Applications of Fourier Series Technique to Inverse Laplace Transform

AUTHOR(S):

ICHIKAWA, Satoshi; KISHIMA, Akira

CITATION:

ICHIKAWA, Satoshi ...[et al]. Applications of Fourier Series Technique to Inverse Laplace Transform. Memoirs of the Faculty of Engineering, Kyoto University 1972, 34(1): 53-67

ISSUE DATE:

1972-01

URL:

http://hdl.handle.net/2433/280874

Applications of Fourier Series Technique to Inverse Laplace Transform

By

Satoshi ICHIKAWA and Akira KISHIMA

(Received September 27, 1971)

This paper describes a method of determining the inverse Laplace transform numerically by applying the Fourier series technique to the matric operational functions. The basis of the method is that by choosing the contour of integration, the inverse Laplace transform is converted into the Fourier transform and it is approximated by a certain Fourier series. In this way numerical Laplace inversion is given and the error introduced can be made as small as desired. Furthermore, Fast Fourier Transform method is applied to the method to reduce the computational time.

The method has the advantage of needing little programming effort in digital computations and is useful in numerical analysis of systems.

Computational algorithms and some numerical examples are given to show usefulness of the method.

1. Introduction

In the previous papers[1),2)] we have reported matric operational calculus based on the Mikusinsky's method to analyse the linear time-invariant physical systems. Then in the engineering applications it is important to get the numerical solutions when the operational solutions are known. In using the above method, we can get the operational solutions easily, but it is pretty difficult to get time solutions because of the restriction of non-commutativity of matrices in multiplications.

In this paper we shall show the numerical inversion method based on the Laplace transform. The Laplace transform is one of the integral transforms, therefore, which restricts the range of applicability to the functions in which the integral transform is convergent. But when it is convergent, relation between the Laplace transform method and the Mikusinsky's method is mathematically isomorphism and in formal calculations they are treated in the same way[3)].

The main purpose of this paper is to get the numerical solutions directly when the matric operational functions are known. For this purpose we shall choose the contour over which the inverse Laplace transform may be evaluated suitably and

* Department of Electrical Engineering, II

convert it into the Fourier transform which may be approximated by a certain Fourier series[4),5)]. And then applying the method to matric operational functions we can get the numerical solutions directly with desired accuracy. This method is suitable for the numerial calculation using digital computers because the main operations are simple repetitive calculations and need little programming effort. Furthermore, to reduce the computational time, Fast Fourier Transform method[6),7),8)] is applied to the numerical inversion. Computational algorithms and some numerical examples will be given to show the usefulness of the method.

2. Numerical Inversion Method

2.1 Principle of analysis[4),5)].

When a matrix function $\boldsymbol{x}$(t) is given, the Laplace transform and its inversion formula are defined as follows:

$$\boldsymbol{X}(s) = \int_0^{\infty} e^{st}\, \boldsymbol{x}(t)dt \tag{2.1}$$

$$\boldsymbol{x}(t) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} e^{st}\, \boldsymbol{X}(s)ds \tag{2.2}$$

where $s=s\mathbf{1}$($\mathbf{1}$: unit matrix) and $a>0$ is arbitrary but must be chosen so that it is greater than the real parts of all the singularities of $\boldsymbol{X}(s)$.

In engineering applications, it is important to get the numerical values of $\boldsymbol{x}(t)$ when its Laplace transform $\boldsymbol{X}(s)$ is known.

For this purpose, as we shall show below, we will convert the Laplace transform into the Fourier transform which may be approximated arbitrarily close by a certain finite Fourier series and error introduced can be made small enough to get the required accuracy. Therefore, this method will permit good numerical values of $\boldsymbol{x}(t)$ over a finite interval of interest from a knowlege of its Laplace transform $\boldsymbol{X}(s)$.

Then we shall present the method. Putting $s=a+i\omega$ the Laplace transform (2.1) and (2.2) are converted into the Fourier transform as follows:

$$\boldsymbol{X}(a+i\omega) = \int_0^{\infty} \boldsymbol{x}(t)e^{-at}\, e^{-i\omega t}\, dt \tag{2.3}$$

$$\boldsymbol{x}(t) = \frac{e^{at}}{2\pi}\int_{-\infty}^{\infty} \boldsymbol{X}(a+i\omega)\, e^{i\omega t}\, d\omega\,. \tag{2.4}$$

Next we shall introduce the following periodic matrix functions:

$$\left.\begin{aligned} &\boldsymbol{x}_n(t) = \boldsymbol{h}(t + 2n\,T),\ \boldsymbol{x}_n(t) = \boldsymbol{x}_n(t + 2m\,T) \\ &n, m = 0, 1, 2, \ldots\ldots,\ 0 \leq t \leq 2T \end{aligned}\right\} \tag{2.5}$$

where $\boldsymbol{h}(t) = \boldsymbol{x}(t)e^{-at}$.

The Fourier series representation of each $\boldsymbol{x}_n(t)$ is given by

$$\boldsymbol{x}_n(t) = \sum_{k=-\infty}^{\infty} \boldsymbol{C}_{n,\,k}\, e^{ik(\pi/T)t} \tag{2.6}$$

and coefficients $\boldsymbol{C}_{n,k}$'s are given by

$$\boldsymbol{C}_{n,k} = \frac{1}{2T}\int_0^{2T} \boldsymbol{x}_n(t)\, e^{-ik(\pi/T)}\, dt \tag{2.7}$$

Furthermore, we shall consider the following infinite matrix series

$$\boldsymbol{x}_p(t) = e^{at} \sum_{n=0}^{\infty} \boldsymbol{x}_n(t)\ . \tag{2.8}$$

By using (2.5), (2.6) and (2.7) we have

$$\boldsymbol{x}_p(t) = \frac{e^{at}}{2T} \sum_{k=-\infty}^{\infty} \left(\int_0^{\infty} \boldsymbol{x}(t) e^{-at}\, e^{-ik(\pi/T)t}\, dt\right) e^{ik(\pi/T)t}\ .$$

The integral in the bracket is reduced to $\boldsymbol{X}(a+ik\,\pi/T)$ by using (2.3), then we have

$$\boldsymbol{x}_p(t) = \frac{e^{at}}{2T} \sum_{k=-\infty}^{\infty} \boldsymbol{X}(a + ik\,\pi/T)\, e^{ik(\pi/T)t}\ . \tag{2.9}$$

The integral (2.4) can be approximated by

$$\boldsymbol{x}(t) \simeq \frac{e^{at}}{2\pi} \sum_{k=-\infty}^{\infty} \boldsymbol{X}(a + ik\Delta\omega)\, e^{ik\Delta\omega t}\, \Delta\omega$$

and putting $\Delta\omega = \pi/T$, we have

$$\boldsymbol{x}(t) \simeq \frac{e^{at}}{2T} \sum_{k=-\infty}^{\infty} \boldsymbol{X}(a + ik\,\pi/T)\, e^{ik(\pi/T)t} = \boldsymbol{x}_p(t)\ . \tag{2.10}$$

More precisely we have

$$\left.\begin{aligned} \boldsymbol{x}_p(t) &= e^{at} \sum_{n=0}^{\infty} \boldsymbol{x}_n(t) = e^{at} \sum_{n=0}^{\infty} e^{-a(t+2nT)}\, \boldsymbol{x}(t + 2nT) \\ &= \boldsymbol{x}(t) + \sum_{n=1}^{\infty} e^{-2anT}\, \boldsymbol{x}(t + 2nT) \\ &\quad 0 \leq t < 2T \end{aligned}\right\} \tag{2.11}$$

Considering the relation

$$\boldsymbol{X}(a + ik\pi/T)\, e^{ik(\pi/T)t} + \boldsymbol{X}(a - ik\pi/T)\, e^{-ik(\pi/T)t}$$
$$= 2R_e\,[\boldsymbol{X}(a + ik\pi/T)\, e^{ik(\pi/T)t}]$$

and truncating the infinite series by K terms we have the following numerical inversion formula

$$\boldsymbol{x}(t) \simeq \frac{e^{at}}{T}\left[\frac{1}{2}\boldsymbol{X}(a) + R_e\sum_{k=1}^{K}\boldsymbol{X}(a + ik\pi/T)\, e^{ik(\pi/T)t}\right] \tag{2.12}$$

and from this equation we can get numerical value of $\boldsymbol{x}(t)$ at any point t in the interval $0 \leq t < 2T$.

The parameter a is chosen such that the error introduced by (2.11) (specially $e^{-2aT}\,\boldsymbol{x}(t+2nT)$) is consistent with the desired accuracy. But it is important that a be made no greater than nessesary since as indicated (2.12) rounoff error and truncation error will be magnified by the factor e^{at} and usually a is chosen constant in the interval $0 \leq t < 2T$ those increase rapidly as a and t increase and too large a value of K is needed to get good numerical solutions, that too long computational time is needed. Therefore we must choose the optimal value of a and the upper limit of t considering the above conditions.

Especially at $t = T$ we have

$$\boldsymbol{x}(T) \simeq \frac{e^{aT}}{T}\left[\frac{1}{2}\boldsymbol{X}(a) + R_e\sum_{k=1}^{K}\boldsymbol{X}(a + ik\pi/T)\,(-1)^k\right] \tag{2.13}$$

and using this relation we can get the numerical solution at the point T.

2.2 Computational algorithm.

Here we shall present the computational algorithm by digital computer using (2,13) when Laplace transform $\boldsymbol{X}(s)$ is given. We can select a time step without regarding the time constants of the system and this useful to treat stiff systems[9].

In Fig 1 a flow chart is shown in order to illustrate the simplicity of the algorithm which is based on the following assumptions.

1) Numerical values are given at the points $T = m\Delta T$ (m: integer) in the interval $\Delta T \leq T \leq T_{\max}$ where ΔT is a time step.

2) In this interval aT is constant and it is selected equal to 4 or 5 because the error introduced by (2.11) is as small as $e^{-8} \sim e^{-10}$ ($3.4 \times 10^{-4} \sim 4.5 \times 10^{-5}$) and in application it can be recognized as sufficiently small.

3) Summation of the series is stopped when condition

$$\frac{e^{aT}}{T}\max\left|R_e\,[\boldsymbol{X}(a + ik\pi/T)]\,ij\right| < \varepsilon \tag{2.14}$$

is satisfied where $\varepsilon > 0$ is sufficiently small.

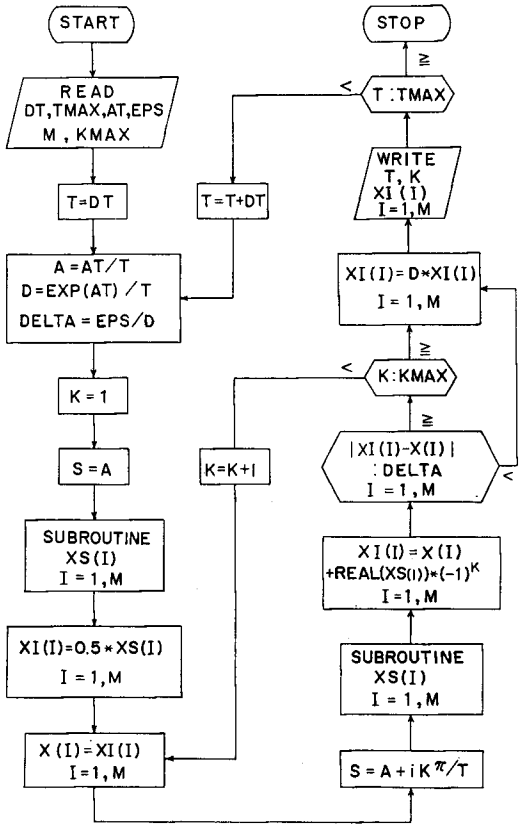

Fig. 1. Flow chart for numerical inversion

2.3 Numerical examples.

Here we shall some numerical examples using the inversion method developed above.

Example 1: $\boldsymbol{X}(s) = (\boldsymbol{s} + \boldsymbol{A})^{-1}\,\boldsymbol{x}(0)$

where $$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & 3 \\ 1 & 2 & 1 \\ -3 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{x}(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Example 2: $\boldsymbol{X}(s) = (\boldsymbol{s}^2 + \boldsymbol{A}\boldsymbol{s} + \boldsymbol{B})^{-1}\,[\dot{\boldsymbol{x}}(0) + (\boldsymbol{s} + \boldsymbol{A})\,\boldsymbol{x}(0)]$

where $$\boldsymbol{A} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 2 & -1 & 2 \\ -1 & 3 & -1 \\ 2 & -1 & 4 \end{pmatrix}, \dot{\boldsymbol{x}}(0) = \begin{pmatrix} 1 \\ 0 \\ 0.4 \end{pmatrix}, \boldsymbol{x}(0) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Example 1 is the Laplace transform of a first order homogeneous differential system and Example 2 is a second order homogeneous differential system.

Table 1. Approximate solution of example 1, a T=5, ε=0.05.

T	*Approximate* $\boldsymbol{x}(T)$				*Exact* $\boldsymbol{x}(T)$		
	x_1	x_2	x_3	K	x_1	x_2	x_3
0.1	0.5995	0.6516	1.1341	128	0.597027	0.649153	1.131822
.2	.2110	.3894	1.1361	133	.213438	.391864	1.138018
.3	− .1223	.2158	1.1039	137	− .119803	.218292	1.040803
.4	− .3844	.1133	.8661	141	− .381869	.115757	.867670
.5	− .5597	.0723	.6493	146	− .562107	.069868	.647916
.6	− .6567	.0682	.4110	150	− .659150	.065696	.409768
.7	− .6769	.0913	.1791	154	− .679356	.088862	.177958
.8	− .6324	.1289	− .0268	158	− .634838	.126439	− .027827
.9	− .5438	.1651	− .1947	161	− .541328	.167617	− .193809
1.0	− .4186	.2016	− .3131	165	− .416113	.204093	− .312283
.1	− .2787	.2278	− .3820	169	− .276195	.230210	− .381212
.2	− .1343	.2453	− .4027	172	− .136814	.242852	− .403383
.3	− .0079	.2436	− .3847	176	− .010402	.241159	− .385278
.4	.0916	.2236	− .3363	179	.094031	.226108	− .335825
.5	.1736	.2025	− .2647	182	.171082	.200021	− .265151
.6	.2213	.1685	− .1831	186	.218788	.166055	− .183457
.7	.2357	.1253	− .1004	189	.238181	.127717	− .100081
.8	.2351	.0909	− .0225	192	.232651	.088448	− .022823
.9	.2047	.0488	.0422	195	.207212	.051295	.042482
2.0	.1702	.0212	.0923	198	.167760	.018679	.092130
.1	.1179	− .0102	.1244	201	.120380	− .007728	.124498
.2	.0732	− .0246	.1400	204	.070769	− .027037	.139809
.3	.0213	− .0416	.1397	207	.023790	− .039091	.139777
.4	− .0143	− .0419	.1272	210	− .016812	− .044362	.127188
.5	− .0510	− .0463	.1055	213	− .048542	− .043803	.105449
.6	− .0677	− .0362	.0781	216	− .070158	− .038681	.078173
.7	− .0791	− .0279	.0488	218	− .081549	− .030416	.048814
.8	− .0860	− .0229	.0205	221	− .083546	− .020428	.020390
.9	− .0752	− .0075	− .0048	224	− .077672	− .010016	− .004714

In Table 1 the approximate solutions of Example 1 are given compared with the exact solutions. In Fig 2 the number of harmonics K is given for various values of ε and T.

In these examples the main operation in the algorithm is to get the inverse matrix of complex elements and is not affected by eigenvalues of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, therefore, we can get numerical solutions at any point T. Then, most of the com-

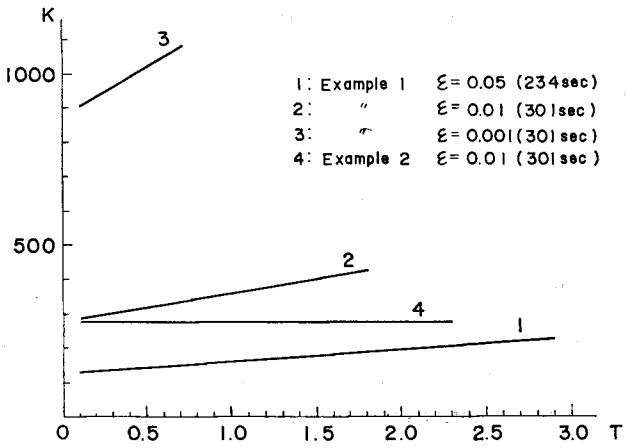

Fig. 2. Number of harmonics K for various T and ε.

putational time is consumed by the time used for matrix inversion which is proportional to $M^3 \cdot K$ (M: order of matrix), therefore, it is affected not by the order of differential equation but by the numbers of unknown variables, as shown in Fig 2.

Especially when $\boldsymbol{X}(s)=(s+\boldsymbol{A})^{-1}$ we can get numerical values of the state transition matrix $e^{-\boldsymbol{A}t}$ at any time t without regard to the eigenvalues of matrix $\boldsymbol{A}$ different from the case of series expansion method[10].

The main advantages of this method in using the analysis of linear time-invariant systems are, 1) system's equations need not be given by normal form (state equations),

Table 2. Approximate solution of Example 3, a T=5, K=150 (27 seconds)

T	Approximate $\boldsymbol{x}(T)$			Exact $\boldsymbol{x}(T)$		
	x_1	x_2	x_3	x_1	x_2	x_3
1.0	0.0000	0.0000	−0.0000	0	0	0
2.0*	.0830	.1660	− .2489	1/6	1/3	−1/2
3.0	.1652	.3300	− .4951	1/6	1/3	−1/2
4.0*	.3344	.1721	− .5066	1/2	0	−1/2
5.0	.4951	− .0010	− .4903	1/2	0	−1/2
6.0*	.7422	.4942	.2539	1	1	1
7.0	1.0030	1.0201	.9899	1	1	1
8.0	.9987	.9844	1.0281	1	1	1
9.0	.9966	.9712	1.0029	1	1	1
10.0	.9861	.9879	.9935	1	1	1

* Discontinuous point: Since this method is based on Fourier series expansion, $x(T)$ is given as

$$x(T) = \frac{1}{2}\,[x(T-0) + x(T+0)]$$

at discontinuous point T.

2) non-homogeneous equations can be treated as well as homogeneous equations and 3) time step of integration can be chosen arbitarily without regard to time constants of the systems.

Example 3: $\boldsymbol{X}(s) = \boldsymbol{s}^{-1} \cdot \exp(-\boldsymbol{A}\boldsymbol{s})\, \boldsymbol{x}_0$

where $$\boldsymbol{A} = \begin{pmatrix} 4 & 0 & 2/3 \\ 0 & 4 & 4/3 \\ 2 & 2 & 4 \end{pmatrix}, \quad \boldsymbol{x}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Example 3 is the Laplace transform of a wave equation when the initial and the boundary conditions are given. In Table 2 the approximate solutions are given compared with the exact solutions where K is fixed to 150.

3. Application of the Fast Fourier Transform

3.1 Application to numerical inversion[5].

The main operation of the numerical inversion given in **2. 3** is to get inverse matrices or exponential functions of complex matrices, and it consumes most of the computational time. Therefore, we must reduce it as much as possible. In the previous method based on (2. 13) we must do so at every required time and as the points increase, too much computational time is needed. But we can avoid doing so by using (2. 12) and the Fast Fourier Transform (F. F. T.) (See Appendix).

Here we shall rewrite (2. 12)

$$\left.\begin{aligned} \boldsymbol{x}(t) = \frac{e^{at}}{T}\left[R_e \sum_{k=0}^{K-1} \boldsymbol{X}(a + ik\pi/T)\, e^{ik(\pi/T)t} - \frac{1}{2}\boldsymbol{X}(a) \right] \\ 0 \le t < 2T \end{aligned}\right\} \tag{3.1}$$

and putting

$$t_n = \frac{n \cdot 2T}{K} \qquad n = 0, 1, \ldots\ldots, K-1$$

we have

$$\boldsymbol{x}(t_n) = \frac{e^{at_n}}{T}\left[R_e \sum_{k=0}^{K-1} \boldsymbol{X}(a + ik\pi/T)\, e^{i2\pi nk/K} - \frac{1}{2}\boldsymbol{X}(a) \right] \tag{3.2}$$

and putting

$$\boldsymbol{x}_n = \sum_{k=0}^{K-1} \boldsymbol{X}(a + ik\pi/T)\, e^{i2\pi nk/K} \tag{3.3}$$

we can adapt F. F. T for $\boldsymbol{x}_n$ and using it we can get numerical value of $\boldsymbol{x}(t_n)$. by (3.2).

In Fig 3 flow chart of the algorithm based on this method is shown.

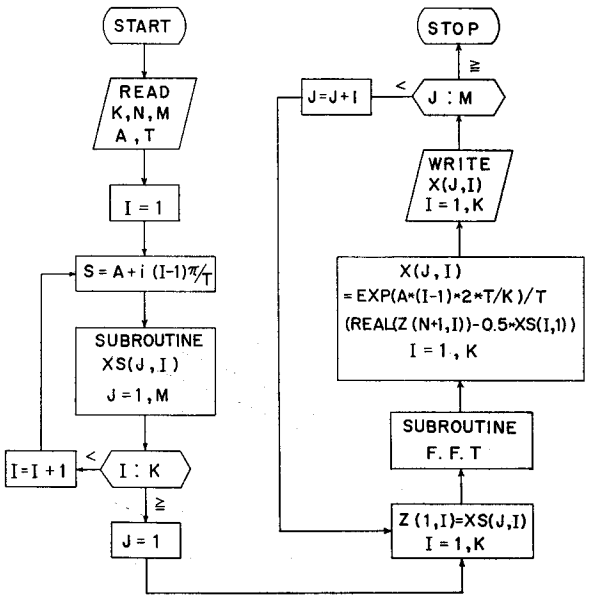

Fig. 3. Flow chart for numerical inversion by F.F.T

Next we may determine the number of terms K at $t=T$ by (2.14). Then error introduced is smaller than ε at every point t in the interval $0\leq t\leq T$, but larger than ε in $T<t<2T$ because in the later interval it is magnified by $e^{a(t-T)}>1$, therefore, we shall take numerical values only in the interval $0\leq t\leq T$.

Furthermore, we shall not need exact value of K determined by (2.14) but minimum value of powers of 2 satisfiing (2.14) in this case.

3.2 Numerical examples.

Here we shall show some numerical examples by the method developed above.

Example 4: $\boldsymbol{X}(s) = (\boldsymbol{s}+\boldsymbol{A})^{-1}\boldsymbol{x}(0)$

where
$$\boldsymbol{A}=\begin{pmatrix} 1 & 0 & 3 \\ 1 & 2 & 1 \\ -3 & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{x}(0)=\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

This example is same as in example 1 and when $aT=5$, $T=3$ and $K=256$ computational time used for CPU is reduced from 234 seconds to 14 seconds by using F. F. T, although time step has gone from 0.1 to 6/256.

Example 5: $\boldsymbol{X}(s) = s^{-1} \exp(-\boldsymbol{A}\sqrt{s})\, \boldsymbol{x}_0$

where $$\boldsymbol{A} = \begin{pmatrix} 4 & 0 & 2/3 \\ 0 & 4 & 4/3 \\ 2 & 2 & 4 \end{pmatrix}, \qquad \boldsymbol{x}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

This example is the operational solution of a heat equation when the initial and the boundary conditions are given and it is pretty difficult to get the time solution of analytical form, but using the above method we can get numerical values directly. In Fig 4 approximate solution when $aT=5$, $T=10$ and $K=128$ is given.

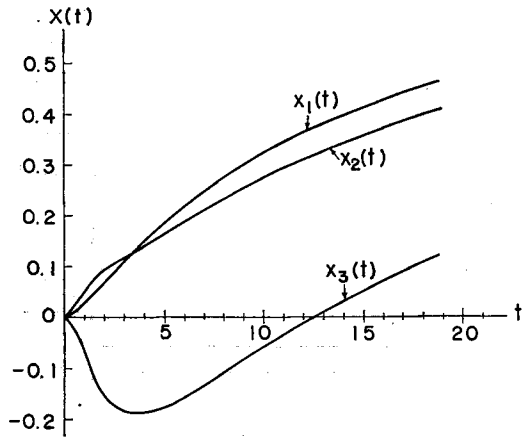

Fig. 4. Approximate solution of Example 5, a T=5, T=10, K=128 (36 seconds)

3.3 Application to linear system with arbitrary input.

Here we shall consider the system whose property is given by

$$\boldsymbol{L}\left(\frac{d}{dt}\right)\boldsymbol{x}(t) = \boldsymbol{B}\,\boldsymbol{u}(t) \tag{3.4}$$

where $\boldsymbol{L}$ is a linear differential operator and $\boldsymbol{u}(t)$ an input function.

Laplace transform of (3.4) is given by

$$\boldsymbol{L}(s)\,\boldsymbol{X}(s) = \boldsymbol{L}_0(s) + \boldsymbol{B}\,\boldsymbol{U}(s) \tag{3.5}$$

and we have

$$\boldsymbol{X}(s) = [\boldsymbol{L}(s)]^{-1}\,[\boldsymbol{L}_0(s) + \boldsymbol{B}\,\boldsymbol{U}(s)] \tag{3.6}$$

where $\boldsymbol{L}_0(s)$ is the term caused by an initial value of $\boldsymbol{x}(t)$, and from (3.6) we can get numerical values of $\boldsymbol{x}(t)$ by the method stated above.

Then we shall consider the case where $\boldsymbol{u}(t)$ and $\boldsymbol{U}(s)$ cannot explicitly be given

by analytical form.

For simplicity we shall assume that input functions are scalar and denote them by $u(t)$ and $U(s)$.

In numerical inversion we do not need a function $U(s)$ but complex frequency spectra of $U(s)$ which can be known from sampled values of $u(t)$, therefore, we can get numerical values of $\boldsymbol{x}(t)$ in this case.

Now we shall present the method. From (2.3) we have

$$U(a+i\omega)=\int_0^{\infty} u_1(t)\, e^{-i\omega t}\, dt$$

and in this equation the function to be transformed is $u_1(t)=u(t)e^{-at}$ and generally decreases rapidly as t increases, therefore, we can approximate it by

$$U(a+i\omega)=\int_0^{2T} u_1(t)\, e^{-i\omega t}\, dt\,.$$

Next we shall approximate it by

$$U(a+i\omega)=\frac{2T}{N}\sum_{n=0}^{N-1} u_1(n\cdot 2T/N)\, e^{-i\omega\,(n\cdot 2T/N)}$$

For numerical inversion we shall need frequency spectra of integer multiples k of π/T, that is

$$U(a+ik\pi/T)=\frac{2T}{N}\sum_{n=0}^{N-1} u_1(n\cdot 2T/N)\, e^{-i(2\pi/N)nk} \tag{3.7}$$

therefore, if we may choose $k=0,1,\ldots\ldots,\ N-1$ and $N\geq K$, we can get necessary frequency spectra of $U(s)$ from sampled values of $u(t)$ by (3.7) using F. F. T, and using them we can get numerical values of $\boldsymbol{x}(t)$ by (3.6) when sampled values of input function are given.

4. Conclusion

As mentioned above, the method of numerically inverting Laplace transform by Fourier series expansion and some numerical examples are given. Furthermore, it is shown that we can reduce computational time remarkably by ppalying the Fast Fourier Transform to the method.

This method can be applied to the analysis of the systems with many variables without regard to time constants of the systems and must be a very powerful tool for system's analysis.

In this paper digital computations were carried out by FACOM 230-60 in

Data Processing Center of Kyoto University.

References

1) Ichikawa, S. and Kishima, A.: This Memoirs., Vol. XXXII, Part 2, pp. 210–222 (1970)
2) Ichikawa, S. and Kishima, A.: This Memoirs., Vol XXXIII, Part 2, pp. 80–93 (1971)
3) J. Mikusinsky: "Operational Calculus" Pergamon Press (1959)
4) M. Silverberg: T.I.E.E.E., Vol. CT-17, pp. 82–86 (1970)
5) H. Dubner and J. Abate: J.A.C.M., Vol. 15, pp. 115–123 (1968)
6) W. T. Cochran et al.: P.I.E.E.E., Vol 55, pp. 1664–1674 (1967)
7) E.O. Brigham and R.E. Morrow: I.E.E.E., Spectrum, Vol 4, pp. 63–70 (1967)
8) J.M. Glassman: T.I.E.E.E., Vol C-19, pp. 105–116 (1970)
9) W. Liniger and R.A. Willoughby: S.I.A.M.J., Numer. Anal., Vol. 7, pp. 47–66 (1970)
10) M.L. Liou: P.I.E.E.E., Vol 54, pp. 20–23 (1966)

Appendix[8)]

When a function $y(t)$ is given, the Fourier transform and its inversion formula is defined as follows:

$$Y(w) = \int_{-\infty}^{\infty} y(t)\, e^{-i\omega t}\, dt\,. \tag{A.1}$$

$$y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y(\omega)\, e^{i\omega t}\, dw\,. \tag{A.2}$$

Then we shall consider the sampled function $y^*(t)$ defined as

$$y^*(t) = \sum_{k=0}^{K-1} y(t)\, \delta(t - kT). \tag{A.3}$$

The Fourier transform $Y^*(\omega)$ is

$$Y^*(\omega) = \sum_{k=0}^{K-1} y(kT)\, e^{-i\omega kT} \tag{A.4}$$

and frequencies are chosen for integer multiples n of $2\pi/KT$ it becomes

$$Y^*\left(\frac{n\cdot 2\pi}{KT}\right) = \sum_{k=0}^{K-1} y(kT)\, e^{-i2\pi nk/K} \tag{A.5}$$

and by this equation we can get the frequency spectra of $y(t)$ from its sampled values.

If K Fourier coefficients ($n=0$ to $K-1$) are computed, matrix representation of (A.5) becomes

$$\boldsymbol{Y}^* = \boldsymbol{W}\boldsymbol{y} \tag{A.6}$$

where

$$\boldsymbol{Y}^* = \begin{pmatrix} Y^*(0) \\ Y^*(2\pi/KT) \\ \vdots \\ Y^*(\overline{K-1}\ 2\pi/KT) \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y(0) \\ y(T) \\ \vdots \\ y(\overline{K-1}\ T) \end{pmatrix},$$

$$\boldsymbol{W} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & w^1 & w^2 \cdots\cdots & w^{K-1} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ 1 & w^{K-1} & w^{2(K-1)} \cdots & w^{(K-1)^2} \end{pmatrix}, \quad w = e^{-i2\pi/K}.$$

Because this equation is the product of matrices, the number of multiplications increases rapidly as K increases. However, when K is chosen suitably and the Fast Fourier Transform (F. F. T) based on the periodicity of powers of w is used, matrix $\boldsymbol{W}$ may be solved into the products of sparse matrices, which significantly reduces the number of multiplications required.

Here we shall present the method. If r is any factor of K, that is $K=rK'$, then, for (i, j) element of matrix $\boldsymbol{W}$ we have

$$Wij = w^{(i-1)(j-1)}.$$

Denote by $\boldsymbol{W}j$ j-th coloum vector of $\boldsymbol{W}$, that is

$$\boldsymbol{W}j = (W_{1j}\ W_{2j} \cdots\cdots\ W_{kj})^t$$

then, from the relations

$$\left.\begin{aligned} &Wi+r, j = w^{(i+r-1)(j-1)}\, w^{r(j-1)}\, Wij \\ &Wi+2r, j = w^{2r(j-1)}\, Wij \\ &\quad\vdots \\ &Wi+(k'-1)\, r, j = w^{(k'-1)r(j-1)} Wij \\ &\quad i = 1, 2, \ldots\ldots, r \end{aligned}\right\}$$

and from the periodicity of $w^r = e^{-i2\pi/k'}$ that is

$$w^{r(j-1)} = w^{r(j-1)MODK'}$$

we have

$$\boldsymbol{W}j^t = \begin{cases} (\boldsymbol{\gamma}_j \quad \boldsymbol{\gamma}_j \cdots\cdots\cdots \boldsymbol{\gamma}_j) & (j-1)\ MODK'=0 \\ (\boldsymbol{\gamma}_j\ w^r \boldsymbol{\gamma}_j \cdots\cdots\cdots w^{(k'-1)r} \boldsymbol{\gamma}_j) & (j-1)\ MODK'=1 \\ \cdots\cdots\cdots\cdots\cdots\cdots & \cdots\cdots\cdots\cdots\cdots \\ (\boldsymbol{\gamma}_j\ w^{(k'-1)r} \boldsymbol{\gamma}_j \ldots w^{(k'-1)2r} \boldsymbol{\gamma}_j) & (j-1)\ MODK'=K'-1 \end{cases} \tag{A.7}$$

where

$$\boldsymbol{\gamma}_j = (1\ w^{j-1}\ w^{2(j-1)} \cdots\cdots\ w^{(r-1)(j-1)})$$

and W can be represented as the product of two matrices as follows

$$W=\begin{pmatrix} \gamma_1 & O_r \ldots\ldots O_r \\ O_r & \gamma_2 \ldots\ldots O_r \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ O_r & O_r \ldots\ldots \gamma_{k'} \\ \gamma_{k'+1} & O_r \ldots\ldots O_r \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ O_r & O_r \ldots\ldots \gamma_k \end{pmatrix} \begin{pmatrix} U_r & U_r \ldots\ldots\ldots\ldots U_r \\ U_r & w^r U_r \ldots\ldots\ldots w^{(k'-1)r} U_r \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ U_r & w^{(k'-1)r} U_r \ldots\ldots w^{(k'-1)^2 r} U_r \end{pmatrix} \tag{A.8}$$

where

$$O_r = (\underbrace{0\ 0 \ldots\ldots 0}_{r}), \qquad U_r = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{pmatrix} \Big\} r \,.$$

The second matrix of the right side of (*A*. 8) is of the same form as W, therefore, if K' has any factor, it can be solved into the product of more simple matrices in the same way as stated above. Specially if $K=2^N$, W is solved into the products of N

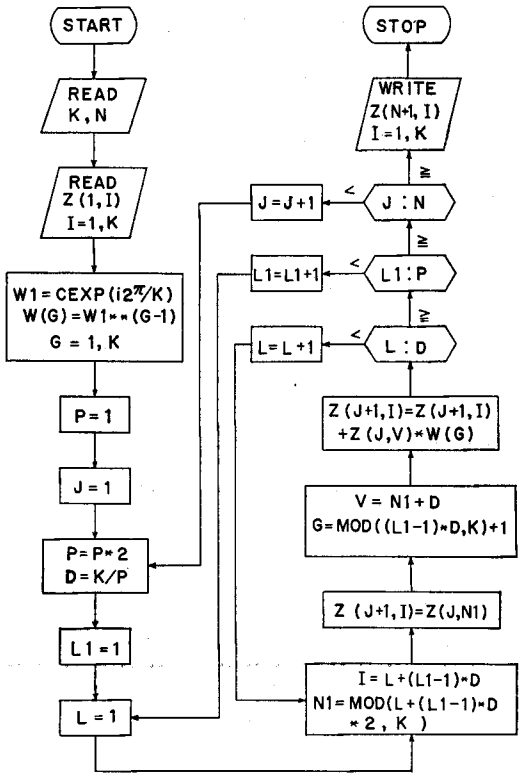

Fig. A-1 Flow chart for Fast Fourier Transform

sparse matrices and for each one numbers of non-zero elements are $2K$ (every row has 1, any power of w and K–2 0s), therefore, in using this method numbers of multiplications are reduced from K^2 to NK and numbers of additions from $K(K-1)$ to NK in comparsion with direct method by (*A*. 6), and computational time can be reduced greatly. The processes of factoring the matrices are regular MOD operations and we can perform them by digital computers, therefore, this method is very attractive for numerical calculations.

In Fig *A*–1 flow chart of above processes is given where $\boldsymbol{W}$ is solved into

$$\boldsymbol{W} = \boldsymbol{W}_1 \boldsymbol{W}_2 \ldots\ldots \boldsymbol{W}_N$$

and

$$\boldsymbol{Z}_1 = \boldsymbol{y},\ \boldsymbol{Z}_2 = \boldsymbol{W}_N \boldsymbol{Z}_1,\ \ldots\ldots,\ \boldsymbol{Z}_N = \boldsymbol{W}_2 \boldsymbol{Z}_{N-1},\ \boldsymbol{Z}_{N+1} = \boldsymbol{W}_1 \boldsymbol{Z}_N = \boldsymbol{Y}^* .$$