Optimization Methods and Software
Vol. 00, No. 00, January 2009, 1–32

RESEARCH ARTICLE

Application of Scaled Nonlinear Conjugate Gradient Algorithms To The Inverse Natural Convection Problem

Jeff C.-F. WONG[a*]

[a]*Department of Mathematics, Lady Shaw Building, The Chinese University of Hong Kong, Shatin, Hong Kong*

Bartosz Protas[b]

[a]*Department of Mathematics & Statistics, McMaster University , 1280 Main Street West Hamilton, Ontario Canada L8S 4K1*

(*v4.4 released October 2008*)

The inverse natural convection problem (INCP) in a porous medium is a highly non-linear problem because of the nonlinear convection and Forchheimer terms. The INCP can be converted into the minimization of a least-squares discrepancy between the observed and the modelled data. It has been solved using different classical optimization strategies that require a monotone descent of the objective function for recovering the unknown profile of the time-varying heat source function. In this investigation we use this PDE-constrained optimization problem as a demanding testbed to compare the performance of several state-of-the-art variants of the conjugate gradients approach. We propose solving the INCP using the scaled nonlinear conjugate gradient (SCG) method: a low-cost and low-storage optimization technique. The method presented here uses the gradient direction with a particular spectral step length and the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating formula without any matrix evaluations. Two adaptive line search approaches are numerically studied in which there is no need for solving the sensitivity problem to obtain the step length directly, and are compared to an exact line search approach. We combine the proposed optimization scheme for INCP with a consistent splitting scheme for solving systems of momentum, continuity and energy equations and a mixed finite element method. We show a number of computational tests which demonstrate that the proposed method performs better than the classical gradient method by improving the number of iterations required and reducing the computational time. We also discuss some practical issues related to the implementation of the different methods.

Keywords: Inverse natural convection problem; adjoint variable method; scaled conjugate gradient method; consistent splitting method; mixed finite elements

AMS Subject Classification: F1.1; F4.3 **(... for example; authors are encouraged to provide two to six AMS Subject Classification codes)**

1. Introduction

The classical conjugate gradient (CCG) algorithm is probably the most widely used approach for minimization of the least-squares functions in inverse heat conduction [1–3] and inverse natural conduction (INC) applications [4–6], to mention just a few references. Based on the CCG method as an iterative regularization technique that requires only first-order information (i.e., the gradient function), the

*Corresponding author. Email: jwong@math.cuhk.edu.hk

ISSN: 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/1055678xxxxxxxxxxxx
http://www.informaworld.com

INC problem generally consists of three coupled systems, namely, the governing systems for the direct problem, the adjoint problem and the sensitivity problem, where the unknown strength of a time-varying heat source $G(t)$ caused by the natural convection flow has to be recovered by measuring some temperatures in the interior of the domain. The adjoint problem is used for computing the conjugate gradient direction, while the sensitivity problem is used for obtaining the optimal step length. The computation cost will rise when the problem size becomes large. In addition, it is an ill-posed problem because a small measurement error induces a large estimated error.

Estimating the unknown profile of $G(t)$ can be considered in the framework of unconstrained optimization as the following minimization problem

$$\min_{G(t)\in\mathbb{R}^n} J(G(t)),$$

where $J : \mathbb{R}^n \to \mathbb{R}$ is the cost functional and its gradient $g = g(G(t)) = \nabla J(G(t))$ can be computed through the adjoint method.

Problems of heat transfer in porous media are encountered in petroleum reservoir engineering, and geothermal reservoir engineering. As a direct problem, various choices of numerical scheme for solving a natural convection problem in a fluid-saturated (or a single-phase) porous medium were reported by Lewis *et al* [7]. Recently, we formulated the INCP in a porous medium that focused on a special treatment of the nonlinear convection and Forchheimer terms using the primitive variable formulation (similar topics but different types of governing equations in a porous medium can be found in [8, 9]). Emphasis is also placed upon the numerical convergence of the unknown time-varying heat source profiles by means of the Fletcher-descent gradient expression using the CCG method [10] (see Algorithm 1). Other descent optimization strategies are based on the classical Newton or Gauss-Newton methods requiring second-order information (i.e., Hessian matrix) for finding the solution of the ill-posed problem [11]. In general, any descent optimization method is guaranteed to decrease the objective function monotonically. However, this fact does not imply convergence from any initial model and therefore an exhaustive line search procedure is required to ensure convergence (e.g., [12]), which is what happens in our numerical experiments [10]. Therefore, an important step in a classical gradient descent optimization approach is the choice of the step length which can be obtained by a line search method. One of the methods relies on the solution of the sensitivity problem. One learns that an efficient line search guarantees a sufficient decrease in the objective function relative to the step length which ensures convergence to a stationary point. The question arises: is it possible to avoid the calculation of the sensitivity problem when searching for the step length?

Another noted optimization technique is the scaled nonlinear conjugate gradient (SCG) algorithm. As reported by Hager and Zhang [13], the SCG method is classified as a non-linear CG method. The SCG has superior characteristics as compared to the CCG because there is no need to solve a system of sensitivity equations, rather it uses the adaptive monotone and nonmonotone learning strategies to obtain the step length, which is based on an iteration process. As we shall see below, the SCG is defined and customized for solving back-propagation learning problems which is also suitable for our purpose. Different versions of SCG for various applications exist in the literature, for example, [14–19]. In this paper, we shall closely follow a SCG version of Andrei's formulation [20, 21]. The key features of this algorithm can be summarized as follows:

(1) The spectral conjugate gradient method is used which not only preserves the general characteristics of classical conjugate gradient methods, but also contains a quasi-Newton BFGS updating formula without involving any matrix computations, other than using a few scalar product calculations.
(2) To ensure better performance, the search direction is periodically restarted when all scaled conjugate gradient algorithms are used. The restart procedure of Birgin and Martínez is used.
(3) The parameter scaling for the gradient is chosen as the spectral gradient by means of using two consecutive values.

The main focus of this work is to evaluate the accuracy and efficiency of two commonly found methods, namely the adaptive monotone and nonmonotone SCG methods for the estimation of the unknown heat source in INCP in a porous medium. Using the INCP as a demanding test problem, our goal is to benchmark these optimization methods in order to understand the different trade-offs affecting their performance, especially in comparison to the classical approaches.

All the field variables in the direct, adjoint and sensitivity problems are computed using a second-order decoupled scheme based on the mixed finite element (FE) formulation which, to the best of our knowledge, is a novel approach in a PDE–constrained optimization problem. In order to reduce the numerical boundary layer effect for the pressure, which stemmed from the Hodge-Helmholtz decomposition in conjunction with the pressure incremental scheme, Shen and Guermond [22] proposed an efficient scheme for solving the decoupling of the velocities and pressure consecutively at each time step, which they named the consistent splitting scheme (CSS). In this work, the CSS was used for divorcing the velocity and pressure field variables from the momentum equations and handling the continuity equations. Hence, the accuracy of the energy equation corresponding to the direct, the adjoint and the sensitivity problems will be improved.

The paper consists of six sections. In Section 2, we introduce the natural convection problem in a porous medium as a direct problem. In Section 3, we introduce the formulation of inverse analysis and the concept of the so-called iterative regularization technique. In Section 3, we also revisit the adjoint variable method for finding the solution of the unknown profile of the time varying heat source using the CCG method, where the adjoint and sensitivity problems are formulated. In Section 4, we introduce two versions of adaptive SCG algorithms for solving the INCP which will be used in the following sections. In Section 5, we examine the feasibility and stability of the CSS with the mixed FE as a vehicle for solving the INCP. Section 6 contains few summarizing remarks. A second-order scheme in space and in time for solving the INCP is formulated in two Appendices.

2. Direct Problem

Let Ω be a bounded domain in the Euclidean space $\mathbb{R}^2$ with a piecewise smooth boundary $\partial\Omega$. A fixed final time is denoted by t_f. Let $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, where $\partial\Omega$ consists of two piecewise smooth boundaries. A brief outline of the governing equations for the direct problem is given below, whereas equations for the adjoint and the sensitivity problems will be presented in Section 3. The aim of the direct problem is to determine a time history of the velocity, the pressure and the temperature fields driven by an *estimated* strength of a time-varying heat source $G(t)$ in $(0, t_f]$. The non-dimensional momentum, continuity and energy equations in a porous medium at the representative elementary volume (REV) scale can be written as follows:

Direct Problem

$$\begin{cases} \frac{1}{\epsilon}\frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{\epsilon^2}(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\nabla p + \frac{R_\nu Pr}{\epsilon}\nabla^2\boldsymbol{u} - \left(\frac{Pr}{Da} + \frac{F_\epsilon}{\sqrt{Da}}|\boldsymbol{u}|\right)\boldsymbol{u} \\ \qquad\qquad +RaPrT\boldsymbol{j} & \text{in } \Omega\times(0,t_f] \\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega\times(0,t_f] \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \partial\Omega\times(0,t_f] \\ \sigma_r\frac{\partial T}{\partial t} + (\boldsymbol{u}\cdot\nabla)T = \alpha_m\nabla^2 T + G(t)\delta_k(x-x^\dagger)\delta_k(y-y^\dagger) & \text{in } \Omega\times(0,t_f] \\ T|_{\partial\Omega_1} = 0 \text{ and } \left.\frac{\partial T}{\partial x}\right|_{\partial\Omega_2} = 0 & \text{on } \partial\Omega\times(0,t_f] \end{cases} \quad (1)$$

with the initial conditions

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{0} \text{ and } T(\boldsymbol{x}, t=0) = 0 \qquad \text{in } \Omega, \quad (2)$$

where $\boldsymbol{u}(\boldsymbol{x},t) = (u,v)$ is the volume-averaged vector field, $p(\boldsymbol{x},t)$ is the volume-averaged pressure scalar field, $T(\boldsymbol{x},t)$ is the volume-averaged temperature scalar field, $|\boldsymbol{u}| = \sqrt{u^2+v^2}$ is the absolute velocity, $\boldsymbol{x} = (x,y)$ is a point of interest in $\mathbb{R}^2$ and t is the time. From left to right in (1), the momentum equation is a balance between the following: unsteady, inertia, pressure gradient, viscous (or Brinkman), Darcy, Forchheimer, and buoyancy terms.

The following non-dimensional variables are defined:

$$x = \frac{x'}{d_x},\ y = \frac{y'}{d_y},\ t = \frac{\alpha_m t'}{d_y^2},\ \boldsymbol{u} = \frac{d_y\boldsymbol{u}'}{\alpha_m},\ T = \frac{T'-T'_{\text{cold}}}{T'_{\text{hot}}-T'_{\text{cold}}},$$

$$p = p^\star - (T'_{\text{cold}} - T'_{\text{ref}})\frac{d_y^3}{\alpha_m^2}\alpha g y,\ p^\star = \frac{d_y^2 p'}{\rho\alpha_m^2},$$

and $T'_{\text{ref}} = \frac{1}{2}(T'_{\text{hot}}+T'_{\text{cold}})$ is the average temperature of the system, ϵ is the porosity of the porous medium, R_ν is the viscosity ratio, $Pr = \dfrac{\nu}{\alpha_m}$ is the Prandtl number, $Da = \dfrac{K}{d_x^2}$ is the Darcy number, and $Ra = \alpha g\dfrac{(T'_{\text{hot}}-T'_{\text{cold}})d_y^3}{\alpha_m\nu}$ is the Rayleigh number, where σ_r is the ratio between the heat capacities of the solid and the fluid phases, K is the permeability, $F_\epsilon = \dfrac{1}{\epsilon^{3/2}}\dfrac{1.75}{\sqrt{150}}$ is the inertial coefficient, α is the thermal expansion, g is the gravitation constant, α_m is the thermal diffusivity, ρ is the density, ν is the kinematic viscosity, $\boldsymbol{j}$ is the unit vector in the y-direction, T'_{hot} is the nominal bottom temperature, T'_{cold} is the dimensional temperature at the top boundary, d_x is half of the width and d_y is half of the depth of the cavity.

The dimensionless strength of the heat source $G(t)$ is related to the dimensional strength $G'(t')$ as follows:

$$G(t) = \frac{G'(t')d_y}{(T'_{\text{hot}}-T'_{\text{cold}})\tilde{k}d_x^3},$$

where $\tilde{k}$ is thermal conductivity. The characteristic temperature of the system T'_{hot} is related to the characteristic magnitude of the dimensional heat source G'_{ref} by

the following relation:

$$T'_{\text{hot}} - T'_{\text{cold}} = \frac{G'_{\text{ref}}}{\tilde{k}}.$$

Combining Ra and the above result gives

$$Ra = \alpha g \frac{d_y^3 G'_{\text{ref}}}{\alpha_m \nu \tilde{k}}.$$

The function $\delta_k(x - x^\dagger)$ approximates the point source in the cavity and becomes the Dirac delta function as k tends to infinity, and is defined by

$$\delta_k(x - x^\dagger) = \frac{k}{2\cosh^2(k(x - x^\dagger))}.$$

Then, $\delta_k(y - y^\dagger)$ can be defined similar to $\delta_k(x - x^\dagger)$. In this work, $k = 20$ is used.

3. Formulation of Inverse Analysis

It is convenient from both the theoretical and computational point of view to formulate inverse problems as optimization problems. The objective function $J = J(G(t))$ of the optimization problem may have the form of the sum of the squares of the difference between the calculated temperature $T(\boldsymbol{x}_m, t) = T(\boldsymbol{x}_m, t; G(t))$ and the measured temperature $T^\dagger(\boldsymbol{x}_m, t)$, namely:

$$J = \frac{1}{2} \sum_{m=1}^{\mathrm{M}} \int_0^{t_f} \Big(T(\boldsymbol{x}_m, t) - T^\dagger(\boldsymbol{x}_m, t) \Big)^2 \, dt, \tag{3}$$

where M denotes the total number of temperature measurements. The calculated temperature $T(\boldsymbol{x}_m, t)$ is obtained from (1) - (2), whereas the measured temperature $T^\dagger(\boldsymbol{x}_m, t)$ at the cavity $\boldsymbol{x}_m = (x_m, y_m)$, $m = 1, \cdots, \mathrm{M}$ is assumed to be known. To get these measured values, one solves (1) - (2) with some known time-varying heat source $G(t)$ at an interior point $\boldsymbol{x}^\dagger = (x^\dagger, y^\dagger)$ of the spatial domain.

The INC problem in a porous medium consists of finding the strength of a time-varying heat source $G(t)$ at an interior point $\boldsymbol{x}^\dagger = (x^\dagger, y^\dagger)$ of the spatial domain while the temperature measurements at the cavity $\boldsymbol{x}_m = (x_m, y_m)$ are available as a function of time. Figure 1 shows the geometry, coordinates, source and sensor (or thermocouple) locations of the INCP.

3.1. *Minimization with classical conjugate gradient method*

The iterative CCG method, as applied to the estimation of the time-varying heat source $G(t)$, is given by [23]:

$$G^{K+1}(t) = G^K(t) - \rho^K d^K(t), \tag{4}$$

where the superscript K denotes the number of iterations (or the loop of CCG iteration). The direction of descent $d^K(t)$ is obtained as a conjugation of the gradient

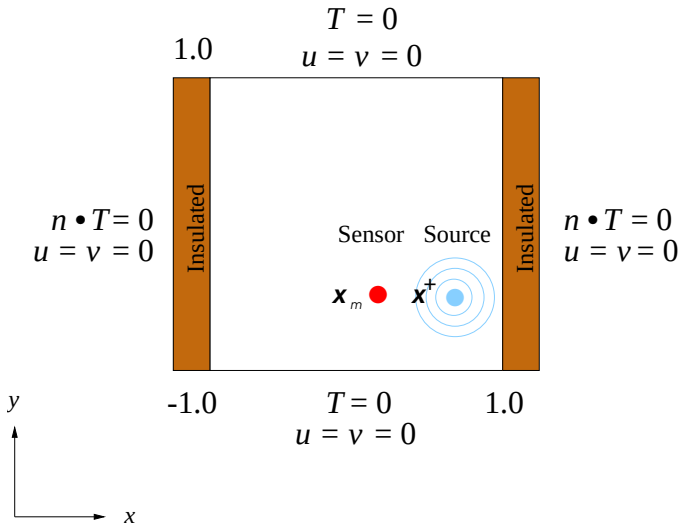


Figure 1. Geometry, coordinates, source and sensor (or thermocouple) locations.

direction $\nabla J^K(t) = \nabla J\left(G^K(t)\right)$ and of the previous direction of descent as:

$$d^K(t) = \nabla J^K(t) + \gamma^K d^{K-1}(t), \tag{5}$$

where $\nabla J^K(t) = \nabla J(G^K(t))$ and γ^K is the conjugation coefficient that can be obtained from the Polak-Ribiere, the Fletcher-Reeves, or the Fletcher conjugate descent expression, just to name a few possible choices.

In order to implement the iterative algorithm given by (4) - (5), one usually needs to develop expressions for the optimal step length ρ^K and for the gradient direction $\nabla J^K(t)$ by making use of two auxiliary problems, known as the sensitivity problem and the adjoint problem respectively.

3.2. *Iterative Regularization Technique as a Stopping Criterion*

Recording and analysis of measured data are important tasks in the formulation of the INC problem. Unfortunately, due to the ubiquity of noise, the measured temperature data are always corrupted by noise to some degree. In this paper, we assume that the measurements containing random errors were obtained by adding an error term to the error-free measurements resulting from the solution of the direct problem:

$$T_{\text{measured}}(= T^{\dagger}) = T_{\text{free}} + \omega\sigma \tag{6}$$

where T_{free} are the error-free measurements; ω is a random variable with normal distribution, zero mean, and unitary standard deviation; and σ is the standard deviation of the measurement errors, which is supposed constant. If a constant error level is assumed throughout the measurement, then

$$T(\boldsymbol{x}_m, t) - T_{\text{measured}}(\boldsymbol{x}_m, t) \simeq \sigma \tag{7}$$

is admissible and advisable. Two cases are considered [24]:

(1) By ignoring the noise, i.e., $\sigma = 0$, formula (4) is iterated until $G(t)$ satisfies

the following stopping criteria:

$$\frac{\| G^{K+1} - G^K \|^2_{L^2(0,t_f)}}{\| G^K \|^2_{L^2(0,t_f)}} \leq \varepsilon_1,$$

where $\| f \|^2_{L^2(0,t_f)} = \int_0^{t_f} (f(t))^2 \, dt$ is the norm of a function f belonging to $L^2(0,t_f)$ space, G^K is the K^{th} iterate for $G(t)$ and ε_1 is a fixed small number ranging from 10^{-3} to 10^{-10}. In what follows, we adopt the simplified notation $\| * \| = \| * \|_{L^2(0,t_f)}$. The selection of tolerance, ε_1 which depends on the choice of the specific time-varying heat source will vary.

(2) In the case of the measured temperature data with noise , i.e., $\sigma \neq 0$, the iterative process is stopped based on the residual criterion or the discrepancy principle [25], i.e. upon fulfilment of the stopping criterion:

$$J(G^K(t)) \leq \varepsilon_2, \tag{8}$$

where ε_2 is defined by

$$\varepsilon_2 = \frac{1}{2} \sum_{m=1}^{\mathrm{M}} \int_0^{t_f} \sigma^2 dt. \tag{9}$$

Equation (9) can be interpreted as the integrated error of the measured data having a constant standard deviation σ on a whole time interval $[0, t_f]$. If the function J has a minimum value that is larger than ε_2, the following criterion is used to stop the iteration:

$$J(G^{K+1}) - J(G^K) < \epsilon_t, \tag{10}$$

where ϵ_t is a prescribed small number.

Many iterative methods exhibit a self-regularizing property in the sense that early termination of the iterative process has a regularizing effect [26].

There are two main features of the iterative regularization technique (IRT):

(1) the iteration index K plays the role of the regularizing parameter α used in the Tikhonov method [27], and
(2) the stopping rule (e.g., (8) or (10)) plays the role of the parameter selection method.

Physically speaking, the expected solution is assumed to be sufficiently accurate and close to the exact one when the difference between the measured and estimated heat source has fallen into the specified order of magnitude of the measurement errors (see (7)).

3.3. *Adjoint Variable Method*

The use of the CG method for the minimization of the objective function J in (3) requires the solution of auxiliary problems, known as sensitivity and adjoint problems. The sensitivity problem given here is used to determine the variation of the dependent variables due to changes in the unknown quantity. Hence, the temperature sensitivity δT can be defined as the directional derivative of T at G

in the direction of δG:

$$\delta T = \lim_{\varepsilon \to 0} \frac{T(G + \varepsilon \delta G) - T(G)}{\varepsilon}. \tag{11}$$

The remaining variables can be defined similar to (11). Based on the definition of the sensitivity problem, the following sets of sensitivity equations for the momentum, continuity and energy equations together with the same initial and boundary conditions as their counterparts given in the direct problem are easily obtained:

Sensitivity Problem

$$\begin{cases} \frac{1}{\epsilon}\frac{\partial \delta \boldsymbol{u}}{\partial t} + \frac{1}{\epsilon^2}\left(\delta \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \delta \boldsymbol{u}\right) = -\nabla \delta p \\ \qquad + \frac{R_\nu Pr}{\epsilon}\nabla^2 \delta \boldsymbol{u} - \frac{Pr}{Da}\delta \boldsymbol{u} - \frac{F_\epsilon}{\sqrt{Da}}\left(\frac{\boldsymbol{u} \cdot \delta \boldsymbol{u}}{|\boldsymbol{u}|}\boldsymbol{u} + |\boldsymbol{u}|\delta \boldsymbol{u}\right) + RaPr\delta T \boldsymbol{j} & \text{in } \Omega \times (0, t_f] \\ \nabla \cdot \delta \boldsymbol{u} = 0 & \text{in } \Omega \times (0, t_f] \\ \delta \boldsymbol{u} = \boldsymbol{0} & \text{on } \partial\Omega \times (0, t_f] \\ \sigma_r \frac{\partial \delta T}{\partial t} + \delta \boldsymbol{u} \cdot \nabla T + \boldsymbol{u} \cdot \nabla \delta T = \alpha_m \nabla^2 \delta T + d(t)\delta_k(x - x^\dagger)\delta_k(y - y^\dagger) & \text{in } \Omega \times (0, t_f] \\ \delta T|_{\partial\Omega_1} = 0 \text{ and } \left.\frac{\partial \delta T}{\partial x}\right|_{\partial\Omega_2} = 0 & \text{on } \partial\Omega \times (0, t_f] \end{cases} \tag{12}$$

with the initial conditions

$$\delta \boldsymbol{u}(\boldsymbol{x}, t = 0) = \boldsymbol{0} \text{ and } \delta T(\boldsymbol{x}, t = 0) = 0 \qquad \text{in } \Omega, \tag{13}$$

where $\delta \boldsymbol{u} = (\delta u, \delta v)$ is the sensitivity of the velocity vector field, δp is the sensitivity of the pressure scalar field, δT is the sensitivity of the temperature scalar field and $d(t)$ is a conjugate search direction that is determined by the iterative procedure of the conjugate gradient method.

To minimize J with respect to G, in an infinite-dimensional space, one needs the gradient of J, ∇J, that is given by

$$\delta J(G; \delta G) = \int_0^{t_f} \nabla J \delta G dt. \tag{14}$$

The function $\delta J(G)$ is interpreted as the directional derivative of J at G in the direction of δG.

With the definition of the objective function in (3), Equation (14) might be expressed using the Dirac delta function as an integral over the space domain and time, where all quantities are determined at the specified measurement location $\boldsymbol{x}_m$. Introducing the adjoint variables $\boldsymbol{\xi}$, q, and η as the Lagrange multipliers, we will ensure that the state variables corresponding to the minimum of objective function (3) will also satisfy the the direct problem. Hence, the direct problem is

a constraint to the minimization of (3). Now, the function $\delta J(G;\delta G)$ becomes

$$\begin{aligned}\delta J(G) = &\int_0^{t_f}\int_\Omega \left(T(\boldsymbol{x}_m,t) - T^\dagger(\boldsymbol{x}_m,t)\right)\delta T \sum_{m=1}^{\mathrm{M}} \delta(x-x_m)\delta(y-y_m)d\Omega dt \\ &+ \int_0^{t_f}\int_\Omega \boldsymbol{\xi}\cdot\left(\frac{1}{\epsilon}\frac{\partial\delta\boldsymbol{u}}{\partial t} + \frac{1}{\epsilon^2}\left(\delta\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\delta\boldsymbol{u}\right) + \nabla\delta p\right. \\ &\left. -\frac{R_\nu Pr}{\epsilon}\nabla^2\delta\boldsymbol{u} + \frac{Pr}{Da}\delta\boldsymbol{u} + \frac{F_\epsilon}{\sqrt{Da}}\left(\frac{\boldsymbol{u}\cdot\delta\boldsymbol{u}}{|\boldsymbol{u}|}\boldsymbol{u} + |\boldsymbol{u}|\delta\boldsymbol{u}\right) - RaPr\delta T\boldsymbol{j}\right)d\Omega dt \\ &+ \int_0^{t_f}\int_\Omega q\left(\nabla\cdot\delta\boldsymbol{u}\right)d\Omega dt \\ &+ \int_0^{t_f}\int_\Omega \eta\left(\sigma_r\frac{\partial\delta T}{\partial t} + \delta\boldsymbol{u}\cdot\nabla T + \boldsymbol{u}\cdot\nabla\delta T - \alpha_m\nabla^2\delta T - \delta G(t)\delta_k(x-x^\dagger)\delta_k(y-y^\dagger)\right)d\Omega dt,\end{aligned} \tag{15}$$

where $\delta(\cdot)$ is the Dirac delta function and $\boldsymbol{\xi} = (\xi^x, \xi^y)$.

By performing integration by parts on the last three terms appearing on the right-hand side of (15), applying the initial and boundary conditions of the sensitivity problem for δT and also requiring that the coefficients of δT in the resulting relations vanish, the following adjoint problem is formed:

Adjoint Problem

$$\begin{cases} \dfrac{1}{\epsilon}\dfrac{\partial\boldsymbol{\xi}}{\partial\tau} - \dfrac{1}{\epsilon^2}\boldsymbol{u}\cdot\nabla\boldsymbol{\xi} = -\nabla q + \dfrac{R_\nu Pr}{\epsilon}\nabla^2\boldsymbol{\xi} - \dfrac{1}{\epsilon^2}\boldsymbol{\xi}\cdot\left(\nabla\boldsymbol{u}\right)^{\mathrm{T}} - \eta\nabla T \\ \qquad -\dfrac{Pr}{Da}\boldsymbol{\xi} - \dfrac{F_\epsilon}{\sqrt{Da}}\dfrac{|\boldsymbol{u}|^2\boldsymbol{\xi} + \left(\boldsymbol{\xi}\cdot\boldsymbol{u}\right)\boldsymbol{u}}{|\boldsymbol{u}|} & \text{in } \Omega\times(0,t_f] \\ \nabla\cdot\boldsymbol{\xi} = 0 & \text{in } \Omega\times(0,t_f] \\ \boldsymbol{\xi} = \boldsymbol{0} & \text{on } \partial\Omega\times(0,t_f] \\ \sigma_r\dfrac{\partial\eta}{\partial\tau} - \boldsymbol{u}\cdot\nabla\eta = \alpha_m\nabla^2\eta + RaPr\xi^y \\ \qquad + \sum_{m=1}^{\mathrm{M}}\left[T(x,y,\tau) - T^\dagger(x,y,\tau)\right]\delta(x-x_m)\delta(y-y_m) & \text{in } \Omega\times(0,t_f] \\ \eta|_{\partial\Omega_1} = 0 \text{ and } \left.\dfrac{\partial\eta}{\partial x}\right|_{\partial\Omega_2} = 0 & \text{on } \partial\Omega\times(0,t_f] \end{cases} \tag{16}$$

with the final conditions

$$\boldsymbol{\xi}(\boldsymbol{x}, t=t_f) = 0 \text{ and } \eta(\boldsymbol{x}, t=t_f) = 0 \qquad \text{in } \Omega, \tag{17}$$

where $\boldsymbol{\xi}$ is the adjoint velocity field, q is the adjoint pressure field, η is the adjoint temperature field and the superscript T denotes the transpose. To solve adjoint problem (16) backward in time with the end condition at the physical time $t = t_f$, one usually proceeds with the change of time variable $\tau = t_f - t$. Through this transformation, the adjoint problem becomes an initial value problem in τ.

In Equation (14), the directional derivative of the functional in the direction $\delta J(G;\delta G)$ simplifies to

$$\delta J(G;\delta G) = \int_0^{t_f}\int_\Omega \eta\delta_k(x-x^\dagger)\delta_k(y-y^\dagger)d\Omega\delta G(t)dt. \tag{18}$$

By comparing (14) and (18), the gradient functional ∇J might be found as follows:

$$\nabla J = \int_{\Omega} \eta \delta_k(x - x^{\dagger}) \delta_k(y - y^{\dagger}) d\Omega. \tag{19}$$

The minimization of the objective function is accomplished by an iteration regularization, using (4) - (5). The sequence of approximations for the unknown strength of a time-varying heat source may be constructed using the following steps of the CCG method, as shown in Algorithm 1.

Algorithm 1 Classical Conjugate Gradient

1: Assume $G^K(t)$, where $G^0(t) = \text{Constant}$.
2: Calculate the velocity and temperature fields by solving (1) and (2) from $t = 0$ to $t = t_f$.
3: Solve adjoint equations (16) and (17) from $\tau = 0$ to $\tau = t_f$.
4: Update the scalar γ^K using the Fletcher-descent expression:

$$\gamma^K = \frac{\int_0^{t_f} \left(\nabla J^K(t)\right)^2 dt}{-\int_0^{t_f} d^{K-1}(t)\nabla J^{K-1}(t)dt}. \tag{20}$$

5: Update the conjugate search direction d^K:

$$\begin{cases} d^0(t) = \nabla J^0(t) = \int_\Omega \eta\delta_k(x - x^\dagger)\delta_k(y - y^\dagger)d\Omega & \text{if } K = 0 \\ d^K(t) = \nabla J^K(t) + \gamma^K d^{K-1}(t) & \text{if } K \geq 1. \end{cases} \tag{21}$$

6: Solve sensitivity equations (12) and (13) from $t = 0$ to $t = t_f$.
7: Determine the optimal step length ρ^K:

$$\rho^K = \text{argmin}\{J(G^K - \rho^K d^K)\} = \frac{\delta J^K}{L^K}, \tag{22}$$

where

$$\delta J^K = \sum_{m=1}^{\text{M}} \int_0^{t_f} \Big(T(\boldsymbol{x}_m, t) - T^\dagger(\boldsymbol{x}_m, t)\Big)\, \delta T(\boldsymbol{x}_m, t)dt \tag{23}$$

and

$$L^K = \sum_{m=1}^{\text{M}} \int_0^{t_f} \left(\delta T(\boldsymbol{x}_m, t)\right)^2 dt. \tag{24}$$

8: Set

$$G^{K+1}(t) = G^K(t) - \rho^K d^K(t).$$

9:

$$\begin{cases} \text{If } \dfrac{\| G^{K+1}(t) - G^{\text{ex}}(t) \|_2^2}{\| G^{\text{ex}}(t) \|_2^2} \text{ or } J(G^K) \text{ is less than tolerance } \varepsilon_1 \text{ or } \varepsilon_2, \text{ STOP} \\ \text{Otherwise, set } K = K + 1, \text{ return to STEP 2} \end{cases}$$

For Algorithm 1, the following observations are made:

- The optimal step length ρ^K in the direction $d^K(t)$ is obtained by minimizing $J(G^K - \rho^K d^K)$ with respect to ρ^K, where

$$J(G^K - \rho^K d^K) = \frac{1}{2}\sum_{m=1}^{\mathrm{M}} \int_0^{t_f} \Big(T(\boldsymbol{x}_m; G^K - \rho^K d^K) - T^\dagger(\boldsymbol{x}_m, t)\Big)^2 dt, \qquad (25)$$

  and by using a first-order Taylor series approximation for the estimated temperature, we have

$$T(\boldsymbol{x}_m; G^K - \rho^K d^K) = T(\boldsymbol{x}_m; G^K) - \delta T(\boldsymbol{x}_m, t)\rho^K. \qquad (26)$$

  Putting (26) into (25) together with (11), expanding the quadratic integrand term, differentiating (25) partially with respect to ρ^K and setting the resulting equation equal to zero, Equation (22) is immediately obtained.
- We observe that the gradient functional is always zero when $t = t_f$. Hence, the final time values of the estimated history of $G(t)$ cannot be predicted and the estimated values $G(t)$ deviate from the exact values near the final time condition. In this work, we let $\nabla J(t_f) = \nabla J(t_f - \Delta t)$. Therefore, the singularity at $t = t_f$ can be avoided and a reasonable inverse solution can be calculated.
- For each iteration in Algorithm 1, we need to compute 15 unknowns for updating $G(t)$. In what follows, we shall use the SCG method in order to avoid the computation of the sensitivity problem. In essence, we simply ignore Step 6 and Step 7. In turn, using the SCG method, we calculate the direct and adjoint problems twice per iteration.

4. Scaled Conjugate Gradient

In what follows, we shall study a discrete version of the given model. All unknown variables are defined in a finite dimensional space. We consider in reasonable detail a computational algorithm implementing the ideas mentioned in the Introduction. The algorithm uses function values only, employing the scalar product to compute all the scaling parameters when required. First we summarize the basic algorithm which is based on the works of Andrei [20, 21].

4.1. *Spectral Conjugate Gradient Method*

Motivated by the work of Birgin and Martínez [28], the general form of the conjugate gradient method is given by

$$d^K(t) = -\theta^K \nabla J(G^K) + \gamma^K s^{K-1}(t), \qquad (27)$$

where θ^K is a scalar parameter that is to be determined, $\gamma^K = \dfrac{\left(\theta^K y^{K-1} - s^{K-1}\right)^T \nabla J(G^K)}{\left(y^{K-1}\right)^T s^{K-1}}$ is obtained by the quadratic function minimization, $s^{K-1} := s^{K-1}(t) = G^K(t) - G^{K-1}(t)$ and $y^{K-1} := y^{K-1}(t) = \nabla J^K - \nabla J^{K-1} = \nabla J(G^K(t)) - \nabla J(G^{K-1}(t))$.

We observe that if $\theta^K = 1$, then we obtain the classical CG method based on the choice of the scalar parameter γ^K. However, if $\gamma^K = 0$, then another class of algorithms based on the choice of the parameter θ^K is considered. Two possibilities

could occur: a positive scalar or a positive definite matrix. The former case would correspond to the steepest descent algorithm if $\theta^K = 1$, while the latter would give the Newton or the quasi-Newton algorithm if $\theta^K = \left(\nabla^2 J(G^K(t))\right)^{-1}$. Hence, Equation (27) preserves a characteristic mixture of the quasi-Newton and the CG methods if $\theta^K \neq 0$ and $\gamma^K \neq 0$.

By following the ideas of Perry [29] and Shanno [30, 31] and rearranging the expressions in (27), the conjugate search direction can be written as

$$d^K = -\left[\theta^K I - \theta^K \frac{s^{K-1}\left(y^{K-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}} + \frac{s^{K-1}\left(s^{K-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}}\right]\nabla J^K = -Q^K \nabla J^K. \quad (28)$$

Shanno [30, 31] concluded that the CG method is exactly equal to the quasi-Newton BFGS method in which at each step the approximation to the inverse Hessian matrix is restarted as the identity matrix I. One observes that

$$\left(y^{K-1}\right)^T Q^K = \left(s^{K-1}\right)^T, \quad (29)$$

is similar, but not identical, to the quasi-Newton equation, which requires that the inverse Hessian matrix H^K should be updated in such a way as to satisfy:

$$H^K y^{K-1} = s^{K-1}. \quad (30)$$

In order to not only guarantee that the matrix Q^K is symmetric and positive definite, but to also get a true quasi-Newton updating expression, Andrei [20] proposed the following update:

$$\begin{aligned} Q^{K,*} = \theta^K I - \theta^K \frac{y^{K-1}\left(s^{K-1}\right)^T + s^{K-1}\left(y^{K-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}} \\ + \left[1 + \theta^K \frac{\left(y^{K-1}\right)^T y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right] \frac{s^{K-1}\left(s^{K-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}} \end{aligned} \quad (31)$$

which satisfies

$$Q^{K,*} y^{K-1} = s^{K-1}. \quad (32)$$

Therefore, the conjugate search direction is

$$d^K = -Q^{K,*}\nabla J^K. \quad (33)$$

There is no need to directly compute the matrix $Q^{K,*}$, since with the computation of only four scalar products the direction now becomes

$$\begin{aligned} d^K = -\theta^K \nabla J^K + \theta^K \frac{\left(\nabla J^K\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}} y^{K-1} \\ - \left[\left(1 + \theta^K \frac{\left(y^{K-1}\right)^T y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right) \frac{\left(\nabla J^K\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}} - \theta^K \frac{\left(\nabla J^K\right)^T y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right] s^{K-1}, \end{aligned} \quad (34)$$

Like the well-known BFGS update to the inverse Hessian, which is currently by-far the best update of the Broyden class, the inverse Hessian is defined by

$$H^K = H^{K-1} - \frac{H^{K-1}y^{K-1}\left(s^{K-1}\right)^T + s^{K-1}\left(y^{K-1}\right)^T H^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}} + \left[1 + \frac{\left(y^{K-1}\right)^T H^{K-1}y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right]\frac{\left(s^{K-1}\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}, \tag{35}$$

with (31). Equation (35) is exactly the same as the quasi-Netwon BFGS method, where at each step the approximation of the inverse Hessian is restarted as the identity matrix multiplied by the scalar θ^K.

By recalling the result established in [21], we have:

THEOREM 4.1 *Assume that J is strongly convex and Lipschitz continuous on the level set $L_0 = \{t \in \mathbb{R}^n | \, J(G(t)) \leq J(G(t_0))\}$. If at every step of the conjugate gradient (4) with d^K given by (34) and the step length ρ^K selected to satisfy the Wolfe conditions (44) and (45), then either $\nabla J^K = 0$ for some K, or $\lim_{K\to\infty} \nabla J^K = 0$.*

4.2. *Birgin and Martínez Restarts*

A more sophisticated and popular restarting criterion has been proposed by Birgin and Martínez [28] which consists of testing to see if the angle between the current direction and the gradient is sufficiently acute. Therefore, at step $R-1$ when:

$$\left(d^{R-1}\right)^T \nabla J^R > -10^{-3} \parallel d^{R-1} \parallel \parallel \nabla J^R \parallel \tag{36}$$

the algorithm is restarted using the direction given by (34).

Otherwise, for $K-1 \geq R$, we consider the same idea used to get (31), i.e. that of modifying the gradient ∇J^K with a symmetric and positive definite matrix which best estimates the inverse Hessian without any additional storage requirements. Therefore, the direction d^K, for $K-1 \geq R$, is computed using a double update scheme as:

$$d^K = -H^K \nabla J^K \tag{37}$$

where

$$H^K = H^R - \frac{H^R y^{K-1}\left(s^{K-1}\right)^T + s^{K-1}\left(y^{K-1}\right)^T H^R}{\left(y^{K-1}\right)^T s^{K-1}} + \left[1 + \frac{\left(y^{K-1}\right)^T H^R y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right]\frac{\left(s^{K-1}\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}} \tag{38}$$

and

$$H^R = \theta^R I - \theta^R \frac{y^{R-1}\left(s^{R-1}\right)^T + s^{R-1}\left(y^{R-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}} + \left(1 + \theta^R \frac{\left(y^{K-1}\right)^T y^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}\right)\frac{s^{K-1}\left(s^{K-1}\right)^T}{\left(y^{K-1}\right)^T s^{K-1}}. \tag{39}$$

The key features of (38) and (39) do not involve any matrix computation. Two following terms, $H^R\nabla J^K$ and $H^R y^{K-1}$, can simply be calculated as:

$$\vartheta \equiv H^R\nabla J^K = \theta^R\nabla J^K - \theta^R\frac{\left(\nabla J^K\right)^T s^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}y^{R-1} + \left[\left(1+\theta^R\frac{\left(y^{R-1}\right)^T y^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}\right)\frac{\left(\nabla J^K\right)^T s^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}} - \theta^R\frac{\left(\nabla J^K\right)^T y^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}\right]s^{R-1} \tag{40}$$

and

$$\varpi \equiv H^R y^{K-1} = \theta^R y^{K-1} - \theta^R\frac{\left(y^{K-1}\right)^T s^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}y^{R-1} + \left[\left(1+\theta^R\frac{\left(y^{R-1}\right)^T y^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}\right)\frac{\left(y^{K-1}\right)^T s^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}} - \theta^R\frac{\left(y^{K-1}\right)^T y^{R-1}}{\left(y^{R-1}\right)^T s^{R-1}}\right]s^{R-1}, \tag{41}$$

which involves 6 scalar products. Using (40) and (41), the descent direction (37) at any non-restart step can be calculated as:

$$d^K = -\vartheta + \frac{\left(\left(\nabla J^K\right)^T s^{K-1}\right)\varpi + \left(\left(\nabla J^K\right)^T \varpi\right)s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}} - \left(1+\frac{\left(y^{K-1}\right)^T \varpi}{\left(y^{K-1}\right)^T s^{K-1}}\right)\frac{\left(\nabla J^K\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}s^{K-1}, \tag{42}$$

which involves only 4 scalar products.

Andrei [20, 21] defined the search direction d^K from (42) as a double quasi-Newton update scheme. Inspection of (42) indicates that the term $\left(y^{K-1}\right)^T s^{K-1} > 0$ is sufficient to ensure that the direction d^K given by (42) is well-defined, and it always preserves a descent direction.

4.3. *Parameter Scaling*

One of the choices of the scalar θ^K is known to lead to the spectral gradient method, namely:

$$\theta^K = \frac{\left(s^{K-1}\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}, \quad \left(y^{K-1}\right)^T s^{K-1} > 0 \tag{43}$$

which was proposed by Barzilai and Borwein [32], and further analyzed by Raydan [33] and Fletcher [34]. Barzilai and Borwein (43) used two new step length approximations to the secant equation underlying the quasi-Newton methods which can be obtained by minimizing $\| \theta^K s^{K-1} - y^{K-1} \|$. Using the property of symmetry, and by minimizing $\| \theta^K s^{K-1} - y^{K-1} \|$ with respect to θ^K, one gets (43). The scalar parameter given by (43) is the inverse of the Rayleigh quotient (e.g., see [19]).

4.4. *Updating the Step length*

The step length is adapted using the choice of Shanno and Phua in CONMIN [35] which exploits the conjugate search values and the step-length from the previous iteration, where

$$\rho^K = \begin{cases} \dfrac{1}{\| \nabla J(G^0) \|}, & K = 0 \\ \rho^{K-1} \dfrac{\| d^{K-1} \|}{\| d^K \|}, & \text{otherwise.} \end{cases}$$

4.5. *Monotone Line Search*

In general, a line search algorithm is an iterative method which generates a sequence of step length estimates ρ^K that satisfy the Armijo-Wolfe (or Powell-Wolfe) conditions as follows (e.g., [12]):

At iteration K, let $0 < \sigma_1 \leq \sigma_2 < 1$ and let d^K be any descent direction. If

$$J(G^K - \rho^K d^K) - J(G^K) \leq \sigma_1 \rho^K \nabla J(G^K) d^K, \tag{44}$$

and

$$\left(\nabla J(G^K - \rho^K d^K)\right)^T d^K \geq \sigma_2 \left(\nabla J(G^K)\right)^T d^K \tag{45}$$

are satisfied, then ρ^K is updated and the updated time-varying heat source is given by $G^K - \rho^K d^K$; otherwise, ρ^K is reduced [12] when conditions (44) and (45) are not satisfied. We can see from the inequalities (44) and (45) that new function evaluations G^{K+1} are required whenever ρ^K is reduced at iteration K. The first inequality (44), called Armijo's condition, guarantees that the error function is sufficiently reduced at each iteration, while the second inequality (45) called the curvature condition prevents the step length from being too small and it is always positive, since it implies that $\left(s^{K-1}\right)^T y^{K-1} > 0$. Here the step length parameters we used are $\sigma_1 = 10^{-3}$ and $\sigma_2 = 0.9$.

4.6. *Nonmonotone Line Search*

Although a number of algorithms can be proved to be globally convergent using Powell-Wolfe conditions (44) and (45), it is not surprising that such methods would be inefficient in the sense that the iterates may be trapped in a narrow curved valley of objective functions. To remedy this situation, Grippo [36] proposed a nonmonotone learning strategy that uses the accumulated information with regard to the most recent values M of the cost functional:

$$J(G^K - \rho^K d^K) - \max_{0 \leq L \leq M} J(G^{K-L}) \leq \sigma_1 \rho^K \nabla J(G^K) d^K, \tag{46}$$

and

$$\left(\nabla J(G^K - \rho^K d^K)\right)^T d^K \geq \sigma_2 \left(\nabla J(G^K)\right)^T d^K, \tag{47}$$

where M is a positive integer, known as a nonmonotone learning horizon. Some salient features of *nonmonotone Wolfe's conditions* are summarized as follows:

- It was shown in [37] that increasing the function values in *the nonmonotone Wolfe conditions* (46) and (47) has no affect on the global convergence properties.
- The general idea behind nonmonotone strategies is that, in practice, the first choice of a trial point by a minimization algorithm hides a lot of useful information about the problem structure and that such knowledge can be deteriorated by the decrease imposition [18].
- In the first inequality, the integer M controls the amount of monotonicity that is permitted, in comparison with (45) and (47), where both inequalities are identical and do not depend on the choice of step length.

An elegant way for adapting the constant M throughout the local estimation of the Lipschitz constant was proposed in [15] and uses Cauchy's method [38, 39]:

$$M^K = \begin{cases} M^{K-1} + 1, & \text{if } \Lambda^K < \Lambda^{K-1} < \Lambda^{K-2}, \\ M^{K-1} - 1, & \text{if } \Lambda^K > \Lambda^{K-1} > \Lambda^{K-2}, \\ M^{K-1}, & \text{otherwise}, \end{cases} \tag{48}$$

where Λ^K is the local estimation of the Lipschitz constant at the K-th iteration

$$\Lambda^K = \frac{\| \nabla J(G^K) - \nabla J(G^{K-1}) \|}{\| G^K - G^{K-1} \|}. \tag{49}$$

4.7. *Four Algorithms*

Let N be the final number of iterations. Let us summarize the most important features of the considered algorithms as follows:

- The governing, adjoint and sensitivity systems (1) - (2), (16) - (17), and (12) - (13) are discretized using the CSS and the mixed FE methods (see Appendices A and B for details), so that the corresponding discrete problems (A1) - (A4), (A5) - (A8) and (A9) - (A12) are obtained.
- The skeleton of Algorithm 2 is similar to that of Algorithm 1, and is referred to as the modified scaled CG method, however, the search direction (see (34) and (42)) and Birgin and Martínez restarts are used.
- The sequence of approximations for the unknown strength of a time-varying heat source may be constructed using the following steps of the adaptive monotone and adaptive nonmonotone SCG methods in Algorithm 3 and Algorithm 4, respectively. Firstly, there is no need to calculate Steps 19 and 20 in Algorithm 2. Secondly, the difference between these two algorithms concerns the determination of the step length via conditions (44) and (45), or (46) and (47). Hence, the direct and adjoint problems need to be computed twice per iteration. We add the condition on the step length that satisfies these two Wolfe's conditions, i.e., $\rho^K = 2\rho^K$, because we try to avoid the situation where ρ^K reduces rapidly. The steps of adaptive monotone SCG methods are listed in Algorithm 3. Algorithm 4 is then constructed when Steps 21 and 22 in Algorithm 3 are replaced by the following

21: **if** the step-length ρ^K satisfies nonmonotone Wolfe's conditions (46) and (47)
22: **if** $J(G^K - \rho^K d^K) - \max_{0 \le L \le M} J(G^{K-L}) < 10^{-3}$ **then**

Algorithm 2 Modified Scaled Conjugate Gradient

Initiate G^0, and select the parameters $0 < \sigma_1 \leq \sigma_2 < 1$.
for $K := 0$ to N **do**
  if $\nabla J(G^K) = 0$ or $J(G^K) < \varepsilon_2$ **then**
    return $G^* = G^K$; $J(G^*) = J(G^K)$
  end if
  Calculate the velocity and temperature fields by solving (1) and (2) from $t = 0$ to $t = t_f$.
  Solve the adjoint equations using (16) and (17) from $\tau = 0$ to $\tau = t_f$.
  if $K := 0$ **then**
    $d^0 = \nabla J(G^0)$;
  else
    Compute the scaling factor $\theta^K = \dfrac{\left(s^{K-1}\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}$.
    Compute the conjugate direction d^K in (34).
    if Condition (36) is satisfied **then**
      Compute the restart direction d^K in (34);
    else
      Compute the standard direction d^K in (42);
    end if
  end if
  Solve the sensitivity equations using (12) and (13) from $t = 0$ to $t = t_f$.
  Determine the optimal step length ρ^K using (22) - (24).
  Set $G^{K+1} = G^K - \rho^K d^K$.
end for

Algorithm 3 Adaptive Monotone Scaled Conjugate Gradient

Initiate G^0, and select the parameters $0 < \sigma_1 \leq \sigma_2 < 1$.
for $K := 0$ to N **do**
  if $\nabla J(G^K) = 0$ or $J(G^K) < \varepsilon_2$ **then**
    return $G^* = G^K$; $J(G^*) = J(G^K)$
  end if
  Calculate the velocity and temperature fields by solving (1) and (2) from $t = 0$ to $t = t_f$.
  Solve the adjoint equations using (16) and (17) from $\tau = 0$ to $\tau = t_f$.
  if $K := 0$ **then**
    $d^0 = \nabla J(G^0)$;
    $\rho^0 = \dfrac{1}{\| \nabla J(G^0) \|}$;
  else
    Compute the scaling factor $\theta^K = \dfrac{\left(s^{K-1}\right)^T s^{K-1}}{\left(y^{K-1}\right)^T s^{K-1}}$.
    Compute the conjugate direction d^K in (34).
    if Condition (36) is satisfied **then**
      Compute the restart direction d^K in (34);
    else
      Compute the standard direction d^K in (42);
    end if
    $\rho^K = \rho^{K-1} \dfrac{\| d^{K-1} \|}{\| d^K \|}$.
  end if
  if the step length ρ^K satisfies Powell-Wolfe's conditions (44) and (45) **then**
    if $J(G^K - \rho^K d^K) - J(G^K) < 10^{-3}$ **then**
      $\rho^K = 2\rho^K$;
    else
      $\rho^K = \rho^K$;
    end if
  else
    $\rho^K = \dfrac{\rho^K}{2}$;
  end if
  Set $G^{K+1} = G^K - \rho^K d^K$.
end for

5. Results and Discussions

The main goal of our computational experiments is to investigate the performance of Algorithms 1–4 on the INCP. We are interested in two main aspects, namely, the rate of convergence of the different Algorithms as quantified by the number of iterations required to reach a certain level of accuracy in the reconstruction of the heat source function, and also in the actual CPU time required to reach such a level of accuracy. This is related to the fact that iterations performed with different algorithms will typically require different computational times. Two sets of computational experiments, corresponding to temperature measurements with and without noise, cf. (6), are presented to address the two questions mentioned above. In both problems we have adopted the following heat source function, also studied in [4], as the exact solution:

$$G^{\text{ex}}(t) = -\frac{750}{4} t \left(t - \frac{1}{3}\right)^2 (t-1) + 1.0, \qquad \text{where } 0 \le t \le 1. \tag{50}$$

The computer implementation of the discussed Algorithms was done in FORTRAN 90, and represents a further development of the code used for studying the direct problem of natural convection in porous media [40, 41]. All computations were performed on a Linux Cluster with one processor of 2GB of RAM and 2.6GHz using the serial FORTRAN compiler at CUHK. For all numerical computations, an initial guess of $G(t)$ is fixed to 0.5. In our investigations we used the following values of the material properties and other parameters: porosity $\epsilon = 0.4$, the viscosity ratio $R_\nu = 1$, the Prandtl number $Pr = 1$, the Darcy number $Da = 10^{-2}$, the ratio of heat capacities $\sigma_r = 1$, the ratio of thermal conductivities $\sigma_m = 1$ and the Rayleigh number $Ra = 10^4$. Unless otherwise stated, a uniform mesh layout $129^2/65^2$ for the mixed FE method is used. The final time is $t_f = 1$ and in the numerical integration of the governing, sensitivity and adjoint systems we used the the time-step $\Delta t = \Delta\tau = 0.025$. The heat source and the sensor are located at $(0.75, -0.75)$ and $(0.43750, -0.6875)$, respectively, cf. Figure 1. We used a single measurement M = 1.

In studying the computational performance of the Algorithms, we will focus on the following quantities:

- K_e — the number of iterations required to reach a certain accuracy, given by e, in the reconstruction of the heat source function G;
- t_d — the cumulative CPU time of the direct solver;
- t_a — the cumulative CPU time of the adjoint solver;
- t_s — the cumulative CPU time of the sensitivity solver;
- $t_{d,J}$ and $t_{a,\nabla J}$ — the cumulative CPU times for the direct and adjoint solvers required to calculate the cost functional and the gradient expression using (44) and (45), or (46) and (47);
- t_{cg} — the cumulative CPU time of the CG or SCG steps which is not included in the times already mentioned above.

For each iteration in Algorithm 1 and Algorithm 2, the total CPU time is $t_d + t_a + t_s + t_{cg}$, whereas in Algorithm 3 and Algorithm 4, the total CPU time is $t_d + t_a + t_{d,J} + t_{a,\nabla J} + t_{cg}$. The relative error between the approximate solution obtained at the K^{th} iteration and exact solution (50) is defined as

$$e(G^K) = \frac{\| G^K - G^{\text{ex}} \|^2}{\| G^{\text{ex}} \|^2}.$$

5.1. *Numerical Performance of Four Algorithms*

For the case with noise–free measurements, Our computational results obtained for the four Algorithms are collected in Figure 2 which shows the reconstructions of the heat source functions obtained after a different number of iterations and Figure 3 which shows the history of the decrease of the objective function $J(G^K)$ with iterations. Additional data concerning the CPU times of the different elements of the four algorithms are presented in Table 1.

In regard to the data shown in Figures 2 - 3 and Table 1, we would like to make the following comments:

- All four algorithms converged to the same minimizer and were able to recover the unknown time-dependence of the heat source function; we add that the value of $G(t)$ at the terminal instant of time $t = t_f$ was also reconstructed correctly.
- The SCG Algorithms showed improved convergence in comparison to the CCG Algorithm requiring significantly fewer iterations to achieve a given level of accuracy of reconstruction; Algorithm 4 converged the fastest requiring about a third of the iterations needed by the CCG Algorithm.
- The faster convergence of the SCG Algorithms was partially offset by the larger computational cost of a single iteration for these approaches; however, the SCG Algorithms still performed better than the CCG Algorithm in terms of the total computational time, with Algorithm 4 exhibiting the best efficiency.

5.2. *Effect of Noise Level*

Finally, we study the effects of random errors in the measured temperature data on the quality of the reconstruction. To fix attention, we will focus exclusively on Algorithm 4, which in the previous section was found to have the best performance of all the Algorithms tested. Here Algorithm 4 is employed with a fixed iteration number as a stopping rule. The noisy data was generated from the solution of the INCP corresponding to the following heat source functions

$$G^{\text{ex}}(t) = -2\cos(2\pi t) + 6, \tag{51}$$

$$G^{\text{ex}}(t) = -1.5\cos(2\pi t) - 0.5\cos(4\pi t) - 0.5\cos(6\pi t) - 1.5\cos(8\pi t) + 6 \tag{52}$$

by adding adding noise with $\sigma = 0.01$, 0.03 and 0.05 according to formula (6), see Figure 4. In Figures 5 and 6, the robustness of reconstruction is assessed by comparing the heat source functions reconstructed from the noisy data against exact functions given in (51) and (52). In the numerical solution of the direct and adjoint problems we used the uniform mesh layout $65^2/33^2$ for the mixed FE method. The final time is $t_f = 1$ with the time-step sizes $\Delta t = \Delta\tau = 0.025$ and $\Delta t = \Delta\tau = 0.0167$ in the reconstructions corresponding to (51) and (52), respectively. Inspection of Figures 5(a), (c), (e) and 6(a), (c), (e) indicates that, as expected, increasing the measurement errors leads to a loss of accuracy in the estimation of the heat source function.

In order to mitigate the effect of the noise and perform further regularization of the reconstructions corresponding to the noisy data from Figure 4, we adopted the technique based on the Sobolev gradients [42]. The Sobolev gradients $\nabla J^S \in H_0^1[0,1]$ can be easily obtained from the usually employed L_2 gradients $\nabla J \in L^2[0,1]$ (see (19)) by solving the following elliptic boundary-value (in time) problem

with homogeneous Neumann boundary condition:

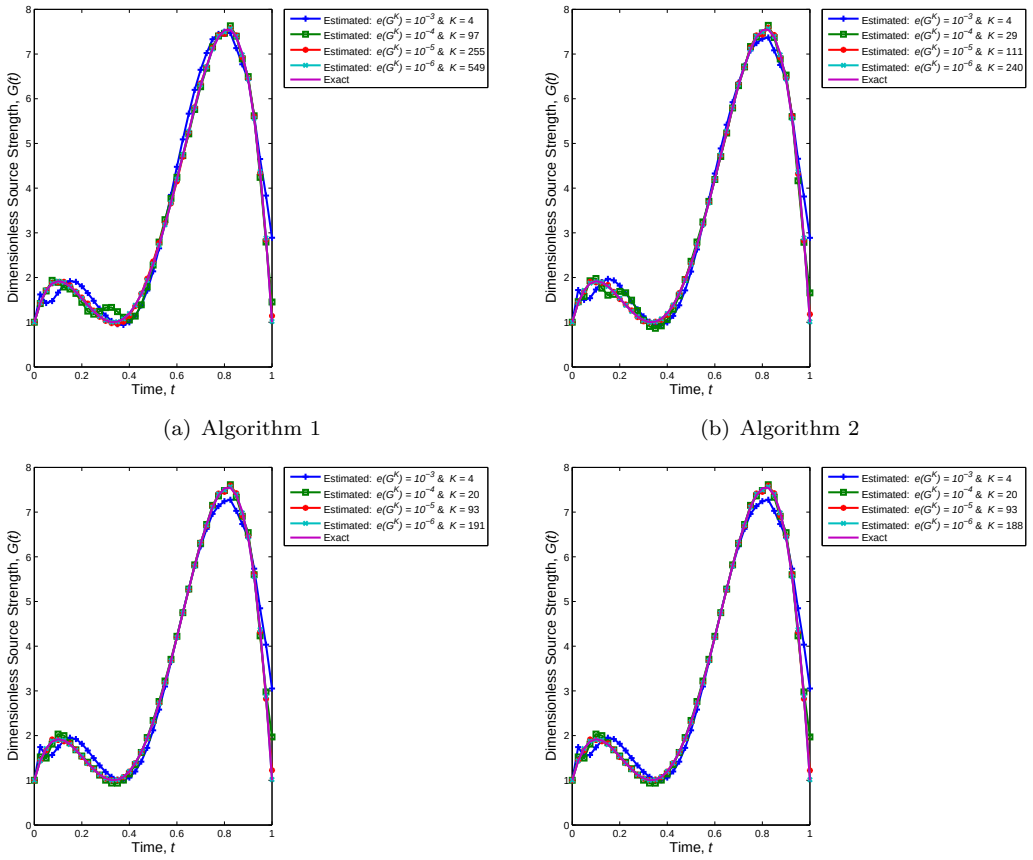
$$\begin{cases} \left(I - \lambda^2 \nabla^2\right) \nabla J^S = \nabla J, & \text{in } [0,1] \\ \boldsymbol{n} \cdot \nabla J^S = 0, & \text{on } \partial[0,1] \end{cases} \tag{53}$$

where $\boldsymbol{n}$ denotes the unit outward normal vector on $\partial[0,1]$ and $\lambda \in \mathbb{R}$ is a regularization parameter representing the cut-off length-scale below which the gradient information is essentially filtered out as result of solving (53) for ∇J^S, see [42]. For discretization of (53) a 1D piecewise linear finite element is used. By constructing the mass and stiffness matrices on the left-hand side of (53) and the lumped mass matrix on the right-hand side of (53), the weak form is solved by an iterative solver. In all our calculations reported here we set $\lambda = 0.1$. Effectiveness of the reconstructions based on the Sobolev gradients described above is demonstrated in Figures 5(b), (d), (f) and 6(b), (d), (f). This data should be compared with the results from Figures 5(a), (c), (e) and 6(a), (c), (e) which were obtained using the "classical" L_2 gradients, cf. (19), with the same number of iterations as the stopping rule. It should be emphasized that the approach based on the Sobolev gradients leads to estimates of the heat source function which are both significantly smoother and also more accurate, except for the endpoints of the interval $[0,1]$.

6. Concluding Remarks

In this investigation we considered the application of a number of state-of-the-art gradient-based minimization algorithms to the solution of a complex PDE-constrained optimization problem. This model problem arises in the context of the time-dependent inverse natural convection problem, and is quite typical for many inverse and optimization problems in fluid mechanics [43]. Determination of the gradient direction in such problems typically involves the solution of a suitably-defined adjoint system. An important question which often arises in practical applications concerns the efficient determination of the length of the step along the descent direction. This study was conducted to assess the trade-offs inherent in the different approaches to this problem, namely, techniques relying on a rather coarse approximation of the optimal step size which have a fairly small computational cost per iteration, and techniques employing a more accurate approximation of the optimal step size and therefore having a larger computational cost per iteration. We carefully examined, both in terms of the rate of convergence and the corresponding computational time, the performance of four state-of-the-art algorithms based on the conjugate gradients. It was found that, when compared to the classical CG method with the step size determined based on the solution of the sensitivity equations, the adaptive nonmonotone SCG algorithm exhibited more rapid convergence. This accelerated convergence was the consequence of a more accurate determination of the step size in the adaptive nonmonotone SCG algorithm, which also resulted in a higher computational cost per iteration of this algorithm. However, the benefit of the more rapid convergence exhibited by this algorithm was not negated by the higher per-iteration cost since it still achieved the best performance in terms of the total CPU time. While the improvement in performance was not very large, essentially it improved by a factor of two or so, the adaptive nonmonotone SCG algorithm could be recommended for the solution of problems such as INCP given the modest additional effort required for its implementation in comparison to the classical CG approaches. We also considered the solution of the reconstruction problem in the presence of noisy measurements and demonstrated

that the use of Sobolev gradients, which are quite straightforward to implement in the context of gradient-based optimization, can have a significant regularizing effect.



(a) Algorithm 1 (b) Algorithm 2

(c) Algorithm 3 (d) Algorithm 4

Figure 2. The reconstructions of the heat source obtained with the four Algorithms after different numbers of iterations.

Algorithm 1	$e(G^K)$	K_e	$J(G^K)$	CPU time in seconds
	10^{-3}	4	1.9079E-04	209.15 + 255.96 + 270.05 + 2.74 = 737.90
	10^{-4}	97	1.0451E-05	5.0363E3 + 6.7110E3 + 7.0088E3 + 66.75 = 1.8823E4
	10^{-5}	255	6.8141E-07	1.2905E4 + 1.7928E4 + 1.8687E4 + 175.26 = 4.9696E4
	10^{-6}	549	3.1323E-09	2.7866E4 + 3.8098E4 + 3.9765E4 + 374.23 = 1.0610E5
Algorithm 2	$e(G^K)$	K_e	$J(G^K)$	CPU time in seconds
	10^{-3}	4	1.1814E-04	212.10 + 298.19 + 274.83 + 2.90 = 788.02
	10^{-4}	29	7.1226E-06	1.4917E3 + 2.072E3 + 2.0586E3 + 19.90 = 5.6425E3
	10^{-5}	111	1.0259E-07	5.3000E3 + 7.3882E3 + 8.1656E3 + 73.33 = 2.0927E4
	10^{-6}	240	9.8179E-10	1.1439E4 + 1.5911E4 + 1.7658E4 + 158.68 = 4.5166E4
Algorithm 3	$e(G^K)$	K_e	$J(G^K)$	CPU time in seconds
	10^{-3}	4	3.4541E-04	215.35 + 257.89 + 218.08 + 258.66 + 3.79 = 953.77
	10^{-4}	20	2.7464E-06	1.0737E3 + 1.3094E3 + 1.0896E3 + 1.2928E3 + 19.10 = 4.7846E3
	10^{-5}	93	1.2573E-08	4.9601E3 + 6.1499E3 + 4.9594E3 + 6.0862E3 + 88.42 = 2.2244E4
	10^{-6}	191	1.0223E-09	1.0067E4 + 1.2706E4 + 1.0039E4 + 1.2559E4 + 181.52 = 4.5552E4
Algorithm 4	$e(G^K)$	K_e	$J(G^K)$	CPU time in seconds
	10^{-3}	4	1.8181E-04	179.49 + 251.55 + 179.84 + 253.18 + 3.56 = 867.62
	10^{-4}	20	2.5622E-06	1.0458E3 + 1.4285E3 + 1.0316E3 + 1.4223E3 + 19.78 = 4.9479E3
	10^{-5}	93	2.0508E-08	4.7762E3 + 6.5171E3 + 4.6655E3 + 6.4516E3 + 90.54 = 2.2501E4
	10^{-6}	188	8.5956E-10	9.5563E3 + 1.3079E4 + 9.3506E3 + 1.2949E4 + 180.76 = 4.5116E4*

Table 1. Comparison of the four Algorithms with respect to the number K_e of iterations required to reach a prescribed accuracy of reconstruction $e(G^K)$ and the corresponding computational time.

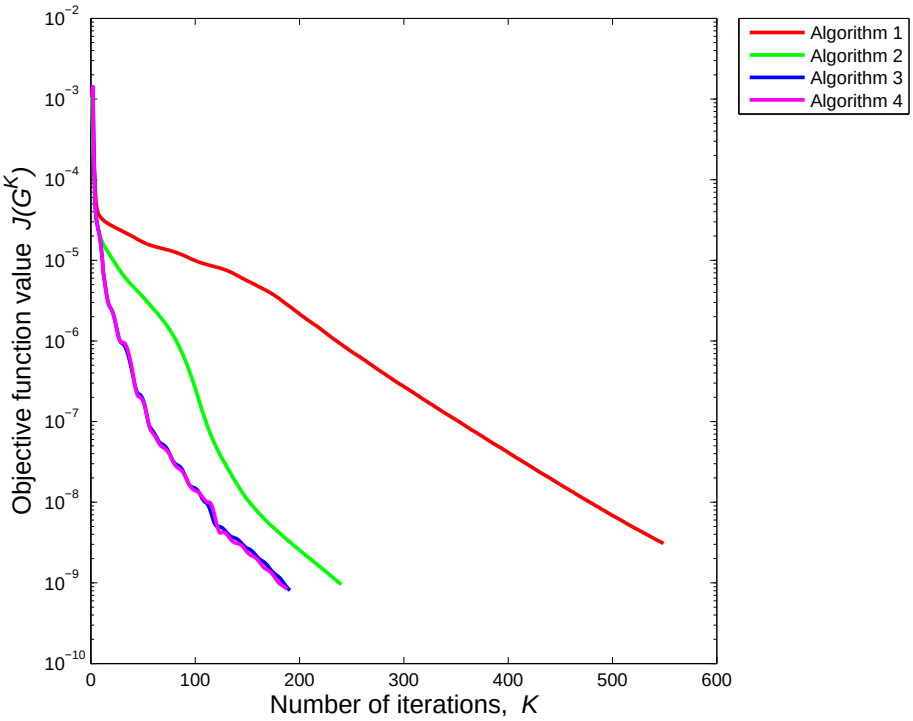
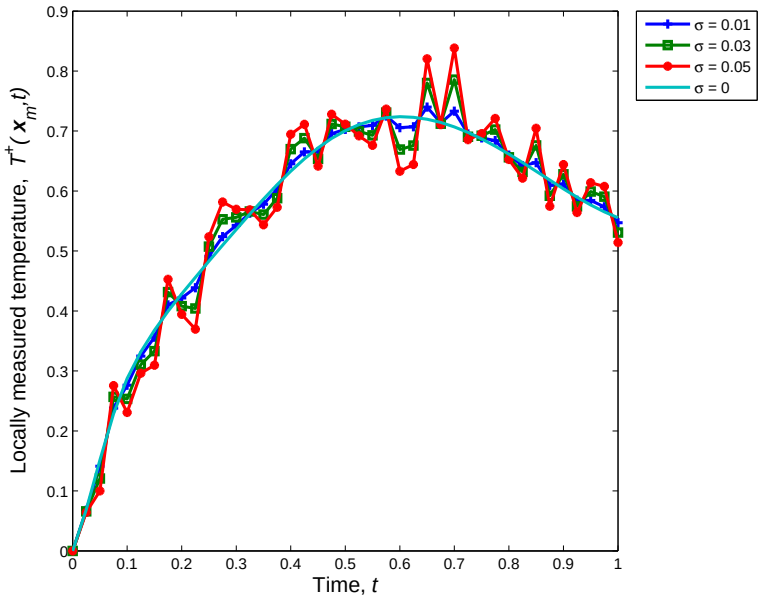
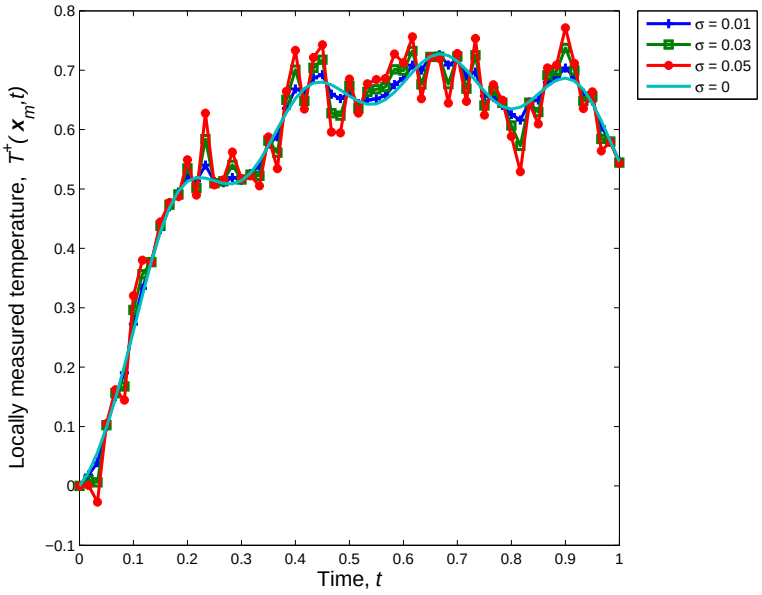


Figure 3. Decrease of objective function $J(G^K)$ as a function of iterations for different Algorithms.

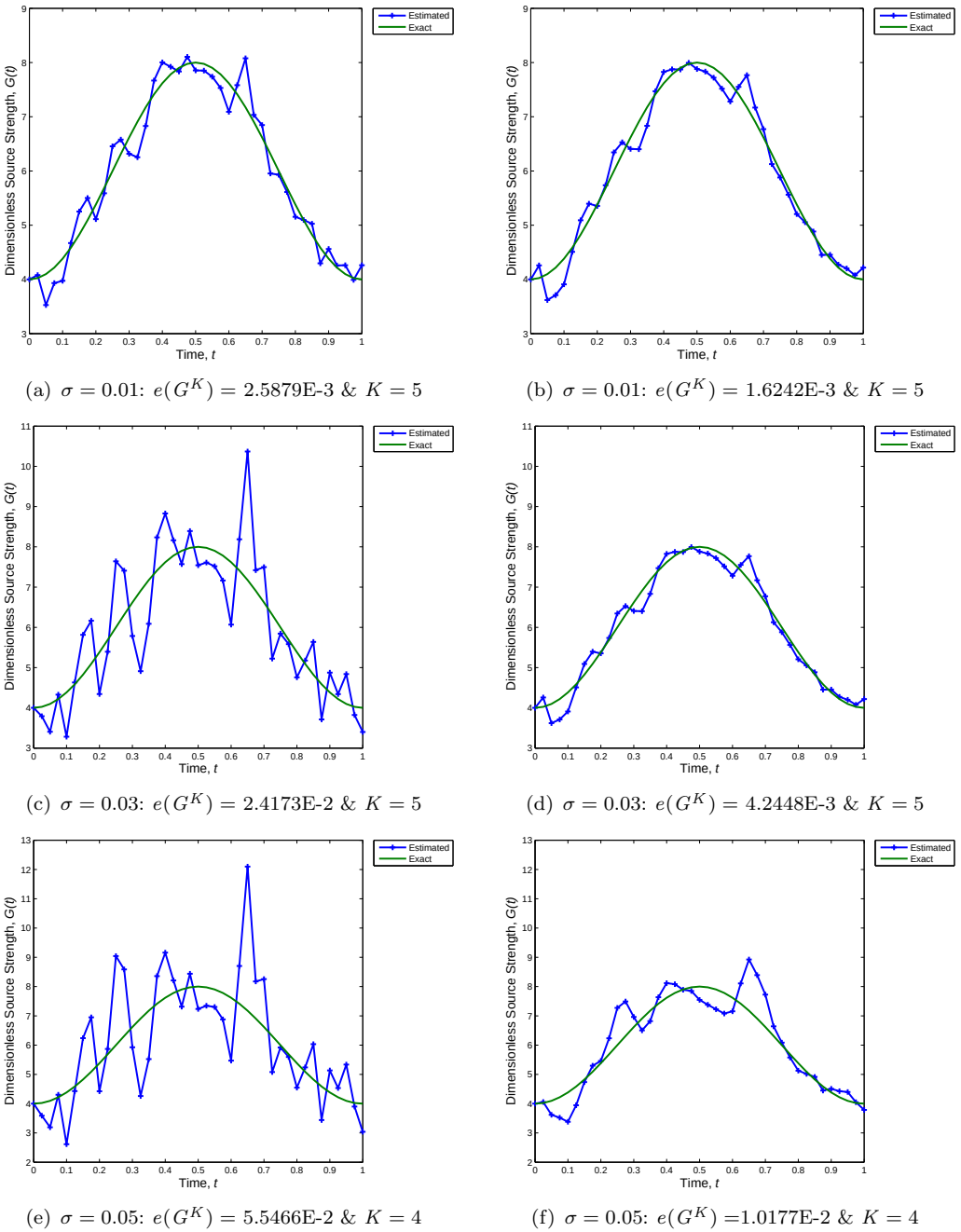


(a) Example (I)



(b) Example (II)

Figure 4. Local temperature data at the location $(0.4375, -0.6875)$ with noise against time.



(a) $\sigma = 0.01$: $e(G^K) = 2.5879\text{E-}3$ & $K = 5$

(b) $\sigma = 0.01$: $e(G^K) = 1.6242\text{E-}3$ & $K = 5$

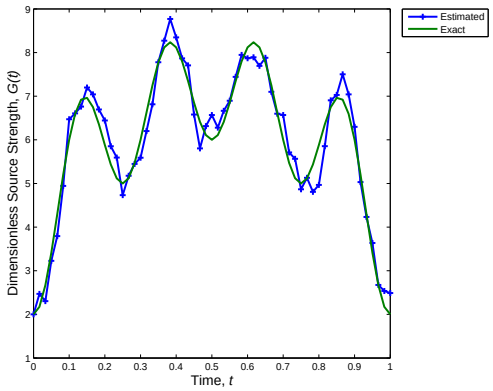
(c) $\sigma = 0.03$: $e(G^K) = 2.4173\text{E-}2$ & $K = 5$

(d) $\sigma = 0.03$: $e(G^K) = 4.2448\text{E-}3$ & $K = 5$

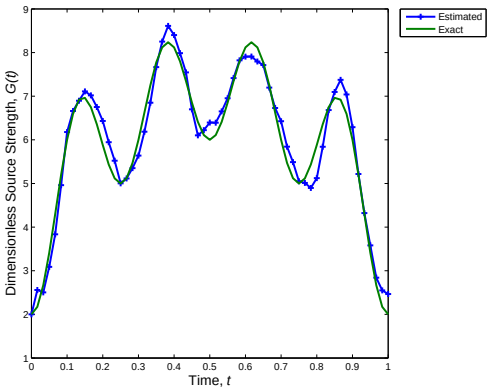
(e) $\sigma = 0.05$: $e(G^K) = 5.5466\text{E-}2$ & $K = 4$

(f) $\sigma = 0.05$: $e(G^K) = 1.0177\text{E-}2$ & $K = 4$

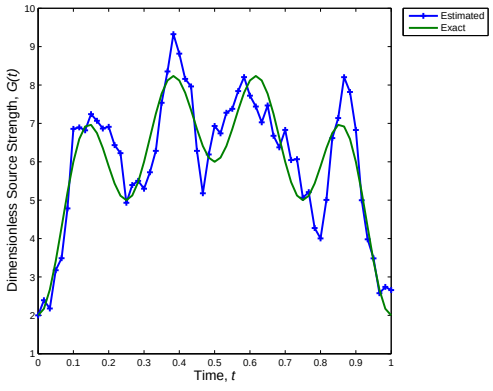
Figure 5. The estimation of the strength of a quadratic heat source varying with time as a sinusoidal function.



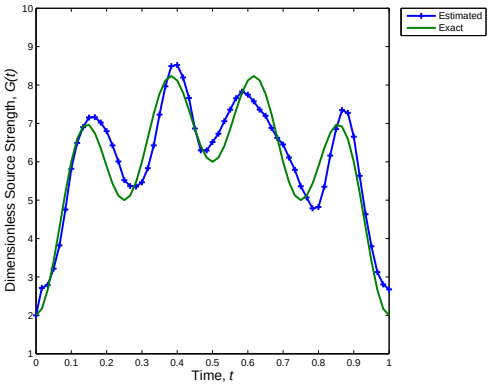
(a) $\sigma = 0.01$: $e(G^K) = 3.1654\text{E-}3$ & $K = 20$



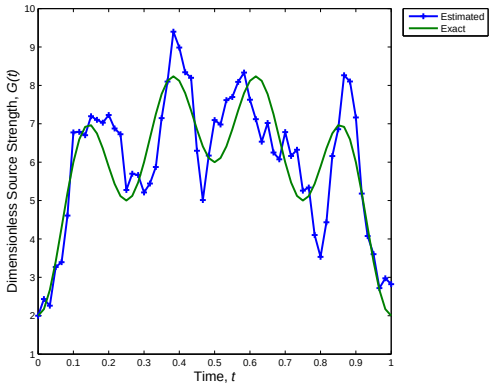
(b) $\sigma = 0.01$: $e(G^K) = 2.3953\text{E-}3$ & $K = 20$



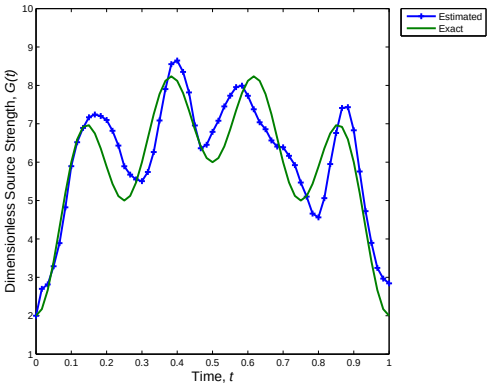
(c) $\sigma = 0.03$: $e(G^K) = 1.3235\text{E-}2$ & $K = 8$



(d) $\sigma = 0.03$: $e(G^K) = 7.1001\text{E-}3$ & $K = 8$



(e) $\sigma = 0.05$: $e(G^K) = 2.1971\text{E-}2$ & $K = 5$



(f) $\sigma = 0.05$: $e(G^K) = 1.2363\text{E-}2$ & $K = 5$

Figure 6. The estimation of the strength of a quadratic heat source varying with time as a symmetric sinusoidal curve-like function.

Appendix A. Discretization in Time using the Consistent Splitting Scheme

This Appendix summarizes the use of the CSS for the direct problem, the adjoint problem and the sensitivity problem.

- Direct Problem: A second-order decoupled approximation to the non-dimensional INCP is defined as follows: Let $\boldsymbol{u}^0 = \boldsymbol{u}^{-1} = \boldsymbol{u}(\boldsymbol{x},0)$, $p^0 = p^{-1} = p(\boldsymbol{x},0)$, and $T^0 = T^{-1} = T(\boldsymbol{x},0)$. Let $(\boldsymbol{u}^n, p^n, T^n)$ be the n^{th} time-step to $(\boldsymbol{u}(\boldsymbol{x}, n\Delta t)$, $p(\boldsymbol{x}, n\Delta t)$, $T(\boldsymbol{x}, n\Delta t))$. For $n = 1$, we use the first-order discretization in space, together with the backward Euler discretization in time to obtain $(\boldsymbol{u}^1, p^1, T^1)$. Then, for $n \geq 1$, find $\boldsymbol{u}^{n+1}$, p^{n+1} and T^{n+1} such that

$$\begin{cases} \dfrac{1}{\epsilon}\left(\dfrac{3\boldsymbol{u}^{n+1} - 4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t}\right) + \dfrac{1}{\epsilon^2}((2\boldsymbol{u}^n - \boldsymbol{u}^{n-1})\cdot\nabla)\boldsymbol{u}^{n+1} + \dfrac{1}{2\epsilon^2}(\nabla\cdot(2\boldsymbol{u}^n - \boldsymbol{u}^{n-1}))\boldsymbol{u}^{n+1} \\ \qquad -\dfrac{R_\nu Pr}{\epsilon}\nabla^2\boldsymbol{u}^{n+1} + \left(\dfrac{Pr}{Da} + \dfrac{F_\epsilon}{\sqrt{Da}}|2\boldsymbol{u}^n - \boldsymbol{u}^{n-1}|\right)\boldsymbol{u}^{n+1} \\ \qquad = -\nabla(2p^n - p^{n-1}) + RaPr(2T^n - T^{n-1})\boldsymbol{j}, \\ \boldsymbol{u}^{n+1}\big|_{\partial\Omega} = \boldsymbol{0}. \end{cases} \tag{A1}$$

$$(\nabla\phi^{n+1}, \nabla b) = \left(\frac{1}{\epsilon}\left(\frac{3\boldsymbol{u}^{n+1} - 4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t}\right), \nabla b\right), \quad \forall b \in H^1(\Omega). \tag{A2}$$

$$p^{n+1} = \phi^{n+1} + 2p^n - p^{n-1} - \frac{R_\nu Pr}{\epsilon}\nabla\cdot\boldsymbol{u}^{n+1}. \tag{A3}$$

$$\begin{cases} \sigma_r\left(\dfrac{3T^{n+1} - 4T^n + T^{n-1}}{2\Delta t}\right) + \boldsymbol{u}^{n+1}\cdot\nabla T^{n+1} - \alpha_m\nabla^2 T^{n+1} \\ \qquad = G(t^{n+1})\delta_k(x - x^\dagger)\delta_k(y - y^\dagger), \\ T^{n+1}\big|_{\partial\Omega_1} = 0, \qquad \text{and} \qquad \left.\dfrac{\partial T^{n+1}}{\partial x}\right|_{\partial\Omega_2} = 0. \end{cases} \tag{A4}$$

- Adjoint Problem: Likewise, for $l \geq 1$, find $\boldsymbol{\xi}^{l+1}$, q^{l+1} and η^{l+1} such that

$$\begin{cases} \dfrac{1}{\epsilon}\left(\dfrac{3\boldsymbol{\xi}^{l+1} - 4\boldsymbol{\xi}^l + \boldsymbol{\xi}^{l-1}}{2\Delta\tau}\right) - \dfrac{1}{\epsilon^2}(\boldsymbol{u}^{l+1}\cdot\nabla)\boldsymbol{\xi}^{l+1} + \dfrac{1}{2\epsilon^2}(\nabla\cdot\boldsymbol{u}^{l+1})\boldsymbol{\xi}^{l+1} - \dfrac{R_\nu Pr}{\epsilon}\nabla^2\boldsymbol{\xi}^{l+1} + \dfrac{Pr}{Da}\boldsymbol{\xi}^{l+1} \\ \quad = -\nabla(2q^l - q^{l-1}) - \dfrac{1}{\epsilon^2}(2\boldsymbol{\xi}^l - \boldsymbol{\xi}^{l-1})\cdot\left(\nabla\boldsymbol{u}^{l+1}\right)^{\mathrm{T}} + \dfrac{1}{2\epsilon^2}\nabla\left(\boldsymbol{u}^{l+1}\cdot(2\boldsymbol{\xi}^l - \boldsymbol{\xi}^{l-1})\right) \\ \qquad -(2\eta^l - \eta^{l-1})\nabla T^{l+1} - \dfrac{F_\epsilon}{\sqrt{Da}}\dfrac{\left(|\boldsymbol{u}^{l+1}|^2\right)(2\boldsymbol{\xi}^l - \boldsymbol{\xi}^{l-1}) + \left((2\boldsymbol{\xi}^l - \boldsymbol{\xi}^{l-1})\cdot\boldsymbol{u}^{l+1}\right)\boldsymbol{u}^{l+1}}{|\boldsymbol{u}^{l+1}|}, \\ \boldsymbol{\xi}^{l+1}\big|_{\partial\Omega} = \boldsymbol{0}. \end{cases} \tag{A5}$$

$$(\nabla\psi^{l+1}, \nabla b) = \left(\frac{1}{\epsilon}\left(\frac{3\boldsymbol{\xi}^{l+1} - 4\boldsymbol{\xi}^l + \boldsymbol{\xi}^{l-1}}{2\Delta\tau}\right), \nabla b\right), \quad \forall b \in H^1(\Omega). \tag{A6}$$

$$q^{l+1} = \psi^{l+1} + 2q^l - q^{l-1} - \frac{R_\nu Pr}{\epsilon}\nabla\cdot\boldsymbol{\xi}^{l+1}. \tag{A7}$$

$$\begin{cases} \sigma_r\left(\dfrac{3\eta^{l+1} - 4\eta^l + \eta^{l-1}}{2\Delta\tau}\right) - \boldsymbol{u}^{l+1}\cdot\nabla\eta^{l+1} - \alpha_m\nabla^2\eta^{l+1} \\ \qquad = RaPr\,(\xi^y)^{l+1} + \sum_{m=1}^{\mathrm{M}}\left[T^{l+1} - \left(T^\dagger\right)^{l+1}\right]\delta(x-x_m)\delta(y-y_m), \\ \eta^{l+1}\big|_{\partial\Omega_1} = 0, \qquad \text{and} \qquad \left.\dfrac{\partial\eta^{l+1}}{\partial x}\right|_{\partial\Omega_2} = 0. \end{cases} \tag{A8}$$

- Sensitivity Problem: Likewise, for $n \geq 1$, find $\delta\boldsymbol{u}^{n+1}$, δp^{n+1} and δT^{n+1} such that

$$\begin{cases} \dfrac{1}{\epsilon}\left(\dfrac{3\delta\boldsymbol{u}^{n+1} - 4\delta\boldsymbol{u}^n + \delta\boldsymbol{u}^{n-1}}{2\Delta t}\right) + \dfrac{1}{\epsilon^2}(\boldsymbol{u}^{n+1}\cdot\nabla)\delta\boldsymbol{u}^{n+1} + \dfrac{1}{2\epsilon^2}(\nabla\cdot\boldsymbol{u}^{n+1})\delta\boldsymbol{u}^{n+1} \\ \qquad -\dfrac{R_\nu Pr}{\epsilon}\nabla^2\delta\boldsymbol{u}^{n+1} + \dfrac{Pr}{Da}\delta\boldsymbol{u}^{n+1} + \dfrac{F_\epsilon}{\sqrt{Da}}|\boldsymbol{u}^{n+1}|\delta\boldsymbol{u}^{n+1} \\ = -\nabla(2\delta p^n - \delta p^{n-1}) + RaPr(2\delta T^n - \delta T^{n-1})\boldsymbol{j} \\ \qquad -\dfrac{1}{\epsilon^2}((2\delta\boldsymbol{u}^n - \delta\boldsymbol{u}^{n-1})\cdot\nabla)\boldsymbol{u}^{n+1} - \dfrac{1}{2\epsilon^2}(\nabla\cdot(2\delta\boldsymbol{u}^n - \delta\boldsymbol{u}^{n-1}))\boldsymbol{u}^{n+1} \\ \qquad -\dfrac{F_\epsilon}{\sqrt{Da}}\left(\dfrac{\boldsymbol{u}^{n+1}\cdot\left(2\delta\boldsymbol{u}^n - \delta\boldsymbol{u}^{n-1}\right)}{|\boldsymbol{u}|^{n+1}}\boldsymbol{u}^{n+1}\right), \\ \delta\boldsymbol{u}^{n+1}\big|_{\partial\Omega} = \boldsymbol{0}. \end{cases} \tag{A9}$$

$$(\nabla\delta\phi^{n+1}, \nabla b) = \left(\frac{1}{\epsilon}\left(\frac{3\delta\boldsymbol{u}^{n+1} - 4\delta\boldsymbol{u}^n + \delta\boldsymbol{u}^{n-1}}{2\Delta t}\right), \nabla b\right), \quad \forall b \in H^1(\Omega). \tag{A10}$$

$$\delta p^{n+1} = \delta\phi^{n+1} + 2\delta p^n - \delta p^{n-1} - \frac{R_\nu Pr}{\epsilon}\nabla\cdot\delta\boldsymbol{u}^{n+1}. \tag{A11}$$

$$\begin{cases} \sigma_r\left(\dfrac{3\delta T^{n+1} - 4\delta T^n + \delta T^{n-1}}{2\Delta t}\right) + \boldsymbol{u}^{n+1}\cdot\nabla\delta T^{n+1} - \alpha_m\nabla^2\delta T^{n+1} \\ \qquad = -\delta\boldsymbol{u}^{n+1}\cdot\nabla T^{n+1} + d(t^{n+1})\delta_k(x-x^\dagger)\delta_k(y-y^\dagger), \\ \delta T^{n+1}\big|_{\partial\Omega_1} = 0, \qquad \text{and} \qquad \left.\dfrac{\partial\delta T^{n+1}}{\partial x}\right|_{\partial\Omega_2} = 0. \end{cases} \tag{A12}$$

The above systems of nonlinear partial differential equations are discretized by the backward difference formula (BDF) in time. The time integrator used in the momentum equations (see (A1), (A5), and (A9)) is fully implicit for the viscous term and semi-implicit for the nonlinear advection term. Specifically, to avoid any restriction on the time step (in other words, to ensure unconditional stability), the advection term $\boldsymbol{u}\cdot\nabla\boldsymbol{u}$ has been replaced by its skew-symmetric counterpart $(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \frac{1}{2}(\nabla\cdot\boldsymbol{u})\boldsymbol{u}$ (e.g., [22]). In the adjoint and the sensitivity problems, ignoring the factor $\dfrac{1}{\epsilon^2}$, the stabilisation terms become $\frac{1}{2}\left[(\nabla\cdot\boldsymbol{u})\boldsymbol{\xi} - \nabla\left(\boldsymbol{u}\cdot\boldsymbol{\xi}\right)\right]$ and $\frac{1}{2}\left[(\nabla\cdot\boldsymbol{u})\delta\boldsymbol{u} + (\nabla\cdot\delta\boldsymbol{u})\boldsymbol{u}\right]$. Of particular interest in (A2) is that the auxiliary pressure ϕ^{n+1}, treated as a correction term, is unknown and will be determined at each

time-step. The pressure p^{n+1} in (A3) is not only updated by the previous linear extrapolation terms $2p^n - p^{n-1}$ but also reduced down by the boundary layer effect along the bounded surface because of the term $\frac{R_\nu Pr}{\epsilon}\nabla \cdot \boldsymbol{u}^{n+1}$. The correction terms ψ^{n+1} in (A7) and $\delta\phi^{n+1}$ in (A11), accordingly, can be interpreted similar to the direct problem. For an in-depth discussion of the numerical implementation, the reader should refer to [40, 41].

Appendix B. Discretization in Space Using Finite Elements

In order to solve the optimality systems of (A1) - (A4), (A5) - (A8), and (A9) - (A12), we use the mixed FE method. Let $\mathcal{T}_h$ be a regular FE mesh consisting of triangular FE mesh. We define X_h, the approximation space for the velocity, as the set of continuous functions that are piecewise quadratic on each triangular mesh of $\mathcal{T}_h$. The approximation space N_h for the auxiliary pressure and pressure consists of continuous functions that are piecewise linear on each triangular mesh. To ensure that the auxiliary pressure and pressure are uniquely defined, we require it to have a mean value of zero. The basic principle guiding the choice of the mixed FE basis functions is that they should satisfy the *inf-sup* compatibility condition and therefore avoid node-to-node pressure oscillations . The element pair used for the present work is a Hood-Taylor type (see e.g., [44]); $\{u_h, v_h, T_h\}$, $\{\xi_h^x, \xi_h^y, \eta_h\}$, and $\{\delta u_h, \delta v_h, \delta T_h\}$, are interpolated in a quadratic fashion, while $\{p_h, \phi_h\}$, $\{q_h, \psi_h\}$, and $\{\delta p_h, \delta\phi_h\}$ are interpolated in a linear manner, as depicted in Figure B1.

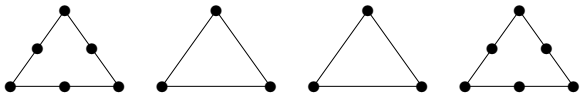


Figure B1. $\mathbb{P}_2$ for $\{u_h, v_h, T_h\}$, $\{\xi_h^x, \xi_h^y, \eta_h\}$, and $\{\delta u_h, \delta v_h, \delta T_h\}$; $\mathbb{P}_1$ for $\{p_h, \phi_h\}$, $\{q_h, \psi_h\}$, and $\{\delta p_h, \delta\phi_h\}$.

References

[1] Beck, JV, Blackwell, B, St-Clair, Jr., CR. Inverse Heat Conduction: Ill-posed Problems. Wiley-Interscience, New York, 1985.

[2] Alifanov, OM. Inverse Heat Transfer Problems. Springer-Verlag, Berlin Heidelberg, 1994.

[3] Ozisik, MN, Orlande, HRB. Inverse Heat Transfer: Fundamentals and Applications. Taylor and Francis, New York, 2000.

[4] Park, HM, Chung, OY. On the solution of an inverse natural convection problem using various conjugate gradient methods. *International Journal for Numerical Methods in Engineering* 2000; **47**: 821 - 842.

[5] Prud'homme, M, Hung Nguyen, T. Whole time-domain approach to the inverse natural convection problem. *Numerical Heat Transfer, Part A: Applications* 1997; **32**: 169 - 186.

[6] Zabaras, N, Yang, GZ. A functional optimization formulation and implementation of an inverse natural convection problem. *Computer Methods in Applied Mechanics and Engineering* 1997; **144**, 245 - 274.

[7] Lewis, RW, Nithiarasu, P, Seetharamu, KN, Fundamentals of the finite element method for heat and fluid flow. John Wiley & Sons: NJ, 2004.

[8] Jasmin, S, Prudhomme, M. Inverse determination of a heat source from a solute concentration generation model in porous medium. *International Communications in Heat and Mass Transfer* 2005; **32**: 43 - 53.

[9] Prudhomme, M, Jasmin, S. Inverse solution for a biochemical heat source in a porous medium in the presence of natural convection. *Chemical Engineering Science* 2006; **61**: 1667 - 1675.

[10] Wong, JCF, Xie, JL. Inverse determination of a heat source from a natural convection in a porous cavity. Submitted to Computers & Fluids.

[11] Alekseev, AK, Navon, IM, Steward, JL. Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems. *Optimization Methods and Software* 2009; **24**: 63 - 87.

[12] Dennis JE Jr, Schnabel, RB. Numerical Methods for Unconstrained Optimization and Non-Linear Equations. Prentice-Hall, Inc, 1983.

[13] Hager, WW, Zhang, H. A new conjugate-gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization* 2005; **16**(1): 170 - 192.
[14] Moller, MF. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks* 1993; **6**: 525 - 533.
[15] Plagianakos, VP, Magoulas, GD, Vrahatis, MN. Deterministic nonmonotone strategies for effective training of multilayer perceptrons. *IEEE Transactions on Neural Networks* 2002; **13**: 1268 - 1284.
[16] Zeev, N. Olga Savasta, O, Cores, D. Non-monotone spectral projected gradient method applied to full waveform inversion. *Geophysical Prospecting* 2006; **54**: 525 - 534.
[17] Bello, L, Raydan, M. Convex constrained optimization for the seismic reflection tomography problem. *Journal of Applied Geophysics* 2007; **62**: 158 - 166.
[18] Livieris, IE, Pintelas, P. A survey on algorithms for training artificial neural networks. Technical Report TR08-01, Department of Mathematics, University of Patras, 2008.
[19] Kostopoulos, AE, Grapsa, TN, Self-scaled conjugate gradient training algorithms. *Neurocomputing* 2009; **72**: 3000 - 3019.
[20] Andrei, N. Scaled conjugate gradient algorithms for unconstrained optimization. *Computational Optimization and Applications* 2007; **38**: 401 - 416.
[21] Andrei, N. A scaled nonlinear conjugate gradient algorithm for unconstrained optimization. *Optimization,* 2008; **57**: 549 - 570.
[22] Guermond, JL, Shen, J. A new class of truly consistent splitting schemes for incompressible flows. *Journal of Computational Physics* 2003; **192**: 262 - 276.
[23] Jarny, Y, Ozisik, MN, Bardon, JP. A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *International Journal of Heat and Mass Transfer* 1991; **34**: 2911 - 2919.
[24] Loulou, T, Scott, EP. An inverse heat conduction problem with heat flux measurements. *International Journal for Numerical Methods in Engineering* 2006; **67**: 1587 - 1616.
[25] Alifanov, OM, Artyukhin, EE, Rumyantsev, SV. Extreme Methods of Solving Ill-posed Problems with Applications to Inverse Heat Transfer Problems. Begell House: New York, 1995.
[26] Engl, HW, Hanke, M, Neubauer, A. Regularization of Inverse Problems. Kluwer: Amsterdam, 1996
[27] Tikhonov, AN, Arsenin, VY. Solution of Ill-posed Problems. Winston and Sons: Washington, DC, 1977.
[28] Birgin, E, Martínez, JM, A spectral conjugate gradient method for unconstrained optimization. *Applied Mathematics and Optimization*, 2001; **43**: 117 - 128.
[29] Perry, JM, A class of conjugate gradient algorithms with a two step variable metric memory, Discussion Paper 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1977.
[30] Shanno, DF, Conjugate gradient methods with inexact searches.*Mathematics of Operations Research*, 1978; : 244 - 256.
[31] Shanno, DF, On the convergence of a new conjugate gradient algorithm, *SIAM Journal on Numerical Analysis*, 1978; **15**: 1247 - 1257.
[32] Barzilai, J, Borwein, JM, Two point step size gradient method. *IMA Journal of Numerical Analysis*, 1988; **8**: 141 - 148.
[33] Raydan, M, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, SIAM Journal on Optimization, 1997; **7**(1): 26 - 33.
[34] Fletcher, R, On the Barzilai-Borwein method, in Optimization and Control Applications, L. Qi, K. Teo, and X. Yang, eds., Series: Applied Optimization, Vol. 96, Springer, Berlin, pp. 235 - 256, 2005.
[35] Shanno, DF, Phua, KH, Algorithm 500. Minimization of unconstrained multivariate functions [E4]. *ACM Transactions on Mathematical Software* 1976; **2**; 87 - 94.
[36] Grippo, L. A class of unconstrained minimization methods for neural network training. *Optimization Methods and Software* 1994; **4**: 135 - 150.
[37] Han, J, Sun, J, Sun W. Global convergence of nonmonotone descent methods for unconstrained optimization problems. *Journal of Computational and Applied Mathematics* 2002; **146**: 89 - 98.
[38] Cauchy, A. Methode generale pour la resolution des systemes d'equations simultanees. *Comp. Rend. Acad. Sci. Paris*, 1847; 536 - 538.
[39] Armijo, L. Minimization of functions having Lipschitz continuous partial derivatives. *Pacific Journal of Mathematics*, 1966; **16**: 1 - 3.
[40] Wong, JCF. Numerical simulation of two-dimensional laminar mixed-convection in a lid-driven cavity using the mixed finite element consistent splitting scheme. *International Journal of Numerical Methods for Heat and Fluid Flows* 2007; **17**: 46 - 93.
[41] Wong, JCF, Yuan P. Mixed finite element formulation of algorithms for double-diffusive convection in a fluid-saturated porous medium. Communications in Numerical Methods in Engineering, 2007; 23(11): 983 - 1022.
[42] Protas, B, Adjoint-based optimization of PDE systems with alternative gradients. *Journal of Computational Physics* 2008; **227** 6490 - 6510.
[43] Gunzburger, MD, Perspectives in flow control and optimization. SIAM, Philadelphia, 2003.
[44] Hood, P, Taylor, C. Navier-Stokes equations using mixed-interpolation. In: Oden, JT, Zienkiewicz, OC, R.H. Gallagher, RH, Taylor, C (ed.) Finite Element Methods in Flow Problems, 57 - 66, UAH Press, Huntsville, 1974.