Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems

A. K. Alekseev[a] and I. M. Navon[b]

a Department of Aerodynamics and Heat Transfer, RSC, ENERGIA, Korolev (Kaliningrad), Moscow Region 141070, Russian Federation

b School of Computational Science and Department of Mathematics, Florida State University, Tallahassee, FL 32306-4120, USA

Abstract

We compare the performance of several robust large-scale minimization algorithms for the unconstrained minimization of an ill-posed inverse problem. The parabolized Navier-Stokes equations model was used for adjoint parameter estimation.

The methods compared consist of two versions of the nonlinear conjugate gradient method (CG), Quasi-Newton (BFGS), the limited memory Quasi-Newton (L-BFGS) [15], Truncated Newton method [19, 20] and a new hybrid algorithm proposed by Morales and Nocedal [16].

For all the methods employed and tested the gradient of the cost function is obtained via an adjoint method. A detailed description of the algorithmic form of minimization algorithms employed in the minimization comparison is provided.

The hybrid method emerged as the best performer for an adequate choice of parameters controlling the number of L-BFGS and Truncated Newton iterations to be interlaced.

1. Introduction

The following specific issues characterize the inverse Computational Fluid Dynamics (CFD) problems posed in the variational statement:

- High CPU time required for the single cost functional computation;
- The computation of the gradient of the cost functional usually is performed using the adjoint model, which requires the same computational effort as the direct model;
- The instability (due to ill posedness) prohibits using Newton type algorithms without prior explicit regularization due to the Hessian of the cost functional being indefinite.

The nonlinear conjugate gradient method is widely used for ill-posed inverse problems [2-3, 14] because it provides the regularization implicitly by neglecting non-dominant Hessian eigenvectors. The large CPU time required for a single cost functional computation justifies the high importance attached to choosing the most efficient large-scale unconstrained optimization method. From this perspective, we will compare the performance of the nonlinear conjugate gradient method along with several quasi-Newton and truncated Newton large-scale unconstrained minimization methods [15, 19-21] and a new hybrid method (see Morales and Nocedal [16]), The problem is an ill-posed inverse computational fluid dynamics parameter identification of entrance boundary parameters from measurements taken in downstream flow-field sections.

The paper is organized as follows. In Section 2, the ill-posed test problem of parameter estimation is presented along with the adjoint derivation required for obtaining the gradient of the cost function with respect to the control parameters. Section 3 consists of a detailed description of the algorithmic form of the large-scale unconstrained minimization methods tested. The numerical tests and their results comparing performance of the above mentioned minimization methods are presented in Section 4. Finally, discussion and conclusion are presented in Section 5.

2. The Test problem

We consider the identification of unknown parameters $(f_\infty(Y) = (\rho(Y), U(Y), V(Y), T(Y)))$ at the entrance boundary (Fig. 1) from measurements taken in a flow-field section $f^{\exp}(X_m, Y_m)$ as a test for the inverse computational fluid dynamics (CFD) problem. The direct measurement of flow-field parameters in zones of interest may be either difficult or impossible to carry out due to different reasons: a lack of access, high heat flux or pressures etc. For example, measurements of parameters in a rocket engine chamber may be very difficult. For the same case, measurements taken in the jet past the nozzle may be carried out without difficulties. Thus, the estimation of inflow parameters from the down flow measurements is a realistic test problem. This problem may be formulated as that of minimization of a cost functional (measuring discrepancy between measured and calculated parameters) with respect to a set of inflow parameters.

The algorithm consists of the flow-field calculation (direct model), the discrepancy gradient (gradient of the cost functional) computation using both forward and adjoint models and an unconstrained optimization method.

The problem has all the features of an ill-posed inverse CFD problem but can be solved relatively fast when using the 2-D parabolized Navier-Stokes equations approximation.

2. 1 The Direct Problem

The two-dimensional parabolized Navier-Stokes equations are used here in a form similar to that presented in Refs. [1, 2]. The flow (Figure 1) is laminar and supersonic along the X- coordinate. These equations describe an under-expanded jet in the supersonic flow.

$$\frac{\partial(\rho U)}{\partial X} + \frac{\partial(\rho V)}{\partial Y} = 0 \tag{1}$$

$$U\frac{\partial U}{\partial X} + V\frac{\partial U}{\partial Y} + \frac{1}{\rho}\frac{\partial P}{\partial X} = \frac{1}{\mathrm{Re}\,\rho}\frac{\partial^2 U}{\partial Y^2} \tag{2}$$

$$U\frac{\partial V}{\partial X} + V\frac{\partial V}{\partial Y} + \frac{1}{\rho}\frac{\partial P}{\partial Y} = \frac{4}{3\rho\,\mathrm{Re}}\frac{\partial^2 V}{\partial Y^2} \tag{3}$$

$$U\frac{\partial e}{\partial X} + V\frac{\partial e}{\partial Y} + (\gamma-1)e\left(\frac{\partial U}{\partial X} + \frac{\partial V}{\partial Y}\right) = \frac{1}{\rho}\left(\frac{\gamma}{\mathrm{Re\,Pr}}\frac{\partial^2 e}{\partial Y^2} + \frac{4}{3\,\mathrm{Re}}\left(\frac{\partial U}{\partial Y}\right)^2\right) \tag{4}$$

$P=\rho RT;\ e=C_v T=R/(\gamma-1)T;\ (X,Y)\in\Omega=(0<X<X_{max};\ 0<Y<1);$

The entrance boundary (A (X=0), Fig. 1) conditions follow:

$e(0,Y)=e_\infty(Y);\ \rho(0,Y)=\rho_\infty(Y));\ U(0,Y)=U_\infty(Y);\ V(0,Y)=V_\infty(Y);$ (5)

The outflow boundary conditions

$$\partial f/\partial Y=0$$

are used on *B, D (Y=0, Y=1).*

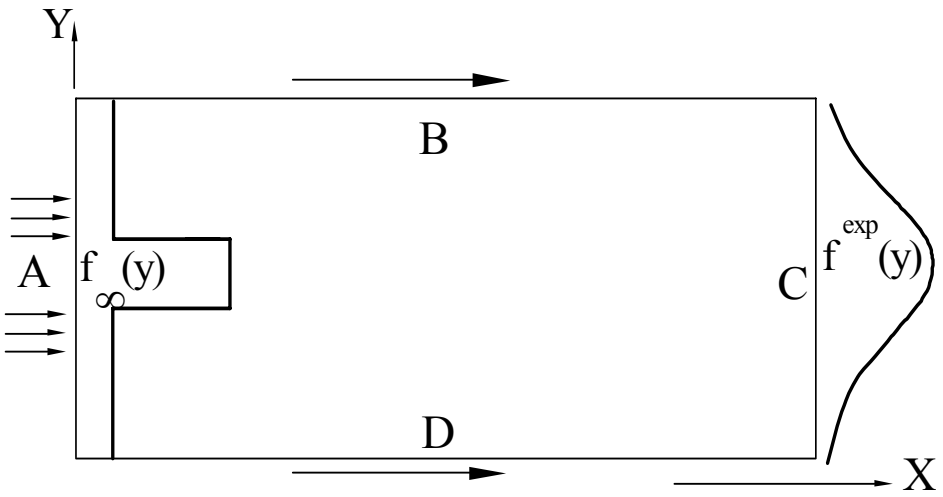

Fig. 1 Flow sketch. A - entrance boundary, C - section of measurements (outflow boundary).

The flow parameters at some set of flow-field points $f^{exp}(X_m, Ym)$ are available. The values $f_\infty(Y)=(\rho(Y), U(Y), V(Y), e(Y))$ on the boundary A are unknown and have to be determined. For this purpose, we minimize the discrepancy between computed and measured values $f^{exp}(X, Y)$ for a set of measurement points.

$$\varepsilon(f_\infty(Y)) = \int_\Omega \left(f^{\exp}(X,Y) - f(X,Y)\right)^2 \delta(X-X_m)\delta(Y-Y_m)dXdY \tag{6}$$

We use the following notations:

C_v- specific volume heat capacity;

e - specific energy, C_vT;
f - flow parameters (ρ,U,V,e);
h-enthalpy;
h_0-total enthalpy
h_x, h_y- spatial steps along X and Y;
M - Mach number;
N_t-number of time steps, N_x – number of spatial nodes along X, N – number of spatial nodes along Y;
L- Lagrangian;
P -pressure ;
Pr - Prandtl number ($Pr=\mu C_v/\lambda$);
R - gas constant;
$\mathrm{Re} = \frac{\rho_\infty U_\infty Y_{\max}}{\mu_\infty}$ -Reynolds number;
T-temperature;
U- velocity component along X;
V- velocity component along Y;
X, Y –coordinates;
δ -Dirac's delta function;
γ - specific heat ratio;
ε - cost functional;
μ -viscosity;
λ - thermal conductivity;
ρ - density;
τ -temporal step;
$\Psi_\rho, \Psi_U, \Psi_V, \Psi_e$ – the *adjoint* variables;
Ω- domain of calculation;

Subscripts:

∞ entrance boundary parameters; an- analytical solution; *corr*- corrected error;

est – estimated point; *exact*- exact solution; k- number of spatial mesh node along Y;

n-number of step along X; *sup*- bound of inherent error; x-component of truncation error connected with Taylor expansion in coordinate X;

t- component of truncation error connected with Taylor expansion in time.

2.2. The Adjoint problem

A fast calculation of the gradient is crucial for implementing the optimization methods tested herein, due to the high CPU time computational cost of the discrepancy calculation as well as due to the relatively large number of control variables. The solution of the adjoint problem is the fastest way for calculating the discrepancy gradient when the number of control parameters is relatively large. The adjoint problem corresponding to Eqs. (1-6) follows:

$$U\frac{\partial\Psi_\rho}{\partial X}+V\frac{\partial\Psi_\rho}{\partial Y}+(\gamma-1)\frac{\partial(\Psi_V e/\rho)}{\partial Y}+(\gamma-1)\frac{\partial(\Psi_U e/\rho)}{\partial X}-$$
$$-\frac{\gamma-1}{\rho}\left(\frac{\partial e}{\partial Y}\Psi_V+\frac{\partial e}{\partial X}\Psi_U\right)+\left(\frac{1}{\rho^2}\frac{\partial P}{\partial X}-\frac{1}{\rho^2\,\mathrm{Re}}\frac{\partial^2 U}{\partial Y^2}\right)\Psi_U+\frac{1}{\rho^2}\left(\frac{\partial P}{\partial Y}-\frac{4}{3\,\mathrm{Re}}\frac{\partial^2 V}{\partial Y^2}\right)\Psi_V-$$
$$-\frac{1}{\rho^2}\left(\frac{\gamma}{\mathrm{RePr}}\frac{\partial^2 e}{\partial Y^2}+\frac{4}{3\mathrm{Re}}\left(\frac{\partial U}{\partial Y}\right)^2\right)\Psi_e+2\left(\rho^{\exp}(X,Y)-\rho(X,Y)\right)\delta(X-X_m)\delta(Y-Y_m)=0\quad,\qquad(7)$$

$$U\frac{\partial\Psi_U}{\partial X}+\frac{\partial(\Psi_U V)}{\partial Y}+\rho\frac{\partial\Psi_\rho}{\partial X}-\left(\frac{\partial V}{\partial X}\Psi_V+\frac{\partial e}{\partial X}\Psi_e\right)+\frac{\partial}{\partial X}\left(\frac{P}{\rho}\Psi_e\right)+$$
$$+\frac{\partial^2}{\partial Y^2}\left(\frac{1}{\rho\,Re}\Psi_U\right)-\frac{\partial}{\partial Y}\left(\frac{8}{3\,Re}\frac{\partial U}{\partial Y}\Psi_e\right)+$$
$$+2\left(U^{\exp}(X,Y)-U(X,Y)\right)\delta(X-X_m)\delta(Y-Y_m)=0\,,\qquad(8)$$

$$\frac{\partial(U\Psi_V)}{\partial X}+V\frac{\partial\Psi_V}{\partial Y}-\left(\frac{\partial U}{\partial Y}\Psi_U+\frac{\partial e}{\partial Y}\Psi_e\right)+$$
$$+\rho\frac{\partial\Psi_\rho}{\partial Y}+\frac{\partial}{\partial Y}\left(\frac{P}{\rho}\Psi_e\right)+\frac{4}{3\,Re}\frac{\partial^2}{\partial Y^2}\left(\frac{\Psi_V}{\rho}\right)+$$
$$+2\left(V^{\exp}(X,Y)-V(X,Y)\right)\delta(X-X_m)\delta(Y-Y_m)=0,\qquad(9)$$

$$\frac{\partial(U\Psi_e)}{\partial X}+\frac{\partial(V\Psi_e)}{\partial Y}-\frac{\gamma-1}{\rho}\left(\frac{\partial\rho}{\partial Y}\Psi_V+\frac{\partial\rho}{\partial X}\Psi_U\right)-(\gamma-1)\left(\frac{\partial U}{\partial X}+\frac{\partial V}{\partial Y}\right)\Psi_e+$$
$$+(\gamma-1)\frac{\partial\Psi_V}{\partial Y}+(\gamma-1)\frac{\partial\Psi_U}{\partial X}+\frac{\gamma}{\mathrm{Re}\;\mathrm{Pr}}\frac{\partial^2}{\partial Y^2}\left(\frac{\Psi_e}{\rho}\right)+$$
$$+2\left(e^{\exp}(X,Y)-e(X,Y)\right)\delta(X-X_m)\delta(Y-Y_m)=0.\qquad(10)$$

The boundary conditions on C ($X=X_{max}$) are:

$$\Psi_f\Big|^{X=X_{\max}}=0;$$

The following boundary condition is used at B and D (Y=0; Y=1):

$$\frac{\partial\Psi_f}{\partial Y}=0.\qquad(11)$$

The discrepancy gradient is determined by the flow parameters and the adjoint variables:

$$\partial\varepsilon / \partial e_\infty(Y) = \Psi_e U + (\gamma - 1)\Psi_U ,$$
$$\partial\varepsilon / \partial \rho_\infty(Y) = \Psi_\rho U + (\lambda - 1)\Psi_U e / \rho ,$$
$$\partial\varepsilon / \partial U_\infty(Y) = \Psi_U U + \rho\Psi_\rho + (\gamma - 1)\Psi_e e , \qquad (12)$$
$$\partial\varepsilon / \partial V_\infty(Y) = \Psi_V U .$$

The flow-field (forward problem, (1-4)) is computed using a finite difference method [1, 2] marching along the X. The method is of first order accuracy in X and second order accuracy in the Y variable. The pressure gradient for supersonic flow is computed from the energy and density. The same algorithm (and the same grid) is used for solving the adjoint problem solution, however the integration is performed in the reverse direction (beginning at the $X=X_{max}$). The grid consists of 50-100 nodes along the Y direction and 50-200 nodes along the X direction. The flow parameters on the entrance boundary $f_\infty(Y_i)=f_i$ $(i=1,N)$ serve as the set of control variables. The input data $f^{exp}(X_m,Y_i)$ $(i=1,...,N)$ are obtained at the outflow section from a preliminary computation. The flow parameters are: the external flow Mach number, $M=5$ (Mach number of the jet is about 3) and the Reynolds number Re is in the range of 10^3 - 10^4. Several tests were performed for an inviscid flow ($Re= 10^8$).
For a systematic analysis of convergence rate for numerical solution techniques that require the gradient of discrete cost function see Hager [13].

3. Description of the Minimization Algorithms

The spatial distribution of parameters on the entrance boundary (A) is determined by applying and comparing the following large-scale optimization methods:

- Conjugate gradients [23, 24, 25,28,29], (nonlinear conjugate-gradient versions),
- Quasi-Newton (BFGS), [23,6-9]),
- Limited memory Quasi-Newton (L-BFGS), [11]
- Truncated Newton method (T-N), [19,20]
- A new hybrid algorithm proposed by Morales and Nocedal [16] that consists of a class of optimization methods that interlace iterations of the limited memory BFGS method (L-BFGS) and a Truncated-Newton method (T-N) in such a way, that the information collected by one type of iteration improves the performance of the other. For algorithmic details about the hybrid method, in particular efficient preconditioning of the conjugate gradient method, see also Morales and Nocedal [17] and [18]. This new algorithm was studied and tested in [4,5] and was demonstrated to be the best performing algorithm.

In this work, we test implementations of the L-BFGS version VA15 of [15] in the Harwell library, the T-N method described by Nash [19, 20] and the hybrid method of Morales and Nocedal [16]. A brief description of the major components of each algorithm is given below. The nonlinear conjugate gradient algorithm CONMIN used in this study is described as well. [19]. The code of Shanno and Phua [25] allows also for the implementation of the Quasi-Newton BFGS method.

The subroutine CONMIN incorporates two nonlinear optimization methods, a nonlinear conjugate gradient algorithm and a variable metric algorithm, with the choice of method being left to the user. The nonlinear conjugate gradient algorithm is the Beale restarted memoryless variable metric algorithm . This method requires approximately 7n double precision words of working storage to be provided by the user. The variable metric method is the BFGS algorithm

with initial scaling documented in Shanno and Phua [25], and required approximately $n^2/2 + \frac{11n}{2}$ double-precision words of working storage.

For a function of n variables we use the following notations: $f_k=f(\mathbf{x}_k)$ denotes a generic cost function where $\mathbf{x}_k$ is the n component vector at the k^{th} iteration. $\mathbf{g}_k=g(\mathbf{x}_k)=\nabla f_k$ is the gradient vector of size n, and $H_k=\nabla^2 f_k$ is the $n\times n$ symmetric Hessian matrix of the second partial derivatives of f with respect to the coordinates. In all the algorithms the new iterate is calculated from

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad ; \tag{13}$$

where $\mathbf{p}_k$ is the descent direction vector, and α_k is the step length. Iterations are terminated when

$$\|\mathbf{g}_k\| < 10^{-6} \max(1, \|\mathbf{x}_k\|). \tag{14}$$

The necessary changes in the programs were made to ensure that all three algorithms use the same termination criterion. In addition, the three methods use the same line search that is based on cubic interpolation, and is subject to the so-called *strong Wolfe conditions*, (see [11] and [12])

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) &\geq -\mu \alpha_k \mathbf{p}_k^T \mathbf{g}_k \\ \left|\mathbf{g}(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{p}_k\right| &\leq \eta \left|\mathbf{g}_k^T \mathbf{p}_k\right| \end{aligned} \tag{15}$$

where $0 < \mu < \eta < 1$, and the superscript T denotes transpose.
The values of the parameters μ and η used were typically 10^{-4} and 0.1 , respectively.

3.1. The nonlinear conjugate gradient algorithm.

Conjugate gradient (CG) uses the analytic derivatives of f, defined by g_k. A step along the current negative gradient vector is taken in the first iteration; successive directions are constructed so that they form a set of mutually conjugate vectors with respect to the Hessian. At each step, the new iterate is calculated from eq. (13) and the search directions are expressed recursively as

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1} \quad . \tag{16}$$

Calculation of β_k with the algorithm incorporated in CONMIN used for nonlinear conjugate gradient is described by Shanno [25].

CONMIN has important advantages such as automatic restarts along a carefully chosen direction (Nocedal and Wright [24]). The Hessian-vector products in the nonlinear CG code were done via finite differencing of gradients.

If one considers the memoryless BFGS formula:

$$H_{k+1} = (I - \frac{s_k y_k^T}{y_k^T s_k})(I - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k} \tag{17}$$

In conjunction with an exact line search for which

$\nabla f_k^T p_k = 0$ for all k, then we obtain

$$p_{k+1} = -H_{k+1}\nabla f_{k+1} = -\nabla f_{k+1} + \frac{\nabla f_k^T y_k}{y_k^T p_k} p_k$$

which is the Hestenes –Stiefel conjugate gradient method and when $\nabla f_{k+1}^T p_k = 0$ the Hestenes-Stiefel formula reduces to the Polak Ribiere formula.

$$\beta_{k+1}^{PR} = \frac{\nabla f_k^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} p_k$$

Automatic restarting is used to preserve a linear convergence rate. For restart iterations, the steplength α_k=1 is used. On the other hand, for no restart iterations,

$$\alpha_{k+1} = \frac{\alpha_k g_k^T p_k}{g_{k-1}^T p_{k-1}} \quad . \qquad (18)$$

3.2 The C-G Descent method

Hager and Zhang (2005) developed a new nonlinear conjugate gradient algorithm for unconstrained optimization problem.

The C-G iterates assume the form

$x_{k+1} = x_k + \alpha_k d_k$,
where the stepsize α_k is positive and where the directions d_k are generated by the rule
$d_{k+1} = -g_{k+1} + \beta_k^N d_k \qquad d_0 = -g_0$,

where

$$\beta_k^N = \frac{1}{d_k^T y_k}(y_k - 2d_k \frac{\|y_k^2\|}{d_k^T y_k})^T g_{k+1} \qquad (19)$$

Here $\|.\|$ is the Euclidean norm, $g_k = \nabla f(x_k)^T$, and $y_k = g_{k+1} - g_k$;

If f is a quadratic and α_k is chosen to achieve the exact minimum of f in the direction d_k, then

$d_k^T g_{k+1} = 0$,

and the formula for β_k^N reduces to the familiar Hestenes–Stiefel scheme.

The advantages of the new conjugate-gradient scheme are described in [28] and [29].
For a wide range of test problems, this method proved to be today's fastest code as illustrated in [28], [29].

3.3. BFGS quasi-Newton Method

A central idea underlying quasi-Newton methods is to use an approximation of the inverse Hessian. In quasi-Newton methods,
instead of the true Hessian, an initial matrix B_0 is chosen (usually B_0 = I) which is subsequently updated by an update formula. Given two points x_k and x_{k+1} , we define $g_k = \nabla f(x_k)$) and $g_{k+1} = \nabla f(x_{k+1})$).
Let $s_k = x_{k+1} - x_k$

$$y_k = g_{k+1} - g_k \tag{20}$$

$B_{k+1} s_k = y_k$. (for the derivation from a quadratic model of the objective function, see Nocedal and Wright [23] Section 8.1) a formula referred to as the secant equation.

The approximate Hessian B_k is used in place of the true Hessian.

Given displacement s_k and change of gradients y_k, the secant equation requires that the symmetric and positive definite matrix B_{k+1} maps s_k into y_k. This is possible only if s_k and y_k satisfy the curvature condition

$$s_k^T y_k > 0. \tag{21}$$

To determine B_{k+1} uniquely the additional condition is imposed that among all symmetric matrices satisfying the secant equation B_{k+1} is in a sense closest to the current matrix B_k i.e. we solve the problem

$$\min_B \|B - B_k\| \tag{22}$$

Subject to $B = B^T$ and $Bs_k = y_k$ where s_k and y_k satisfy $s_k^T y_k > 0$ and B_k is symmetric and positive definite.

Using a weighted Frobenius norm the unique solution of (20) as shown in Nocedal and Wright [23] is the Davidon [6]- Fletcher Powell (DFP) updating formula originally proposed by Davidon ([6] and popularized by Fletcher and Powell.[9]

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T \tag{23}$$

With $\gamma_k = \dfrac{1}{y_k^T s_k}$.

Instead of imposing conditions on the Hessian approximations B_k , we impose similar conditions on their inverses H_k. The updated approximation H_{k+1} must be symmetric and positive definite, and must satisfy the secant equation and now written as

$$H_{k+1} y_k = s_k.$$

The condition of closeness to H_k is now specified by

$$\min_H \|H - H_k\| \qquad (24)$$

subject to $H = H^T \quad Hy_k = s_k$

using a weighted Frobenius norm.

The unique solution H_{K+1} to (23) is given by

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \qquad (25)$$

Where

$$\rho_k = \frac{1}{y_k^T s_k}.$$

Numerical experiments have shown that BFGS formula's performance is superior over the DFP formula. Hence, BFGS is often preferred over DFP. As Nocedal and Wright [24] note , the DFP and BFGS updating formulae are dual of each other , one being obtained from the other via interchanges $s \leftrightarrow y$ and $B \leftrightarrow H$.

3.4. Limited memory BFGS algorithm.

The L-BFGS method is an adaptation of the BFGS method to large problems, achieved by changing the Hessian update of the latter. (See [22] and [15] for details). Thus, in the BFGS, {Dennis & More 1977 [7]}{Dennis & Schnabel 1983 [8] } eq. (3) is used with an approximation $\tilde{H}_k$ to the inverse Hessian which is updated by

$$\tilde{H}_{k+1} = V_k^T \tilde{H}_k V_k + \rho_k \mathbf{s}_k \mathbf{s}_k^T \quad , \qquad (26)$$

where $V_k = I - \rho_k \mathbf{y}_k \mathbf{s}_k^T$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_{k,}$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_{k,}$, $\boldsymbol{\rho}_k = 1/(\mathbf{y}_k^T \mathbf{s}_k)$, and I is the identity matrix. The search direction is given by

$$\mathbf{p}_{k+1} = -\tilde{H}_{k+1} \mathbf{g}_{k+1} \quad . \qquad (27)$$

In the L-BFGS, instead of forming the matrices $\widetilde{H}_k$ explicitly (which would require a large memory for a large problem) one only stores the vectors $\mathbf{s}_k$ and $\mathbf{y}_k$ obtained in the last m iterations which define $\widetilde{H}_k$ implicitly; a cyclical procedure is used to retain the latest vectors and discard the oldest ones. Thus, after the first m iterations, eq. (18) becomes

$$\begin{aligned}
&\tilde{H}_{k+1} = (V_k^T \cdots V_{k-m}^T)\tilde{H}_{k-1}^0 (V_{k-m} \cdots V_k) \\
&+\rho_{k-m}(V_k^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T \\
&\times (V_{k-m-1}^T \cdots V_k) \\
&+\rho_{k-m-1}(V_k^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T \\
&\times (V_{k-m+2}^T \cdots V_k) \\
&\vdots \\
&+\rho_k s_k s_k^T,
\end{aligned} \tag{28}$$

with the initial guess $\widetilde{H}_{k+1}^0$ which is the diagonal matrix

$$\widetilde{H}_{k+1}^0 = \frac{\mathbf{y}_{k+1}^T \mathbf{s}_{k+1}}{\mathbf{y}_{k+1}^T \mathbf{y}_{k+1}} I \quad . \tag{29}$$

Many previous studies have shown that typically $3 \le m \le 7$, where $m > 7$ does not improve the performance of L-BFGS. Here we used a value of m=5.

3.5. The truncated Newton algorithm (Hessian-free T-N)

In the T-N, a search direction is computed by finding an *approximate* solution to the Newton equations,

$$H_k \mathbf{p}_k = -\mathbf{g}_k \quad . \tag{30}$$

The use of an approximate search direction is justified because an exact solution of the Newton equation at a point far from the minimum is unnecessary and computationally wasteful in the framework of a basic descent method. Thus, for each *outer* iteration [eq. (13)], there is an *inner* iteration loop making use of the conjugate gradient method that computes this approximate direction, $\mathbf{p}_k$. The conjugate gradient inner algorithm is preconditioned by a scaled two-step limited memory BFGS method with Powell's restarting strategy used to reset the preconditioner periodically. A detailed description of the preconditioner may be found in Ref.[19] The Hessian vector product $H_k\mathbf{v}$ for a given $\mathbf{v}$ required by the inner conjugate gradient algorithm is obtained by a finite difference approximation,

$$H_k \mathbf{v} \approx [\mathbf{g}(x_k + h\mathbf{v}) - \mathbf{g}(\mathbf{x}_k)]/h \quad . \tag{31}$$

A major issue is how to adequately choose h {Wang et al [26]}; in this work we use $h = \varepsilon^{1/2}(1+\|x_k\|)$, where ε is the machine precision and $\|\bullet\|$ denotes the Euclidian norm. The inner algorithm is terminated using the quadratic truncation test, which monitors a sufficient decrease of the quadratic model $q_k = \mathbf{p}_k^T H_k \mathbf{p}_k / 2 + \mathbf{p}_k^T \mathbf{g}_k$:

$$(1 - q_k^{i-1} / q_k^i) \le c_q / i \tag{32}$$

where i is the counter for the inner iteration and c_q is a constant, $0 < c_q \le 1$. For additional details see {Nash 1984 [20]}).

An important qualification is the following. Since gradient evaluations are the dominant cost in inverse problems, a truncated Newton method that computes Hessian-vector products by a finite-difference of gradients may compute many gradient evaluations per iteration. This problem may become even more severe in 3-D plus time inverse problems due to increase in dimensionality .

As such, forward+adjoint calls is the most relevant metric to use in comparing it to the other methods, as we provide in present research (communication of anonymous reviewer).

3.6. The hybrid method.

The hybrid method consists of interlacing in a dynamical way the L-BFGS method with the T-N discussed above. The limited memory matrix H_m plays a dual role of preconditioning the inner conjugate gradient iteration in the T-N method as well as providing the initial approximation of the inverse of the Hessian matrix in the L-BFGS iteration. In this way information gathered by each method improves the performance of the other without increasing the computational cost.

The hybrid method alleviates the shortcomings of both L-BFGS and T-N. The T-N requires fewer iterations than the L-BFGS to reach the solution, while the computational cost of each iteration is high and the curvature information gathered in the process is lost after the iteration has been completed. L-BFGS, on the other hand, performs inexpensive iterations, with poorer curvature information – a process that can become slow on ill-conditioned problems.

Algorithmically, implementation of the hybrid-enriched method includes an advanced preconditioning of the CG iteration, a dynamic strategy to determine the lengths of the L-BFGS and T-N cycles, as well as a standard stopping test for the inner CG iteration. In the enriched method that will be tested below, *k1* steps of the L-BFGS method are alternated with *k2* steps of the T-N method, where the choice of *k1* and *k2* will be discussed below. We illustrate this as

$$[\text{k1} * (\text{L-BFGS}) \rightarrow \text{k2} * (\text{T-N(PCG)}). \rightarrow \tilde{H}(m), \text{repeat}], \tag{33}$$

where $\tilde{H}(m)$ is a limited memory matrix that approximates the inverse of the Hessian matrix (Eq. 20), and m denotes the number of correction pairs stored. The L-BFGS cycle starts from the initial unit or a weighted unit matrix, $\tilde{H}(m)$ is updated using the most recent m pairs, and the matrix obtained at the last L-BFGS cycle is used to precondition the first of the *k2* T-N iterations. In the remaining *k2-1* iterations, the limited memory matrix $\tilde{H}(m)$ is updated using information generated by the inner preconditioned CG (PCG) iteration and it is used to

precondition the next T-N iteration. At the end of the T-N steps, the most current $\widetilde{H}(m)$ matrix is used as the initial matrix in a new cycle of L-BFGS steps.

The enriched algorithm is described below:

Choose a starting point **x**, the memory parameter *m*, and an initial choice of the
length k1 of the L-BFGS cycle; set method ←'L-BFGS'; first ←.true.
While a convergence test is not satisfied:
Repeat:
compute **p**: call PCG (*B,* **p, g**, method, status, *maxcg*); (either a PCG or an L-BFGS step)
compute α: call LNSRCH (α);
compute **x**+ = **x** + α**p**;
store **s** = **x**+ - **x** and **y** = **g**+ -**g**; call ADJUST (k1, k2*,* α, method, status, first, *maxcg*);
End repeat
End while.

The vectors **s** and **y** are used to update the limited memory matrix $\widetilde{H}(m)$. The parameter 'method' in the compute **p** step of the pseudo-code can assume the values 'L-BFGS' or 'T-N', and 'maxcg' determines the maximum number of CG iterations allowed. This procedure returns a search direction **p**, and the value of 'status', which is used by procedure ADJUST to modify the lengths k1 and k2 of the L-BFGS and T-N cycles. The procedure LNSRCH is a standard backtracking line search routine enforcing the Wolfe conditions (eq. 5). A more detailed description of procedures PCG and ADJUST is provided in Morales and Nocedal [18] .

4. **Numerical Tests**

The computations have the following algorithmic structure: the forward problem (1-5) is solved for parameters f (Y_∞) and the flow-field values of $\rho(X, Y)$, $U(X, Y)$, $V(X, Y)$, $T(X, Y)$ are stored. The discrepancy (cost functional) $\varepsilon^n(f)$ is calculated, the adjoint problem (7-10) is solved and the gradient of the cost $\nabla\varepsilon^n$ is calculated from (12). Then, the new control parameters are calculated using the chosen optimizer. The optimization algorithm uses the following prescribed termination criterion: $\|\nabla\varepsilon\| < 10^{-6}\max\left(1, \|f_\infty\|\right)$.

Figures 2-6 represent typical results of solution of this problem by different minimization methods and are compared with the exact data.

Figure 2 presents the result of inflow temperature estimation from the error-free outflow data (within computer accuracy). Figure 3 presents the inflow density illustrating the development of the instability (the constant density being equal to the unity at the exact solution). This result may be considered as an illustration of the problem's ill posedness. Figures 4 and 5 provide the total density distribution in the flow-field for the exact solution and the result of the calculation. Figure 6 illustrates the adjoint density field.

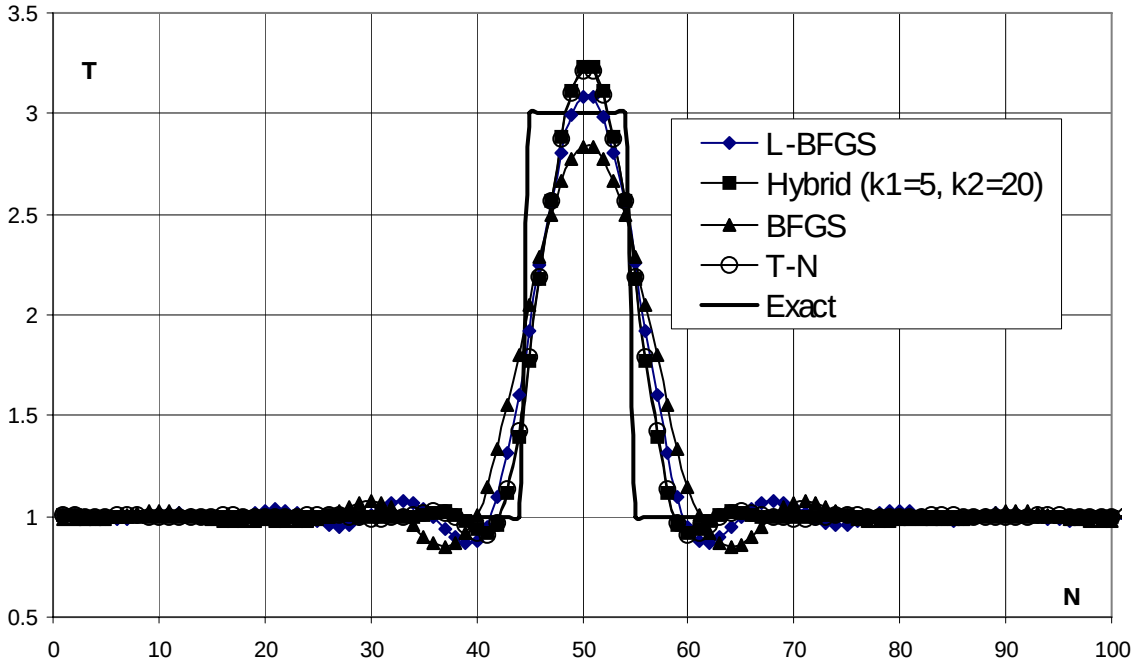

Fig. 2 Inflow temperature calculation

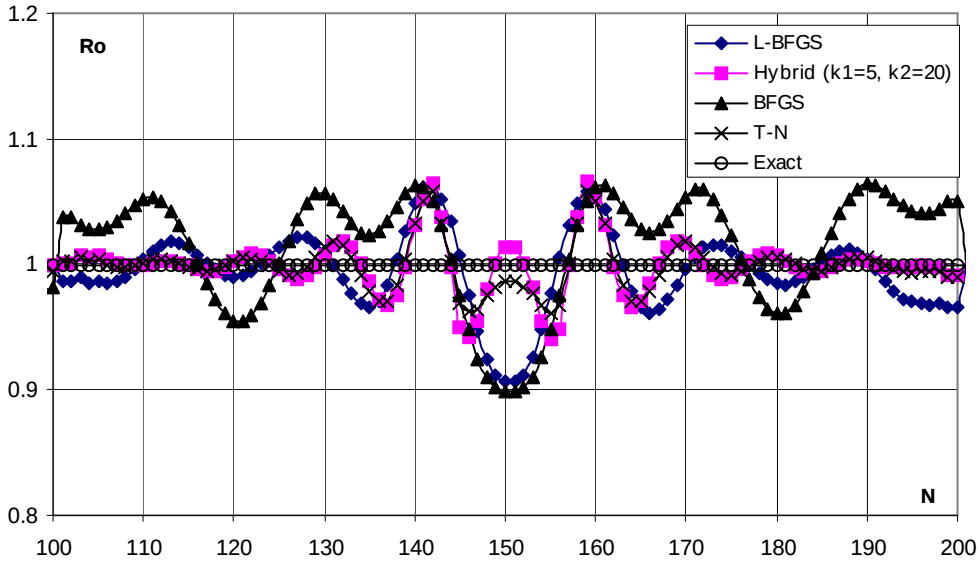

Fig. 3 Inflow density calculation

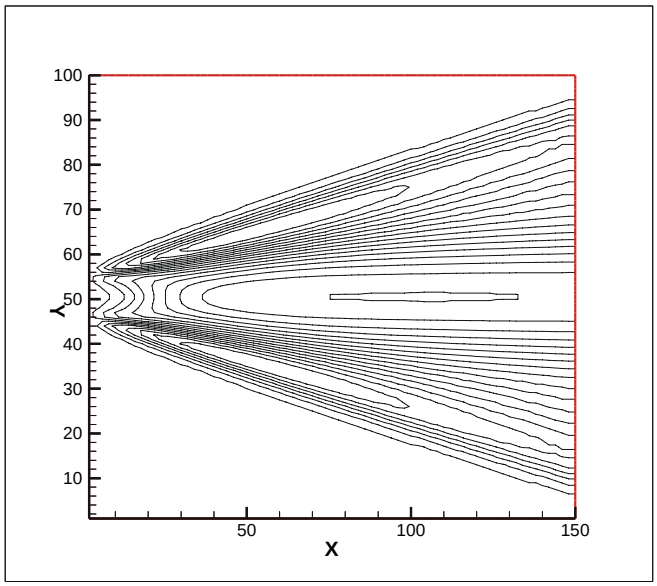

Fig. 4. Target density field

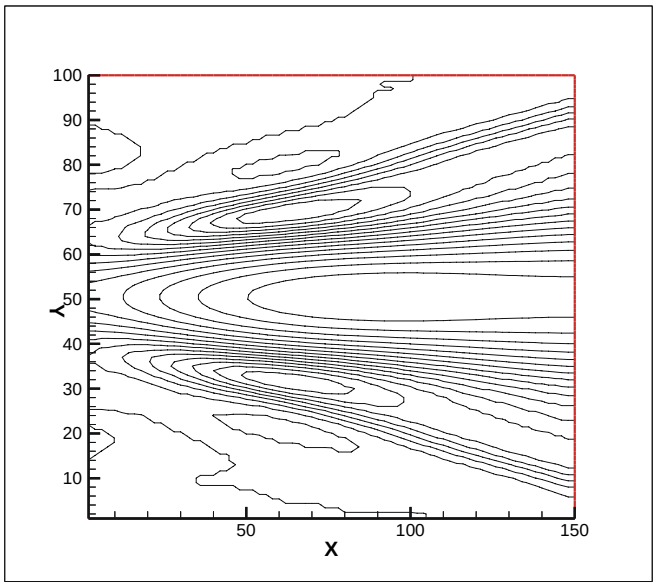

Fig. 5. Computed density field

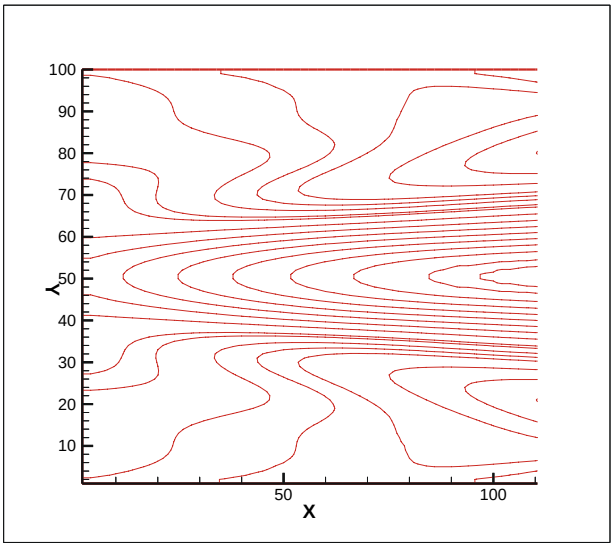

Fig. 6. Adjoint density field

Figure 7 presents the Hessian of the cost spectrum for this problem near the exact solution (1) and the spectrum of the uniform flow (2). The horizontal axis presents the number of the eigenvalues in decreasing order of their magnitude while the vertical axis presents their magnitude normalized with respect to that of the largest eigenvalue. Most eigenvalues are very close to zero, thus prohibiting the use of the standard Newton method for this problem.

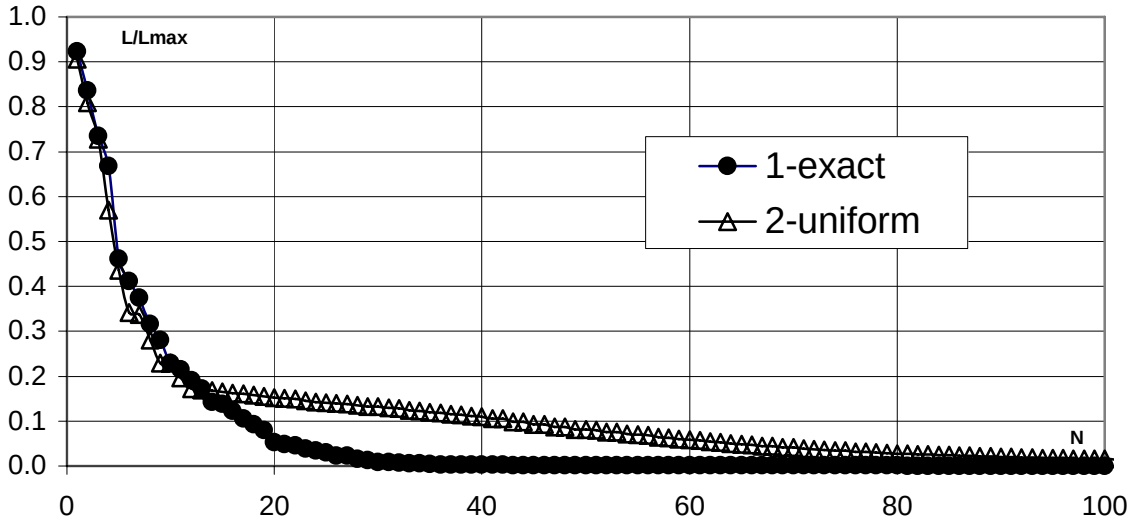

Fig. 7. Hessian eigenvalues ordered by magnitude.

Figures 8 and 9 present a comparison of different minimization methods applied to a viscous flow (Reynolds number equals 10^3). The history of optimization is presented as the dependence of the logarithm of the discrepancy vs. the number of direct +adjoint problem calls (proportional to CPU time). Figure 8 presents the results demonstrated by CG ([25] and [28] options), BFGS, L-BFGS and T-N. The T-N and L-BFGS are implemented here in the framework of the hybrid algorithm (by choosing either L-BFGS calls k1=0 or T-N calls k2=0 respectively). BFGS exhibits the best convergence rate during the first few iterations but then stops converging quickly. Another problem with this method is its lack of robustness: very often a suitable (determined by trial and error) initial guess should be chosen in order for this method to perform adequately. The hybrid method was tested also on this problem by selecting a combination of L-BFGS calls (k1) and T-N calls (k2). The hybrid method performances (for different k1 and k2) compared to those of T-N and L-BFGS, are presented in Figure 9.

Figures 10-13 present results of another test (for inviscid flow). Figure 10 represents the history of minimization iterations for the CG, BFGS, L-BFGS and T-N unconstrained minimization methods.

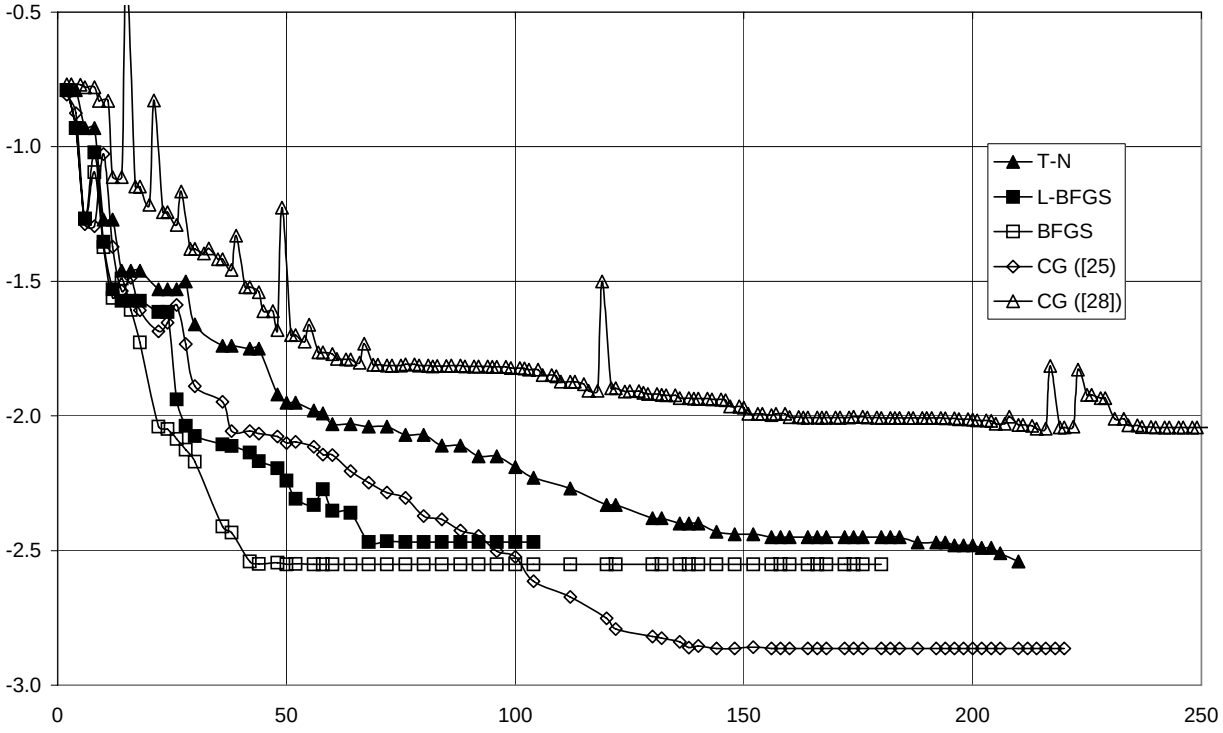

Fig. 8. History the optimization (logarithm of discrepancy) vs. the number of forward +adjoint problem calls, CG [25], CG [28], BFGS, L-BFGS and T-N.

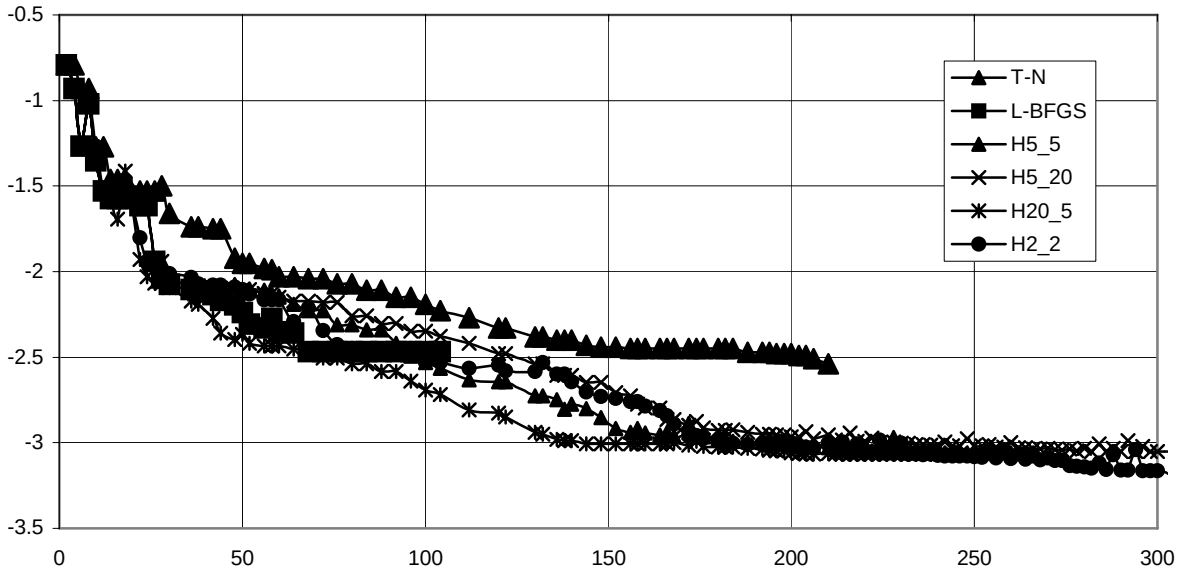

Fig. 9. History the optimization (logarithm of discrepancy) vs. the number of forward +adjoint problem calls, T-N, L-BFGS and Hybrid

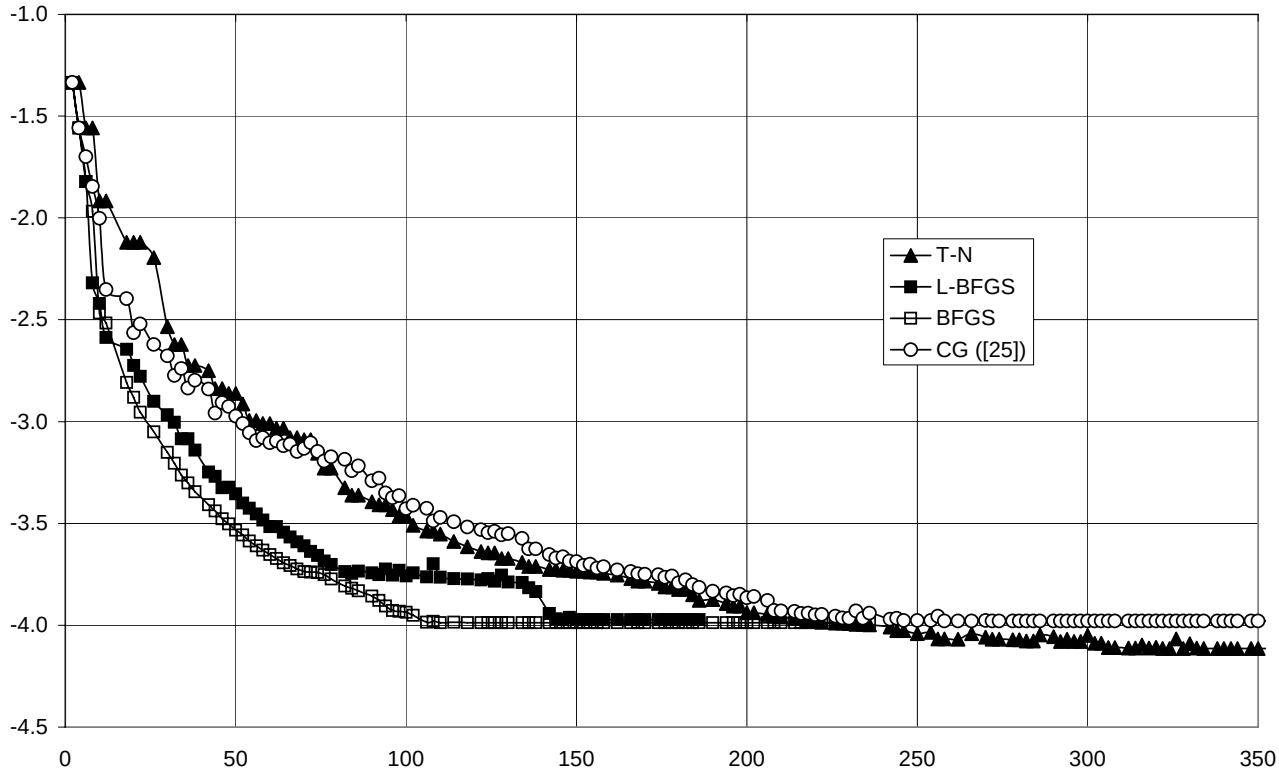

Fig. 10. The comparison of T-N, L-BFGS, CG and BFGS (discrepancy vs. direct +adjoint calls)

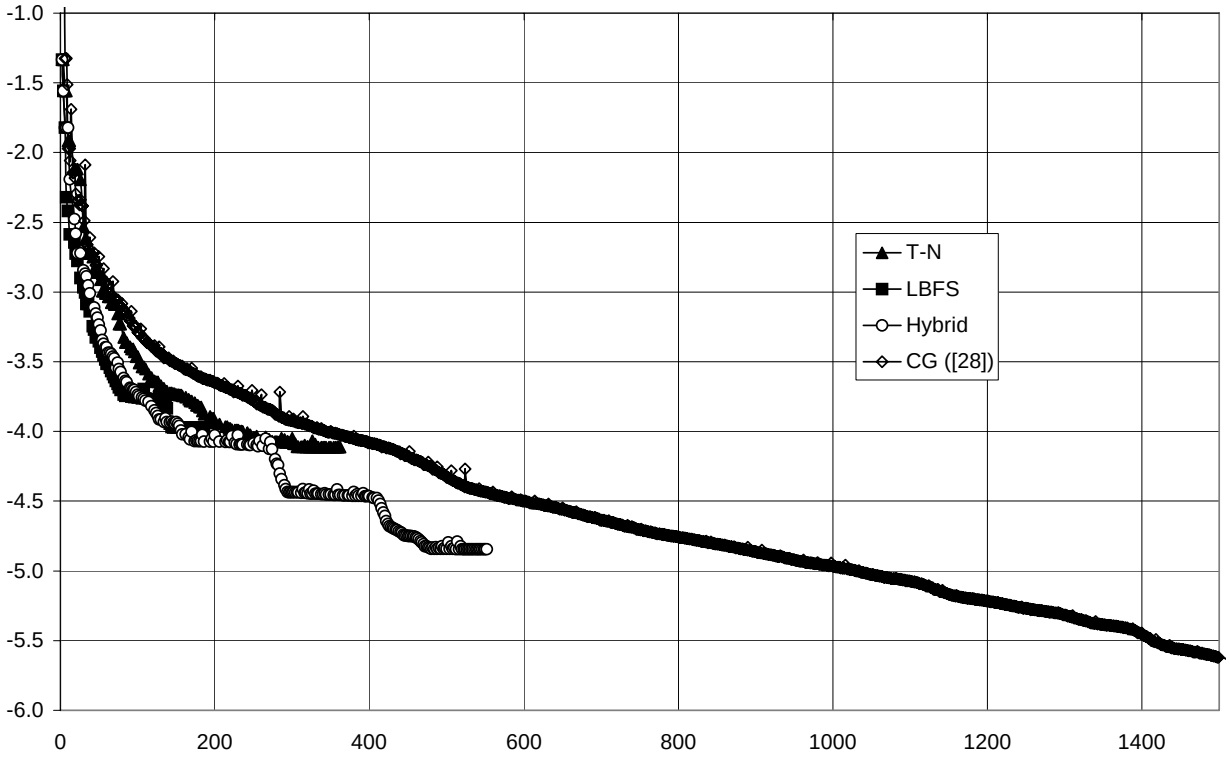

Fig. 11. The comparison of T-N, L-BFGS ,Hybrid (k1=5, k2=20), and CG [27} *(discrepancy vs. direct +adjoint calls)*

Figure 11 shows a comparison between T-N, L-BFGS and the hybrid algorithms where we also have plotted the cost functional versus the number *of direct+adjoint* calls. The comparison of Figures 10 and 11 shows that the hybrid method achieves the best results from the viewpoint of both quality and speed due to the possibility of tune-up of the parameters k1 and k2.

Figs. 12-13 present a comparison of results obtained for the considered minimization methods versus the exact result.

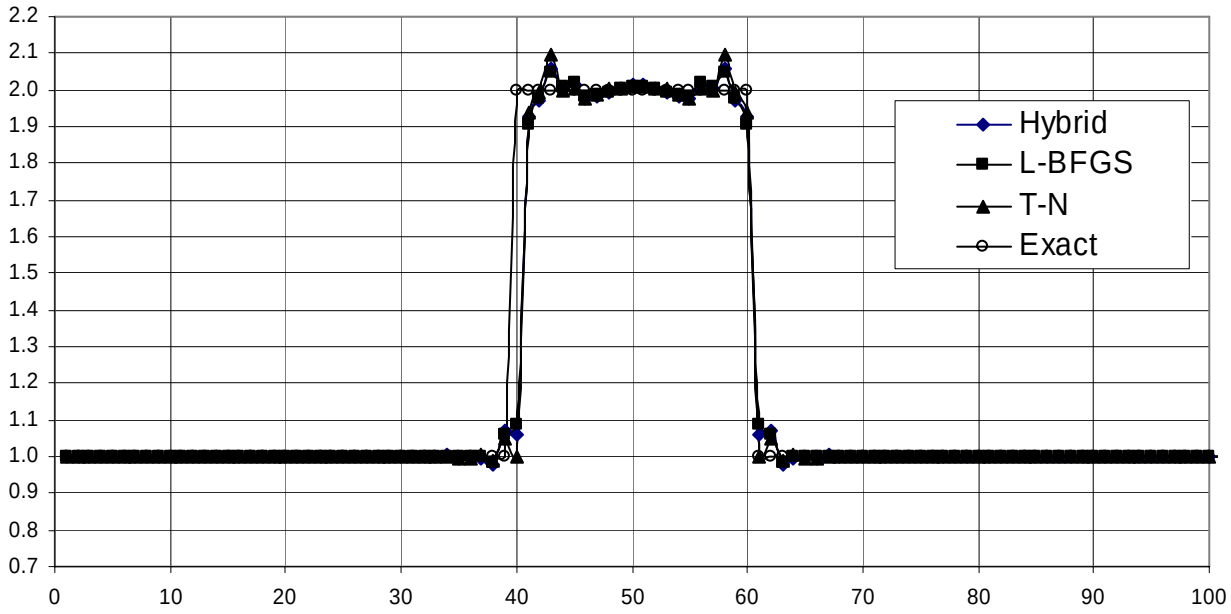

Fig. 12. The inflow temperature

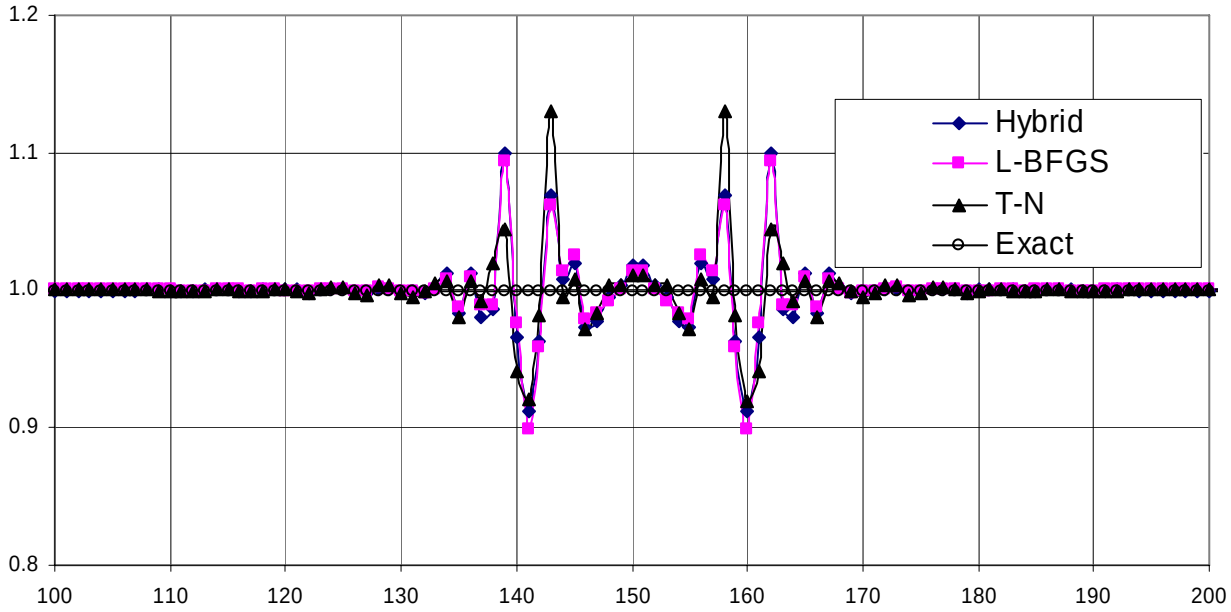

Fig. 13. Inflow density

The structure of calculation time in terms of number of direct and adjoint problem calls and consumed CPU time is presented in Tables 1 and 2 consequently. The CPU time corresponds to Celeron (800 MHz) processor and Windows-98 operational system. The specifics of present tests are the high computational burden of direct and adjoint problems in comparison with other operations (Hessian generation and inversion, linear search etc.) that consume about 2% of total computational time. This is connected with relatively low dimensionality of control parameters (400) and high expenses of direct and adjoint problems. The adjoint problem call is less time consuming if compare with direct one due to linearity (lack of additional iterations).

Table 1.

Method	Direct problem calls	Adjoint problem calls	Percentage of direct problem CPU time	Percentage of adjoint problem CPU time	Other operations
LBFGS	93	93	59.5%	37.9%	2.6%
T-N	182	180	59.9%	38.0%	2.1%
Hybrid	250	250	59.7%	38.1%	2.2%
C-G[25]	176	175	59.5%	38.1%	2.4%
C-G[28]	*1562*	*771*	*76.8%*	*23.1%*	*0.4%*
BFGS	120	116	59.5%	38.5%	2.0%

Table 2

Method	Direct problem calls	Adjoint problem calls	Number of inner CG iterations	direct problem CPU time, sec	adjoint problem CPU time, sec	Total time, sec
LBFGS	93	93	-	82.11	52.30	138.01
T-N	182	180	46	160.89	102.06	268.60
Hybrid	250	250	48	221.49	141.35	371.0
C-G [25]	176	175	-	158.7	101.6	266.7
C-G [28]	*1562*	*771*	-	*1450.2*	*436.2*	*1894.5*
BFGS	120	116	-	105.9	68.6	177.9

5. Discussion and Conclusions

The problem of inflow parameters estimation from the outflow measurements is an ill-posed one. A study of the spectrum of the Hessian of the discrepancy (cost) with respect to the control variables (Figure 7) confirms the problem's ill posedness. The Newton method is expected to be largely unstable due to large number of Hessian eigenvalues that are close to zero. This is related to the irreversible losses of information in dissipation and "gradient catastrophe", see for example Figure 14 (where we see the impact of disturbance magnitude (pressure ratio) on the quality of inflow parameter estimation).

According to the theory of ill-posed problems, these processes should engender instability. Some oscillations are indeed detectable in the numerical calculations (see Figures. 3 and 13). Nevertheless, they are of lesser size than expected. The possible reason may lie in the numerical viscosity of the forward and adjoint solvers. As a result, the approximation of the highly oscillating gradient is violated and the optimization breaks down before the significant instability develops.

Another source of stability may be caused by the general properties of gradient-based methods. The steepest descent and conjugate gradients methods are known to possess regularization properties [3], [14]. These properties are connected with a search in the subspace of the dominant Hessian eigenvectors (corresponding to maximal eigenvalues). The discrepancy gradient may be presented as the action of the Hessian by the distance to the exact solution.

$$\nabla\varepsilon(x^n) = -H\,\Delta x^n \tag{34}$$

Here the superscript n denotes the minimization iteration count.

For example, the steepest descent method has a form $x^{n+1} - x^n = -\tau\nabla\varepsilon\left(x^n\right)$. It may be recast in the form (x^* being the exact solution): $x^{n+1} - x^n - x^* = -\tau\nabla\varepsilon\left(x^n\right) - x^*$; $x^{n+1} - x^* = x^n - x^* - \tau\nabla\varepsilon\left(x^n\right); -\Delta x^{n+1} = -\Delta x^n - \tau\nabla\varepsilon\left(x^n\right) = -\Delta x^n + \tau H\Delta x^n = -\left(E - \tau H\right)\Delta x^n$;

And finally

$$\Delta x^{n+1} = \left(E - \tau H\right)\Delta x^n \tag{35}$$

Here E is the unit matrix. If the initial guess Δx^0 is expanded over Hessian eigenvectors ($\Delta X_0 = C_j U_j$, where λ_j are the eigenvalues), the components that are connected with maximum eigenvalues (dominant or leading vectors) will be represented in the gradient with maximum weights. These components of the initial guess will be maximally reduced during iterations and will be absent from the final solution.

$$\Delta X^n = C^j U^j (1-\tau\lambda_j)^n \quad \tau \sim b/\lambda_{\max}; \quad 0<b<1; \tag{36}$$

On the other hand, the components of the initial guess Δx^0 connected with small eigenvalues do not participate in the iterations. Thus the search along the gradient (or some combination of gradients under different iterations) means the search is conducted in the subspace of the Hessian dominant eigenvectors. The subspace of eigenvectors with the small eigenvalues is implicitly neglected, thus providing for the regularization effect. In practice, the convergence is fast during the first iterations and then slows-down after a relatively small number of iterations, whose number is possibly close to the number of Hessian dominant eigenvectors.

For the present problem, the methods under the consideration are found to provide a much faster convergence rate in comparison with the nonlinear conjugate gradient method CONMIN and a similar stability. The other conjugate gradient method tested [28], [29] performs extremely well for the inviscid case attaining the best local minimum and warrants future in-depth research. This may be caused by the same mechanism of self-regularization as for the gradient-based methods. Thus, the methods considered in this research display applicability for the inverse problem solution using iterative regularization.

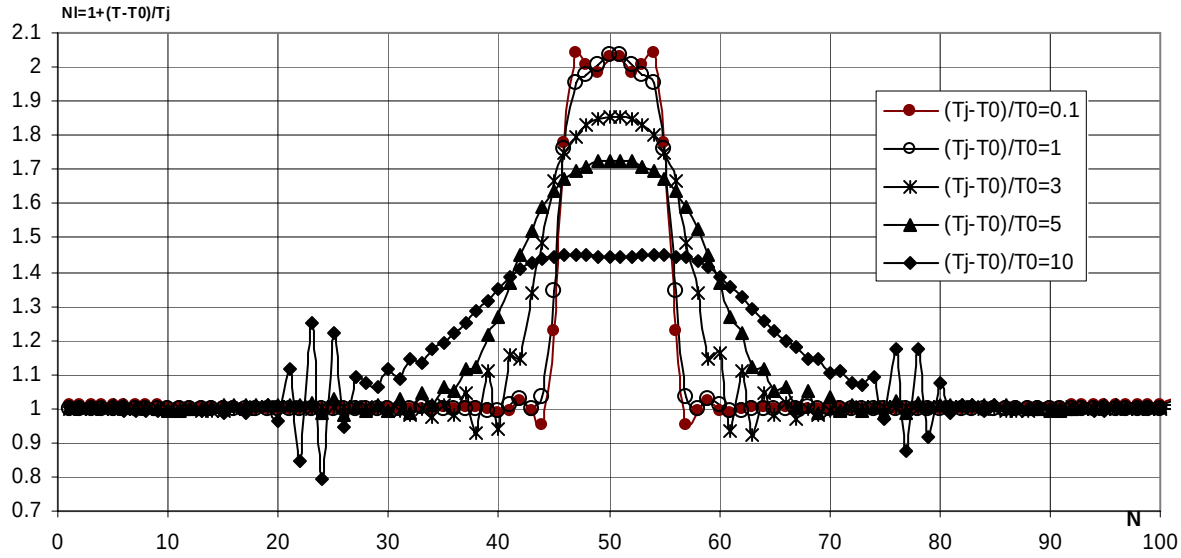

Fig. 14. The quality of solution as a function of the disturbance magnitude (inviscid flow)

The robust large-scale unconstrained minimization methods considered (T-N, L-BFGS and hybrid) were found to be applicable for the inverse problem solution without requiring any special regularization. From this viewpoint these methods exhibit a similarity to the method of nonlinear conjugate gradients while exhibiting a better performance. The BFGS method may be effectively used if a small range of convergence is required. The L-BFGS method provided both fast convergence and a good quality of results for our case. The Truncated Newton method provided a good final quality of optimization while exhibiting a relatively slower rate of convergence. The version of the conjugate gradient method of Hager [28, 29] demonstrated moderate results for viscous flow and excellent results for inviscid flow, attaining the best local minimum, albeit requiring a rather large number of iterations.

Fig. 9 shows also the impact of tuning the k1 and k2 parameters in the hybrid algorithm. A suitable tuning, which is obviously case dependent, permits the hybrid method to achieve in our case the best performance amongst large-scale unconstrained optimization methods tested.

The numerical results obtained for our test case demonstrate that the new hybrid method (also referred to as the enriched method [15]) suitably tuned should be considered a serious competitor to both the Truncated Newton and L-BFGS methods, especially since it is known (see e.g. [20]) that Newton-type methods are more effective than L-BFGS on ill-conditioned problems.

Acknowledgements

The authors acknowledge the support from the School of Computational Science, Florida State University. The expert comments of two anonymous reviewers and the suggestions of the journal Editor-in-Chief , Dr William Hager, sizably improved the presentation and content of the paper.

The second author acknowledges the support from the NSF grant number ATM-0201808 managed by Dr. Linda Pang whom he would like to thank for her support.

References

1. A. K. Alekseev, "On Estimation of entrance boundary parameters from downstream measurements using adjoint approach", Int. J. for Numerical Methods in Fluids, v. 36, pp. 971-982, 2001.

2. A. K. Alekseev and I. M. Navon, "The analysis of an ill-posed problem using multiscale resolution and second order adjoint techniques", Computer Methods in Appl. Mech. and Eng., v.190, issue 15-17, pp. 1937-1953, 2001.

3. O. M. Alifanov, E.A Artyukhin. and S.V. Rumyantsev, "Extreme Methods for Solving Ill-Posed Problems with Applications to Inverse Heat Transfer Problems", Begell House Inc. Publishers, New York, NY, 1996.

4. D. N. Daescu and I. M. Navon, "An Analysis of a Hybrid Optimization method for variational data assimilation", International Journal of Computational Fluid Dynamics, Vol. 17 (4), pp. 299-306, 2003.

5. B. Das, H. Meirovitch, and I. M. Navon, "Performance of Enriched Methods for Large Scale Unconstrained Optimization as applied to Models of Proteins" Journal of Computational Chemistry. Vol. 24, Number 10, pp. 1222-1231, 2003.

6. W.C. Davidon, Variable metric method for minimization, SIAM Journal on Optimization, Vol 1, pp1-17, 1991.

7. Dennis, J.E., Jr.; More, J.J., "Quasi-Newton Methods, Motivation and Theory", SIAM Rev, Vol.19, pp. 46- 89, 1977.

8. Dennis, J.E., Jr. and Schnabel, R.B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall: Englewood Cliffs, NJ 1983, 378 pp.

9. R. Fletcher and M.J.D. Powell, A rapidly convergent descent method for minimization, *Comput. J.*, Vol **6**, pp. 163–168, 1963.

10. Gilbert, J.C. and Nocedal, J., "Global Convergence Properties of Conjugate Gradient Methods", SIAM Journal on Optimization, Vol. 2, pp. 21-42, 1992

11. Gilbert, J.C. On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization. SIAM Journal on Optimization Vol 7 (3), pp 780-813, 1997.

12. P.E Gill, Murray, W. Report SOL 79-15, Department of Operation Research, Stanford University, Stanford, CA ,1979.

13. Hager W.W**., "**Runge-Kutta methods in optimal control and the transformed adjoint system ", Numerische Mathematik, Vol 87 (2): 247-282, 2000.

14. Per Christian Hansen, "Rank Deficient and Discrete Ill-posed problems." SIAM, Philadelphia, 247pp. 1998.

15. D.C. Liu, and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Minimization", Mathematical Programming, Vol. 45, pp. 503-528, 1989.

16. J. L. Morales and J. Nocedal, "Enriched Methods for Large-Scale Unconstrained Optimization", Computational Optimization and Applications, Vol. 21, pp. 143–154, 2002.

17. J.L. Morales, and J. Nocedal, "Automatic preconditioning by limited memory quasi-Newton updating ", SIAM J. Optim. Vol.10, No.4, pp.1079 –1096, 2000.

18. J.L. Morales, and J. Nocedal, "Algorithm PREQN: FORTRAN subroutines for preconditioning the conjugate gradient method", ACM Transactions on Mathematical Software. Vol. 27, pp. 83-91, 2001.

19. S.G. Nash, "Preconditioning of truncated Newton methods," SIAM J. Sci. Stat. Comput. Vol. 6, pp. 599-616, 1985.

20. S.G. Nash, “Newton-type minimization via the Lanczos method,” SIAM J. Numer. Anal.” Vol. 21, pp. 770-788, 1984.
21. S.G. Nash and J. Nocedal, “A Numerical study of the limited memory BFGS method and the truncated-Newton method for large-scale optimization,” SIAM J. Optim., Vol. 1, pp. 358-372, 1991.
22 J. Nocedal, "Updating quasi-Newton matrices with limited storage," Math. Comp. 35 (1980) 773--782.
23. J. Nocedal, “Theory of algorithms for unconstrained minimization”, Acta Numerica, Vol 1, pp 199-242, 1992.
24. J. Nocedal and S.J. Wright, “Numerical Optimization”, Springer Verlag, 656pp, 1999.
25. D.F. Shanno and K.H. Phua, “Remark on algorithm 500. Minimization of unconstrained multivariate functions”, ACM Transactions of Mathematical Software, Vol. 6, pp. 618-622, 1980.
26. D.F Shanno, “Conjugate gradient methods with inexact searches”. Mathematics of Operations Research, Vol.3, pp. 244-256, 1978.
27. Z. Wang, I.M. Navon, X. Zou and F.X. LeDimet, , “A Truncated –Newton Optimization Algorithm in Meteorology Applications with Analytic Hessian/vector Products”, Comput Opt Appl, Vol. 4, pp. 241- 262, 1995.
28. W.W. Hager, H. Zhang, A new conjugate gradient method with guaranteed descent and efficient line search, SIAM J. OPTIM., Vol. 16, No. 1, pp. 170–192, 2005
29. W. W. Hager , H. Zhang, Algorithm 851: CG_DESCENT, A conjugate gradient method with guaranteed descent, ACM Transactions on Mathematical Software, 32, pp. 113-137, (2006).

Figure Captions

Fig. 1 Flow sketch. A - entrance boundary, C - section of measurements (outflow boundary).

Fig. 2 Inflow temperature calculation.

Fig. 3 Inflow density calculation.

Fig. 4. Target density field.

Fig. 5. Computed density field.

Fig. 6. Adjoint density field.

Fig. 7. Hessian eigenvalues (ordered).

Fig. 8. History the optimization (logarithm of discrepancy) vs. the number of forward +adjoint problem calls for CG, BFGS, T-N and L-BFGS.

Fig. 9. History the optimization (logarithm of discrepancy) vs. the number of forward +adjoint problem calls for T-N, L-BFGS and Hybrid.

Fig. 10. The comparison of T-N, L-BFGS, CG and BFGS (discrepancy, vs. direct +adjoint calls)

Fig. 11. The comparison of T-N, L-BFGS and Hybrid (k1=5, k2=20) (discrepancy, vs. direct +adjoint calls)

Fig. 12. The inflow temperature

Fig. 13. Inflow density

Fig. 14. The quality of solution depending on the disturbance magnitude (inviscid flow)

Keywords

1. Large scale minimization methods
2. Inverse problem
3. Adjoint method
4. Ill-posedness
5. Parameter estimation