Ann. Univ. Ferrara - Sez. VII - Sc. Mat.
Supplemento al Vol. XLV, 1-0 (2000)

Parallel inexact Newton and interior point method

L. BERGAMASCHI (*) AND G. ZILLI(*)

SUNTO - In questo lavoro vengono presentati risultati ottenuti nella soluzione di sistemi nonlineari di grandi dimensioni con i metodi Newton e Quasi-Newton inesatti combinati con un solutore iterativo a blocchi di tipo row-action (Cimmino). Si propone inoltre un opportuno partizionamento della matrice Jacobiana (o di sue approssimazioni). In tale modo si ottengono blocchi di righe mutuamente ortogonali che permettono di risolvere facilmente i sottoproblemi ai minimi quadrati introdotti dal solutore lineare. Vengono presentati alcuni risultati numerici ottenuti sul CRAY-T3E dall'applicazione di tale metodo ad alcuni problemi test nonlineari derivanti dalla discretizzazione di equazioni differenziali alle derivate parziali. Sono infine mostrati alcuni risultati preliminari che dimostrano l'efficienza del metodo applicato a problemi di complementarietà nolineare misti risolti con metodi di punto interno.

ABSTRACT - In this paper we present the results obtained in the solution of sparse and large systems of non-linear equations by inexact Newton methods combined with an block iterative row-projection linear solver of Cimmino-type. Moreover, we propose a suitable partitioning of the Jacobian matrix A. In view of the sparsity, we obtain a mutually orthogonal row-partition of A that allows a simple solution of the linear least squares subproblems. We present numerical results obtained on a CRAY-T3E when this method is used to solve both non linear problems arising from dicretization of PDEs. Preliminary sequential results are also shown in the solution of nonlinear mixed complementary problems solved with interior point methods.

1. – The Inexact Newton and Quasi-Newton Cimmino method

We are concerned with the numerical solution of a system of nonlinear equations

$$F(x) = 0 \qquad F = (f_1, ..., f_n)^T \tag{1}$$

(*)Indirizzo dell'autore: Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate Universitá di Padova, Via Belzoni 7, 35131 Padova, ITALY

where $F : R^n \to R^n$ is a nonlinear C^1 function, and its Jacobian matrix $J(x)$ is sparse and n is large. Moreover, in Section 5 we solve the nonlinear systems which arise from the application of the primal-dual interior point method (see [7]) to mixed complementary problems.

Let $As = b$ be the linear system to be solved. Here below we give the general lines of the block iterative Cimmino method. Let us partition A into p row-blocks: $A_i, i = 1, \dots, p$, i.e. $A^T = [A_1, A_2, \dots, A_p]$ and partition the vector b conformally. Then the original system is premultiplied (*preconditioning*) by

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

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 $i = 1, \dots, p$, $P_i = A_i^+ A_i$ 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 $\delta_{i,k}$

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

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

Combining the classic Newton method and the block Cimmino method we obtain the block Inexact Newton-Cimmino algorithm [9] and [11], in which at a major outer iteration the linear system $J(x_k)s = -HF(x_k)$, where $J(x^*)$ is the Jacobian matrix, is solved in *parallel* by the block Cimmino method (3).

In the Inexact Newton method [3] for solving $F(x) = 0$, the step s_k satisfies

$$||J(x_k)s_k + F(x_k)|| \le \eta_k ||F(x_k)|| \tag{5}$$

where η_k is the forcing term. In [3] it is proved that the method has a local, linear convergence in an appropriate norm, and it may converge superlinearly or even quadratically under convenient assumptions.

ALGORITHM: INEXACT NEWTON-CIMMINO

```
Let $x_0$ be the initial vector, $k = 0$.
WHILE $||F(x_k)|| > \varepsilon_1 \, ||F(x_0)||$ DO
    • compute $A = J(x_k)$
    • partition $A^T = (A_1^T, \dots, A_p^T)$,   $F \equiv F(x_k)^T = (F_1^T, \dots, F_p^T)$
    • compute the preconditioner $H$
    • solve $HA\Delta x_k = HF$ by the Conjugate Gradient method noting that
      the product $y = HAv$ is implemented in parallel as

          $y = \sum_{i=1}^{p} A_i^T (A_i A_i^T)^{-1} A_i v_i$

      where every term of the sum is computed by a different processor $i$.
    • $x_{k+1} = x_k + \Delta x_k$   $k = k+1$
END WHILE
```

The exit test for the Cimmino iterations is $||r_{k,m}|| \le \varepsilon_2 ||F(x_k)||$.

To overcome the expensive computation of the Jacobian matrix at every nonlinear iteration a Quasi Newton method has been developed, which tries the solution of $F(x) = 0$ by computing at each Newton iteration an approximation B_k of $J(x_k)$. We now report the following Broyden-like method:

ALGORITHM: INEXACT QUASI-NEWTON CIMMINO

```
Let $x_0$ be the initial vector, $k = 0$, $A \equiv J(x_0)$.
Let $H$ be defined by (2). Set $B_0 = HA$.
WHILE $||F(x_k)|| > \varepsilon_1 \, ||F(x_0)||$ DO
    Compute an approximation $s_k$ to the solution of the linear system
    (6)    $B_k s = -HF(x_k)$
    Set
        $x_{k+1} = x_k + s_k$
        $y_k = H(F(x_{k+1}) - F(x_k))$,
        $u_k = \frac{(y_k - B_k s_k)}{||s_k||_{HA}}$
        $v_k = \frac{HAs_k}{||s_k||_{HA}}$
        $B_{k+1} = B_k + u_k v_k^T$,   $k = k+1$.
END DO
```

Concerning the solution of (6), it can be shown [1] that it can be reduced to the system $HAs = z$ which is solved approximately with the CG with $s_0 = 0$ as the initial vector. $HAs = z$ is solved *concurrently* with the row-projection (Cimmino) method.

Note that one of the most expensive operations is represented by the construction of the preconditioner H. In the Quasi-Newton method this preprocessing stage is carried out once and for all since at each outer iteration we have to solve a linear system with the same $A = B_0$ as the coefficient matrix.

2. – Convergence results

Now we give a convergence theorem under weak regularity conditions for F. Referring to system (6) for each $k = 0, 1, \ldots$, let us set

$$\varepsilon_k = \frac{||s - s_k||_{HA}}{||s||_{HA}}. \tag{7}$$

Our hypotheses are the following:

Assumptions 2.1 *Let $D \subset R^n$ be an open convex set. Let $F : D \to R^n$ be Frechet differentiable. Let $x^* \in D$ such that $F(x^*) = 0$ and $J(x^*)$ is non singular. Let us set*

$$w(x,y) = \frac{||F(x) - F(y) - J(x^*)(x-y)||_2}{||x-y||_2}, \qquad \forall\ x, y \in D.$$

Let $S(r) = \{x : ||x^* - x||_2\} \le r\}$ and let $R > 0$ such that $S(r) \subset D$ for $0 < r \le R$. Then, for any $0 < r \le R$, define

$$\omega(r) = \sup_{(x,y)\in S(r)} w(x,y). \tag{8}$$

Accordingly, we assume that, for every $0 < \varepsilon < 1$, it is $\lim_{r\to 0} \sum_{k=0}^{\infty} \omega(\varepsilon^k r) = 0$.

Theorem 2.1 *Let Assumptions 2.1 hold. For every $\varepsilon \in (0,1)$ there exist a $\delta > 0$ and a $\eta > 0$ such that if $||x^* - x_0||_{HA} \le \delta$, $||J(x^*) - A||_2 \le \eta$ and $\varepsilon_k \le \varepsilon_0 < \varepsilon$ for each k, then B_k is invertible, the sequence of iterates $\{x_k\}$ converges to x^* and*

$$||x^* - x_{k+1}||_{HA} \le \varepsilon\ ||x^* - x_k||_{HA} \tag{9}$$

Moreover if $\lim_{k\to\infty} \varepsilon_k = 0$, the convergence is q*-superlinear.*

The proof of the theorem is reported in [1].

3. – Numerical results

In this section we report some numerical results in which we solve the subproblems (4) *concurrently* with the iterative Lanczos algorithm LSQR [6]. We present also the results of the Cimmino algorithm applied to a linear problem (the nonsymmetric Sameh problem) since the parallelism of the whole Newton-Cimmino method essentially depends on the performance of the solver of the linearized system. In this section we always assume that the number of processors corresponds to the number of blocks, i.e. $p = q$.

We now briefly describe the main features of the test problems, which come from discretization by 5-point Finite Differences of linear and nonlinear PDE. In the three test cases the domain Ω is the unitary square, discretized into a regular $l \times l$ grid with a total number of unknowns equals to $n = l \times l$ ($l = 64$ in our runs).

- *Poisson problem,*

$$-\Delta u - \frac{u^3}{1 + x^2 + y^2} = 0, \quad \text{in } \Omega$$

$$u(0,y) = 1, \quad u(1,y) = 2 - e^y, \quad u(x,0) = 1, \quad u(x,1) = 2 - e^x.$$

- *Bratu problem* [2]

$$(10) \qquad -\Delta u - \lambda e^u = 0 \ \text{ in } \Omega, \qquad u = 0 \ \text{ on } \partial\Omega, \qquad \lambda \in \mathbb{R}$$

  It is known that there exists a critical value of λ, λ^* such that problem (10) has two solutions for $\lambda < \lambda^*$ and no solutions for $\lambda > \lambda^*$.

- *Sameh problem* [9]

$$-\Delta u + 1000\, e^{xy}\, (u_x - u_y) = 0, \qquad u = x + y \quad \text{on } \partial\Omega.$$

All the results are obtained on the CRAY T3E installed (1998) at CINECA, Bologna, Italy. The outer exit test is based on the relative residual $||F(x_k)|| \leq \varepsilon_1\, ||F(x_0)||$. At each nonlinear iteration k_{CG} Cimmino iterations are performed until the following test is satisifed: $||r_{k,m}|| \leq \varepsilon_2 ||F(x_k)||$. Each of these iterations essentially consists of a conjugate gradient procedure in which the pseudoresiduals $\delta_{i,k}$ (4) are *concurrently* computed by k_{LSQR} LSQR iterations, within a tolerance ε_3. The Fortran code runs under **MPI** implementation. Table 1 refer to LSQR Inexact Newton-Cimmino method. As already remarked in [1], we may note that the most expensive part of the algorithm is represented by the LSQR solution of the underdetermined subproblems (see Table 3). From this Table it is worth noting that with $p = 1$ the Cimmino method acts as an *exact* preconditioner applied to $n \times n$ problem and hence the solution is attained at the very first iteration (with $k_{LSQR} = 2258$ LSQR iterations). With $p \geq 2$,

Table 1: Time (in seconds) and speedups T_2/T_p obtained with the *Inexact Newton-Cimmino method.* Moreover, the average number of outer k_{CG}, the inner k_{LSQR} iterations performed at each step and the overall number of LSQR iterations are shown.

Poisson problem, $\varepsilon_1 = 10^{-3}, \varepsilon_2 = 10^{-4}$.						
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$
Time	34.93	353.47	233.93	122.12	53.83	27.14
speedup	–	1	1.51	2.89	6.56	13.02
k_{NEWT}	2	2	2	2	2	2
k_{CG}	1	89	212	303	391	525
k_{LSQR}	4276	1715	813	372	161	68
$k_{CG} \times k_{LSQR}$	4276	165400	176960	113920	62841	35428
Bratu problem, $\varepsilon_1 = 10^{-4}, \varepsilon_2 = 10^{-5}$.						
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$
Time	43.04	537.12	294.13	197.45	108.99	69.72
speedup	–	1	1.82	2.72	4.92	7.70
k_{NEWT}	4	4	4	4	4	4
k_{CG}	1	48	86	141	230	318
k_{LSQR}	1300	1021	341	278	119	62
$k_{LSQR} \times k_{CG}$	1300	49000	47888	40336	27512	19676
Sameh linear problem, $, \varepsilon_2 = 10^{-3}$.						
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$
Time	9.4	28.7	18.5	11.4	8.2	6.1
speedup	–	1	1.6	2.5	3.5	4.7
LSQR time (%)	99.9%	99.8%	99.7%	99.2%	96.3%	96.1%
k_{CG}	1	28	45	64	90	127
k_{LSQR}	2258	384	211	118	73	38
$k_{CG} \times k_{LSQR}$	2258	10740	9481	7542	6569	4843

the number of Cimmino iterations and the corresponding CPU time increase largely, due to the ill-conditioning of the Laplace equation. For these problems, it is therefore reasonable to evaluate the performance of the parallel Cimmino method with respect to the T_2 CPU time ($p = 2$). The speedups reported in Table 1 are consistently computed as $S_p = T_2/T_p$, $p \geq 2$. In other problems [1], this behavior does not occur. In Table 1, we finally note that starting from 8 processors we obtain a total CPU time which is less than the $p = 1$ CPU time.

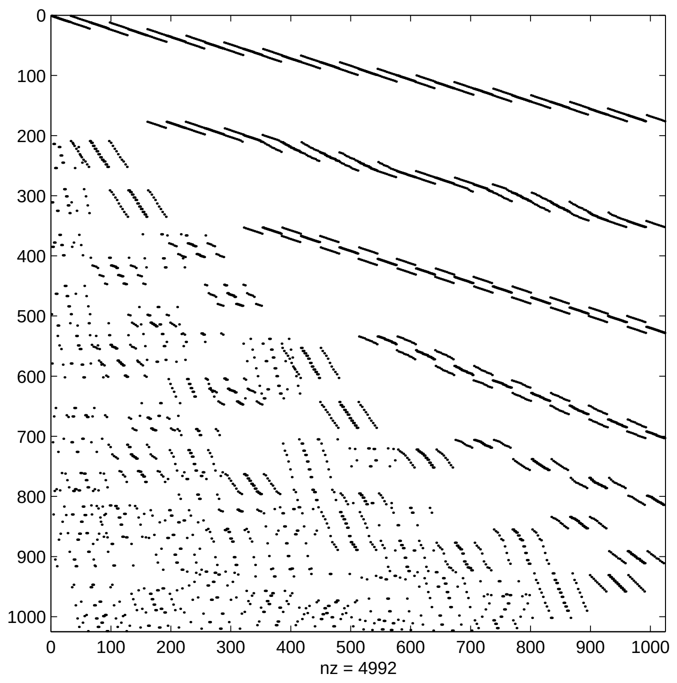

Figure 1: Pattern of the Sameh linear test problem after the reordering in 7 row-orthogonal blocks

4. – Row-orthogonal partitioning

To overcome the problem of the costly solution of the least squares subproblems (4), we adopt a suitable block row partitionings of the matrix A in such a way that $A_i A_i^T = I, \quad i = 1, \dots q$, and consequently, $A_i^+ = A_i^T$. This allows to simplify the solution of the least squares subproblems at each step of the inner iteration. This partitioning [8] is always possible for every sparse matrix and produces a number q of blocks A_i whose rows are mutually orthogonal. As an example, we show in Figure 1 the pattern of the Sameh linear test problem after the reordering in 7 row-orthogonal blocks. The numerical results relative to the row-partitioning for the same test problems of section 1 are shown in Table 2.

From Tables 1 and 2 we can make the following observations. The Conjugate Gradient applied to the row-orthogonal partitioned matrices takes a larger number of iterations with respect to the LSQR case. For example, in the Poisson test case the number of outer iterations are constantly equal to 1123 in the orthogonal case while they range from 1 to 525 using the LSQR algorithm. The same behavior is evidenced by the other test problems.

Poisson problem, $\varepsilon_1 = 10^{-4}, \varepsilon_2 = 10^{-5}$.					
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
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, $\varepsilon_1 = 10^{-4}, \varepsilon_2 = 10^{-5}$.					
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
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

Linear problem, $\varepsilon_2 = 10^{-8}$					
$n = 4096$	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$
Time (speedup)	3.16	2.30 (1.4)	2.00 (1.6)	2.04 (1.6)	2.30 (1.4)
k_{CG}	696	695	694	694	694

Table 2: Time (in seconds), speedups T_1/T_p, number of outer and inner iterations k_{NEWT}, k_{CG}, obtained for solving the three test problems, using the row-orthogonal partitioning.

On the contrary, the orthogonal variant of the algorithm is more convenient from the point of view of CPU time, as can be seen by comparing the overall CPU times in the three test cases for a fixed number of processors. Note that now a single iteration is much less costly than in the LSQR algorithm, since the solution of a linear subproblem is replaced by a cheap multiplication for a diagonal matrix (or even the identity). Moreover, the speedups in the second algorithm are correctly computed as T_1/T_p because in this case the CPU time decreases from 1 processor forward. However, they 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. Table 3 refers to the linear problem, and yields the time spent by the most expensive routine (the matrix-vector product) and by the MPI_ALLREDUCE routine. Note the good speedup obtained by the matrix-vector product, while the time spent by the MPI_ALLREDUCE routine becomes a large part of the overall CPU time as the number of processors increase, going to more than 50% in the case with 16 processors. Differently, in the LSQR case the times spent by the communication represent a little portion compared to the much higher times required to solve the least squares subproblems.

Table 3: CPU time needed by the matrix vector products and by the reduce operation in the linear test case.

p	$w = A_i v$	(T_p/T_1)	$w = A_i^T v$	(T_p/T_1)	mpi_allreduce (%)	overall
1	0.70	(-)	0.80	(-)	0.18 (06)	3.16
2	0.35	(2.0)	0.43	(1.8)	0.46 (20)	2.30
4	0.19	(3.7)	0.27	(3.0)	0.75 (25)	2.00
8	0.10	(7.0)	0.18	(4.4)	1.04 (51)	2.04
16	0.05	(13.9)	0.12	(6.7)	1.33 (58)	2.30

5. – Inexact Newton for non linear complementary problems

Let us consider the following system of constrained equations:

$$H(v,s,z) = \begin{pmatrix} F(v,s,z) \\ SZe \end{pmatrix} = 0 \qquad (s,z) \geq 0 \tag{11}$$

where $F : \mathbb{R}^{n+2m} \to \mathbb{R}^n$, 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 method for the solution of (11) requires the solution of the nonlinear system $H(x) = 0$. Using the Newton method we have to solve at every iterations a linear system of the form

$$H'(x_k)\Delta x = -H(x_k) + \sigma_k \mu_k e_0 \tag{12}$$

where $\mu_k = (s_k^T z_k)/m$, $\sigma_k \in]0,1[$.

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 (12) becomes

$$H'(x_k)\Delta x = -H(x_k) + \sigma_k \mu_k e_0 + r_k \tag{13}$$

where r_k is the residual of the iterative method applied to the linear system satisfying $||r_k|| \leq \eta_k \mu_k$, with η_k the *forcing term* of the inexact Newton method [3]. Global convergence is assured by means of backtracking.

We have applied the inexact interior point algorithm for the solution of the following obstacle Bratu problem [2] and the Lubrication problem [2].

5.1. – *The Bratu-obstacle problem.* –

$$\begin{aligned} f(v) &= z_1 - z_2 \\ z_1^T(v - v_l) &= 0 \\ z_2(v_u - v) &= 0 \\ s_1 &= v - v_l \\ s_2 &= v_u - v \end{aligned} \tag{14}$$

Table 4: Results for the obstacle Bratu problem with three different mesh sizes and four values of λ (nl=non linear, it= iterations, s=seconds).

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

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

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

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.

For the obstacle problem system (13) at step k can be written as

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

where $D_l = \mathrm{diag}((v - v_l)_1, \ldots, (v - v_l)_n)$, and $D_u = \mathrm{diag}((v_u - v)_1, \ldots, (v_u - v)_n)$, $B = f'(v)$.

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. The Schur complement approach yields a system in the Δv unknown only:

$$C\Delta v = r, \quad \text{where} \quad C = B + D_l^{-1}Z_1 + D_u^{-1}Z_2,\ r = b_1 + D_l^{-1}b_2 - D_u^{-1}b_3.$$

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 bactracking larger than the allowed maximun (=5) have been recorded. Actually, for $\lambda > 6.8$ the algorithm did not achieve convergence (this result is well documented in the literature, see [5]). The sequential results for the cases $\lambda = 1, 4, 6$ are reported in Table 4. The CPU times refer to the computation ob a 600 Mhz Alpha workstation with 512 Mb RAM.

5.2. – *The Lubrication Problem.* – A very difficult complementary problem, from the point of view of nonlinearity, is represented by the Elastohydrodynamic Lubrication Problem [2] which consist of two integral equations coupled with a partial differential equation – the Reynold's equation. Each problem is characterized by the pair (α, λ). The discretization of this problems yields a higly nonlinear dense system of equations. We solve the linearized system with a direct method.

Table 5: Results for the Lubrication Problem with $\alpha = 2.832$, $\lambda = 6.057$

n	nl it.	CPU				
		tot	Jacobian	LU factor.	LU solver	$\|\|H\|\|$
200	14	1.30	0.61 (46%)	0.28 (22%)	0.01(1%)	0.18840E-06
1000	19	69.41	20.75 (29%)	35.59 (51%)	0.68(0%)	0.13107E-06
2000	22	478.59	97.70 (20%)	320.80 (67%)	3.06(0%)	0.69400E-06

In Table 5 we show the results obtained with $\alpha = 2.832$, $\lambda = 6.057$ using $n = 200, 1000$ and 2000 points of discretization of the interval $[-3, 2]$. Note that for large problems the evaluation and factorization of the Jacobian matrix become the largest part of the total CPU time (80% and 87% in the cases with $n = 1000$ and 2000). Work is in progress to ascertain the possibility of using the Quasi-Newton method which will allow the Jacobian computation $J(x_0)$ and factorization only once, hence drastically reducing the CPU time.

As a final remark, further work is undergoing to apply the parallel Inexact Newton and Quasi-Newton Cimmino algorithms of Section 3 to the mixed complementary problems.

REFERENCES

[1] L. Bergamaschi, I. Moret, and G. Zilli, *Inexact block quasi-newton methods for sparse systems of nonlinear equations,* J. Fut. Generat. Comput. Sys. (2000) (submitted).

[2] Bongartz, I., Conn, A.R., Gould, N.I.M., and Toint, P.L., *CUTE:Constrained and unconstrained testing environment*, Research Report, IBM T.J. Watson Research Center, Yorktown Heights, NY, (1993.)

[3] Dembo, R.S., Eisenstat, S.C., and Steihaug, T., *Inexact Newton methods,* SIAM. J. Numer. Anal., **19**, (1982). pp. 400–408,

[4] Dirske, S.P. and Ferris, M. C. *MCLIB: A collection of nonlinear mixed complementary problems*, Tech. Rep., CS Depth., University of Winsconsin, Madison, WS, (1994).

[5] Fokkema D. R., Slejipen G.L.G. and Van der Vorst H.A., *Accelerated Inexact Newton schemes for large systems of nonlinear equations.* SIAM J. Sci. Comput., **19 (2)**, pp. 657-674, (1997).

[6] Paige, G.G. and Saunders, M.A., LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software* **8**, 43-71, 1982.

[7] Wright, S.J. *Primal-Dual Interior-Point Methods*, Siam, Philadelphia, (1997).

[8] Zilli, G., *Parallel implementation of a row-projection method for solving sparse linear systems,* Supercomputer, **53, X-1**, pp. 33-43, (1993).

[9] Zilli, G., *Parallel method for sparse non-symmetric linear and non-linear systems of equations on a transputer network,* Supercomputer, **6, XII-4**, pp. 4-15, (1996).

[10] Zilli, G. and Bergamaschi, L., *Parallel Newton methods for sparse systems of nonlinear equations,* Rendiconti del Circolo Matematico di Palermo, **II-58**, pp. 247-257, (1999).

[11] Zilli, G. and Bergamaschi, L., *Truncated block Newton and Quasi Newton methods for sparse systems of nonlinear equations. Experiments on parallel platforms* In: M. Bubak, J. Dongarra, J. Wasniewski (Eds.), Recent advances in PVM and MPI, Lectures Notes in Computer Sciences, 1332, Springer, pp. 390-397, (1997).