STUDIA UNIV. BABEŞ-BOLYAI, MATHEMATICA, XXXVI, 3, 1991

ON SOME PARALLEL METHODS IN LINEAR ALGEBRA

GH. COMAN*

Received : March 2, 1991
AMS subject classification: 65FXX, 68Q10

REZUMAT. - Asupra unor metode paralele în algebra liniară. Sînt studiate din punct de vedere al complexităţii mai multe metode numerice de inversare a matricelor şi de rezolvare a sistemelor algebrice liniare.

The parallel computation had became an actual problem in many application fields.

Of course, not each mathematical method can be efficientely projected in a parallel version.

To characterize the depth of the parallelism of a given method there exists specifically criterions. Such criterions are the speed and the efficiency. The goal of this paper is to discus some methods in linear algebra from the parallelism point of view.

Let X be a linear space, X_0 a subset of X, $(Y, \|.\|)$ a normed linear space and $S, S: X_0 \to Y$, a given operator. The problem: for given $\varepsilon>0$ and $x \in X_0$ find an $y \in Y$ such that $\|S(x) - y\| \leq \varepsilon$ is called a S - problem, x is the problem element, S is the solution operator and $s = S(x)$ is the solution element. $\tilde{s} \in Y$ for which $\| \tilde{s} - s\| \leq \varepsilon$ is called an ε - approximation of the solution s.

In order to solve a S - problem there are necessary some informations on the problem element x. So, let Z be a set (the set of informations). The operator $\mathfrak{I}: X \to Z$ is called the informational operator and $\mathfrak{I}(x)$, $x \in X_0$, is the information on

* University of Cluj-Napoca, Faculty of Mathematics, 3400 Cluj-Napoca, Romania

x. To compute a solution of a S - problem for a given information $\mathfrak{I}(x)$ we need an algorithm, which is defined as an application α: $\mathfrak{I}(X_0)\to Y$. So, for a given $x \in X_0$, $\alpha(\mathfrak{I}(x))$ is the approximation of the solution $S(x)$ given by the algorithm α with the information $\mathfrak{I}(x)$ as the input data. If α $(\mathfrak{I}(x))$ is an ε - approximation of $S(x)$ then $\mathfrak{I}$ and α are called ε - admissible. So, to solve a S - problem means to find an ε - admissible informational operator and an ε - admissible algorithm for it.

DEFINITION 1. A couple $(\mathfrak{I},\alpha)$ with $\mathfrak{I}:X\to Z$ and α : $\mathfrak{I}$ $(X_0)\to Y$ is called a method associated to a S - problem.

If $\mathfrak{I}$ and α are ε - admissible then the corresponding method is called also ε - admissible.

Next, one denotes by $M(S)$ the set of all admissible methods for the problem $\dot{S}$. A method $\mu\in M(S)$, $\mu = (\mathfrak{I},\alpha)$, is called a serial method if all the computations are described as a single instructions stream (α is a serial algorithm). If the computations are described as a multiple instructions streams then μ is called a parallel method (α is a parallel algorithm).

To distinguish the two kind of methods one denotes by $M_s(S)$ the set of all serial methods for the problem S and by $M_p(S)$ the set of all parallel methods for S.

For a method $\mu\in M(S)$ one denotes by $CP(\mu;x)$, $x \in X_0$, its computational complexity for the element x or the local complexity, while

$$CP(\mu) = \sup_{x\in X_0} CP(\mu;x)$$

is the complexity of the method μ for the problem S(global complexity) [3].

DEFINITION 2. The method $\overline{\mu} \in M_s$ (S) for which

$$CP(\overline{\mu}) = \inf_{\mu \in M(S)} CP(\mu)$$

is called the optimal method with regard to the complexity.

Now, let μ be a serial method, $\mu \in M_s$ (S).

Generally speaking, by a parallel method $\mu_p \in M_p$ (S), associated to μ we understand a method in which all the operations, independent to each others, are performed in parallel (in the same time). So, we can image the serial method divided in many parts (segments - streams of instructions) independently or partial independently from the computation point of view, say $\mu_1,\ldots,\mu_r$. Then

$$CP(\mu_p) = \max_{1\le i\le r} CP(\mu_i)$$

is the complexity of the corresponding parallel method μ_p.

DEFINITION 3. Let S be a given problem, $\mu_p \in M_p(S)$ a parallel method and $\overline{\mu_s} \in M_s$ (S) the optimal serial method with regard to the complexity.

Then

$$S(\mu_p) = \frac{CP(\overline{\mu}_s)}{CP(\mu_p)}$$

is called the speed of the parallel method μ_p.

Remark 1. The speed is also denoted by $S(\mu_p;r)$, where r is the number of the instructions streams of the method μ_p.

Obviously, $S(\mu_p; r) \le r$.

Remark 2. A more practical value to judge the parallel version μ_p of a serial method μ_s is

$$s(\mu_p;r) = \frac{CP(\mu_s)}{CP(\mu_r)}.$$

We also have $s(\mu_p;r) \geq S(\mu_p;r)$.

DEFINITION 4. The value

$$E(\mu_p) = \frac{S(\mu_p;r)}{r}$$

is called the efficiency of the parallel method μ_p.

As $0 \leq S(\mu_p;r) \leq r$ it follows that $0 \leq E(\mu_p) \leq 1$.

Next, we consider first some examples.

E.1. Let $\mathcal{E}$ be the following expression :

$\mathcal{E} = t_1 \ \rho \ t_2 \ \rho \ \dots \ \rho \ t_n$

where ρ is an associative operation.

The serial computational complexity of $\mathcal{E}$ is

$CP(\mathcal{E}) = (n - 1) \ CP(\rho)$,

where $CP(\rho)$, is the complexity of the operation ρ .

A parallel version $\mathcal{E}_p$ of the expression $\mathcal{E}$ is obtained as follows: in the first step we compute, say $t_i^1 := t_{2i-1}\rho \ t_{2i}$, for all possible i. To do it more clear, let $m \in \mathrm{N}$ be such that $2^{m-1} <$ $< n \leq 2^m$. If $n < 2^m$ then we supplement the expression $\mathcal{E}$ by

$t_{n+1} = \dots = t_{2m} = 0$, i.e.

$\mathcal{E} = t_1 \ \rho \ t_2 \ \rho \ \dots \ \rho t_n \ \rho \ t_{n+1} \ \rho \ \dots \ \rho \ t_{2^m}$

so,

$t_i^1 \ := t_{2i-1} \ \rho \ t_{2i}$, $i = 1,\dots,2^{m-1}$.

In the second step we have

$t_i^2 := t_{2i-1}^1 \ \rho \ t_{2i}^1$, $\qquad i = 1,\dots,2^{m-2}$

and so on

$$t_i^k := t_{2i-1}^{k-1} \rho \, t_{2i}^{k-1}, \qquad i = 1,\dots,2^{m-k}$$

for $k = 3,\dots,m$. Finally, we have $\mathscr{E} = t_1^m$. Hence, the necessary steps to compute $\mathscr{E}$ is m. Taking into account that $2^{m-1} < n \le 2^m$, one obtains $m = [\log_2 n]$, where $[x]$, $x \in \mathbb{R}$ is the integer with the property $x \le [x] < x + 1$.

It follows that

$CP(\mathscr{E}_p) = [\log_2 n] \, CP(\rho)$.

So, we have

$$s(\mathscr{E}_p; [n/2]) = \frac{n-1}{[\log_2 n]}$$

and

$$E(\mathscr{E}_p) = \frac{n-1}{[n/2]\,[\log_2 n]} \approx \frac{2}{[\log_2 n]}$$

where $[x]$ is the integer part of x.

Remark 3. If we consider the binary tree associated to the expression $\mathscr{E}$ then the complexity of the parallel computation of $\mathscr{E}$ is the depth of the tree [5].

E.2. Let be $X = M_n(\mathbb{R})$, $X_0 = X$, $Y = \mathbb{R}$ and $S : X \to Y$, $A \to \det A$. Hence, S is the problem to compute the determinant $\det A$ of the matrix A. The method used consists in the transformation of the determinant

$$\det A = \begin{vmatrix} a_{11} & a_{12}\dots & a_{1n} \\ a_{21} & a_{22}\dots & a_{2n} \\ \dots & \dots & \dots \\ a_{n1} & a_{n2}\dots & a_{nn} \end{vmatrix} .$$

in the form

$$\det A := a_{11}^1 * \ldots * a_{nn}^n * \begin{vmatrix} 1 & a_{11}^2 & \ldots & a_{1n}^2 \\ 0 & 1 & \ldots & a_{2n}^3 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & . & 1 \end{vmatrix}$$

using the operations :

$$a_{ij}^1 := a_{ij} \ , \ i,j = 1,\ldots,n$$

$$a_{ip}^{p+1} := \frac{a_{ip}^p}{a_{pp}^p} \ , \ i = p+1,\ldots,n$$

$$a_{ij}^{p+1} := a_{ij}^p - a_{pj}^p * a_{ip}^{p+1} \ , \quad i,j = p+1,\ldots,n$$

for $p = 1,\ldots,n-1$.

So, we have

$$\det A = a_{11}^1 * a_{22}^2 * \ldots * a_{nn}^n \ .$$

Remark 4. Next we suppose that $CP(+) = 1$ (a unit time) and $CP(*) = CP(/) = 3$.

If one denotes by μ_s the serial method to compute det A, one obtains

$$CP(\mu_s) = \frac{1}{6}(n-1)(8n^2+5n+18) \ .$$

A parallel version of the considered method using n parallel instructions strems (n processors) is :

begin

det A: = 1;

for p: = 1 step 1 until n -1 do

begin

(det A: = det A * a_{pp}^p; $(p + 1 \le j \le n)$ $a_{pj}^{p+1} := \frac{a_{pj}^p}{a_{pp}^p}$) ;

((p+1 $\le j \le n$) for i:=p+1 step 1 until n do

$$a_{ij}^{p+1} := a_{ij}^{p} - a_{ip}^{p} * a_{pj}^{p+1})$$

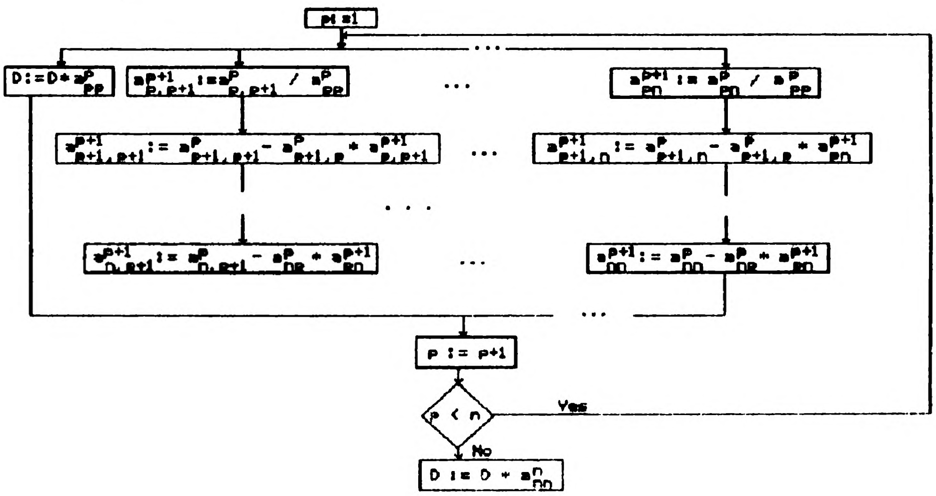
end

det A := **det** $A * a_{nn}^{n}$

end

Remark 5. $(I, (1 \leq k \leq m)\ I_k)$ means that the instructions $I, I_1, \ldots, I_m$ are performed in parallel.

For a better illustration of the parallel method, say μ_p, we give the next diagram ($D :=$ det A) :



The complexity of the parallel method μ_p, as it can be easy seen, is:

$CP(\mu_p; n) = n(2n - 1)$.

So,

$$s(\mu_p; n) = \frac{(n-1)(8n^2 + 5n + 18)}{/6n(2n - 1)} \approx \frac{2}{3} n + \frac{1}{12}$$

and

$$E(\mu_p) = \frac{2}{3} .$$

E.3. For $X = M_n\ (\mathbb{R})$, $X_0 = \{A \mid \det A \neq 0,\ A \in X\}$, $Y = M_n\ (\mathbb{R})$ and $S(A) = A^{-1}$, S is the problem to compute the inverse of a matrix.

We use the method based on the succesive transformations of the matrix $[A \mid I_n]$ in the matrix $[I_n \mid A]$, where I_n is the unit matrix of order n. The transformations are : first one denotes the elements of the matrix $[A \mid I_n]$ by t^1_{ij}, $i=1,\dots,n$; $j=1,\dots,2n$. Now,

$$t^{p+1}_{pj} := \frac{t^p_{pj}}{t^p_{pp}}\ ,\ j=p+1,\dots,2n$$

$$t^{p+1}_{ij} := t^p_{ij} - t^p_{ip} * t^{p+1}_{pj}\ ,\ i=1,\dots,n,\ i \neq p\ ;\ j=p+1,\dots,2n$$

$$t^n_{nj} := \frac{t^n_{nj}}{t^n_{nn}}\ ,\ j=n+1,\dots,2n\ ,$$

for all $p = 1,\dots,n-1$.

So,

$$A^{-1} = (t^n_{ij})\ i = \overline{1,n}\ ;\quad j = \overline{n+1\ ,\ 2n}$$

If μ_s is the corresponding serial method then

$$CP(\mu_s) = \frac{3}{2}\, n\,(4n^2 - 5n + 3) .$$

A parallel method, μ_p, can be projected as follows :

begin

$$t^2_{12} := \frac{t^1_{12}}{t^1_{11}}\ ;$$

for $p := 1$ step 1 until $n - 1$ do

begin for $j := p+1$ step 1 until $2n$ do

$$\left(t_{p,j+1}^{p+1} := \frac{t_{p,j+1}^{p}}{t_{pp}^{p}} \; ; \; (1\le i\le n,\ i\ne p)\ t_{ij}^{p+1} := t_{ij}^{p} - t_{ip}^{p} * t_{pj}^{p+1}\right)$$

end;

$$\left((n+1\le j\le 2n)\quad t_{nj}^{n} := \frac{t_{nj}^{n}}{t_{nn}^{n}}\right)$$

end

We have

$$CP(\mu_p;n) = 6(n^2 - n + 1)$$

and

$$s(\mu_p;n) \approx n - \frac{1}{4}$$

respectively

$$E(\mu_p) \approx 1.$$

Remark 6. From these three examples we can see that the matrix inversion permites a very good parallelism ($E(\mu_p) \approx 1$), while for the determinant computation $E(\mu_p) \approx 2/3$ and in the first example

$$E(\mathscr{E}_p) \approx 2/[\log_2 n] .$$

Linear algebraic systems.

If $X = \{[A|b] \mid A\in M_n(\mathbb{R}),\ b\in M_{n,1}(\mathbb{R})\}$, $X_0=\{[A|b]\in X \mid \det A\ne 0\}$ $S([A|b])=A^{-1}b$ then S is the problem to solve the system $As=b$.

Next, there are discused serial and parallel versions for some well known numerical methods for the solution of linear algebraic systems.

I. **Cramer's method.** Taking into account that the solution is given by $s_i = D_i/D, i=1,\dots,n$, where D=det A and D_i is the determinant obtained by D changing the i-th column vector by b.

So, we have to compute $n+1$ determinants of order n, with the complexity $CP(\mu_s)$ from the example E_2, and n divisions. It follows that the serial complexity of Cramer's method μ_s^c is $CP(\mu_s^c) = (n+1)CP(\mu_s) + nCP(/)$, i.e.

$$CP(\mu_s^c) = \frac{1}{6}(8n^4 + 5n^3 + 1on^2 + 13n - 18). \tag{1}$$

A natural parallel method here is to compute in parallel the $(n+1)$ determinants and than to perform the n divisions. So,

$$CP(\mu_p^c) = \frac{1}{6}(8n^3 - 3n^2 + 13n) \tag{2}$$

where μ_p^c is the mentioned parallel method.

Hence, one obtains

$$s(\mu_p^c; n+1) = (n+1) - \frac{18}{8n^3-3n^2+13n} \approx n+1$$

and

$$E(\mu_p^c) \approx 1. \tag{3}$$

As a conclusion we can remark the very good parallelism of Cramer's method $\left(E(\mu_p^c) \approx 1\right)$.

II. **Gaussian elimination method.** As, it is well known first the given matrix $[A|b] \in X_0$ is transformed in the matrix $[T_n \mid \bar{b}]$, where T_n is an upper triungular matrix ($T_n = (a_{ij}^i)$ $i=1,\ldots,n;\ j=i+1,\ldots,n;\ a_{ii}^i=1$) using the relations

$$a_{pj}^p := a_{pj}^p \,/\, a_{pp}^p,\ j=p+1,\dots,n;\ b_p^p \,/\, a_{pp}^p$$

$$a_{ij}^{p+1} := a_{ij}^p - a_{ip}^p * a_{pj}^p\ ,\ i,j=p+1,\dots,n$$

$$b_i^{p+1} := b_i^p - a_{ip}^p * b_p^p\quad ,\ i = p+1,\dots,n$$

for $p = 1,\dots,n-1$, and $b_n^n := \dfrac{b_n^n}{a_{nn}^n}$, where for the begining $a_{ij}^1 := a_{ij}$, $b_i^1 := b_i$, $i,j=1,\dots,n$.

The complexity of this computation is $n(n^2-1)/3*[CP(+)+CP(*)]$ + + $n(n+1)/2 * CP(/)$. Now the triangular system $T_n s = b$ is solved by back substitution method:

$$s_n := b_n^n$$

$$s_i := b_i^i - \sum_{j=i+1}^{n} a_{ij}^i * x_j,\quad i=n-1,\dots,1,$$

with the computational complexity $n(n-1)/2 * [CP(+) + CP(*)]$. It follows that

$$CP(\mu_s^G) = \frac{1}{6}\ (8n^3 + 21n^2 - 11n)\,. \tag{4}$$

A parallel version μ_p^G of the Gauss method is :

begin

for p:=1 step 1 until n-1 do

begin

$$\left((p+1\le j\le n+1)\quad a_{ij}^p := \frac{a_{pj}^p}{a_{pp}^p}\right);$$

for i:=p+1 step 1 until n do

begin $\left((p+1\le j\le n+1)\quad a_{ij}^{p+1} := a_{ij}^p - a_{ip}^p * a_{pj}^p\right)$; $b_i^{p+1} := b_i^p - a_{ip}^p * b_p^p$ end

end;

$a^n_{n,n+1} := \dfrac{a^n_{n,n+1}}{a^n_{nn}}$

for k:=1 step 1 until n - 1 do

$\big((k \le i \le n-1) \quad a^n_{n-i,n+1} := a^n_{n-i,n+1} - a_{n-i,n-k+1} * a^n_{n-k+1,n+1}\big)$

end

where $a^p_{p,n+1} = b^p_p$.

So, $s_i := a^n_{i,n+1}$, $i=1,\dots,n$.

It follows that

$$CP(\mu^G_p;n) = 2n^2 + 5n - 11 \tag{5}$$

and

$$S(\mu^G_p;n) \approx \frac{2}{3}\,n + \frac{1}{2}$$

respectively

$$E(\mu^G_p) \approx \frac{2}{3}\,. \tag{6}$$

III. **Total elimination method.** The matrix $[A|b] \in X_0$ is transformed in the matrix $[\,I_n \mid b^n]$.

First,

$a^1_{ij} := a_{ij}$, $a^1_{i,n+1} = b_i$, $i,j=1,\dots,n$.

Now, one applies the succesive transformations

$a^{p+1}_{pj} := \dfrac{a^p_{pj}}{a^p_{pp}}$, $j=p+1,\dots,n+1$;

$a^{p+1}_{ij} := a^p_{ij} - a^p_{ip} * a^{p+1}_{pj}$; $i=1,\dots,n$; $i \ne p$; $j=p+1,\dots,n+1$

for all $p = 1,\dots,n$.

So, the solution is $s_i := a^{n+1}_{i,n+1}$, $i=1,\dots,n$.

The computational complexity of this method in the serial version (μ^T_s) is

$$CP(\mu_s^T) = \frac{1}{2}(4n^3 + 3n^2 - n). \tag{7}$$

As a parallel version (μ_p^T) of the total elimination method is the following :

begin

$a_{12}^2 := \dfrac{a_{12}^1}{a_{11}^1}$;

for p:=1 step 1 until $n-1$ do

begin

for j:=p+1 step 1 until n do

$$\left(a_{p,j+1}^{p+1} := \frac{a_{p,j+1}^p}{a_{pp}^p}; \quad (1 \le i \le n \quad i \ne p) \quad a_{ij}^{p+1} : a_{ij}^p - a_{ip}^p * a_{pj}^{p+1} \right)$$

$$\left(a_{p+1,p+2}^{p+2} := \frac{a_{p+1,p+2}^{p+1}}{a_{p+1,p+2}^{p+1}}; \quad (1 \le i \le n, \ i \ne p) \quad a_{i,n+1}^{p+1} := a_{i,n+1}^p - a_{ip}^p * a_{p,n+1}^{p+1} \right)$$

end

$$\left(a_{n,n+1}^{n+1} := \frac{a_{n,n+1}^n}{a_{nn}^n}; \quad (1 \le i \le n-1) \quad a_{i,n+1}^{n+1} := a_{i,n+1}^n - a_{in}^m * a_{n,n+1}^{n+1} \right)$$

end

We have

$$CP(\mu_p^T) = 2n^2 + 2n + 3. \tag{8}$$

So,

$$s(\mu_p^T; n) \approx n - \frac{1}{4}$$

and

$$E(\mu_p^T) \approx 1. \tag{9}$$

IV. **Iterative methods.** One considers two iterative methods.

IV.1. **Jacobi iteration.** For a given $x^{(0)} = (x_1^{(0)}, \ldots, x_n^{(0)})^T$, the sequence of the succesive approximation $x^{(m+1)}$ is given by

$$x_i^{(m+1)} = \frac{1}{a_{ii}} (b_i - \sum_{\substack{j=1 \\ j \neq i}} a_{ij} x_j^{(m)}), \; i = 1,\ldots,n.$$

If $CPI(\mu_s^J)$ is the computational complexity of one iteration then the serial complexity of the Jacobi method is

$$CP(\mu_s^J) = m_J(\varepsilon) \quad CPI(\mu_s^J),$$

where $m_J\ (\varepsilon)$ is the iterations number for which $x^{(m_J(\varepsilon))}$ is an ε-approximation of the solution. So, we have

$$CP(\mu_s^J) = (4n^2 - n)\ m_J(\varepsilon)\ . \tag{10}$$

A parallel version of the method μ_s^J is to compute, in parallel, each $x_i^{(m+1)}$, $i=1,\ldots,n$.

Hence,

$$CP(\mu_p^J\ ;n) = (4n-1)\ m_J\ (\varepsilon)\ . \tag{11}$$

It follows that

$$S(\mu_p^J; n) = n$$

and

$$E\left(\mu_p^J\right) = 1$$

IV.2. **Gauss - Siedel iteration.** Starting with $x^{(0)}$, the iterations are given by

$$x_i^{(m+1)} = \frac{1}{a_{ii}} (b_i - \sum_{j=1}^{i-1} a_{ij} x^{(m+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(m)}), i=1,\ldots,n.$$

The serial complexity of the Gauss-Siedel method is

$$CP(\mu_s^{GS}) = (4n^2 - n)\ m_{GS}(\varepsilon)\ , \tag{13}$$

where $m_{GS}(\varepsilon)$ is the iterations number.

It is obviously that the parallelism of the Gauss-Siedel method is more less than of the Jacobi iteration. Certainly we solve for $x_2^{(m+1)}$ using already the "new" value $x_1^{(m+1)}$, for $x_3^{(m+1)}$ it is used the "new" values $x_1^{(m+1)}$, $x_2^{(m+1)}$ and so on. Hence, $x_2^{(m+1)}$ can be computed only when the computation of $x_1^{(m+1)}$ is finished and the computation of $x_3^{(m+1)}$ must wait for $x_1^{(m+1)}$ and $x_2^{(m+1)}$ and so on. It follows that a parallel version μ_p^{GS} is to do the computation begining with the first line ($x_1^{(m+1)}$) than the second one ($x_2^{(m+1)}$) and so on. One obtains

$$CP(\mu_p^{GS}) = n([\log_2 n] + 6)\, m_{GS}(\varepsilon)$$

and

$$E(\mu_p^{GS}) = \frac{4(1 - 1/n)}{[\log_2 n] + 6} .$$

Conclusions. Taking into account the serial and parallel complexity of the above methods for linear algebraic systems it follows:

PROPOSITION 1. $CP(\mu_s^{G}) < CP(\mu_S^{T}) < CP(\mu_s^{C})$, $\forall n > 2$.

The proof follows directly by (1), (4) and (7).

Remark 7. Of the Gauss - Siedel procedure may be viewed as an acceleration of Jacobi method, so we generally have $m_{GS}(\varepsilon) \leq m_J(\varepsilon)$ i.e.

$$CP(\mu_s^{GS}) < CP(\mu_s^{J}) .$$

Now, from (2) and (10), it follows :

PROPOSITION 2. *If* $m_{GS}(\varepsilon) \leq [n/3]$ *then*

$$CP(\mu_s^{GS}) < CP(\mu_s^{G}) .$$

Remark 8. For the systems with a lagre number of equation (such that $[(n/3) + 1]$ iterations are sufficient to get a good approximation the Gauss - Siedel iteration is better than of the Gauss elimination method.

The following two propositions give some informations regarding with the parallel methods.

PROPOSITION 3. $CP(\mu_p^T) < CP(\mu_p^G) < CP(\mu_p^C) \quad \forall n > 2$.

The proof is based on the relations (2), (5) and (8).

Remark 9. For the parallel version μ_p^G and μ_p^T we have $CP(\mu_p^G) > CP(\mu_p^T)$ just if in the serial case the relation is $CP(\mu_s^G) < CP(\mu_s^T)$. So, generally a good serial method does not conduct to a good parallel version.

PROPOSITION 4. *If* $m_T(\varepsilon) < [n/2]$ *then* $CP(\mu_p^T) < CP(\mu_p^T)$.

Remark 10. In the parallel case it can be done just $[n/2]$ iterations without passing the complexity of the best parallel method μ_p^T

Finally, from (3), (6), (9) and (12) it follows that the best parallelism is possessed by the Jacobi iteration method ($E(\mu_p^T)=1$). Also, a good parallelism has the total elimination method $(E(\mu_p^T) \approx 1 - \frac{1}{4n})$ and the Cramer's method $(E(\mu_p^C) \approx 1)$. But the complexity of the Cramer method is, in both serial and parallel versions, a polynomial function on degree with a unity greater than the other ones. So, the Cramer's method is never recommended from the computational complexity point of view.

REFERENCES

1. Blum E.K., *Numerical Analysis and computation.* Theory and practice, Addison - Wesley Publishing Company, 1972.
2. Coman Gh., *On the parallel complexity of some numerical algorithms for solving linear systems.* "Babeş-Bolyai" University, Cluj-Napoca Research Seminars, Preprint Nr. 6, 7-16, 1987.

3. Coman Gh., Johnson D., *The complexity of algorithms*. "Babeş-Bolyai" University, Cluj-Napoca, 1987.
4. Fadeeva V.N., Fadeev D.K., *Parallel computation in linear algebra*. Kibernetica, 6, 28-40, 1977.
5. Heller D., *A survey of parallel algorithms in numerical linear algebra*. SIAM Rev. 20, 740-777, 1978.
6. Solodovnikov V.I., *Upper bounds on complexity of solving systems of linear equations*. Zap.naucin.seminars (LOMI) , 159-187, 1982.
7. Stone H.S., *An efficient parallel algorithm for the solution of a tridiagonal linear system of equations*. J.ACM, 20, 27-38, 1973.
8. Traub J.F., Wozniakowski H., *A General Theory of Optimal Algorithms*, Academy Press, 1980.