Convergence Acceleration and Optimal Parameter Estimation at FFT-based 2D-NILT Method

Lubomír Brančík

***Abstract* — Laplace transforms in two variables can very be useful for solving certain partial differential equations, namely those describing transient behaviour of linear dynamical systems. In practice, it is often either too difficult or even impossible to obtain corresponding originals by analytic methods. In such cases methods that enable getting original numerically have to be applied. The 2D-NILT method based on FFT, recently published and verified in Matlab language, seems to be well usable. Its main advantage lies in high speed of calculation, however, it is necessary to connect it always with proper technique of acceleration of the convergence to achieve required accuracy. It was shown either the epsilon or the quotient-difference algorithms are very suitable for this purpose. In the paper an error analysis, comparison and evaluation of the optimal NILT parameters are newly presented.**

I. Introduction

THE paper closelly follows the author's previous works devoted to ways of getting originals of two-dimensional Laplace transforms numerically. Namely, the 2D-NILT method is based on the summation of the infinite two-dimensional complex Fourier series which approximate the definition Bromwich integral, in part evaluated by the FFT [1]. To ensure a desired accuracy of results both the ε–algorithm of Wynn [2] and the quotient–difference algorithm of Rutishauser [3] were partially verified at the process of acceleration of the convergence of the infinite series [4,5]. As is shown e. g. in [2] both these algorithms should theoretically lead to the same results, more precisely they are both equivalent to the Padé approximation method in case that power series are considered. However, in practice, due to the different ways of their numeration they lead to slightly different results. Herein both methods are compared in light of their efficiency and numerical stability. Moreover the error analysis is performed leading to the evaluation of optimal NILT parameters, namely to the number of terms for the FFT and the accelerating algorithms. The computations have been performed using the Matlab language environment, for many representative transforms of rational, irrational and transcendental patterns known originals [6].

II. Approximate Formula and Error Analysis

The two-dimensional Laplace transform of the real function of two variables $f(t_1,t_2)$, $t_1 \geq 0, t_2 \geq 0$, fits [6]

$$F(s_1,s_2) = \int_0^\infty\int_0^\infty f(t_1,t_2)e^{-s_1t_1-s_2t_2}dt_1dt_2 \ . \quad (1)$$

Author is with the Institute of Theoretical and Experimental Electrical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic, e-mail: *brancik@feec.vutbr.cz*

Under the basic premise $|f(t_1,t_2)| < Me^{\alpha_1t_1+\alpha_2t_2}$, M , α_1, α_2 as positive real constants, and $F(s_1,s_2)$ defined on a region $\{[s_1,s_2] \in \mathbf{C}^2 : \mathrm{Re}[s_1] > \alpha_1 \wedge \mathrm{Re}[s_2] > \alpha_2\}$, the original function $f(t_1,t_2)$ is given by

$$f(t_1,t_2) = -\frac{1}{4\pi^2}\int_{c_1-j\infty}^{c_1+j\infty}\int_{c_2-j\infty}^{c_2+j\infty}F(s_1,s_2)e^{s_1t_1+s_2t_2}ds_1ds_2 \ . \quad (2)$$

The double integral has to be evaluated numerically. Hereafter it will be shown that the simplest way of the numerical integration leads to the approximate formula at which the relative error is theoretically adjustable. Substituting $s_i = c_i + j\omega_i$, $i = 1,2$, in (2), and applying the rectangular rule of the integration, with generalized frequency steps $\Omega_i = 2\pi/\tau_i$, $i = 1,2$, we can get the approximate formula in the form

$$\tilde{f}(t_1,t_2) = \frac{e^{c_1t_1+c_2t_2}}{\tau_1\tau_2}\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}F(c_1+jn_1\Omega_1,c_2+jn_2\Omega_2)e^{jn_1\Omega_1t_1+jn_2\Omega_2t_2} \ . \quad (3)$$

Reflecting (1) and $s_i = c_i + jn_i\Omega_i$, $i = 1,2$, the result is

$$F(c_1+jn_1\Omega_1,c_2+jn_2\Omega_2) = \int_0^\infty\int_0^\infty f(t_1,t_2)e^{-c_1t_1-c_2t_2}e^{-jn_1\Omega_1t_1-jn_2\Omega_2t_2}dt_1dt_2 \ . \quad (4)$$

Further, divide integration ranges into infinite numbers of intervals τ_i, $i = 1,2$. We get from (4)

$$F(c_1+jn_1\Omega_1,c_2+jn_2\Omega_2) = \sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}\int_{l_1\tau_1}^{(l_1+1)\tau_1}\int_{l_2\tau_2}^{(l_2+1)\tau_2}g(t_1,t_2)e^{-jn_1\Omega_1t_1-jn_2\Omega_2t_2}dt_1dt_2 \ , \quad (5)$$

where $g(t_1,t_2)$ is the exponentially dumped original function. Now for each $[t_1,t_2] \in \langle l_1,l_1+1)\tau_1 \times \langle l_2,l_2+1)\tau_2$ a new function

$$g_{l_1}^{l_2}(t_1,t_2) = f(t_1,t_2)e^{-c_1t_1-c_2t_2} \quad (6)$$

can be defined up to be the periodical function on the whole 2D interval, with the periods τ_1 and τ_2. In this case (5) can be rewritten into

$$F(c_1+jn_1\Omega_1,c_2+jn_2\Omega_2) = \tau_1\tau_2\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}c_{l_1,n_1}^{l_2,n_2} \ , \quad (7)$$

where the terms

$$c_{l_1,n_1}^{l_2,n_2} = \frac{1}{\tau_1\tau_2}\int_{l_1\tau_1}^{(l_1+1)\tau_1}\int_{l_2\tau_2}^{(l_2+1)\tau_2}g_{l_1}^{l_2}(t_1,t_2)e^{-jn_1\Omega_1t_1-jn_2\Omega_2t_2}dt_1dt_2 \quad (8)$$

act as the coefficients of the 2D complex Fourier series of the function (9), namely

$$g_{l_1}^{l_2}(t_1,t_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}c_{l_1,n_1}^{l_2,n_2}\,e^{jn_1\Omega_1 t_1+jn_2\Omega_2 t_2}\ . \quad (9)$$

After substituting (7) into (3) we have

$$\widetilde{g}(t_1,t_2)=\widetilde{f}(t_1,t_2)e^{-c_1t_1-c_2t_2}=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}\left(\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}c_{l_1,n_1}^{l_2,n_2}\right)e^{jn_1\Omega_1t_1+jn_2\Omega_2t_2}\ . \quad (10)$$

Now interchanging the summations and considering (9) the result is

$$\widetilde{g}(t_1,t_2)=\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}g_{l_1}^{l_2}(t_1,t_2)\ . \quad (11)$$

We found that the exponentially dumped approximate original function can be also expressed as the infinite double sum of newly defined periodical functions (6). Now take the 2D interval of our interest, in which the original $f(t_1,t_2)$ is required, as $[t_1,t_2]\in\langle 0,\tau_1\rangle\times\langle 0,\tau_2\rangle$. Then, considering (6) in its alternative form

$$g_{l_1}^{l_2}(t_1,t_2)=f(l_1\tau_1+t_1,l_2\tau_2+t_2)e^{-c_1(l_1\tau_1+t_1)-c_2(l_2\tau_2+t_2)}\ , \quad (12)$$

the function $f(t_1,t_2)$ occurs in the expression

$$g_0^0(t_1,t_2)=f(t_1,t_2)e^{-c_1t_1-c_2t_2}\ . \quad (13)$$

Further, arranging (11) into the form

$$\widetilde{g}(t_1,t_2)=g_0^0(t_1,t_2)+\sum_{l_1=1}^{\infty}g_{l_1}^0(t_1,t_2)+\sum_{l_2=1}^{\infty}g_0^{l_2}(t_1,t_2)+\sum_{l_1=1}^{\infty}\sum_{l_2=1}^{\infty}g_{l_1}^{l_2}(t_1,t_2),\ (14)$$

substituting (12), (13), and considering (10), we have

$$\widetilde{f}(t_1,t_2)=f(t_1,t_2)+\sum_{l_1=1}^{\infty}f(l_1\tau_1+t_1,t_2)e^{-c_1l_1\tau_1}+\sum_{l_2=1}^{\infty}f(t_1,l_2\tau_2+t_2)e^{-c_2l_2\tau_2}$$
$$+\sum_{l_1=1}^{\infty}\sum_{l_2=1}^{\infty}f(l_1\tau_1+t_1,l_2\tau_2+t_2)e^{-c_1l_1\tau_1-c_2l_2\tau_2}. \quad (15)$$

By this we succeeded to find the absolute error as

$$\varepsilon(t_1,t_2)=\widetilde{f}(t_1,t_2)-f(t_1,t_2)\ . \quad (16)$$

Now we can determine a limit absolute error

$$\varepsilon_M(t_1,t_2)\geq\varepsilon(t_1,t_2) \quad (17)$$

as follows.

Take into account the basic assumption for the validity of (2), namely $\left|f(t_1,t_2)\right|<Me^{\alpha_1t_1+\alpha_2t_2}$, and substitute into (15). We get the limit absolute error as

$$\varepsilon_M(t_1,t_2)=Me^{\alpha_1t_1+\alpha_2t_2}\left[\sum_{l_1=1}^{\infty}e^{l_1\tau_1(\alpha_1-c_1)}+\sum_{l_2=1}^{\infty}e^{l_2\tau_2(\alpha_2-c_2)}+\sum_{l_1=1}^{\infty}e^{l_1\tau_1(\alpha_1-c_1)}\sum_{l_2=1}^{\infty}e^{l_2\tau_2(\alpha_2-c_2)}\right]$$
$$=Me^{\alpha_1t_1+\alpha_2t_2}\left[\frac{1}{e^{\tau_1(c_1-\alpha_1)}-1}+\frac{1}{e^{\tau_2(c_2-\alpha_2)}-1}+\frac{1}{e^{\tau_1(c_1-\alpha_1)}-1}\cdot\frac{1}{e^{\tau_2(c_2-\alpha_2)}-1}\right]\ , \quad (18)$$

where the basic assumptions $c_1>\alpha_1\wedge c_2>\alpha_2$ and the formula for the sum of the infinite geometric series were applied. Besides, from the last equation, the limit relative error is defined as

$$\varepsilon_{Mr}(t_1,t_2)=\frac{\varepsilon_M(t_1,t_2)}{Me^{\alpha_1t_1+\alpha_2t_2}}=\varepsilon_{Mr}\ . \quad (19)$$

We can see, this error has the constant value on all the interval. From (18) and (19) it is possible to derive

$$e^{-\tau_1(c_1-\alpha_1)}+e^{-\tau_2(c_2-\alpha_2)}-e^{-\tau_1(c_1-\alpha_1)}\cdot e^{-\tau_2(c_2-\alpha_2)}=\frac{\varepsilon_{Mr}}{1+\varepsilon_{Mr}}\ . \quad (20)$$

Let both terms influence the limite relative error in the same way, which is valid for $\tau_1(c_1-\alpha_1)=\tau_2(c_2-\alpha_2)$. Then it is possible to find relations between the above coefficients as

$$c_i=\alpha_i-\frac{1}{\tau_i}\ln\left(1-\frac{1}{\sqrt{1+\varepsilon_{Mr}}}\right),\quad i=1,2\ , \quad (21)$$

when, for values $\varepsilon_{Mr}<<1$, it is approximately valid

$$c_i\approx\alpha_i-\frac{1}{\tau_i}\ln\frac{\varepsilon_{Mr}}{2}\ ,\quad i=1,2\ . \quad (22)$$

The equations (21) or (22) enable choosing paths of integration from the required limit relative errors.

III. NUMERATION AND CONVERGENCE ACCELERATION

When supposing real valued originals $f(t_1,t_2)$ then the equation (3) can be decomposed into the form

$$\widetilde{f}(t_1,t_2)=\frac{E_{c_1,c_2}(t_1,t_2)}{\tau_1\tau_2}\Big\{2\mathrm{Re}\Big[\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}F_{-n_1,-n_2}E_{-n_1,-n_2}(t_1,t_2)+ \quad (23)$$
$$+\sum_{n_1=0}^{\infty}\Big(\sum_{n_2=0}^{\infty}F_{-n_1,n_2}E_{n_2}(t_2)\Big)E_{-n_1}(t_1)-\sum_{n_1=0}^{\infty}F_{-n_1,0}E_{-n_1}(t_1)-\sum_{n_2=0}^{\infty}F_{0,-n_2}E_{-n_2}(t_2)\Big]+F_{0,0}\Big\}\ .$$

where the designations were introduced as follows

$$F_{n_1,n_2}=F(c_1+jn_1\Omega_1,c_2+jn_2\Omega_2)\ , \quad (24)$$

$$E_{n_1,n_2}(t_1,t_2)=e^{j(n_1\Omega_1t_1+n_2\Omega_2t_2)}=E_{n_1}(t_1)E_{n_2}(t_2)\ , \quad (25)$$

$$E_{c_1,c_2}(t_1,t_2)=e^{c_1t_1+c_2t_2}=E_{c_1}(t_1)E_{c_2}(t_2)\ , \quad (26)$$

and equalities $F_{-n_1,-n_2}=F^*_{n_1,n_2}$, $E_{-n_1,-n_2}=E^*_{n_1,n_2}$, $E_{-n_1}=E^*_{n_1}$, and $E_{-n_2}=E^*_{n_2}$ were taken into account.

In practice it is usually necessary to find a solution on the whole 2D region, and the computation is done on some chosen grid of discrete points. Expressing the original at $t_{k_i}=k_iT_i$, $k_i=0,1,\cdots,N_i-1$, $i=1,2$, where T_i act as the sampling periods in the original domain, the approximate formula matching (23) in the discrete form $\widetilde{f}^{k_1,k_2}=\widetilde{f}(k_1T_1,k_2T_2)$ is equal to

$$\widetilde{f}^{k_1,k_2}=\frac{E_{c_1,c_2}^{k_1,k_2}}{\tau_1\tau_2}\Big\{2\mathrm{Re}\Big[\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty}F_{-n_1,-n_2}E_{-n_1,-n_2}^{k_1,k_2}+ \quad (27)$$
$$+\sum_{n_1=0}^{\infty}\Big(\sum_{n_2=0}^{\infty}F_{-n_1,n_2}E_{n_2}^{k_2}\Big)E_{-n_1}^{k_1}-\sum_{n_1=0}^{\infty}F_{-n_1,0}E_{-n_1}^{k_1}-\sum_{n_2=0}^{\infty}F_{0,-n_2}E_{-n_2}^{k_2}\Big]+F_{0,0}\Big\}\ ,$$

where designations, matching (25) and (26), look like

$$E_{n_1,n_2}^{k_1,k_2}=e^{j(k_1T_1n_1\Omega_1+k_2T_2n_2\Omega_2)}=E_{n_1}^{k_1}E_{n_2}^{k_2}\ , \quad (28)$$

$$E_{c_1,c_2}^{k_1,k_2}=e^{c_1k_1T_1+c_2k_2T_2}=E_{c_1}^{k_1}E_{c_2}^{k_2}\ . \quad (29)$$

As it has followed from the error analysis the relative error is predictable on the interval $\mathrm{O}_{\mathrm{err}} = \langle 0,\tau_1\rangle \times \langle 0,\tau_2\rangle$, while (27) leads to $\mathrm{O}_{\max} = \langle 0,(N_1-1)T_1\rangle \times \langle 0,(N_2-1)T_2\rangle$. To fulfill the necessary condition, that is $\mathrm{O}_{\max} \subset \mathrm{O}_{\mathrm{err}}$, the variables τ_i are chosen appropriately as $\tau_i = N_i T_i$, $i = 1,2$. In practical calculations, to have some margins, we choose the interval as $\mathrm{O}_{\mathrm{calc}} = \langle 0,t_{1\max}\rangle \times \langle 0,t_{2\max}\rangle$, where $t_{i\max} = (M_i - 1)T_i$, and $M_i = N_i/2$, $i = 1,2$.

It is clear that to come near to the predicted relative error the infinite sums in (27) must be evaluated as much precisely as possible. In practice, of course, only certain finite numbers of terms can be summed which evidently leads to additional truncation errors. These can be suppressed by means of a proper chosen method for accelerating the convergence of infinite series. Due to the exponential terms (28) the infinite series in (27) can be decomposed in the power series while denoting $z_i = e^{jk_iT_i\Omega_i}$, $i = 1,2$. It is known that just for such the power series either the epsilon or the quotient-difference algorithms are very suitable to be used for the accelerating process [2,3]. Theoretically, both methods lead to the same results, more precisely they are equivalent to usage of the Padé approximation method. However, due to the different paths of their numeration they can lead to somewhat different results as round-off errors can play their role.

At first, to accelerate calculation the first $N_i = 2^{m_i}$, m_i integer, $i = 1,2$, terms are summed using the FFT, and subsequently residual infinite series are undergone to the accelerating algorithm. In practice, of course, again only a finit number of terms can be considered for this process, let us say $2P+1$. Thus we can think the residual finite sum as

$$S(z,P) = \sum_{n=0}^{2P} G_n z^n , \qquad (30)$$

however, the value $S(z,\infty) = \lim_{P\to\infty} S(z,P)$ is what we want to know. Applying a method for accelerating convergence of the series a relatively small P can be chosen to get a result coming near to the $S(z,\infty)$.

$$\begin{array}{cccccc}
 & \varepsilon_0^{(0)} & & & & \\
\varepsilon_{-1}^{(1)} & & \varepsilon_1^{(0)} & & & \\
 & \varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & & \\
\varepsilon_{-1}^{(2)} & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & \\
 & \varepsilon_0^{(2)} & & \varepsilon_2^{(1)} & & \varepsilon_4^{(0)} \\
\varepsilon_{-1}^{(3)} & & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & \\
 & \varepsilon_0^{(3)} & & \varepsilon_2^{(2)} & & \\
\varepsilon_{-1}^{(4)} & & \varepsilon_1^{(3)} & & & \\
 & \varepsilon_0^{(4)} & & & &
\end{array}
\qquad
\begin{array}{ccccc}
e_0^{(0)} & & & & \\
 & q_1^{(0)} & & & \\
e_0^{(1)} & & e_1^{(0)} & & \\
 & q_1^{(1)} & & q_2^{(0)} & \\
e_0^{(2)} & & e_1^{(1)} & & e_2^{(0)} \\
 & q_1^{(2)} & & q_2^{(1)} & \\
e_0^{(3)} & & e_1^{(2)} & & \\
 & q_1^{(3)} & & & \\
e_0^{(4)} & & & &
\end{array}$$

a) b)

Fig. 1. The epsilon and q-d algorithm diagrams

The ε-algorithm can shortly be explained by the diagram in Fig.1a, the q-d algorithm in Fig.1b (valid exactly for $P = 2$) [2,3].

A. Epsilon algorithm

The first column is formed with

$$\varepsilon_{-1}^{(i)} = 0 , \quad i = 1, \cdots ,2P , \qquad (31)$$

the second one with partial sums

$$\varepsilon_0^{(i)} = \varepsilon_0^{(i-1)} + G_i z^i , \quad i = 1, \cdots ,2P , \qquad (32)$$

when starting with $\varepsilon_0^{(0)} = G_0$,

and the columns for $r = 1, \cdots ,2P$ are created as

$$\varepsilon_r^{(i)} = \varepsilon_{r-2}^{(i+1)} + [\varepsilon_{r-1}^{(i+1)} - \varepsilon_{r-1}^{(i)}]^{-1} , \quad i = 0, \cdots ,2P - r . \qquad (33)$$

The sequence of serial approximations $\varepsilon_0^{(0)}, \varepsilon_2^{(0)}, \varepsilon_4^{(0)},\cdots$ converges usually much more quickly than the original sequence of the partial sums (32). Thus, starting with $2P+1$ partial sums, the $\varepsilon_{2P}^{(0)}$ term is the wanted result.

B) Quotient-difference algorithm

The result is obtained through a continued fraction

$$v(z,P) = d_0 / (1 + d_1 z / (1 + \cdots + d_{2P} z)) , \qquad (34)$$

where its coefficients are done along Fig.1b as follows. The first two columns are formed as

$$e_0^{(i)} = 0 , \quad i = 0,\cdots,2P , \qquad (35)$$

$$q_1^{(i)} = G_{i+1}/G_i , \quad i = 0,\cdots,2P-1 , \qquad (36)$$

and the successive columns are given by the rules:

for $r = 1,\cdots,P$,

$$e_r^{(i)} = q_r^{(i+1)} - q_r^{(i)} + e_{r-1}^{(i+1)} , \quad i = 0,\cdots,2P-2r , \qquad (37)$$

for $r = 2,\cdots,P$,

$$q_r^{(i)} = q_{r-1}^{(i+1)} e_{r-1}^{(i+1)} / e_{r-1}^{(i)} , \quad i = 0,\cdots,2P-2r-1 . \qquad (38)$$

Finally the coefficients d_n, $n = 0,\cdots,2P$, are given by

$$d_0 = F_0 ,\ d_{2m-1} = -q_m^{(0)} ,\ d_{2m} = -e_m^{(0)} ,\ m = 1,\cdots,P . \qquad (39)$$

In contrast to the epsilon algorithm, the quotient-difference algorithm does not require the repetitional assesment of these coefficients for each new variable z, which results in slightly faster computation.

IV. Optimal parameter estimation

In this section it will be experimentally determined how to choose the 2D-NILT formula parameters, namely numbers of terms for the FFT (marked by *N*) and additional terms for the accelerating algorithm (marked by *P*), so that (22) could be applied correctly. For this purpose the root-mean-square error

$$L_2 = \sqrt{\sum_{i=1}^{64}\sum_{j=1}^{64} [f(x_i,y_j) - \bar{f}(x_i,y_j)]^2 / (64\times 64)} \qquad (40)$$

was introduced to show an overall error quantitatively, handling the grid of 64 x 64 points. About 50 various 2D rational, irrational and transcendental transforms taken from [6] have been tested, see examples in Fig.2.

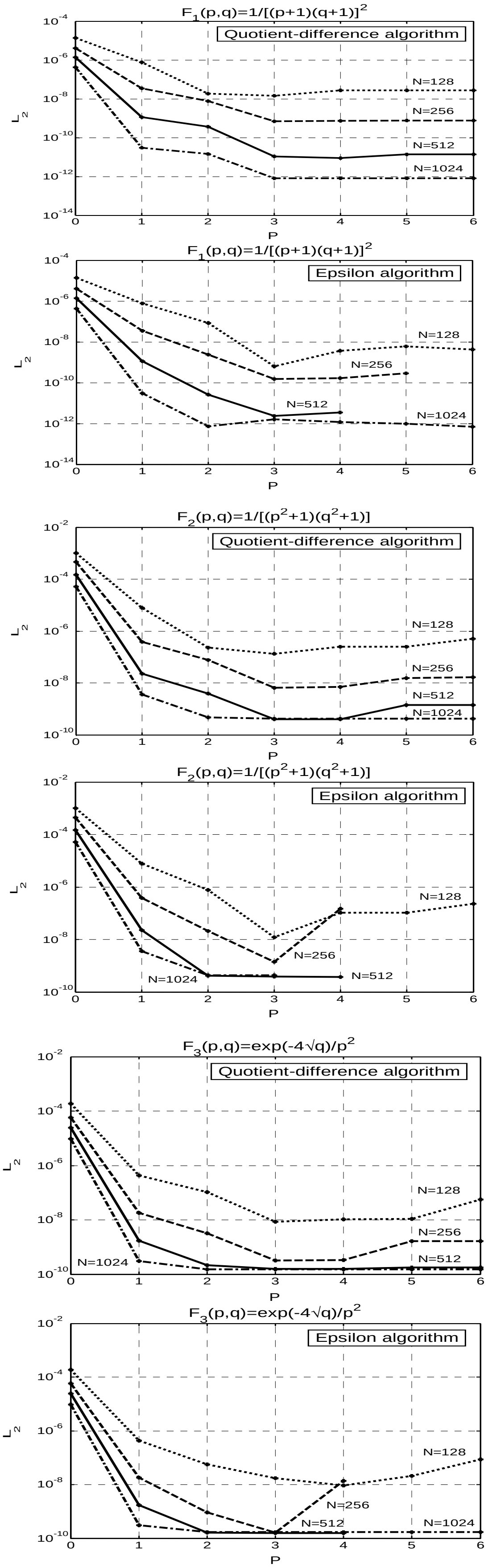

Fig. 2. Experimental 2D-NILT parameter estimation

As is obvious without using accelerating algorithm (the case $P=0$) the error cannot be suppressed under approximately 10^{-5}. To suppress the error significantly the accelerating algorithm is essential. From the other hand it seems that there is some optimal number of terms to be used, likely $P = 3$. From this number the error is often increasing or an instability can arise. The most probably round-off errors start to play their role. The q-d algorithm shows better numerical stability compared to the ε-algorithm while their efficiencies are roughly equivalent. It is due to the fact that in (33) there is often necessary to divide by a difference of two big close numbers, while it does not exist at the q-d algorithm. Generally, it seems for routine purposes the number of terms for the FFT is sufficient $N = 512$.

V. Conclusions

The comparison were also done with other recently published 2D-NILT method, namely based on the fast Hartley transform computations [7]. The appropriate Laplace transforms under testing in Fig. 2 were chosen just on this account. The results taken up from this paper are shown in Tab.1.

TABLE I. RMS ERRORS OBTAINED IN FHT METHOD [7]

Original	$\bar{f}_1(t_1,t_2)$	$\bar{f}_2(t_1,t_2)$	$\bar{f}_3(t_1,t_2)$
L_2	$2.81\cdot10^{-7}$	$1.75\cdot10^{-5}$	$1.10\cdot10^{-6}$

As we can see RMS errors are about 4 orders bigger in average when comparing with the discussed 2D-NILT method, while choosing above approved parameters. The relevant program procedure has been developed in Matlab language. Besides its basic version, see e.g. [5], a generalized one was made to enable solving tasks as in [8], where 2D vector transforms are handled with.

References

[1] L. Brančík, J. Valsa, “A Fast Computing Method of Numerical Inversion of Two-Dimensional Laplace Transforms Using FFT Algorithms“, in *Proc. of the SSCC'98*, Vol. II, Durban, South Africa, September 1998, p. 102-106.

[2] J. R. Macdonald, ”Accelerated Convergence, Divergence, Iteration, Extrapolation, and Curve Fitting”, J. Appl. Phys., 10, 1964, p. 3034-3041.

[3] P. Henrici, “Quotient-difference algorithms“, Mathematical Methods for Digital Computers, Vol. 2, A. Ralston and H. S. Wilf, eds., John Wiley, NewYork, 1967, p. 37-62.

[4] L. Brančík, “An Improvement of FFT-based Numerical Inversion of Two-Dimensional Laplace Transforms by Means of ε–algorithm“, in *Proc. of the IEEE ISCAS'2000*, Geneva, Switzerland, May 2000, p. 581-584.

[5] L. Brančík, “Numerical Inversion of Two-Dimensional Laplace Transforms Based on FFT and Quotient-Difference Algorithm”, in *Proc. of the ICFS‘2002*, Tokyo, Japan, March 2002, p. 13-15 – 13-20.

[6] V. A. Ditkin, A. P. Prudnikov, “Operational Calculus in Two Variables and its Applications”, New York, Pergamon 1962.

[7] C. Hwang, M. J. Lu, “Numerical Inversion of 2-D Laplace Transforms by Fast Hartley Transform Computations“, Journal of the Franklin Institute, 336, 1999, p. 955-972.

[8] L. Brančík, “Improved Method of Numerical Inversion of Two-Dimensional Laplace Transforms for Dynamical Systems Simulation,” in *Proc. the 9th IEEE ICECS'2002*, Dubrovnik, Croatia, September 2002, p. 385 – 388.

Acknowledgements

This work was supported by the grant GAČR No. 102/03/0241 and realized within the scope of the Research programmes MSM 262200011 and MSM 262200022.