

COLLOCATION AND LEAST SQUARES METHOD FOR 2D HEAT CONDUCTION EQUATION*

L. G. SEMIN

Institute of Theoretical and Applied Mechanics SB RAS,

Novosibirsk, Russia

e-mail: l_semin@ngs.ru

Для уравнения теплопроводности построен метод, основанный на комбинации метода коллокации и метода наименьших квадратов. Он имеет ряд привлекательных свойств, таких как легкость задания граничных условий различных типов, легкость повышения порядка аппроксимации, и др. На тестах с известными точными решениями на последовательности сеток показана сходимость численного решения к точному и исследован порядок сходимости. Приведены формулы для ускорения сходимости итераций.

Introduction

In this study the method based on combined application of collocation method and least squares method for heat conduction equation is presented.

Application of the least squares method often improves the properties of a numerical method. For example, it is known that an increase in the number of interpolation nodes leads to higher numerical instability in the problem of finding an approximant in the form of a Lagrange polynomial for a discretely assigned function. The application of the least squares method makes solution of this problem more stable. The main difference between the least squares method and the Lagrange method is that the former does not require the equality of interpolant values and discrete values of interpolated function at the nodes. A minimum of a functional, which usually consists of a sum of weighted residual squares for all nodes, must be achieved instead.

In turn, the collocation method is simple in implementation and gives good results when solving boundary-value problems for ordinary differential equations, both linear and nonlinear [1, 2], for one-dimensional nonlinear parabolic equations [3] and for nonrigid elliptic problems on uniform grids [4]. However, ill-conditioned matrices are obtained when this method is applied on grids with local refinement. That is why this method in combination with adaptive irregular grids did not give more precise results in comparison with the solution of the problems on uniform grids.

The method called collocation and least squares (CLS) method has shown good computational properties when solving elliptic equations on triangular grids [5], stationary Stokes and Navier — Stokes equations [6, 7, 8]. It has several attractive features. One of them is that it is easy to

*This study was supported by INTAS YSF grant N 04-83-3860.

© Институт вычислительных технологий Сибирского отделения Российской академии наук, 2006.

realize boundary conditions of any type: Dirichlet, von Neumann, mixed conditions. Another one is that it is easy to increase the approximation order.

In the next section we describe the CLS method for stationary heat equation, then we generalize this approach for the case of time-depended equation, show the results of some numerical experiments, describe the method for accelerating the iterations and give short summary of the study.

1. CLS method for stationary equation

Let us consider boundary value problem for stationary two-dimensional heat conduction equation with convective transfer along x_1 axis:

$$C_d \left(\frac{\partial^2 \theta}{\partial x_1^2} + \frac{\partial^2 \theta}{\partial x_2^2} \right) + C_c u(x_1, x_2) \frac{\partial \theta}{\partial x_1} + f(x_1, x_2) = 0, \quad (x_1, x_2) \in \Omega, \quad (1)$$

$$a\theta + b \frac{\partial \theta}{\partial n} = g(x_1, x_2), \quad (x_1, x_2) \in \partial\Omega, \quad (2)$$

where C_d and C_c are diffusion and convective coefficients, $\Omega = (X_{11}, X_{12}) \times (X_{21}, X_{22})$, $\partial\Omega$ is the boundary of Ω , n is the normal vector to the $\partial\Omega$.

We shall find the numerical solution as piece-wise polinomial function. Let us introduce rectangular grid (x_1^i, x_2^j) and local coordinates in each cell: $y_1 = (x_1 - x_{1m})/h_1$, $y_2 = (x_2 - x_{2m})/h_2$, $\bar{\theta}(y_1, y_2) = \theta(x_1, x_2)$, where (x_{1m}, x_{2m}) are coordinates of m -th cell center, h_1, h_2 are half-widths of the m -th cell in x_1 and x_2 directions, correspondingly. Then equation (1) in local variables can be written as follows:

$$C_d \left(\frac{\partial^2 \bar{\theta}}{\partial y_1^2} + \frac{h_1^2}{h_2^2} \frac{\partial^2 \bar{\theta}}{\partial y_2^2} \right) + C_c u(y_1, y_2) \frac{h_1^2}{h_2} \frac{\partial \bar{\theta}}{\partial y_1} + h_1^2 \bar{f}(y_1, y_2) = 0. \quad (3)$$

We find approximate solution in each cell as linear combination of basic functions:

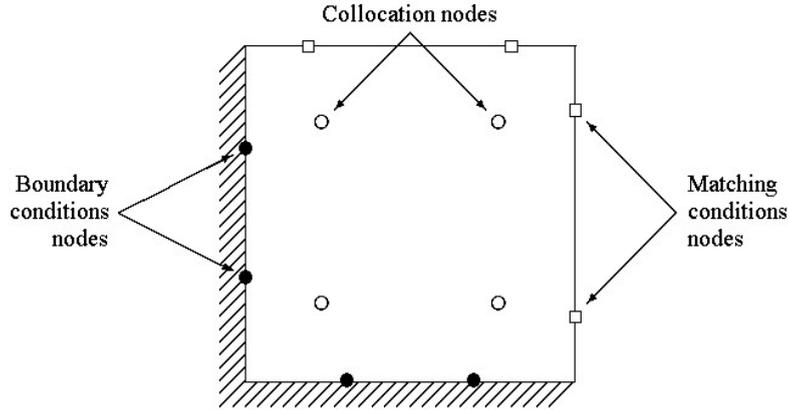
$$\bar{\theta} = \sum_j c_j \varphi_j, \quad (4)$$

where φ_j are basic functions, c_j are unknown coefficients. We take basic functions belonging to the space of second order polynomials and write them down as:

j	1	2	3	4	5	6
φ_j	1	y_1	y_2	$y_1 \cdot y_2$	$y_1^2 - y_2^2$	$y_1^2 + y_2^2$

Thus, the first five terms in (4) compose harmonic part of the approximate solution, and the sixth one is non-harmonic part.

We shall find coefficients c_j from the following conditions. First, we require the equation (3) to be satisfied in four points inside each cell (these are collocation equations). Second, we specify matching conditions on the intercell boundaries at two points on the each cell side. If the cell side is on the domain boundary, then we specify boundary conditions at two points on this side. And because the function θ is explicitly differentiable, there are no troubles in specifying boundary conditions of different types in each cell; for example, we do not need additional points to approximate the derivatives, in contrast to finite-difference methods. Figure shows a schematic location of the matching, boundary and collocation points in a cell. The continuity



Location of matching, boundary and collocation nodes in a cell.

of the following expressions is considered as matching conditions:

$$\frac{\partial \theta}{\partial n} + \eta \theta, \quad (5)$$

where n is the outside unit vector normal to the cell, η is a positive parameter. The latter can affect the conditionality of the obtained system of linear algebraic equations and the convergence rate. As a result, the following system of linear algebraic equations is obtained:

$$\sum_{k=1}^6 B_{lk} c_{km} = \bar{F}_l, \quad l = 1, \dots, 12. \quad (6)$$

This system is overdetermined: it involves 12 equations and 6 unknowns. We shall find its solution by least-squares method in following way. To define what is considered to be a solution of this system, let us compose two functionals:

$$\Phi_1 = \sum_{l=1}^8 \left(\sum_{j=1}^6 B_{lj} c_{jm} - \bar{F}_l \right)^2, \quad \Phi_2 = \sum_{l=9}^{12} \left(\sum_{j=1}^6 B_{lj} c_{jm} - \bar{F}_l \right)^2.$$

The first functional corresponds to the sum of residual squares of equations obtained from the matching or boundary conditions, the second one corresponds to the sum of the residual squares of the collocation equations. Solution of (6) is found from minimizing these functionals, with Φ_1 being minimized with respect to c_{jm} , $j = 1, \dots, 5$, at fixed c_{6m} , and Φ_2 being minimized with respect to c_{6m} at fixed others c_{jm} .

Thus, we have the following system of linear algebraic equations for determining the coefficients c_{jm} , $j = 1, \dots, 6$ in each cell:

$$\sum_{j=1}^6 D_{lj} c_{jm} = F_l, \quad l = 1, \dots, 6. \quad (7)$$

2. CLS method for time-dependent equation

Now let us consider initial problem for time-dependent equation:

$$\frac{\partial \theta}{\partial t} = C_d \left(\frac{\partial^2 \theta}{\partial x_1^2} + \frac{\partial^2 \theta}{\partial x_2^2} \right) + C_c u(x_1, x_2) \frac{\partial \theta}{\partial x_1} + f(x_1, x_2); \quad (x_1, x_2) \in \Omega, \quad (8)$$

$$a\theta + b\frac{\partial\theta}{\partial n} = g(x_1, x_2, t), \quad (x_1, x_2) \in \partial\Omega; \quad (9)$$

$$\theta(x_1, x_2, 0) = \psi(x_1, x_2). \quad (10)$$

Again, let us introduce rectangular grid (x_1^i, x_2^j, t^k) . We shall find numerical solution on each time level k as piece-wise polynomial function which depends only on spatial coordinates, as described in the previous section. That is, we find approximate solution in each cell as linear combination of basic functions:

$$\bar{\theta}^k = \sum_j c_j^k \varphi_j, \quad (11)$$

where k stands for the index of time level, and basis functions φ_j do not depend on time. To approximate time derivative we use finite difference. The simplest approximation is the following:

$$\frac{\partial\bar{\theta}}{\partial t} \approx \frac{\bar{\theta}^{k+1} - \bar{\theta}^k}{\Delta t}, \quad \Delta t = t^{k+1} - t^k, \quad (12)$$

$$\bar{\theta}^k = \bar{\theta}(y_1, y_2, t^k) = \theta(x_1, x_2, t^k).$$

Substituting this approximation into the equation (8) we can write down the implicit scheme:

$$\frac{\theta^{k+1} - \theta^k}{\Delta t} = C_d \left(\frac{\partial^2 \theta^{k+1}}{\partial x_1^2} + \frac{\partial^2 \theta^{k+1}}{\partial x_2^2} \right) + C_c u(x_1, x_2) \frac{\partial \theta^{k+1}}{\partial x_1} + f(x_1, x_2). \quad (13)$$

Now to find the approximate solution we need to find coefficients c_j^k in (11) on each time level. We again introduce local coordinates in each spatial cell. Substitution of the representation (11) into (13) gives us the approximate equation in local coordinates. Then we apply the CLS method described in the previous section taking this equation as collocation one and derive the system of linear algebraic equations of the form (7) to find coefficients c_j^k in each cell on each time level.

3. Finding solution in the whole domain

The CLS approach leads to the system of equations of the form (7) both for stationary and non-stationary equations. This is the system for determining unknown coefficient in a single cell. The assemblage of these systems over all cells of the domain gives a global system of linear algebraic equations for finding the solution in the whole domain. An iterative method, in which the solution is improved for each cell individually, is applied to solve this system. At each iteration when improving the solution in m -th cell, the solution in adjacent cells either has already been improved or is taken from previous iteration step. This method can be considered as optimized Schwartz method without overlapping with Robin transmission conditions (5) where each cell is meant to be a subdomain.

4. Numerical experiments

The formulae of the method were tested numerically on problems with exact solution being known. We used these tests also to investigate numerically the convergence order of the method.

Firstly the simplest tests were performed to be ensured in the correctness of the formulae derived.

The polynomial belonging to the basis was taken as exact solution, and in each cell this solution was taken as the initial approximation. For time-dependent equation, linear function in time was taken. After the first iteration step all coefficients c_j of the numerical solution improvement were equal to zero. This fact means that the method does not damage the exact solution obtained.

Another test was made for every particular cell type. In the cell of interest, either the perturbed exact solution or zero value was taken as initial approximation. Boundary and matching conditions were taken from exact solution. Iterations were performed over the cell of interest only. The iterative process gave exact solution in this cell in a few iteration steps.

Third of the tests with exact solution belonging to the basis was like the second one but iterations were performed over the whole domain. In all such tests numerical solution converged to the exact one.

These tests assure that the method will give exact solution for the exact solution belonging to the basis and for solutions depending on time linearly.

Other tests were performed in order to establish the convergence order of the method. In these tests the exact solutions were beyond the basis.

4.1. Tests for the stationary equation

Among the tests for the stationary equation there were the next ones:

the exact solution is $\theta = x_1^3 + x_2^3$, $f(x_1, x_2) = -C_d(6x_1 + 6x_2) - 3C_c x_1^2$, $C_d = 1$, $C_c = 1$, $u(x_1, x_2) = 1$;

the exact solution is $\theta = \exp(-x_1^2)$, $f(x_1, x_2) = (-C_d(-2 + 4x_1^2) + 2C_c x_1)\exp(-x_1^2)$, $C_d = 1$, $C_c = 1$, $u(x_1, x_2) = 1$;

the exact solution is $\theta = \sin(x_1)\exp(-x_2)$, $f(x_1, x_2) = -C_c \cos(x_1)\exp(-x_2)$, $C_d = 1$, $C_c = 1$, $u(x_1, x_2) = 1$.

The numerical solution converged to the exact one. All these tests show second order of convergence. Table 1 shows the results for the third of the tests mentioned above. One can see that the difference between exact and calculated solutions decreased greater than four times while the grid step decreased by factor of 2. This fact means that the numerical solution converges to the exact one with the second order.

The convergence order can be increased by adding polynomials of higher order to the basis.

4.2. Tests for the time-dependent equation

Among the tests for the time-dependent equation there were the tests with exact solutions containing polynomials of different orders in time and space, trigonometric and exponential

Таблица 1. Establishing the convergence order. Exact solution is $\theta = \sin(x_1)\exp(-x_2)$, domain is $0 \leq x_1 \leq 1, -1 \leq x_2 \leq 1$

N	Grid	Grid step	Error	Error decrease factor
1	10×20	0.1	$3.213 \cdot 10^{-4}$	
2	20×40	0.05	$4.328 \cdot 10^{-5}$	7.4
3	40×80	0.025	$7.094 \cdot 10^{-6}$	6.1
4	80×160	0.0125	$1.454 \cdot 10^{-6}$	4.9
5	160×320	0.00625	$3.294 \cdot 10^{-7}$	4.4

functions. Table 2 demonstrates the results on establishing the convergence order for the test where exact solution is $\theta = \exp(t) + \exp(-x_2)\sin(x_1)$. One can see that the difference between exact and calculated solutions decreased by factor of 2 while the grid step decreased by factor of 2. This fact means that the numerical solution converges to the exact one with the first order.

To increase convergence order, one should use finite difference approximation of time derivative of higher order. The method allows one to write down such difference scheme easily.

Таблица 2. Establishing the convergence order. Exact solution is $\theta = \exp(t) + \exp(-x_2)\sin(x_1)$, domain is $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $0 \leq t \leq 1$

N	Number of time steps	Time step	Error	Error decrease factor
1	4	0.25	$2.195 \cdot 10^{-2}$	
2	8	0.125	$1.137 \cdot 10^{-2}$	1.9
3	16	0.0625	$5.794 \cdot 10^{-3}$	2.0
4	32	0.03125	$2.924 \cdot 10^{-3}$	2.0
5	64	0.015625	$1.468 \cdot 10^{-3}$	2.0
6	128	0.0078125	$7.343 \cdot 10^{-4}$	2.0

5. Acceleration of convergence

The convergence of the iterative method mentioned above is slow if the grid is fine. To accelerate convergence we have used approach proposed by A.G. Sleptsov [9]. This approach consists of projecting the error to subspace of residuals of previous iterations.

Let we need to solve a system of linear algebraic equations:

$$Ax = b. \quad (14)$$

Let us transform it to the equivalent form:

$$x = Tx + f \quad (15)$$

and write down the iterative method

$$x_{n+1} = Tx_n + f. \quad (16)$$

We introduce the following designations: $z_n = Tx_n + f - x_n = x_{n+1} - x_n$ is the residual of the iteration, $y_n = x - x_n = y_{n+1} + z_n$ is the error, where x is the exact solution of (15). From the last equality it is easy to obtain the equation:

$$(I - T)y_n = z_n. \quad (17)$$

If one solves it exactly then the exact solution is given by the equality $x = x_n + y_n$. But this is usually as hard to do as to solve the equation (14). So, one solves the equation (17) approximately and writes down the correction $x = x_n + \tilde{y}_n$ instead of exact expression. Many of acceleration methods (such as Bubnov-Galerkin and steepest descent) are equivalent to application of a projection method to find \tilde{y}_n in the form

$$\tilde{y}_n = \sum_{i=1}^m a_i \varphi_i$$

taking as φ_i the residuals z_i . Disadvantage of this approach is that to compute \tilde{y}_n one needs to compute z_{n+1} , that is to correct x_{n+1} one needs to calculate x_{n+2} . Hence, one has to perform extra computations.

To overcome this, let us replace equation (17) by

$$(T^{-1} - I)y_{n+1} = z_n. \quad (18)$$

Again, the approximate solution of (18) is searched for by some projection method in the form

$$\tilde{y}_{n+m+1} = \sum_{i=1}^m a_i z_{n+i}. \quad (19)$$

That is, we want to use last m residuals to find approximation of the error. So, to correct x_{n+m+1} we need only to calculate x_{n+m+1} and do not need to calculate x_{n+m+2} . As numerical experiments have shown, this method not only accelerates a convergent iteration process, but also can correct divergent process.

In the paper [9] the formulae for $m = 2$ are given. We have found that the best number m of residuals taken for the acceleration is 3 or 4. At bigger m the iterations accelerate not essentially, but significantly more memory is required to store the data. At $m = 3$ this method gives 10-fold acceleration as compared to iterations without the method.

We found coefficients a_i in (19) by least-squares method. For the case of $m = 3$ these coefficients can be found by the following system of linear algebraic equations:

$$Qa = p, \quad (20)$$

where

$$Q = \begin{pmatrix} (z_n - z_{n+1}, z_n - z_{n+1}) & (z_{n+1} - z_{n+2}, z_n - z_{n+1}) & (z_{n+2} - z_{n+3}, z_n - z_{n+1}) \\ (z_n - z_{n+1}, z_{n+1} - z_{n+2}) & (z_{n+1} - z_{n+2}, z_{n+1} - z_{n+2}) & (z_{n+2} - z_{n+3}, z_{n+1} - z_{n+2}) \\ (z_n - z_{n+1}, z_{n+2} - z_{n+3}) & (z_{n+1} - z_{n+2}, z_{n+2} - z_{n+3}) & (z_{n+2} - z_{n+3}, z_{n+2} - z_{n+3}) \end{pmatrix},$$

$$p = \begin{pmatrix} (z_{n+3}, z_n - z_{n+1}) \\ (z_{n+3}, z_{n+1} - z_{n+2}) \\ (z_{n+3}, z_{n+2} - z_{n+3}) \end{pmatrix}, \quad a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix},$$

(\cdot, \cdot) denotes a scalar product of two vectors.

6. Summary

In this paper a collocation and least-squares method for two-dimensional heat conducting equation is presented, where a possibility of variable heat transfer along x axe is taken into account. Simultaneous usage of the collocation method and the least-squares technique allows one to blend together the advantages of both methods. One of the advantages of the CLS method is the easiness of specifying boundary conditions of different types: Dirichlet, von Neumann or mixed, without introducing additional points to approximate derivatives, in contrast to many other numerical methods.

One more attractive feature of this method is that the solution computed on coarse grid can be used on fine grid without any interpolation and hence without loss of accuracy because when one introduces fine grid then the only thing he needs to use the solution computed previously is to recalculate coefficients c_j in new local coordinate systems of new cells.

In addition one can increase the approximation order in the space simply by adding polynomials of higher order to basic functions.

It is also easy to write down an implicit scheme. It is well known that implicit schemes usually are more stable than explicit ones and do not put rigid restrictions on the time step value. In the present study the first order finite difference approximation of time derivative is considered, but the method is not limited by this particular finite difference, and higher order approximations can be used.

This method can also be generalized to problems with curvilinear boundary because in general it does not matter where one has to specify boundary conditions: on a straight line or on a curve.

References

- [1] ASCHER U., CHRISTIANSEN J., RUSSELL R.D. A collocation solver for mixed order systems of boundary value problems // *Mathematics of Computation*. 1979. Vol. 33. P. 659–679.
- [2] DE BOOR C., SWARTZ B. Collocation at Gaussian points // *SIAM J. of Numer. Anal.* 1973. Vol. 10, N 4. P. 582–606.
- [3] PLYASUNOVA A.V., SLEPTSOV A.G. Collocation-grid method for solving nonlinear parabolic equations // *Rus. J. Theor. & Appl. Mech.* 1991. Vol. 1, N 1. P. 15–26.
- [4] SLEPTSOV A.G. Grid-projection solution of elliptic problem for a irregular grid // *Rus. J. Numer. Anal. & Math. Modelling*. 1993. Vol. 8, N 6. P. 501–525.
- [5] SLEPTSOV A.G., SHOKIN YU.I. Adaptive projection-grid method for elliptic problems // *Zhurnal vychislitelnoi matematiki i matematicheskoi fiziki*. 1997. Vol. 31, N 5. P. 572–586.
- [6] SEMIN L.G., SLEPTSOV A.G., SHAPEEV V.P. Collocation and least-squares method for Stokes equations // *Computational Technologies*. 1996. Vol. 1, N 2. P. 90–98.
- [7] SEMIN L.G., SHAPEEV V.P. Collocation-grid method for solving boundary problems for Navier — Stokes equations // *Proc. Intern. Conf. on the Methods of Aerophys. Research. Pt II. Novosibirsk*, 1998. P. 186–191.
- [8] SHAPEEV V.P., SEMIN L.G., BELYAEV V.V. The collocation and least squares method for numerical solution of Navier — Stokes equations // *Proc. The Steklov Institute of Mathematics*. 2003. Suppl. 2. P. S115–S137.
- [9] SLEPTSOV A.G. On the convergence acceleration of the linear iterations. II // *Russ. J. of Theor. and Appl. Mech.* 1991. Vol. 1, N 3. P. 213–219.