Consiglio Nazionale delle Ricerche

# ISTITUTO DI ELABORAZIONE DELLA INFORMAZIONE

PISA

A VERY RAPILLY CONVERGENT ITERATIVE METHOD FOR PARALLEL INVERSION OF MATRICES

B. Codenotti, F. Romani

Nota interna B84-08

Luglio 1984

A VERY RAPIDLY CONVERGENT ITERATIVE METHOD FOR PARALLEL INVERSION OF MATRICES

B. CODENOTTI and F. ROMANI
Istituto di Elaborazione dell'Informazione - CNR,
via S.Maria 46, PISA, ITALY.

ABSTRACT. A very fast iterative method is presented, for the inversion of matrices of the form A=I-P, where P is a convergent matrix. The method is well suitable for parallel implementation.

The convenient number of iterations required is the logarithm of the number of iterations required by a classical iterative method applied to the splitting I-P of the matrix. Asymptotically, the method requires O(log n A(log n log log n)) steps on $n^3$ processors, where A(p) is the complexity of the arithmetic operations with p digits. Further, a detailed study of the total complexity for finite values of n shows the relations among the number of steps, the spectral radius and the dimension of the matrix.

KEY WORDS. Matrix Inversion, Iterative Methods, Parallel Algorithms.



## 1. INTRODUCTION AND PRELIMINARIES

Let us consider a nxn non singular matrix A, that can be written as

$$A = I - P$$

with the spectral radius of P less than one. The matrix A can be obtained from a transformation applied to a more general matrix A' (e.g. as in Jacobi or Gauss-Seidel methods [3,5]).

In this paper, we present a very rapidly convergent iterative method for the inversion of the matrix A, well suitable for parallel implementation.

The convenient number of iterations required by this method is the logarithm of the number of iterations required by a classical iterative methods applied to the splitting I-P of A. Each iteration requires O(log n) steps on $n^3$ processors.

A detailed error analysis allows us to investigate the total complexity for finite values of n, and to derive relations among the number of steps, the spectral radius and the dimension of the matrix. With numerical experiments, simulating the parallel environment, the theoretical error bound and the estimated number of iterations are compared with the effective ones.

In the following, ||.|| denotes the spectral norm and fl(a(op)b) denotes the value of a(op)b, computed using a floating point arithmetic. Moreover, log(x) is the logarithm to base 2 of x, while LOG(x) is the logarithm to base 10 of x.

## 2. THE ITERATIVE METHOD

Let $A = I-P$ be a non singular matrix and assume that $r(P)<1$. Hence

$$A^{-1} = (I-P)^{-1} = \sum_{k=0}^{\infty} P^{k} .$$

Let $X_i$ be nxn matrices defined by the following recurrence relations:

$$(2.1)\qquad \begin{aligned} &X_0 = I, \qquad P_0 = P, \\ &X_{i+1} = P_i X_i + X_i, \\ &P_{i+1} = P_i^{2} . \end{aligned} \qquad i>0$$

It is easy to see that

$$X_i = \prod_{k=0}^{i-1} (I + P^{2^k}) = \sum_{k=0}^{2^i-1} P^k = A^{-1}(I - P^{2^i})$$

and the analytic error $E_A(i)$ is bounded as follows

$$(2.2)\qquad E_A(i) = \| X_i - A^{-1} \| \le \| P^{2^i} \| \, \| A^{-1} \| .$$

REMARK.

The iterative method (2.1) is formally equivalent to the classical method:

$$(2.3)\qquad \begin{aligned} &Y_0 = 0 , \\ &Y_{k+1} = P Y_k + I , \\ &\text{with} \qquad X_k = Y_{2^k} . \end{aligned}$$

Hence the number of iterations required to get a given error bound by (2.1) is the logarithm to base 2 of the one required by (2.3). Each iteration of (2.1) has the cost of two matrix multiplications and one matrix addition. The parallel complexity of one iteration is

$$\left.\begin{array}{l} 2 \ \text{MULTIPLICATIONS} \\ 2\lceil \log n \rceil + 1 \ \text{ADDITIONS} \end{array}\right\} \text{ on } n^3 \text{ PROCESSORS.}$$

To estimate the total complexity of the method, it is necessary to determine the convenient number of iterations and a careful error analysis is required.

We restrict ourselves to the case $P^H = P$, that allows to use the equality $\| P^k \| = r^k$, $k \ge 0$.

## 3. ERROR ANALYSIS

In the case $P^H = P$, the relation (2.2) can be written

$$E_A(i) \le r^{2^i} \, \| A^{-1} \| , \qquad r = r(P).$$

Let us recall some basic results from error analysis theory [4].

LEMMA 3.1

Let A,B be nxn matrices; let u be the relative computer precision and assume $n\,u < 0.1$. Then

$$\|fl(AB)-AB\| \le c_1\, u\, \|A\|\,\|B\| , \qquad \text{where}$$

$$c_1 = \begin{cases} O(n^2) & \text{if the serial algorithm is used,} \\ O(n \log n) & \text{if parallel addition is performed (Fan-in algorithm),} \end{cases}$$

and

$$\|fl(A+B)-(A+B)\| \le u\, c_2\, \|A+B\| , \qquad \text{where}$$

$$c_2 = n^{1/2} .$$

The proof follows from the results of [4] by using the 2-norm instead of the euclidean norm. ∎

PROPOSITION 3.1

Let $fl(P_{i+1})$ be the computed value of $P_{i+1}$ in floating point arithmetic, and let $dP_i$ be the absolute error

$$P_i - fl(P_i).$$

It follows :

$$\| dP_i \| \le c_1\, u\, (2^i - 1)\, r^{2^i} .$$

Proof.

The following equality holds:



$$fl(P_{i+1}) = fl(P_i)^2 + M_i , \quad \text{where}$$

$$\|M_i\| \le c_1\, u\, \|P_i\|^2 \le c_1\, u\, r^{2^{i+1}} . \text{ Hence}$$

$$dP_{i+1} = P_i\, dP_i + dP_i\, P_i + M_i , \quad \text{and}$$

$$\|dP_{i+1}\| \le 2\, \|P_i\|\, \|dP_i\| + \|M_i\| .$$

Assume that $P_0 = P$, $\|dP_0\|=0$. It holds :

$$\|dP_{i+1}\| \le \sum_{j=0}^{i} \Big( 2^{i-j} \prod_{k=j+1}^{i} \|P_k\| \Big) \|M_j\| =$$

$$= \sum_{j=0}^{i} 2^{i-j} \Big( \prod_{k=j+1}^{i} r^{2^k} \Big) \|M_j\| \le$$

$$\le \sum_{j=0}^{i} 2^{i-j}\, r^{(2^{i-j}-1)2^{j+1}}\, c_1\, u\, r^{2^{j+1}} =$$

$$= c_1\, u\, (2^{i+1} - 1)\, r^{2^{i+1}} . \qquad ∎$$

PROPOSITION 3.2

Let $E_R(i) = \|fl(X_i) - X_i\|$ be the 2-norm of the absolute roundoff error in the floating point computation of $X_i$; let a be the maximal eigenvalue of P $(-r\le a\le r)$. Then $E_D(i)$ is bounded by

$$u\,\|A^{-1}\|\,(1+r^{2^{i+1}})\left(c_2\frac{1-a^2}{1-r}+\sum_{j=1}^{i}\frac{1-a^{2^{j+1}}}{1-r^{2^{j+1}}}\,(c_1 2^j r^{2^j}+c_2)\right).$$

Proof.

We have

$fl(X_{i+1})=(P_i+dP_i)fl(X_i)+fl(X_i)+M'_i+A_i$ , where

$\|M'_0\|=0$ , $\|M'_i\|\le c_1 u\,\|fl(P_i)\|\,\|X_i\|$ , $i>0$

and $\|A_i\|\le c_2 u\,\|X_{i+1}\|$.

Hence , if $dX_i = X_i - fl(X_i)$, we obtain

$$X_{i+1}+dX_{i+1} = P_iX_i + P_i dX_i + dP_i X_i + dP_i dX_i + X_i + dX_i + M'_i + A_i .$$

It is easy to see that $dX_i$ and $dP_i$ are $o(u)$; then using the (2.1) and dropping out $o(u^2)$ terms, we obtain

$$dX_{i+1} \doteq (I+P_i)\,dX_i + dP_i X_i + M''_i + A_i , \quad \text{where}$$

$\|M''_0\|=0$ and $\|M''_i\|\le c_1 u\,\|P_i\|\,\|X_i\|$ , $i>0$.

For $i>0$, it holds the following equality:

$$dX_{i+1} = \sum_{j=0}^{i}\prod_{k=j+1}^{i}(I+P_k)(dP_jX_j+M''_j+A_j) =$$



$$= \prod_{k=1}^{i}(I+P_k)A_0 + \sum_{j=1}^{i}\prod_{k=j+1}^{i}(I+P_k)(dP_jX_j+M''_j+A_j) .$$

Taking the 2-norm, we get

$$E_D(i+1)\le u\,c_2\,\Big\|\prod_{k=1}^{i}(I+P_k)\Big\|\,\|X_1\| +$$

$$+\sum_{j=1}^{i}\Big\|\prod_{k=j+1}^{i}(I+P_k)\Big\|\,(\|dP_j\|\,\|X_j\|+\|M''_j\|+\|A_j\|).$$

Observe that, for $j>0$

$$\Big\|\prod_{k=j+1}^{i}(I+P_k)\Big\| = \Big\|\sum_{k=0}^{2^{i-j+1}-1}P^{k2^j}\Big\| = \|(I-P^{2^{i+1}})(I-P^{2^j})^{-1}\| \le$$

$$\le (1+r^{2^{i+1}})/(1-r^{2^j}).$$

From the definition of a, we get

$\|A^{-1}\| = 1/(1-a)$ ;

$\|X_j\| = (1-a^{2^j})/(1-a) = (1-a^{2^j})\,\|A^{-1}\|$;

$(1-a^{2^j}) \le (1-a^{2^{j+1}})$.

Then,

$$E_R(i+1) \le u\, c_2 \frac{1+r^{2^{i+1}}}{1-r^2}(1+a) +$$

$$+ u\,\|A^{-1}\| \sum_{j=1}^{i} \frac{1+r^{2^{i+1}}}{1-r^{2^{j+1}}} \left( c_1 2^j r^{2^j} (1-a^{2^j}) + c_2 (1-a^{2^{j+1}}) \right) =$$

$$= u\,\|A^{-1}\| (1+r^{2^{i+1}}) \left( c_2 \frac{1-a^2}{1-r} + \sum_{j=1}^{i} \frac{1-a^{2^{j+1}}}{1-r^{2^{j+1}}} (c_1 2^j r^{2^j} + c_2) \right).$$

COROLLARY 3.1

Using fan-in algorithm for vector inner product, we get

$$E_E(i) \le u\,\|A^{-1}\| (1+r^{2^i})\, Z(n,r,i), \text{ where}$$

$$Z(n,r,i) = \begin{cases} n^{1/2} + \sum_{j=1}^{i-1} (n \log n\, 2^j r^{2^j} + n^{1/2}), & |a|=r, \\ \frac{n^{1/2}}{1-r^2} + \sum_{j=1}^{i-1} \frac{n \log n\, 2^j r^{2^j} + n^{1/2}}{1-r^{2^{j+1}}}, & |a|<r. \end{cases}$$



The total error $E_T(i)$ is the sum of the analytic error $E_A(i)$ and the roundoff error $E_R(i)$. The best choice for $i$ minimizes $E_T(i)$; namely $E_{OPT} = \min_i \{ E_A(i) + E_R(i) \}$.

Let $e(r,u,n)$ be defined as

$$\min_i ( r^{2^i} + u ( 1 + r^{2^i} ) Z(n,r,i) )$$

and let $I(r,u,n)$ be the value of $i$ for which the minimum is attained. Then

$$E_{OPT} \le e(r,u,n)\, \|A^{-1}\|.$$

The values of $e(r,u,n)$ and $I(r,u,n)$ can be investigated by numerical computation. For the estimated number of iterations we get the empirical formula:

$I(r,u,n) = I'(r,u) - w$, $\quad 0 \le w < 1$, where

$I'(r,u) = \lceil \log\log(1/u) - \log\log(1/r) \rceil$. The validity of this formula has been tested in the interval:

$1 \le n \le 10^8$, $-r \le a \le r$, $10^{-5} \le u \le 10^{-90}$, $10^{-1} \le r \le 1-10^{-10}$.

In this interval $I(r,u,n)$ is independent from $n$ and slightly increases with the number of arithmetic digits. Table I shows the values of the upper bound to $I(r,u,n)$ together with the values of $-\text{LOG LOG}(1/r)$ for various values of $r$ and $u$.

TABLE I

| r | -LOG LOG(1/r) | $u=10^{-5}$ | $u=10^{-20}$ | $u=10^{-50}$ | $u=10^{-75}$ |
|---|---|---|---|---|---|
| $10^{-10}$ | -1.000 | 1 | 1 | 3 | 3 |
| 0.1 | 0.0 | 3 | 5 | 6 | 7 |
| 0.2 | 0.156 | 3 | 5 | 7 | 7 |
| 0.3 | 0.282 | 4 | 6 | 7 | 8 |
| 0.4 | 0.400 | 4 | 6 | 7 | 8 |
| 0.5 | 0.521 | 5 | 7 | 8 | 8 |
| 0.6 | 0.654 | 5 | 7 | 8 | 9 |
| 0.7 | 0.810 | 6 | 8 | 9 | 9 |
| 0.8 | 1.014 | 6 | 8 | 10 | 10 |
| 0.9 | 1.340 | 7 | 9 | 11 | 11 |
| 0.99 | 2.360 | 11 | 13 | 14 | 15 |
| 0.999 | 3.362 | 14 | 16 | 17 | 18 |
| 0.9999 | 4.362 | 17 | 19 | 21 | 21 |
| 0.99999 | 5.362 | 21 | 23 | 24 | 25 |
| 0.999999 | 6.362 | 24 | 26 | 27 | 28 |
| 0.9999999 | 7.362 | 27 | 29 | 31 | 31 |
| 0.99999999 | 8.362 | 31 | 33 | 34 | 35 |
| 0.999999999 | 9.362 | 34 | 36 | 37 | 38 |
| 0.9999999999 | 10.362 | 37 | 39 | 41 | 41 |

The function LOG(e(r,u,n)/u) is an estimate of the number of decimal digits lost in the computed result; fig.1 plots this function versus -LOG LOG(1/r) for $u=10^{-50}$, n=10 and n=1000, with $|a|=r$ and a=0.

## 4. IMPLEMENTATION OF THE METHOD

The implementation of (2.1) for parallel computation is straightforward. A simple condition to terminate the iterations is to check if, in the used floating point arithmetic

$$P_i X_i = X_i .$$

This condition is always reached in a finite number of



FIG. 1

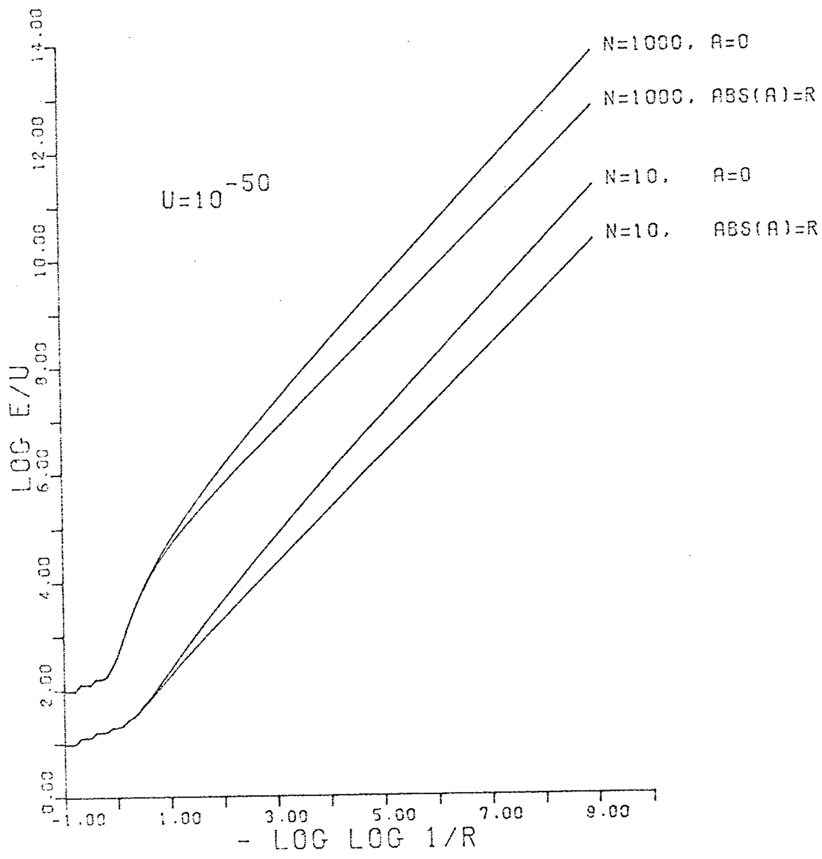


Fig.1. Theoretical Error bounds.

iterations $I''$ if the matrix $P$ is numerically convergent. All the numerical experiments, described in the following section, use this termination condition. The value $I''$ (i.e. the actual number of iterations) has been computed and compared with the estimated value $I'(r,u)$. Table II shows the frequencies of the difference $I''-I'$ for 400 matrices with $n=10$ and $u=16^{-5}$.

TABLE II

| $I''-I'$ | relative frequency |
|---|---|
| -1 | .0100 |
| 0 | .5375 |
| 1 | .4450 |
| 2 | .0075 |

The good agreement between the real and the theoretical number of iterations allows to use the theoretical value $I'(r,u)$ as a measure of the actual number of iterations. Hence, for the parallel complexity of the method, we get

$$\lceil \log\log(1/u) - \log\log(1/r) \rceil \, ( 2 \lceil \log n \rceil + 3 )$$

steps on $n^3$ processors, provided that the value of $e(r,u,n)$ ensures the desired accuracy of the result.

## 5. NUMERICAL EXPERIMENTS

The method has been implemented according to the considerations of section 4, using fan-in algorithm to compute inner products, in order to simulate the numerical behaviour of a parallel implementation.



The method has been tested with several kinds of symmetric positive definite matrices and compared with a standard direct method. Namely, we have used LINPACK routines for factorization and solution of positive definite matrices [2].

Four types of test matrices are considered:

1) Symmetric tridiagonal Toeplitz matrices of the form

$$\begin{matrix} 1 & x & 0 & . & . & 0 \\ x & 1 & x & . & . & 0 \\ 0 & x & 1 & x & . & 0 \\ . & . & x & . & . & . \\ . & . & . & . & . & . \\ 0 & . & . & . & . & x \\ 0 & 0 & . & . & x & 1 \end{matrix} \qquad x > 0, \quad a = r = 1 + 2x\cos(\pi / n+1).$$

2) Symmetric Toeplitz matrices of the form

$$\begin{matrix} 1 & x & x & . & . & x \\ x & 1 & x & . & . & x \\ x & x & 1 & x & . & x \\ . & . & x & . & . & . \\ . & . & . & . & . & . \\ x & . & . & . & 1 & x \\ x & x & . & . & x & 1 \end{matrix} \qquad x > 0, \quad a = x, \quad r = (n-1)\,x.$$

3) Random symmetric definite positive matrices with $a=r$.

4) Random symmetric definite positive matrices with $a<r$.

Fig.2 plots the logarithm to base 10 of the error divided by the computer relative precision versus $-\mathrm{LOG}\,\mathrm{LOG}(1/r)$ in the case $n=10$ and $u=16^{-5}$, together with the corresponding upper bounds.

Fig.3 shows the same function plotted versus $r$ with a different distribution of the instances of $r$.

Fig.4 compares these results with the errors produced by

FIG.2

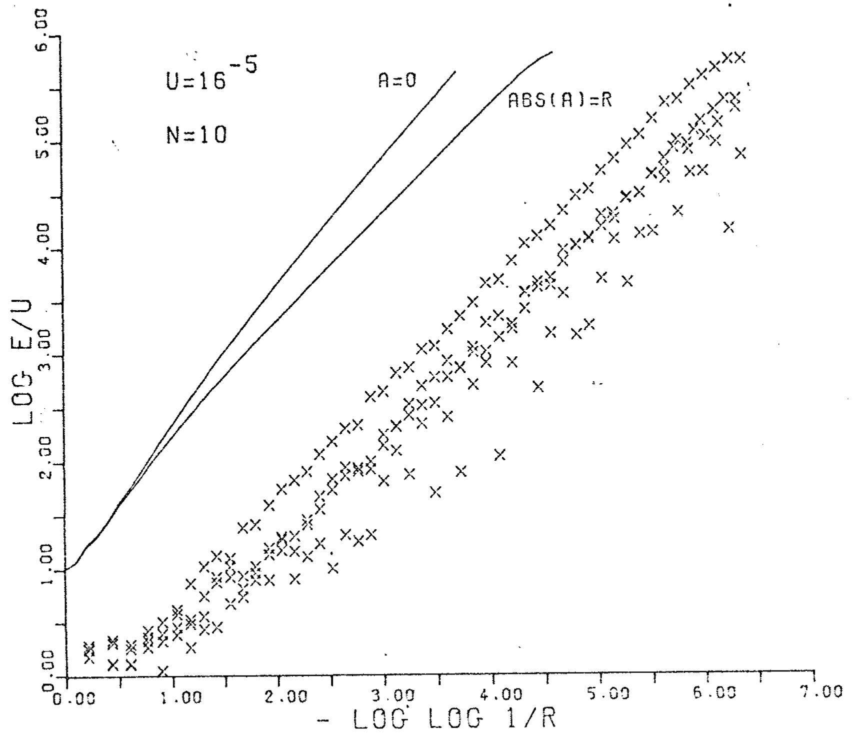


Fig.2. Effective and theoretical errors, plotted versus - LOG LOG(1/r).

FIG.3

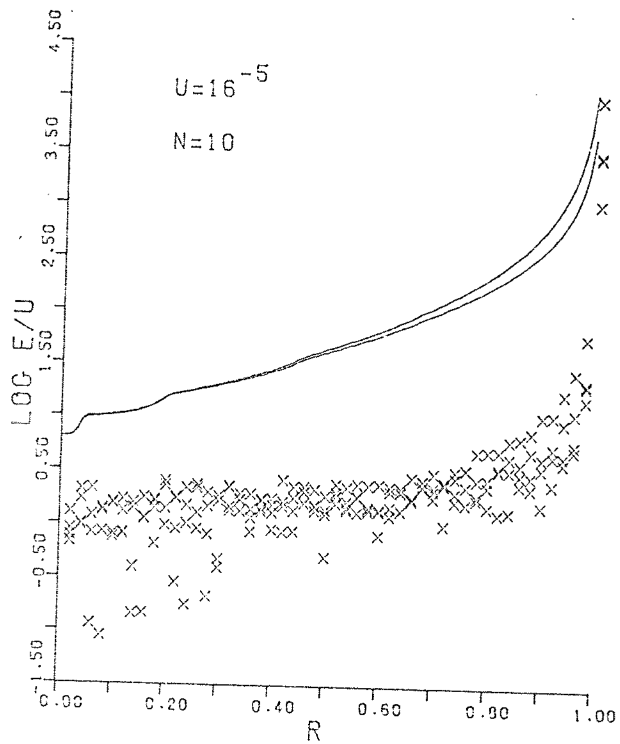


Fig.3. Effective and theoretical errors, plotted versus r.

FIG.4

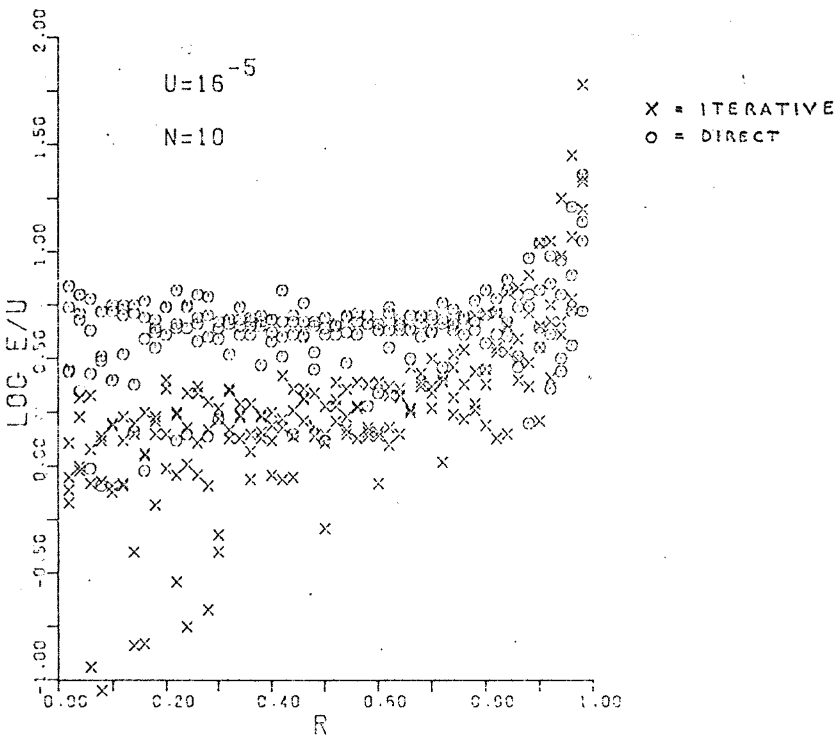


Fig.4. Effective errors of iterative and direct methods.

using LINPACK routines with the same precision.

## 6. ASYMPTOTIC COMPLEXITY

It is interesting to study the asymptotic complexity of the method (2.1) for $n$ .

Let $r \le r_0 < 1$ be the spectral radius of P; our objective is to get a relative error bounded by $e_0$. This can be achieved with the conditions :

(6.1) $\quad e_A \le e_0 / 2$ ,

(6.2) $\quad e_R \le e_0 / 2$ .

From (6.1), it follows

$$r^{2^i} < e_0 / 2 \text{ , that is true if}$$

$$i \le i_0 < \lceil \log \log (1/e_0) - \log \log (1/r_0) \rceil = O(1).$$

Further, the (6.2) is true if

$$e_R \le u\,(1+ r_0^{2^{i_0}})\, Z\,(n,r_0,i_0) \le u\,(1+ e_0 / 2)\, Z\,(n,r_0,i_0) \le$$

$$\le e_0 / 2 \text{ , that is}$$

$$u \le e_0 / ((2+e_0) Z(n,r_0,i_0)).$$

Now, from the COROLLARY 3.1, we get

$$Z(n, r_0, i_0) = O(n \log n),$$

and the (6.2) holds with $1/u = O(n \log n)$, i.e. $O(\log n \log\log n)$ arithmetic digits have to be used.

Let $A(p)$ be the bit complexity of the arithmetic as a function of the number of digits p, then the overall asymptotic complexity of the method is

$$C(\log n\, A(\log n \log\log n)) \quad \text{bit operations.}$$

REMARK

The condition $1/u = O(n \log n)$ derives from the bound to the roundoff error produced by matrix multiplication. Hence, it is common to all the algorithms using matrix multiplications. In particular, we can assume the asymptotic complexity of Csanky's algorithm [1] for parallel inversion of matrices to be

$$C(\log^2 n\, A(\log n \log\log n)).$$



REFERENCES

[1] L. CSANKY, Fast Parallel Matrix Inversion Algorithms. SIAM J. Comput., 5(1976) pp.618-623.

[2] J.J. DONGARRA et al., LINPACK Users's Guide. SIAM, Philadelphia, 1979.

[3] R.S. VARGA, Matrix Iterative Analysis. Prentice Hall, Englewood Cliffs, N.J., 1962.

[4] J.H. WILKINSON, Rounding Errors in Algebraic Processes. Englewood Cliffs, N.J., 1963.

[5] H. WOZNIAKOWSKI, Round-Off Error Analysis of Iterations for Large Linear Systems. Numer. Math. 30(1978) pp. 301-314.