Theorem 2. $\max \{f_{G_p}(x)\,;\ x_{vi} \in [0, 1],\ v \in V,\ 1 \leqslant i \leqslant p\}$ *is* $\lambda(G, p)\,(\lambda(G, p)-1)/2$ *and all optimal solutions have integer components. If* x *is an optimal solution then the set of vertices* $v \in V$, *such that there exists* i_v *with* $x_{vi_v}=1$, *induces a p-colourable subgraph of maximum order in* G.

Another approach of the vertex colouring problem is to introduce the parameter $\mu(G, p)$ defined as the minimum cardinality of a set A of edges with property that if we remove A from E we obtain a p-colourable graph. Clearly, G has a p-colouring iff $\mu(G, p)=0$.

Let $g_G : [0, 1]^{np} \to R$ be the Motzkin-Strauss function of the graph G_p:

$$g_G(x) = \sum_{i=1,p} \sum_{vw \in E} x_{vi} x_{wi} + \sum_{v \in V} \sum_{i \neq j} x_{vi} x_{vj}.$$

Theorem 3. *The optimal value for the quadratic program*

$$\text{(2)} \qquad \min g_G(x)$$

$$0 \leqslant x_{vi} \leqslant 1,\ \sum_{i=1,p} x_{vi} = 1, \text{ for all } v \in V \text{ and } i=\overline{1, p}$$

is $\mu(G, p)$ *and each optimal solution has integer components.*

Proof. Observe that, if x is a feasible solution then g_G can be written:

$$g_G(x) = \sum_{i=1,p} \sum_{vw \in E} x_{vi} x_{wi} + \sum_{v \in V} \Big(1 - \sum_{i=1,p} x_{vi}^2\Big).$$

Let x^0 be an optimal solution of (2) and suppose that it has a non integer component, say x_{vi}^0. Since $\sum_{k=1,p} x_{vk}^0 = 1$, there exists j, $j \neq i$ and $0 < x_{vj}^0 < 1$. Let us consider $\bar{x}^0$ the vector with components:

$$\bar{x}_{wk}^0 = \begin{cases} x_{vi}^0 - y & \text{if } w=v \text{ and } k=i \\ x_{vj}^0 + y & \text{if } w=v \text{ and } k=j \\ x_{wk}^0 & \text{otherwise} \end{cases}$$

where $-x_{vj}^0 \leqslant y \leqslant x_{vi}^0$. Obviously, $\bar{x}^0$ is a feasible solution and we have

$$g_G(x^0) - g_G(\bar{x}^0) = 2y^2 + y\left(\frac{\partial g_G}{\partial x_{vi}}(x^0) - \frac{\partial g_G}{\partial x_{vj}}(x^0)\right).$$

It is clear that y can be chosen such that $g_G(x^0) > g_G(\bar{x}^0)$ which contradicts that x^0 is an optimal solution. Therefore, all optimal solutions have integer components. Moreover, if x^0 is such a solution, and A is the set of edges vw so that there exists i with $x_{vi}^0 = x_{wi}^0 = 1$, then $G-A$ has a p-colouring given by $c(v)=i$, and $g_G(x^0)=|A|$. On the other hand, if A is a set of edges such that $G-A$ has a p-colouring $c : V \to \{1, 2, \ldots, p\}$ then, taking $x_{vc(v)}^A = 1$ for all $v \in V$ and $x_{vi}^A = 0$ otherwise, we obtain a feasible solution x^A for (2) with $g_G(x^A) \leqslant |A|$. This proves that the minimum value of (2) is $\mu(G, p)$.

REFERENCES

1. Motzkin T. S., Strauss E. G. — *Maxima for graphs and a new proof of a theorem of Turan*, Canad. Journal of Math. vol. 17, p. 533–540, 1965

Received 1.VI. 88

Faculty of Mathematics
University of Iași
6600 — Iași, România

ANALELE ȘTIINȚIFICE ALE UNIVERSITĂȚII „AL. I. CUZA"
Tomul XXXIV, s. I a. Matematică-Informatică, 1988, f. 4

SYSTOLIC NETWORKS FOR NUMERICAL DIFFERENTIATION BY USING RICHARDSON-ROMBERG EXTRAPOLATION METHOD

BY

OCTAV BRUDARU

1. Introduction. Systolic arrays processors offer an effective way to use a large number of processors in parallel in order to obtain a high computation rate, low parts count, modularity, simplicity and small size [5]—[8]. In recent years many systolic array algorithms have been proposed in order to solve problems in the field of numerical analysis [8].

In [3] they are presented some systolic arrays for estimating derivatives based on the Lagrange polynomial interpolation with unequally-spaced arguments.

This paper presents linear and two-dimensional systolic arrays implementing the Richardson-Romberg extrapolation method applied to central and one-sided divided differences in order to estimate the derivatives of many functions choosen as in [4].

In Section 2 we tre at the case of central divided differences and propose a linear systolic array and a two-dimensional systolic array. We prove the correctness of the working of these arrays and analyse their performances.

In Section 3, we discuss similar results for the case of one-sided divided differences.

2. The case of central divided differences. As described in [4], suppose that $f : D \to R$, $D \subset R$. Given a positive integer n and a step size h_1, they are computed

$$\text{(1)} \qquad h_k = (n-k+1)\, h_1/n, \quad k=2, \ldots, n,$$

and the central divided differences

$$\text{(2)} \qquad T_{0,k} = [f(x+h_k) - f(x-h_k)]\, c_k, \quad k=1, \ldots, n.$$

where $c_k = 1/(2h_k)$, $k=1, \ldots, n$.

The clock tick (CT) is defined as the time to perform a division or both a multiplication and an addition. The time is denoted by t which is a count of the number of CTs. We shall suppose that each processor is active during each pulse number. The processors used to compute (2) are given in Figure 1 (a)—(b).

They work so that: (a) $y(t+1) = -x(t)$ and (b) $z(t+1) = (x(t) + y(t))\,u(t)$. The cell CDD to compute (2) is depicted in Figure 1 (c). If L is

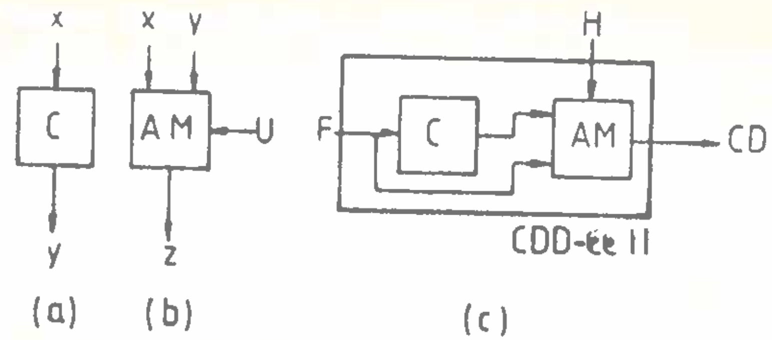


Fig. 1 — Processing elements to compute the central divided differences

the label of an input (output) of a processor, then $L(t)$ designates the value circulating through this one during the t-th pulse number.

Proposition 1. *If* $F(t+2k-1)=f(x-h_k)$, $F(t+2k)=f(x+h_k)$ *and* $H(t+2k)=c_k$, *then* $CD(t+2k+1)=T_{0,k}$, $k=1, \ldots, n$.

Proof. If $k\in\{1, \ldots, n\}$ and $t\geq 0$, at the time $t+2k-1$, $f(x-h_k)$ enters F and after 1 CT the AM-processor receives $-f(x-h_k)$, c_k and $f(x+h_k)$, because $H(t+2k)=c_k$ and $F(t+2k)=f(x+h_k)$. Thus, $CD(t+2k+1)=T_{0,k}$.

Eventually, c_k's could be computed inside of CDD by multiplying $1/h_1$ by $g_k=n/(2(n-k+1))$.

2.1. A linear systolic array. Starting from $T_{0,k}$, $k=1, \ldots, n$, the extrapolated values are given by the following formula:

$$T_{m,k}=T_{m-1,k+1}+(T_{m-1,k+1}-T_{m-1,k})\, b_{k+m,m} \tag{3}$$

where $k=2, \ldots, n-m$, $m=1, \ldots, n-1$ and $b_{ij}=(n-i-1)/[j(2n-2i-2+j)]$.

The linear systolic array to compute (2) uses the processors depicted in Figure 2 (a)—(b), where (a) $y(t+1)=x(t)$ and (b) $z_{out}(t+1)=z_{in}(t)+x(t)\,y(t)$ and the RL-cell given in Figure 2 (c).

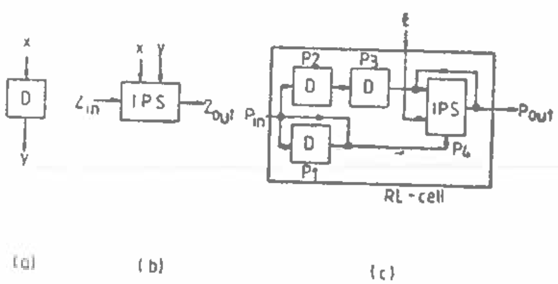


Fig. 2 — Processing elements to compute the extrapolated values

Further, we denote by $X(t)=\varnothing$ the fact that no value circulates through X during the t-th CT.

Proposition 2. *If* $P_{in}(t)\neq\varnothing$, $P_{in}(t+2)\neq\varnothing$ *and* $P_{in}(t+1)=P_{in}(t+3)=\varnothing$ *then* $P_{out}(t+4)=P_{in}(t+2)+E(t+3)\,[P_{in}(t+2)+E(t+2)\,P_{in}(t)]$, $t\geq 0$.

Proof. If $a=P_{in}(t)$, $b=P_{in}(t+2)$, $u=E(t+2)$ and $v=E(t+3)$ then during the $t+2$-th CT a, u, b reach y-input, x-input, z_{in}-input of P_4, respectively. It results that $P_{out}(t+3)=b+ua$. Therefore, during the $t+3$-th CT, P_1 receives $b+ua$, v and b at y-input, x-input and z_{in}-input, respectively and therefore $P_{out}(t+4)=b+v(b+ua)$.

Observe that by sending proper values to RL-cell, this one is able to compute (3).

The linear array implementing Richardson-Romberg extrapolation method using the central divided differences, called $LARRC$, is depicted in Figure 3. We note that, because only one of two consecutive output values of CDD is meaningfull, it is suited that CDD emits only the significant values.

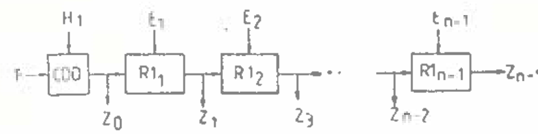


Fig. 3 — The linear systolic array implementing (3)

Theorem 3. *If*

$$F(2k-1)=f(x-h_k),\quad F(2k)=f(x+h_k),\quad H(2k)=c_k,\quad k=1, \ldots, n, \quad \textit{and} \tag{4}$$

$$E_m(2k+4m-1)=-1,\quad E_m(2k+4m)=b_{k+m,m},$$
$$k=1, \ldots, n-m,\quad m=1, \ldots, n-1 \quad \textit{then} \tag{5}$$

$$Z_m(2k+1+4m)=T_{m,k},\quad k=1, \ldots, n-m,\quad m=1, \ldots, n-1. \tag{6}$$

Proof. From (4) and by taking $t=0$ in Proposition 1, it results that $Z_0(2k+1)=T_{0,k}$, $k=1, \ldots, n$. From the assumption $Z_0(2s)=\varnothing$ for each positive integer s, and by applying Proposition 2 with $t=2k+1$, we obtain $Z_1(2k+5)=Z_0(2k+3)+E_1(2k+4)\,[Z_0(2k+3)+E_1(2k+3)\,Z_0(2k+1)]$. Because $E_1(2k+3)=-1$ and $E_1(2k+4)=b_{k+1,1}$, it results that $Z_1(2k+5)=T_{0,k+1}+b_{k+1,1}\,(T_{0,k+1}-T_{0,k})$, $k=1, \ldots, n-1$, and from (3), we have $Z_1(2k+5)=T_{1,k}$, $k=1, \ldots, n-1$. Now, suppose that $1\leq m<n-1$ and $Z_m(2k+1+4m)=T_{m,k}$, $k=1, \ldots, n-m$. Applying again Proposition 2 with $t=2k+1+4m$, we have $Z_{m+1}(2k+1+4m+4)=Z_m(2k+1+4m+2)+E_{m+1}(2k+1+4m+3)\,[Z_m(2k+1+4m+2)+E_{m+1}(2k+1+4m+2)\,Z_m(2k+1+4m)]$. Because $Z_m(2k+4m+3)=T_{m,k+1}$, $Z_m(2k+1+4m)=T_{m,k}$, $E_{m+1}(2k+4m+3)=-1$ and $E_{m+1}(2k+4m+4)=b_{k+m+1,m+1}$, $k=1, \ldots, n-(m+1)$, one obtains $Z_{m+1}(2k+4(m+1)+1)=T_{m,k+1}+b_{k+m+1,m+1}(T_{m,k+1}-T_{m,k})$, and using (3) it results $Z_{m+1}(2k+4(m+1)+1)=T_{m+1,k}$, $k=1, \ldots, n-(m+1)$, i.e. (4)—(5) implies (6).

Let us remark that $f(x+h_1)$ enters $LARRC$ during the first CT, while $T_{n-1,1}$ is available during the $4n-1$ CT, i.e. the computation of $T_{n-1,1}$ needs $4n-2$ CTs. Observe that $LARRC$ is able to estimate $f'_s(x(s, h))$, $h=1, \ldots, r(s)$, $s=1, \ldots, S$, if the values of f_{s+1} are sent to F-input after the complete introducing of f_s-values, $s=1, \ldots, S-1$, i.e. at every $2n$ CTs. The value of $f_1'(x(1, 1))$ appears after $4n-2$ CTs, while the remaining are obtained with the rate of one at every $2n$ CTs.

We note that any $T_{k,m}$ extrapolated value may be a candidate for $f'(x)$. The manner to choose such candidate will be discussed in Section 2.3.

Let us remark that, because of increasing roundoff errors, it is indicated to execute no more than five extrapolations.

2.2. A two-dimensional systolic array. In this section we present a triangular systolic array ($TARRC$) implementing (3). This network uses the basic cell (RT) depicted in Figure 4(a). This cell contains a register keeping a value, during the entire processing activity.

The processor in Figure 4(b) performs $z(t+1)=y(t)-x(t)$, $t \geq 0$.

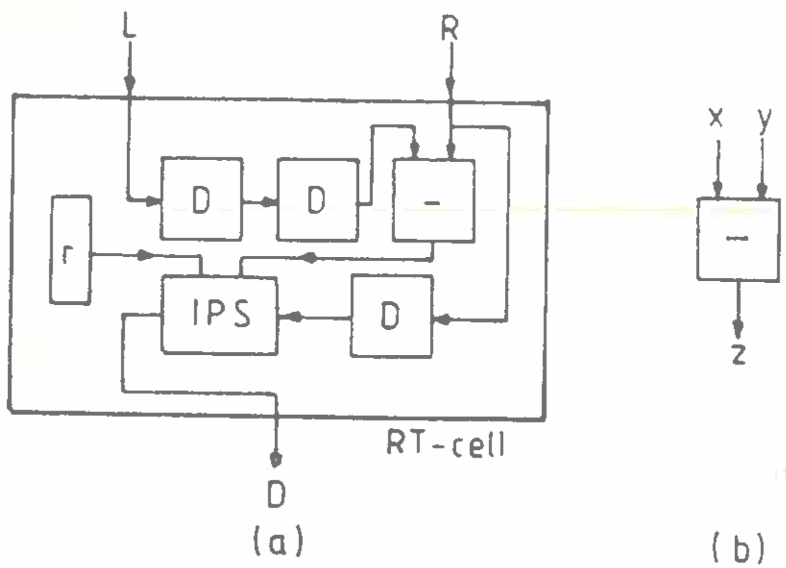


Fig. 4 — The basic cell for the triangular array

Proposition 4. *Under the above assumptions*

$$D(t+h)=R(t+2)+r(R(t+2)-L(t)), \quad t \geq 0.$$

The proof is quite simple.

The structure of $TARRC$ is presented (for $n=5$) in Figure 5. We suppose that after a reset command the register in RT_{km} keeps the value $r_{km}=b_{k+m,\,m}$, $k=1, \ldots, n-m$, $m=1, \ldots, n-1$.

The fact that $TARRC$ correctly implements (3) is stated by

Theorem 5. *If*

$$I_k(t+2k+1)=T_{0,\,k}, \quad k=1, \ldots, n$$

then

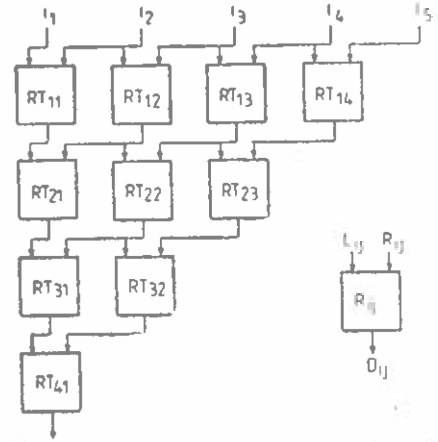


Fig. 5 — The triangular systolic array implementing (3)

$$D_{n,\,k}(t+2k+4m+1)=T_{m,\,k}$$

$k=1, \ldots, n-m$, $m=1, \ldots, n-1$.

Proof. For $m=1$ and for any fixed $k \in \{1, \ldots, n-1\}$ we have $L_{1k}(t+2k+1)=T_{0,\,k}$ and $R_{1k}(t+2k+3)=I_{k+1}(t+2k+3)=T_{0,\,k+1}$ and by applying Proposition 4 with $2k+1$ instead of t, we obtain $D_{1k}(t+2k+5)=R_{1k}(t+2k+3)+r_{1k}[R_{1k}(t+2k+3)-L_{1k}(t+2k+1)]=T_{0,\,k+1}+b_{k+1,\,1}(T_{0,\,k+1}-T_{0,\,k})$, i.e., following (3) it results $D_{1k}(t+2k+5)=T_{1,\,k}$, $k=1, \ldots n-1$.

Now, suppose that $1 \leq m < n-1$ and $D_{mk}(t+2k+4m+1)=T_{m,\,k}$. For each $k \in \{1, \ldots, n-(m+1)\}$ we have $L_{m+1,\,k}(t+2k+4m+1)=D_{m,\,k}(t+2k+4m+1)=T_{m,\,k}$, $R_{m+1,\,k}(t+2k+4m+3)=D_{m,\,k+1}(t+2k+4m+3)=T_{m,\,k+1}$. By applying again Proposition 4 we obtain $D_{m+1,\,k}(t+2k+4m+5)=T_{m,\,k+1}+b_{k+m+1,\,m+1}(T_{m,\,k+1}-T_{m,\,k})=T_{m+1,\,k}$ and this completes the proof.

Observe that $T_{0,\,1}$ enters $TARRC$ during the $t+3$-th CT and $T_{n-1,\,1}$ is obtained during the $t+4(n-1)+3$-th CT, i.e the complete computation requires $4(n-1)$ CTs. Let us suppose that we have to estimate $f_s'(x(s))$, $s=1, \ldots, S$. The corresponding central divided differences $T^s_{0,\,k}$, $k=1, \ldots, n$ may be pumped through the associated inputs with the rate of one at every CT. By taking $t=s$ in Theorem 5 we obtain that if $I_k(s+2k+1)=T^s_{m,k}$ $k=1, \ldots, n$ then $D_{mk}=(s+2k+4m+1)=T^s_{m,\,k}$, $s=1, \ldots, S$ (the contents of registers of RT-cells are preserved). Thus, $TARRC$ has 1 CT as period.

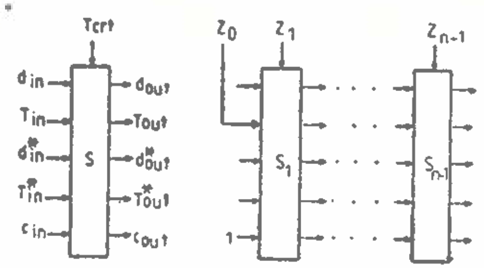
Suppose that the values of f_s are received from the host or another systolic array to compute f_s [2] through a way denoted by W_s, $s=1, \ldots, S$. In order to supply $TARRC$ with the $T^s_{0,\,k}$ values, the way W_s could be connected to a CCD-cell (Figure 1 (c)) whose outputs are sent to a demultiplexer distributing these values to I_k, $k=1,\ldots,n$.

2.3. The choice of the derivative value. Usually, in the sequential algorithm, the computation of T-values is performed along the diagonals of the triangular scheme. The computation of the elements $T_{h,\,j-h}$, $h=0, \ldots, j-1$, belonging to the j-th diagonal, is ended as soon as the values of $d_{jh}=|T_{h,j-h}-T_{h-1,\,j-h+1}|$, $h=1, \ldots, j-1$, stop decreasing, and the desired value of $f'(x)$ is $T_{r,\,j-r}$, where $d_{jr} \leqslant d_{jh}$, $h=1, \ldots, j-1$, $j \in \{2, \ldots, n\}$.

Because, for the presented arrays, it is fruitless to stop the computation, we give a systolic implementation of the above technique operating the choice for all diagonals.

The cell able to do a step of the above computation is depicted in Figure 6(a). Its working is defined by the following procedure:

```
begin if c_in(t)=0 then x_out(t+2) := x_in(t), x∈{d, T, d*, T*, c},
      else begin d := |T_crt(t+1) − T_in(t)|, d_out(t+2) := d,
                 T_out(t+2) := T_crt(t+1),
                 if d > d_in(t) then c_out(t+2) := 0,
                   x_out(t+2) := x_in(t), x∈{d*, T*},
                 else begin if d ≥ d*_in(t) then
                                x_out(t+2) : x_in(t),
                                x∈{d*, T*, c}
                            else d*_out(t+2) := d,
                                 T*_out(t+2) := T_crt(t+1),
                                 c_out(t+2) := c_in(t)
                      end
           end
end
```



(a) (b)

Fig. 6 – Systolic array to select the derivative value

If a linear array of $n-1$ serially connected cells of the above type (Figure 6 (b)) is connected to *LARRC* and d_{in} and d^*_{in} of the left-most cell receive a higher value as initial values, then it is clearly that the best candidate of each diagonal can be extracted from T^* of the right-most cell. A similar solution could be adopted in the case of the triangular array. A more adequate way is to redefine *RL* and *RT* cells in order to support the computation performed by the cell in figure 6(a).

3. The case of one-sided divided differences. According to this variant [4] the sequence of one-sided divided differences is given by

$$T'_{0,\,k}=(f(x+h_k)-f(x))\,c_k,\quad k=1, \ldots, n \tag{7}$$

while the extrapolated values are defined by

$$T'_{m,\,k}=T'_{m-1,\,k+1}+(T'_{m-1,\,k+1}-T'_{m-1,\,k})\,b_{k+m,\,m} \tag{8}$$

$k=1, \ldots, n-m$, $m=1, \ldots, n-1$, where c_k's are previously defined and $b'_{ij}=(n-i+1)/j$, $j=1, \ldots, i-1$, $i=2, \ldots, n$.

In order to use the previously systolic networks it suffices to pump $T'_{0,k}$ with the same rate as $T'_{0,\,k}$, $k=1, \ldots, n$. This could be done by using a cell keeping in a register the value of $-f(x)$, introduced by a reset command, and by sending $f(x+h_k)$ and c_k with the rate of one pair every 2 *CTs*.

Both *LARRC* and *TARRC* can be used to process these values.

An alternative way is to redefine the basic cells of the above arrays in order to accept one extrapolated value at every 1 *CT*.

REFERENCES

1. Brudaru O. – *Systolic Arrays to Solve Nonlinear Equations by a Noniterative Method*, An. şt. Univ. Iaşi (to appear)
2. Brudaru O. – *A Note on the Design of Linear Systolic Arrays for Some Real Functions Computation*, Bul. I.P.I. Iaşi (to appear)
3. Brudaru O. – *Systolic Arrays for Estimating Derivatives by Using Lagrange Polynomial Interpolation*, (submitted for publication)
4. Fillipi S. and Angels H. – *Altes und Nueues zur Numerischen Differentiation*, *Elektronische Datenvearbeitung*, iss 2, 1966, 57–65
5. Kung, H. T., Leiserson C. E. – *Systolic Arrays* (for VLSI), Sparse Matrix Proc. (I.S. Buff, G. W. Stewart – eds.), Society for Industrial and Applied Mathematics, Philadelphia, 1979, 256–282
6. Kung H. T. – *Why Systolic Architectures ?*, Computer (IEEE), 15(1), 1982, 37–46
7. Quinton P. – *An Introduction to Systolic Architectures*, Future Parallel Computers LNCS, (272), P. Treleaven, M. Vanneschi (eds.), Springer-Verlag, 1986, 387–400
8. Schreiber R. – *A Survey of Systolic Computation*, Research Report, Computer Science Department, Rensselaer Polytechnic Institute Troy, New York, 1986.

Received 4.IV.88

Institutul Politehnic Iaşi
Centrul de Calcul
România