PUC

Series: Monografias em Ciência da Computação

Nº 1/80

A METHOD FOR THE NUMERICAL INVERSION OF LAPLACE TRA[redacted]

G. Honig
U. Hirdes

DEPARTAMENTO DE INFORMÁTICA

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225 - CEP 22453
Rio de Janeiro — Brasil

PUC/RJ - DEPTO. DE INFORMÁTICA

Series: Monografias em Ciência da Computação, Nº 1/80

Series Editor: Daniel A. Menascé March, 1980

A METHOD FOR THE NUMERICAL INVERSION OF LAPLACE TRANSFORMS*

G. Hönig+
U. Hirdes++

* This work have been submitted for publication elsewhere. As a courtesy to the publisher it should not be widely distributed.

+ Depto. de Informática-PUC/RJ (on leave from Kernforschungsanlage Jülich).

++ Institut für Festkörperforschung, Kernforschungsanlage Jülich.

Contents

Abstract

A Method for the Numerical Inversion of Laplace Transforms

In this paper a numerical inversion method for Laplace transforms, based on a Fourier series expansion developed by Durbin (1973), is presented. The disadvantage of the inversion methods of that type, the encountered dependence of discretization and truncation error on the free parameters, is removed by the simultaneous application of a procedure for the reduction of the discretization error, a method for accelerating the convergence of the Fourier series and a procedure that computes approximately the 'best' choice of the free parameters. Suitable for a given problem, the inversion method allows the adequate application of these procedures. Therefore, in a big range of applications a high accuracy can be achieved with only a few function evaluations of the Laplace transform. The inversion method is implemented as a FORTRAN subroutine.

1. Introduction

The significance of numerical Laplace inversion is obvious from the big range of applications. Well known in engineering Laplace transformation methods are also used in order to solve differential and integral equations and to assist when other numerical methods are applied (see [7], [1o]).

A number of numerical inversion methods has been developed during the last few years. In what follows we confine ourselves to the methods using Fourier series approximations. Many tests (see [4], [5], [8], [2], [1]) have demonstrated the simplicity and the accuracy of these methods.

Fourier series were first used by Dubner and Abate [4] in 1968 for the numerical inversion of Laplace transforms. Durbin | 5 | improved the method in 1973.

Other authors, Simon et al. [8] in 1972, Veillon [9] in 1974 and Crump | 2 | in 1976, used different acceleration methods in order to speed up the convergence of the Fourier series derived in | 4 | and | 5 |. Some of these methods at times achieve a considerable reduction of the truncation error. However, they fail in other cases and their efficiency heavily depends on the choice of the parameters of the methods of Durbin or Dubner and Abate; and the choice of these parameters was somewhat arbitrary.

The biggest disadvantage of the above mentioned methods is the dependence of the discretization and truncation errors on the free parameters: by a suitable choice of these parameters the discretization error becomes arbitrarily small, but at the same time the truncation error grows to infinity and vice versa. A first step in surmounting this problem was taken in |1|, 1977 (see part 3), where the so-called "Korrektur"-method was presented. It allows a remarkable reduction of the discretization error without increasing the truncation error.

Nevertheless the accuracy of the "Korrektur"-method also depends on a 'good' choice of the free parameters. The procedure introduced in part 5 gets closer to the solution of the problem. It allows the approximate computation of optimal parameters for all above mentioned methods (for the definition of "optimal" see part 5).

Part 4 describes a new method for the acceleration of convergence of the Fourier series. It is applicable if the infinite series for the approximation of the inversion integral does not alternate (see part 4). In this case all other tested acceleration methods fail.

Finally, in parts 2 and 4, those of the above mentioned inversion and acceleration methods that were found to be best by numerical tests as well as by the derived error estimates are summarized.

A new algorithm for the numerical Laplace inversion, taking into consideration the new,above mentioned,procedures and implemented as a FORTRAN subroutine [1], is described in part 6.

This subroutine can be used as a "black box": the only input parameters are the t-values for which $f(t) = L^{-1}[F(s)]$ shall be computed and of course the Laplace transform $F(s)$. On the other hand - for a concrete problem - the user can make an optimal choice of all free parameters in order to have the most efficient combination of the algorithms contained in the subroutine.

The efficiency of the inversion method is shown by the examples in part 7.

[1] See part 8, subroutine LAPIN (LAPlace INversion).

2. The method of Durbin

The Laplace transform of a real function $f: R \to R$ with $f(t) = o$ for $t<o$ and its inversion formula are defined as

$$F(s) = L[f(t)] = \int_{o}^{\infty} e^{-st} f(t)\, dt \, , \qquad (1)$$

$$f(t) = L^{-1}[F(s)] = \frac{1}{2\pi i} \int_{v-i\infty}^{v+i\infty} e^{st} F(s)\, ds \qquad (2)$$

with $s = v + iw$; $v, w \in R$.

$v \in R$ is arbitrary, but greater than the real parts of all the singularities of $F(s)$. The integrals in (1) and (2) exist for $Re(s) > a \in R$ if

(a) f is locally integrable,

(b) there exist a $t_o \geq o$ and k , $a \in R$ such that $|f(t)| \leq ke^{at}$ for all $t \geq t_o$, (3)

(c) for all $t \in (o,\infty)$ there is a neighbourhood in which f is of bounded variation.

In the following we always assume that f fulfils the conditions (3) and in addition that there are no singularities of $F(s)$ to the right of the origin. Therefore (1) and (2) are defined for all $v>o$. The possibility to choose $v>o$ arbitrarily, is the basis of the methods of Dubner/Abate and Durbin. The latter is now described.

Using the inversion formula (see [5])

$$f(t) = \frac{e^{vt}}{\pi} \int_{o}^{\infty} [Re\{F(s)\} \cos(wt) - Im\{F(s)\} \sin(wt)] dw \, , \qquad (4)$$

which is equivalent to (2), and a Fourier series expansion of $h(t) = e^{-vt} f(t)$ in the interval $[o,2T]$, Durbin derived the approximation formula

$$f(t) = \frac{e^{vt}}{T} \Big[-\frac{1}{2} \operatorname{Re}\{F(v)\} + \sum_{k=o}^{\infty} \operatorname{Re}\{F(v + i \frac{k\pi}{T})\} \cos \frac{k\pi}{T} t - \sum_{k=o}^{\infty} \operatorname{Im}\{F(v + i \frac{k\pi}{T})\} \sin \frac{k\pi}{T} t\Big] - F1(v,t,T) \tag{5}$$

for $o < t < 2T$.

$F1(v,t,T)$ is the discretization error, given by

$$F1(v,t,T) = \sum_{k=1}^{\infty} e^{-2vkT} f(2kT + t) . \tag{6}$$

Since there are no singularities of $F(s)$ in the right halfplane we have[2)] a $c > o$, $m \geq o$ and a $t_o \geq o$, such that

$$|f(t)| \leq ct^m \quad \text{for all} \quad t \geq t_o . \tag{7}$$

From (6) and (7) the following estimates for the discretization error can be deduced (see [5]):

a) $m = o$

$$|F1(v,t,T)| \leq \frac{c}{e^{2vT} - 1} \tag{8}$$

b) $m > o$

$$|F1(v,t,T)| \leq K(2T)^m e^{-2vT} \left(\frac{\alpha_1}{2vT} + \dots + \frac{\alpha_{m+1}}{(2vT)^{m+1}}\right) \tag{9}$$

where $K, \alpha_1, \dots, \alpha_{m+1} \in R$.

[2)] See Doetsch [3], Bd. 1

These estimates show that the discretization error can be made arbitrarily small by choosing v sufficiently large.

As the infinite series in (5) can only be summed up to a finite number N of terms, there also occurs a truncation error given by

$$FA(N,v,t,T) = \frac{e^{vt}}{T} \Big[\sum_{k=N+1}^{\infty} \{ \mathrm{Re}\{F(v + \frac{ik\pi}{T})\} \cos \frac{k\pi}{T} t - \mathrm{Im}\{F(v + \frac{ik\pi}{T})\} \sin \frac{k\pi}{T} t \} \Big] . \tag{1o}$$

Hence the approximate value for $f(t)$ is

$$f_N(t) = \frac{e^{vt}}{T} \Big[-\frac{1}{2} \mathrm{Re}\{F(v)\} + \sum_{k=o}^{N} \{ \mathrm{Re}\{F(v + \frac{ik\pi}{T})\} \cos \frac{k\pi t}{T} - \mathrm{Im}\{F(v + \frac{ik\pi}{T})\} \sin \frac{k\pi t}{T} \} \Big] \tag{11}$$

3. The reduction of the discretization error by the "Korrektur"-method

It is obvious from (8) and (9) that the discretization error can be made arbitrarily small if the product vT is sufficiently large.

Unfortunately, the truncation error (1o) may diverge for large values of vT.

The "Korrektur"-method allows a reduction of the discretization error without enlarging the truncation error.
With (11) equation (5) can be written in the form

$$f(t) = f_{\infty}(t) - F1(v,t,T) \quad . \tag{12}$$

The "Korrektur"-method uses the approximation formula

$$f(t) = f_{\infty}(t) - e^{-2vT} f_{\infty}(2T+t) - F2(v,t,T) \quad . \tag{13}$$

As stated in the theorem below, the discretization error $F2(v,t,T)$ is much smaller than $F1(v,t,T)$.

Taking into consideration the truncation error (1o), we find that the approximate value for $f(t)$, using the "Korrektur"-method (13), is given by

$$f_{NK}(t) = f_N(t) - e^{-2vT} f_{N_O}(2T + t) \quad . \tag{14}$$

It should be mentioned that the truncation error of the "Korrektur"-term $e^{-2vT} f_{\infty}(2T + t)$, is much smaller than $FA(N,v,t,T)$ if $N=N_O$. We can therefore choose $N_O < N$ [3),]

[3)] see part 6.

which means that only a few additional function evaluations of F(s) are necessary to achieve a considerable reduction of the discretization error. This error reduction is pointed out in the following theorem.

Theorem 3.1 : Suppose f is a real function with $f(t) = o$ for $t < o$ that possesses the properties (3) and $F(s) = L[f(t)]$ its Laplace transform that has no singularities to the right of the origin, and suppose the "Korrektur"-formula (13) is used for the numerical inversion of $F(s)$, then

a) $$|F2(v,t,T)| \leq \frac{2c}{e^{2vT}(e^{2vT} - 1)} \qquad (15)$$

if $m = o$,

b) $$|F2(v,t,T)| \leq \qquad (16)$$

$$3^m \; e^{-2vT}\{K(2T)^m \; e^{-2vT}(\frac{\alpha_1}{2vT} + \ldots + \frac{\alpha_{m+1}}{(2vT)^{m+1}})\}$$

if $m > o$ and $(m!)/2^m - 1 \leq 2vT$.

For the definition of $c, m, K, \alpha_1, \ldots, \alpha_{m+1}$ see equations (7), (8) and (9). For the proof see [1] or [6].

Remarks: A comparison with the method of Durbin (see (8) and (9)) shows a reduction of the error bound by the factor $2e^{-2vT}$ for $m=o$ and $3^m e^{-2vT}$ for $m>o$. The condition $m!/2^m - 1 \leq 2vT$ might be difficult to fulfill for large m , and hence the application of the "Korrektur"-method is not recommended in such cases.

As mentioned before, an adequate reduction of the total error can be obtained by the "Korrektur"-method only if (for fixed N and T) the parameter v is suitable. This is illustrated in Fig. 1. It shows the error curves of the method of Durbin (F1 and FA) and of the "Korrektur"-method (F2 and FA). A successful application of the "Korrektur"-method requires a $v < v_o$. For $v > v_o$ the truncation error dominates and the reduced discretization error F2 does not lead to a noticeable reduction of the total error.

Fig. 1

The opposite holds if the methods for the acceleration of convergence, that we describe in the following part, are applied. Acceleration of convergence is only sensible if $v \geq v_o$ or, which is the same, if the truncation error dominates.

However, a simultaneous application of an acceleration method and the "Korrektur"-method (as realized in the subroutine LAPIN) is recommended, if the parameters are optimally chosen by the procedures introduced in part 5.

4. Acceleration of convergence

In this chapter three acceleration methods used in the subroutine LAPIN (see chapter 8) are briefly described: The ε-algorithm, the minimum-maximum method and a method based on curve fitting. The latter is new, whereas the ε-algorithm was already used in [2] and [9], the minimum-maximum method in [1] in order to speed up the convergence of the series approximating the Laplace inversion integral. We have also tested other acceleration methods such as the Euler transformation, or Aitken's extrapolation procedure (see [8]). But these turned out to be less efficient than the above mentioned methods.

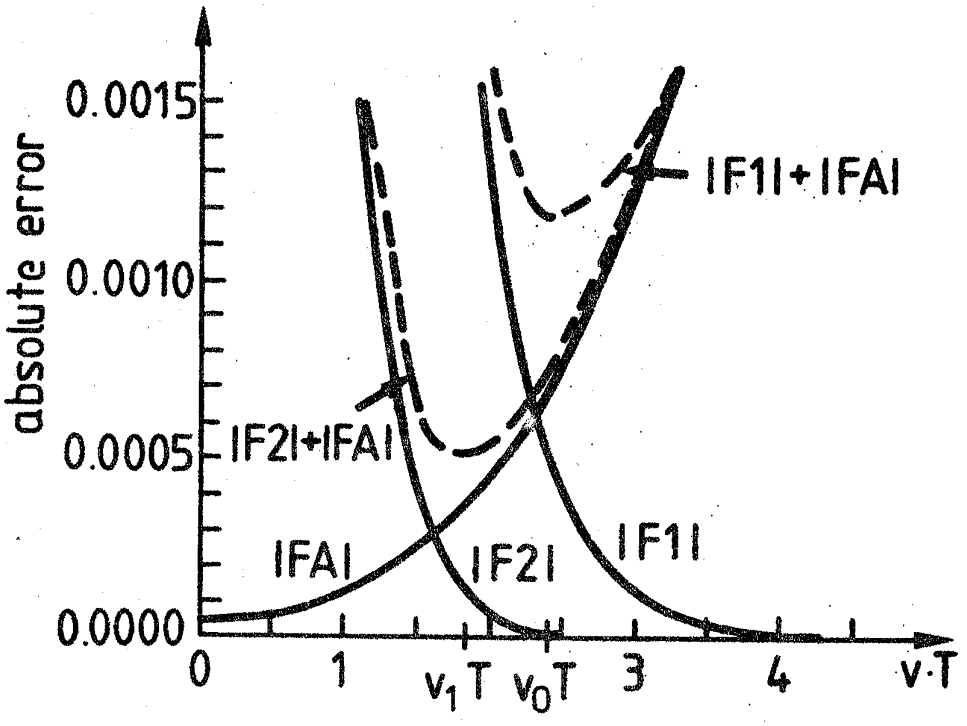
We can consider $f_N(t)$ (see (11)) for fixed t as a discrete function of N and define:

$f_N(t)$ as a function of N is monotonous if

$$|f_N(t) - f_\infty(t)| \geq |f_M(t) - f_\infty(t)|$$

for all N, M with $N \leq M$.

For a non-monotonous $f_N(t)$ the ε-algorithm (EPAL) and the minimum-maximum method (MINIMAX) in general significantly increase the rate of convergence. However, they fail - as do the Euler transformation and Aitken's extrapolation procedure - for a monotonous $f_N(t)$. The method based on curve fitting (CFM) leads in this case to a considerable improvement in the results.


absolute error
0.0015
0.0010
0.0005
0.0000
0
1
v_1T
v_0T
3
4
$v \cdot T$
|F1|+|FA|
|F2|+|FA|
|FA|
|F2|
|F1|

With

$$c_k := \frac{e^{vt}}{T} \left[\mathrm{Re}\{F(v + \frac{ik\pi}{T})\} \cos \frac{k\pi t}{T} - \mathrm{Im}\{F(v + \frac{ik\pi}{T})\} \sin \frac{k\pi t}{T}\right] ,$$

$$k = o,1,2,...$$

we can replace (11) by

$$f_N(t) = \frac{1}{2} c_o + \sum_{k=1}^{N} c_k . \qquad (17)$$

The ε-*algorithm* applied to (17) is defined as follows.

Let $N = 2q + 1$, $q \in N$,

$$s_m := \sum_{k=1}^{m} c_k \text{ and}$$

$$\begin{aligned} \varepsilon_{p+1}^{(m)} &:= \varepsilon_{p-1}^{(m+1)} + 1/(\varepsilon_p^{(m+1)} - \varepsilon_p^{(m)}) \\ \varepsilon_o^{(m)} &:= o \qquad (18) \\ \varepsilon_1^{(m)} &:= s_m , \end{aligned}$$

then the sequence $\varepsilon_1^{(1)}, \varepsilon_3^{(1)}, \varepsilon_5^{(1)}, \ldots, \varepsilon_{2q+1}^{(1)} = \varepsilon_N^{(1)}$, converges to $f_\infty(t) - c_o/2$. For a non-monotone $f_N(t)$ in general it converges faster than the sequence of the partial sums s_m , $m = 1,2,...$ of the untransformed series.

The *minimum-maximum method* also increases the rate of convergence only in the case of a non-monotonous $f_N(t)$. The method works as follows. Having found three neighbouring stationary values of $f_N(t)$ as a function of N , say a maximum at $N = N1$ and $N = N3$, and a minimum at $N = N2$, an interpolating function for the maximum values at $N = N1$ and $N = N3$ is constructed. The mean value of the minimum at $N = N2$ and the function value of the interpolating function at N = N2 yields the new approximation for $f_\infty(t)$. For a more detailed description of MINIMAX see [1].

The *method based on curve fitting* (CFM), applicable only for monotonous $f_N(t)$, consists in fitting the parameters of any function that has a horizontal asymptote $\gamma_A(x) \equiv \gamma$, by demanding that this function is an interpolating function for the points $(N, f_N(t))$, $0 \leq N_0 \leq N \leq N_1$. The function value of the asymptote γ is the desired approximation for $f(t)$. With the simple rational function

$$r(x) := \frac{\alpha}{x^2} + \frac{\beta}{x} + \gamma , \tag{19}$$

For example, we achieved high accuracy already for small N_1. The CFM fills an important gap: now it is also possible to speed up the convergence of the Fourier series (5) in case of a monotonous $f_N(t)$ [4]. The subroutine LAPIN (see part 8) automatically chooses between CFM and EPAL (MINIMAX) depending on whether $f_N(t)$ is a monotonous or a non-monotonous function of N. Tests have shown that for non-monotonous $f_N(t)$ EPAL mostly is superior to MINIMAX in accuracy. However, we found examples where the ε-algorithm falsifies the results, but the minimum-maximum method did not. In these cases, the subroutine LAPIN chooses by itself the MINIMAX method to accelerate the convergence of the series (for more details, see part 8, significance of the parameter H). Furthermore, the ε-algorithm is not applicable for arbitrarily large N in (17), because of overflows occuring in (18) for larger N.

[4] For instance, $f_N(t_o)$ often is a monotonous function of N, if f(t) is a step function with the discontinuity in t_o.

5. The choice of optimal parameters

We already mentioned that a good choice of the free parameters N and $CON = vT$ is not only important for the accuracy of the results but also for the application of the "Korrektur"-method and the methods for the acceleration of convergence (see Fig. 1). These methods do not improve the result if the parameters are chosen badly.

Two methods are now presented which approximately determine the 'optimal' v for fixed N and T. The main difference between the two methods is the definition of 'optimal parameters'.

Definition A: For fixed N and T the parameter v of the method of Durbin (see (12)), with or without the application of the "Korrektur"-method or a method for the acceleration of convergence, is optimal if the absolute values of discretization and truncation error are equal.

Definition B: For fixed N and T the parameter v of the above mentioned methods (see definition A) is optimal if the sum of the absolute values of discretization and truncation error is minimal (see Fig. 1, where the optimal v is given by v_o and v_1).

Method A: Let $\bar{f}_N(t)$, $\bar{f}_{NK}(t)$ be the approximate values for $f(t)$ computed by the method of Durbin (12) and the "Korrektur"-method (13) respectively. The bar indicates that one of the methods for the acceleration of convergence may have been applied.

Let $R(N,v,t,T)$ be the expression in the brackets on the right side of (1o). Neglecting the dependence of v in R we can write for fixed t,T : $R(N,v,t,T) \cong R(N)\cdot\delta$, where $\delta \in [-1,+1]$ indicates the application of an acceleration method ($\delta \equiv 1$ if none of the acceleration methods is used). The truncation error of $\bar{f}_N(t)$ or $\bar{f}_{NK}(t)$ is therefore given by

$$FA(N,v,t,T) \cong \frac{e^{vt}}{T} R(N)\delta \ . \qquad (2o)$$

With (5), (6) and (2o) we find

$$\bar{f}_N(t) \cong f(t) - \frac{e^{vt}}{T} R(N)\delta + O(e^{-2vT}) \ . \qquad (21)$$

Let v_1, v_2 be large and $v_1 \neq v_2$ [5], then

$$\bar{f}_N^1(t) - \bar{f}_N^2(t) \cong R(N)\delta\cdot(e^{v_2 t} - e^{v_1 t})/T \qquad (22)$$

or

$$R(N)\delta \cong T \frac{\bar{f}_N^1(t) - \bar{f}_N^2(t)}{e^{v_2 t} - e^{v_1 t}} \ . \qquad (23a)$$

Similarly we get for the "Korrektur"-method

$$R(N)\delta \cong T \frac{\bar{f}_{NK}^1(t) - \bar{f}_{NK}^2(t)}{e^{v_2 t} - e^{v_1 t}} \ . \qquad (23b)$$

The discretization error (6) of the method of Durbin can be written as [3]

$$F1(v,t,T) = e^{-2vT} f(2T + t) + O(e^{-4vT}) \ . \qquad (24a)$$

[5] We found that for example $v_1 :\mp 2o$, $v_2 := v_1 - 2$ is a good choice.

From (13) we find for the discretization error of the "Korrektur"-method

$$F2(v,t,T) = \sum_{k=2}^{\infty} e^{-2vTk}\{f(2kT+t) - f(2T(3k-2)+t)\} .$$

Hence [3)]

$$F2(v,t,T) = e^{-4vT}\{f(4T+t) - f(8T+t)\} + O(e^{-6vT}) . \quad (24b)$$

Using definition A, the equations (23) and (24) easily lead to the following equations for the optimal parameter $v = v^A_{OPT}$:

$$v^A_{OPT} \cong - \frac{1}{2T + t} \ln \left|\frac{R(N)\delta}{T\,\bar{f}_N(2T+t)}\right| , \quad (25a)$$

(method of Durbin)

and

$$v^A_{OPT} \cong - \frac{1}{4T+t} \ln \left|\frac{R(N)\delta}{T\{\bar{f}_N(4T+t) - \bar{f}_N(8T+t)\}}\right| \quad (25b)$$

("Korrektur"-method)

Because of definition A an upper bound for the total absolute error $|f(t)-\bar{f}_N(t)|$ or $|f(t)-\bar{f}_{NK}(t)|$ for $v=v^A_{OPT}$ is approximately given by

$$TERR \leqq 2\,\frac{e^{v^A_{OPT} \cdot t}}{T}\,|R(N)\delta| .$$

<u>Method B</u>: We now use definition B and assume that $R(N,v,t,T)$ is no longer a constant function of v to get a more accurate approximation formula for v_{OPT} .

The dependence of v is assumed to be as follows

$(R(N,v,t,T) := R(N,v)\delta$ because t,T are fixed, $v_1>o$ arbitrary):

$$R(N,v)\delta \cong R(N,v_1)\delta \cdot \Big[\mathrm{Re}\{F(v + i\tfrac{N\pi}{T})\}\cos\tfrac{N\pi t}{T} - \mathrm{Im}\{F(v + i\tfrac{N\pi}{T})\}\sin\tfrac{N\pi t}{T}\Big]\Big/\Big[\mathrm{Re}\{F(v_1 + i\tfrac{N\pi}{T})\}\cos\tfrac{N\pi t}{T} - \mathrm{Im}\{F(v_1 + i\tfrac{N\pi}{T})\}\sin\tfrac{N\pi t}{T}\Big] . \tag{27}$$

The acceleration factor δ can be computed very easily from (23a) without additional function evaluations:

$$\delta = \frac{\bar{f}_N^1(t) - \bar{f}_N^2(t)}{f_N^1(t) - f_N^2(t)} . \tag{28}$$

Let $AR(v_1,v,N)$ be the expression in brackets on the right side of (27) and $R(N,v_1)\delta := R(N)\delta$, where $R(N)\delta$ is computed by method A. Then (27) is expressible in the form

$$R(N,v)\delta = R(N)\delta \cdot AR(v_1,v,N) . \tag{29}$$

Definition B implies (for the method of Durbin):

$$\frac{\partial}{\partial v}\Big[\,|R(N,v)\delta|\frac{e^{vt}}{T} + |F1(v,t,T)|\,\Big]_{|v=v_{OPT}^B} = o . \tag{3o}$$

Approximating the derivative of $R(N,v)$ by finite differences, we find the following iteration procedure to solve equation (3o):

$$\Big[\frac{\partial}{\partial v}R(N,v)\Big]^{(i)} \cong \frac{R(N,v^{(i-1)}) - R(N,v_1^{(i-1)})}{v^{(i-1)} - v_1^{(i-1)}} ,\quad i = 1,2,\ldots,s , \tag{31a}$$

$$v^{(o)} = v_{OPT}^A ,\quad v_1^{(o)} = v_1 \tag{31b}$$

$$v_1^{(i)} = v^{(i-1)} ,\quad i = 1,\ldots,s-1 . \tag{31c}$$

$$v^{(i)} = -\frac{1}{2T + t}\ \ln\left[\frac{\frac{\partial}{\partial v}|R(N,v)^{(i)}|\ \ \delta + |R(N,v^{(i-1)})|\delta\cdot t}{2T^2|\bar{f}_N(2T + t)|}\right],$$

$$i = 1,2,\ldots,s\ . \qquad (31d)$$

In (31b) v^A_{OPT} and v_1 are defined by method A (see equation (25) and (22) respectively). Equation (31d) follows from (3o) substituting (31a) for the derivative of $R(N,v)$ and (24a) for $F1(v,t,T)$.

Hence the optimal parameter $v = v^B_{OPT}$ for fixed N,T,t is given by

$$v^B_{OPT} := v^{(s)} \qquad (32)$$

An analogous result holds for the "Korrektur"-method: Replacing F1 by F2 in (3o), we only have to substitute (31d) by

$$v^{(i)} = -\frac{1}{4T + t}\ \ln\left[\frac{[\frac{\partial}{\partial v}\ |R(N,v)^{(i)}|\ \ \delta + |R(N,v^{(i-1)})|\ \delta\cdot t}{4T^2|\bar{f}_N(4T+t) - \bar{f}_N(8T+t)|}\right],$$

$$i = 1,2,\ldots,s\ . \qquad (31e)$$

It is sufficient to choose $s=1$ or $s=2$, as numerical tests have shown.

Again it is possible to compute approximately an upper bound for the total error, if the parameter $v = v^B_{OPT}$ is used for the computation of $\bar{f}_N(t)$ [6].

[6] A similar equation holds for $\bar{f}_{NK}(t)$.

We find

$$\mathrm{TERR} \cong \frac{e^{v^B_{OPT}\cdot t}}{T} \mid R(N, v^B_{OPT}) \mid \delta + e^{-2v^B_{OPT}\cdot T} \mid \bar{f}_N(2T+t) \mid . \quad (33)$$

<u>Remarks</u>: The simplifications made in order to derive the equations (25) and (31) do in general not cause any difficulties. Besides a few exceptions, which are discussed later, the optimal parameters v^A_{OPT} and v^B_{OPT} are a good approximation for the true optimal parameters.

If the ε-algorithm is used in order to accelerate the convergence of the series (5), it is not possible to compute the acceleration factor δ for large N with the desired accuracy (the ε-algorithm breaks down after a certain $N_o < N$ depending on v,t and T ; N_o is not known in advance; see also the final remarks of part 4). Therefore the minimum-maximum method allows a more accurate determination of the optimal parameters by method A or B, and hence of the total error (26) or (33).

6. Remarks about the use of the algorithm

First some input parameters of the subroutine LAPIN are described (see also part 8):

T1, TN Lower and upper bound of the interval in which $f(t)$ shall be approximated.

IMAN The choice of this parameter decides whether the free parameters of the subroutine are placed automatically or not,

IMAN = o automatical,

IMAN = 1 manual choice of parameters.

ILAPIN If ILAPIN = 1 , the approximate value $f_N(t)$ (see 11) is computed with T=t , if ILAPIN = 2 with T=TN . For T=t the computation of sine and cosine terms and of the imaginary parts of $F(s)$ in (11) is cancelled, on the other hand $\mathrm{Re}\{F(s)\}$ becomes dependent on t . Hence ILAPIN = 1 is recommended if the inverse is computed only for a few t-values.

IKONV IKONV = 1 implies the use of the acceleration method MINIMAX, IKONV = 2 the acceleration of convergence with EPAL. In both cases the subroutine automatically chooses the curve fitting method (CFM) for monotonous $f_N(t)$.

ICON If this parameter is zero (ICON = o) the parameter v is not optimally chosen by method A or B (see part 5). For ICON = 1 the subroutine LAPIN chooses an optimal v for $t = t_{[N/2]}$ (see part 8) and uses this v for the computation of $f(t)$ in the whole interval.

If ICON = 2 an optimal v is computed for all $t_k \varepsilon (T1,TN)$ (see part 8 for the definition of t_k).

IKOR IKOR = 1 leads to the application of the "Korrektur"-method, IKOR = o to the approximation of f(t) without the "Korrektur"-method.

The following table shows the possible combinations of the above described algorithms offered by the subroutine LAPIN.

	ILAPIN	ICON	IKOR	IKONV
IMAN = 1	1	o/1/2	o/1	1/2
	2	o/1	o/1	1/2
IMAN = o	1	1	o	2

Table 1

The method A for an optimal choice of the free parameters is used if ILAPIN = 2 , method B if ILAPIN = 1 .

The application of the "Korrektur"-method is not recommended first, in the case that the absolute value of the "Korrektur"-term f(2T + t) is small or equal to zero: $|f(2T + t)| << 1$, (no noticeable improvement of the accuracy occurs), and second, if f(t) is a rapidly increasing function as $t \to \infty$ (see the remarks to theorem 3.1).

The same holds for the application of the methods for an optimal choice of the free parameters if the auxiliary quantities (these are $f(2T+t)$ in (24a) and $f(4T+t)$, $f(8T+t)$, in (24b)) are equal to zero. Although the denominators in (25) do not vanish (because of the discretization and truncation errors), the computed optimal parameters might be misleading. In these cases, it is very helpful to have a global conception about the Laplace inverse $f(t)$. If not already known, this can be obtained easily by the subroutine LAPIN choosing either IMAN = o or IMAN = 1, ILAPIN = 2, ICON = o, IKOR = o and IKONV = 1. The latter combination of the parameters coincides with the method of Durbin, at which the MINIMAX-method is used to speed up the convergence of the Fourier series.

Let NS1 be the number of function evaluations of $F(s)$ used to approximate $f(t)$ and NS2 the number of function evaluations used to approximate the "Korrektur" term $f(2T+t)$. For given NS1 we found that $NS1/2 \leq NS2 \leq NS1$ if $NS1 \leq 100$ and $NS1/10 \leq NS2 \leq NS1/2$ if $NS1 \geq 100$ is a good choice for NS2.

As stated above, occuring overflows in (18) often make it impossible to transform a given number NS1 of terms of the series (17) by the ε-algorithm. If it breaks down after $N < NS1$ iterations the accuracy is diminished with that the optimal parameters and the total error (33) are calculated. The MINIMAX-algorithm does not cause such problems.

Finally, we would like to mention that it can be necessary to know the s-values in advance for that $F(s)$ has to be evaluated (for instance, if $F(s)$ is a not analytically known solution of a differential or integral equation). These s-values are defined by T and $CON = \nu \cdot T$ ($s = \nu+ik\pi/T$, $k = o(1)NSUM$). In Table 2, the values of CON and T, for that the 'auxiliary quantities' ($f_{NS1}(T)$) must be computed, are listed.

T	t_o	t_o	$2t_1+t_o$	$4t_1+t_o$	$8t_1+t_o$
CON	20	18	5	5	5

	ILAPIN = 1
	ICON = 1
		ILAPIN = 2
		ICON = 1
			ILAPIN = 1
			ICON = 2

t_o	$T1+(TN-T1)\left[\frac{N}{2}\right]/(N+1)$	$T1+(TN-T1)\left[\frac{N}{2}\right]/(N+1)$	$T1+k(TN-T1)/(N+1)$, $k=1(1)N$ [7]
t_1	$T1+(TN-T1)\left[\frac{N}{2}\right]/(N+1)$	TN	$T1+k(TN-T1)/(N+1)$, $k=1(1)N$ [7]

Table 2

If ICON = o no auxiliary quantities are needed. If ICON > o, IKOR = o requires the computation of $f_{NS1}(t_o)$ and $f_{NS1}(2t_1+t_o)$, IKOR = 1 the computation of $f_{NS1}(t_o)$, $f_{NS1}(4t_1+t_o)$ and $f_{NS1}(8t_1+t_o)$ with the above given values of CON, t_o and t_1.

[7] N = number of t-values for that f(t) shall be approximated.

7. Numerical examples [8)]

From the following examples one can get some idea of the accuracy of the subroutine LAPIN and the possibilities it offers to find an 'optimal' solution for a given problem by a suitable choice of the different algorithms described above. The parameters used in the following tables are defined in part 6. The code numbers refer to the parameters ILAPIN-ICON-IKOR-IKONV (1-2-o-1, for example, means: ILAPIN=1, ICON=2, IKOR=o and IKONV=1). IMAN=o coincides with 1-1-o-2.

Table 3 shows the errors occuring if $F(s) = s(s^2+1)^{-2}$ is inverted. Three different parameter combinations of LAPIN are

$F(t) = t \cdot \sin(t)/2$, $F(s) = s(s^2+1)^{-2}$

method	Durbin (v·T=5)	L A P I N			
Code		2-1-o-1	IMAN=o	1-2-1-1	
NS1+NS2	2ooo	3o	6o	16o+4o	
		real absolute error			by (33)
t 1	0.4 D-02	0.2 D-02	0.1 D-11	0.351 D-10	0.273 D-10
3	0.6 D-01	0.1 D-01	0.1 D-11	0.211 D-09	0.236 D-09
5	0.1 D-01	0.2 D-02	0.1 D-12	0.573 D-09	0.648 D-09
7	0.8 D-01	0.9 D-02	0.2 D-09	0.786 D-09	0.979 D-09
9	0.2 D-01	0.9 D-03	0.1 D-10	0.774 D-09	0.191 D-08
11	0.1 D-01	0.3 D-03	0.2 D-10	0.590 D-09	0.273 D-08
13	0.6 D-01	0.2 D-01	0.9 D-10	0.107 D-08	0.372 D-08
15	0.5 D-02	0.1 D-01	0.2 D-09	0.175 D-08	0.356 D-08
17	0.7 D-02	0.2 D-01	0.1 D-09	0.225 D-08	0.569 D-08
19	0.1 D-00	0.5 D-02	0.4 D-10	0.321 D-09	0.633 D-08
CPU-time sec (for t=1(1)20)	1.21	0.04	0.42	1.14	

Table 3

[8)] All calculations were performed in double precision on the IBM 370/168 computer of KFA Jülich.

compared with the method of Durbin. With only a few function evaluations of F(s) and little CPU-time, LAPIN computes the Laplace inverse with a considerable accuracy. For instance, the first two columns show that the method of Durbin needs about 60 times more function evaluations and 30 times more CPU-time than LAPIN to get a comparable accuracy. Column 3 shows - and this was confirmed by many other examples - that using the subroutine LAPIN as a 'black box' (IMAN=o) yields excellent results. From a comparison between columns 5 and 6 it is obvious that the absolute error computed by (33) is a good approximation for the real error (such an error estimation is only possible if ICON=2; it is most accurate for IKONV=1).

As a second example, the Laplace transform of the step function

$$U(t-1o) := \begin{cases} o & t < 1o \\ o.5 & t = 1o \\ 1 & t > 1o \end{cases}$$

was inverted (see Table 4). Comparing the results of Durbin with

f(t) = U(t-1o) , F(s) = exp(-1os)/s

method	Durbin (v·T=5)	L A P I N				LAPIN
Code		2-1-o-1	IMAN=o			1-2-1-2
NS1+NS2	2ooo	5o	6o			35o+15o
	real absolute error					real absolute error
t 5	0.6 D-02	0.4 D-05	0.5 D-06		t 9.o	0.1 D-13
6	0.6 D-02	0.1 D-04	0.5 D-06		9.2	0.3 D-12
7	0.6 D-02	0.6 D-04	0.5 D-06		9.4	0.8 D-14
8	0.7 D-02	0.5 D-03	0.5 D-06		9.6	0.1 D-09
9	0.7 D-02	0.1 D-01	0.6 D-06		9.8	0.6 D-05
1o	0.6 D-02	0.5 D-04*	0.6 D-06*		1o.o	0.3 D-09*
11	0.5 D-02	0.3 D-01	0.8 D-05		1o.2	0.1 D-05
12	0.6 D-02	0.6 D-02	0.5 D-06		1o.4	0.8 D-10
13	0.6 D-02	0.1 D-02	0.5 D-06		1o.6	0.2 D-09
14	0.6 D-02	0.1 D-02	0.5 D-06		1o.8	0.8 D-10
15	0.6 D-02	0.2 D-02	0.5 D-06		11.o	0.2 D-12
CPU-time sec (for t=1(1)2o)	2.02	0.06	0.32		CPU-time sec	13.7o

* CFM was used to accelerate convergence.

Table 4

those obtained from LAPIN we come to the same conclusion as in the example above. The high accuracy at t = 1o is possible only if the curve-fitting method is used to speed up the convergence of the series. EPAL and MINIMAX fail because $f_N(1o)$ is monotonous in N . But also at points very close to the discontinuity at t = 1o a high accuracy is obtained by LAPIN as demonstrated in the last column of Table 4.

$f(t) = (-t^3+9t^2-18t+6)/6$, $F(s) = (s-1)^3/s^4$

method	LAPIN			LAPIN		
code	1-2-o-2 (without Korrektur)			1-2-1-2 (with Korrektur)		
NS1+NS2	1oo			6o+4o		
	absolute error		$v_{OPT}\cdot T$	absolute error		$v_{OPT}\cdot T$
	real	by (33)		real	by (33)	
t 1	0.1 D-10	0.2 D-10	12.2	0.3 D-12	0.4 D-14	9.4
3	0.9 D-10	0.1 D-10	14.7	0.6 D-12	0.5 D-13	9.9
6	0.1 D-09	0.2 D-10	15.4	0.2 D-12	0.2 D-12	10.1
9	0.5 D-10	0.7 D-10	16.0	0.8 D-13	0.1 D-12	10.5
CPU-time sec (t=1(1)1o)	1.49			0.81		

Table 5

From Table 5 one gets some idea of the effect of the Korrektur method. It can be seen that (with the same number of function evaluations) the Korrektur method (right part of Table 5) is clearly more accurate. This is not only caused by the reduction of the discretization error. The optimal parameter $v_{OPT}\cdot T$ is smaller if the Korrektur method is applied. Hence the truncation error is reduced too.

As a final example, we show in Table 6 the dependence of the optimal parameter v_{OPT} on the number NS1 of function evaluations (v_{OPT} also depends on t). It is obvious that primarily

$f(t) = 1-\exp(t)\mathrm{erfc}(\sqrt{t})$; $F(s) = 1/(s\sqrt{s}+1))$

method code	LAPIN 1-2-o-2		
NS1	absolute error at t=1 real	by (33)	$v_{OPT} \cdot T$
1o	0.103 D-04	0.416 D-05	6.61
2o	0.352 D-10	0.900 D-10	11.96
3o	0.356 D-11	0.801 D-11	13.16

Table 6

the possibility to calculate v_{OPT} for a given NS1 (and NS2) makes the inversion methods based on Fourier series expansions accurate and efficient.

Acknowledgement

The authors wish to thank K. Mika for valuable suggestions and comments.
For making possible the stay at the Departamento de Informática of Pontifícia Universidade Católica (PUC), Rio de Janeiro, the first author wishes to thank the director of the department, Prof. Carlos José Pereira de Luçena, the Conselho Nacional de Desenvolvimento Científicio e Tecnológico (CNPq), Gesellschaft für Mathematik und Datenverarbeitung Bonn (GMD), PUC Rio de Janeiro and KFA Jülich.

References

[1] Albrecht, P., Honig, G.: Die numerische Inversion der Laplace-Transformierten. Angewandte Informatik Nr. 8, 336-345 (1977).

[2] Crump, K.S.: Numerical inversion of Laplace transforms using a Fourier series approximation. J. ACM Vol. 23, Nr. 1, 89-96 (1976).

[3] Doetsch, G.: Handbuch der Laplace-Transformation. Bd. I,II,III, Verlag Birkhäuser, Basel 195o.

[4] Dubner, H., Abate, J.: Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. J. ACM Vol. 15, Nr. 1, 115-123 (1968).

[5] Durbin, F.: Numerical inversion of Laplace transforms: an effective improvement of Dubner and Abate's method. The computer Journal, Vol. 17, Nr. 4, 371-376 (1973).

[6] Honig, G.: Zur numerischen Lösung partieller Differentialgleichungen mit Laplace-Transformation (Diss. Uni. Dortmund D29o). Berichte der Kernforschungsanlage Jülich-Jül-155o, November 1978.

[7] Kehr, K., Honig, G., Richter, D.: Stochastic theory of spin depolarization of muons diffusing in the presence of traps. Z. Phys. B 32, 49-58 (1978).

[8] Simon, R.M., Stroot, M.T., Weiss, G.H.: Numerical inversion of Laplace transforms with application to percentage labeled experiments. Computers and Biomed. Rev. 5, 596-6o7 (1972).

[9] Veillon, F.: Quelques méthodes nouvelles pour le calcul numérique de la transformée inverse de Laplace. Th. U. de Grenoble, März 1972.

[1o] Warzee,G.: Application de la transformée de Laplace et de la méthode des éléments finis à la résolution de l'équation de conduction thermique instationnaire. C. r. Acad. Sci., Paris, Séri. A278, 1265-1266 (1974).

8. Appendix

The FORTRAN subroutine LAPIN

(a) Purpose

Suppose f(t) is a real function and F(s) its Laplace transform, that has no singularities to the right of the origin, then the subroutine LAPIN calculates from F(s) for a programmers-chosen set of t-values, the corresponding values for f(t) , using the above described algorithms.

(b) Usage

```
CALL LAPIN(F,T1,TN,N,IMAN,ILAPIN,IKONV,NS1,NS2,ICON,IKOR,
                                                CON,H,E,IER)
```

(c) Description of the parameters

Type:

```
INTEGER*4   N,IMAN,ILAPIN,IKONV,NS1,NS2,ICON,IKOR,IER
REAL*8      T1,TN,CON,H,E
COMPLEX*16  F
```

Input parameters: T1,TN,N,IMAN,ILAPIN,IKONV,NS1,NS2,ICON,IKOR,CON

Output parameters: H,IER

Significance of the parameters .

F Laplace transform - external function, to declare as EXTERNAL in the calling program;

T1,TN lower and upper bound of the interval in which $f(t)$ shall be approximated;

N number of t-values for which $f(t)$ shall be computed, the t-values are given by

$t_k = T1 + (TN-T1)k/(N+1)$, $k = 1(1)N$;

IMAN = o , all further parameters are placed automatically;

= 1 , a manual choice of the following parameters is possible;

ILAPIN ILAPIN = 1 implies the application of the approximation formula with $T=t$,

ILAPIN = 2 with $T = TN$ ($T=2TN$ for the "Korrektur"-terms);

IKONV if IKONV = 1 the minimum-maximum method, if IKONV = 2 the ε-algorithm is used to accelerate the convergence of the series;

NS1 number of function evaluations of $F(s)$ used to approximate $f(t)$ $(f_N(t) = f_{NS1}(t))$;

NS2 number of function evaluations of $F(s)$ used to approximate the "Korrektur"-term $f(2T+t)$ $(f_N(2T+t) = f_{NS2}(2T+t))$;

ICON = o no optimal choice of the free parameters,

= 1 optimal choice of the free parameters for $t = t_{[N/2]}$

= 2 optimal choice of the free parameters for $t = t_k$, $k = 1(1)N$;

IKOR = o no application of the "Korrektur"-method,

= 1 application of the "Korrektur"-method;

CON by the choice of CON = vT , the free parameter v is determined in case that ICON = o ;

H matrix of dimension 6 by N , contains on return in the

<u>1st row</u> the approximation values for $f(t_k)$, $k = 1(1)n$;

<u>2nd row</u> work area;

<u>3rd row</u> the computed optimal parameter $CON_{OPT} = v_{OPT} \cdot T$ for $t = t_k$, $k = 1(1)N$, ($CON_{OPT} = 18$ if the truncation error (23) is zero, $CON_{OPT} = 1$ if the discretization (24) is zero or of very small absolute value);

<u>4th row</u> the code for the used acceleration method:

= o no acceleration of convergence,

= 1 MINIMAX,

= 2 EPAL,

= 3 CFM,

= P $\geq$ 4 EPAL, overflow occured after P-2 iterations (NS1= P-2 values of F(s) are used for the approximation of f(t)): note: the output parameter on the 4th row of H may be different from the input parameter IKONV . The subroutine LAPIN always uses

(a) CFM if $f_N(t)$ is monotonous in N ,

(b) MINIMAX if the application of EPAL falsifies the results,

(c) no acceleration method if only 1 or 2 stationary values of $f_N(t)$ as a function of N are found;

<u>5th row</u> absolute error calculated by (33) if ICON = 2 and ILAPIN = 1; if ICON = 1 the error estimation is valid only at $t = t_{[N/2]}$;

<u>6th row</u> contains a control number of up to 6 digits $(n_1 n_2 n_3 n_4 n_5 n_6)$, the digits refer to the used auxiliary quantities, the "Korrektur"- term $f_{NS2}(2t_1+t_o)$ and to

$f_{NS1}(t_o)$ as follows (see Table 2 for the definition of t_o and t_1):

$n_1 \rightarrow f^1_{NS1}(t_o)$ (CON = 2o)

$n_2 \rightarrow f^2_{NS2}(t_o)$ (CON = 18)

$n_3 \rightarrow f_{NS1}(2t_1+t_o)$

$n_4 \rightarrow f_{NS1}(4t_1+t_o)$

$n_5 \rightarrow f_{NS1}(8t_1+t_o)$

$n_6 = n_{61}+n_{62}$; $n_{61} \rightarrow f_{NS1}(t_o)$; $n_{62} \rightarrow f_{NS2}(2t_1+t_o)$.

If one of the digits is zero the application of the ε-algorithm falsifies the results of the corresponding auxiliary quantity (MINIMAX is used). Otherwise the digits are equal to 1 (besides n_{62}, which is equal to 2).

Note: $n_3 = o$ always if IKOR = 1, $n_4 = n_5 = o$ always if IKOR = o.

Whenever a zero in the control number indicates the falsification of the results, it is recommended to increase NS1 or NS2.

E matrix of dimension 3 by NS1, work area;

IER error parameter, control of input data:

= o no error ,

= 1 TN < T1

= 1o N < 1 ,

= 1oo IMAN < o or IMAN > 1 ,

= 1ooo ILAPIN < 1 or ILAPIN > 2 ,

= 1oooo IKONV < 1 or IKONV > 2 ,

= 1ooooo NS1 < 1 or (NS2 < 1 and IKOR = 1) ,

= 1oooooo ICON < o or ICON > 2 or (ICON = 2 and ILAPIN = 2) ,

= 1ooooooo IKOR < o or IKOR > 1 ,

= 1oooooooo CON ≤ o .

E.g. IER = 11 means that 1 and 1o occured.

```
      SUBROUTINE LAPIN(F.T1.TN.N.IMAN.ILAPIN.IKONV.NS1.NS2.ICON.
     *                 IKOR.CON.H.E.IER)
C
      IMPLICIT REAL*8(A-H.O-Z)
      COMPLEX*16 F.FUNC
      INTEGER*4 RICHT.RICHTA.HMONO
      EXTERNAL F
      DIMENSION H(6.N).E(3.NS1).E3(3)
C
      CALL MASKE
      IER = 0
      IF (TN.LT.T1) IER = 1
      IF (N.LT.1) IER = IER+10
      I = IMAN+1
      GO TO (100.110).I
      IER = IER+100
      GO TO 120
C
C PARAMETERS FOR IMAN = 0
  100 IF (IER.NE.0) GO TO 120
      ILAPIN = 1
      IKONV = 2
      ICON = 1
      IKOR = 0
      NS1 = 60
      NS2 = 0
      GO TO 200
C
C IMAN = 1
  110 IF (ILAPIN.LT.1 .OR. ILAPIN.GT.2) IER = IER+1000
      IF (IKONV.LT.1 .OR. IKONV.GT.2) IER = IER+10000
      IF (NS1.LT.1 .OR. (NS2.LT.1 .AND. IKOR.EQ.1)) IER = IER+100000
      IF (ICON.LT.0 .OR. ICON.GT.2 .OR. (ICON.EQ.2 .AND. ILAPIN.EQ.2))
     *   IER = IER+1000000
      IF (IKOR.LT.0 .OR. IKOR.GT.1) IER = IER+10000000
      IF (ICON.EQ.0.AND.CON.LE.0.D0) IER = IER+100000000
      IF (IER.EQ.0) GO TO 200
  120 WRITE(6.1) IER.T1.TN.N.IMAN.ILAPIN.IKONV.NS1.NS2.ICON.IKOR.CON
    1 FORMAT(///' *** ERROR ***'.8X.'IER ='.I12//
     * ' T1      I '.D10.3.18X.'1 I  TN < T1'/
     * ' TN      I '.D10.3/
     * ' N       I '.I10.17X.'10 I  N < 1'/
     * ' IMAN    I '.I10.16X.'100 I  IMAN < 0  OR  IMAN > 1'/
     * ' ILAPIN  I '.I10.15X.'1000 I  ILAPIN < 1  OR  ILAPIN > 2'/
     * ' IKONV   I '.I10.14X.'10000 I  IKONV < 1  OR  IKONV > 2'/
     * ' NS1     I '.I10.13X.'100000 I  NS1 < 1  OR  (NS2 < 1  AND  '.
     *                                 'IKOR = 1)'/
     * ' NS2     I '.I10/
     * ' ICON    I '.I10.12X.'1000000 I  ICON < 0  OR ICON > 2  OR'.
     *                                 '  (ICON = 2  AND  ILAPIN = 2)'/
     * ' IKOR    I '.I10.11X.'10000000 I  IKOR < 0  OR  IKOR > 1'/
     * ' CON     I '.D10.3.10X.'100000000 I  CON <= 0'//)
      RETURN
C
  200 PI = 4.D0*DATAN(1.D0)
      CON1 = 20.D0
      CON2 = CON1-2.D0
      ABSF = 0.D0
C
      J3 = (3-ICON)/2+N*(ICON/2)
      TA = T1
      TB = TN
C
      DO 830 L3=1.J3
      HMONO = 0
      KOR1 = IKOR
C
C COMPUTATION OF THE OPTIMAL PARAMETERS
  210 IF (ICON-1) 360.230.220
  220 TA = T1+L3*(TN-T1)/(N+1)
      TB.= TA
```

```
C
  230 NH = N/2
      T = TA+(TB-TA)*NH/(N+1)
      TK = (2-ILAPIN)*T+(ILAPIN-1)*TB
C
C COMPUTATION OF THE TRUNCATION ERROR (RNSUM)
      TO = T
      CON = CON1
      JUMP = 1
      GO TO 370
  240 FN = H(1,L3)
      FNS1 = E(1,NS1)
      CON = CON2
      JUMP = 2
      GO TO 370
  250 IF (FN.NE.H(1,L3).AND.FNS1.NE.E(1,NS1)) GO TO 255
      CONOPT = CON
      ABSF = 0.D0
      GO TO 320
  255 RNSUM = TK*(FN-H(1,L3))/(DEXP(CON1)-DEXP(CON2))
      IF (ILAPIN.EQ.2) GO TO 260
C
C COMPUTATION OF THE ACCELERATION FACTOR (DEL)
      RACC = T*(FNS1-E(1,NS1))/(DEXP(CON1)-DEXP(CON2))
      DEL = RNSUM/RACC
C
  260 IF (IKOR.EQ.1) GO TO 280
C
C OPTIMAL PARAMETERS (METHOD A)
      TO = 2.D0*TK+T
      CON = CON1/4.D0
      JUMP = 3
      GO TO 370
  270 FN = H(1,L3)
      CONOPT = -TK/(2.D0*TK+T)*DLOG(DABS(RNSUM/(TK*FN)))
      GO TO 310
C
C OPTIMAL PARAMETERS FOR THE KORREKTUR METHOD (METHOD A)
  280 TO = 4.D0*TK+T
      CON = CON1/4.D0
      JUMP = 4
      GO TO 370
  290 FN = H(1,L3)
      TO = 8.D0*TK+T
      JUMP = 5
      GO TO 370
  300 FN = FN-H(1,L3)
      CONOPT = -TK/(4.D0*TK+T)*DLOG(DABS(RNSUM/(TK*FN)))
C
C OPTIMAL PARAMETERS (METHOD B)
  310 IF (ILAPIN.EQ.1) GO TO 315
      ABSF = DABS(DEXP(CONOPT)*RNSUM*2.D0/TK)
      GO TO 320
  315 V1 = CON1/T
      V2 = CONOPT/T
      EINS = 1.D0
      IF (NS1/2*2.NE.NS1) EINS = -1.D0
      RNSUMK = RNSUM*(DREAL(F(DCMPLX(V2,PI*NS1/T)))*EINS)/
                     (DREAL(F(DCMPLX(V1,PI*NS1/T)))*EINS)
     *
      ABRN = (RNSUMK-RNSUM)/(V2-V1)
      CONOPT = -DLOG(DABS((ABRN/T+RNSUMK)/(T*(IKOR*2+2)*FN)))/
                (3.D0+2.D0*IKOR)
     *
      V1 = V2
      V2 = CONOPT/T
      RNSUMK = RNSUMK*(DREAL(F(DCMPLX(V2,PI*NS1/T)))*EINS)/
                     (DREAL(F(DCMPLX(V1,PI*NS1/T)))*EINS)
     *
      ABSF = DEXP(CONOPT)/T*DABS(RNSUMK)+
             DABS(DEXP(-2.D0*CONOPT)*FN*(IKOR-1)
     *
                  +DEXP(-4.D0*CONOPT)*FN*IKOR)
     *
C
  320 IF (CONOPT.LE.0.D0) CONOPT = 1.D0
C
```

```
C LAPLACE-INVERSION WITH OPTIMAL PARAMETERS
C
      TO = TA
      TT = TB
      I1 = L3
      J1 = (2-ICON)*N+(ICON-1)*L3
      CON = CONOPT
      KOR1 = IKOR
      JUMP = 6
      GO TO 380
C
  360 TO = T1
      TT = TN
      I1 = 1
      J1 = N
      KOR1 = IKOR
      JUMP = 6
      GO TO 380
C
  370 KOR1 = 0
      TT = TO
      I1 = L3
      J1 = L3
C
  380 DELTA = (TT-TO)/DFLOAT(N+1)
      K2 = 1
      NSUM = NS1
C
C COMPUTATION OF THE T-VALUES FROM TO,TT
  400 IF (ILAPIN.EQ.1) GO TO 420
      TE = K2*TT
      V = CON/TE
      RAL = -0.5D0*DREAL(F(DCMPLX(V,0.D0)))
      PITE = PI/TE
C
  405 DO 410 L=1,NSUM
      W = (L-1)*PITE
      FUNC = F(DCMPLX(V,W))
      E(2,L) = DREAL(FUNC)
      E(3,L) = DIMAG(FUNC)
  410 CONTINUE
C
  420 DO 800 K1=I1,J1
C
      TL = TO+K1*DELTA
      IF (ILAPIN.EQ.2) GO TO 440
C
      IF (K2.EQ.2) TL = 3.D0*TL
      V = CON/TL
      FAKTOR  =  DEXP(V*TL)/TL
      RAL = -0.5D0*DREAL(F(DCMPLX(V,0.D0)))
      PIT = PI/TL
      EINS = 1.D0
      SURE = 0.D0
C
C METHOD OF DURBIN
  425 DO 430 L=1,NSUM
      W = (L-1)*PIT
      SURE = SURE+DREAL(F(DCMPLX(V,W)))*EINS
      EINS = -EINS
      E(1,L) = FAKTOR*(RAL+SURE)
  430 CONTINUE
      GO TO 460
C
  440 IF (K2.EQ.2) TL = TL+TE
      FAKTOR = DEXP(V*TL)/TE
      SURE = 0.D0
      SUIM = 0.D0
      DO 450 L=1,NSUM
      W = (L-1)*PITE
      SURE = SURE+E(2,L)*DCOS(W*TL)
      SUIM = SUIM+E(3,L)*DSIN(W*TL)
      E(1,L) = FAKTOR*(RAL+SURE-SUIM)
  450 CONTINUE
C
C SEARCH FOR STATIONARY VALUES
  460 NMAX = NSUM*2/3
      MONOTO = 0
      K = 0
      RICHTA = DSIGN(1.5D0,(E(1,NSUM)-E(1,NSUM-1)))
      DO 500 L=1,NMAX
      J = NSUM-L
      RICHT = DSIGN(1.5D0,(E(1,J)-E(1,J-1)))
      IF (RICHT.EQ.RICHTA) GO TO 500
      K = K+1
      E3(K) = E(1,J)
      IF (K.EQ.3) GO TO 510
      RICHTA = RICHT
  500 CONTINUE
      IF (K.EQ.0) GO TO 700
      H(K2,K1) = E(1,NSUM)
      IF (K2.EQ.1) H(4,K1) = 0
      GO TO 790
  510 KE = 1
  520 RICHTA = RICHT
      KE = 3-KE
      IF ((E3(KE)-E(1,J))*RICHTA.GT.0.D0) GO TO 560
  540 IF (J.LE.NSUM/3) GO TO 550
      J = J-1
      RICHT = DSIGN(1.5D0,(E(1,J)-E(1,J-1)))
      IF (RICHT-RICHTA) 520,540,520
  550 MONOTO = 1
      IF (IKONV.EQ.2) GO TO 630
C
C MINIMUM-MAXIMUM METHOD (MINIMAX)
  560 H(K2,K1) = (E3(1)+E3(3))/4.D0+E3(2)/2.D0
      IF (K2.EQ.1) H(4,K1) = 1
      GO TO 790
C
C EPSILONALGORITHM (EPAL)
  630 K = 0
      NSUMM1 = NSUM-1
      E2 = E(1,1)
      DO 660 L=1,NSUMM1
      E1 = E(1,1)
      TM = 0.D0
      LP1 = L+1
      DO 650 M=1,L
      MM = LP1-M
      TM1 = E(1,MM)
      DIVI = E(1,MM+1)-E(1,MM)
      IF (DABS(DIVI).GT.1.D-20) GO TO 640
      K = L
      GO TO 670
  640 E(1,MM) = TM+1.D0/DIVI
      TM = TM1
  650 CONTINUE
      E2 = E1
  660 CONTINUE
C
  670 IF (DABS(E1).GT.DABS(E2)) E1 = E2
      IF(DABS(E(1,1)).GT.DABS(E1)) E(1,1) = E1
      H(K2,K1) = E(1,1)
      IF (K2.EQ.1) H(4,K1) = K+2
      GO TO 790
C
C CURVE FITTING (CFM)
  700 X1 = NSUM-2.D0
      X2 = NSUM-1.D0
      X3 = NSUM-0.D0
      Y1 = E(1,NSUM-2)
      Y2 = E(1,NSUM-1)
      Y3 = E(1,NSUM)
      B = ((Y3-Y1)*X3*X3*(X1+X2)/(X1-X3)
     *     -(Y2-Y1)*X2*X2*(X1+X3)/(X1-X2))/(X3-X2)
      A = ((Y2-Y1)-B*(X1-X2)/(X1*X2))*(X2*X2*X1*X1)/(X1*X1-X2*X2)
      H(K2,K1) = Y1-(A/X1+B)/X1
      MONOTO = 1
      IF (K2.EQ.1) H(4,K1) = 3
C
C
  790 H(3,K1) = CON
      H(5,K1) = ABSF
      HMONO = HMONO+K2*MONOTO*10**(6-JUMP)
      IF (JUMP.LT.6) GO TO 800
      H(6,K1) = (2-K2)*HMONO+(K2-1)*(2*MONOTO+H(6,K1))
      HMONO = HMONO/10*10
C
  800 CONTINUE
      IF (KOR1.EQ.0) GO TO (240,250,270,290,300,830),JUMP
      IF (K2.EQ.2) GO TO 810
      K2 = 2
      NSUM = NS2
      GO TO 400
C
C KORREKTUR METHOD
  810 FAKTOR = -DEXP(-2.D0*CON)
      DO 820 K=I1,J1
      H(1,K) = H(1,K)+FAKTOR*H(2,K)
  820 CONTINUE
C
      GO TO (240,250,270,290,300,830),JUMP
  830 CONTINUE
      RETURN
      END
```