Optimization Methods & Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

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

A. K. Alekseev , I. M. Navon & J. L. Steward

To cite this article: A. K. Alekseev , I. M. Navon & J. L. Steward (2009) Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems, Optimization Methods & Software, 24:1, 63-87, DOI: 10.1080/10556780802370746

To link to this article: https://doi.org/10.1080/10556780802370746

Published online: 04 Mar 2011.

Optimization Methods & Software
Vol. 24, No. 1, February 2009, 63–87

Taylor & Francis
Taylor & Francis Group

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

A.K. Alekseev[a], I.M. Navon[b]* and J.L. Steward[b]

[a]*Department of Aerodynamics and Heat Transfer, RSC, ENERGIA, Korolev (Kaliningrad), Moscow Region, Russian Federation;* [b]*Department of Scientific Computing, Florida State University, Tallahassee, FL, USA*

(*Received 22 July 2007; revised version received 25 July 2008*)

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 equation model was used for adjoint parameter estimation.

The methods compared consist of three versions of nonlinear conjugate-gradient (CG) method, quasi-Newton Broyden–Fletcher–Goldfarb–Shanno (BFGS), the limited-memory quasi-Newton (L-BFGS) [D.C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale minimization*, Math. Program. 45 (1989), pp. 503–528], truncated Newton (T-N) method [S.G. Nash, *Preconditioning of truncated Newton methods*, SIAM J. Sci. Stat. Comput. 6 (1985), pp. 599–616, S.G. Nash, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal. 21 (1984), pp. 770–788] and a new hybrid algorithm proposed by Morales and Nocedal [J.L. Morales and J. Nocedal, *Enriched methods for large-scale unconstrained optimization*, Comput. Optim. Appl. 21 (2002), pp. 143–154].

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.

For the inviscid case, the CG-descent method of Hager [W.W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and efficient line search*, SIAM J. Optim. 16 (1) (2005), pp. 170–192] performed the best followed closely by the hybrid method [J.L. Morales and J. Nocedal, *Enriched methods for large-scale unconstrained optimization*, Comput. Optim. Appl. 21 (2002), pp. 143–154], while in the viscous case, the hybrid method emerged as the best performed followed by CG [D.F. Shanno and K.H. Phua, *Remark on algorithm 500. Minimization of unconstrained multivariate functions*, ACM Trans. Math. Softw. 6 (1980), pp. 618–622] and CG-descent [W.W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and efficient line search*, SIAM J. Optim. 16 (1) (2005), pp. 170–192]. This required an adequate choice of parameters in the CG-descent method as well as controlling the number of L-BFGS and T-N iterations to be interlaced in the hybrid method.

Keywords: large-scale minimization methods; inverse problems; adjoint parameter estimation; ill-posed problems

AMS Subject Classification: 90C90; 90C30; 49J20; 47A52

*Corresponding author. Email: navon@scs.fsu.edu

ISSN 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/10556780802370746
http://www.informaworld.com

1. Introduction

The following specific issues characterize inverse computational fluid dynamics (CFD) problems posed in the variational sense:

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

The nonlinear conjugate-gradient (CG) method is widely used for ill-posed inverse problems [3,4,20] because it provides regularization implicitly by neglecting nondominant 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 CG method along with several quasi-Newton and truncated Newton (T-N) large-scale unconstrained minimization methods [22,26–28] and a new hybrid method [23]. The problem is an ill-posed inverse CFD parameter identification of entrance boundary parameters from measurements taken in downstream flow-field sections. A similar study addressing computational experience with several limited-memory quasi-Newton and TN methods for data assimilation with the shallow water equation model using the 1990s state-of-the-art optimization methods is described in [35].

The paper is organized as follows. In Section 2, the ill-posed parameter estimation test problem 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 $f_\infty(Y) = (\rho(Y), U(Y), V(Y), T(Y))$ (see definitions below) at the entrance boundary (Figure 1) from measurements taken in a flow-field section $f^{\exp}(X_m, Y_m)$ as a test inverse 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, if not impossible, due to the extreme environment there. For the same case, measurements taken in the jet past the nozzle may be carried out

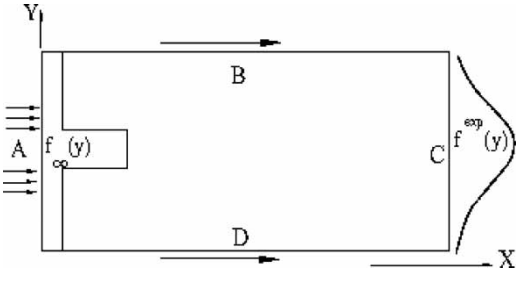

Figure 1. Flow sketch. A – entrance boundary; C – section of measurements (outflow boundary).

without difficulties. Thus, the estimation of inflow parameters from downflow measurements is a realistic test problem. This problem may be formulated as a 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 quickly when using the two-dimensional parabolized Navier–Stokes equation approximation.

2.1 *The direct problem*

The two-dimensional parabolized Navier–Stokes equations are used here in a form similar to that presented in [2,3]. The flow (Figure 1) is laminar and supersonic along the X-coordinate. These equations describe an under-expanded jet in supersonic flow.

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

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

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

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

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

The entrance boundary (A, $(X = 0)$, Figure 1) conditions follow:

$$e(0, Y) = e_\infty(Y); \quad \rho(0, Y) = \rho_\infty(Y); \quad U(0, Y) = U_\infty(Y); \quad V(0, Y) = V_\infty(Y). \tag{5}$$

The outflow boundary conditions $\partial f/\partial y = 0$ are used on B and D ($Y = 0$, $Y = 1$).

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

$$\varepsilon(f_\infty(Y)) = \sum_{m=1}^{M}\int_\Omega (f^{\exp}(X, Y) - f(X, Y))^2\delta(X - X_m)\delta(Y - Y_m)\,\mathrm{d}X\,\mathrm{d}Y. \tag{6}$$

Notation

C_v	specific volume heat capacity
e	specific energy, $C_v T$
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
Re	$\dfrac{\rho_\infty U_\infty Y_{\max}}{\mu_\infty}$ – Reynolds number
T	temperature
U	velocity component along X
V	velocity component along Y
X, Y	coordinates
γ	specific heat ratio
δ	Dirac's delta function
ε	cost functional
λ	thermal conductivity
μ	viscosity
ρ	density
τ	temporal step
$\Psi_\rho, \Psi_U, \Psi_V, \Psi_e$	the adjoint variables
Ω	domain of calculation

Subscripts

∞	entrance boundary parameters
corr	corrected error
est	estimated point
exact	exact solution
k	number of spatial mesh node along Y
n	number of steps along X
sup	bound of inherent error
t	component of truncation error connected with Taylor expansion in time
x	component of truncation error connected with Taylor expansion in coordinate X

We consider the initial boundary problem for parabolized Navier–Stokes equations (1–5), describing supersonic viscous flow evolving along X from $X = 0$. However, we have our experimental information at the downflow points. We may consider the inverse problem as the one having initial conditions at the outflow section $X_{\max}$ and transferring it to the inflow section. Let us consider its properties.

By substituting $\partial\rho/\partial X = -(\rho/U)(\partial U/\partial X + F$ from Equation (1) and $\partial e/\partial X = -((\gamma - 1)e/U)(\partial U/\partial X) + F_1$ from Equation (4) to (2), we get

$$\frac{\partial U}{\partial X}(U - \gamma RT/U) = \frac{1}{\mathrm{Re}\,\rho}\frac{\partial^2 U}{\partial Y^2} + F_2.$$

Here F, F_1 and F_2 are the remaining terms. This equation is similar to the heat conduction equation. For supersonic flow, the calculation starts from $X_{\max}$ and the viscosity becomes negative.

Instead of attenuation (at positive viscosity), we have amplification of small disturbances. Hence, the problem is unstable and it is equivalent to the inverse heat problem that is well known to be an ill-posed problem.

To show this in more detail, let us consider the evolution of harmonic disturbances of the following form:

$$\begin{pmatrix} \Delta\rho \\ \Delta U \\ \Delta V \\ \Delta e \end{pmatrix} \begin{pmatrix} \Delta\rho_0 \\ \Delta U_0 \\ \Delta V_0 \\ \Delta e_0 \end{pmatrix} e^{i(\omega x - ky)}.$$

Equations (1–5), assume the form

$$\mathbf{A}_{ij}\frac{\partial \Delta U_j}{\partial x} + \mathbf{B}_{ij}\frac{\partial \Delta U_j}{\partial y} + \mathbf{D}_{ij}\frac{\partial^2 \Delta U_j}{\partial y^2} + \mathbf{b}_i = 0, \quad i = 1, \ldots, 4,$$

where

$$\mathbf{A} = \begin{pmatrix} U & \rho & 0 & 0 \\ (\gamma-1)e/\rho & U & 0 & (\gamma-1) \\ 0 & 0 & U & 0 \\ 0 & (\gamma-1)e & 0 & U \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} V & 0 & \rho & 0 \\ 0 & V & 0 & 0 \\ (\gamma-1)e/\rho & 0 & V & (\gamma-1) \\ 0 & 0 & (\gamma-1)e & V \end{pmatrix}.$$

The resulting characteristic matrix is

$$\mathbf{C} = \begin{pmatrix} iU\omega - ikV & i\rho\omega & -ik\rho & 0 \\ i(\gamma-1)e/\rho\omega & iU\omega - ikV + k^2/(\rho\,\mathrm{Re}) & 0 & i\omega \\ -i(\gamma-1)ek/\rho & 0 & iU\omega - ikV + 4k^2/(3\rho\,\mathrm{Re}) & -ik \\ 0 & i(\gamma-1)e\omega & -ik(\gamma-1)e & iU\omega - iVk + \gamma k^2/(\rho\,\mathrm{Re}\,Pr) \end{pmatrix}.$$

From the condition $\det(\mathbf{C}) = 0$, one may find a relationship for the frequency $\omega = \omega(k)$. The determinant may be recast in the form

$$\begin{aligned} \det(\mathbf{C}) = {} & (U\omega - kV)\left(\frac{U\omega - kV - ik^2}{\rho\,\mathrm{Re}}\right)\left(\frac{U\omega - kV - i4k^2}{3\rho\,\mathrm{Re}}\right)\left(\frac{U\omega - Vk - i\gamma k^2}{\rho\,\mathrm{Re}\,\mathrm{Pr}}\right) \\ & - (U\omega - kV)\left(\frac{U\omega - kV - ik^2}{\rho\,\mathrm{Re}}\right)\left(\frac{U\omega - kV - i4k^2}{3\rho\,\mathrm{Re}}\right)(ik^2(\gamma-1)^2 e) \\ & + (U\omega - kV)(\gamma-1)^2 e\omega^2\left(\frac{U\omega - kV - 4ik^2}{3\rho\,\mathrm{Re}}\right) \\ & - \omega^2(\gamma-1)e\left(\frac{U\omega - Vk - 4ik^2}{3\rho\,\mathrm{Re}}\right)\left(\frac{U\omega - Vk - i\gamma k^2}{\rho\,\mathrm{Re}\,\mathrm{Pr}}\right) \\ & + k^2\left(\frac{U\omega - Vk - ik^2}{\rho\,\mathrm{Re}}\right)(\gamma-1)e\left(\frac{U\omega - Vk - i\gamma k^2}{\rho\,\mathrm{Re}\,\mathrm{Pr}}\right) \\ = {} & 0. \end{aligned}$$

With the tolerance of $O(1/\mathrm{Re})$, we may find one of the solutions as $iU\omega_1 - ikV + 4k^2/(3\rho\mathrm{Re}) = 0$ (this solution is exact for the case $\gamma/\mathrm{Pr} = 4/3$).

By substituting $\omega_1 = Vk/U + i4k^2/3\,\mathrm{Re}\,\rho U$ into $e^{i(\omega x - ky)}$, we obtain a multiplier $\exp(-(4k^2/3\,\mathrm{Re}\,\rho U)x)$, meaning that there is attenuation of small disturbances when x increases and the disturbances are amplified when x decreases. Therefore, the problem is ill-posed.

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 to calculate the discrepancy gradient when the number of control parameters is relatively large. The adjoint problem corresponding to Equations (1–6) follows [3]:

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

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

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

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

The boundary conditions on C ($X = X_{\max}$) are $\Psi_f|^{X=X_{\max}} = 0$.

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

$$\frac{\partial\Psi_f}{\partial Y} = 0. \tag{11}$$

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

$$\frac{\partial\varepsilon}{\partial e_\infty(Y)} = \Psi_e U + (\gamma-1)\Psi_U,$$

$$\frac{\partial\varepsilon}{\partial\rho_\infty(Y)} = \Psi_\rho U + \frac{(\lambda-1)\Psi_U e}{\rho},$$

$$\frac{\partial \varepsilon}{\partial U_\infty(Y)} = \Psi_U U + \rho \Psi_\rho + (\gamma - 1)\Psi_e e, \tag{12}$$

$$\frac{\partial \varepsilon}{\partial V_\infty(Y)} = \Psi_V U.$$

The flow-field (forward problem, (1–4)) is computed using a finite difference method [2,3] marching along 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; however, the integration is performed in the reverse direction (beginning at $X = X_{\max}$). The grid is rectangular and consists of 50–100 nodes along the Y direction and 50–200 nodes along the X direction (see [3] for more information regarding the discretization strategy). The flow parameters on the entrance boundary $f_\infty(Yi) = f_i (i = 1, \ldots, N)$ serve as the set of control variables. The input data $f_{\exp}(X_m, Y_i)(i = 1, \ldots, N)$ are obtained at the outflow section from a preliminary computation. The flow parameters are the external flow Mach number, $M = 5$ (the Mach number of the jet is about 3) and the Reynolds number Re in the range of $10^3 - 10^4$. Several tests were performed for an 'inviscid' flow ($\mathrm{Re} = 10^8$).

For a systematic analysis of the convergence rate for numerical solution techniques that require the gradient of discrete cost function, see [17].

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:

(1) conjugate gradients [18,30,31,33] (non-linear CG version);
(2) quasi-Newton (Broyden–Fletcher–Goldfarb–Shanno (BFGS)), [8–11,30];
(3) limited-memory quasi-Newton (L-BFGS) [12];
(4) T-N method [26,27];
(5) a new hybrid algorithm proposed by Morales and Nocedal [23] that consists of a class of optimization methods that interlace iterations of the L-BFGS method and a T-N method 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, the efficient preconditioning of the GC method, see also [24,25]. This new algorithm was studied and tested in [6,7] and was demonstrated to be the best performing algorithm.

In this work, we test implementations of the L-BFGS version VA15 of [22] in the Harwell library, the T-N method described by Nash [26,27] and the hybrid method of Morales and Nocedal [23]. A brief description of the major components of each algorithm is given below. The nonlinear CG algorithm CONMIN used in this study is described as well [26]. The code of Shanno and Phua [33] allows also for the implementation of the quasi-Newton BFGS method.

The subroutine CONMIN incorporates two nonlinear optimization methods, a nonlinear CG algorithm and a variable metric (Newton method) algorithm, with the choice of method left to the user. The nonlinear GC algorithm is the Beale restarted GC strategy [1,5]. 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 [33], and requires approximately $n^2/2 + 11n/2$ 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 ith iteration, ∇f_k is the gradient vector

of size n evaluated at $\mathbf{x}_k$ and $\mathbf{H}_k = \nabla^2 f_k$ is the $n \times n$ symmetric Hessian matrix of the second partial derivatives of f with respect to the coordinates evaluated at $\mathbf{x}_k$. In all the algorithms, the new iterate is calculated from

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k^{\mathrm{T}} \mathbf{x}_k, \tag{13}$$

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

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

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* [12,14].

$$f(\mathbf{x}_k) - f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \geq -\mu \alpha_k \mathbf{p}_k^{\mathrm{T}} \nabla f_k, |\nabla f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)^{\mathrm{T}} \mathbf{p}_k| \leq \eta |\nabla f_k^{\mathrm{T}} \mathbf{p}_k| \tag{15}$$

where $0 < \mu < \eta < 1$.

The values of the parameters μ and η used were 10^{-4} and 0.1, respectively.

3.1 *The nonlinear CG algorithm*

CG uses derivatives of f, defined by ∇f_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 Equation (13) and the search directions are expressed recursively as

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

Calculation of β_k with the algorithm incorporated in CONMIN used for nonlinear CG is described in [32].

CONMIN has important advantages such as automatic restart along a carefully chosen direction [31] and global convergence properties [13]. The Hessian vector products in the nonlinear CG code were done via finite differencing of gradients. As will be shown in Section 3.5, the Hessian vector product is accurate to the order of $\sqrt{\varepsilon_m}$, where ε_m is the machine accuracy (2^{-53} for this double precision application).

If one considers the memoryless BFGS formula,

$$\mathbf{H}_{k+1} = \left(\mathbf{I} - \frac{\mathbf{s}_k \mathbf{y}_k^{\mathrm{T}}}{\mathbf{y}_k^{\mathrm{T}} \mathbf{s}_k}\right) \left(\mathbf{I} - \frac{\mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}}{\mathbf{y}_k^{\mathrm{T}} \mathbf{s}_k}\right) + \frac{\mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}}{\mathbf{y}_k^{\mathrm{T}} \mathbf{s}_k}, \tag{17}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k = \alpha_k \mathbf{p}_k$ and $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$. In conjunction with an exact line search for which $\nabla f_k^{\mathrm{T}} \mathbf{p}_k = 0$ for all k, then we obtain $\mathbf{p}_{k+1} = -\mathbf{H}_{k+1} \nabla f_{k+1} = -\nabla f_{k+1} + (\nabla f_k^{\mathrm{T}} \mathbf{y}_k / \mathbf{y}_k^{\mathrm{T}} \mathbf{p}_k) \mathbf{p}_k$, which is the Hestenes–Stiefel CG Method, and when $\nabla f_{k+1}^{\mathrm{T}} \mathbf{p}_k = 0$, the Hestenes–Stiefel formula reduces to the Polak–Ribiere formula:

$$\beta_{k+1}^{\mathrm{PR}} = \frac{\nabla f_k^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2} \mathbf{p}_k. \tag{18}$$

As shown in [30], CONMIN is related to the BFGS variable metric method and increased storage requirements for CONMIN results in fewer function evaluations. Indeed in terms of requiring the fewest number of function evaluations, CONMIN is on top for the examples tested in [30]. Automatic restarting is used to preserve a linear convergence rate. For restart iterations, the step length $\alpha_k = 1$ is used. On the other hand, for no restart iterations,

$$\alpha_{k+1} = \frac{\alpha_k \nabla f_k^{\mathrm{T}} \mathbf{p}_k}{\nabla f_{k-1}^{\mathrm{T}} \mathbf{p}_{k-1}}. \tag{19}$$

3.2 *The CG-descent method*

Hager and Zhang [18] developed a new nonlinear CG algorithm for unconstrained optimization problems.

The CG iterates assume the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{20}$$

where the stepsize α_k is positive and where the directions $\mathbf{d}_k$ are generated by the rule $\mathbf{d}_{k+1} = -\nabla f_{k+1} + \beta_k^N \mathbf{d}_k$, $\mathbf{d}_0 = -\nabla f_0$, while $\beta_k^N = 1/\mathbf{d}_k^{\mathrm{T}}\mathbf{y}_k(\mathbf{y}_k - 2\mathbf{d}_k\|\mathbf{y}_k^2\|/\mathbf{d}_k^{\mathrm{T}}\mathbf{y}_k)^{\mathrm{T}}\nabla f_{k+1}$.

Here $\|\cdot\|$ is the Euclidean norm, and $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$. If f is a quadratic and α_k is chosen to achieve the exact minimum of f in the direction $\mathbf{d}_k$, then $\mathbf{d}_k^{\mathrm{T}}\nabla f_{k+1} = 0$, and the formula for β_k^N reduces to the familiar Hestenes–Stiefel scheme.

The advantages of the new CG scheme are described in [18].

A judicious choice of parameters is required to obtain optimal results, in particular, for problems that are associated with PDE-constrained optimization, see the user manual that comes with the free code distribution. A program searching the parameter space for the CG-descent method for a given optimization problem was developed by one of the authors.

3.3 *BFGS quasi-newton method*

The evaluation of the Hessian matrix is impractical or costly for large-scale minimization. A central idea underlying quasi-Newton methods is to use an approximation of the inverse Hessian. The form of the approximation differs among methods. In quasi-Newton methods, instead of the true Hessian $\mathbf{H}$, an initial matrix $\tilde{\mathbf{H}}_0$ is chosen (usually $\tilde{\mathbf{H}}_0 = \mathbf{I}$), which is subsequently updated by an update formula. The approximate Hessian $\tilde{\mathbf{H}}_k$ is then used in place of the true Hessian.

Given displacement $\mathbf{s}_k$ and change of gradients $\mathbf{y}_k$, the secant equation requires that the symmetric and positive-definite matrix $\tilde{\mathbf{H}}_{k+1}$ maps $\mathbf{s}_k$ into $\mathbf{y}_k$. This is possible only if $\mathbf{s}_k$ and $\mathbf{y}_k$ satisfy the curvature condition

$$\mathbf{s}_k^{\mathrm{T}}\mathbf{y}_k > 0. \tag{21}$$

To determine $\tilde{\mathbf{H}}_{k+1}$ uniquely, the additional condition is imposed that among all symmetric matrices satisfying the secant equation, $\tilde{\mathbf{H}}_{k+1}$ is in a sense closest to the current matrix $\tilde{\mathbf{H}}_k$, i.e. we solve the problem

$$\min_{\tilde{\mathbf{H}}} \left\| \tilde{\mathbf{H}} - \tilde{\mathbf{H}}_k \right\| \tag{22}$$

subject to $\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^{\mathrm{T}}$ and $\tilde{\mathbf{H}}\mathbf{s}_k = \mathbf{y}_k$ and $\tilde{\mathbf{H}}_k$ is positive-definite.

Using a weighted Frobenius norm, the unique solution of Equation (22), as shown in [31], is the Davidon–Fletcher–Powell (DFP) updating formula originally proposed by Davidon [8] and popularized by Fletcher and Powell [11].

$$\tilde{\mathbf{H}}_{k+1} = (\mathbf{I} - \gamma_k \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}})\tilde{\mathbf{H}}_k(\mathbf{I} - \gamma_k \mathbf{s}_k \mathbf{y}_k^{\mathrm{T}}) + \gamma_k \mathbf{y}_k \mathbf{y}_k^{\mathrm{T}} \tag{23}$$

with $\gamma_k = 1/\mathbf{y}_k^{\mathrm{T}}\mathbf{s}_k$.

Instead of imposing conditions on the Hessian approximations $\tilde{\mathbf{H}}_k$, we impose similar conditions on their inverses $\tilde{\mathbf{B}}_k$. The updated approximation $\tilde{\mathbf{B}}_{k+1}$ must be symmetric and positive definite,

and must satisfy the secant equation now written as

$$\tilde{\mathbf{B}}_{k+1}\mathbf{y}_k = \mathbf{s}_k. \tag{24}$$

The condition of closeness to $\tilde{\mathbf{B}}_k$ is now specified by

$$\min_{\tilde{\mathbf{B}}} \left\| \tilde{\mathbf{B}} - \tilde{\mathbf{B}}_k \right\| \tag{25}$$

using a weighted Frobenius norm, subject to $\tilde{\mathbf{B}} = \tilde{\mathbf{B}}^{\mathrm{T}}$, Equation (24), and $\tilde{\mathbf{B}}$ being positive definite.

The unique solution $\tilde{\mathbf{B}}_{k+1}$ to Equation (25) is given by

$$\tilde{\mathbf{B}}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}_k^{\mathrm{T}})\tilde{\mathbf{B}}_k(\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}) + \rho_k \mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}, \tag{26}$$

where $\rho_k = 1/y_k^{\mathrm{T}} s_k$.

The quasi-Newton methods that build up an approximation of the inverse Hessian are often regarded as the most sophisticated optimization methods for solving unconstrained problems. It can be shown [see [31] for motivation] that as long as $\tilde{\mathbf{B}}_k$ exists at the true minimum $\mathbf{x}^*$, the initial guess $\mathbf{x}_0$ is 'sufficiently' near x^*, and the curvature condition holds, the BFGS methods will converge. Indeed, if the remainder $\mathbf{r}_k = \tilde{\mathbf{B}}_k \mathbf{p}_k + \nabla f_k$ can be bounded in relation to ∇f_k between 0 and 1, that is, if $\|\mathbf{r}_k\| \leq \eta_k \|\nabla f_k\|$ for some $\eta_k \leq \eta \in [0, 1)$ where η is a constant, any quasi-Newton method is guaranteed to converge. If $\lim_{k\to\infty} \eta_k = 0$, the rate of convergence will be superlinear, and if $\tilde{\mathbf{B}}_k$ is Lipschitz continuous for $\mathbf{x}_k$ near $\mathbf{x}^*$and $\eta_k = O(\|\nabla f_k\|)$, the rate of convergence will be quadratic [31].

The BFGS formula (26) is straightforward to apply as the BFGS update formula can be used exactly like the DFP formula. Numerical experiments have shown that the performance of the BFGS formula is superior to the DFP formula. Hence, BFGS is often preferred over DFP. As Nocedal and Wright [31] note, the DFP and BFGS updating formulae are dual of each other, one being obtained from the other via interchanges $\mathbf{s} \leftrightarrow \mathbf{y}$ and $\tilde{\mathbf{B}} \leftrightarrow \tilde{\mathbf{H}}$.

Both the DFP and BFGS updates are symmetric rank 2 corrections that are constructed from the vectors $\mathbf{s}_k$ and $\mathbf{y}_k$. Weighted combinations of these formulae will therefore also have the same properties. This observation leads to a whole collection of updates known as the Broyden family.

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. Thus, in the BFGS [9,10], Equation (24) is used with an approximation $\tilde{\mathbf{B}}_k$ to the inverse Hessian, which is updated by

$$\tilde{\mathbf{B}}_{k+1} = \mathbf{V}_k^{\mathrm{T}} \tilde{\mathbf{B}}_k \mathbf{V}_k + \rho_k \mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}, \tag{27}$$

where $\mathbf{V}_k = \mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}}$, $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$ and $\rho_k = 1/(\mathbf{y}_k^{\mathrm{T}} \mathbf{s}_k)$. The search direction is given by

$$\mathbf{p}_{k+1} = -\tilde{\mathbf{B}}_{k+1} \mathbf{g}_{k+1}. \tag{28}$$

In the L-BFGS method, instead of forming the matrices $\tilde{\mathbf{B}}_k$ 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 $\tilde{\mathbf{B}}_k$ implicitly; a cyclical procedure is used to retain the latest vectors and discard the oldest ones. Thus, after the first m iterations, Equation (18) becomes

$$\begin{aligned}\tilde{\mathbf{B}}_{k+1} = (\mathbf{V}_k^{\mathrm{T}} \cdots \mathbf{V}_{k-m}^{\mathrm{T}})\tilde{\mathbf{B}}_{k-1}^{0}(\mathbf{V}_{k-m} \cdots \mathbf{V}_k) + \rho_{k-m}(\mathbf{V}_k^{\mathrm{T}} \cdots \mathbf{V}_{k-m+1}^{\mathrm{T}})\mathbf{s}_{k-m}\mathbf{s}_{k-m}^{\mathrm{T}}(\mathbf{V}_{k-m-1}^{\mathrm{T}} \cdots \mathbf{V}_k) \\ + \rho_{k-m-1}(\mathbf{V}_k^{\mathrm{T}} \cdots \mathbf{V}_{k-m+2}^{\mathrm{T}})\mathbf{s}_{k-m+1}\mathbf{s}_{k-m+1}^{\mathrm{T}}(\mathbf{V}_{k-m+2}^{\mathrm{T}} \cdots \mathbf{V}_k) \cdots + \rho_k \mathbf{s}_k \mathbf{s}_k^{\mathrm{T}} \end{aligned} \tag{29}$$

with the initial approximation $\tilde{\mathbf{B}}_{k+1}^{0}$ the diagonal matrix

$$\tilde{\mathbf{B}}_{k+1}^{0} = \frac{\mathbf{y}_{k+1}^{\mathrm{T}}\mathbf{s}_{k+1}}{\mathbf{y}_{k+1}^{\mathrm{T}}\mathbf{y}_{k+1}}\mathbf{I}. \tag{30}$$

It should be noted that this is only one of the possible ways to choose the initial approximation; other choices are possible as well to try to improve the L-BFGS approximation (in fact, this is exactly what is done in the implementation of the hybrid algorithm below). Many previous studies have shown that $3 \leq m \leq 7$ is sufficient and $m > 7$ usually does not improve the performance of the L-BFGS algorithm. Here we used a value of $m = 5$.

3.5 *The T–N algorithm*

In the T-N method, also known as the Hessian-free Newton (HFN) method, a search direction is computed by finding an *approximate* solution to the Newton equations,

$$\mathbf{H}_k \mathbf{p}_k = -\nabla f_k. \tag{31}$$

The use of an approximate search direction $\mathbf{p}_k$ 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 (13), there is an *inner* iteration making use of the CG method that computes this approximate direction, $\mathbf{p}_k$. The CG inner algorithm is preconditioned by a scaled two-step L-BFGS method, with Powell's restarting strategy used to reset the preconditioner periodically. A detailed description of the preconditioner may be found in [26]. The Hessian vector product $\mathbf{H}_k\mathbf{v}$ for a given $\mathbf{v}$ required by the inner CG algorithm is obtained by a finite difference approximation,

$$\mathbf{H}_k\mathbf{v} \approx \frac{\nabla f(\mathbf{x}_k + h\mathbf{v}) - \nabla f(\mathbf{x}_k)}{h}. \tag{32}$$

A major issue is how to adequately choose h [34]; in this work, we use $h = \varepsilon^{1/2}(1 + \|\mathbf{x}_k\|)$, where ε is the machine precision and $\|\cdot\|$ denotes the Euclidean norm. Using this approximation, the Hessian will be accurate up to $O(h)$ [34]. The inner algorithm is terminated using the quadratic truncation test, which monitors a sufficient decrease of the quadratic model $q_k = \mathbf{p}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{p}_k/2 + \mathbf{p}_k^{\mathrm{T}}\nabla f_k$:

$$\frac{1 - q_k^{i-1}}{q_k^i} \leq \frac{c_q}{i}, \tag{33}$$

where i is the counter for the inner iteration and c_q is a constant, $0 < c_q < 1$. The inner algorithm is also terminated if an imposed upper limit on the number of inner iterations, M, is reached, or when a loss of positive-definiteness is detected in the Hessian (i.e. when $\mathbf{v}^{\mathrm{T}}\mathbf{H}_k\mathbf{v} < 10^{-12}$).

T-N methods can be extended to more general non-convex problems in much the same way as Newton's method [27].

3.6 *The hybrid method*

The hybrid method consists of interlacing in a dynamical way the L-BFGS method with the T-N method discussed above. The limited-memory matrix $\mathbf{H}_m$ plays a dual role of preconditioning the inner CG 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 HFN/T-N. One notes that the strengths and weaknesses of the HFN and L-BFGS methods are complementary. The HFN method requires much fewer iterations to approach the solution, but the computational effort invested in one iteration can be very high while curvature information gathered in the process is lost once the iteration is completed. The L-BFGS method, on the other hand, performs inexpensive iterations, but the quality of the curvature information it collects may be poor, and as a consequence it can be slow on ill-conditioned problems. The enriched algorithm aims to combine the best features of both methods in a dynamic manner [25].

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

$$[k1 * (L - \text{BFGS}) \rightarrow k2 * (T - N(\text{PCG})) \rightarrow \tilde{\mathbf{B}}(m), \text{repeat}], \tag{34}$$

where $\tilde{\mathbf{B}}(m)$ is again a limited-memory matrix that approximates the inverse of the Hessian matrix (20), and m denotes the number of correction pairs stored. The L-BFGS cycle starts from the initial unit or a weighted unit matrix, $\tilde{\mathbf{B}}(m)$ 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 $\tilde{\mathbf{B}}(m)$ is updated using information generated by the inner preconditioned CG (PCG) iteration to precondition the next T-N iteration. At the end of the T-N steps, the most current $\tilde{\mathbf{B}}(m)$ matrix is used as the initial matrix in a new cycle of L-BFGS steps.

A more detailed description of this algorithm is provided by Morales and Nocedal [25] and in [6,7].

4. Numerical tests

The computations have the following algorithmic structure: the forward problem (1–5) is solved for parameters $f(Y_\infty)$ 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 Equation (12). Then, the new control parameters are calculated using the chosen optimizer. The optimization algorithm uses the following prescribed termination criterion: $\|\nabla\varepsilon\| < 10^{-6} \max(1, \|f_\infty\|)$.

Figures 2–5 represent the solution of this problem by different minimization methods compared with the exact data.

Figure 2 presents the result of inflow temperature estimation from the outflow data. Figure 3 presents the inflow density illustrating the development of the instability (the constant density

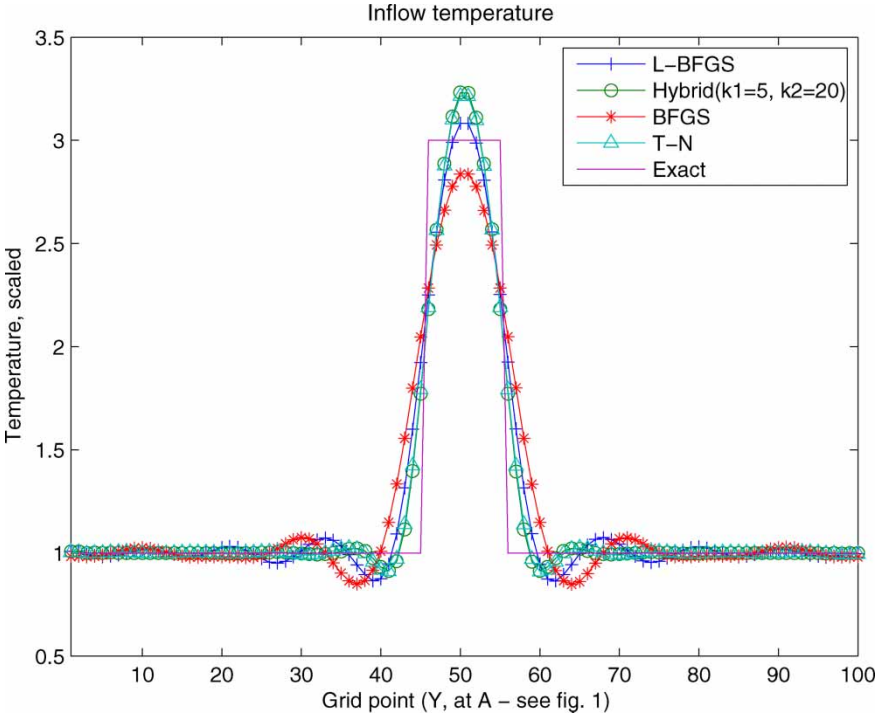

Figure 2. Inflow temperature calculation.

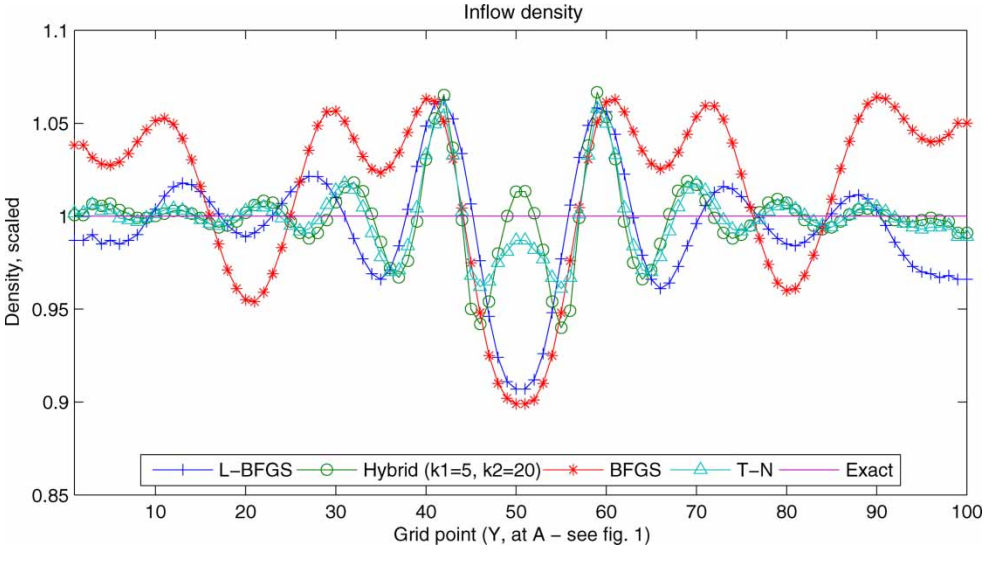

Figure 3. Inflow density calculation.

being equal to unity at the exact solution). Figure 4 provides the total density distribution in the flow-field for the exact solution and the result of the calculation. Figure 5 illustrates the adjoint density field.

Figure 6 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

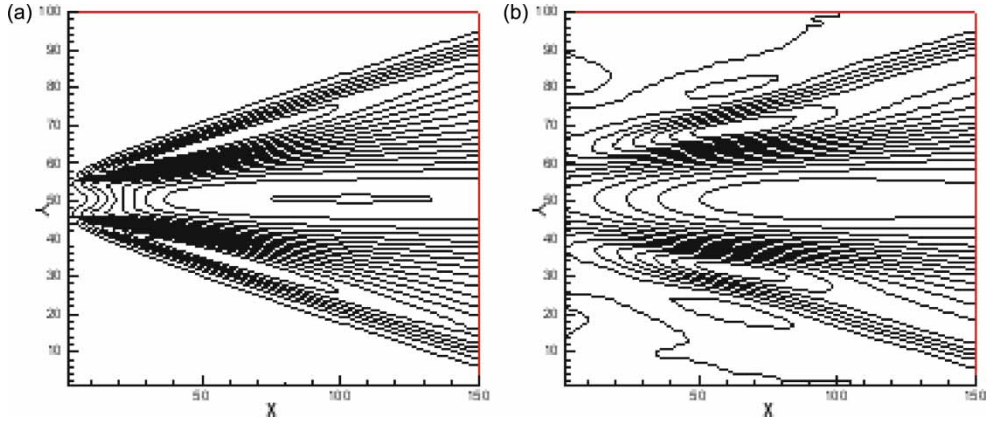

Figure 4. Target versus computed density field.

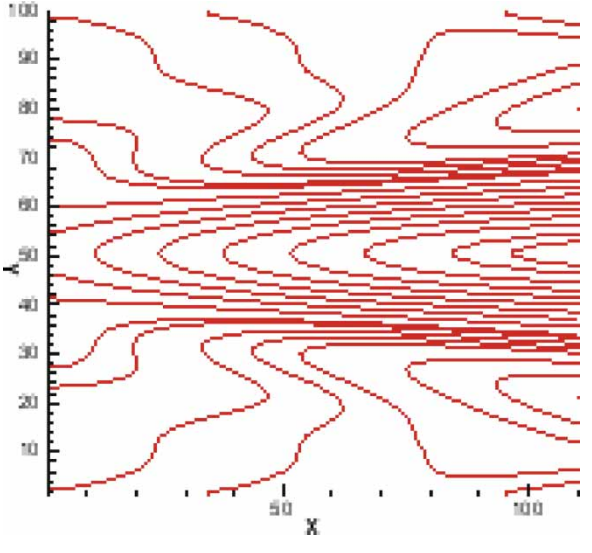

Figure 5. Adjoint density field.

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.

Figures 7 and 8 present a comparison of different minimization methods applied to a viscous flow ($\mathrm{Re} = 10^3$). 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 7 presents the results demonstrated by CG ([33] and [18] 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

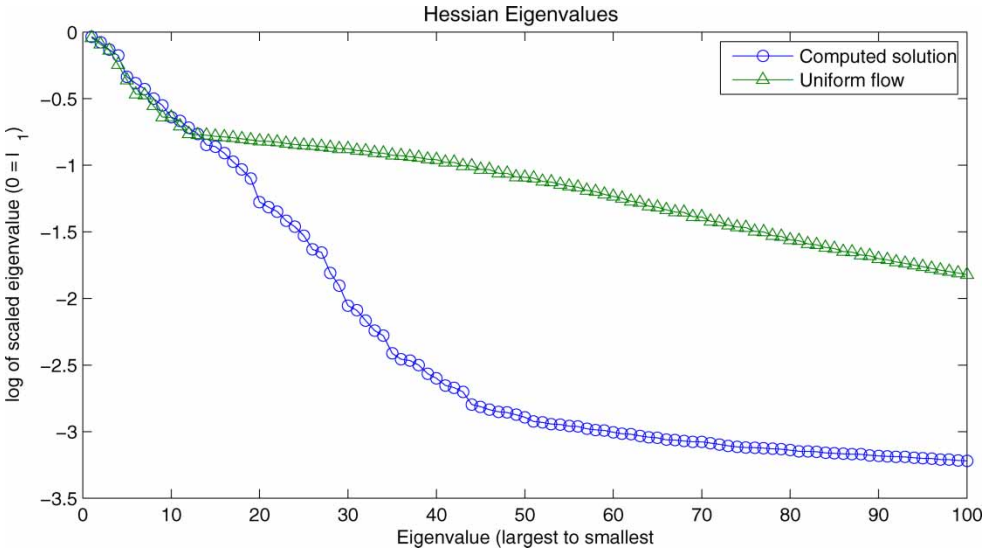

Figure 6. Hessian eigenvalues ordered by magnitude.

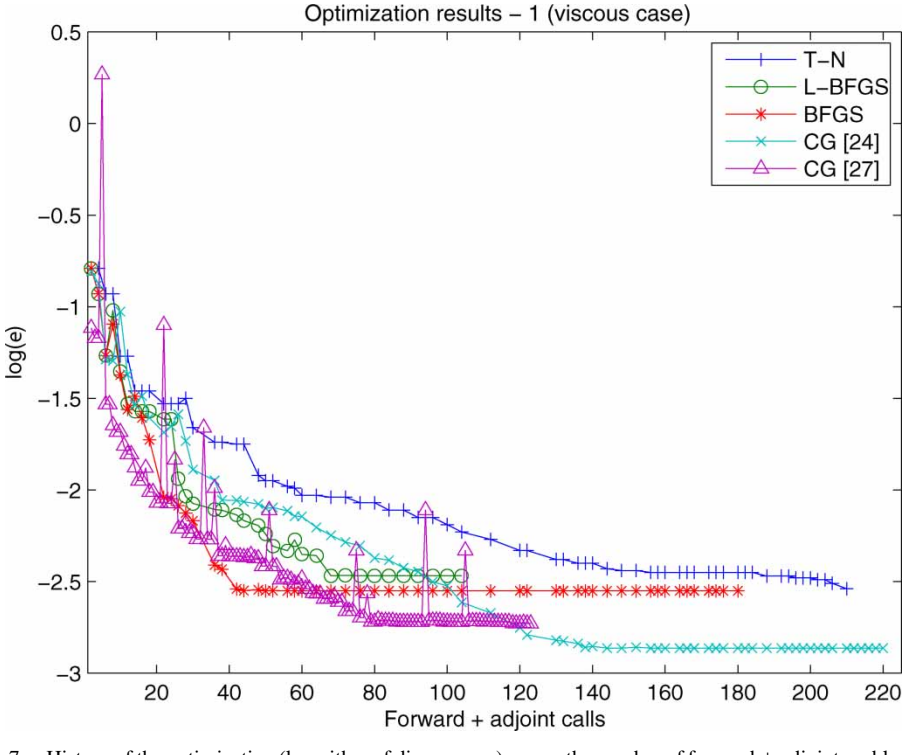

Figure 7. History of the optimization (logarithm of discrepancy) versus the number of forward + adjoint problem calls, CG [33], CG [18], BFGS, L-BFGS and T-N for viscous case.

tested also on this problem by selecting a combination of L-BFGS calls ($k1$) and T-N calls ($k2$). A simplistic trial-and-error search of the parameter space showed that the optional combination was $k1 = 5$ and $k2 = 20$ for this problem. The hybrid method performances (for different $k1$ and $k2$) compared with those of T-N and L-BFGS are presented in Figure 8.

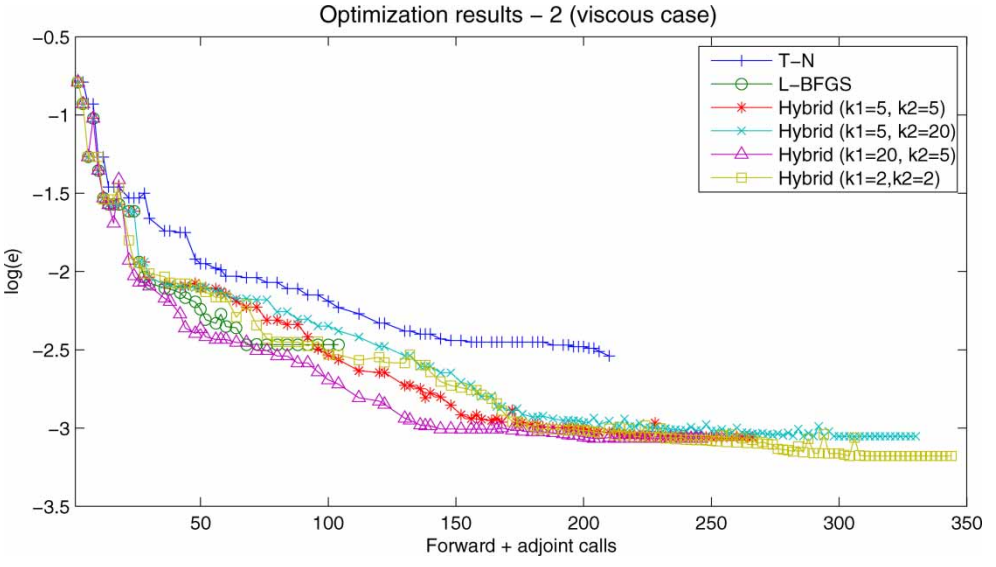

Figure 8. History the optimization (logarithm of discrepancy) versus the number of forward + adjoint problem calls, T-N, L-BFGS and hybrid ($k1 = 5$, $k2 = 20$) for the viscous case.

Figures 9–12 present results of another test (for inviscid flow, Reynolds number of 10^8). Figure 9 represents the history of minimization iterations for the CG, BFGS, L-BFGS and T-N unconstrained minimization methods.

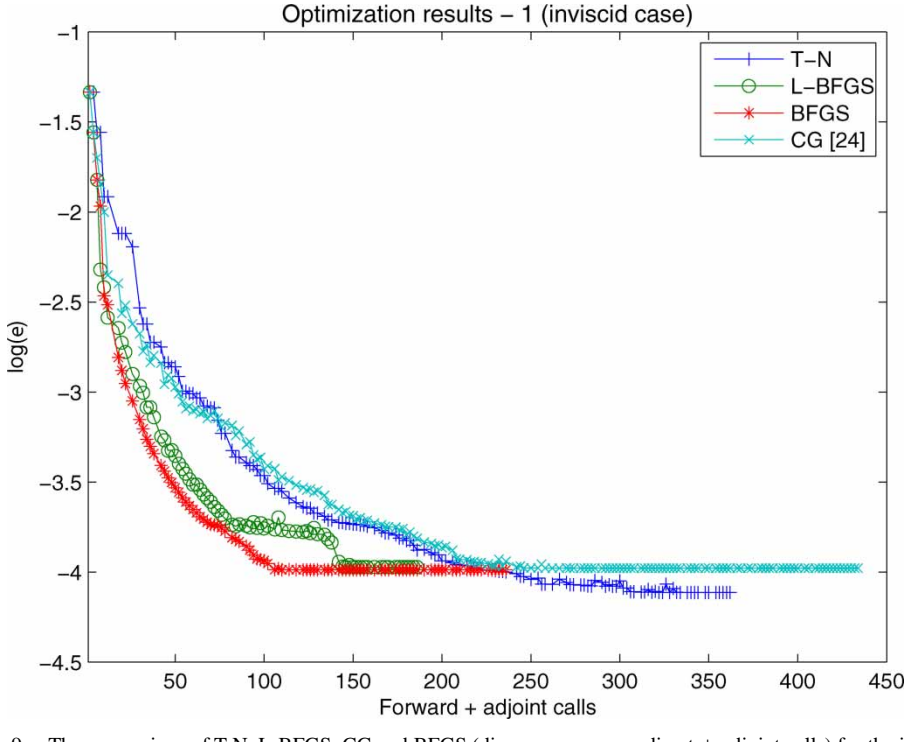

Figure 9. The comparison of T-N, L-BFGS, CG and BFGS (discrepancy versus direct + adjoint calls) for the inviscid case.

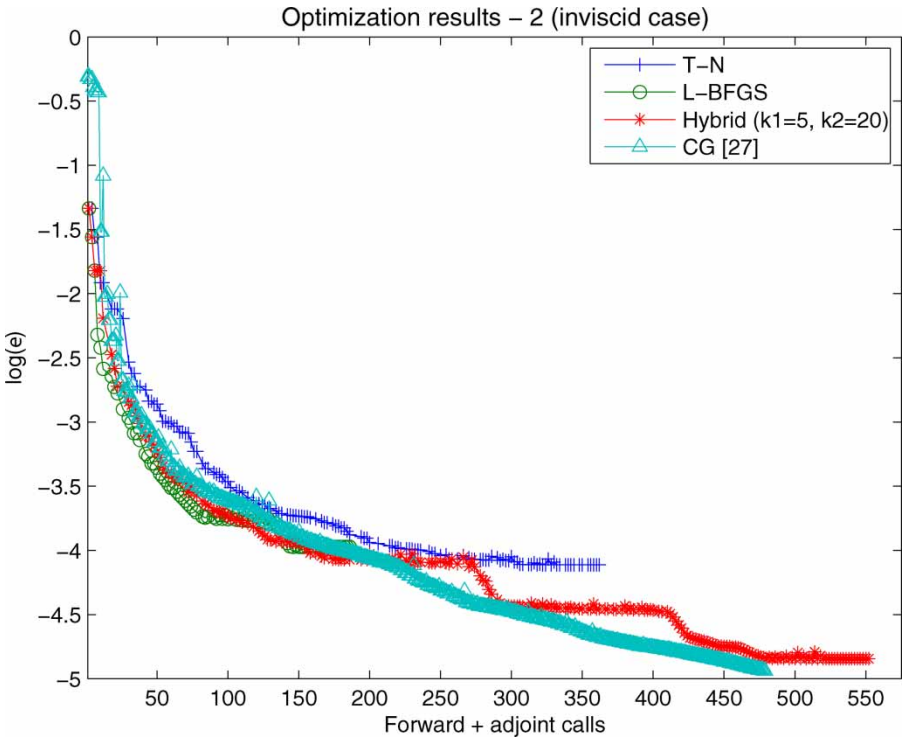

Figure 10. The comparison of T-N, L-BFGS, hybrid ($k1 = 5$, $k2 = 20$) and CG [18] (discrepancy versus direct + adjoint calls) for the inviscid case.

Figure 10 shows a comparison between T-N, L-BFGS and the hybrid algorithms for the inviscid case where we also have plotted the cost functional versus the number of direct + adjoint calls. The comparison of Figures 9 and 10 shows that the CG-descent method achieves the best results from the viewpoint of both quality and speed followed immediately by the hybrid method.

Figures 11 and 12 present a comparison of results obtained for the considered minimization methods versus the exact result.

The calculation time in terms of the number of direct and adjoint problem calls and the consumed CPU time is presented in Tables 1 and 2, respectively. The CPU time corresponds to the Celeron (800 MHz) processor and the Windows-98 operational system. The specifics of the 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 only about 2% of total computational time. This is connected with the relatively low dimensionality of control parameters (400) and high expense of solving the direct and adjoint problems. Solving the adjoint problem call is less time consuming than solving the direct one due to the linearization of the forward problem during the adjoint process.

Table 3 displays the norm of solution error as the sum of square discrepancies of optimal and exact solutions for the quasi-Newton methods.

4.1 *Quality of the adjoint model*

The adjoint of the parabolized Navier–Stokes equations was derived from a differentiate-then-discretize (continuous adjoint) approach.

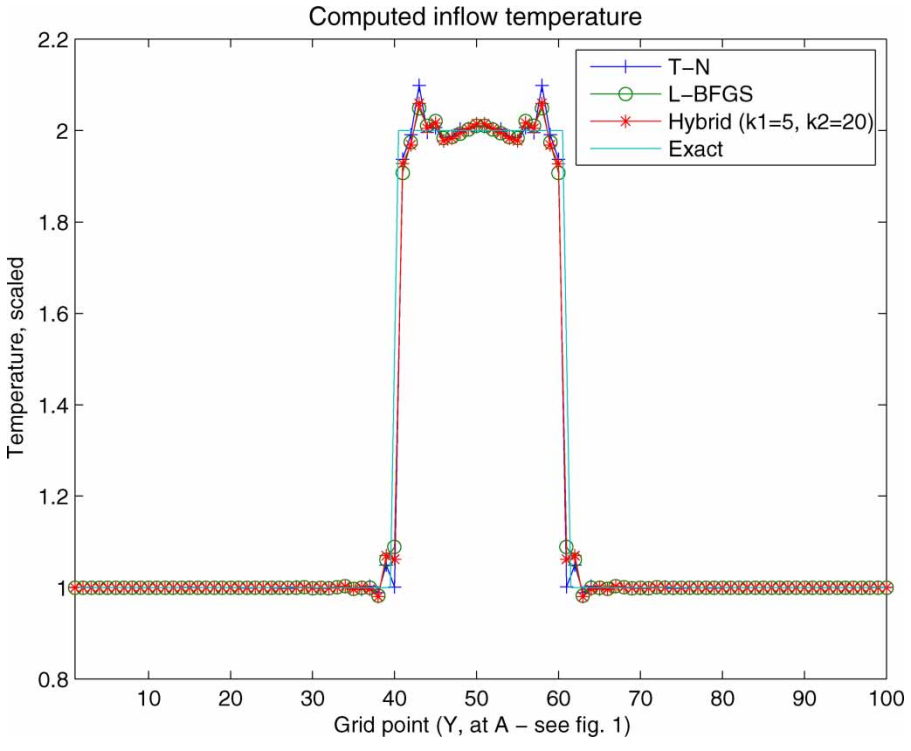

Figure 11. Inflow density.

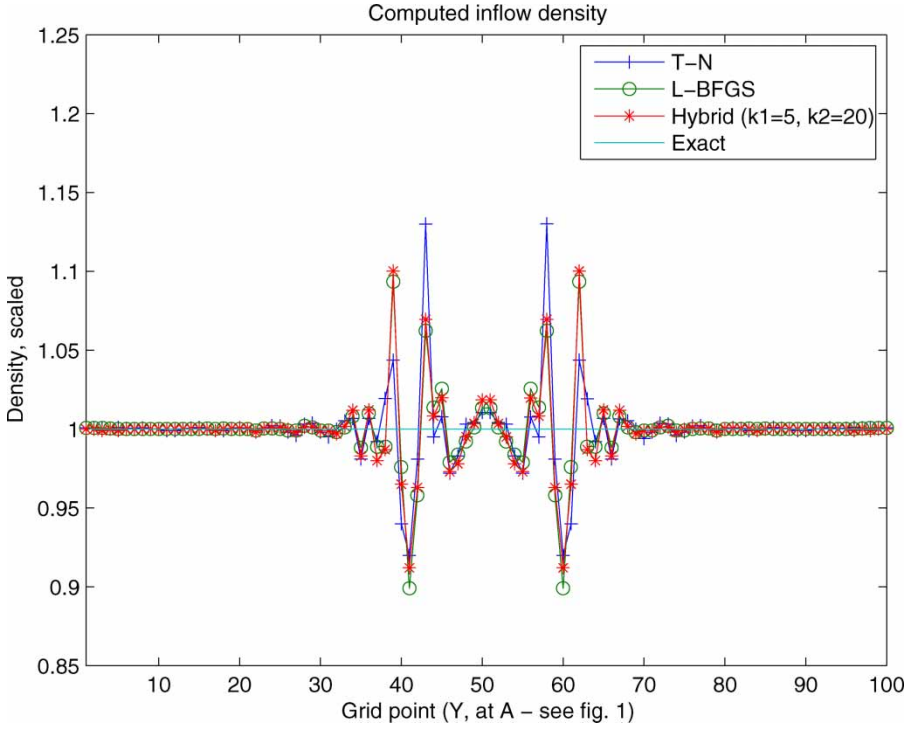

Figure 12. Inflow temperature.

Table 1. Direct solver performance.

Method	Direct calls	Adjoint calls	% Direct CPU time	% Adjoint CPU time	% Other ops
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
CG [33]	176	175	59.5	38.1	2.4
CG [18]	161	318	33.1	66.9	0.4
BFGS	120	116	59.5	38.5	2.0

Table 2. Adjoint solver performance.

Method	Direct calls	Adjoint calls	Number of inner CG iterations	Direct CPU time (s)	Adjoint CPU time (s)	Total time (s)
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
CG [33]	176	175	–	158.7	101.6	266.7
CG [18]	161	318	–	87.2	272.9	360.1
BFGS	120	116	–	105.9	68.6	177.9

Table 3. Norm of final solution error.

Method	LBFGS	Hybrid	BFGS	T-N	CG [33]	CG [18]
Norm of solution error	2.5186	2.5237	2.4618	2.5154	2.5106	2.5129

A verification of the quality of the gradient of the cost functional with respect to the control variables yields around two digits of accuracy.

A more significant test is the alpha test [29]. The alpha test verification of the correctness of the gradient is described below.

Let $J(\mathbf{x}+\alpha\mathbf{h}) = J(\mathbf{x}) + \alpha\mathbf{h}^{\mathrm{T}}\nabla J(\mathbf{x}) + O(\alpha^2)$ be a Taylor expansion of the cost function $J = \varepsilon$ around $\mathbf{X}$. The term α is a small scalar, and $\mathbf{h}$ is a vector of unit length (such as $\mathbf{h} = \nabla J / \|\nabla J\|$). Rewriting the above formulas, a function of α can be defined as

$$\varphi(\alpha) = \frac{J(\mathbf{x}+\alpha\mathbf{h}) - J(X)}{\alpha\mathbf{h}^{\mathrm{T}}\nabla J(X)} = 1 + O(\alpha).$$

For values of α that are small but not too close to the machine zero, one should expect to obtain a value for $\varphi(\alpha)$ that is close to 1.

The values of $\varphi(\alpha)$ are shown in Figures 13 and 14 as a function of α . It is clear that, for a value of α between 10^{-2} and 10^{-8}, a near unit value of $\varphi(\alpha)$ is obtained for both inviscid and viscous cases.

This validates the quality of the adjoint model for use in obtaining the gradient of the cost function with respect to the control variables for both the inviscid and viscid cases, respectively. It is anticipated that these conclusions will hold with a higher accuracy for a discrete gradient as well. An upcoming paper with the same authors describes and compares the use of the differentiate-then-discretize (used in this study) versus a discretize-then-differentiate gradient obtained from an automatic differentiator [15,16].

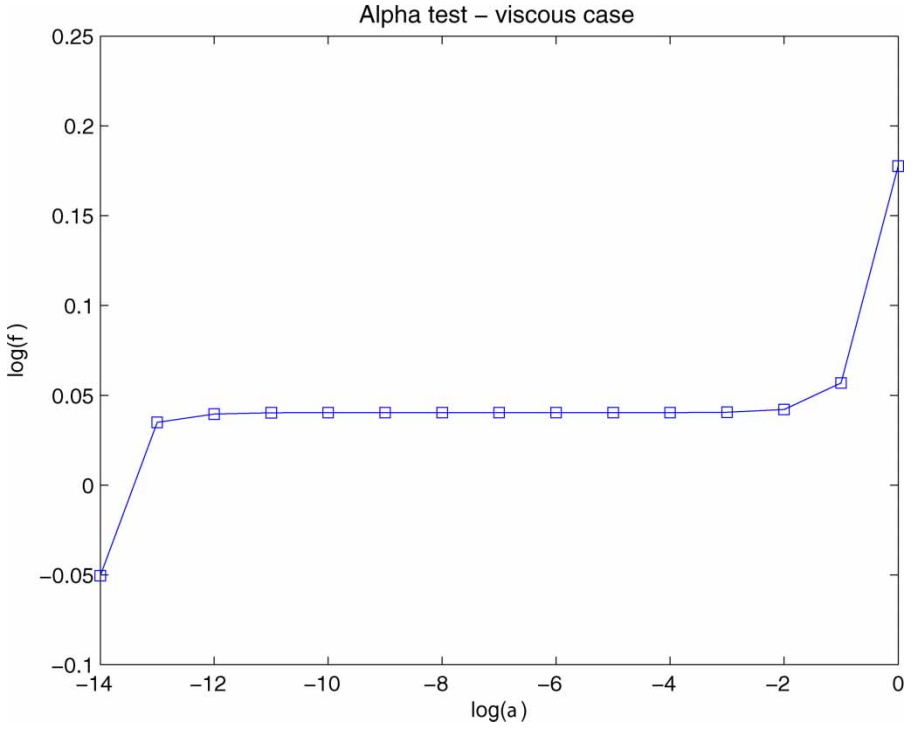

Figure 13. Verification of gradient calculation: variation of ϕ with respect to α (viscous case).

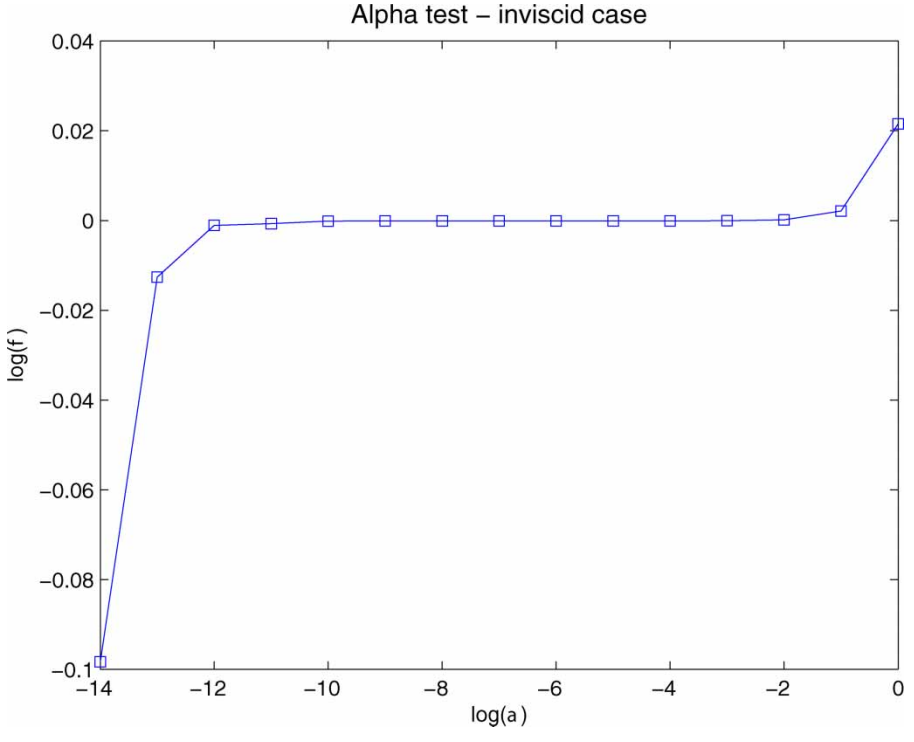

Figure 14. Verification of gradient calculation: variation of ϕ with respect to α (inviscid case).

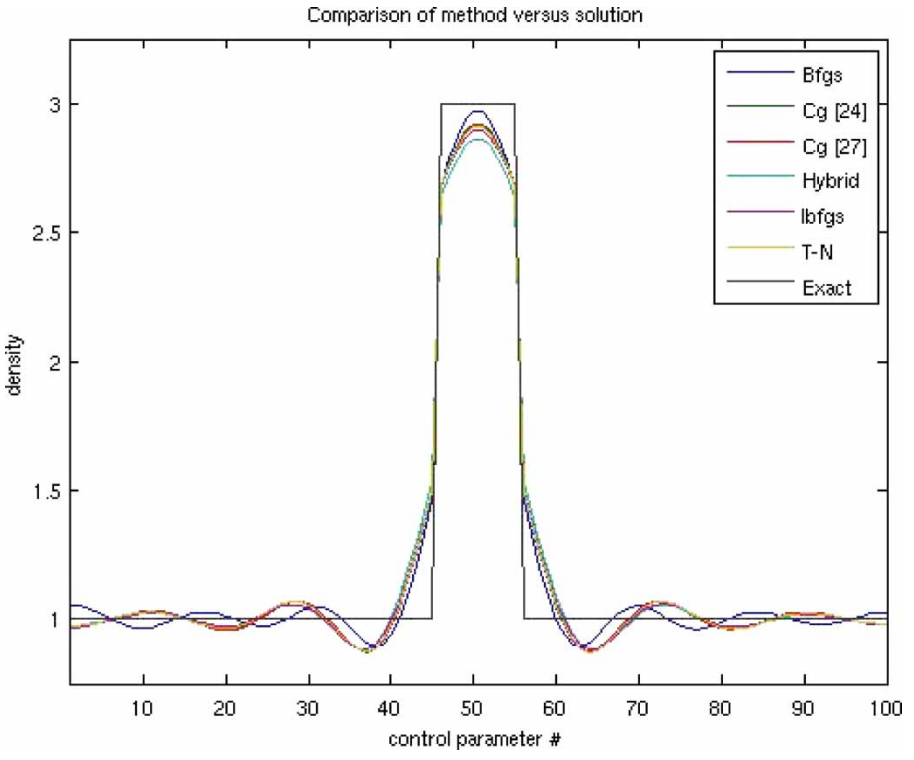

Figure 15. Comparison of methods (viscous case).

4.2 *Issues of ill-posedness and multiple minima*

As the problem we are addressing is an ill-posed inverse parameter estimation, the issue of uniqueness of the local minima attained has to be placed in the context of the accuracy the different minimizers attain. Moreover the CG methods used in the comparison have a self-regularization property [20].

Additional tests conducted show that while the minimum of the cost function attained by the various methods is not identical, the solutions they attain are equal within a range of 5×10^{-3}.

Figure 15 shows all of the viscous case solutions of the various minimization methods employed in this research. As is evident from the figure, the solutions obtained are within the aforementioned range.

As seen in Figure 15, it becomes evident that further research along these lines would benefit from the use of non-smooth optimization techniques. Breaking the interval into sub-domains may improve the final error norm of these solutions.

4.3 *Sensitivity to initial guess*

Several tests (not shown) were run to determine the sensitivity of various methods to the choice of the initial guess. Perturbations of order ranging from 10^{-5} to 10^{0} were added to the initial guess (which was determined from engineering experience and intuition).

The results show that perturbations up to the order 10^{-2} converge to minima of the same order as the unperturbed problem. Perturbations of the order of 10^{-1} still converge to minima of lower

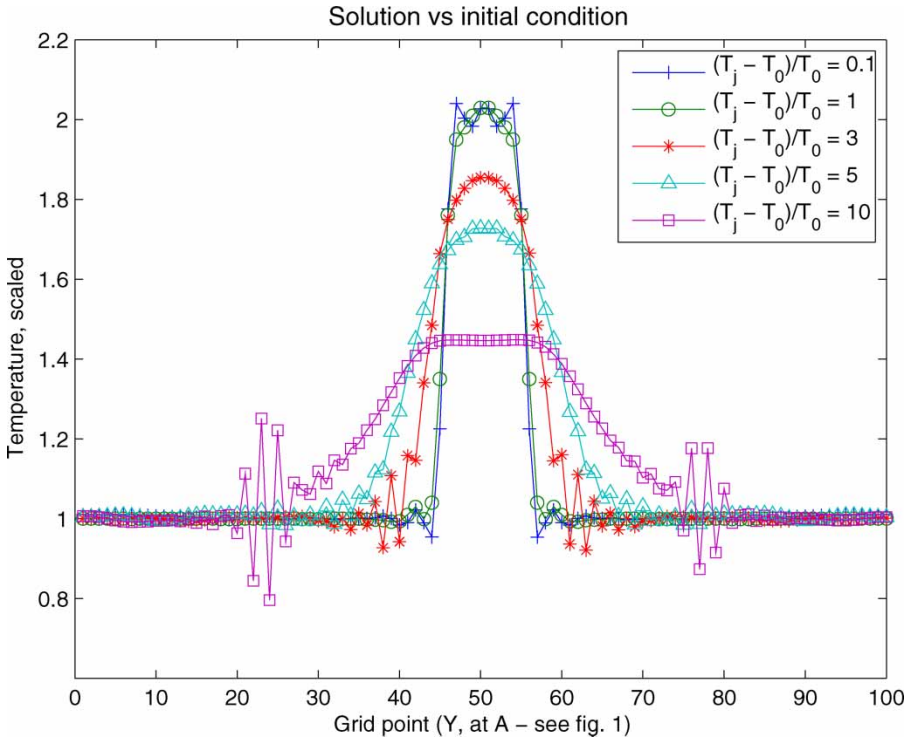

Figure 16. The quality of solution as a function of the disturbance magnitude (inviscid flow).

order of accuracy while perturbations of 10^0 yield solutions which are not Lipschitz continuous, i.e. physically invalid.

While all methods displayed reasonable robustness to these perturbations, the method of Hager and Zhang [18] emerged as the most robust when the debug parameter was set to 'true'.

5. Discussion and conclusions

The problem of inflow parameter 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 6) 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 loss of information (entropy increasing) under dissipation and shock formation; see for example Figure 16, 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 12). 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 CG methods are known to possess regularization properties [4,20]. 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.

$$\nabla\epsilon(\mathbf{x}^n) = -\mathbf{H}\Delta\mathbf{x}^n. \tag{35}$$

Here the superscript n denotes the minimization iteration count.

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

And finally

$$\Delta\mathbf{x}^{n+1} = (\mathbf{I} - \tau H)\,\Delta\mathbf{x}^n. \tag{36}$$

If the initial guess $\Delta\mathbf{x}^0$ is expanded over Hessian eigenvectors ($\mathbf{H}\mathbf{U}_\alpha = \lambda_\alpha\mathbf{U}_a$, where U_α, λ_α are the eigenvectors and eigenvalues), $\Delta x^0 = \sum_j C_j U_j$, 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.

$$\Delta x^n = \sum_j C_j U_j (1 - \tau\lambda_j)^n, \quad \tau \sim \frac{b}{\lambda_{\max}} m, \;\; 0 < b < 1. \tag{37}$$

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 minimization methods under consideration (L-BFGS, TN and hybrid) are found to provide a much faster convergence rate in comparison with the usual nonlinear CG method (excluding the new CG descent algorithm) and a similar stability. 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.

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 CGs while exhibiting 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 TN method provided a good final quality of optimization while exhibiting a relatively slower rate of convergence. The version of CG of Hager [18,19] demonstrated excellent results for both viscous flow and inviscid flow when one follows closely the instructions in the user manual. Private communications with Prof. Hager helped us with this task.

Figure 8 shows also the impact of tuning the $k1$ and $k2$ parameters in the hybrid algorithm [23]. A suitable tuning, which is obviously case dependent, permits the hybrid method to achieve in our case the second best performance among large-scale unconstrained optimization methods tested for the inviscid case, the best performance being exhibited by CG-descent of Hager [18]. The hybrid algorithm achieves the best results for the viscous case followed closely by the CG-descent algorithm of Hager.

Therefore, the numerical results obtained for our test case demonstrate that the new hybrid method (also referred to as the enriched method [23]) and the new CG-descent method [18], once suitably tuned, should be considered serious alternatives to both the T–N and L-BFGS methods, especially since it is known (see e.g. [28]) that Newton-type methods are more effective than the limited-memory quasi-Newton L-BFGS method on ill-conditioned problems.

Another implication of this research is the possibility of reusing existing minimization techniques for the minimization of noisy functions as the minimization methods here proved to be robust in the presence of noise, especially the method of Hager [18], see Kelley [21] for more on noisy function minimization.

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 Dr William Hager sizably improved the presentation and content of the paper and are gratefully acknowledged.

Professor Navon acknowledges the support from NSF grants number ATM-0201808, managed by Dr Linda Pang, and CCF-0635162, managed by Dr Eun K. Park.

References

[1] L.M. Adams and J.L. Nazareth, *Linear and nonlinear conjugate gradient-related methods*, Proceedings of the AMS-IMS-SIAM Summer Research Conference held at the University of Washington, July 1995, SIAM, 1996.

[2] A.K. Alekseev, *On estimation of entrance boundary parameters from downstream measurements using adjoint approach*, Int. J. Numer. Methods Fluids 36 (2001), pp. 971–982.

[3] A.K. Alekseev and I.M. Navon, *The analysis of an ill-posed problem using multiscale resolution and second order adjoint techniques*, Comput. Methods Appl. Mech. Eng. 190(15–17) (2001), pp. 1937–1953.

[4] 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.

[5] E.M.L. Beale, *A derivation of conjugate gradients*, in *Numerical Methods for Nonlinear Optimization*, F.A. Lootsma, ed., Academic Press, London, 1972.

[6] D.N. Daescu and I.M. Navon, *An analysis of a hybrid optimization method for variational data assimilation*, Int. J. Comput. Fluid Dyn. 17(4) (2003), pp. 299–306.

[7] B. Das, H. Meirovitch, and I.M. Navon, P*erformance of enriched methods for large scale unconstrained optimization as applied to models of proteins*, J. Comput. Chem. 24(10) (2003), pp. 1222–1231.

[8] W.C. Davidon, *Variable metric method for minimization*, SIAM J. Optim. 1 (1991), pp. 1–17.

[9] J.E. Dennis, Jr and J.J. More, *Quasi-Newton methods, motivation and theory*, SIAM Rev. 19 (1977), pp. 46–89.

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

[11] R. Fletcher and M.J.D. Powell, *A rapidly convergent descent method for minimization*, Comput. J. 6 (1963), pp. 163–168.

[12] J.C. Gilbert, *On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization*, SIAM J. Optim. 7(3) (1997), pp 780–813.

[13] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods*, SIAM J. Optim. 2 (1992), pp. 21–42.

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

[15] M.D. Gunzburger, *Adjoint equation-based methods for control problems in viscous, incompressible flows*, Flow Turbul. Comb. 65 (2000), pp. 249–272.

[16] M.D. Gunzburger, *Perspectives in Flow Control and Optimization (Advances in Design and Control)*, SIAM, 2003.

[17] W.W. Hager, *Runge–Kutta methods in optimal control and the transformed adjoint system*, Numerische Mathematik 87(2) (2000), pp. 247–282.

[18] W.W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and efficient line search*, SIAM J. Optim. 16(1) (2005), pp. 170–192.

[19] ———, *Algorithm 851: CG DESCENT, A conjugate gradient method with guaranteed descent*, ACM Trans. Math. Softw. 32 (2006), pp. 113–137.

[20] P.C. Hansen, *Rank Deficient and Discrete Ill-posed Problems*, SIAM, Philadelphia, 1998, 247pp.

[21] C. T. Kelley, *Iterative Methods for Optimization*, SIAM, Philadelphia, 1999, xvi + 180pp.

[22] D.C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale minimization*, Math. Program. 45 (1989), pp. 503–528.

[23] J.L. Morales and J. Nocedal, *Enriched methods for large-scale unconstrained optimization*, Comput. Optim. Appl. 21 (2002), pp. 143–154.
[24] ———, *Automatic preconditioning by limited memory quasi-Newton updating*, SIAM J. Optim. 10(4) (2000), pp.1079–1096.
[25] ———, *Algorithm PREQN: FORTRAN subroutines for preconditioning the conjugate gradient method*, ACM Trans. Math. Softw. 27 (2001), pp. 83–91.
[26] S.G. Nash, *Preconditioning of truncated Newton methods*, SIAM J. Sci. Stat. Comput. 6 (1985), pp. 599–616.
[27] ———, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal. 21 (1984), pp. 770–788.
[28] 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. 1 (1991), pp. 358–372.
[29] I.M. Navon, X. Zou, J. Derber, and J. Sela, *Variational data assimilation with an adiabatic version of the NMC spectral model*, Monthly Weather Rev. 120(7) (1992), pp. 1433–1446.
[30] J. Nocedal, *Theory of algorithms for unconstrained minimization*, Acta Numerica 1 (1992), pp. 199–242.
[31] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Verlag, 1999, 656pp.
[32] D.F. Shanno, *Conjugate gradient methods with inexact searches*, Math. Oper. Res. 3 (1978), pp. 244–256.
[33] D.F. Shanno and K.H. Phua, *Remark on algorithm 500. Minimization of unconstrained multivariate functions*, ACM Trans. Math. Softw. 6 (1980), pp. 618–622.
[34] 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. 4 (1995), pp. 241–262.
[35] X. Zou, I.M. Navon, M. Berger, K.H. Phua, T.Schlick, and F.X. Le Dimet, *Numerical experience with limited memory quasi-Newton and truncated Newton methods*, SIAM J. Optim. 3(3) (1993), pp. 582–608.