BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Volumul 69 (73), Numărul 3, 2023
Secţia
CONSTRUCŢII DE MAŞINI
DOI:10.2478/bipcm-2023-0028

FINITE DIFFERENCE METHOD FOR SOLVING THE TWO-DIMENSIONAL LAPLACE EQUATION IN CURVILINEAR COORDINATES

BY

GABRIEL-DUMITRU TCACIUC* and ARISTOTEL POPESCU

“Gheorghe Asachi” Technical University of Iaşi,
Faculty of Mechanical Engineering, Iaşi, Romania

Received: December 15, 2023
Accepted for publication: December 30, 2023

Abstract. This paper presents a numerical method for solving the two-dimensional Laplace equation in curvilinear coordinates using a finite difference scheme. Laplace equation is a differential equation used to describe various physical phenomena studied within different branches of engineering. The two-dimensional Laplace equation has analytic solutions only for simple shaped domains and simple boundary conditions (first and second kind). Even so, the results are usually complicated to interpret and use. The present paper proposes a method to solve the equation for domains bounded by complex inner and outer curves. The boundary curves are defined by a set of points from which the equations of curves are approximated by using Fourier series. The domain is generated by blending the two curves using a blending function. The results are a set of continuous parametric equations which are defining the domain in angle and radius variables. Following a method presented in the literature, the Laplace equation is transformed from Cartesian coordinates to curvilinear coordinates, as the Fourier series is easily differentiable. Next, the new Laplace equation is

*Corresponding author; *e-mail*: gabriel-dumitru.tcaciuc@academic.tuiasi.ro
© 2023 Gabriel-Dumitru Tcaciuc and Aristotel Popescu
This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

approximated using the finite difference method and a MATLAB code is developed to solve the system of linear equations using successive over-relaxation method (SOR). The presented method is used to simulate steady two-dimensional heat conduction in a polygonal shape domain, with uniform and non-uniform Dirichlet boundary conditions. Results are analyzed in order to determine the model's precision, speed and convergence criteria.

Keywords: finite difference method, curvilinear coordinates, heat conduction, Fourier series, Laplace equation.

1. Introduction

Heat and mass transfer are permanent phenomena in every physical system which exchanges thermal energy with its surroundings or with other systems. Understanding and modeling the phenomena are necessary when designing any kind of mechanical system, especially those designed to move heat or fluids (heat exchangers, pumps, fans, compressors etc.).

Modeling these phenomena starts from governing differential equations, that express conservation of mass, momentum and energy. Because these equations are linear and non-linear PDE's, direct analytic solutions can be obtained only in very few cases, often for simple geometries and after applying enough simplifying assumptions. For the large majority of cases, solutions must be approximated using numerical methods. One of the widely used methods is the finite difference method, which consists of approximating the derivatives in discrete points, named nodes. This means that the original continuous domain must be discretized in a grid of nodes, the process being relatively simple when the geometries are rectangular, cylindrical or spherical. In practice, the geometries are very often more complex and are treated in general by other methods, such as the finite element method. However, the finite difference scheme can be applied on complex geometries by writing the governing equations in a new coordinate system more suited for the geometry. The new coordinates are called generalized coordinates or curvilinear coordinates.

The method has been applied extensively for heat transfer and fluid flows. Schneider et al. derived a finite difference scheme in general orthogonal coordinates for heat conduction by approximating the derivatives using Taylor expansion (Schneider *et al*., 1975). Robertson formulated the equation of heat conduction in generalized coordinates and showed its applications on different coordinate system like skewed, parabolic-rectilinear and conical (Robertson, 1979). More generally, Lau derived a 3D finite-difference scheme in generalized coordinates for potential problems (Lau, 1978). On more specific problems, the method was applied by Fan to compare simulated results for a fin-and-tube surface (Fan *et al*., 2012). Bigham et al. used curvilinear coordinates with a finite element scheme for curvilinear microchannels (Bigham *et al*., 2011). Flow and

heat transfer in curvilinear ducts with elliptic grid generation was studied by Öztop (Öztop, 2005).

In this paper, a curvilinear finite difference method for heat conduction with Dirichlet boundary condition is developed. The method is applied on domains in which inner and outer boundaries are different regular polygon shapes. Polygon shapes are approximated using Fourier series.

2. Mathematical model

Consider a regular polygon with N_p sides, inscribed in a circle with radius R_p which has its vertices defined by the coordinates x_k and y_k:

$$\begin{cases} x_k = R_p \cos\left(\dfrac{2\pi}{N_p}k\right) \\ y_k = R_p \sin\left(\dfrac{2\pi}{N_p}k\right) \end{cases}, k = 0,1,2,...,N_p \tag{1}$$

Sides are defined as the set of points which form a straight line between vertices:

$$\begin{aligned} S_k = \{&(x,y) \in \mathrm{R}, x = x_k + \delta(x_{k+1} - x_k), \\ &y = y_k + \delta(y_{k+1} - y_k), \delta \in [0,1)\} \end{aligned} \tag{2}$$

Now the polygon can be defined as the reunion between every side set:

$$P = \bigcup_{k=0}^{N_p - 1} S_k \tag{3}$$

If P is a finite set, Eq. (2) takes the following form:

$$\begin{aligned} S_k = \bigcup_{m=0}^{M-1} \Bigg\{ &(x,y) \in \mathrm{R}, x = x_k + \frac{m}{M+1}(x_{k+1} - x_k), \\ &y = y_k + \frac{m}{M+1}(y_{k+1} - y_k) \Bigg\} \end{aligned}, \tag{4}$$

where M is the number of points between two adjacent vertices.

Defining the polygon in this way is satisfactory from a mathematical point of view, but is difficult to apply curvilinear coordinates transformation, because there is no single continuous function which defines the entire shape.

Therefore, an approximated continuous parametric function is proposed as follows:

$$P:\begin{cases} x = R_p \phi_x(\theta) \\ y = R_p \phi_y(\theta) \end{cases} \tag{5}$$

where $\phi_x(\theta)$, $\phi_y(\theta)$ are functions approximated using finite Fourier series and θ is the angle coordinate:

$$\begin{cases} \phi_x(\theta) = \sum_{n=0}^{N} b_n \cos(n\theta) \\ \phi_y(\theta) = \sum_{n=0}^{N} a_n \sin(n\theta) \end{cases} \tag{6}$$

The Fourier series coefficients, a_n and b_n, have the following expressions:

$$\begin{cases} a_n = \dfrac{1}{\pi}\int_0^{2\pi} y \sin(n\theta)\, d\theta \\ b_n = \dfrac{1}{\pi}\int_0^{2\pi} y \cos(n\theta)\, d\theta \end{cases}, \tag{7}$$

where x and y are computed using the finite set definition at R_p = 1. The computation is performed by defining the set (4) as two column vectors:

$$\begin{cases} X_{k(M+1)+m+1} = x_k + \dfrac{m}{M+1}(x_{k+1} - x_k) \\ Y_{k(M+1)+m+1} = y_k + \dfrac{m}{M+1}(y_{k+1} - y_k) \end{cases} \tag{8}$$

The angles between vertices and the horizontal axis, measured in trigonometric direction, are given by the following column vector:

$$\Theta_{k(M+1)+m+1} = \frac{2\pi}{N_p} k + \cos^{-1}\left(\frac{x_k X_{k(M+1)+m+1} + y_k Y_{k(M+1)+m+1}}{\sqrt{X^2_{k(M+1)+m+1} + Y^2_{k(M+1)+m+1}}} \right), \tag{9}$$

where k = 0, 1, 2, …, N_p -1 and m = 0, 1, 2, …, M.

This manipulation orders the elements in trigonometric direction and allows an index change from $k(M+1)+m+1$ to $u = 1, 2, \ldots, N_pM$.

Fourier coefficients are computed by applying the trapezoidal rule of integration to Eqs. (7):

$$\begin{cases} a_n = \dfrac{1}{2\pi}\displaystyle\sum_{u=1}^{N_p(M+1)-1}\left[Y_u \sin\left(n\Theta_u\right)+Y_{u+1}\sin\left(n\Theta_{u+1}\right)\right]\left(\Theta_{u+1}-\Theta_u\right) \\ b_n = \dfrac{1}{2\pi}\displaystyle\sum_{u=1}^{N_p(M+1)-1}\left[X_u \cos\left(n\Theta_u\right)+X_{u+1}\cos\left(n\Theta_{u+1}\right)\right]\left(\Theta_{u+1}-\Theta_u\right) \end{cases} \tag{10}$$

By allowing the radius R_p from Eqs. (5) to become variable, the function will produce a two-dimensional domain. Let's assume that R_p become the variable r and takes values from an inner radius R_c to an outer radius R_s. Eqs. (5) become:

$$\begin{cases} x = r\displaystyle\sum_{n=0}^{N} b_n \cos\left(n\theta\right) \\ y = r\displaystyle\sum_{n=0}^{N} a_n \sin\left(n\theta\right) \end{cases}, r\in\left[R_c, R_s\right], \theta\in\left(0, 2\pi\right] \tag{11}$$

If different curves are desired for inner and outer boundaries, then one of the curves must be blended into the other curve progressively, as the radius varies. This action is performed by varying the series coefficients as a function of radius. In the present paper, the analyzed domain is bounded by an inner circle and an outer square and it is defined by the following relations:

$$\begin{aligned} \Omega = \Big\{&\left(x, y\right)\in \mathrm{R}^2 : x = r\psi_x\left(r,\theta\right), y = r\psi_y\left(r,\theta\right), \\ &r\in\left[R_c, R_s\right], \theta\in\left(0, 2\pi\right]\Big\} \end{aligned}, \tag{12}$$

where ψ_x and ψ_y functions are:

$$\begin{cases} \psi_x = b_1\cos\left(\theta\right)+\dfrac{r-R_c}{R_s-R_c}\displaystyle\sum_{n=2}^{N} b_n\cos\left(n\theta\right) \\ \psi_y = a_1\sin\left(\theta\right)+\dfrac{r-R_c}{R_s-R_c}\displaystyle\sum_{n=2}^{N} a_n\sin\left(n\theta\right) \end{cases} \tag{13}$$

Figure 1 shows the discrete domain generated with Eq. (12), using 15 non-zero Fourier coefficients. A low number of steps in radial and angular directions were chosen for display purposes (un-cluttered distribution).

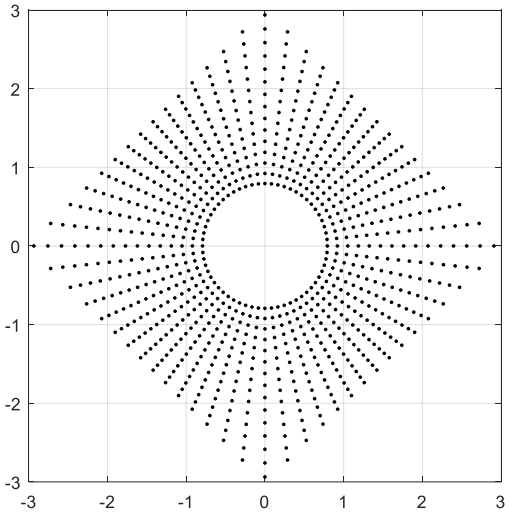


Fig. 1 – Discrete domain.

3. Laplace equation in curvilinear coordinates

The Laplace equation is transformed from Cartesian coordinates (x, y) to curvilinear coordinates (r,θ), using the Mazumder methodology (Mazumder, 2015). In Cartesian coordinates, the equation is:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}=0\,, \tag{14}$$

where T represents the nondimensional temperature, but it can be any other scalar.

The general expression of Eq. (14) in curvilinear coordinates is:

$$\frac{\partial}{\partial r}\left(\frac{B_{11}}{J}\frac{\partial T}{\partial r}+\frac{B_{21}}{J}\frac{\partial T}{\partial \theta}\right)+\frac{\partial}{\partial \theta}\left(\frac{B_{21}}{J}\frac{\partial T}{\partial r}+\frac{B_{22}}{J}\frac{\partial T}{\partial \theta}\right)=0 \tag{15}$$

After algebraic manipulations, Eq. (15) is rewritten in a more compact form:

$$C_1\frac{\partial^2 T}{\partial r^2}+C_2\frac{\partial T}{\partial r}+C_3\frac{\partial^2 T}{\partial \theta}+C_4\frac{\partial^2 T}{\partial r\partial \theta}+C_5\frac{\partial^2 T}{\partial \theta^2}=0 \tag{16}$$

The general Dirichlet boundary conditions are:

$$\begin{cases} T\left(R_c,\theta\right)=f_c\left(\theta\right) \\ T\left(R_s,\theta\right)=f_s\left(\theta\right) \end{cases} \tag{17}$$

The coefficients from Eq. (16) are computed using the expressions from Table 1.

Table 1

Coefficients definition

Coefficient symbol	Formula
C_1	$\frac{B_{11}}{J}$
C_2	$\frac{1}{J}\left(B_{11r}+B_{12\theta}\right)-\frac{1}{J^2}\left(B_{11}J_r+B_{12}J_\theta\right)$
C_3	$\frac{1}{J}\left(B_{21r}+B_{22\theta}\right)-\frac{1}{J^2}\left(B_{21}J_r+B_{22}J_\theta\right)$
C_4	$\frac{2B_{21}}{J}$
C_5	$\frac{B_{22}}{J}$
J_r	$x_{rr}y_\theta+x_ry_{r\theta}-x_{r\theta}y_r-x_\theta y_{rr}$
J_θ	$x_{r\theta}y_\theta+x_ry_{\theta\theta}-x_{\theta\theta}y_r-x_\theta y_{r\theta}$
B_{11}	$y_\theta^2+x_\theta^2$
B_{21}	$-y_\theta y_r-x_\theta x_r$
B_{12}	$-y_\theta y_r-x_\theta x_r$
B_{22}	$y_r^2+x_r^2$
B_{11r}	$2y_\theta y_{r\theta}+2x_\theta x_{r\theta}$
B_{21r}	$-y_{r\theta}y_r-y_\theta y_{rr}-x_{r\theta}x_r-x_\theta x_{rr}$
$B_{12\theta}$	$-y_{\theta\theta}y_r-y_\theta y_{r\theta}-x_{\theta\theta}x_r-x_\theta x_{r\theta}$
$B_{22\theta}$	$2y_ry_{r\theta}+2x_rx_{r\theta}$
J	$x_ry_\theta-x_\theta y_r$
x_r	$b_1\cos\theta+\frac{2r-R_c}{R_s-R_c}\sum_{n=2}^{N}b_n\cos\left(n\theta\right)$

y_r	$a_1 \sin\theta + \dfrac{2r - R_c}{R_s - R_c} \sum_{n=2}^{N} a_n \sin(n\theta)$
x_θ	$-rb_1 \sin\theta - \dfrac{r^2 - rR_c}{R_s - R_c} \sum_{n=2}^{N} nb_n \sin(n\theta)$
y_θ	$ra_1 \cos\theta + \dfrac{r^2 - rR_c}{R_s - R_c} \sum_{n=2}^{N} na_n \cos(n\theta)$
x_{rr}	$\dfrac{2}{R_s - R_c} \sum_{n=2}^{N} b_n \cos(n\theta)$
y_{rr}	$\dfrac{2}{R_s - R_c} \sum_{n=2}^{N} a_n \sin(n\theta)$
$x_{\theta\theta}$	$-rb_1 \cos\theta - \dfrac{r^2 - rR_c}{R_s - R_c} \sum_{n=2}^{N} n^2 b_n \cos(n\theta)$
$y_{\theta\theta}$	$-ra_1 \sin\theta - \dfrac{r^2 - rR_c}{R_s - R_c} \sum_{n=2}^{N} n^2 a_n \sin(n\theta)$
$x_{r\theta}$	$-b_1 \sin\theta - \dfrac{2r - R_c}{R_s - R_c} \sum_{n=2}^{N} nb_n \sin(n\theta)$
$y_{r\theta}$	$a_1 \cos\theta + \dfrac{2r - R_c}{R_s - R_c} \sum_{n=2}^{N} na_n \cos(n\theta)$

4. Finite difference method

The differential equation (16) is transformed into a system of algebraic equations by approximating the derivatives using the finite difference method. First order partial derivatives are approximated with first order forward difference, second order and cross partial derivatives with second order central difference. After some algebraic manipulations, an algebraic equation for the (i, j) node is obtained:

$$\begin{aligned} &G^1_{i,j}T_{i,j} + G^2_{i,j}T_{i+1,j} + G^3_{i,j}T_{i-1,j} + G^4_{i,j}T_{i,j+1} + G^5_{i,j}T_{i,j-1} \\ &+G^6_{i,j}T_{i+1,j+1} - G^6_{i,j}T_{i-1,j+1} - G^6_{i,j}T_{i+1,j-1} + G^6_{i,j}T_{i-1,j-1} = 0 \end{aligned}, \tag{18}$$

where $G^1_{i,j},\ldots,G^6_{i,j}$ are:

$$\begin{cases} G_{i,j}^1 = -2\left(C_1\right)_{i,j} - \left(C_2\right)_{i,j}\Delta r - \dfrac{\left(\Delta r\right)^2}{\Delta\theta}\left(C_3\right)_{i,j} - 2\left(\dfrac{\Delta r}{\Delta\theta}\right)^2\left(C_5\right)_{i,j} \\ G_{i,j}^2 = \left(C_1\right)_{i,j} + \left(C_2\right)_{i,j}\Delta r \\ G_{i,j}^3 = \left(C_1\right)_{i,j} \\ G_{i,j}^4 = \left(\dfrac{\Delta r}{\Delta\theta}\right)^2\left(C_5\right)_{i,j} + \dfrac{\left(\Delta r\right)^2}{\Delta\theta}\left(C_3\right)_{i,j} \\ G_{i,j}^5 = \left(\dfrac{\Delta r}{\Delta\theta}\right)^2\left(C_5\right)_{i,j} \\ G_{i,j}^6 = \dfrac{\Delta r}{4\Delta\theta}\left(C_4\right)_{i,j} \end{cases} \tag{19}$$

where subscripts *i, j* are the position in radial and angular direction, respectively.

Eq. (18) is solved using the successive over-relaxation method (SOR):

$$\begin{aligned} T_{i,j}^{k+1} = \frac{\beta}{G_{i,j}^1}\Big[& G_{i,j}^6\left(T_{i-1,j+1}^k + T_{i+1,j-1}^k - T_{i+1,j+1}^k - T_{i-1,j-1}^{k+1}\right) \\ & - G_{i,j}^2 T_{i+1,j}^k - G_{i,j}^3 T_{i-1,j}^{k+1} - G_{i,j}^4 T_{i,j+1}^k - G_{i,j}^5 T_{i,j-1}^{k+1}\Big] + \left(1-\beta\right)T_{i,j}^k \end{aligned}, \tag{20}$$

where β is the over-relaxation parameter.

5. Results

Consider a domain bounded by two closed curves, square for the outer curve and circular for the inner one. The corresponding radii are R_s = 3 and R_c = 1 respectively. The numbers of Fourier coefficients and points between adjacent vertices are N = 60 and M = 5000 respectively. The boundary conditions are:

$$\begin{aligned} & T\left(R_c,\theta\right) = 1 \\ & T\left(R_s,\theta\right) = 0 \end{aligned}, \quad \theta \in \left(0, 2\pi\right] \tag{21}$$

A Matlab code has been written to compute all the coefficients used in the present model. The successive over-relaxation method (SOR) is implemented into a loop which terminates when the following condition is satisfied:

$$\max\left(\left|T^{k+1}-T^{k}\right|\right)<\varepsilon \tag{22}$$

where $\varepsilon = 10^{-6}$.

In order to test the model's precision, speed and convergence, simulations were carried out for different input parameters. The results are presented in Table 2.

Table 2
Simulation results

Z	K	τ (s)	T_m	β_{opt}
20	90	0.0364	0.4065	1.72
	180	0.1029	0.4065	1.77
	360	0.3355	0.4065	1.87
	720	1.3089	0.4065	1.92
40	90	0.1276	0.4018	1.78
	180	0.3378	0.4018	1.78
	360	1.2372	0.4018	1.8
	720	5.4560	0.4017	1.84
80	90	0.4556	0.3995	1.87
	180	1.3356	0.3995	1.83
	360	5.1658	0.3994	1.77
	720	25.204	0.3994	1.77
120	90	1.0090	0.3987	1.91
	180	2.9070	0.3987	1.88
	360	11.366	0.3986	1.83
	720	45.419	0.3985	1.78

The parameters from Table 2 are:

Z – number of steps in radial direction;
K – number of steps in angular direction;
τ – elapsed time in the code loop until Eq. (22) is achieved;
T_m – average temperature;
β_{opt} – optimal over-relaxation parameter.

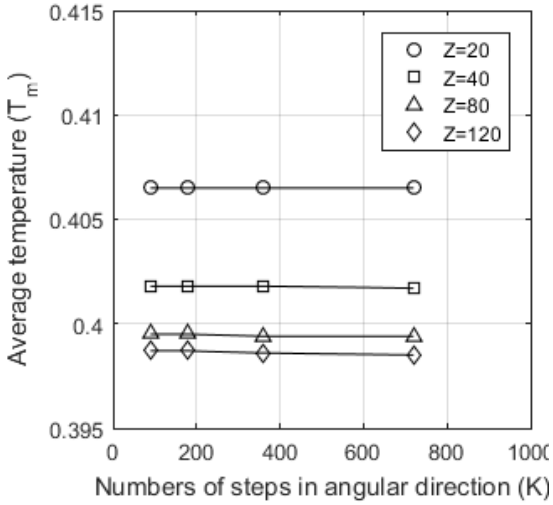


Fig. 2 – Average temperature as a function of K.

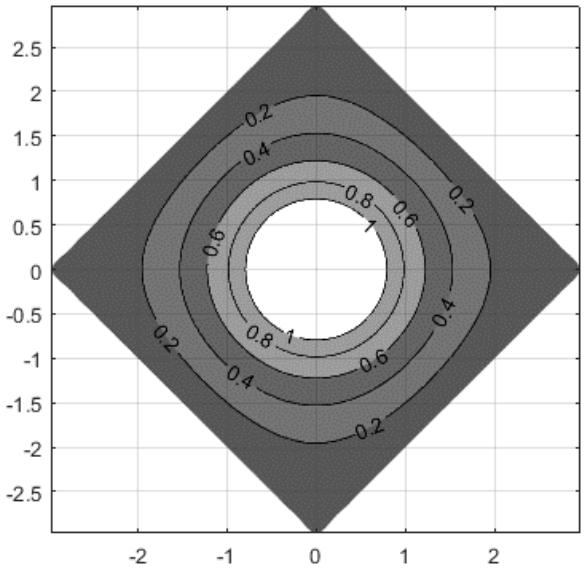


Fig. 3 – Contour plot of temperature field.

One can notice that the average temperature is dependent more on the step size in radial direction than in angular direction. As the radial step decreases, the average temperature tends to become constant. Also, it can be noted that the results were dependent on the parameters N and M, which were not taken in consideration in the present study. A future study will investigate the influence of these parameters and the model's accuracy.

6. Conclusions

The proposed model was found to be very useful in modeling different steady-state diffusion phenomena, where the domain can be approximated with the presented method. The model has been tested for simple regular polygons, but it can be extended to any other types of closed curves. The boundary conditions can be specified on the whole curve as a single function, which constitutes an advantage as this prevents abrupt changes or discontinuity points.

REFERENCES

Bigham S., Shokouhmand H., Isfahani R.N., Yazdani S., *Fluid flow and heat transfer simulation in a constricted microchannel: effects of rarefaction, geometry, and viscous dissipation*, Numerical Heat Transfer, Part A: Applications, 59(3), 209-230 (2011).

Fan J.F., Ding W.K., He Y.L., Tao W.Q, *Three-dimensional numerical study of fluid and heat transfer characteristics of dimpled fin surfaces*, Numerical Heat Transfer, Part A: Applications, 62(4), 271-294 (2012).

Lau P.C.M., *Curvilinear finite difference method for three-dimensional potential problems*, Journal of Computational Physics 32.3 (1979): 325-344.

Mazumder Sandip, *Numerical methods for partial differential equations: finite difference and finite volume methods*, Academic Press, 2015.

Öztop H.F., *Numerical study of flow and heat transfer in curvilinear ducts: applications of elliptic grid generation*, Applied mathematics and computation 168.2 (2005): 1449-1460.

Robertson S.R., *A finite-difference formulation of the equation of heat conduction in generalized coordinates*, Numerical Heat Transfer 2.1 (1979): 61-80.

Schneider G.E., Strong A.B., Yovanovich M.M., *A physical approach to the finite difference solution of the conduction equation in orthogonal curvilinear coordinates*, American Society of Mechanical Engineers (1975).

METODĂ CU DIFERENȚE FINITE PENTRU REZOLVAREA ECUAȚIEI LAPLACE BIDIMENSIONALE ÎN COORDONATE CURBILINII

(Rezumat)

Această lucrare prezintă o metodă de rezolvare a ecuației Laplace bidimensională în coordonate curbilinii, utilizând o schemă cu diferențe finite. Ecuația Laplace este o ecuație diferențială folosită la modelarea diferitelor fenomenele fizice studiate în inginerie. Ecuația Laplace bidimensională are soluții analitice doar pentru domenii cu geometrie și condiții la limită simple (de prima și-a doua speță). Chiar și în aceste condiții, rezultatele au o utilitate practică limitată. Metoda prezentată înlătură

aceste limitări prin aproximarea domeniului de analizat folosind serii Fourier. Domeniul este definit de o curbă exterioară și una interioară, ambele închise. Curbele sunt definite de un set de puncte, care mai apoi sunt aproximate într-o funcție continuă cu variabilele rază și unghi. Folosind o metodă din literatura de specialitate, ecuația Laplace este transformată din coordonate carteziene în coordonate polare modificate. Ecuația diferențială rezultată este aproximată într-un sistem de ecuații algebrice, câte o ecuație pentru fiecare nod din domeniul discretizat. Sistemul de ecuații este rezolvat folosind metoda supra-relaxării succesive, metodă iterativă care este implementată într-un program scris în limbajul de programare MATLAB.

Simularea difuzei se face pentru condiții la limită de speța întâi uniforme, pentru o serie de cazuri în care variază lungimea pasului spațial pe direcții radială și tangențială.

Rezultatele se analizează pentru a determina viteza, precizia și condițiile de convergență ale modelului.