OptIC project: An intercomparison of optimization techniques for parameter estimation in terrestrial biogeochemical models

Cathy M. Trudinger,[1] Michael R. Raupach,[2] Peter J. Rayner,[3] Jens Kattge,[4] Qing Liu,[5] Bernard Pak,[1] Markus Reichstein,[4] Luigi Renzullo,[6] Andrew D. Richardson,[7] Stephen H. Roxburgh,[8,9] Julie Styles,[10,11] Ying Ping Wang,[1] Peter Briggs,[2] Damian Barrett,[6] and Sonja Nikolova[2,12]

Received 10 November 2006; revised 1 March 2007; accepted 20 March 2007; published 1 June 2007.

[1] We describe results of a project known as OptIC (Optimisation InterComparison) for comparison of parameter estimation methods in terrestrial biogeochemical models. A highly simplified test model was used to generate pseudo-data to which noise with different characteristics was added. Participants in the OptIC project were asked to estimate the model parameters used to generate this data, and to predict model variables into the future. Ten participants contributed results using one of the following methods: Levenberg-Marquardt, adjoint, Kalman filter, Markov chain Monte Carlo and genetic algorithm. Methods differed in how they locate the minimum (gradient-descent or global search), how observations are processed (all at once sequentially), or the number of iterations used, or assumptions about the statistics (some methods assume Gaussian probability density functions; others do not). We found the different methods equally successful at estimating the parameters in our application. The biggest variation in parameter estimates arose from the choice of cost function, not the choice of optimization method. Relatively poor results were obtained when the model-data mismatch in the cost function included weights that were instantaneously dependent on noisy observations. This was the case even when the magnitude of residuals varied with the magnitude of observations. Missing data caused estimates to be more scattered, and the uncertainty of predictions increased correspondingly. All methods gave biased results when the noise was temporally correlated or non-Gaussian, or when incorrect model forcing was used. Our results highlight the need for care in choosing the error model in any optimization.

Citation: Trudinger, C. M., et al. (2007), OptIC project: An intercomparison of optimization techniques for parameter estimation in terrestrial biogeochemical models, *J. Geophys. Res.*, *112*, G02027, doi:10.1029/2006JG000367.

1. Introduction

[2] The roles of terrestrial and oceanic biogeochemical processes in the climate system and the earth system are now well recognized [*Steffen et al.*, 2004; *Field and Raupach*, 2004]. There is a corresponding recognition of the need to include these processes in climate and earth system models, especially for long-term (multiannual and greater) simulations. Data assimilation into models has been under intensive development in meteorological and ocean forecasting since the 1980s, and has led to major improvements in forecast ability. Increasingly, data assimilation and parameter estimation methods (collectively termed model-data synthesis) are being used to constrain models of biogeochemical cycles (both in stand-alone form and as components of climate system models) with multiple sources of data [*Raupach et al.*, 2005]. Various methods have been used recently for parameter estimation in biogeochemical models, including gradient methods [*Wang et al.*, 2001; *Rayner et al.*, 2005], Kalman filter [*Williams et al.*, 2005], genetic algorithm [*Barrett*, 2002; *Roxburgh et al.*,

[1] Marine and Atmospheric Research, CSIRO, Aspendale, Victoria, Australia.

[2] Marine and Atmospheric Research, CSIRO, Canberra, ACT, Australia.

[3] Laboratoire CEA/CNRS/UVSQ, LSCE/IPSL, Paris, France.

[4] Max Planck Institute for Biogeochemistry, Jena, Germany.

[5] School of Earth and Atmospheric Sciences, Georgia Institute of Technology, Atlanta, Georgia, USA.

[6] Land and Water, CSIRO, Canberra, ACT, Australia.

[7] Complex Systems Research Center, University of New Hampshire, Durham, New Hampshire, USA.

[8] School of Biological, Earth and Environmental Sciences and Bushfire Cooperative Research Center, University of New South Wales, Sydney, New South Wales, Australia.

[9] ENSIS, Canberra, ACT, Australia.

[10] Oregon State University, Corvallis, Oregon, USA.

[11] Now at South Australian Research and Development Institute, Loxton, South Australia, Australia.

[12] Now at Bureau of Rural Sciences, Canberra, ACT, Australia.

0148-0227/07/2006JG000367$09.00

2006] and global search [*Braswell et al.*, 2005; *Knorr and Kattge*, 2005]. These are examples of parameter estimation in terrestrial carbon models with stock, flux and/or atmospheric CO_2 concentration measurements. There are also many applications in other areas of biogeochemistry and ecology, such as marine ecosystem models [*Losa et al.*, 2004].

[3] Different parameter estimation methods use different assumptions, and have different strengths and weaknesses. The OptIC project was designed to comparatively evaluate several methods for parameter estimation in biogeochemical models, with focus on the following: (1) ability to handle data inadequacies such as noise, correlations and gaps; (2) ways to optimize the model-data synthesis process in the absence of good error specifications for the data; (3) ways to handle multiple data sources with quite different properties (such as sampling interval, temporal averaging properties and error structures); (4) coping with issues like the observability of parameters (i.e., how well they can be inferred from the observations), including collinearity (also known as exchangeability or equifinality) where two or more parameters have a similar effect on observations and are therefore difficult to distinguish [*Aalderlink and Jovin*, 1997]; and (5) uncertainty analysis.

[4] The OptIC project was initially carried out between September and December 2005, and was open to anyone who wished to be involved. Participants were provided with pseudo data from a highly simplified test model, to which noise had been added, and they were asked to use the parameter estimation method of their choice to estimate the model parameters used to generate the data. There were 21 experiments with different model parameters and/or types of noise (from simple uncorrelated Gaussian noise to correlated noise, random drifts, noise from various distributions, missing data and incorrect forcing). Ten participants submitted results, with some submitting more than one set of results (generally with variations of the same method). This paper describes the model, experiments and approach taken in the OptIC project, and analysis of the results. More information about the OptIC project is available at http://www.globalcarbonproject.org/ACTIVITIES/OptIC.htm including model code, data sets and results.

[5] The outline of this paper is as follows. In section 2 we will describe the OptIC project, including the model, approach, noise types, experiments and optimization methods used. Section 3 gives the results. Discussion is given in section 4 and summary in section 5. The forcing function used in the model is described in Appendix A, and the optimization methods used by participants are described in Appendix B.

2. OptIC Project

2.1. OptIC Model

[6] The test model used in the OptIC project is a highly simplified representation of the carbon dynamics in a terrestrial biosphere model (TBM), with two state variables corresponding conceptually to stores of living biomass carbon (x_1) and litter and soil carbon (x_2). These variables are governed by the equations

$$\frac{dx_1}{dt} = \underbrace{F(t)\left(\frac{x_1}{p_1 + x_1}\right)\left(\frac{x_2}{p_2 + x_2}\right)}_{\text{Net Primary Production(NPP)}} - \underbrace{k_1 x_1}_{\text{Litterfall}} + \underbrace{s_0}_{\text{Seed production}} \tag{1}$$

$$\frac{dx_2}{dt} = \underbrace{k_1 x_1}_{\text{Litterfall}} - \underbrace{k_2 x_2}_{\text{Heterotrophic respiration}}, \tag{2}$$

where $F(t)$ is a forcing term describing input into the biomass store x_1 by net primary production (NPP), the flux of carbon into biomass by growth; p_1 and p_2 are scales for the limitation of production by lack of x_1 and x_2, respectively; k_1 and k_2 are rate constants for the decay of x_1 and x_2, respectively; and s_0 is a "seed production" term for x_1.

[7] This test model is not intended to be an actual TBM but rather a simplified version of a TBM with enough of the mathematical properties of a real TBM to answer the questions posed in the OptIC objectives. These properties include (1) nonlinearity, through the form of the NPP term; (2) multiple stores (here two); and (3) data streams giving information about the stores or fluxes (such as NPP) but not directly about the parameters, which must be estimated.

[8] The OptIC model is like an actual TBM in the following sense [*Raupach*, 2007]. All living biomass carbon (leaf, wood, root) is lumped into a single store x_1, and all litter and soil carbon into a single store x_2. These are respectively governed by equations of the form dx_1/dt = (NPP) − (litterfall) and dx_2/dt = (litterfall) − (heterotrophic respiration). The flux terms on each right hand side are identified in equations (1) and (2) (except for seed production, which is discussed shortly). Litterfall is parameterized as a flux k_1x_1 which is an outflow from x_1 and an inflow to x_2, where k_1 is a rate constant. Likewise, heterotrophic respiration is parameterized as an outflow flux k_2x_2 from the x_2 pool. NPP depends on the availability of essential resources (light, water and nutrients) and also on the biomass invested in organs for resource acquisition (leaves for light, roots for water and nutrients). The light and water resources are together represented by a forcing term $F(t)$ equal to the NPP under given light and water inputs, without limitation by either lack of biomass investment in resource-gathering organs or nutrient availability. The time dependence in $F(t)$ accounts for fluctuating availability of light and water through variation in weather and climate. The actual NPP is less than $F(t)$ because of lack of biomass investment in resource-gathering organs and lack of nutrients, described respectively by the factors $x_1/(p_1 + x_1)$ and $x_2/(p_2 + x_2)$ (of Michaelis-Menten form). To account for nutrient limitation, x_2 (litter and soil carbon) is used rather than a soil nutrient store, since these two stores tend to vary together. We consider observations of x_1 and x_2, which can be seen as surrogates for remote sensing data on biomass (such as NDVI-based estimates of green leaf area index) and in situ store measurements (such as soil carbon).

[9] The parameter s_0 is a small "seed production" term for x_1. It represents growth of biomass from seed, assumed

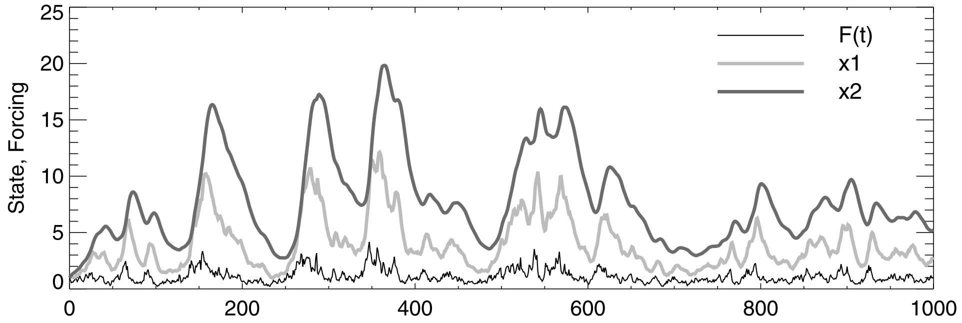

Figure 1. Test model behavior with reference parameter choices p_1 = 1, p_2 = 1, k_1 = 0.2, k_2 = 0.1, s_0 = 0.01. Forcing function $F(t)$ is log-Markovian with p_0 = 1, σ_m = 0.5, T_m = 10 Δt, and discretization interval Δt = 1 (see text).

(unrealistically) to be a constant growth flux independent of x_1, x_2 and external conditions. It is included for the following reason: if s_0 is absent (s_0 = 0), then "extinction" of the test model biosphere is possible because $(x_1, x_2) = (0, 0)$ is a stable equilibrium solution of equations (1) and (2). This does not occur if $s_0 > 0$ [*Raupach*, 2007]. Throughout, the forcing term $F(t)$ is assumed to be externally specified and is supplied to participants. It is created such that ln F(t) is an AR(1) random process (an autoregressive random process with a memory of one time step; see Appendix A).

[10] The test model has five parameters: p_1, p_2, k_1, k_2 and s_0. Figure 1 shows the behavior of the model with "reference" parameter choices p_1 = 1, p_2 = 1, k_1 = 0.2, k_2 = 0.1, s_0 = 0.01, and a reference forcing function $F(t)$ as described in Appendix A. Despite the test model being quite simple in form, it has some subtle properties as a dynamical system [*Raupach*, 2007]. For particular ranges of parameter values the model exhibits two stable states, and flips between "active" and "dormant" basins of attraction when triggered by the forcing.

2.2. Approach

[11] The OptIC project involved the estimation of optimum or minimum-error values for four parameters (p_1, p_2, k_1, k_2) in the test model (s_0 was set throughout to 0.01), using artificially generated data from a forward run of the model in which these parameters were assigned "true" values unknown to participants. The time series (($x_1(t)$, $x_2(t)$) from this forward run were treated as observations for the parameter estimation, but to mimic real data, they were subjected to various kinds of degradation (noise, correlations, drifts, gaps) before being supplied to participants. Thus the "observations" are $z_1(t) = x_1(t)$ + noise, and $z_2(t) = x_2(t)$ + noise. The observation series could contain missing values, possibly representing a large fraction of the data. In two cases, the forcing was degraded to some extent.

[12] Participants in the intercomparison were given Fortran90 code for a forward run of the test model plus an example driver file for this code and corresponding output (to confirm implementation of the forward model). For each experiment they received data sets of the forcing, $F(t)$, and noisy observations $z_1(t)$ and $z_2(t)$. They were told that true parameters would lie within known "prior" ranges shown in Table 1. They did not know what type of noise was added in each case, but were given a list of possibilities (see section 2.3), as well as Fortran90 code to generate these noise types for use in testing their method.

[13] An optimization process generally involves the minimization of some objective or cost function describing the mismatch between simulated and observed quantities. Optionally a term involving departures of model parameters from prior values may be included. Both types of mismatch must be weighted according to our confidence in the observations, the model and our prior knowledge. In OptIC it was up to participants to choose the form of their cost function, including weights, as well as any additional information required by their optimization method, such as initial parameter values or uncertainties, initial values for x_1 and x_2 or how to treat negative observations.

[14] Participants were asked to submit their best estimates of the parameters, and, if possible, the parameter covariance matrix, for each experiment. The forcing time series was specified for 12,000 time steps, but the noisy observations were available only for the first 10,000 time steps. Participants were asked to calculate $x_1(t)$ and $x_2(t)$ for the full 12,000 time steps, using their best estimates of the parameters and the given forcing $F(t)$ for each experiment. The last 2000 time steps were to test how differences in parameter estimates affect predictions of x_1 and x_2 when there is no data for assimilation.

2.3. Types of Observation Noise

[15] Below is a list of the types of noise that were added to the data. In each experiment the noise properties of z_2 were the same as those of z_1 (but with different measure of spread, σ). In the following, y_i is the uncorrupted observation (in our case $y_i = x_{1,\mathrm{i}}$ or $x_{2,\mathrm{i}}$), z_i is the observation with noise added, and w_i is a Gaussian random number with zero mean and unit variance. In most cases the noise added to the observations has zero mean.

[16] 1. Gaussian random noise with constant standard deviation σ was generated using

$$z_i = y_i + \sigma w_i. \tag{3}$$

[17] 2. Gaussian random noise with standard deviation σy_i (proportional to signal y_i) was generated using

$$z_i = y_i(1 + \sigma w_i). \tag{4}$$

Table 1. Prior Ranges for Parameters

Parameter	Minimum Value	Maximum Value
p_1	0.5	5
p_2	0.5	5
k_1	0.03	0.9
k_2	0.01	0.12
s_0[a]	0.01	0.01

[a]Parameter s_0 is fixed at 0.01.

[18] 3. Random noise with lognormal (skewed) distribution added to signal used

$$z_i = y_i + \exp(\sigma w_i) - \exp\left(0.5\sigma^2\right). \qquad (5)$$

[19] The last term is the population mean of the noise term $\exp(\sigma w_i)$, which is subtracted to give an unbiased (zero-mean) overall noise term.

[20] 4. Random noise with lognormal distribution multiplied by signal used

$$z_i = y_i \exp(\sigma w_i) - y_i\left[\exp\left(0.5\sigma^2\right) - 1\right]. \qquad (6)$$

[21] The noise population mean, the last term, is subtracted to give an unbiased overall noise.

[22] 5. For Gaussian random noise with a time-invariant correlation between noise in z_1 and z_2 at each time instant, we reversed the prewhitening procedure described by *Rodgers* [2000, Appendix C]. For a given covariance matrix $\mathbf{C}$, we multiplied a vector of random normal deviates with variance 1 by a matrix $\mathbf{X}$ where $\mathbf{C} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$. $\mathbf{X}$ is not unique and we use the Cholesky decomposition (*Press et al.*, 1986) for efficiency.

[23] 6. Noise correlated in time (Markov sequence) used

$$\begin{aligned} z_i &= y_i + m_i \\ m_i &= am_{i-1} + b\sigma_m w_i \quad \left(\text{with } a = \exp(-\Delta t/T_m),\ b = \sqrt{1-a^2}\right), \end{aligned} \qquad (7)$$

where σ_m is the standard deviation of the Markov sequence and T_m is its integral timescale. This is the timescale over which the noise is temporally correlated. The above is similar to generation of the forcing (Appendix A).

[24] 7. Gaussian random noise plus drifts of different magnitudes, resetting to zero at random intervals (mimicking calibration drifts) was generated using

$$\begin{aligned} z_i &= y_i + \sigma_n w_{n,i} + \zeta_i \\ \text{with } \zeta_i &= \zeta_{i-1} + k \qquad \text{if } H\left(p_d - w_{r,i}\right) = 0 \\ \text{and } \zeta_i &= 0, \quad k = \sigma_d w_{d,i} \quad \text{if } H\left(p_d - w_{r,i}\right) = 1, \end{aligned} \qquad (8)$$

where ζ_i is the drift part of the noise, $H(x)$ is the Heaviside step function ($H(x) = 0$ for $x < 0$ and $H(x) = 1$ for $x > 0$), p_d is the (small) probability of a resetting of the drift to zero, $w_{n,i}$, $w_{d,i}$ and $w_{r,i}$ are Gaussian random numbers with zero mean and unit variance, σ_d is the standard deviation of the zero-mean Gaussian distribution from which drift rate is drawn, and σ_n is the standard deviation of the Gaussian random noise. The drift rate k was held constant except for occasional resets.

2.4. Experiments

[25] There were 21 experiments in total. The project began with 16 experiments, but a further 5 experiments were added after some initial analysis of results. Eleven experiments (denoted A1 through A11) involved the same set of parameter values, initial values and true forcing time series, but different types of noise. Participants were told that experiments A1–A11 had the same parameters, but were asked to treat these as individual experiments, to give the best indication of how the different types of error influence the solution. There were also 10 experiments for which participants were told that each involved a different set of parameter values, a different forcing time series $F(t)$, and different types of noise imposed on the data. In reality, six of these experiments (B1–B6) had identical parameters with different noise types. The forcing time series for these six experiments was calculated with the same forcing parameters, but with different random seed terms. The remaining four experiments (C1–C4) had different model and forcing parameters. Tables 2 and 3 list the true parameter values and noise types for the 21 OptIC experiments. Figure 2 shows the noisy observations and true values of x_1 and x_2 for 500 time steps of each experiment. Parameter sets 5 and 6 (used in Experiments C3 and C4, respectively) show dual-mode behavior. Parameter sets 3 and 4 (used in Experiments C1 and C2, respectively) have relatively large x_1 and x_2 relative to p_1 and p_2 which may affect the observability of these parameters. Note that we generated noisy observations over 14,000 time steps then discarded observations from the first 2000 time steps to remove the effect of the choice of initial x_1 and x_2. Experiment A7 noise was Gaussian + drifts, and discarding the first 2000 time steps for this experiment left the noisy observations in the middle of a relatively large drift at the new initial time (Figure 2). This is a realistic situation as calibration might well be most uncertain farther back in time.

[26] Experiments A10 and A11 had the same noisy observations as A1, so these are not shown again in Figure 2. These two cases were supplied with forcing that had been degraded, and this forcing is shown in Figure 3. In both cases the supplied forcing was generated from the true forcing by applying a block running mean, then adding Gaussian noise and noise correlated in time (noise type 6). The error in the forcing for Experiment A10 was moderate and for Experiment A11 was severe. Figure 3 also shows x_1 and x_2 calculated with the true and incorrect forcing, as an indication of how sensitive the model is to the

Table 2. Values of the True Parameters for the Six Parameter Sets Used in the OptIC Experiments

Parameter Set	Experiments	p_1	p_2	k_1	k_2
1	A1–A11	1.04	1.35	0.23	0.08
2	B1–B6	2.44	2.45	0.11	0.031
3	C1	2.44	2.45	0.11	0.011
4	C2	0.77	2.73	0.033	0.025
5	C3	1.14	1.55	0.23	0.11
6	C4	4.6	1.45	0.63	0.011

Table 3. Noise Types for Each Experiment[a]

Experiment	Noise Type	Code
A1	Gaussian additive	G+
A2	Gaussian but noise in x_2 correlated with noise in x_1	GC
A3	Gaussian with 99% of x_2 data missing	GM
A4	Gaussian with 99% of x_1 and 90% of x_2 data missing	GM
A5	Gaussian with Gaussian extreme outliers	GE
A6	Gaussian + temporally correlated (Markov)	GT
A7	Gaussian + drifts	GD
A8	lognormal additive	L+
A9	lognormal multiplicative	L×
A10	Gaussian additive for obs; incorrect F(x)	IF
A11	Gaussian additive for obs; incorrect F(x)	IF
B1	Gaussian additive	G+
B2	Gaussian additive	G+
B3	Gaussian but noise in x_2 correlated with noise in x_1	GC
B4	lognormal additive	L+
B5	Gaussian + temporally correlated (Markov)	GT
B6	Gaussian + drifts	GD
C1	Gaussian additive	G+
C2	Gaussian additive	G+
C3	Gaussian multiplicative	G×
C4	lognormal additive	L+

[a] The third column gives a code for each noise type that will be used in the figures.

forcing (and therefore how hard it will be to estimate parameters with the incorrect forcing).

2.5. Optimization Methods and Other Choices

[27] Ten participants submitted results to the OptIC project, some submitting results using more than one approach or variations of the same approach, to give 15 submissions for the first round of experiments (A1–A3, A6–A8, B1–B6, C1–C4). There were slightly fewer submissions for the second round of experiments (A4–A5, A9–A11). Table 4 shows the list of parameter estimation methods used, and a code for each submission that will be used in figures and discussion. The methods are ordered in the table and later in the analysis roughly from local gradient-based searches to sequential methods to global searches using stochastic selection. The methods are described in more detail in Appendix B, and briefly here: The Levenberg-Marquardt method is a gradient-descent method that combines information about the gradient and the second derivative to efficiently locate the minimum [*Doherty*, 1999; *Press et al.*, 1986]. The adjoint method calculates the sensitivity of the model outputs to model parameters (for example using an automatic differentiation tool [*Giering*, 1997, 2000]) then uses a down-gradient search method to estimate the parameters. The Kalman filter is a sequential technique for state estimation, and here the Extended Kalman filter (EKF) and Ensemble Kalman filter (EnKF) were used with parameters included in the state vector (C. M. Trudinger et al., Using the Kalman filter for parameter estimation in biogeochemical models, submitted to *Environmetrics*, 2007) (hereinafter referred to as Trudinger et al., submitted manuscript, 2007). The Metropolis algorithm is a Markov chain Monte Carlo (MCMC) method that uses random walks to sample a probability distribution [*Metropolis et al.*, 1953]. Genetic algorithms are a type of search technique that involve a population of solutions that can reproduce, mutate, combine, or die, based on ideas of evolution and natural selection in biology [*Haupt and Haupt*, 1998].

[28] These optimization methods differ in a number of ways. One significant difference is in the way that they locate the minimum (such as gradient-descent or global search). Another is in how observations are processed, either sequentially in the Kalman filters or all at once ('batch processing') in the other methods. They also make different assumptions about the statistics. The MCMC methods, for example, allow representation of probability density functions (pdfs) that are non-Gaussian [*Knorr and Kattge*, 2005], whereas many other methods (such as Levenberg-Marquardt, EKF, EnKF) assume that pdfs are Gaussian. Adjoint methods minimize a cost function directly. If, as is normal, this cost function is quadratic, this is equivalent to a Gaussian assumption. Some methods are equivalent for the linear case with Gaussian noise but involve different approximations for nonlinear, non-Gaussian problems that can give rise to different solutions.

[29] Participants were not told what cost function to use, and choices varied. In all but two cases the cost function used can be written as

$$\Theta = \sum_i \left(\frac{(x_1(t_i) - z_1(t_i))^2}{w_1^2} + \frac{(x_2(t_i) - z_2(t_i))^2}{w_2^2} \right), \qquad (9)$$

where (inverse) weights w_1 and w_2 differed among participants. Some participants (*LM2*, *Adj1*, *Met2*, *Gen1*) used the same weights for both x_1 and x_2 in all experiments (i.e., $w_1 = w_2 = 1.0$ or $w_1 = w_2 = 0.01$). Others (*EKF, EnKF, Met1* and *Met1r*) ran a preliminary calculation with constant or arbitrary weights to determine the standard deviation of residuals for x_1 and x_2, then used these values as weights for z_1 and z_2. *Adj2* and *Met3* used the standard deviation of observations as weights, and *Gen1a* used the mean of observations as weights (with each of these methods treating z_1 and z_2 separately in each experiment). *LM1* used a linear function of the observation values as weights. These linear functions were based on linear regression of the standard deviation of residuals from a preliminary calculation against observations, where residuals had been binned according to the magnitude of the observation. In most cases the absolute part of the linear function dominated compared to the relative part, but in a couple of cases (experiments A8, B4 and C3) the relative part dominated. *LM2z* used weights in the cost function given by max$\{z_1(i)$,min[avg(z_1),avg(z_2)]$\}$ for $z_1(i)$ and similar for $z_2(i)$. *Met2z* was identical to *Met2* but with weights given by max[1.0, z_1(i)] for z_1(i) with similar for $z_2(i)$. An important distinction among these choices is whether weights vary for each observation or not; see Table 4.

[30] Two participants submitted a 'robust' version of their optimization, using cost functions that do not follow the form in equation (9). *LM1r* ignored the highest 2% of summands in the cost function, while *Met1r* used absolute deviations instead of squared deviations. These were submitted by the same participants as *LM1* and *Met1*, respec-

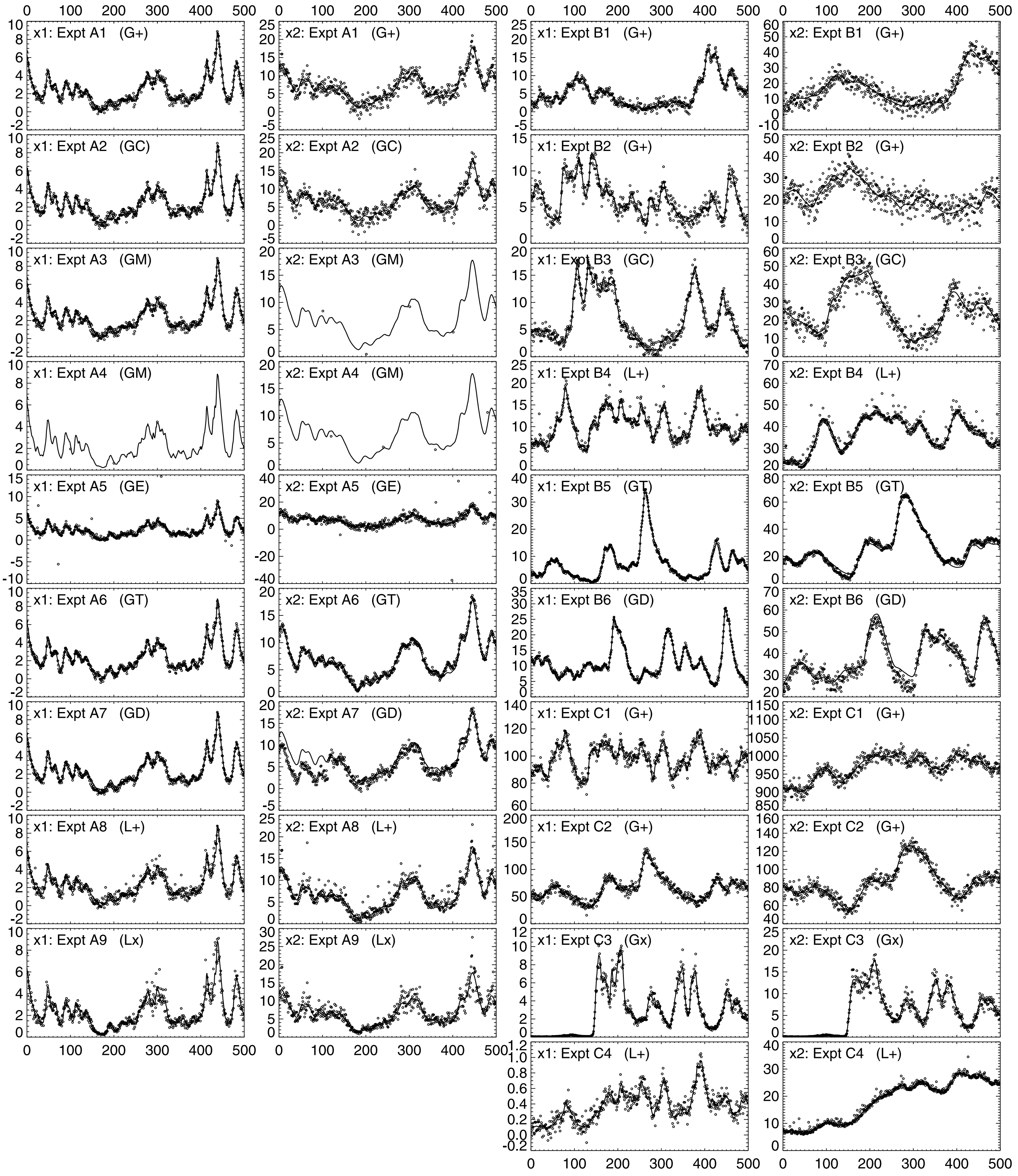

Figure 2. Five hundred time steps of noisy observations (symbols) and true values (solid lines) for variables x_1 and x_2 in all experiments. A10 and A11 are not shown because they used the same noisy observations as Experiment A1. Observations shown for Experiment C3 start from t = 500, to show both 'dormant' and 'active' periods. The noise type for each experiment is shown in brackets (see Table 3 for noise types).

tively, allowing comparison of the robust and nonrobust versions.

[31] True initial values of x_1 and x_2 were not supplied. Participants either used the first available observations for initial x_1 and x_2 or estimated initial x_1 and x_2 in their optimization. Some experiments had negative observations. Participants either ignored these or replaced them with zero. After some initial analysis of the results, participants were asked to repeat the calculations with all methods using common weights in the cost function (the standard deviation

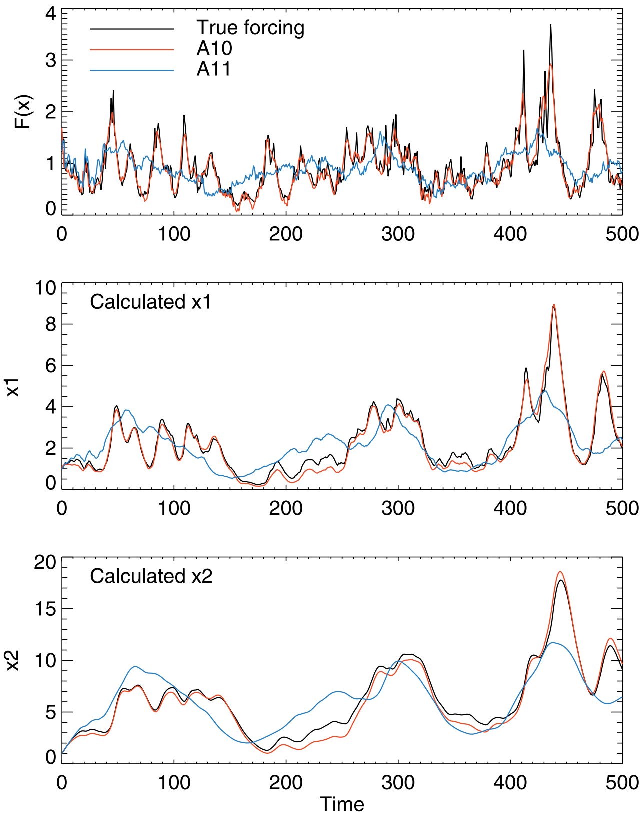

Figure 3. Forcing used to generate observations for experiments A1–A11 (black line), with the forcing supplied to participants for experiments A10 (red line) and A11 (blue line). x_1 and x_2 calculated with each of these forcing time series are also shown.

of residuals was chosen for this purpose and values were supplied to participants).

3. Results

3.1. General Features

[32] Figure 4 shows the parameter estimates divided by the true parameters for the 21 experiments. Each row corresponds to one parameter (p_1, p_2, k_1, or k_2). Experiments are ordered horizontally by parameter set. The codes along the top of each plot (G+, GC, etc.) refer to the noise type of each experiment (Table 3). (Also see auxiliary material[1] for figures showing all parameter estimates separately).

[33] Participants generally did very well at estimating the parameters k_1 and k_2 (to within about 5% of truth for most cases). The parameters p_1 and p_2 were significantly harder to estimate, because they are somewhat collinear or exchangeable (that is, they have a similar affect on observations, so can be hard to distinguish), but the estimates were mostly within 40% of truth. The estimated error covariance matrices suggested a strong anticorrelation in estimates of p_1 and p_2; that is, where an estimate of p_1 was too high it was likely that the estimate for p_2 would be too low. This accords with the behavior seen in the ensemble of estimates themselves. Both *LM2z* and *Met2z* had results that were consistently farther from the truth than the other submissions.

[34] Figure 5 shows contours of RMS differences from true x_1 and x_2, of x_1 and x_2 calculated for an array of incorrect parameters. Two parameters are varied at a time, while keeping the other two parameters fixed at their true values. x_2 values are multiplied by 0.5 because their magnitude is about twice that of x_1, to weight both x_1 and x_2 evenly. The true parameters for this figure correspond to the parameters in Experiments A1–A11. Note that this figure shows the difference of calculated x from true x, and the picture would change slightly if we considered the difference from noisy x. The 'long valley' (elongated contours with slope around -1) in the plot of p_1 and p_2 shows the collinearity of p_1 and p_2. This makes these two parameters difficult to estimate separately, and if this were a real application it might be advisable to try to rewrite the model equations so that all parameters were more easily observable [*Aalderlink and Jovin*, 1997]. However, this situation occurs often in real applications [e.g., *Richardson and Hollinger*, 2005], so it is instructive to see how the parameter estimation methods handle it. Note also that our model does not appear to have multiple minima, something that can be a problem for local down-gradient methods.

Table 4. Parameter Estimation Methods Used in the OptIC Project[a]

Code	Method	Weights
LM1	Monte Carlo then Levenberg-Marquardt	**lin(z)**
LM1r	as *LM1*, but ignore 2% highest summands in cost function	**lin(z)**
LM2	Levenberg-Marquardt	0.01
LM2z	Levenberg-Marquardt	**fn1(z)**
Adj1	down-gradient search using model adjoint	sd(resids)
Adj2	down-gradient search using model adjoint	sd(obs)
EKF	extended Kalman filter (with parameters in state vector)	sd(resids)
EnKF	ensemble Kalman filter (with parameters in state vector)	sd(resids)
Met1	Metropolis Markov chain Monte Carlo	sd(resids)
Met1r	as *Met1* but absolute deviations not least squares	sd(resids)
Met2	Metropolis Markov chain Monte Carlo	1.0
Met2z	as *Met2* but weights depend on noisy observations	**fn2(z)**
Met3	Metropolis Markov chain Monte Carlo	sd(obs)
Gen1	genetic algorithm (followed by downhill simplex)	1.0
Gen1a	genetic algorithm (followed by downhill simplex)	avg(obs)

[a] The first column shows the codes used to identify the methods in discussion and figures (where different numbers indicate submissions of the same method by different participants, e.g., *LM1*, *LM2* and Adj1 are by three different participants, but *Met2* and *Met2z* are by the same participant). The second column describes the method. The third column shows the weights used in the cost function (determined separately for z_1 and z_2 and for each experiment), where '1.0' refers to $w_1 = w_2 = 1.0$, '0.01' is $w_1 = w_2 = 0.01$, 'sd(obs)' is the standard deviation of observations, 'sd(resids)' is the standard deviation of residuals after a preliminary run of the calculation, 'avg(obs)' is the average of observations, 'lin(z)' is a linear function of the observation values, 'fn1(z)' is max$\{z_1(i)$,min[avg(z_1), avg(z_2)]$\}$ for $z_1(i)$ and similar for $z_2(i)$, and 'fn2(z)' is max[$z_1(i)$, 1.0] for $z_1(i)$ with similar for $z_2(i)$. Boldface emphasizes the cases with weights that vary with each observation.

[1] Auxiliary materials are available in the HTML. doi:10.1029/2006JG000367.

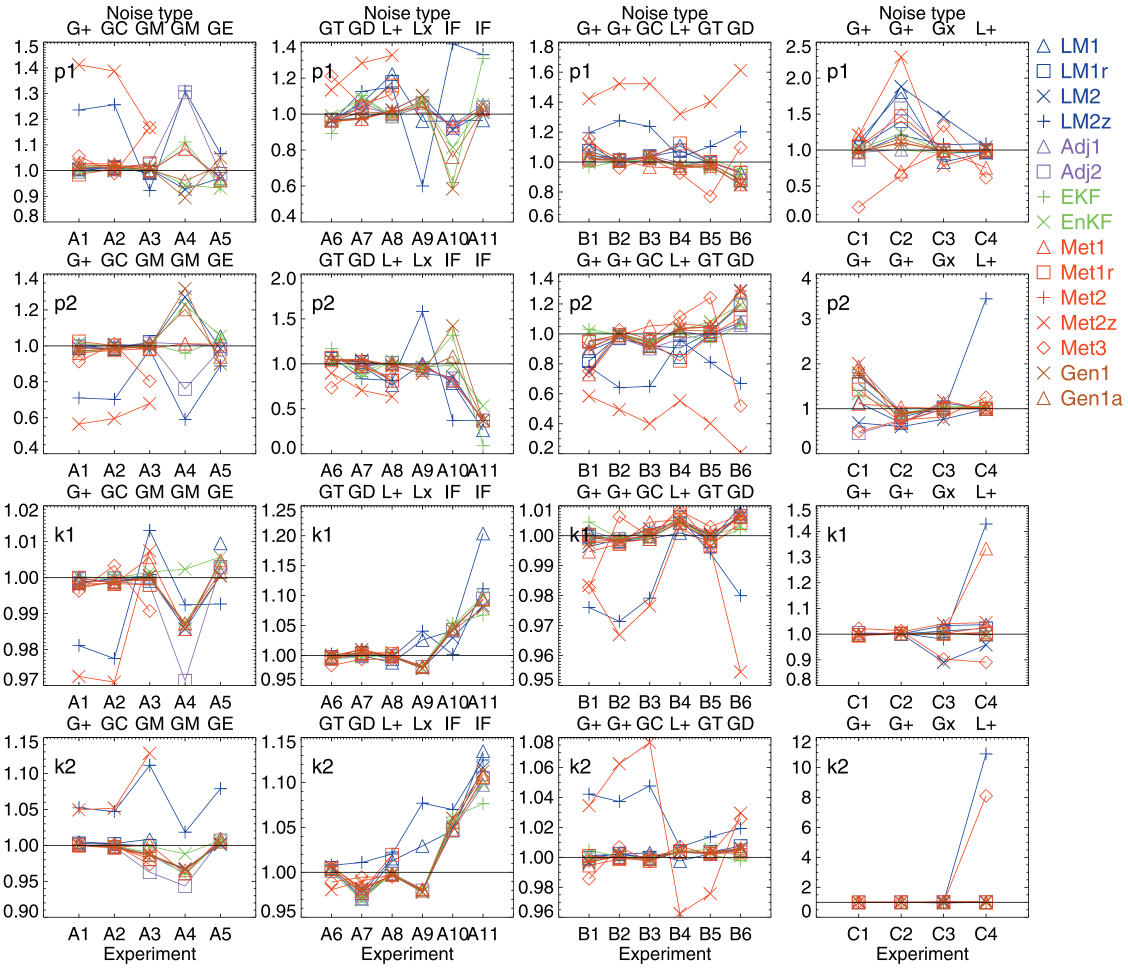

Figure 4. Parameter estimates divided by true estimates for all experiments. Labels on the lower x axis refer to Experiment, and labels on the upper x axis refer to the noise type for that experiment (see Table 3).

[35] We can analyze the error covariance matrix using the methods of principal component analysis [*Preisendorfer*, 1988], in which eigenvectors and eigenvalues of the covariance matrix reveal patterns of uncertainty in the estimates [*Menke*, 1989]. For example, the eigenvalues estimated by *Adj2* for experiment A1 were

$$\lambda = \left[9.7\mathrm{e}^{-8}, 1.0\mathrm{e}^{-6}, 4.9\mathrm{e}^{-4}, 3.5\mathrm{e}^{-2}\right],$$

and the eigenvectors are given by the columns of

$$\begin{bmatrix} 0.026 & 0.048 & 0.89 & -0.46 \\ 0.012 & 0.021 & 0.46 & 0.89 \\ 0.16 & 0.98 & -0.057 & 0.0032 \\ 0.99 & -0.16 & -0.020 & 0.0013 \end{bmatrix},$$

where the parameters are ordered as (p_1, p_2, k_1, k_2).

[36] These are typical of the eigenvalues and eigenvectors of covariance matrices estimated by the other methods for this and other experiments. The smaller eigenvalues correspond to the directions in parameter space that are best resolved by the optimization. If the ratio of an eigenvalue to the smallest eigenvalue is large, then the corresponding eigenvector defines a direction in parameter space that is not well resolved. In our example above, the first eigenvector (i.e., the best resolved direction) is dominated by a single element (0.99), corresponding to parameter k_2. The second eigenvector is also dominated by a single element (0.98), corresponding to parameter k_1. The third and fourth eigenvectors correspond to directions that are not as well resolved (indicated by smaller eigenvalues), and they each contain two significant components rather than one (0.89 and 0.46), indicating some correlation between p_1 and p_2. In general, the principal component analysis confirms that k_1 and k_2 are the most easily observed parameters, and that it is more difficult to separate the effect of p_1 and p_2.

[37] Participants estimated x_1 and x_2 for 12,000 time steps, on the basis of observations of only the first 10,000 time steps. A measure of skill is the RMS difference of predicted x_1 and x_2 from true x_1 and x_2, normalized by the mean x_1 and x_2 for each experiment. Figure 6 shows this measure for x_1 over the range $0 < t < 10{,}000$ to indicate how well the estimated x_1 matches the true values when there are observations, and for $t > 10{,}000$, indicating how well the estimated parameters can forecast x_1 into the future. (A corresponding figure for x_2 is given in the auxiliary material.) Although the estimated parameters may not

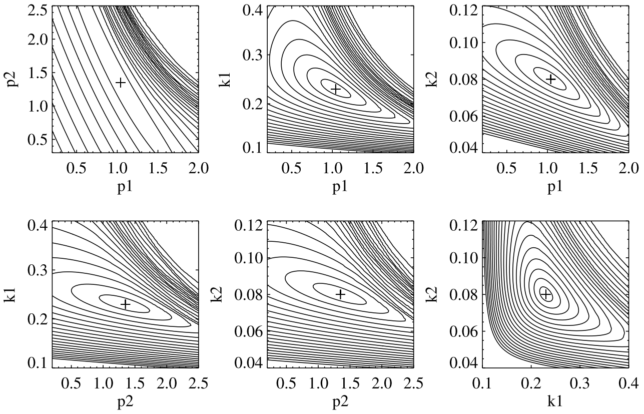

Figure 5. Contours of RMS differences of calculated x_1 and x_2 from true x_1 and x_2 for pairs of parameters for Experiments A1–A11. In each panel, two parameters were varied while the other two parameters were fixed at the true values. In calculating the RMS differences, x_2 values were weighted by 0.5 because their magnitude is about twice that of x_1. The plus symbols indicate the true parameters. The contour interval is 0.2.

match the true parameters exactly, we wondered whether the model may still be able to predict x_1 and x_2 well because of the collinearity of parameters. These results showed that when a method gave parameter estimates that were farther from the true parameters, it also predicted x_1 and x_2 that were farther from the truth. Thus, although there is a strong correlation between some parameters, particularly p_1 and p_2, the best match with true x_1 and x_2 was in fact obtained with the true parameters.

3.2. Choice of Cost Function

[38] In some cases there were substantial differences between implementations of the same method. Much of this difference seems to have been due to the choice of cost function. Cost functions based on (inverse) weights w_1 and w_2 (equation (9)) that depended on the observations (methods *LM2z*, *Met2z* and in some cases *LM1*) did not do as well as those using constant weights.

[39] For example, *Met2z* used the same optimization method as *Met2*, with the only difference being the weights in the cost function (*Met2z* weights varied with the noisy observations apart from a lower threshold for small z_1 and z_2, *Met2* weights were fixed to 1.0). *Met2* was more successful at estimating the parameters than *Met2z*. Similarly, *LM2* was more successful than *LM2z*.

[40] *LM1* used weights that were a linear function of observations (with nonzero intercepts). The linear function was determined from residuals after a preliminary optimization, as described earlier. The linear increase in the weights dominated over the constant part for experiments A8, A9, A11, B4 and C3, but in all of these cases except A9, *LM1* was less successful at estimating the parameters relative to performance of *LM1* for other experiments. The noise in experiments A8, A11 and C3 did not depend on the true x_1 and x_2, and use of weights that varied linearly with observations was not particularly successful. Experiment A9 had lognormal multiplicative noise that did increase with the true x_1 and x_2. In this case the parameter estimates from *LM1* differed from all of the other methods but were not significantly better or worse (in particular, k_1 and k_2 from *LM1* were too high, but from all other methods were too low by a similar amount). Experiment C3 had Gaussian multiplicative noise (i.e., noise that depended on the true x_1 and x_2), but *LM1* was less successful than the majority of methods. Most of the experiments did not have noise that varied with the magnitude of x_1 and x_2, but even in the cases where it did, use of weights in the cost function that varied with observations was not particularly successful.

[41] Two participants included a 'standard' and a 'robust' case calculated with the same optimization method. *LM1r* was identical to *LM1* but ignored the highest 2% of summands in the cost function. *LM1r* estimates were closer to the truth in ten of the experiments, similar in one experiment and worse in five experiments (B1, B4, B5, B6 and C1, covering a range of different noise types). (We quantified the differences between robust and standard cases by comparing for each experiment the sum of the absolute

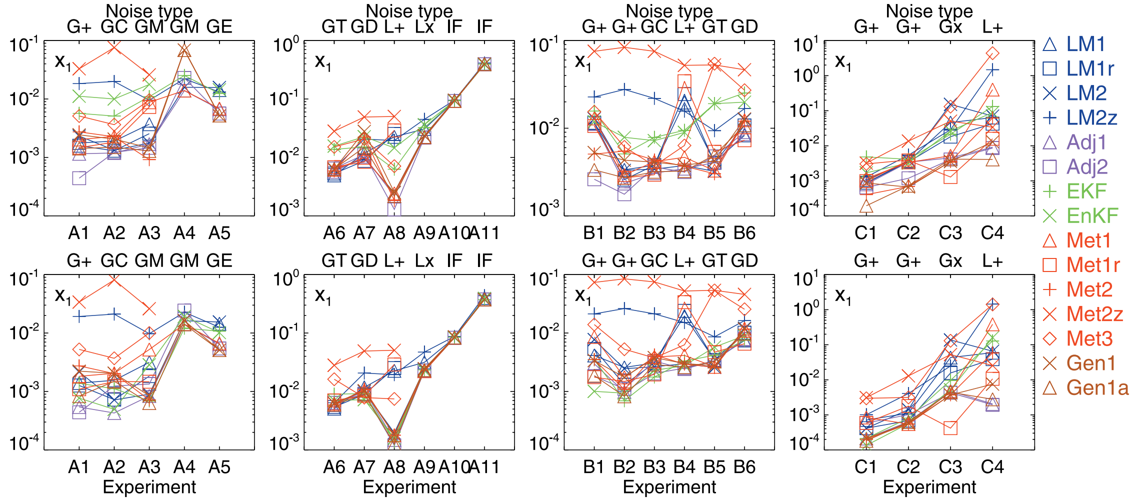

Figure 6. RMS difference of predicted x_1 from true x_1 normalized by mean x_1, for (top) 0 < t < 10,000 and (bottom) t > 10,000.

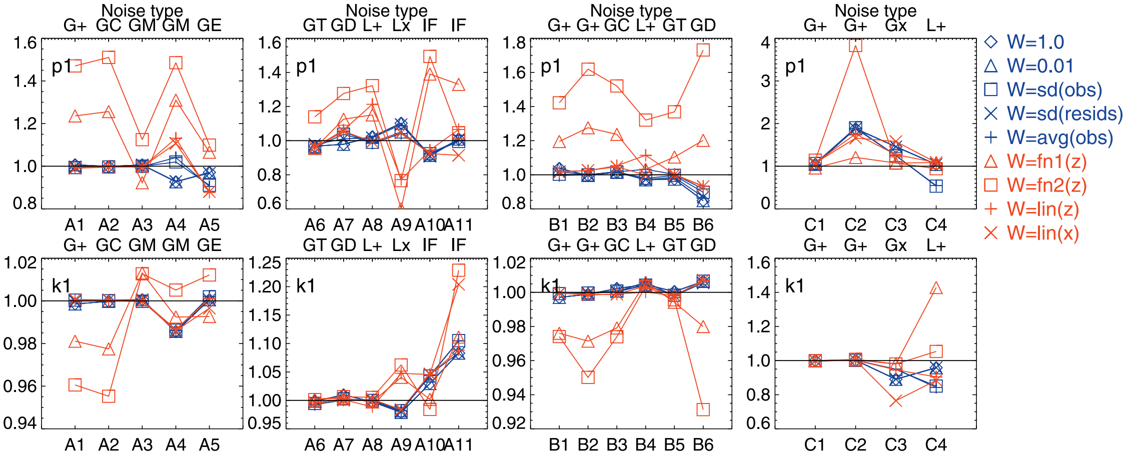

Figure 7. Parameter estimates for p_1 and k_1 divided by true estimates from running the Levenberg-Marquardt method PEST for a range of weights in the cost function (weight descriptions match Table 4). A corresponding figure for p_2 and k_2 is given in the auxiliary material.

deviations of each parameter estimate from truth divided by the standard deviation of all estimates (excluding *LM2z* and *Met2z*) of that parameter.) *Met1r* was identical to *Met1*, but used absolute deviations rather than squared deviations. *Met1r* was better than *Met1* in eleven of the experiments, slightly worse in two and significantly worse than *Met1* in three experiments (A8 and B4 with lognormal noise added and C2 with Gaussian noise multiplied). Neither *LM1r* nor *Met1r* did the 5 experiments A4, A5, A9, A10 and A11. Thus both 'robust' methods outperformed the corresponding 'nonrobust' case in the majority of experiments. There was more variation in results between *Met1* and *Met1r* (with absolute deviations) than between *LM1* and *LM1r* (ignored highest 2% of summands).

[42] In order to compare the choice of weights in the cost function without the effect of differences between methods, Figure 7 shows parameters p_1 and k_1 estimated by a single method with a range of different cost function weights, covering many of the choices in the OptIC results (a corresponding figure for p_2 and k_2 is included in the auxiliary material). The Levenberg-Marquardt package PEST [*Doherty*, 1999] that was used in the *LM2* and *LM2z* calculations was chosen for this purpose. The cases shown in blue have weights that are constant with time, while those in red have weights that vary for each observation (although recall that the linear function as described above was essentially constant for many experiments). The spread of results due to different cost function weights was similar to the spread from the full set of OptIC results. The results of *Met2z* showed strong correspondence with 'W = fn2(z)' that used the same weights. Cases with constant weights generally showed very similar results (Experiments C3 and C4, the two cases with active and dormant periods, had the most difference between results for different choices of constant weights).

[43] The cases 'W = lin(z)' and 'W = lin(x)' in Figure 7 both used the linear functions that were used by *LM1*, but 'lin(z)' implemented this as a function of noisy observations while 'lin(x)' implemented it as a function of true values of x_1 and x_2. Obviously true values would not be available in a real application, but this comparison was to test whether weighting by noisy observations would bias the results because an observation that is accidentally low is given more weight than one that is accidentally high by the same amount [*Evans*, 2003]. Of the five experiments that had a significant linear variation in weights, there were three in which the results with 'lin(x)' were significantly better than with 'lin(z)' (A8, A9 and B4), one where the results were similar for both (C3) and one where the results were slightly better for 'lin(z)' (A11). These results suggest that a bias can be introduced by weighting by noisy observations. The potential for bias would vary for different applications, dependent on the relative magnitudes of the noise and signal, but is certainly something to be aware of. Experiment A9 was slightly better with 'lin(x)' than constant weights, but was significantly worse with 'lin(z)' than constant weights. Figure 7 highlights the need for care in choosing the error model (including cost function).

3.3. Comparison of Methods

[44] The range of choices that OptIC participants made for the cost function weights (and its dominant effect on the results) was very interesting, but it obscured the comparison of the parameter estimation methods themselves. Therefore participants were asked to submit a second round of results determined using common cost function weights, corresponding to the standard deviation of residuals. Values for the weights were provided, but in a couple of cases participants used the standard deviations of residuals they had calculated themselves (these were very close to the values provided). Figure 8 shows estimates of parameters divided by true estimates calculated with the specified weights. (Also see auxiliary material for figures showing all parameter estimates separately.)

[45] There are fewer methods in this figure than in the original set of results, for three reasons: (1) the original set of 15 results had three cases where the same method was used with different cost function weights (*LM2* and *LM2z*; *Met2* and *Met2z*; *Gen1* and *Gen1a*), (2) 'robust' cases (*LM1r* and *Met1r*) were not resubmitted with specified weights, and (3) the participant who used *Met2* did not submit results in the second round. This left 9 methods that

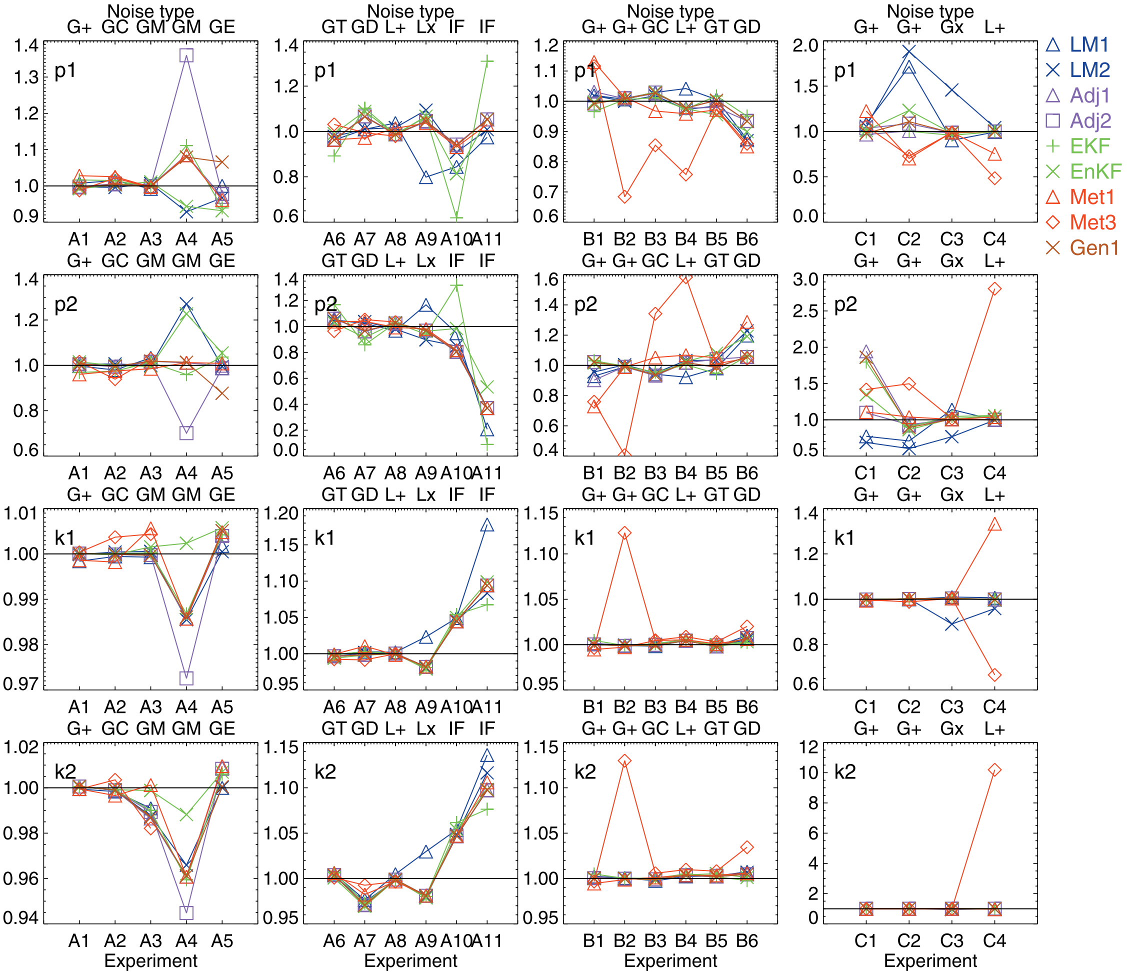

Figure 8. Parameter estimates for divided by true estimates calculated using weights in the cost function that correspond to the standard deviation of residuals.

used specified weights for comparison. The range of results was reduced from the original range mainly because we have removed the cases using weights dependent on the observations. There is a small amount of further reduction in spread from using the same values for the weights (based on the standard deviation of residuals), compared to the original spread of results from methods that used constant weights. An exception is *Met3*. This method is often furthest from the truth for experiments B1–B6 and C1–C4 in Figure 8. The reason for this is likely to be due to tuning of the optimization. Selection of an appropriate step size is known to be time consuming, and if not done properly can lead to problems. Some of the runs with this method did not converge. The original results from *Met3* (with weights of 1.0) were better than the results with supplied weights. Overall, the results in Figure 8 do not show any method or group of methods as being consistently more successful at estimating the model parameters.

3.4. Comparison of Noise Types

[46] Experiments A1–A11 have the same true parameters with different noise types. Experiments B1–B6 have the same parameters as each other and the same noise types as some of A1–A11. These two sets of experiments show similar patterns for common noise types.

[47] The cases with uncorrelated Gaussian noise (A1, B1 and B2) show similar results in terms of how well participants estimated the parameters and x_1 and x_2, relative to the other cases. Experiments with noise in x_2 correlated with noise in x_1 (A2 and B3) have a similar spread of results to the uncorrelated Gaussian cases, suggesting that this correlation did not make it more difficult to estimate the parameters.

[48] Experiment A3, with 99% of x_2 observations missing, has much greater spread in estimates for k_2. There is also greater uncertainty in this estimate compared to A1. Figure 9 shows the parameter estimates with uncertainties for these two experiments, where we use the square root of the diagonal element in the covariance matrix as a measure of the uncertainty in the parameter estimate. (It is worth noting that as there are some strong correlations in the estimated covariance matrices, particularly between p_1 and p_2, that the full uncertainty is not reflected in just the diagonal terms of the covariance matrix, but that the entire covariance matrix is required to characterize the uncertainty fully. Despite this, the diagonals can still provide a convenient measure of the uncertainty, useful for comparison between different experiments and methods.) The estimate of k_2 is vulnerable to missing or degraded x_2 observations,

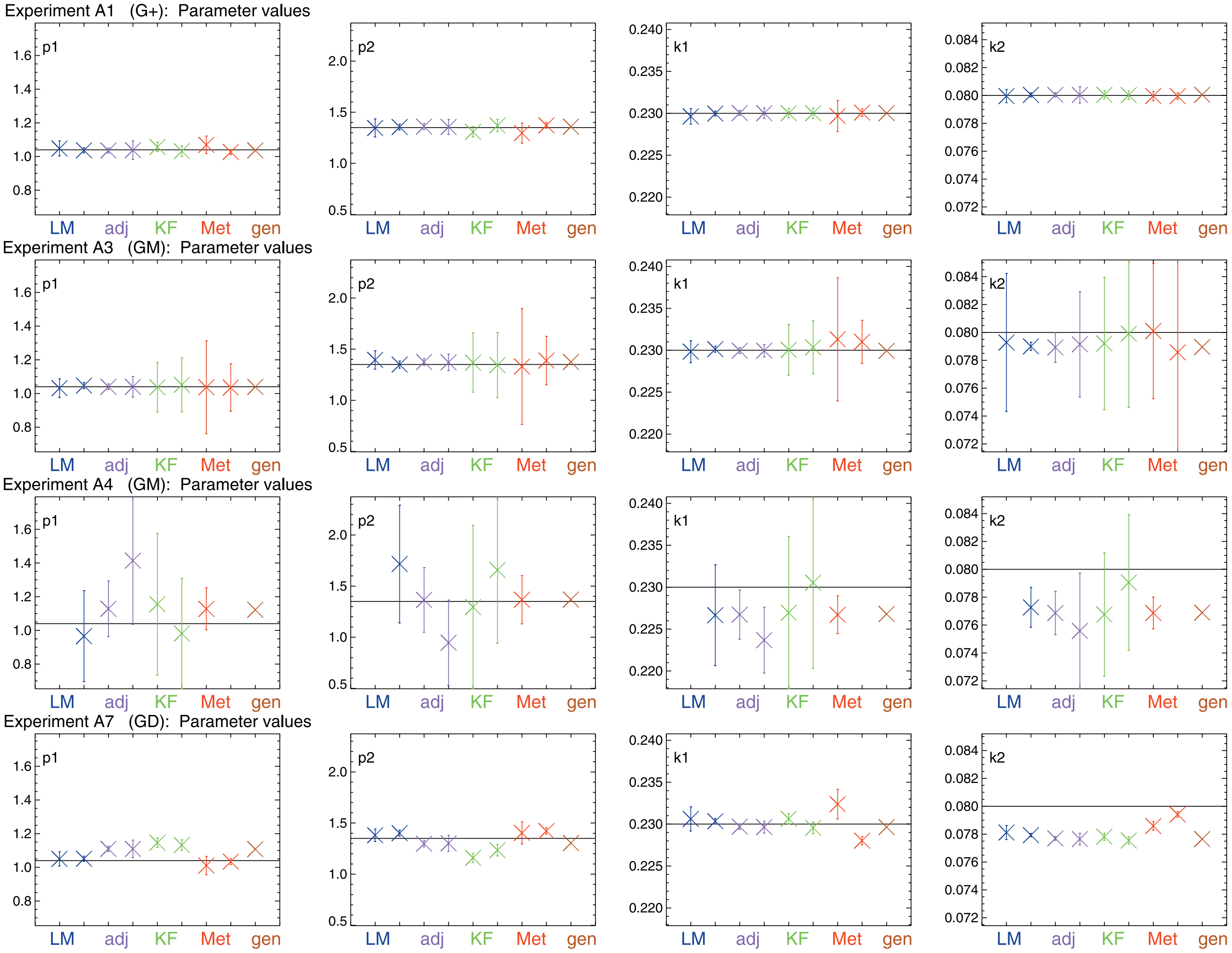

Figure 9. Parameter estimates with 1σ uncertainty range for experiments A1, A3, A4, and A7. The uncertainty range is the square root of the diagonal elements in the error covariance matrix (this was not available for the genetic algorithm). Results were generated using weights in the cost function that correspond to the standard deviation of residuals. Methods are ordered from left to right as listed in the key for Figure 8 (*LM1* and *Met3* were not submitted for Experiment A4). The horizontal lines show the true values of the parameters.

and this is reflected in greater uncertainties in the estimate of k_2. The variable x_2 corresponds conceptually to litter and soil carbon, which can be difficult to measure regularly or accurately, and hard to calibrate. Experiment A4 has most of x_1 and x_2 missing, and like A3, the spread of estimates is generally greater than A1 but this is accompanied by larger uncertainties (Figure 9).

[49] We now turn to the question of bias in parameter estimation. Cases with lognormal noise (A8 and B4), and temporally correlated noise (Markov in A6 and B5 and drifts in A7 and B6) all generally led not only to larger spread compared to Gaussian cases, but also to biases in estimates of the parameters. Importantly, biases are often not reflected in the uncertainties (Figure 9), leading to the risk of misleading interpretation. In real systems, noise is often not Gaussian, and Figure 9 demonstrates the problems that can arise when a Gaussian assumption is made. Characteristics of the residuals (observed minus predicted x_1 and x_2), such as distribution and autocorrelation, may suggest improvements that could be made to the error model to improve the results. The residuals for these cases clearly show non-Gaussian distributions or correlation in time.

[50] Two experiments (A10 and A11) supplied incorrect forcing, F(t). In both cases the estimated parameters were biased. Experiment A11 was particularly difficult, with many optimization methods wanting a negative value for parameter p_2, so the results often show it sitting at the lower boundary of the prior range or outside the range if participants allowed that outcome.

3.5. Comparison of Parameter Sets

[51] The experiments that gave participants the most trouble were A11 and C1–C4. In C3 and C4, the model flipped between dormant and active behavior, meaning that for some of the time the sensitivity to parameters was low. In C1, the parameter p_2 was particularly difficult to estimate because the mean value of x_2 was very large (around 1000) compared to the value of p_2 (i.e., with large x_2, the term $x_2/(p_2 + x_2)$ is less sensitive to variations in p_2). In C2, the true values of the rate constants k_1 and k_2 were similar, giving

similar mean values for x_1 and x_2 so that parameters p_1 and p_2 were even more difficult to distinguish than in other cases. A11 had severely incorrect forcing. For these difficult cases, parameter estimates were usually within 30% of true values for k_1 and k_2, and 100% for p_1 and p_2.

4. Discussion

[52] The choice of cost function appears to have been a very important component in the OptIC results, apparently more important than the choice of optimization method. *Evans* [2003] discussed choice of the cost function in biogeochemistry in detail. In a review of the literature on parameter estimation in biogeochemical models, he claimed that "it was hard to find two groups of workers who made the same choice for the form of the misfit function", with most of the differences being in the form of the weights. He also demonstrated with an example for the subtropical North Atlantic that the results of parameter estimation can vary substantially for different choices of the cost function. Our results from OptIC also showed that a range of initially reasonable choices for the cost function led to varying results.

[53] Analysis of the statistics of the residuals can be used to improve an optimization calculation. In most cases, OptIC participants did not allow the statistics of the residuals to alter their optimization approach. A major exception was *LM1*, where the cost function weights were determined for each experiment by analysis of the variation of residuals as a function of the observations. Unfortunately this method was not particularly successful, even when the residuals suggested that noise varied with the observations. This lack of success was apparently due to bias introduced because observations that are accidentally high are weighted lower than observations that are accidentally low. This result may be relevant for optimizations using eddy covariance flux measurements, following the conclusion of *Richardson et al.* [2006] that flux measurement errors scale with the magnitude of the flux. A number of participants determined the standard deviation of residuals from an initial optimization and used these as weights in the cost function, but did not alter their approach when the residuals were clearly non-Gaussian or temporally correlated. Examples of possible approaches to deal with the different error types could involve the use of transformations to make the noise more closely Gaussian, or, in the case of the EKF or EnKF, inclusion of an additional component with autoregressive evolution in the noise (or in the forcing for the cases with incorrect forcing) to cope with temporal correlations.

[54] In OptIC, the more complicated error models happened to be less successful than simpler assumptions. This finding will not necessarily hold for real applications with more complex models and error structures, but it serves as a strong warning for care in choice of the error model, and testing of the sensitivity to different options. In absence of good knowledge of the characteristics of the errors, simpler options may be better than complicated ones. A good optimization should begin with characterization of the statistics of the system (including noise), and the approach should be tailored to meet specific needs, but care is needed to avoid unintended consequences of the choices made.

[55] Two 'robust' cost functions were included, and they outperformed the corresponding standard least squares cost functions in the majority of experiments. In one case, 'robust' was defined as using absolute deviations, while in the other the highest 2% of summands in the cost function were ignored. Both approaches were similar in that they reduced the influence of outliers. *Richardson and Hollinger* [2005] and *Richardson et al.* [2006] have argued for the use of absolute rather than squared deviations for optimizations using flux tower measurements of energy or CO_2 fluxes (which they show can be described by a double-exponential distribution rather than a normal distribution). Here the robust methods were generally more successful for a range of different noise types including Gaussian.

[56] Our results showed no benefit from methods that allowed non-Gaussian pdfs for the posterior parameter estimate. Other applications where the model is more complicated, for example with multiple minima, might be expected to benefit from methods that allow non-Gaussian pdfs, but this would still not correct for an inappropriately specified error model.

[57] A few participants used a combination of global search and local gradient methods (e.g., *LM1* and *LM1r* used Monte Carlo then Levenberg-Marquardt, *Gen1* and *Gen1a* used a genetic algorithm then downhill simplex). Results for our experiments were not significantly different for local gradient or global search methods, so we are unable to assess how a combination of these approaches compares to either type of method on their own.

[58] There are a number of metrics for intercomparing methods, which are relevant for real applications, such as computational cost, setup cost, flexibility and simplicity of implementation, and information obtained from the inversion. The different methods varied substantially in the number of model integrations used (i.e., number of times the OptIC model was called to run over the entire time range), and therefore in the time taken to run the optimization calculations. The OptIC model, having simple equations with only two variables and four parameters, is very fast to run, so it was possible to run a very large number of integrations. However, in a more realistic application with a larger and/or more complex model, or with more parameters to estimate, limiting the optimization to a practical number of model iterations would be a much higher consideration. For some methods used in the OptIC project, it took significantly more iterations to estimate the covariance matrix than to estimate the parameters. Methods can be grouped into four categories in terms of number of iterations used: methods that took up to a few hundred iterations (*LM2*, *LM2z*, *Adj2*, *EKF*, *EnKF*), a few thousand iterations (*LM1*, *LM1r*), around 10,000 (*Gen1*, *Gen1a*) and 100,000 or more iterations (*Met1*, *Met1r, Met2, Met2z*) to estimate both the parameters and covariance matrix for each experiment. The global search methods used more iterations than the down-gradient and sequential methods, but some of the methods using a high number of iterations could also be run successfully with fewer iterations if required.

[59] There are significant differences between methods in terms of setup of the optimization calculation and ability to handle changes to model code. Adjoints can be expensive to

set up, but can handle simple changes to model code relatively easily. The EKF requires the Jacobian for the model, which may be difficult to specify for a complicated model, and would need to be updated if the model code changed. Monte-Carlo methods (EnKF and the global search methods) can easily cope with changes to model code, but they can require experience to achieve the most from the method due to choices of settings. The Levenberg-Marquardt method is available in easy-to-use packages [e.g., *Doherty*, 1999], copes easily with changes to model code, but is expensive when many parameters must be estimated simultaneously. All methods involve some choices, such as stopping criteria, or specification of model error for the Kalman filter.

[60] Another issue is the number of parameters that can be inverted simultaneously by the different methods. A high number of parameters may be a problem for down-gradient approaches like Levenberg-Marquardt, depending on the model and whether there are multiple minima. This could also be a problem for the Metropolis approach due to larger number of iterations required for many parameters, and therefore computational time.

[61] The Kalman filters (sequential methods) performed as well as the batch methods for these experiments, with the *EnKF* outperforming the *EKF* (as has also been shown in many other studies of strongly nonlinear models [e.g., *Verlaan and Heemink*, 2001]). The Kalman filter starts with the initial guess for the state variables (including the four parameters), and updates their estimates sequentially as it runs through time processing the observations. The state variable estimates at any time are due to observations up to and including that time. The parameter estimates from the final time are taken as the best estimates, because they have been influenced by all observations. The early estimates of x_1 and x_2 therefore do not correspond to the final (best) parameter estimates, and consequently the *EKF* and *EnKF* have relatively high RMS differences of x_1 from truth for $0 < t < 10,000$ in Figure 6 (the differences are less for the $t > 10,000$, or for the range $5000 < t < 10,000$). Use of the Kalman smoother would improve this (this was not done here). (The Kalman smoother involves first running the Kalman filter to process all data, then running the calculation backward in time from the final estimate to give state variable estimates at all times that depend on all observations.)

[62] A particular strength of the Kalman filter is the ability to assimilate observations allowing for errors in the forcing or the model. The OptIC model was known exactly, so our experiments did not exploit this advantage of the KF for model error. The EnKF can be used to account for errors in the forcing [e.g., *Moradkhani et al.*, 2006; *Slater and Clark*, 2006]. For the two OptIC experiments with incorrectly specified forcing, the error in the forcing was temporally correlated. This could have been included in the EnKF using state augmentation, which involves adding an extra component to the state vector [*Gelb*, 1974; *Reichle et al.*, 2002].

[63] *Zimmerman et al.* [1998] compared seven different inverse approaches for identifying aquifer transmissivity. They found that the most important factor for achieving a successful solution was the time and experience devoted by the user of the method. As choice of the cost function was clearly important in OptIC, our results suggest that experience and time devoted to looking at statistics of the residuals and refining the error model would be beneficial to obtaining the best solution in an optimization. In addition, the results from *Met3* with supplied weights were generally farther from the truth than other methods, and it appears that tuning of the method was an issue, which is related to experience and time devoted to the calculations. One participant initially used fixed values of 0.1 for the initial values of x_1 and x_2. This gave incorrect results, particularly for experiments with true initial x_1 and x_2 that were significantly different from 0.1. This further highlights the need for care in making the various choices that go into an optimization calculation.

[64] Some features of our model and experiments were simpler than might be expected in many real-world applications (such as the small number of parameters to estimate, short integration time, and a perfectly known model without multiple minima), while others were as difficult as would be found in some real-world applications (such as a strongly nonlinear model, collinearity of parameters, complicated and unknown types of noise, missing observations and incorrect forcing). Thus, while the tests in this paper do not probe all possible aspects of a comparison of optimization methods, our conclusion of the importance of the error model (including the choice of cost function) will still be highly relevant for the more complex cases. Provided the statistical inputs to the problem are correctly specified, the choice of optimization method can be made on computational or logistical grounds.

5. Conclusions

[65] The OptIC project used pseudo-data from a highly simplified test model to compare parameter estimation methods in an international intercomparison, with a focus on issues relevant to parameter estimation in terrestrial biogeochemical models. Overall we found the different types of methods equally successful at estimating the parameters. The choice of cost function differed among participants, and the most prominent feature of the results was that this choice had significantly greater impact on the results than the choice of optimization method. In other words, it mattered much more how participants defined the minimum than how they located it.

[66] The cost function involves the mismatch between simulated and observed quantities, often weighted according to confidence in the observations. Some participants used weights that were a function of the (noisy) observations, but they were less successful than those who used constant weights. Even when the residuals suggested a relationship between the magnitude of the noise and the signal, weighting by noisy observations was not particularly successful, apparently owing to bias caused because observations that are accidentally low are weighted higher than observations that are accidentally high. These results highlight the need for care in choosing the error model in an optimization, and may be particularly relevant for optimizations involving eddy covariance flux measurements where the errors have been shown to scale with the magnitude of the flux [*Richardson et al.*, 2006]. We recommend than an optimization involve analysis of the statistics of the residuals combined with testing of the sensitivity of results to the choice of error model.

[67] Two participants submitted a 'standard' (i.e., least squares) and a 'robust' version of their results, where 'robust' was defined (in slightly different ways) to downweight the influence of outliers in the cost function. Both robust methods generally outperformed the corresponding standard case.

[68] When common (specified) weights were used in the cost function, no group of methods (down-gradient, global search or sequential) was clearly more or less successful at estimating model parameters. The Kalman filter, a sequential state estimate method that has not commonly been used for parameter estimation until recently, was as successful at estimating the parameters as the batch methods. Results differed more owing to different noise types than optimization methods, suggesting again the importance of the error model.

[69] Some clear patterns emerged for the different types of noise. When the noise added to one variable was correlated with the noise in the other, this did not seem to affect parameter estimation. When most of the observations were missing, the parameters were not as well estimated, but uncertainties in the parameters increased to reflect the reduced information. When the noise was non-Gaussian or temporally correlated, biases occurred in all optimization methods, but the biases were not reflected in the uncertainties.

[70] In two experiments the forcing supplied to participants was degraded, one slightly and one severely. In both cases the estimated parameters were biased. This has implications for real applications, where forcing would rarely be perfectly known. In addition, the model itself would not be known exactly in a real application.

[71] We used a highly nonlinear model, with collinearity and complicated noise types, thereby addressing some of the issues faced in real-world problems. However, the model did not have multiple minima, many parameters to estimate or expensive computational cost, which are issues that could make some methods more appropriate than others in a different application. Although ours was a relatively simple example in some ways, our clear conclusion of the importance of the error model will still be highly relevant for more complex problems with additional issues to address.

[72] All optimization methods have choices to be made, and these have the potential to influence the results. In our application, no group of methods (down-gradient or global search, batch or sequential) was clearly more or less successful at estimating parameters. The main criterion for success was the choice of error model and thence the cost function being minimized. The OptIC model and data sets are available on the OptIC website http://www.globalcarbonproject.org/ACTIVITIES/OptIC.htm.

Appendix A

[73] The forcing function, $F(t)$, was supplied to participants. It was calculated as a random process such that ln $F(t)$ is AR(1) with specified mean and standard deviation. Specifically, we took $F(t) = p_0\exp(m(t))$, where p_0 is a measure of the mean magnitude of $F(t)$, and $m(t)$ is a dimensionless Markov process with zero mean, standard deviation σ_m, and timescale T_m and Gaussian increments (the Ornstein-Uhlenbeck process). The process $m(t)$ obeys the Langevin equation, the stochastic differential equation $dm/dt = -(m/T_m) + (\sigma_m\sqrt{2/T_m})\xi(t)$, where $\xi(t)$ is Gaussian white noise [*Arnold*, 1974; *Legg and Raupach*, 1982]. In finite difference form, $F(t)$ is given by

$$F(t_i) = F_i = p_0 \exp(m_i)$$

$$m_i = am_{i-1} + b\sigma_m w_i \quad \left(\text{with } a = \exp(-\Delta t/T_m),\ b = \sqrt{1-a^2}\right), \tag{A1}$$

where F_i and m_i are sequences of F(t) and m(t) at times t_i with increments Δt, and w_i is a Gaussian random number with zero mean and unit variance. This formulation ensured that F(t) was always positive. The quantities determining F(t) are p_0, σ_m and T_m. In the reference case, the values used were $p_0 = 1$, $\sigma_m = 0.5$ and $T_m = 10\ \Delta t$, with discretization interval $\Delta t = 1$ time unit. These properties of F(t) were given quantities, not model parameters to be estimated.

Appendix B

[74] This appendix describes the parameter estimation methods used in the OptIC project, and the assumptions made by each of the participants.

B1. Levenberg-Marquardt (LM1, LM1r, LM2, LM2z)

[75] Levenberg-Marquardt is a gradient-descent method for parameter estimation in nonlinear models. The algorithm combines the method of steepest descent and use of the Hessian (second derivative) to find the minimum, where the combination of methods changes as you approach the minimum.

[76] *LM1* started with a 1000 model run Monte-Carlo type exploration of parameter space to obtain the best initial guess, then performed the Levenberg-Marquardt algorithm. A bootstrapping approach was taken for uncertainties. *LM1r* was identical to *LM1* but with 2% highest summands in the cost function ignored. *LM2* performed the Levenberg-Marquardt algorithm from PEST (*Doherty*, 1999). *LM2z* was the same as *LM2* but with weights in the cost function that varied with observations.

B2. Adjoint (Adj1, Adj2)

[77] The adjoint of a model allows you to compute the sensitivity of model outputs to model inputs (including parameters). This can be used in a down-gradient line search to estimate model parameters.

[78] *Adj1* used the automatic differentiation tool TAMC [*Giering*, 1997] to generate the model adjoint and the Hessian (matrix of second partial derivatives from which uncertainties can be inferred). The gradient search method used was a limited-memory quasi-Newton method for large-scale bound-constrained or unconstrained optimization [*Byrd et al.*, 1995; *Zhu et al.*, 1997].

[79] *Adj2* used the automatic differentiation tool TAF [*Giering*, 2000] to generate the model adjoint, and the Hessian. The cost function included the squared weighted difference between modeled and observed x_1 and x_2, plus the squared weighted difference of parameters from a prior estimate (prior estimates for the parameters varied between cases). The gradient search method used was a steepest descent method.

B3. Kalman Filter (EKF, EnKF)

[80] The Kalman filter is a sequential method for estimating the state of a system. The Extended Kalman filter (EKF), which involves linearization of the model equations about the current trajectory, and the Ensemble Kalman filter (EnKF), which uses Monte Carlo techniques, can be applied to nonlinear systems [*Gelb*, 1974; *Evensen*, 2003].

[81] *EKF* and *EnKF* involved estimation of parameters by including them in the state vector, and estimating them sequentially in addition to the variables x_1 and x_2. Model error was zero for both variables and parameters. A preliminary run of the Kalman filter was performed, and two to three times the standard deviation of residuals was taken as the observation error for subsequent runs. This choice was based on experience with the EKF and EnKF that has shown better results for parameter estimation when greater uncertainties are used than would be suggested by residuals (Trudinger et al., submitted manuscript, 2007). Increasing the observation error gave better results for parameter estimation than increasing the model error. The EnKF calculations used 300 ensemble members, although results were similar down to about 50 ensemble members. The parameter estimates were usually close to their final values after about 1000 time steps. Trudinger et al. (submitted manuscript, 2007) describes the EKF and EnKF calculations for OptIC in more detail.

B4. Markov Chain Monte Carlo (Met1, Met1r, Met2, Met2z, Met3)

[82] Markov chain Monte Carlo (MCMC) methods are a family of techniques that use Monte Carlo sampling to generate a discrete approximation of the posterior probability distribution of the parameter(s) one seeks to estimate. These distributions often (except for trivial cases) elude direct calculation. MCMC invokes Bayes' rule to decompose a parameter's posterior distribution into a product of conditional distributions that are simpler to sample from. This quality has lead to MCMC being used in numerous parameter estimation problems [*Gilks et al.*, 1996].

[83] MCMC methods create a sequence of samples by performing random walks through parameter space. Each new value relies only on the value from the previous iteration (i.e., Markovian property). The size of each step is governed by a proposal distribution. The rules to decide the size of each step form the various instances of sampling algorithms. The Metropolis sampler, or more generally Metropolis-Hastings sampler [*Hastings*, 1970], is one of the more popular MCMC sampling algorithms [*Andrieu et al.*, 2003]. The algorithm requires "tuning" so that samples are generated at a rate that is sufficiently high to efficiently survey parameter space and ensure the resulting parameter estimates are globally optimal. There are many diagnostic measures to assess whether sequences have converged to the desired distribution. One of the most popular is that of *Gelman and Rubin* [1992] which uses multiple runs and iterations terminate once the ratio of the between- and within-run variance is close to unity. Given a sufficiently long sequence of parameter values, the final parameter estimates may be taken from the mean, median or mode of each distribution, or we may choose the maximum a posteriori estimate (i.e., the estimate corresponding to the maximum of the posterior distribution).

[84] *Met1* used the Metropolis algorithm from Numerical Recipes [*Press et al.*, 1986] with squared deviations in the cost function, while *Met1r* used the same algorithm with absolute deviations. *Met2* used the Metropolis algorithm. *Met2z* was the same as *Met2* but with weights in the cost function that varied with observations.

[85] *Met3* also used the Metropolis algorithm however with standardized data as inputs (i.e., the observations were scaled to have zero mean and unit variance). This corresponded to having weights of sd(obs) in equation (9). Initial tuning runs were conducted to ensure that candidate values were accepted at a rate between 20–75%. The *Gelman and Rubin* [1992] convergence statistic was used on four parallel runs. Convergence to the desired posterior distribution was observed for some data sets to occur by 40,000 iterations. For other data sets, when sequences failed to reach convergence by 40,000 iterations, iterations were manually terminated and visual appraisal was used to select the single "best" run. Using the values from all available runs, the maximum a posteriori estimates were chosen to be the final parameter values presented.

B5. Genetic Algorithm (Gen1, Gen1a)

[86] Genetic algorithms start with a population of randomly generated individuals. The cost function is evaluated for every individual in the population, then multiple individuals are stochastically selected from the current population (based on their fit to observations), and modified (mutated or recombined) to form a new population (the 'next generation'). The new population is then used in the next iteration of the algorithm, and this is repeated for a number of generations.

[87] *Gen1* and G*en1a* used an initial population size of 600 and the cost function of each was calculated. The best 48 solutions were retained and allowed to evolve for a further 250 generations with a mutation probability of 0.03. The genetic algorithm implementation was from *Haupt and Haupt* [1998] and was followed by a downhill simplex routine 'Amoeba' [*Press et al.*, 1986]. *Gen1* and *Gen1a* differ in their choice of cost function weights (1.0 or the average of observations, respectively).

[88] **Acknowledgments.** We thank the European Space Agency and IGBP for support and Ian Enting for useful discussions. A. D. R. would like to thank Dave Hollinger, Rob Braswell, and Jeff Gove for their valuable assistance. This paper is a contribution to Theme 1 of the Global Carbon Project (www.globalcarbonproject.org). The GCP acknowledges with appreciation the support of the Australian Climate Change Science Program.

References

Aalderlink, R. H., and J. Jovin (1997), Identification of the parameters describing primary production from continuous oxygen signals, *Water Sci. Technol.*, *36*, 43–51.

Andrieu, C., N. de Freitas, A. Doucet, and M. I. Jordan (2003), An introduction to MCMC for machine learning, *Mach. Learning*, *50*, 5–43.

Arnold, L. (1974), *Stochastic Differential Equations: Theory and Applications*, 228 pp., John Wiley, Hoboken, N. J.

Barrett, D. J. (2002), Steady state turnover time of carbon in the Australian terrestrial biosphere, *Global Biogeochem. Cycles*, *16*(4), 1108, doi:10.1029/2002GB001860.

Braswell, B. H., W. J. Sacks, E. Linder, and D. S. Schimel (2005), Estimating diurnal to annual ecosystem parameters by synthesis of a carbon flux model with eddy covariance net ecosystem exchange observations, *Global Change Biol.*, *11*, 335–355.

Byrd, R. H., P. Lu, and J. Nocedal (1995), A limited memory algorithm for bound constrained optimization, *SIAM J. Sci. Stat. Comput.*, *16*(5), 1190–1208.

Doherty, J. (1999), *PEST: Model-Independent Parameter Estimation*, Watermark Computing, Townsville, Queensland, Australia.

Evans, G. T. (2003), Defining misfit between biogeochemical models and data sets, *J. Mar. Syst.*, *40–41*, 49–54.

Evensen, G. (2003), The ensemble Kalman filter: Theoretical formulation and practical implementation, *Ocean Dyn.*, *53*, 343–367.

Field, C. B., and M. R. Raupach (Eds.) (2004), *The Global Carbon Cycle: Integrating Humans, Climate, and the Natural World*, 526 pp., Island Press, Washington, D. C.

Gelb, A. (1974), *Applied Optimal Estimation*, 374 pp., MIT Press, Cambridge, Mass.

Gelman, A., and D. B. Rubin (1992), Influence from iterative simulation using multiple sequences, *Stat. Sci.*, *7*(4), 457–511.

Giering, R. (1997), *Tangent Linear and Adjoint Model Compiler, Users Manual*, Max-Planck Inst. For Meteorol., Hamburg, Germany.

Giering, R. (2000), Tangent linear and adjoint biogeochemical models, in *Inverse Methods in Global Biogeochemical Cycles*, *Geophys. Monogr. Ser.*, vol. 114, edited by P. Kasibhatla et al., pp. 33–48, AGU, Washington, D. C.

Gilks, W. R., S. Richardson, and D. J. Spieglhalter (1996), *Markov Chain Monte Carlo in Practice*, 520 pp., CRC Press, Boca Raton, Fla.

Hastings, W. K. (1970), Monte Carlo sampling using Markov chains and their applications, *Biometrika*, *57*(1), 97–109.

Haupt, R., and S. Haupt (1998), *Practical Genetic Algorithms*, John Wiley, Hoboken, N. J.

Knorr, W., and J. Kattge (2005), Inversion of terrestrial ecosystem model parameter values against eddy covariance measurements by Monte Carlo sampling, *Global Change Biol.*, *11*, 1333–1351.

Legg, B. J., and M. R. Raupach (1982), Markov-chain simulation of particle dispersion in inhomogeneous flows: The mean drift velocity induced by a gradient in Eulerian velocity variance, *Boundary Layer Meteorol.*, *24*, 3–13.

Losa, S. N., G. A. Kivman, and V. A. Ryabchenko (2004), Weak constraint parameter estimation for a simple ocean ecosystem model: What can we learn about the model and data?, *J. Mar. Syst.*, *45*, 1–20.

Menke, W. (1989), *Geophysical Data Analysis: Discrete Inverse Theory*, Revised edition, *Int. Geophys. Ser.*, 289 pp., Elsevier, New York.

Metropolis, H., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953), Equations of state calculations by fast computing machines, *J. Chem. Phys.*, *21*, 1087–1092.

Moradkhani, H., K. Hsu, Y. Hong, and S. Sorooshian (2006), Investigating the impact of remotely sensed precipitation and hydrologic model uncertainties on the ensemble streamflow forecasting, *Geophys. Res. Lett.*, *33*, L12401, doi:10.1029/2006GL026855.

Preisendorfer, R. W. (1988), *Principal Component Analysis in Meteorology and Oceanography*, 425 pp., Elsevier, New York.

Press, W. H., B. P. Flannery, S. A. Teukolski, and W. T. Vetterling (1986), *Numerical Recipes*, pp. 523–526, Cambridge Univ. Press, New York.

Raupach, M. R. (2007), Dynamics of resource production and utilisation in two-component biosphere-human and terrestrial carbon systems, *Hydrol. Earth Syst. Sci.*, *11*, 875–889.

Raupach, M. R., P. J. Rayner, D. J. Barrett, R. DeFries, M. Heimann, D. S. Ojima, S. Quegan, and C. Schmullius (2005), Model-data synthesis in terrestrial carbon observation: Methods, data requirements and data uncertainty specifications, *Global Change Biol.*, *11*, 378–397, doi:10.1111/j.1365-2486.2005.00917.x.

Rayner, P. J., M. Scholze, W. Knorr, T. Kaminski, R. Giering, and H. Widmann (2005), Two decades of terrestrial carbon fluxes from a carbon cycle data assimilation system (CCDAS), *Global Biogeochem. Cycles*, *19*, GB2026, doi:10.1029/2004GB002254.

Reichle, R. H., D. B. McLaughlin, and D. Entekhabi (2002), Hydrologic data assimilation with the ensemble Kalman filter, *Mon. Weather Rev.*, *130*, 103–114.

Richardson, A. D., and D. Y. Hollinger (2005), Statistical modelling of ecosystem respiration using eddy covariance data: Maximum likelihood parameter estimation, and Monte Carlo simulation of model and parameter uncertainty, applied to three simple models, *Agric. For. Meteorol.*, *131*, 191–208.

Richardson, A. D., et al. (2006), A multi-site analysis of random error in tower-based measurements of carbon and energy fluxes, *Agric. For. Meteorol.*, *136*, 1–18.

Rodgers, C. D. (2000), *Inverse Methods for Atmospheric Sounding: Theory and Practice*, 238 pp., World Sci., Hackensack, N. J.

Roxburgh, S. H., S. W. Wood, B. G. Mackey, G. Woldendorp, and P. Gibbons (2006), Assessing the carbon sequestration potential of managed forests: A case study from temperate Australia, *J. Appl. Ecol.*, *43*, 1149–1159, doi:10.1111/j.1365-2664.2006.01221.x.

Slater, A. G., and M. P. Clark (2006), Snow data assimilation via an ensemble Kalman filter, *J. Hydrometeorol.*, *7*, 478–493.

Steffen, W. L., et al. (2004), *Global Change and the Earth System: A Planet Under Pressure*, 336 pp., Springer, New York.

Verlaan, M., and A. W. Heemink (2001), Nonlinearity in data assimilation applications: A practical method for analysis, *Mon. Weather Rev.*, *126*, 1578–1589.

Wang, Y. P., R. Leuning, H. A. Cleugh, and P. A. Coppin (2001), Parameter estimation in surface exchange models using nonlinear inversion: How many parameters can we estimate and which measurements are most useful?, *Global Change Biol.*, *7*, 495–510.

Williams, M., P. A. Schwarz, B. E. Law, J. Irvine, and M. R. Kurpius (2005), An improved analysis of forest carbon dynamics using data assimilation, *Global Change Biol.*, *11*, 89–105.

Zhu, C., R. H. Byrd, and J. Nocedal (1997), L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization, *Trans. Math. Software*, *23*(4), 550–560.

Zimmerman, D. A., et al. (1998), A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow, *Water Resour. Res.*, *34*, 1373–1414.

D. Barrett and L. Renzullo, Land and Water, CSIRO, GPO Box 1666, Canberra, ACT 2601, Australia. (damian.barrett@csiro.au; luigi.renzullo@csiro.au)

P. Briggs and M. R. Raupach, Marine and Atmospheric Research, CSIRO, GPO Box 3023, Canberra, ACT 2601, Australia. (peter.briggs@csiro.au; michael.raupach@csiro.au)

J. Kattge and M. Reichstein, Max-Planck-Institute for Biogeochemistry, Hans-Knöll-Str. 10, D-07745 Jena, Germany. (jkattge@bgc-jena.mpg.de; markus.reichstein@bgc-jena.mpg.de)

Q. Liu, School of Earth and Atmospheric Sciences, Georgia Institute of Technology, Atlanta, GA 30332, USA. (liuq1@ornl.gov)

S. Nikolova, Bureau of Rural Sciences, GPO Box 858, Canberra ACT 2601, Australia. (sonja.nikolova@affa.gov.au)

B. Pak, C. M. Trudinger, and Y. P. Wang, Marine and Atmospheric Research, CSIRO, PMB 1, Aspendale, Vic 3195, Australia. (bernard.pak@csiro.au; cathy.trudinger@csiro.au; yingping.wang@csiro.au)

P. J. Rayner, Laboratoire CEA-CNRS-UVSQ, LSCE/IPSL, Bat. 701 LSCE - CEA de Saclay Orme des Merisiers, F-91191 Gif/Yvette, Paris, France. (peter.rayner@cea.fr)

A. D. Richardson, Complex Systems Research Center, University of New Hampshire, Morse Hall, 39 College Road, Durham, NH 03824, USA. (andrew.richardson@unh.edu)

S. H. Roxburgh, Ensis, PO Box E4008, Kingston, ACT 2604, Australia. (stephen.roxburgh@ensisjv.com)

J. Styles, SARDI/PIRSA, PO Box 411, Loxton, SA 5333, Australia. (styles.julie@saugov.sa.gov.au)