An Implementation of MacMahon's Partition Analysis in Ordering the Lower Bound of Processing Elements for the algorithm of LU-Decomposition

Halil Snopce, Ilir Spahiu, Lavdrim Elmazi

Abstract—A lot of Scientific and Engineering problems require the solution of large systems of linear equations of the form $Ax = b$ in an effective manner. LU-Decomposition offers good choices for solving this problem. Our approach is to find the lower bound of processing elements needed for this purpose. Here is used the so called "Omega calculus", as a computational method for solving problems via their corresponding Diophantine relation. From the corresponding algorithm is formed a system of linear diophantine equalities using the domain of computation which is given by the set of lattice points inside the polyhedron. Then is run the Mathematica program DiophantineGF.m. This program calculates the generating function from which is possible to find the number of solutions to the system of Diophantine equalities, which in fact gives the lower bound for the number of processors needed for the corresponding algorithm. There is given a mathematical explanation of the problem as well.

Keywords—generating function, lattice points in polyhedron, lower bound of processor elements, system of Diophantine equations and Ω calculus.

I. INTRODUCTION

THERE are a lot of studies concerning the processor-time-minimal schedules and optimizing of different arrays [1-11]. It is known that for the algorithms of matrix product, Gaussian elimination and Transitive closure the number of processors is $3n^2/4$, $n^2/4$ and $n^2/3$ respectively. Transformation of the problem from geometrical into combinatorial analysis can be seen at [11, 12]. Mathematical guide for the analysis can be seen at [12-20]. An application for nested loop algorithms of the formulae for the number of solutions of Diophantine system of equalities is given in [22]. A general and uniform technique for deriving lower bounds of processing elements (as a piecewise polynomial function) is presented at [11]. At the same article is shown that the nodes of the dag can be viewed as lattice points in convex polyhedron. Adding to this the linear constraint of the schedule, there will be form a system of Diophantine equations where the number of solutions is a lower bound. In this article, using the steps mentioned above, we have obtained the optimal lower bound for the number of processors required by the systolic algorithm for DFT and for the algorithm of LU-Decomposition. At the beginning we give some definitions and lemas which are used for mathematical explanation followed by an example. We have used the same example for applying the mathematica program DiophantineGF.m. We show that the corresponding generating function is not changed. So for the problems mentioned above we can use this program.

II. SOME DEFINITIONS AND PROPERTIES

Definition 1: Let $a \in R^n$ and $b \in R$. Then a hyperplane consists of the set $\{x \in R^n \setminus a^T x = b\}$ and a halfspace consists of the set $\{x \in R^n \setminus a^T x \geq b\}$. If A is an mxn matrix and $b \in R^m$ then a polyhedron P consists of the set $P = \{x \in R^n \setminus Ax \leq b\}$, in other words, a polyhedron is the intersection of finitely many halfspaces.

Definition 2: For a multiple Laurent series, $\sum_{v_1,..,v_k=-\infty}^{\infty} A_{v_1,...,v_k} \lambda_1^{v_1} ... \lambda_k^{v_k}$, the operator $\Omega_{\geq}$ is defined by:

$$\Omega_{\geq} \sum_{v_1,..,v_k=-\infty}^{\infty} A_{v_1,...,v_k} \lambda_1^{v_1} ... \lambda_k^{v_k} = \sum_{v_1,..,v_k=0}^{\infty} A_{v_1,...,v_k}.$$

Two of many identities presented in [17] are given below:

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

Lema 1: For any integer $s \geq 0$,

$$\Omega_{\geq} \frac{1}{(1-\lambda x)\left(1-\dfrac{y}{\lambda^s}\right)} = \frac{1}{(1-x)(1-x^s y)}$$

Lema 2: $\Omega_{\geq} \dfrac{1}{(1-\lambda^2 x)\left(1-\dfrac{y}{\lambda}\right)} = \dfrac{1+xy}{(1-x)(1-xy^2)}$

MacMahon leaves the verification of many of his identities to the reader. Below is given prove for first lema.

From geometric series expansion there will be:

$$\frac{1}{(1-\lambda x)\left(1-\dfrac{y}{\lambda^s}\right)} = \sum_{a_i \geq 0}(\lambda x)^{a_1}\left(\frac{y}{\lambda^s}\right)^{a_2} = \sum_{a_i \geq 0}\lambda^{a_1-a_2 s}x^{a_1}y^{a_2}$$

If $a_2 s > a_1$, then λ will have a negative power. To prevent this from happening, let $a_1 - a_2 s = b$, forcing the restriction $b \geq 0$ and making appropriate substitution into the crude generating function there will be:

$$\sum_{a_i \geq 0}\lambda^{a_1-a_2 s}x^{a_1}y^{a_2} = \sum_{a_2,b \geq 0}\lambda^{b}x^{a_2 s+b}y^{a_2} =$$

$$= \sum_{a_2,b \geq 0}(\lambda x)^{b}\left(x^s y\right)^{a_2} = \frac{1}{(1-\lambda x)(1-x^s y)}$$

Now with the substitution for $\lambda = 1$, the desired identity will be fulfilled. (Using the mentioned conditions above, actually is used the defined $\Omega_{\geq}$ operator).

III. MATHEMATICAL EXPLANATION OF THE ALGORITHM

If there is a polyhedron $P = \{x \in R^n \setminus Ax \geq b\}$, then for each defining halfspace $a_i x - tb_i \geq 0$ embeding $\lambda^{(a_i x - tb_i)}$ into a crude generating function is a general idea of this process. Because of considering only the positive constraints $t \geq 0$, embedding y^t into the crude generating function, then the form of obtained function is:

$$F(\lambda, y) = \sum_{x_i,t \geq 0}\lambda_1^{(a_1 x - tb_1)}\lambda_2^{(a_2 x - tb_2)}...\lambda_m^{(a_m x - tb_m)}y^t$$

For example let $t \geq 0$ and $P = \{(x_1, x_2) \in R^2 / x_2 + 2x_1 \geq 2, 2 \geq x_2, 1 \geq x_1\}$ be a given polyhedron. Then the number of integer points

contained in $t \cdot P$ is equivalent to the number of integer solutions of the system:

$$\begin{aligned} x_2 + 2x_1 - 2t &\geq 0 \\ 2t - x_2 &\geq 0 \\ t - x_1 &\geq 0 \\ t &\geq 0 \end{aligned} \quad (1)$$

In fact this is the polyhedron with $P = \{x \in R^2 / Ax \leq b\}$ where:

$$A = \begin{bmatrix} -2 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } b = \begin{bmatrix} -2 \\ 2 \\ 1 \end{bmatrix}$$

The corresponding crude generating function to this system will be:

$$\sum_{x_i,t \geq 0}\lambda_1^{x_2+2x_1-2t}\lambda_2^{2t-x_2}\lambda_3^{t-x_1}y^t$$

With an additional transformations there will be:

$$\sum_{x_i,t \geq 0}\lambda_1^{x_2+2x_1-2t}\lambda_2^{2t-x_2}\lambda_3^{t-x_1}y^t =$$

$$= \sum_{x_i,t \geq 0}\left(\frac{\lambda_1^2}{\lambda_3}\right)^{x_1}\left(\frac{\lambda_1}{\lambda_2}\right)^{x_2}\left(\frac{\lambda_2^2 \lambda_3}{\lambda_1^2}y\right)^{t}$$

This corresponds to the following crude rational generating function:

$$\frac{1}{\left(1-\dfrac{\lambda_1^2}{\lambda_3}\right)\left(1-\dfrac{\lambda_1}{\lambda_2}\right)\left(1-\dfrac{\lambda_2^2 \lambda_3}{\lambda_1^2}y\right)}$$

To find the corresponding rational generating function, there are used the two lemas given above ($\Omega_{\geq}(\lambda)$ means that the given identity is used for parameter λ).

An implications of using lema 1 for $s = 1$, lema 1 for s=2 and lema 2 respectively are:

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

$$\Omega_{\ge}(\lambda_3)\frac{1}{\left(1-\frac{\lambda_1^2}{\lambda_3}\right)\left(1-\frac{\lambda_1}{\lambda_2}\right)\left(1-\frac{\lambda_2^2\lambda_3}{\lambda_1^2}y\right)}=$$

$$=\frac{1}{\left(1-\frac{\lambda_1}{\lambda_2}\right)\left(1-\frac{\lambda_2^2}{\lambda_1^2}y\right)\left(1-\lambda_2^2y\right)}$$

$$\Omega_{\ge}(\lambda_1)\frac{1}{\left(1-\frac{\lambda_1}{\lambda_2}\right)\left(1-\frac{\lambda_2^2}{\lambda_1^2}y\right)\left(1-\lambda_2^2y\right)}=\frac{1}{\left(1-\lambda_2^2y\right)\left(1-\frac{1}{\lambda_2}\right)(1-y)}$$

$$\Omega_{\ge}(\lambda_2)\frac{1}{\left(1-\lambda_2^2y\right)\left(1-\frac{1}{\lambda_2}\right)(1-y)}=\frac{1}{(1-y)}\cdot\frac{1+y\cdot1}{(1-y)(1-y\cdot1^2)}$$

So, the conclusion is:

$$\Omega_{\ge}\frac{1}{\left(1-\frac{\lambda_1^2}{\lambda_3}\right)\left(1-\frac{\lambda_1}{\lambda_2}\right)\left(1-\frac{\lambda_2^2\lambda_3}{\lambda_1^2}y\right)}=\frac{1+y}{(1-y)^3}$$

This is the same rational generating function which can be obtained via Mathematica program *DiophantineGF.m*. Transformation of the set of inequalities (1) to a set of Diophantine equations is done using an integral slack variables $s_1, s_2, s_3 \ge 0$ and the corresponding system is:

$$\begin{aligned}2x_1 + x_2 - s_1 &= 2t\\ -x_2 - s_2 &= -2t\\ -x_1 - s_3 &= -t\\ -s_4 &= -t\end{aligned} \tag{2}$$

Because the program *DiophantineGF.m* essentially requires three arguments (A,b,c) of the Diophantine system $Ax = bt + c$, the main computation is performed by the call *DiophantineGF[A,b,c]*. The result is the rational generating function. The form of (A,b,c) is found from (2) and given below:

$$A=\begin{bmatrix}2 & 1 & -1 & 0 & 0 & 0\\0 & -1 & 0 & -1 & 0 & 0\\-1 & 0 & 0 & 0 & -1 & 0\\0 & 0 & 0 & 0 & 0 & -1\end{bmatrix},\quad b=\begin{bmatrix}2\\-2\\-1\\-1\end{bmatrix},\quad c=\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$$

The original result from the program *DiophantineGF.m* is given below as well:

In[1]:=<<DiophantineGF.m

In[2]:=a={{2,1,-1,0,0,0},
　　　　　{0,-1,0,-1,0,0},
　　　　　{-1,0,0,0,-1,0},
　　　　　{0,0,0,0,0,-1}};
In[3]:=b={2,-2,-1,-1};c={0,0,0,0};
In[4]:=DiophantineGF[a,b,c]

$$Out[1]=-\frac{1+t}{(-1+t)^3}$$

With substitution $t = y$ the result is same like in previous mathematical explanation.

$$-\frac{1+t}{(-1+t)^3}=\frac{1+t}{(1-t)^3}=\frac{1+y}{(1-y)^3}$$

IV. LOWER BOUND OF PROCESSOR ELEMENTS (PEs) OF THE SYSTOLIC ARRAY FOR DISCRETE FOURIER TRANSFORM (DFT) BASED ON MATRIX MULTIPLICATION

The algorithm for the writing the 2 dimensional DFT which is used for designing the corresponding systolic array is given below (taken from [1]):

Algorithm 1
Internal computations

for $j_1 = 0$ to $n_1 - 1$ do
　　for $j_2 = 0$ to $n_2 - 1$ do
　　　　for $j_3 = 0$ to $n_1 - 1$ do
　　　　　$z(j_1,j_2,j_3) = z(j_1,j_2,j_3-1)$
　　　　　$+ \omega_1(j_1,-1,j_3) \times x(-1,j_2,j_3);$ (3)

for $j_1 = 0$ to $n_1 - 1$ do
　　for $j_2 = 0$ to $n_2 - 1$ do
　　　　for $j_3 = n_1$ to $n_1 + n_2 - 1$ do
　　　　　$y(j_1,j_2,j_3) = y(j_1,j_2-1,j_3)$
　　　　　$+ z(j_1,j_2,n_1-1) \times \omega_2(-1,j_2,j_3+n_1);$ (4)

Output computations

$$\lfloor y_{j_1j_3}\rfloor_{n_1 \times n_2} = [y(j_1,n_2-1,j_3)]_{n_1 \times n_2};$$

From above the conclusion is that the computational structure is characterized by an index space:

$$P_{\text{int}} = \left\{(j_1,j_2,j_3)^T\right\} \in Z^3 \text{ where:}$$

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

$$0 \leq j_1 \leq n_1 - 1, \ 0 \leq j_2 \leq n_2 - 1, \ 0 \leq j_3 \leq n_1 + n_2 - 1.$$

The data dependence vectors for variables from (3) and (4) are

$$(0,0,1)^T, (0, j_2+1, 0)^T, (j_1+1, 0, 0)^T \text{ and}$$

$$(0,1,0)^T, (0, 0, j_3 - n_1 + 1)^T, (j_1+1, 0, -n)^T, \text{ respectively.}$$

In this case (j_1, j_2, j_3) are lattice points inside 3-dimensional convex polyhedron whose faces are defined by the inequalities which are the consequence of the *algorithm 1*. The obtained inequalities by the converting the geometrical into a combinatorial interpretation are given below:

$$j_1 \leq n_1 - 1, \ j_2 \leq n_2 - 1, \ j_3 \leq n_1 + n_2 - 1$$

The next step is transforming this into the system of equalities putting the slack variables $s_1, s_2, s_3 \geq 0$ and augmenting this by the condition of linear schedule for the corresponding dag which is given with $j_1 + j_2 + j_3 = 4n - 2$ (taking the case $n_1 = n_2 = n_3 = n$). This ranges from 1 to $4n - 2$, and in this case is taken the halfway point which is:

$$j_1 + j_2 + j_3 = \frac{4n-2}{2} = 2n - 1$$

The corresponding system of Diophantine equalities is:

$$
\begin{aligned}
j_1 + j_2 + j_3 & & & = 2n - 1 \\
j_1 & + s_1 & & = n - 1 \\
j_2 & + s_2 & & = n - 1 \\
j_3 & + s_3 & & = 2n - 1
\end{aligned}
\qquad (5)
$$

(A, b, c) form is found from the system (5):

$$
A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad
b = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 2 \end{bmatrix}, \quad
c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}
$$

The next step is to run the program *DiophantineGF.m*:

```
In[1]:=<<DiophantineGF.m
In[2]:=a={{1,1,1,0,0,0},
        {1,0,0,1,0,0},
        {0,1,0,0,1,0}
```

```
        {0,0,1,0,0,1}};
In[3]:=b={2,1,1,2};c={-1,-1,-1,-1};
In[4]:=DiophantineGF[a,b,c]
Formula
```

$$\text{Out[1]} := -\frac{t(1+t)}{(-1+t)^3}$$

Binomial Formula : C[n,2]+C[1+n,2]

Power Formula : n^2

This means that the lower bound for the number of PEs of systolic array for 2 dimensional DFT is n^2. In [1] is given a table of number of processors elements of systolic arrays, where can be seen that the array obtained along the projection direction $(0,0,1)^T$, i.e. along j_3 axis, is optimal in terms of number of PEs. This number is $n_1 n_2$ which is the same with our result for $n_1 = n_2 = n$.

V. Lower Bound of Processor Elements for LU Factorization

The algorithm for factoring the $n \times n$ matrix $A = (a_{ij})$ into the product of the lower triangular matrix $L = (l_{ij})$ and the upper triangular matrix $U = (u_{ij})$; that is, $A = LU$, where the main diagonal of either L or U consists of all ones is presented below (taken from [21]):

Algorithm 2

INPUT dimension n; the entries $a_{ij}, 1 \leq i, j \leq n$ of A; the diagonal $l_{11} = \dots = l_{nn} = 1$ of L or the diagonal $u_{11} = \dots = u_{nn} = 1$ of U.

Step 1: Select l_{11} and u_{11} satisfying $l_{11} u_{11} = a_{11}$.
   If $l_{11} u_{11} = 0$ then OUTPUT (‚Factorization impossible');
   STOP.

Step 2: For $j = 2, \dots, n$ set $u_{1j} = a_{1j} / l_{11}$; (First row of U.) $l_{j1} = a_{j1} / u_{11}$. (first column of L).

Step 3: For $i = 2, \dots, n-1$ do Steps 4 and 5.

Step 4: Select l_{ii} and u_{ii} satisfying

$$l_{ii} u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik} u_{ki}. \text{ If } l_{ii} u_{ii} = 0 \text{ then OUTPUT}$$

('Factorization impossible');
   STOP.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

Step 5: For $j = i+1, \ldots, n$ set

$$u_{ij} = \frac{1}{l_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right]; \text{(ith row of } U \text{ .)}$$

$$l_{ji} = \frac{1}{u_{ii}} \left[a_{ij} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right]. \text{ (ith column of } L \text{ .)}$$

Step 6: Select l_{nn} and u_{nn} satisfying

$$l_{nn} u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk} u_{kn} .$$

Step 7: OUTPUT (l_{ij}, $1 \le j \le i$, $1 \le i \le n$);

OUTPUT (u_{ij}, $i \le j \le n$, $1 \le i \le n$);

STOP.

The computational structure is characterized by the index space:

$$P_{\text{int}} = \left\{ (i,j,k) / 0 \le i \le n-1, i+1 \le j \le n-1, i+1 \le k \le n-1 \right\}$$

There is used a translation of the loop taking $0 \le i \le n-1$ as opposed to $1 \le i \le n$, because it is implicit. The same is done with the other index points.

The array computation for the algorithm above ($n \times n \times n$ mesh) is given by $G_n = (P_{\text{int}}, A)$, where:

$$A = \{(i,j,k),(i',j',k') / (i,j,k) \in P_{\text{int}}, (i',j',k') \in P_{\text{int}}$$
$$and \; i' = i+1, j' = j, k' = k \; or$$
$$j' = j+1, i' = i, k' = k \; or \; k' = k+1, i' = i, j' = j\}$$

In this case (i,j,k) are lattice points inside 3-dimensional convex polyhedron whose faces are defined by the inequalities which are the consequence of the *algorithm 2*.

Converting the geometrical interpretation of the problem explained above, into a combinatorial interpretation, exactly into finding of solutions to the system of Diophantine equations the following four inequalities are got: $j \ge i+1, j \le n-1, k \ge i+1, k \le n-1$ (there is no specification of the case $0 \le i,j,k$ and $i \le n-1$). The result of the transforming this into the system of equalities putting the slack variables $s_1, s_2, s_3, s_4 \ge 0$ is:

$$j = i+1+s_1$$
$$j + s_2 = n-1$$
$$k = i+1+s_3$$
$$k + s_4 = n-1$$

Augmenting this by the condition of linear schedule for the corresponding dag which is given with $i+j+k = 3n-1$ (this ranges from 1 to $3n-1$) and taking the halfway point in this schedule, which means $i+j+k = \dfrac{3n-1}{2}$, then the result is the corresponding system of Diophantine equalities where the number of solutions is a lower bound for the number of processors:

$$
\begin{aligned}
2i + 2j + 2k & & &= 3n-1 \\
i - \; j + \; s_1 & & &= \; -1 \\
j \; & + s_2 & &= n-1 \quad\quad (6)\\
i - \; k & + s_3 & &= \; -1 \\
k & + s_4 & &= n-1
\end{aligned}
$$

From system (6) the values of (A, b, c) are found:

$$
A = \begin{bmatrix}
2 & 2 & 2 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix},
\quad
b = \begin{bmatrix} 3 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix},
\quad
c = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}
$$

Running the program *DiophantineGF.m* is found the lower bound of PEs for this algorithm:

```
In[1]:=<<DiophantineGF.m
In[2]:=a={{2,2,2,0,0,0,0},
        {1,-1,0,1,0,0,0},
        {0,1,0,0,1,0,0},
        {1,0,-1,0,0,1,0},
        {0,0,1,0,0,0,1}};
In[3]:=b={3,0,1,0,1};c={-1,-1,-1,-1,-1};
In[4]:=DiophantineGF[a,b,c]
Formula
```

$$\text{Out[1]}:= -\frac{t^3 \left(1 + 3t^2\right)}{(-1+t)^3 (1+t)^3 (1+t)^2}$$

Binomial Formula : 1/32 (-21 C[2+1/2 (10+n),2]+6 C[2+1/2 (-9+n),2]+59 C[2+1/2 (8+n),2]-22 C[2+1/2 (-7+n),2]-47 C[2+1/2 (6+n),2]+34 C[2+1/2 (-5+n),2]-23 C[2+1/2 (4+n),2]+14 C[2+1/2 (-3+n),2]-12 C[1/4 (7+n),0]+8 C[1/4 (-5+n),0]+21 C[-5+n,2]-57 C[4+n,2]+4 C[1/4 (-3+n),0]+49 C[-3+n,2]-19 C[2+n,2]+14 C[-1+n,2])

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:6, 2009

Simplifying the binomial coefficients above is found the lower bound of PEs, which is $\dfrac{n^2 - n}{4}$.

REFERENCES

[1] M.P. Bekakos, Highly Parallel Computations-Algorithms and Applications, Democritus University of Thrace, Greece, pp. 139-209, 2001.

[2] N. Petkov, Systolic Parallel Processing, North-Holland, 1993

[3] Special issue on: The Future of Micro-Processors, Computer, vol. 30, no.9, September 1997

[4] S.Y.Kung, VLSI Array Processors, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[5] Kung, H.T. and Leiserson, C.E., Systolic arrays for (VLSI), Introduction to VLSI Systems, Addison-Wesley Ltd., Reading, MA, 1980.

[6] W. Shang, J.A.B. Fortes, Time optimal linear schedule for algorithms with uniform dependencies, IEEE Trans. Comput. 40 (6) (1991) 723-742.

[7] C. Scheiman, P. Cappello, A processor-time minimal systolic array for the 3D rectilinear mesh, in: Proc. Int. Conf. On Application Specific Array Processors, Strasbourg, France, July, 1995, pp.26-33

[8] C. Scheiman, P. Cappello, A period processor-time minimal schedule for cubical mesh algorithms, IEEE Trans. Comput. 40 (6) (1991) 723-742.

[9] C. Scheiman, P. Cappello, A processor-time minimal systolic array for transitive closure, IEEE Trans. Parallel Distributed Syst. 3 (3) (1992) 257-269.

[10] A. Benaini, Y. Robert, Spacetime-minimal systolic arrays for Gaussian elimination and the algebric path problem, in: Proc. Int. conf. On Application Specific Array Processors, IEEE Computer Society, Princeton, 1990, pp. 746-757.

[11] P. Cappello, Omer Egecioglu, Processor lower bound formulas for array computations and parametric Diophantine systems, Int. J. Found. Comput. Sci. 9 (4) (1998) 351-378

[12] Ph. Clauss, C. Mongenet, G.R. Perrin, Calculus of space-optimal mappings of systolic algorithms on processor arrays, in: Proc. Int. Conf. on Application Specific Array Processors, IEEE Computer Society, Princeton, 1990, pp. 4-18

[13] K. Down, High Performance Computing, O' Reilly & Associates, Inc., California, 1993

[14] P. Clauss, V. Loechner, Parametric analysis of polyhedral iteration spaces, J. VLSI Signal Process. 19 (1998) 179-194

[15] D.K. Wilde, A library for doing polyhedral operations, Master's thesis, Corvallis, Oregon, December 1993, Also published as IRISA technical report PI 785, Rennes, France, December 1993

[16] R.P. Stanley, Linear Homogeneous Diophantine equations and magic labelings of graphs, Duke Math. J. 40 (1973) 607-632

[17] Major Percy A. Macmahon, F.R.S., D.Sc., LL.D., Combinatory Analysis, volume II, Cambridge University Press, 1916 (Reprinted: Chelsea, New York, 1960)

[18] George E. Andrews, Peter Paule, A. Riese, MacMahon's Partition Analysis: The Omega Package, European Journal of Combinatorics, October 2001, vol. 22, iss. 7, pp. 887-904 (18) Academic press., London

[19] G.E, Andrews, MacMahon's Partition Analysis II: Fundamental Theorems, Ann. Comb. 4 (2000), 327-338.

[20] Guouce Xin, A Fast Algorithm for MacMahon's Partition Analysis. Elect. Journal of Comb. (11), 2004, R58, 20 pp.

[21] Richard L. Burden, J. Douglas Faires, Numerical Analysis, 6th edition, 1997.

[22] Philippe Clauss and Vincent Loechner. Parametric analysis of polyhedral iteration spaces. Journal of VLSI Signal Processing, 19:179-194, 1998.

Authors: Halil Snopce[1], Ilir Spahiu[2] , Lavdrim Elmazi[3]

[1,3] South East European University CST Faculty, Department of computer science, 1200 Ilindenska bb, Tetovo,Republic of Macedonia
[2] Faculty of Pedagogy "ss Clement of Ohrid", "Ss. Cyril and Methodius" University in Skopje, Republic of Macedonia

E-mail: [1]h.snopce@seeu.edu.mk, [2]i.spahiu.@seeu.edu.mk [3]l.elmazi@seeu.edu.mk