Solving Systems of Linear Equations in Complex Domain: Complex E-Method

Milos Ercegovac, Jean-Michel Muller

Solving Systems of Linear Equations in Complex Domain: Complex E-Method

Miloš D. Ercegovac
Computer Science Department
3732 Boelter Hall
University of California at Los Angeles
Los Angeles, CA 90024, USA
milos@cs.ucla.edu

Jean-Michel Muller
CNRS-Laboratoire LIP, projet Arénaire
Ecole Normale Supérieure de Lyon
46 Allée d'Italie
69364 Lyon Cedex 07, France
Jean-Michel.Muller@ens-lyon.fr

This is LIP research report number 2007-2
LIP is a laboratory of CNRS, Ecole Normale Supérieure de Lyon,
INRIA and Université Claude Bernard Lyon 1

Abstract

The E-method, introduced in [2, 3], allows efficient parallel solution of diagonally dominant systems of linear equations in real domain using simple and highly regular hardware. Since the evaluation of polynomials and certain rational functions can be achieved by solving the corresponding linear systems, the E-method is an attractive general approach for function evaluation. We generalize the E-method to complex linear systems, and show some potential applications such as the evaluation of complex polynomials and rational functions.

1 Introduction

In this report we propose an extension of a digit-iterative method for solving systems of linear equations, the E-method [2, 3, 7], to allow the use of the complex number system. The proposed approach is suitable for hardware implementation. The main characteristics of the method are: (i) m-digit solution is computed in about m steps, each step consisting of a sum of number-by-digit products, (ii) the cycle time depends on the number of nonzero

coefficients, (iii) the cycle time does not depend on the precision m (if redundant additions are used), (iv) for a system of order n, the shortest latency requires n elementary units for the real part, and n units for the imaginary part, and (v) the elementary units are interconnected with digit-wide links. The approach is particularly efficient when the coefficient matrix is sparse. This happens when the E-method is used to evaluate polynomials (one off-diagonal element) and rational functions (two off-diagonal elements). Other examples are a tridiagonal system (two off-diagonal elements), powers of the argument (one off-diagonal element), and special expressions.

We first introduce the transform which allows the E-method to be used in the complex field **C**. Then we show how to use the complex E-method (CE-method) in evaluating complex polynomials and rational functions as particularly interesting cases. Evaluation of consecutive powers of a complex argument is a special case of polynomial evaluation.

With the exception of complex addition and multiplication, other complex operations are typically not implemented in hardware. Online algorithms for complex arithmetic have been proposed and implemented in FPGAs in [8, 9]. Based on this work, algorithms and implementations for complex FIR filters, complex matrix inversion, and complex Householder transform have been developed [10, 11, 12]. Recently, hardware-oriented methods for complex division and square root have been introduced [4, 6]. The method proposed in this report extends complex arithmetic to complex polynomials, complex powers, and rational functions - a significant extension of the domain of hardware implementation for complex arithmetic.

Complex polynomials appear in many areas such as digital signal and image processing, control systems, and applied mathematics, in general. A Horner type method for evaluating complex polynomials is proposed in [1] at the algorithm level, implicitly assuming a software implementation. The method uses $O(n)$ multiplications and $O(n)$ additions for a complex polynomial of degree n. If these multiplications and additions are performed in a sequential order, the latency of the method is about $n \times T_{MULT-ADD}$ which is significantly slower than our method. If a parallel algorithm for polynomial evaluation is used, the total time is about $\log n \times T_{MULT-ADD}$ which is still slower than our method.

The case of rational functions with complex coefficients and argument is even more attractive: using the proposed CE-method, we avoid explicit complex division and produce the result, as mentioned above, in time proportional to the desired precision. We are not aware of prior special algorithms for evaluation of complex rational functions in hardware.

In the next section we describe the transformation which maps computation from the complex to the real domain. In Section 3 we show the CE-method. In Section 4 iterations and convergence conditions are considered. Implementation aspects are discussed in Section 5.

2 Complex-Real (CR) Transforms

Complex numbers can be represented by 2×2 skew-symmetric matrices

$$x + iy \leftrightarrow \begin{pmatrix} x & -y \\ y & x \end{pmatrix} \tag{1}$$

This isomorphism holds for complex addition and multiplication which are used in the proposed method :

$$\begin{aligned}(a+ib)+(c+id) &\leftrightarrow \begin{pmatrix} a & -b \\ b & a \end{pmatrix} + \begin{pmatrix} c & -d \\ d & c \end{pmatrix} \\ &= \begin{pmatrix} a+c & -b-d \\ b+d & a+c \end{pmatrix} \leftrightarrow (a+c)+i(b+d) \end{aligned} \tag{2}$$

$$\begin{aligned}(a+ib)\times(c+id) &\leftrightarrow \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \times \begin{pmatrix} c & -d \\ d & c \end{pmatrix} \\ &= \begin{pmatrix} ac-bd & -(ad+bc) \\ ad+bc & ac-bd \end{pmatrix} \leftrightarrow (ac-bd)+i(bc+ad) \end{aligned} \tag{3}$$

Consequently, an $m \times n$ matrix of complex numbers can be represented as a $2m \times 2n$ matrix of real numbers. For $n \times n$ complex matrices, considered in this paper, the transform from the complex domain to the real domain is defined next.

Definition 1 *The* CR-transform *of the n-dimensional* complex *linear system*

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & \cdots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & \cdots & a_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n,1} & a_{n,2} & a_{n,3} & a_{n,4} & \cdots & a_{n,n} \end{pmatrix} \times \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_n \end{pmatrix} \tag{4}$$

is the $2n$*-dimensional* real *linear system*

$$\begin{pmatrix} a^r_{1,1} & -a^i_{1,1} & a^r_{1,2} & -a^i_{1,2} & \cdots & a^r_{1,n} & -a^i_{1,n} \\ a^i_{1,1} & a^r_{1,1} & a^i_{1,2} & a^r_{1,2} & \cdots & a^i_{1,n} & a^r_{1,n} \\ a^r_{2,1} & -a^i_{2,1} & a^r_{2,2} & -a^i_{2,2} & \cdots & a^r_{2,n} & -a^i_{2,n} \\ a^i_{2,1} & a^r_{2,1} & a^i_{2,2} & a^r_{2,2} & \cdots & a^i_{2,n} & a^r_{2,n} \\ a^r_{3,1} & -a^i_{3,1} & a^r_{3,2} & -a^i_{3,2} & \cdots & a^r_{3,n} & -a^i_{3,n} \\ a^i_{3,1} & a^r_{3,1} & a^i_{3,2} & a^r_{3,2} & \cdots & a^i_{3,n} & a^r_{3,n} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \\ a^r_{n,1} & -a^i_{n,1} & a^r_{n,2} & -a^i_{n,2} & \cdots & a^r_{n,n} & -a^i_{n,n} \\ a^i_{n,1} & a^r_{n,1} & a^i_{n,2} & a^r_{n,2} & \cdots & a^i_{n,n} & a^r_{n,n} \end{pmatrix} \times \begin{pmatrix} z^r_1 \\ z^i_1 \\ z^r_2 \\ z^i_2 \\ z^r_3 \\ z^i_3 \\ \vdots \\ z^r_n \\ z^i_n \end{pmatrix} = \begin{pmatrix} t^r_1 \\ t^i_1 \\ t^r_2 \\ t^i_2 \\ t^r_3 \\ t^i_3 \\ \vdots \\ t^r_n \\ t^i_n \end{pmatrix} \tag{5}$$

where $a_{j,k} = a^r_{j,k} + ia^i_{j,k}$, $z_j = z^r_j + iz^i_j$ *and* $t_j = t^r_j + it^i_j$. These two linear systems are *equivalent.*

□

In other words, the real linear system (5) is obtained from the complex linear system (4) by replacing each element $x + ix$ by the 2×2 matrix defined in (1). In the next section we consider a hardware-oriented method for solving such a system.

3 Complex E-method

The E-method [2, 3], provides an iterative approach of solving diagonally dominant real linear systems. The method has characteristics desirable for efficient hardware implementation: the basic operators are bit-vector multiplexers, redundant adders of $[p : 2]$ type, with $p \in \{3, 4, 6\}$ for radix-2, and registers. The overall structure consists of n elementary units, interconnected digit-serially. The method computes one digit of each component of the solution per iteration in the MSDF (Most Significant Digit First) manner which allows digit-serial communication between the modules which operate concurrently. The time to obtain the solution to m digits of precision is about m cycles (iterations). The amount of hardware required is roughly related to the number of nonzero terms of the matrix of the system, which makes the E-method very efficient in hardware resources when the matrix of the system is sparse. Typical applications of the E-method are evaluation of polynomial and rational functions, since these correspond to sparse linear systems. The solution of the linear system

$$\begin{pmatrix} 1 & -x & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -x & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 1 & -x \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix}$$

is

$$\begin{pmatrix} p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n \\ p_1 + p_2 x + \cdots + p_n x^{n-1} \\ \vdots \\ p_{n-1} + p_n x \\ p_n \end{pmatrix}$$

that is, the first component of the solution is

$$p_0 + p_1 x + p_2 x^2 + \cdots + p_n x^n$$

whereas the solution of the linear system

$$\begin{pmatrix} 1 & -x & 0 & 0 & 0 & \cdots & 0 \\ q_1 & 1 & -x & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ q_{n-1} & 0 & \cdots & 0 & 0 & 1 & -x \\ q_n & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix}$$

is

$$\begin{pmatrix} \frac{p_0+p_1x+p_2x^2+\cdots+p_nx^n}{1+q_1x+q_2x^2+\cdots+q_nx^n} \\ \frac{(p_1-p_0q_1)+(p_2-p_0q_2)x+\cdots+(p_n-p_0q_n)x^{n-1}}{1+q_1x+q_2x^2+\cdots+q_nx^n} \\ \vdots \\ \vdots \\ \frac{(p_n-q_np_0)+(p_nq_1-q_np_1)x+\cdots+(p_nq_{n-1}-q_np_{n-1})x^{n-1}}{1+q_1x+q_2x^2+\cdots+q_nx^n} \end{pmatrix}$$

That is, the first component of the solution is the rational function

$$\frac{p_0 + p_1x + p_2x^2 + \cdots + p_nx^n}{1 + q_1x + q_2x^2 + \cdots + q_nx^n}.$$

Now, let us turn to the evaluation of complex polynomials of a complex argument. We wish to evaluate

$$p(z) = p_0 + p_1z + p_2z^2 + \ldots + p_nz^n$$

where the p_j's and z are complex numbers. As in the real case, the desired value $p(z)$ is clearly equal to the first component of the solution of the linear system

$$\begin{pmatrix} 1 & -z & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & -z & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & -z & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix} \times \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ \vdots \\ p_n \end{pmatrix} \tag{6}$$

The E-method cannot directly solve the linear system (6), but now if we define real numbers x and y as $x + iy = z$, and p_j^r and p_j^i as $p_j = p_j^r + ip_j^i$, then we can apply the CR-transform of (6), and get the following linear system:

The matrix is

$$E = \begin{pmatrix} 1 & 0 & -x & y & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -y & -x & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & -x & y & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & -y & -x & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & -x & y \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & -y & -x \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The first two components of the solution s of the linear system

$$\begin{pmatrix} 1 & 0 & -x & y & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -y & -x & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & -x & y & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & -y & -x & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & -x & y \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & -y & -x \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} s_0^r \\ s_0^i \\ s_1^r \\ s_1^i \\ \vdots \\ s_{n-1}^r \\ s_{n-1}^i \\ s_n^r \\ s_n^i \end{pmatrix} = \begin{pmatrix} p_0^r \\ p_0^i \\ p_1^r \\ p_1^i \\ \vdots \\ p_{n-1}^r \\ p_{n-1}^i \\ p_n^r \\ p_n^i \end{pmatrix} \tag{7}$$

are equal to the real and imaginary parts of

$$p_0 + p_1 z + p_2 z^2 + \cdots + p_n z^n.$$

For instance, in the case $n = 3$, we get

$$s = \begin{pmatrix} -3\,xy^2p_3^r + x^3p_3^r - 3\,yx^2p_3^i + x^2p_2^r - 2\,xyp_2^i + xp_1^r + y^3p_3^i - y^2p_2^r - yp_1^i + p_0^r \\ -y^3p_3^r + 3\,yx^2p_3^r - 3\,y^2xp_3^i + 2\,yxp_2^r - y^2p_2^i + yp_1^r + x^3p_3^i + x^2p_2^i + xp_1^i + p_0^i \\ -y^2p_3^r + x^2p_3^r - 2\,yxp_3^i + xp_2^r - yp_2^i + p_1^r \\ x^2p_3^i + 2\,xyp_3^r - y^2p_3^i + yp_2^r + xp_2^i + p_1^i \\ xp_3^r - yp_3^i + p_2^r \\ yp_3^r + xp_3^i + p_2^i \\ p_3^r \\ p_3^i \end{pmatrix}$$

The linear system (7) is easily solved by the E-method, provided that it is diagonally dominant (see Section 4 for details on the iterations and convergence conditions). Note that the E-method does not evaluate directly the expressions given for the solution s_0. These would require at least 16+16 full multiplications, that, assuming enough multipliers, would take at least 3 consecutive multiply times. Moreover, the reduction of product terms would require a [10:2] reduction. Of course, all the interconnections are of full precision. Instead, as explained later, the complex E-method computes s_0 on 14 serial-parallel (left-to-right) multipliers, including the additions, in about one serial-parallel multiplication time. In this approach, the interconnections are digit-serial.

Now, let us turn to rational functions of a complex argument with rational coefficients (assuming the degree-0 coefficient of the denominator is 1). We wish to evaluate

$$R(z) = \frac{p_0 + p_1 z + p_2 z^2 + \cdots + p_n z^n}{1 + q_1 z + q_2 z^2 + \cdots + q_n z^n}$$

where the p_j's, the q_j's and z are complex numbers. Clearly, $R(z)$ is equal to the first component of the solution of the linear system

$$\begin{pmatrix} 1 & -z & 0 & 0 & 0 & \dots & 0 \\ q_1 & 1 & -z & 0 & 0 & \dots & 0 \\ q_2 & 0 & 1 & -z & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_n & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \times \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} = \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} \tag{8}$$

The E-method cannot directly solve the linear system (8), but it suffices to take the CR-transform of that system. Define $z = x + iy$, $p_j = p_j^r + ip_j^i$ and $q_j = q_j^r + iq_j^i$. The CR-transform results in the following linear system

$$\begin{pmatrix} 1 & 0 & -x & y & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -y & -x & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ q_1^r & -q_1^i & 1 & 0 & -x & y & 0 & 0 & \dots & 0 & 0 \\ q_1^i & q_1^r & 0 & 1 & -y & -x & 0 & 0 & \dots & 0 & 0 \\ q_2^r & -q_2^i & 0 & 0 & 1 & 0 & -x & y & \dots & 0 & 0 \\ q_2^i & q_2^r & 0 & 0 & 0 & 1 & -y & -x & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ q_n^r & -q_n^i & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\ q_n^i & q_n^r & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \times \begin{pmatrix} s_0^r \\ s_0^i \\ s_1^r \\ s_1^i \\ s_2^r \\ s_2^i \\ \vdots \\ s_n^r \\ s_n^i \end{pmatrix} = \begin{pmatrix} p_0^r \\ p_0^i \\ p_1^r \\ p_1^i \\ p_2^r \\ p_2^i \\ \vdots \\ p_n^r \\ p_n^i \end{pmatrix} \tag{9}$$

Again, that system is easily solved by the E-method, provided that it satisfies the convergence conditions (see Section 4), and

$$R(z) = s_0^r + is_0^i.$$

4 Iteration, and convergence conditions

To make the presentation simpler, we will focus on radix-2 iterations only. Adaptation to higher radices is rather straightforward. The radix-2 E-method consists in solving the n-dimensional linear system

$$Ax = P$$

by using the following basic recursion on residuals:

$$w^{(j)} = 2 \times \left[w^{(j-1)} - Ad^{(j-1)} \right] \tag{10}$$

with $w^{(0)} = [p_0, p_1, \dots, p_n]^t$, and $d^{(j)} = [d_0, d_1, \dots, d_n]^t$ where the digits $d_k^{(j)}$ are in $\{-1, 0, 1\}$. Define the number $D_k^{(j)} = d_k^{(0)}.d_k^{(1)}d_k^{(2)} \dots d_k^{(j)}$ (the $d_k^{(j)}$ are the digits of a radix-2 signed-digit representation of $D_k^{(j)}$). By induction, we easily get,

$$w^{(j)} = 2^j \left[w^{(0)} - AD^{(j-1)} \right]. \tag{11}$$

Using (11), one can show that if the residuals $|w_k^{(j)}|$ are bounded, then for all k, $D_k^{(j)}$ goes to y_k as j goes to infinity.
The problem at step j is to find a *selection function* that gives a value of the digits $d_k^{(j)}$ from

the residuals $w_k^{(j)}$ such that the values $w_k^{(j+1)}$ will remain bounded. In [3], the following selection function (a form of rounding) is proposed

$$s(x) = \begin{cases} \text{sign } x \times \lfloor |x + 1/2| \rfloor, & \text{if } |x| \leq 1 \\ \text{sign } x \times \lfloor |x| \rfloor, & otherwise, \end{cases} \tag{12}$$

and applied to the following cases:

1. $d_k^{(j)} = s(w_k^{(j)})$, i.e., the selection uses a non-redundant $w_k^{(j)}$;
2. $d_k^{(j)} = s(\hat{w}_k^{(j)})$, where $\hat{w}_k^{(j)}$ is an *approximation* to $w_k^{(j)}$ (in practice, $\hat{w}_k^{(j)}$ is deduced from a few digits of $w_k^{(j)}$ by the means of a rounding or a truncation)

Let us now see what this gives in two cases: complex polynomial evaluation and complex rational function evaluation.

4.1 Polynomial evaluation

We wish to evaluate a degree-n polynomial

$$p_n z^n + p_{n-1} z^{n-1} + \cdots + p_0$$

at the complex point $z = x + iy$, with $p_k = p_k^r + i p_k^i$. The matrix of the CR-transform, obtained in Section 3 is

$$A = \begin{pmatrix} 1 & 0 & -x & y & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -y & -x & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & -x & y & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & -y & -x & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & -x & y \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & -y & -x \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Let us slightly modify the notations w and d of iteration (10), to adapt them to the complex case. The residual vector $w^{(j)}$ will be denoted

$$w^{(j)} = [w_{0,r}^{(j)}, w_{0,i}^{(j)}, w_{1,r}^{(j)}, w_{1,i}^{(j)}, \ldots, w_{n,r}^{(j)}, w_{n,i}^{(j)}],$$

and its initial value will be given by

$$\begin{cases} w_{k,r}^{(0)} &= p_k^r \\ w_{k,i}^{(0)} &= p_k^i \end{cases}$$

The digit-vector $d^{(j)}$ will be denoted

$$d^{(j)} = [d^{(j)}_{0,r}, d^{(j)}_{0,i}, d^{(j)}_{1,r}, d^{(j)}_{1,i}, \cdots, d^{(j)}_{n,r}, d^{(j)}_{n,i}].$$

Therefore, iteration (10) becomes

- for $k = 0, \ldots, n-1$,

$$\left\{ \begin{array}{rcl} w^{(j)}_{k,r} & = & 2\left[w^{(j-1)}_{k,r} - d^{(j-1)}_{k,r} + x d^{(j-1)}_{k+1,r} - y d^{(j-1)}_{k+1,i}\right] \\ w^{(j)}_{k,i} & = & 2\left[w^{(j-1)}_{k,i} - d^{(j-1)}_{k,i} + y d^{(j-1)}_{k+1,r} + x d^{(j-1)}_{k+1,i}\right] \end{array} \right. \tag{13}$$

- for $k = n$,

$$\left\{ \begin{array}{rcl} w^{(j)}_{n,r} & = & 2\left[w^{(j-1)}_{n,r} - d^{(j-1)}_{n,r}\right] \\ w^{(j)}_{n,i} & = & 2\left[w^{(j-1)}_{n,i} - d^{(j-1)}_{n,i}\right] \end{array} \right.$$

Now, let us examine the convergence conditions. The iterations converge to the desired result if vector $w^{(j)}$ is bounded. Define constants ξ, α and Δ (with $0 \leq \Delta < 1$) such that

1. $|x| + |y| \leq \alpha$;
2. for any k between 0 and n,

$$\left\{ \begin{array}{lcl} |p^r_k| & \leq & \xi \\ |p^i_k| & \leq & \xi \\ |w^{(j)}_{k,r} - \hat{w}^{(j)}_{k,r}| & \leq & \frac{\Delta}{2} \\ |w^{(j)}_{k,i} - \hat{w}^{(j)}_{k,i}| & \leq & \frac{\Delta}{2} \end{array} \right.$$

Since $|d^{(j-1)}_{k,r} - \hat{w}^{(j-1)}_{k,r}| \leq 1/2$ and $|d^{(j-1)}_{k,i} - \hat{w}^{(j-1)}_{k,i}| \leq 1/2$, from (13) we find

$$|w^{(j)}_{k,r}| \leq 2\left(\frac{1}{2} + \frac{\Delta}{2} + \alpha\right) = 1 + \Delta + 2\alpha. \tag{14}$$

The same bound holds for $|w^{(j)}_{k,i}|$. For this bound to be feasible, we must assure that a suitable choice of $d^{(j)}_{k,r}$ and $d^{(j)}_{k,r}$ in $\{-1, 0, 1\}$ is possible. This requires that $|w^{(j)}_{k,r}|$ and $|w^{(j)}_{k,i}|$ should be less than $3/2$. This immediately gives the following condition

$$\Delta + 2\alpha \leq \frac{1}{2}. \tag{15}$$

Now, let us turn to the initial values. Since $|w^{(0)}_{k,r}|$ and $|w^{(0)}_{k,i}|$ must also be less than $3/2$, we get

$$\xi \leq \frac{3}{2}. \tag{16}$$

Consider the following example: we wish to evaluate

$$p(z) = (1+i)\,z^3 - (0.5+1.25\,i)\,z^2 + z + 1.$$

at point

$$z = \frac{1}{100} + \frac{i}{10}.$$

We assume that $\Delta = 0$ (that is, we use non-redundant residuals). We get:

- initialization:

$$w^{(0)} = [p_0^r, p_0^i, p_1^r, p_1^i, p_2^r, p_3^i, p_4^r, p_4^i]^t = [1, 0, 1, 0, -0.5, -1.25, 1, 1]^t.$$

- Step 1: from $w^{(0)}$ and the selection function, we get

$$s^{(0)} = [1, 0, 1, 0, 0, -1, 1, 1]^t,$$

  which gives

$$w^{(1)} = [0.02, 0.2, 0.2, -0.02, -1.18, -0.28, 0, 0]^t.$$

- Step 2: from $w^{(1)}$ and the selection function, we get

$$s^{(1)} = [0, 0, 0, 0, -1, 0, 0, 0]^t,$$

  which gives

$$w^{(2)} = [0.04, 0.4, 0.38, -0.24, -0.36, -0.56, 0, 0]^t.$$

- after 20 iterations, the number

$$d_{0,r}^{(0)}.d_{0,r}^{(1)}d_{0,r}^{(2)}\cdots d_{0,r}^{(20)} + i \times d_{0,i}^{(0)}.d_{0,i}^{(1)}d_{0,i}^{(2)}\cdots d_{0,i}^{(20)}$$

  is equal to

$$\frac{533789}{524288} + \frac{57727}{524288}\,i \approx 1.018121719 + 0.110105514\,i$$

  whereas the exact value of $p(z)$ is

$$p(z) = 1.018121 + 0.110106\,i$$

Exactly as in the real case, even if polynomial p and point z do not satisfy the convergence constraints, one can easily "transform" them using mere shifts, so that $p(z)$ can be computed using the E-method. Once Δ is chosen, and α is defined as $\frac{1}{4} - \Delta/2$, this is done as follows:

1. Find the smallest integer k such that $|\Re(z/2^k)| + |\Im(z/2^k)|$ should be less than α;
2. Now, $p(z) = \pi(t)$, where the degree-m coefficient of polynomial π is $2^{mk}p_m$. If at least one of the coefficients of π has the absolute value of its real or imaginary part greater than $\xi = 3/2$, then divide π by 2^ℓ, where ℓ is the smallest integer such that $\rho = \pi/2^\ell$ has the absolute value of the real and imaginary parts of its coefficients less than ξ;
3. What we actually compute using the E-method is $\rho(z/2^k)$. This result will then be multiplied by 2^ℓ (a simple left-shift) to get $p(z)$.

4.2 Rational function evaluation

We now wish to evaluate

$$R(z) = \frac{p_0 + p_1 z + p_2 z^2 + \cdots + p_n z^n}{1 + q_1 z + q_2 z^2 + \cdots + q_n z^n}$$

at the complex point $z = x + iy$, with $p_k = p_k^r + i p_k^i$ and $q_k = q_k^r + i q_k^i$. The matrix of the CR-transform obtained in Section 3 is

$$\begin{pmatrix}
1 & 0 & -x & y & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
0 & 1 & -y & -x & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\
q_1^r & -q_1^i & 1 & 0 & -x & y & 0 & 0 & \dots & 0 & 0 \\
q_1^i & q_1^r & 0 & 1 & -y & -x & 0 & 0 & \dots & 0 & 0 \\
q_2^r & -q_2^i & 0 & 0 & 1 & 0 & -x & y & \dots & 0 & 0 \\
q_2^i & q_2^r & 0 & 0 & 0 & 1 & -y & -x & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
q_n^r & -q_n^i & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \\
q_n^i & q_n^r & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 1
\end{pmatrix}$$

Therefore, iteration (10) becomes

- for $k = 0$,

$$\left\{ \begin{array}{rcl} w_{0,r}^{(j)} & = & 2\left[w_{0,r}^{(j-1)} - d_{0,r}^{(j-1)} + x d_{1,r}^{(j-1)} - y d_{1,i}^{(j-1)}\right] \\ w_{0,i}^{(j)} & = & 2\left[w_{0,i}^{(j-1)} - d_{0,i}^{(j-1)} + y d_{1,r}^{(j-1)} + x d_{1,i}^{(j-1)}\right] \end{array} \right.$$

- for $k = 1, \dots, n-1$,

$$\left\{ \begin{array}{rcl} w_{k,r}^{(j)} & = & 2\left[w_{k,r}^{(j-1)} - d_{k,r}^{(j-1)} - q_k^r d_{0,r}^{(j-1)} + q_k^i d_{0,i}^{(j-1)} + x d_{k+1,r}^{(j-1)} - y d_{k+1,i}^{(j-1)}\right] \\ w_{k,i}^{(j)} & = & 2\left[w_{k,i}^{(j-1)} - d_{k,i}^{(j-1)} - q_k^i d_{0,r}^{(j-1)} - q_k^r d_{0,i}^{(j-1)} + y d_{k+1,r}^{(j-1)} + x d_{k+1,i}^{(j-1)}\right] \end{array} \right. \tag{17}$$

- for $k = n$,

$$\left\{ \begin{array}{rcl} w_{n,r}^{(j)} & = & 2\left[w_{n,r}^{(j-1)} - d_{n,r}^{(j-1)} - q_n^r d_{0,r}^{(j-1)} + q_n^i d_{0,i}^{(j-1)}\right] \\ w_{n,i}^{(j)} & = & 2\left[w_{n,i}^{(j-1)} - d_{n,i}^{(j-1)} - q_n^i d_{0,r}^{(j-1)} - q_n^r d_{0,i}^{(j-1)}\right] \end{array} \right.$$

Similarly to the polynomial case, define constants ξ, α, and Δ (with $0 \leq \Delta < 1$) so that

$$\begin{cases} \forall k, |p_k^r| \leq \xi \\ \forall k, |p_k^i| \leq \xi \\ \forall k, |x| + |y| + |q_k^r| + |q_k^i| \leq \alpha \\ \forall k, |w_{k,r}^{(j)} - \hat{w}_{k,r}^{(j)}| \leq \frac{\Delta}{2} \\ \forall k, |w_{k,i}^{(j)} - \hat{w}_{k,i}^{(j)}| \leq \frac{\Delta}{2} \end{cases} \tag{18}$$

As in the polynomial case, we find that

$$|w_{k,r}^{(j)}|, |w_{k,i}^{(j)}| \leq 1 + \Delta + 2\alpha.$$

Again, for this bound to be valid, we must be sure that it is possible to find a suitable choice of $d_{k,r}^{(j)}$ and $d_{k,i}^{(j)}$. This requires that $|w_{k,r}^{(j)}|$ and $|w_{k,i}^{(j)}|$ should be less than $3/2$, which gives the conditions

$$\begin{cases} \Delta + 2\alpha & \leq & 1/2 \\ \xi & \leq & 3/2. \end{cases} \tag{19}$$

Unfortunately, as in the real case, there is no simple rule of transformation that allows to evaluate any rational function. In the real case, this problem is discussed in [3, 13].

4.3 Approximations with real coefficients

Many useful math functions (e.g., the elementary functions) have polynomial or rational approximations whose coefficients are real numbers. A straightforward example is the following Padé approximation for the exponential function:

$$e^z \approx \frac{1 + 1/2\, z + 1/9\, z^2 + \frac{1}{72}\, z^3 + \frac{1}{1008}\, z^4 + \frac{1}{30240}\, z^5}{1 - 1/2\, z + 1/9\, z^2 - \frac{1}{72}\, z^3 + \frac{1}{1008}\, z^4 - \frac{1}{30240}\, z^5} \tag{20}$$

Having real coefficients would not simplify a polynomial evaluation much, since the coefficients of the polynomial only appear in the initialization of the residual vector w. And yet, this would significantly simplify a rational function evaluation, since recurrence (17) becomes simpler if the terms q_k^i are equal to zero.

4.4 Implementation Aspects

In this section we discuss implementation aspects of the complex E-method in general terms. The main difference from implementation of real domain E-method is that the number of non-zero off-diagonal elements doubles: for the polynomial case to two, and for the rational case to four elements. This has two consequences. First, the bounds on the elements are smaller by a factor of two, and second, the cycle time is increased as explained later in this

section. The corresponding implementations considered for the real domain E-method are in [2, 3, 7].

A general scheme for evaluation of complex polynomials is shown in Figure 1 for $n = 3$ and the corresponding elementary unit (PEU) is illustrated in Figure 2. A bit-parallel bus transmits x and y values in a broadcast mode, while the real and imaginary coefficients p^r and p^i are loaded in separate cycles. Note that the initialization cycles could be shorter than the iteration cycles.

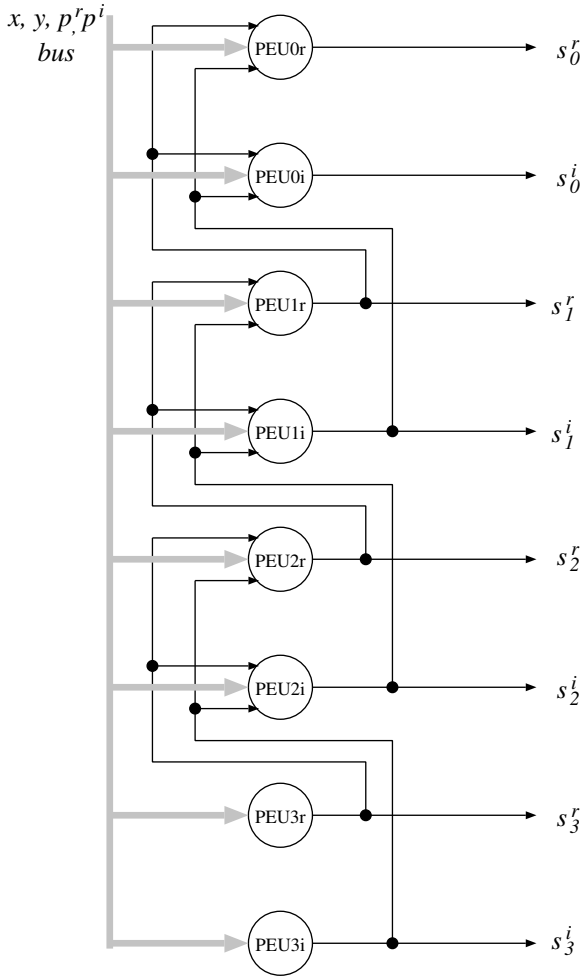

Figure 1: Overall scheme for evaluating complex polynomial of degree $n = 3$.

A block diagram of an Elementary Unit for polynomial evaluation (PEU) is shown in Figure 2.

The modules in Figure 2 are:

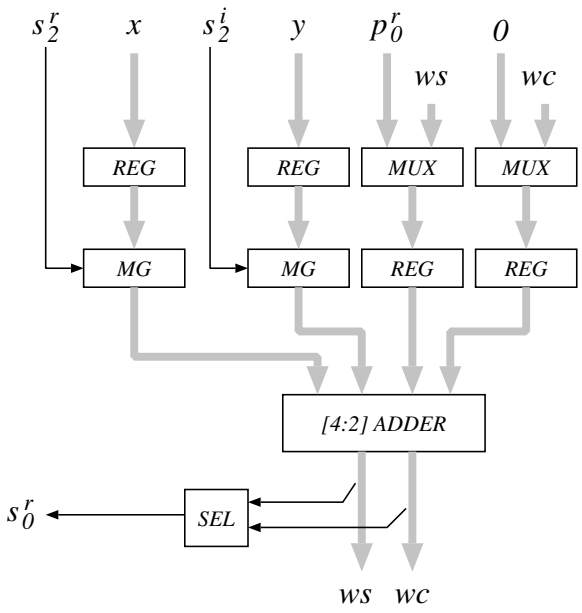

Figure 2: Block diagram of Elementary Unit for polynomials (EU_0.)

- Registers (4)
- Multiple generators MG (2), producing $\{-1, 0, 1\} \times x$ and $\{-1, 0, 1\} \times y$, with buffers
- Multiplexer MUX for initializing the residual
- A [4:2] adder
- Output digit selection SEL (a table or a gate network):

The cycle time, in terms of a full adder (complex gate) delay t, is estimated as

$$T_{PEU} = t_{BUFF} + t_{MG} + t_{SEL} + t_{[4:2]} + t_{REG} \approx (0.4 + 0.3 + 1 + 1.3 + 0.9)t = 3.9t \quad (21)$$

The cost, again in terms of area of a full adder A is estimated as

$$\begin{aligned} A_{PEU}(m) &= A_{SEL} + A_{BUFF} + A_{MG} + A_{[4:2]} + A_{REG} \\ &\approx [5 + 2 \times 0.4 + (m + 2)(2 \times 0.45 + 2.3 + 4 \times 0.6)]A \approx 6 + 6mA \end{aligned} \quad (22)$$

A general scheme for evaluation of complex rational functions is shown in Figure 3 for $n = 3$ and the corresponding elementary unit (REU) is illustrated in Figure 4. As mentioned above, a bit-parallel bus transmits x and y values in a broadcast mode, while the real and imaginary coefficients p^r, p^i, q^r and q^i are loaded in separate cycles. Note that the initialization cycles could be shorter than the iteration cycles.

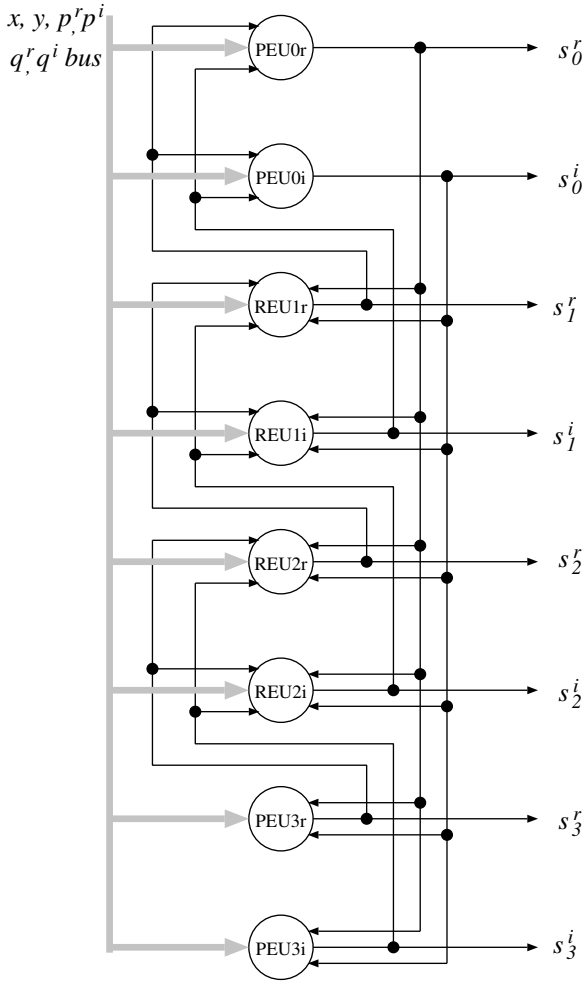

Figure 3: Overall scheme for evaluating complex rational function of degree $n = 3$.

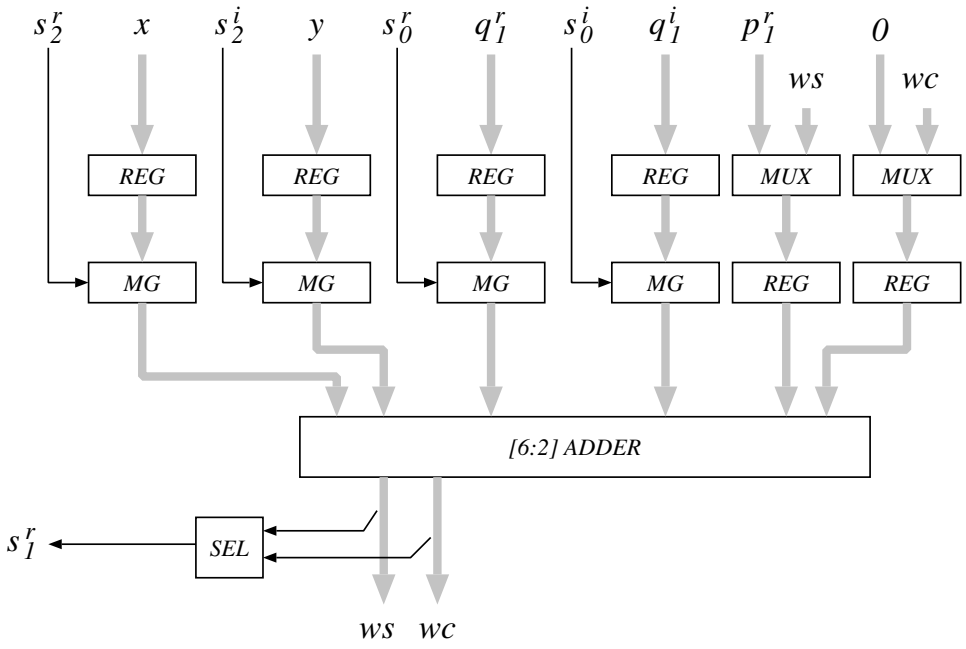

Figure 4: Block diagram of Elementary Unit for rational function evaluation (REU_0).

A block diagram of an Elementary Unit for rational function evaluation (REU) is shown in Figure 4.

The modules in Figure 4 are:

- Registers (6)
- Multiple generators MG (4), producing $\{-1, 0, 1\} \times x$ etc. with buffers
- Multiplexer MUX for initializing the residual
- A [6:2] adder
- Output digit selection SEL (a table or a gate network):

The cycle time, in terms of a full adder (complex gate) delay t, is estimated as

$$T_{REU} = t_{BUFF} + t_{MG} + t_{SEL} + t_{[6:2]} + t_{REG} \approx (0.4 + 0.3 + 1 + 2.3 + 0.9) = 4.9t \quad (23)$$

The cost, again in terms of area of a full adder A is estimated as

$$\begin{aligned} A_{REU}(m) &= A_{SEL} + A_{BUFF} + A_{MG} + A_{[6:2]} + A_{REG} \\ &\approx [5 + 2 \times 0.4 + (m+2)(4 \times 0.5 + 3.3 + 6 \times 0.6)]A \approx 20 + 7mA \end{aligned} \quad (24)$$

About Comparisons with Real E-Method

For a "general" function, there is no way of comparing the real and complex methods, since they do not apply to the same (real or complex) domain. However, for the usual elementary functions, there are schoolbook transformation rules that makes it possible to re-write a complex-valued function in terms of real-valued functions. So, let us compare both methods with such a function. Assume we wish to approximate $\exp(z)$, where $\max(|\Re(z)|, |\Im(z)|) \leq 1/2$. The Padé approximation of Eq. (20) has a degree-5 numerator and a degree-5 denominator, with real coefficients. The error of this approximation in the considered domain is less than 2.2×10^{-12}. If one wishes to reach a similar accuracy using the real E-method and the formula

$$e^{x+iy} = e^x(\cos y + i \sin y),$$

then

- with polynomial approximations, one would need a degree-9 polynomial for the exp function, a degree-8 polynomial for the cos and a degree-9 polynomial for the sin;
- with rational approximations, a (4/4)-fraction for the exp function, a (4/4)-fraction for the cos and a (5/5)-fraction for the sin, where an (n/m)-fraction is a rational fraction whose numerator has degree n, and whose denominator has degree m.

4.5 Potential Applications

For computing complex square roots with moderate (e.g., single) precision, our method can be of interest. In [6] we adapted the real digit-recurrence square-root iteration to the complex case. The basic iteration is simple, yet there is a prescaling initial step that requires a look-up in a rather big tables and a small multiplication if the required precision is large enough, the method is of much interest (the initial step can then be neglected). If this is not the case, it is much simpler to use a Padé approximation of the square-root and the complex E-method. For instance, for computing $\sqrt{1+z}$ with $\max(|\Re(z)|, |\Im(z)|) \leq 1/2$ (this domain is large enough, so that reduction to it is straightforward), then one can use the Padé approximation

$$\sqrt{1+z} \approx \frac{1 + \frac{15}{4}\,z + \frac{45}{8}\,z^2 + \frac{275}{64}\,z^3 + \frac{225}{128}\,z^4 + \frac{189}{512}\,z^5 + \frac{35}{1024}\,z^6 + \frac{15}{16384}\,z^7}{1 + \frac{13}{4}\,z + \frac{33}{8}\,z^2 + \frac{165}{64}\,z^3 + \frac{105}{128}\,z^4 + \frac{63}{512}\,z^5 + \frac{7}{1024}\,z^6 + \frac{1}{16384}\,z^7}$$

It has real coefficients only, and the error is less than 9.3×10^{-10}.

4.6 Summary

We have presented a method for solving diagonally-dominant linear systems in complex domain by a digit-recurrence algorithm. This is a generalization of the real-domain E-method. The method is particularly area/cost-effective for solving systems with sparse coefficient matrices. Specifically, the method is suitable for evaluating complex polynomials, integer powers of a complex number, and complex rational functions. The latency is roughly m cycles for m bits of precision and independent of the order of the system. This does not take into account potentially needed scaling steps. The cycle time is independent of m. We discussed the transform from real to complex numbers, the iteration and convergence conditions. The application of the method to polynomials, rational functions, and division are described. Implementation is given at a high level with estimates of the cost and latency. A detailed design and its hardware implementation with FPGAs are considered.

References

[1] K. Benmahammed, Evaluation of Complex Polynomials in One and Two Variables. Multidimensional Systems and Signal Processing, 5, 245-261, 1994.

[2] M.D. Ercegovac. *A general method for evaluation of functions and computation in a digital computer*. PhD thesis, Dept. of Computer Science, University of Illinois, Urbana-Champaign, 1975.

[3] M.D. Ercegovac. A general hardware-oriented method for evaluation of functions and computations in a digital computer. *IEEE Trans. Comp.*, C-26(7):667–680, 1977.

[4] M.D. Ercegovac and J.-M. Muller. Complex Division with Prescaling of Operands. *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pp. 293-303, 2003.

[5] M.D. Ercegovac and J.-M. Muller, Design of a complex divider. *Proc. SPIE on Advanced Signal Processing Algorithms, Architectures, and Implementations XII*, pp. 51-59, 2004.

[6] M.D. Ercegovac and J.-M. Muller. Complex Square Root with Operand Prescaling. *IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pp. 293-303, 2004.

[7] M.D. Ercegovac and T. Lang. *Digital Arithmetic*, Morgan Kaufmann Publishers - an Imprint of Elsewier Science, San Francisco, 2004.

[8] R. McIlhenny and M.D. Ercegovac. On-Line Algorithms for Complex Number Arithmetic. Proc. 32nd Asilomar Conference on Signals, Systems and Computers, pages 172-176, 1998.

[9] R. McIlhenny, *Complex Number On-line Arithmetic for Reconfigurable Hardware: Algorithms, Implementations, and Applications*, PhD Dissertation, UCLA Computer Science Department, 2002.

[10] R. McIlhenny and M.D. Ercegovac, On the Design of an On-Line Complex FIR Filter. Proc. 38th Asilomar Conference on Signals, Systems and Computers, pp. 478-482, 2004.

[11] R. McIlhenny and M. D. Ercegovac, On the Design of an On-line Complex Matrix Inversion Unit. Proc. 39th Asilomar Conference on Signals, Systems and Computers, 5 pps., 2005.

[12] R. McIlhenny and M. D. Ercegovac, On the Design of an On-line Complex Householder Transform. Proc. 40th Asilomar Conference on Signals, Systems and Computers, 5 pps., 2006.

[13] N. Brisebarre and J.-M. Muller. Functions approximable by E-fractions. 38th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, California, Nov. 2004.

[14] A.H. Nutall, Efficient Evaluation of Polynomials and Exponentials of Polynomials for Equispaced Arguments, IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-35, pp. 1486-1487, 1987.

[15] F. W. J. Olver, Error Bounds for Polynomial Evaluation and Complex Arithmetic, IMA Journal of Numerical Analysis 6, 373-379, 1986.

[16] J.H. Reif, Approximate Complex Polynomial Evaluation in Near Constant Work Per Point, STOC 97, pp. 30-39, 1997.