Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012

The optimization problem in model-reduced gradient-based history matching

Sławomir P. Szklarz * **Marielba Rojas** **
Małgorzata P. Kaleta ***

* *Delft Institute of Applied Mathematics, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (e-mail: s.p.szklarz@tudelft.nl).*
** *Delft Institute of Applied Mathematics and Department of Geotechnology, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands (e-mail: marielba.rojas@tudelft.nl).*
*** *Shell Global Solutions International, Kessler Park 1, Urals Buildings, 2288 GS Rijswijk, The Netherlands (e-mail: malgorzata.kaleta@shell.com)*

Abstract: We present preliminary results of a performance evaluation study of several gradient-based state-of-the-art optimization methods for solving the nonlinear minimization problem arising in model-reduced gradient-based history matching. The issues discussed also apply to other areas, such as production optimization in closed-loop reservoir management.

Keywords: history matching, reservoir simulation, model order reduction, nonlinear programming, gradient-based optimization.

1. INTRODUCTION

Accurate numerical modeling of complex systems in applications such as oil-reservoir simulation, weather forecasting, and ocean modeling, usually involves calibration of the model based on observations of the system's behavior and on model predictions. Model calibration is known as data assimilation in weather and ocean modeling, and as history matching in reservoir simulation.

The model-calibration problem is usually posed as the nonlinear least-squares problem of minimizing the misfit between the model prediction and the available observations. In most applications of interest, discretization yields a very large number of model parameters and therefore, large-scale nonlinear optimization methods must be used. For example, in reservoir simulation, the number of discretized parameters such as permeability and porosity is typically of the order of $10^5 - 10^7$. Moreover, in simulation-based applications only function and gradient information is available. Hence, the choice of methods is limited to first-order optimization methods, and to second-order methods that only require approximate second-order information.

In recent years, model-order reduction techniques have been proposed in the context of model calibration (see Kaleta et al. (2011); Kaleta (2011); Vermeulen and Heemink (2006)) as an alternative to computing the adjoint of the original system in gradient-based history matching. In this approach, a sequence of approximate models in a reduced parameter space is constructed and a sequence of corresponding nonlinear least-squares problems for the reduced model is solved in order to determine a small number of parameters that are then used to reconstruct an estimate of the original parameters. The behavior of the original objective function is monitored at these approximations to the original parameters, to check for decrease in value. Algorithmically, the scheme is an inner-outer iteration where the inner iteration corresponds to computing the solution of a small-scale nonlinear least-squares problem, and the outer iteration controls the decrease of the original objective function. In Kaleta et al. (2011); Kaleta (2011); Vermeulen and Heemink (2006), model-order-reduction techniques are used to construct the reduced models. In particular, the Proper Orthogonal Decomposition (POD) proposed in Karhunen (1946); Loéve (1946), and Balanced POD, proposed in Moore (1981), were extensively explored.

We consider the efficient numerical solution of the nonlinear least-squares problem arising in model-reduced history matching proposed in Vermeulen and Heemink (2006), and extended in Kaleta et al. (2011) and Kaleta (2011), i.e. the inner iteration in the scheme described above. The problem solved in the inner iteration can be written as:

$$\min_{\eta} J_u(\eta, z) \equiv \frac{1}{2} \sum_{i=1}^{N_o} v_i(\eta, z)^T C_i^{-1} v_i(\eta, z) + \frac{1}{2} w(\eta, \theta)^T C_p^{-1} w(\eta, \theta) + \lambda^T f(\eta, z) \quad (1)$$

⋆ Contribution to invited session "Closed-loop production optimization in reservoir engineering". This research was carried out within the context of the **Recovery Factory Programme**, a joint program of Shell Global Solutions International and the Delft University of Technology.

where:

θ is a vector in the original parameter space;
η is a vector in the reduced parameter space;
$x = x(t)$ are the original state variables;
$z = z(t)$ are the reduced state variables;
t is time;
N_o is the number of time steps at which observations are gathered;
$v_i(\eta, z) = d_i - h_i(\eta, z)$;
d_i is the observation at time t_i;
θ^p is a vector of known parameter values (the prior);
$w(\eta, \theta) = \theta^p - \theta_k - \Phi_\eta \eta$;
k is the outer iteration index;
C_p is the covariance matrix of the prior's errors;
C_i is the covariance matrix of the observation errors at time t_i;
$h_i(\eta, z)$ describes the relationship between the reduced state variables and the observations.

In (1), $f(\eta, z)$ is a linear approximation, in the reduced variable and parameter spaces, to the dynamic operator that describes the system (reservoir simulator), and λ is the vector of Lagrange multipliers or adjoint-state vector of the reduced model. Note that the parameter-estimation problem is an inverse problem that is usually ill-posed. In this kind of problems, regularization techniques are needed in order to control the effect of errors (noise). In (1), the information corresponding to the prior (second term in (1)) is a regularization term.

Constructing a reduced model requires reduction of both states and parameters. To reduce the state variables, we construct the matrix Φ_z with N_z orthonormal columns corresponding to the dominant directions or *data patterns* of so-called *snapshots* of the system's behavior. Analogously, to represent the parameters in a reduced-order space, we construct the matrix Φ_η with N_η columns corresponding to the dominant eigenvectors (*parameter patterns*) of a low-rank approximation of the covariance matrix of the parameter field. The original parameters θ and the reduced parameters η (approximately) satisfy $\theta = \theta^p + \Phi_\eta \eta$. See Kaleta (2011); Kaleta et al. (2011) for further details. If N_x is the number of original state variables and N_θ is the number of components of the discretized parameter field, then N_z and N_η are chosen such that $N_z \ll N_x$ and $N_\eta \ll N_\theta$.

Problem (1) can also be formulated as a bound-constrained minimization problem. In this case, we are interested in solving:

$$\begin{aligned} \min_{\eta} J_c(\eta, z) \equiv & \frac{1}{2} \sum_{i=1}^{N_o} v_i(\eta, z)^T C_i^{-1} v_i(\eta, z) \\ & + \lambda^T f(\eta, z) \\ \text{subject to} \quad & L \leq \eta \leq U \end{aligned} \tag{2}$$

where all quantities are defined as in problem (1), and the inequalities are understood component-wise. In solving both (1) and (2), we assume that either relevant gradients or approximation to them are available.

A considerable simplification can be made in the minimization problems (1) and (2) by linearizing the functions h_i at given values θ_k and $x_i = x(t_i)$. This yields the expression:

$$\begin{aligned} & h_i(\eta, z) = \\ & h_i(\theta^p, x_i^p) + \frac{\partial h_i(\theta, x_i)}{\partial \theta} \Phi_\eta \eta + \frac{\partial h_i(\theta, x_i)}{\partial x_i} \Phi_z z, \end{aligned} \tag{3}$$

where $x_i^p = x^p(t_i)$ is the value of the state variables (at time t_i) corresponding to the prior θ^p. Note that with this choice of h_i, the functions J_u and J_c become convex quadratics. Therefore, the resulting optimization problem has a unique solution and most (descent) optimization methods are usually very efficient on this kind of problems. Note that although this simplification introduces additional approximation errors, this is often the only practicable approach since the nonlinear problems are usually very challenging. However, whenever possible, the nonlinear case should be preferred.

In this work, we present preliminary results of a performance evaluation study of several gradient-based optimization methods for solving (1). Note that, although the work focuses on history matching, most of the issues arise in model calibration in other areas as well as in production optimization in closed-loop reservoir management (see Jansen (2011a)).

The presentation is organized as follows. In Section 2, we present an overview of state-of-the-art optimization methods suitable for solving (1) and (2). In Section 3, we describe the reservoir model used as test problem. In Section 4, we describe our experiments, and present and discuss the results. Concluding remarks are presented in Section 5.

2. OPTIMIZATION METHODS

In this section, we review gradient-based methods for solving the problem:

$$\min_{\eta \in \Omega} f(\eta) \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be continuous and differentiable, and Ω is a convex set. Problem (1) and (2) are minimization problems of type (4), with $\Omega = \mathbb{R}^n$ in (1), and $\Omega = \{\ell_i \leq \eta_i \leq u_i, i = 1, 2, \ldots, p\}$ in (2). In practice, the problems are possibly large-scale.

2.1 Unconstrained minimization

Efficient derivative-based techniques exist for solving problem (4) in the unconstrained case, i.e. when $\Omega = \mathbb{R}^n$. Most of the techniques are so-called descent methods where the search direction is chosen so that the objective function decreases along that direction. Two of the main search directions are the Newton direction and the Cauchy or gradient direction. In order to compute the Newton direction, second-order information is needed. However, several gradient-based approximations exist that yield efficient and accurate methods. In this work, we focus on the following state-of-the-art families of methods which are representative of the approximate-Newton and Cauchy minimization approaches, namely: quasi-Newton methods, in particular the BFGS and limited-memory BFGS approaches combined with line search or trust-region globalization strategies (see Conn et al. (2000); Nocedal and Wright (1999)); and the spectral-projected-gradient (SPG) methods proposed in Birgin et al. (2000)

In quasi-Newton methods, an approximation to the Newton direction is computed by approximating the Hessian of the objective function at the current iterate. So-called secant approximations of the Hessian matrices are the most popular and efficient. In particular, the BFGS approach is very efficient for medium-scale problems when all previous gradients and iterates can be stored. In the large-scale case, the limited-memory BFGS approach is preferred since it requires the storage of only a few of the previous gradients and iterates. Indeed, Oliver et al. (2008) report that limited-memory BFGS was the most efficient method for solving the problems in their history-matching test set. Gauss-Newton-type methods are also based on approximating Hessian matrices.

Most practical minimization methods that follow descent directions converge to a local minimizer provided that the starting point is close enough to the solution. Robust versions incorporate so-called *globalization strategies* that guarantee convergence regardless of the choice of starting point. The most popular globalization strategies are line search and trust region, and they can be incorporated into most algorithms.

As the name indicates, projected-gradient methods are based on iterations that follow projected, and possibly modified negative gradient directions. The methods are designed for constrained minimization on convex sets and are efficient when projections onto such sets can be computed inexpensively. The spectral-projected gradient methods are among the most efficient of these approaches, thanks to the incorporation of a non-monotonic line search strategy.

Note that, with the exception of projected-gradient techniques, all methods above require the solution of a sequence of linear systems with a symmetric and positive definite matrix. In the medium-scale case, Cholesky factorizations are typically used for solving the systems, whereas large-scale problems are solved with the Conjugate-Gradients method. For more information on solutions of linear systems, see for example Golub and Van Loan (1996).

2.2 Bound-constrained minimization

If the feasible set Ω in (4) is a convex set different from $\mathbb{R}^n$, for example a (hyper)box as in (2), we can use projected-gradient-type methods. In particular, SPG methods are very efficient in practice since they usually identify the optimal face of the feasible set very quickly.

Other techniques include: active set methods, which iteratively construct a set of active constraints; interior-point methods, which generate iterates in the interior of the feasible set; sequential quadratic programming (SQP) methods, which solve a sequence of simple problems (quadratic objective, linear constraints) by using Newton's method to solve a related nonlinear system of equations. All techniques can be globalized by means of line search or trust-region mechanisms.

3. RESERVOIR MODEL

The test problem in our study was a synthetic 2D reservoir model describing isothermal, slightly-compressible, two-phase (oil-water) flow in the five-spot well configuration shown in Figure 1, with four production wells on the corners and one injection well in the center.

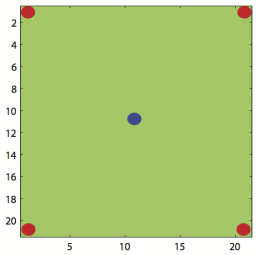

Fig. 1. Well configuration for a simple 2D reservoir model.

The initial reservoir pressure was $30 \times 10^6 Pa$ and the initial water saturation was assumed to be connate water saturation. Water was injected at a rate of $0.003 m^3/s$. The production wells were operated at a constant pressure of $25 \times 10^6 Pa$.

Gravity and capillary pressures were neglected, while porosity was assumed to be uniform with value 0.3. The only uncertain parameter was the permeability field. The data was generated as follows. The "true" field was chosen from 1000 model realizations. The remaining samples were used to obtain the prior permeability θ^p and the prior covariance matrix C_p. Bottom-hole pressure and production rates were assumed to have Gaussian errors with zero mean and standard deviations equal to 10% and 5% of the actual data, respectively. Synthetic production data was obtained by adding to the true data Gaussian errors with zero mean and variance equal to a fixed percentage of the true data. In addition, observations were assumed to be independent, and therefore the covariance matrices C_i were diagonal. Simulations were performed with the academic reservoir simulator SimSim described in Jansen (2011b).

Observations were taken at the five wells every 30 days during a period of 250 days. This resulted in eight assimilation times for a total of 40 observations. Hence, $N_o = 8$. The dimension of the field was $700m \times 700m \times 2m$ divided into $21 \times 21 \times 1$ uniform Cartesian grid blocks. The reduced model was constructed by means of the POD method. See Kaleta (2011) for more details.

4. NUMERICAL STUDY

Our numerical study consisted of comparing several Matlab implementations of the methods described in Section 2.1 when used for solving problems (1) and (2) on the test problem described in Section 3. In this work, we present results for problem (1), in the convex-quadratic case obtained by using the linearization (3).

We compared the performance of the codes for solving only the first minimization problem in the inner-outer iteration model-reduced, gradient-based, history-matching procedure described in Section 1, i.e. we solved the inner problem for $k = 1$ only, with k the outer-iteration index and $\theta_1 = \theta^p$. The reason for this choice was two-fold. Firstly, it is usually in the first inner iteration that a significant reduction in the objective function is achieved; and secondly, running the complete algorithm can be too

lengthy in the context of this study where several methods were used for the inner iteration. The test problem was the reservoir model described in Section 3. The number of original parameters was $N_\theta = 441$ and the number of original state variables was $N_x = 882$. Comparisons were performed with a number of reduced parameters $N_\eta = 20$ and $N_\eta = 300$. The number of reduced state variables was $N_z = 109$ when $N_\eta = 20$, and $N_z = 358$ when $N_\eta = 300$.

The comparisons were performed using the idea of *targets*. In this approach, we solved the problem with a state-of-the-art code (the *best* code) with default settings. Then, we adjusted the settings of the other codes to try to match, with a relative error of 10^{-3} or less, the minimum objective function value (target) computed with the best code.

The experiments were carried out in MATLAB R2010b on an ASUS F5Z with an AMD Athlon X2 dual-core QL-60 1.90GHz processor and 2GB RAM, running the Windows 7 Professional (32-bit) operating system. The floating-point arithmetic was IEEE standard double precision with machine precision $2^{-52} \approx 2.2204 \times 10^{-6}$.

We compared the following Matlab implementations of the optimization methods described in Section 2.1:

- KBFGS: Trust-region, quasi-Newton BFGS option of KNITRO version 7.0. See Byrd et al. (2006).
- KLBFGS: Trust-region, quasi-Newton limited-memory BFGS option of KNITRO version 6.0. See Byrd et al. (2006).
- MBFGS: Line-search, quasi-Newton BFGS option of Matlab's `fminunc` routine.
- IBFGS: Line-search, quasi-Newton BFGS routine in IMMOPTIBOX. See Nielsen (2010).
- SPG: the SPG method. See Birgin et al. (2000, 2001).

We also tested the trust-region option of Matlab's `fminunc` routine. However, in spite of the very low number of function and gradient evaluations reported, the code was considerably slower than the others. This seems to be due to the fact that the Hessian matrices are approximated by means of finite differences which require additional function evaluations. For this reason, this option is not included in our study. We also emphasize that the codes above are at a different stage of development. Therefore, timings are not reported since most of the codes are not programmed in the most efficient manner, which possibly leads to high overhead. Moreover, the KNITRO package is not programmed in Matlab, but a Mexfile interface to the C/C++ code is provided with the software. The KNITRO package is the most mature, robust, and optimized code of the ones considered, being a free-trial version of a commercial code. Therefore, in all experiments the target was the minimum objective function value computed by KNITRO's full BFGS method using the default settings.

The results are shown in Tables 1 and 2, where we report for each method: the number of function evaluations (F), the number of gradient evaluations (G), the value of the objective function J_u at the optimum (J_u^*), and the number of conjugate-gradients iterations when applicable (CG).

Several comments are in order regarding the results in Tables 1 and 2. We summarize them as follows:

Table 1. Unconstrained, convex, quadratic problem, $N_\eta = 20$.

Method	F	G	CG	J_u^*
KBFGS	23	17	81	24.04
KLBFGS	28	34	276	24.04
MBFGS	76	76	n/a	24.05
IBFGS	13	14	n/a	24.04
SPG ($m = 30$)	63	61	n/a	24.04

Table 2. Unconstrained, convex, quadratic problem, $N_\eta = 300$.

Method	F	G	CG	J_u^*
KBFGS	778	466	1818	29.61
KLBFGS	259	189	2082	29.62
MBFGS	236	236	n/a	29.62
IBFGS	143	163	n/a	29.62
SPG ($m = 5$)	400	230	n/a	29.57

- The SPG method can get lower function values than the other methods tested. This was observed in all our experiments.
- For the convex quadratic case, large values for the backtracking parameter m for the SPG method seem to speed up convergence.
- The IBFGS implementation of the (full) BFGS approach was consistently the most efficient in terms of number of function and gradient evaluations. Note that this method does not require additional calculations such as CG iterations.
- The KNITRO package involves other significant calculations (CG iterations) besides function and gradient evaluations, and this may affect the actual time required to solve a problem.

In our experiments, the size of the optimization problem is small enough (20 or 300) that full-BFGS methods are affordable. However, note that the smaller dimension (20) is about 5% of the original dimension (441). For real problems of dimension $10^5 - 10^7$, even this small percentage becomes significant and therefore, large-scale techniques such as limited-memory BFGS and SPG are the only options. Observe also that, of the methods tested, only the commercial package KNITRO implements the LBFGS approach. Therefore, the SPG method becomes very attractive. Moreover, given that (unlike KNITRO), the method does not require any additional calculation besides function and gradient evaluations, we expect that an efficient implementation of the method will be very competitive on this kind of problems, especially if appropriate tuning of the backtracking parameter m is performed.

Figure 2 shows four permeability fields, in natural-log scale, for the test problem described in Section 3: the true field; the prior; the fields obtained after one outer iteration when the inner minimization problem is solved with IBFGS, using reduced models or orders $N_\eta = 20$ and $N_\eta = 300$.

We observe in Figure 2 that the quality of the first-iteration approximation to the permeability field corresponding to $N_\eta = 20$ seems better than the one corresponding to $N_\eta = 300$. Indeed, we can also confirm this quantitatively by computing the Root Mean Square Error for the fields. The values of the error for the prior field,

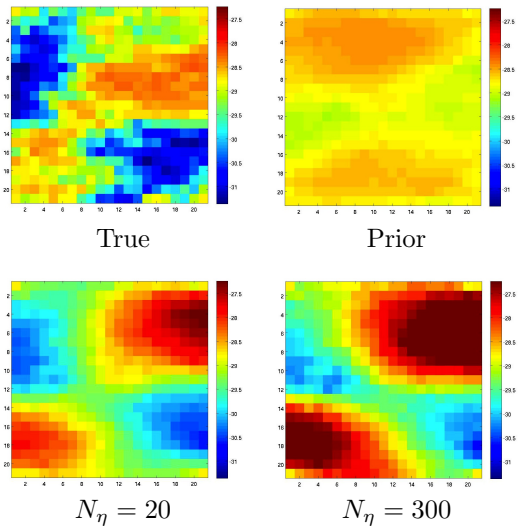

Fig. 2. Permeability fields for a 2D reservoir model.

for the solution corresponding to $N_\eta = 20$, and for the solution corresponding to $N_\eta = 300$ are, respectively: 1.16, 0.69, and 1.17. Therefore, it appears that not only is a large number of patterns not necessary from the point of view of recovering relevant parameter features, but also including too many patterns might bring in perturbations (noise) into the approximation. This seems to indicate that reducing the original model to a low-dimensional space has an additional regularization effect in the parameter estimation problem.

Finally, we note that the convex quadratic problem can also be formulated as a linear least squares problems and therefore matrix-computations techniques such as the QR factorization or the LSQR method proposed in Paige and Saunders (1982) can in principle be used in place of a nonlinear optimization approach. This is certainly possible in our case where the corresponding coefficient matrix is usually small and sparse. In practice, observations are gathered monthly for several years, and though sparse the coefficient matrix becomes very large and only large-scale techniques such as LSQR can be used. We emphasize that this approach only applies to the convex quadratic case. Since the ultimate goal is to preserve the nonlinearity in the objective function, it seemed appropriate to choose methods for solving nonlinear problems. This choice also makes it easier to compare the accuracy of solutions obtained using the convex-quadratic objective function and the nonlinear function.

5. CONCLUDING REMARKS

We have presented preliminary results from a numerical evaluation of nonlinear optimization software for solving the minimization problem in model-reduced gradient-based history matching. Our results indicate that the implementation of the well-known BFGS approach in Nielsen (2010) is the most efficient of the software tested and should be preferred for problems of medium scale. The results also indicate that the SPG method is a promising (publicly available) option for solving large-scale instances of the minimization problem considered. We showed that it is also possible to formulate the unconstrained minimization problem as an equivalent bound-constrained problem, and also to treat the nonlinear problems in the reduced parameter space. These extensions are the subject of Szklarz et al. (2012).

Even though in general the main computational cost in model-reduced gradient-based history matching is concentrated on the collection of the snapshots and on the linearization of the system's equations, the cost of solving a sequence of large and possibly highly-nonlinear optimization problems might become significant in practice. Therefore, choosing efficient methods for solving these problems becomes a relevant issue. Studies such as the one presented in this work might be useful in guiding the choice of appropriate methods.

REFERENCES

Birgin, E., Martínez, J., and Raydan, M. (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optimiz.*, 10(4), 1196–1211.

Birgin, E., Martínez, J., and Raydan, M. (2001). Algorithm 813: SPG - software for convex constrained optimization. *ACM T. Math. Software*, 27(3), 340–349.

Byrd, R., Nocedal, J., and Waltz, R. (2006). KNITRO: An Integrated Package for Nonlinear Optimization. In G. di Pillo and M. Roma (eds.), *Large-Scale Nonlinear Optimization*, 35–59. Springer-Verlag.

Conn, A., Gould, N., and Toint, P. (2000). *Trust Region Methods*. SIAM, Philadelphia.

Golub, G. and Van Loan, C. (1996). *Matrix Computations*. John Hopkins University Press, Baltimore, third edition.

Jansen, J. (2011a). Adjoint-based optimization of multi-phase flow through porous media a review. *Comput. Fluids*, 46, 40–51.

Jansen, J. (2011b). SimSim: A Simple Reservoir Simulator. Delft University of Technology, Department of Geotechnology, Delft, The Netherlands.

Kaleta, M. (2011). *Model-reduced Gradient-based History Matching*. Ph.D. thesis, Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands.

Kaleta, M., Hanea, R., Jansen, J., and Heemink, A. (2011). Model-reduced gradient-based history matching. *Computat. Geosci.*, 15, 135–153.

Karhunen, K. (1946). Zur spektral theorie stochasticher prozesse. *Annales Academiae Scientiarum Fennicae*.

Loéve, M. (1946). Functions aleatoires de second ordre. *Revue Science*, 84, 195–206.

Moore, M. (1981). Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE T. Automat. Contr.*, 26, 17–32.

Nielsen, H. (2010). A Matlab Toolbox for Optimization and Data Fitting. Department of Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark. Available at `http://www2.imm.dtu.dk/~hbn/immoptibox/`.

Nocedal, J. and Wright, S. (1999). *Numerical Optimization*. Springer, New York.

Oliver, D., Reynolds, A., and Liu, N. (2008). *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. Cambridge University Press, Cambridge.

Paige, C. and Saunders, M. (1982). LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM T. Math. Software*, 8(1), 43–71.

Szklarz, S., Rojas, M., and Kaleta, M. (2012). Efficient solution of the optimization problem in model-reduced gradient-based history matching. Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands, in preparation.

Vermeulen, P. and Heemink, A. (2006). Model-reduced variational data assimilation. *Mon. Weather Rev.*, 134(10), 2888–2899.

Appendix A. EXPERIMENT SETTINGS

A.1 Unconstrained minimization

The initial point $\eta_0 = 0$ was used for all methods. The parameter m, which determines the backtracking scope in the line-search strategy of the SPG method, was set to 30 when $N_\eta = 20$, and 5 when $N_\eta = 300$. Other settings relate to the stopping criteria used in the codes and are summarized in Table A.1. In the table, $\varepsilon_f, \varepsilon_g, \varepsilon_x \in (0, 1)$, and $f_k = f(\eta_k)$, $f_{k-1} = f(\eta_{k-1})$, $g_k = \nabla f(\eta_k)$, $g_0 = \nabla f(\eta_0)$

Table A.1. Stopping criteria for unconstrained optimization methods

Method	Stopping Criteria
KBFGS	$\|g_k\|_\infty \leq \max\{q * optol, optol_{abs}\}$, with $q = \max\{1, \min\{\|f_k\|, \|g_k\|_\infty\}\}$
KLBFGS	as in KBFGS
MBFGS	$\|\eta_k - \eta_{k-1}\|_\infty < \varepsilon_x(1 + \|\eta_{k-1}\|_\infty)$ or $\|g_k\|_\infty < \varepsilon_g\|(1 + \|g_0\|_\infty)$
IBFGS	$\|\eta_k - \eta_{k-1}\|_2 \leq \varepsilon_x(\varepsilon_x + \|\eta_{k-1}\|_2)$ or $\|g_k\|_\infty \leq \varepsilon_g$
SPG	$\|\eta_k - \eta_{k-1}\|_2 \leq \varepsilon_x$ or $\|f_k - f_{k-1}\| \leq \varepsilon_f$ or $\|P_\Omega(\eta_k - g_k) - \eta_k\|_\infty \leq \varepsilon_g$

In Table A.1, P_Ω in the stopping criteria for the SPG method denotes projection onto the feasible set Ω. As mentioned before, $\Omega = \mathbb{R}^n$ in the unconstrained case. Thus, in this case the stopping criteria becomes $\|g_k\|_\infty \leq \varepsilon_f$.

The following settings were used in all experiments. For KBFGS and KLBFGS, the default values were used; for MBFGS, $\varepsilon_x = 10^{-6}$, $\varepsilon_g = 10^{-6}$; for IBFGS, $\varepsilon_x = 10^{-8}$, $\varepsilon_g = 10^{-6}$; for SPG when $N_\eta = 20$, $\varepsilon_x = 10^{-4}$, $\varepsilon_f = 10^{-4}$, $\varepsilon_g = 10^{-6}$ for SPG when $N_\eta = 300$, $\varepsilon_x = 10^{-12}$, $\varepsilon_f = 10^{-12}$, $\varepsilon_g = 10^{-6}$. Default values were used for other stopping criteria such as maximum number of iterations.