DEMONSTRATIO MATHEMATICA
Vol. XXIX No 4 1996

Halina Grabarska

AN APPROXIMATION OF SOLUTIONS OF THE FIRST FOURIER PROBLEM FOR THE HEAT EQUATION WITH AN APPLICATION OF THE ORTHOGONALIZATION PROCEDURE

1. Introduction

In the papers [1], [2] some methods for solving approximately the Fourier problems for certain parabolic equations have been proposed. Those methods, based on the least squares method, led to linear systems of equations.

In this paper the above-mentioned methods are applied to a numerical analysis of the first Fourier problem for the heat equation in one space variable. In view of "bad" properties of the systems received, a new method is proposed. An orthogonalization of a system of functions forming an approximate solution basis is an important tool in this new method.

2. Preliminaries

In [3] it was proved that the polynomials

$$h_j(x,t) = j!\sum_{k=0}^{[j/2]} \frac{x^{j-2k}t^k}{(j-2k)!k!}, \qquad j = 0,1,2,\ldots, \tag{1}$$

form a complete system in the space of strong solutions of the heat equation (by a strong solution we mean that it is of class C^2 with respect to x and of class C^1 with respect to t)

$$h_t = h_{xx}. \tag{2}$$

In order to solve the initial boundary value problem (the first Fourier problem) for (2) with conditions

$$\begin{cases} h(-1,t) = f_1(t), & 0 < t < t_0, \\ h(1,t) = f_2(t), & 0 < t < t_0, \\ h(x,0) = g(x), & -1 \le x \le 1 \end{cases} \tag{3}$$

we can apply an approximate method based on the least squares method (see [1], [2]). Thus we will seek the approximate solution $h^N(x,t)$ as a linear combination of the first $N+1$ functions of the form (1), i.e.,

$$h^N(x,t) = \sum_{j=0}^{N} c_j h_j(x,t), \tag{4}$$

where the coefficients $c_0, \ldots, c_N$ are evaluated from the linear system of $N+1$ equations received by minimizing the function

$$S(c_0,\ldots,c_N) = \int_0^{t_0} [f_1(\tau) - h^N(-1,\tau)]^2 d\tau + \int_0^{t_0} [f_2(\tau) - h^N(1,\tau)]^2 d\tau + \int_{-1}^{1} [g(\xi) - h^N(\xi,0)]^2 d\xi. \tag{5}$$

That system has the form

$$Ac = b, \tag{6}$$

where

$$A = [a_{ij}]_{i,j=0}^N, \qquad c = [c_0,\ldots,c_N]^T, \qquad b = [b_0,\ldots,b_N]^T,$$

$$a_{ij} = \int_0^{t_0} [h_i(-1,\tau)h_j(-1,\tau) + h_i(1,\tau)h_j(1,\tau)]d\tau + \int_{-1}^{1} g(\xi)h_j(\xi,0)d\xi,$$

$$b_j = \int_0^{t_0} [f_1(\tau)h_j(-1,\tau) + f_2(\tau)h_j(1,\tau)]d\tau + \int_{-1}^{1} g(\xi)h_j(\xi,0)d\xi.$$

After some calculations, we get

$$a_{ij} = \begin{cases} 0, & \text{when } i+j \text{ is odd}, \\ 2\left[\dfrac{1}{i+j+1} + i!j!\displaystyle\sum_{l=0}^{[i/2]}\sum_{k=0}^{[j/2]} \dfrac{t_0^{k+l+1}}{(j-2k)!(i-2l)!k!l!(k+l+1)}\right], & \text{when } i+j \text{ is even}. \end{cases} \tag{7}$$

3. Conditioning of the linear system of equation

Our goal will be a numerical analysis of the system (6) and in particular an analysis of the conditioning of that system. In general, a numerical problem is ill-conditioned, if "small" changes in the problem input data produce "large" changes in the solution (see for example [4], [6]). Usually for a numerical problem a so-called condition number is defined. It is the ratio of variations in solutions to input data perturbation. Certainly, an attempt to solve problems with a big condition number is very risky. Due to roundoff errors, one can obtain results completely different from the exact solution. In the case of linear systems of equations the condition number is defined by the formula (see for example [5])

$$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|, \tag{8}$$

where $\| \cdot \|$ is a matrix norm.

Let us come back to our system (6). The explicit form (7) of the elements of the matrix A may suggest the ill-conditioning of the system (6), because the term $\frac{1}{i+j+1}$ occurs in the well-known Hilbert matrix (see for example [5]). Actually the condition numbers of the system (6) are big. It is illustrated in the case of the spectral norm $\|A\|_2 = \max_{\lambda \in Sp(A^T A)} \sqrt{\lambda}$ by the following table.

	N	cond(A)
$t_0 = 0$	2	14
	5	$1.9 \cdot 10^3$
	10	$9.4 \cdot 10^6$
$t_0 = 0.5$	2	4
	5	$2.7 \cdot 10^3$
	10	$4.7 \cdot 10^9$
$t_0 = 1$	2	10
	5	$2.1 \cdot 10^4$
	10	$5.2 \cdot 10^{11}$
$t_0 = 2$	2	35
	5	$2.4 \cdot 10^5$
	10	$7.2 \cdot 10^{13}$
$t_0 = 5$	2	$2.9 \cdot 10^2$
	5	$6.7 \cdot 10^6$
	10	$9.2 \cdot 10^{16}$

4. Orthogonalization

It is obvious that, dealing with so ill-conditioned problem, one should seek the approximation $h^N(x,t)$ in another way than by solving the system (6). One of such ways is the Gram-Schmidt orthogonalization of the system of functions (1) with respect to the scalar product

$$(f,g) = \int_0^{t_0} f(-1,\tau)g(-1,\tau)d\tau + \int_0^{t_0} f(1,\tau)g(1,\tau)d\tau + \int_{-1}^{1} f(\xi,0)g(\xi,0)d\xi. \tag{9}$$

It is clear that after the orthogonalization, i.e., substituting the functions $h_0, \ldots, h_N$ by functions $g_0, \ldots, g_N$ satisfying

$$(g_i, g_j) = \begin{cases} 0 & \text{for } i \neq j, \\ e_j & \text{for } i = j,\ e_j \neq 0, \end{cases}$$

the approximation $h^N(x,t)$ will be given by the formula

$$h^N(x,t) = \sum_{j=0}^{N} d_j g_j(x,t), \tag{10}$$

where

$$d_j = \frac{1}{e_j}\Big[\int_0^{t_0} [f_1(\tau) g_j(-1,\tau) + f_2(\tau) g_j(1,\tau)] d\tau + \int_{-1}^{1} g(\xi) g_j(\xi, 0) d\xi \Big]. \tag{11}$$

The Gram-Schmidt procedure is given by the formula

$$\begin{cases} g_0(x,t) = h_0(x,t), \\ g_i(x,t) = h_i(x,t) - \sum_{j=0}^{i-1} \frac{(h_i, g_j)}{(g_j, g_j)} g_j(x,t), \quad i = 1, 2, \ldots, N, \end{cases} \tag{12}$$

where $(\cdot, \cdot)$ is the scalar product (9).

As we see, the most important point here is to evaluate scalar products (h_i, g_j) and (g_j, g_j). We will deal with this problem in next two sections.

5. Recurrence formulae for the polynomials $h_j(x,t)$

We will now determine recurrence relations for the polynomials $h_j(x,t)$ very useful in evaluating the values of these polynomials at a point (x,t). By the formula

$$h_j(x,t) = (-t)^{j/2} \cdot H_j\left(\frac{x}{(-4t)^{1/2}}\right),$$

where $H_j(z)$ is the Hermite polynomial (see [1] or [3]), and by the recurrence formula for the Hermite polynomials

$$H_j(z) = 2zH_{j-1}(z) - 2(j-1)H_{j-2}(z),$$

we get

$$h_j(x,t) = (-t)^{j/2}\left[2\frac{x}{(-4t)^{1/2}} H_{j-1}\left(\frac{x}{(-4t)^{1/2}}\right) - 2(j-1)H_{j-2}\left(\frac{x}{(-4t)^{1/2}}\right)\right]$$

$$= (-t)^{j/2}\left[2\frac{x}{(-4t)^{1/2}}(-t)^{-(j-1)/2} h_{j-1}(x,t) - 2(j-1)(-t)^{-(j-2)/2} h_{j-2}(x,t)\right]$$

$$= x h_{j-1}(x,t) - 2(j-1) t h_{j-2}(x,t).$$

Thus

$$(13) \qquad \begin{cases} h_0(x,t) = 1, \ h_1(x,t) = x, \\ h_j(x,t) = xh_{j-1}(x,t) - 2(j-1)th_{j-2}(x,t), \quad j = 2,3,\dots \end{cases}$$

6. Algorithm for finding the approximation $h^N(x,t)$

The recurrence formula (13) yields a simple algorithm for determining values of the polynomials $h_j(x,t)$. Therefore it is convenient to seek the functions $g_j(x,t)$ as linear combinations of the functions $h_0,\dots,h_i$, i.e.,

$$(14) \qquad g_i(x,t) = h_i(x,t) + \sum_{j=0}^{i-1} \beta_{ij} h_j(x,t),$$

where β_{ij} are unknown coefficients to be determined. Setting

$$(15) \qquad \alpha_{ij} = \frac{(h_i, g_j)}{(g_j, g_j)},$$

we can write the formula (12) as

$$(16) \qquad g_i(x,t) = h_i(x,t) + \sum_{j=0}^{i-1} \alpha_{ij} g_j(x,t), \qquad i = 1,2,\dots,N.$$

If now in this formula we substitute (14) for $g_j(x,t)$, then changing the order of summation we get

$$g_i(x,t) = h_i(x,t) + \sum_{j=0}^{i-1} \alpha_{ij} \Big[h_j(x,t) + \sum_{k=0}^{j-1} \beta_{jk} h_k(x,t) \Big] =$$
$$= h_i(x,t) + \sum_{j=0}^{i-1} \Big[\alpha_{ij} + \sum_{k=j+1}^{i-1} \alpha_{ik}\beta_{kj} \Big] h_j(x,t).$$

Comparing that with (14), we get a recurrence formula for the coefficients β_{ij} as follows

$$(17) \qquad \beta_{ij} = \alpha_{ij} + \sum_{k=j+1}^{i-1} \alpha_{ik}\beta_{kj}, \quad i = 1,2,\dots,N; \ j = 0,1,\dots,i-1.$$

So the coefficients β_{ij} can be evaluated if α_{ij} are known. However, substituting the relations (14) into (15), we get

$$\alpha_{ij} = -(h_i, g_j)/(g_j, g_j) = -\frac{1}{e_j}\Big(h_i, h_j + \sum_{k=0}^{j-1} \beta_{jk} h_k \Big) =$$

$$= -\frac{1}{e_j}\Big[(h_i, h_j) + \sum_{k=0}^{j-1} \beta_{jk}(h_i, h_k)\Big].$$

Let us remark that $(h_i, h_j) = a_{ij}$, where a_{ij} are given by (7). Therefore we are entitled to write

$$\alpha_{ij} = -\frac{1}{e_j}\Big[a_{ij} + \sum_{k=0}^{j-1} \beta_{jk} a_{ik}\Big]. \tag{18}$$

The coefficients e_j can be computed in the following fashion

$$\begin{aligned} e_j = (g_j, g_j) &= \Big(h_j + \sum_{k=0}^{j-1} \beta_{jk} h_k, h_j + \sum_{k=0}^{j-1} \beta_{jk} h_k\Big) = \\ &= a_{jj} + 2\sum_{k=0}^{j-1} \beta_{jk} a_{jk} + \sum_{k=0}^{j-1}\sum_{l=0}^{j-1} \beta_{jk}\beta_{jl} a_{kl}. \end{aligned} \tag{19}$$

As seen, the problem of computing α_{ij} and β_{ij} is reduced to the evaluation of the elements a_{ij}. These can be in turn found from the following formula being a consequence of (7)

$$a_{ij} = \begin{cases} 0, & \text{when } i+j \text{ is odd,} \\ 2\Big[\dfrac{1}{i+j+1} + \displaystyle\sum_{l=0}^{[i/2]}\sum_{k=0}^{[j/2]} \dfrac{r_{lk}}{k+l+1}\Big], & \text{when } i+j \text{ is even,} \end{cases} \tag{20}$$

where

$$r_{lk} = \frac{t_0^{k+l+1}\, i!\, j!}{(j-2k)!(i-2l)!k!l!}.$$

It is convenient to complete the coefficients r_{lk} with the algorithm

$$\begin{aligned} r_{00} &= t_0, \\ r_{l+1,0} &= t_0 \frac{(i-2l)(i-2l-1)}{l+1} r_{l,0}, \quad l = 0, 1, \dots, [i/2], \\ r_{l,k+1} &= t_0 \frac{(j-2k)(j-2k-1)}{k+1} r_{l,k}, \quad l = 0, 1, \dots, [i/2], \\ & \qquad\qquad\qquad\qquad\qquad\qquad k = 0, 1, \dots, [j/2]-1. \end{aligned}$$

Taking into account (14), we will propose another form of the formula (10) yielding the approximation h^N. Substituting (14) into (10) and changing the order of summation, we get

$$h^N(x,t) = \sum_{j=0}^{N} d_j g_j(x,t) = \sum_{j=0}^{N} d_j \Big[h_j(x,t) + \sum_{k=0}^{j-1} \beta_{jk} h_k(x,t)\Big] =$$

$$= \sum_{j=0}^{N} d_j h_j(x,t) + \sum_{j=0}^{N-1} \Big(\sum_{k=j+1}^{N} d_k \beta_{kj} \Big) h_j(x,t) =$$

$$= d_N h_N(x,t) + \sum_{j=0}^{N-1} \Big[d_j + \sum_{k=j+1}^{N} d_k \beta_{kj} \Big] h_j(x,t).$$

Hence setting

$$\gamma_j = d_j + \sum_{k=j+1}^{N} d_k \cdot \beta_{kj}, \qquad j = 0, 1, \ldots, N-1, \tag{21}$$

we can rewrite the formula (10) in the following form

$$h^N(x,t) = d_N h_N(x,t) + \sum_{j=0}^{N-1} \gamma_j h_j(x,t). \tag{22}$$

We propose the following algorithm for finding the approximation $h^N(x,t)$:

1° For $i, j = 0, 1, \ldots, N$ compute a_{ij} according to the formula (20) (remark: $a_{ij} = a_{ji}$).

2° Set $e_0 = a_{00}$.

3° For $i = 1, 2, \ldots, N$
  (a) and for $j = 0, 1, \ldots, i-1$ compute α_{ij} from (18),
  (b) and for $j = 0, 1, \ldots, i-1$ compute β_{ij} from (17),
  (c) evaluate e_i from (19).

4° For $j = 0, 1, \ldots, N$ evaluate d_j, using the formula (11).

5° For $j = 0, 1, \ldots, N-1$ evaluate γ_j, using (21).

6° In order to evaluate the approximation $h^N(x,t)$ at a point (x,t) make use of the formula (22). Evaluate values of the functions $h_j(x,t)$ from the recurrence relations (13).

7. Final remarks

At the end we would like to emphasize the virtues of the above described method. The first advantage is that the received approximation is determined at each point of the domain in which we seek solutions, because it is given in an analytical form. The second very important property is that the accuracy improvement of the approximation is connected only with adding consecutive terms to $h^N(x,t)$ determined by the formula (10). The popular in applications finite-difference methods have no these advantages.

References

[1] D. Colton, *The approximation of solutions to initial boundary value problems for parabolic equations in one space variable*, Quart. Appl. Math., January (1976), 377–386.

[2] H. Grabarska, *The approximation of solution of the Fourier problem of the first kind for systems of parabolic equations*, Demonstratio Math. 23 (1990), 1033–1042.

[3] D.V. Widder, *Analytic solutions of the heat equation*, Duke Math. J. 29 (1962), 497–503.

[4] J. & M. Jankowscy, *Przegląd metod i algorytmów numerycznych.* cz. 1, WNT, Warszawa 1981.

[5] A. Kiełbasiński, H. Schwetlick, *Numeryczna algebra liniowa.* WNT, Warszawa 1992.

[6] J. H. Wilkinson, *Błędy zaokrągleń w procesach algebraicznych*, PWN, Warszawa 1967.

INSTITUTE OF MATHEMATICS
WARSAW UNIVERSITY OF TECHNOLOOGY
00-661 WARSAW, POLAND

Received April 29, 1996.