Contents

Introduction **3**

Acknowledgements **7**

1 Optimization framework **9**

1.1 Optimality conditions 9

1.2 Optimization in Banach spaces 11

2 Newton's methods **15**

2.1 Classical Newton's method 16

2.1.1 Globally convergent modifications of Newton's method 19

2.2 Inexact Newton methods 26

2.2.1 Local convergence 29

2.2.2 Global Inexact Newton methods 33

2.2.3 Line–Search Inexact Newton methods 35

2.2.4 Trust–Region methods 38

2.2.5 Nonmonotone Inexact Newton methods 41

2.3 Newton methods and dynamic systems 51

3 Interior–Point Methods **63**

3.1 Barrier methods 63

3.2 Perturbed Karush–Kuhn–Tucker systems 65

3.3 Newton Interior–Point Methods 68

3.3.1 Newton Line–Search IP Methods 69

3.4 Newton Interior-Point Methods as Inexact Newton Methods . 76

3.5 Previously proposed IP algorithms 79

3.5.1 LOQO 79

3.5.2 KNITRO 80

3.5.3 IPOPT 83

4 Description of the algorithm 85
4.1 The interior–point iteration 86
4.2 The search direction 89
4.2.1 The direct approach 90
4.2.2 The iterative approach: Hestenes method 90
4.2.3 The iterative approach: PCG 93
4.3 The nonmonotone version 97

5 Convergence analysis 99
5.1 Convergence theorems 99
5.2 Convergence in the nonmonotone case 112
5.3 A global convergence failure 114

6 Description of the test problems 117
6.1 Elliptic boundary control problems 117
6.1.1 Optimality condition in a special case: boundary elliptic control problem with Neumann boundary conditions117
6.1.2 Statement of the problem 121
6.1.3 Discretization and optimization formulation 122
6.1.4 Test problems: general description 127
6.1.5 Test problems: discretization technique. 129
6.2 Elliptic distributed control problems 143
6.2.1 Statement of the problem 143
6.2.2 Discretization and optimization formulation 145
6.2.3 Test problems: general description 146
6.2.4 Test problems: discretization techniques 147
6.3 Figures 158

7 Numerical experience 169
7.1 Implementation of the algorithm 169
7.2 The results 171
7.3 Results in the nonmononone case 173
7.4 Tables 175

Bibliography 191

Introduction

This thesis is concerned with the three following main topics: the analysis of the Newton interior–point methods for nonlinear programming, the analysis of the inexact Newton method for the solution of nonlinear system of equations and the numerical solution of optimal control problems by means of mathematical programming techniques.

The theory of interior–point methods has been developed since the 70's and they were initially proposed for linear programming, but they became unpopular soon because of their inherent ill-conditioning. Only recently, the introduction of new algorithms with "global" convergence properties led to a rediscovery of these methods and, in particular, good practical performances has been observed in nonlinear programming.

The study presented here is about a special class of interior–point methods, the Newton interior–point methods, which, in a more general framework, address to systems of nonlinear equations with nonnegativity bounds on some variables.

This remark allows the point of view adopted in this thesis, which consists in considering the Newton interior–point method as a special case of the inexact Newton methods for the solution of nonlinear systems of equations. In this framework, an interior–point algorithm has been proposed, and for this algorithm the convergence theory has been developed.

Furthermore, the analysis of the inexact Newton methods has yielded the introduction of the nonmonotone inexact Newton methods, and thus the introduction of a nonmonotone interior–point algorithm.

Crucial issues, which are deeply investigated here, in the analysis of the Newton interior–point methods are the solution, at each iterate, of the *perturbed Newton equation*, that is a linear system with symmetric coefficients matrix, and the modification of the computed direction, in order to guarantee the "global" convergence of the sequence of the iterates.

Finally, the study presented in this thesis is motivated by the intention of producing an efficient algorithm for the numerical solution optimal control

problems. Indeed, from the continuous formulation of an optimal control problem, a discrete version can be formulated as a large scale nonlinear programming problem involving structured and sparse jacobian and hessian matrices.
The transcription of a set of elliptic control problems into mathematical programming problems has been examined, taking into account also the relations between the continuous and the discrete formulation.
The numerical experience showed the good stability and efficiency of the proposed software. The performances obtained are better or comparable with the ones of some existing softwares.

In summary, the following significant contributions have been produced:

1. A “global” convergence theory (Chapter 5) of the Newton line–search interior–point method for the solution of Karush–Kuhn–Tucker systems has been formulated in the framework of the inexact Newton method (Section 2.2), allowing an approximate solution of the perturbed Newton equation, for example by means of an iterative inner solver.

2. A nonmonotone “global” convergence theory has been introduced and formulated for the inexact Newton method for the solution of nonlinear systems (Section 2.2.5) and for the line–search Newton interior–point method for KKT systems (Section 5.2). The nonmonotone case allows less restrictive choices of the perturbation parameter, of the inner stopping criterion and of the line–search acceptance rule (Section 4.3).

3. Different solvers of the perturbed Newton equation for the computation of the direction have been considered. In particular:

   (a) By employing elimination techniques to the perturbed Newton equation, the inner linear system at each iterate can be written in a “condensed” form (Section 3.3.1), which is equivalent to the optimality conditions of a quadratic programming problem. Thus, the Hestenes multipliers method has been proposed as iterative solver of the perturbed Newton equation in condensed form (Section 4.2.2). At each iteration of the Hestenes method, a linear system whose coefficients matrix is of the same dimension of the primal optimization variable, symmetric and positive definite, is solved.

   (b) The system in condensed form has been also solved by means of the conjugate gradient method with a suitable preconditioner

(PCG) (Section 4.2.3); in this case the main computational task at each interior–point iteration is the factorization of the preconditioner. The implementation choices adopted here lead to two different algorithms. The first one performs a block factorization of the preconditioner, which requires the Choleski factorization of a symmetric positive definite matrix of the same dimension of the equality constraints. The other one, provides the factorization of the whole preconditioner, which is a quasidefinite matrix whose dimension is the number of the primal variables plus the number of the equality constraints.

(c) An algorithm for the supernodal block factorization of quasidefinite matrices performing the minimum degree reordering has been proposed, allowing a dynamic regularization. This algorithm is a variant of the package of Ng and Peyton for sparse, symmetric, positive definite matrices and it has been employed for the direct factorization of the whole preconditoner.

(d) The performances of the algorithms implementing an iterative inner solver have been compared to the ones obtained by the algorithm employing a direct factorization of the condensed matrix (Section 4.2.1). The factorization routine chosen for this task is the MA27 subroutine, of the Harwell Subroutine Library.

4. The algorithm in the four versions described above has been coded in Fortran 90, also in the nonmonotone case. The evaluation of such software has been made on a testset of nonlinear and quadratic programming problems arising from the finite differences discretization of elliptic control problems (Chapter 6). The best version of the proposed method is the one implementing the PCG method with the direct factorization of the whole preconditioner, which has been able to solve problems with up to one million primal variables. The performances of this variant of the algorithm have shown to be better of the performances of LOQO, KNITRO and IPOPT (Table 3.1 in Chapter 3, Table 7.13 in Chapter 7), while the performances of the other variants with iterative inner solver are also comparable with the ones of these existing softwares.

Acknowledgments

I am very indebted to Professor Hans Mittelmann, for his precious suggestions and observations which have enriched my thesis. I thank him for his warm hospitality during my visit at the Arizona State University as well as his valuable scientific advices.

Chapter 1

Optimization framework

In this chapter the basic topics in constrained optimization are introduced. The finite dimension case is considered and the classical optimality conditions are reported. For sake of completeness, in the second section we briefly recall the basic results in the more general case of the optimization in Banach spaces, which also includes the optimal control theory.

1.1 Optimality conditions

The more general form of a nonlinear programming problem (NLP) is the following

$$\begin{array}{ll} \min & f(x) \\ s.t. & \begin{array}{l} g_1(x) = 0 \\ g_2(x) \geq 0 \end{array} \end{array} \tag{1.1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$, the equality and inequality constraints $g_1 : \mathbb{R}^n \to \mathbb{R}^{neq}$, $g_2 : \mathbb{R}^n \to \mathbb{R}^m$ are supposed to be twice continuously differentiable. The symbol $\nabla f(x)$ denotes the gradient of the objective function $\nabla f = (\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n})^t$, while the matrices $\nabla g_1(x)$ and $\nabla g_2(x)$ denote the transpose of the jacobian of the constraints, i.e.

$$\nabla g_1(x) = \begin{pmatrix} \frac{\partial (g_1)_1(x)}{\partial x_1} & \cdots & \frac{\partial (g_1)_{neq}(x)}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial (g_1)_1(x)}{\partial x_n} & \cdots & \frac{\partial (g_1)_{neq}(x)}{\partial x_n} \end{pmatrix} \quad \nabla g_2(x) = \begin{pmatrix} \frac{\partial (g_2)_1(x)}{\partial x_1} & \cdots & \frac{\partial (g_2)_m(x)}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial (g_2)_1(x)}{\partial x_n} & \cdots & \frac{\partial (g_1)_m(x)}{\partial x_n} \end{pmatrix}.$$

Here $(g_1)_i(x)$ or $(g_2)_i(x)$ denote the generic i–th component of $g_1(x)$ and $g_2(x)$ respectively.
The Lagrangian function associated to the problem (1.1) can be written as

$$\mathcal{L}(x, \lambda, w) = f(x) - \lambda^t g_1(x) - w^t g_2(x) \tag{1.2}$$

where the vectors λ and w are the Lagrange multipliers for the equality and inequality constraints respectively.
A point x_* is a local solution of the minimum problem (1.1) if it is feasible, which means that the equality and inequality constraints are satisfied in x_*, and if $f(x_*) \leq f(x)$ for any x in the neighborhood of x_* $N_\delta(x_*) = \{x \in \mathbb{R}^n : \|x - x_*\| < \delta\}$, for some $\delta > 0$. Under suitable assumptions on the constraints, necessary and sufficient conditions can be given to characterize the local minima. One of these assumptions involves the notion of regularity of a point $x \in \mathbb{R}^n$ with respect to a set of constraints.

Definition 1.1 A feasible point x satisfying $(g_2)_i(x) = 0$ $\forall i \in I(x)$, where $I(x)$ is a subset of $\mathcal{I} = \{i \in \mathbb{N} : 1 \leq i \leq m\}$, is said to be a regular point of the constraints g_1 and g_2 if the gradient vectors $\nabla(g_1)_i(x)$ $\nabla(g_2)_j(x)$ with $i = 1, \cdots, neq$, and $j \in I(x)$ are linearly independent.

Definition 1.2 A point x is a Karush-Kuhn-Tucker (KKT) point if there exist two vectors $\lambda_1 \in \mathbb{R}^{neq}$ and $\lambda_2 \in \mathbb{R}^m$ such that

$$\nabla f(x) - \nabla g_1(x)\lambda - \nabla g_2(x)w = 0 \tag{1.3}$$

$$g_1(x) = 0 \tag{1.4}$$

$$w^t g_2(x) = 0 \tag{1.5}$$

$$w \geq 0 \quad g_2(x) \geq 0 \tag{1.6}$$

The condition (1.3) is equivalent to set the gradient of the lagrangian function respect to the variables $x_1, ...x_n$ equal to zero. Condition (1.5) is usually called complementarity condition: indeed, from (1.5) and (1.6) it follows

$$w_i(g_2)_i(x) = 0 \tag{1.7}$$

Conditions (1.3)–(1.6) are the first order necessary conditions for the minimum problem (1.1), also called Karush-Kuhn-Tucker optimality conditions.

Proposition 1.1 If a regular point x_* is a local minimum for the problem (1.1), then x_* is a KKT point.

The proof of the proposition above can be found for example in [54, p. 314]. Sufficient conditions can be stated with some more information about the convexity of the lagrangian function in a neighborhood of the KKT point.

Proposition 1.2 Let x_* be a regular point for the problem (1.1). If x_* is a KKT point such that the hessian matrix of the lagrangian function

$$\nabla^2_{xx}\mathcal{L}(x_*, \lambda, w)) = \nabla^2 f(x_*) - \sum_{1=1}^{neq} \lambda_i \nabla^2 (g_1)_i(x_*) - \sum_{1=1}^{m} w_i \nabla^2 (g_2)_i(x_*)$$

is positive definite on the space

$$M = \{y \in \mathbb{R}^n : \nabla g_1(x_*)^t y = 0, \nabla (g_2)_i(x_*)^t y = 0, \quad \forall i \in A(x_*)\}$$

where

$$A(x_*) = \{i : 1 \leq i \leq m, \quad (g_2)_i(x_*) = 0, \quad w_i > 0\}$$

then x_* is a minimum point.

Here we denote by $\nabla^2 f$, $\nabla^2 (g_1)_i(x)$ and $\nabla^2 (g_2)_i(x)$ the hessian matrices of $f(x)$, $(g_1)_i(x), i = 1, ...neq$, $(g_2)_i(x), i = 1, ...m$. For instance:

$$\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}.$$

The proof of the proposition can be found in [54, p. 316]. The set $\{(g_2)_i(x) : i \in A(x)\}$ is referred as the set of the constraints which are active at the point x.

1.2 Optimization in Banach spaces

The previous theorems can be incorporated in a more general theory where an optimization problem can be expressed as: given a subset of a vector space, find the vector which minimizes a given functional in that subset. For example, any variational problem and any optimal control problem belong to this class. In the following we report the theorems which characterize the solution of the minimum problem when the feasible subset is defined by means of functional equalities or inequalities.

Consider the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.}t. & H(x) = 0 \end{array} \tag{1.8}$$

where f is a real valued functional on a Banach space X and H is a mapping from X into a Banach space Z. In analogy with the definition of regular point given in the previous section, we report the definition of regular point for the mapping H and the Lagrange multiplier's theorem.

Definition 1.3 Let H be a continuously Fréchet differentiable [1] transformation from an open set D in a Banach space X into a Banach space Y. If $x_0 \in D$ is such that $H'(x_0)$ maps X onto Y, the point x_0 is said to be a regular point of the transformation H.

Theorem 1.1 (Lagrange multiplier's theorem [53, p.243]) If the continuously differentiable functional f has a local extremum subject to the constraint $H(x) = 0$ at the regular point x_0, then there exists an element z_0^* in the dual space [2] Z^* of Z such that the lagrangian functional

$$\mathcal{L}(x) = f(x) + z_0^* H(x)$$

is stationary at x_0, i.e. $f'(x_0) + z_0^* H'(x_0) = 0$.

For the inequality constraints, we have to define a relation which indicates the positivity of the elements in a vector space.

Definition 1.4 Let P a convex cone in a vector space X. For $x, y \in X$, we write $x \geq y$ (with respect to P) if $x - y \in P$. The cone P defining this relation is called the *positive cone* in X.

Now we can consider the following inequality constrained minimum problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.}t. & G(x) \geq 0 \end{array} \tag{1.9}$$

where f is defined in a vector space X and G is a mapping from X into the normed space Z having positive cone P. Under some assumption (see [53, p.

[1] Let be T a transformation defined on an open domain D in a normed space X and having range in a normed space Y. If, for fixed $x \in D$ and each $h \in X$, there exists $\delta T(x; h) \in Y$ which is linear and continuous with respect to h such that

$$\lim_{\|h\| \to 0} \frac{\|T(x+h) - T(x) - \delta T(x; h)\|}{\|h\|} = 0$$

then T is said to be Fréchet differentiable at x and $\delta T(x; h)$ is said to be the Fréchet differential of T at x with increment h [53, p.172].

[2] The dual space X^* of a normed vector space X is the space of all bounded linear functionals on X [53, p. 106].

249, problem 9, p.267]) it is possible to prove the Generalized Kuhn–Tucker Theorem, which ensures the existence of a multiplier z_* corresponding to a minimum x_* of the problem (1.9) such that $z_* \in Z^*$ and that the Lagrangian

$$f(x) + \langle G(x), z_* \rangle$$

is stationary at x_*, where $\langle , \rangle$ indicates the inner product in the space Z. Furthermore the complementarity condition $\langle G(x_*), z_* \rangle$ holds in the solution.

Chapter 2

Newton's methods

The Newton's method is strictly related to the idea of optimization: it can be considered an optimization method itself for the unconstrained case and furthermore, for the constrained case, it can be used to solve the optimality conditions in order to find a Karush–Kuhn–Tucker point. In the first section the classical Newton's method is presented, and for sake of completeness, its basic principles, features and convergence results are also reported. Then, particular attention is given to the class of the inexact Newton methods, whose convergence theorems have been revisited and reported in the second section. In Section 2.2.5 the inexact Newton method is extended to the nonmonotone case for which it has been possible to prove convergence results analogous to the one stated in the classical monotone case. The "global" convergence of a nonmonotone line–search backtracking algorithm is proved with Theorem 2.13.

The introduction of the nonmonotone inexact Newton method is an original contribution of this thesis.

Finally, in the third section, some observations about the relations between Newton's method and Eulero's method for the solution of a dynamic system are explained, following the line proposed in [40].

From this point of view, the failure of the Newton method in some examples known in literature can be justified with new arguments.

2.1 Classical Newton's method

The Newton's method's is based on the linearization idea, which means to construct a linear approximation of the nonlinear problem

$$\begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = 0$$

that in vector form can be written as

$$F(x) = 0 \tag{2.1}$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denotes a mapping defined in some open subset E of $\mathbb{R}^n$, whose components are the f_i.

The more natural linear model of F in a neighborhood of a given point x_k can be obtained by the Taylor's expansion of F of the first order:

$$L_k(x) = F(x_k) + F'(x_k)(x - x_k),$$

where $F'(x)$ is the jacobian matrix of F defined by

$$F'(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix}.$$

The linear function $L_k(x)$ has the property to agree with F at x_k and it is a good approximation of F in a neighborhood of x_k.

The classical Newton's method builds a sequence of points x_k such that $L_k(x_{k+1}) = 0$, which means that, the iterates can be computed by solving the *Newton equation*

$$F'(x_k)s_k^N + F(x_k) = 0 \tag{2.2}$$

where the solution vector s_k^N is the *Newton step*, and with the updating rule

$$x_{k+1} = x_k + s_k^N.$$

We recall the local convergence results for the Newton method which can be derived from the fixed–point theorem. Indeed, the Newton iteration can be written as

$$x_{k+1} = x_k - F'(x_k)^{-1}F(x_k) \tag{2.3}$$

which is a special case of a method of successive approximations

$$x_{k+1} = K(x_k). \tag{2.4}$$

We recall the standard fixed–point theorem and the definition of contraction mapping:

Definition 2.1 Let $E \subset \mathbb{R}^n$. A mapping $K : E \to \mathbb{R}^n$ is a *contraction mapping* on E if K is Lipschitz continuous [1] on E with Lipschitz constant $\gamma < 1$.

Theorem 2.1 (Contraction Mapping Theorem) Let C a closed subset of E and let K a contraction mapping on C with Lipschitz constant $\gamma < 1$ such that $K(x) \in C$ for any $x \in C$. Then, there exists a unique *fixed point* x_* of K such that $x_* = K(x_*)$ in C and the sequence $\{x_k\}$ generated by the iteration defined in (2.4) converges Q–linearly [2] to x_* for all initial iterate $x_0 \in C$.

For the proof of the previous theorem see for example [63, p.36].
The contraction mapping theorem can be also applied to the Newton iteration (2.3), but it allows to show only linear convergence, as pointed out in [47], thus the convergence of the Newton's method is proved following other ways. Indeed, there exist many local convergence theorems for the Newton's method, which also give an estimate of the convergence speed. We can

[1] A function K(x) is Lipschitz continuous on a set E with Lipschitz constant γ if

$$\|K(x) - K(y)\| \leq \gamma \|x - y\|$$

for all $x, y \in E$.

[2] The convergence rate considered here is the Q–order convergence rate. For sake of completeness we report the definitions. Let $\{x_k\}$ be a sequence in $\mathbb{R}^n$ that converges to x_*. The convergence is Q–linear if there exists a positive constant $t < 1$ such that

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \leq t$$

for k sufficiently large.
The convergence is said to be Q–superlinear if

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} = 0$$

for k sufficiently large. Q-quadratic convergence is obtained if

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^2} \leq M$$

for k sufficiently large where M is a positive constant.

distinguish two approaches: the more standard approach is to state some assumptions on F and on the jacobian matrix F' in a neighborhood of the solution or at the solution itself, while the Kantorovich approach provides conditions on the starting point x_0. The standard assumption are

A1 There exists a solution x_* of the problem 2.1.

A2 $F'(x_*)$ is nonsingular.

A3 The jacobian $F' : E \to \mathbb{R}^{n \times n}$ is Lipschitz continuous, where E is a neighborhood of x_*.

or the following *affine invariant assumption*

A3' There exists a $\omega \geq 0$ such that

$$\|F'(x)^{-1}(F'(x+sv) - F'(x))v\| \leq s\omega\|v\|^2$$

for all $s \in [0,1]$, $v \in \mathbb{R}^n$ and for any $x \in D$, $x + vs \in D$, where D is a neighborhood of x_* in which $F'(x)$ is nonsingular.

and they guarantee the convergence of the Newton sequence, starting from an initial point sufficiently close to the solution, as claimed in the following result.

Theorem 2.2 Suppose that A1 and A2 hold. If the hypothesis A3 or the hypothesis A3' is verified, then there exists a positive number δ such that if $N_\delta(x_*) \subset E \cap D$ [3] and if $x_0 \in N_\delta(x_*)$, then the Newton iteration (2.3) converges quadratically to x_*.

Proof. If A3 holds, then the thesis follows from Theorem 5.1.2 in [47, p.71] or also Theorem 5.2.1 in [26, p.90].
Suppose now that A3' is verified. We observe that, for the continuity of F', from A1 there exists a positive number $\bar{\delta}$ such that the matrix $F'(x)$ is nonsingular for any $x \in N_{\bar{\delta}}(x_*)$. Then, if $\delta = \min\{\bar{\delta}, \frac{2}{\omega}\}$, the thesis follows from Theorem 4.10 in [27, p.97]. □

Remark. Under the affine invariant condition A3', the radius of attraction δ of the Newton method depends on the quantity $2/\omega$, where ω is the affine invariance constant. An analogous result can be obtained under the assumption A3: in this case the radius of the attraction region for the Newton

[3] Here and in the following $N_\delta(x_*)$ denotes the neighbourhood of x_* with radius δ, i.e. the set $\{x \in \mathbb{R}^n : \|x - x_*\| \leq \delta\}$.

method is $\delta \leq 2/(3\gamma\|F'(x_*)^{-1}\|)$ (see [63, p.44]).

As mentioned before, in the Kantorovich approach the convergence theorem is proved under suitable hypotheses on the starting point x_0. Such hypotheses are

K1 $\|F'(x_0)\|$ is nonsingular.

K2 There exist two positive constants β and η such that

$$\|F'(x_0)\| \leq \beta, \text{ and } \|F'(x_0)^{-1}F(x_0)\| \leq \eta$$

K3 There exists δ such that F' is Lipschitz continuous with Lipschitz constant γ in $N_\delta(x_0)$.

Theorem 2.3 (Kantorovich) Assume that K1–K3 hold. If $\beta\eta\gamma \leq 1/2$ and $\delta > \delta_0$ where $\delta_0 = (1 - \sqrt{1 - 2\beta\eta\gamma})/(\beta\gamma)$, then there exists a unique solution x_* of the problem (2.1) in the closure of $N_{\delta_0}(x_0)$ and the iteration (2.3) is well defined and converges to x_* with a R–quadratic convergence rate[4].

For a proof of the Kantorovich theorem we refer to [61], we only observe that the form of Theorem 2.3 is similar to the form of the contraction mapping theorem 2.1, as pointed out also in [26]. Indeed, both theorems identify a region in which, under some assumptions a unique root of F exists and, starting from a point belonging to that region, the iterations of type (2.4) converge to the solution.

2.1.1 Globally convergent modifications of Newton's method

In the previous section, the Newton's method has been shown to be Q–quadratically convergent to a solution of the problem $F(x) = 0$, when the initial point of the sequence is sufficiently close to the solution. In other words, if good properties, as the nonsingularity of the jacobian matrix, hold at the solution, there is a "good" region around this solution in which such properties hold too. This section deal with the two major ideas to get into

[4] Let $\{x_k\} \subset \mathbb{R}^n$ and $x_* \in \mathbb{R}^n$. Then $\{x_k\}$ converges to x_* R-(quadratically–superlinearly, linearly) if there exists a sequence $\{\xi_k\}$ converging Q–(quadratically–superlinearly, linearly) to zero such that

$$\|x_k - x_*\| \leq \xi_k.$$

this region, the line–search and the trust–region, which provide the criteria for evaluating when the Newton step is unsatisfactory and the strategies to proceed.

Line-search strategy

The general framework of the line search approach applied to the Newton's method is the following: at the iterate k, given the Newton step s_k^N, check the *acceptance rule*, then, if it is not satisfied, reduce the steplength along the direction s_k^N such that reduced step can be accepted by the rule. In this contest, it is natural to introduce the idea of the *merit functions*, which are real valued functions employed to measure the progress toward the solution. For the problem (2.1), a measure of the distance from the solution is the function

$$f(x) = \frac{1}{2}\|F(x)\|^2, \tag{2.5}$$

where $\|\cdot\|$ indicates the euclidean norm in $\mathbb{R}^n$, and a reasonable acceptance criterion is to require that in two successive Newton iterations the value of f decreases:

$$f(x_{k+1}) < f(x_k).$$

It is evident the relation between the root–finding problem (2.1) and the unconstrained minimum problem

$$\begin{array}{cc} \min & f(x) \\ x \in \mathbb{R}^n & \end{array} \tag{2.6}$$

since a solution of (2.1) is also a solution of (2.6), but it could exist a local minimizer of f which is not a root of F. We recall that all the strategies explained here can be applied to the root finding problem but also to a general unconstrained minimum problem of the form (2.6), where f is a generic real valued function. Indeed, for the minimum problem we can introduce the definition of *descent direction* as follows.

Definition 2.2 A vector s_k is a *descent direction* for the problem (2.6) in the point x_k if

$$\nabla f(x_k)^t s_k < 0.$$

If s_k is a descent direction for the function f in x_k, then the decrease of the function f is guaranteed, for sufficiently small values of α_k. Indeed, by setting $f_k(\alpha) = f(x_k + \alpha s_k)$, we have $f_k'(0) = \nabla f^t(x_k) s_k < 0$. Hence, for sufficiently small values of α_k, Taylor's theorem ensures that $f(x_k + \alpha s_k) =$

$f_k(\alpha) < f_k(0) = f(x_k)$.
We observe that the vector $-\nabla f(x_k)$, called *steepest descent direction* is a descent direction while the Newton step is a descent direction for the problem (2.6) with f defined as in (2.5) at each Newton iterate x_k.
In general, provided a descent direction s_k at a given point x_k, the best candidate along this direction for the next iterate is the point $x_k + \alpha_k s_k$, where α_k the solution of the one dimensional problem

$$\min \quad f(x_k + \alpha s_k), \quad \alpha > 0.$$

Nevertheless, in practical algorithms it is not necessary to compute the exact solution of that minimum problem, but it is sufficient to require that in the new point there is a sufficient decrease of f. A well known accepting rule for such decrease is the *Armijo* condition

$$f(x_k + \alpha_k s_k) \leq f(x_k) + \beta \alpha_k \nabla f(x_k)^t s_k. \tag{2.7}$$

The Armijo condition implies that the decrease of f is at least a multiple of the distance between two successive iterates. Indeed, if s_k is a descent direction for f in x_k and if we set

$$\omega = -\frac{\beta s_k^t \nabla f(x_k)}{\|s_k\|} > 0,$$

then by means of (2.7) we obtain

$$\frac{f(x_k) - f(x_{k+1})}{\|x_k - x_{k+1}\|} = \frac{f(x_k) - f(x_{k+1})}{\alpha_k \|s_k\|} \geq \omega,$$

and from this we have

$$f(x_k) - f(x_{k+1}) \geq \omega \|x_k - x_{k+1}\|.$$

This means that if the distance between two successive iterates is large, then is also large the amount of the decrease of f.
The *Wolfe* condition is often associated to (2.7) and it is expressed by requiring that the following inequality holds for a fixed value of γ such that $0 < \beta < \gamma < 1$.

$$s_k^t \nabla f(x_{k+1}) > \gamma s_k^t \nabla f(x_k). \tag{2.8}$$

Such condition prevents α_k to become too small, as pointed out for example in [60, pp. 39–40].
The convergence theory for the line–search methods with (2.7) and (2.8) as acceptance rules can be resumed by the following theorem.

Theorem 2.4 Let $\{x_k\}$ be a sequence of iterate such that $x_{k+1} = x_k + \alpha_k s_k$, where, at each iterate k, s_k is a descent direction for f and α_k satisfies the Armijo-Wolfe conditions (2.7) and (2.8). Suppose that f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set E containing the level set $\Omega = \{x : f(x) \le f(x_0)\}$, where x_0 is the starting point of the iteration and assume that the gradient ∇f is Lipschitz continuous in E. Then the *Zoutendijk condition*

$$\sum_{k=1}^{\infty} \cos^2 \xi_k \|\nabla f(x_k)\| < \infty \tag{2.9}$$

holds, where ξ_k is the angle between s_k and the steepest descent direction $-\nabla f(x_k)$.

If the descent direction s_k is chosen such that the angle ξ_k is bounded away from 90 degrees, then $\cos \xi_k$ is bounded away from 0 and the Zoutendijk condition implies that

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0. \tag{2.10}$$

It is worth to stress that condition (2.10) does not guarantee that the sequence $\{x_k\}$ is convergent. If we assume that the sequence $\{x_k\}$ has a limit point x_*, then under the assumptions of Theorem 2.4 we can conclude that x_* is a stationary point for f, that is $\nabla f(x_*) = 0$. On the other hand, when we consider the root–finding problem and we choose f as in (2.5), if we assume that there exist a limit point x_* of the sequence $\{x_k\}$ such that $F'(x_*)$ is nonsingular, then (2.10) implies that $F(x_*) = 0$.

The implementation of the line–search strategy is often obtained by means of a *backtracking* procedure, that consists in starting from the full step s_k, then reducing the steplength α_k by a value $\theta_k < 1$ until the acceptance rule is satisfied, according to the following scheme:

Scheme 2.1 (Backtracking technique)

```
At the iteration $k$, given a descent direction $s_k$ and a scalar $\theta_k < 1$
Set $\alpha_k = 1$.
Until the acceptance rule is satisfied do:
    Set $\alpha_k = \theta_k \alpha_k$
Set $x_{x+1} = x_k + \alpha_k s_k$.
```

The previous scheme allows to retain the full step if it satisfies the acceptance rule, and this choice can make the backtracking algorithm very effective when the Newton step is taken. Indeed, when the iterate is close to the solution, the quadratic convergence rate of the Newton method can be maintained, because the full Newton step can be accepted.
In general, a good acceptance rule for the backtracking technique should be not too restrictive in order to allow sufficiently large step length and thus a significant progress toward the solution at each iteration.

Trust-region strategy

In the trust–region approach, at the iterate k a quadratic function $m_k(s) = m_k(x_k + s)$ is chosen as model of the function f. Then, a region of the space around x_k is determined such that $m_k(s)$ is trusted to be a good approximation of f in that region. Generally the trust–region is chosen as a sphere or an ellipse centered in the current point x_k, hence determining the trust–region means to choose the radius of the region. Once determined the radius δ_k, the step s_k is computed as the minimizer of $m_k(s)$ over the trust–region, namely s_k solves the following constrained quadratic programming problem:

$$\begin{array}{ll} \min & m_k(s) \\ s.t. & \|s\| \le \delta_k \end{array} \tag{2.11}$$

Referring to the unconstrained minimum problem (2.6), a choice for the quadratic model is

$$m_k(s) = f(x_k) + \nabla f(x_k)^t s + \frac{1}{2} s^t \nabla^2 f(x_k) s \tag{2.12}$$

and if we refer to the root–finding problem, we could choose f as in (2.5), obtaining

$$\nabla f(x_k) = F'(x_k)^t F(x_k)$$

and

$$\nabla^2 f(x_k) = F'(x)^t F'(x) + \sum_{i=1}^{n} f_i(x) \nabla^2 f_i(x).$$

A slightly different quadratic model for the problem (2.1) with the same linear term but different hessian matrix, is the following:

$$\begin{aligned} m_k(s) &= \frac{1}{2}\|F'(x)s + F(x)\|^2 \qquad (2.13) \\ &= \frac{1}{2}F(x_k)^t F(x_k) + (F'(x_k)^t F(x_k))^t s + \frac{1}{2} s^t (F'(x_k)^t F'(x_k)) s. \end{aligned}$$

We observe that if $F'(x_k)$ is nonsingular, then the matrix $F'(x_k)^t F'(x_k)$ is positive definite and the Newton step s_k^N is the unique global minimizer of $m_k(s)$ in (2.13).
The step computed by minimizing the quadratic model is accepted if it satisfies an acceptance rule: as for the line–search approach, it is generally required that the new iterate $x_k + s_k$ gives a sufficient reduction of a merit function, which can be the object function f of the minimum problem (2.6) itself or the least squares function (2.5). If this does not occur, the step is rejected, the radius of the trust–region is reduced and the process is repeated until an acceptable step is computed according to the following general scheme:

Scheme 2.2 (Trust–region technique)

```
Until the acceptance rule is satisfied do:
    Choose the trust–region radius $\delta_k$.
    Compute a vector $s_k$, solution of the problem (2.11).
Update the iterate $x_{k+1} = x_k + s_k$.
Update the radius $\delta_{k+1}$.
```

The main difference between the line–search and the trust–region approach, is that in the latter case the radius δ_k controls not only the step length, but also the direction. Indeed, by changing the value of δ_k, the solutions of (2.11) give different vectors.
A frequent choice for the acceptance rule is the following: let us define the *actual reduction* of the function f as

$$ared_k(s_k) = f(x_k) - f(x_k + s_k) \tag{2.14}$$

which corresponds to the amount of the decrease of f, and the *predicted reduction*

$$pred_k(s_k) = m_k(0) - m_k(s_k) \tag{2.15}$$

namely the decrease of the quadratic model function in two successive iterates. The step s_k is accepted if the ratio $\nu = \frac{ared_k(s_k)}{pred_k(s_k)}$ is grater than a positive fixed quantity t

$$ared_k(s_k) > t \cdot pred_k(s_k).$$

Furthermore, in many practical algorithms, the updating rule for the trust–region radius for the next iterate depends on the value of ν: if it is grater than a fixed quantity $u > t$, it means that the quadratic model gives a good approximation of the function f and the radius can be increased. The aim of this procedure is to allow larger step when we are close to the solution: for example, if the quadratic model is (2.13), the full Newton step could be taken, improving the convergence rate.
About the solution of the quadratic subproblem (2.11), we report the Levenberg–Marquardt characterization of the solutions:

Theorem 2.5 The vector s_* is a global solution of the trust–region subproblem

$$\begin{array}{ll} \min & c + b^t s + \frac{1}{2} s^t A s \\ s.t. & \|s\| \leq \delta \end{array} \tag{2.16}$$

if and only if there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:

$$\begin{aligned} (A + \lambda I)s_* &= -b && (2.17) \\ \lambda(\delta - \|s_*\|) &= 0 && (2.18) \\ (A + \lambda I) & \quad \text{is a positive semidefinite matrix} && (2.19) \end{aligned}$$

For the proof we refer for example to [60, p.84] and we report the formulation of the conditions (2.17)–(2.19) for the case (2.13):

$$\begin{aligned} (F'(x_k)^t F'(x_k) + \lambda I)s_* &= -F'(x_k)^t F(x_k) \\ \lambda(\delta - \|s_*\|) &= 0. \end{aligned}$$

The previous characterization of the exact solution of the trust–region subproblems has also a practical importance. However, it is not necessary to compute an exact solution of the quadratic subproblem, but it is sufficient to compute a direction with some suitable properties which allow to prove the convergence of the algorithm. Referring to the generic problem (2.16), such properties can be formulated by requiring that the following inequality holds at each iteration:

$$m_k(0) - m_k(s_k) \geq c\|b\| \min\left(\delta_k, \frac{\|b\|}{\|A\|}\right). \tag{2.20}$$

Under this condition, it is possible to prove convergence theorems (see [60, p.89–93]) whose thesis is the same as in the line–search framework: indeed,

under some assumptions, condition (2.10) is guaranteed, but the convergence of the sequence $\{x_k\}$ is not ensured.
The first approximation of the solution of (2.11) can be found by calculating the *Cauchy point*: it is defined as $s_k^C = \tau_k s_k^S$, where s_k^S is a solution of a linear version of (2.11), that is $s_k^S = \text{argmin}\ (c + b^t s)$, and the scalar τ_k minimizes $m_k(\tau s_k^S)$ subject to the trust–region bound $\|\tau s_k^S\| \leq \delta_k$. It is easy to prove (see [60, p.70]) that the Cauchy point satisfies (2.20).
In order to improve the accuracy of the approximation of the solution of (2.11) provided by the Cauchy point, many methods for the computation of a direction which satisfies (2.20) have been proposed, and the two more popular techniques are the *dogleg* method and the Steihaug implementation of the conjugate gradient method.
The former one finds an approximate solution of the problem (2.16) by minimizing the quadratic model along a path consisting of two line segments. The first segment has origin in the current point and terminates in the unconstrained minimizer along the steepest descent direction for (2.16): in other words the first line is the vector $-(b^t b/b^t Ab)b$. The second segment runs from the previous point to the unconstrained minimizer of (2.16) defined by $-A^{-1}b$. This procedure is justified because the dogleg path is an approximation of the curve $s_*(\delta)$, that is the curve of the exact solutions of (2.16) depending on the values of the radius δ.
The Steihaug approach consists in a modification of the classical conjugate gradient method, which terminates either when the conjugate gradient iterates violates the trust region bound $\|s\| \leq \delta$ or when a direction of negative curvature in A is encountered. For further details and proofs we refer to [67].

2.2 Inexact Newton methods

The inexact Newton methods have been firstly proposed in [25]. The idea of these methods is to give a condition on the direction along which the new iterate will be computed, guaranteeing the convergence to a solution. Such condition requires that, at each step k, the direction s_k satisfies

$$\|F'(x_k)s_k + F(x_k)\| \leq \eta_k \|F(x_k)\| \tag{2.21}$$

for some *forcing term* $\eta_k \in [0, 1)$.
If (2.21) holds, then the direction s_k, called *inexact Newton step* at the level η_k, can be considered an approximation of the Newton direction. Indeed, the left-hand-side of (2.21) is the norm of the residual of the Newton equation.

From (2.21) we observe that the ratio between the residual of the Newton equation and the "outer" residual $\|F(x_k)\|$ is controlled by the forcing term η_k.
An *inexact Newton method* is any method which, given an initial guess x_0, generates a sequence $\{x_k\}$ as follows:

```
For $k = 0, 1, 2, \ldots$
    Find some $\eta_k \in [0, 1)$ and a vector $s_k$ that satisfy (2.21);
    Set $x_{k+1} = x_k + s_k$.
```

Before to give further theoretical details, it is worth to make some observations about condition (2.21) and to illustrate the situation with an example. Consider the system of two nonlinear equation

$$F(x) = \begin{pmatrix} x_1 + x_2 - 5 \\ x_1 x_2 - 4 \end{pmatrix} = 0. \tag{2.22}$$

The two solution of (2.22) are $(1, 4)$ and $(4, 1)$ and the jacobian matrix $F'(x)$ is given by

$$F'(x) = \begin{pmatrix} 1 & 1 \\ x_2 & x_1 \end{pmatrix}.$$

Supposing that $x_k = (0, 3)$, we have that $F'(x_k)$ is nonsingular and the Newton step s_k^N is the vector $(1.\bar{3}, 0.\bar{6})$, drawn with the black solid arrow in the figure 2.1, where the contour lines of $\|F(x)\|$ are also reported. In the same figure, the inexact Newton steps s_k which satisfy (2.21) (with $\eta_k = 0.3$) are the dotted arrows, and the circles indicates the points $x_{k+1} = x_k + s_k$, which form a "cloud" of points around $x_{k+1}^N = x_k + s_k^N$. If the forcing term η_k is increased, then the cloud has a larger diameter. Furthermore, it is worth to stress that

- the method is independent of the way to compute the direction s_k;

- $F'(x_k)$ is not required to be nonsingular.

Note that condition (2.21) guarantees that the inexact Newton step is a descent direction for the scalar function

$$\Phi(x) = \frac{1}{2}\|F(x)\|_2^2. \tag{2.23}$$

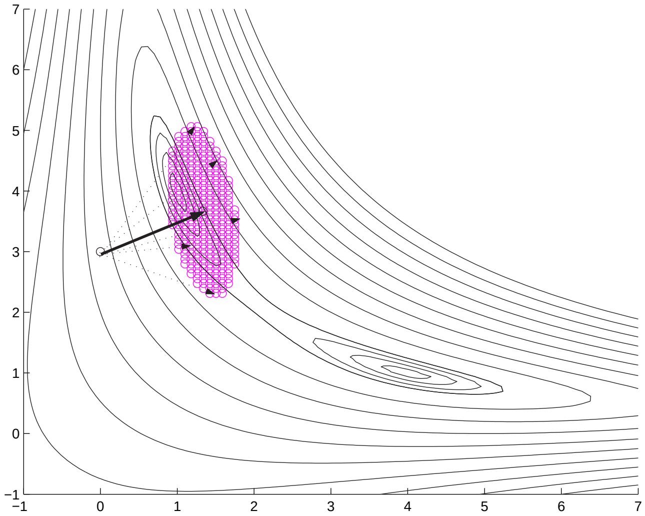


Figure 2.1: Inexact Newton step: nonsingular jacobian matrix

Indeed we have the following inequality (we omit the iteration index):

$$\begin{aligned}
\nabla\Phi(x)^t s &= F(x)^t F'(x)s \\
&= F(x)^t[-F(x) + F'(x)s + F(x)] \\
&= -\|F(x)\|_2^2 + F(x)^t(F'(x)s + F(x)) \\
&\leq -(1-\eta)\|F(x)\|_2^2 \leq 0.
\end{aligned}$$

When $F'(x_k)$ is nonsingular, there exists a unique Newton direction, as in the previous example, and a practical way to compute an inexact step is to apply an iterative inner solver to the Newton equation

$$F'(x_k)s = -F(x_k)$$

until condition (2.21) is satisfied. Thus condition (2.21) represents an adaptive stopping criterion for the iterative inner solver where the accuracy of the solution of the linear system depends on the value $\|F(x_k)\|$ which is large for the initial iterations, and it become smaller when x_k is approaching to the solution. It follows that the inexact Newton method particularly suited for large scale problems, because unnecessary and costly computations can be avoided.

If the matrix $F'(x_k)$ is singular, then two cases can occur: either the Newton equation is possible and admits an infinite number of solutions, and then

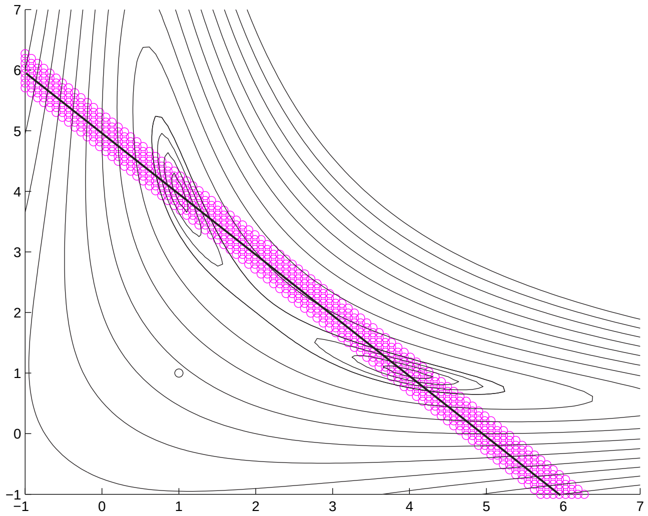


Figure 2.2: Inexact Newton step: singular jacobian matrix case 1

there exists an infinite number of Newton directions, or the Newton equation has no solutions.

Figure 2.2 refers again to the example (2.22), but now x_k is the point $(1, 1)$, where the jacobian is the singular matrix

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

and $F(x_k) = (-3, -3)^t$. In this case the system $F'(x_k)s = -F(x_k)$ admits an infinite number of solutions, which are all the vectors with origin in x_k and end point in the points lying on the line $x_1 + x_2 = 5$. The circles around the line $x_1 + x_2 = 5$ are the points $x = x_k + s$ where s is an inexact Newton step with $\eta_k = 0.1$.

Starting from the point $x_k = (2, 2)$, the system has no solutions, for the Rouché–Capélli theorem, but with $\eta_k = 0.9$ there exist infinite inexact Newton steps s; the points marked with a black circle in the figure 2.3 represent $x = x_k + s$.

2.2.1 Local convergence

The inexact Newton method has a local linear convergence property, under the following standard assumptions:

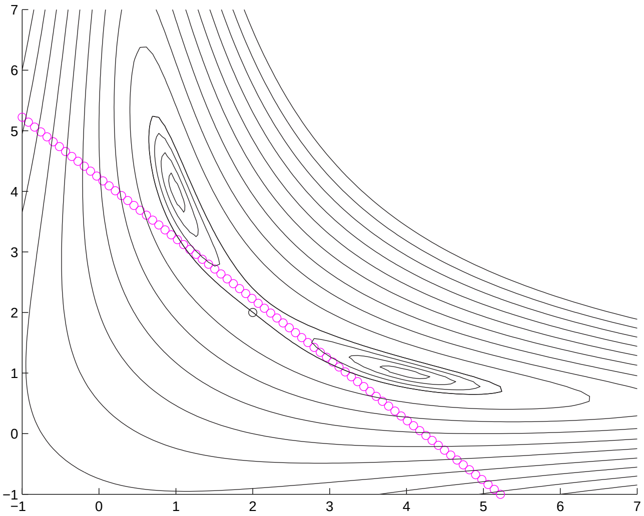


Figure 2.3: Inexact Newton step: singular jacobian matrix case 2

(A1) There exists a point $x_* \in \mathbb{R}^n$ with $F(x_*) = 0$;

(A2) F is continuously differentiable in a neighborhood of x_*;

(A3) $F'(x_*)$ is nonsingular.

The local convergence theorem can be formulated as follows.

Theorem 2.6 (Theorem 2.3 in [25]) Assume that $\eta_k < t < 1$ and (A1)-(A3) hold. There exists $\epsilon > 0$ such that, if $\|x_0 - x_*\| < \epsilon$, then the sequence of inexact Newton iterates $\{x_k\}$ converges to x_*.
Moreover, the convergence rate is linear, that is

$$\|x_{k+1} - x_*\|_* \leq t\|x_k - x_*\|_*$$

where $\|y\|_* \equiv \|F'(x_*)y\|$.

Remarks. The proof of the theorem is carried out exploiting the assumption (A2) which guarantees the smoothness of F and employing the two following lemmas.

Lemma 2.1 [61, §2.3.3] Assume that the matrix $F'(x)$ is invertible. Then, for any $\epsilon > 0$ there exists $\delta > 0$ such that $F'(x)$ is invertible and

$$\|F'(x)^{-1} - F'(y)^{-1}\| < \epsilon,$$

for all $y \in N_\delta(x)$.

Lemma 2.2 [61, §3.1.5] For any x and $\epsilon > 0$, there exists $\delta > 0$ such that

$$\|F(z) - F(y) - F'(y)(z-y)\| \le \epsilon \|z-y\|,$$

for all $z, y \in N_\delta(x)$.

About the rate of convergence of the method, the sequence of the forcing terms η_k plays an important role: indeed, from the proof of the previous theorem, the linear rate is due to the bound $\eta_k < 1$. The following theorem shows that, choosing the forcing sequence in an appropriate way, the convergence rate can be improved.

Theorem 2.7 Let $\{x_k\}$ the sequence generated by the inexact Newton method. Assume that (A1)–(A3) hold and that the sequence $\{x_k\}$ converges to x_*. Then $\{x_k\}$ converges to x_* with the same rate of convergence as the sequence $\{\|F(x_k)\|\}$ converges to 0.
Furthermore the rate of convergence is superlinear if $\eta_k \to 0$ and it is quadratic if $\eta_k = \mathcal{O}(\|F(x_k\|)$.

Proof. Let $\beta = \|F'(x_*)^{-1}\|$ and $\alpha = \max[\|F'(x_*)\| + \frac{1}{2\beta}, 2\beta]$. From Taylor's expansion it follows that

$$F(x_k) = F(x_*) + F'(x_*)(x_k - x_*) + \mathcal{O}(\|x_k - x_*\|^2),$$

thus

$$\|F(x_k) - F(x_*) - F'(x_*)(x_k - x_*)\| \le \mathcal{O}(\|x_k - x_*\|^2).$$

Since $\{x_k\}$ converges to x_*, for k sufficiently large the following relation holds (see Lemma 2.2):

$$\|F(x_k) - F(x_*) - F'(x_*)(x_k - x_*)\| \le \frac{1}{2\beta}\|x_k - x_*\|.$$

Now, the value of $F(x_k)$ can be obtained as

$$F(x_k) = F'(x_*)(x_k - x_*) + [F(x_k) - F(x_*) - F'(x_*)(x_k - x_*)] \qquad (2.24)$$

and then

$$\begin{aligned} \|F(x_k)\| &\le \|F'(x_*)\|\|x_k - x_*\| + \|F(x_k) - F(x_*) - F'(x_*)(x_k - x_*)\| \\ &\le \left[\|F'(x_*)\| + \frac{1}{2\beta}\right]\|x_k - x_*\| \\ &\le \alpha\|x_k - x_*\| \qquad (2.25) \end{aligned}$$

On the other hand, (2.24) can be written also as

$$F(x_k) = F'(x_*)(x_k - x_*) - [F(x_*) - F(x_k) + F'(x_*)(x_k - x_*)]. \tag{2.26}$$

By multiplying (2.26) by $F'(x_*)^{-1}$ and taking norms, we can conclude that the following inequality holds:

$$\begin{aligned} \|F(x_k)\| &\geq \|F'(x_*)^{-1}\|^{-1}\|x_k - x_*\| - \|F(x_k) - F(x_*) - F'(x_*)(x_k - x_*)\| \\ &\geq \left[\|F'(x_*)^{-1}\|^{-1} - \frac{1}{2\beta}\right]\|x_k - x_*\| \\ &= \frac{1}{2\beta}\|x_k - x_*\| \\ &\geq \frac{1}{\alpha}\|x_k - x_*\|. \end{aligned} \tag{2.27}$$

Comparing the inequalities (2.25) and (2.27), it has been proved that

$$\frac{1}{\alpha}\|x_k - x_*\| \leq \|F(x_k)\| \leq \alpha\|x_k - x_*\| \tag{2.28}$$

(cfr. Lemma 3.1 in [25]). This is sufficient to conclude that

$$\frac{1}{\alpha} \leq \limsup_{k\to\infty} \frac{\|F(x_k)\|}{\|x_k - x_*\|} \leq \alpha$$

which means that $\{\|F(x_k)\|\}$ and $\{\|x_k - x_*\|\}$ converge to zero with the same rate of convergence.

Thus, it is possible to prove the last part of the theorem, either on the sequence $\|F(x_k)\|$, or on the sequence $\|x_k - x_*\|$.

Since $F'(x_*)$ is nonsingular, then there exists a positive number δ such that $F'(x)$ is nonsingular for all $x \in N_\delta(x_*)$.

Let L be the maximum value of $\|F'(x)^{-1}\|$ in the compact set $N_\delta(x_*)$. For k sufficiently large (such that $x_k \in N_\delta(x_*)$), from (2.21) and from

$$(x_{k+1} - x_k) = F'(x_k)^{-1}([F'(x_k)(x_{k+1} - x_k) + F(x_k)] - F(x_k))$$

it follows that

$$\begin{aligned} \|x_{k+1} - x_k\| &\leq \|F'(x_k)^{-1}\|[\|F'(x_k)(x_{k+1} - x_k) + F(x_k)\| + \|F(x_k)\| \\ &\leq 2L\|F(x_k)\|. \end{aligned}$$

By using the last expression together with the Taylor's expansion, it results that

$$\begin{aligned} F(x_{k+1}) &= F(x_k) + F'(x_k)(x_{k+1} - x_k) + \mathcal{O}(\|x_{k+1} - x_k\|^2) \\ &= F(x_k) + F'(x_k)(x_{k+1} - x_k) + \mathcal{O}(\|F(x_k)\|^2). \end{aligned}$$

Hence, from (2.21), the following inequality is proved

$$\|F(x_{k+1})\| \leq \eta_k \|F(x_k)\| + \mathcal{O}(\|F(x_k)\|^2). \tag{2.29}$$

Then, dividing (2.29) by $\|F(x_k)\|$, we obtain

$$\limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|} = \limsup_{k\to\infty} \eta_k,$$

and we have that, if $\eta_k \to 0$, then it follows the superlinear convergence. Furthermore, if $\eta_k = \mathcal{O}(\|F(x_k)\|)$, proceeding as before, we obtain

$$\limsup_{k\to\infty} \frac{\|F(x_{k+1})\|}{\|F(x_k)\|^2} = C$$

for some constant C, which guarantees the quadratic convergence rate of the sequences $\{\|F(x_k)\|\}$ and $\{\|x_k - x_*\|\}$. □

The next step is to globalize the method by introducing another condition in order to obtain algorithms with global convergence properties, under appropriate assumptions.

2.2.2 Global Inexact Newton methods

The "globalization" of the inexact Newton methods can be obtained by adding a step to its scheme: the first step consists in finding a direction which satisfies (2.21), then, in the second step, such direction is modifies in order to guarantee a sufficient progress toward the solution. The idea of "sufficient progress" toward the solution can be expressed by requiring a suitable decrease of $\|F(x)\|$ in the next iterate.

The general scheme for the global method can be written as follows:

```
Let $x_0 \in \mathbb{R}^n$ and $\beta \in (0,1)$ be given.
For $k = 0, 1, 2, \ldots$
    Find $\eta_k \in [0,1)$, $\lambda_k \in (0,1)$ and a vector $s_k$ that satisfy
```

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta_k \|F(x_k)\| \tag{2.30}$$

and

$$\|F(x_k + s_k)\| \leq \lambda_k \|F(x_k)\|. \tag{2.31}$$

```
    Set $x_{k+1} = x_k + s_k$.
```

The theoretical foundation for the convergence of the sequence $\{x_k\}$ generated by the previous algorithm has been developed in [33] under the assumptions (A2),(A3) and the following further hypothesis:

(A4) x_* is a limit point of the sequence $\{x_k\}$.

The first step is to analyze when the algorithm *breaks down*, that is when, at the iterate k, it is impossible to compute the next iterate x_{k+1} which satisfies (2.30) and (2.31). The next lemma shows that, if at the k-th iterate there exists an inexact Newton step, then the sequence does not break down. Thus, the possibility to construct the sequence depends only on the first requirement, the condition (2.30).

Lemma 2.3 (Lemma 3.1 in [33]) Let x and $\beta \in (0, 1)$ be given and assume that there exists a vector $\bar{s}$ and a scalar $\bar{\eta} < 1$ that satisfy

$$\|F'(x)\bar{s} + F(x)\| \leq \bar{\eta}\|F(x)\|.$$

Then, there exist $\eta_{min} \in [0, 1)$ such that

$$\|F'(x)s + F(x)\| \leq \eta\|F(x)\| \text{ and } \|F(x + s)\| \leq \lambda\|F(x)\|,$$

where $s = \frac{1-\eta}{1-\bar{\eta}}\bar{s}$, $\lambda = 1 - \beta(1 - \eta)$ for any $\eta \in [\eta_{min}, 1)$.

The previous lemma also shows that there exist some relations between η_k and λ_k, which represent the rates of the reduction of the residual $\|F'(x_k)s + F(x_k)\|$ and of $\|F(x_k)\|$ respectively at two successive iterates.
We also report the following theorem, which is crucial for the convergence proof.

Theorem 2.8 (Theorem 3.3 in [33]) Let $\{x_k\}$ be a sequence such that

$$\lim_{k\to\infty} F(x_k) = 0. \tag{2.32}$$

and for each iteration k the following conditions hold:

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta\|F(x_k)\|,$$

$$\|F(x_{k+1})\| \leq \|F(x_k)\|,$$

where $s_k = x_{k+1} - x_k$ and $\eta < 1$.
If x_* is a limit point of $\{x_k\}$, then $F(x_*) = 0$ and if $F'(x_*)$ is nonsingular, then the sequence $\{x_k\}$ converges to x_*.

The proof of the previous theorem can be also found in [63]. It can be observed that Theorem 2.8 is proved under the hypothesis (2.32). A sufficient condition that guarantees (2.32) can be obtained by requiring an appropriate reduction of the value of $\|F(x)\|$ at each iteration. More precisely, if

$$\|F(x_{k+1})\| \leq \lambda_k \|F(x_k)\| \tag{2.33}$$

where $0 < \lambda_k \leq \lambda < 1$, then (2.32) holds (see Theorem 6.7 in [63]).
Referring to Lemma 2.3, we can found an analogous necessary condition in terms of the forcing parameters η_k: assuming that $\lambda_k = 1 - \beta(1 - \eta_k)$ for a fixed $\beta \in (0, 1)$, the following inequality can be obtained:

$$\begin{aligned} \|F(x_k)\| &\leq \|F(x_0)\| \prod_{0 \leq j < k} [1 - \beta(1 - \eta_j)] \\ &\leq \|F(x_0)\| \exp\left[-\beta \sum_{0 \leq j < k} (1 - \eta_j)\right]. \end{aligned} \tag{2.34}$$

If the series $\sum_{k \geq 0}(1 - \eta_k)$ is divergent, then condition (2.32) holds and, since $(1 - \eta_k) > 0$, then a necessary and sufficient condition for the divergence of the series is

$$\lim_{k \to \infty} (1 - \eta_k) \neq 0. \tag{2.35}$$

This is the theoretical background for every sequence with the properties (2.30) and (2.31); in the next sections, different algorithms are presented in this framework. All of them provide a method to determine the sequences η_k and λ_k such that (2.32) or (2.35) holds, so that the convergence proof is obtained by applying Theorem 2.8.

2.2.3 Line–Search Inexact Newton methods

In this section we consider the class of inexact Newton methods in which a line–search procedure is employed in order to find the parameter λ_k satisfying (2.31). We will describe a backtracking algorithm and we will give the convergence proof by showing that a relation of the type of (2.35) holds. Following Lemma 2.3, the algorithm is composed of two steps: the first one is to find an inexact Newton step $\bar{s}$ at some level $\bar{\eta}_k$; the second one is to find a new forcing term η_k by increasing $\bar{\eta}_k$ until (2.30) and (2.31) are satisfied with

$$\lambda_k = 1 - \beta(1 - \eta_k)$$

and

$$s_k = \frac{1 - \eta_k}{1 - \bar{\eta}_k} \bar{s}_k. \tag{2.36}$$

The equality (2.36) means that an increase of the forcing parameter corresponds to a reduction of the inexact Newton step: by introducing a damping parameter α for the step length, that is

$$\alpha_k = \frac{1-\eta_k}{1-\bar{\eta}_k}$$

we obtain

$$\eta_k = 1 - \alpha_k(1-\bar{\eta}_k). \tag{2.37}$$

Our choice is to express the algorithm in terms of the damping parameter instead of the forcing term, tacking into account the relation (2.37). Hence the two–step (predictor–corrector) algorithm with backtracking strategy can be written as follows:

Algorithm 2.1

```
Set $x_0 \in \mathbb{R}^n$, $\beta \in (0,1)$, $0 < \theta_{min} < \theta_{max} < 1$, $\eta_{max} \in (0,1)$, $k = 0$.
For $k = 0, 1, 2, ...$
    Determine $\bar{\eta}_k \in [0, \eta_{max}]$ and $\bar{s}_k$ that satisfy
        $\|F'(x_k)\bar{s}_k + F(x_k)\| \leq \bar{\eta}_k \|F(x_k)\|.$
    Set $\alpha_k = 1$.
    While $\|F(x_k + \alpha_k \bar{s}_k)\| > (1 - \alpha_k \beta(1-\bar{\eta}_k))\|F(x_k)\|$
        Choose $\theta \in [\theta_{min}, \theta_{max}]$
        Set $\alpha_k = \theta\alpha_k$.
    Set $x_{k+1} = x_k + \alpha_k \bar{s}_k$
```

Furthermore, the relation (2.35) can be also translated in terms of the damping parameter α_k, by means of (2.37), in the following way:

$$\begin{array}{rcl} \lim_{k\to\infty}(1-\eta_k) & = & \lim_{k\to\infty} 1 - (1-\alpha_k(1-\bar{\eta}_k)) \\ & = & \lim_{k\to\infty} \alpha(1-\bar{\eta}_k). \end{array} \tag{2.38}$$

Since $\bar{\eta}_k \leq \eta_{max} < 1$, then (2.35) is equivalent to

$$\lim_{k\to\infty} \alpha_k > 0 \tag{2.39}$$

Condition (2.39) means that there exists a positive number τ such that $\alpha_k > \tau$ for infinitely many k. The following theorem shows that (2.39) holds, and then the convergence of the whole sequence is proved.

Theorem 2.9 Let $\{x_k\}$ the sequence generated by Algorithm 2.1 and assume that (A2)–(A4) hold. Then, there exists a positive number τ such that $\alpha_k > \tau$ for infinitely many k.

Proof. Denoting $\|F'(x_*)^{-1}\| = K$, we can find $\delta > 0$ such that

(i) $F'(x)^{-1}$ exists whenever $x \in N_\delta(x_*)$,

(ii) $\|F'(x)^{-1}\| \leq 2K \qquad \forall x \in N_\delta(x_*)$

(iii) $\|F(x) - F(y) - F'(y)(x-y)\| \leq \frac{(1-\beta)(1-\eta_{max})}{2K(1+\eta_{max})}\|y-x\| \qquad \forall x, y \in N_{2\delta}(x_*)$.

Since x_* is a limit point, there exist infinitely many k such that the following condition holds for any $x_k \in N_\delta(x_*)$:

$$\begin{aligned}\|\bar{s}_k\| &\leq \|F'(x_k)^{-1}\|(\|F'(x_k)\bar{s}_k + F(x_k)\| + \|F(x_k)\|) \\ &\leq 2K(1+\eta_{max})\|F(x_k)\|.\end{aligned} \tag{2.40}$$

Since $s_k = \alpha\bar{s}_k$, formula (2.40) can be written as

$$\|s_k\| \leq \Gamma\alpha\|F(x_k)\| \tag{2.41}$$

where $\Gamma = 2K(1+\eta_{max})$. Now we show that if $\alpha \leq \frac{\delta}{\Gamma\|F(x_k)\|}$, then $\|F(x_k + \alpha_k\bar{s}_k)\| < (1 - \alpha_k\beta(1-\bar{\eta}_k))\|F(x_k)\|$, thus the *while loop* in the Algorithm 2.1 terminates.
By means of condition (ii) and formulae (2.37) and (2.41) we can write

$$\begin{aligned}\|F(x_k+s_k)\| &\leq \|F(x_k) + F'(x_k)s_k\| + \|F(x_k+s_k) - F(x_k) - F'(x_k)s_k\| \\ &\leq \eta\|F(x_k)\| + \frac{(1-\beta)(1-\eta_{max})}{\Gamma}\|s_k\| \\ &\leq ((1-\alpha)(1-\bar{\eta}) + (1-\beta)\alpha(1-\bar{\eta}))\|F(x_k)\|,\end{aligned}$$

thus

$$\|F(x_k + \alpha\bar{s}_k)\| \leq (1 - \alpha\beta(1-\bar{\eta}))\|F(x_k)\|. \tag{2.42}$$

This inequality shows that the backtracking condition (2.42) is satisfied for $\alpha \leq \frac{\delta}{\Gamma\|F(x_k)\|}$ and since α is reduced at each step by a factor $\theta \leq \theta_{max} < 1$ the *while loop* terminates.
Suppose now that the *while loop* has been executed at least once, and denote α_k the final value (i.e. the value of α for which (2.42) is satisfied) and $\bar{\alpha_k}$ the previous one. At the last but one step the condition (2.42) is not satisfied, then we have

$$\bar{\alpha_k} > \frac{\delta}{\Gamma\|F(x_k)\|}$$

thus

$$\alpha_k = \theta\bar{\alpha_k} > \frac{\delta\theta_{min}}{\Gamma\|F(x_k)\|} \geq \frac{\delta\theta_{min}}{\Gamma\|F(x_0)\|}.$$

Hence the thesis of the theorem has been proved with $\tau = min(1, \frac{\delta\theta_{min}}{\Gamma\|F(x_0)\|})$.
□

2.2.4 Trust–Region methods

In this section a particular class of trust–region methods is considered and it is shown as the convergence theorem for this class of algorithms can be derived from the results in the previous section, in a slightly different way than in [33].
Consider a sequence $\{x_k\}$ such that $x_{k+1} = x_k + s_k$: recalling the standard notation (2.14) and (2.15) we have

$$\begin{aligned} ared(s_k) &\equiv \|F(x_k)\| - \|F(x_k + s_k)\| && (2.43) \\ pred(s_k) &\equiv \|F(x_k)\| - \|F'(x_k)s_k + F(x_k)\|. && (2.44) \end{aligned}$$

Theorem 2.8 can be adapted for any sequence $\{x_k\}$ such that $x_{k+1} = x_k + s_k$ and such that $pred(s_k) \geq 0$, observing that we can define an "a posteriori" forcing term in the following way:

$$\eta_k \equiv \begin{cases} \|F(x_k) + F'(x_k)s_k\|/\|F(x_k)\| & \|F(x_k)\| \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.45}$$

Using this notation, if we require that

$$ared(s_k) \geq \beta \cdot pred(s_k), \tag{2.46}$$

then (2.30) is satisfied with the equality and (2.31) is satisfied with $\lambda_k = (1 - \beta(1 - \eta_k))$. Indeed, if condition (2.46) holds, it follows that

$$\|F(x_k)\| - \|F(x_k + s_k)\| \geq \beta(\|F(x_k) + F'(x_k)s_k\|),$$

hence

$$\begin{aligned} \|F(x_k + s_k)\| &\leq \|F(x_k)\| - \beta(\|F(x_k)\| + \|F'(x_k)s_k + F(x_k)\|) \\ &= \|F(x_k)\| \left(1 - \beta\frac{\|F'(x_k)s_k + F(x_k)\|}{\|F(x_k)\|}\right) \\ &= (1 - \beta(1 - \eta_k)). \end{aligned}$$

In the following we will consider a particular trust–region scheme (see [33], section 4), and we will show the convergence employing Theorem 2.8.

Algorithm 2.2

```
Set $x_0 \in \mathbb{R}^n$, $0 < \beta < u < 1$, $0 < \theta_{min} < \theta_{max} < 1$, $\bar{\delta}_0 > 0$.
For $k = 0, 1, 2, ...$
    Set $\delta_k = \bar{\delta}_k$ and determine $s_k$ such that
        $s_k \in arg \min_{\|\bar{s}\| \le \delta_k} \|F'(x_k)\bar{s} + F(x_k)\|.$    (2.47)
    While $ared(s_k) < \beta \cdot pred(s_k)$
        Choose $\theta \in [\theta_{min}, \theta_{max}]$
        Set $\delta_k = \theta \delta_k$.
        Choose $s_k \in arg \min_{\|\bar{s}\| \le \delta_k} \|F'(x_k)\bar{s} + F(x_k)\|$.
    Set $x_{k+1} = x_k + s_k$
    If $ared(s_k) \ge u \cdot pred(s_k)$, choose $\bar{\delta}_{k+1} \ge \delta_k$, else choose $\bar{\delta}_{k+1} \ge \theta_{min}\delta_k$.
```

The aim of the following propositions is to prove that $\lim_{k\to\infty} \|F(x_k)\| = 0$, hence the convergence of the sequence by means of Theorem 2.8. The procedure used here is quite similar to the one presented in section 4 of [33] and it exploits the three following results, whose proofs are not reported here but they can be found in [33].

Lemma 2.4 Let $\{x_k\}$ be the sequence generated by Algorithm 2.2 and suppose that x_* is a limit point of $\{x_k\}$ such that

$$\|s_k\| \le \Gamma\{\|F(x_k)\| - \|F(x_k) + F'(x_k)s_k\|\} \tag{2.48}$$

for k sufficiently large. If $\{x_k\}$ converges to x_*, then $\liminf_{k\to\infty} \delta_k > 0$.

The previous lemma is a straightforward consequence of Lemma 4.1 in [33].

Lemma 2.5 (Lemma 4.2 in [33]) If x_* is a limit point of $\{x_k\}$ such that $F'(x_*)$ is nonsingular, then there exists a positive scalar Γ and $\epsilon > 0$ such that, for any $\delta > 0$,

$$s \in arg \min_{\|\bar{s}\| \le \delta} \|F'(x)\bar{s} + F(x)\| \tag{2.49}$$

satisfies

$$\|s\| \le \Gamma\{\|F(x)\| - \|F(x) + F'(x)s\|\} \tag{2.50}$$

for any $x \in N_\epsilon(x_*)$.

It can be noticed that inequality (2.50) is equivalent to

$$\|s\| \leq \Gamma pred(s)$$

where Γ is independent from x. Lemma 2.5 shows that (2.50) holds for $s = s_k$ when the iterates are sufficiently near to a limit point in which the jacobian matrix is nonsingular. The next lemma shows that to (2.50), where s is chosen as in (2.49), also holds in a neighborhood of a non stationary point of $\|F(x)\|$.
We recall that x_* is a stationary point of $\|F(x)\|$ if and only if

$$\|F(x_*)\| \leq \|F(x_*) + F'(x_*)s\|$$

for any $s \in \mathbb{R}^n$ (see for example [34, Proposition 2.1]).

Lemma 2.6 (Lemma 4.3 in [33]) If x_* is a nonstationary point of $\|F(x)\|$, then there exist a positive scalar Γ, $\epsilon_* > 0$ and $\delta_* > 0$ such that (2.50) holds for any s chosen as in (2.49) for $x \in N_{\epsilon_*}(x_*)$, $\delta < \delta_*$.

Theorem 2.10 Assume that Algorithm 2.2 does not break down. Then every limit point of $\{x_k\}$ are stationary point of $\|F\|$. If x_* is a limit point of $\{x_k\}$ such that $F'(x_*)$ is nonsingular, then $F(x_*) = 0$ and x_k converges to x_*. Furthermore, for k sufficiently large, the full Newton step is taken.

Proof. Suppose that x_* is a limit point of $\{x_k\}$ that is nonsingular for $\|F(x)\|$. We want to show by contradiction that this implies

$$\lim_{k\to\infty} \delta_k = 0. \tag{2.51}$$

If (2.51) does not hold, then there exists a positive number $\delta > 0$ and a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ converging to x_* such that $\delta_{k_j} > \delta$ for j sufficiently large. Then we have

$$\begin{aligned}
0 &= \lim_{j\to\infty} \left\{\|F(x_{k_j})\| - \|F(x_{k_{j+1}})\|\right\} \\
&\geq \lim_{j\to\infty} \left\{\|F(x_{k_j})\| - \|F(x_{k_j+1})\|\right\} \\
&= \lim_{j\to\infty} ared(s_{k_j}) \\
&\geq \beta \lim_{j\to\infty} pred(s_{k_j}) \\
&\geq \beta \lim_{j\to\infty} \left\{ \|F(x_{k_j})\| - \min_{\|s\|\leq\delta_{k_j}} \|F(x_{k_j}) + F'(x_{k_j})s\| \right\} \\
&\geq \beta \left\{\|F(x_*)\| - \|F(x_*) + F'(x_*)s_*\|\right\} \\
&> 0
\end{aligned} \tag{2.52}$$

Thus (2.51) holds. This implies that

$$\lim_{k\to\infty} \|s_k\| = 0,$$

hence the sequence $\{x_k\}$ satisfies the Cauchy condition, and we can conclude that it converges to x_*. Therefore, the hypotheses of Lemma 2.4 are satisfied, since (2.50) holds with Γ defined in Lemma 2.6, and we found a contradiction: indeed Lemma 2.4 claims that the sequence δ_k is uniformly bounded away from zero, while assuming x_* nonstationary yields to (2.51). Thus, x_* is a stationary point of $\|F(x)\|$ and if $F'(x_*)$ is nonsingular, than we must have $F(x_*) = 0$. For the continuity of F, it follows that $\lim_{k\to\infty} F(x_k) = 0$ and can employ Theorem 2.8 to conclude that $\{x_k\}$ converges to x_*.
Moreover, when we are close to the solution, $F'(x_k)$ is nonsingular, and the norm of the Newton step is given by $\|s_k^N\| = \|F'(x)^{-1}F(x)\|$. Approaching to the solution, since $F(x)$ tends to zero, we have $\|s_k^N\| \le \tau < \delta_k$, where τ is defined as in Lemma 2.4. □

2.2.5 Nonmonotone Inexact Newton methods

In this section we present a nonmonotone variant of the inexact Newton method, in which the tolerance for the residual of the Newton equation and for the norm of F in the new point does not depend on the value of $\|F\|$ in the previous iterate, but on the maximum of the last N values, where N is a fixed positive integer. First of all, it is useful to introduce the following notations. Given $N \in \mathbb{N}$ and a sequence $\{x_k\}$, we denote by $x_{\ell(k)}$ the element with the following property

$$\|F(x_{\ell(k)})\| = \max_{0\le j\le \min(N,k)} \|F(x_{k-j})\|. \tag{2.53}$$

Note that we have $k - min(N,k) \le \ell(k) \le k$.
The modified scheme of the inexact Newton methodcan be written as follows:

```
Let $x_0 \in \mathbb{R}^n$ and $\beta \in (0,1)$ be given.
For $k = 0, 1, 2, \dots$
    Find some $\eta_k \in [0,1)$ and a vector $s_k$ that satisfy
```

$$\|F(x_k) + F'(x_k)s_k\| \le \eta_k \|F(x_{\ell(k)})\| \tag{2.54}$$

and

$$\|F(x_k + s_k)\| \le \lambda_k \|F(x_{\ell(k)})\|. \tag{2.55}$$

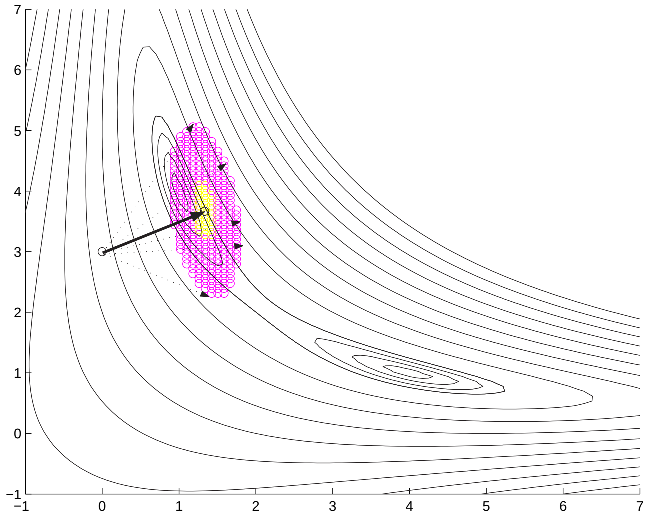


Figure 2.4: Inexact Newton step: singular jacobian matrix case 1

Set $x_{k+1} = x_k + s_k$.
According to (2.30), we define the vector s_k satisfying (2.54) *nonmonotone inexact Newton step* at the level η_k. Note that the sequence $\{\|F(x_k)\|\}$ satisfying (2.54) and (2.55) is nonmonotone, but $\{\|F(x_{\ell(k)})\|\}$ is a monotone nonincreasing subsequence of it. Furthermore, the nonmonotone step is not a descent direction for the merit function defined in (2.23). Referring to the example (2.22), the situation is depicted in Figure 2.4: the smaller gray region contains the points allowed by the monotone rule (2.30), while the points lying in the larger colored region satisfy the nonmonotone condition (2.54). Thus, the set of the nonmonotone inexact Newton steps is larger than in the monotone case. This could be an advantage in the choice of the direction s_k and it means also that the Newton equation can be solved with a coarser accuracy. Furthermore, the second condition (2.55) is less restrictive than (2.31), allowing larger step sizes. For the general scheme, an analogous property as in the monotone case holds: indeed, a sequence which satisfies the conditions (2.54) and (2.55) breaks down only if at the step k it does not exist a nonmonotone inexact Newton step, as stated in the following lemma.

Lemma 2.7 Let $\{x_k\}$ be a sequence such that (2.54) and (2.55) hold for any k. Suppose that there exist $\bar{\eta} \in [0,1)$, $\bar{s}$ satisfying

$$\|F(x_k) + F'(x_k)\bar{s}\| \leq \bar{\eta}\|F(x_{\ell(k)})\|.$$

Then, there exist η, λ and a vector s such that

$$\|F(x_k) + F'(x_k)s\| \leq \eta\|F(x_{\ell(k)})\| \tag{2.56}$$

$$\|F(x_k + s)\| \leq \lambda\|F(x_{\ell(k)})\| \tag{2.57}$$

hold for any where $\eta \in [\bar{\eta}, 1)$, and $\lambda < 1$.

Proof. Let $s = \alpha\bar{s}$. Then we have

$$\begin{array}{rcl}\|F(x_k) + F'(x_k)s\| & = & \|F(x_k) - \alpha F(x_k) + \alpha F(x_k) + \alpha F'(x_k)\bar{s}\| \\ & \leq & (1-\alpha)\|F(x_k)\| + \alpha\|F(x_k) + F'(x_k)\bar{s}\| \\ & \leq & (1-\alpha)\|F(x_{\ell(k)})\| + \alpha\bar{\eta}\|F(x_{\ell(k)})\| \\ & = & \eta\|F(x_{\ell(k)})\|,\end{array}$$

so (2.56) is proved. Now let

$$\varepsilon = \frac{(1-\beta)(1-\bar{\eta})}{\|\bar{s}\|}\|F(x_{\ell(k)})\|, \tag{2.58}$$

where $\beta < 1$ and let $\delta > 0$ be sufficiently small (see Lemma 2.2) such that

$$\|F(x_k + s) - F(x_k) - F'(x_k)s\| \leq \varepsilon\|s\| \tag{2.59}$$

whenever $\|s\| < \delta$. Choosing $\alpha_{max} = \min(1, \frac{\delta}{\|\bar{s}\|})$, for any $\alpha \in (0, \alpha_{max}]$ we have $\|s\| < \delta$ and then, using (2.58) and (2.59), we obtain the following inequality

$$\begin{array}{rcl}\|F(x_k + s)\| & \leq & \|F(x_k + s) - F(x_k) - F'(x_k)s\| + \|F(x_k) + F'(x_k)s\| \\ & \leq & \varepsilon\alpha\|\bar{s}\| + \eta\|F(x_{\ell(k)})\| \\ & = & ((1-\beta)(1-\bar{\eta})\alpha + (1-\alpha(1-\bar{\eta})))\|F(x_{\ell(k)}\| \\ & = & (1-\beta\alpha(1-\bar{\eta}))\|F(x_{\ell(k)})\|\end{array}$$

that completes the proof with $\lambda = 1 - \beta\alpha(1-\bar{\eta})$ or, expressing α in terms of η,

$$\lambda = 1 - \beta(1-\eta). \tag{2.60}$$

Furthermore, observing that $\eta > \bar{\eta}$ we have

$$\|F(x_k + s)\| \leq (1 - \beta\alpha(1-\eta))\|F(x_{\ell(k)})\|,$$

thus we could also choose $\lambda = 1 - \beta\alpha(1-\eta)$. □

A further result, analogous to the Theorem 2.8, can be proved also in the nonmonotone case.

Theorem 2.11 Let $\{x_k\}$ a sequence such that $\lim_{k\to\infty} F(x_k) = 0$ and for each k the following conditions hold:

$$\|F(x_k) + F'(x_k)s_k\| \leq \eta \|F(x_{\ell(k)})\|, \tag{2.61}$$

$$\|F(x_{k+1})\| \leq \|F(x_{\ell(k)})\|, \tag{2.62}$$

where $s_k = x_{k+1} - x_k$ and $\eta < 1$. If x_* is a limit point of $\{x_k\}$, then $F(x_*) = 0$ and if $F'(x_*)$ is nonsingular, then the sequence $\{x_k\}$ converges to x_*.

Proof. If x_* is a limit point of the sequence $\{x_k\}$, there exists a subsequence $\{x_{k_j}\}$ of $\{x_k\}$ convergent to x_*. By the continuity of F, we obtain

$$F(x_*) = F\left(\lim_{j\to\infty} x_{k_j}\right) = \lim_{j\to\infty} F(x_{k_j}) = 0.$$

Furthermore, since $\{x_{\ell(k)}\}$ is a subsequence of $\{x_k\}$, also the sequence $\{F(x_{\ell(k)})\}$ converges to zero when k diverges. Denote $K = \|F'(x_*)^{-1}\|$ and $\delta > 0$ be sufficiently small such that $F'(y)^{-1}$ exists whenever $y \in N_\delta(x_*)$; thus we can suppose

$$\|F'(y)^{-1}\| \leq 2K, \tag{2.63}$$

$$\|F(y) - F(x_*) - F'(x_*)(y - x_*)\| \leq \frac{1}{2K}\|y - x_*\|.$$

Then for any $y \in N_\delta(x_*)$ we have

$$\begin{array}{rcl} \|F(y)\| & = & \|F'(x_*)(y - x_*) + F(y) - F(x_*) - F'(x_*)(y - x_*)\| \\ & \geq & \|F'(x_*)(y - x_*)\| - \|F(y) - F(x_*) - F'(x_*)(y - x_*)\| \\ & \geq & \frac{1}{K}\|y - x_*\| - \frac{1}{2K}\|y - x_*\| \\ & = & \frac{1}{2K}\|y - x_*\|. \end{array}$$

Then

$$\|y - x_*\| \leq 2K\|F(y)\| \tag{2.64}$$

holds for any $y \in N_\delta(x_*)$.

Now, let $\epsilon \in (0, \frac{\delta}{4})$ and since x_* is a limit point of $\{x_k\}$, there exists a k sufficiently large that

$$x_k \in N_{\frac{\delta}{2}}(x_*)$$

and

$$x_{\ell(k)} \in S_\epsilon \equiv \left\{y : \|F(y)\| < \frac{\epsilon}{K(1+\eta)}\right\}.$$

Note that since $x_{\ell(k)} \in S_\epsilon$ then also $x_{k+1} \in S_\epsilon$ because $\|F(x_{k+1})\| \leq \|F(x_{\ell(k)})\|$.
For the direction s_k, by (2.61), (2.62) and (2.63), the following inequality holds:

$$\begin{aligned} \|s_k\| &\leq \|F'(x_k)^{-1}\|(\|F(x_k)\| + \|F(x_k) + F'(x_k)s_k\|) \\ &\leq 2K(\|F(x_{\ell(k)})\| + \eta\|F(x_{\ell(k)})\|) \\ &= 2K(1+\eta)\|F(x_{\ell(k)})\| < 2\epsilon < \tfrac{\delta}{2}. \end{aligned}$$

Since $s_k = x_{k+1} - x_k$, the previous inequality implies $\|x_{k+1} - x_*\| < \delta$ and from (2.64) we obtain

$$\|x_{k+1} - x_*\| \leq 2K\|F(x_{k+1})\| < 2K\frac{\epsilon}{K(1+\eta)} < \frac{\delta}{2}$$

that implies $x_{k+1} \in N_{\frac{\delta}{2}}(x_*)$. Therefore $x_{\ell(k+1)} \in S_\epsilon$, since $\|F(x_{\ell(k+1)})\| \leq \|F(x_{\ell(k)})\|$. It follows that, for any j sufficiently large, $x_j \in N_\delta(x_*)$, and from (2.64)

$$\|x_j - x_*\| \leq 2K\|F(x_j)\|.$$

Since $F(x_j)$ converges to 0 we can conclude that x_j converges to x_*. □

The previous one is a convergence theorem under the hypothesis A4 and the assumption $\lim_{k\to\infty} F(x_k) = 0$. It can be observed, taking into account (2.60) and making the same remarks as before, that the following inequality holds:

$$\begin{aligned} \|F(x_k)\| &\leq \|F(x_0)\| \prod_{0\leq j<k} [1 - \beta(1 - \eta_{\bar{\ell}(k_j)})] \\ &\leq \|F(x_0)\| \exp\left[-\beta \sum_{0\leq j<k} (1 - \eta_{\bar{\ell}(k_j)})\right]. \end{aligned}$$

Here $\eta_{\bar{\ell}(k_j)}$ indicates the subsequence of $\eta_{\ell(k)}$ such that

$$\bar{\ell}(k_j) = \underbrace{\ell\,(\ell\,(\ell\,(\ \ldots\ell}_{j}\ \ (\ k)))) \quad .$$

A sufficient condition for the divergence of the series is

$$\lim_{k\to\infty} \left(1 - \eta_{\ell(k)}\right) \neq 0$$

which, in terms of α becomes

$$\lim_{k\to\infty} \alpha_{\ell(k)} \neq 0. \qquad (2.65)$$

In the following, we will restrict our attention to a particular nonmonotone inexact Newton algorithm, with a line–search procedure. For such algorithm, the convergence proof is given by showing that (2.65) holds, from which it follows

$$\lim_{k\to\infty} \|F(x_k)\| = 0 \qquad (2.66)$$

and the convergence of $\{x_k\}$ to x_* by theorem 2.11.

Algorithm 2.3

```
1. Set $x_0 \in \mathbb{R}^n$, $\beta \in (0,1)$, $0 < \theta_{min} < \theta_{max} < 1$, $\eta_{max} \in (0,1)$, $k = 0$.
2. Determine $\bar\eta_k \in [0, \eta_{max}]$, $\bar s_k$ that satisfy
       $\|F(x_k) + F'(x_k)\bar s_k\| \le \bar\eta_k \|F(x_{\ell(k)})\|.$
   Set $\alpha_k = 1$.
3. While $\|F(x_k + \alpha_k \bar s_k)\| > (1 - \alpha_k\beta(1-\bar\eta_k))\|F(x_{\ell(k)})\|$
    3a. Choose $\theta \in [\theta_{min}, \theta_{max}]$;
    3b. Set $\alpha_k = \theta\alpha_k$.
4. Set $x_{k+1} = x_k + \alpha_k \bar s_k$.
   $k = k + 1$
   Go to Step 2.
```

From the proof of Lemma 2.7 it follows that the nonmonotone backtracking condition

$$\|F(x_k + \alpha \bar s_k)\| < (1 - \alpha\beta(1-\bar\eta_k))\|F(x_{\ell(k)})\| \qquad (2.67)$$

is satisfied for $\alpha < \alpha_{max}$, where α_{max} depends on k.
Indeed, since the value of α_k is reduced by a factor $\theta < \theta_{max} < 1$ at the step 3a, then there exists a positive integer p such that $(\theta_{max})^p < \alpha_{max}$, thus *the while loop* terminates at most after p steps. For the nonmonotone algorithm it is also possible to prove the same result stated in Theorem 2.9 under the same assumptions.

Theorem 2.12 Let $\{x_k\}$ be a sequence generated by the algorithm (**??**) and assume that (A2)–(A4) hold. Then there exists a positive number τ such that $\alpha_k > \tau$ for infinitely many k.

Proof. The proof of Theorem 2.12 can be easily derived from the proof of Theorem 2.9 given in the previous section for the algorithm 2.1, since for the nonmonotone adaptive tolerance we have $\|F(x_k)\| \leq \|F(x_{\ell(k)})\|$.
For sake of completeness we report the whole proof, which can be found also in [12].
Denoting $\|F'(x_*)^{-1}\| = K$, we can find $\delta > 0$ such that

(i) $F'(x)^{-1}$ exists whenever $x \in N_\delta(x_*)$,

(ii) $\|F'(x)^{-1}\| \leq 2K \qquad \forall x \in N_\delta(x_*)$

(iii) $\|F(x) - F(y) - F'(y)(x-y)\| \leq \frac{(1-\beta)(1-\eta_{max})}{2K(1+\eta_{max})}\|y-x\| \qquad \forall x, y \in N_{2\delta}(x_*)$.

Since x_* is a limit point, there exist infinitely many k such that $x_k \in N_\delta(x_*)$ for which the following condition holds:

$$\begin{aligned} \|\bar{s}_k\| &\leq \|F'(x_k)^{-1}\|(\|F'(x_k)\bar{s}_k + F(x_k)\| + \|F(x_k)\|) \\ &\leq 2K(1+\eta_{max})\|F(x_{\ell(k)})\|. \end{aligned} \tag{2.68}$$

Since $s_k = \alpha\bar{s}_k$, formula (2.68) can be written as

$$\|s_k\| \leq \Gamma\alpha\|F(x_{\ell(k)})\| \tag{2.69}$$

where $\Gamma = 2K(1+\eta_{max})$.
Now we show that if $\alpha \leq \frac{\delta}{\Gamma\|F(x_{\ell(k)})\|}$, then the *while loop* terminates. We can write by means of condition (ii), Lemma 2.7and formula (2.69)

$$\begin{aligned} \|F(x_k+s_k)\| &\leq \|F(x_k) + F'(x_k)s_k\| + \|F(x_k+s_k) - F(x_k) - F'(x_k)s_k\| \\ &\leq \eta\|F(x_{\ell(k)})\| + \tfrac{(1-\beta)(1-\eta_{max})}{\Gamma}\|s_k\| \\ &\leq ((1-\alpha)(1-\bar{\eta}) + (1-\beta)\alpha(1-\bar{\eta}))\|F(x_{\ell(k)})\|. \end{aligned}$$

Thus

$$\|F(x_k+\alpha\bar{s}_k)\| \leq (1-\alpha\beta(1-\bar{\eta}))\|F(x_{\ell(k)})\|$$

This inequality shows that the backtracking condition (2.67) is satisfied for $\alpha \leq \frac{\delta}{\Gamma\|F(x_{\ell(k)})\|}$ and since α is reduced at every step by a factor $\theta \leq \theta_{max} < 1$ the *while loop* terminates. Suppose now that the *while loop* has been executed at least once, let denote α_k the final value (i.e. the value of α for which (2.67) is satisfied) and $\bar{\alpha_k}$ the previous one. At the penultimate step the condition (2.67) is not satisfied, so necessarily we have

$$\bar{\alpha_k} > \frac{\delta}{\Gamma\|F(x_{\ell(k)})\|}$$

and so

$$\alpha_k = \theta\bar{\alpha_k} > \frac{\delta\theta_{min}}{\Gamma\|F(x_{\ell(k)})\|} \geq \frac{\delta\theta_{min}}{\Gamma\|F(x_0)\|}.$$

Hence the thesis has been proved with $\tau = min(1, \frac{\delta\theta_{min}}{\Gamma\|F(x_0)\|})$. □

From the proof of the previous theorem, it is useful to put in evidence that the property $\alpha_{k_j} > \tau$ holds in particular if $\{x_{k_j}\}$ is a subsequence of $\{x_k\}$ converging to x_*, and the following corollary can be derived.

Corollary 2.1 Suppose that Algorithm 2.3 does not break down. If x_* is a limit point of the sequence $\{x_k\}$ such that $F'(x_*)$ is nonsingular and $\{x_{k_j}\}$ is a subsequence converging to x_*, then the sequence $\{\alpha_{k_j}\}$ is bounded away from zero.

Exploiting Corollary 2.1, it can be proved that (2.66) holds, from which it follows the convergence of the sequence to x_*. This result is proved employing a technique similar to the one used for the convergence theorem in section 3 of [44].

Theorem 2.13 Suppose that Algorithm 2.3 does not break down and that the norm of inexact Newton step is bounded for every k by a positive constant M

$$\|\bar{s}_k\| \leq M. \tag{2.70}$$

Assume also that one of the two following properties holds:

$$F \text{ is Lipschitz continuous;} \tag{2.71}$$

$$\text{the set } \Omega(0) = \{x \in \mathbb{R}^n : \|F(x)\| \leq \|F(x_0)\|\} \text{ is compact.} \tag{2.72}$$

If x_* is a limit point of the sequence $\{x_k\}$ such that $F'(x_*)$ is invertible then $F(x_*) = 0$ and $\{x_k\}$ converges to x_* when k diverges.

Proof. Since $\|F(x_{\ell(k)})\|$ is a monotone nonincreasing, bounded sequence, then there exists $L \geq 0$ such that

$$L = \lim_{k\to\infty} \|F(x_{\ell(k)})\|.$$

Thus, writing the backtracking condition (2.42) for the iterate $\ell(k)$, we obtain

$$\|F(x_{\ell(k)})\| \leq (1 - \alpha_{\ell(k)-1}\beta(1 - \bar{\eta}_{\ell(k)-1}))\|F(x_{\ell(\ell(k)-1)})\|. \tag{2.73}$$

When k diverges, we can write

$$L \leq L - L \cdot \lim_{k\to\infty} \alpha_{\ell(k)-1}\beta(1 - \bar{\eta}_{\ell(k)-1}). \tag{2.74}$$

Since β is a constant and $1 - \bar{\eta}_j \geq 1 - \eta_{max} > 0$ for any j, (2.74) yields

$$L \cdot \lim_{k\to\infty} \alpha_{\ell(k)-1} \leq 0$$

that implies

$$L = 0$$

or

$$\lim_{k\to\infty} \alpha_{\ell(k)-1} = 0. \tag{2.75}$$

Suppose that $L \neq 0$, so that (2.75) holds. Let $\hat{\ell}(k) = \ell(k + N + 1)$ so that $\hat{\ell}(k) > k$ and we show by induction that for any $j \geq 0$ we have

$$\lim_{k\to\infty} \alpha_{\hat{\ell}(k)-j} = 0 \tag{2.76}$$

and

$$\lim_{k\to\infty} \|F(x_{\hat{\ell}(k)-j})\| = L. \tag{2.77}$$

For $j = 1$, since $\{\alpha_{\hat{\ell}(k)-1}\}$ is a subsequence of $\{\alpha_{\ell(k)-1}\}$, (2.75) implies (2.76). From (2.70) we also obtain

$$\lim_{k\to\infty} \|x_{\hat{\ell}(k)} - x_{\hat{\ell}(k)-1}\| = 0. \tag{2.78}$$

If (2.71) holds, from $|\|F(x)\| - \|F(y)\|| \leq \|F(x) - F(y)\|$ and (2.78) we obtain

$$\lim_{k\to\infty} \|F(x_{\hat{\ell}(k)-1})\| = L. \tag{2.79}$$

If, instead of (2.71), condition (2.72) holds, then, exploiting the uniform continuity of F in $\Omega(0)$, we can again derive (2.79).
Assume now that (2.76) and (2.77) hold for a given j, we have

$$\|F(x_{\ell(k)-j})\| \leq (1 - \alpha_{\ell(k)-(j+1)}\beta(1 - \eta_{\ell(k)-(j+1)}))\|F(x_{\ell(\ell(k)-(j+1))})\|.$$

Using the same arguments employed above, since $L > 0$, we obtain

$$\lim_{k\to\infty} \alpha_{\hat{\ell}(k)-(j+1)} = 0$$

and then

$$\lim_{k\to\infty} \|x_{\hat{\ell}(k)-j} - x_{\hat{\ell}(k)-(j+1)}\| = 0,$$

$$\lim_{k\to\infty} \|F(x_{\hat{\ell}(k)-(j+1)})\| = L.$$

Thus, we conclude that (2.76) and (2.77) hold for any $j \geq 1$. Now, for any k, we can write

$$\|x_{k+1} - x_{\hat{\ell}(k)}\| \leq \sum_{j=1}^{\hat{\ell}(k)-k-1} \alpha_{\hat{\ell}(k)-j} \|\bar{s}_{\hat{\ell}(k)-j}\|$$

so that, since we have $\hat{\ell}(k) - k - 1 \leq N$, we have

$$\lim_{k\to\infty} \|x_{k+1} - x_{\hat{\ell}(k)}\| = 0. \tag{2.80}$$

Furthermore, we have

$$\|x_{\hat{\ell}(k)} - x_*\| \leq \|x_{\hat{\ell}(k)} - x_{k+1}\| + \|x_{k+1} - x_*\| \tag{2.81}$$

Since x_* is a limit point of $\{x_{k+1}\}$ and (2.80) holds, (2.81) implies that x_* is a limit point for the sequence $\{x_{\hat{\ell}(k)}\}$. From (2.78) we conclude that x_* is also a limit point for the sequence $\{x_{\hat{\ell}(k)-1}\}$, which contradicts the Corollary 2.1. Indeed, there exists a $\tau > 0$ such that $\alpha_{\hat{\ell}(k)-1} > \tau$ for infinitely many k. Hence, we necessarily have $L = 0$, that implies

$$\lim_{k\to\infty} \|F(x_k)\| = 0.$$

Now, Theorem 2.11 completes the proof. □

We report also the following result.

Theorem 2.14 Under the hypothesis of Theorem 2.13 we have that the sequence $\{\|F(x_k)\|\}$ converges and

$$\lim_{k\to\infty} \|F(x_k)\| = \lim_{k\to\infty} \|F(x_{\ell(k)})\|.$$

Proof. If $\lim_{k\to\infty} \|F(x_{\ell(k)})\| = 0$, then $\lim_{k\to\infty} \|F(x_k)\| = 0$.
If $\lim_{k\to\infty} \|F(x_{\ell(k)})\| = L > 0$, using the same arguments in the first part of the proof of Theorem 2.13, we can conclude that (2.80) holds. If (2.71) or (2.72) holds, then $\lim_{k\to\infty} \|F(x_k)\| = L = \lim_{k\to\infty} \|F(x_{\ell(k)})\|$. □

About the local convergence rate, it is possible to prove the same result given in the previous section.

Theorem 2.15 Let $\{x_k\}$ be the sequence generated by Algorithm 2.3 and assume that the hypothesis of Theorem 2.13 hold. Then, the sequence $\{x_k\}$ locally converges to x_* with the same rate of convergence as the sequence $\|F(x_k)\|$ converges to 0.
Furthermore the rate of the local convergence is superlinear if $\eta_k \to 0$ and quadratic if $\eta_k = \mathcal{O}(\|F(x_k\|)$.

Proof. Under the assumptions of the theorem, the sequence $\{x_k\}$ converges to x_* and $\lim_{k\to\infty} \|F(x_k)\| = 0$. Thus, for the continuity of F, it follows that $F(x_*) = 0$. This is sufficient to conclude that (2.28) holds (see Theorem 2.7) for k sufficiently large, which implies that $\|x_k - x_*\|$ and $\|F(x_k)\|$ converge to zero with the same rate.
Furthermore, since $x_{\ell(k)}$ is a subsequence of x_k we have that

$$\mathcal{O}(\|F(x_k)\|) \geq \mathcal{O}(\|F(x_{\ell(k)})\|),$$

but we also have $\|F(x_k)\| \leq \|F(x_{\ell(k)})\|$, thus

$$\mathcal{O}(\|F(x_k)\|) = \mathcal{O}(\|F(x_{\ell(k)})\|). \tag{2.82}$$

Proceeding as in the proof of Theorem 2.7, it can be proved that the following inequality holds for each k sufficiently large:

$$\|F(x_{k+1})\| \leq \eta_k \|F(x_{\ell(k)})\| + \mathcal{O}(\|F(x_k)\|).$$

Tacking into account of (2.82), the statement of the theorem is proved. □

2.3 Newton methods and dynamic systems

This section is slightly apart from the rest of the chapter, but it is still related to the Newton method for the solution of a nonlinear system of equations. Our aim is to introduce a dynamic system associated to the problem (2.1) and to interpret the root–finding problem from this point of view.
First of all, we introduce some definitions and results in the dynamic systems

framework.
Let

$$\frac{dx(t)}{dt} = F(x(t)) \quad t \geq t_0 \tag{2.83}$$

be an autonomous [5] differential equation where F is a continuous Lipschitz function, $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$.

Definition 2.3 Let x_* be an *equilibrium* point of $F(x)$, namely $F(x_*) = 0$.

- It is said that the point x_* is *stable* for the system (2.83) if, for any given $\epsilon > 0$, there exists a $\delta > 0$ such that $\|x(t_0) - x_*\| \leq \delta$ implies $\|x(t; x(t_0)) - x_*\| \leq \epsilon$ for all $t \geq t_0$.
  Here, $x(t, x(t_0))$ denotes the solution of the differential equation (2.83) with the initial condition $x(t_0)$.

- It is said that the point x_* is *asymptotically stable* for the system (2.83) if it is stable and, moreover, there exists $\gamma > 0$ such that for $\|x(t_0) - x_*\| < \gamma$ one has

$$\lim_{t \to \infty} \|x(t; x(t_0)) - x_*\| = 0.$$

- Let $V(x)$ be a continuous and differentiable function such that $V(x)$ is always positive except in the equilibrium point x_*, in which it is equal to zero.
  We shall indicate with $\dot{V}(x)$ the derivative of $V(x)$ respect to the independent variable t on the solutions of the equation (2.83).

$$\dot{V}(x) = \left(\frac{\partial V(x)}{\partial x}\right)^t \cdot \frac{dx(t)}{dt} = \nabla_x V(x) \cdot F(x).$$

  The function V is called a *Lyapunov function* for the problem (2.83).

Taking into account the previous definitions, we can state the following theorem. For the proof we refer to [15]

Theorem 2.16 (Lyapunov's Theorem)

[5] A system of differential equations $\frac{dx(t)}{dt} = F(x(t), t)$ is said to be autonomous when $\frac{\partial F}{\partial t} = 0$, that is when F does not depend directly from the independent variable t.

If $\dot{V}(x)$ is negative in a neighbourhood of x_*, then x_* is asymptotically stable for the equation (2.83).
If $\dot{V}(x)$ is positive in a neighbourhood of x_*, then x_* is unstable for the equation (2.83).

Now let us consider the initial value problem

$$\frac{dx}{dt} = -F'(x(t))^{-1}F(x(t)) \qquad x(t_0) = x_0. \tag{2.84}$$

The iteration of the Euler's method with a time discretisation step equal to one applied to (2.84) is equivalent to the Newton iteration applied to the problem (2.1) when the full Newton step is taken. Thus, the time discretisation step corresponds to the steplength controlled by the damping parameter in the line–search type Newton methods and the initial condition of the dynamic system represents the starting point of the Newton sequence. Furthermore, the Lyapunov function V defined above plays the role of a merit function for the equilibrium points of the dynamic system. Thus, the dynamic system (2.84) is in some sense associated to the original problem (2.1).

Example 2.1

Consider the simple one–dimensional equation

$$\sin x = 0$$

and its associated differential problem

$$\frac{dx}{dt} = -\tan x \quad x(t_0) = x_0. \tag{2.85}$$

The analytic solution of (2.85) is a periodic function which can be written as

$$x(t) = \begin{cases} k\pi + \arcsin(e^{-t} \cdot e^{t_0} \cdot \sin x_0) \text{ if } x \in [k\pi, k\pi + \frac{\pi}{2}] \\ (k+1)\pi - \arcsin(e^{-t} \cdot e^{t_0} \cdot \sin x_0) \text{ if } x \in (k\pi + \frac{\pi}{2}, (k+1)\pi] \end{cases}$$

for $k = 0, \pm 1, \pm 2 \ldots$ and it is plotted in figure 2.5 for different initial points x_0. It is interesting to observe the behaviour of the Newton's method compared to the exact solution curve, in particular the influence of the starting point and of the step length can be shown on this simple example.
In figures 2.6 and 2.7, the exact solution (dotted line) is plotted together with the paths obtained by the classical Newton's method (dotted line with

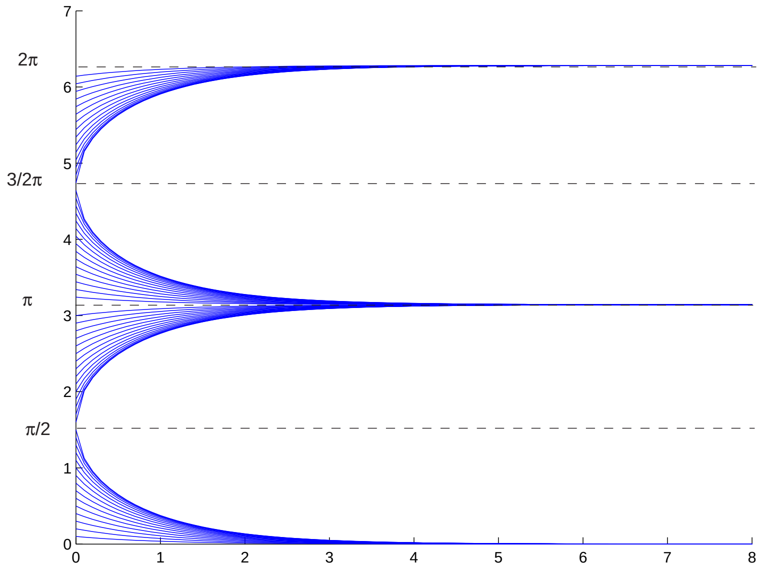


Figure 2.5: Analytic solution curve of $\frac{dx}{dt} = -\tan(x)$

circles), the Euler's method with step equal to 0.01 (solid line), and the Newton's method with Armijo (dotted line with squares) and Eisenstat–Walker (dotted with diamonds) steplenght selection rules, for ten different starting points from 1.1 to 2.

It can be observed that the Euler's method with uniform step size 0.01 provides a very good approximation to the exact solution curve, such that in the figures the two lines are superimposed. Furthermore, the Newton's method always get a solution of the equation $\sin x = 0$, but in several cases it is attracted by a solution which is not the one closest to the starting point. It has also to be noticed that in such cases the Newton trajectory crosses the regions in which the first derivative of the sine function is zero. When a line–search strategy is applied, as we could expect, the Newton's method path is much more stick to the solution curve, since the step length, which corresponds to the time discretization step, is smaller. From the numerical example, we can observe that the path generated by the Newton's method with the Armijo backtracking rule approximates quite well the analytic solution of (2.85), while the Eisenstat-Walker rule, allowing larger step length, is not always close to that curve. This capability to stick to the curve $x(t)$ which solves (2.84), due to a small step length, is not always a desirable

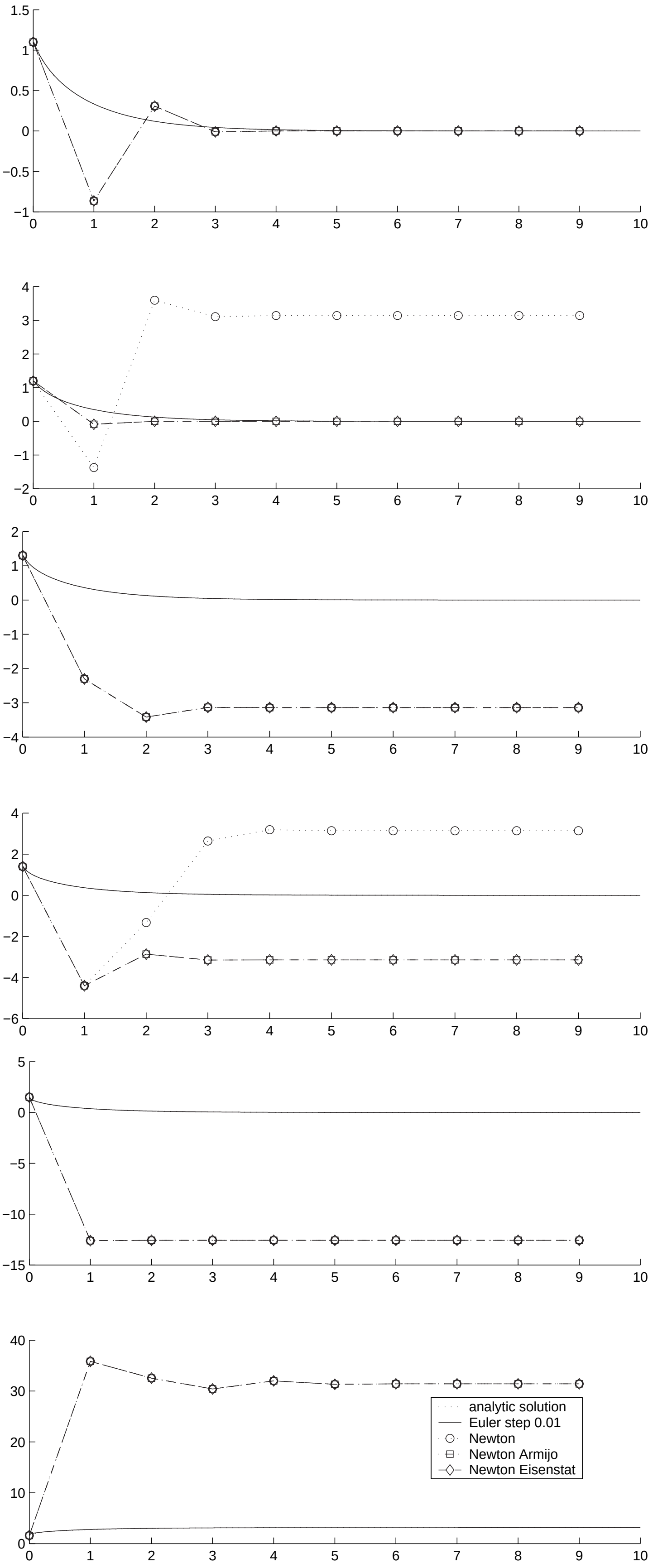


Figure 2.6: Comparisons

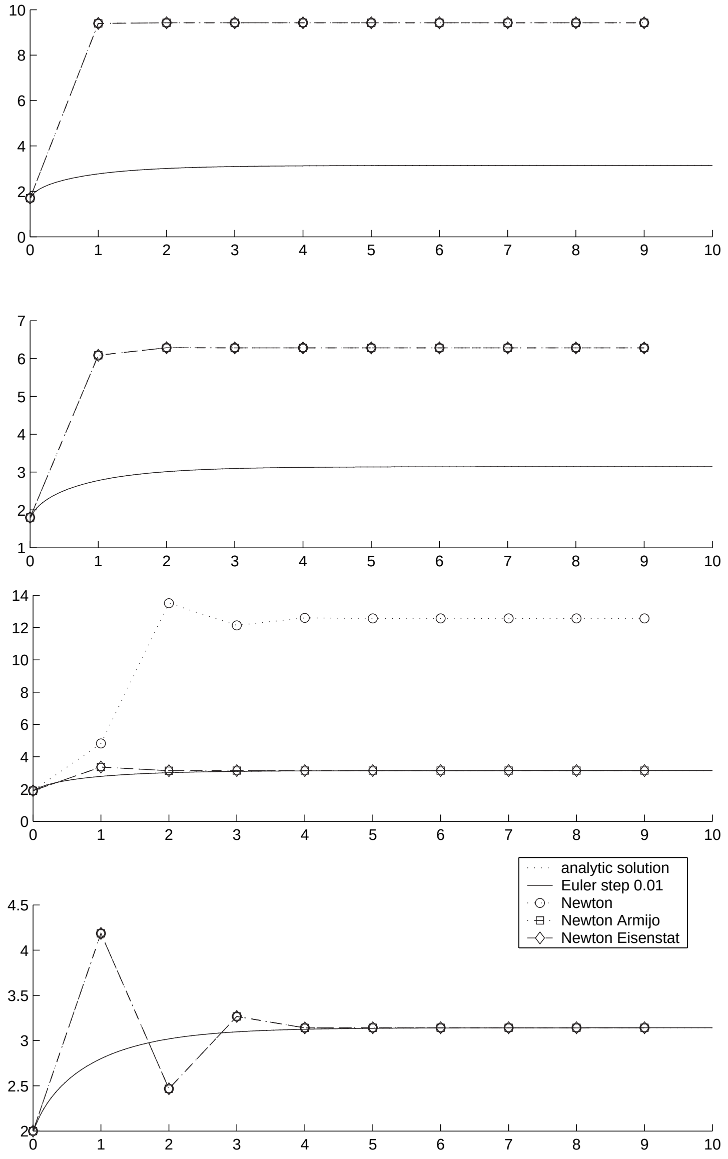


Figure 2.7: Comparisons

feature of the Newton's method, as shown in the following example.

Example 2.2 (Powell)

Consider the following nonlinear system

$$F(x) \equiv \begin{pmatrix} x_1 \\ 10x_1/(x+0.1) + 2x_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.86}$$

This example has been firstly proposed in [62], where the author proved that the Newton's method, starting from the point $x_0 = (3,1)^t$, does not converge to the unique solution $x = (0,0)^t$, when the step length is chosen as the first local minimizer of the least squares function (2.5). Indeed, the iterative process leads to the point $x_\infty = (1.8016, 0.0000)^t$, which is neither a solution of the system (2.86) nor a stationary point of the least squares merit function, as showed in [19]. Indeed, the vector $F(x_\infty)$ does not belong to the null space of the jacobian matrix of F,

$$F'(x) = \begin{pmatrix} 1 & 0 \\ 1/(x_1+0.1)^2 & 4x_2 \end{pmatrix},$$

computed in the points of the x_1 axis, where it is singular.
Here we give another explanation of such behaviour of the Newton's method, according with the observations above.
Taking into account that the inverse of F' is the following matrix

$$F'(x)^{-1} = \begin{pmatrix} 1 & 0 \\ -1/4x_2(x_1+0.1)^2 & 1/(4x_2) \end{pmatrix},$$

the dynamic system associated to the nonlinear system (2.86) is given by

$$\begin{align} \frac{dx_1}{dt} &= -x \tag{2.87} \\ \frac{dx_2}{dt} &= \frac{x_1}{4x_2(x_1+0.1)^2} - \frac{1}{4x_2}\left(\frac{10x_1}{x_1+0.1} + 2x_2^2\right) \tag{2.88} \end{align}$$

which yields

$$\frac{dx_2}{dx_1} = \frac{1}{4x_2}\left(-\frac{1}{(x_1+0.1)^2} + \frac{10}{x_1+0.1} + \frac{2x_2^2}{x_1}\right). \tag{2.89}$$

By applying standard techniques, it is possible to obtain the analytic solution of (2.89), depending on the initial value $x_0 = (x_1(t_0), x_2(t_0))^t = ((x_0)_1, (x_0)_2)^t$

$$x_2 = \pm\sqrt{x_1\left(-\frac{5}{x_1+0.1} + \frac{5}{(x_0)_1+0.1} + \frac{(x_0)_2^2}{(x_0)_1}\right)}. \tag{2.90}$$

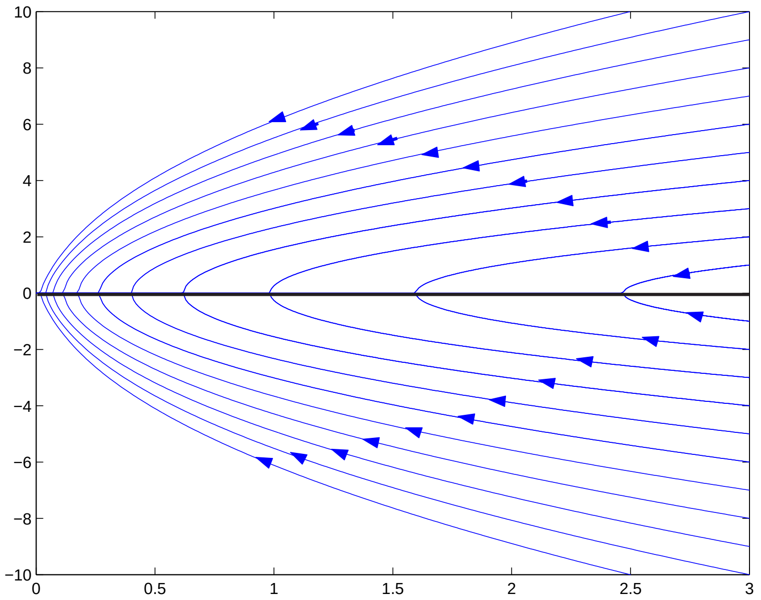


Figure 2.8: Powell example: phase plane curves

which is plotted in figure 2.8 in the phase plane (x_1, x_2). It is easy to see that the Newton direction $s_k^N = -F'(x_k)^{-1}F(x_k)$ is tangent to the curve (2.90) which the point x_k belongs to. In figure 2.9 the situation is depicted: the curves are the exact solutions of the differential equation (2.89), and the line is the Newton path (taking the full step). The same path is obtained when we use a backtracking (Eisenstat–Walker or Armijo) strategy, thus the lines related to the backtracking cases are superimposed. In all these cases, the iterative process gets the solution. The black circles are the first three iterates of the Newton's methods when the step length α_k is the first local minimizer of the scalar function $\phi(\alpha) = 1/2\|F(x_k + \alpha s_k^N)\|^2$, as in [62]. At the first iterate, the step is shortened with $\alpha \approx 0.395$ and this leads the next iterate very close to the x_1 axis, where the jacobian matrix is singular. This yields a Newton step almost orthogonal to the axis and the first local minimizer of the merit function along this direction is very close to the previous iterate, such that the step length is of order 10^{-3}. At the next iterate the situation is the same, and the sequence sticks to one of the solution curves and stagnates in a neighborhood of the point x_∞.

In the previous example, the jacobian matrix is singular on the line $x_2 = 0$

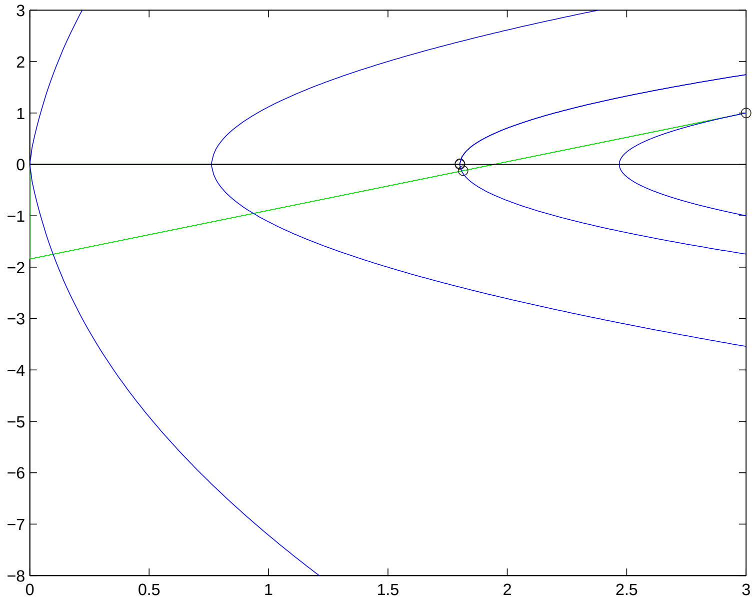


Figure 2.9: Numerical results

and the orbits in the phase plane end in that line.
This behaviour can be generalized, following [40], by defining a Lyapunov function for the points belonging to the set Ω, defined as

$$\Omega = \left\{ x \in \mathbb{R}^n : F'(x) \text{ is singular} \right\},$$

as a continuous and differentiable function V always positive except in Ω where it is equal to zero. For example we could take

$$V(x) = \frac{1}{2} \left(detF'(x) \right)^2 . \tag{2.91}$$

Here $detF'(x)$ denotes the determinant of the jacobian matrix $F'(x)$.
If we have $\dot{V}(x) < 0$ in some domain containing Ω, the function V is decreasing with respect to the variable t, on the solution curves of the differential equation. Thus, the function $V(x)$, with increasing values of t, will be able to become arbitrarily small on $x(t)$, which means that the distance between $x(t)$ and Ω will become arbitrarily small. Thus the solutions of the differential equation are directed towards the points of Ω. These points of Ω are called *end points*.

This is the case of the Powell example. Indeed, choosing the Lyapunov function as in (2.91) and taking into account (2.88), on the solution curves of (2.87)–(2.88) we have

$$\begin{aligned}
\dot{V}(x) &= \nabla_x V(x)^t \cdot \frac{dx}{dt} \\
&= -detF'(x) \cdot \nabla_x (detF'(x))^t \cdot F'(x)^{-1} F(x) \\
&= 4x_2 \, (0 \quad 4) \left(\begin{array}{c} -x_1 \\ \frac{x_1}{4x_2(x_1+0.1)^2} - \frac{1}{4x_2}\left(\frac{10x_1}{x_1+0.1} + 2x_2^2\right) \end{array} \right) \\
&= 4\frac{x_1 - 10x_1(x_1+0.1) - 2x_2^2(x_1+0.1)^2}{(x_1+0.1)^2} \\
&= -4\frac{10x_1^2 + 2x_2^2(x_1+0.1)^2}{(x_1+0.1)^2} \\
&\leq 0 \qquad (2.92)
\end{aligned}$$

If $\dot{V}(x) > 0$ in some domain containing Ω, the function V with increasing value of t, will be arbitrarily large on $x(t)$, then the distance between $x(t)$ and Ω will become arbitrarily large. Then, the solutions are diverging from the points of Ω, which are called *initial points.*

This last case is verified for the system (2.22), whose jacobian matrix is singular on the line $x_1 = x_2$. Outside this line, the inverse of the jacobian matrix is given by

$$F'(x)^{-1} = \frac{1}{x_1 - x_2} \left(\begin{array}{cc} x_1 & -1 \\ -x_2 & 1 \end{array} \right)$$

and the dynamic system associated to (2.22) is

$$\begin{aligned}
\frac{dx_1}{dt} &= -\frac{1}{x_1 - x_2}(x_1^2 - 5x_1 + 4) \\
\frac{dx_2}{dt} &= -\frac{1}{x_1 - x_2}(-x_2^2 + 5x_2 - 4)
\end{aligned}$$

which yields the following differential equation

$$\frac{dx_2}{dx_1} = -\frac{(x_2-4)(x_2-1)}{(x_1-4)(x_1-1)} \qquad (2.93)$$

By using standard techniques, it is possible to calculate the analytic expression of the solutions of (2.93), which can be written as

$$\frac{x_1 - 1}{x_1 - 4} = K\frac{x_2 - 4}{x_2 - 1}, \quad K = \frac{(x_0)_1 - 1}{(x_0)_1 - 4}\frac{(x_0)_2 - 4}{(x_0)_2 - 1}$$

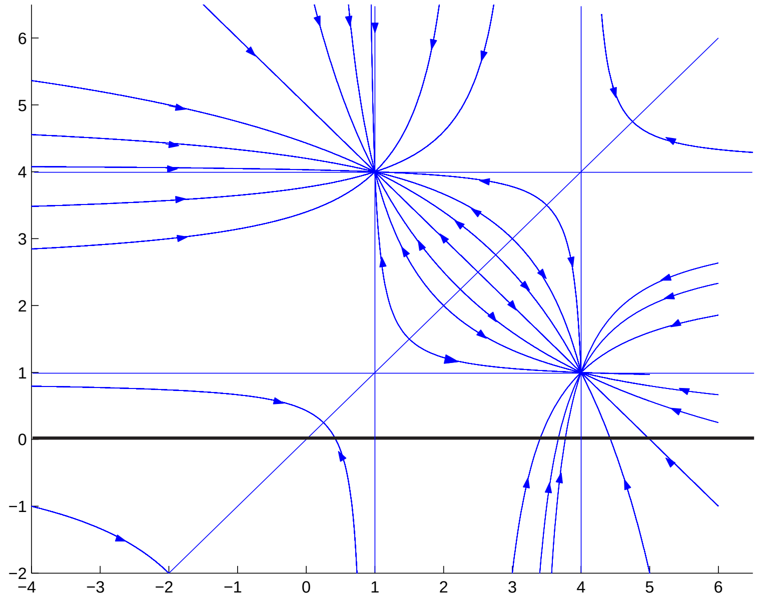


Figure 2.10: Solution curves of (2.93)

where $x_0 = ((x_0)_1, (x_0)_2)^t$ is the initial point. The solution curves in the phase plane are plotted in figure 2.10. In order to classify the points where the jacobian matrix is singular, we define the Lyapunov function as in (2.91) and proceeding as before we obtain $\dot{V}(x) = -(x_1^2+x_2^2-5(x_1+x_2)+8)$, whose value on the points of the line $x_1 = x_2$ is $-2(x_1^2-5x_1+4)$. This implies that $\dot{V}$ is positive in the points (y, y) where y belongs to the interval $(1, 4)$ (initial points), is zero for $y = 4$ and $y = 1$, and negative otherwise (end points). Finally, we can observe that, following the solution trajectories with initial point in the region S defined as

$$S = \{(x_1, x_2) : x_1 \geq 4 \text{ and } x_2 \geq 4\} \cup \{(x_1, x_2) : x_1 \leq 1 \text{ and } x_2 \leq 1\}$$

it is impossible to reach the solutions $(4, 1)$ and $(1, 4)$.

Chapter 3

Interior–Point Methods

The aim of this chapter is to present the class of interior–point (IP) methods for the solution of the NLP problem (1.1) in a quite general and unitary way. In the first two sections the barrier methods and the perturbation of the KKT conditions are explained, tacking into account the connections between these methods and the framework of the path–following methods. afterwards a simple scheme, which includes only the basic features and principles of the interior–point methods, can be written. Then, we restrict our attention to a particular class of IP methods, the Newton interior–point methods, which are strictly related to the solution of a nonlinear system with bound constraints. The last section deals with the relations between the Newton IP methods and the class of inexact Newton methods, and it is crucial for the next chapter.

3.1 Barrier methods

The idea of the barrier methods for the solution of the general nonlinear programming problem (1.1) is to replace the inequality constraints by adding a logarithmic term to the object function, obtaining the following *barrier problem*

$$\begin{array}{ll} \min & f(x) - \rho \sum_{i=1}^{m} \log(g_2)_i(x) \\ s.t. & g_1(x) = 0. \end{array} \tag{3.1}$$

where ρ is a positive scalar parameter. Since the barrier term becomes large when x is close to the boundary of the *feasible region*, the set $\mathcal{F} \equiv \{x \in \mathbb{R}^n : g_2(x) \geq 0\}$, the solution of (3.1) must be in the interior of $\mathcal{F}$.
The size of the *barrier parameter* ρ indicates the degree of influence of the logarithmic barrier term, thus it is reasonable to expect thatwhen ρ is close

to zero, the solution of (3.1) is close to the solution of the original problem (1.1). In particular, the barrier method consists in solving (approximately) a sequence of subproblems like (3.1) where $\rho = \rho_k$ is the barrier parameter of the k–th subproblem, providing that $\lim_{k\to\infty} \rho_k = 0$. Furthermore, the solution of the previous subproblem is used as starting point for the computation of the next solution.
Theorem 7 in [74] states the local convergence of the barrier methods applied to a nonlinear inequality constrained programming problem (see also [37]). Since any constrained minimum problem can be written in an equivalent way as an inequality constrained problem, it applies also to (3.1).
The solution of the barrier subproblem at the iterate k can be obtained by solving the KKT conditions, represented by the following nonlinear system:

$$\begin{array}{rcl} \nabla f(x) - \rho_k \ \sum_{i=1}^m \nabla (g_2)_i(x) \frac{1}{(g_2)_i(x)} - \nabla g_1(x)\lambda_1 & = & 0 \\ g_1(x) & = & 0. \end{array} \tag{3.2}$$

The unknowns of the system (3.2) are x and λ, and the system (3.2) is called the *primal system*. Furthermore, comparing the first equation of (3.2), and (1.3) it can be observed a correspondence between the components of the multiplier w and the terms $\frac{\rho_k}{g_2(x)}$. Indeed, denoting by x_* the solution of (1.1) and λ_*, w_* the multipliers associated to x_*, if standard sufficient optimality conditions hold at the solution then the sequence of the minimizers of (3.1) x_{ρ_k} converges to the solution x_*, the multipliers sequence $\{\lambda_{\rho_k}\}$ associated to x_{ρ_k} converges to λ_*, and the quantity $\frac{\rho_k}{g_2(x)}$ tends to w_* (see Theorem 8 in [74]).
The solutions x_ρ describe a parametrized curve in $\mathbb{R}^n$ whose parameter is ρ. Such curve is usually called homotopy path, or, in the framework of linear programming, *central path*. Under smoothness assumptions (Theorem 8 (iv) in [74]), is smooth. This suggest that a barrier method can be considered as a *path following* method.
In the path–following context, in [52] the authors investigate the conditions for the existence of the central path by considering the homotopy

$$P(x, \rho) = f(x) + \frac{1}{2\rho}\sum_{i=1}^{neq} (g_1)_i(x)^2 - \rho \sum_{i=1}^{m} \ln(g_2)_i(x)$$

which transform the constrained minimum problem (1.1) into a parametrized set of unconstraint problems of the form

$$\min P(x, \rho). \tag{3.3}$$

In particular, Theorem 3.1 claims that, if the hessian of the lagrangian of the problem (3.1) is nonsingular at the solution, then for every ρ sufficiently small there exists a unique solution of (3.1) which determines the homotopy path.

The primal approach (3.2) has been shown to present some drawbacks: the radius of convergence of the Newton method applied to (3.2) converges to 0 as the barrier parameter is close to zero [75]. Moreover, the first Newton step after a change of ρ_k is not very good and it tends to violate the constraint $g_2(x) \geq 0$. In order to avoid these drawbacks it is generally preferred another formulation of (3.2), which can be also derived from a different point of view, as described in the following section.

3.2 Perturbed Karush–Kuhn–Tucker systems

An equivalent way to state the problem (1.1) can be obtained by introducing the *slack variables*, i.e. a vector $s \in \mathbb{R}^m$, on the inequality constraints: this leads to the following reformulation of (1.1):

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_1(x) = 0 \\ & g_2(x) - s = 0 \\ & s \geq 0. \end{array} \tag{3.4}$$

The lagrangian function for the problem (3.4) is given by

$$\mathcal{L}(x, \lambda, w, s, z) = f(x) - \lambda^t g_1(x) - w^t(g_2(x) - s) - z^t s \tag{3.5}$$

where $z \in \mathbb{R}^m$ is the multiplier of the constraint $s \geq 0$. The KKT optimality conditions can be derived by differentiating $\mathcal{L}$ with respect to all its variables and the system of nonlinear equations obtained in such way has to be completed by the complementarity conditions as follows:

$$\begin{array}{rlcl} \mathcal{L}_x = & \nabla f(x) - \lambda^t \nabla g_1(x) - w^t \nabla g_2(x) & = & 0 \\ \mathcal{L}_\lambda = & -g_1(x) & = & 0 \\ \mathcal{L}_w = & -g_2(x) + s & = & 0 \\ & SWe_m & = & 0 \\ & s, w \geq 0. & & \end{array} \tag{3.6}$$

Here and in the following we denote $S = diag(s)$, $W = diag(w)$, $e_m = (1, ..., 1)^t \in \mathbb{R}^m$. In (3.6) we have taken into account of the equality

$$\mathcal{L}_s = w - z = 0$$

which implies $w = z$. The system (3.6) is also called a *primal–dual system*, because the multiplier w represents the *dual variable* for the problem (3.4). The constraints $s, w \geq 0$ define a feasible region $\mathcal{F}$ which is the nonnegative orthant of the plane (s, w). Defining a new variable $v \in \mathbb{R}^{n+neq+2m}$ as $v = (x^t, \lambda^t, s^t, w^t)^t$ and $H_1(v)$ as the vector $(\mathcal{L}_x^t, \mathcal{L}_\lambda^t, \mathcal{L}_w^t)^t$, then (3.6) can be written as

$$\begin{gathered} H(v) = \begin{pmatrix} H_1(v) \\ SWe_m \end{pmatrix} = 0 \\ s, w \geq 0. \end{gathered} \tag{3.7}$$

Hence, the nonlinear programming problem (1.1) leads to a nonlinear system, denoted here as $H(v) = 0$, with bounds on some variables, expressed by the constraints $s, w \geq 0$. It is worth to stress that a solution of (3.7) is a KKT point for the nonlinear problem (1.1), not necessarily a minimum point, for which the second order conditions should be verified.

If we solve the system $H(v) = 0$ with the Newton's method, at each iteration k we have to compute the vector Δv_k which is a solution of the Newton equation

$$H'(v_k)\Delta v_k = -H(v_k). \tag{3.8}$$

Writing the last m equations of the linear system (3.8), the ones related to the complementarity conditions, we obtain the following equalities

$$S_k \Delta w_k + W_k \Delta s_k = -S_k W_k e_m, \tag{3.9}$$

which imply that, if the i–th component of s_k is zero, then (3.9) becomes

$$(W_k)_i (\Delta s_k)_i = 0.$$

This yields $(\Delta s_k)_i = 0$, thus $(s_{k+j})_i = (s_k)_i$ for all $j = 1, 2, ...$, and the iterate sticks on the boundary of the feasible region. The same stagnation of the iterates occurs if we have $(w_k)_i = 0$, which implies $(w_{k+j})_i = (w_k)_i$ for $j = 1, 2, ...$

The above observations suggest the idea to perturb the system (3.7) only on the complementarity equations so that the cases $(s_k)_i = 0$ and $(w_k)_i = 0$ for any component i are excluded. The perturbed system can be written as

$$\begin{gathered} H(v) = \begin{pmatrix} H_1(v) \\ SWe_m \end{pmatrix} = \begin{pmatrix} 0 \\ \rho e_m \end{pmatrix} = \rho \tilde{e} \\ s, w > 0 \end{gathered} \tag{3.10}$$

where ρ is a positive scalar called *perturbation parameter* and $\tilde{e}$ denotes the vector $(0_{n+neq+m}, e_m^t)^t$.

Since ρ has to be positive, the variable s and w, whose components solve the complementarity conditions $(s_k)_i(w_k)_i = \rho$, for $i = 1, ..., m$, must stay in the interior of the feasible region.
The nonlinear system $H(v) = \rho\tilde{e}$ represents the *perturbed KKT conditions* for the problem (3.4).
In the framework of the interior–point methods we have to generate a sequence of perturbed problems like (3.10), where the perturbation parameter ρ_k decreases and such that $\lim_{k\to\infty} \rho_k = 0$. Thus, the k-th problem of the sequence is

$$\begin{array}{l} H(v) = \rho_k\tilde{e} \\ s, w > 0, \end{array} \tag{3.11}$$

which is equivalent to

$$\begin{array}{lcl} \nabla f(x) - \lambda^t\nabla g_1(x) - w^t\nabla g_2(x) & = & 0 \\ -g_1(x) & = & 0 \\ -g_2(x) + s & = & 0 \\ SWe_m & = & \rho_k e_m \\ & s, w > 0. & \end{array} \tag{3.12}$$

By introducing a "measure" $\mathcal{M}$ of the perturbed KKT conditions expressed by the vector $H_\rho(v) = H(v) - \rho\tilde{e}$ (for example $\mathcal{M}(H_\rho(v)) = \|H_\rho(v)\|_2$) and tacking into account the observations above, it is possible to write a general scheme for the whole class of the interior–point methods: we will call Interior–Point method every method which can be written as follows.

Scheme 3.1

1. Choose the initial guess v_0 s.t. $(s_0, w_0) > 0$, the stopping tolerance τ, the measure $\mathcal{M}$;

2. For $k = 0, 1, 2, ...,$ until $\mathcal{M}(H(v_k)) > \tau$

   2a. Choose the perturbation parameter ρ_k and the inner tolerance tol_{ρ_k};

   2b. Compute a new point v_{k+1} such that:

   $$\begin{array}{l} \mathcal{M}(H_{\rho_k}(v_{k+1})) < tol_{\rho_k} \\ (s_{k+1}, w_{k+1}) > 0 \end{array}$$

   2c. Set $k = k + 1$

The step 2 of the scheme implies that, for any k, v_{k+1} is an approximate solution of (3.11) and the accuracy of this solution is determined by the tolerance tol_{ρ_k}.
The scheme 3.1 can also describe a barrier method: indeed, if we consider the k-th barrier problem for the solution of (3.4)

$$\begin{array}{ll} \min & f(x) - \rho_k \sum_{i=1}^{m} \log s_i \\ s.t. & g_1(x) = 0 \\ & g_2(x) - s = 0, \end{array} \tag{3.13}$$

then the optimality conditions for (3.13) are represented by the following primal system

$$\begin{array}{lcll} \nabla f(x) - \lambda^t \nabla g_1(x) - w^t \nabla g_2(x) & = & 0 & \\ -g_1(x) & = & 0 & \\ -g_2(x) + s & = & 0 & \\ \rho_k \frac{1}{s_i} - w_i & = & 0 & i = 1, ...m. \end{array} \tag{3.14}$$

By multiplying the last m equations by the component s_i we obtain the primal–dual system (3.11).
Thus, there is an equivalence between this barrier method and the IP methods, hence the scheme 3.1 can describe both a barrier method and an interior–point method for the solution of the perturbed KKT conditions (3.12). Moreover, it can be observed that any solution of (3.14) or (3.12), due to the last m equations, lies on the curve $s_i w_i = \rho_k$ in the plane (s_i, w_i).

3.3 Newton Interior–Point Methods

The one described by the scheme 3.1 is a wide class of methods, allowing many choices for $\mathcal{M}$, for the perturbation parameter ρ_k and for the method employed at the step 2b; in this section we consider the case when the step 2a is performed by applying a Newton–type method to the perturbed KKT conditions (3.10). In order to obtain global convergence properties, we include in the Newton's method one of the two globalization techniques, the trust region or the line–search, which require the choice of a suitable merit function and a criterion for the acceptance of the trial step. The scheme 3.1 for this choice of the inner solver can be written as follows:

Scheme 3.2

1. Choose the initial guess v_0 s.t. $(s_0, w_0) > 0$, the stopping tolerance τ, the measure $\mathcal{M}$;

2. For $k = 0, 1, 2, \ldots$, until $\mathcal{M}(H(v_k)) > \tau$

   2a. Choose the perturbation parameter ρ_k and the inner tolerance tol_{ρ_k};

   2b. By applying the global Newton method (line–search or trust region) to the problem (3.12), compute a new point v_{k+1} such that

   $$\begin{array}{l} \mathcal{M}(H_{\rho_k}(v_{k+1})) < tol_{\rho_k} \\ (s_{k+1}, w_{k+1}) > 0 \end{array}$$

   2c. Set $k = k + 1$.

It has to be noticed that, even though from a theoretical point of view there is no difference in solving (3.14) or (3.12) at the step 2b., these two problems do not lead to a Newton algorithmic equivalence, as pointed out in [35]. Indeed, applying the Newton method to (3.14) or (3.12) generates different iterates.
The inherent ill conditioning of the primal system when ρ_k is close to zero is well known in literature since the late 1960s [58], and for this reason the interior methods were not very popular in the 1970s. In more recent works, several authors have pointed out that the computation of the search direction by means of linear equations solvers applied directly to the ill conditioned systems produces more accurate solutions than one could expect [77, 76]). Indeed, observing the behaviour of the interior methods, it seems that they get the solution before that the conditioning is too poor. Nevertheless, it is always convenient to carefully choose the formulation of the barrier subproblem, in order to avoid the worse effects of the ill conditioning. For example, it is easy to see that the derivatives of the left hand side of the last m equations of (3.14), $\rho_k \frac{1}{s_i} - w_i = 0 \quad i = 1, \ldots m$, are not bounded when the slack variables approach to zero, which is not the case with the perturbed complementarity conditions $s_i w_i = \rho_k$ of the primal–dual system (3.12).Thus, in the last 10 years all the main proposed algorithms, as the ones presented in [35, 2, 70, 66, 18, 71, 69], follows a primal–dual approach.

3.3.1 Newton Line–Search IP Methods

The Newton line–search IP methods are based on the solution of the *perturbed Newton equation*

$$H'(v_k)\Delta v_k = -H(v_k) + \rho_k \tilde{e} \tag{3.15}$$

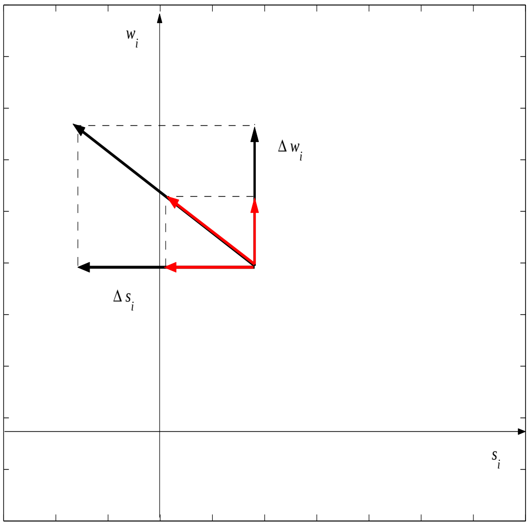


Figure 3.1: Restoring feasibility

which has to be solved at each IP iteration k and it is obtained by applying the Newton method to (3.12). The Newton step does not guarantee the feasibility of the new iterate, then, if some components of $s_k + \Delta s_k$ or $w_k + \Delta w_k$ are negative, the step is reduced until $s_k + \Delta s_k > 0$ or $w_k + \Delta w_k > 0$. Defining α_k as the step length, we have that

$$\alpha_k = \gamma \min \left\{ -\frac{(s_k)_i}{(\Delta s_k)_i}, -\frac{(w_k)_i}{(\Delta w_k)_i}, \text{ where } (s_k)_i + (\Delta s_k)_i < 0, \quad (w_k)_i + (\Delta w_k)_i < 0 \right\} \tag{3.16}$$

where γ is a parameter less than one.

The resulting vector is depicted in figure 3.1. The damping parameter α_k has also to guarantee that the new iterate is sufficiently close to the central path, i.e. the new iterate is a sufficiently good approximation of the solution of the k–th barrier subproblem (3.11). Thus, the damping parameter is reduced until some *centrality conditions*, which express the idea of sufficient proximity to the central path, are satisfied. Then, the step size is reduced again until a sufficient reduction of a given merit function is reached. To be sure that by reducing the step length a sufficient reduction of the merit function can be reached, we should have that the Newton step is a descent direction for the merit function: this propriety can be guaranteed by a suitable choice of the perturbation parameter, as we will see below. Then a new iterate is computed and the loop is repeated until some measure of the KKT conditions satisfies a fixed tolerance. Hence, at every IP iteration, instead of having an inner tolerance tol_{ρ_k} which evaluates when the subproblem (3.11) is solved with a sufficient accuracy, the number of Newton iterations is *a priori* fixed to one and the closeness to the central path is obtained by means of the reduction of the step length until the centrality conditions are satisfied. We can resume the remarks above in the following scheme:

Scheme 3.3

- Choose an initial guess v_0 s.t. $(s_0, w_0) > 0$;
- Choose the perturbation parameter

$$\rho_k = \sigma_k \mu_k \text{ s.t.} \quad \begin{array}{l} \sigma_k \in [0,1) \\ \mu_k = \frac{s_k^t w_k}{m} \end{array}$$

- Solve the perturbed Newton equation

$$H'(v_k)\Delta v_k = -H(v_k) + \rho_k \tilde{e}$$

- Move along the direction computed at the previous step: choose the damping parameter α_k such that the new iterate satisfies

  feasibility;

  centrality conditions;

  sufficient decrease of the merit function;

- Update the iterate $v_{k+1} = v_k + \alpha_k \Delta v_k$.

In determining the direction, we implicitly assume that a solution of the perturbed Newton equation exists at each iteration k: this is a crucial point for this kind of methods, and it is usually assumed that a sufficient condition for the nonsingularity of $H'(v_k)$ holds at each iteration. In order to make some general observations, for the moment we simply assume that the matrix $H'(v_k)$ is nonsingular.

1. *The perturbation parameter.* The choice of the perturbation parameter as product of the two scalar factors σ_k and μ_k, where

$$\mu_k = \frac{s_k^t w_k}{m} \tag{3.17}$$

is typical for many interior–point algorithms, for example in [35] and [70], and it measures the average value of the product pairs $(s_k)_i(w_k)_i$. It can also be justified with the following remark.
Consider the function

$$\phi(v) = \frac{1}{2}\|H(v)\|_2^2. \tag{3.18}$$

The gradient of $\phi(v)$ is given by

$$\nabla\phi(v) = H'(v)^t H(v).$$

If Δv_k is the solution of the perturbed Newton equation (3.15), then $\Delta v_k = H'(v_k)^{-1}(-H(v_k) + \rho_k\tilde{e})$, hence

$$\begin{aligned}\nabla\phi(v_k)^t\Delta v_k &= (H'(v_k)^t H(v_k))^t[H'(v_k)^{-1}(-H(v_k) + \rho_k\tilde{e})]\\ &= H'(v_k)^t(-H(v_k) + \rho_k\tilde{e})\\ &= (-\|H(v_k)\|^2 + \rho_k H(v_k)^t\tilde{e}).\end{aligned}$$

This implies that $\nabla\phi(v_k)^t\Delta v_k < 0$ if and only if $\rho_k < \frac{\|H(v_k)\|}{s^t w}$. Thus, in order to obtain a descent direction for the nonlinear least–squares merit function ϕ, a possible choice is (3.17). In [35], the authors show that their algorithm converges to a zero of the merit function (3.18) with the perturbation parameter chosen as in (3.17). For different merit functions, we have to ensure the descent property of the Newton step.

2. *Centrality conditions.* The centrality conditions can be derived as follows [35, 78]: at the current point v, define, for a given direction Δv and for the step length α, the point $v(\alpha) = v + \alpha\Delta v$ and the two functions

$$\begin{aligned}f^I(\alpha) &= \min(W(\alpha)s(\alpha)) - \gamma\tau_1 w(\alpha)^t s(\alpha)/m \qquad (3.19)\\ f^{II}(\alpha) &= w(\alpha)^t s(\alpha) - \gamma\tau_2\|H_1(v(\alpha))\|, \qquad (3.20)\end{aligned}$$

where $\gamma \in (0, 1)$ is a fixed parameter and τ_1 and τ_2 are two constants depending on the initial value v_0. The function f^I can be considered a measure of the distance of $v(\alpha)$ from the central path in the plane (s, w): indeed (3.21) defines the one sided ∞–norm of neighborhood of the central path. The function f^{II} is a weighted difference between the complementarity term and the other components of the vector $H(v(\alpha))$. By requiring that

$$\begin{aligned} f^I(\alpha) &\geq 0 \qquad (3.21)\\ f^{II}(\alpha) &\geq 0 \qquad (3.22) \end{aligned}$$

it follows that $(s(\alpha), w(\alpha)) > 0$ and, roughly speaking, that the point $v(\alpha)$ lies in a neighborhood of the central path, and the complementarity part of $H(v(\alpha))$ is not too small compared to the vector $H_1(v(\alpha))$. Other path following strategies are shown in [3] and can be obtained by combining different centrality conditions and merit functions.

3. *Merit functions.* The *damping parameter* α_k has also to guarantee the sufficient decrease of the chosen merit function. A classical choice for the merit function is the nonlinear least squares function $\phi(v)$ defined in (3.18). This merit function has the advantage that it is differentiable and it gives good convergence properties to the algorithm, even if a solution of $\phi(v) = 0$ is not necessarily a minimum point for the nonlinear problem. Many authors [2, 70, 71] have proposed different merit functions which in general follows the idea of a penalty–barrier function.
   The damping parameter is reduced until a backtracking rule is satisfied. In the most part of the algorithms, the backtracking rule is the classical Armijo–Wolfe condition

$$\phi(v(\alpha)) < \phi(v(0)) + \beta\nabla\phi(v(0))\Delta v_k \qquad (3.23)$$

   where β is a scalar parameter less than one and $\phi(v)$ is the chosen merit function.

4. *The solution of the perturbed Newton equation.* The computation of the search direction is the main computational task of the method. The jacobian matrix of $H(v)$ is a nonsymmetric matrix whose block structure is

$$H'(v) = \begin{pmatrix} Q & B & C & 0\\ B^t & 0 & 0 & 0\\ C^t & 0 & 0 & I_m\\ 0 & 0 & S & W \end{pmatrix} \qquad (3.24)$$

   where

$$\begin{array}{l} Q = \nabla^2_{xx}\mathcal{L}(x, \lambda, s, w)\\ B^t = -\nabla g_1^t(x)\\ C^t = -\nabla g_2^t(x), \end{array} \qquad (3.25)$$

thus the perturbed Newton equation can be written as

$$
\begin{aligned}
Q\Delta x + B\Delta\lambda + C\Delta w &= -\mathcal{L}_x && (3.26)\\
B^t\Delta x &= -\mathcal{L}_\lambda && (3.27)\\
C^t\Delta x + \Delta s &= -\mathcal{L}_w && (3.28)\\
S\Delta w + W\Delta s &= -SWe_m + \rho_k e_m && (3.29)
\end{aligned}
$$

Since the matrix (3.24) is nonsymmetric, practical algorithms do not solve the system (3.26)–(3.29), but they apply some elimination techniques in order to obtain an equivalent formulation of the system. From the last block of equations (3.29), we get the relation

$$\Delta s = -W^{-1}S\Delta w - (Se_m - W^{-1}\rho_k e_m), \qquad (3.30)$$

thus, by substituting in (3.26), we obtain a 3 × 3 block system in the *reduced form*

$$\begin{pmatrix} Q & B & C \\ B^t & 0 & 0 \\ C^t & 0 & E \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_x \\ -\mathcal{L}_\lambda \\ -\mathcal{L}_w + Se_m - \rho_k W^{-1} e_m \end{pmatrix}. \qquad (3.31)$$

where $E = -W^{-1}S$. By a further substitution from the third block of equations

$$\Delta s = \quad S^{-1}WC^t\Delta x + S^{-1}W\mathcal{L}_w - w + \rho_k s, \qquad (3.32)$$

we obtain the following *condensed form* of the system (3.26)–(3.29)

$$\begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda_1 \end{pmatrix} = \begin{pmatrix} c \\ q \end{pmatrix}, \qquad (3.33)$$

with

$$
\begin{aligned}
A = &\quad Q + CS^{-1}WC^t\\
c = &\quad -\mathcal{L}_x - C[S^{-1}W\mathcal{L}_w - w + \rho_k s]\\
q = &\quad -\mathcal{L}_\lambda.
\end{aligned}
$$

For the nonsingularity of the matrices in (3.31) and (3.33) we can exploit the following result:

Theorem 3.1 [36, pp.161–163] Let M_1 a $p \times p$ nonsingular matrix and let M_2, M_3 and M_4 be $q \times q$, $p \times q$, $q \times p$ matrices respectively. If $M_2 - M_4M_1^{-1}M_3$ is a nonsingular matrix, then the matrix

$$M = \begin{pmatrix} M_1 & M_3 \\ M_2 & M_4 \end{pmatrix}$$

is nonsingular and its inverse is given by

$$M^{-1} = \begin{pmatrix} M_1 & M_3 \\ M_2 & M_4 \end{pmatrix}^{-1} = \begin{pmatrix} M_1^{-1} + M_1^{-1} M_3 M_0 M_4 M_1^{-1} & -M_1^{-1} M_3 M_0 \\ -M_0 M_4 M_1^{-1} & M_0 \end{pmatrix}^{-1}$$

where $M_0 = (M_1 - M_4 M_1^{-1} M_3)^{-1}$.

Theorem 3.2 (Theorem 6 in [11]) The coefficient matrix in (3.33) is nonsingular if one of the following conditions hold:

C3' the matrices A and $B^T A^{-1} B$ are nonsingular;

C3" B^T is a full row–rank matrix and A is positive definite on the null space of B^T: $\mathcal{N}(B^T) = \{\boldsymbol{x} \in \mathbb{R}^n \ : \ B^T \boldsymbol{x} = 0\}$.

Proof. If C3' holds, it is immediate to prove that the following matrix is the inverse of the coefficient matrix in (3.33), by applying Theorem 3.1:

$$\begin{pmatrix} A^{-1} - A^{-1}B(B^tA^{-1}B)^{-1}B^tA^{-1} & A^{-1}B(B^tA^{-1}B)^{-1} \\ (B^tA^{-1}B)^{-1}B^tA^{-1} & -(B^tA^{-1}B)^{-1} \end{pmatrix}$$

For the condition C3", see [54, p. 424]. □

It has to be noticed that the diagonal matrices $W^{-1}S$, $S^{-1}W$, S and W lead to an ill conditioning of the systems obtained by means of the elimination techniques: in [78] the author shows that "the error in the computed step due to the finite precision may become large as μ_k decreases, but it does not interfere with the convergence of the iterate to the solution since it belongs almost entirely to the null space of the gradients of the constraints active in the solution". He also points out that the centrality conditions (3.21) and (3.22) give an important contribution in the error analysis, since they provide an estimate bound for the variables s and w in a neighborhood of the solution.

For the solution of the perturbed Newton equation we could distinguish two different approaches, the direct and the iterative one.

The matrices in (3.31) and (3.33) in general are not positive definite, thus the factorization subroutines can only exploit the symmetry of the matrices. For this reason, *regularization* techniques have been proposed for both direct [70] and iterative [1] approach. The aim of such regularization is to make the matrix (3.33) positive definite by adding positive quantities to the diagonal of the left up block A. Then, the system (3.33) can be be solved by factorizing the matrix with a Cholesky–like in the direct case, or by means of a preconditioned conjugate gradient if an iterative approach is followed.

The matrices in (3.31) and (3.33) play a fundamental role in all the interior–point algorithms and they are a central interest of this dissertation, thus we refer to the next chapter for a deeper investigation of the techniques employed for the computation of the direction.

3.4 Newton Interior-Point Methods as Inexact Newton Methods

In this section we will show the conditions under which any algorithm following the scheme 3.4 can be considered as a special case of inexact Newton methods.

Suppose that the vector Δv_k is a solution of the perturbed Newton equation (3.15) and define the vector r_k as the residual of the Newton equation for the problem $H(v) = 0$ as follows:

$$r_k = H'(v_k)\Delta v_k + H(v_k). \tag{3.34}$$

We have that

$$r_k = \rho_k \tilde{e}. \tag{3.35}$$

This means that solving the perturbed Newton equation is equivalent to "approximately" solve the Newton equation, with residual equal to the vector $\rho_k \tilde{e}$. From the definition of $\tilde{e}$ we have that

$$r_k = \begin{pmatrix} 0 \\ \sigma_k \mu_k e_m \end{pmatrix},$$

thus

$$\|r_k\| = \sigma_k \mu_k \sqrt{m}. \tag{3.36}$$

Since the residual condition (2.21) of the inexact Newton method and tacking into account (3.34), such condition for the problem $H(v) = 0$ can be written as

$$\|r_k\| \leq \sigma_k \|H(v_k)\|. \tag{3.37}$$

using σ_k as forcing term.

It has to be noticed that σ_k belongs to the interval $(0, 1)$, so that it can be chosen as forcing parameter. From (3.36), we can derive a condition on the parameter μ_k such that Δv_k (i.e. a solution of the perturbed Newton equation) is an inexact Newton step for the problem $H(v) = 0$ at the level σ_k. Indeed, it is easy to see that, if

$$\mu_k \leq \frac{\|H(v_k)\|}{\sqrt{m}}, \tag{3.38}$$

then (3.37) is satisfied.
It is straightforward to prove that the classical choice for μ_k as in (3.17) satisfies the condition (3.38), since we have

$$\mu_k^{(1)} = \frac{s_k^t w_k}{m} \leq \mu_k^{(2)} = \frac{\|H(v_k)\|}{\sqrt{m}} \tag{3.39}$$

(see [28]).
Now consider the case when the vector Δv_k is an approximate solution of the perturbed Newton equation, but such that the perturbed complementarity conditions are solved exactly. This gives the following residual vector (for the problem $H(v) = 0$)

$$r_k = \begin{pmatrix} \bar{r}_k \\ \sigma_k \mu_k e_m \end{pmatrix} \tag{3.40}$$

where $\bar{r}_k$ is the residual vector of the first $n+neq+m$ equations of the linear system

$$H'(v_k)\Delta v_k + H(v_k) = 0.$$

It follows that

$$\|r_k\|^2 = \|\bar{r}_k\|^2 + \sigma_k^2 \mu_k^2 m \tag{3.41}$$

from which we can derive conditions on $\|\bar{r}_k\|$ and on μ_k such that the approximate solution of the perturbed Newton equation is also an inexact Newton step for the problem $H(v) = 0$.
By requiring that

$$\|\bar{r}_k\| \leq \delta_k \|H(v_k)\| \tag{3.42}$$

and that (3.38) holds, we obtain the following inequality:

$$\|r_k\| \leq (\delta_k + \sigma_k)\|H(v_k)\|. \tag{3.43}$$

Hence, choosing δ_k and σ_k such that $0 < \delta_k + \sigma_k < 1$, we can conclude that Δv_k is an inexact Newton step at the level $(\delta_k + \sigma_k)$. It can be observed that, if the forcing term δ_k is chosen equal to zero, then we are in the previous case.
Suppose now that the direction Δv_k satisfies the condition (3.43), so that it

is an inexact Newton step at the level $(\delta_k + \sigma_k)$.
In order to obtain an inexact Newton sequence, we should require that the new iterate, computed along the vector Δv_k, guarantees a sufficient decrease of the function $\|H(v)\|$. In other words, the norm condition (2.29) of the inexact Newton method should hold, and this can be guarantee by means of a backtracking technique, as the one in the scheme 2.1.
By introducing the damping parameter α_k for the step length, the norm condition becomes

$$\|H(v_k + \alpha_k \Delta v_k)\| \leq (1 - \beta\alpha_k(1 - \sigma_k - \delta_k))\|H(v_k)\|. \tag{3.44}$$

We can summarize all the considerations above by introducing a new scheme for the class of Newton interior–point methods with line–search.

Scheme 3.4

- Choose an initial guess v_0 s.t. $(s_0, w_0) > 0$;
- Choose the parameters

$$\rho_k = \sigma_k \mu_k \text{ s.t. } \quad \begin{array}{l} \sigma_k, \delta_k \in [0, 1) \\ \sigma_k + \delta_k < 1 \\ \mu_k \in [\mu_k^{(1)}, \mu_k^{(2)}] \end{array}$$

- Compute a direction Δv_k such that

$$\|\bar{r}_k\| \leq \delta_k \|H(v_k)\|$$

- Move along the direction computed at the previous step: choose the damping parameter α_k such that the new iterate satisfies

  feasibility;

  centrality conditions;

  sufficient decrease of the merit function (condition (3.44));

- Update the iterate $v_{k+1} = v_k + \alpha_k \Delta v_k$.

This scheme is different from scheme 3.2 in two crucial points: the choice of the perturbation parameter in a larger interval and the way to determine the search direction. In particular, under some assumption, condition (3.43) can be employed as stopping criterion for an iterative solver applied to the perturbed Newton equation at each IP iteration. The application of an iterative inner solver is an useful tool in the implementation of the Newton interior–point methods and it can improve the effectiveness of the algorithm. On the other hand, viewing the interior–point methods as special case of inexact Newton methods gives a strong theoretical foundation for the convergence of the algorithm, and convergence theorems can be proved by means of the ones stated in Chapter 2 (see [28, 4]).

3.5 Previously proposed IP algorithms

In this section we present three optimization algorithm belonging to the interior–point class which have shown to be very efficient on nonlinear large scale test problems.
The first one, LOQO, is similar to the one proposed in the next chapters, while the second one, Knitro, follows a quite different approach, employing sequential quadratic programming ant trust region techniques.
The third algorithm, called IPOPT, is a barrier method implementing a filter line–search technique, a new approach followed also in [69].
At the end of this section, the table 3.1 summarizes the numerical results of these three softwares on some of the test problems described in the Chapter 6.

3.5.1 LOQO

The LOQO algorithm [70, 65] belongs to the class described by the scheme 3.4 and it addresses to a basic problem formulation with inequality constraints only

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_2(x) \geq 0. \end{array} \tag{3.45}$$

which after introducing the slack variables becomes

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_2(x) - s = 0 \\ & s \geq 0. \end{array} \tag{3.46}$$

The search direction is computed by solving the reduced form of the system, which in this case has the following form:

$$\begin{pmatrix} -Q & \nabla g_2(x) \\ \nabla g_2(x)^t & SW^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta w \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \nabla g_2(x)^t w \\ -g_2(x) + s + SW^{-1}(w - \rho S^{-1} e_m) \end{pmatrix} \tag{3.47}$$

Along this direction the steplenght is reduced such that the new iterate is feasible and in order to guarantee a sufficient reduction of the merit function

$$\Phi_{\beta,\rho} = f(x) - \rho \sum_{i=1}^{m} \ln(s_i) + \frac{\beta}{2} \|g_2(x) - s\|.$$

Indeed, if the matrix Q is positive definite, then the direction computed by solving the system (3.47) is a descent direction for that merit function, as shown in [70].

Thus, having a positive definite hessian matrix is crucial and the LOQO algorithm provides to this issue in two ways. First of all, the inequality constraints which are simple bounds can be treated separately, and by operating the substitution (3.32) they produce a positive diagonal matrix to be added to the hessian matrix. This can improve the stability of the algorithm, and for this reason, in the LOQO code, also the variables x_i which are not bounded, are written as the difference of two nonnegative variables t_i^- and t_i^+.

$$\begin{array}{rcl} x_i - t_i^- + t_i^+ & = & 0 \\ t_i^-, t_i^+ & \geq & 0 \end{array}$$

Furthermore, the hessian matrix Q is modified by a diagonal perturbation, whenever during the factorization a pivotal element of the wrong sign occurs.

In summary, the search direction is the solution of a system with the same right hand side of (3.47), whose matrix has the following form:

$$\begin{pmatrix} -(Q + E_n) & \nabla g_2(x) \\ \nabla g_2(x)^t & E_m \end{pmatrix}. \tag{3.48}$$

In the more recent versions, LOQO also adopted the filter technique of Fletcher and Leyffer for the line search, as discussed in [6].

In [45], the authors analyze the global convergence to a first order optimality point for a general algorithm combining features of the previously mentioned versions of LOQO but the global convergence proof for the LOQO algorithm as not been published. Good practical performances have been observed for the solution of nonlinear programming problems.

3.5.2 KNITRO

The Knitro algorithm [18, 17] follows the scheme 3.1. It applies to the problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_1(x) = 0 \\ & g_2(x) \leq 0 \end{array}$$

and, after introducing the slack variables, at each step it solves a barrier subproblem of the type

$$\begin{array}{ll} \min & f(x) - \rho_k \sum_{i=1}^{m} \ln s_i \\ \text{s.t.} & g_1(x) = 0 \\ & g_2(x) + s = 0 \\ & s \geq 0. \end{array}$$

The lagrangian function can be written as

$$\mathcal{L}(x, s, \lambda, w) = f(x) - \rho \sum_{i=1}^{m} \ln s_i + \lambda^t g_1(x) + w^t (g_2(x) + s)$$

and the measure M of the optimality conditions of the barrier subproblem is defined as follows:

$$\max\{\|\mathcal{L}_x\|_\infty, \|Sw - \rho e_m\|_\infty, \|g_1(x)\|_\infty, \|g_2(x) + s\|_\infty\}.$$

The solution of each barrier subproblem is computed by sequential quadratic programming and trust–region techniques. Indeed, at each step the direction $(\Delta x_k, \Delta s_k)^t$ is computed as solution of the following quadratic subproblem:

$$\begin{array}{ll} \min_{\Delta x, \Delta s} & \nabla f(x_k)^t \Delta x_k + \frac{1}{2} \Delta x \nabla^2_{xx} \mathcal{L}(x_k . s_k, \lambda, w) \Delta x - \rho e^t S_k^{-1} \Delta s + \frac{1}{2} \Delta s^t \rho S_k^{-2} \Delta s \\ \text{s.t.} & \nabla g_1(x_k)^t \Delta x + g_1(x_k) = r_{\mathcal{E}} \\ & \nabla g_2(x_k)^t \Delta x + \Delta s + g_2(x_k) + s_k = r_{\mathcal{I}} \\ & (\Delta x, \Delta s) \in T_k \end{array} \tag{3.49}$$

where λ and w are Lagrangian multipliers estimates, T_k indicates the trust–region at the iterate k and $r_{\mathcal{E}}$ $r_{\mathcal{I}}$ are the smallest residual vectors such that the constraints of (3.49) are consistent.

The solution of the quadratic subproblem (3.49) is computed in two steps: the normal step, which tries to satisfy the constraints of (3.49), and the tangential step which attempts to achieve the optimality. By omitting the iteration index, the normal step consists in solving the problem

$$\begin{array}{ll} \min_v & \|\nabla g_1(x_k)^t v_x + g_1(x_k)\|_2^2 + \|\nabla g_2(x_k)^t v_x + g_2(x_k) + s_k\|_2^2 \\ \text{s.t.} & \|(v_x, S_k^{-1} v_s)^t\|_2 \leq 0.8 \Delta_k \\ & v_s \geq -\tau s/2 \end{array} \tag{3.50}$$

where τ is a constant in $(0, 1)$ and the inequality constraints of define the trust region T_k. The solution of (3.50) is obtained approximately with the dogleg method. After the computation of the normal vector $v = (v_x, v_s)^t$, the residuals in (3.49) are defined as

$$r_{\mathcal{E}} = \nabla g_1(x_k)^t v_x + g_1(x_k), \quad r_{\mathcal{I}} = \nabla g_2(x_k)^t v_x + v_s + g_2(x_k) + s_k$$

and the tangential step is computed as the solution of the problem

$$\begin{array}{ll} \min & \nabla f(x_k)^t \Delta x_k - \rho e^t S_k^{-1} \Delta s + \frac{1}{2}\{\Delta x \nabla^2_{xx} \mathcal{L}(x_k.s_k, \lambda, w) \Delta x + \Delta s^t \rho S_k^{-2} \Delta s\} \\ \text{s.t.} & \nabla g_1(x_k)^t \Delta x = \nabla g_1(x_k)^t v_x \\ & \nabla g_2(x_k)^t \Delta x + \Delta s = \nabla g_2(x_k)^t v_x + v_s \\ & \|(\Delta x, S_k^{-1} \Delta s)^t\|_2 \leq \Delta_k \\ & \Delta s \geq -\tau s. \end{array} \tag{3.51}$$

The tangential subproblem is solved with a preconditioned conjugate gradient method, following the Steihaug approach. More details about the solution of the quadratic subproblems can be found in [43].
The merit function whose decrease determines if the computed step is accepted or rejected and the change of the trust region radius in the next iteration is

$$\phi(x, s; \nu) = f(x) - \rho \sum_{i=1}^{m} \ln s_i + \nu \|(g_1(x), \quad g_2(x) + s\|_2.$$

At each iteration of the Knitro algorithm, three systems of linear equations hav to be solved, requiring the factorization of only one matrix, whose form is

$$\begin{pmatrix} I & \hat{A} \\ \hat{A}^t & 0 \end{pmatrix}$$

where the $\hat{A}$ indicates the jacobian matrix of the constraints $(\nabla g_1, \nabla g_2)$. The first version of the Knitro code performed the direct factorization of this matrix, by employing the routine MA27 of the Harwell Subroutine Library, while more recent version allow the choice of the conjugate gradient method as inner iterative solver. The two versions of the code are referred as Knitro-Direct and Knitro-Iterative respectively.
The global convergence of Knitro has been proved in [17].

3.5.3 IPOPT

The last interior–point algorithm that we consider is IPOPT [71, 73], which is a barrier algorithm applied to the following nonlinear problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_1(x) = 0 \\ & x \geq 0 \end{array} \tag{3.52}$$

which can also describe a nonlinear programming problem formulated as in (3.4).

The measure of the optimality condition violation for the barrier subproblems is the function

$$\mathcal{M}(x, \lambda, w; \rho) = \max\left\{ \frac{\|\nabla f(x) + \nabla g_1(x)^t\lambda - w\|_\infty}{s_d}, \|g_1(x)\|_\infty, \frac{\|Wx - \rho e_n\|_\infty}{s_c} \right\}$$

where λ and w are the multipliers of the equality and inequality constraints and s_d, s_c are two scaling factors used in order to adapt the termination rule to the critical cases, when the gradients of the constraints are nearly linearly dependent. Scaling techniques are also applied in the computation of the direction for the solution of the barrier problem, which is performed by solving the following system in condensed form whose matrix is

$$\begin{pmatrix} Q_k + X_k^{-1}W_k & \nabla g_1(x_k) \\ \nabla g_1(x_k)^t & 0 \end{pmatrix}. \tag{3.53}$$

The IPOPT code provides also a regularization of the matrix by modifying the diagonal entries. The factorization, after the scaling and the regularization, is performed with the MA27 subroutine.

After the computation of the direction, the step size is shortened so that the new iterate is feasible, by allowing a different step size for the variable w: this feature can lead to a modification of the direction.

Then, in the framework of filter methods, the new point is accepted if it produces a sufficient decrease of the barrier merit function

$$\varphi_\rho(x) = f(x) - \rho\sum_{i=1}^{n} \ln x_i$$

or a sufficient progress towards the minimization of the constraints violation $\theta(x) = \|g_1(x)\|$.

The acceptable points are of two types: a φ-type point is computed when the constraint violation in the current point is less then a fixed constant and

Prob.	LOQO 6.2		Knitro direct		Knitro iterative		IPOPT	
	it	sec	it	sec	it	sec	it	sec
6.1.3-199	39	108	19	72	12	94	25	276
6.1.3-299	*	*	20	278	12	322	20	1143
6.1.3-399	*	*	21	786	15	1020	28	3618
6.1.3-499	*	*	22	1585	14	1754	22	7374
6.1.3-599			*	*	16	2876	m	m
6.2.6-99	131	51	34	17	45	33	91	29
6.2.6-199	143	427	44	180	41	263	74	302
6.2.6-299	*	*	41	674	101	1637	113	1670
6.2.6-399	*	*	40	1829	109	4693	90	3518
6.2.6-499	*	*	42	3498	*	*	88	7034

Table 3.1: Comparison on the test problems 6.1.3 and 6.2.6 of Chapter 6: the symbol '*' indicates a failure of the algorithm, while 'm' means that the code ran out the available memory.

it must satisfy the Armijo rule for the merit function φ_ρ and a "switching condition" which guarantees that the direction is a descent direction for φ_ρ but prevents the orthogonality to the vector $\nabla\varphi_\rho$.
The other type of acceptable point is such that one of the following inequality holds

$$\begin{aligned} \theta(x_{k+1}) &\leq (1-\gamma_\theta)\theta(x_k) \\ \varphi_\rho(x_{k+1}) &\leq \varphi(x_k) - \gamma_\varphi\theta(x_k) \end{aligned}$$

where γ_θ and γ_φ are two positive constants chosen in $(0,1)$.
In order to obtain acceptable points, the algorithm performs "second–order corrections", whose aim is to reduce the violation of the constraints by applying a Newton–type method on them
The corrected step is computed as the solution of a system whose matrix is the matrix (3.53), thus only one factorization is needed for each iteration.
Finally, the perturbation parameter is updated and a new iterate is performed.
A more detailed description of the IPOPT algorithm is given in [71], while the convergence theorems and an extensive numerical experimentation can be found in [73].

Chapter 4

Description of the algorithm

The algorithm proposed in this dissertation is a line–search inexact Newton interior–point method (see Section 3.4), solving a sequence of perturbed Newton equations.

The search direction can be computed by applying direct or iterative methods to the perturbed Newton equation, which is not considered in its full version (3.26)–(3.29) but in the reduced or condensed form as explained in Section 3.3.1.

The iterative solvers chosen for the computation of the search direction are the Hestenes method, a new contribution in this thesis, and the preconditioned conjugate gradient method, and they are discussed in sections 4.2.2 and 4.2.3.

The preconditioned conjugate gradient method for the solution of linear systems of the form (3.33) has been proposed from several authors (for example [1, 50, 51]) and it is employed in many optimization codes, for example [18, 71]. The novelty of the approach presented in this thesis consists in its association with the inexact Newton framework and in the implementation, as explained in Section 4.2.3.

Furthermore, an efficient solver for systems with a quasidefinite matrix including the possibility of performing a dynamical regularization is proposed here.

The global convergence of the algorithm is ensured by a line–search strategy which can be extended to the new nonmonotone case.

In particular, by means of the theorems in Section 2.2.5, the whole algorithm can be generalized in a nonmonotone way allowing different choices not only for the backtracking rule, but also for the perturbation parameter and for the inner stopping criterion, as explained in Section 4.3.

4.1 The interior–point iteration

The algorithm follows a primal–dual approach, thus at each outer iteration the linear system (3.15) has to be solved. The general framework is summarized in the Scheme 3.4, and in this section we deal with the interior–point iteration, while the solution of the perturbed Newton equation is the subject of the next sections.

The stopping criterion
The merit function chosen is the nonlinear least squares function (3.18) and the iterations stop when $\|H(v_k)\|$ reach a fixed tolerance *tol*. For the stopping criterion we have also considered the quantity

$$\frac{|gap|}{1+|gap|} \tag{4.1}$$

where *gap* is the difference between the objective function of the primal problem (1.1) and the objective function of the following dual problem [1], whose variables are x, λ and w:

$$\begin{array}{ll} \min & f(x) - \lambda^t g_1(x) - w^t g_2(x) - \nabla f(x)^t x + (\lambda^t \quad w^t) \begin{pmatrix} \nabla g_1(x) \\ \nabla g_2(x) \end{pmatrix} x \\ s.t. & \nabla f(x) - (\nabla g_1(x) \quad \nabla g_2(x)) \begin{pmatrix} \lambda \\ w \end{pmatrix} = 0. \end{array}$$

Thus the stopping criterion can be written as follows:

$$\left\{ \begin{array}{l} \|H(v_k)\| \le tol \\ \text{or} \\ \frac{|gap|}{1+|gap|} \le tol \end{array} \right. \tag{4.2}$$

The parameters At the beginning of the algorithm, the constants τ_1 and τ_2 used for the centrality condition (3.21) and (3.22) are initialized as

$$\begin{array}{l} \tau_1 = \min(0.99, 10^{-7} \min(S_0 W_0 e_m)/0.5[s_0^t w_0/m] \\ \tau_2 = 10^{-7} s_0^t w_0 / \|H_1(v_0)\| \end{array} \tag{4.3}$$

(see also [35]), then two safeguard values δ_{max} and σ_{max} for the forcing terms δ_k and σ_k respectively are chosen. The aim of this settings is to give an upper

[1] We recall that the dual formulation of the problem (1.1) must lead to the same KKT conditions, see also [70]

bound for the choice of the forcing terms such that $\delta_k+\sigma_k \leq \delta_{max}+\sigma_{max} < 1$. Our choices are

$$\delta_{max} = \frac{0.8}{1+0.5\frac{\tau_2\sqrt{2}}{\min(1,\tau_2)}} \tag{4.4}$$

$$\sigma_{max} = \frac{\delta_{max}0.5\tau_2\sqrt{2}}{\min(1,\tau_2)}\cdot 1.1 \tag{4.5}$$

and they contribute to the convergence of the algorithm, as showed in Theotem 5.3 of the next chapter.

The forcing terms and the perturbation parameter
In the interior–point iteration, we must choose the forcing term $\delta_k \in [0, \delta_{max}]$ and $\sigma_k \in [0, \sigma_{max}]$.
In order to improve the rate of convergence of the sequence, we should choose $\delta_k, \sigma_k \approx \|H(v_k)\|$, as suggested by the convergence results for the inexact Newton method in Chapter 2.
We follow this approach, and we introduce some control on the size of the forcing term in order to avoid that it becomes too small, but accelerating the convergence when the iterates are close to the solution.
The initial value δ_0 is set equal to $\min(\delta_{max}, 0.8\|H(v_k)\|)$, while, tacking into account considerations above, the settings for all the successive iterations are

$$\min(\delta_{max}, \max(5.0\cdot 10^{-5}, \|H(v_k)\|, 0.5\|H_1(v_k)\|/\|H_1(v_{k-1})\|)) \tag{4.6}$$

if $\|H(v_k)\| < 10^{-3}$, and

$$\min(\delta_{max}, \max(5.0\cdot 10^{-5}, \min(0.999\delta_{k-1}, \|H(v_k)\|, 0.5\|H_1(v_k)\|/\|H_1(v_{k-1})\|))) \tag{4.7}$$

otherwise.
The forcing term σ_k is chosen in $[0, \sigma_{max}]$ of the same order than δ_k, as

$$\sigma_k = \min(\sigma_{max}, \max(1.1\cdot\frac{0.5\tau_2\delta_k\sqrt{2}}{\min(1,\tau_2)}, 0.01\|H(v_k)\|))$$

and this choice contributes to the convergence of the algorithm, as we will show in the next chapter, and it also influences the convergence rate.
After the forcing terms, the user is allowed to choose the perturbation parameter as $\rho_k = \sigma_k\mu_k$, where

$$\mu_k = \mu_k^{(1)} = \frac{s_k^t w}{m} \tag{4.8}$$

or

$$\mu_k = \mu_k^{(2)} = \frac{\|H(v_k)\|}{\sqrt{m}}. \tag{4.9}$$

Then, the search direction Δv_k is computed such that the property (3.42) holds and the damping parameter is initially set equal to 1.

Feasibility
We recall that the feasibility conditions can be expressed as follows: find α_k such that $s_k + \alpha_k \Delta s_k > 0$ and $w_k + \alpha_k \Delta w_k > 0$.
Thus, we compute α_k as in (3.16), where γ represents the percentage of movement to the boundary and it can be chosen adaptively by means of the following rule [3]:

$$\gamma = \max(0.8, \min(0.9995, 1 - 100 s_k^t w)) \tag{4.10}$$

if the step length has been reduced by the formula (3.16),

$$\gamma = \max(0.8, 1 - 100 s_k^t w) \tag{4.11}$$

if the full direction Δv_k does not bring out of the feasible region.
The meaning of (4.10) and (4.11) is to allow the sequence to be close to the boundary of the feasible region only when we are sufficiently close to the solution. This adaptive choice has been shown to have good performances when we use the iterative inner solvers for the Newton equation presented in the next sections, while for the direct approach it is sufficient to fix the value of γ to 0.995.

The centrality conditions
After the feasibility conditions, also the centrality conditions have to be checked and the damping parameter α_k is reduced by a factor of 0.5 until (3.21) and (3.22) are satisfied.

The line–search
Finally, we must guarantee a sufficient decrease of the merit function $\phi(v)$, by using a backtracking strategy along Δv_k until the condition (3.44), with $\beta = 10^{-4}$, is satisfied.

For sake of completeness we report the scheme of the algorithm which resumes all the topics above.

Algorithm 4.1

- Choose an initial guess v_0 s.t. $(s_0, w_0) > 0$;

```
- Choose the parameters
              $\tau_1, \tau_2$ as in (4.3)
              $\sigma_{max}, \delta_{max}$ as in (2.53) and (4.5)
- For $k = 0, 1, 2, ...$ until (4.2) is satisfied
    - Choose the forcing terms $\delta_k$ and $\sigma_k$ as in (4.6) and (4.7)
    - Choose $\mu_k \in \left\{\mu_k^{(1)}, \mu_k^{(2)}\right\}$
    - Set $\rho_k = \sigma_k \mu_k$
    - Compute a direction $\Delta v_k$ such that
              $\|\bar{r}_k\| \leq \delta_k \|H(v_k)\|$
    - Set $\alpha_k = 1$
    - Feasibility: compute $\alpha_k$ as in (3.16) where $\gamma$ is defined in (4.10)
      –(4.11)
    - Centrality:
      While $f^I(\alpha_k) < 0$
        - Set $\alpha_k \leftarrow 0.5\alpha_k$
      While $f^{II}(\alpha_k) < 0$
        - Set $\alpha_k \leftarrow 0.5\alpha_k$
    - Backtracking:
      While $\|H(v_k + \alpha_k \Delta v_k)\| > (1 - \beta(1 - \delta_k - \sigma_k)\alpha_k)\|H(v_k)\|$
        - Set $\alpha_k \leftarrow 0.5\alpha_k$
    - Update the iterate $v_{k+1} = v_k + \alpha_k \Delta v_k$.
```

4.2 The search direction

The scheme of the Algorithm 4.1 shows that the main computational effort is spent in the approximate solution of the perturbed Newton equation, since the other issues do not give a significant contribution to the complexity of the algorithm. Thus, the choice of the more suitable method for the computation of the search direction Δv_k at each iteration has a crucial importance for the effectiveness of the whole algorithm. In the following sections we consider three different inner solvers, one direct method and two iterative methods, which lead to four versions of the Algorithm 4.1, depending on the choice of the inner solver and on its implementation.

4.2.1 The direct approach

As observed in Section 3.3.1, the perturbed Newton equation in its full form (3.26)-(3.29) is not symmetric and it has no suitable properties. Thus, we consider the reduced form (3.31). Actually, in this direct approach, we prefer a slightly different reduction of the system, which is an intermediate step between the reduced form (3.31) and the condensed form (3.33). Namely, we perform the substitution (3.32) only for the components of the vector Δw corresponding to a box constraint. Thus we obtain a system of the same form of (3.31), but the matrices involved are different. In particular, the left–up block of the matrix of the system is given by the sum of the hessian matrix of the lagrangian Q plus a positive semidefinite diagonal matrix F whose nonzero entries correspond to the components of the variable x which are bounded. Thus, the computation of the matrix F does not require any computational. After the partial substitution, the size of the system is $n + neq + 2(m - b)$, where b indicates the number of the box constraints. The matrix of this partially reduced system is a symmetric but not definite matrix, thus the factorization has to be performed by means of a symmetry preserving algorithm as the Bunch–Parlett [16] triangular factorization.

4.2.2 The iterative approach: Hestenes method

Consider now the perturbed Newton equation in the condensed form (3.33). We recall that, if B^t is a full row–rank matrix, the coefficient matrix of (3.33)

$$M = \begin{pmatrix} A & B \\ B^t & 0 \end{pmatrix}$$

is nonsingular if and only if the matrix A is nonsingular on the null space of B^t ([42]), i.e. Z^tAZ is a nonsingular matrix, where Z is the $n \times (n - neq)$ matrix such that $B^tZ = 0$ and $Z^tZ = I$. In particular, a sufficient condition for the nonsingularity of M is that the matrix Z^tAZ is positive definite (see also [54, p. 424]). This condition holds if the hessian matrix of the lagrangian function Q is positive definite on the null space of B^t. Note that this assumption is also the one required for the local SQP method ([60, p. 531]).

Setting $y_1 = \Delta x$ and $y_2 = \Delta\lambda$, the system (3.33), can be viewed as the Lagrange necessary conditions for the minimum point of the following quadratic problem

$$\begin{array}{ll} \min & \frac{1}{2} y_1^t A y_1 - c^t y_1 \\ \text{s.t.} & B^t y_1 - q = 0. \end{array}$$

This quadratic problem can be solved efficiently by Hestenes' multipliers scheme ([46, p. 308]), that consists in updating the dual variable by the rule

$$y_2^{(j+1)} = y_2^{(j)} + \chi(B^t y_1^{(j)} - q),$$

where χ is a positive parameter (penalty parameter) and $y_1^{(j)}$ minimizes the augmented lagrangian function of the quadratic problem

$$\mathcal{L}_\chi(y_1, y_2) = \frac{1}{2} y_1^t A y_1 - y_1^t c + y_2^t (B^t y_1 - q) + \frac{\chi}{2}(B^t y_1 - q)^t (B^t y_1 - q).$$

This means that $y_1^{(j)}$ is the solution of the linear system of order n

$$(A + \chi BB^t) y_1 = -By_2^{(j)} + c + \chi Bq \qquad (4.12)$$

Note that, since B^t has full row–rank, the null space of BB^t is equal to the null space of B^t, then the matrix A is positive definite on the null space of BB^t. Then, it is immediate the following theorem.

Theorem 4.1 ([54, p. 408]) There exists a positive parameter χ^* such that for all $\chi > \chi^*$, the matrix $A + \chi BB^t$ is positive definite.

This result enables us to solve the system (4.12) by applying a Cholesky factorization.

In order to choose the parameter χ, we observe that, for any $x \neq 0$, we must have $x^t(A + \chi BB^t)x > 0$. When $B^t x = 0$, we have $x^t A x > 0$. If $B^t x \neq 0$, $x^t BB^t x > 0$. Then, it follows that

$$\chi > \max(0, \max_{x \notin \mathcal{N}(B^t)} \frac{-x^t A x}{x^t BB^t x})$$

Since $\|A\| \geq (-x^t A x)/(x^t x)$ for any natural norm and also for the Frobenius norm $\|\cdot\|_F$, and $x^t BB^t x/(x^t x) \geq \tau_{min}$, where τ_{min} is the minimum nonzero eigenvalue of BB^T or of $B^T B$, we can choose as χ the following value:

$$\chi > \frac{\|A\|_F}{\tau_{min}}$$

In general it is difficult to determine an estimate of τ_{min}. Numerical evidence shows that a good approximation of τ_{min} is $\min(1, t_{min})$, where t_{min} is the minimum diagonal entry of the matrix $B^T B$, although $t_{min} \geq \tau_{min}$. Furthermore, in order to avoid that the value of χ is too small (the matrix

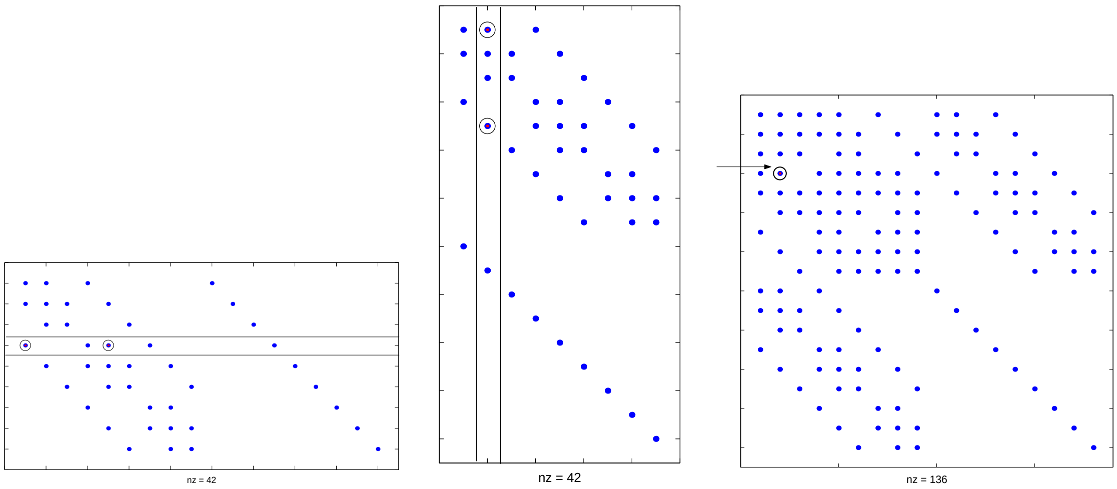


Figure 4.1: Preprocess phase: save the indices of the nonzero contribution of the scalar product

is not positive definite) or too large (too ill–conditioned system), it is convenient to use safeguards. In the numerical experiments of the last chapter, the following value of χ produced good results:

$$\chi = \min(\max(10^7, \frac{\max\{\|A\|_F, 1\}}{\min\{t_{min}, 1\}}), 10^8). \tag{4.13}$$

Now, we discuss the implementation of the method. We assume that the hessian matrix Q of the lagrangian function and the jacobian matrix B^T of the equality constraints are stored in a *column compressed format* ([64]). Then, at any step of the IP method, the implementation of Hestenes' multipliers scheme requires the computation of the matrix $T = A + \chi BB^t$ and its Cholesky factorization $T = L_n L_n^t$. The other operations related to each iteration (i. e. sparse matrix–vector products $B(-y_2^{(j)} + \chi q)$ and $B^t y_1^{(j)}$ and solution of the triangular systems equivalent to (4.12)) have a negligible computational complexity. In order to execute only necessary operations to form T, it is convenient to execute a preprocess procedure that builds a data structure which stores the indices of the nonzero entries of T. For any nonzero entry t_{ij} of T, in the same data structure we also store the pairs of indices of the elements of C, C^t, B and B^t that give a nonzero contribution in the scalar product forming the entry, as depicted in Figure 4.1.

The preprocess routine also computes the symbolic Cholesky factorization of the sparse, symmetric and positive definite matrix T. To exploit the sparsity of T, its factorization can be obtained by a very efficient Fortran package (version 0.3) of Ng and Peyton (included in the package LIPSOL, downloadable from *www.caam.rice.edu/˜zhang/lipsol*). This package *a priori* computes the symbolic factor of T (i.e. the indices of the nonzero entries of L_n and the information to form these entries), using the multiple minimum degree ordering of Liu to minimize the fill–ins in L_n and the supernodal block factorization to take advantage of the presence of the cache memory in modern computer architectures ([59]). The *a priori* procedure of Liu for reordering of T and the computation of its symbolic factorization can be executed only one time in the preprocess routine.

In conclusion, the time for solving an NLP problem by the IP method combined with Hestenes' multipliers method is subdivided in two part, the *preprocess time* and the time for computing the solution (*solution time*). We observe that the preprocess time is dependent on the strategy used to perform the matrix–matrix products needed in the method for computing T.

4.2.3 The iterative approach: PCG

A different approach for solving the inner system arising at each step of an IP scheme uses a Preconditioned Conjugate Gradient (PCG) method, as suggested in [50] (see also [32], [29], [51], [7]). As in the previous section, we propose to solve the condensed form of the system (3.33) instead of the reduced form (3.31), but, unlike as it arises for the Hestenes' multipliers scheme, in this case we can avoid to explicitly compute the matrix $A = Q + CS^{-1}WC^T$. Indeed, at any step of the PCG scheme, the matrix A is required only in the matrix–vector product $t = Mp$, where

$$M = \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}, \qquad p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}, \qquad p_1 \in \mathbb{R}^n, \quad p_2 \in \mathbb{R}^{neq}.$$

The product Mp can be executed by sparse matrix–vector products only, using a temporary array $\hat{t}$ to store the partial results:

$$\begin{array}{rcl} t_1 & \leftarrow & C^t p_1 \\ \hat{t} & \leftarrow & S^{-1} W t_1 \\ t_1 & \leftarrow & C\hat{t} \\ t_1 & \leftarrow & t_1 + Qp_1 + Bp_2 \\ t_2 & \leftarrow & B^t p_1 \end{array}$$

As preconditioner in the PCG scheme, we can consider the indefinite preconditioner in [50]:

$$\bar{M} = \begin{pmatrix} \bar{A} & B \\ B^t & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^t\bar{A}^{-1} & I \end{pmatrix} \begin{pmatrix} \bar{A} & 0 \\ 0 & -B^t\bar{A}^{-1}B \end{pmatrix} \begin{pmatrix} I & \bar{A}^{-1}B \\ 0 & I \end{pmatrix} \tag{4.14}$$

where we assume that $\bar{A}$ is a positive diagonal approximation of A.
For sake of completeness, we report the main theoretical results about the preconditioner (4.14)(for further details and proofs of the following theorems, see [50]).

Theorem 4.2 If $\bar{A}$ is a positive definite matrix , then the matrix $M\bar{M}^{-1}$ has at least $2 \cdot neq$ unit eigenvalues.
If $A\bar{A}^{-1} - I$ is a nonsingular matrix, then only neq linearly independent eigenvectors corresponding to these eigenvalues exist; the other eigenvalues of the matrix $M\bar{M}^{-1}$ are exactly the eigenvalues of the matrix $Z^tAZ(Z^t\bar{A}Z)^{-1}$.
If Z^TAZ is a positive definite matrix, all the eigenvalues of the matrix $M\bar{M}^{-1}$ are positive.
Moreover, if $vZ^t\bar{A}Zv = v^tZ^tAZv$ for some $v \in \mathbb{R}^n$, then all the eigenvalues of the matrix $M\bar{M}^{-1}$ are included in the interval determined by the extremal eigenvalues of the matrix $Z^tAZ(Z^t\bar{A}Z)^{-1}$.

Theorem 4.3 Consider the PCG method with preconditioner (4.14), where the matrix $\bar{A}$ is positive definite, applied to the system

$$M \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

If a breakdown does not occur, then we obtain the solution $\begin{pmatrix} v_1^* \\ v_2^* \end{pmatrix}$ after at most $n - neq + 2$ iterations.

Theorem 4.4 Let the matrix Z^tAZ be positive definite. Consider the PCG method with the preconditioner (4.14), where $\bar{A}$ is a positive definite matrix, applied to the system (3.33), starting with the initial point $v_1^0 = \bar{A}^{-1}B(B^t\bar{A}^{-1}B)^{-1}y_2$, $v_2^0 = 0$. The PCG method finds the solution of the system after at most $n - neq$ iterations and the following condition holds

$$\|v_1^i - v_1^*\| \leq 2\sqrt{k} \left(\frac{1 - \sqrt{k}}{1 + \sqrt{k}} \right)^i \|v_1^0 - v_1^*\| \tag{4.15}$$

where k is the spectral condition number of $Z^tAZ(Z^t\bar{A}Z)^{-1}$.

In the implementation of the PCG scheme, we can choose the diagonal matrix $\bar{A} = diag(\bar{a}_{ii})$ as follows

$$\bar{a}_{ii} = \begin{cases} a_{ii} = q_{ii} + \sum_{j=1}^{m} c_{ij}^2 w_j / s_k & \text{if } a_{ii} > 10^{-8} \\ 1.5 \cdot 10^{-8} & \text{otherwise.} \end{cases} \quad i = 1, ..., n \tag{4.16}$$

At any step of the PCG scheme, we have to compute the solution of the system

$$\bar{M} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{4.17}$$

We can determine the solution of this system in two different ways that produce a very different performance, especially for large scale problems.

In the first case (IP-PCG1), at the beginning of the PCG method we compute the symmetric positive definite matrix $T = B^t \bar{A}^{-1} B$ and its Cholesky factorization $T = L_{neq} L_{neq}^t$; then, taking into account of $\bar{M}^{-1}$ from (4.14), the solution of (4.17) can be determined by the following procedure

$$\begin{array}{lcl} z_1 & \leftarrow & \bar{A}^{-1} r_1 \\ z_2 & \leftarrow & r_2 - B^t z_1 \\ t_2 & \leftarrow & -L_{neq}^{-1} z_2 \\ z_2 & \leftarrow & L_{neq}^{-T} t_2 \\ z_1 & \leftarrow & z_1 - \bar{A}^{-1} B z_2 \end{array}$$

where t_2 is an neq–vector used to store the partial products.

As in the implementation of Hestenes' method, a preprocess routine can build a data structure that stores the information needed to compute the nonzero contribution to each nonzero scalar product. The preprocess routine can also determine the minimum degree reordering of the matrix T and its symbolic Cholesky factor. For these last tasks and for computing the elements of L_{neq}, we can use the package of Ng and Peyton. With this approach, the preprocess phase is less expensive than that of the IP method combined with the Hestenes multipliers' scheme, even for NLP problems with equality and box constraints. Indeed, we have to compute the entries of the matrix T and to solve systems with T as coefficient matrix, whose size is neq instead of the size n of the matrix $A + \chi B B^t$, where $neq < n$.

Now, we discuss the other way to implement the PCG algorithm that avoids the computation of the matrix–matrix product $B^t \bar{A}^{-1} B$.

We call this second version of the PCG algorithm IP-PCG2.

We observe that the matrix $\bar{M}$ can be factorized in a Cholesky–like form

$$L_{n+neq} D L_{n+neq}^t, \tag{4.18}$$

where L_{n+neq} is a lower triangular matrix with diagonal entries equal to one and D is a nonsingular diagonal matrix. In order to reduce the fill–in in the lower triangular factor, we can perform a minimum degree reordering of the matrix $\bar{M}$. But, it is not assured that the symmetrically permuted matrix $P\bar{M}P^t$ can be factorized in the Cholesky–like form.
Nevertheless, we can obtain a factorization in the form (4.18) if we use for the matrix $\bar{M}$ the regularization technique described in [1]; in other words, instead of using the preconditioner $\bar{M}$, we compute the factorization of

$$\bar{\bar{M}} = \bar{M} + \begin{pmatrix} R_1 & 0 \\ 0 & -R_2 \end{pmatrix}$$

where R_1 and R_2 are non negative diagonal matrices such that $P\bar{\bar{M}}P^t$ admits a factorization of the form (4.18). The computation of R_1 and R_2 can be obtained during the factorization procedure. If a pivot d_i is too small ($|d_i| < 10^{-15} \max_{j<i} |d_j|$), we put $d_i = \sqrt{\epsilon}$ if $1 \leq i \leq n$, or $d_i = -\sqrt{\epsilon}$ if $n+1 \leq i \leq n+neq$, where ϵ is the machine precision.
The dynamic computation of the elements of R_1 and R_2 reduces the perturbation to a minimum. This approach is used in [7] for linear and quadratic programming problems with equality and box constraints.
The Cholesky–like factorization of $\bar{\bar{M}}$ can be obtained by a modification of the Ng and Peyton package. In particular, we modify the subroutine PCHOL such that we compute $L_{n+neq}DL^T_{n+neq}$ with diagonal elements of L_{n+neq} equal to 1.
Consequently, it is necessary to construct suitable subroutines (MMPYM and SMXPYM) to update the blocks of the factor L_{n+neq}, and to modify the subroutine BLKSVT for the computation of the solution of the system

$$L_{n+neq}DL^t_{n+neq}z = r.$$

The routines for performing the minimum degree reordering, for determining the supernodes and for the computation of the symbolic factor are unchanged. Consequently, the effectiveness of the package of Ng and Peyton due to a suitable use of the cache memory is maintained. This new package, called BLKFCLT, is downloadable from *http://dm.unife.it/blkfclt.*

At the iterate k, the termination rule for both the iterative solvers is the adaptive stopping rule (3.43) which makes the approximate solution of the system (3.33) an inexact Newton step at the level $\delta_k + \sigma_k$.

4.3 The nonmonotone version

Following the same procedure described in Section 3.4, we can extend the interior–point method to nonmonotone choices, in the context of the nonmonotone inexact Newton method presented in Section 2.2.5. Indeed, by allowing the choice of the parameter μ_k in the interval

$$\mu_k \in \left[\frac{{s_k}^t w_k}{m}, \frac{\|H(v_{\ell(k)})\|}{\sqrt{m}} \right], \tag{4.19}$$

if the direction Δv_k computed by approximately solving the system (3.33) satisfies the condition

$$\|\bar{r}_k\| \leq \delta_k \|H(v_{\ell(k)})\|, \tag{4.20}$$

then, such direction is a nonmonotone inexact Newton step at the level $\delta_k + \sigma_k$. Moreover we can also include a nonmonotone backtracking rule

$$\|H(v_k + \alpha_k \Delta v_k)\| \leq (1 - \alpha\beta(1 - (\delta_k + \sigma_k))\|H(v_{\ell(k)})\|. \tag{4.21}$$

In summary, we can allow nonmonotone choices on three crucial issues: on the perturbation parameter, on the inner adaptive stopping criterion and on the backtracking rule. We observe that the first two choices influence the direction itself, while a less restrictive backtracking rule allows to retain larger stepsizes than in the monotone case. The resulting algorithm is a nonmonotone Newton interior–point algorithm, whose convergence properties are investigated in the next chapter.

Chapter 5

Convergence analysis

In this chapter we state the convergence theorems for the Algorithm 4.1. The hypotheses under which the convergence is proved, are quite similar to the one in [35], but here the convergence properties of the inexact Newton methods can be exploited in the proof. The line of the proof is the following: supposing that the sequence $\{H(v_k)\}$, where the iterates v_k are generated by the Algorithm 4.1, is bounded away from zero yields a contradiction. Before the convergence result, under the hypotheses made, we prove the boundedness of the sequence $\{v_k\}$. Furthermore, we show that, supposing that $\{H(v_k)\}$ is bounded away from zero, then $\{\Delta v_k\}$ is bounded, and the damping parameter, after the reductions due to the feasibility requirement, to the centrality conditions and to the backtracking technique, is uniformly bounded away from zero. This is crucial in order to apply the inexact Newton theory in this context.
Finally, a slightly modification of the hypotheses and of the proof allows us to prove the convergence also in the nonmonotone case.

5.1 Convergence theorems

First of all we define a subset of $\mathbb{R}^{n+neq+2m}$ which contains all the iterates of the sequence generated by the Algorithm 4.1.
Given $\epsilon \geq 0$, we define the set $\Omega(\epsilon)$ as follows:

$$\Omega(\epsilon) = \{v : 0 \leq \epsilon \leq \|H(v)\|^2 \leq \|H(v_0)\|^2, \text{ s. t. } (3.21) \text{ and } (3.22) \text{ hold}\}. \tag{5.1}$$

The set $\Omega(\epsilon)$ is a closed set.
Indeed, let v_* be an accumulation point of the sequence $\{v_k\}$, where $v_k \in$

$\Omega(\epsilon)$. The definition of continuity of $\Phi(v) = \|H(v)\|_2$ implies that

$$\lim_{k\to\infty} \Phi(v_k) = \Phi(\lim_{k\to\infty} v_k) = \Phi(v_*)$$

and since $\Phi(v_k) \leq \Phi(v_k)$ for all k, we have that

$$\lim_{k\to\infty} \Phi(v_k) \leq \lim_{k\to\infty} \Phi(v_0)$$

i.e. $\Phi(v_*) \leq \Phi(v_0)$.
Analogously, we have

$$\lim_{k\to\infty} \frac{\min_{i=1,\ldots,m}(S_k W_k e_m)}{(s_k^t w_k)/m} = \frac{\min_{i=1,\ldots,m}(S_* W_* e_m)}{(s_*^t w_*)/m} \geq \frac{\tau_1}{2}$$

and

$$\lim_{k\to\infty} \frac{s_k^t w_k}{\|H_1(v_k)\|} = \frac{s_*^t w_*}{\|H_1(v_*)\|} \geq \frac{\tau_2}{2}$$

thus, $\boldsymbol{v}_*$ is a point of $\Omega(\epsilon)$.

Moreover it is straightforward to observe that $v_k \in \Omega(0)$, since the backtracking condition (3.44) yields

$$\|H(v_k)\| \leq \|H(v_0)\|.$$

Let assume that the following conditions hold ([31], see also [35]):

C1 in $\Omega(0)$, $f(x)$, $g_1(x)$, $g_2(x)$ are twice continuously differentiable; the gradients of the equality constraints are linearly independent and $H_1'(v)$ is Lipschitz continuous;

C2 the sequences $\{x_k\}$ and $\{w_k\}$ are bounded;

C3 in any compact subset of $\Omega(0)$ where s is bounded away from zero, the matrix $H'(v)$ is nonsingular.

In general, in literature, the condition C3 is replaced by a sufficient condition to assure that, at each iterate k, there exists a unique solution of the perturbed Newton equation, for example the conditions C3' or C3" in Section 3.3.1.
The boundedness of the sequence $\{x_k\}$ can be assured by enforcing box constraints $-l_i \leq (x_k)_i \leq l_i$ for sufficiently large $l_i > 0$, $i = 1, ..., n$.

Theorem 5.1 Let $\{v_k\}$ be a sequence generated by the Algorithm 4.1 and assume that the hypotheses C1, C2 and C3 hold. Then, the sequences $\{\lambda_k\}$ and $\{s_k\}$ are bounded.

Proof.
From the assumptions C1 and C2 and from the definition of the vector $H(v_k)$, at each iteration k we have

$$\|\mathcal{L}_x(x_k, \lambda_k, w_k, s_k)\| = \|\nabla f(x_k) + B_k\lambda_k + C_k w_k\| \leq \|H(v_k)\| \leq \|H(v_0)\|,$$

where B_k and C_k indicate the matrix C and B defined in (3.25) evaluated in x_k. Then, since B_k is a full column–rank matrix, we can write

$$\lambda_k = (B_k^t B_k)^{-1} B_k^t(-\nabla f(x_k) - C_k w_k + \mathcal{L}_{\S_k})$$

and for C1 and C2 the sequence $\{\lambda_k\}$ is bounded.
Furthermore,

$$\begin{aligned} \|s_k\| &\leq \|s_k - g_2(x_k)\| + \|g_2(x_k)\| \\ &\leq \|H(v_k)\| + \|g_2(x_k)\|. \end{aligned}$$

Then the sequence $\{s_k\}$ is bounded. □

The previous theorem shows that, if the hypotheses C1 and C3 hold, then the hypothesis C2 is sufficient to ensure the boundedness of the whole sequence $\{v_k\}$.
The next theorem claims that, if the sequence $\|H(v_k)\|$ is bounded away from zero, then the sequences $\{s_k\}$ and $\{w_k\}$ are componentwise bounded away form zero, the norm of the inverse of the jacobian matrix and also the sequence $\{\|\Delta v_k\|\}$ are bounded.

Theorem 5.2 If the sequence $\{v_k\} \in \Omega(\epsilon)$, with $\epsilon > 0$, then

(a) $s_k^t w_k$, $(s_k)_i(w_k)_i$, $i = 1, \ldots m$, are bounded above and below away from zero for any $k \geq 0$; $\|H_1(v_k)\|$ is bounded above for any $k \geq 0$;

(b) if C1 and C2 hold, s_k and w_k are componentwise bounded away from zero;

(c) if C1, C2 and C3 hold, then the sequence of matrices $\{H'(v_k)^{-1}\}$ is bounded;

(d) if C1, C2 and C3 hold, then the sequence $\{\Delta v_k\}$ is bounded.

Proof.
(a) The above boundedness of $(s_k)_i(w_k)_i$, $i = 1, \ldots, m$, and $s_k^t w_k$ follows from the inequality

$$\begin{aligned}(s_k)_i(w_k)_i \leq s_k^t w_k = \|S_k W_k e_m\|_1 &\leq \sqrt{m}\|S_k W_k e_m\| \\ = \sqrt{m}\|H(v_k)\| &\leq \sqrt{m}\|H(v_0)\|. \end{aligned} \tag{5.2}$$

Furthermore in $\Omega(\epsilon)$, $\epsilon > 0$, from (3.21) and (3.22), we have $(s_k)_i(w_k)_i > 0$ for any $k \geq 0$ and $i = 1, \ldots, m$. From the inequality

$$\begin{aligned}\epsilon &\leq \|H(v_k)\| \leq \|H_1(v_k)\| + \|S_k W_k e_m\| \\ &\leq (s_k^t w_k)/(\gamma_k \tau_2) + \|S_k W_k e_m\|_1 \\ &= (1 + 1/(\gamma_k \tau_2)) s_k^t w_k, \end{aligned} \tag{5.3}$$

it follows that, for $k \geq 0$ and $i = 1, \ldots, m$,

$$s_k^t w_k \geq \epsilon \tau_2/(\tau_2 + 2), \tag{5.4}$$

and, from (3.21),

$$(s_k)_i(w_k)_i \geq \epsilon \tau_1 \tau_2/(2m(\tau_2 + 2)). \tag{5.5}$$

Finally,

$$\|H_1(v_k)\| \leq \|H(v_k)\| \leq \|H(v_0)\|.$$

(b)Since $(s_k)_i(w_k)_i$ are bounded below away from zero and s_k is bounded above for the previous theorem, for any k, it follows that w_k is bounded below away from zero. Analogously, for the same argument, s_k is bounded away from zero.
(c) Rearranging the rows and the columns of the matrix $H'(v_k)$, we obtain the following matrix

$$\begin{bmatrix} W_k & S_k & 0 & 0 \\ I & 0 & C_k{}^T & 0 \\ 0 & C_k & Q_k & B_k \\ 0 & 0 & B_k^t & 0 \end{bmatrix}. \tag{5.6}$$

Since s_k and w_k are bounded above and componentwise below away from 0, the matrix (5.6) can be factorized in the form $L_k U_k$, where L_k is the matrix

$$\begin{bmatrix} I & 0 & 0 & 0 \\ W_k^{-1} & I & 0 & 0 \\ 0 & -C_k E_k^{-1} & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \tag{5.7}$$

and U_k is the matrix

$$\begin{bmatrix} W_k & S_k & 0 & 0 \\ 0 & -E_k & C_k^t & 0 \\ 0 & 0 & F_k & B_k \\ 0 & 0 & B_k^t & 0 \end{bmatrix}, \tag{5.8}$$

with $E_k = W_k^{-1}S_k$ and $F_k = Q_k + C_k E_k^{-1} C_k^t$. Since L_k and $H'(v_k)$ are nonsingular bounded matrices, the block triangular matrix U_k is a nonsingular matrix with nonsingular and bounded diagonal blocks. The inverse of the matrix $H'(v_k)$ is given by $U_k^{-1}L_k^{-1}$. Since all the blocks of the matrices U_k^{-1} and L_k^{-1} are bounded, then $H'(v_k)^{-1}$ is also bounded in $\Omega(\epsilon)$, $\epsilon > 0$, i.e.

$$\|H'(v_k)^{-1}\| \leq \bar{M}, \tag{5.9}$$

for $v_k \in \Omega(\epsilon)$, $\epsilon > 0$ and for $k \geq 0$, with $\bar{M}$ a positive scalar.
(d) Since (3.40), Δv_k has the following form

$$\Delta v_k = H'(v_k)^{-1}(-H(v_k) + r_k + \sigma_k \mu_k \tilde{e}). \tag{5.10}$$

From (5.9), (5.1), (3.38) and (3.43), we have that

$$\|\Delta v_k\| \leq \bar{M}(1 + \delta_k + \sigma_k)\|H(v_0)\| < 2\bar{M}\|H(v_0)\|,$$

because $\delta_k + \sigma_k \leq \delta_{\max} + \sigma_{\max} < 1$. Then the proof is completed. □

In the following, we analyze the three steps for the computation of the damping parameter α_k in the algorithm 4.1, the reduction for the feasibility, for the centrality conditions and for the sufficient decrease of the merit function and we show that it is uniformly bounded away from zero in $\Omega(\epsilon)$. If we call $\alpha_k^{(1)}$ the value of the damping parameter after the reduction needed for the feasibility of the iterate, it is easy to see that $\alpha_k^{(1)}$ is bounded away from zero in $\Omega(\epsilon)$, with $\epsilon > 0$, i.e. $\alpha_k^{(1)} \geq \alpha^{(1)} > 0$, since we set

$$\alpha_k^{(1)} = \min\left\{ \min_{(\Delta s_k)_i < 0} \frac{-(s_k)_i}{(\Delta s_k)_i}, \min_{(\Delta w_k)_i < 0} \frac{-(w_k)_i}{(\Delta w_k)_i}, 1 \right\},$$

where, for any iteration k, $(s_k)_i$ and $(w_k)_i$ are bounded away from zero and $(\Delta s_k)_i$ and $(\Delta w_k)_i$ are bounded in $\Omega(\epsilon)$. with $\epsilon > 0$.
Now, we analyze the damping parameter after the reduction for the centrality conditions; the following theorem (see [39]) shows that, if the current

iterate satisfies the centrality conditions and the direction satisfies the inexact residual condition (3.43), then there exist two positive numbers $\hat{\alpha}_k^{(2)}$ and $\check{\alpha}_k^{(2)}$ such that the centrality functions $f_k^I(\alpha)$ and $f_k^{II}(\alpha)$, defined as $f^I(v_k+\alpha\Delta v_k)$ and $f^{II}(v_k+\alpha\Delta v_k)$ defined as in (3.19) and (3.20) for $v=v_k$ and $\Delta v=\Delta v_k$, are nonnegative for $\alpha\in(0,\hat{\alpha}_k^{(2)}]$ and for $\alpha\in(0,\check{\alpha}_k^{(2)}]$ respectively.

Theorem 5.3 Let $\{v_k\}$ be a sequence generated by the Algorithm 4.1; let us also assume $\sigma_k\in[\sigma_{\min},\sigma_{\max}]\subset(0,1)$ and $\delta_k\in[0,\delta_{\max}]\subset[0,1)$, and

$$\sigma_k>\delta_k(1+\gamma_k\tau_2) \tag{5.11}$$

Then, if $f_k^I(0)\geq 0$, there exists a positive number $\hat{\alpha}_k^{(2)}>0$, such that $f_k^I(\alpha)\geq 0$ for all $\alpha\in(0,\hat{\alpha}_k^{(2)}]$.
Then, if $f^{II}(0)\geq 0$, there exists a positive number $\check{\alpha}_k^{(2)}>0$, such that $f_k^{II}(\alpha)\geq 0$ for all $\alpha\in(0,\check{\alpha}_k^{(2)}]$.

Proof. Set

$$(N_k)_i=|(\Delta s_k)_i(\Delta w_k)_i-\frac{\gamma_k\tau_1}{m}\Delta s_k^t\Delta w_k|\qquad i=1,...,m.$$

The fourth block equations of the perturbed Newton equation in componentwise is

$$(s_k)_i(\Delta w_k)_i+(w_k)_i(\Delta s_k)_i=-(s_k)_i(w_k)_i+\sigma_k\mu_k. \tag{5.12}$$

Summing for any $i=1,...,m$, we have

$$s_k^t\Delta w_k+w_k^t\Delta s_k=-s_k^tw_k+m\sigma_k\mu_k. \tag{5.13}$$

Thus, for $\alpha\in(0,1]$, we can define the following quantities by

$$\begin{aligned}(f_k^I(\alpha))_i &= ((s_k)_i+\alpha(\Delta s_k)_i)\,((w_k)_i+\alpha(\Delta w_k)_i)-\\ &\quad -\frac{\tau_1\gamma_k}{m}(s_k+\alpha\Delta s_k)^t(w_k+\alpha\Delta w_k).\end{aligned}$$

By easy computation and by using (5.12) and (5.13), we can deduce

$$\begin{aligned}(f_k^I(\alpha))_i &= (1-\alpha)\left[(s_k)_i(w_k)_i-\frac{\tau_1\gamma_k}{m}s_k^tw_k\right]+\alpha\sigma_k\mu_k(1-\tau_1\gamma_k)+\\ &\quad +\alpha^2\left((\Delta s_k)_i(\Delta w_k)_i-\frac{\tau_1\gamma_k}{m}\Delta s_k^t\Delta w_k\right).\end{aligned}$$

Hence,

$$\begin{aligned}(f_k^I(\alpha))_i &= (1-\alpha)(f_k^I(0))_i+\alpha\sigma_k\mu_k(1-\tau_1\gamma_k)+\\ &\quad +\alpha^2\left((\Delta s_k)_i(\Delta w_k)_i-\frac{\tau_1\gamma_k}{m}\Delta {s_k}^t\Delta w_k\right).\end{aligned} \tag{5.14}$$

Since $f_k^I(0) \geq 0$, we have $(f_k^I(0))_i \geq 0$. Then

$$\begin{aligned}(1-\alpha)(f_k^I(0))_i &= (f_k^I(\alpha))_i - \alpha\sigma_k\mu_k(1-\tau_1\gamma_k) - \\ &\quad -\alpha^2\left((\Delta s_k)_i(\Delta w_k)_i - \frac{\tau_1\gamma_k}{m}\Delta {s_k}^t\Delta w_k\right).\end{aligned}$$

Thus

$$\begin{aligned}(f_k^I(\alpha))_i &\geq \alpha\sigma_k\mu_k(1-\tau_1\gamma_k) + \alpha^2\left((\Delta s_k)_i(\Delta w_k)_i - \frac{\tau_1\gamma_k}{m}\Delta {s_k}^t\Delta w_k\right) \\ &\geq \alpha\sigma_k\mu_k(1-\tau_1\gamma_k) - \alpha^2(N_k)_i.\end{aligned}$$

Set $N_k = \max_{i=1,\dots,m}(N_k)_i$; for any α such that

$$\alpha \geq ((1-\tau_1\gamma_k)\sigma_k\mu_k)/N_k > 0, \tag{5.15}$$

we have $f_k^I(\alpha) \geq 0$. Thus we define

$$\hat{\alpha}_k^{(2)} = \max_{\alpha\in(0,1]}\{\alpha : f_k^I(t) \geq 0,\, \forall t \leq \alpha\}.$$

We prove now the second part of the theorem.
By assumptions C1, we have that $H_1'(v)$ is Lipschitz continuous with Lipschitz constant Γ.
Set

$$M_k = \left|\Delta s_k^t\Delta w_k - \gamma_k\tau_2\frac{\Gamma}{2}\|\Delta v_k\|^2\right|$$

and let $\bar{r}_k$ be the vector composed by the first three block components of the vector r_k defined in (3.40) and (3.43). By the mean value theorem for vector valued functions (e.g. see [26, p.74]), we can write for $\alpha \in (0,1]$

$$\begin{aligned}H_1(v_k+\alpha\Delta v_k) &= H_1(v_k) + \alpha H_1'(v_k)\Delta v_k + \\ &\quad +\alpha\left(\int_0^1 (H_1'(v_k+\xi\alpha\Delta v_k) - H_1'(v_k))\,d\xi\right)\Delta v_k \\ &= (1-\alpha)H_1(v_k) + \alpha\bar{r}_k + \\ &\quad +\alpha\left(\int_0^1 (H_1'(v_k+\xi\alpha\Delta v_k) - H_1'(v_k))\,d\xi\right)\Delta v_k.\end{aligned} \tag{5.16}$$

From the Lipschitz continuity for the derivative of $H_1(v)$, we obtain

$$\begin{aligned}\|H_1(v_k+\alpha\Delta v_k)\| &\leq (1-\alpha)\|H_1(v_k)\| + \alpha\|\bar{r}_k\| + \\ &\quad +\alpha\left(\int_0^1 \Gamma\|\xi\alpha\Delta v_k\|d\xi\right)\|\Delta v_k\|,\end{aligned}$$

or, by (3.43)

$$\|H_1(v_k+\alpha\Delta v_k)\| \leq (1-\alpha)\|H_1(v_k)\| + \alpha\delta_k\|H(v_k)\| + \frac{\Gamma}{2}\alpha^2\|\Delta v_k\|^2. \tag{5.17}$$

From the definition of $f_k^{II}(\alpha)$ and by using (5.13), we have that

$$\begin{aligned} f_k^{II}(\alpha) &= s_k^t w_k + \alpha(-s_k^t w_k + \sigma_k \mu_k m) + \\ &\quad +\alpha^2 \Delta {s_k}^t \Delta w_k - \gamma_k \tau_2 \|H_1(v_k + \alpha \Delta v_k)\|. \end{aligned}$$

If we multiply (5.17) by $-\gamma_k \tau_1$, changing the sign, then we have a lower bound of $-\gamma_k \tau_2 \|H_1(v_k + \alpha \Delta v_k)\|$ that gives

$$\begin{aligned} f_k^{II}(\alpha) &\geq (1-\alpha) f_k^{II}(0) + \alpha(\sigma_k \mu_k m - \gamma_k \tau_2 \delta_k \|H(v_k)\|) + \\ &\quad +\alpha^2 (\Delta s_k^t \Delta w_k - \gamma_k \tau_2 \frac{\Gamma}{2} \|\Delta v_k\|^2). \end{aligned}$$

Then, by the hypothesis $f_k^{II}(0) \geq 0$, $\mu_k \geq \frac{s_k^t w_k}{m}$ and (5.3), we obtain

$$f_k^{II}(\alpha) \geq \alpha((\frac{\sigma_k}{1+\gamma_k \tau_2} - \delta_k)\gamma_k \tau_2 \|H(v_k)\| - \alpha M_k).$$

If condition (5.11) holds, then for any α such that

$$\alpha \geq ((\frac{\sigma_k}{1+\gamma_k \tau_2} - \delta_k)\gamma_k \tau_2 \|H(v_k)\| / M_k > 0, \tag{5.18}$$

we have $f_k^{II}(\alpha) \geq 0$. Thus we define

$$\check{\alpha}_k^{(2)} = \max_{\alpha \in (0,1]} \{\alpha : f_k^{II}(t) \geq 0, \, \forall t \leq \alpha\}.$$

This completes the proof. □

Let us define

$$\tilde{\alpha}_k = \min\{\hat{\alpha}_k^{(2)}, \check{\alpha}_k^{(2)}, 1\} \in (0,1];$$

then, under the hypotheses of Theorem 5.2, N_k and M_k are uniformly bounded in $\Omega(\epsilon)$ and

$$\tilde{\alpha}_k \geq \tilde{\alpha} > 0.$$

Consequently, we have

$$\alpha_k^{(2)} \equiv \min\{\tilde{\alpha}_k, \alpha_k^{(1)}\} \geq \alpha^{(2)} \equiv \min\{\tilde{\alpha}, \alpha^{(1)}\} > 0.$$

To select the final value of the damping parameter at the iteration k, we perform the backtracking technique described in [33] until an *acceptable*

$$\alpha_k = \theta^{\bar{t}} \alpha_k^{(2)}$$

is found, where $\bar{t}$ is the smallest nonnegative integer such that α_k satisfies the backtracking condition

$$\|H(v_k + \alpha_k \Delta v_k)\| \leq (1 - \beta\alpha_k(1 - (\sigma_k + \delta_k)))\, \|\boldsymbol{H}(v_k)\| \tag{5.19}$$

with $\theta, \beta \in (0, 1)$.
We have to prove now that $\bar{t}$ is a finite number independent on k.

Theorem 5.4 Under the hypotheses of Theorems 5.2 and 5.3, the backtracking procedure terminates in a finite number of steps.

Proof. From (5.12), (5.16), (3.34) and (3.35), we have, for $\alpha \in (0, 1]$ and for $i = 1, ..., m$:

$$\begin{aligned}((s_k)_i + \alpha(\Delta s_k)_i)((w_k)_i + \alpha(\Delta w_k)_i) &= (s_k)_i(w_k)_i + \alpha(-(s_k)_i(w_k)_i + \sigma_k\mu_k) + \\ &\quad +\alpha^2(\Delta s_k)_i(\Delta w_k)_i\end{aligned}$$

and

$$\begin{aligned}H_1(v_k + \alpha\Delta v_k) &= (1-\alpha)H_1(v_k) + \alpha\bar{r}_k + \\ &\quad +\alpha\left(\int_0^1 (H_1'(v_k + \xi\alpha\Delta v_k) - H_1'(v_k))\, d\xi\right)\Delta v_k.\end{aligned}$$

We can write

$$\begin{aligned}H(v_k + \alpha\Delta v_k) &= \begin{pmatrix} H_1(v_k + \alpha\Delta v_k) \\ (S_k + \alpha\Delta S_k)(W_k + \alpha\Delta W_k) \end{pmatrix} \\ &= (1-\alpha)\begin{pmatrix} H_1(v_k) \\ S_k W_k \boldsymbol{e}_m \end{pmatrix} + \alpha\begin{pmatrix} \bar{r}_k \\ 0 \end{pmatrix} + \alpha\begin{pmatrix} 0 \\ \sigma_k\mu_k e_m \end{pmatrix} + \\ &\quad +\alpha\begin{pmatrix} \left(\int_0^1 (H_1'(v_k + \xi\alpha\Delta v_k) - H_1'(v_k) d\xi\right)\Delta v_k \\ 0 \end{pmatrix} + \\ &\quad +\alpha^2\begin{pmatrix} 0 \\ \Delta S_k \Delta W_k e_m \end{pmatrix}.\end{aligned}$$

Thus

$$\begin{aligned}\|H(v_k + \alpha\Delta v_k)\| &\leq (1-\alpha)\|H(v_k)\| + \alpha\|r_k\| + \alpha\sigma_k\mu_k\|e_m\| + \\ &\quad +\alpha\|\Delta v_k\| \int_0^1 \|H_1'(v_k + \xi\alpha\Delta v_k) - H_1'(v_k)\| d\xi + \\ &\quad +\alpha^2\|\Delta S_k \Delta W_k e_m\|.\end{aligned}$$

From the Lipschitz continuity for the derivative of $H_1(v)$, from (3.43), we have

$$\begin{aligned}\|H(v_k + \alpha\Delta v_k)\| &\leq (1-\alpha)\|H(v_k)\| + \alpha(\sigma_k + \delta_k)\|H(v_k)\| + \\ &\quad +\alpha^2(\|\Delta S_k \Delta W_k e_m\| + \frac{\Gamma}{2}\|\Delta v_k\|^2).\end{aligned}$$

Therefore, we can affirm that

$$(1-\beta\alpha(1-(\sigma_k+\delta_k)))\|H(v_k)\|-\|H(v_k+\alpha\Delta v_k)\|\geq$$

$$\geq(1-\beta)\alpha(1-(\sigma_k+\delta_k))\|H(v_k)\|-\alpha^2(1+\tfrac{\Gamma}{2})\|\Delta v_k\|^2)$$

is nonnegative for $\alpha\in(0,\hat{\alpha}]$ with

$$\hat{\alpha}=\frac{(1-\beta)(1-(\sigma_k+\delta_k))\|H(v_k)\|}{(1+\frac{\Gamma}{2})\|\Delta v_k\|^2}>0$$

Since $\hat{\alpha}$ is bounded away from zero in $\Omega(\epsilon)$, $\epsilon>0$, it is possible to find a nonnegative integer $\bar{t}$ such that $0<\theta^{\bar{t}}\alpha_k^{(2)}\leq\min\{\hat{\alpha},1\}$; then the value $\alpha_k=\theta^{\bar{t}}\alpha_k^{(2)}$ is bounded below by a strictly positive number, say $\breve{\alpha}$.
This completes the proof. □

Set $\bar{\alpha}=\min\{\alpha^{(2)},\breve{\alpha}\}$, we observe that, since

$$(1-\beta\alpha_k(1-(\sigma_k+\delta_k)))\leq(1-\beta\bar{\alpha}(1-(\sigma_{\max}+\delta_{\max})))<1,$$

inequality (5.19) asserts that

$$\|H(v_{k+1})\|<\|H(v_k)\|. \qquad (5.20)$$

We prove now the following result (see [39]) which shows that the strict feasibility of the initial vectors $s_0>0$ and $w_0>0$ is sufficient to guarantee the positivity of the centrality functions $f^I(\alpha)$ and $f^{II}(\alpha)$ at each iterate k.

Proposition 5.1 Let $f^I(\alpha)$ and $f^{II}(\alpha)$ be the centrality functions defined in (3.19) and (3.20); set

$$\tau_1=\frac{\min_{i=1,m}(S_0W_0e_m)}{(\frac{s_0^tw_0}{m})};\qquad \tau_2=\frac{s_0^tw_0}{\|H_1(v_0)\|}$$

and let be given a sequence of parameters $\{\gamma_k\}$ with

$$1>\gamma_0\geq\gamma_1\geq\ldots\geq\gamma_k\geq\gamma_{k+1}\geq\ldots\geq\frac{1}{2}.$$

If $s_0>0$, $w_0>0$, then

$$\begin{array}{llll} f_k^I(\alpha) & \geq 0 & \text{for all} & \alpha\in(0,\hat{\alpha}_k^{(2)}] \\ f_k^{II}(\alpha) & \geq 0 & \text{for all} & \alpha\in(0,\breve{\alpha}_k^{(2)}] \end{array}$$

for any $k=0,1,\ldots$

Proof. For $k = 0$, the definitions of τ_1 and τ_2 give

$$\begin{aligned} f_0^I(0) &= (1-\gamma_0)\min_i(S_0 W_0 e_m) > 0 \\ f_0^{II}(0) &= (1-\gamma_0)s_0^t w_0 > 0. \end{aligned}$$

Theorem (5.4) assures that there exist $\hat{\alpha}_0^{(2)} > 0$ and $\check{\alpha}_0^{(2)} > 0$ such that

$$\begin{aligned} f_0^I(\alpha) &\geq 0 \qquad \text{for all} \quad \alpha \in (0, \hat{\alpha}_0^{(2)}] \\ f_0^{II}(\alpha) &\geq 0 \qquad \text{for all} \quad \alpha \in (0, \check{\alpha}_0^{(2)}]. \end{aligned}$$

Thus, we have $f_0^I(\alpha_0) \geq 0$ and $f_0^{II}(\alpha_0) \geq 0$, where α_0 is the final value of the damping parameter obtained after the backtracking procedure.
For $k = 1$, the centrality functions are

$$\begin{aligned} f_1^I(\alpha) &= \min_{i=1,\ldots,m}(S_1(\alpha)W_1(\alpha)e_m) - \gamma_1\tau_1\left(\frac{s_1(\alpha)^t w_1(\alpha)}{m}\right) \\ f_1^{II}(\alpha) &= s_1(\alpha)^t w_1(\alpha) - \gamma_1\tau_2\|H_1(v_1(\alpha))\|, \end{aligned}$$

where $s_1(\alpha) = s_1 + \alpha\Delta s_1$, $w_1(\alpha) = w_1 + \alpha\Delta w_1$ and $v_1(\alpha) = v_1 + \alpha\Delta v_1$.
We have

$$\begin{aligned} f_1^I(0) &= \min_{i=1,\ldots,m}(S_1 W_1 e_m) - \gamma_1\tau_1\left(\frac{s_1^t w_1}{m}\right) \\ f_1^{II}(0) &= s_1^t w_1 - \gamma_1\tau_2\|H_1(v_1)\|. \end{aligned}$$

Since

$$\begin{aligned} f_0^I(\alpha_0) &= \min_{i=1,\ldots,m}(S_1 W_1 e_m) - \gamma_0\tau_1\left(\frac{s_1^t w_1}{m}\right) \geq 0 \\ f_0^{II}(\alpha_0) &= s_1^t w_1 - \gamma_0\tau_2\|H_1(v_1)\| \geq 0 \end{aligned}$$

and $\gamma_1 \leq \gamma_0$, we have

$$f_1^I(0) \geq f_0^I(\alpha_0) \geq 0 \;\text{ and }\; f_1^{II}(0) \geq f_0^{II}(\alpha_0) \geq 0.$$

Thus, Theorem 5.4 assures that there exist $\hat{\alpha}_1^{(2)} > 0$ and $\check{\alpha}_1^{(2)} > 0$ such that

$$\begin{aligned} f_1^I(\alpha) &\geq 0 \qquad \text{for all} \quad \alpha \in (0, \hat{\alpha}_1^{(2)}] \\ f_1^{II}(\alpha) &\geq 0 \qquad \text{for all} \quad \alpha \in (0, \check{\alpha}_1^{(2)}]. \end{aligned}$$

Hence, we have $f_1^I(\alpha_1) \geq 0$ and $f_1^{II}(\alpha_1) \geq 0$, where α_1 is the step–length obtained after the execution of the backtracking procedure.

Thus, in the next steps ($k = 2, 3, ...$) of the process we have

$$f_k^I(0) \geq f_{k-1}^I(\alpha_{k-1}) \geq 0 \text{ and } f_k^{II}(0) \geq f_{k-1}^{II}(\alpha_{k-1}) \geq 0.$$

This completes the proof. □

Theorem 5.5 Under the hypotheses C1, C2 and C3, the Newton IP Algorithm 4.1, with $tol = 0$, generates a sequence $\{v_k\}$ such that:

(a) the sequence $\{\|H(v_k)\|\}$ converges to zero and each limit point of the sequence $\{v_k\}$ satisfies the KKT conditions for (**??**); furthermore, if v_* is a limit point of the sequence $\{v_k\}$ such that $H'(v_*)$ is nonsingular, then v_k converges to v_* when k diverges;

(b) if the sequence $\{v_k\}$ converges to v_* with $H'(v_*)$ nonsingular matrix, $\sigma_k = \mathcal{O}(\|H(v_k)\|^\xi)$, $0 < \xi < 1$, and $\delta_k = \mathcal{O}(\|\boldsymbol{H}(v_k)\|)$, then there exists an index $\bar{k}$ such that $\alpha_k = 1$ for $k \geq \bar{k}$. Thus, the Newton IP algorithm has a superlinear local convergence rate.

Proof.
Part (a) (see [32, Theor. 3.1]). The algorithm 4.1 generates a sequence $\{\|H(v_k)\|\}$ which is monotone nonincreasing, and bounded. Consequently, this sequence has limit, say, $H_* \geq 0$. If $H_* = 0$, we have the result. Suppose that $H_* > 0$, then the sequence $\{v_k\}$ and its limit points belong to $\Omega(\epsilon)$, with $\epsilon = (H_*)^2$. If v_* is one this limit points, we have that $H'(v_*)$ is a nonsingular matrix, then from theorem 2.8 ([33, Theor. 6.1]), we deduce that $H(\boldsymbol{v}_*) = 0$. This contradicts the assumption that $H^* > 0$. Hence, the sequence $\{\|H(v_k)\|\}$ must converge to zero.
Furthermore, if v_* is a limit point of the sequence $\{v_k\}$ such that $H'(v_*)$ is nonsingular, the same theorem also guarantees that v_k converges to v_* when k diverges.
Part (b) (see [30]). From (3.34), (3.35 and (3.43), (3.38), we have

$$\|\Delta v_k\| \leq \|H'(v_k)^{-1}\|(1 + \sigma_k + \delta_k)\|H(v_k)\|$$

where $H'(v_k)^{-1}$ is a bounded matrix, then, for $k \geq \bar{k}$, we have, $\|\Delta s_k\| = \mathcal{O}(\|H(v_k)\|)$ and $\|\Delta w_k\| = \mathcal{O}(\|H(v_k)\|)$. Then, for k sufficiently large, the conditions (3.21) and (3.22) are satisfied for $\alpha_k^{(2)} = 1$. Indeed, for k sufficiently large, $(\Delta s_k)_i < 0$ and $(\Delta w_k)_i < 0$ are negligible with respect $(s_k)_i$ and $(w_k)_i$ and then $\alpha_k^{(1)} = 1$.

Furthermore, from the definition of $(f_k^I(\alpha))_i$ and (5.14), we observe that

$$\begin{array}{rcl} (f_k^I(1))_i & = & (s_k)_i(1)(w_k)_i(1) - (\gamma_k\tau_1/m)s_k(1)^t w_k(1) \\ & \geq & \sigma_k\mu_k(1-\tau_1\gamma_k) - (1+\tau_1\gamma_k/m)\|\Delta s_k\|\|\Delta w_k\|. \end{array}$$

Since, from (3.38) and (5.3), we have

$$\|H(v_k)\|/((1+1/(\gamma_k\tau_2))m) \leq \mu_k \leq \|H(v_k)\|/\sqrt{m},$$

then $\mu_k = \mathcal{O}(\|H(v_k)\|)$ and $\sigma_k\mu_k = \mathcal{O}(\|H(v_k)\|^{\xi+1})$, while $\|\Delta s_k\|\|\Delta\lambda_k\| = \mathcal{O}(\|H(v_k)\|^2)$. Hence the criterion (3.21) is satisfied for $\hat{\alpha}_k^{(2)} = 1$, with k sufficiently large.
As far as the criterion (3.22) is concerned,

$$\begin{array}{rcl} f_k^{II}(1) & = & s_k(1)^t w_k(1) - \tau_2\|H_1(v_k(1))\| \\ & \geq & m\sigma_k\mu_k - (\gamma_k\tau_2\delta_k\|H(v_k)\| + (1+\gamma_k\tau_2)\|\Delta v_k\|^2), \end{array}$$

so, for sufficiently large k, $\check{\alpha}_k^{(2)} = 1$ satisfies (3.22).
Then $\alpha_k^{(2)} = \min\left(\alpha_k^{(1)}, \hat{\alpha}_k^{(2)}, \check{\alpha}_k^{(2)}, 1\right) = 1$.
Now we prove that the backtracking procedure determines $\alpha_k = 1$ for sufficiently large k.

$$\begin{aligned} \|H(v_k(1))\| &= \|H(v_k + \Delta v_k)\|, \\ &\leq \|H(v_k+\Delta v_k) - (H(v_k) + H'(v_k)\Delta v_k)\| \\ &\quad + \|H(v_k) + H'(v_k)\Delta v_k\|. \end{aligned}$$

For the Lemma 2.2 in [25] (see also the footnote in p. 403) and from the residual condition (2.30) with forcing term $\eta_k = \sigma_k + \delta_k$, it follows that

$$\begin{aligned} \|H(v_k(1))\| &\leq o(\|\Delta v_k\|) + (\delta_k+\sigma_k)\|H(v_k)\| \\ &= o(\|H(v_k)\|) + (\delta_k+\sigma_k)\|H(v_k)\|. \end{aligned}$$

Hence, we have

$$\begin{aligned} &(1-\beta(1-(\delta_k+\sigma_k)))\|H(v_k)\| - \|H(v_k(1))\| \\ &\quad \geq (1-\beta)(1-(\delta_k+\sigma_k))\|H(v_k)\| - o(\|H(v_k)\|) \\ &\quad = (1-\beta)\|H(v_k)\| - (1-\beta)(\delta_k+\sigma_k)\|H(v_k)\| - o(\|H(v_k)\|) \\ &\quad = (1-\beta)\|H(v_k)\| - (\mathcal{O}(\|H(v_k)\|^{1+\xi}) + \mathcal{O}(\|H(v_k)\|^2)) - o(\|H(v_k)\|) \\ &\quad \geq 0. \end{aligned}$$

Then, there exists an index $\bar{k} \geq 0$ such that $\alpha_k = 1$ for all $k \geq \bar{k}$. It follows that

$$\eta_k = 1 - \alpha_k(1 - (\delta_k + \sigma_k)) = \delta_k + \sigma_k, \quad \text{for } k \geq \bar{k},$$

and then, from Corollary 3.5(a) in [25], the sequence $\{v_k\}$ converges to v_* superlinearly. □

5.2 Convergence in the nonmonotone case

The convergence in the nonmonotone case, presented in Section 4.3, can be proved in a very similar way as in the previous section. However, we need a stronger hypothesis on the jacobian matrix $H'(v_k)$. More precisely, instead of C3, we will assume

C3"' in any compact subset of $\Omega(0)$ the matrix $H'(v)$ is nonsingular.

Furthermore, we observe that Theorem 5.1 holds also in this case: indeed, the inequalities employed in the proof depend only on the property $\|H(v_k)\| \leq \|H(v_0)\|$, which holds also in the nonmonotone case. From this remark, it follows that the iterates generated by the Algorithm 4.1 with nonmonotone choices belong to the set $\Omega(0)$ defined in the previous section. Furthermore, we can prove the boundedness of the sequence $\|\Delta v_k\|$ in $\Omega(0)$, as claimed in the following theorem.

Theorem 5.6 Assume that the hypotheses C1, C2 and C3"' hold. Then the sequence $\|\Delta v_k\|$, where the sequence $\{v_k\}$ is generated by the Algorithm 4.1 with the nonmonotone choices described in Section 4.3, is bounded in $\Omega(0)$.

Proof.
Since the sequence $\{v_k\}$ is bounded, then the matrix $H'(v_k)$ is nonsingular which means that its inverse $H'(v_k)^{-1}$ exists for any k. From the hypothesis C1, $H'(v)$ is a continuous function from $\mathbb{R}^n$ into $\mathbb{R}^{n\times n}$. Hence, also $H'(v)^{-1}$ and $\|H'(v)^{-1}\|$ are continuous functions, and there exists a positive number $\bar{M}$ such that $\|H'(v_k)^{-1}\| \leq M$ for each k.
Then the result follows by employing the same arguments as in the part (d) of theorem 5.2. □.

We observe that Theorem 5.3 holds also in the nonmonotone case, and in $\Omega(\epsilon)$ with $\epsilon > 0$ the damping parameter after the reduction due to the feasibility requirement and the centrality conditions is uniformly bounded away from zero. This property allows us to use the theorems in section 4.3 in the following convergence theorem ([12]).

Theorem 5.7 Let $\{v_k\}$ be the sequence generated by the algorithm 4.1 with the nonmonotone choices described in section 4.3. Then

(a) if v_* is a limit point of the sequence $\{v_k\}$ such that $H'(v_*)$ is nonsingular, then $H(v_*) = 0$;

(b) if v_* is a limit point of the sequence $\{v_k\}$ such that $H'(v_*)$ is nonsingular, then $\lim_{k\to\infty} \|H(v_k)\| = 0$ and $\{v_k\}$ converges to v_*.

(c) if the sequence $\{v_k\}$ converges to v_* with $H'(v_*)$ nonsingular matrix, $\sigma_k = \mathcal{O}(\|H(v_k)\|^{\xi})$, $0 < \xi < 1$, and $\delta_k = \mathcal{O}(\|H(v_k)\|)$, than there exists an index $\bar{k}$ such that $\alpha_k = 1$ for $k \geq \bar{k}$. Thus, the nonmonotone IP method has a superlinear local convergence rate.

Proof.
(a) Suppose that $H(v_*) = H_* > 0$. Then we have that

$$\lim_{k\to\infty} \|H(v_{\ell(k)})\| = L \geq H_* > 0.$$

By employing the same arguments as in the proof of Theorem 2.13, we obtain that v_* is a limit point also of the sequence $v_{\ell(k)-1}$ and that the damping parameter $\alpha_{\ell(k)-1}$ converges to zero as k diverges.
Since $L > 0$, then $v_{\ell(k)-1} \in \Omega(\epsilon)$, for some $\epsilon > 0$, hence Theorem 5.3 imply that the damping parameter after the reduction due to the feasibility requirement and to the centrality conditions is uniformly bounded away from zero. This result together with Corollary 2.1 gives the contradiction. Thus we necessarily have $H(v_*) = 0$.
(b) Suppose that

$$\lim_{k\to\infty} \|H(v_{\ell(k)})\| = L > 0. \tag{5.21}$$

The boundedness of the sequence $\{v_k\}$ guarantees that there exists a limit point v_* of the sequence $v_{\ell(k)}$. Thus (5.21) implies that the iterates v_k and the limit point v_* belong to $\Omega(\epsilon)$ with $\epsilon > 0$.
From the hypotheses made, if $v_* \in \Omega(\epsilon)$ then $H'(v_*)$ is nonsingular and the part (a) of the theorem claims that $H(v_*) = 0$, which is a contradiction. Thus we must have $L = 0$, which guarantees $\lim \|H(v_k)\| = 0$.
Now, the result follows from Theorem 2.11.
(c) If we suppose that $\{v_k\}$ converges to v_*, then we have that $\lim_{k\to\infty} \|H_{\ell(k)}\| = \lim \|H(v_k)\| = 0$. Thus the result follows by the part (c) of Theorem 5.5. □

5.3 A global convergence failure

It is well known in literature that in the following simple example in $\mathbb{R}$, many algorithm with global convergence properties fails to converge:

$$\begin{array}{ll} \min & x_1 \\ s.t. & x_1 - 1 \geq 0 \\ & x_1 - 0.5. \geq 0 \end{array}$$

By introducing the slack variables on the inequality constraints, we obtain the following equivalent formulation in $\mathbb{R}^3$

$$\begin{array}{ll} \min & x_1 \\ s.t. & x_1 - x_2 - 1 \geq 0 \\ & x_1 - x_3 - 0.5 \geq 0 \\ & x_2, x_3 \geq 0. \end{array} \tag{5.22}$$

The unique solution of (5.22) is the point $(x_1, x_2, x_3)^t = (1, 0, 0.5)^t$, but starting from the feasible point $((x_0)_1.(x_0)_2, (x_0)_3) = (-2, 3, 1)$ the algorithm 4.1 does not converge. It has been proved in [72] that any method which uses a search direction satisfying the linearization of the constraints fails on this test problem, for all the feasible initial points with $(x_0)_1 < 0$. The resulting sequence is plotted in figure 5.3. In this case, the vector $H(v)$ has the following form:

$$H(v) = \begin{pmatrix} 1 - 2w_1x_1 - w_2 \\ x_1^2 - x_2 - 1 \\ x_1 - x_3 - 0.5 \\ x_2w_1 \\ x_3w_2 \end{pmatrix} \tag{5.23}$$

where w_1 and w_2 are the multipliers of the inequality constraints. It can be observed that there exist two solutions of the system $H(v) = 0$: $v_* = (1, 0, 0.5, 0.5, 0)^t$, which is the minimum point with its corresponding multipliers, and $v_{**} = (-1, 0, -1.5, -0.5, 0)^t$, which is not a KKT point. The Newton's method applied to the system $H(v) = 0$ with starting point $v_0 = (-2, 3, 1, 1, 1)^t$ is attracted by the solution v_{**}. This remark suggests that the Newton direction, starting from a point sufficiently close to v_{**}, leads to the point v_{**} and a line–search procedure with every merit function, will not modify such behaviour, as also confirmed by the failure of LOQO.

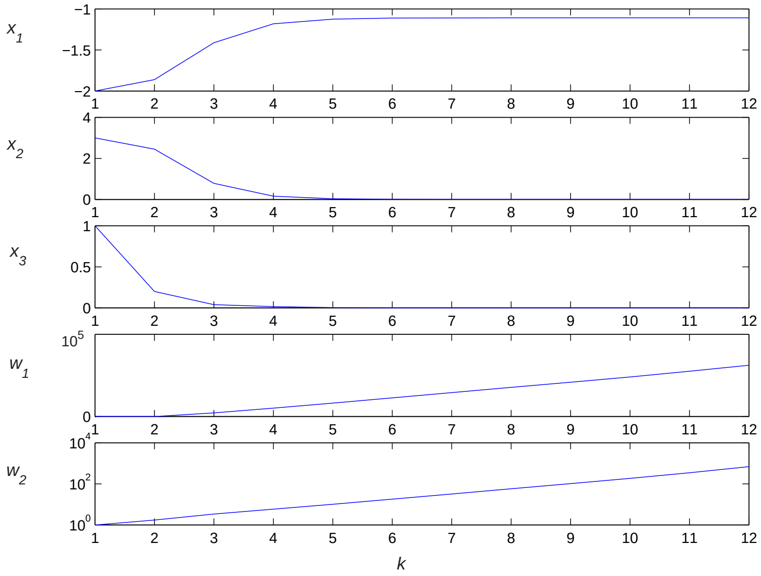


Figure 5.1: The sequence $\{v_k\}$ generated by the Algorithm 4.1 in the Wächter-Biegler counterexample

A positive result on this example is given by some algorithms implementing a trust–region strategy, for example the one presented in [5], or a filter strategy (see [38] and [73]).

Another possibility is to compute a suitable starting point by means of a procedure to be executed before the Newton interior–point iterations and we choose to perform such preprocess procedure by executing some steps of the projected gradient method, implemented as follows.

- For $k = 0, 1, 2, ...$
  - Compute $\Delta v_k = -H'(v_k)^t H(v_k)$
  - Set

$$\begin{array}{lcll} x_{k+1} & = & x_k + \Delta x_k & \\ \lambda_{k+1} & = & \lambda_k + \Delta \lambda_k & \\ (s_{k+1})_i & = & \max\{(s_k)_i + (\Delta s_k)_i, \epsilon\} & i = 1, \cdots, m \\ (w_{k+1})_i & = & \max\{(w_k)_i + (\Delta w_k)_i, \epsilon\} & i = 1, \cdots, m \end{array} \tag{5.24}$$

  - Set $g_k = v_{k+1} - v_k$ and $\alpha_k = 1$
  - While $\|H(v_{k+1})\|^2 > \|H(v_k)\|^2 - \beta\alpha\Delta v_k^t g_k$

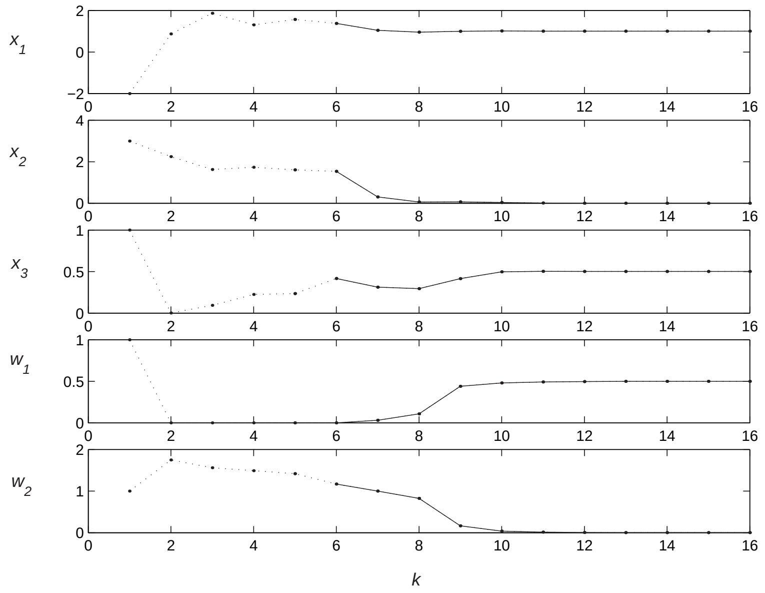


Figure 5.2: Preprocess with the projected gradient: the dotted line refers to the projected gradient iterations, the solid line to the IP iterations (4.1)

- $*$ Set $\alpha_k = \alpha_k \theta$
- $*$ Set

$$\begin{array}{lcll} x_{k+1} & = & x_k + \alpha_k \Delta x_k & \\ \lambda_{k+1} & = & \lambda_k + \alpha_k \Delta \lambda_k & \\ (s_{k+1})_i & = & \max\{(s_k)_i + \alpha_k (\Delta s_k)_i, \epsilon\} & i = 1, \cdots, m \\ (w_{k+1})_i & = & \max\{(w_k)_i + \alpha_k (\Delta w_k)_i, \epsilon\} & i = 1, \cdots, m \end{array} \tag{5.25}$$

- $*$ Set $g_k = v_{k+1} - v_k$

The result obtained by performing 5 steps of the projected gradient method on the problem $H(v) = 0$, where H is defined in (5.23), before the IP iterations, is depicted in figure 5.3. The projected gradient iterations bring the points v_k in the good region and then the IP iterations leads to the solution v_*.

Chapter 6

Description of the test problems

In this chapter we analyze the test problems we have considered for the numerical experience. We dealt with optimal control problems in two dimensions with an elliptic state equation and control and state constraints. The boundary conditions are the Dirichlet or the Neumann conditions and the control variable is defined either in the boundary (boundary control problems) or on the whole domain (distributed control).

In the following we analyze the continuous formulation of the problem, reporting the necessary optimality conditions. We also derived such optimality conditions in the unconstrained case of boundary control with Neumann conditions. Then, we describe the discretization technique and the nonlinear programming problems arising from such discretization.

We also report the values obtained for the cost functional and the graphs of the state and control function for a fixed value of the meshsize.

6.1 Elliptic boundary control problems

6.1.1 Optimality condition in a special case: boundary elliptic control problem with Neumann boundary conditions

In this section necessary optimality conditions are derived for a special class of elliptic control problems, where the state variable y is defined on a bounded set $\Omega \in \mathbb{R}^2$ and the control variable u is only defined on the boundary $\Gamma = \partial\Omega$, supposed to be piecewise smooth. The objective functional $F : \Omega \times \partial\Omega \rightarrow \mathbb{R}$ is defined as the following

$$F(y,u) = \int_{\Omega} f(x, y(x))dx + \int_{\Gamma} g(x, y(x), u(x))dx \tag{6.1}$$

and the state equation is the following elliptic partial differential equation with Neumann boundary condition

$$-\Delta y(x) + d(x, y(x)) = 0 \qquad x \in \Omega, \tag{6.2}$$

$$\partial_{\nu} y(x) = b(x, y(x), u(x)) \qquad x \in \Gamma. \tag{6.3}$$

The functions $f : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$, $g : \Gamma \times \mathbb{R}^2 \rightarrow \mathbb{R}$, $d : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$, and $b : \Gamma \times \mathbb{R}^2 \rightarrow \mathbb{R}$ are assumed to be C^2 functions on their respective domains. Under appropriate assumption on d, it can be proved that the state equation admits for each $u \in L^{\infty}(\Gamma)$ a weak solution $y \in C(\bar{\Omega}) \cap H^1(\Omega)$ [1].
Suppose now that $\bar{u}$ is a solution of the optimal control problem (6.1)-(6.3) and $\bar{y}$ the corresponding state variable, then

$$\int_{\Omega} \left[-\sum_{i=1}^{2} \frac{\partial \bar{y}}{\partial x_i} \frac{\partial q}{\partial x_i} + d(x, \bar{y}(x))q(x) \right] dx = \int_{\Gamma} b(x, \bar{y}(x), \bar{u}(x))q(x)dx \tag{6.4}$$

holds for each test function $q \in H^1(\Omega)$. Indeed, by multiplying (6.2) by a test function $q(x) \in H^1(\Omega)$, and by integrating on the domain Ω we obtain

$$\int_{\Omega} \left[-\sum_{i=1}^{2} \frac{\partial^2 \bar{y}}{\partial x_i^2} + d(x, y(x)) \right] q(x)dx = 0$$

which can be written also as

$$-\int_{\Omega} \sum_{i=1}^{2} \frac{\partial(\partial \bar{y}/\partial x_i)}{\partial x_i} q(x)dx + \int_{\Omega} d(x, \bar{y}(x))q(x)dx = 0$$

and this implies that

$$\int_{\Omega} \left[\sum_{i=1}^{2} \frac{\partial \bar{y}}{\partial x_i} \frac{\partial q}{\partial x_i} \right] dx - \int_{\Omega} \left[\sum_{i=1}^{2} \frac{\partial}{\partial x_i} \left(q \frac{\partial \bar{y}}{\partial x_i} \right) \right] dx + \int_{\Omega} d(x, \bar{y}(x))q(x)dx = 0.$$

[1] We recall that, given a subset X of $\mathbb{R}^n$, $L^2(X)$ is the space of square integrable functions, $L^{\infty}(X)$ is the space of the functions which are bounded almost everywhere in X and $H^1(X)$ is the space of the functions of $L^2(X)$ with the derivative belonging to $L^2(X)$.

For the Green theorem we have

$$\begin{aligned}
\int_\Omega \left(\frac{\partial}{\partial x_1}\left(q\frac{\partial \bar{y}}{\partial x_1}\right) - \frac{\partial}{\partial x_2}\left(-q\frac{\partial \bar{y}}{\partial x_2}\right)\right) dx &= \int_\Gamma \left(-q\frac{\partial \bar{y}}{\partial x_2}dx_1 + q\frac{\partial \bar{y}}{\partial x_1}dx_2\right) \\
&= \int_\Gamma q\left(-\frac{\partial \bar{y}}{\partial x_2}dx_1 + \frac{\partial \bar{y}}{\partial x_1}dx_2\right) \\
&= \int_\Gamma q\left(\frac{\partial \bar{y}}{\partial x_1}, \ \frac{\partial \bar{y}}{\partial x_2}\right)^t \left(\begin{array}{c} \cos\gamma_2 \\ \cos\gamma_1 \end{array}\right) ds
\end{aligned}$$

where ds is an element of Γ, γ_2 and γ_1 are the angles between the outward normal dv to the element of the surface ds and the directions x_1 and x_2 respectively. This means that

$$\begin{aligned}
dx_1 &= -ds\cos\gamma_1 \\
dx_2 &= ds\cos\gamma_2.
\end{aligned}$$

Then we have from (6.3)

$$\begin{aligned}
\int_\Omega \left[\sum_{i=1}^{2} \frac{\partial \bar{y}}{\partial x_i}\frac{\partial q}{\partial x_i} + d(x,\bar{y}(x))q(x)\right] dx &= \int_\Gamma q(x)\frac{\partial \bar{y}}{\partial \nu}dx \\
&= \int_\Gamma q(x)\cdot b(x,\bar{y}(x),u(x))dx
\end{aligned}$$

which implies (6.4).
Therefore, in the solution the objective functional can be written as

$$\begin{aligned}
F(\bar{y},\bar{u}) &= \int_\Omega f(x,\bar{y}(x))dx + \int_\Gamma g(x,\bar{y}(x),\bar{u}(x))dx + \\
&\quad + \int_\Omega \left[\sum_{i=1}^{2} \frac{\partial \bar{y}}{\partial x_i}\frac{\partial q}{\partial x_i} + d(x,\bar{y}(x))q(x)\right] dx - \int_\Gamma b(x,\bar{y}(x),\bar{u}(x))q(x)dx.
\end{aligned}$$

Now consider a one–parameter family of controls $\bar{u}(x) + ah(x)$ where $h \in L^\infty(\Gamma)$ and indicate with $z(a,x)$ the corresponding state variable such that the pair $(z(a,x),\bar{u}(x)+ah(x))$ satisfies the state equation, and assume that $z(\cdot,x) \in C^1(\mathbb{R})$.
The evaluation of the objective functional in that pair, for a fixed h, depends

only on the parameter a:

$$\begin{aligned} F(a) &= \int_\Omega f(x,z(a,x))dx + \int_\Gamma g(x,z(a,x),\bar{u}(x)+ah(x))dx + \\ &+ \int_\Omega \left[\sum_{i=1}^{2} \frac{\partial z(a,x)}{\partial x_i}\frac{\partial q}{\partial x_i} + d(x,z(a,x))q(x)\right] dx + \\ &- \int_\Gamma b(x,z(a,x),\bar{u}(x)+ah(x))q(x)dx. \end{aligned}$$

Since $\bar{u}$ is a minimizer control, $F(a)$ assumes its minimum value for $a = 0$, hence $F'(0) = 0$. Differentiating the previous expression with respect to a and evaluating in $a = 0$, the following relation holds:

$$\begin{aligned} F'(0) &= \int_\Omega f_y(x,\bar{y})z_a dx + \int_\Gamma [g_y(x,\bar{y},\bar{u})z_a + g_u(x,\bar{y},\bar{u})h]dx + \\ &+ \int_\Omega \left[\sum_{i=1}^{2} \frac{\partial z_a}{\partial x_i}\frac{\partial q}{\partial x_i} + d_y(x,\bar{y})z_a q(x)\right] dx + \\ &- \int_\Gamma [b_y(x,\bar{y},\bar{u})z_a + b_u(x,\bar{y},\bar{u})h]q(x)dx. \end{aligned}$$

If $\bar{q}$ satisfies the *adjoint equation*

$$-\Delta\bar{q}(x) + d_y(x,\bar{y}(x))\bar{q} + f_y(x,\bar{y}) = 0 \qquad x \in \Omega, \tag{6.5}$$

$$\partial_\nu \bar{q}(x) = b_y(x,\bar{y},\bar{u})\bar{q} - g_y(x,\bar{y},\bar{u}) \qquad x \in \Gamma \tag{6.6}$$

then

$$\int_\Omega \left[\sum_{i=1}^{2} \frac{\partial z_a}{\partial x_i}\frac{\partial q}{\partial x_i} + \left(d_y(x,\bar{y})q(x) + f_y(x,\bar{y})\right) z_a\right] dx - \int_\Gamma [b_y(x,\bar{y},\bar{u}) - g_y(x,\bar{y},\bar{u})] z_a dx = 0.$$

Consequently

$$F'(0) = \int_\Gamma [g_u(x,\bar{y},\bar{u}) - b_u(x,\bar{y},\bar{u})\bar{q}]h dx$$

and $F'(0) = 0$ for any h if the *minimum condition*

$$g_u(x,\bar{y},\bar{u}) - b_u(x,\bar{y},\bar{u})\bar{q} = 0 \tag{6.7}$$

holds. Conditions (6.5)–(6.6) and (6.7) represent the necessary conditions for the optimal control problem (6.1)-(6.3), in which the function $\bar{q}$, usually called *adjoint variable*, plays for the optimal control problem, the role the multiplier have in the optimization problem.

6.1.2 Statement of the problem

Here we consider the class of the constrained optimal control problems, where the state and the control variable have to satisfy an elliptic state equation and also some inequality constraint. We consider the following elliptic boundary control problem with Neumann boundary conditions: given a bounded domain $\Omega \subset \mathbb{R}^2$ with piecewise smooth boundary Γ, determine a boundary control function $u \in L^\infty(\Gamma)$ which minimizes the cost functional

$$F(y,u) = \int_\Omega f(x,y(x))dx + \int_\Gamma g(x,y(x),u(x))dx, \tag{6.8}$$

subject to the elliptic state equation

$$-\Delta y(x) + d(x,y(x)) = 0 \qquad \text{for } x \in \Omega, \tag{6.9}$$

and to the Neumann boundary conditions

$$\partial_\nu y(x) = b(x,y(x),u(x)) \qquad \text{for } x \in \Gamma. \tag{6.10}$$

Here ∂_ν denotes the derivative in the direction of the outward unit normal ν of Γ. We also introduce the following control and state inequality constraints

$$\begin{array}{lcll} C(x,y(x),u(x)) & \leq & 0 & x \in \Gamma, \\ S(x,y(x)) & \leq & 0 & x \in \bar{\Omega}. \end{array} \tag{6.11}$$

Here $\bar{\Omega} = \Omega \cup \Gamma$. The functions $f : \Omega \times \mathbb{R} \to \mathbb{R}$, $g : \Gamma \times \mathbb{R}^2 \to \mathbb{R}$, $d : \Omega \times \mathbb{R} \to \mathbb{R}$, $b : \Gamma \times \mathbb{R}^2 \to \mathbb{R}$, $C : \Gamma \times \mathbb{R}^2 \to \mathbb{R}$, $S : \bar{\Omega} \times \mathbb{R} \to \mathbb{R}$ are assumed to be C^2 functions.

When the elliptic boundary problem has Dirichlet conditions, the problem (6.8)–(6.11) becomes: determine a boundary control function $u \in L^\infty(\Gamma)$ which minimizes the cost functional

$$F(y,u) = \int_\Omega f(x,y(x))dx + \int_\Gamma g(x,u(x))dx, \tag{6.12}$$

subject to the state equation (6.9), the Dirichlet conditions

$$y(x) = b(x,u(x)) \qquad \text{for } x \in \Gamma, \tag{6.13}$$

and the inequality constraints on control and state

$$\begin{array}{lcll} C(x,u(x)) & \leq & 0 & x \in \Gamma, \\ S(x,y(x)) & \leq & 0 & x \in \Omega. \end{array} \tag{6.14}$$

Here $f : \Omega \times \mathbb{R} \to \mathbb{R}$, $g : \Gamma \times \mathbb{R} \to \mathbb{R}$, $d : \Omega \times \mathbb{R} \to \mathbb{R}$, $b : \Gamma \times \mathbb{R} \to \mathbb{R}$, $C : \Gamma \times \mathbb{R} \to \mathbb{R}$, $S : \Omega \times \mathbb{R} \to \mathbb{R}$, and f, g, d, b, C, S are C^2 functions.

A third version of an elliptic control problem is the following: determine a boundary control function $u \in L^\infty(\Gamma)$ which minimizes the cost functional

$$F(y,u) = \int_\Omega f(x,y(x))dx + \int_{\Gamma_\alpha} g(x,y(x),u(x))dx + \int_{\Gamma_\beta} K(x,u(x))dx, \tag{6.15}$$

where $\Gamma = \Gamma_\alpha \cup \Gamma_\beta$ with disjoint sets $\Gamma_\alpha, \Gamma_\beta \subset \Gamma$ that consist of finitely many connected components, subject to the state equation (6.9), to the boundary conditions of Neumann and Dirichlet type:

$$\partial_\nu y(x) = b_1(x,y(x),u(x)) \qquad \text{for } x \in \Gamma_\alpha, \tag{6.16}$$

$$y(x) = b_2(x,u(x)) \qquad \text{for } x \in \Gamma_\beta, \tag{6.17}$$

and the inequality constraints on control and state

$$\begin{array}{rcll} C(x,u(x)) & \leq & 0 & x \in \Gamma, \\ S(x,y(x)) & \leq & 0 & x \in \Omega. \end{array} \tag{6.18}$$

Here $f : \Omega \times \mathbb{R} \to \mathbb{R}$, $g : \Gamma_\alpha \times \mathbb{R}^2 \to \mathbb{R}$, $d : \Omega \times \mathbb{R} \to \mathbb{R}$, $K : \Gamma_\beta \times \mathbb{R} \to \mathbb{R}$, $b_1 : \Gamma_\alpha \times \mathbb{R}^2 \to \mathbb{R}$, $b_2 : \Gamma_\beta \times \mathbb{R} \to \mathbb{R}$, $C : \Gamma \times \mathbb{R} \to \mathbb{R}$, $S : \Omega \times \mathbb{R} \to \mathbb{R}$, and $f, g, K, d, b_1, b_2, C, S$ are C^2 functions.

For the general class of elliptic control problems, the theory of necessary conditions has not been yet fully developed. First order necessary optimality conditions for linear elliptic equations $-\Delta y(x) + y(x) = 0$ and pure Neumann conditions may be found in [22], [23], [24]. Problem (6.8)–(6.11) is considered as a mathematical programming problem in Banach spaces to which the first order Karush Kuhn Tucker conditions are applicable. For this approach, see [55].

For Dirichlet boundary conditions, a weak formulation of first order necessary conditions for linear elliptic equations may be found in [9]. Furthermore, first order conditions are derived in [55] in a purely formal way. This form of conditions is justified by its analogy in the first order necessary conditions for the discretized version of elliptic problem.

Also in the case of the problem (6.15)–(6.18), first order conditions are derived in a purely formal way in [57].

6.1.3 Discretization and optimization formulation

In the application of nonlinear programming techniques to optimal control, we use a full discretization approach [21], [10], [48], where both the control and the state variables are discretized and the integration method is included as an explicit equality constraint at each gridpoint. This technique leads to a large scale nonlinear programming problem (NLP) with a sparse structure of the jacobian of the constraints. We consider the standard situation where the elliptic operator is the laplacian and $\Omega = (0,1) \times (0,1)$. Given a positive integer N, we define the stepsize h as

$$h = \frac{1}{N+1}$$

and we consider the mesh points

$$x_{ij} = (ih, jh), \quad 0 \le i,j \le N+1.$$

In particular, denoting the following subsets of indices as follows

$$\begin{array}{lcl} I(\Omega) & \doteq & \{(i,j) : 1 \le i, j, \le N\}, \\ I_1(\Gamma) & \doteq & \{(i,0) : 1 \le i \le N\}, \\ I_2(\Gamma) & \doteq & \{(0,j) : 1 \le j \le N\}, \\ I_3(\Gamma) & \doteq & \{(N+1,j) : 1 \le j \le N\}, \\ I_4(\Gamma) & \doteq & \{(i,N+1) : 1 \le i \le N\}, \\ I(\Gamma) & \doteq & \cup_{k=1}^{4} I_k(\Gamma), \\ I(\bar{\Omega}) & \doteq & I(\Omega) \cup I(\Gamma), \\ I(\Gamma_\alpha) & \doteq & \{(i,j) : x_{ij} \in \Gamma_\alpha\}, \\ I(\Gamma_\beta) & = & I(\Gamma) - I(\Gamma_\alpha), \end{array}$$

we have $x_{ij} \in \Omega$ for $(i,j) \in I(\Omega)$, $x_{ij} \in \Gamma$ for $(i,j) \in I(\Gamma)$, $x_{ij} \in \Gamma_\alpha$ for $(i,j) \in I(\Gamma_\alpha)$ and $x_{ij} \in \Gamma_\beta$ for $(i,j) \in I(\Gamma_\beta)$. As usual, we denote the approximations of the state and control variables in the mesh points as

$$\begin{array}{lcll} y(x_{ij}) & \approx & y_{ij} & (i,j) \in I(\bar{\Omega}), \\ u(x_{ij}) & \approx & u_{ij} & (i,j) \in I(\Gamma). \end{array}$$

Now, we define the vector z as the vector whose entries are the approximations of the control and state variables.
When the Neumann boundary conditions hold, z is given by

$$z \doteq \Big((y_{ij})_{(i,j)\in I(\bar{\Omega})}, (u_{ij})_{(i,j)\in I(\Gamma)} \Big) \in \mathbb{R}^{N^2+8N}. \tag{6.19}$$

The laplacian operator $\Delta y(x)$ is approximated by using the standard five points formula for each $x_{ij}, (i,j) \in I(\Omega)$; thus, according to the previous notation, we have

$$-\Delta y(x_{ij}) \approx \frac{1}{h^2}\{4y_{ij} - y_{i+1,j} - y_{i-1,j} - y_{i,j+1} - y_{i,j-1}\}. \tag{6.20}$$

The values of the normal derivative, needed for the Neumann boundary conditions, are approximated in the mesh points by y^{ν}_{ij}/h, where

$$y^{\nu}_{ij} \doteq \begin{cases} y_{i0} - y_{i1}, & \text{for } j = 0 \quad i = 1, \ldots, N \\ y_{0j} - y_{1j}, & \text{for } i = 0 \quad j = 1, \ldots, N \\ y_{N+1,j} - y_{N,j}, & \text{for } i = N+1 \quad j = 1, \ldots, N \\ y_{i,N+1} - y_{i,N}, & \text{for } j = N+1 \quad i = 1, \ldots, N \end{cases} \tag{6.21}$$

Then, the discrete form of the elliptic equation and the discrete Neumann boundary conditions lead to the equality constraints

$$G^h_{ij}(z) \doteq 4y_{ij} - y_{i+1,j} - y_{i-1,j} - y_{i,j+1} - y_{i,j-1} + h^2 d(x_{ij}, y_{ij}) = 0, \tag{6.22}$$

for $(i,j) \in I(\Omega)$ and

$$B^h_{ij}(z) \doteq y^{\nu}_{ij} - hb(x_{ij}, y_{ij}, u_{ij}) = 0 \qquad \text{for } (i,j) \in I(\Gamma). \tag{6.23}$$

The control and state inequality constraints (6.11) lead to the inequality constraints on the variable z

$$\begin{aligned} C_{ij}(x_{ij}, y_{ij}, u_{ij}) &\leq 0 \quad (i,j) \in I(\Gamma), && (6.24) \\ S_{ij}(x_{ij}, y_{ij}) &\leq 0 \quad (i,j) \in I(\bar{\Omega}). && (6.25) \end{aligned}$$

When Dirichlet boundary conditions are given, they are included in the discrete relations

$$y_{ij} = b(x_{ij}, u_{ij}) \qquad \text{for } (i,j) \in I(\Gamma). \tag{6.26}$$

Then, the number of the optimization variables is reduced, so that we define

$$z \doteq \big((y_{ij})_{(i,j)\in I(\Omega)}, (u_{ij})_{(i,j)\in I(\Gamma)}\big) \in \mathbb{R}^{N^2+4N}. \tag{6.27}$$

The equality constraints agree with those in (6.22) where y_{i0}, y_{iN+1}, y_{0j}, y_{N+1j} are replaced by $b(x_{i0}, u_{i0})$, $b(x_{iN+1}, u_{iN+1})$, $b(x_{0j}, u_{0j})$, $b(x_{N+1j}, u_{N+1j})$ respectively. The control and state inequality constraints (6.14) give rise to the inequality constraints

$$\begin{aligned} C_{ij}(x_{ij}, u_{ij}) &\leq 0 \quad (i,j) \in I(\Gamma), \\ S_{ij}(x_{ij}, y_{ij}) &\leq 0 \quad (i,j) \in I(\Omega). \end{aligned} \tag{6.28}$$

When Dirichlet and Neumann boundary conditions are given, these conditions are included in the discrete relations

$$\begin{aligned} B^h_{ij}(z) &= y^{\nu}_{ij} - hb_1(x_{ij}, y_{ij}, u_{ij}) \qquad (i,j) \in I(\Gamma_\alpha), && (6.29) \\ y_{ij} &= b_2(x_{ij}, u_{ij}) \qquad (i,j) \in I(\Gamma_\beta). && (6.30) \end{aligned}$$

Then the number of variables is reduced, so that we define

$$z \doteq \big((y_{ij})_{(i,j)\in I(\Omega)\cup\in I(\Gamma_\alpha)}, (u_{ij})_{(i,j)\in I(\Gamma)}\big) \in \mathbb{R}^{N^2+\tau(N)}. \tag{6.31}$$

Here $\tau(N)$ is the number of variables related to the meshpoints on the edges of Γ, where we have to compute y_{ij} (meshpoints on Γ_α) and u_{ij} (meshpoints on Γ). Then, the equality constraints are given by

$$\begin{aligned} G^h_{ij}(z) &= 0 \qquad (i,j)\in I(\Omega), && (6.32)\\ B^h_{ij}(z) &= 0 \qquad (i,j)\in I(\Gamma_\alpha). && (6.33)\end{aligned}$$

The control and state inequality constraints agree with those in (6.28).
The approximations of the functionals (6.8), (6.12) and (6.15) are obtained by the rectangular rule and they are given by

$$F^h(z) \doteq h^2 \sum_{(i,j)\in I(\Omega)} f(x_{ij}, y_{ij}) + h \sum_{(i,j)\in I(\Gamma)} g(x_{ij}, y_{ij}, u_{ij}) \tag{6.34}$$

for Neumann boundary conditions, by

$$F^h(z) \doteq h^2 \sum_{(i,j)\in I(\Omega)} f(x_{ij}, y_{ij}) + h \sum_{(i,j)\in I(\Gamma)} g(x_{ij}, u_{ij}) \tag{6.35}$$

for Dirichlet boundary conditions, or by

$$\begin{aligned} F^h(z) &\doteq h^2 \sum_{(i,j)\in I(\Omega)} f(x_{ij}, y_{ij}) + h \sum_{(i,j)\in I(\Gamma_\alpha)} g(x_{ij}, y_{ij}, u_{ij}) + \\ &+ h \sum_{(i,j)\in I(\Gamma_\beta)} K(x_{ij}, u_{ij}) && (6.36)\end{aligned}$$

for mixed Neumann and Dirichlet boundary conditions.
Thus, for every N, we obtain a NLP problem; if we state Neumann conditions, the optimization variable z belongs to $\mathbb{R}^{N^2+8N}$ and the discrete boundary conditions (6.23) are included in the equality constraints:

$$\begin{array}{ll} \quad\min & F^h(z) \\ G^h_{ij}(z) = 0 & \qquad (i,j)\in I(\Omega), \\ B^h_{ij}(z) = 0 & \qquad (i,j)\in I(\Gamma), \\ C_{ij}(z) \le 0 & \qquad (i,j)\in I(\Gamma), \\ S_{ij}(z) \le 0 & \qquad (i,j)\in I(\bar\Omega). \end{array} \tag{6.37}$$

With Dirichlet conditions, the problem becomes

$$\begin{array}{ccc} \min & F^h(z) & \\ G^h_{ij}(z)=0 & & (i,j)\in I(\Omega), \\ C_{ij}(z)\leq 0 & & (i,j)\in I(\Gamma), \\ S_{ij}(z)\leq 0 & & (i,j)\in I(\Omega). \end{array} \tag{6.38}$$

with $z\in\mathbb{R}^{N^2+4N}$. With mixed boundary conditions, the NLP problem is as follows:

$$\begin{array}{ccc} \min & F^h(z) & \\ G^h_{ij}(z)=0 & & (i,j)\in I(\Omega), \\ B^h_{ij}(z)=0 & & (i,j)\in I(\Gamma_\alpha), \\ C_{ij}(z)\leq 0 & & (i,j)\in I(\Gamma), \\ S_{ij}(z)\leq 0 & & (i,j)\in I(\Omega). \end{array} \tag{6.39}$$

with $z\in\mathbb{R}^{N^2+\tau(N)}$.
The lagrangian functions of (6.37), (6.38) and (6.39) are respectively given by

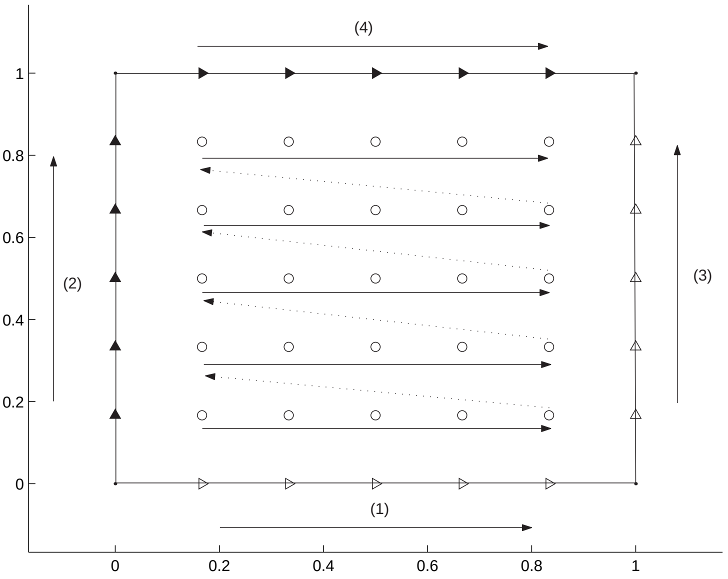
$$\begin{array}{rcl} L(z,q,\mu,\lambda) & = & h^2\sum_{(i,j)\in I(\Omega)} f(x_{ij},y_{ij}) + h\sum_{(i,j)\in I(\Gamma)} g(x_{ij},y_{ij},u_{ij}) + \\ & & +\sum_{(i,j)\in I(\Omega)} q_{ij}G^h_{ij}(z) + \sum_{(i,j)\in I(\bar{\Omega})} \mu_{ij}S(x_{ij},y_{ij}) + \\ & & +\sum_{(i,j)\in I(\Gamma)}[q_{ij}B^h_{ij}(z)+\lambda_{ij}C(x_{ij},y_{ij},u_{ij})], \end{array} \tag{6.40}$$

$$\begin{array}{rcl} L(z,q,\mu,\lambda) & = & h^2\sum_{(i,j)\in I(\Omega)} f(x_{ij},y_{ij}) + h\sum_{(i,j)\in I(\Gamma)} g(x_{ij},u_{ij}) + \\ & & +\sum_{(i,j)\in I(\Omega)}[q_{ij}G^h_{ij}(z)+\mu_{ij}S(x_{ij},y_{ij})] + \\ & & +\sum_{(i,j)\in I(\Gamma)} \lambda_{ij}C(x_{ij},u_{ij}), \end{array} \tag{6.41}$$

$$\begin{array}{rcl} L(z,q,\mu,\lambda) & = & h^2\sum_{(i,j)\in I(\Omega)} f(x_{ij},y_{ij}) + h\sum_{(i,j)\in I(\Gamma_\alpha)} g(x_{ij},y_{ij},u_{ij}) + \\ & & +h\sum_{(i,j)\in I(\Gamma_\beta)} K(x_{ij},u_{ij}) + \\ & & +\sum_{(i,j)\in I(\Omega)}[q_{ij}G^h_{ij}(z)+\mu_{ij}S(x_{ij},y_{ij})] + \\ & & +\sum_{(i,j)\in I(\Gamma_\alpha)} q_{ij}B^h_{ij}(z) + \sum_{(i,j)\in I(\Gamma)} \lambda_{ij}C(x_{ij},u_{ij}), \end{array} \tag{6.42}$$

where the Lagrange multipliers $q=(q_{ij})_{(i,j)\in I(\bar{\Omega})}$ for (6.40), $q=(q_{ij})_{(i,j)\in I(\Omega)}$ for (6.41), $q=(q_{ij})_{(i,j)\in I(\Omega)\cup I(\Gamma_\alpha)}$ for (6.42) are associated with the equality constraints and $\mu=(\mu_{ij})_{(i,j)\in I(\bar{\Omega})}$ (or $(i,j)\in I(\Omega)$) and $\lambda=(\lambda_{ij})_{(i,j)\in I(\Gamma)}$ are related to the inequality constraints $S_{ij}(z)\leq 0$ and $C_{ij}(z)\leq 0$ respectively. The ordering of the discrete variables y_{ij} and u_{ij} in the array z

Figure 6.1: Ordering of the discrete variables



determines the structure of the jacobian matrix of the equality constraints and of the hessian matrix of the lagrangian function. The Figure 6.1 depicts the strategy chosen here: the first entries of z are the $y_{ij}, \quad (i,j) \in I(\Omega)$ in lexicographic order from $(i,j) = (1,1)$ to $(i,j) = (N,N)$. Then, when we have boundary Neumann conditions, we store in the array z the boundary values y_{ij}, where $(i,j) \in I_k(\Gamma)$, for $k = 1,2,3,4$. Finally, we store in the array z the discrete control variables u_{ij} in the same order than the boundary entries y_{ij}. For the problems (6.38) and (6.39), we use the same strategy.

6.1.4 Test problems: general description

In the following, we consider elliptic problems where the cost functional is of tracking type

$$F(y,u) = \frac{1}{2}\int_\Omega (y(x) - y_d(x))^2 dx + \frac{\alpha}{2}\int_\Gamma (u(x) - u_d(x))^2 dx, \tag{6.43}$$

with given function $y_d \in C(\bar{\Omega})$, $u_d \in L^\infty(\Gamma)$ and a nonnegative weight $\alpha \geq 0$. The control and state constraints are supposed to be box constraints of the simple type

$$y(x) \le \psi(x) \qquad \text{on } \Omega \text{ or } \bar{\Omega}, \tag{6.44}$$

$$u_1(x) \le u(x) \le u_2(x) \qquad \text{on } \Gamma, \tag{6.45}$$

with functions $\psi \in C(\bar{\Omega})$ and $u_1, u_2 \in L^\infty(\Gamma)$.

We assume that an optimal solution of the optimal control problem considered exists and we denote by $\bar{y}(x)$ and $\bar{u}(x)$ the optimal state function and the optimal control function respectively. If the function b in (6.10) is such that $b_u = 1$ (or respectively b in (6.13) is such that $b_u = 1$ or b_1 in (6.16) is such that $b_{1u} = 1$), the optimal control $\bar{u}(x)$ is completely determined.

In the case of Neumann boundary condition, if we denote by $\bar{q}(x)$ the adjoint state corresponding to $\bar{y}(x)$ and $\bar{u}(x)$, we have:

- case $\alpha > 0$:

$$\bar{u}(x) = \begin{cases} u_d(x) + \bar{q}(x)/\alpha & \text{if } u_d(x) + \bar{q}(x)/\alpha \in (u_1(x), u_2(x)), \\ u_1(x) & \text{if } u_d(x) + \bar{q}(x)/\alpha \le u_1(x), \\ u_2(x) & \text{if } u_d(x) + \bar{q}(x)/\alpha \ge u_2(x), \end{cases} \tag{6.46}$$

- case $\alpha = 0$: we obtain an optimal control of bang–bang or singular type:

$$\bar{u}(x) = \begin{cases} u_1(x) & \text{if } \bar{q}(x) < 0, \\ u_2(x) & \text{if } \bar{q}(x) > 0, \\ \text{singular} & \text{if } \bar{q}(x) = 0 \text{ on } \Gamma_S \subset \Gamma, \quad \int_{\Gamma_S} dx > 0. \end{cases} \tag{6.47}$$

For $\alpha = 0$, the adjoint state function $\bar{q}(x)$ on the boundary plays the role of a switching function. The isolated zeros of $\bar{q}(x)|_\Gamma$ are switching points of a bang–bang control.

For Dirichlet boundary conditions, we obtain the same results if we replace $\bar{q}(x)$ formally by $-\partial_\nu \bar{q}(x)$. For $\alpha = 0$, the outward normal derivatives $-\partial_\nu \bar{q}(x)|_\Gamma$ plays the role of a switching function. The isolated zeros of $-\partial_\nu \bar{q}(x)|_\Gamma$ are the switching points of a bang–bang control.

For mixed boundary conditions, if $b_{1u} = 1$ and $\bar{q}(x)$ denotes the adjoint state, we have:

- case $\alpha > 0$: for $x \in \Gamma(\alpha)$, $\bar{u}(x)$ is as in (6.46), while for $x \in \Gamma_\beta$, $\bar{u}(x)$ is as in (6.46) with $\bar{q}(x)$ replaced by $-\partial_\nu \bar{q}(x)$;

- case $\alpha = 0$: we obtain an optimal control of bang–bang or singular type; for $x \in \Gamma_\alpha$, $\bar{u}(x)$ is as in (6.47) while for $x \in \Gamma_\beta$, $\bar{u}(x)$ is as in (6.47) with $\bar{q}(x)$ replaced by $-\partial_\nu \bar{q}(x)$; here Γ is replaced by Γ_α or Γ_β.

Then, for $\alpha = 0$, the switching function is given by $\bar{q}(x)$ on Γ_α and by $-\partial_\nu \bar{q}(x)$ on Γ_β. The isolated zeros of the switching function are the switching points of a bang–bang control.

The discrete counterpart of $\bar{q}(x)$ is the vector of the Lagrange multipliers $q = q_{ij}$. For Dirichlet boundary conditions, $\partial_\nu \bar{q}(x)|_\Gamma$ is replaced by q^ν_{ij}/h, where q^ν_{ij} is given by the finite differences of (6.21). In this case, we assume that q_{ij} are equal to zero on Γ ($q_{i0} = q_{iN+1} = q_{0j} = q_{N+1j} = 0$). For mixed boundary conditions, $\bar{q}(x)$ is replaced by the Lagrange multipliers on Γ_α and $\partial_\nu \bar{q}(x)$ is replaced by q^ν_{ij}/h on Γ_β with q^ν_{ij} as in (6.21). Furthermore, $q_{ij} = 0$ for $(i,j) \in I(\Gamma_\beta)$.

In all the described test problems, the choice of symmetric function $y_d(x)$ and $u_d(x)$ in the tracking functional implies that the optimal control is the same on every edge of Γ.

6.1.5 Test problems: discretization technique.

When the discretization techniques described in section 6.1.3 are applied to a cost functional of tracking type (6.43), $F^h(z)$ can be written as follows:

$$F^h(z) = \frac{1}{2}h^2 \sum_{(i,j)\in I(\Omega)} (y_{ij} - y_d(x_{ij}))^2 + \frac{\alpha}{2}h \sum_{(i,j)\in I(\Gamma)} (u_{ij} - u_d(x_{ij}))^2.$$

The hessian matrix H of $F^h(z)$ is a diagonal matrix, given by

$$H = diag(h^2 I_{n_1}, 0_{n_2}, h\alpha I_{n_3}), \tag{6.48}$$

where $n_1 = N^2$, $n_2 = 0$, $n_3 = 4N$ for Dirichlet boundary conditions, $n_1 = N^2$,$n_2 = 4N$,$n_3 = 4N$ for Neumann boundary conditions. For mixed boundary conditions, n_1, n_2, n_3 depend on the choice of Γ_α and Γ_β (see problems 6.1.9 and 6.1.10).

Now, we determine the jacobian matrix J of the equality constraints. For Dirichlet boundary conditions, J is an $N^2 \times (N^2 + 4N)$ matrix, given by

$$J = [Y + D, E], \tag{6.49}$$

where the $N^2 \times N^2$ matrix D is

$$D = h^2 diag\left(\frac{\partial d(x_{ij}, y_{ij})}{\partial y_{ij}}\right),$$

the $N^2 \times 4N$ matrix E is a sparse matrix with non null entries equal to $-\frac{\partial b(x_{ij},u_{ij})}{\partial u_{ij}}$, so that

$$e_{kl} = \begin{cases} -\frac{\partial b(x_{i0},u_{i0})}{\partial u_{i0}} & l,k,i = 1,\ldots,N \\ -\frac{\partial b(x_{0j},u_{0j})}{\partial u_{0j}} & j = 1,\ldots,N \quad k = (j-1)N+1, l = N+j \\ -\frac{\partial b(x_{N+1j},u_{N+1j})}{\partial u_{N+1j}} & j = 1,\ldots,N \quad k = jN, l = 2N+j \\ -\frac{\partial b(x_{iN+1},u_{iN+1})}{\partial u_{iN+1}} & i = 1,\ldots,N \quad k = N^2-N+i, l = 4N-N+i \end{cases} \tag{6.50}$$

and, finally, Y is an $N \times N$ block tridiagonal matrix with $N \times N$ diagonal blocks given by

$$\begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \tag{6.51}$$

and off diagonal blocks equal to $-I_N$.

For Neumann boundary conditions, J is an $(N^2+4N)\times(N^2+8N)$ matrix, that can be written as follows

$$J = \left[\begin{array}{ccc} Y+D & B^t & 0_{4N} \\ B & T & S \end{array} \right], \tag{6.52}$$

where Y and D are $N^2 \times N^2$ matrices as in (6.49) and S, T are the following $4N \times 4N$ diagonal matrices:

$$S = diag\left(-h\frac{\partial b(x_{ij},y_{ij},u_{ij})}{\partial u_{ij}}\right), \quad (i,j) \in I(\Gamma) \tag{6.53}$$

$$T = diag\left(1 - h\frac{\partial b(x_{ij},y_{ij},u_{ij})}{\partial y_{ij}}\right), \quad (i,j) \in I(\Gamma) \tag{6.54}$$

and B^t is a sparse $N^2 \times 4N$ matrix where the nonzero entries are equal to 1 and whose indices are the same of the nonzero entries of E in (6.49). We point out that $S = -hI_{4N}$ if $b_u = 1$.

For mixed boundary conditions, the structure of J is similar to (6.52), but the sizes of B, T and S depend on the choice of Γ_α and Γ_β (see problems 6.1.9 and 6.1.10). The hessian matrix $\bar{H}$ of the lagrangian function is equal to H in (6.48) for Dirichlet boundary conditions, while for Neumann conditions, $\bar{H}$ is given by

$$\bar{H} = H + \begin{pmatrix} \bar{Y} & 0 & 0 \\ 0 & \bar{T} & \bar{V} \\ 0 & \bar{V}^t & \bar{S} \end{pmatrix}, \tag{6.55}$$

where the $N^2 \times N^2$ matrix $\bar{Y}$, the $4N \times 4N$ matrices $\bar{T}$, $\bar{S}$ and $\bar{V}$are given by

$$\bar{Y} = diag\left(h^2 q_{ij}\frac{\partial^2 d(x_{ij}, y_{ij})}{\partial y_{ij}^2}\right), \quad (i,j) \in I(\Omega), \tag{6.56}$$

$$\bar{T} = diag\left(-hq_{ij}\frac{\partial^2 b(x_{ij}, y_{ij}, u_{ij})}{\partial y_{ij}^2}\right), \quad (i,j) \in I(\Gamma), \tag{6.57}$$

$$\bar{S} = diag\left(-hq_{ij}\frac{\partial^2 b(x_{ij}, y_{ij}, u_{ij})}{\partial u_{ij}^2}\right), \quad (i,j) \in I(\Gamma), \tag{6.58}$$

$$\bar{V} = diag\left(-hq_{ij}\frac{\partial^2 b(x_{ij}, y_{ij}, u_{ij})}{\partial y_{ij} u_{ij}}\right), \quad (i,j) \in I(\Gamma). \tag{6.59}$$

Note that, if $b_u = 1$, then $\bar{S} = \bar{V} = 0_{4N}$.
For mixed boundary conditions, the hessian matrix $\bar{H}$ of the lagrangian function is similar to (6.55), but the size of $\bar{T}$, $\bar{S}$, and $\bar{V}$ depends on the choice of Γ_α and Γ_β (see problems 6.1.9 and 6.1.10).
For convenience, the numerical results reported in the following of this section for all the test problems are referred to the fixed stepsize $h = 1/(N+1)$, with $N = 99$.

Problem 6.1.1 (Example 5.5 in [55])

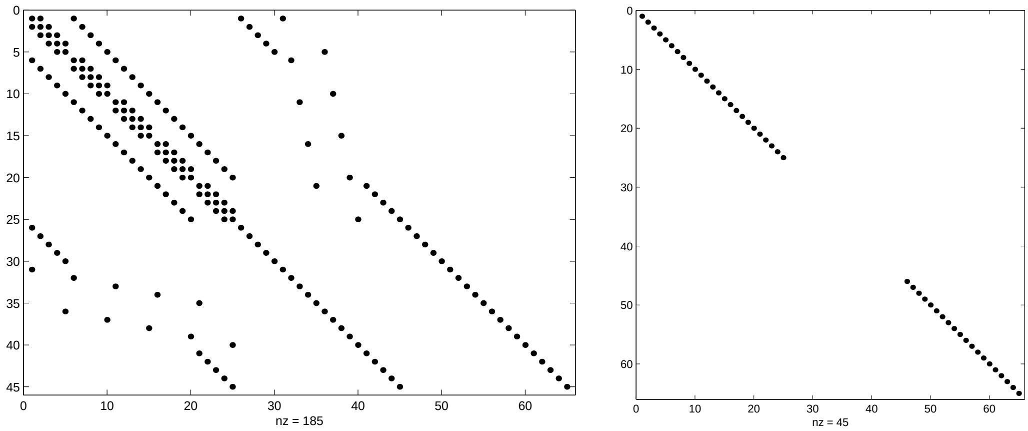
We consider the following elliptic control problem with Neumann boundary conditions: minimize the functional (6.43)

$$F(y,u) = \frac{1}{2}\int_\Omega (y(x) - y_d(x))^2 dx + \frac{\alpha}{2}\int_\Gamma (u(x) - u_d(x))^2 dx,$$

subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) = 0, & y_d(x) = 2 - 2(x_1(x_1-1) + x_2(x_2-1)), \\ \text{on } \Gamma: & \partial_\nu y(x) = u(x) - y(x)^2, & 3.7 \le u(x) \le 4.5, \quad u_d(x) \equiv 0, \alpha = 0.01 \\ \text{on } \bar{\Omega}: & y(x) \le 2.071. & \end{array}$$

This problem leads to a NLP problem. The structure of the jacobian and hessian matrices J and H are depicted in Figure 6.2. The pictures, here and in the following, are obtained with $N = 5$.



Jacobian matrix J Hessian matrix H

Figure 6.2: Problem 6.1.1

Problem 6.1.1	
Variables	$N^2 + 8N$
Constraints	$N^2 + 4N$
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	$4N$
Nonzeros $\nabla^2 f(x)$	$N^2 + 4N$
Nonzeros $\nabla g_1(x)$	$5N^2 + 12N$

The hessian matrix H of F^h is a positive semidefinite matrix, because the entries related to $y_{ij}, (i,j) \in I(\Gamma)$ are equal to zero, while the hessian matrix $\bar{H}$ of the lagrangian function is an indefinite diagonal matrix (see Figure 6.3).

The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.55224597$. The optimal control is a continuous function and, on the bottom edge of Γ, it is such that

- $\bar{u}(x) = 3.7$ for $x = (x_1, 0)$, with $x_1 \in (0, 0.18) \cup (0.82, 1)$
- $\bar{u}(x) = 4.5$ for $x = (x_1, 0)$, with $x_1 \in (0.36, 0.64)$.

Since $b_u = 1$, on the edges of Γ, we have

$$\bar{u}(x) = \begin{cases} \bar{q}(x) \cdot 100 & \text{if } \bar{q}(x) \cdot 100 \in (3.7, 4.5) \\ 3.7 & \text{if } \bar{q}(x) \cdot 100 \leq 3.7 \\ 4.5 & \text{if } \bar{q}(x) \cdot 100 \geq 4.5 \end{cases}$$

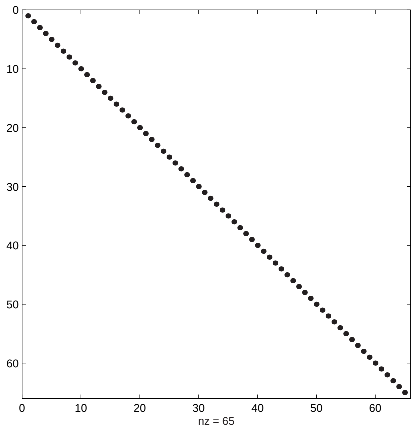


Figure 6.3: Problem 6.1.1: Hessian matrix $\bar{H}$

The active set for the state constraint $y(x) \leq 2.071$ is given by the midpoints of the edges of Γ. The dual variable for this active inequality constraint is 0.0004478692. At $x_{i0} = (0.5, 0)$, we have $y_{i0} = 2.071$, $q_{i0} = -0.04651456$.

Problem 6.1.2 (Example 5.6 in [55])

We consider the following elliptic control problem with nonlinear Neumann boundary conditions: minimize the functional (6.43) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) = 0, & y_d(x) = 2 - 2(x_1(x_1-1) + x_2(x_2-1)), \\ \text{on } \Gamma: & \partial_\nu y(x) = u(x) - y(x)^2, & 6 \leq u(x) \leq 9, \quad u_d(x) \equiv 0, \alpha = 0, \\ \text{on } \bar{\Omega}: & y(x) \leq 2.835. & \end{array}$$

The obtained programming problem is an NLP problem where the jacobian matrix J and the hessian matrix H have the same structure of those of the previous problem (see Figure 6.2), but the entries of the hessian matrix of F^h related to the control variables u_{ij} are equal to zero. These entries are zero also in $\bar{H}$.

Problem 6.1.2	
Variables	$N^2 + 8N$
Constraints	$N^2 + 4N$
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	$4N$
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$5N^2 + 12N$

Furthermore, the nonlinearity of the Neumann conditions leads to nonconstant diagonal entries (the ones related to the y_{ij}, $(i,j) \in I(\Gamma)$) in the hessian $\bar{H}$ of the Lagrangian, which is an indefinite diagonal matrix (see Figure 6.4). The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.015078$. The optimal control is a bang–bang control, that, on the edges of Γ, is given by

$$\bar{u} = \begin{cases} 6 & \text{if } q_{ij} < 0 \\ 9 & \text{if } q_{ij} > 0 \end{cases}$$

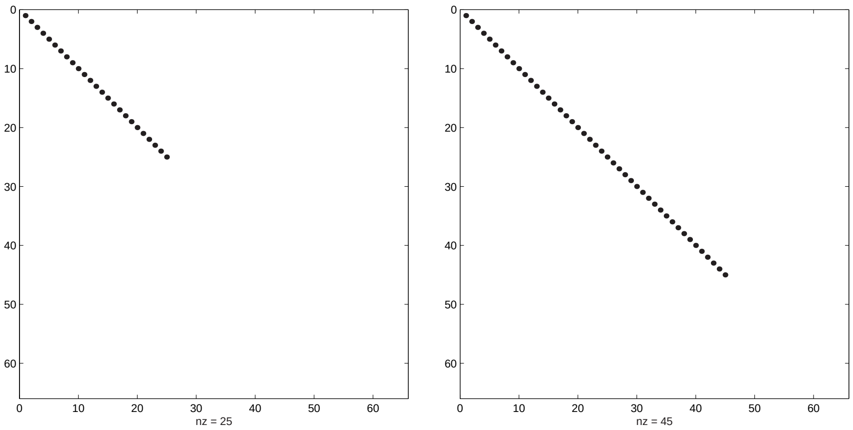
where $j = 1$ for the bottom edge, $j = N$ for the top edge, $i = 1$ for the left edge and $i = N$ for the right edge. The switching points on the bottom edge of Γ are approximately (0.33,0) and (0.67,0). The optimal state is equal to 2.835 at the midpoints of the edges of Γ. The dual variable for this active inequality constraint is $\mu_{ij} = 0.00002895$.

Problem 6.1.3 (Example 5.7 in [55])

We consider the following elliptic control problem with Neumann boundary conditions: minimize the functional (6.43) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) - y(x) + y(x)^3 = 0, & y_d(x) = 2 - 2(x_1(x_1 - 1) + x_2(x_2 - 1)) \\ \text{on } \Gamma: & \partial_\nu y(x) = u(x), & 1.8 \le u(x) \le 2.5, \quad u_d(x) \equiv 0, \alpha = 0.01 \\ \text{on } \bar{\Omega}: & y(x) \le 2.7. & \end{array}$$

By means of the discretization techniques, a NLP problem is obtained; the structures of the jacobian matrix J and of the hessian matrix H of F^h are the same of those in problem 6.1.1 (see Figure 6.2), but the hessian matrix $\bar{H}$ of the lagrangian function has the form in Figure 6.5.



Hessian matrix H Hessian matrix $\bar{H}$

Figure 6.4: Problem 6.1.2

Problem 6.1.3	
Variables	$N^2 + 8N$
Constraints	$N^2 + 4N$
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	$4N$
Nonzeros $\nabla^2 f(x)$	$N^2 + 4N$
Nonzeros $\nabla g_1(x)$	$5N^2 + 12N$

In this case the first N^2 entries of the diagonal depend on the values of $y_{ij}, (i,j) \in I(\Omega)$. The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.264163$ The optimal control is a continuous function and, on the bottom edge of Γ, it is such that

- $\bar{u}(x) = 1.8$ for the points $x = (x_1, 0)$, $x_1 \in (0, 0.15) \cup (0.85, 1)$
- $\bar{u}(x) = 2.5$ for the points $x = (x_1, 0)$, $x_1 \in (0.29, 0.71)$.

Indeed, on the edges of Γ, we have

$$\bar{u}(x) = \begin{cases} \bar{q}(x) \cdot 100 & \text{if } \bar{q}(x) \cdot 100 \in (1.8, 2.5) \\ 1.8 & \text{if } \bar{q}(x) \cdot 100 \leq 1.8 \\ 2.5 & \text{if } \bar{q}(x) \cdot 100 \geq 2.5 \end{cases}$$

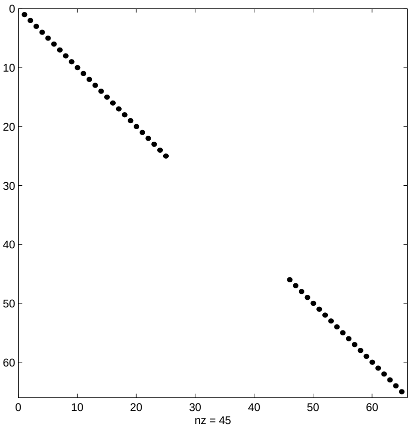


Figure 6.5: Problem 6.1.3: Hessian matrix $\bar{H}$

The active set for the state constraint $y(x) \leq 2.7$ comprises the points adjacent to the corners of the domain. The dual variable for this active inequality constraint is $\mu_{ij} = 0.0034573$.

Problem 6.1.4 (Example 5.8 in [55])

The cost functional and the constraints are the same of problem 6.1.3, but we choose $\alpha = 0$; thus the jacobian matrix J has the same structure than in problem 6.1.3, while the structures of the hessian matrix H of F^h and of the hessian matrix $\bar{H}$ of the Lagrangian are given in Figure 6.6.

Problem 6.1.4	
Variables	$N^2 + 8N$
Constraints	$N^2 + 4N$
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	$4N$
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$5N^2 + 12N$

The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.165531$. The optimal control is a bang–bang control, and, on the bottom edge of Γ, it is given by

$$\bar{u}(x) = \begin{cases} 1.8 & \text{if } q_{ij} < 0 \\ 2.5 & \text{if } q_{ij} > 0 \end{cases}$$

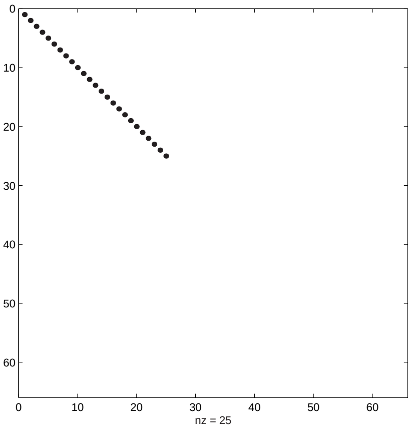


Figure 6.6: Problem 6.1.4: Hessian matrices H and $\bar{H}$

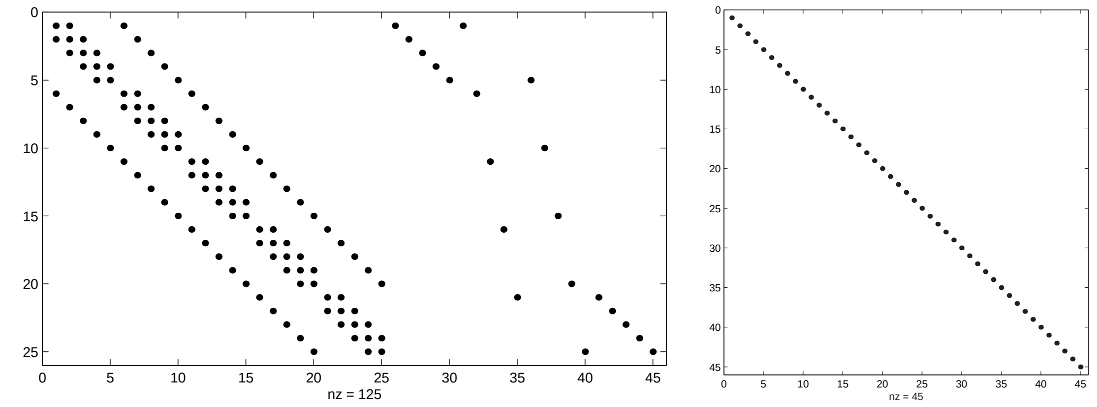
where $j = 1$ for the bottom edge, $j = N$ for the top edge, $i = 1$ for the left edge and $i = N$ for the right edge of Γ. The switching points on the bottom edge of Γ are approximately $(0.21, 0)$ and $(0.79, 0)$. Again, the optimal state is active at the points adjacent to the corners of the domain. The dual variable for this active inequality constraint is $\mu_{ij} = 0.030118$.

Problem 6.1.5 (Example 5.1 in [55])

We consider the following elliptic control problem with Dirichlet boundary conditions: minimize the functional (6.43) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) = 20, & \begin{array}{l} y(x) \leq 3.5, \\ y_d(x) = 3 + 5x_1(x_1 - 1)x_2(x_2 - 1), \end{array} \\ \text{on } \Gamma: & y(x) = u(x), & 0 \leq u(x) \leq 10, \quad u_d(x) \equiv 0, \alpha = 0.01. \end{array}$$

This control problem leads to a strictly convex quadratic programming problem (QP) whose jacobian and hessian matrices J and H are structured as shown in Figure 6.7.



Jacobian matrix J Hessian matrix $H = \bar{H}$

Figure 6.7: Problem 6.1.5

Problem 6.1.5	
Variables	$N^2 + 4N$
Constraints	N^2
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	$N^2 + 4N$
Nonzeros $\nabla g_1(x)$	$5N^2$

For $N = 99$, the minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.196525$. The control constraints are not active while the state variable attains its upper bound only in the center $x_{ij} = (0.5, 0.5)$ of the unit square with dual variable $\mu_{ij} = 0.24602$. Here $q_{ij} = -0.21312$, $y_{ij} = 3.5$, $y_{ij} - y_d(x_{ij}) = 0.1875$. Furthermore $y(0.4, 0.5) = 3.449163$ and $u(0, 0.5) = 1.690270$.

Problem 6.1.6 (Example 5.2 in [55])

The cost functional and the constraints are the same of problem 6.1.5, except that we choose $\alpha = 0$ instead of $\alpha = 0.01$, Then, the jacobian matrix J has the same structure as in Figure 6.7, while the diagonal entries of the hessian matrix H related to the variables u_{ij} are equal to 0 (see Figure 6.8). The programming problem is a convex QP problem.

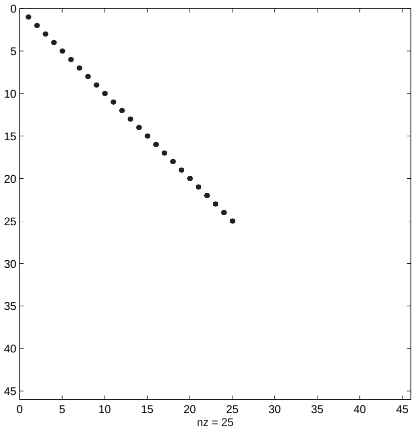


Figure 6.8: Problem 6.1.6: Hessian matrix $H = \bar{H}$

Problem 6.1.6	
Variables	$N^2 + 4N$
Constraints	N^2
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$5N^2$

In this case we can expect either a bang–bang or a singular control. We observe the following numerical results ($N = 99$):

- the minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.096695$;

- both the control and state constraint do not become active; the optimal control is totally singular on Γ; from the numerical point of view, this means that the multipliers q_{i1}, q_{iN}, q_{1j}, q_{Nj} are equal to zero.

Problem 6.1.7 (Example 5.3 in [55])

We consider the following elliptic control problem with Dirichlet boundary conditions: minimize the functional (6.43) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) = 20, & \begin{array}{l} y(x) \leq 3.2, \\ y_d(x) = 3 + 5x_1(x_1-1)x_2(x_2-1), \end{array} \\ \text{on } \Gamma: & y(x) = u(x), & 1.6 \leq u(x) \leq 2.3, \quad u_d(x) \equiv 0, \alpha = 0.01. \end{array}$$

The discretized problem is a strictly convex QP problem and the structures of jacobian and hessian matrices are the same of problem 6.1.5 (see Figure 6.7).

Problem 6.1.7	
Variables	$N^2 + 4N$
Constraints	N^2
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	$N^2 + 4N$
Nonzeros $\nabla g_1(x)$	$5N^2$

For $N = 99$, the minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.321010$. Furthermore $y(x) = 3.2$ at the center point $x_{ij} = (0.5, 0.5)$. The corresponding multiplier is $\mu_{ij} = 0.642704$; the optimal control is continuous and, on the bottom edge of Γ, it is such that

- $u_{i0} = 2.3$ for the points on the edge having the x_1 coordinate in $(0.002, 0.18) \cup (0.82, 0.98)$;
- $u_{i0} = 1.6$ for the points on the edge having the x_1 coordinate in $(0.23, 0.77)$;

Indeed, in view of $\alpha = h = 0.01$, we have

$$u_{i,0} = \begin{cases} q_{i,1} \cdot 10^4 & \text{if } q_{i,1} \cdot 10^4 \in (1.6, 2.3) \\ 1.6 & \text{if } q_{i,1} \cdot 10^4 \leq 1.6 \\ 2.3 & \text{if } q_{i,1} \cdot 10^4 \geq 2.3 \end{cases}$$

Problem 6.1.8 (Example 5.4 in [55])

The data are the same of problem 6.1.7, but $\alpha = 0$, so the hessian matrix H is positive semidefinite with zero entries in correspondence of the variables u_{ij}. We have a convex QP problem.

Problem 6.1.8	
Variables	$N^2 + 4N$
Constraints	N^2
Upper bounds	$N^2 + 4N$
Lower bounds	$4N$
Linear equalities	N^2
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$5N^2$

The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.249178$. The optimal control is a bang–bang control and

$$u_{i,0} = \begin{cases} 1.6 & \text{if } q_{i,1} < 0 \\ 2.3 & \text{if } q_{i,1} > 0 \end{cases}$$

The switching points on the bottom edge of Γ are $(0.2, 0)$ and $(0.8, 0)$. The optimal state is active at the center point $x_{ij} = (0.5, 0.5)$ and the multiplier related to this active inequality constraint is $\mu_{ij} = 0.73378$.

Problem 6.1.9 (Example 4.1 in [57])

We consider the following elliptic control problem with mixed Dirichlet and Neumann boundary conditions: given $\Gamma_\beta = \{(x_1, 1) : 0 \le x_1 \le 1\}$ and $\Omega_0 = [0.25, 0.75]^2$, minimize the cost functional

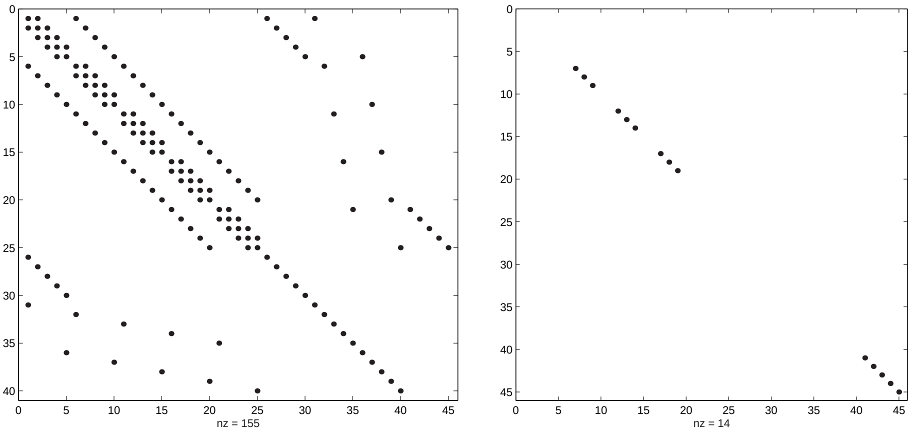
$$F(y,u) = \frac{1}{2}\int_{\Omega_0} (y(x) - 1)^2 dx + \frac{\alpha}{2}\int_{\Gamma_\beta} u(x)^2 dx \tag{6.60}$$

subject to

$$\begin{array}{lll} \text{on } \Omega : & -\Delta y(x) = 0, & \begin{array}{l} 0 \le y(x) \le 3.15 \text{ on } \Omega_0 \\ 0 \le y(x) \le 10 \text{ on } \Omega - \Omega_0 \end{array} \\ \text{on } \Gamma_\alpha : & \delta_\nu y(x) = 0 & \text{for } x_2 = 0, 0 \le x_1 \le 1 \\ & \delta_\nu y(x) = y(x) - 5 & \text{for } x_1 \in \{0,1\}, 0 \le x_2 \le 1 \\ \text{on } \Gamma_\beta : & y(x) = u(x) & 0 \le u(x) \le 10 \\ & & \alpha = 0.005 \end{array}$$

In this case, $z \equiv ((y_{ij})_{(i,j)\in I(\Omega)\cup I(\Gamma_\alpha)}, (u_{ij})_{(i,j)\in I(\Gamma_\beta)}) \in \mathbb{R}^{N^2+4N}$. The programming problem is a convex QP problem. The jacobian matrix J corresponding to the equality constraints is given by

$$J = \begin{bmatrix} Y & U^t & E \\ U & T & 0_N \end{bmatrix}$$



Jacobian matrix J Hessian matrix $H = \bar{H}$

Figure 6.9: Problem 6.1.9

where Y is a block tridiagonal $N^2 \times N^2$ matrix as in (6.49), $\begin{bmatrix} U^t & E \end{bmatrix} = B^t$ is a $N^2 \times 4N$ matrix as in (6.52), U is a sparse $3N \times N^2$ matrix, T is a diagonal $3N \times 3N$ matrix as in (6.54). The hessian matrix H of $F^h(y, u)$ is a square diagonal matrix of order $N^2 + 4N$. The hessian matrix $\bar{H}$ of the lagrangian function is equal to H. The structures of J and H are reported in Figure 6.9 The diagonal entries of H corresponding to the indices of $x_{ij} \in \Omega - \Omega_0$ are equal to zero. In the same way, the diagonal entries of H corresponding to $y_{ij}, (i, j) \in I(\Gamma_\alpha)$ are equal to zero. The entries related to u_{ij} are equal to $h\alpha$.

Problem 6.1.9	
Variables	$N^2 + 4N$
Constraints	$N^2 + 3N$
Upper bounds	$N^2 + N$
Lower bounds	$N^2 + 4N$
Linear equalities	$N^2 + 3N$
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	$((N + 1)/2)^2 + N$
Nonzeros $\nabla g_1(x)$	$5N^2 + 3N$

The minimum of the cost functional is $F(\bar{y}, \bar{u}) = 0.26284923$. The state constraint $y \leq 3.15$ for $x \in \Omega_0$ becomes active at the points $(\frac{1}{4}, \frac{3}{4})$ and $(\frac{3}{4}, \frac{3}{4})$ while the state constraint $y \leq 10$ in $\Omega - \Omega_0$ does not become active. Since no control is applied on the boundary Γ_α, we have

$$u_{iN+1} = \begin{cases} q_{iN}/(\alpha h) & \text{if } q_{iN}/(\alpha h) \in (0,10) \\ 0 & \text{if } q_{iN}/(\alpha h) \leq 0 \\ 10 & \text{if } q_{iN}/(\alpha h) \geq 0 \end{cases}$$

The graph of the discrete function $q_{iN}/(\alpha h)$ is reported in Figure 6.25.

Problem 6.1.10 (Example 4.1 in [57] with $\alpha = 0$)

The cost functional and the constraints are the same of the previous problem, but in this case we choose $\alpha = 0$. The optimal control is a bang–bang control, given by

$$u_{i,N+1} = \begin{cases} 0 & \text{if } q_{iN} \leq 0 \\ 10 & \text{if } q_{iN} \geq 0 \end{cases}$$

The programming problem is a convex QP problem. The jacobian matrix of the equality constraints is the same of the problem 6.1.9 (see Figure 6.9); the hessian matrix H is a diagonal matrix as that of the problem 6.1.9, but the entries corresponding to u_{ij} are equal to 0 (see Figure 6.10).

Problem 6.1.10	
Variables	$N^2 + 4N$
Constraints	$N^2 + 3N$
Upper bounds	$N^2 + N$
Lower bounds	$N^2 + 4N$
Linear equalities	$N^2 + 3N$
Nonlinear equalities	0
Nonzeros $\nabla^2 f(x)$	$((N+1)/2)^2$
Nonzeros $\nabla g_1(x)$	$5N^2 + 3N$

6.2 Elliptic distributed control problems

6.2.1 Statement of the problem

We consider the following elliptic distributed control problem with mixed Neumann and Dirichlet boundary conditions: given a bounded domain $\Omega \subset \mathbb{R}^2$ with piecewise smooth boundary Γ, where $\Gamma = \Gamma_\alpha \cup \Gamma_\beta$ with disjoints sets $\Gamma_\alpha, \Gamma_\beta \subset \Gamma$ that are composed of finitely many smooth and connected components, determine a distributed control function $u \in L^\infty(\Omega)$ that minimizes the cost functional

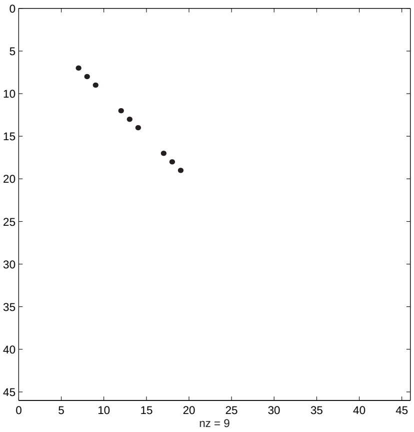


Figure 6.10: Problem 6.1.10: Hessian matrix $H = \bar{H}$

$$F(y,u) = \int_{\Omega} f(x, y(x), u(x))dx + \int_{\Gamma_\alpha} g(x, y(x))dx, \tag{6.61}$$

subject to the elliptic state equation

$$-\Delta y(x) + d(x, y(x), u(x)) = 0 \text{ for } x \in \Omega, \tag{6.62}$$

and to the Neumann and Dirichet boundary conditions

$$\partial_\nu y(x) = b_1(x, y(x)) \qquad \text{for } x \in \Gamma_\alpha \tag{6.63}$$

$$y(x) = b_2(x) \qquad \text{for } x \in \Gamma_\beta \tag{6.64}$$

and mixed control–state or pure state inequality constraints

$$\begin{array}{lcll} C(x, y(x), u(x)) & \leq & 0 & x \in \Omega \\ S(x, y(x)) & \leq & 0 & x \in \Omega \cup \Gamma_\alpha \end{array} \tag{6.65}$$

The functions $f : \Omega \times \mathbb{R}^2 \to \mathbb{R}$, $g : \Gamma_\alpha \times \mathbb{R} \to \mathbb{R}$, $d : \Omega \times \mathbb{R}^2 \to \mathbb{R}$, $b_1 : \Gamma_\alpha \times \mathbb{R} \to \mathbb{R}$, $b_2 : \Gamma_\beta \times \mathbb{R} \to \mathbb{R}$, $C : \Omega \times \mathbb{R}^2 \to \mathbb{R}$, and $S : \Omega \cup \Gamma_\alpha \times \mathbb{R} \to \mathbb{R}$ are assumed to be C^1 functions. As for boundary control problem, also for the distributed control problem, first order necessary conditions known in literature (see [14] and [13], [8] for linear elliptic equations and [20], [49], [68] for nonlinear elliptic equations of Lotka–Volterra type) have been formally extended in [56]. In this way, the necessary conditions are consistent with their counterparts in the discretized problems, given by the KKT conditions.

6.2.2 Discretization and optimization formulation

For the distributed control, we can use the same discretization and optimization techniques described in section 6.1.3 for boundary control. Also in this case, we consider the standard situation where the elliptic operator is the laplacian and $\Omega = (0,1) \times (0,1)$. Given a positive integer N and $h = \frac{1}{N+1}$ consider the mesh points

$$x_{ij} = (ih, jh), \quad 0 \leq i, j \leq N+1.$$

Assume the same notations stated in 6.1.3. We define the vector z as the vector whose entries are the approximations of the state variables y_{ij}, $(i,j) \in I(\Omega) \cup I(\Gamma_\alpha)$ and of the control variables u_{ij}, $(i,j) \in I(\Omega)$:

$$z \doteq \big((y_{ij})_{(i,j)\in I(\Omega)\cup I(\Gamma_\alpha)}, (u_{ij})_{(i,j)\in I(\Gamma)}\big) \in \mathbb{R}^{2N^2+\tau(N)}. \tag{6.66}$$

where $\tau(N)$ is the number of index pairs of $I(\Gamma_\alpha)$. The remaining state variables y_{ij}, $(i,j) \in I(\Gamma_\beta)$ are determined by the Dirichlet condition (6.64) as

$$y_{ij} = b_2(x_{ij}) \text{ for } \quad (i,j) \in I(\Gamma_\beta). \tag{6.67}$$

The derivative $\partial_\nu y(x_{ij})$ in the direction of the outward normal is approximated by y^ν_{ij}/h, where y^ν_{ij} is defined in (6.21). Then the discrete form of the Neumann boundary condition (6.63) leads to the equality constraints

$$B^h_{ij}(z) \doteq y^\nu_{ij} - hb_1(x_{ij}, y_{ij}) = 0, \quad \text{ for } (i,j) \in I(\Gamma_\alpha). \tag{6.68}$$

The application of the five points formula to the elliptic equation (6.62) yields the following equality constraint for all $(i,j) \in I(\Omega)$

$$G^h_{ij}(z) \doteq 4y_{ij} - y_{i+1,j} - y_{i-1,j} - y_{i,j+1} - y_{i,j-1} + h^2 d(x_{ij}, y_{ij}, u_{ij}) = 0, \tag{6.69}$$

Note that the discrete Dirichlet conditions (6.67) are used in this equation to substitute the variables y_{ij} for $(i.j) \in I(\Gamma_\beta)$. The control and state inequality constraints (6.65) yield the inequality constraints

$$C(x_{ij}, y_{ij}, u_{ij}) \leq 0 \qquad \text{for } (i,j) \in I(\Omega), \tag{6.70}$$

$$S(x_{ij}, y_{ij}) \leq 0 \qquad \text{for } (i,j) \in I(\Omega) \cup I(\Gamma_\alpha). \tag{6.71}$$

The discretized form of the cost functional (6.61) is

$$F^h(z) \doteq h^2 \sum_{(i,j)\in I(\Omega)} f(x_{ij}, y_{ij}, u_{ij}) + h \sum_{(i,j)\in I(\Gamma_\alpha)} g(x_{ij}, y_{ij}). \tag{6.72}$$

In summary, for any N, we have the following nonlinear programming problem:

$$\begin{array}{ccc} \min & F^h(z) & \\ G^h_{ij}(z)=0 & & (i,j)\in I(\Omega), \\ B^h_{ij}(z)=0 & & (i,j)\in I(\Gamma_\alpha), \\ C(x_{ij},y_{ij},u_{ij})\leq 0 & & (i,j)\in I(\Omega), \\ S(x_{ij},y_{ij})\leq 0 & & (i,j)\in I(\Omega)\cup I(\Gamma_\alpha), \end{array} \tag{6.73}$$

with $z\in\mathbb{R}^{2N^2+\tau(N)}$.
The lagrangian function of the NLP problem (6.73) is given by

$$\begin{array}{rcl} L(z,q,\lambda,\mu) & = & h^2\sum_{(i,j)\in I(\Omega)} f(x_{ij},y_{ij},u_{ij})+h\sum_{(i,j)\in I(\Gamma_\alpha)} g(x_{ij},y_{ij})+ \\ & & +\sum_{(i,j)\in I(\Omega)}[q_{ij}G^h_{ij}(z)+\lambda_{ij}C(x_{ij},y_{ij},u_{ij})+\mu_{ij}S(x_{ij},y_{ij})]+ \\ & & +\sum_{(i,j)\in I(\Gamma_\alpha)}[\mu_{ij}S(x_{ij},y_{ij})+q_{ij}B^h_{ij}(z)], \end{array} \tag{6.74}$$

where the Lagrange multipliers $q=(q_{ij})_{(i,j)\in I(\Omega)\cup I(\Gamma_\alpha)}$, $\lambda=(\lambda_{ij})_{(i,j)\in I(\Omega)}$ and $\mu=(\mu_{ij})_{(i,j)\in I(\Omega)\cup I(\Gamma_\alpha)}$ are associated respectively with the equality constraints (6.69) and (6.68) and with the inequality constraints (6.70) and (6.71). The ordering of the discrete variables y_{ij} and u_{ij} in the array z is described in subsection 6.1.3 (see Figure 6.1).

6.2.3 Test problems: general description

In the following, we consider elliptic problems where the cost functional is of tracking type (except for the last problems):

$$F(y,u)=\frac{1}{2}\int_\Omega(y(x)-y_d(x))^2dx+\frac{\alpha}{2}\int_\Omega(u(x)-u_d(x))^2dx, \tag{6.75}$$

with given function $y_d\in C(\bar\Omega)$, $u_d\in L^\infty(\Omega)$ and a nonnegative weight $\alpha\geq 0$. The control and state constraints are supposed to be box constraints of the simple type

$$y(x)\leq\psi(x)\quad \text{on } \Omega, \tag{6.76}$$

$$u_1(x)\leq u(x)\leq u_2(x)\quad \text{on } \Omega, \tag{6.77}$$

with functions $\psi\in C(\bar\Omega)$ and $u_1,u_2\in L^\infty(\Omega)$. We assume that an optimal solution $\bar y(x)$ and $\bar u(x)$ of the optimal control problems exists. If $d(x,y,u)$ in the state equation (6.62) is linear in the control variable u, the optimal control $\bar u(x)$ is completely determined. If we denote by $\bar q(x)$ the adjoint state corresponding to $\bar y(x)$ and $\bar u(x)$, we have:

- case $\alpha \geq 0$: for $x \in \Omega$

$$\bar{u}(x) = \begin{cases} u_d(x) + \bar{q}(x)/\alpha & \text{if } u_d(x) + \bar{q}(x)/\alpha \in (u_1(x), u_2(x)), \\ u_1(x) & \text{if } u_d(x) + \bar{q}(x)/\alpha \leq u_1(x), \\ u_2(x) & \text{if } u_d(x) + \bar{q}(x)/\alpha \geq u_2(x), \end{cases} \tag{6.78}$$

- case $\alpha = 0$: we obtain an optimal control of bang–bang or singular type:

$$\bar{u}(x) = \begin{cases} u_1(x) & \text{if } \bar{q}(x) < 0, \\ u_2(x) & \text{if } \bar{q}(x) > 0, \\ \text{singular} & \text{if } \bar{q}(x) = 0 \text{ on } \Omega_S \subset \Omega, \quad \int_{\Omega_S} dx > 0. \end{cases} \tag{6.79}$$

The discrete counterpart of $\bar{q}(x)$ is the vector of the Lagrange multipliers $q = (q_{ij})$, where we set $q_{ij} = 0$ for $(i, j) \in I(\Gamma_\beta)$.

6.2.4 Test problems: discretization techniques

When the discretization techniques described in section 6.2.2 are applied to a cost functional of tracking type (6.75), $F^h(z)$ can be written as follows:

$$F^h(z) = \frac{1}{2}h^2 \sum_{(i,j)\in I(\Omega)} (y_{ij} - y_d(x_{ij}))^2 + \frac{\alpha}{2}h \sum_{(i,j)\in I(\Omega)} (u_{ij} - u_d(x_{ij}))^2.$$

The hessian matrix H of $F^h(z)$ is a diagonal matrix, given by

$$H = diag(h^2 I_{n_1}, 0_{n_2}, h\alpha I_{n_3}), \tag{6.80}$$

where $n_1 = N^2$, $n_2 = \tau(N)$, $n_3 = N^2$. If $\Gamma_\alpha = \emptyset$ and $\Gamma_\beta = \Gamma$ (Dirichlet boundary conditions only), then $n_2 = 0$. If $\Gamma_\alpha = \Gamma$ and $\Gamma_\beta = \emptyset$ (Neumann boundary conditions), then $n_2 = 4N$.

Now, we determine the jacobian matrix J of the equality constraints. J is a sparse $(N^2 + \tau(N)) \times (2N^2 + \tau(N))$ matrix, that can be written as follows:

$$J = \begin{bmatrix} Y + D & \bar{U}^t & \bar{E} \\ \bar{U} & T & 0 \end{bmatrix} \tag{6.81}$$

where Y is an $N \times N$ block tridiagonal matrix as in (6.49), D is a square diagonal matrix of order N^2 with diagonal entries of the form

$$\left(h^2 \frac{\partial d(x_{ij}, y_{ij}, u_{ij})}{\partial y_{ij}} \right), \quad (i, j) \in I(\Omega),$$

$\bar{U}^t$ is a sparse $N^2 \times \tau(N)$ matrix with non null entries equal to -1, $\bar{E}$ is a square diagonal matrix of order N^2 with diagonal entries $h^2 \frac{\partial d(x_{ij}, y_{ij}, u_{ij})}{\partial u_{ij}}$, $(i, j) \in I(\Omega)$ and finally, T is a square diagonal matrix of order $\tau(N)$ with diagonal entries

$$\left(1 - h\frac{\partial b_1(x_{ij}, y_{ij})}{\partial y_{ij}}\right), \quad (i, j) \in I(\Gamma_\alpha).$$

If $\Gamma_\alpha = \emptyset$, $\Gamma_\beta = \Gamma$, J becomes equal to the following $N^2 \times 2N^2$ matrix:

$$J = [Y + D, \bar{E}]. \tag{6.82}$$

If $\Gamma_\alpha = \Gamma$, $\Gamma_\beta = \emptyset$, J is a $(N^2 + 4N) \times (2N^2 + 4N)$ matrix.
The hessian matrix $\bar{H}$ of the lagrangian function has the following form:

$$\bar{H} = H + \begin{pmatrix} \bar{Y} & 0 & \bar{Z} \\ 0 & \bar{T} & 0 \\ \bar{Z}^t & 0 & \bar{S} \end{pmatrix}, \tag{6.83}$$

where the $N^2 \times N^2$ matrices $\bar{Y}$, $\bar{Z}$, $\bar{S}$ and the $\tau(N) \times \tau(N)$ matrix $\bar{T}$ are given by:

$$\bar{Y} = diag\left(h^2 q_{ij} \frac{\partial^2 d(x_{ij}, y_{ij}, u_{ij})}{\partial y_{ij}^2}\right), \quad (i, j) \in I(\Omega), \tag{6.84}$$

$$\bar{S} = diag\left(-hq_{ij} \frac{\partial^2 d(x_{ij}, y_{ij}, u_{ij})}{\partial u_{ij}^2}\right), \quad (i, j) \in I(\Omega), \tag{6.85}$$

$$\bar{Z} = diag\left(h^2 q_{ij} \frac{\partial^2 d(x_{ij}, y_{ij}, u_{ij})}{\partial y_{ij} \partial u_{ij}}\right), \quad (i, j) \in I(\Omega), \tag{6.86}$$

$$\bar{T} = diag\left(-hq_{ij} \frac{\partial^2 b_1(x_{ij}, y_{ij})}{\partial y_{ij}^2}\right), \quad (i, j) \in I(\Gamma_\alpha). \tag{6.87}$$

If $\Gamma_\alpha = \emptyset$, $\Gamma_\beta = \Gamma$, $\bar{H}$ becomes a $2N^2 \times 2N^2$ matrix with the following form:

$$\bar{H} = H + \begin{pmatrix} \bar{Y} & \bar{Z} \\ \bar{Z}^t & \bar{S} \end{pmatrix}$$

For convenience, the numerical results reported in the following of this section for all the test problems are referred to the fixed stepsize $h = 1/(N+1)$, with $N = 99$ and in some cases also with $N = 199$.

Problem 6.2.1 (Example 1 in [56])

We consider the following elliptic control problem with Dirichlet boundary conditions ($\Gamma_\alpha = \emptyset$): minimize the cost functional (6.61) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) - y(x) + y(x)^3 = u, & \begin{array}{l} y(x) \le 0.185, \quad 1.5 \le u(x) \le 4.5, \\ y_d(x) = 1 + 2(x_1(x_1-1) + x_2(x_2-1)), \end{array} \\ \text{on } \Gamma: & y(x) = 0, & \quad u_d(x) \equiv 0, \alpha = 0.001. \end{array}$$

The discretization techniques lead to a NLP problem.

Problem 6.2.1	
Variables	$2N^2$
Constraints	N^2
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	0
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	$2N^2$
Nonzeros $\nabla g_1(x)$	$6N^2 - 4N$

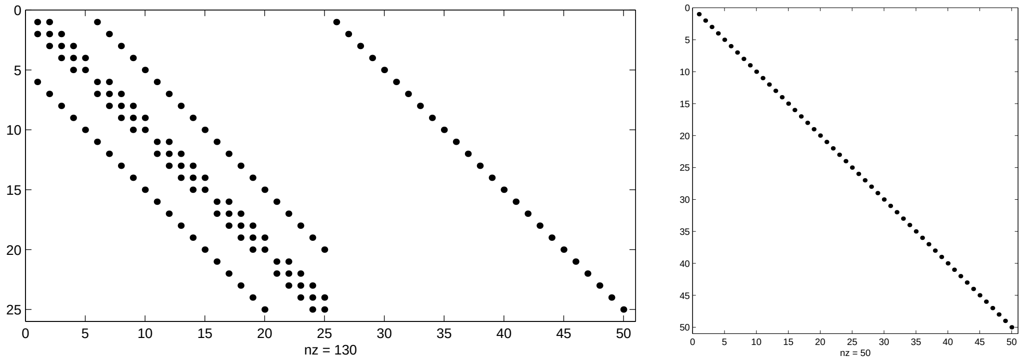
In Figure 6.11, the structure of the jacobian matrix J and that of the hessian matrix $\bar{H}$ of the lagrangian function are reported (for $N = 5$). Since $d_u(x, y, u) = 1$ and $\alpha > 0$, we have that

$$u_{ij} = \left\{ \begin{array}{ll} q_{ij} \cdot 10^3 & \text{if } q_{ij} \cdot 10^3 \in (1.5, 4.5) \\ 1.5 & \text{if } q_{ij} \cdot 10^3 \le 1.5 \\ 4.5 & \text{if } q_{ij} \cdot 10^3 \ge 4.5 \end{array} \right\} \tag{6.88}$$

The state constraint is active at the center $(0.5, 0.5)$. For $N = 99$, $F(\bar{y}, \bar{u}) = 0.0621615$; for $N = 199$, $F(\bar{y}, \bar{u}) = 0.0644263$.

Problem 6.2.2 (Example 2 in [56])

The data are the same of the previous problem, but in this case $\alpha = 0$. The matrix J has the same structure than in the problem 6.2.1 (see Figure 6.11); the hessian matrix $\bar{H}$ of the lagrangian function is a diagonal matrix, but the diagonal entries of $\bar{H}$ corresponding to $u_{ij}, \quad (i,j) \in I(\Omega)$ are equal to 0 (see Figure 6.12).



Jacobian matrix J Hessian matrix $\bar{H}$

Figure 6.11: Problem 6.2.1

Problem 6.2.2	
Variables	$2N^2$
Constraints	N^2
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	0
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$6N^2 - 4N$

Since $\alpha = 0$, we obtain a bang–bang control, having the following form:

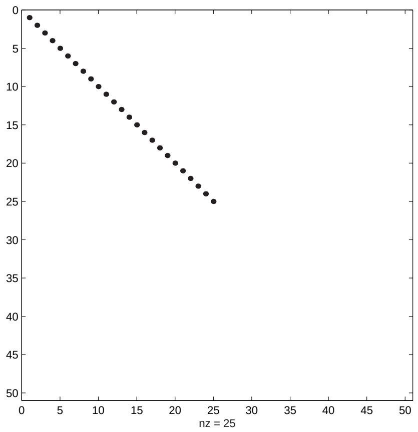
$$\bar{u}(x) = \left\{ \begin{array}{ll} 1.5 & \text{if } \bar{q}(x) < 0 \\ 4.5 & \text{if } \bar{q}(x) > 0 \end{array} \right\}$$

For $N = 99$, $F(\bar{y}, \bar{u}) = 0.0564479$; for $N = 199$, $F(\bar{y}, \bar{u}) = 0.0586978$.

Problem 6.2.3 (Example 3 in [56])

We consider the following elliptic control problem with Dirichlet boundary conditions: minimize the functional (6.61) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) - \exp(y(x)) = u, & \begin{array}{l} y(x) \le 0.11, \quad -5 \le u(x) \le 5, \\ y_d(x) = \sin(2\pi x_1)\sin(2\pi x_2), \end{array} \\ \text{on } \Gamma: & y(x) = 0, & \quad u_d(x) \equiv 0, \alpha = 0.001. \end{array}$$



Hessian matrix $\bar{H}$

Figure 6.12: Problem 6.2.2

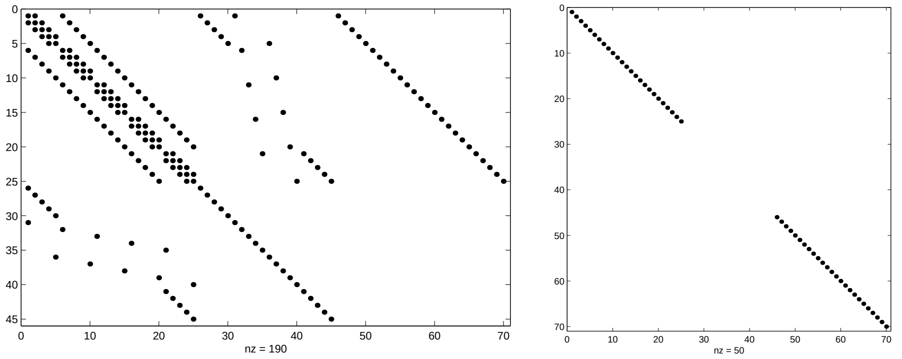
The structure of the jacobian matrix J and of the hessian matrix $\bar{H}$ for the discretized NLP problem is the same of the problem 6.2.1 (see Figure 6.11).

Problem 6.2.3	
Variables	$2N^2$
Constraints	N^2
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	0
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	$2N^2$
Nonzeros $\nabla g_1(x)$	$6N^2 - 4N$

From (6.78), we have

$$u_{ij} = \begin{cases} q_{ij} \cdot 1000 & \text{if } q_{ij} \cdot 1000 \in (-5, 5) \\ -5 & \text{if } q_{ij} \cdot 1000 \leq -5 \\ 5 & \text{if } q_{ij} \cdot 1000 \geq 5 \end{cases}$$

The state constraint is active at the points $(0.26, 0.26)$, $(0.74, 0.74)$. For $N = 99$, at the point $(0.26, 0.26)$, we have $q_{ij} = 0.00858$, $y_{ij} = 0.11$, $y_d(x_{ij}) = 1$, $\mu_{ij} = 0.00251$. Furthermore $q_{i+1j} = 0.00912$, $q_{i-1j} = 0.00926$, $q_{ij+1} = 0.00912$, $q_{ij-1} = 0.00926$. For $N = 99$, $F(\bar{y}, \bar{u}) = 0.110263$; for $N = 199$, $F(\bar{y}, \bar{u}) = 0.1102685$.



Jacobian matrix J Hessian matrix $\bar{H}$

Figure 6.13: Problem 6.2.4

Problem 6.2.4 (Example 4 in [56])

We consider the following elliptic control problem with Neumann boundary conditions: minimize the functional (6.61) subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) - \exp(y(x)) = u, & \begin{array}{l} y(x) \le 0.371, \quad -8 \le u(x) \le 9, \\ y_d(x) = \sin(2\pi x_1)\sin(2\pi x_2), \end{array} \\ \text{on } \Gamma: & \partial_\nu y(x) + y(x) = 0, & \ u_d(x) \equiv 0, \alpha = 0.001. \end{array}$$

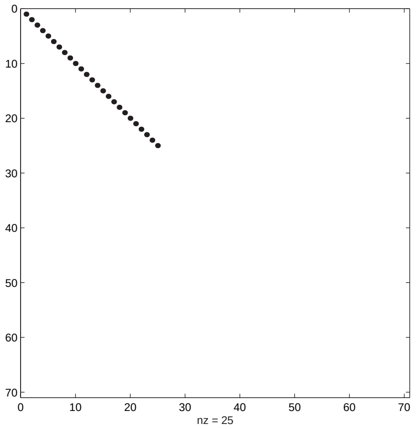
The Figure 6.13 illustrates the structure of the matrices of the NLP problem.

Problem 6.2.4	
Variables	$2N^2+4N$
Constraints	N^2+4N
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	$4N$
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	$2N^2$
Nonzeros $\nabla g_1(x)$	$6N^2+4N$

In this case $\Gamma_\alpha = \Gamma$ and $\Gamma_\beta = \emptyset$. Since $\alpha > 0$ and $d_u(x,y,u) = 1$, we have

$$u_{ij} = \begin{cases} q_{ij} \cdot 1000 & \text{if } q_{ij} \cdot 1000 \in (-8, 9) \\ -8 & \text{if } q_{ij} \cdot 1000 \le -8 \\ 9 & \text{if } q_{ij} \cdot 1000 \ge 9 \end{cases}$$

For $N = 99$, $F(\bar{y}, \bar{u}) = 0.07806389$; for $N = 199$, $F(\bar{y}, \bar{u}) = 0.07842597$. We report also the values of y and u at the point $(0.5, 0.5)$: $y_{ij} = -0.009152$



Hessian matrix $\bar{H}$

Figure 6.14: Problem 6.2.2

$(N = 99)$ and $y_{ij} = -0.008243$ $(N = 199)$ while $u_{ij} = -1.619699$ $(N = 99)$ and $u_{ij} = -1.588730$ $(N = 199)$.

Problem 6.2.5 (Example 5 in [56])

This problem has the same data as the previous one, except for the choice $\alpha = 0$. The jacobian matrix J has the same structure of that in Figure 6.13; the hessian matrix $\bar{H}$ of the lagrangian function has diagonal entries corresponding to the variables u_{ij} equal to zero (see Figure 6.14).

Problem 6.2.5	
Variables	$2N^2 + 4N$
Constraints	$N^2 + 4N$
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	$4N$
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	N^2
Nonzeros $\nabla g_1(x)$	$6N^2 + 4N$

In this case, the optimal control is a bang–bang control having the form:

$$\bar{u}(x) = \begin{cases} -8 & \text{if } \bar{q}(x) < 0 \\ 9 & \text{if } \bar{q}(x) > 0 \end{cases}$$

For $N = 99$, $F(\bar{y}, \bar{u}) = 0.0526639$.

Problem 6.2.6 (Example 4.2 in [57])

We consider the following elliptic control problem with Neumann boundary conditions: minimize the functional

$$\int_\Omega (Mu(x)^2 - Ku(x)y(x))dx \tag{6.89}$$

subject to

$$\begin{array}{lll} \text{on } \Omega: & -\Delta y(x) = y(x)(a(x) - u(x) - by(x)) & y(x) \leq \psi(x), \\ & & u_1 \leq u(x) \leq u_2, \\ \text{on } \Gamma: & \partial_\nu y(x) = 0. & \end{array} \tag{6.90}$$

where

$$a(x) = 7 + 4\sin(2\pi x_1 x_2) \tag{6.91}$$

$$b = 1, \quad M = 1, \quad K = 0.8, \quad u_1 = 1.7, \quad u_2 = 2, \quad \psi(x) = 7.1 \quad .$$

The discrete Neumann conditions

$$y_{ij}^\nu = 0 \qquad (i, j) \in I(\Gamma)$$

suggest to reduce the number of variables $y_{ij}, \quad (i, j) \in I(\Gamma) \cup I(\Omega)$. In other words, from the equality constraints (6.68), we obtain

$$\begin{aligned} &y_{0j} = y_{1j}, \\ &y_{N+1j} = y_{Nj}, \\ &y_{i0} = y_{i1}, \\ &y_{iN+1} = y_{iN}. \end{aligned}$$

Thus the jacobian matrix J is an $N^2 \times 2N^2$ matrix with the form

$$J = [\tilde{Y} + D \quad \bar{E}]$$

where $\tilde{Y}$ is an $N \times N$ block tridiagonal matrix with the off diagonal block equal to $-I_N$ and the diagonal block of the form

$$\tilde{Y}_{11} = \tilde{Y}_{NN} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 3 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 3 & -1 \\ & & & -1 & 2 \end{bmatrix},$$

$$\tilde{Y}_{ii} = \begin{bmatrix} 3 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 3 \end{bmatrix}, \qquad i = 2, \dots, N-1.$$

Furthermore, the matrices D and $\bar{E}$ are as in (6.81). The matrix H has the form

$$\begin{pmatrix} 0_{N^2} & -Kh^2 I_{N^2} \\ -Kh^2 I_{N^2} & 2h^2 M I_{N^2} \end{pmatrix}$$

and the matrix $\bar{H}$ is equal to

$$\bar{H} = H + \begin{pmatrix} \bar{Y} & \bar{Z} \\ \bar{Z}^t & \bar{S} \end{pmatrix}$$

where $\bar{Y}$, $\bar{Z}$, and $\bar{S}$ are as in (6.83) (in this case $\bar{S}$=0). The structures of matrices J and $\bar{H}$ are depicted in Figure6.15.

Problem 6.2.6	
Variables	$2N^2$
Constraints	N^2
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	0
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	$4N^2$
Nonzeros $\nabla g_1(x)$	$6N^2 - 4N$

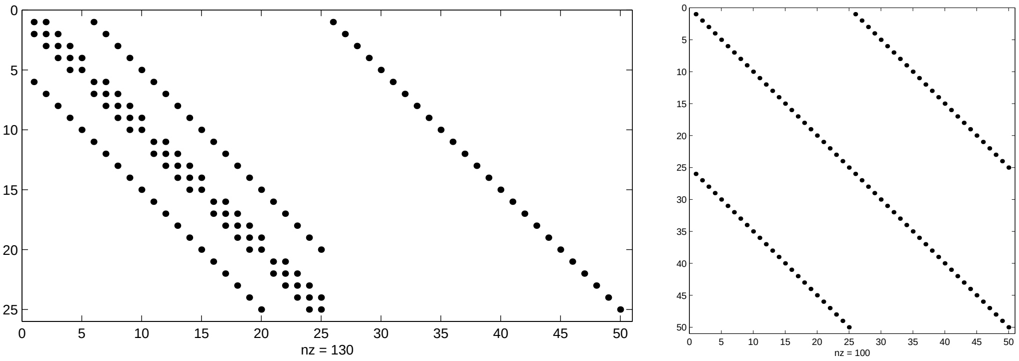
The discretized problem is again a NLP problem.
The state variable attains its upper bound at the two points $(0.21, 0.99)$ and $(0.99, 0.21)$ close to the boundary. For $N = 99$, $F(\bar{y}, \bar{u}) = -6.576428$; for $N = 199$, $F(\bar{y}, \bar{u}) = -6.620092$.

Problem 6.2.7 Example 4.2 in [57]

The problem has the same data of the previous problem, but in this case we choose

$$b = 1, \quad M = 0, \quad K = 1, \quad u_1 = 2, \quad u_2 = 6, \quad \psi(x) = 4.8 \quad .$$

The structure of the jacobian matrix J is the same of the previous problem (see Figure 6.15). The matrix $\bar{H}$ has the diagonal entries corresponding to



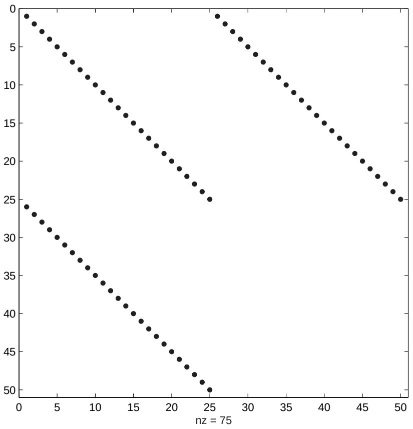
Jacobian matrix J Hessian matrix $\bar{H}$

Figure 6.15: Problem 6.2.6

the variables u_{ij} equal to zero (see Figure 6.16).

Problem 6.2.7	
Variables	$2N^2$
Constraints	N^2
Upper bounds	$2N^2$
Lower bounds	N^2
Linear equalities	0
Nonlinear equalities	N^2
Nonzeros $\nabla^2 f(x)$	$3N^2$
Nonzeros $\nabla g_1(x)$	$6N^2 - 4N$

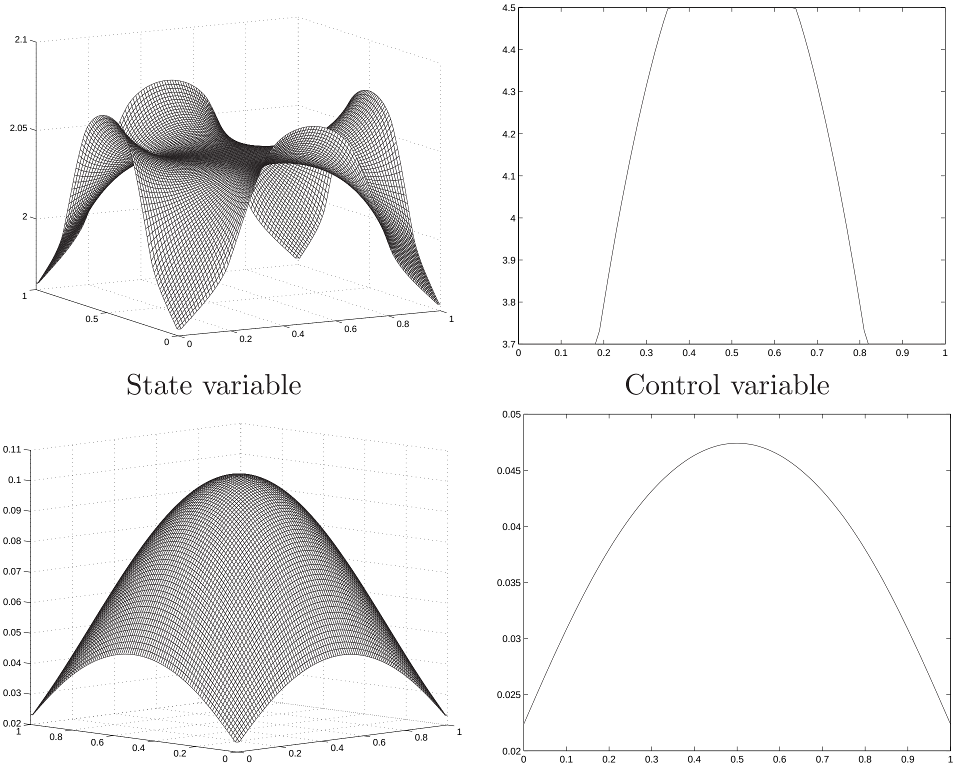
In this case, the optimal control is a bang–bang control. For $N = 99$, $F(\bar{y}, \bar{u}) = -18.73615$; for $N = 199$, $F(\bar{y}, \bar{u}) = -18.86331$.



Hessian matrix $\bar{H}$

Figure 6.16: Problem 6.2.7

6.3 Figures



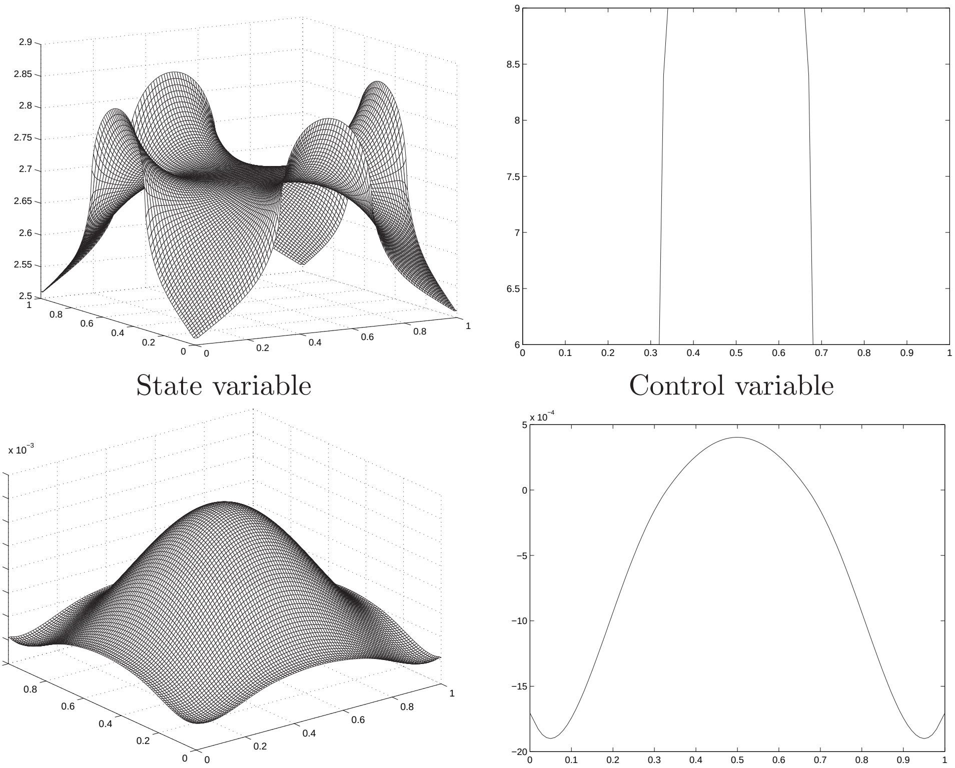
State variable

Control variable

Adjoint variable

Switching function

Figure 6.17: Problem 6.1.1



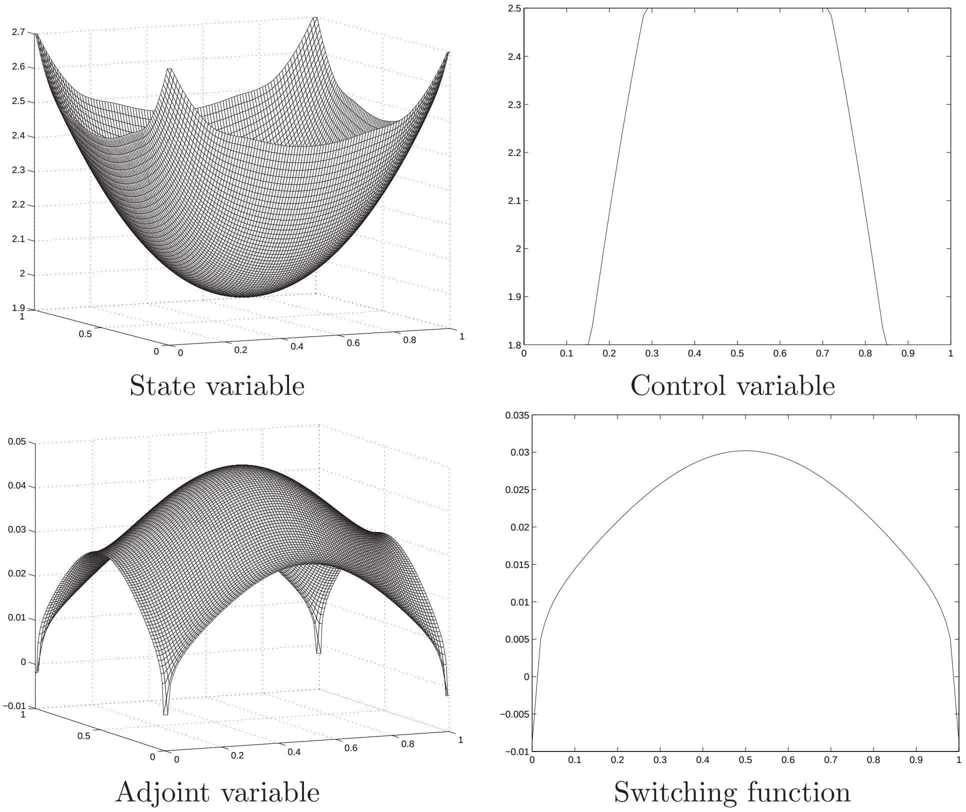
State variable

Control variable

Adjoint variable

Switching function

Figure 6.18: Problem 6.1.2



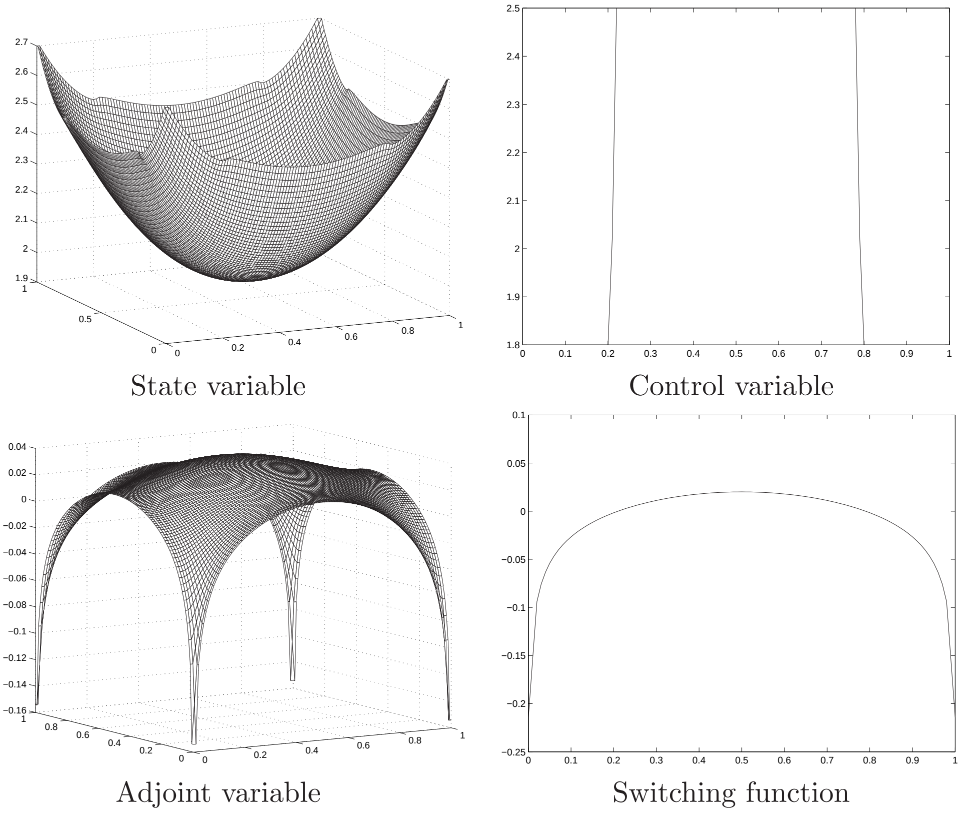
State variable

Control variable

Adjoint variable

Switching function

Figure 6.19: Problem 6.1.3



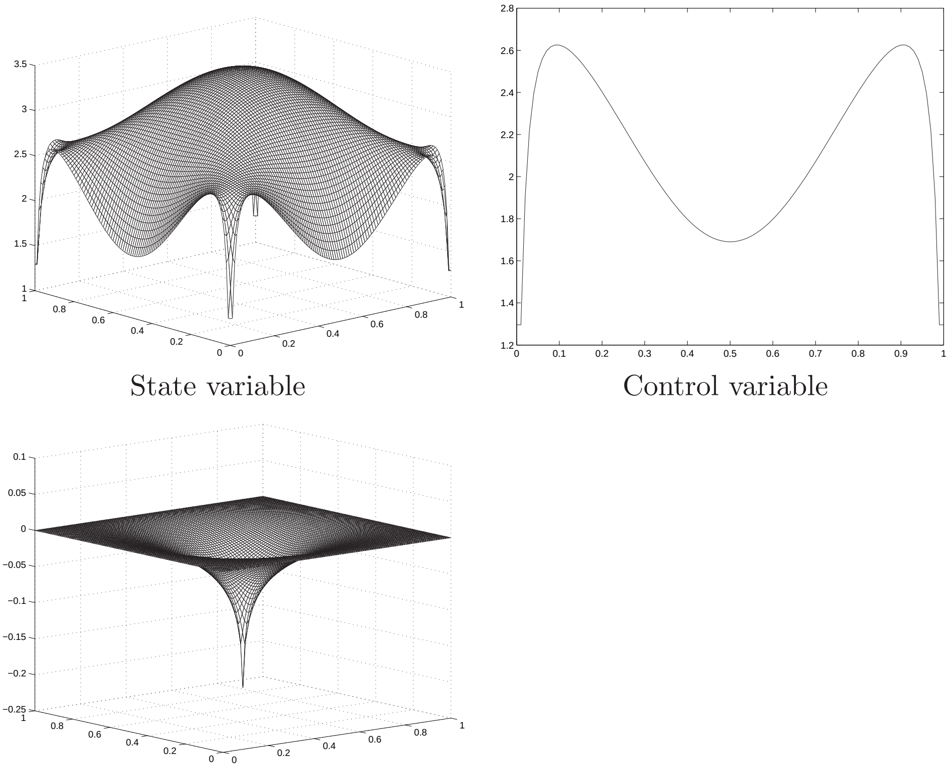
State variable

Control variable

Adjoint variable

Switching function

Figure 6.20: Problem 6.1.4

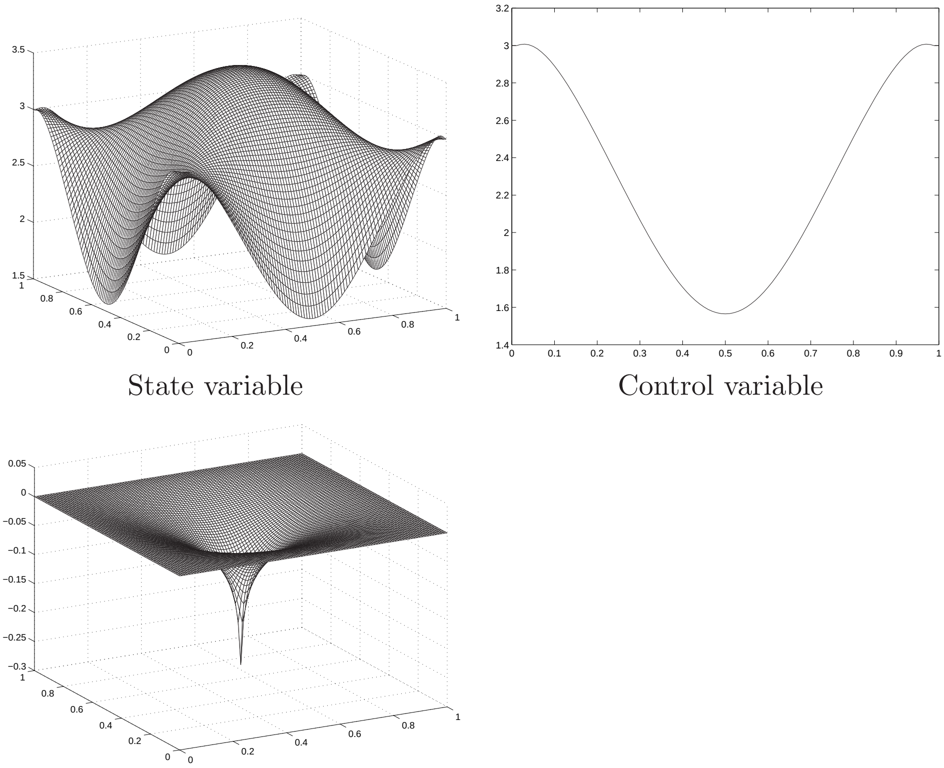


State variable

Control variable

Adjoint variable

Figure 6.21: Problem 6.1.5

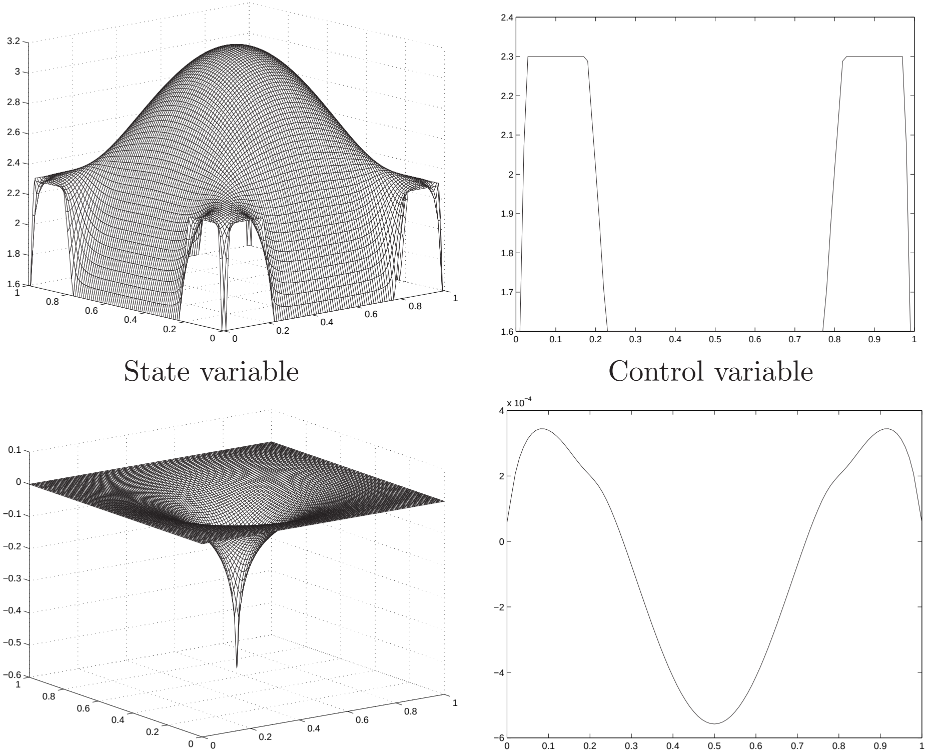


State variable

Control variable

Adjoint variable

Figure 6.22: Problem 6.1.6



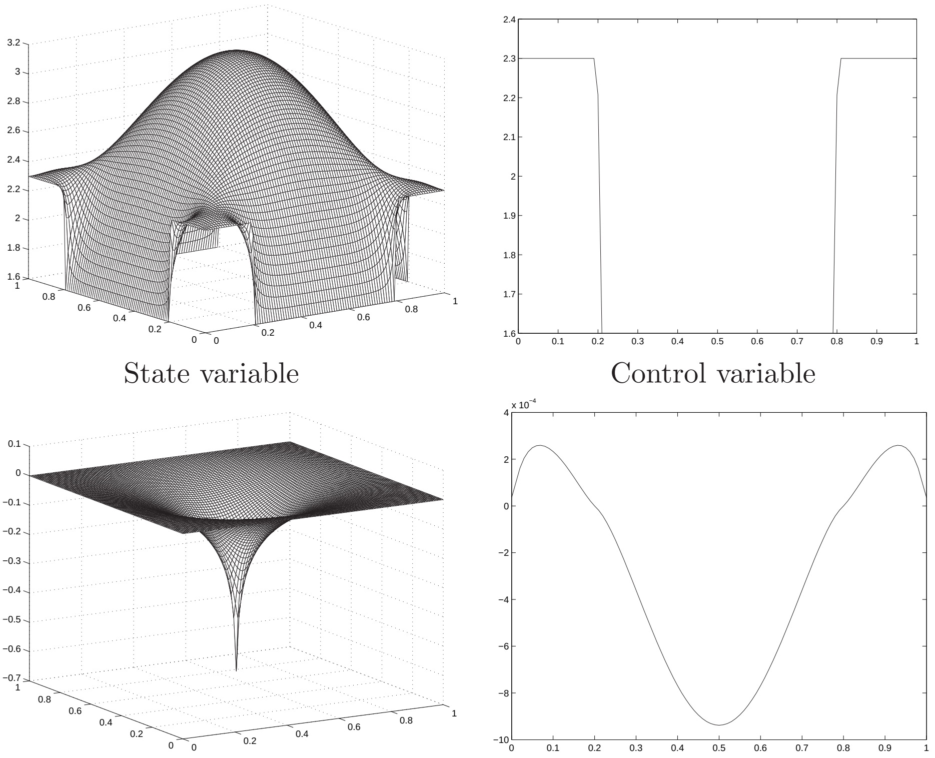
State variable

Control variable

Adjoint variable

Switching function

Figure 6.23: Problem 6.1.7



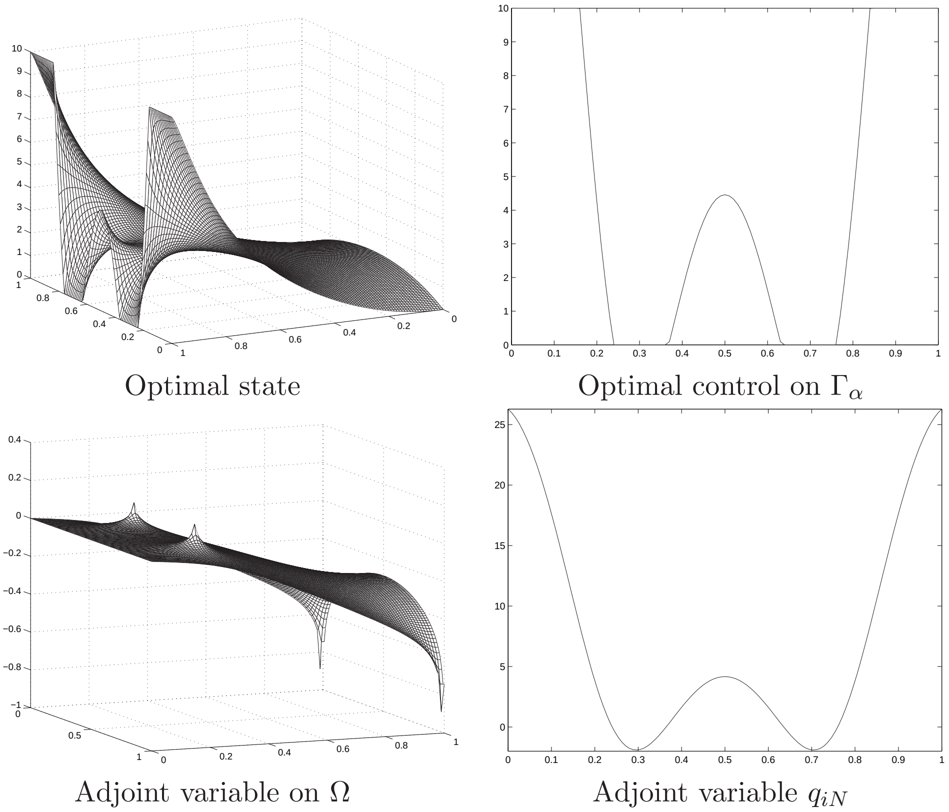
State variable

Control variable

Adjoint variable

Switching function

Figure 6.24: Problem 6.1.8



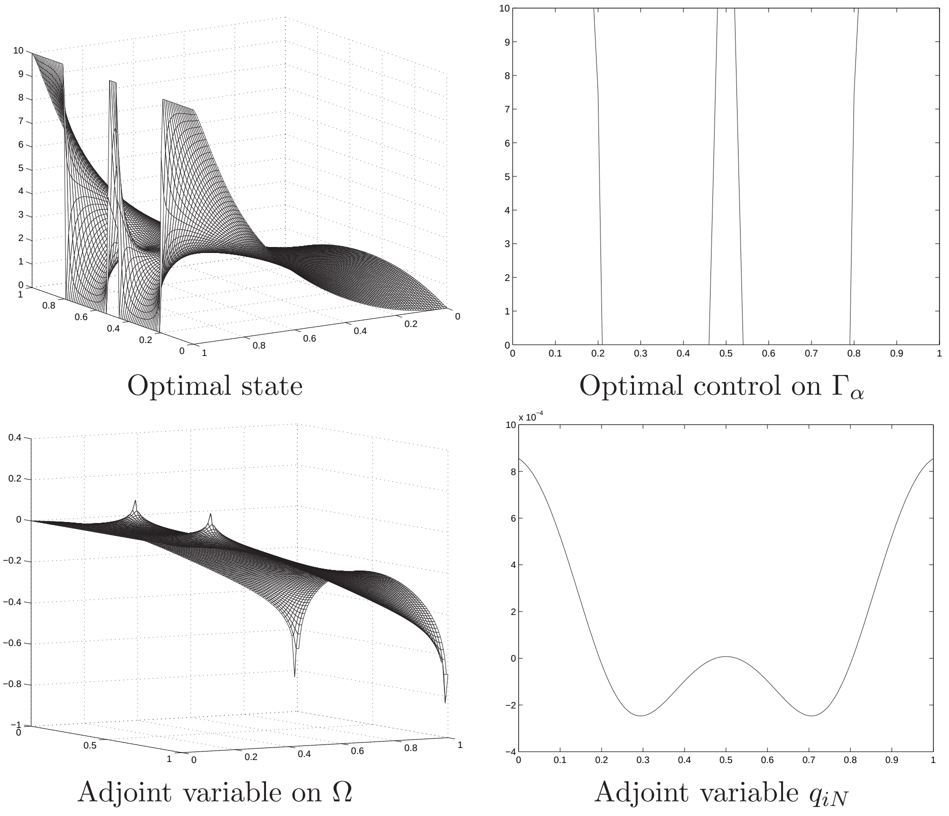
Optimal state

Optimal control on Γ_α

Adjoint variable on Ω

Adjoint variable q_{iN}

Figure 6.25: Problem 6.1.9



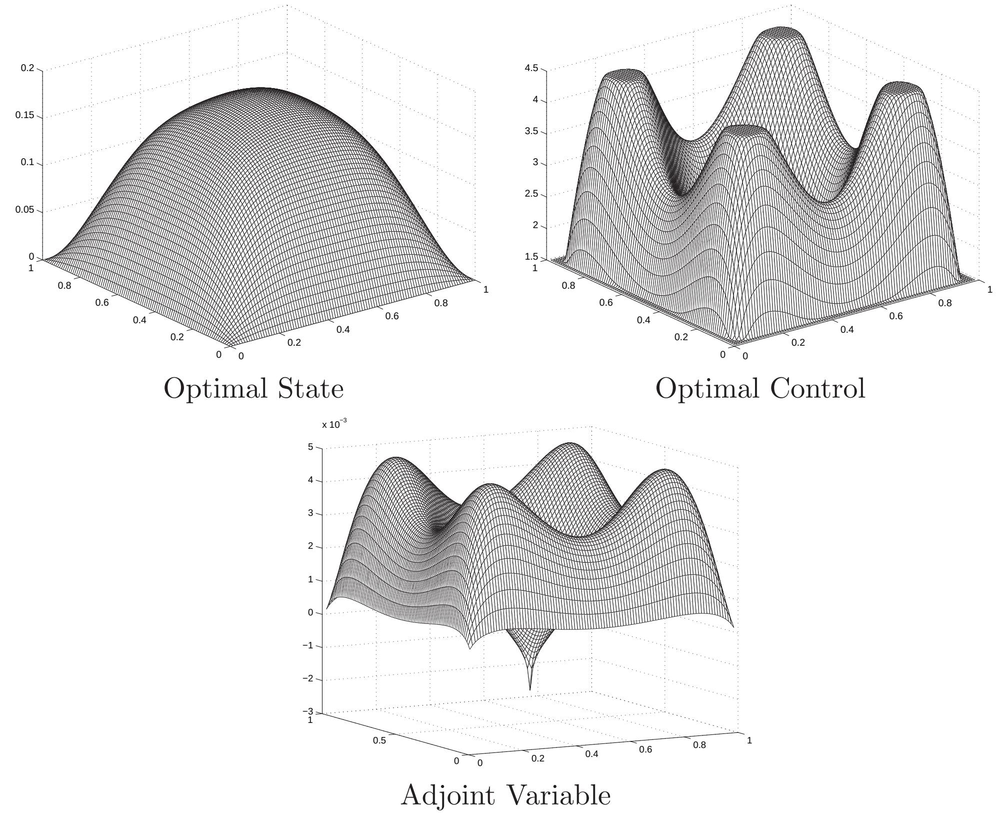
Optimal state

Optimal control on Γ_α

Adjoint variable on Ω

Adjoint variable q_{iN}

Figure 6.26: Problem 6.1.10

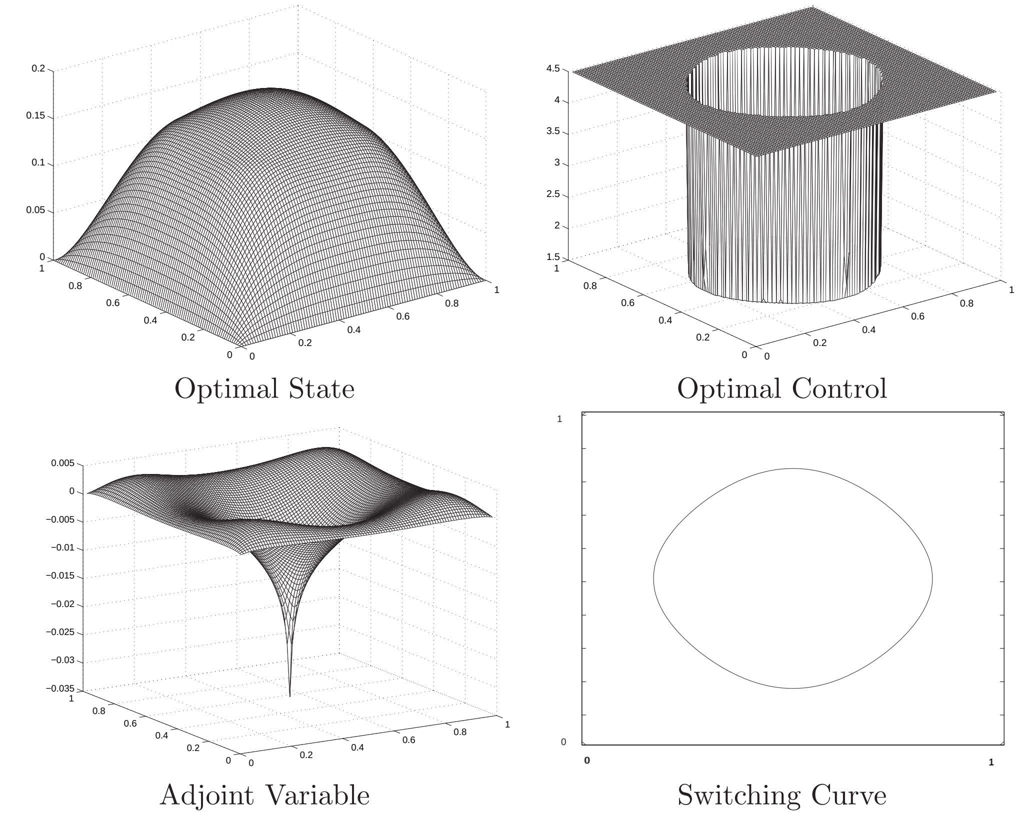


Optimal State

Optimal Control

Adjoint Variable

Figure 6.27: Problem 6.2.1



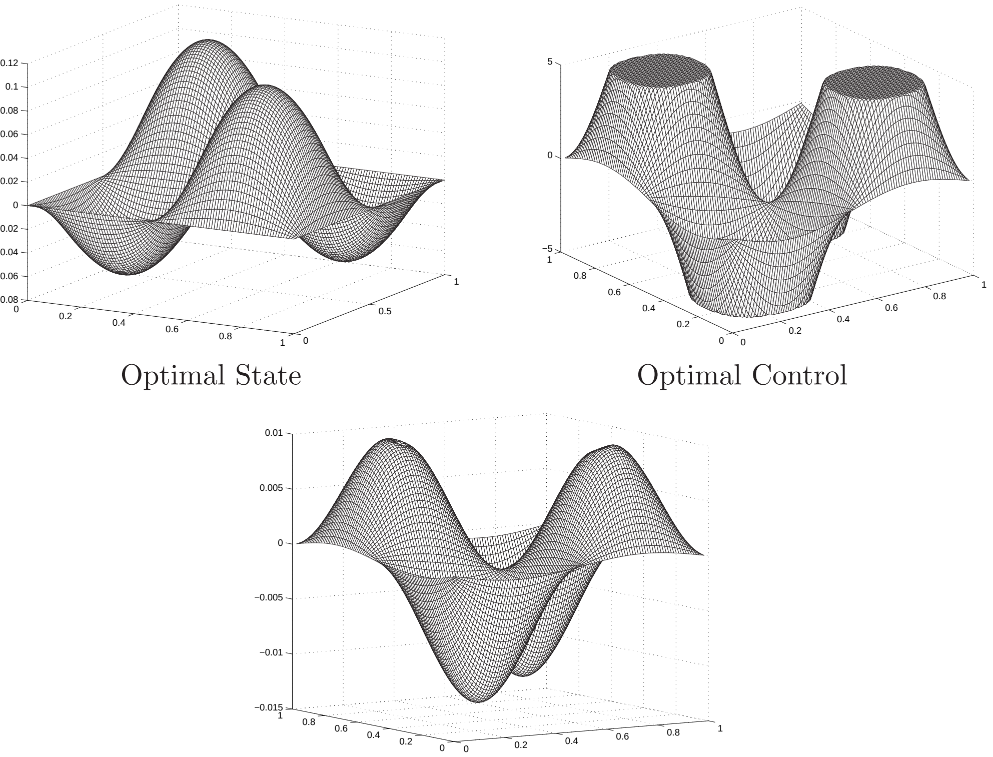
Optimal State

Optimal Control

Adjoint Variable

Switching Curve

Figure 6.28: Problem 6.2.2

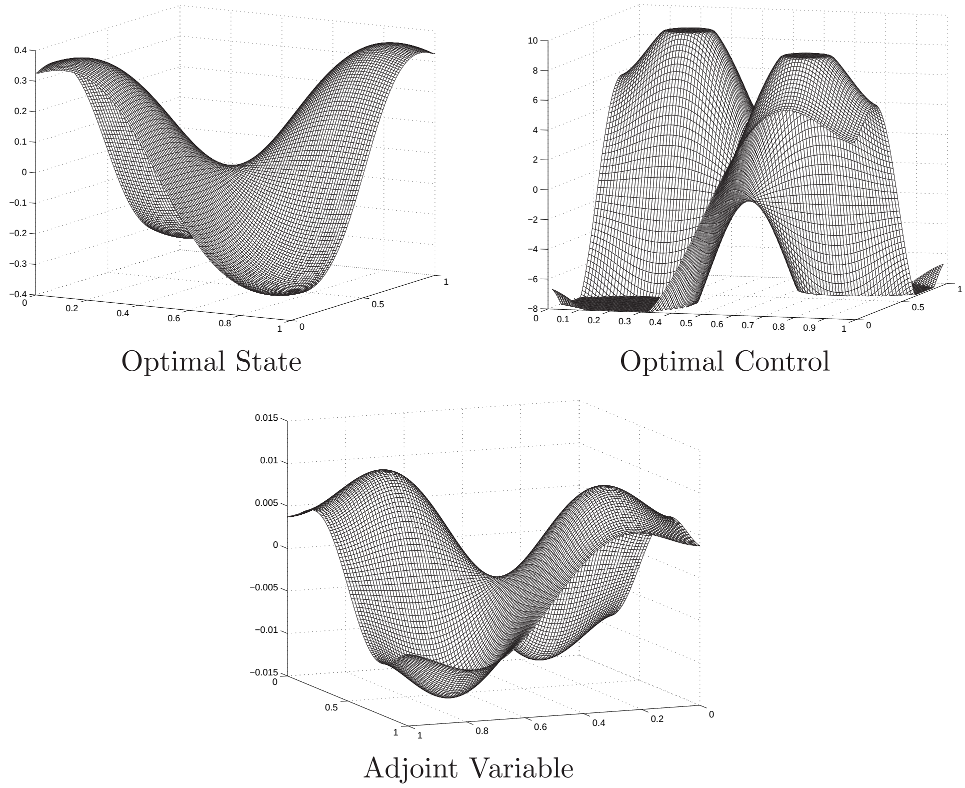


Optimal State

Optimal Control

Adjoint Variable

Figure 6.29: Problem 6.2.3

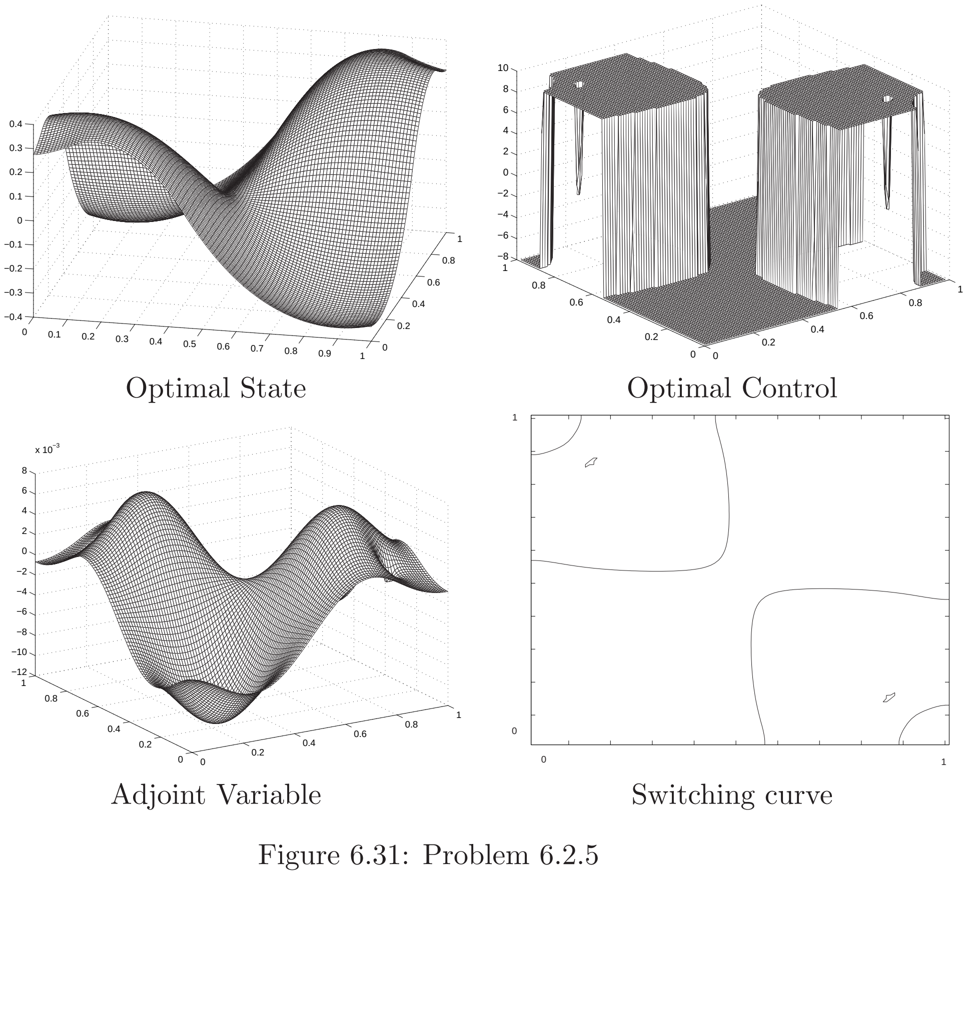


Optimal State

Optimal Control

Adjoint Variable

Figure 6.30: Problem 6.2.4



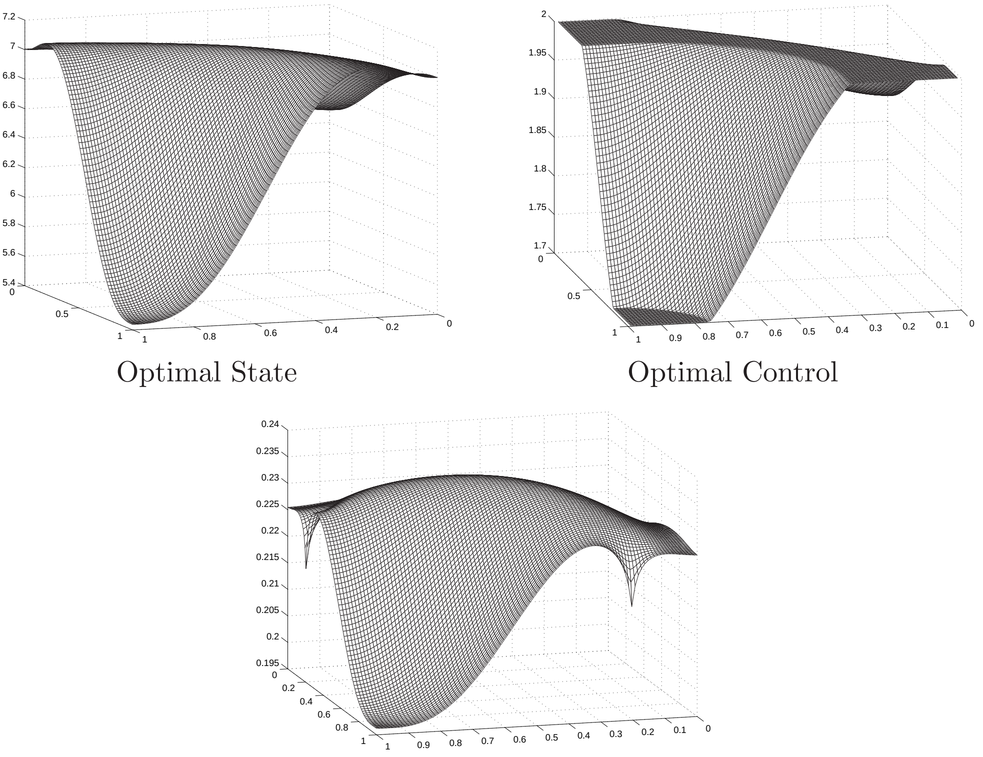
Optimal State

Optimal Control

Adjoint Variable

Switching curve

Figure 6.31: Problem 6.2.5

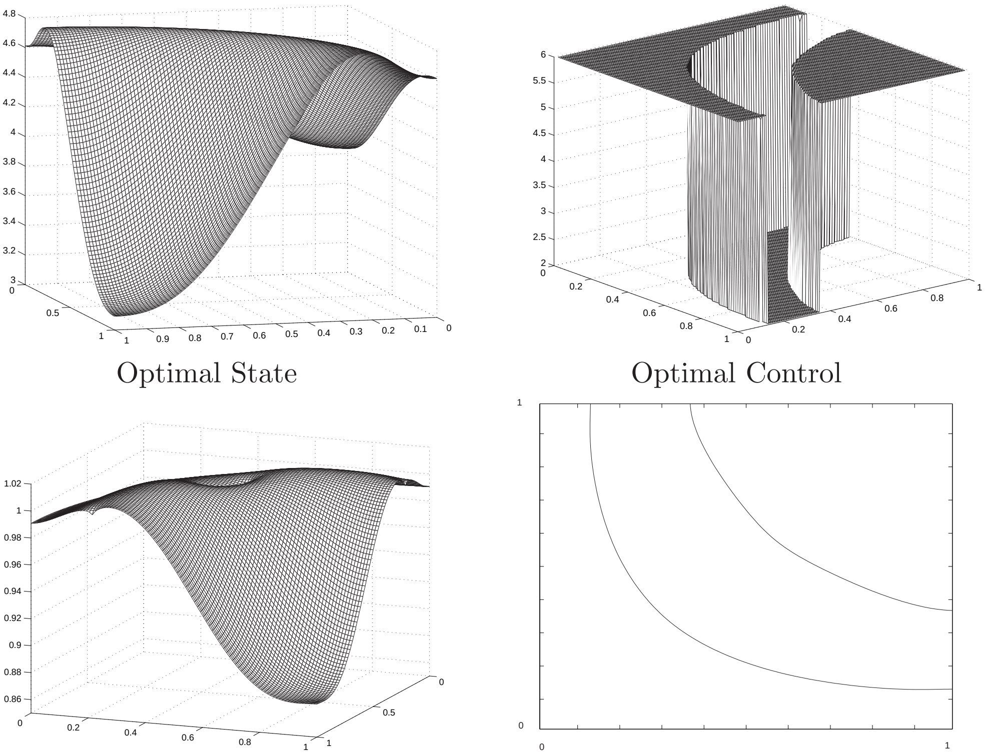


Optimal State

Optimal Control

Adjoint Variable

Figure 6.32: Problem 6.2.6



Optimal State

Optimal Control

Adjoint Variable

Figure 6.33: Problem 6.2.7

Chapter 7

Numerical experience

The nonlinear programming problems problems described in the previous section constitute a test set for the optimization codes, and they also are very interesting from a strictly numerical point of view for their structure and for the possibility to use them in order to test the numerical stability and efficiency of the algorithms. In this chapter we present the numerical results of the Algorithm 4.1 on this test set, by increasing the number of mesh points. Furthermore, we report the value of the cost functional for any considered case.

We experimented also the influence of the nonmonotone choices presented in Section 4.3 in both cases of direct and iterative inner solver.

7.1 Implementation of the algorithm

The Algorithm 4.1 has been implemented in Fortran 90 programming language in four different versions, for each of the inner solvers described in Section 4.2.

In case of the direct solution of the perturbed Newton equation described in section 4.2.1, we choose as linear solver the subroutine MA27 of the Harwell Subroutine Library and we refer to this version as IP-MA27. The version with the Hestenes multipliers' method (Section 4.2.2) as inner solver is called IP-Hestenes, while the two different implementations of the preconditioned conjugate gradient method (Section 4.2.3) yield the codes IP-PCG1 and IP-PCG2 respectively.

All the versions deal with a programming problem of the form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & g_1(x) = 0 \\ & \tilde{g}_2(x) \geq 0 \\ & -P_l x + l \geq 0 \\ & P_u x - u \geq 0 \end{array} \tag{7.1}$$

where P_l end P_u are two diagonal rectangular matrices of size $nl \times n$ and $nu \times n$ respectively, where nl and nu indicate the number of the components of x bounded below and above respectively, and $\tilde{g}_2 : \mathbb{R}^n \to \mathbb{R}^{\tilde{m}}$ are the inequality constraints which are not simple bound. Following this notation, the total number of the inequality constraints is $m = \tilde{m} + n_l + n_u$.
The rectangular matrices P_l end P_u have unitary diagonal entries corresponding to the component of x which is bounded below and above respectively. The vectors $l \in \mathbb{R}^{nl}$ and $u \in \mathbb{R}^{nu}$ are the lower and upper bounds.
The initial values for the multipliers and for the slack variables are set to 1 while the value $(x_0)_i$ are set equal to zero if the i–th component x_i is a free variable, equal to $(u_i + l_i)/2$ if x_i is bounded above and below, and equal to $u_i - 1$ or $l_i + 1$ if x_i is bounded above or below respectively. In the following, we specify if different choices for the starting values have been made.
For the codes employing an iterative method as inner solver, the initial value of the inner iterations has been fixed equal to the null vector.
All the results in this section have been obtained with the choice $\mu_k = s_k^t w_k/\sqrt{m}$: even if the choice of the perturbation parameter is crucial, in these test problems the two safeguard values of the perturbation parameter μ^1 and μ^2 of formula (3.39) are very close (indeed we have $s_k^t w_k \sim \|H(v_k)\|$), and there is no significant difference in the results, in terms of iterations number and execution time.
Moreover, the maximum value of inner iterations has been set to 15 for the IP-Hestenes code, to neq for IP-PCG1 and to $n + neq$ for IP-PCG2.
The other settings are described in Section 4.1.
An explicit computation of the matrices $Q = A + \chi BB^t$, $B^t B$ and of the preconditioner $\bar{M}$ is needed for the factorization, which is performed in two phases: the symbolic factorization exploiting only the structure of the matrices and can be made once before the first iteration, and the actual factorization performed at each iteration. As explained in Section 4.2.2, for the IP-Hestenes and IP-PCG1 codes the structure of the matrices Q and $B^t B$ respectively is computed with a preprocess routine. This preprocess is not needed for the code IP-PCG2, since it does not require the computation of a matrix–matrix product.

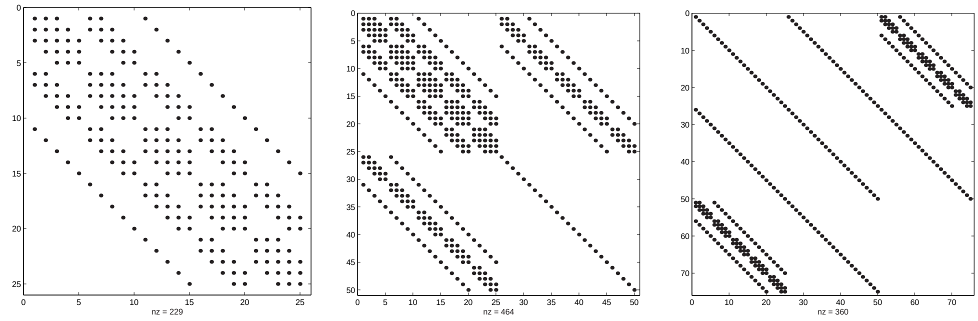
We declare a failure of the algorithm when a maximum number of iterations, fixed to 1500 is reached or when the backtracking procedure produces a damping parameter smaller than 10^{-8}. e *The test problems*

We will refer to the test problems of the previous chapter in the following way: 6.1.1-199 indicates the test problem 6.1.1 with a mesh of $N = 199$ point per axis, etc. Following this notation, 6.1.*-* are related to the boundary control problems, while the script P6.2.*-* refers to the distributed case.
The minimum values of the objective functional are listed in Tables 7.4 and 7.5. The differences on the minimum values obtained by the different solvers are not significant, of the order of 10^{-8}.
In Tables 7.1 and 7.2 the test problems are described: for each test problem the number of primal variables n, the number of equality (neq) constraints, the number of lower (nl) and upper (nu) are reported, while the last two columns are related to the number of nonzero entries of the jacobian (nnziac) and hessian of the lagrangian (nnzhess) matrices.
In Tables 7.1 and 7.2 we have taken into account that some test problems have the same structure.

We observe that the inequality constraints are only box constraints, thus the matrix $CS^{-1}WC^t$ is a diagonal matrix and the computation of the block A in the condensed system (3.33) is inexpensive.
The sparsity pattern of the matrices A, B^tB and of the preconditioner $\bar{M}$ is showed in figure 7.1 for the test problem 6.2.6 with $N = 5$. In Table 7.3 the number of nonzero entries of one triangular part (including the diagonal elements) of the matrices Q, B^tB and of the preconditioner $\bar{M}$ is reported in the columns "nnzhes", "nnzpcg1" and "nnzpcg2" respectively, while the number of nonzero entries of the Cholesky factor is listed in the columns "Lhes", "Lpcg1" and "Lpcg2". The different values of the meshpoint number is reported in the column "Grid". It can be observed that the number of nonzero entries in the Cholesky factor is quite similar in the three cases. In the two cases IP-Hestenes and IP-PCG1, even if a matrix–matrix product is involved, the matrices $A + \chi BB^t$ and B^tB have a density at most equal to 0.1%, while the ratio of the nonzero entries of the Cholesky factor is at most equal to 15.3%.

7.2 The results

In tables 7.6, 7.7, 7.8, 7.9 and 7.10 we compare the performances of the different versions of the code implementing the iterative inner solvers, while in tables 7.12 and 7.11 there are the data related to the code with the di-



IP-Hestenes matrix IP-PCG1 matrix IP-PCG2 matrix

Figure 7.1: Sparsity pattern of the matrix factorized by the codes. The figure refers to problem 6.2.6

rect inner solver. Our comparison takes into account the number of outer iterations, reported in the column "it.", where in brackets the number of inner iterations is also reported, and the execution time of the algorithms. The times are reported in seconds in the column "prep.+iter", specifying the two partial times, the one employed by the preprocess routine in the computation of the matrices structure and the one needed for the iterations. The total CPU time is listed in the column "total".

The symbol "*" indicates that the algorithm failed because of a too small damping parameter which produces a stagnation of the iterates. The symbol "i" denotes that the maximum number of iterations (fixed to 1000) has been reached and "m" indicates that the needs of memory is grater than the available memory.

The codes ran on a workstation HP zx6000 with an Intel Itanium2 processor 1.3 GHz with 2Gb of RAM and they have been compiled with a "+O3" optimization level of the HP compiler.

In the tables 7.12 and 7.11 only the successful tests have been reported in the boundary and distributed case. In any other case, we observed a failure of the algorithm after a few iterates, due to the fill-in of the Cholesky factor which exceeds the available memory. Indeed, the Gauss factor computed by the subroutine MA27 not only depends on the matrix structure, and at each iteration the fill-in can change.

In the iterative case, we can observe that the more expensive computational task for the codes IP-Hestenes and IP-PCG1 is the preprocess phase. We can also notice that the preprocess time is smaller for the IP-PCG1 code,

since the size of the matrix B^tB is neq, while Q is an $n \times n$ matrix. This gain in terms of time is more significant when the test problem arises from a distributed control problem, since in such case the number of equality constraints is an half of the number of the variables. On the other hand, beside a "heavy" preprocess phase, the iterations are very fast. This fact could be exploited when different problems with the same structure or the same problem with different parameters have to be solved in sequence.
A further improvement of the iterations times can be obtained as explained in the following section.
In terms of total time, the more effective code is the IP-PCG2, which does not require the preprocess phase and needs almost the same number of outer and inner iterations than the version IP-PCG1.
At each iteration of the IP-PCG2 code, the preconditioner $\bar{M}$ is factorized as explained in section 4.2.3. The size of $\bar{M}$ is $n + neq$, thus the iterations are not so fast as in the two other cases, but the total execution time is smaller since for IP-PCG2 the preprocess is not needed.
Another characteristic of this code is to require a relatively little memory occupancy: this allows us to solve very large scale problems up to one million primal variables.
In table 7.13, the performances of IP-PCG2 are compared in terms of execution time with the direct and iterative versions of Knitro (version 3.1), reported in the columns Knitro-D and Knitro-I respectively, on the test problem 6.1.1. The comparison puts in evidence the good stability and efficiency of IP-PCG2 on this kind of test problems. The faster Knitro version is the one implementing the direct inner solver, but it runs out the memory for a discretization with a number of meshpoints per axis grater than 499. The Knitro-Iterative code can solve problems with a discretization grid up to 699 meshpoints per axis, then it runs out the available memory too. The faster code among the considered three is the IP-PCG2. Our comparison also included the IP-PCG1, whose results are not reported in table 7.13. It showed to be faster than the Knitro codes but slower than IP-PCG2: for example, it solves the problem 6.1.1-499 in 1997 seconds and 6.1.1-899 in 19835 seconds.

7.3 Results in the nonmononone case

In this section we analyse the results produced when the nonmonotone choices explained in section 4.3 are taken.

The direct case

In table 7.14 we consider the code IP-MA27 with nonmonotone backtracking rule and nonmonotone choice of the perturbation parameter. Our goal was to investigate the effects of such choices in some critical cases and for this reason we have considered starting points which lead the algorithm to perform many backtracking loops. We choose $(x_0)_i = 1$ for the problem P6.1.1, $(x_0)_i = 0.01$ for P6.1.2, $(x_0)_i = 0.5$ for P6.1.3 and $(x_0)_i = 3.995$ for P6.1.4 for any $i = 1, ..., n$. In the table, the failures of the algorithm produced by a stagnation of the iterates due to the backtracking reductions are indicated by the symbol "-". The columns with the label "b." refer to the total number of backtracking reductions, while in the columns with the label "it." the total number of iterations ir reported.

We can observe that in some cases the nonmonotone algorithm prevents the algorithm from the failures.

The iterative case

In this case we allow the nonmonotonicity in the inner stopping criterion and in the backtracking rule, while the perturbation parameter is chosen as in the monotone case as $s_k^t w_k/m$.

We experimented different degrees of nononotonicity by varying the parameter N which, following the notation in (2.53), defines the size of the "memory".

In general the experiments have shown that the nonmonotone rules can be useful also in the iterative case, producing a decrease of the number of inner iterations and thus a reduction of the execution time.

This good behaviour has been observed above all in the IP-Hestenes algorithm, as shown in table 7.15, where for different values of nonmonotonicity degree N, we report the number of outer and inner iterations (in brackets) in the columns labelled with the "it." symbol, and the total and iterations time (in brackets) in seconds in the columns "sec".

We observe a reduction of the inner iterations number, while the number of outer iterations is maintained.

An explanation of this fact is that the nonmonotone stopping criterion can avoid unnecessary Hestenes iterations when the system is ill conditioned and the tolerance is small. Indeed, in many of these cases, the first Hestenes step produces a residual which is only a little grater than the adaptive tolerance. Hence, it is forced to execute other steps until the desired tolerance is satisfied, but when the system is ill conditioned, a too small tolerance could not be reached.

This observation seems to be confirmed by the numerical experience, since with a certain degree of nonmonotonicity in many cases the Hestenes inner

solver tends to execute only one step for each outer iteration.
The remarks above, suggested us a variant of the IP-Hestenes code, where, staying in this nonmonotone contest, the number of inner iterations is fixed to 1. Some results of this modification are reported in the table 7.16. We can observe that the number of outer iterations, which now coincides with the number of inner iterations and which is reported in the column "it.", does not increase and the iterations time, reported in brackets in the column "sec." together with the total time, becomes very small.
We experimented the nonmonotone stopping criterion also on the IP-PCG2 code and the results are reported in table 7.17. We can observe that, setting the nonmonotonicity parameter $N = 4$, the iteration time can be shortened up to a 50% of the time needed in the monotone case, but for other values of N, the results of the monotone algorithm can not be improved.

In general, the nonmonotone inner stopping rule with a moderate degree of nonmonotonicity produces a decreasing in the number of the inner iteration, while for larger value of the nonmonotonicity degree, it can happens that this decreasing is offset by an increasing of the number of outer iterations. This suggests that the nonmonotone flavour should be carefully handled, and possibly adapted to each case.

7.4 Tables

P	Grid	n	neq	nu	nl	nziac	nzhess
6.1.1	99	10593	10197	10593	39204	50193	10593
	199	41193	40397	41193	158404	200393	41193
	299	91793	90597	91793	357604	450593	91793
	399	162393	160797	162393	636804	800793	162393
	499	252993	250997	252993	996004	1250993	252993
	599	363593	361197	363593	1435204	1801193	363593
6.1.2	99	10593	10197	10593	39204	50193	10197
	199	41193	40397	41193	158404	200393	40397
	299	91793	90597	91793	357604	450593	90597
	399	162393	160797	162393	636804	800793	160797
	499	252993	250997	252993	996004	1250993	250997
	599	363593	361197	363593	1435204	1801193	361197
6.1.3	99	10593	10197	10593	39204	50193	10593
	199	41193	40397	41193	158404	200393	41193
	299	91793	90597	91793	357604	450593	91793
	399	162393	160797	162393	636804	800793	162393
	499	252993	250997	252993	996004	1250993	252993
	599	363593	361197	363593	1435204	1801193	363593
6.1.4	99	10593	10197	10593	39204	50193	9801
	199	41193	40397	41193	158404	200393	39601
	299	91793	90597	91793	357604	450593	89401
	399	162393	160797	162393	636804	800793	159201
	499	252993	250997	252993	996004	1250993	249001
	599	363593	361197	363593	1435204	1801193	358801
6.1.5 and 6.1.7	99	10197	9801	10197	396	49005	10197
	199	40397	39601	40397	796	198005	40397
	299	90597	89401	90597	1196	447005	90597
	399	160797	159201	160797	1596	796005	160797
	499	250997	249001	250997	1996	1245005	250997
	599	361197	358801	361197	2396	1794005	361197
6.1.6 and 6.1.8	99	10197	9801	10197	396	49005	9801
	199	40397	39601	40397	796	198005	39601
	299	90597	89401	90597	1196	447005	89401
	399	160797	159201	160797	1596	796005	159201
	499	250997	249001	250997	1996	1245005	249001
	599	361197	358801	361197	2396	1794005	358801
6.1.9	119	14637	14518	14280	14637	71519	3840
	179	32757	32578	32220	32757	161279	8460
	279	78957	78678	78120	78957	390879	20160
	379	145157	144778	144020	145157	720479	36870
	479	231357	230878	229920	231357	1150079	58560
	579	337557	336978	335820	337557	1679679	85260
6.1.10	119	14637	14518	14280	14637	71519	3721
	179	32757	32578	32220	32757	161279	8281
	279	78957	78678	78120	78957	390879	19881
	379	145157	144778	144020	145157	720479	36491
	479	231357	230878	229920	231357	1150079	58081
	579	337557	336978	335820	337557	1679679	84681

Table 7.1: Description of the test problems: boundary control

P	Grid	n	neq	nu	nl	nziac	nzhess
6.2.1	99	19602	9801	19602	9801	59202	19602
	199	79202	39601	79202	39601	238402	79202
	299	178802	89401	178802	89401	537602	178802
	399	318402	159201	318402	159201	956802	318402
	499	498002	249001	498002	249001	1496002	498002
6.2.2	99	19602	9801	19602	9801	59202	9801
	199	79202	39601	79202	39601	238402	39601
	299	178802	89401	178802	89401	537602	89401
	399	318402	159201	318402	159201	956802	159201
	499	498002	249001	498002	249001	1496002	249001
6.2.3 and 6.2.4	99	19998	10197	19602	9801	59598	19602
	199	79998	40397	79202	39601	239198	79202
	299	179998	90597	178802	89401	538798	178802
	399	319998	160797	318402	159201	958398	318402
	499	499998	250997	498002	249001	1497998	498002
6.2.5	99	19998	10197	19602	9801	59598	10197
	199	79998	40397	79202	39601	239198	40397
	299	179998	90597	178802	89401	538798	90597
	399	319998	160797	318402	159201	958398	160797
	499	499998	250997	498002	249001	1497998	250997
6.2.6	99	19602	9801	19602	9801	58410	39204
	199	79202	39601	79202	39601	236810	158404
	299	178802	89401	178802	89401	535210	357604
	399	318402	159201	318402	159201	953610	636804
	499	498002	249001	498002	249001	1492010	996004
6.2.7	99	19602	9801	19602	9801	58410	29403
	199	79202	39601	79202	39601	236810	118803
	299	178802	89401	178802	89401	535210	268203
	399	318402	159201	318402	159201	953610	477603
	499	498002	249001	498002	249001	1492010	747003

Table 7.2: Description of the test problems: distributed control.

Problem	Grid	nnzhes	Lhes	nnzpcg1	Lpcg1	nnzpcg2	Lpcg2
6.1.1	99	70783	622759	69991	621571	60786	718637
6.1.2	199	281583	3181444	279195	3179056	241586	3416032
6.1.3	299	632383	8374469	628795	8370881	542386	9084296
6.1.4	399	1123183	16252152	1118395	16247364	9631186	20102932
	499	1753983	26855490	1747995	26849502	1503986	28784753
	599	2524783	41135305	2517595	41128117	2164786	43488232
6.1.5	99	69595	621571	67619	619595	59202	716261
6.1.6	199	279195	3179056	275219	3175080	238401	3411256
6.1.7	299	628795	8370881	622819	8364905	537602	9011520
6.1.8	399	1118395	16247364	1110419	16239388	956802	20093356
	499	1747995	26849502	1738019	26839526	1496002	28772777
	599	2517595	41128117	2505619	41116141	2155202	43473654
6.1.9	119	100315	945546	99720	944951	86156	1029560
6.1.10	179	226075	2541572	225180	2540677	194036	2733190
	279	547675	7167732	546280	7166337	469836	8619291
	379	1009275	14501957	1007380	14500062	865636	15396152
	479	1610875	24901311	1608480	24898916	1381436	26203761
	579	2352475	37810473	2349580	37807578	2017236	48288922
6.2.1	99	126029	715465	67619	619595	78012	735071
6.2.2	199	512029	3409660	275219	3175080	316012	3488866
6.2.3	299	1158029	8900195	622819	8364905	714012	9253530
	399	2064029	20090160	1110419	16239388	1272012	20405866
	499	3230029	28768781	1738019	26839526	1990012	29266787
6.2.4	99	128401	717837	69595	621571	79596	737447
6.2.5	199	516801	3414432	517993	3179056	319196	3493642
	299	1165201	8907367	1166993	8370881	718796	9260706
	399	2073601	20099732	2075994	16247364	1278396	20418142
	499	3242001	28780753	3244994	26849502	3244994	29278763
6.2.6	99	72816	715465	67619	619595	78012	735071
6.2.7	199	295620	3409660	510837	3175080	316012	3488866
	299	1158029	8900195	622819	8364905	714012	9079001
	399	2064029	20090160	1110419	16239388	1272012	20408566
	499	3230029	28768781	1738019	26839526	1990012	29266787

Table 7.3: Nonzero entries of the matrices and of the Choleski factors

Boundary control

Problem	min	Problem	min
6.1.1-99	0.55224625	6.1.6-99	0.09669507
6.1.1-199	0.55436881	6.1.6-199	0.10044221
6.1.1-299	0.55507372	6.1.6-299	0.10170115
6.1.1-399	0.55542568	6.1.6-399	0.10233242
6.1.1-499	0.55580371	6.1.6-499	0.10271175
6.1.1-599	0.55577731	6.1.6-599	0.10296487
6.1.2-99	0.01507867	6.1.7-99	0.32100965
6.1.2-199	0.01560172	6.1.7-199	0.32812152
6.1.2-299	0.01577842	6.1.7-299	0.33050688
6.1.2-399	0.01586721	6.1.7-399	0.33170235
6.1.2-499	0.01592062	6.1.7-499	0.33242047
6.1.2-599	0.01595628	6.1.7-599	0.33289956
6.1.3-99	0.26416255	6.1.8-99	0.24917848
6.1.3-199	0.26728343	6.1.8-199	0.25587655
6.1.3-299	0.26832628	6.1.8-299	0.25812464
6.1.3-399	0.26884799	6.1.8-399	0.25925143
6.1.3-499	0.26916120	6.1.8-499	0.25992872
6.1.3-599	0.26937006	6.1.8-599	0.26038061
6.1.4-99	0.16553111	6.1.9-119	0.25908196
6.1.4-199	0.16778056	6.1.9-179	0.25305430
6.1.4-299	0.16854245	6.1.9-279	0.24894250
6.1.4-399	0.16892441	6.1.9-379	0.24705220
6.1.4-499	0.16915379	6.1.9-479	0.24596630
6.1.4-599	0.16930712	6.1.9-579	0.24526176
6.1.5-99	0.19651967	6.1.10-119	0.15741541
6.1.5-199	0.20077162	6.1.10-179	0.15128350
6.1.5-299	0.20219539	6.1.10-279	0.14694570
6.1.5-399	0.20290856	6.1.10-379	0.14492090
6.1.5-499	0.20333682	6.1.10-479	0.14375680
6.1.5-599	0.20362247	6.1.10-579	0.14299524

Table 7.4: Minimum values of the objective functional: boundary control

Distributed control			
Problem	min	Problem	min
6.2.1-99	0.06216164	6.2.5-99	0.05266390
6.2.1-199	0.06442591	6.2.5-199	0.05293239
6.2.1-299	0.06519262	6.2.5-299	0.05302628
6.2.1-399	0.06557820	6.2.5-399	0.05307458
6.2.1-499	0.06581034	6.2.5-499	0.05310603
6.2.2-99	0.05644747	6.2.6-99	-6.57642757
6.2.2-199	0.05869688	6.2.6-199	-6.62009226
6.2.2-299	0.05946010	6.2.6-299	-6.63464408
6.2.2-399	0.05984417	6.2.6-399	-6.64192346
6.2.2-499	0.06007572	6.2.6-499	-6.64629219
6.2.3-99	0.11026306	6.2.7-99	-18.73618438
6.2.3-199	0.11026872	6.2.7-199	-18.86331163
6.2.3-299	0.11026969	6.2.7-299	-18.90575104
6.2.3-399	0.11027035	6.2.7-399	-18.92698093
6.2.3-499	0.11027047	6.2.7-499	-18.93972227
6.2.4-99	0.07806386		
6.2.4-199	0.07842594		
6.2.4-299	0.07854995		
6.2.4-399	0.07861255		
6.2.4-499	0.07865054		

Table 7.5: Minimum values of the objective functional: distributed control

	IP-Hestenes			IP-PCG1			IP-PCG2	
Problem	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	total
6.1.1 99	29(32)	2.22+2.03	4.25	37(30)	2.21+2.46	4.67	37(72)	5.24
199	54(59)	36.38+22.87	59.25	45(37)	35.81+18.31	54.12	45(95)	38.9
299	181(186)	206.35 +246.8	453.15	52(47)	197.29+68.09	256.38	52(116)	156.49
399	327(341)	833.79+961.08	1794.92	58(53)	758.23+174.82	933.05	58(137)	493.07
499	501(527)	1933.8+2768.7	4702.5	63(59)	1635.64+341.65	1977.37	63(158)	845.76
599	*	*	*	66(62)	1902.21+701.21	2603.55	66(181)	1377.61
6.1.8 99	34(34)	2.2+2.3	4.6	35(39)	2.3+2.4	4.7	35(37)	4.5
199	40(40)	37.8+17.3	55.1	41(45)	37.8+17.3	55.1	41(41)	32.6
299	55(56)	172.8+73.5	246.3	51(55)	201.8+68.13	269.9	50(52)	140.3
399	143(144)	558.1+420.8	979	58(64)	655.9+171.6	827.5	59(60)	441.7
499	197(198)	1418.3+1076.1	2494.4	66(76)	1621.4+364+9	1986.4	70(73)	829.1
599	242(243)	2997.4+2367.5	5364.9	74(87)	3456.5+714.5	4171.1	79(89)	1560.2
6.1.3 99	21(23)	3.02+1.46	4.49	29(36)	2.22+2.05	4.28	28(79)	4.31
199	26(27)	47.83+10.87	58.71	33(42)	45.87+14.46	60.35	33(91)	30.02
299	39(45)	162.15+52.88	215.03	36(47)	194.79+49.7	243.52	37(109)	115.84
399	36(39)	831.0+105.29	936.34	39(54)	617.47+117.91	735.42	38(120)	312.78
499	65(87)	2062.11+360.03	2422.22	42(55)	1522.11+232.93	1755.12	41(146)	535.14
599	*	*	*	44(60)	3928.54+427.12	3928.54	43(159)	925.84
6.1.4 99	31(31)	2.2+2.1	4.3	33(42)	2.3+2.3	4.6	31(44)	4.2
199	38(98)	34.5+18.9	53.5	40(51)	36.6+17.2	53.8	38(59)	31.5
299	41(58)	172.7+56.3	228.9	45(60)	189.9+61.3	251.3	40(59)	114.7
399	43(45)	560.3+125.3	685.7	49(66)	603.6+147.2	750.8	45(67)	341.7
499	*	*	*	51(67)	1499.3+283.7	1783.1	50(76)	601.7
599	*	*	*	69(82)	3621.0+662.8	4284	82(153)	1660.4

Table 7.6: Numerical results: boundary control

	IP-Hestenes			IP-PCG1			IP-PCG2	
Problem	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	total
6.1.5 99	28(28)	2.1+1.9	4	31(31)	2.1+2.1	4.2	28(34)	3.6
199	33(33)	33.8+13.6	47.4	37(37)	35.7+15.2	50.9	32(42)	26
299	40(55)	169.0+54.8	223.8	41(41)	194.9+53.9	248.8	36(50)	99.6
399	45(75)	548.0+137.4	685.4	44(45)	751.0+128.1	879.1	39(62)	298.3
499	49(79)	1380.6+275.1	1655.8	46(47)	1583.6+248.5	1832.1	43(73)	520.2
599	51(126)	2978.9+534.4	3513.3	48(50)	3336.1+456.7	3792.9	46(77)	925.3
6.1.6 99	30(30)	2.1+2.0	4.1	35(37)	2.1+2.4	4.5	30(39)	3.9
199	33(33)	33.2+13.6	46.7	37(41)	35.4+15.4	50.8	32(41)	25.8
299	40(55)	168.3+54.3	222.6	41(47)	231.6+55.8	287.4	37(54)	102.8
399	46(61)	546.9+136.7	683.7	44(50)	634.8+129.4	764.2	40(65)	306.1
499	50(95)	1377.3+288.2	1665.5	47(56)	1587.4+258.6	1846.1	44(75)	533.5
599	51(126)	2971.5+532.8	3504.5	49(59)	3290.6+470.1	3760.8	46(77)	925.2
6.1.7 99	39(39)	2.1+2.7	4.8	42(42)	2.1+2.8	4.9	40(54)	5.2
199	46(46)	33.2+19.0	52.1	46(46)	35.6+18.9	54.5	45(72)	37.4
299	64(99)	168.4+89.2	257.6	52(52)	193.6+69.5	263.1	49(87)	138.7
399	96(141)	549.3+288.5	837.8	55(58)	636.0+160.5	796.5	53(95)	407.9
499	127(169)	1387.3+703.4	2090.7	57(64)	1583.1+311.9	1895.1	52(94)	630
599	159(202)	3020.9+1548.1	4569.1	59(69)	3309.4+563.6	3873.1	52(98)	1051.9
6.1.8 99	40(40)	2.1+2.7	4.8	43(43)	2.1+2.9	5	41(52)	5.3
199	48(48)	33.2+20.5	53.7	50(50)	35.4+20.6	56	47(74)	38.8
299	66(106)	168.9+93.1	292	55(55)	195.3+72.8	268.2	52(90)	146.8
399	101(156)	546.1+304.3	851.4	60(63)	637.0+174.6	808.7	58(105)	447.2
499	133(195)	1404.1+745.6	2149.8	62(70)	1589.4+337.4	1926.9	57(105)	693.1
599	167(364)	3033+1725.1	4758.1	65(76)	3595.4+622.9	4218.4	58(112)	1175.5

Table 7.7: Numerical results: boundary control

	IP-Hestenes			IP-PCG1			IP-PCG2	
Problem	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	total
6.1.9 119	41(41)	4.4+4.3	8.7	45(47)	4.5+4.8	9.3	48(74)	9.6
179	75(87)	17.1+25.7	42.8	51(56)	23.2+16.9	40.1	46(73)	30.1
279	63(63)	133.1+69.4	202.5	57(63)	140.7+62.4	203.4	51(90)	136.2
379	82(97)	448.9+212.5	661.4	62(78)	482.3+161.9	644.2	55(112)	302.1
479	95(185)	1226.6+523.1	1749.7	65(84)	1255.2+341.1	1596.3	58(129)	638.5
579	110(275)	2562+997.8	3560	67(96)	2658.7+570.9	3229.6	61(149)	1545.1
6.1.10 119	44(44)	4.4+4.6	9	49(52)	4.5+5.2	9.8	44(70)	8.9
179	56(56)	22.4+18.4	40.8	59(65)	23.2+19.5	42.7	55(89)	36
279	72(87)	129.1+80.1	209.2	67(79)	140.9+74.3	215.2	63(117)	169.2
379	101(251)	452.9+290.5	743.5	77(100)	483.6+204.9	688.6	74(165)	408.6
479	105(360)	1202.8+629.1	1832	81(113)	1235.4+429.9	1665.3	78(190)	864.7
579	119(374)	2558.6+1123.8	3682.5	86(130)	2660.6+738.9	2299.6	83(224)	2118.1

Table 7.8: Numerical results: boundary control

	IP-Hestenes			IP-PCG1			IP-PCG2	
Problem	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	total
6.2.1 99	23(23)	4.8+2.2	7.1	26(25)	2.5+1.9	4.36	24(23)	3.3
199	28(193)	123.1+26.4	149.5	28(26)	41.5+12.1	53.7	27(26)	22.6
299	*	*	*	30(29)	218.3+41.2	259.5	28(27)	81.2
399	*	*	*	31(56)	706.4+100.9	807.4	29(28)	222
499	*	*	*	32(69)	2166.8+196.3	2363.2	29(28)	351.8
6.2.2 99	28(28)	4.8+2.6	7.5	31(45)	2.5+2.4	4.9	29(28)	3.9
199	31(166)	78.8+25.8	104.6	33(53)	41.7+15.7	57.4	30(29)	24.74
299	*	*	*	34(64)	217.7+51.5	269.2	32(31)	92.8
399	*	*	*	36(95)	704.9+125.7	830.7	33(32)	252.3
499	*	*	*	37(132)	1741.9+252.1	1994.1	33(32)	399.1
6.2.3 99	25(25)	4.8+2.4	7.2	31(26)	2.5+2.2 4.7		25(22)	3.4
199	31(196)	119.0+27.9	147	33(27)	41.5+14.5	55.7	26(23)	21.7
299	43(403)	694.4+131.1	825.6	34(28)	218.4+46.1	264.5	28(25)	80.8
399	89(1184)	2339.9+758.3	3098.2	37(58)	869.4+119.4	988.9	30(27)	229.1
499	*	*	*	36(61)	1742.25+212.91	1955.3	29(26)	350.23
6.2.4 99	24(54)	5.0+2.9	7.9	20(16)	3.08+1.45	4.53	20(38)	3.11
199	27(237)	80.6+29.4	109.9	21(17)	40.82+9.11	49.94	21(37)	19.02
299	35(335)	424.2+108.3	532.5	22(18)	227.65+30.02	257.67	22(39)	68.21
399	36(351)	1480.1+256.3	1736.5	23(19)	752.20+69.30	821.5	23(42)	184.77
499	*	*	*	23(19)	2119.59+127.88	2247.47	23(42)	290.58

Table 7.9: Numerical results: distributed control

	IP-Hestenes			IP-PCG1			IP-PCG2	
Problem	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	time(prep.+iter.)	total	it.(inn.)	total
6.2.599	48(63)	5.0+4.8	9.9	56(152)	2.6+5.2	7.8	47(43)	6.4
199	68(383)	80.7+58.2	138.9	78(712)	52.7+74.7	127.4	65(61)	53.9
299	104(1439)	421.5+403.4	825.4	91(1356)	226.9+320.7	547.7	80(77)	230.9
399	155(2255)	1489.0+1376.1	2865.2	107(1436)	718.6+704.4	1423.1	93(92)	709.7
499	i	i	i	116(3125)	1798.6+2040.4	3839.1	104(104)	1256
6.2.699	28(29)	5.77+2.7	8.48	35(70)	2.46+3.03	5.5	34(122)	6.28
199	48(49)	118.03+25.11	143.17	51(88)	41.25+25.19	66.44	51(178)	53.2
299	81(111)	686.30+131.49	817.99	56(97)	223.61+85.79	309.41	54(177)	173.82
399	102(153)	2292.11+477.5	2769.7	71(130)	727.06+239.67	966.73	64(221)	553.78
499	101(166)	5496.66+699.3	6196.11	62(107)	1849.82+361.12	2210.95	61(209)	823.08
6.2.799	51(51)	4.8+4.9	9.7	51(90)	2.5+4.2	6.7	35(70)	5.5
199	62(107)	118.7+35.6	154.3	63(284)	41.4+41.8	83.2	51(88)	45.8
299	68(188)	684.8+127.4	812.4	70(493)	217.14+164.1	381.29	54(94)	158.7
399	80(1010)	2299.4+654.9	2954.3	81(1014)	703.2+522.5	1225.8	65(109)	515.9
499	90(1170)	3808.8+1150.7	4959.6	87(1331)	1733.9+1083.6	2817.7	80(115)	989.4

Table 7.10: Numerical results: distributed control

IP-MA27

Prob.	iter	time	Prob.	iter.	time
6.1.1 99	29	27.38	1.2 99	24	23.1
199	37	349.66	199	26	258.7
6.1.2 99			1.3 99	26	22.1
199	35	339.1	199	31	269.1
6.1.3 99	24	22.52	1.4 99	27	22.9
199	27	250.28	199	33	285.1
6.1.4 99	25	22.7	1-1 119	31	48.1
199	30	269.7	179	34	406.7
6.1.5 99	24	21.7	1-0 119	35	54.6
199	26	370	179	40	581.5

Table 7.11: Boundary control problems with direct inner solver

IP-MA27

Prob.	iter	time
6.2.6 99	25	24.71
199	26	304.11

Table 7.12: Distributed control problems with direct inner solver

N	IP-PCG2	KNITRO-I	KNITRO-D
99	6	40	17
199	46	321	127
299	243	1353	759
399	799	4990	1939
499	1372	10343	
599	2512	17577	7447*
699	6575	30069	
799	7279		
899	10290		
999	20892		
1099	20168		
1199	27624		

KNITRO: opttol=1e-9 * on 2GHz Opteron

Table 7.13: Comparison IP-PCG2-Knitro v3.1 on the test problem 6.1.1 on a 3.2MHz Pentium 4

Table 7.14: Numerical results: nonmonotone algorithm with direct inner solver

	Monotone		Nonmonotone					
	$N = 0$		$N = 2$		$N = 4$		$N = 9$	
P	it	b.	it	b.	it	b.	it	b.
6.1.1-49	27	1	26	0	26	0	26	0
6.1.2-49	30	15	–	–	26	2	26	2
6.1.3	–	–	25	0	25	0	25	0
6.1.4	33	6	–	–	–	–	34	2
6.1.1-99	46	55	–	–	–	–	33	0
6.1.2-99	–	–	–	–	–	–	32	13
6.1.3-99	–	–	26	1	27	0	27	0
6.1.4-99	31	0	31	0	31	0	31	0
6.1.2-199	–	–	–	–	–	–	41	19
6.1.3-199	–	–	26	1	27	0	27	0
6.1.4-199	29	2	32	1	32	1	32	1

	$N = 2$		$N = 3$		$N = 4$	
Prob.	it	sec	it	sec	it	sec
6.2.7-199	62(92)	113.2(34.5)	62(77)	112.1(33.4)	62(77)	112.1(33.4)
6.2.7-299	68(143)	538.5(119.4)	68(113)	533.4(114.2)	68(113)	533.1(114)
6.2.7-399	80(995)	2101.1(1447.7)	80(905)	2056.5(614.1)	80(815)	2037(589.8)
6.2.7-499	90(1170)	4959(1150)	89(1139)	4984.3(1154.4)	89(1052)	4891.3(1082.4)
	$N = 5$		$N = 7$			
	it	sec	it	sec		
	62(62)	111.0(32.3)	62(62)	111.0(32.3)		
	68(98)	530.6(111.4)	68(68)	525.3(106.0)		
	80(636)	1962.5(520.1)	80(500)	1915.2(472.7)		
	89(974)	4899.0(1069.1	89(824)	4821.5(991.6)		
	$N = 2$		$N = 3$		$N = 4$	
Prob.	it	sec	it	sec	it	sec
6.2.5-99	48(63)	9.8(4.8)	48(48)	9.62(4.5)	48(48)	9.62(4.5)
6.2.5-199	68(278)	131.4(50.6)	68(173)	123.7(42.9)	68(113)	119(38.5)
6.2.5-299	103(1408)	822(392.1)	104(1334)	810(380.1)	103(1213)	787.1(357.2)
6.2.5-399	148(2158)	2804.1(1311.2)	150(2145)	2806.6(1313.7)	150(2145)	2806.6(1313.7)
	$N = 5$		$N = 7$			
	48(48)	9.62(4.5)	48(48)	9.62(4.5)		
	68(98)	118.2(37.4)	68(68)	116.5(35.4)		
	103(996)	748.5(318.6)	104(449)	647.8(222.6)		
	150(2100)	2790.9(1298)	150(2014)	2736.8(1244.8)		

Table 7.15: Results for IP-Hestenes with nonmonotone stopping and backtracking rules

Prob	it.	time	Prob	it.	time
6.2.7-199	62	111(32.3)	6.2.5-99	48	9.6(4.6)
6.2.7-299	68	525.3(106)	6.2.5-199	68	116.5(35.5)
6.2.7-399	80	1456(330)	6.2.5-299	105	591.3(164.9)
6.2.7-499	91	4428.6(612.7)	6.2.5-399	156	2104.3(634)

Table 7.16: IP-Hestenes with the number of inner iterations fixed to one.

	N=1		N=2		N=3		N=4	
Prob.	it(inn.)	time	it(inn.)	time	it(inn.)	time	it(inn.)	time
6.2.7-99	35(70)	5.5	39(70)	5.9	42(72)	6.3	25(52)	3.9
6.2.7-199	51(88)	45.8	54(94)	48.4	62(105)	55.3	28(37)	24.3
6.2.7-299	54(94)	158.7	57(95)	167.2	73(114)	212.5	32(51)	93.8
6.2.7-399	65(109)	515.9	79(144)	630.5	32(38)	248.6	35(37)	269.9

Table 7.17: IP-PCG2 with the nonmonotone inner stopping rule

Bibliography

[1] A. Altman and J. Gondzio (1999). *Regularized symmetric indefinite systems in Interior–Point methods for linear and quadratic optimization*, Optim. Meth. Software, **11**, 12, pp. 275–302.

[2] M. Argaez and R. A. Tapia (2002). *On the global convergence of a modified augmented lagrangian linesearch Interior–Point method for nonlinear programming* J. Optim. Theory Appl., **114**,1, pp. 1–25.

[3] M. Argaez, R. A. Tapia and L. Velasquez (2002). *Numerical comparisons of path–following strategies for a primal–dual interior–point method for nonlinear programming* J. Optim. Theory and Appl., **114**,2, pp. 255–272.

[4] S. Bellavia (1998). Inexact Interior–Point Method, J. Optim. Theory Appl., **96**, 1, pp. 109–121.

[5] S. Bellavia, M. Macconi and B. Morini (2004). *STRSCNE: A scaled trust region solver for constrained nonlinear equations*, Comput. Optim. Appl., **98**, pp. 31–50.

[6] H. Benson, D. F. Shanno and R.J. Vanderbei (2001). *Interior-Point methods for nonconvex nonlinear programming: filter methods and merit functions*, Technical Report of Operations Research and Financial Engineering, Princeton University, ORFE-00-06.

[7] L. Bergamaschi, J. Gondzio and G. Zilli (2003). *Preconditioning indefinite systems in Interior Point methods for optimization*, to appear on Comput. Optim. Appl.

[8] M. Bergounioux, K. Ito and K. Kunish (1997). *Primal–dual strategy for constrained optimal control problems*, SIAM J. Control Optim. **35**, pp. 1524–1543.

[9] M. Bergounioux and K. Kunish (1999). *Augmented Lagrangian techniques for elliptic state constrained optimal control problems*, SIAM J. Control Optim. **37**, pp. 1176–1194.

[10] J. T. Betts (2001). *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, Philadelphia.

[11] S. Bonettini, E. Galligani and V. Ruggiero (2004). *An inexact Newton method combined with Hestenes multipliers' scheme for the solution of Karush–Kuhn–Tucker Systems*, Appl. Math. Comput., to appear.

[12] S. Bonettini (2004). *A Nonmonotone Inexact Newton Method* Optim. Meth. Software, to appear.

[13] F. Bonnans and E. Casas (1989). *Optimal control of semilinear multistate systems with state constraints*, SIAM J. Control Optim., **27**, pp. 303–325.

[14] F. Bonnans and E. Casas (1995). *An extension of Pontryagin's principle for state–constrained optimal control of semilinear elliptic equations and variational inequalities*, SIAM J. Control Optim., **33**, pp. 274–298.

[15] F. H. Branin (1972). *Widely convergent method for finding multiple solution of simultaneous nonlinear equations*, IBM J. Res. Devel., **16**, pp. 504–525.

[16] J. R. Bunch and B. N. Parlett (1971). *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., **8**, pp. 639–655.

[17] R. H. Byrd, J.C. Gilbert, and J. Nocedal (2000). *A trust region method based on Interior Point techniques for nonlinear programming*, Math. Programming A, **89**, pp.149–185.

[18] R. H. Byrd, M. E. Hribar and J. Nocedal (1999). *An interior point algorithm for large–scale nonlinear programming*, SIAM J. Optimization, **9**, 4, pp. 877–900.

[19] R. H. Byrd, M. Marazzi and J. Nocedal (2002). *On the convergence of Newton iterations to non–stationary points*, Report OTC 2001/01, Optimization Technology Center, Northwestern University, Evanston,IL.

[20] A. Canãda, J.L. Gámez and J. A. Montero (1998). *Study of an optimal control problem for diffusive nonlinear elliptic equations of logistic type*, SIAM J. Control Optim., **36**, pp. 1171–1189.

[21] M.D. Canon, C. D. Cullum and E. Polak (1970). Theory of Optimal Control and Mathematical Programming, Mc Graw–Hill, New York.

[22] E. Casas (1993). *Boundary control with pointwise state constraints*, SIAM J. Control Optim., **31**, pp. 993–1006.

[23] E. Casas, F. Trölzsch and A. Unger (1996). *Second order sufficient optimality conditions for a nonlinear ellptic control problem*, J. Anal. Appl., **15**, pp. 687–707.

[24] E. Casas, F. Trölzsch and A. Unger (2000). *Second order sufficient optimality conditions for some state constrained control problems of semilinear elliptic equations*, SIAM J. Control Optim., **38**, pp. 1369–1391.

[25] R. S. Dembo, S. C. Eisenstat and T. Steihaug(1982). *Inexact Newton methods*, SIAM J. Numer. Anal., **19**, pp. 400–408.

[26] J. E. Dennis and R. B. Schnabel (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice–Hall, Inc., Englewood Cliffs, New Jersey.

[27] P. Deuflhard and A. Hohomann (1995). Numerical Analysis. A First Course in Scientific Computation, Walter de Gruyter, Berlin–New–York.

[28] C. Durazzi (2000). *On the Newton Interior–Point method for nonlinear programming problems*, J. Optim. Theory Appl., **104**, 1, pp. 73–90.

[29] C. Durazzi and V. Ruggiero (2003). *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems*, Numer. Linear Algebra Appl., **10**, pp. 673–688 .

[30] C. Durazzi and V. Ruggiero (2003). *A Newton inexact Interior–Point method for large scale nonlinear optimization problems*, Annali Univ. Ferrara, Sez. VII, Sc. Matem. IL, pp. 333–357.

[31] C. Durazzi and V. Ruggiero (2004). *Global convergence of the Newton Interior–Point method for nonlinear programming*, J. Optim. Theory Appl. **120**, pp. 199–208.

[32] C. Durazzi, V. Ruggiero and G. Zanghirati (2001). *Parallel interior–point method for linear and quadratic programs with special structure*, J. Optim. Theory Appl., **110**, pp. 289–313.

[33] S. C. Eisenstat and H. F. Walker (1994). *Globally convergent inexact Newton methods*, SIAM J. Optimization, **4**, pp. 393–422.

[34] M. El Hallabi and R. A. Tapia (1993). *A global convergence theory for arbitrary norm trust–region methods for nonlinear equations*, Tech Report TR93-41, Department of Mathematical Sciences, Rice University, Houston, TX, September 1993; revised May, 1995.

[35] A. S. El–Bakry, R. A. Tapia, T. Tsuchiya and Y. Zhang (1996). *On the formulation and theory of the Newton Interior–Point method for nonlinear programming*, J. Optim. Theory Appl., **89**, 3, pp. 507–541.

[36] D. K. Faddeev and V. N. Faddeeva (1963). Computational Methods for Linear Algebra, W. H. Freeman Co., San Francisco.

[37] A. V. Fiacco and G. P. McCormick (1968). Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, reprinted by SIAM Publications, 1990.

[38] R. Fletcher and S. Leyffer (2002). *Nonlinear programming without a penalty function*, Math. Programming, **91**, pp. 239–269.

[39] E. Galligani (2004) *Analysis of the convergence of an inexact Newton method for solving Karush–Kuhn–Tucker systems*, to appear on Atti Sem. Matem. Fis. Univ. Modena.

[40] I. Galligani and D. Trigiante (1974). *Numerical methods for solving large algebraic systems*, IAC Pubblication N. 98, ser. 3.

[41] J. Gondzio (1995).*HOPDM (version 2.12)–A fast LP solver based on a primal-dual Interior Point method*, European J. Oper. Res., **85**, pp. 221–225

[42] N.I.M. Gould (1985). *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*, Math. Programming, **32**, pp. 90–99.

[43] J. Nocedal, M.E. Hribar and N.I.M. Gould (2001). *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Computing, **23**, 4, pp. 1375-1394.

[44] L. Grippo, F. Lampariello and S. Lucidi (1986). *A nonmonotone line search technique for Newton's method*, SIAM J. Num. Anal., **23**, pp. 707–716.

[45] I. Griva, D. F. Shanno and R. J. Vanderbei (2004). *Convergence analysis of a primal-dual interior-point method for nonlinear programming*, Technical Report downloadable from http://www.princeton.edu/ rvdb/techreps_pdf.html.

[46] M.R. Hestenes (1975). Optimization Theory. The Finite Dimensional Case, J. Wiley & Sons, New York.

[47] C. T. Kelley (1995). Iterative Methods for Solving Linear and Nonlinear Equations, SIAM, Philadelphia

[48] F. Leibfritz and E. W. Sachs (1994). *Numerical Solution of Parabolic State Constrained Control Problems usaing SQP and Interior–Point–Methods*, Large Scale Optimization: Stare of the Art, W. W. Hager et al. eds., Kluwer Academic Publisher, pp. 245–258.

[49] A. Leung and S. Stojanovic (1993). *Optimal control for Volterra–Lotka equations*, J. Math. Anal. Appl., **173**, pp. 603–619.

[50] L. Lukšan, J. Vlček (1998). *Indefinitely preconditioned Inexact Newton method for large sparse equality constrained non–linear programming problems*, Numer. Linear Algebra Appl., **5**, pp. 219–247.

[51] L. Lukšan, C. Matonoha and J. Vlček (2004). *Interior–Point method for nonlinear nonconvex optimization*, to appear on Numer. Linear Algebra Appl.

[52] B. N. Lundberg, A. B. Poore and B. Yang (1990). *Smooth penalty functions and continuation methods for constrained optimization*, Lectures in Applied Mathematics, SIAM, Philadelphia, **26**, pp. 389–412.

[53] D.G. Luenberger (1969). Optimization by Vector Space Methods, John Wiley and Sons, New York.

[54] D.G. Luenberger (1984). Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Reading.

[55] H. D. Mittelmann and H. Maurer (1999). *Optimization techiques for solving elliptic control problems with control and state constraints: Part 1. Boundary control*, Comput. Optim. Appl., **16**, pp. 29–55.

[56] H. D. Mittelmann and H. Maurer (2001). *Optimization techiques for solving elliptic control problems with control and state constraints: Part 2. Distributed control*, Comput. Optim. Appl., **18**, pp. 141–160.

[57] H. D. Mittelmann and H. Maurer (2000). *Solving elliptic control problems with Interior Point and SQP Methods: control and state constraints*, J. Comput. Appl. Math., **120**, pp. 175–195.

[58] W. Murray (1971) *Analytical expressions for the eigenvalues and eigenvectors of the hessian matrices of barrier and penalty functions*, J. Optim. Theory Appl., **7**, pp.189–196.

[59] (1993). J.W. Liu, E.G. Ng and B.W. Peyton; *On finding supernodes for sparse matrix computations*, SIAM J. Matrix Anal. Appl., **14**, pp.242–252.

[60] J. Nocedal and S. J. Wright (1999) Numerical Optimization, Springer–Verlag, New–York.

[61] J. M. Ortega and W. C. Rheimboldt (1970). Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York.

[62] M. J. D. Powell (1970). *A hybrid method for nonlinear equations*, in P. Rabinowitz editor, Numerical Methods for Nonlinear Algebraic Equations, Gordon and Breach, London, pp. 87–114.

[63] W.C. Rheinboldt (1998). Methods for Solving Systems of Nonlinear Equations, Second Edition, SIAM, Philadelphia.

[64] Y. Saad (1996). Iterative Methods for Sparse Linear System, PSW Publ. Co., Boston MA.

[65] D. F. Shanno and R. J. Vanderbei (1999). *Interior-Point methods for nonconvex nonlinear programming: orderings and higher-order methods*, Technical Report of Statistics and Operations Research, Princeton University SOR-99-5.

[66] D. F. Shanno and E. M. Simantiraki (1997). *Interior–Point methods for linear and nonlinear programming*, The State of the Art in Numerical Analysis, I.S. Duff and G. A. Warson eds., Clarendon Press, Oxford.

[67] T. Steihaug (1983). *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., **20**, 3, pp.626–638.

[68] S. Stojanovic (1991). *Optimal damping control and nonlinear elliptic problem*, SIAM J. Control Optim., **29**, pp. 594–608.

[69] M. Ulbrich, S. Ulbrich and L. N. Vicente (2000). *A globally convergent primal–dual Interior–Point filter method for nonconvex nonlinear programming*, Tecnical Report TR00–12, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA, revised March 2003.

[70] R. J. Vanderbei and D. F. Shanno (1999). *An Interior–Point algorithm for nonconvex nonlinear programming*, Comput. Optim. Appl., **13**, pp. 231–252.

[71] A. Wächter and L. T. Biegler (2004). *On the implementation of an Interior-Point filter line-search algorithm for large-scale nonlinear programming*, Research Report RC 23149, IBM T. J. Watson Research Center, Yorktown, USA

[72] A. Wächter and L. T. Biegler (2000). *Failure of global convercence for a class of interior point methods for nonlinear programming*, Math. Programming Series A, **83**, 3, pp. 565–574.

[73] A. Wächter (2002). An Interior–Point Algorithm for Large–Scale Nonlinear Optimization with Applications in Process Engineering, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

[74] M. H. Wright (1991). *Interior methods for constrained optimization*, Acta Numerica, pp. 341–407.

[75] M. H. Wright (1995). *Why a pure primal newton barrier step may be infeasible*, SIAM J. Optimization, **5**, 1, pp. 1–12.

[76] M. H. Wright (1998). *Ill–conditioning and computational error in interior methods for nonlinear programming*, SIAM J. Optimization, **9**, 1, pp. 84–111.

[77] M. H. Wright (2004). *The interior methods revolution in optimization: history, recent developments and lasting consequences*, Bulletin of American Mathematical Society (New series), S-0273-0979(04)01040-7.

[78] S. J. Wright (2001). *effects of finite precision arithmetic on interior–point methods for nonlinear programming*, SIAM J. Optim., **12**, 1, pp. 36-78.