Inexact Newton methods and mixed nonlinear complementary problems

L. Bergamaschi and G. Zilli

Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate Università di Padova, Via Belzoni 7, 35131 Padova, ITALY

Abstract. In this paper we present the results obtained in the solution of sparse and large systems of nonlinear equations by Inexact Newton-like methods [6]. The linearized systems are solved with two preconditioners particularly suited for parallel computation. We report the results for the solution of some nonlinear problems on the CRAY T3E under the MPI environment. Our methods may be used to solve more general problems. Due to the presence of a logarithmic penalty, the interior point solution [10] of a nonlinear mixed complementary problem [7] can indeed be viewed as a variant of an Inexact Newton method applied to a particular system of nonlinear equations. We have applied this inexact interior point algorithm for the solution of some nonlinear complementary problems. We provide numerical results in both sequential and parallel implementations.

1 The Inexact Newton-Cimmino method

Consider the system on nonlinear equations

$$G(x) = 0 \qquad G = (g_1, ..., g_n)^T \tag{1}$$

where $G : R^n \to R^n$ is a nonlinear C^1 function, and its Jacobian matrix $J(x)$. For solving (1) we use an iterative procedure which combines a Newton and a Quasi-Newton method with a row-projection (or row-action) linear solver of Cimmino type [11], particularly suited for parallel computation. Here below, referring to block Cimmino method, we give the general lines of this procedure.

Let $As = b$ be the linearized system to be solved. Let us partition A into p row-blocks: $A_i, i = 1, \ldots, p$, i.e. $A^T = [A_1, A_2, \ldots, A_p]$ and partition the vector b conformally. Then the original system is premultiplied (*preconditioning*) by

$$H_p = [A_1^+, \ldots, A_i^+, \ldots, A_p^+] \tag{2}$$

where $A_i^+ = A_i^T (A_i A_i^T)^{-1}$ is the Moore-Penrose pseudo inverse of A_i.

We obtain the equivalent system $H_p A s = H_p b$,

$$(P_1 + \cdots + P_p)s = \sum_{i=1}^{p} A_i^+ A_i s = \sum_{i=1}^{p} A_i^+ b_i = H_p b, \tag{3}$$

where for each $i = 1, \ldots, p$, $P_i = A_i^+ A_i$ is the orthogonal projection onto range(A_i^T). As A is non singular, the matrix $H_p A \sum_{i=1}^{p} A_i^+ A_i$ is symmetric and positive definite. Then the solution of (3) is approximated by the Conjugate Gradient(CG) method. The q (underdetermined) linear least squares subproblems in the pseudoresidual unknowns $\delta_{i,k}$

$$A_i \delta_{i,k} = (b_i - A_i s_k), \quad 1 \leq i \leq p \tag{4}$$

must be solved at *each* conjugate gradient iteration ($k = 1, 2...$).

Combining the classic Newton method and the block Cimmino method we obtain the block Inexact Newton-Cimmino algorithm [11], in which at a major outer iteration the linear system $J(x_k)s = -G(x_k)$, where $J(x^*)$ is the Jacobian matrix, is solved in *parallel* by the block Cimmino method.

In [12, 4] a simple p-block partitioning of the Jacobian matrix A was used for solving *in parallel* a set of nonlinear test problems with sizes ranging from 1024 to 131072 on a CRAY T3E under the MPI environment. The least squares subproblems (4) were solved *concurrently* with the iterative Lanczos algorithm LSQR.

In this paper (see in section 4) we adopt a suitable block row partitioning of the matrix A in such a way that $A_i A_i^T = I, \quad i = 1, \ldots p$, and consequently, $A_i^+ = A_i^T$. This simplify the solution of the subproblems (4).

Due to the costly communication routines needed in this approach we have also implemented in parallel the preconditioned BiCGstab for the solution of the linearized system. As the preconditioner we choose AINV [2] which is based on the approximate sparse computation of the inverse of the coefficient matrix.

2 Inexact Newton method for nonlinear complementary problems

The methods of section 1 may be used to solve more general problems as the nonlinear mixed complementary problems [7] (including linear and nonlinear programming problems, variational inequalities, control problems, etc.).

Let us consider the following system of constrained equations:

$$F(v, s, z) = \begin{pmatrix} G(v, s, z) \\ SZe \end{pmatrix} = 0 \qquad (s, z) \geq 0 \tag{5}$$

where $G : \mathbb{R}^{n+2m} \to \mathbb{R}^n$ is a nonlinear function of v, $S = \text{diag}(s_1, \ldots, s_m)$, $Z = \text{diag}(z_1, \ldots, z_m)$, $e = (1, \ldots 1)^T$. The interior point methods [10] for the solution of (5) require the solution of the nonlinear system $F(x) = 0$. Using the Newton method we have to solve at every iterations a linear system of the form

$$F'(x_k)\Delta x = -F(x_k) + \sigma_k \mu_k e_0 \tag{6}$$

where $\mu_k = (s_k^T z_k)/m$, $\sigma_k \in]0, 1[$, that is an Inexact Newton method [6].

An interior point method in which the linearized system is solved approximately (by means of an iterative method) will be called *inexact (truncated) interior point method.* In this framework system (6) becomes

$$F'(x_k)\Delta x = -F(x_k) + \sigma_k \mu_k e_0 + r_k \tag{7}$$

where r_k is the residual of the iterative method applied to the linear system satisfying $\|r_k\| \leq \eta_k \mu_k$, and η_k is, for every k, the *forcing term* of the inexact Newton method [6]. Global convergence is assured by means of backtracking [1].

3 Numerical results I (sequential)

We have applied the *inexact interior point methods* for the solution of two nonlinear complementary problems: the (sparse) obstacle Bratu problem [5, 8], and the (dense) Lubrication problem [9, 5].

3.1 The obstacle Bratu problem

This problem can be formulated as a nonlinear system of equations:

$$f(v) = z_1 - z_2, \quad Z_1 S_1 e = 0, \quad Z_2 S_2 e = 0, \quad s_1 = v - v_l, \quad s_2 = v_u - v, \tag{8}$$

with the constraint $s_i, z_i \geq 0,\ i = 1, 2$. The nonlinear function $f(v)$ is defined as

$$f(v) = Av - \lambda h^2 E(v)e, \qquad E(v) = \text{diag}(\exp(v_1), \ldots, \exp(v_n)),$$

where A is the matrix arising from FD discretization of the Laplacian on the unitary square with homogeneous Dirichlet boundary conditions, v_l, v_u are the obstacles and h is the grid spacing.

The system (7) at step k can be written as

$$\begin{bmatrix} B & 0 & 0 & -I & I \\ 0 & Z_1 & 0 & S_1 & 0 \\ 0 & 0 & Z_2 & 0 & S_2 \\ -I & I & 0 & 0 & 0 \\ I & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta s_1 \\ \Delta s_2 \\ \Delta z_1 \\ \Delta z_2 \end{bmatrix} = \begin{bmatrix} -f + z_1 - z_2 \\ -Z_1 S_1 e + \sigma_k \mu_k e \\ -Z_2 S_2 e + \sigma_k \mu_k e \\ -s_1 + v - v_l \\ -s_2 + v_u - v \end{bmatrix} \equiv \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

where $B = f'(v)$. Taking into account the simple structure of some of the block matrices, we can use a Schur complement approach to reduce the original system $(5n \times 5n)$ to a system with n rows and n columns. In this way we obtain a system in the Δv unknown only: $C\Delta v = r$, where

$$C = B + S_1^{-1} Z_1 + S_2^{-1} Z_2, \quad r = b_1 + S_1^{-1}(b_2 - Z_1 b_4) - S_2^{-1}(b_3 - Z_2 b_5).$$

Once this nonsymmetric system has been solved (we used the BiCGstab solver), we can compute $\Delta z_1, \Delta z_2$ and $\Delta s_1, \Delta s_2$ by:

$$\Delta z_1 = S_1^{-1} Z_1 (b_2 - b_4 - \Delta v), \qquad \Delta s_1 = b_4 + \Delta v$$

$$\Delta z_2 = S_2^{-1} Z_2 (b_3 - b_5 + \Delta v), \qquad \Delta s_2 = b_5 - \Delta v.$$

We may note that matrix C is is obtained by adding to B the two nonnegative diagonal matrices $D_l^{-1} Z_1$ and $D_u^{-1} Z_2$ thus enhancing its diagonal dominance. The algorithm has been tested for different grids $h = 1/32, 1/64, 1/128$ with values of $n = 1024, 4096, 16384$, respectively, for different values of $\lambda = 1, 4, 6, 10$. The initial vectors for the experiments are $v^{(0)} = s^{(0)} = z^{(0)} = [1, \ldots, 1]^T$ with the obstacles $v_l = [0, \ldots, 0]^T, v_u = [4, \ldots, 4]^T$. For the last λ-value we reported a failure since a number of backtracking larger than the allowed maximum (=5) have been recorded. Actually, for $\lambda > 6.8$ the algorithm did not achieve convergence (this result is well documented in the literature, see [8]). The sequential results for the cases $\lambda = 1, 4, 6$ are reported in Table 1. The CPU times refer to the computation on a 600 Mhz Alpha workstation with 512 Mb RAM.

Table 1. Results obtained with the *inexact interior point Newton method* for the obstacle Bratu problem with three different mesh sizes and four values of λ (nl= nonlinear, it= iterations, s=seconds on a 600 Mhz Alpha workstation).

λ	n	nl it.	tot lin it.	CPU (s)	$\|H\|$
1	1024	13	397	0.24	0.12371E-09
4	1024	11	335	0.23	0.21711E-08
6	1024	12	374	0.25	0.13904E-12
1	4096	14	873	2.61	0.12668E-08
4	4096	13	828	2.56	0.13289E-10
6	4096	12	840	2.57	0.44356E-11
1	16384	15	1779	36.21	0.34762E-08
4	16384	14	1700	34.46	0.30286E-09
6	16384	14	2021	40.68	0.79480E-09

3.2 The Lubrication Problem

A very difficult problem from the point of view of nonlinearity is represented by the Elastohydrodynamic Lubrication Problem [5] which consists of two integral equations coupled with a differential equation – the Reynold's equation. Given the parameters α, λ and an inlet point x_a, find the pressure $p(x)$, the thickness $h(x)$, the free boundary x_b and the variable k satisfying:

$$h(x) = x^2 + k - \frac{2}{\pi} \int_{x_a}^{x_b} \ln|x - s| \, ds \quad \text{in} \quad [x_a, \infty)$$

$$\frac{d}{dx}\left(\frac{h^3(x)}{e^{\alpha p}} \frac{dp}{dx} \right) = \lambda \frac{dh}{dx} \quad \text{in} \quad [x_a, x_b], \qquad \frac{2}{\pi} \int_{x_a}^{x_b} p(s) \, ds = 1$$

with the free boundary conditions $p(x_a) = 0$ and $p(x_b) = \dfrac{dp}{dx}(x_b) = 0$. The discretization of this problems yields a highly nonlinear and dense system of

equations. We solve the linearized system with a direct method (Lapack routines). In Table 2 we show the results obtained with the *inexact interior point*

Table 2. Results obtained for the Lubrication Problem with the *inexact interior point Newton method* with $\alpha = 2.832$, $\lambda = 6.057$.

n	nl it.	CPU (s)	Jacobian	LU factor	LU solver	$\|H\|$
200	14	1.30	0.61	0.28	0.01	0.18840E-06
1000	19	69.41	20.75	35.59	0.68	0.13107E-06

Newton method with $\alpha = 2.832$, $\lambda = 6.057$ using $n = 200$ and $n = 1000$ points of the discretization of the interval $[-3, 2]$. The initial vector is chosen as

$$x_i^{(0)} = \frac{3}{4}\left(1 - \frac{|x_a + hi|}{2}\right), \qquad h = \frac{5}{n}$$

From the table we note that the major part of the computation is represented by the construction and factorization of the Jacobian matrix. This suggests that a Quasi-Newton approach may drastically reduce the CPU time of a single iteration. In Figure 1 the nonlinear convergence profile is provided, showing the superlinear rate of the convergence of the Inexact Newton method. Figures 2 and 3 display the plots of the film thickness and the pressure, respectively. They compare well with the results of the literature [9].

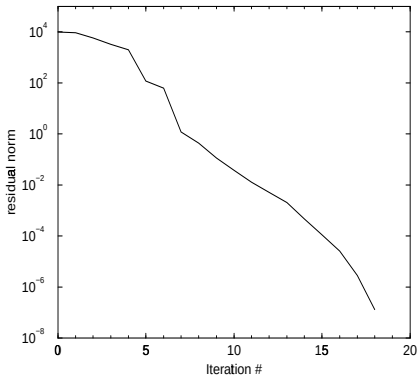

Fig. 1. Convergence profile

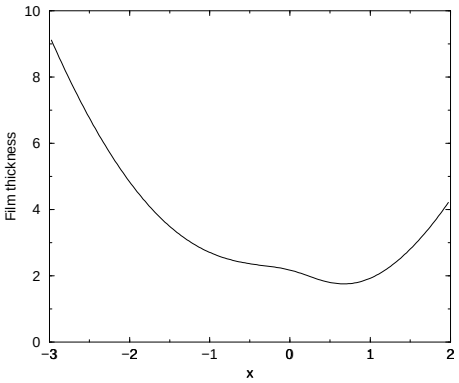

Fig. 2. Film thickness

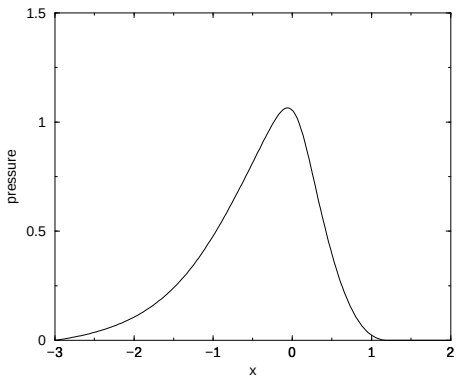

Fig. 3. Pressure

4 Numerical results II (parallel)

4.1 Parallel results for nonlinear problems

In this section we show the results obtained in the solution of the nonlinear system (1) applying the Newton-Cimmino method. As we mentioned above at the end of section 1, to overcome the problem of the costly solution of the least square subproblems (4), we adopt a suitable *block row partitioning* of the matrix A in such a way that $A_i A_i^T = I, \quad i = 1, \ldots q$, and consequently, $A_i^+ = A_i^T$. This partitioning [11] is always possible for every sparse matrix and produces a number q of blocks A_i whose rows are mutually orthogonal. The numerical results of Table 3 were obtained on a CRAY T3E under the MPI environment for two sparse problems (also solved in [12, 4] adopting a simple block partitioning, using the iterative algorithm LSQR), which arise from Finite Difference discretization in the unit square Ω of the following PDEs:

$$\text{1. Poisson problem} \qquad -\Delta u - \frac{u^3}{1+x^2+y^2} = 0 \quad \text{in } \Omega, \quad + \text{ b.c.} \tag{9}$$

$$\text{2. Bratu problem [8]} \qquad -\Delta u - \lambda e^u = 0 \quad \text{in } \Omega, \quad \lambda \in \mathbb{R}, \quad + \text{ b.c.} \tag{10}$$

The linear system is solved using a tolerance $\varepsilon_2 = 10^{-5}$ while the Newton iteration stops whenever the relative residual norm is less than $\varepsilon_2 = 10^{-4}$. From

Table 3. Time (in seconds), speedups $S_p = T_1/T_p$, number of outer and inner iterations k_{NEWT}, k_{CG}, obtained for solving the two test problems, using the row-orthogonal partitioning on a CRAY T3E under the MPI environment.

Poisson problem					
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
Time (speedup)	7.22	5.74 (1.3)	5.79 (1.3)	6.05 (1.2)	6.83 (1.1)
k_{NEWT}	2	2	2	2	2
k_{CG}	1123	1123	1123	1123	1123

Bratu problem					
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
Time (speedup)	6.70	5.11 (1.3)	4.96 (1.4)	5.33 (1.3)	8.01 (0.8)
k_{NEWT}	4	4	4	4	4
k_{CG}	630	630	630	630	630

Table 3 we can see that the speedups are not completely satisfactory, reaching the maximum value of 1.4 for $p = 4$ processors. This fact is mainly due to the cost of the communication routine MPI_ALLREDUCE which performs the communication of the local pseudoresiduals and their sums on every processor. This operation is costly, and its cost increases with the number of processors.

A most effective parallel solution was obtained with the use of a standard Krylov method with AINV as a preconditioner [2]. AINV is based on the incomplete construction of a set of biconjugate vectors. This process produces two triangular factors Z and W and a diagonal matrix D so that: $A^{-1} \approx ZD^{-1}W^T$. Therefore, application of the preconditioner consists in two matrix-vector products and a diagonal scaling. These matrix-vector products have been parallelized exploiting data locality as in [3], minimizing in this way the communication among processors. The incompleteness of the process is driven by a tolerance parameter ε. Previous (sequential) experimental results show that a choice of $\varepsilon \in [0.05, 0.1]$ leads to a good convergence of the Krylov subspace methods, very similar to that obtained using the ILU preconditioner. In our test cases we choose $\varepsilon = 0.05$.

In Table 4 we show the results when BiCGstab is employed as the linear solver using both AINV and the diagonal scaling (Jacobi) as the preconditioners. The CPU time on p processors (T_p) is measured in seconds on a CRAY T3E. From the results we note that for the small problem ($n = 4096$), as expected, the speedups S_p are not very high. However, for the $n = 65\,536$ problem they reach a value of 19 (AINV) and 21 (Jacobi) on 32 processors. Note that in all the runs the CPU time needed by AINV is less than the one required by Jacobi. Moreover, the AINV preconditioner shows a degree of parallelism comparable with that of the diagonal scaling.

Table 4. Results obtained on the CRAY T3E for the Bratu problem employing the AINV and Jacobi preconditioners.

		AINV(0.05)				Jacobi			
n	p	T_p	nl	lin	S_p	T_p	nl	lin	S_p
4096	1	0.74	3	57	–	0.92	4	134	–
	2	0.47	3	57	1.58	0.57	4	133	1.61
	4	0.33	3	57	2.24	0.42	4	135	2.19
	8	0.25	3	57	2.96	0.33	4	134	2.78
16 384	1	5.06	3	107	–	5.46	4	243	–
	2	2.82	3	107	1.79	3.32	4	268	1.64
	4	1.53	3	107	3.31	1.82	4	270	3.00
	8	0.90	3	107	5.62	1.17	4	266	4.67
	16	0.61	3	107	8.29	0.74	4	257	7.38
65 536	1	40.75	4	224	–	41.56	4	497	–
	2	21.02	4	223	1.93	20.42	4	479	2.03
	4	11.00	4	225	3.70	13.31	5	582	3.12
	8	5.79	4	224	7.03	5.76	4	462	7.21
	16	3.48	4	224	11.70	3.24	4	466	12.82
	32	2.15	4	219	18.95	2.05	4	493	20.27

4.2 Parallel results for nonlinear complementary problems

As in section 4.1, we also solved in parallel the obstacle Bratu problem (8) via the *inexact interior point method*, using the BiCGstab method as linear solver with AINV and Jacobi as preconditioners. In Table 5 we show the results obtained. The same considerations of section 4.1 hold, even with larger speedup values.

Table 5. Results obtained on a CRAY T3E for the obstacle Bratu problem employing the AINV and Jacobi preconditioners.

		AINV(0.05)				Jacobi			
n	p	T_p	nl	lin	S_p	T_p	nl	lin	S_p
4096	1	5.30	14	402	–	5.82	14	946	–
	2	2.96	14	405	1.73	3.23	14	938	1.76
	4	1.69	14	400	2.75	1.99	14	941	2.60
	8	1.05	14	400	3.89	1.23	14	948	3.69
	16	0.73	14	403	4.83	0.92	14	943	4.72
16 384	1	41.66	15	823	–	46.05	15	1959	–
	2	21.56	15	820	1.93	24.11	15	1943	1.91
	4	11.10	15	795	3.75	13.12	15	1967	3.51
	8	6.21	15	819	6.71	7.13	15	1968	6.46
	16	3.75	15	828	11.11	4.43	15	1970	10.40
	32	2.56	15	821	16.27	2.88	15	1950	15.98
65 536	1	321.32	16	1688	–	346.82	16	4026	–
	2	162.25	16	1651	1.93	180.30	16	4022	1.92
	4	84.64	16	1676	3.79	94.47	16	4022	3.67
	8	50.22	16	1672	6.39	51.39	16	4046	6.74
	16	24.31	16	1706	13.21	28.28	16	4043	12.26
	32	14.04	16	1675	22.88	15.05	16	4027	23.04

5 Conclusions and future topics

In this paper we experimented that the Inexact Newton method performs well in solving nonlinear problems and mixed nonlinear complementary problems both in sequential and in parallel computations. We adopted two different parallel preconditioners in the iterative solution of sparse problems: the row-action Cimmino method [11] and the incomplete inverse AINV [2]. While the latter obtains good results (speedup values up to 23 with 32 processor), the former heavily suffers for the overhead due to the MPI communication routines. Future work will address the parallel implementation of an Inexact Quasi-Newton interior point method applied to the solution of mixed nonlinear complementary problems.

References

1. S. Bellavia An Inexact Interior Point method *Journal of Optimization Theory and Applications*, vol. 96, 1 (1998).

2. M. Benzi and M. Tůma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, *SIAM J. Sci. Comput.*, 19 (1998), pp. 968–994.
3. L. Bergamaschi and M. Putti. Efficient parallelization of preconditioned conjugate gradient schemes for matrices arising from discretizations of diffusion equations. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing*, March, 1999. (CD–ROM).
4. L. Bergamaschi, I. Moret, and G. Zilli, Inexact block Quasi-Newton methods for sparse systems of nonlinear equations, *J. Fut. Generat. Comput. Sys.* (2000) (in print).
5. I. Bongartz, I., A. R. Conn, N. I. M. Gould and P. L. Toint, CUTE:*Constrained and unconstrained testing environment*, Research Report, IBM T.J. Watson Research Center, Yorktown Heights, NY, 1993.
6. R. S. Dembo, S C. Eisenstat, and T. Steihaug, Inexact Newton methods, *SIAM. J. Numer. Anal.* 19, 400-408, 1982.
7. S. P. Dirske, and M. C. Ferris *MCLIB: A collection of nonlinear mixed complementary problems*, Tech. Rep., CS Depth., University of Winsconsin, Madison, WS, 1994.
8. D. R. Fokkema, G. L. G. Slejipen and H. A. Van der Vorst, Accelerated Inexact Newton schemes for large systems of nolinear equations, *SIAM J. Sci. Comput.*, 19 (2), 657-674, 1997.
9. M. M. Kostreva, Elasto-hydroninamic lubrication: A non-linear complementary problem. *International Journal for Numerical Methods in Fluids*, 4:377-397, 1984
10. S. J. Wright, *Primal-Dual Interior-Point Methods*, Siam, Philadelphia, 1997.
11. G. Zilli, Parallel method for sparse non-symmetric linear and non-linear systems of equations on a transputer network,
12. G. Zilli and L. Bergamaschi, Parallel Newton methods for sparse systems of nonlinear equations, *Rendiconti del Circolo Matematico di Palermo* II-58, 247-257, 1999.