COMPUTER-AIDED NUMERICAL INVERSION OF LAPLACE TRANSFORM

UMESH KUMAR*

Senior Member, IEEE, Fellow IETE, EE Department, IIT Delhi-110 016, India

This paper explores the technique for the computer aided numerical inversion of Laplace transform. The inversion technique is based on the properties of a family of three parameter exponential probability density functions. The only limitation in the technique is the word length of the computer being used. The Laplace transform has been used extensively in the frequency domain solution of linear, lumped time invariant networks but its application to the time domain has been limited, mainly because of the difficulty in finding the necessary poles and residues. The numerical inversion technique mentioned above does away with the poles and residues but uses precomputed numbers to find the time response. This technique is applicable to the solution of partially differentiable equations and certain classes of linear systems with time varying components.

Keywords: Laplace transform, invariant networks, linear system

THEORETICAL ASPECTS OF THE TECHNIQUE

The three parameter exponential probability density functions for a random variable T are

$$p_{m,n}(a, t) = \frac{((m + n)/m)!}{n!(m - 1)!}(1 - e^{-at})ae^{-mat} \quad a > 0, \quad 0 \leq t \leq \infty$$

$$m = 1, 2, 3 \quad n = 0, 1, 2 \tag{1}$$

Plots of typical density functions are illustrated in Figure 1 for selected values of $m = n$. If the density of (1) is differentiated, the peak
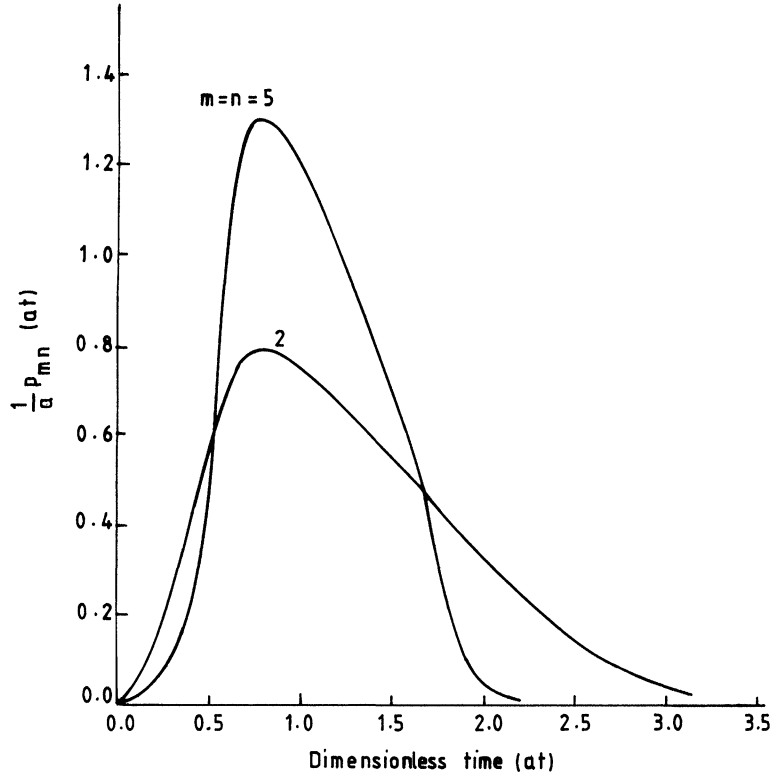
*e-mail: umesh@ee.iitd.ernet.in

189

FIGURE 1   Three parameter exponential probability density functions.

is found to occur at

$$t = \ln\left(\frac{n+m}{m}\right)\bigg/ a \tag{2}$$

The expected value of some function f of the random variable T is given by

$$E[f(t)] = \int_0^\infty f(t)p_{m,n}(a,t)dt \tag{3}$$

Obtaining binomial expansion of a portion of the density function gives

$$(1 - e^{-at})^n = \sum_{i=0}^\infty \binom{n}{i}(-1)^i e^{-iat} \tag{4}$$

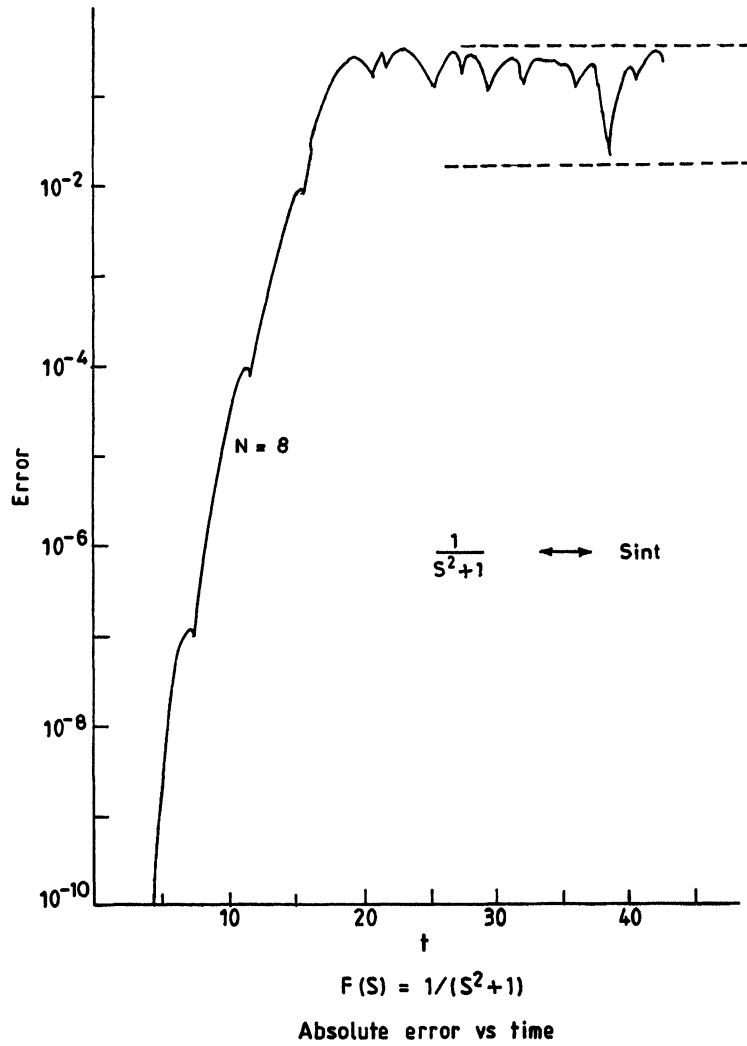$$F(S) = 1/(S^2+1)$$

Absolute error vs time

FIGURE 2   Broken lines indicate bounds of errors for other values of N.

The expected value of $f(t)$ is then

$$E[f(t)] = \frac{(n+m)!}{n!(m-1)!} \int_0^\infty \sum_{i=0}^n \binom{n}{i} (-1)^i e^{-(i+m)at} a \cdot dt \qquad (5)$$
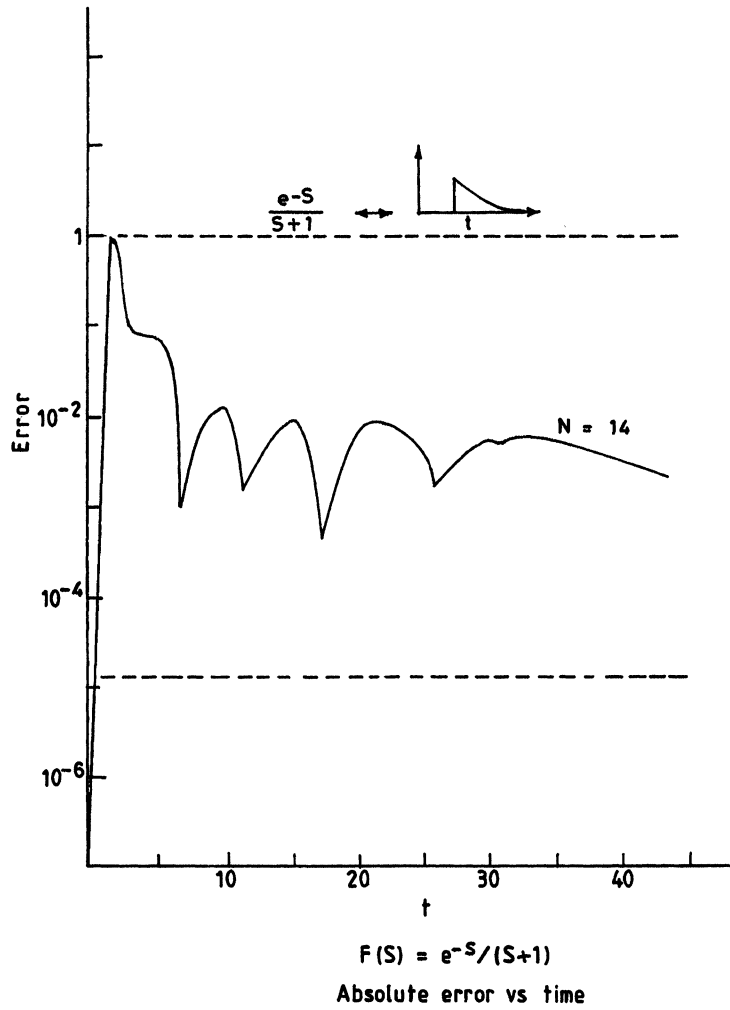
$$F(S) = e^{-S}/(S+1)$$

Absolute error vs time

FIGURE 3   Broken lines indicate bounds of errors.

changing order of the summation and integration, the result is

$$E[f(t)] = \frac{(n+m)!}{n!(m-1)!}\sum_{i=0}^{n}\binom{n}{i}(-1)^{i}a\int_{0}^{\infty}f(t)e^{-(i+m)at}dt \qquad (6)$$

$$F(S) = 1/(S+1)^2$$
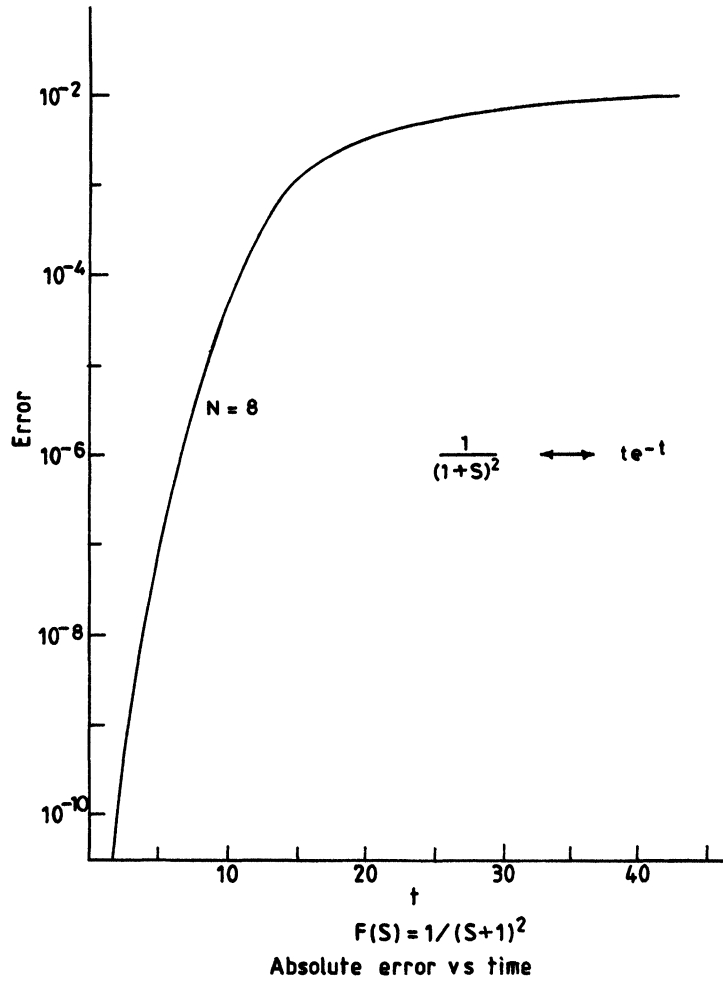
Absolute error vs time

FIGURE 4

The integral can be identified as Laplace transform of $f(t)$ which is denoted by $F(s)$

$$E[f(t)] = \frac{(n+m)!}{n!(m-1)!} \sum_{i=0}^{n} \binom{n}{i} (-1)^i a F((i+m)a) \qquad (7)$$
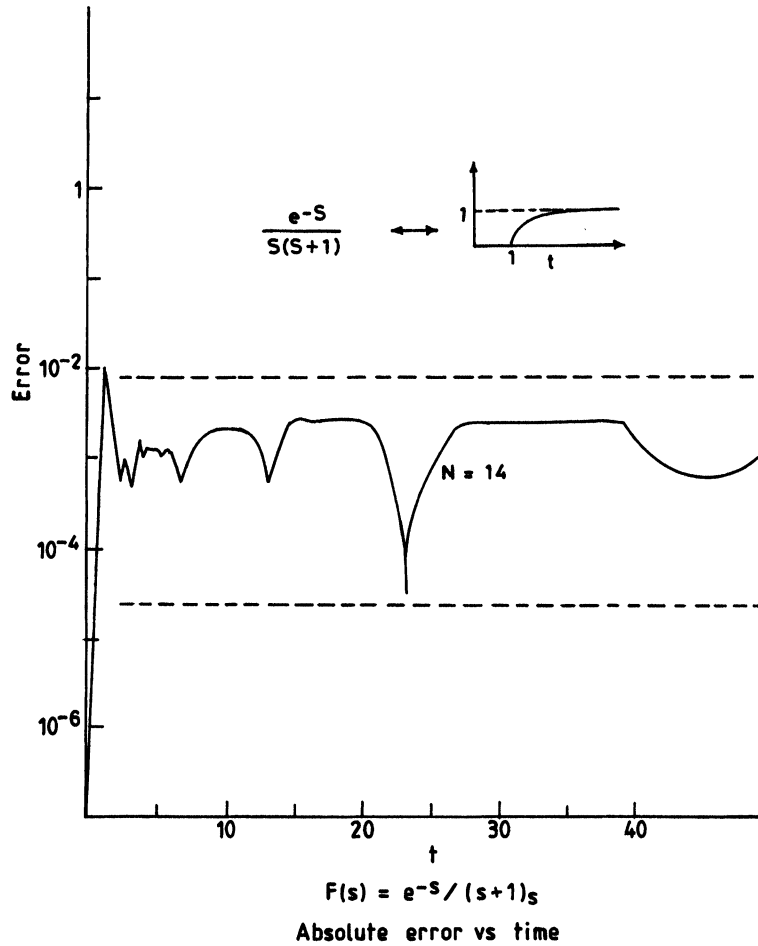
FIGURE 5　Broken lines indicate bounds of error for other values of N.

where the function $F(s)$ is defined as

$$F(s) = \int_0^\infty f(t)\mathrm{e}^{-st}dt \qquad (8)$$

If the limit is taken as m and n approach infinity, the density function becomes very peaked and behave like a Dirac delta function
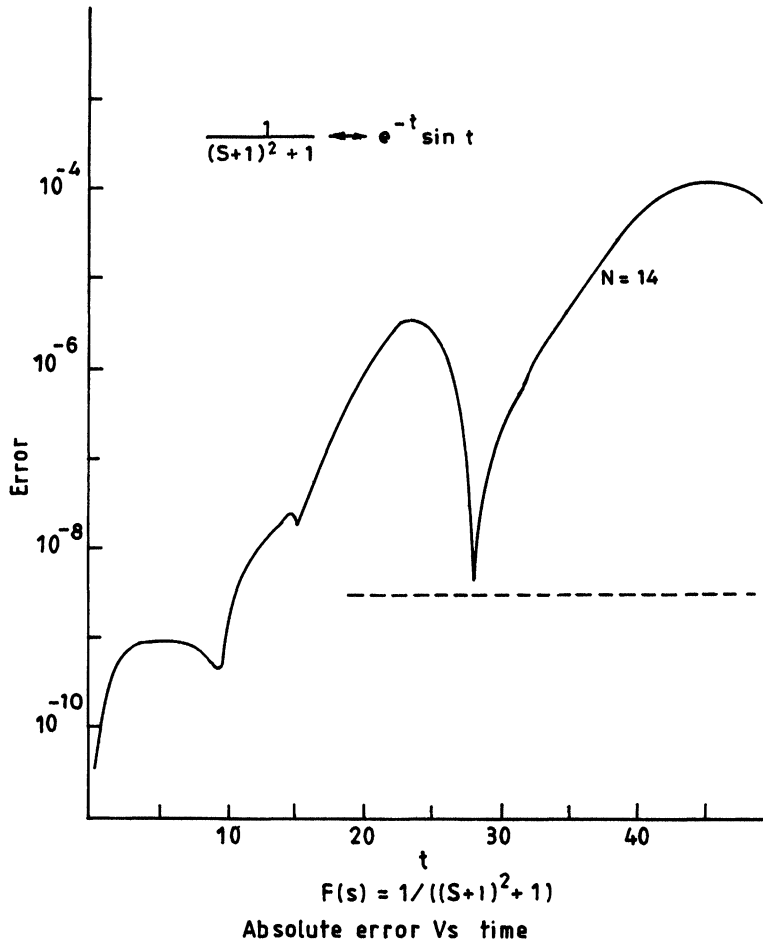
FIGURE 6   Broken line indicates bounds of errors for other values of N.

located at the mode (peak) of the density and hence the density tends
to sample the function $f(t)$ at

$$t = \ln\left(\frac{n+m}{m}\right)\bigg/ a$$

$$\text{or}\quad f\left[\frac{\ln((n+m)/m)}{a}\right] = \lim_{m,n\to\infty} E[f(t)] \tag{9}$$

$$F(S) = e^{-S}(1 - e^{-S})/S$$
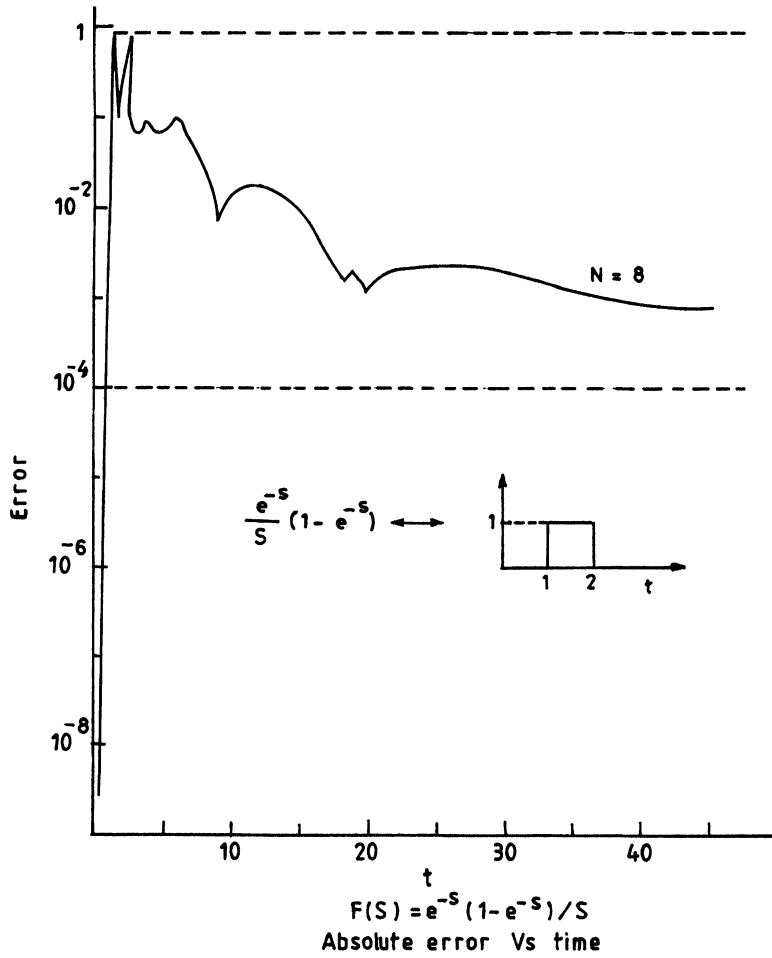
Absolute error Vs time

FIGURE 7   Broken lines indicate bounds of errors for other N.

for convenience sake 'a' is assigned the value

$$a = \frac{\ln((m+n)/m)}{t}$$

A combination of (7) and (9) yields

$$f(t) = \lim_{m,n \to \infty} \left\{ \frac{\ln((m+n)/m)}{t} \frac{(m+n)!}{n!(m-1)!} \sum_{i=0}^{n} \binom{n}{i} (-1)^i \right.$$
$$\left. F\left[\frac{i+m}{t} \times \ln\left(\frac{n+m}{m} \right) \right] \right\} \tag{11}$$

$$F(S) = e^{-S}/S$$

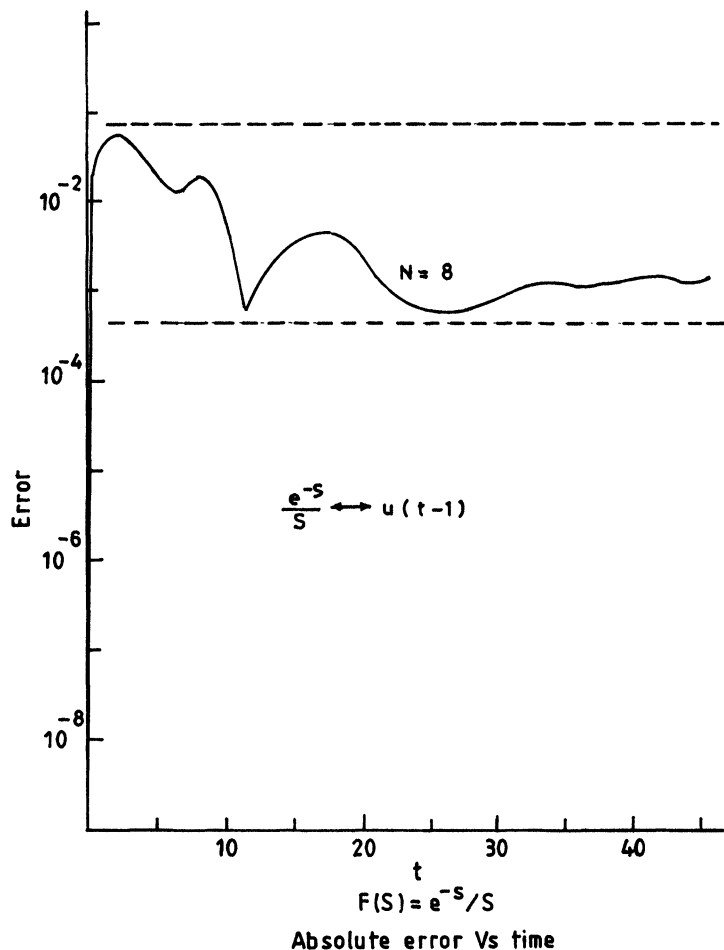**Absolute error Vs time**

FIGURE 8   Broken lines indicate bounds of errors for other values of N.

If the S-domain function $F(s)$ is known, then (11) may be employed to evaluate the corresponding time domain function for finite m and n, the estimate of the inverse transform $f_{m,n}(t)$ may be written as:

$$f_{m,n}(t) = \frac{\ln((m+n)/m)}{t} \sum_{i=0}^{n} \frac{(m+n)!(-1)^i}{(m-1)!(i)!(n-1)!} F\left[\left(\frac{i+m}{i}\right)\ln\left(\frac{m+n}{m}\right)\right] \tag{12}$$

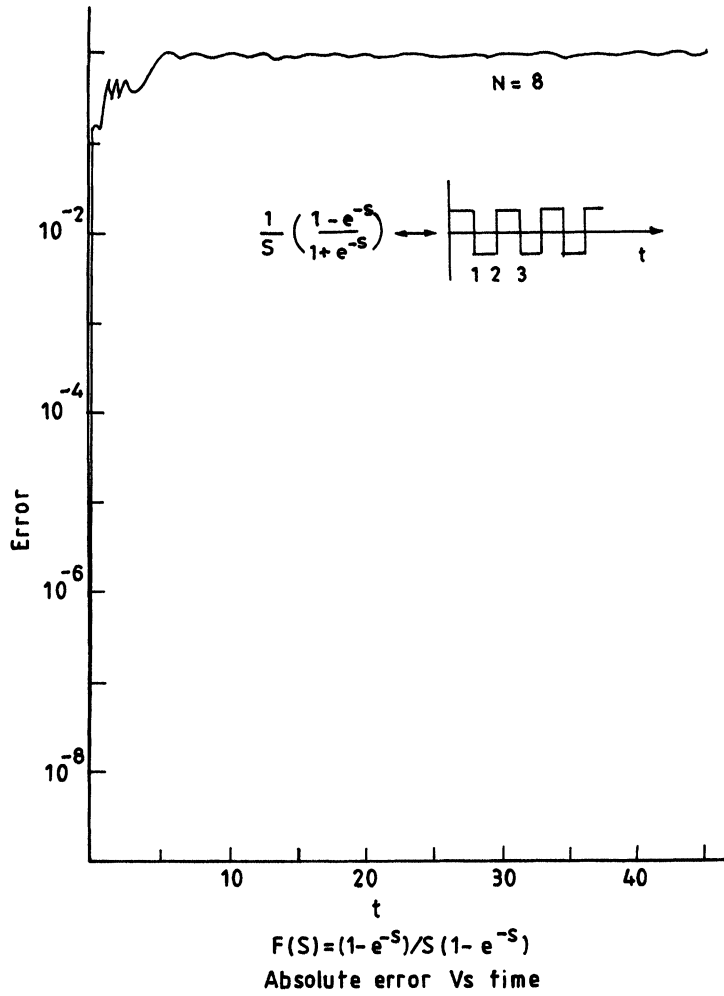$$F(S) = (1 - e^{-S})/S(1 - e^{-S})$$
Absolute error Vs time

FIGURE 9

if $m = n$, then

$$f_n(t) = \frac{\ln 2}{t} \sum_{i=0}^{n} \frac{(2n)!(-1)^i}{(m-1)!\, i!\, (n-1)!} F((i+n)\ln(2/t)) \qquad (13)$$

Expression (13) is called the Gaver algorithm. Improved accuracy may be obtained if the weighted average of $f_1(t)$, $f_2(t)$,...,$f_{N/2}(t)$ is

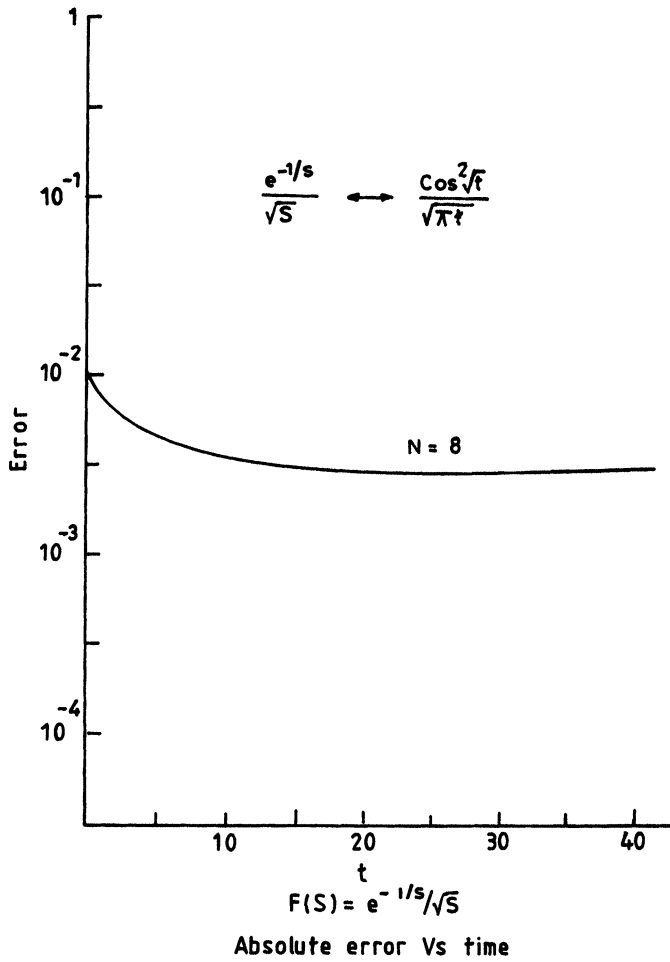$$F(S) = e^{-1/s}/\sqrt{S}$$

Absolute error Vs time

FIGURE 10

taken for a fixed t, such that

$$\overline{f_N(t)} = \sum_{i=1}^{N/2} x_i f_i(t) \qquad (14)$$

where the weights x_i are given by

$$x_i = \frac{(-1)^{i-1}}{(N/2)!} \binom{N/2}{i} i((N/2) + 1 - i)^{(N/2)-1} \qquad (15)$$

$$\frac{e^{-\sqrt{S}}}{\sqrt{S}} \quad \longleftrightarrow \quad \frac{e^{-1/4t}}{\sqrt{\pi t}}$$

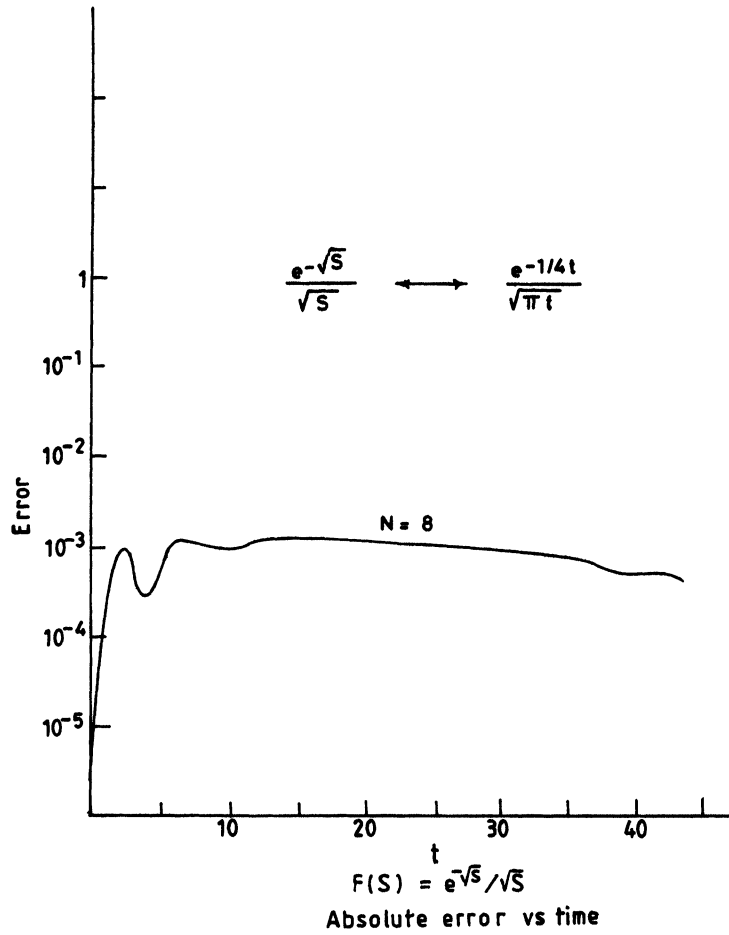$$F(S) = e^{-\sqrt{S}}/\sqrt{S}$$

Absolute error vs time

FIGURE 11

and N must be an even integer. By recombination of similar terms in (14) we find that the averaged approximate inverse transform is given by

$$\overline{f_N(t)} = \frac{\ln 2}{t} \sum_{i=1}^{N} V_i F\left(\frac{i \ln 2}{t}\right) \qquad (16)$$

$$\frac{999}{(S+1)(10^3+S)} \longleftrightarrow e^t - e^{-1000t}$$

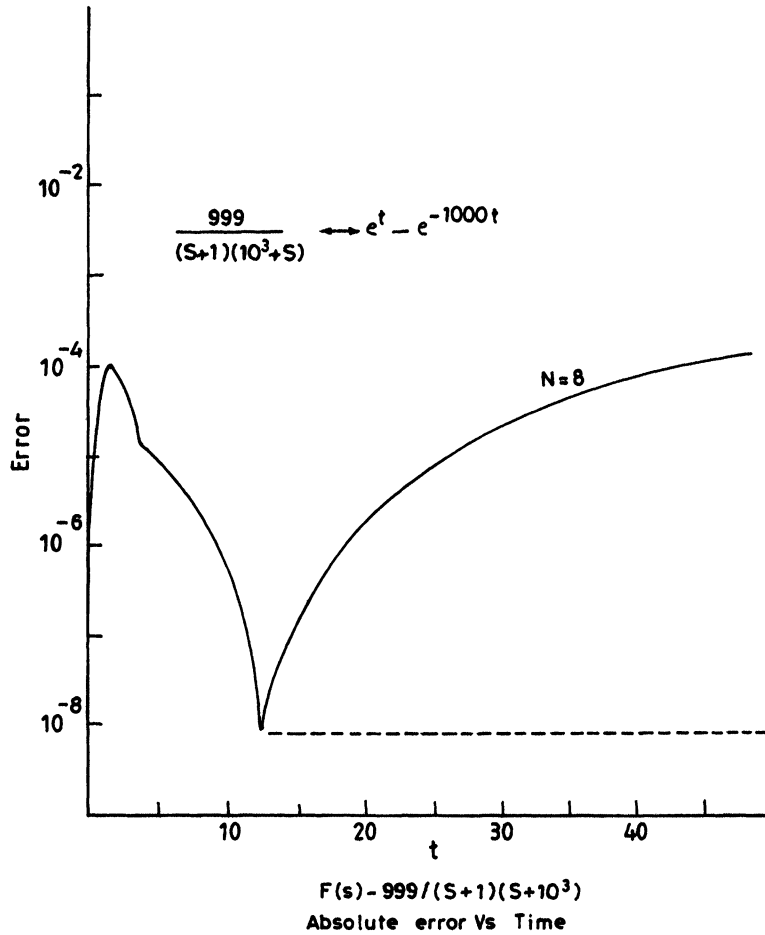F(s) - 999/(S+1)(S+10³)
Absolute error Vs Time

FIGURE 12   Broken line indicates bounds of errors for other values of N.

V_i is given by

$$V_i = (-1)^{(N/2)+i} \sum_{k=(i+1)/2}^{\text{Min }(i,N/2)} \frac{k^{N/2}(2k)!}{((N/2)-k)!k!(k-1)!(i-k)!(2k-i)!} \quad (17)$$

and k is computed using integer arithmatic. This expression is called the stehfest algorithm.

$$F(s) = 1/\sqrt{s}$$

Absolute error Vs Time
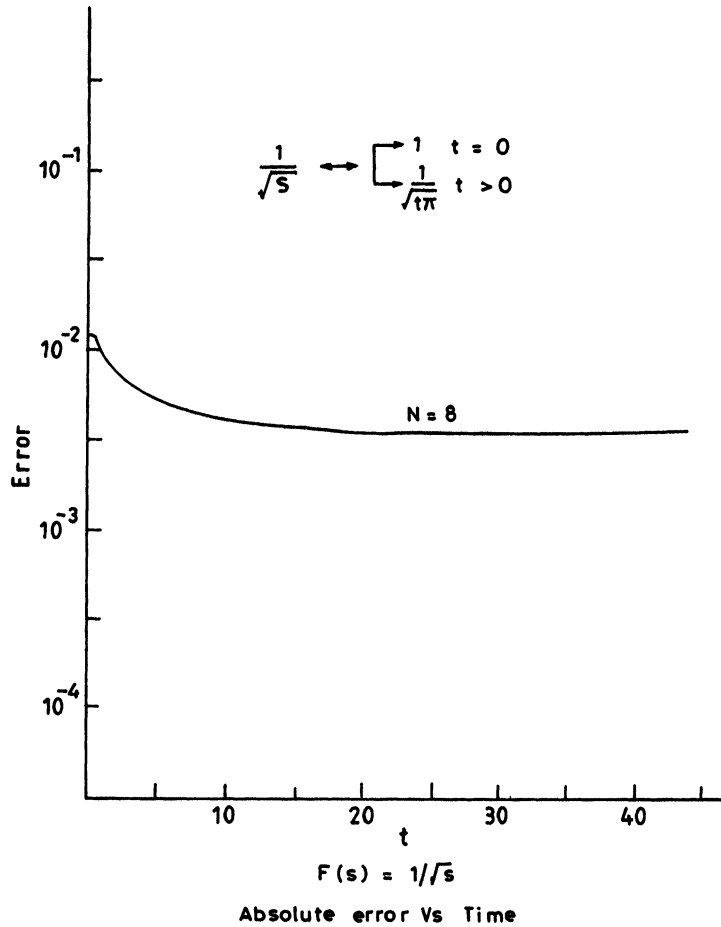
FIGURE 13

The analytical form of $F(s)$ is not limited to polynomial ratios. Here $F(s)$ can include transcendental function or transport lag terms $\exp(-ts)$ which arise in the analysis of distributed parameter systems.

The method of "extrapolation to the limit" which Gaver used, leads to less accurate results for the same N, because not so many powers of 'n' cancel out. Moreover, with this method N must be a power of 2, so that in general one cannot make the best use of the available computer precision.
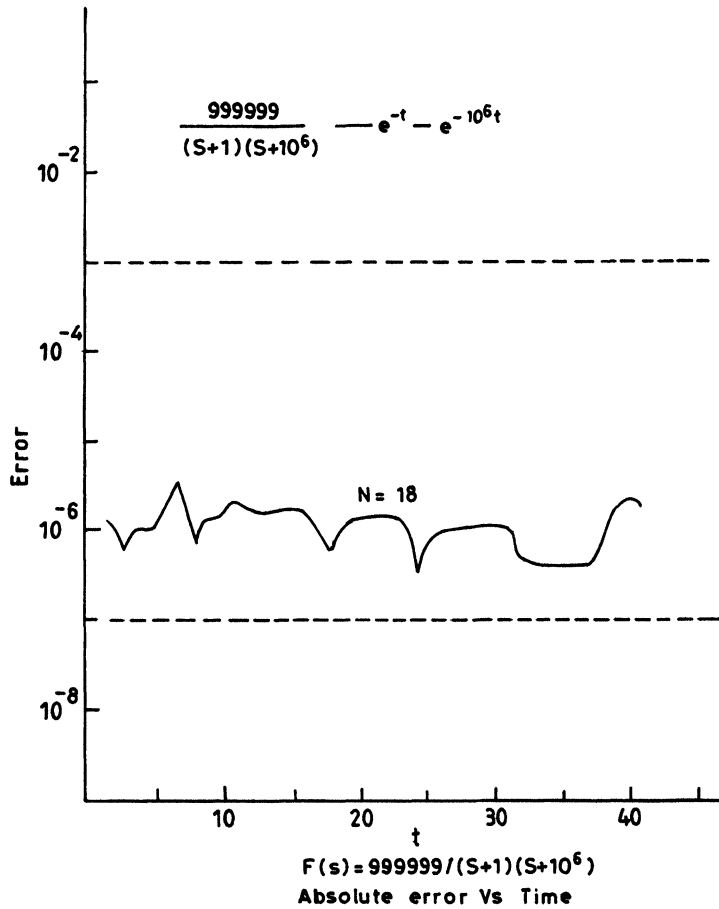
FIGURE 14 Broken lines indicate bounds of errors for other values of N.

Theoretically $\bar{f}_n$ becomes more accurate for large values of N. Practically, rounding errors worsen the results if N becomes too large because V_i with greater and greater absolute values occurs. For given $F(s)$ and T, N at which the accuracy is maximal increases with the number of significant figures used. For fixed computer precision the optimal value of N is smaller *i.e.*, maximum accuracy is greater and $\bar{f}_n$ converges to $f(t)$ faster. It was also found that with increasing N, the number of correct figures first increases linearly and then owing to rounding errors, decreases linearly. The optimum N is approximately

$$F(S) = (1-e^{-S})/S^2(1+e^{-S})$$

Absolute error vs time
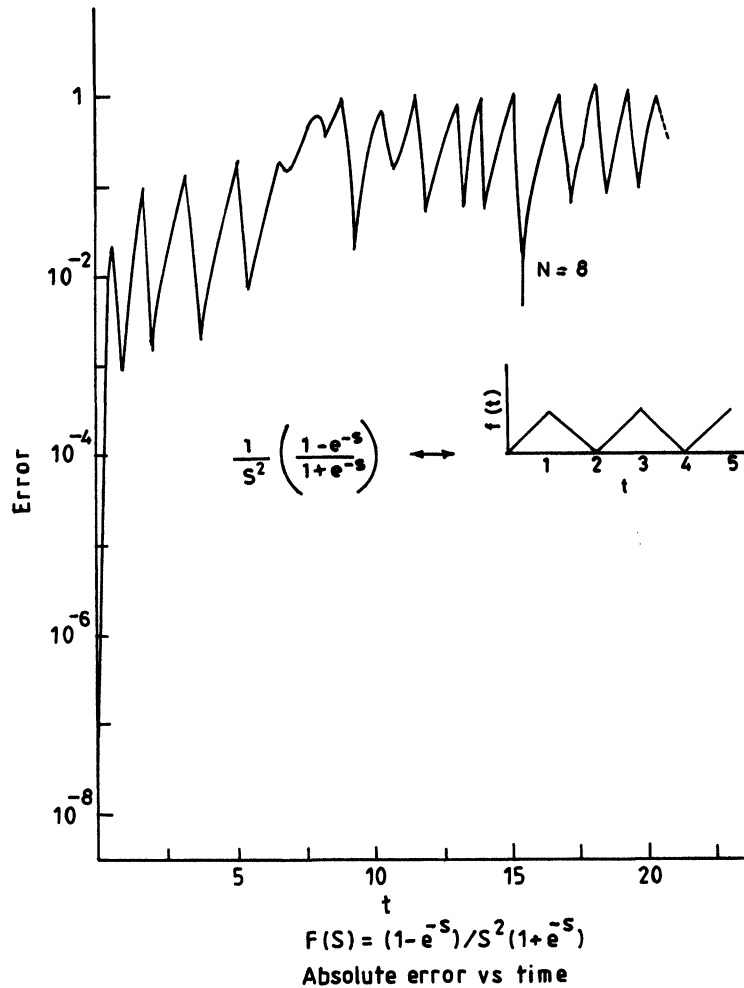
FIGURE 15

proportional to the number of digits the machine is working with. Evaluating an unknown function from its Laplace transform, one should compare the results for different N, to see whether the function is smooth enough (less oscillatory), what accuracy can be reached and what the optimum N is. One should also make sure that the unknown function $f(t)$ has not any discontinuities, salient points, sharp peaks or rapid oscillations.

$$F(S) = 1/S(1-e^{-S})$$

Absolute error vs time
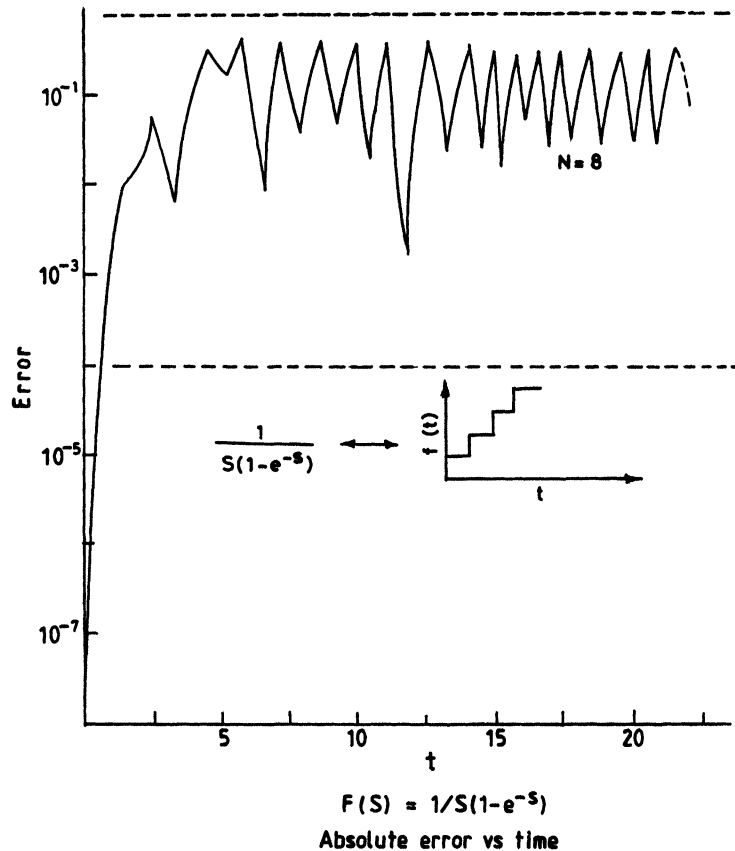
FIGURE 16   Broken lines indicate bounds of errors for other values of N.

If the Gaver algorithm is employed, the coefficients with alternating signs create a problem in that large numbers of very nearly equal size must be subtracted from one another, thus implying the need for retention of a large number of significant digits. As the value of N is increased this becomes a word length problem for any digital processor. Similarly if the averaged approximate inverse algorithm is employed, there will be a value N beyond which a degradation of the inverse will occur due to finite digital processor word length. The algorithm works very well when oscillatory solutions are not anticipated, otherwise the asymptotic character of the inverse function soon dominates.

$$F(S) = \pi(1 + \pi^2)$$

Absolute error vs time

FIGURE 17

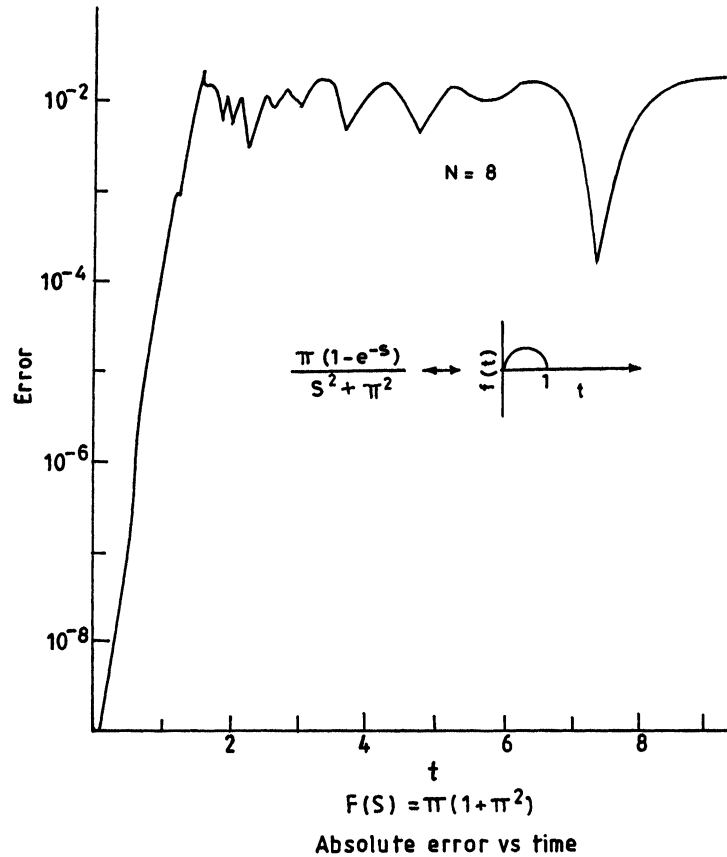An alternative method on similar lines is: Consider the Laplace transform inversion formula

$$V(t) = \frac{1}{2\pi j} \int_{C-j\infty}^{C+j\infty} V(s)e^{st}ds \qquad (A)$$

where s is the complex frequency variable and C is the arbitrary positive constant such that $\text{Res}_i < C$ where S_i are the poles of $V(s)$. Using the substitution

$$(A) \text{ changes to } z = St \qquad (B)$$

$$V(t) = \frac{1}{2\pi j t} \int_{C'-j\infty}^{C'+j\infty} V(z/t)e^z dz \qquad (C)$$

Approximate the function e^z in (C) by a rational function. The Pade approximation was used.

$$\xi_{N,M}(z) = \frac{P_N(z)}{Q_M(z)} \qquad (D)$$

where $P_N(z)$ and $Q_M(z)$ are polynomials of order N, M respectively. As is well known, the Pade approximation formally equates a rational function to some terms of a series

$$\sum_{i=0}^{N} a_{iz}^i \Big/ 1 + \sum_{i=1}^{M} f_i z^i = \sum_{i=0}^{M+N} C_i z^i + \sum_{M+N+1}^{\infty} C_i z^i \qquad (E)$$

the coefficients C_i for $i \le M + N$ are the coefficients of the Taylor expansion of e^z. Therefore the Pade approximation $\xi_{N,M}(z)$ has the first $M+N+1$ terms of its Taylor expansion equal to the Taylor expansion of e^z. The remaining terms of both expansions differ.

It is not necessary to solve the system of equations arising from (E). A closed form exists

$$\xi_{N,M}(z) = (M+N)! + (M+N-1)!\binom{N}{1}z + (M+N-2)\binom{N}{2}z^2$$
$$+ \cdots + M!\binom{N}{N}z^N \Big/$$
$$(M+N)! - (M+N-1)!\binom{M}{1}z + (M+N-2)\binom{M}{2}z^2$$
$$+ \cdots + (-1)^M (N)!\binom{M}{M}z^M \qquad (F)$$

The function $\xi_{N,M}$ can be written in a doubly infinite Pade table

$$\begin{array}{cccc}
\xi_{0,0} & \xi_{1,0} & \xi_{2,0} & \cdots \\
\xi_{0,1} & \xi_{1,1} & \xi_{2,1} & \cdots \\
\xi_{0,2} & \xi_{1,2} & \xi_{2,2} & \cdots \\
\vdots & \vdots & \vdots &
\end{array}$$

Any sequence of these functions converges to e^z for any z as $\lim (M + N) \to \infty$.

All poles of (F) are simple and for M not differing considerably from N, all are in the right half plane. Although this has been used for the sake of simplicity, it can be shown that left half plane poles of (F) do not alter the results.

Inverting (D) into (C), the approximation $\hat{V}(t)$ of $V(t)$ becomes,

$$\hat{V}(t) = \frac{1}{2\pi jt} \int_{C'-j\infty}^{C'+j\infty} V(z/t)\xi_{N,M}(z)dz. \tag{G}$$

Integral (G) can be evaluated by residue calculus by closing the path of integration along an infinite arc either to the right or to the left. In order that arc does not contribute to the integral M,N is taken such that the function

$$F(z) = V(z/t)\xi_{N,M}(z) \tag{H}$$

has at least two more finite poles than zeros. Then

$$\oint F(z)dz = \pm 2\pi j \sum (\text{residue at poles inside the closed path}). \tag{I}$$

where the positive sign applies when the path is closed in the left half plane (counter-clockwise), whereas the negative one applies for the other case. Important properties of the method will be derived by alternatively closing the paths in both half planes. For $N < M$,

$$\xi_{N,M}(z) = \sum_{i=1}^{M} K_i/z - z_i \tag{J}$$

where z_i are the poles of $\xi_{N,M}(z)$ and k_i are the corresponding residues.

Closing the path of integration around the poles of $\xi_{N,M}(z)$ in the right half plane,

$$\hat{V}(t) = -1/t \sum_{i=1}^{M} K_i \, V(z_i/t) \tag{K}$$

Real time functions can be evaluated using only the poles z_i in the upper half plane. For M even

$$\hat{V}(t) = -1/t \sum_{i=1}^{M'} k_i \, V(z_i/t) - 1/t \sum_{i=1}^{M'} K_i^* \, V(z_i^*/t)$$

$$= -1/t \sum_{i=1}^{M'} 2\mathrm{Re}[K_i \, V(z_i/t)]$$

$$= -1/t \sum_{i=1}^{M'} \mathrm{Re}[K_i' \, V(z_i/t)] \qquad\qquad \text{(L)}$$

where $M' = M/2$ and $K_i' = 2K_i$, k_i is defined by (J) when M is odd, $M' = (M + 1)/2$ and $K_i' = k_i$ for the residue corresponding to the real pole.

Periodic functions are calculated correctly for small times irrespective of whether they are smooth, with discontinuities or with discontinuities in their derivatives. The error grows until it reaches a region where it fluctuates between two limits. Some of these functions cannot be described by Taylor series expansion and thus the accuracy for larger times cannot be expected to be good, nevertheless the error remains bounded.

Delayed functions with discontinuities in their response or in their derivatives have largest errors at the point of discontinuity. The phenomenon is the well known Gibbs effect and the error remains reasonably small afterwards. Smooth delayed functions do not exhibit the Gibbs phenomenon and the error is small. Unstable functions with poles in the right half plane are inverted correctly for small times but then the error grows without bound.

Manual verification of algorithm for $(F(s) = 1/s, f(t) = 1)$
Take $N = 2$

$$V_i = (-1)^{(N/2)+i} \sum_{k=(i+1)/2}^{\mathrm{Min}(i,N/2)} \frac{(K)^{N/2}(2k)!}{((-N/2) - k)!k!(k - 1)!(i - k)!(2k - i)!}$$

$$V_1 = (-1)^{1+1} \sum_{k=1}^{1} \frac{1!\,(2x1)!}{(1 - 1)!1!(1 - 1)!(1 - 1)!(2 \times 1 - 1)!} = 2$$

Similarly,

$$V_2 = -2$$

$$\overline{f_N(t)} = \frac{\ln 2}{t} \sum_{i=1}^{N} V_i F(i \ln 2/t)$$

$$\overline{f_2(t)} = \frac{\ln 2}{t} \left[2 \times \frac{t}{\ln 2 \times 1} - 2\frac{t}{2 \times \ln 2} \right]$$

$$= \frac{\ln 2}{t} \times \frac{t}{\ln 2} [2 - 1] = 1$$

Thus we see that we get an exact numerical inversion of the given Laplace transform.

Possible Application

Consider the solution to the one dimensional diffusion equation

$$\frac{\partial^2 V}{\partial x^2} = k^2 \frac{\partial v}{\partial t} \tag{18}$$

with boundary conditions

$$V(0, t) = V_0 u(t) \tag{19}$$

where $u(t)$ is the unit step function and

$$V(\infty, t) < \infty \tag{20}$$

and initial condition

$$V(x, 0) = 0 \tag{21}$$

Laplace transform of (18) using (21) gives an ordinary differential equation,

$$\frac{\partial^2 \bar{V}}{\partial x^2} - k^2 s \bar{V} = 0 \tag{22}$$

which has the solution

$$\bar{V}(x, s) = Ae^{-kx\sqrt{s}} + Be^{kx\sqrt{s}} \tag{23}$$

Application of the boundary condition yields

$$\bar{V}(x,s) = \frac{V_0}{S} e^{-kx\sqrt{s}} \tag{24}$$

The exact inverse given by contour integration in the S-plane is

$$V(x,t) = V_0 \operatorname{erfc}[-kx/2\sqrt{t}] \tag{25}$$

Equation (24) can be inverted approximately using the algorithm.

COMPUTER IMPLEMENTATION

The algorithm is implemented on the computer in the package LINV. This package does the following:

(a) Calculate all V_i's.
(b) Calculate the averaged approximate inverse transform.
(c) Calculates the error if the inverse of the Laplace Transform is known.
(d) Calculates the maximum error and also the bounds of errors for various values of N.

The above algorithm has been made more accurate by the use of a sophisticated multiplication algorithm MUT. This does increase processing time; but this is a one time expense. Once the tables of V_i's for different values of N are ready, one only has to feed these precomputed numbers.

Thus, the program achieves a very important aim of error analysis of the algorithm.

Calculations are made for several functions, the error calculated, and an attempt made to localize these errors.

CONCLUSIONS

Several important conclusions can be made:

(a) It was found that with increasing N the number of correct figures first increases nearly linearly and then owing to rounding errors,

decreases linearly. The optimum N is approximately proportional to the number of digits the machine is working with.

(b) The nature of the Gaver-Stehfest algorithms with large coefficients and alternating signs makes it necessary to subtract very large numbers of almost the same size, which creates word-length computational problems. Particular choices of N were made because of the limitation on N due to the finite word length of the digital processor used. With 32 bits, we get a 7 decimal digit resolution. In cases where $N = 18$, the inverse function will demonstrate considerable scatter due to truncation error in the processor.

(c) It is clear that functions with oscillatory inverses present difficulty to the method.

(d) There is considerable improvement due to the multiplication algorithm used.

(e) Transforms for lightly damped sinusoids and Bessel functions require large mainframe computers with long word-length.

The accuracy of the method was considered from the theoretical and empirical points of view. It was found that the accuracy is high for small time. Examples were used to illustrate the idea.

References

[1] Stehfest, H., "Numerical inversion of the Laplace Transforms", *Communications of the ACM*, **13**(1), 47–49, January, 1970.

[2] Papoulis, A. (1957). "A new method of inversion of the Laplace Transform", *Qt. Appl. Math.*, **14**(4), 405–414.

[3] Dubner, H. and Abate, J. (1968). "Numerical inversion of Laplace Transform and the finite Fourier Cosine Transform", *J. Ass. Comput. Mach.*, **15**(1), 115–123.

[4] Salzer, H. E. (1958). "Tables for the numerical calculation of inverse Laplace Transforms", *J. Math. Phys.*, **37**, 89–109.

[5] Wing, O. (1967). "An efficient method of numerical inversion of Laplace Transform", *Arch. Comput.*, **2**, 153–166.

[6] Bellman, R., Kalaba, R. E. and Lockett, J. A. (1966). *"Numerical inversion of the Laplace Transform"*, American Elsevier, New York.

[7] Vlach, J. (1969). "Numerical Method for transient responses of Linear networks with lumped, distributed or mixed parameters", *J. Franklin Inst.*, **288**, 91–113.

[8] Singhal, K. and Vlach, J. (1971). "Program for numerical inversion of Laplace Transform", *Electronics Letters*, **7**, 413–415.

[9] Nakhla, M. R., Singhal, K. and Vlach, J., "Simple method for numerical inversion of Laplace Transform", *Proc. Sixteenth Midwest Symposium on Circuit Theory*, pp. XIV. 5-1-9, April, 1973.

[10] Wellekens, C. J. (1970). "Generalization of Vlach's method for numerical inversion of the Laplace Transform", *Electronics Letters*, **5**, 742–744.

[11] Zakian, V., "Numerical inversion of Laplace Transform", *Electronics Letters*, pp. 120–121, March, 1969.

[12] Piessens, R., "A Bibliography on numerical inversion of Laplace Transform", Report TW20, *Applied Mathematics and Programming Division*, Katholieke Universiteit Leuven, Belgium, July, 1974.

[13] Wood, K. C., "TI-59, inverts Laplace Transforms for time domain analysis", *Electronics*, **53**(22), 178–179, October 9, 1980.

[14] Wyatt, M. A., "Home Computer Displays inverse Laplace Transform", *Electronics*, **54**(6), 163–164, March 24, 1981.

[15] Librach, H., "Calculator plots time response of inverse Laplace Transforms", *Electronics*, **54**, 149, July 14, 1981.

[16] Jacquot, R. G., Steadman, J. W. and Rhodine, C. N., "Gaver-Stehfest Algorithm for Approximate inversion of Laplace Transforms", *IEEE Circuits and Systems Magazine*, pp. 4–8, March, 1983.

[17] Kishore Singhal and Jiri Vlach "Computation of time domain response by Numerical Inversion of the Laplace Transform", *Journal of the Franklin Institute*, **299**(2), 109–126, February, 1976.

[18] Krylov, V. I. and Skoblya, N. S. (1977). "*Handbook of Numerical Inversion of Laplace Transforms*", Moscow, MIR Publishers.