A Truncated Newton Method for Solving Parameter Identification Problems*

Susana Gómez[1] *José Luis Morales*[2]

[1]IIMAS, Universidad Nacional Autónoma de México
México, D.F., México

[2]Instituto Tecnológico Autónomo de México
Río Hondo 1, Tizapán San Angel. México, D.F., México

Abstract

In this work we present a truncated Newton method able to deal with large scale bound constrained optimization problems. These problems are posed during the process of identifying parameters in mathematical models. Specifically, we propose a new termination rule for the inner iteration, that is effective in this class of applications. Preliminary numerical experimentation is presented in order to illustrate the merits of the rule.

1 Introduction

In this paper we present a numerical method able to deal with very large, non-linear, unstable, bound constrained optimization problems. These problems arise when the parameters of a mathematical model that describes certain physical phenomena, have to be identified. Generally these calibration problems have a dynamical behaviour modelled with partial differential equations.

This situation can be formulated through the following optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) = ||\phi(x) - d\,||, \qquad \phi : \mathbf{R}^n \to \mathbf{R}^m, \quad d \in \mathbf{R}^m \\ \text{subject to} & l \leq x \leq u, \quad l, x, u \in \mathbf{R}^n, \end{array} \tag{1}$$

*The first author was supported in part by the UNAM-CRAY research grant program, SC-002985, and a CONACyT-NSF research project. The second author was supported in part by a CONACyT-NSF research project.

Received december 1996, revised version accepted june 1997.

where the unknown parameters are the decision variables and the objective function is some norm of the difference between the vector of observed data d and the vector of model predictions $\phi(x)$. Thus, the computed parameters are the best in the sense that the error in the predictions is minimized. Then problem (1) is an inverse problem, which in general is badly posed in the Hadamard sense: i) there can be multiple local solutions (a situation that will not be addressed in this paper), ii) the problem is ill conditioned.

Frequently the vector of observations has a significant amount of error in the measured data. This represents a complicating factor from the numerical point of view due to the ill conditioning. In addition, the number of parameters may be quite large as a consequence of the discretization of the differential model.

Therefore, suitable numerical methods must be able to solve (1) in a stable way using reasonable amounts of computational resources. The optimization approach has shown to be successful to solve this class of problems. For example, an implementation of the limited memory quasi-Newton (LMQN) method developed by Byrd, Zhu, Lu and Nocedal [12] has been used to calibrate a hydraulic model and a seismic model (see [1], [2]).

On the other hand, truncated Newton (TN) methods are well suited to deal with optimization problems in the large scale setting. In practice, they perform well and are relatively easy to implement. Furthermore, under some conditions they are able to retain most of the nice convergence properties of original Newton's method. When the local quadratic model is ill-conditioned, a well known potential disadvantage is due to the fact that the conjugate gradient (CG) based inner iteration requires some preconditioning technique in order to accelerate its convergence. However, for the applications already described, problem-oriented preconditioners can be generated (for example using multiscale grids) to alleviate ill-conditioning. In this spirit, it is known that LMQN methods do not work well if the distribution of eigenvalues is ill, and it is not yet clear how to incorporate preconditioners in them. Finally, the stabilization properties of the CG method could be useful to deal with the stabilization problem, see for instance Nemirovskii [8] and Plato [9]. Therefore, TN methods seem worth to be explored in the context of parameter identification problems.

In this work we propose and study a TN method that uses the CG method to solve the system of linear equations defining the descent direction at each Newton step. Since in our approach the overall performance of the method strongly depends on the CG iteration, special attention has been devoted to the termination rule. A new rule has been developed to identify when the inner iteration has computed a satisfactory descent direction. The rule has been tested on two classes of problems: (a) a set of difficult academic problems, (b) a synthetic example of an inverse parameter identification problem that calibrates an aquifer model based on the flux equations.

In this paper the development of the new stopping condition for the CG iteration of a TN method is presented. The development of *ad-hoc* preconditioners and the

analysis of the stabilization properties of the method, are the subject of current research and will be reported in forthcoming papers. The paper is organized as follows: section 2 deals with basic issues in the truncated Newton method. In section 3 the test problems are described along with numerical experimentation on both sets of problems. Some remarks and comments are given in section 4.

2 Truncated Newton Methods

In essence, TN methods attempt to retain the rapid quadratic convergence of the classical Newton method while making storage and computation feasible for large scale applications. In the absence of constraints, Newton methods are iterative techniques that find local minima by minimizing local quadratic models of the objective function f in (1). Thus, at the k-th iteration, the following subproblem is solved

$$\underset{p_k}{\text{minimize}} \qquad f(x_k + p_k) \approx Q_k(p_k) = f(x_k) + g_k^T p_k + \frac{1}{2} p_k^T H_k p_k \,, \tag{2}$$

where g_k and H_k denote the gradient and Hessian, respectively, of f at x_k. Its solution p_k, which is computed by solving the system of linear equations

$$H_k p_k = -g_k, \tag{3}$$

is then used to generate the next iterate with the rule $x_{k+1} = x_k + p_k$.

In practice, however, Newton's method requires some modifications in order to be effective. First of all, the actual matrix of coefficients in the system of linear equations (3) may be some modification $\bar{H}_k$ of the true Hessian which ensures that a descent direction is always computed. Second, under certain conditions, for example when the iterates are far away from the local optimum, the system (3) may be solved only approximately. When p_k is just an approximate solution of (3) without any reference to the procedure used to compute it, the resulting method is referred to as an *inexact* Newton method. If the procedure used to compute p_k is an iterative technique, then the term *truncated* Newton is used instead. We will discuss in the following subsections some details of the implementation of a TN method that are crucial to deal with our applications.

2.1 Computing Feasible Descent Directions

We will assume that the elements of the Hessian matrix are not available. Therefore, the CG method seems appropriate to solve (3) since it does not require the Hessian explicitly. We always start the method with the initial approximation $p_k^{(0)} = 0$. Then at each iteration a new approximation is computed $p_k^{(i)}$. The computational cost involved is some linear algebra operations and a matrix-vector product $H_k v$. This product is approximated using differences of the gradient vector as follows

$$H_k v \approx \frac{g(x_k + hv) - g(x_k)}{h}, \tag{4}$$

where $h = (1 + ||x_k||)\sqrt{\epsilon_M}$, and ϵ_M is the relative machine precision.

Then, the cost of one CG iteration is dominated by the computation of $g(x_k + hv)$. In practical situations this fact has an important consequence: the overall performance of the method will strongly depend on the rule to stop the CG iteration.

In our applications, the parameters are generally subject to satisfy some bounds imposed by the physical problem. The algorithms deal with this situation as follows. At each Newton iteration, a variable at its bound is kept fixed only if the corresponding component of the gradient vector points inside the feasible region. Then an unbounded step on the free variables is computed by solving the Newton equations (3) with the preconditioned (see next subsection) CG method. During this computation, some precautions are needed to prematurely stop the CG loop, for example when an indefinite Hessian is detected. In this particular case, the implementation tests the double product $d^{(i)T} H_k d^{(i)}$ at each inner iteration; where $d^{(i)}$ is the current direction computed by the CG method. If this quantity is less than a prescribed tolerance ϵ, ($\epsilon = 10^{-10}$ in our implementation), then the procedure is terminated. In any case the stopping rules ensure that the CG iteration always finishes with a descent direction. Then a line search is performed along the descent direction.

It could happen however, that the computed direction is infeasible for a variable at its bound. To deal with this situation, before the line search is performed, the maximum allowable feasible step is determined which will be zero, if an infeasible direction has been computed. Then the active set is modified, and a new search direction will be computed based on this new active set. The line search routine is an implementation of the Moré and Thuente [4] algorithm. It uses quadratic and cubic interpolation to iteratively compute a sufficient decrease stepsize α_k. The search is terminated when α_k satisfies the following, *strong Wolfe conditions*,

$$|g(x_k + \alpha_k p_k)^T p_k| \leq \xi |g(x_k)^T p_k|, \tag{5}$$

$$f(x_k) - f(x_k + \alpha_k p_k) \geq -\mu \alpha_k p_k^T g(x_k), \tag{6}$$

where $\mu = 10^{-4}$, and $\xi = 0.9$. Then the trial point, $\bar{x}_{k+1} = x_k + \alpha_k p_k$ is computed, and the active set of bound-constraints is updated.

2.2 Preconditioning

In spite of its finite termination property, the CG method is not effective as it stands. This algorithm, as many others of its class intended to solve systems of linear equations of the form $Ax = b$, is very sensitive to the eigenvalue distribution of the coefficients

matrix A. Specifically, the CG method converges linearly with rate $(\kappa^{1/2}-1)/(\kappa^{1/2}+1)$, where κ is the spectral condition number of A. Therefore, the basic goal in preconditioning is to improve the eigenvalue distribution of A. The standard approach consists in premultiplying the original system by a non singular matrix M that is a good approximation of the inverse of A, i.e.

$$MAx = Mb, \quad \bar{A} = MA, \quad \bar{b} = Mb. \tag{7}$$

Then the new system $\bar{A}x = \bar{b}$, the preconditioned system, is solved; observe that in this case the CG method will require matrix-products of the form $MAv = Mv'$.

The implementated algorithm uses the preconditioner developed by Nash [5]. This preconditioner is based on two steps of the LMQN-BFGS method and a diagonal scaling built with information collected during the inner iteration. Therefore, the matrix M is never stored explicitly. A detailed description of the preconditioner and the scaling can be found in [5].

2.3 Stopping the CG Method

There are several theoretically supported truncation rules that give good results in practical situations. For example the following rule, due to Dembo and Steighaug [3],

$$||r_k|| \leq \eta_k ||g_k||, \quad \eta_k = \min\{c/k, ||g_k||\}, \quad c \leq 1, \tag{8}$$

where $r_k = -H_k p_k - g_k$ is the residual, and η_k is called the forcing sequence. In [3] the authors prove that the outer iteration achieves asymptotic superlinear convergence if the precision in the inner iteration is forced using (8). However in our context, this condition can be overly restrictive if, for example, the problem is ill conditioned and a good preconditioner is not available. In this case the experiments show that the number of outer iterations is low but in contrast, the corresponding of CG iterations may be unaffordable.

Other rules try to avoid an excessive number of inner iterations by limiting the rate of convergence of the outer algorithm. The quadratic stopping condition due to Nash [5], uses a measure of the quality of the local quadratic model generated by Newton's method to assess the approximation to the Newton step. The inner iteration is terminated when the following test is satisfied

$$i\left(1 - \frac{Q_k(p_k^{(i-1)})}{Q_k(p_k^{(i)})}\right) \leq 0.5, \tag{9}$$

where Q_k is defined in (2), and $p^{(i)}$ is the current approximation to the Newton direction computed by the CG method at step i. Experimentally, this condition combines

a very reasonable number of inner iterations with a potentially large number of outer iterations. These results are supported by the fact that with this rule the outer iteration is expected to have linear convergence [7].

As an alternative, we present in this paper another way to decide when to stop the CG iteration. This rule is not based on the rate of convergence, instead it estimates the quality of the approximation produced by the CG method with respect to the Newton direction. In order to do so, we suggest to compute the angle between the vectors $H_k p^{(i)}$ and $-g_k$ i.e.,

$$\cos\theta^{(i)} = \frac{-g_k^T H_k p^{(i)}}{||g_k||\,||H_k p^{(i)}||}. \tag{10}$$

Observe that this is an indirect way to estimate the alignment between the Newton direction $p_N = -H_k^{-1} g_k$ and the computed direction $p_k^{(i)}$.

In order to define the stopping condition we start with two simple observations. First we note that $1 - \cos\theta^{(i)} \to 0$ as the CG method approaches the solution p_N. Second, the Newton direction also satisfies the relation $g_k^T H_k p_N + g_k^T g_k = 0$. These observations suggest to stop the CG method when the following test is satisfied:

$$|1 - \cos\theta^{(i)}| \leq \begin{cases} \text{TOL} & \text{if} \quad k = 0, \\ |g_k^T H_k p^{(i)} - g_k^T g_k| / ||g_k||^2 & \text{otherwise,} \end{cases} \tag{11}$$

where TOL is defined as follows

$$\text{TOL} = \max\{10^{-7}, \eta_0 ||g_0||\}, \quad \eta_0 = ||g_0||. \tag{12}$$

The quantity $|1 - \cos\theta^{(i)}|$ does not tend monotonically to zero. However, our numerical experiments have shown that good directions can be computed when for example $|1 - \cos\theta^{(i)}| \leq 10^{-1}$.

3 Test Problems and Numerical Experiments

In this section we describe the two classes of problems and the numerical experimentation conducted so far.

3.1 Modelling an Aquifer in 2-D

Consider an unsteady flow in an inhomogeneous, isotropic and confined aquifer with the following partial differential equation governing the flux

$$\frac{\partial}{\partial x}\left(T(x,y)\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left((T(x,y)\frac{\partial h}{\partial y}\right)=S\frac{\partial h}{\partial t}+Q, \tag{13}$$

with the following initial and boundary conditions:

$$\begin{aligned} h(x,y,0) &= h_0(x,y), \qquad \forall \quad x,y\in\Omega, && (14)\\ h(x,y,t) &= h_1(x,y,t), \qquad \forall \quad x,y\in\delta\Omega_1, && (15)\\ T\frac{\partial h}{\partial n} &= h_2(x,y,t), \qquad \forall \quad x,y\in\delta\Omega_2, && (16) \end{aligned}$$

where h is the hydraulic head; T is the transmisivity, S is the storage coefficient; Q is the source and sink term (assumed to be known); Ω is the flow region; $\delta\Omega$, the boundary of the aquifer, is given by $\delta\Omega_1\cup\delta\Omega_2$; $\partial/\partial n$ denotes the normal derivative; finally h_0, h_1, h_2 are specified functions.

In order to exploit the aquifer it is necessary to solve (13) in $h(x,y,t)$, the head distribution at every point of the domain at any time. However, $T(x,y)$ the transmisivity parameter is not known. Fortunately, some data measurements of the heads $h^*(x_i,y_j,t_k)$ are known. Thus, this information can be used to calibrate the model by finding the transmisivity T that better reproduces these data. This can be carried out by solving the following least squares problem,

$$\begin{aligned} &\text{minimize} && f(T)=\sum_{i,j,k}\left[h(x_i,y_j,t_k;T(x_i,y_j))-h^*(x_i,y_j,t_k)\right]^2\\ &\text{subject to} && T_{min}\le T\le T_{max}. \end{aligned}$$

The interested reader can find a detailed description of this problem in [10], [11].

3.2 Academic Problems

We have chosen a standard subset of academic problems as a reference to compare the complexity of our application problem. They are considered as difficult problems and normally used for testing purposes. A detailed description of some of these problems can be found in [5]. They share some similarities, for example at the initial point their Hessian matrices have the same set of eigenvalues: $\lambda_i = i$ for $i = 1,\ldots,N$, where N is the dimension of the problem. At the solution the Hessian matrices have, except for problem SQRT, at least one eigenvalue very close to zero.

3.3 Numerical Results

All numerical experimentation was performed on an ALPHA DECSTATION in double precision with $\epsilon_M \approx 10^{-16}$. All subprograms and routines were written in FORTRAN.

quadratic				residual				angle			
iter	f-g	cg	f-g + cg	iter	f-g	cg	f-g + cg	iter	f-g	cg	f-g + cg
271	501	3, 553	4, 054	273	502	11, 658	12, 160	389	501	595	1, 096

Tab. 1: Aquifer problem. Computational costs for three different stopping conditions. Number of variables, after discretization, 78.

problem	quadratic			residual			angle		
	iter	f-g	cg	iter	f-g	cg	iter	f-g	cg
QOR	15	16	45	8	9	45	10	11	41
matrix SQRT	36	42	388	22	35	615	29	36	418
Tridiagonal	19	20	64	11	12	75	34	35	58
Trigonometric	39	84	354	39	75	3, 029	40	59	89
Calc of var 1	35	38	469	16	20	625	42	44	409
f-g + cg	1, 520			4, 540			1, 200		

Tab. 2: Academic problems. Computational cost for three different stopping conditions. Number of variables = 100.

In the experiments we tested three stopping conditions for the CG inner iteration: the rule based on the residual (8) due to Dembo and Steihaug [3], denoted by *residual*; the rule (9) due to Nash and Sofer [6] denoted by *quadratic*, and the proposed rule denoted by *angle*. A limit of 50 iterations CG per outer iteration was imposed in all cases. With respect to the outer iteration, this was stopped with the following condition

$$||g(x_k)||_\infty < 10^{-6}(1 + f(x_k)). \tag{17}$$

The numerical results are summarized in Tables 1, 2, where information is displayed as follows: (iter) is the number of Newton iterations, (f-g) is the number of function and gradient evaluations, and (cg) is the number of conjugate gradient iterations needed. Note that the actual computational cost has to be calculated by adding the number of function and gradient evaluations to the CG iterations. We recall that the cost of one CG iteration is one gradient due to the finite differences approximation to the matrix-vector product (4).

Table 1 shows the performance for the three rules with the aquifer problem, where the total cost appears in the fourth column. Table 2 displays the results for the academic problems, where the last row gives the acumulated computational cost for the five problems.

4 Some Final Remarks

Based on the limited numerical experiments presented here, we can assert that it is possible to define a practical stopping condition not based on the rate of convergence. For ill conditioned problems this rule seems to behave better than the known rules. The idea, based on geometric arguments, requires however a careful analysis in order to establish its theoretical support.

References

[1] R.M. Alvarez, S. Gómez and A. Pérez, "Towards the calibration of an aquifer model". In: *XI Int. Conf. on Computational Methods in Water Resources*, July 1996, Cancún, México.

[2] G. Chavent, F. Clement and S. Gómez, "Migration Based Travel Time Inversion of 2D Simple Structures: The Synclay Model". Accepted in *Geophysics 1996.*Also in french in Rapport de Recherche INRIA No. 2860, 1997.

[3] R.S. Dembo and T. Steihaug, "Truncated-Newton Methods for Large-Scale Unconstrained Optimization". *Math. Prog.*, 26(1983), pp. 190-212.

[4] J.J. Moré and D.J. Thuente, "Line search procedures with guaranteed sufficient decrease". *ACM Trans. Math. Software*, 20(1994), pp. 286-307.

[5] S. Nash, "Preconditioning of Truncated-Newton Methods". *SIAM J. Sci. Statist. Comput.*, 6(1985), pp. 599-616.

[6] S. Nash and A. Sofer, "Assessing a Search Direction within a Truncated Newton Method". *Oper. Research. Lett.*, 9, (1990), pp. 219-220.

[7] S. Nash and J. Nocedal, "A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large Scale Optimization". *SIAM J. Opt.*, 1(1991), pp. 358-372.

[8] A.S. Nemirovskii, "The Regularizing Properties of the Adjoint Gradient Method in Ill-Posed Problems". *U.S.S.R. Comput. Math. Phys.*, 26(1986), pp. 7-16.

[9] R. Plato, "Optimal Algorithms for Linear Ill-Posed Problems Yield Regularization Methods". *Numer. Funct. Anal. and Opt.*, 11(1990), pp. 111-118.

[10] W.W-G Yeh, "Review of Parameter Identification Procedures in Groundwater Hydrology: the Inverse Problem". *Water Resources Research*, 22(1986), pp. 95-108.

[11] Y.S. Yoon and W.W-G Yeh, "Parameter Identification in an Inhomogeneous Medium with the Finite-Element Method". *Society of Petroleum Engineers Journal*, No (1976), pp. 217-226.

[12] C. Zhu, R.H. Byrd, P. Lu and J. Nocedal, *LBFGS-B: Fortran Subroutines for Large-Scale Bound Constrained Optimization*. Report NAM-11, EECS Department, Northwestern University, 1994.