A note on the iterative solution of weakly nonlinear elliptic control problems with control and state constraints

Emanuele Galligani

Dipartimento di Matematica Pur e e Applicata “G. Vitali”
Università degli Studi di Modena e Reggio Emilia
Via Campi 213/b, 41100, Modena, Italy
galligani@unimo.it

Technical Report n. 79, October 2007
Dipartimento di Matematica, Università di Modena e Reggio Emilia

Abstract. This work concerns with the solution of optimal control problems by means of nonlinear programming methods. The control problem is transcribed into a finite dimensional nonlinear programming problem by finite difference approximation. An iterative procedure for the solution of this nonlinear program is presented. An extensive numerical analysis of the behaviour of the method is reported on boundary control and distributed control problems with boundary conditions of Dirichlet or Neumann or mixed type.

1 Reaction-diffusion problems

When the physical system under consideration involves a reaction process accompanied by diffusion, the physical principle of conservation leads to a set of partial differential equations for the unknown quantities of the system. These quantities may be mass concentration in chemical reaction processes, temperature in heat conduction, neutron flux in nuclear reactors, population density in population dynamics, and many others. In certain problems of physical and engineering sciences, a combination of these quantities is involved in a set of equations.

The reaction–diffusion equation with convection has the form

$$\frac{\partial \phi}{\partial t} - \text{div} (\sigma \text{grad} \phi) + \mathbf{v} \cdot \text{grad} \phi + \alpha \phi + g(x, t, \phi) = 0$$

(1)

where $$\phi = \phi(x, t)$$ is the density function at time $$t$$ and position $$x$$ in a diffusion medium $$R$$, $$\sigma = \sigma(x, t)$$ is the diffusion coefficient, $$\alpha = \alpha(x, t)$$ is the absorption term, $$\mathbf{v}$$ and is the velocity vector and $$-g(x, t, \phi(x, t))$$ is the rate of change due to a reaction.

When the physical system involves two or more density functions $$\phi_1, \phi_2, ..., \phi_r$$, the reaction–diffusion processes is described by a set of equations of the form

$$\frac{\partial \phi_i}{\partial t} - \text{div} (\sigma_i \text{grad} \phi_i) + \mathbf{v}_i \cdot \text{grad} \phi_i + \alpha_i \phi_i + g_i(x, t, \phi_1, \phi_2, ..., \phi_r) = 0$$

$$i = 1, ..., r$$

(2)

and $$\sigma_i(x, t), \alpha_i(x, t)$$ and $$\mathbf{v}_i$$ are the terms referred to $$\phi_i(x, t)$$.

*This research was supported by the Italian Ministry of Education and Research (MIUR) project: PRIN–2006 n. 2006018748.
This work was presented at 22nd Biennial Conference on Numerical Analysis "NA07", Dundee, June 26–29, 2007.
The multigroup diffusion approximation to the Boltzmann’s transport equation leads to the coupled system (2) with

\[ g_i(x, t, \varphi_1, ..., \varphi_r) = \sum_{j=1}^{r} a_{ij}(x, t) \varphi_j(x, t) - s_i(x, t, \varphi_1, ..., \varphi_r) \]  

(3)

where \( a_{ii} \geq 0 \) and \( a_{ij} \leq 0 \) for \( i \neq j \) are piecewise continuous functions on \( R \) and \( s_i \) are source terms \((i = 1, ..., r)\).

When the reaction diffusion process reaches a steady state, the density function \( \varphi(x, t) \) and the coefficients are independent of time \( t \). This implies that

\[ \frac{\partial \varphi}{\partial t} = 0 \]

and therefore the governing equation with the effect of convection becomes

\[ -\text{div}(\sigma \text{grad}\varphi) + \nu \cdot \text{grad}\varphi + \alpha \varphi + g(x, \varphi) = 0 \]  

(4)

In the case of \( r \) density functions, the corresponding equations become \((i = 1, ..., r)\)

\[ -\text{div}(\sigma_i \text{grad}\varphi_i) + \nu_i \cdot \text{grad}\varphi_i + \alpha_i \varphi_i + g_i(x, \varphi_1, \varphi_2, ..., \varphi_r) = 0 \]  

(5)

In a variety of interesting applications to Ecology and Nuclear Physics linear and nonlinear eigenvalue problems arise.

For example, in the computation of the steady spatial distribution of a population of diffusing consumers (predators) in a special resource–consumer (prey–predator) ecological system we must consider a diffusion model of the type \(([10])\)

\[ -\text{div}(\sigma \text{grad}\varphi) + \alpha \varphi + g(x, \varphi) = \frac{1}{\lambda} \theta \varphi \quad \lambda > 0 \]  

(6)

with \( \sigma = \sigma(x) > 0, \theta = \theta(x) > 0, \alpha = \alpha(x) \geq 0 \) piecewise continuous functions on \( R \); \( g(\varphi) = \gamma(\varphi) \varphi \) and the trophic function \( \gamma(\varphi) \) is a continuously differentiable positive and monotone nondecreasing function of \( \varphi \) for all \( \varphi \geq 0 \). The constant \( \lambda \) (eigenvalue) is positive.

On the boundary surface \( \Gamma \) of \( R \) we have the homogeneous condition \( \varphi(x) = 0, \; x \in \Gamma \).

When the diffusion medium \( R \) is a bounded domain the reaction–diffusion–convection equations are supplemented by suitable boundary conditions on the boundary surface \( \Gamma \). The appropriate condition on the boundary depends on the physical mechanism surrounding the diffusion medium. Besides the three basic types of boundary conditions (Dirichlet, or Neumann, or Robin boundary conditions), in certain cases it is necessary to impose a nonlinear boundary condition.

For time–dependent problems there is also an initial condition at \( \text{time } t = 0 \). Typical expressions of \( g(x, t, \varphi) \) are

1. Enzyme–substrate reaction model \(([29], [40])\):

\[ g(\varphi) = \frac{a \varphi}{1 + b \varphi} \quad a, b > 0 \]  

(7)

We remark that,

\[ g'(\varphi) = \frac{a}{(1 + b \varphi)^2}; \quad g''(\varphi) = -\frac{2b}{(1 + b \varphi)^3} \]

Here, \( g > 0 \) and \( g' > 0 \) for \( \varphi > 0 \).

2. Chemical reaction model \(([1])\):

\[ g(\varphi) = -a(c - \varphi)e^{(-b/(1+\varphi))} \quad a, b, c > 0 \]  

(8)

(e.g. \( 1 \leq a \leq 5.0 \cdot 10^{-4}, \; b = 15, \; c = 0.4 \)).
We remark that, 

\[ g'(\varphi) = \frac{ae^{-\frac{b}{1+\varphi}}(\varphi^2 + (b+2)\varphi + (1-bc))}{(1+\varphi)^2} \]
\[ g''(\varphi) = \frac{a}{(1+\varphi)^4}e^{-\frac{b}{1+\varphi}}(b^2(\varphi - c) + 2b(1+c)(1+\varphi)) \]

Here, \( g > 0, g' > 0 \) and \( g'' > 0 \) for \( \varphi > c \).

3. Ginzburg–Landau oscillating BZ reaction model ([30], [31], [12])

\[ g(\varphi) = -a\varphi(1 - \varphi^2) \quad a > 0 \quad (9) \]

4. Spatially distributed communities model ([10]):

\[ g(\varphi) = \frac{a\varphi^2}{b+\varphi} \quad a > 0, b > 0 \quad (10) \]

or

\[ g(\varphi) = a\varphi \log(1 + \varphi) \quad a > 0 \quad (11) \]

(e.g. \( \sigma = 1.5, \alpha = 0.7, v = 0 \) and \( R = [0,100] \times [0,100] \), with \( a = 2.0 \cdot 10^{-2}, b = 3.0 \) for (10) and \( a = 5.0 \cdot 10^{-3} \) for (11) for the equation (4) subject to Dirichlet boundary conditions and \( \sigma = 1.5, \alpha = 0.085, \theta = 0.12, \lambda = 1.25 \) and \( R = [0,100] \times [0,100] \), with \( a = 0.02, b = 3.0 \) for (10) and \( a = 0.005 \) for (11) for the eigenvalue problem (6) with homogeneous Dirichlet boundary conditions).

We remark that, for (10) we have

\[ g'(\varphi) = \frac{a\varphi^2 + 2ab\varphi}{(b + \varphi)^2}, \quad g''(\varphi) = \frac{2ab^2}{(b + \varphi)^3} \]

and for (11) we have

\[ g'(\varphi) = a\log(1 + \varphi) + \frac{a\varphi}{1 + \varphi}; \quad g''(\varphi) = \frac{a(\varphi + 2)}{(1 + \varphi)^2} \]

then, in both the cases, \( g > 0, g' > 0 \) and \( g'' > 0 \) for \( \varphi \geq 0 \).

5. Fischer’s population growth model ([40], [45]):

\[ g(\varphi) = -a\varphi(b - c\varphi) \quad a, b, c > 0 \quad (12) \]

or

\[ g(\varphi) = -a\varphi(\varphi - \theta)(1 - \varphi) \quad a > 0, \quad 0 < \theta < 1 \quad (13) \]

We remark that, for (13)

\[ g'(\varphi) = -a(\varphi - \theta)(1 - \varphi) - a\varphi((1 + \theta) - 2\varphi) \]

Here, \( g \geq 0 \) and \( g' \geq 0 \) for \( \varphi \geq 1 \).

6. Budworm population dynamics model ([40]):

\[ g(\varphi) = -\frac{\varphi^2}{1 + \varphi} + r\varphi(1 - \frac{\varphi}{q}) \quad r, q > 0, \quad (14) \]
7. Radiation model:

\[ g(\varphi) = \beta e^{\alpha \varphi} \quad \beta \in \mathbb{R}, \alpha > 0 \]  

In thermal ignition and combustion problems we have \( \beta < 0 \) and \( \alpha > 0 \), and then, in that case,

\[ g'(\varphi) = \beta \alpha e^{\alpha \varphi} < 0 \]

See [45] and [46] for an analysis of the existence of the solution of thermal ignition and combustion problems described with

\[ \frac{\partial \varphi}{\partial t} - \Delta \varphi + g(\varphi) = 0 \]

In Bratu eigenvalue problem (e.g. [9], [39]) we have \( g(\varphi) = \lambda e^{\varphi} \) in

\[ -\Delta \varphi + \alpha \frac{\partial \varphi}{\partial x} + g(\varphi) = 0 \]

The solution exists for \( \lambda \geq 0 \) [28] and in the unit square domain the branch of solutions has a limit point at \( \lambda = \lambda_c = -6.80812... \) and \( u = u_c \) with \( u_c(0.5, 0.5) = 1.39166... \).

8. Molecular interaction model ([47]):

\[ g(\varphi) = \varphi^2 \]  

9. A nonlinear oscillator ([49])

\[ g(\varphi) = -\lambda \sin \varphi \]  

in an eigenvalue problem

\[ -\Delta \varphi + g(\varphi) = 0 \]

In [2] a nonlinearity in several variables is

\[ g_i(\varphi) = - (|\varphi_i| + \sin \varphi_i \cos \varphi_{i-1}) \quad i = 1, ..., r \]  

(with \( \varphi_0 = 0 \).)

Another interesting case of this type is the complex Bratu problem ([39]). Taking \( \varphi = \varphi_1 + i\varphi_2 \), the complex Bratu problem is equivalent to the following system

\[
\begin{align*}
-\Delta \varphi_1 &= \lambda e^{\varphi_1} \cos \varphi_2 \\
-\Delta \varphi_2 &= \lambda e^{\varphi_1} \sin \varphi_2
\end{align*}
\]

In [27], the authors show that the critical point \( \{\lambda_c, u_c\} \) of the real valued Bratu problem is a bifurcation point for the complex value problem if we allow complex valued solutions.

There is a huge amount of literature concerning the numerical solution of these initial and boundary value problems. When \( R \) is a two–three dimensional domain, these problems belong to the class of very large scientific computing problems for which the use of computers with vector–parallel architecture is appropriate.

Much of the work which has been done on the use of parallel and vector computers for solving with the finite difference or finite element discretization methods the partial differential equations (1)–(6) has treated the theme of decomposition of a problem into independent portions and reordering of the unknowns of the discretized equations in order to enhance such a decomposition.

Special attention have received the Multigrid and Domain–Decomposition methods (see e.g. [26, Chap. 12] and [50]).

Also the Operator Splitting and Alternating Direction methods have been extensively considered ([35]). In the large class of the Operator Splitting methods we have developed and analyzed the Additive Operator Splitting (AOS) methods. These methods are characterized by having a high degree of decomposition. Generally, this property increases the degree of multiprogramming (DOM), i.e., the number of active processes of a multiprocessor system. (DOM is a measure of the performance of a multiprocessor system). This makes the Additive Operator Splitting methods ideally suited for implementation on parallel computers.
We discretize, with respect to the spatial variable, the partial differential equations (1)–(6) with the box–integration method ([52, Chap. 6]). This provides to apply the theory of monotone operators for the study of stability of the approximate solution.

The large sets of nonlinear algebraic equations generated by the discretization of the equations (1)–(6) are solved by using the Newton’s method. This method is one of the most used and powerful tool for solving nonlinear equations in several variables. Newton’s method is attractive for its local convergence properties even if it may fail to converge because not properly initialized: initialization may be therefore a complicated question when one applies this method to difference schemes generated by the discretization of partial differential equations, especially of elliptic type as (4) or (6).

In neighbourhoods of exact solution Newton’s method generates quadratically convergent approximations. To maintain a superlinear convergence and to enlarge the attraction basin of the exact solution, one may use various Newton–like methods.

In order to solve efficiently and accurately large systems of nonlinear difference equations generated by the discretization with the Finite Difference method of the equation (4) or (6), we have developed and analyzed a Modified Newton–Iterative method and a Simplified Newton–Iterative method (a two–stage iterative method) which are very well suited for implementation on vector–parallel computers ([21], [24], [19], [20]).

In the recent past there has been given, also, notable consideration to distributed and boundary optimal control problems for the time–dependent and steady–state reaction–diffusion equations. Important control problems for the stationary equation (4), subject to control and state inequality constraints, with a quadratic cost functional have been investigated in [36], [37], [38], [30], [31], [32], [51], [11].

By introducing a suitable finite–difference discretization scheme developed in [38], [36], [37], these elliptic control problems are transcribed into a large finite–dimensional nonlinear constrained optimization problem with a sparse structure (see [23, sections 2,3,5]).

In particular, for the nonlinear elliptic equation with \(g(\varphi)\) given by 3., 5. and 7. elliptic boundary control problems with Neumann boundary condition and elliptic distributed control problems with Neumann and Dirichlet boundary conditions with control and state constraints of simple type have been considered; even bang–bang controls have been included.

2 Newton–like approximation approach

The transcription of control problems governed by equation (4) leads to a nonlinear mathematical programming problem of the form

\[
\text{minimize} \quad f(z) \quad (18)
\]

subject to

\[
\begin{align*}
    g_1(z) &= 0 \\
    g_2(z) &\geq 0
\end{align*} \quad (19)
\]

where \(z \in \mathbb{R}^n\), \(f : \mathbb{R}^n \rightarrow \mathbb{R}\), \(g_1 \in \mathbb{R}^{neq}\) and \(g_2 \in \mathbb{R}^m\) (\(neq < n\)).

We assume that \(f(z), g_1(z), g_2(z)\) are twice continuously differentiable and the first and second derivatives of the objective function and constraints are available.

The Karush–Kuhn–Tucker (KKT) optimality conditions for problem (18)–(19), written in slack variable form, are

\[
\begin{align*}
    E_1 &= \nabla f(z) - \nabla g_1(z)\lambda_1 - \nabla g_2(z)\lambda_2 = 0 \\
    E_2 &= -g_1(z) = 0 \\
    E_3 &= -g_2(z) + s = 0 \\
    E_4 &= \Lambda_2 Se_m = 0
\end{align*} \quad (20)
\]

with \(s \geq 0; \quad \lambda_2 \geq 0\).
where \( s, \lambda_2 \in \mathbb{R}^m, \Lambda_2 = \text{diag}(\lambda_2), S = \text{diag}(s). \)

The vector \( e_m \) indicates the vector of \( m \) components whose values are equal to 1.

The system (20) can be written as

\[
\begin{align*}
H(v) &= 0 \\
\begin{array}{c}
s \\
\lambda_2
\end{array} &\geq 0; \quad \lambda_2 \geq 0
\end{align*}
\]

where

\[
v = \begin{pmatrix} z \\ \lambda_1 \\ \lambda_2 \\ s \end{pmatrix}, \quad H(v) = \begin{pmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{pmatrix} H_1(v)
\]

The Jacobian matrix of \( H \) is the matrix of order \( n + neq + 2m \)

\[
H'(v) = \begin{pmatrix} Q & B & C & 0 \\ B^T & 0 & 0 & 0 \\ C^T & 0 & 0 & I \\ 0 & 0 & S & \Lambda_2 \end{pmatrix}
\]

where \( Q \) is the Hessian matrix of the Lagrangian function of the problem (20),

\[
Q \equiv L(z) = \nabla^2 f(z) - \sum_{i=1}^{neq} \lambda_{1,i} \nabla^2 g_{1,i}(z) - \sum_{i=1}^{m} \lambda_{2,i} \nabla^2 g_{2,i}(z)
\]

and \( B = -\nabla g_1(z) \) and \( C = -\nabla g_2(z) \).

Here, \( \nabla^2 f(z), \nabla^2 g_{1,i}(z), \nabla^2 g_{2,i}(z) \) are the Hessian matrices of the function \( f(z) \) and of the \( i \)-th component of the constraints \( g_1(z) \) and \( g_2(z) \), respectively; then, \( \lambda_{1,i} \) and \( \lambda_{2,i} \) are the \( i \)-th component of \( \lambda_1 \) and \( \lambda_2 \), respectively.

Under the classic assumptions on problem (18)–(19) of existence, smoothness, regularity, second–order sufficiency, strict complementarity (see i.e. [34, Chap. 10]), the Jacobian matrix \( H'(z^*, \lambda_1^*, \lambda_2^*, s^*) \) of \( H(v) \) in (21) is nonsingular: here \( (z^*, \lambda_1^*, \lambda_2^*) \) is the solution to problem (18)–(19) and \( s^* = g_2(z^*) \).

This result motivates the use of the Newton method for solving the nonlinear system (21) with an initial guess \( v^{(0)} \), satisfying \( \lambda_{1,i}^{(0)} > 0, s^{(0)} > 0 \) in a neighbourhood of \( v^* = (z^*, \lambda_1^*, \lambda_2^*, s^*) \).

In order to avoid that the iterates of Newton’s method, when they reach the boundary of the nonnegative orthant \( (s, \lambda_2) \geq 0 \), are forced to stick to the boundary, we replace the complementarity equation \( E_4 = 0 \) in (21) with a perturbed complementarity equation \( \Lambda_2 S e_m = \rho e_m, \rho > 0 \). Thus, at each stage \( k \) \( (k = 0, 1, 2, \ldots) \) of Newton’s method we must solve the perturbed Newton equation

\[
H'(v^{(k)}) \Delta v + H(v^{(k)}) = \rho_k \tilde{e}
\]

where \( \tilde{e} = (0_{n+neq+m}, e_m^T)^T \) and the parameter \( \rho_k \) goes to 0 when \( k \) diverges.

Then, we must update the current iterate by

\[
v^{(k+1)} = v^{(k)} + \alpha_k \Delta v^{(k)}
\]

where \( \Delta v^{(k)} \) is the solution of the linear system (22).

This method determines a sequence \( \{v^{(k)}\} \) that strictly satisfies the constraints in the perturbed KKT system

\[
\begin{align*}
H(v) &= \rho_k \tilde{e} \\
\begin{array}{c}
s \\
\lambda_2
\end{array} &\geq 0; \quad \lambda_2 > 0
\end{align*}
\]

and the KKT optimality conditions (21) only in the limit.

Since the iterates \( v^{(k)} \) reach the solution from the interior of the nonnegative orthant (feasible region), such a method is also referred as Interior Point method [53].

The crucial points in the analysis of this iterative procedure are the choices of the parameters \( \rho_k \) and \( \alpha_k \) and the solution of the linear system (22).
• We define $\rho_k = \sigma_k \mu_k$, where $0 < \sigma_{\min} \leq \sigma_k \leq \sigma_{\max} < 1$ and $\mu_k$ satisfies the condition

$$\|\mu_k \tilde{e}\| \leq \|H(v^{(k)})\|$$

(\| \cdot \| denotes the 2–norm). Thus, equation (22) may be interpreted as the $k$–th iteration of the Inexact Newton method for solving the KKT system (21) with the forcing term $\sigma_k \in (0, 1)$ [13]. Therefore, the vector $\sigma_k \mu_k \tilde{e}$ has the meaning of a residual.

In the Inexact Newton method with forcing term $\sigma_k$ we have

$$\|H'(v^{(k)}) \Delta v^{(k)} + H(v^{(k)})\| \leq \sigma_k \|H(v^{(k)})\|$$

(26)

We can prove ([14]) that for values of $\mu_k$ that satisfy

$$\mu_k^{(1)} = \frac{s^{(k)} T \lambda_2^{(k)}}{m} \leq \mu_k \leq \frac{\|H(v^{(k)})\|}{\sqrt{m}} \equiv \mu_k^{(2)}$$

(27)

the vector $\Delta v^{(k)}$ that satisfies (26) is a descent direction at $v^{(k)}$ for the merit function $\Phi(v) = \|H(v)\|^2$, i.e. $\nabla \Phi(v^{(k)})^T \Delta v^{(k)} \leq 0$.

Computation of the exact solution $\Delta v^{(k)}$ of the equation (22) can be or may not be justified when $v^{(k)}$ is relatively far from the solution of (22). Therefore, one might prefer to compute some approximate solution of (22) by performing some iterations of an inner linear iterative solver for the equation (22) with an adaptive stopping criterion of the form

$$\|r^{(k)}\| \leq \delta_k \|H(v^{(k)})\|$$

(28)

where

$$r^{(k)} = H'(v^{(k)}) \Delta v^{(k)} + H(v^{(k)}) - \sigma_k \mu_k \tilde{e}$$

(29)

We indicate the approximate solution of (22) again with $\Delta v^{(k)}$.

The choice $\delta_k = 0$ implies that the equation (22) must be solved exactly.

Solving only approximately the equation (22) causes a reduction of the forcing term in the Inexact Newton method.

• The damping parameter $\sigma_k$ in (23) must be selected in such a way to guarantee that the merit function $\Phi(v)$ can be reduced at each iteration and the components of the vectors $s^{(k)}$ and $\lambda_2^{(k)}$ remain positive for all $k = 0, 1, 2, \ldots$.

In order to generate a sequence of iterates $v^{(k)}$ with $(s^{(k)}, \lambda_2^{(k)}) > 0$ that converges to a solution $v^*$ of the system (21) under standard assumptions on KKT systems, it is necessary to define a path to be followed by the iterates $v^{(k)}$ which forces these from coming too close to the boundary of the nonnegative orthant $(s, \lambda_2) \geq 0$.

For the generation of this trajectory, we use the following two functions of $\alpha$ defined in [17] (see also [42, p. 402]), $k = 0, 1, 2, \ldots$:

$$\varphi^{(k)}(\alpha) \equiv \min_{i=1,m} \left( S^{(k)}(\alpha) \lambda_2^{(k)}(\alpha) e_m \right) - \gamma_k \tau_1 \left( \frac{s^{(k)}(\alpha) T \lambda_2^{(k)}(\alpha)}{m} \right)$$

$$\psi^{(k)}(\alpha) \equiv s^{(k)}(\alpha)^T \lambda_2^{(k)}(\alpha) - \gamma_k \tau_2 \|H_1(v^{(k)}(\alpha))\|$$

where

$$\tau_1 \leq \frac{\min_{i=1,m} \left( S^{(0)}(\alpha) \lambda_2^{(0)}(\alpha) e_m \right)}{\left( \frac{s^{(0)} T \lambda_2^{(0)}}{m} \right)}; \quad \tau_2 \leq \frac{s^{(0)} T \lambda_2^{(0)}}{\|H_1(v^{(0)})\|}; \quad \gamma_k \in \left[ \frac{1}{2}, 1\right]$$
and

\[ v^{(k)}(\alpha) = \begin{pmatrix} z^{(k)}(\alpha) \\ \lambda_1^{(k)}(\alpha) \\ \lambda_2^{(k)}(\alpha) \\ s^{(k)}(\alpha) \end{pmatrix} = \begin{pmatrix} z^{(k)} \\ \lambda_1^{(k)} \\ \lambda_2^{(k)} \\ s^{(k)} \end{pmatrix} + \alpha \begin{pmatrix} \Delta z^{(k)} \\ \Delta \lambda_1^{(k)} \\ \Delta \lambda_2^{(k)} \\ \Delta s^{(k)} \end{pmatrix} = v^{(k)} + \alpha \Delta v^{(k)} \]

with \(s^{(0)} > 0\) and \(\lambda_2^{(0)} > 0\).

Thus \(\tau_1 > 0\) and \(\tau_2 > 0\).

Clearly \(v^{(k)} = v^{(k)}(0)\) and \(v^{(k+1)} = v^{(k)}(\alpha_k)\).

The vector \(\Delta v^{(k)}\) is the approximate solution of (22) with residual (28)–(29).

We can prove [22] that there exist two positive numbers \(\alpha_k^{(1)}\) and \(\alpha_k^{(2)}\) in \([0,1]\) such that

\[ \varphi^{(k)}(\alpha) \geq 0 \quad \forall \alpha \in (0, \alpha_k^{(1)}) \]

\[ \psi^{(k)}(\alpha) \geq 0 \quad \forall \alpha \in (0, \alpha_k^{(2)}) \]

when \(\varphi^{(k)}(0) \geq 0\) and \(\psi^{(k)}(0) \geq 0\).

If we define

\[ \tilde{\alpha}_k = \min\{\alpha_k^{(1)}, \alpha_k^{(2)}, 1\} \in (0,1] \]

and

\[ \tilde{\eta}_k = 1 - \tilde{\alpha}_k (1 - (\sigma_k + \delta_k)) \quad \text{with} \quad (\sigma_k + \delta_k) \leq \sigma_{max} + \delta_{max} < 1 \]

we have \(\tilde{\eta}_k < 1\) and

\[ \|H'(v^{(k)}) \tilde{\alpha}_k \Delta v^{(k)} + H(v^{(k)})\| \leq \tilde{\eta}_k \|H(v^{(k)})\| \tag{30} \]

That is, the vector \(\tilde{\alpha}_k \Delta v^{(k)}\) satisfies the condition on the residual of the Inexact Newton method with forcing term \(\tilde{\eta}_k\).

We select the step length \(\alpha_k\) in (23) by performing a reduction of \(\tilde{\alpha}_k\) by using the Inexact Newton Backtracking algorithm described in [16] until an acceptable \(\alpha_k = \theta \tilde{\alpha}_k\) is found, where \(t\) is the smallest nonnegative integer such that \(\alpha_k\) satisfies

\[ \|H(v^{(k)} + \alpha_k \Delta v^{(k)})\| \leq (1 - \beta \alpha_k (1 - (\sigma_k + \delta_k))) \|H(v^{(k)})\| \tag{31} \]

with \(\theta, \beta \in (0,1)\).

Since \((1 - \beta \alpha_k (1 - (\sigma_k + \delta_k))) < 1\), this inequality asserts that

\[ \|H(v^{(k+1)})\| \leq \xi_k \|H(v^{(k)})\| \quad 0 < \xi_k \leq \bar{\xi} < 1 \tag{32} \]

Inequalities (30) and (32) are the two basic conditions for the convergence of Inexact Newton method [48, Sect. 6.4].

Besides, if \(\|H(v^{(k)})\| \neq 0\), for all \(k = 0, 1, 2, \ldots\), we can prove [22] that the inner products \(s^{(k)^T} \lambda_1^{(k)}\) are bounded above and bounded away from zero and all components of \(S^{(k)} \lambda_2^{(k)} e_m\) are bounded above and bounded away from zero. Specifically, \((s^{(k)}, \lambda_2^{(k)}) > 0\) for all \(k = 0, 1, 2, \ldots\); the sequence \(\{\Phi(v^{(k)})\}\) is monotone and nonincreasing; therefore, \(\Phi(v^{(k)}) \leq \Phi(v^{(0)})\) for all \(k = 1, 2, \ldots\).

In [22] and [5] the convergence properties of the method (22)–(23) are stated under standard assumptions on the Karush–Kuhn–Tucker systems considered in [17] and [15].

- By omitting the iteration index \(k\), the linear perturbed Newton equation (22) can be written as

\[
\begin{align*}
Q \Delta z + B \Delta \lambda_1 + C \Delta \lambda_2 &= -E_1 \\
B^T \Delta z &= -E_2 \\
C^T \Delta z + \Delta s &= -E_3 \\
S \Delta \lambda_2 + \lambda_2 \Delta s &= -E_4 + \rho e_m
\end{align*}
\]
From the last equation we can deduce

$\Delta s = \Lambda^{-1}_2 [-S \Delta \lambda_2 - E_4 + \rho e_m]$  

and then the system (22) can be rewritten in the reduced form

$$
\begin{pmatrix}
Q & B & C \\
B^T & 0 & 0 \\
C^T & 0 & -\Lambda^{-1}_2 S
\end{pmatrix}
\begin{pmatrix}
\Delta z \\
\Delta \lambda_1 \\
\Delta \lambda_2
\end{pmatrix} =
\begin{pmatrix}
-E_1 \\
-E_2 \\
g_2(x) - \rho \Lambda^{-1}_2 e_m
\end{pmatrix}  \tag{33}
$$

The coefficient matrix of this system is symmetric and indefinite.

System (22) is not necessarily an ill-conditioned system of equations. In actual computations more care must be taken during the reduction of the system (22) to avoid cancellation in the reduction process (33) due to very small elements in $\Lambda^{-1}_2 S$.

By a further substitution, from the third block equation in (33), we obtain

$\Delta \lambda_2 = S^{-1}_2 [\Lambda_2 C^T \Delta z + \Lambda_2 E_3 - E_4 + \rho e_m]$  

Thus, the system (22) can be written in the condensed form

$$
\begin{pmatrix}
A & B \\
B^T & 0
\end{pmatrix}
\begin{pmatrix}
\Delta z \\
\Delta \lambda_1
\end{pmatrix} =
\begin{pmatrix}
c \\
-E_2
\end{pmatrix}  \tag{34}
$$

with

$A = Q + CS^{-1}_2 \Lambda_2 C^T$
$c = -E_1 - CS^{-1}_2 [\Lambda_2 E_3 - E_4 + \rho e_m]$  

In actual computations, as the variables $s_i$ and $\lambda_{2,i}$ approach the bound, the elements of $S^{-1}_2 \Lambda_2$ may become very large. In this case it is necessary to introduce a preconditioning strategy for improving greatly the performance of the method.

It is well known that, if the symmetric matrix $Q$ is positive definite on the null space of $B^T$ and $B^T$ is a full row–rank matrix, the condensed system (34) may be solved with the Hestenes method of multipliers ([34, p. 406], [25]). The standard assumptions on the Karusk–Kuhn–Tucker systems considered in [17] and [15] assure that this property on $Q$ and $B^T$ holds.

However, numerical experience on optimal control of physical systems described by elliptic reaction–diffusion equations with box constraints of simple–type on the control and the state ([6], [8]), has shown that it is more efficient to solve system (34) with the preconditioned conjugate gradient method with an indefinite preconditioner ([33]) together with a sparse Choleski–like factorization on the preconditioner matrix.

The Fortran 90 package that implements the sparse Choleski–like factorization of sparse matrices, called BLKFCLT and introduced in [7], is downloadable from the website dm.unife.it/blkfelt.

## 3 Numerical examples

### 3.1 Elliptic control problems

In the following we state the elliptic control problems with control and state constraints and we consider $R \subset \mathbb{R}^2$ a bounded domain with piecewise smooth boundary $\Gamma$; the point $x = (x_1, x_2)^T$ belongs to $\bar{R} = R \cup \Gamma$.

We consider the elliptic boundary control problem with Neumann boundary condition (EBC) and elliptic distributed control problems with Neumann and Dirichlet boundary conditions (EDC–I, EDC–II, EDC–III).
In these problems, $\partial_{\nu} y(x)$ denotes the differentiation along the normal $\nu$ to $\Gamma$ directed away from $R$.

An optimal solution of each of these problems is denoted by $u^*$ and $y^*$. We assume that an optimal solution $z^* = (y^*, u^*)^T$ exists.
3.2 Description of the numerical experiments

The Newton–like approximation approach described in Section 2 and in the tables below referred as INM–PCG (Inexact Newton Method–Preconditioned Conjugate Gradient) for the solution of the optimization problem (18)–(19), has been implemented in a Fortran 90 code on a workstation HPzx6000 with Intel Itanium2 processor 1.3GHz with 2Gb of RAM. The code has been compiled with a +03 optimization option of the Fortran HP compiler.

The figures 1–8 show the computed state and the control variables with different values of the parameters for the control problems with the number $N$ of grid points for each axis $x_1, x_2$ equal to 99. In all the problems the domain $\bar{R} = R \cup \Gamma$ is the unit square in the Cartesian $x_1 x_2$ plane.

We listed below the values of the parameters:

- **EBC** (figures 1 and 2): $u_l(x) = 1.8, u_r(x) = 2.5, y_r(x) = 2.7, y_d(x) = 2 - 2(x_1(x_1 - 1) + x_2(x_2 - 1)), u_d(x) = 0; \gamma = 0.01$ (for Figure 1); $\gamma = 0$ (for Figure 2);

- **EDC–I** (figures 3 and 4): $u_l(x) = 1.5, u_r(x) = 4.5, y_r(x) = 0.185, y_d(x) = 1 + 2(x_1(x_1 - 1) + x_2(x_2 - 1)), u_d(x) = 0; \gamma = 0.001$ (for Figure 3); $\gamma = 0$ (for Figure 4);

- **EDC–II** (figures 5 and 6): $y_d(x) = \sin(2\pi x_1) \sin(2\pi x_2), u_d(x) = 0$;
  - for Dirichlet condition (Figure 5): $u_l(x) = -5, u_r(x) = 5, y_r(x) = 0.11, \gamma = 0.001$;
  - for Neumann condition (Figure 6): $u_l(x) = -8, u_r(x) = 9, y_r(x) = 0.371, \gamma = 0$;

- **EDC–III** (figures 7 and 8): $u_l(x) = 1.7, u_r(x) = 2, y_r(x) = 7.1, a(x) = 7 + 4 \sin(2\pi x_1 x_2), b = 1$;
  - $M = 1$ and $K = 0.8$ (for Figure 7); $M = 0$ and $K = 1$ (for Figure 8);
Figure 2: problem EBC; $\gamma = 0$

Figure 3: problem EDC–I; $\gamma = 0.001$
Figures 9 and 10 show the structures of the Jacobian matrix of the equality constraints and the Hessian matrix of the Lagrangian function when $N = 5$. Furthermore, in the tables 1–4 (tables 9–12 of [6]), we report a comparison of INM–PCG method with KNITRO codes version 4.0.2. Both direct ([3], [4]) and iterative ([41]) inner solvers for KNITRO are considered.

Here, the term $1e - 6$ indicates $10^{-6}$; opttol is a tolerance parameter used by the KNITRO codes and the INM–PCG method stops when the Euclidean norm of $H(v^{(k)})$ is less than $10^{-8}$; the choose of the values for the starting points is described in [6].

In the experiments, INM–PCG method uses the same input AMPL models ([18]) of the discretized PDE problems as KNITRO.

References

Figure 6: problem EDC–II, Neumann condition; $\gamma = 0$

Figure 7: problem EDC–III; $M = 1, K = 0.8$

Figure 8: problem EDC–III; $M = 0, K = 1$
structure of \((\nabla g_1(z))^T\)

structure of \(Q\)

Figure 9: problem EBC

structure of \((\nabla g_1(z))^T\)

structure of \(Q\)

Figure 10: problem EDC–III
<table>
<thead>
<tr>
<th>$N$</th>
<th>Solver</th>
<th>opttol</th>
<th>$f(x^{(it)})$</th>
<th>$|H(u^{(it)})|$</th>
<th>time</th>
<th>it (inn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>INM–PCG</td>
<td>0.2641625459</td>
<td>3.5e-9</td>
<td>5.9</td>
<td>28(24)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>0.2642862472</td>
<td>4.5e-7</td>
<td>5.5</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.264163747</td>
<td>4.7e-9</td>
<td>7.0</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2641625452</td>
<td>7.2e-11</td>
<td>7.7</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>0.2643159559</td>
<td>5.5e-7</td>
<td>7.0</td>
<td>11(38)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2641637501</td>
<td>5.0e-9</td>
<td>13.0</td>
<td>15(173)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2641625452</td>
<td>5.2e-10</td>
<td>16.3</td>
<td>17(247)</td>
</tr>
<tr>
<td>199</td>
<td>INM–PCG</td>
<td>0.2672834461</td>
<td>9.0e-9</td>
<td>40.0</td>
<td>35(34)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>0.2676809839</td>
<td>5.6e-7</td>
<td>37.2</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2672858418</td>
<td>5.5e-9</td>
<td>49.5</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2672839122</td>
<td>9.2e-10</td>
<td>54.4</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>0.2676583552</td>
<td>8.0e-7</td>
<td>57.8</td>
<td>14(85)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2672838028</td>
<td>4.8e-9</td>
<td>100.5</td>
<td>21(231)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2672835172</td>
<td>3.7e-10</td>
<td>155.2</td>
<td>23(536)</td>
</tr>
<tr>
<td>299</td>
<td>INM–PCG</td>
<td>0.2683261906</td>
<td>6.0e-10</td>
<td>125.9</td>
<td>37(30)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>0.2683490951</td>
<td>7.6e-7</td>
<td>174.6</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2683296025</td>
<td>9.0e-9</td>
<td>191.7</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2683262257</td>
<td>2.3e-10</td>
<td>217.3</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>0.2694845168</td>
<td>4.0e-7</td>
<td>258.2</td>
<td>19(110)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2683272338</td>
<td>6.3e-9</td>
<td>361.6</td>
<td>25(217)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2683263348</td>
<td>5.0e-10</td>
<td>494.2</td>
<td>27(488)</td>
</tr>
<tr>
<td>399</td>
<td>INM–PCG</td>
<td>0.26884799937</td>
<td>4.3e-9</td>
<td>352.3</td>
<td>39(34)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>0.268471969</td>
<td>3.1e-7</td>
<td>401.9</td>
<td>11(70)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>0.2688521643</td>
<td>9.6e-9</td>
<td>523.9</td>
<td>14(127)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>0.2688480413</td>
<td>3.6e-10</td>
<td>916.4</td>
<td>17(509)</td>
</tr>
</tbody>
</table>

Table 1: Comparison of Inexact Newton Method + Preconditioned Conjugate Gradient with KNITRO v4.0.2 on test problem EBC ($\gamma = 0.01$)
| $N$ | Solver  | opttol | $f(x^{(it)})$ | $||H(v^{(it)})||$ | time | it (inn) |
|-----|---------|--------|---------------|----------------|------|--------|
| 99  | INM–PCG | 1e-6   | 6.216167657e-2 | 8.0e-9 | 5.9   | 23(22) |
|     |         | 1e-8   | 6.21720059e-2  | 4.2e-8 | 6.95  | 14     |
|     |         | 1e-9   | 6.21637312e-2  | 9.1e-10| 7.85  | 16     |
|     | KNITRO–D| 1e-6   | 6.290104777e-2 | 1.9e-7 | 3.51  | 4(11)  |
|     |         | 1e-8   | 6.21659812e-2  | 2.5e-9 | 6.4   | 8(30)  |
|     |         | 1e-9   | 6.2162183013e-2| 4.1e-10| 8.86  | 10(66) |
| 199 | INM–PCG | 1e-6   | 6.44262870e-2  | 8e-9   | 33.3  | 25(24) |
|     |         | 1e-8   | 6.519188743e-2 | 1.3e-7 | 20.39 | 6      |
|     |         | 1e-9   | 6.442826536e-2 | 2.3e-10| 45.2  | 15     |
|     | KNITRO–D| 1e-6   | 6.5203856841   | 8e-7   | 16.5  | 3(8)   |
|     |         | 1e-8   | 6.444127289e-2 | 3.6e-9 | 33.7  | 7(25)  |
|     |         | 1e-9   | 6.44335931e-2  | 7.8e-10| 44.1  | 9(47)  |
| 299 | INM–PCG | 1e-6   | 6.5193140696e-2| 9.3e-9 | 100.1 | 26(25) |
|     |         | 1e-8   | 6.5970554302e-2| 1.1e-7 | 66.3  | 6      |
|     |         | 1e-9   | 6.5269064297e-2| 1.8e-10| 92.4  | 9      |
|     | KNITRO–D| 1e-6   | 6.519765283e-2 | 6.2e-11| 159.1 | 16     |
|     |         | 1e-8   | 6.597880027e-2 | 5.2e-7 | 52.1  | 3(8)   |
|     |         | 1e-9   | 6.525495675e-2 | 1.1e-9 | 91.7  | 6(17)  |
|     | KNITRO–I| 1e-6   | 6.519876526e-2 | 3.0e-10| 147.2 | 10(46) |
| 399 | INM–PCG | 1e-6   | 6.5578165106e-2| 8.8e-10| 279.8 | 28(27) |
|     |         | 1e-8   | m              | m      | m     | m      |
|     |         | 1e-9   | m              | m      | m     | m      |
|     | KNITRO–D| 1e-6   | m              | m      | m     | m      |
|     |         | 1e-8   | m              | m      | m     | m      |
|     |         | 1e-9   | m              | m      | m     | m      |
|     | KNITRO–I| 1e-6   | m              | m      | m     | m      |
|     |         | 1e-8   | m              | m      | m     | m      |
|     |         | 1e-9   | m              | m      | m     | m      |

Table 2: Comparison of Inexact Newton Method + Preconditioned Conjugate Gradient with KNITRO v4.0.2 on test problem EDC–I ($\gamma = 0.001$)
Table 3: Comparison of Inexact Newton Method + Preconditioned Conjugate Gradient with KNITRO v4.0.2 on test problem EDC–III (M = 1, K = 0.8).

<table>
<thead>
<tr>
<th>N</th>
<th>Solver</th>
<th>opttol</th>
<th>$f(x^{(it)})$</th>
<th>$|H(v^{(it)})|$</th>
<th>time</th>
<th>it (inn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>INM–PCG</td>
<td>-6.57642730</td>
<td>9.4e-8</td>
<td>10.1</td>
<td>29(66)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>-6.57629592</td>
<td>3.9e-7</td>
<td>13.8</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-6.57640594</td>
<td>8.9e-9</td>
<td>14.8</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-6.57642744</td>
<td>1.1e-10</td>
<td>16.8</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-6.57585240</td>
<td>6.3e-7</td>
<td>9.9</td>
<td>6(30)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-6.57642667</td>
<td>5.2e-10</td>
<td>42.2</td>
<td>12(494)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-6.57642667</td>
<td>5.2e-10</td>
<td>42.2</td>
<td>12(494)</td>
</tr>
<tr>
<td>199</td>
<td>INM–PCG</td>
<td>-6.62009225</td>
<td>3.9e-8</td>
<td>46.3</td>
<td>27(61)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>-6.61987115</td>
<td>4.7e-8</td>
<td>63.2</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-6.62007441</td>
<td>5.6e-9</td>
<td>72.9</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-6.62009178</td>
<td>3.7e-10</td>
<td>79.5</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-6.61925123</td>
<td>1.2e-7</td>
<td>67.4</td>
<td>6(17)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-6.62009137</td>
<td>2.0e-9</td>
<td>185.2</td>
<td>12(198)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-6.62009225</td>
<td>6.0e-9</td>
<td>984.2</td>
<td>36(1802)</td>
</tr>
<tr>
<td>299</td>
<td>INM–PCG</td>
<td>-6.63464400</td>
<td>5.5e-9</td>
<td>126.3</td>
<td>27(55)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-6.63316140</td>
<td>1.3e-7</td>
<td>240.8</td>
<td>6(16)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-6.63462556</td>
<td>6.2e-9</td>
<td>373.5</td>
<td>9(38)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

* Current solution estimate cannot be improved by the code.

References:


<table>
<thead>
<tr>
<th>(N)</th>
<th>Solver</th>
<th>(\text{opttol})</th>
<th>(f(x^{(it)}))</th>
<th>(|H(v^{(it)})|)</th>
<th>time</th>
<th>it (inn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>INM–PCG</td>
<td>1e-6</td>
<td>-18.73614837</td>
<td>5.9e-9</td>
<td>13.9</td>
<td>49(54)</td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-8</td>
<td>-18.73611604</td>
<td>9.3e-9</td>
<td>14.8</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-18.73614224</td>
<td>6.8e-10</td>
<td>17.4</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-18.73221830</td>
<td>4.2e-7</td>
<td>20.5</td>
<td>19(133)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-18.73608419</td>
<td>7.8e-9</td>
<td>33.3</td>
<td>32(214)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-18.73614833</td>
<td>1.8e-10</td>
<td>38.6</td>
<td>38(235)</td>
</tr>
<tr>
<td>199</td>
<td>INM–PCG</td>
<td>1e-6</td>
<td>-18.86331163</td>
<td>4.9e-9</td>
<td>87.4</td>
<td>60(73)</td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>-18.84530188</td>
<td>4.5e-7</td>
<td>38.2</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-18.86315409</td>
<td>4.0e-9</td>
<td>119.7</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-18.86328069</td>
<td>8.1e-10</td>
<td>141.0</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-18.84592606</td>
<td>4.5e-7</td>
<td>88.4</td>
<td>14(118)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-18.86315369</td>
<td>4.7e-9</td>
<td>213.1</td>
<td>33(307)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-18.86329639</td>
<td>5.5e-10</td>
<td>248.2</td>
<td>40(339)</td>
</tr>
<tr>
<td>299</td>
<td>INM–PCG</td>
<td>1e-6</td>
<td>-18.905751265</td>
<td>3.3e-9</td>
<td>278.1</td>
<td>67(93)</td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>-18.862397636</td>
<td>5.0e-7</td>
<td>126.7</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-18.905307698</td>
<td>6.1e-9</td>
<td>361.2</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>-18.905681444</td>
<td>9.7e-10</td>
<td>472.5</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>-18.836892321</td>
<td>8.4e-7</td>
<td>309.1</td>
<td>14(153)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>-18.905301531</td>
<td>6.1e-9</td>
<td>578.6</td>
<td>28(282)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>399</td>
<td>INM–PCG</td>
<td>1e-6</td>
<td>-18.926980012</td>
<td>2.9e-7</td>
<td>825.2</td>
<td>76(113)</td>
</tr>
<tr>
<td></td>
<td>KNITRO–D</td>
<td>1e-6</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>KNITRO–I</td>
<td>1e-6</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-8</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-9</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 4: Comparison of Inexact Newton Method + Preconditioned Conjugate Gradient with KNITRO v4.0.2 on test problem EDC–III (\(M = 0, K = 1\))


