Solution of bounded nonlinear systems of equations using homotopies with inexact restoration

E. G. Birgin *   Nataša Krejić †   J. M. Martínez ‡

February 25, 2002.

Abstract

Nonlinear systems of equations often represent mathematical models of chemical production processes and other engineering problems. Homotopic techniques (in particular, the bounded homotopies introduced by Paloschi) are used for enhancing convergence to solutions, especially when a good initial estimate is not available. In this paper, the homotopy curve is considered as the feasible set of a mathematical programming problem, where the objective is to find the optimal value of the homotopic parameter. Inexact restoration techniques can then be used to generate approximations in a neighborhood of the homotopy, the size of which is theoretically justified. Numerical examples are given.

Key words: Nonlinear programming, nonlinear systems, homotopies, bounded homotopies, homotopy methods, inexact restoration.

C.R. categories: G.4, G.1.5, G.1.6.

1 Introduction

Many models in chemistry, engineering and applied physics require the solution of nonlinear systems of equations (NLS) with bounded variables. In particular, this problem has many applications in chemical engineering, so that it is not surprising that valuable contributions for the computational solution of NLS can be found in the chemical engineering literature. See [14, 15, 16, 22, 23] and references therein.

Assume that $\Omega = \{x \in I\!R^n \mid \ell \leq x \leq u\}$, the mapping $F : \Omega \to I\!R^n$ has continuous first partial derivatives, $\ell, u \in I\!R^n$ and $\ell < u$. Then, the mathematical problem consists in finding

*Department of Computer Science IME-USP, University of São Paulo, Rua do Matão 1010, Cidade Universitária, 05508-900, São Paulo SP, Brazil. This author was supported by PRONEX-Optimization 76.79.1008-00 and FAPESP (Grants 99/08029-9 and 01-04597-4). e-mail: egbirgin@ime.usp.br

†Institute of Mathematics, University of Novi Sad, Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia. This author was supported by a visiting professor grant of FAPESP (99/03102-0). e-mail: natasa@unsim.im.ns.ac.yu

‡Department of Applied Mathematics, IMECC-UNICAMP, University of Campinas, CP 6065, 13081-970 Campinas SP, Brazil. This author was supported by PRONEX-Optimization 76.79.1008-00, FAPESP (Grant 01-04597-4), CNPq and FAEP-UNICAMP. e-mail: martinez@ime.unicamp.br

$x \in \Omega$ such that

$$F(x) = 0. \tag{1}$$

When bounds are not present, a classical strategy for solving the problem (1) is given by the homotopic approach. See, among others, [2, 3, 4, 5, 17, 18, 24, 25]. This approach has been extended to bounded problems by Paloschi [14, 15, 16]. See, also, [22, 23]. Often, there are solutions of (1) not belonging to Ω but they do not have physical meaning. Hence imposing natural bounds is essential for obtaining practical solutions of the problem. In other cases, bounds are introduced artificially due to ill-conditioning or singularity of the Jacobian.

Homotopic methods are used when strictly local Newton-like methods for solving (1) fail because a sufficiently good initial guess of the solution is not available. Moreover, in some applications areas, homotopy methods are now the rule of choice. See [25]. The homotopic idea consists in defining

$$H : \Omega \times I\!R \to I\!R^n$$

such that $H(x, 1) \equiv F(x)$ and the system $H(x, 0) = 0$ is easy to solve. The solution of $H(x, t) = 0$ is used as initial guess for solving $H(x, t') = 0$, with $t' > t$. In this way, the solution of the original problem is progressively approximated. For many engineering problems, natural homotopies are suggested by the very essence of the physical situation.

In general, the set of points (x, t) that satisfy $H(x, t) = 0$ define a curve in $I\!R^{n+1}$. Homotopic methods are the procedures to "track" this curve in such a way that its "end point" $(t = 1)$ can be safely reached. Since the intermediate points of the curve (for which $t < 1$) are of no interest by themselves, it is not necessary to compute them very accurately. So, an interesting theoretical problem with practical relevance is to choose the accuracy to which intermediate points are to be computed. If an intermediate point (x, t) is computed with high accuracy, the tangent line to the curve that passes through this point can be efficiently used to predict the points corresponding to larger values of the parameter t. This prediction can be very poor if (x, t) is far from the true zero-curve of $H(x, t)$. On the other hand, accurate computing of all intermediate points can be unaffordable.

The discussion above led us to establish a relation between the homotopic framework for solving nonlinear equations and inexact restoration methods [9, 10, 11], a family of recently introduced methods for nonlinear programming (NLP). The idea is to look at the homotopic problem as the nonlinear optimization problem

$$\text{Minimize } (t-1)^2$$

$$\text{subject to } H(x, t) = 0, \;\; x \in \Omega. \tag{2}$$

Therefore, the homotopic curve is the feasible set of (2). The nonlinear programming problem (2) could be solved by any constrained optimization method, but inexact restoration algorithms seem to be closely connected to the classical predictor-corrector procedure used in the homotopic approach. Moreover, they give theoretically justified answers to the accuracy problem.

Let us survey here how inexact restoration (IR) algorithms proceed in a general nonlinear program given by

$$\text{Minimize } f(w) \;\text{ subject to } w \in \mathcal{C}. \tag{3}$$

In (3), $\mathcal{C} \subset I\!R^m$ is the feasible region, defined by a set of equalities and/or inequalities. Given the current iterate w, an iteration of IR consists of two phases. In the first phase a "more feasible" point y is computed and in the second phase a "more optimal point" z is calculated on the "tangent" approximation to $\mathcal{C}$ that passes through y. If, according to some merit function, z is better than w, then z is the new current iterate. Otherwise, a trust region centered at y is reduced and a new trial point z is computed at the intersection of the tangent set with the trust region. It can be proved that the iteration finishes with a trial point z which is better than w. The criterion by which the intermediate point y is more feasible than the current point w is rigorously defined in the IR framework. These features give an answer to the question of how accurate the intermediate point should be. Under suitable assumptions, the theory ensures that convergence to a solution takes place. Inexact Restoration algorithms are related to gradient projection, reduced gradient and gradient restoration techniques. See [1, 8, 12, 13, 19, 20]. The main differences will be discussed in the following sections.

This paper is organized as follows. In Section 2 we discuss the relation between nonlinear programming and continuation. In Section 3 we define the algorithm and discuss some of its properties. Section 4 is devoted to numerical examples and some conclusions are drawn in Section 5.

2 Nonlinear programming and continuation

The feasible set of the nonlinear programming problem (3) can always be represented as

$$\mathcal{C} = \{w \in I\!R^n \mid h(w) = 0,\ \ell \le w \le u\}, \tag{4}$$

where $h : I\!R^n \to I\!R^m$. (When an inequality constraint $g(w) \le 0$ appears in the original formulation, it can be replaced by $g(w) + z = 0,\ z \ge 0$.)

Feasible methods for solving (3-4) generate a sequence of approximations $w^k \in \mathcal{C}$ such that $f(w^{k+1})$ is sufficiently smaller than $f(w^k)$ for all k. Reduced gradient, gradient projection and gradient restoration techniques [1, 8, 12, 13, 19, 20] belong to this class of methods. Usually, each iteration of a feasible method consists of two phases. In the "predictor phase", given a feasible $y \in \mathcal{C}$, a "better" approximation z is computed in the tangent set to $\mathcal{C}$ that passes through y. In the "corrector phase", feasibility is restored. Starting with (the generally infeasible) z one tries to find a new feasible point w such that $f(w)$ is sufficiently smaller than $f(y)$. In reduced gradient (GRG) methods [1, 8] the restored point is obtained (if possible) by modifying only the value of m "basic" variables. In gradient projection methods, the restoration is along a nearly orthogonal direction to the feasible manifold. If $h(w)$ is nonlinear, completely feasible points cannot be obtained, and we can say that restoration is, in some sense, always inexact.

Two questions arise: (i) how feasible the restored points must be in order to preserve (theoretical and practical) convergence?, and (ii) is it necessary to take care of feasibility at all?

In many problems it is not worthwhile to care a lot with feasibility because fast convergence can be obtained by dealing with the primal-dual system of optimality conditions as a whole. Such problems are generally solved by sequential quadratic programming (SQP) techniques. SQP can be viewed as an adaptation of Newton's method to the solution of the optimality conditions of (3). A pure Newton strategy cannot be used because one wants to distinguish minimizers from other stationary points, so the Newtonian technique is modified using line searches or trust regions. See [7] and references therein.

In some other problems it is important to preserve feasibility of the iterates or, at least, to recover easily feasibility from current iterates. Frequently, feasible points are useful even without being optimal, whereas infeasible points are not. The feasibility of iterates in reduced gradient and gradient projection methods is controlled in the corrector phase of the iteration. This phase involves the application of an iterative algorithm (usually Newton's chord or quasi-Newton) and the feasibility tolerance is related to its convergence criterion. Usually, this tolerance is a "small positive number" related to machine precision and scaling quantities.

The inexact-restoration techniques for nonlinear programming introduced theoretical criteria that control feasibility and preserve convergence of the overall process. Moreover, the requirements on the restored point are quite general, so that one is not restricted to orthogonal-like or basic-like restoration steps. This is important because the nature of many problems calls for specific restoration procedures.

The identification of the homotopy path with the feasible set of a nonlinear programming problem allows us to use IR criteria to define the closeness of corrected steps to the path. The solutions of (1) correspond exactly to those of (2), so the identification proposed here is quite natural.

The correspondence between the feasibility phase of the feasible algorithms for (3) and the corrector phase of predictor-corrector continuation methods is immediate. The IR technique provides a criterion for declaring convergence of the subalgorithm used in the correction. The optimality phase of the feasible algorithms for (3) corresponds to the predictor phase of continuation methods. The IR technique determines how long predictor steps can be taken and establishes a criterion for deciding whether they should be accepted or not.

Turning points of the homotopy path correspond to local minimizers of (2). Inexact restoration helps to find them but says nothing about how to continue the path through them. From the point of view of (local) nonlinear programming, at a turning point the problem is solved. Other singular points of the homotopy paths generally correspond to the points where some constraint qualification is not fulfilled. Again, IR says nothing about how to continue the homotopy path in this case.

Inexact restoration (as classical feasible methods, SQP and practical versions of penalty and augmented-Lagrangian methods) is a *local* technique for nonlinear programming in the sense that convergence is guaranteed to critical points. The set of critical points is an approximation of the set of local minimizers that generally includes saddle points and nonregular points of the feasible set. When in nonlinear programming a local minimizer is not satisfactory, some *global* tool must be used to jump out from its convergence basin. Similar procedures are used when one wants to find more than a single minimizer. Many heuristic nonlinear programming techniques have been proposed for that purpose. However, we feel that for this specific task the amount of continuation ideas that can be used in nonlinear programming is larger than the number of nonlinear programming global heuristics that could be useful in numerical continuation.

Inexact-restoration tools can be used in connection to any curve-tracking method for tracing homotopies. Imbreeding between nonlinear programming and continuation techniques seems to be natural and necessary, by means of the identification between (1) and (2).

3 Inexact-restoration algorithm

Let us first introduce the notation that will be used throughout the rest of this paper. The symbol $\|\cdot\|$ will denote the Euclidean norm and $\mathcal{B}(y,\delta)$ the ball with the center y and radius δ. For $x \in I\!R^n$ and $t \in I\!R$ we denote $\tilde{x} = (x,t) = (x, x_{n+1}) \in I\!R^{n+1}$, and $\tilde{f}(\tilde{x}) = f(x_{n+1}) = (x_{n+1}-1)^2$. So, the problem we are going to consider can be written as

$$\text{Minimize } f(x_{n+1})$$

$$\text{subject to } H(\tilde{x}) = 0, \; x \in \Omega, \tag{5}$$

where $\Omega = \{x \in I\!R^n : \ell \le x \le u\}$, $\ell, u \in I\!R^n$ and H is the homotopy mapping. The solution of this problem is $\tilde{x}^* = (x^*, 1)$, and $F(x^*) = 0$. The Inexact-Restoration algorithm for obtaining x^* is iterative. The generated sequence will be called $\{\tilde{x}^k\}$. The parameters $\eta > 0, \theta_{-1} \in (0,1), \delta_{min} > 0, \delta_{max} > \delta_{min}, r \in (0,1), \beta > 0$ and the sequence of positive numbers $\{w_k\}$ such that $\sum w_k < \infty$, are given as well as the initial approximation x^0. We also define $t_0 = 0$. In the implementation, we use $\eta = 1$, $\theta_{-1} = 0.8$, $\delta_{min} = 10^{-3}$, $\delta_{max} = 1$, $w_k \equiv 1.1^{-k}$, $r = 0.1$ and $\beta = 10^6$.

Assume that $x^k \in \Omega, t_k, \theta_{k-1}, \ldots, \theta_{-1}$ have been computed. The steps for obtaining $\tilde{x}^{k+1}$ are given below.

Algorithm 1.
Step 1. *Initialize the trust region and the penalty parameter.*
Define

$$\theta = \min\{1, \min\{\theta_{k-1}, \ldots, \theta_{-1}\} + w_k\}.$$

and

$$\delta \leftarrow 1.$$

Step 2. *Feasibility phase of the iteration.*
Compute the point $\tilde{y} \in \Omega \times I\!R$ such that

$$\|H(\tilde{y})\| \le r\|H(\tilde{x}^k)\| \quad \text{and} \quad \|\tilde{y} - \tilde{x}^k\| \le \beta \|H(\tilde{x}^k)\|. \tag{6}$$

(Observe that $\tilde{y} = \tilde{x}^k$ is the only point that satisfies (6) if $\|H(\tilde{x}^k)\| = 0$.)
Step 3. *Tangent Cauchy direction*
Let $\mathcal{N}$ be the null-space of $H'(\tilde{y})$. Define

$$\pi = (\tilde{y} + \mathcal{N}) \cap (\Omega \times I\!R).$$

Compute

$$d^k_{tan} = p - \tilde{y}$$

where p is the projection of $\tilde{y} - \eta \nabla \tilde{f}(\tilde{y})$ on π. In general, π is a line segment, so this projection can be trivially computed.

If $\tilde{y} = \tilde{x}^k$ and $\|d_{tan}^k\| = 0$, terminate the execution of the algorithm.

If $\|d_{tan}^k\| = 0$ but $\tilde{y} \neq \tilde{x}^k$, define

$$\tilde{x}^{k+1} = \tilde{y},\ \theta_k = \theta_{k-1},$$

and terminate the iteration.

Step 4. *Trial point in the tangent set.*

Compute $\tilde{z} \in \pi$ the minimizer of $\tilde{f}(\tilde{z})$ on $\pi \cap \mathcal{B}(\tilde{y}, \delta)$. Observe that, since π is in general a segment and $\tilde{f}$ is quadratic, this step is trivial.

Step 5. *Predicted reduction*

Define, for all $\mu \in [0, 1]$,

$$\text{Pred }(\mu) = \mu[\tilde{f}(\tilde{x}^k) - \tilde{f}(\tilde{z})] + (1-\mu)[\|H(\tilde{x}^k)\| - \|H(\tilde{y})\|].$$

Compute

$$\theta \leftarrow \max\{\mu \in [0, \theta] \mid \text{ Pred }(\mu) \geq \frac{1}{2}[\|H(\tilde{x}^k)\| - \|H(\tilde{y})\|]\}$$

and define

$$Pred \leftarrow \text{ Pred }(\theta).$$

Step 6. *Compare actual and predicted reduction.*

Compute

$$Ared = \theta[\tilde{f}(\tilde{x}^k) - \tilde{f}(\tilde{z})] + (1-\theta)[\|H(\tilde{x}^k)\| - \|H(\tilde{z})\|].$$

If

$$Ared < \frac{1}{10} Pred$$

update $\delta \leftarrow \|\tilde{z} - \tilde{y}\|/2$ and go to Step 4.

Else, define

$$\tilde{x}^{k+1} = \tilde{z},\ \theta_k = \theta.$$

If $Ared \geq 2\ Pred$, update

$$\delta \leftarrow \min\{\delta_{max}, \max\{\delta_{min}, 2\delta\}\}.$$

Terminate the iteration.

As we mentioned in the introduction, Algorithm 1 is obtained as a special case of the inexact restoration algorithm given in [10], taking into account the specific problem we deal with here. See, also, [9, 11]. Some steps deserve further comments. Since $H'(\tilde{y}) \in I\!R^{n\times(n+1)}$, the null-space of this matrix is, very likely, a line. For some homotopies it can be assumed that it is a line with probability 1. See [24, 25]. We are going to assume that this is the case in the rest of this paper. Therefore, the set π is a line segment and the projection p can be trivially computed. By the same reason, it is also trivial to compute the trial point $\tilde{z}$ in Step 4.

The merit function used in [10] is the addition of a Lagrangian plus a nonsmooth penalty term. In [9], a classical augmented Lagrangian is used. Therefore, in each iteration an estimation of the Lagrange multipliers at the solution is needed. In our case, the gradient of the objective function at the desired solution is null so that the optimal Lagrange multipliers are zero. Therefore, the Lagrangian term is not necessary in the merit function from which *Ared* and *Pred* are derived.

The algorithm for the feasibility phase of the inexact restoration algorithm (Step 2 of Algorithm 1) is given in Appendix 1 (Algorithm 2).

The theoretical (convergence) results related to the algorithm defined in this section follow as particular cases of the results proved in [10]. Assuming that H is Lipschitz-continuous, it can be proved that the algorithm is well defined, that is, each iteration necessarily finishes after a finite number of reduction of the trust region radius. In the remaining results it is assumed that Phase 1 of the algorithm can always be completed. In this case one can prove that every limit point of the generated sequence belongs to the homotopy curve and an optimality condition of the nonlinear program (2) is satisfied by a certain limit point. Under regularity conditions, this implies that this limit point is a solution of (2).

4 Numerical examples

We present four examples. The first is a bound-constrained problem taken from [14]. The other three examples are unconstrained problems. These problems were introduced in [6] as difficult problems in the sense that the purely local Newton method, given by

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

does not converge for different reasons regardless of the number of iterations allowed.

Example 1: Process of production of dibutyl phtalate. ([14])

This problem models a process of production of dibutyl phtalate. The process involves two adiabatic continuous stirred-tank reactors (CSTR) and the model consists of seven equations. See [21]. We used the bounds $\ell = (50, 0, 0, 0, 0, 0, 0)$, $u = (100, 10, 100, 10, 1, 10, 1)$, and the initial point $x^0 = (70, 6, 50, 8, 0.1, 2, 0.9)$.

The Newton homotopy fails for this problem. The path enters a region with negative values for some of the variables and becomes unbounded afterwards (see [14]). The use of the Newton bounded homotopy combined with the affine-scaling-invariant regularization form of Paloschi's penalty solves this problem. Figure 1 shows the restarted $(\tilde{y}^k)$ points which, according to (6), are approximate solutions of $H(y,t) = 0$. The solution is $x^* \approx (80.52, 61.11, 6.97, 7.20, 0.55, 3.65, 0.06)$. It should be noted that the value of $H(x^*, 1) \approx 7.35 \times 10^{-9}$ while $F(x^*) \approx 0.12$. This can be easily explained looking at Paloschi's bounded homotopies formulae [14].

Example 2: Augmented Powell badly scaled ([6])

$$\begin{aligned} F_{3i-2}(x) &= 10^4 x_{3i-2}x_{3i-1} - 1, \\ F_{3i-1}(x) &= \exp(-x_{3i-2}) + \exp(-x_{3i-1}) - 1.0001, \\ F_{3i}(x) &= \varphi(x_{3i}), \end{aligned}$$

for $i = 1, \ldots, n/3$ (n multiple of 3), where

$$\varphi(t) = \begin{cases} \dfrac{t}{2} - 2, & \text{if} \quad t \le -1 \\[2ex] \dfrac{1}{1998}(-1924 + 4551t + 888t^2 - 592t^3), & \text{if} \quad t \in [-1, 2] \\[2ex] \dfrac{t}{2} + 2, & \text{if} \quad t \ge 2 \end{cases},$$

with $x_0 = (0, 1, -4, 0, 1, -4, \ldots)$.

Example 3: Tridimensional valley ([6])

$$\begin{array}{lcl} F_{3i-2}(x) & = & (c_2 x_{3i-2}^3 + c_1 x_{3i-2}) \exp(\frac{-x_{3i-2}^2}{100}) - 1, \\ F_{3i-1}(x) & = & 10(\sin(x_{3i-2}) - x_{3i-1}), \\ F_{3i}(x) & = & 10(\cos(x_{3i-2}) - x_{3i}), \end{array}$$

for $i = 1, \ldots, n/3$ (n multiple of 3), where

$$\begin{array}{lcl} c_1 & = & 1.003344481605351, \\ c_2 & = & -3.344481605351171 \times 10^{-3}, \end{array}$$

with $x_0 = (-4, 1, 2, 1, 2, 1, 2, \ldots)$.

Example 4: Diagonal of three variables premultiplied by a quasi-orthogonal matrix ([6])

$$\begin{array}{lcl} F_{3i-2}(x) & = & 0.6x_{3i-2} + 1.6x_{3i-1}^3 - 7.2x_{3i-1}^2 + 9.6x_{3i-1} - 4.8, \\ F_{3i-1}(x) & = & 0.48x_{3i-2} - 0.72x_{3i-1}^3 + 3.24x_{3i-1}^2 - 4.32x_{3i-1} - x_{3i} + 0.2x_{3i}^3 + 2.16, \\ F_{3i}(x) & = & 1.25x_{3i} - 0.25x_{3i}^3, \end{array}$$

for $i = 1, \ldots, n/3$ (n multiple of 3), with $x_0 = (50, 0.5, -1, 50, 0.5, -1, \ldots)$.

Problems 2–4 (with $n = 51, 33, 33$, respectively) were solved using Newton's, regularizing and affine-scale-invariant homotopies. For Problem 2, all homotopies found a solution of (1). In Problem 3, the Newton and affine-scale-invariant homotopies found solutions of (1) while the regularizing homotopy stopped at a local minimizer of (2). Finally, for Problem 4, all homotopies stopped at local minimizers of (2) that are not solutions of (1). Figures 2–4 show the trajectories (t, x_1) of Newton's homotopy for this set of problems.

5 Conclusions

We considered the resolution of nonlinear systems using homotopies as a nonlinear programming problem and exploited a recently introduced NLP technique, inexact restoration, to produce approximate feasible points that are close to the homotopy path. Inexact-restoration ultimately guarantees the points are on the curve and finding of solutions at least in the NLP sense. The proposed preliminary algorithm appeared to be successful in solving some reported problems. Many efficient techniques for tracing homotopies through singular points have been introduced in the literature. These techniques are not the subject of the present research. In the algorithm we used several classical homotopies, including the bounded homotopies of Paloschi, which are quite useful for generating continuous paths to the solutions when natural bounds are present or when artificial bounds are induced by the behavior of unbounded homotopies.

In this research we considered the inexact restoration procedure as an independent technique for tracing approximate homotopies. Considering the formidable advances of homotopy methods in the last 15 years, there is probably a more promising approach that deserves future research. The inexact restoration criteria implicit in Steps 2, 5 and 6 of Algorithm 1 can be used in connection with other predictor-corrector algorithms for tracing homotopic curves, providing auxiliary tools for deciding on the acceptance or rejection of trial points.

6 Appendix 1

The algorithm below describes the first phase (restoration) of the IR algorithm.

Algorithm 2.

Initialize $\tilde{y} \leftarrow \tilde{x}^k$, $\nu = 0$.

Step 1. *Terminate if the desired conditions have been satisfied.*

If $\tilde{y}$ satisfies (6), return.

Step 2. *Stop if the number of iterations is exhausted.*

If $\nu > 10$, stop declaring "Failure of Phase 1".

Step 3. *Compute the projection on the linearized homotopy-curve*

Define

$$\mathcal{L} = \{w \in I\!R^{n+1} \mid H'(\tilde{y})(w - \tilde{y}) + H(\tilde{y}) = 0\}.$$

If $\mathcal{L} \cap (\Omega \times I\!R) = \emptyset$ stop declaring "Failure of Phase 1". Else, compute $\tilde{y}_{new}$, the projection of $\tilde{y}$ on $\mathcal{L} \cap (\Omega \times I\!R)$, set $\tilde{y} \leftarrow \tilde{y}_{new}$, $\nu \leftarrow \nu + 1$ and go to Step 1.

As we can see above, Algorithm 2 can fail in its task of finding $\tilde{y}$ satisfying (6). A theorem proved in [10] indicates that such a failure is unlikely to occur.

The implementation of Step 3 of Algorithm 2 requires some linear algebra, which is sketched below. Again, the fact that the null space of $H'(y)$ is one-dimensional plays an essential role in the procedure. Let us write, for simplicity, $A = H'(\tilde{y}) \in I\!R^{n\times(n+1)}$, rank $(A) = n$. We compute the rectangular LU factorization of A. So,

$$AP = LU,$$

where P is a permutation matrix, $L \in I\!R^{n\times(n+1)}$ is the lower-triangular with nonnull diagonal entries, $U \in I\!R^{(n+1)\times(n+1)}$ is the upper-triangular with unitary diagonal. This factorization allows us to preserve possible sparsity of A. To find a particular solution of $Av = b$, observe that this system is equivalent to

$$LUs = b, \quad v = Ps. \tag{7}$$

For solving $LUs = b$ we fix $(Us)_{n+1} = 1$ and proceed by back-substitution. The problem of finding a basis of the null-space of A corresponds to taking $b = 0$ in (7). In this case, the situation is even simpler because $b = 0$ implies that $Us = e_{n+1}$ (the last element of the canonical basis of $\mathbb{R}^{n+1}$).

Having a particular point of $\mathcal{L}$ and a basis of the (one-dimensional) null-space of $H'(\tilde{y})$, the problem of finding the projection of $\tilde{y}$ on $\mathcal{L} \cap (\Omega \times \mathbb{R})$ can be solved as follows. First, we compute the projection of $\tilde{y}$ on $\mathcal{L}$. With the parametric representation of $\mathcal{L}$ this merely requires the minimization of a one-dimensional quadratic. If this projection belongs to $\Omega \times \mathbb{R}$, we are done. Otherwise, we consider the finite set

$$\mathcal{F} = \{x \in \mathcal{L} \mid x_i = \ell_i \quad \text{or } x_i = u_i\}.$$

If $\mathcal{F} \cap (\Omega \times \mathbb{R}) = \emptyset$, then $\mathcal{L} \cap (\Omega \times \mathbb{R}) = \emptyset$ and nothing can be done. Otherwise, the projection of $\tilde{y}$ on $\mathcal{L} \cap (\Omega \times \mathbb{R})$ is the point of $\mathcal{F} \cap (\Omega \times \mathbb{R})$ which is closest to $\tilde{y}$.

References

[1] Abadie, J., Carpentier, J., Generalization of the Wolfe Reduced-Gradient Method to the Case of Nonlinear Constraints, in *Optimization*, edited by R. Fletcher, Academic Press, New York, 37-47, 1968.

[2] Allgower, E., Georg, K., Simplicial and continuation methods for approximating fixed points and solutions to systems of equations, *SIAM Review* 22 (1980), 28–85.

[3] Allgower, E., Georg, K., Estimates for piecewise linear approximations of implicitly defined manifolds, *Applied Mathematics Letters* 5 (1989), 1–7.

[4] Allgower, E., Georg, K., *Numerical continuation methods: an introduction*. Series in Computational Mathematics 13, Springer, Berlin, 1990.

[5] Allgower, E., Georg, K., Numerical path following, in *Handbook of Numerical Analysis* Vol. 5, edited by P. G. Ciarlet and J. L. Lions, North Holland, Amsterdam, 1997, 3–207.

[6] Friedlander, A., Gomes-Ruggiero, M. A., Kozakevich, D. N., Martínez, J. M., and Santos, S. A., Solving nonlinear systems of equations by means of quasi-Newton methods with a nonmonotone strategy. *Optimization Methods and Software* 8 (1997), 25–51.

[7] Gomes, F. A. M., Maciel, M. C., Martínez, J. M., Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. *Mathematical Programming* 84 (1999), 161-200.

[8] Lasdon, L. S., Reduced Gradient Methods, in *Nonlinear Optimization 1981*, edited by M. J. D. Powell, Academic Press, New York, 235-242, 1982.

[9] Martínez, J. M., Two-phase model algorithm with global convergence for nonlinear programming. *Journal of Optimization Theory and Applications* 96 (1998), 397–436.

[10] Martínez, J. M., Inexact-restoration methods with Lagrangian tangent decrease and a new merit function function for nonlinear programming, *Journal of Optimization Theory and Applications* 111 (2001), 39–58.

[11] Martínez, J. M., Pilotta, E. A., Inexact - restoration algorithm for constrained optimization, *Journal of Optimization Theory and Applications* 104 (2000), 135–163.

[12] Miele, A., Huang, H. Y., Heideman, J. C., Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions, Ordinary and Conjugate Gradient Version, *Journal of Optimization Theory and Applications* 4 (1969), 213-246.

[13] Miele A., Levy, A. V., Cragg, E. E., Modifications and Extensions of the Conjugate-Gradient Restoration Algorithm for Mathematical Programming Problems, *Journal of Optimization Theory and Applications* 7 (1971), 450-472.

[14] Paloschi, J. R., Bounded homotopies to solve systems of algebraic nonlinear equations, *Computers and Chemical Engineering* 19 (1995), 1243–1254.

[15] Paloschi, J. R., Bounded homotopies to solve systems of sparse of algebraic nonlinear equations, *Computers and Chemical Engineering* 21 (1997), 531–541.

[16] Paloschi, J. R., Using sparse bounded homotopies in the SPEEDUP simulation package, *Computers and Chemical Engineering* 9 (1998), 1181-1187.

[17] Rheinboldt, W. C., Numerical continuation methods: a perspective, *Journal of Computational and Applied Mathematics* 124 (2000), 229-244.

[18] Rheinboldt, W. C., Burkardt, J. V., A locally parametrized continuation process, *ACM Transactions on Mathematical Software* 9 (1983), 215–235.

[19] Rosen, J. B., The Gradient Projection Method for Nonlinear Programming, Part 1, Linear Constraints, *SIAM Journal on Applied Mathematics* 8 (1960), 181-217.

[20] Rosen, J. B., The Gradient Projection Method for Nonlinear Programming, Part 2, Nonlinear Constraints, *SIAM Journal on Applied Mathematics* 9 (1961), 514-532.

[21] Seader, J. D., Kuno, M., Lin, W., Johnson, S., Unsworth, K., and Wiskin, J., Mapped continuation methods for computing all solutions to general systems of nonlinear equations, *Computers and Chemical Engineering* 14 (1990), 71–85.

[22] Tanskanen, J., Pohjola, V. J., A robust method for predicting state profiles in a reactive distillation, *Computers and Chemical Engineering* 24 (2000), 81–88.

[23] Tanskanen, J., Pohjola, V. J., Minimum internal recycle in nonideal homogeneous multicomponent distillation, *Chemical Engineering Science* 55 (2000), 2713-2726.

x1

x3

[24] Watson, L. T., Billups, S. C., Morgan, A. P., Algorithm 652 HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Transactions on Mathematical Software* 13 (1987), 281–310.

[25] Watson, L. T., Sosonkina, M., Melville, R. C., Morgan, A. P., Walker, H. F., Algorithm 777. HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms, *ACM Transactions on Mathematical Software* 23 (1997), 514-549.

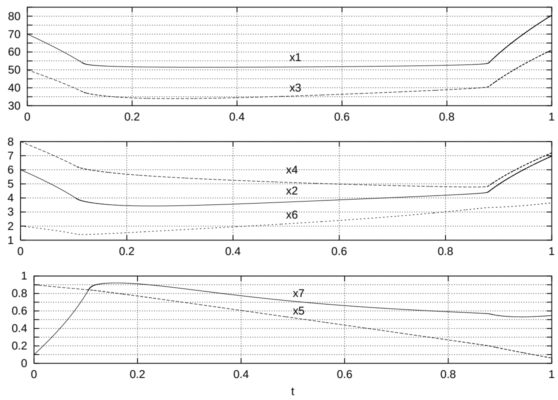

Figure 1: Approximate solutions of $H(y,t) = 0$ for Problem 1.

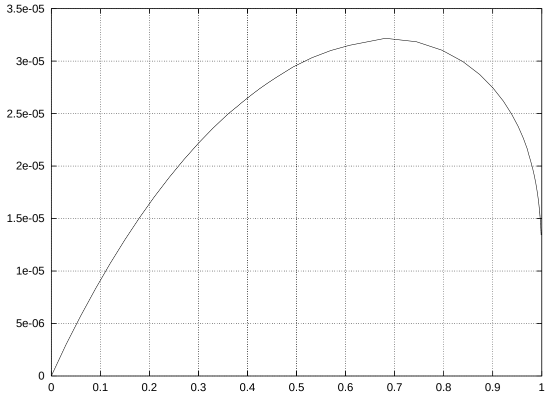

Figure 2: Approximate solutions of $H(y,t) = 0$ for Problem 2 (just first variable).

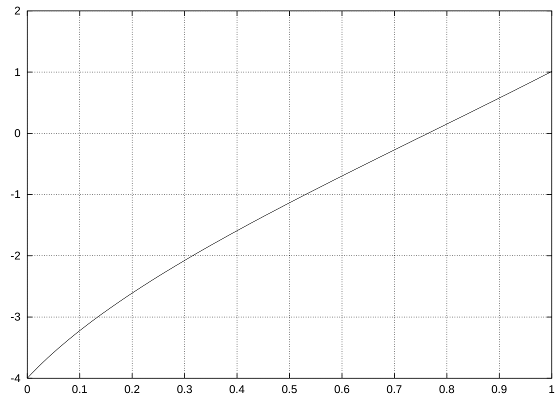

Figure 3: Approximate solutions of $H(y, t) = 0$ for Problem 3 (just first variable).

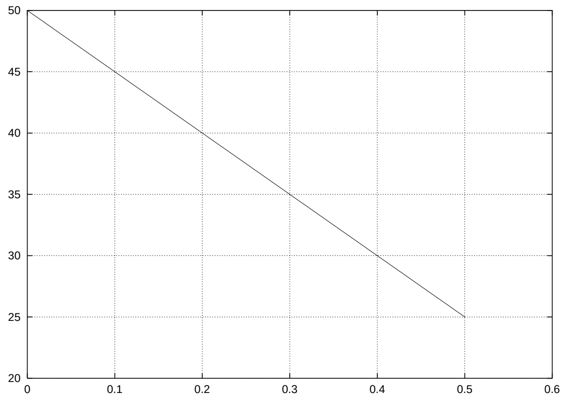

Figure 4: Approximate solutions of $H(y, t) = 0$ for Problem 4 (just first variable).Note that the method stops at a local minimizer of the NLP with $t \approx 0.5$.

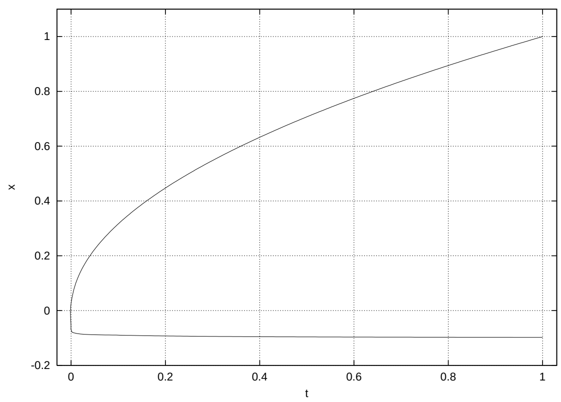

Figure 5: Newton "homotopy" for $F(x) = x^2 - 1$, $\Omega = [0, 2]$ and $\delta = 0.1$. Note that, for $t = 1$, $x = 1 \in \Omega$ is a solution of $F(x) = 0$ but $x = -0.1 \in \Omega' = [-0.1, 2.1]$ is not.