problem! We have also had success applying TAMC to 1D hydrodynamics codes [8].

3. OPTIMIZATION

The ML or MAP solution is found by minimizing a scalar functional 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. Fortunately, there is a technique to efficiently calculate these gradients as described in the previous section. The best-known approaches to gradient-based optimization of functions of many parameters are steepest descent and conjugate gradient. Another set of techniques that make use of the gradient are referred to as quasi-Newton methods. The general idea is to build up an approximate expression for the Hessian (the second-derivative matrix of φ with respect to all the variables), or the inverse Hessian. Davidon pioneered the quasi-Newton movement with what he called the variable metric method [14]. We have had good success using the Davidon algorithm, although at times the conjugate-gradient algorithm converges better.

The BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm was an outgrowth of the Davidon algorithm, which had become known as DFP (Davidon-Fletcher-Powell). The BFGS algorithm has the advantage over the DFP in that it does not require accurate line minimizations along the quasi-Newton directions to build up the approximate Hessian [15]. Thus, BFGS potentially reduces the number of function evaluations required to complete an optimization procedure. In the kth iteration of the optimization procedure, the change in the parameter vector from its present value x_k is based on the gradient $g_k = \frac{\partial \varphi}{\partial x}|_{x_k}$ and the present estimate of the inverse Hessian[1] C_k. The position of the minimum is estimated by the Newton formula: $x^* = x_k - \alpha C_k g_k$. Starting with $\alpha = 1$, a line search is conducted to find the value of α that minimimizes $\varphi(x^*)$, which yields the new estimate $x_{k+1} = x^*$. Then, designating the change in position by $s_k = x_{k+1} - x_k$ and the corresponding change in the gradient by $y_k = g_{k+1} - g_k$, the BFGS update formula for the inverse Hessian is

$$C_{k+1} = V_k^{\mathrm{T}} C_k V_k + c_k \, s_k s_k^{\mathrm{T}} \ , \qquad (2)$$

where $V_k = I - c_k \, y_k s_k^{\mathrm{T}}$ and $c_k = (s_k^{\mathrm{T}} y_k)^{-1}$.

Important to note for applications involving many variables is that the inverse Hessian does not have to be stored as a matrix. Rather, whenever the product of the inverse Hessian with a vector is desired, for example to estimate the next step, the expansion (2) may be used. Therefore, only the parameter vector and the gradient at each iteration of the optimization need to be stored.

More recently, there has appeared a limited-memory version of BFGS [16], as well as the truncated Newton [17] algorithm, both of which have the advantage that they do not require keeping the full sequence of optimization iterations. They both seem to have strengths for different kinds of problems [18]. Although we have not yet implemented either of these in the BIE, we think that the limited-memory BFGS algorithm will be very useful for large optimization problems.

[1] We avoid the prevalent confusing designation of the inverse Hessian by H.

4. DEMONSTRATION: TIME-DEPENDENT DIFFUSION

As an example of the success of the ADICT approach, we summarize our previous paper [1], in which we investigated the diffusion of infrared light through tissue. We solved the problem of inversion of time-resolved data to obtain the distribution of diffusion coefficients through which the light passed. Suppose that the intensity of diffused light at position (x, y) and time t is denoted by $U(x, y, t)$ and the source strength by $R(x, y, t)$. Then the time evolution of U through a region described by diffusion coefficients $D(x, y)$ and absorption coefficients $\mu_a(x, y)$ is given by the diffusion equation

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial U}{\partial x}\right) + \frac{\partial}{\partial y}\left(D\frac{\partial U}{\partial y}\right) - c\mu_a U + R \,, \tag{3}$$

where the spatial and temporal dependence of the parameters has been suppressed. The quantity c is the speed of light.

We approach the computational problem in terms of discrete samples of U on a spatial grid with locations specified as a subscript s and in time with a superscript n. When the spatial position subscript is dropped, the resulting quantity is a column vector obtained by either row-ordering or column-ordering the corresponding two-dimensional field (e.g., U^n).

For simplicity we assume that the measurements are degraded by additive uncorrelated Gaussian noise. The minus-log-likelihood of the observations Y, given D and μ_a, is

$$\varphi = -\log P(Y|D, \mu_a) = \tfrac{1}{2}\chi^2 = \sum_{s,n} \frac{(Y_s^n - \tilde{U}_s^n)^2}{2\sigma_{s,n}^2} \,, \tag{4}$$

where $\sigma_{s,n}^2$ is the noise variance at spatial position s and time n and the tilde on the U indicates only those positions at which the light intensities are measured. The general strategy for estimating the parameters, D and μ_a is to minimize $\frac{1}{2}\chi^2$, or if prior information is to be incorporated by means of a Bayesian formulation, the minus-log-posterior, which is $\frac{1}{2}\chi^2$ plus the minus-log-prior.

To compute (4), given D and μ_a, we need to solve the diffusion equation (3) forward in time to obtain the diffuse intensity U_s^n for all time n and spatial positions s. We will briefly summarize the approach to this calculation employed in Ref. [1], which should be consulted for the details.

4.1. Solving the Forward Problem

The general approach taken to solve this forward problem is to use the finite-difference method in which the spatial and temporal derivatives in Eq. (3) are replaced by their finite-difference approximations. This substitution results in a difference equation that needs to be solved forward in time. When solving the difference equation for U^{n+1}, the finite-difference approximations to the spatial derivatives can be evaluated either at time index $n + 1$ or n. In the implicit method[2] for solving differential equations, the spatial derivatives are evaluated at

[2] The implicit method is presented here for clarity, although a slightly different method, the Alternating-Directions Implicit (ADI) method, was used in Ref. [1] for calculational efficiency.

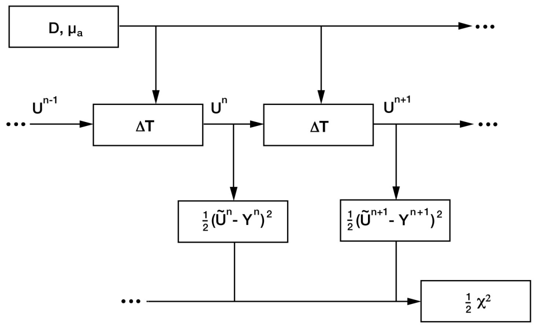


Figure 2: Data flow diagram for the forward calculation of the diffusion problem. The boxes labeled ΔT compute the update to the system for each time step based on the spatial distribution of D and μ_a provided by the topmost box. The predicted intensities are compared to the corresponding measurements Y to obtain $\varphi = \frac{1}{2}\chi^2$. In the adjoint differentiation calculation, the data flow is reversed.

the time instance $(n+1)$ when computing the diffuse intensity U^{n+1}. The implicit method is unconditionally stable for any value of Δt.

Substituting in Eq. (3), the equation to be solved to obtain U^{n+1} from U^n is, in vector notation,

$$A\,U^{n+1} = U^n + \bar{R}^{n+1/2} \;, \tag{5}$$

where A is a sparse matrix (because derivatives involve only local variables) whose elements depend on the D and μ_a values. $\bar{R}^{n+1/2}$ denotes the integrated source strength between time instances n and $n+1$.

The procedure for calculating the time-evolution of U is depicted in Fig. 2. The transformations denoted by ΔT essentially involve solving Eq. (5) to move forward by one time step. These calculations depend on the spatial distribution of the diffusion constant D and absorption coefficient μ_a. The minus-log-likelihood $(\frac{1}{2}\chi^2)$ is the accumulation of the sum of the squares of the differences between the measurements Y_s^n and their predicted values $\tilde{U}_s^n$. Thus it gets a contribution from each measurement time. The assumed time-independent distribution of D and μ_a is used in each time step calculation.

4.2. Gradient Computation

We designate the unknown parameters by the column vector $\theta = [D\ \mu_a]^T$. We need the derivative or sensitivity of $\varphi(\theta)$ with respect to θ to facilitate the solution of the inverse problem. The ADICT principle requires us to work backwards in time differentiating the same discretized equations that are used to compute the forward solution. We present a brief outline of the approach.

The sensitivity of φ with respect to U^n is obtained by using the sensitivity of

φ with respect to U^{n+1}. Application of the chain rule yields

$$\frac{d\varphi}{dU^n} = \left[\frac{dU^{n+1}}{dU^n}\right]^T \frac{d\varphi}{dU^{n+1}} + \frac{\partial\varphi}{\partial U^n} , \tag{6}$$

where $\frac{\partial\varphi}{\partial U^n}$ denotes the change in φ when only U^n is varied, keeping all other variables constant, while $\frac{d\varphi}{dU^n}$ denotes the total change in φ when U^n is varied along with all variables that depend on U^n. Differentiating Eq. (4) with respect to U_s^n, we obtain

$$\frac{\partial\varphi}{\partial U_s^n} = \frac{Y_s^n - \tilde{U}_s^n}{\sigma_s^2} , \tag{7}$$

which provide the second term in (6). As indicated in Fig. 2, similar contributions are derived from $\frac{1}{2}\chi^2$ at each time step for which there is a measurement. Differentiating the time-step update, Eq. (5), with respect to U^n, we obtain

$$\frac{dU^{n+1}}{dU^n} = A^{-1} . \tag{8}$$

Using Eqs. (7) and (8), we obtain the sensitivity of φ with respect to U^n

$$\frac{d\varphi}{dU^n} = (A^{-1})^T \frac{d\varphi}{dU^{n+1}} + \frac{\partial\varphi}{\partial U^n} \quad . \tag{9}$$

In the diagram this result comes from each ΔT transformation and flows backwards from U^{n+1} to U^n.

Similar use of the chain rule yields the sensitivity of φ with respect to θ, which result flows out of the top of the ΔT box in Fig. 2 and gets added to the total derivative of φ with respect to the D vector.

4.3. Inversion

The problem of reconstructing the unknown parameters D and μ_a from the measurements Y_s^n is an ill-posed inverse problem. Some form of regularization is necessary to make the solution well behaved. We accomplish this by employing an image model in the reconstruction process that incorporates our *a priori* knowledge regarding the unknown fields D and μ_a. Markov random fields (MRF) have been extensively used in image processing applications. We model D as a generalized Gaussian MRF (GGMRF) [19] with an energy function of the form

$$V\left(\frac{D}{\sigma_D}\right) = \sum_{\{s,r\}\in\mathcal{N}} \frac{b_{s-r}}{p} \left|\frac{D_s - D_r}{\sigma_D}\right|^p , \tag{10}$$

where $\mathcal{N}$ is the set of all neighboring pixel pairs, b is the weight assigned to the specific neighbors, and σ_D is the scale parameter of the model. The popular choice of $p = 2$ in the signal-processing literature yields a quadratic cost function, which tends to excessively penalize large deviations resulting in blurred edges. It is possible to provide good edge preservation in the reconstructed image for $p \approx 1$

[20]. Furthermore, the form of the model facilitates the estimation of the strength of this prior directly from the data [20]. In our formalism, the above energy function is taken to the minus-log-prior, which is added to the minus-log-likelihood (4) to obtain our objective function, the minus-log-posterior. Minimization of the minus-log-posterior yields the MAP solution.

Our example consists of a simulation of time-resolved data for a 6.4-cm-square section of tissue. Figure 3a shows as a 64×64 image the original diffusion coefficients, which range in value from 0.7 to 1.4 $\mathrm{cm}^2\mathrm{ns}^{-1}$. The absorption coefficients are set to a constant value of 0.1 cm^{-1}. The values of these coefficients, as well as the physical dimensions of the problem, have been chosen to reflect the properties of real tissue, since this modality is envisioned as potentially useful in medical imaging. Although the above method can be used to estimate D and μ_a simultaneously, we will restrict ourselves to the simpler case of just estimating D and assume that μ_a is known.

We assume that there are four pulsed sources placed at the midpoints of each side of the square region. There are 52 detectors evenly spaced around the perimeter, which measure the time-dependent signal in response to each pulsed source. Gaussian noise is added to the simulated signals with an rms value of 3% of the rms signal value over the 1.0 ns observation time, corresponding to a signal-to-noise ratio of 30 dB. The time step used is $\Delta t = 0.005$ ns and the detector resolution is 0.02 ns. Figure 4 shows the type of measurements obtained for one source position.

Figure 3b shows the MAP reconstruction for $p = 1.1$ obtained using 70 iterations of the conjugate gradient algorithm (taking about nine hours on an HP 9000/755). The reconstruction is remarkably good considering that effectively only four views (in the nomenclature of computed tomography) are used. This result confirms the value of incorporating ADICT into a simulation code to solve this inversion problem in which roughly 4000 parameters are determined from approximately 10000 measurements (54 detectors $\times$ 50 time samples $\times$ 4 source positions).

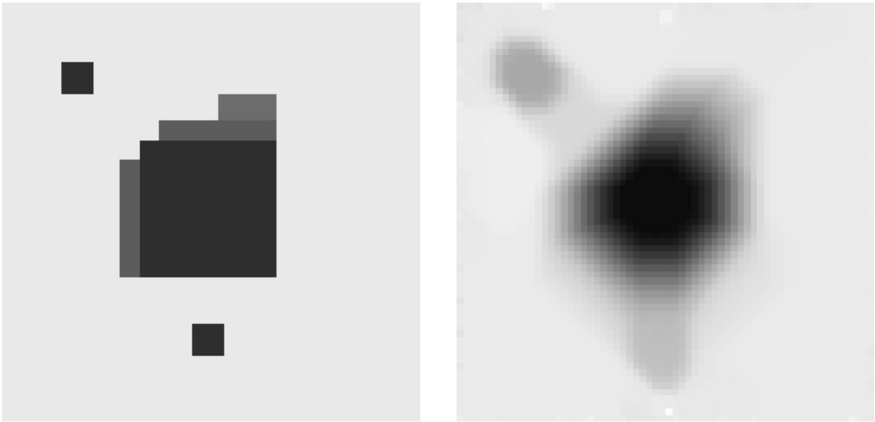
Inversion of these diffusion data is a truly difficult nonlinear problem. This task is much more complicated than inversion from x-ray radiographic measurements because the paths that the photons take depend on the distribution of diffusion coefficients that are being determined.

5. OTHER ENABLING TECHNIQUES

There are several other techniques that complement adjoint differentiation and optimization and help complete the suite of tools needed to conduct inference using large-scale simulations.

5.1. Multiscale optimization

When optimizing nonlinear models it is possible to encounter objective functions with multiple local minima, which can introduce difficulties for gradient-based algorithms. However, for many kinds of problems, the desired minimum can be found either by knowledgeably choosing a good starting point or by using a multiresolution approach, that is, by first finding the minimum at coarse resolution



(a) (b)

Figure 3: The original distribution of diffusion coefficients (a) and their reconstructed values (b) derived from time-dependent measurements made around the periphery in response to short pulses introduced at the middle of each of the four sides.

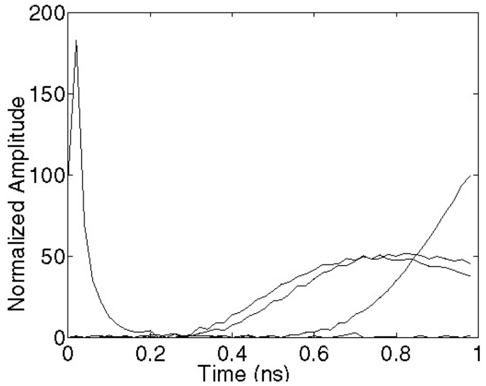


Figure 4: The simulated time-dependent measurements for a selected set of detectors presented as a function of time after a short pulse of light is introduced into the midpoint of the top edge of Fig. 3a. The first signal is from a detector located close to the source. The two signals that peak around 0.8 ns are from detectors positioned at the midpoint of the two sides and the late signal from the bottom side.

and then working toward finer resolutions [21, 22]. If this approach does not work for a particular problem, to find the global minimum it may be necessary to resort to stochastic optimization algorithms (simulated annealing or genetic algorithm), which are notoriously inefficient compared to gradient-based approaches. It might be possible to combine the best of both approaches through a hybrid algorithm in which the starting point of a gradient-based algorithm is chosen stochastically.

5.2. Deformable Geometric Models

It is often desirable to use a high-level model to describe an object or situation of interest. As an example, we have found it very useful to employ a deformable geometric model to represent the boundary of an object that we wish to reconstruct from projection data [21, 23]. In three dimensions, such a model consists of a surface, which might be represented by a grid of triangles [9, 24], as an example. An even higher-level model would use spline-based patches, which would result in fewer parameters, but not necessarily much less computation time. High-level models are often invoked to help regularize or control the inversion problem in the belief that it should be easier to solve a problem involving fewer variables. However, we have come to realize that it may be more desirable to use a very flexible description involving many parameters. The flexibility of such a model can be controlled through the use of a prior or graded constraint function, which effectively reduces the number of degrees of freedom of that model. Such constraints often take the form of an integral of the square of a derivative of some quantity, which basically acts to smooth that quantity. The advantage of this general approach is that it allows one to choose the prior that is most appropriate for the problem and even locally turn off the control when that is demanded by the data or circumstance [25].

5.3. Intermediate Elemental Representation

In constructing the Bayes Inference Engine [5], we have uncovered another basic tool for model building. In order to easily accommodate a variety of deformable geometric models in conjunction with a variety of potential measurement scenarios, we decided to employ an intermediate elemental representation for the object of interest. Every high-level model is converted to the elemental representation before the measurement process can be accomplished. For a physical object the elemental representation is a discretized density image (array of pixels) in two dimensions or a voxelated array in three dimension. This approach permits implementation of a new high-level object model without the necessity of writing a new code to calculate the result of each type of measurement.

Beyond its calculational advantage, we have come to recognize the underlying value of the elemental representation as a basic modeling tool. Its importance lies in the fact that when one uses and sees only a high-level model, deficiencies in matching the data can only be displayed in terms of the gradients of φ with respect to the model parameters. It can be difficult to recognize which aspects of the model do not accommodate the data. However, by displaying the gradients in the elemental representation, it can become evident how the high-level model needs to be changed, and possibly augmented, to better match the data [25].

5.4. Markov Chain Monte Carlo

One aspect of complex models that does not typically receive enough attention is the assessment of uncertainties in the parameters estimated from measurements. The probable reason for this neglect is that uncertainties are perceived to be difficult to estimate when there are numerous parameters. In a likelihood formulation the parameter uncertainties are described by the likelihood function. In a Bayesian formulation, they are described by the posterior probability distribution. The problem of assessing uncertainties amounts to characterizing the likelihood or the posterior. The Markov Chain Monte Carlo (MCMC) technique provides an effective means to characterize a probability distribution [26, 27, 28, 23]. MCMC generates a sequence of random samples of an arbitrary probability distribution. The only requirement of the basic algorithm is that one be able to evaluate the relative value of the probability density function for any set of parameter values, which in our context simply means doing the forward calculation to obtain ϕ plus the minus-log-prior.

This technique seems to be an essential tool to deal with the issue of uncertainties in estimated parameters. One valuable use of MCMC is to visualize the degree of model variation allowed by the uncertainties, which only requires that one has a way of displaying the model [29, 23]. Most useful is the ability to characterize the uncertainty in the model in whatever way one wishes, e.g., in terms of the variance in the parameters or of quantities derived from the model. In uncertainty assessment, it is crucially important to include the effects of correlations among the uncertainties, which MCMC does.

6. DISCUSSION

We have presented some useful tools that permit one to efficiently estimate parameters of a complicated forward model from measurements. The use of a forward model is important because many aspects of a model that describes a physical system and measurement scenario may not easily be directly inverted. This general approach allows one to construct complete models to fully account for the observations.

The primary technique in the toolkit is adjoint differentiation implemented according to the ADICT principle, which yields derivatives of a functional (objective function) based on a forward computational code with respect to all the parameters in the computational model. The derivatives of the computed functional are desirable when minimizing that functional. In the particular approach that we suggest, adjoint differentiation is accomplished through code rather than by storing derivative matrices. The resulting calculational time for the derivatives is comparable to that of the forward model calculation. ADICT can, in principle, be implemented for any code that is differentiable. Adjoint differentiation is being applied to some large simulation codes, e.g., to calculate the dynamics of the ocean and atmosphere.

ADICT permits the use of gradient-based optimization algorithms. We have had success with a quasi-Newton method of optimization and we therefore suggest using the limited-memory BFGS algorithm [16] for problems involving many parameters of mixed type. An important message that we wish to get across is that analysts should not necessarily be afraid of using models that contain large numbers of variables. With reasonable constraints on the models, one can easily accommodate many parameters. Our experience indicates that the penalties associated with using many parameters, either in terms of computational speed or ill posedness, are often not as severe as one might imagine.

Work is continuing on many fronts. The photon-diffusion work presented in Sect. 4. is progressing in several directions [30]. For example, we have demonstrated that it is feasible to reconstruction both D and μ_a from the kind of data used in this paper. Also, the diffusion simulation code has been extended to obtain three-dimensional reconstructions. These new results will be presented elsewhere [31]. Other work presently underway includes reconstruction of time-evolving 3D objects [24], improving the efficiency of the basic MCMC algorithm [32], and implementation of adjoint differentiation in a 2D hydrodynamics code [33, 7]. The current status of our developments may be found on the web: `http://bayes.lanl.gov/~kmh/publications.html`.

If one is going to rely on a simulation code to interpret data and make predictions, one needs to make certain that the simulation adequately matches real world behavior. This process is called validation [34]. While much needs to be learned about validation and how to go about it, we believe that the techniques presented here will play an important role in the methodology that evolves. Because the understanding of the uncertainties in predictions is a key component in validation, the MCMC or adjoint sensitivity techniques could become central elements in the validation process.

7. ACKNOWLEDGEMENTS

We have had helpful conversations with John Skilling, Steve Gull, Ralf Giering, Rudy Henninger, Maria Rightley, Paul Maudlin, and Bryan Travis. This work was supportedby the United States Department of Energy under contract W-7405-ENG-36.

References

1. S. S. Saquib, K. M. Hanson, and G. S. Cunningham, "Model-based image reconstruction from time-resolved diffusion data," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3034**, pp. 369–380, 1997.
2. G. Burgers, R. Giering, and M. Fischer, "Construction of the adjoint of the HOPE OGCM," *Ann. Geophysicae.*, **C14**, p. 390, 1996.
3. F. C. Vossepoel, G. Burgers, and P. J. vanLeeuwen, "Radar altimeter data assimilation in an intermediate ENSO model," *Ann. Geophysicae.*, **C14**, p. 429, 1996.
4. W. C. Thacker, "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.

5. K. M. Hanson and G. S. Cunningham, "A computational approach to Bayesian inference," 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.
6. R. J. Henninger, P. J. Maudlin, and E. N. Harstad, "Differential sensitivity theory applied to the MESA2D code for multi-material problems," in *Shock Compression of Condensed Matter*, S. C. Schmidt, et al., ed., pp. 283–287, Amer. Inst. Physics, Woodbury, NY, 1996.
7. R. J. Henninger, P. J. Maudlin, and M. L. Rightley, "Accuracy of differential sensitivity for one-dimensional shock problems," in *Shock Compression of Condensed Matter*, S. C. Schmidt, et al., ed., Amer. Inst. Physics, Woodbury, NY, 1998 (to be published).
8. M. L. J. Rightley, R. J. Henninger, and K. M. Hanson, "Adjoint differentiation of hydrodynamic codes," in *CNLS Newsletter*, Center for Nonlinear Studies, Los Alamos National Laboratory, Nov., 1997 (WWW: http://cnls.lanl.gov/Publications/newsletters.html).
9. X. L. Battle, G. S. Cunningham, and K. M. Hanson, "3D tomographic reconstruction using geometrical models," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3034**, pp. 346–357, 1997.
10. D. A. Harville, *Matrix Algebra from a Statistician's Perspective*, Springer, New York, 1997.
11. C. Bischof, A. Carle, P. Khademi, and A. Mauer, "The ADIFOR 2.0 system for the automatic differentiation of FORTRAN 77 programs," Tech. Rep. ANL-MCS-P481-1194, Argonne National Laboratory, 1995.
12. J. E. Horwedel, E. M. Oblow, B. A. Worley, and F. G. Pin, "GRESS 3.0 Gradient Enhanced Software System," Tech. Rep. PSR-231, RSIC Peripheral Shielding Routine Collection, Oak Ridge National Laboratory, 1994.
13. R. Giering, "Tangent linear and adjoint model compiler," Tech. Rep. TAMC 4.7, Max-Planck-Institut fuer Meteorologie, 1997 (e-mail: giering@dkrz.de).
14. W. C. Davidon, "Variable metric method for minimization," Tech. Rep. ANL-5990, Argonne National Laboratory, 1966.
15. R. Fletcher, "A new approach to variable metric algorithms," *Computer Journal*, **13**, pp. 317–322, 1970.
16. D. C. Liu and J. Nocedal, "On the limited memory BFGS methods for large scale optimization," *Math. Programming*, **45**, pp. 503–528, 1989.
17. S. G. Nash, "Preconditioning of truncated-Newton methods," *SIAM J. Sci. Stat. Comput.*, **6**, pp. 599–616, 1985.
18. S. G. Nash and J. Nocedal, "A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization," Tech. Rep. NAM 02, Northwestern University - Dep't. Elec. Eng. and Comp. Sci., 1989.
19. C. A. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. on Image Proc.*, **2**, pp. 296–310, 1993.
20. S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields, with applications to Bayesian tomography," Tech. Rep. TR-ECE 95-24, School of Electrical and Computer Engineering, Purdue University, 1995.
21. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Inter. J. Comp. Vision*, **1**, pp. 321–331, 1988.
22. G. S. Cunningham, I. Koyfman, and K. M. Hanson, "Improved convergence of gradient-based reconstructions using multi-scale models," in *Image Processing*, M. H. Loew and K. M. Hanson, eds., *Proc. SPIE*, **2710**, pp. 145–155, 1996.

23. K. M. Hanson, G. S. Cunningham, and R. J. McKee, "Uncertainty assessment for reconstructions based on deformable models," *Int. J. Imaging Systems and Technology*, **8**, pp. 506–512, 1997.
24. G. S. Cunningham, K. M. Hanson, and X. L. Battle, "Three-dimensional reconstructions from low-count SPECT data using deformable models," *Opt. Express*, 1998 (to be published).
25. K. M. Hanson, R. L. Bilisoly, and G. S. Cunningham, "Kinky tomographic reconstruction," in *Image Processing*, M. H. Loew and K. M. Hanson, eds., *Proc. SPIE*, **2710**, pp. 156–166, 1996.
26. J. Besag, P. Green, D. Higdon, and K. Mengersen, "Bayesian computation and stochastic systems," *Stat. Sci.*, **10**, pp. 3–66, 1995.
27. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London, 1996.
28. R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, New York, 1996.
29. 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.
30. A. H. Hielscher, S. S. Saquib, and K. M. Hanson, "Model-based iterative image reconstruction for photon migration tomography," in *Methods for Solving Ill-Posed Inverse Imaging Problems: Medical and Nonmedical Applications*, R. L. Barbour, M. J. Carvlin, and M. A. Fiddy, eds., *Proc. SPIE*, **3171**, pp. 106–117, 1997.
31. A. H. Hielscher, A. Klose, D. M. Catarious, Jr., and K. M. Hanson, "Tomographic imaging of breast and brain tissue by time-resolved, model-based, iterative image reconstruction," in *Proc. Conf. on Advances in Optical Imaging and Photon Migration*, Optical Society of America, Orlando, 1998 (to be published).
32. K. M. Hanson and G. S. Cunningham, "Posterior sampling with improved efficiency," in *Medical Imaging: Image Processing*, K. M. Hanson, ed., *Proc. SPIE*, **3338**, 1998 (to be published).
33. R. J. Henninger, P. J. Maudlin, M. L. Rightley, and K. M. Hanson, "Application of forward and adjoint techniques to hydrocode sensitivity analysis," in *Proc. 9th Nuclear Explosives Code Developers' Conference*, F. Graviana, et al., ed., Lawrence Livermore National Laboratory, 1996 (proceedings are classified; article available from authors as report LA-CP-96-235.
34. W. L. Oberkampf, M. N. Sindir, and A. T. Conlisk, "Guide for the verification and validation of computational fluid dynamics simulations," Tech. Rep. G-077-98, Amer. Inst. Aeronautics and Astronautics, 1998.