Second International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, September 29 - October 02, 2011

A prolongation technique for solving partial differential equations with a multigrid method

Gabriela Nuţ

Abstract

The purpose of this paper is to introduce a new prolongation method for solving partial differential equations by a numerical method of multilevel type. This new technique is compared with others already existing in the literature, by means of some numerical results.

1 Introduction

The partial differential equations (PDE's) are used for modeling various real processes and phenomena in many different fields (fluid mechanics, thermodynamics, economics,...). Solving a PDE is thus very important, practically and theoretically. Because the analytical solution is not always available, it is important to consider numerical methods for approximating the solution of such an equation.

It is well known from the literature that the most used discretization methods are the finite difference and the finite element methods (see [1], [3]). In this paper, we use them both. In order to do this, as in [2], [4] and [5], the domain is divided in rectangular subdomains, having the same step on both Ox and Oy directions.The solution of the systems generated through discretization is obtained by Gauss full elimination method. The first level on wich the solution is computed is denoted by l_0, then this particular solution is used for generating the solutions on higher order levels, according with [4] and [5]. The grid on the l level l is divided by the one from the l_0 level in subdomains. On each of these, the system of linear equations obtained through the discretization method will be solved.

2 The problems

1. Convection-diffusion equations

It is known that the general expression of a convection-diffusion problem in two dimensions is given by:

$$\begin{cases} m\Delta u + \mathbf{n} \bigtriangledown u = f, & (x,y) \in \Omega, \\ u = g, & (x,y) \in \partial\Omega, \end{cases}$$

where $\mathbf{n} = (n_1, n_2)$ a flow velocity field and m is the coefficient of diffusion or viscosity.

As an example, let's consider the following problem:

$$\begin{cases} -e\Delta u + au_x = f, & (x,y) \in (0,1)\times(0,1) = \Omega, \\ u = 0, & (x,y) \in \partial\Omega, \end{cases} \tag{1}$$

where $f = x(1-x)\sin qy$.
The exact solution in this case can be determined analytically and has the expression:
$u_{ex} = \left(Ax^2 + Bx + C\right)\sin qy$, with $A = \frac{1}{t}, B = \frac{1-2aA}{t}, C = \frac{2eA-aB}{t}, t = eq^2$.

2. Poisson's equation

The second problem we consider is given by a Poisson equation of the form:

$$\begin{cases} -\Delta u = f, & (x,y) \in \Omega, \\ u = g, & (x,y) \in \partial\Omega. \end{cases} \tag{2}$$

Remark. Even if the exact solutions of these two problems are relatively easy to be determined, in the following paragraphs we shall compute also their numerical approximations, in order to introduce our prolongation method and establish its efficiency.

3 Finite difference and finite element discretizations

The partial differential equations will be replaced by a liniar system of equations through the discretization methods such as finite difference and finite element discretization.

In order to achieve this, and keeping the notations used in [4], we choose a grid step $h_l = \frac{1}{2^{l+1}}$, l being the number of the level. The corresponding number of grid points is $n_l = 2^{l+1}-1$ on each direction. The grid that has a step $h_l = \frac{1}{n_l+1} = \frac{1}{2^{l+1}}$ will contain the points $(x_i, y_j), i, j = 1, 2, \ldots, n_l$, and will be denoted by G_l. The value of the exact solution in the point (x_i, y_j) is denoted by $u_{i,j}$.

Remark. The numerical solution together with all discretizations involved are made for the convection-diffusion equation (1), because the Poisson's equation (2) is obtained by replacing parameter a by zero.

3.1 Second order finite difference discretization

Expanding in Taylor series the values of the function in the grid points, as in [4], one can compute approximations of the derivatives from the differential equation:

$$u_{i+1,j} = u_{i,j} + \frac{h}{1!}\frac{\partial u}{\partial x}(x_i,y_j) + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}(x_i,y_j) + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i,y_j) + \ldots; \tag{3}$$

$$u_{i-1,j} = u_{i,j} - \frac{h}{1!}\frac{\partial u}{\partial x}(x_i,y_j) + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}(x_i,y_j) - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}(x_i,y_j) + \ldots. \tag{4}$$

The approximation for the second order partial derivative is then:

$$\frac{\partial^2 u}{\partial x^2}(x_i,y_j) = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2). \tag{5}$$

An analogous result holds for the y-direction derivative:

$$\frac{\partial^2 u}{\partial y^2}(x_i,y_j) = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2). \tag{6}$$

The approximation for the first order partial derivative in the x-direction is :

$$\frac{\partial u}{\partial x}(x_i, y_j) = \frac{u_{i+1,j} - u_{i-1,j}}{2h} + O(h^2). \tag{7}$$

Equation (1) will have the following discrete formulation:

$$\begin{cases} -e\left(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2} + \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}\right) + a\frac{u_{i+1,j}-u_{i-1,j}}{2h} = f_{i,j}, \\ \qquad\qquad\qquad\qquad\qquad\qquad i, j = 1, 2, ..., n_l + 1, \\ \\ u_{i,j} = 0, \qquad i \in \{0, n_l + 2\} \text{ or } j \in \{0, n_l + 2\}. \end{cases} \tag{8}$$

The difference equations above are abbreviated by the stencil notation:

$$\frac{1}{h^2}\left[\!\!\left[\begin{array}{ccc} & -e & \\ -e - \frac{ah}{2} & 4e & -e + \frac{ah}{2} \\ & -e & \end{array} \right]\!\!\right] u = f, \tag{9}$$

where :

$$\begin{aligned} \left[\!\!\left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & k \end{array} \right]\!\!\right] u(i,j) &= au(i-1, j+1) + bu(i, j+1) + cu(i+1, j+1) + \\ &+ du(i-1, j) + eu(i, j) + fu(i+1, j) + gu(i-1, j-1) \\ &+ hu(i, j-1) + ku(i+1, j-1). \end{aligned} \tag{10}$$

3.2 Finite element discretization

According to [1], in order to apply the finite element discretization, some transformations of the given equation have to be made. So, the equation to be discretized is multiplied by a test function v, then is integrated on the domain Ω:

$$-e\iint_\Omega v\Delta u dxdy + a\iint_\Omega v\frac{\partial u}{\partial x}dxdy = \iint_\Omega fvdxdy.$$

Using Green's formula, the equation above becomes :

$$\begin{gathered} e\iint_\Omega \nabla u \nabla v dxdy - e\int_{\partial\Omega} v\frac{\partial u}{\partial n}ds + a\iint_\Omega v\frac{\partial u}{\partial x}dxdy = \iint_\Omega fvdxdy, \\ u, v \in H^1(\Omega), \iint_\Omega \left[\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + v^2\right] dxdy < \infty. \end{gathered} \tag{11}$$

The functions u and v are approximated using some continuous functions, Φ_i ($\Phi_i(x_j, y_j) = \delta_{ij}, i, j = 1, ..., N$, $N = n_l^2$ being the number of interior points of the grid on level l), through the relations: $u \approx \sum_{i=1}^N u_i\Phi_i, v \approx \sum_{j=1}^N v_j\Phi_j$, where $u_i = u(x_i, y_i), i = 1, ..., N$. Replacing these approximations in equation (11), the system obtained is:

$$\sum_{j=1}^N K_{ij}u_j = F_i, i = 1, ..., N, \tag{12}$$

where:

$$K_{ij} = \iint_\Omega \left[e\left(\frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_j}{\partial x} + \frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_j}{\partial y}\right) + a\Phi_i\frac{\partial\Phi_j}{\partial x}\right] dxdy, \tag{13}$$

$$F_i = \iint_\Omega f\Phi_i dxdy. \tag{14}$$

As $\Omega = \bigcup_{i=1}^{(n_l+1)^2}\Omega_i$, the above integrals can be rewritten as sums of integrals on the interior domains. Because the functions Φ_i are different from zero only on the subdomains immediately next to the point (x_i, y_i), the sums corresponding to the node i in the system (23) will only contain the integrals on the Ω_A, Ω_B, Ω_C and Ω_D domains:

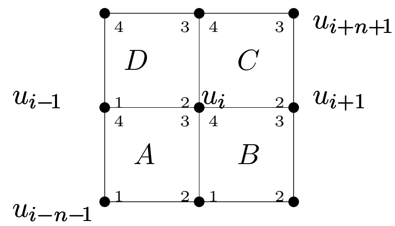

Fig. 1.

$$K_{i,i-n-1}^{(A)}u_{i-n-1} + K_{i,i-n}^{(A+B)}u_{i-n} + K_{i,i-n+1}^{(B)}u_{i-n+1} + K_{i,i-1}^{(A+D)}u_{i-1} + K_{i,i}^{(A+B+C+D)}u_i +$$
$$+K_{i,i+1}^{(B+C)}u_{i+1} + K_{i,i+n-1}^{(D)}u_{i+n-1} + K_{i,i+n}^{(C+D)}u_{i+n} + K_{i,i+n+1}^{(C)}u_{i+n+1} = F_i^{(A+B+C+D)},$$
$$i = 1, ..., N. \qquad (15)$$

In the equation (15) the superscript (A) or (A+B) means that the corresponding integrals in (13) and (14) are computed on Ω_A or $\Omega_A \cup \Omega_B$.

Further we denote the restrictions on Ω_A with Ψ_3^A for Φ_i, Ψ_4^A for Φ_{i-1}, Ψ_1^A for Φ_{i-n-1} and Ψ_2^A for Φ_{i-n}. If the domain Ω_A is $[a,b]\times[c,d]$, then the following expresions can be obtained:

$$\Psi_1^A = \frac{(x-b)(y-d)}{(b-a)(d-c)}, \Psi_2^A = -\frac{(x-a)(y-d)}{(b-a)(d-c)}, \Psi_3^A = \frac{(x-a)(y-c)}{(b-a)(d-c)}, \Psi_4^A = -\frac{(x-b)(y-c)}{(b-a)(d-c)}$$
$$\left(\Psi_k^A = \epsilon\frac{(x-\alpha)(y-\beta)}{(b-a)(d-c)}, k = 1, ..., 4\right). \qquad (16)$$

The restrictions of K and F on a domain Ω_A are:

$$k_{ij}^A = \iint_{\Omega_A}\left[e\left(\frac{\partial\Psi_i^A}{\partial x}\frac{\partial\Psi_j^A}{\partial x} + \frac{\partial\Psi_i^A}{\partial y}\frac{\partial\Psi_j^A}{\partial y}\right) + a\Psi_i^A\frac{\partial\Psi_j^A}{\partial x}\right]dxdy, \quad i,j = 1, ..., 4. \qquad (17)$$

$$f_k^A = \iint_{\Omega_A} f(x,y)\Psi_k^A(x,y)dxdy, \quad k = 1, ..., 4, \qquad (18)$$

With these notations, in equation (15) all the coefficients can be determined like in the following model:

$$\begin{aligned}
K_{i,i-n}^{(A+B)} &= \iint_{\Omega_A\cup\Omega_B}\left[e\left(\frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_{i-n}}{\partial x} + \frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_{i-n}}{\partial y}\right) + a\Phi_i\frac{\partial\Phi_{i-n}}{\partial x}\right]dxdy = \\
&= \iint_{\Omega_A}\left[e\left(\frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_{i-n}}{\partial x} + \frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_{i-n}}{\partial y}\right) + a\Phi_i\frac{\partial\Phi_{i-n}}{\partial x}\right]dxdy + \\
&+ \iint_{\Omega_B}\left[e\left(\frac{\partial\Phi_i}{\partial x}\frac{\partial\Phi_{i-n}}{\partial x} + \frac{\partial\Phi_i}{\partial y}\frac{\partial\Phi_{i-n}}{\partial y}\right) + a\Phi_i\frac{\partial\Phi_{i-n}}{\partial x}\right]dxdy = \\
&= \iint_{\Omega_A}\left[e\left(\frac{\partial\Psi_3^A}{\partial x}\frac{\partial\Psi_2^A}{\partial x} + \frac{\partial\Psi_3^A}{\partial y}\frac{\partial\Psi_2^A}{\partial y}\right) + a\Psi_3^A\frac{\partial\Psi_2^A}{\partial x}\right]dxdy + \\
&+ \iint_{\Omega_B}\left[e\left(\frac{\partial\Psi_4^B}{\partial x}\frac{\partial\Psi_1^B}{\partial x} + \frac{\partial\Psi_4^B}{\partial y}\frac{\partial\Psi_1^B}{\partial y}\right) + a\Psi_4^B\frac{\partial\Psi_1^B}{\partial x}\right]dxdy = k_{32}^A + k_{41}^B. \qquad (19)
\end{aligned}$$

Thus the equation (15) can be rewritten as:

$$\left[\!\!\left[\begin{array}{ccc} k_{24}^{D} & k_{23}^{D}+k_{14}^{C} & k_{13}^{C} \\ k_{34}^{A}+k_{21}^{D} & k_{33}^{A}+k_{44}^{B}+k_{11}^{C}+k_{22}^{D} & k_{43}^{B}+k_{12}^{C} \\ k_{31}^{A} & k_{32}^{A}+k_{41}^{B} & k_{42}^{B} \end{array}\right]\!\!\right] u_i = f_3^A + f_4^B + f_1^C + f_2^D,\, i=1,...,N. \quad (20)$$

If $x = \frac{d-c}{b-a}$, $y = \frac{d-c}{h_0}$, and replacing (16) in (17),(18), the values on Ω_A for the model problem (1) are as follows:

$$\left(k_{ij}^{A}\right)_{i,j=\overline{1:4}} = \frac{e}{6}\left[\!\!\left[\begin{array}{cccc} 2x+\frac{2}{x} & -2x+\frac{1}{x} & -x-\frac{1}{x} & x-\frac{2}{x} \\ -2x+\frac{1}{x} & 2x+\frac{2}{x} & x-\frac{2}{x} & -x-\frac{1}{x} \\ -x-\frac{1}{x} & x-\frac{2}{x} & 2x+\frac{2}{x} & -2x+\frac{1}{x} \\ x-\frac{2}{x} & -x-\frac{1}{x} & -2x+\frac{1}{x} & 2x+\frac{2}{x} \end{array}\right]\!\!\right] + y\frac{ah_0}{12}\left[\!\!\left[\begin{array}{cccc} -2 & 2 & 1 & -1 \\ -2 & 2 & 1 & -1 \\ -1 & 1 & 2 & -2 \\ -1 & 1 & 2 & -2 \end{array}\right]\!\!\right] \quad (21)$$

$$f_k^A = \frac{\epsilon}{(b-a)(d-c)}\left(\frac{x^4}{4} - (1+\alpha)\frac{x^3}{3} + \alpha\frac{x^2}{2}\right)\Bigg|_a^b \left(\frac{(y-\beta)\cos qy}{q} - \frac{\sin qy}{q^2}\right)\Bigg|_c^d . \quad (22)$$

4 Prolongation methods

The systems of equations (9) or (20) generated in the previous section can be written on any level l. Each system contains n_l^2 unknowns. The solution is exactly computed on a level l_0, for example on l_0=2 or l_0=3 using Gauss elimination method with partial pivoting. Thus the exact solution on the level l_0, for the problem is approximated by $u_i, i \in \{1, 2, ..., n_{l_0}^2\}$ (Fig. 2.), wich only contains an error due to the discretization.

In order to solve problem on the level l, the grid already obtained has to be further divided. Thus, each domain from the grid, $\Omega_k, k = 1, ..., (n_0+1)^2$, will be splitted into $(n_i+1)^2$ subdomains, where $n_i = 2^{l_i+1}-1$, and $l_i = l - l_0 - 1$. On each subdomain Ω_k, the discretization of the differential equation leads to a system whose matrix has the same form as the one on l_0 level. But on the level l_0 the boundary values were given in the hypothesis. For the systems on the level l to be precisely solved on Ω_k, one has to determine as accurate as possible the n_i interior values on each of the sides of the domain Ω_k. Two possible ways to accomplish this are given in the following subsections.

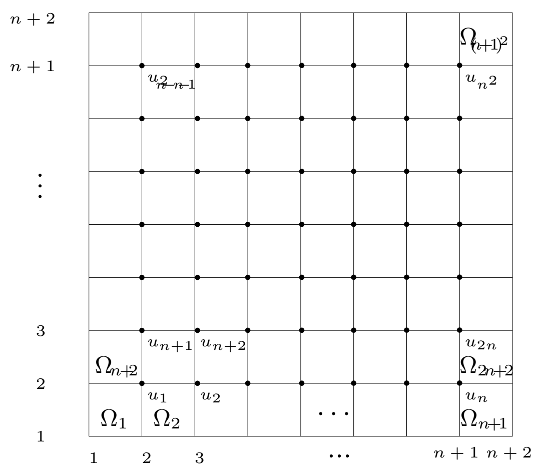

Fig. 2.

4.1 Pondered arithmetic mean prolongation

As in [2], the value of the approximation on level l is denoted by $u^{(l)}$. On the borders of Ω_k, they are defined through the following relations ($n = n_{l_0}$, $n_i = n_{l_i}, l_i = l - l_0 - 1, N = n_i + 1$):

$u_{jN+1,iN+1}^{(l)} = u_{(i-1)n+j}^{(l_0)}, i = 0, .., n, j = 1, .., n$ for the common points of the grids G_{l_0} and G_l.

For the grid points of G_l that do not belong to G_{l_0} :

$$u_{jN+1,iN+1+k}^{(l)} = \frac{1}{N}\left(ku_{in+j}^{(l_0)} + (N-k)u_{(i-1)n+j}^{(l_0)}\right), i = 0, ..., n, j = 1, ..., n;$$

$$u_{jN+1+k,iN+1}^{(l)} = \frac{1}{N}\left(ku_{(i-1)n+j+1}^{(l_0)} + (N-k)u_{(i-1)n+j}^{(l_0)}\right), i = 1, ..., n, j = 0, ..., n, k = 1, ..., n_i.$$

4.2 Stellar prolongation

In what follows, we introduce a new type of prolongation which we call "stellar prolongation" because the nodes involved in computation are in the shape of a star.

In order to determine more accurately the values of the solution on the borders of Ω_k, instead of pondered arithmetic mean prolongation one can use the solutions of the systems obtained discretizing the initial equation in the grid points corresponding to the values a_i and b_i, $i = 1, 2, ..., n^2 + n, n = n_{l_0}$ from Fig. 4 and Fig. 5.

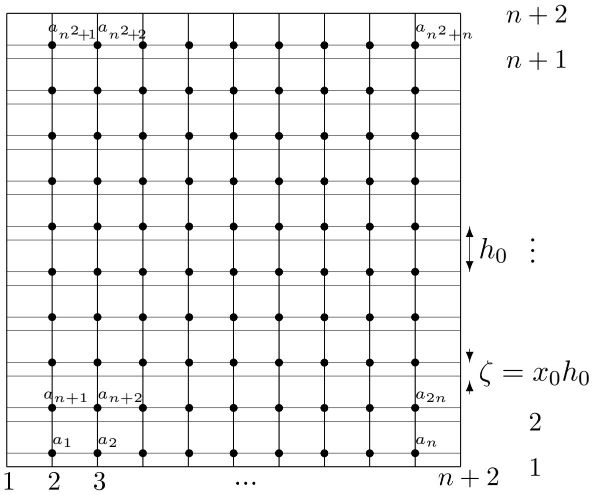

Fig. 4.

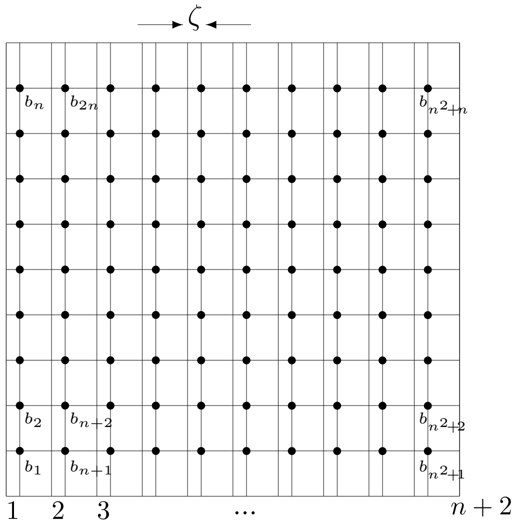

Fig. 5.

4.2.1 Finite difference discretization

The values $a_k, k = 1, 2, ..., n_0(n_0 + 1)$ depend on their vertical distance, ζ, from the old grid G_0 (marked with the thin lines in Fig.4) and will be further denoted by $a_k(\zeta)$, $\zeta = jh, j = 1, ..., n_i$. They are the solutions of the following system:

$$Aa = T \tag{23}$$

where the matrix A is:

$$A = \begin{bmatrix} C & D & \Theta & ... & \Theta & \Theta \\ S & C & D & ... & \Theta & \Theta \\ \Theta & S & C & ... & \Theta & \Theta \\ \vdots & & & \ddots & & \\ \vdots & & & & \ddots & \\ \Theta & \Theta & \Theta & ... & S & C \end{bmatrix}, \tag{24}$$

$$C = \begin{bmatrix} q_c & q_r & 0 & ... & 0 \\ q_l & q_c & q_r & ... & 0 \\ 0 & q_l & q_c & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_c \end{bmatrix}, D = \begin{bmatrix} q_u & 0 & 0 & ... & 0 \\ 0 & q_u & 0 & ... & 0 \\ 0 & 0 & q_u & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_u \end{bmatrix}, S = \begin{bmatrix} q_d & 0 & 0 & ... & 0 \\ 0 & q_d & 0 & ... & 0 \\ 0 & 0 & q_d & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_d \end{bmatrix},$$

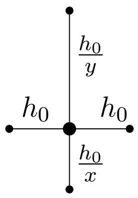

Fig. 6.

For the first and the last line of blocks in A, the neighbours of the node where the discretization is made are illustrated in Fig. 6. For the first line: $x = \frac{h_0}{\zeta}$, $y = 1$, for the last one: $x = 1$, $y = \frac{h_0}{h_0 - \zeta}$, whereas for the remaining lines: $x = 1$ and $y = 1$.

$$\left[\!\!\left[\begin{array}{ccc} & q_u & \\ q_l & q_c & q_r \\ & q_d & \end{array} \right]\!\!\right] = \left[\!\!\left[\begin{array}{ccc} & -e\alpha y & \\ -e - \frac{ah_0}{2} & e[2 + \alpha(x+y)] & -e + \frac{ah_0}{2} \\ & -e\alpha x & \end{array} \right]\!\!\right].$$

The vector of constant terms, T, has the components:

$$t_{in+j} = h_0^2 f\left(jh_0, (ih_0 + \zeta)\right) - \left[\!\!\left[\begin{array}{ccc} & q_u & \\ q_l & q_c & q_r \\ & q_d & \end{array} \right]\!\!\right] u_{fr}\left(jh_0, (ih_0 + \zeta)\right),$$
$$i = 0, ..., n_0, j = 1, ..., n_0,$$

u_{fr} being a function which is zero inside the domain Ω on wich the system is solved and equal to the border values on $\partial\Omega$ and h_0 is the grid step on l_0 level.

According to the kind of discretization that is used, the values of the parameter α are: $\alpha = \frac{xy}{x+y}$ (symmetric finite differences), $\alpha = x$ (backward finite differences), $\alpha = y$ (forward finite differences).

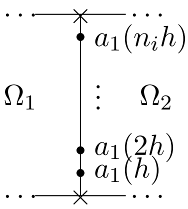

Fig. 7.

As ζ takes the values $h, 2h, ..., n_i h$, the values $a_k(\zeta)$ obtained from the system (23) will be used as border data on the vertical sides of Ω_k (for example, on the right vertical side of Ω_1 and the left side for Ω_2, they are corresponding to the points marked with a dot in Fig.7).

The values $b_k(\zeta), k = 1, 2, ..., n_0(n_0+1)$ depend on their horizontal pozition, ζ (see Fig.5) and are computed by solving a system whose matrix is also of the form (24), but in which:

$$C = \begin{bmatrix} q_c & q_u & 0 & ... & 0 \\ q_d & q_c & q_u & ... & 0 \\ 0 & q_d & q_c & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_c \end{bmatrix}, D = \begin{bmatrix} q_r & 0 & 0 & ... & 0 \\ 0 & q_r & 0 & ... & 0 \\ 0 & 0 & q_r & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_r \end{bmatrix}, S = \begin{bmatrix} q_l & 0 & 0 & ... & 0 \\ 0 & q_l & 0 & ... & 0 \\ 0 & 0 & q_l & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & q_l \end{bmatrix},$$

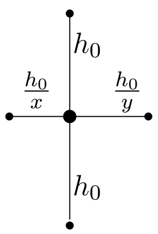

Fig. 8.

For the first and the last line of blocks in B, the nodes involved in the discretization are illustrated in Fig. 8. For the first line $x = \frac{h_0}{\zeta}$, $y = 1$, for the last one $x = 1$, $y = \frac{h_0}{h_0 - \zeta}$, and for the remaining lines: $x = 1$ and $y = 1$.

$$\left[\!\!\left[\begin{array}{ccc} & q_u & \\ q_l & q_c & q_r \\ & q_d & \end{array} \right]\!\!\right] = \left[\!\!\left[\begin{array}{ccc} & -e & \\ -e\alpha x + ah_0\rho\delta & e[2 + \alpha(x+y)] + a\gamma\rho & e\alpha y + ah_0\rho\beta \\ & -e & \end{array} \right]\!\!\right].$$

The constant terms vector now has the components:

$$t_{in+j} = h_0^2 f\left((ih_0+\zeta), jh_0\right) - \left[\!\!\left[\begin{array}{ccc} & q_u & \\ q_l & q_c & q_r \\ & q_d & \end{array} \right]\!\!\right] u_{fr}\left((ih_0+\zeta), jh_0\right),$$
$$i = 0, ..., n_0, j = 1, ..., n_0.$$

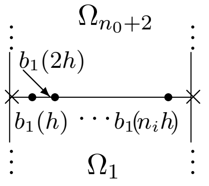

Fig. 9.

The values $b_k(\zeta), \zeta = 1, ..., n_i$ obtained from the system (23), with the matrix components described above, will be used as border data on the horizontal sides of Ω_k (for example, on the lower horizontal side of Ω_{n_0+2} and the upper side for Ω_1, they are corresponding to the points marked with a dot in Fig.9).

4.2.2 Finite element discretization

If the discretization is made by the finite element method, the same computing method is used, the only changes are in the matrices components. If we denote:

$$\left[\!\!\left[\begin{array}{ccc} l_1 & l_2 & l_3 \\ l_4 & l_5 & l_6 \\ l_7 & l_8 & l_9 \end{array} \right]\!\!\right] = \left[\!\!\left[\begin{array}{ccc} k_{24}^D & k_{23}^D + k_{14}^C & k_{13}^C \\ k_{34}^A + k_{21}^D & k_{33}^A + k_{44}^B + k_{11}^C + k_{22}^D & k_{43}^B + k_{12}^C \\ k_{31}^A & k_{32}^A + k_{41}^B & k_{42}^B \end{array} \right]\!\!\right],$$

where k_{ij} is given by (21), than the matrix A has:

$$C = \begin{bmatrix} l_5 & l_6 & 0 & ... & 0 \\ l_4 & l_5 & l_6 & ... & 0 \\ 0 & l_4 & l_5 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_5 \end{bmatrix}, D = \begin{bmatrix} l_2 & l_3 & 0 & ... & 0 \\ l_1 & l_2 & l_3 & ... & 0 \\ 0 & l_1 & l_2 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_2 \end{bmatrix}, S = \begin{bmatrix} l_8 & l_9 & 0 & ... & 0 \\ l_7 & l_8 & l_9 & ... & 0 \\ 0 & l_7 & l_8 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_8 \end{bmatrix},$$

and the components of the constant terms vector:

$$t_{in+j} = f_3^A + f_4^B + f_1^C + f^D - \left[\!\!\left[\begin{array}{ccc} l_1 & l_2 & l_3 \\ l_4 & l_5 & l_6 \\ l_7 & l_8 & l_9 \end{array} \right]\!\!\right] u_{fr}\left(jh, (i+x_0)h\right),$$
$$i = 0, ..., n, j = 1, ..., n.$$

For the B matrix:

$$C = \begin{bmatrix} l_5 & l_2 & 0 & ... & 0 \\ l_8 & l_5 & l_2 & ... & 0 \\ 0 & l_8 & l_5 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_5 \end{bmatrix}, D = \begin{bmatrix} l_6 & l_3 & 0 & ... & 0 \\ l_9 & l_6 & l_3 & ... & 0 \\ 0 & l_9 & l_6 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_6 \end{bmatrix}, S = \begin{bmatrix} l_4 & l_1 & 0 & ... & 0 \\ l_7 & l_4 & l_1 & ... & 0 \\ 0 & l_7 & l_4 & ... & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & ... & l_4 \end{bmatrix}.$$

The components of the constant terms vector:

$$t_{in+j} = f_3^A + f_4^B + f_1^C + f_2^D - \left[\!\!\left[\begin{array}{ccc} l_1 & l_2 & l_3 \\ l_4 & l_5 & l_6 \\ l_7 & l_8 & l_9 \end{array} \right]\!\!\right] u_{fr}\left(\left(i + x_0 \right) h, jh \right),$$
$$i = 0, ..., n, j = 1, ..., n,$$

In the matrix A: $x = x_0$ and $y = x_0$ for the first line of blocks in (21), on Ω_A and Ω_B, $x = 1$ and $y = 1$ on Ω_C and Ω_D, while $x = 1$ and $y = 1$ for the last line of blocks on Ω_A and Ω_B, and $x = 1 - x_0$, $y = 1 - x_0$ on Ω_C and Ω_D. For the remainder of the lines: $x = 1$ and $y = 1$.

For the matrix B: $x = \frac{1}{x_0}$ and $y = 1$ for the first line of blocks on Ω_A and Ω_D, $x = 1$ and $y = 1$ on Ω_B and Ω_C. The last line has: $x = 1$ and $y = 1$ on Ω_A and Ω_D, and on Ω_B and Ω_C $x = \frac{1}{1-x_0}$ and $y = 1$. For the other lines: $x = 1$ and $y = 1$.

5 Solving method

The differential equation is first discretized on a grid G_0 and the solutions obtained solving the system resulted are the values $u_i, i = 1, ..., n_0^2$ situated in the corners of the subdomains Ω_k, $k = 1, ..., n_0(n_0 + 1)$ (Fig. 2).

These values are then used to compute a_k and $b_k, k = 1, ..., n_0(n_0 + 1)$ for each $\zeta = jh$, $j = 1, ..., n_i$ (as in Section 4.1 or 4.2).

Thus, on every subdomain Ω_{iN_0+j}, i=$0, ..., n_0, j$=$1, ..., n_0, N_0$=n_0+1 the values on the frontiers are now known:

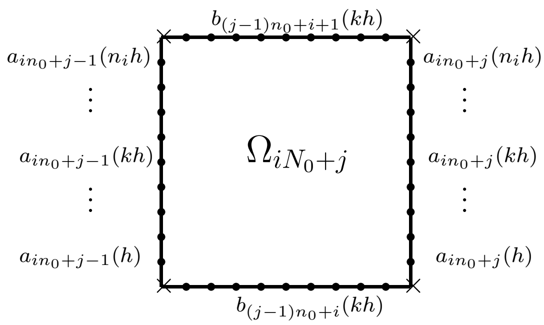

Fig. 10.

$$u^{(l)}_{(j-1)N+1,iN+1+k} = a_{in_0+j-1}(kh), u^{(l)}_{(j-1)N+1+k,iN+1} = b_{(j-1)n_0+i}(kh),$$

$k = 1, ..., n_0(n_0 + 1), j = 1, ..., n_i, h = \frac{1}{n_l+1}$ (see Fig. 10).

The problem is now discretized the on each subdomain $\Omega_{iN_0+j}, i = 0, ..., n_0, j = 1, ..., n_0$ and the solution obtained will represent the components of the final solution on the grid G_l.

Reuniting the solutions computed on the grid corresponding to the level l_0 and the ones from every subdomain, the final solution on the work level l is obtained.

6 Numerical results

In order to give some numerical results, we denote by:
FD-PAM: the finite difference discretization with pondered arithmetic mean prolongation,
FD-SP: the finite difference discretization with stellar prolongation,
FEM-SP: the finite element discretization with stellar prolongation.

We have computed the infinity norm of the difference between the computed solution and the exact solution. If the grid on the level l is $G_l = \{(x_i, y_j), i, j = 1, 2, ..., n_l\}$, then the error is:

$$\xi = ||u_{ex} - u_l||_\infty = max\{|u_{ex}(x, y) - u_l(x, y)|, (x, y) \in G_l\}.$$

In the following table there are the errors for the convection-diffusion problem (1) ($u_{ex} \leq 1.7839 \cdot 10^4$).

Level	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	32.8788	25.3345	0.7798
l=4	32.8788	25.3345	0.7877
l=5	32.8788	25.3345	0.7877

In order to compare the previous methods applied on a Poisson problem, we consider the following problems of this type, and their exact solutions.

$$\textbf{\textit{Pr.1}} \begin{cases} -\Delta u = -4, & (x,y) \in (0,1) \times (0,1), \\ u(x,y) = x^2 + y^2 = u_{ex}(x,y), & (x,y) \in \partial\Omega. \end{cases}$$

$$\textbf{\textit{Pr.2}} \begin{cases} -\Delta u = -\frac{3y}{x+1} - \frac{y^3}{(x+1)^3}, & (x,y) \in (0,1) \times (0,1), \\ u(x,y) = 0.5\frac{y^3}{x+1} = u_{ex}(x,y), & (x,y) \in \partial\Omega. \end{cases}$$

$$\textbf{\textit{Pr.3}} \begin{cases} -\Delta u = -\frac{3y}{x+0.1} - \frac{y^3}{(x+0.1)^3}, & (x,y) \in (0,1) \times (0,1), \\ u(x,y) = 0.5\frac{y^3}{x+0.1} = u_{ex}(x,y), & (x,y) \in \partial\Omega. \end{cases}$$

$$\textbf{\textit{Pr.4}} \begin{cases} -\Delta u = 2\pi^2 \sin \pi x \cos \pi y, & (x,y) \in (0,4) \times (0,1), \\ u(x,y) = 0, & (x,y) \in \partial\Omega; \end{cases}$$

$u_{ex}(x,y) = \sin \pi x \cos \pi y.$

$$\textbf{\textit{Pr.5}} \begin{cases} -\Delta u = \alpha \sin \frac{\pi y}{b}, & (x,y) \in [0,\lambda] \times [0,b], \\ u(x,y) = 0, & (x,y) \in \partial\Omega. \end{cases}$$

$u_{ex}(x,y) = -\alpha \left(\frac{b}{\pi}\right)^2 \sin \frac{\pi y}{b} \left(e^{\frac{\pi x}{b}} - 1\right),$

$\alpha = \frac{F\pi}{Rb}, \lambda = 10^7, b = 2\pi \cdot 10^6, F = 0.3 \cdot 10^{-7}, R = 0.6 \cdot 10^{-3}.$

Pr.1	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	$3.9 \cdot 10^{-03}$	$3.3 \cdot 10^{-03}$	0.5968
l=4	$3.4 \cdot 10^{-03}$	$3.3 \cdot 10^{-03}$	0.5968
l=5	$3.5 \cdot 10^{-03}$	$3.3 \cdot 10^{-03}$	0.5967

Pr.2	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	$4.9075 \cdot 10^{-3}$	$3.4025 \cdot 10^{-3}$	$2.2406 \cdot 10^{-4}$
l=4	$4.9075 \cdot 10^{-3}$	$3.4025 \cdot 10^{-3}$	$2.2406 \cdot 10^{-4}$
l=5	$4.9075 \cdot 10^{-3}$	$3.4428 \cdot 10^{-3}$	$2.2406 \cdot 10^{-4}$

Pr.3	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	0.4407	0.1930	0.0195
l=4	0.4407	0.4493	0.0195
l=5	0.4407	0.4600	0.0195

Pr.4	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	0.0170	0.0185	0.0042
l=4	0.0170	0.0185	0.0042
l=5	0.0170	0.0185	0.0042

Pr.5	ξ_{FD-PAM}	ξ_{FD-SP}	ξ_{FEM-SP}
l=3	$2.4991 \cdot 10^{-20}$	$1.7752 \cdot 10^{-20}$	$1.1379 \cdot 10^{-20}$
l=4	$2.4991 \cdot 10^{-20}$	$1.7752 \cdot 10^{-20}$	$1.3814 \cdot 10^{-20}$
l=5	$2.5688 \cdot 10^{-20}$	$1.7752 \cdot 10^{-20}$	$1.5718 \cdot 10^{-20}$

7 Conclusions

The numerical results indicate that our stellar prolongation method is more efficient than others used in the literature. Even if we applied it on some particular cases, we expect this behavior to be the same on other more general problems, too.

References

[1] E.B. Becker, G.F. Carey, J.T. Oden, *Finite Elements. An Introduction*, Texas Institute for Computational Mecanics, UT Austin, 1981.

[2] A. Brandt, Multilevel Adaptive Solutions to Boundary Value Problems. *Mathematics of Computation*, 31, 333-390, 1977.

[3] W. Hackbush, *Elliptic Differential Equations*, Springer-Verlag, New York, 1992.

[4] W. Hackbush, *Multigrid Method and Applications*, Springer, Berlin, Heidelberg, 1985.

[5] U. Trottenberg, C. Oosterlee, A. Schuller, *Multigrid*, Elsevier Academic Press, London, 2001.

Gabriela Nuţ
Babeş-Bolyai University, Cluj-Napoca
Faculty of Mathematics and Computer Science
1, Mihail Kogălniceanu str., Cluj Napoca, 400084
ROMANIA
E-mail: *djantai@yahoo.com*