

Brief note

THE EFFECT OF MULTIGRID PARAMETERS IN A 3D HEAT DIFFUSION EQUATION

F. DE OLIVEIRA *

State University of Ponta Grossa, Department of Mathematics and Statistics
Ponta Grossa – PR, BRAZIL
E-mail: fabiane1910@yahoo.com.br

S.R. FRANCO

State University of Central West, Department of Mathematics
Irati – PR, BRAZIL
Federal University of Paraná
Graduate Program in Numerical Methods in Engineering
Curitiba – PR, BRAZIL
E-mail: romero@unicentro.br

M.A. VILLELA PINTO

Federal University of Paraná, Department of Mechanical Engineering
Curitiba – PR, BRAZIL
E-mail: marcio_villela@ufpr.br

The aim of this paper is to reduce the necessary CPU time to solve the three-dimensional heat diffusion equation using Dirichlet boundary conditions. The finite difference method (FDM) is used to discretize the differential equations with a second-order accuracy central difference scheme (CDS). The algebraic equations systems are solved using the lexicographical and red-black Gauss-Seidel methods, associated with the geometric multigrid method with a correction scheme (CS) and V-cycle. Comparisons are made between two types of restriction: injection and full weighting. The used prolongation process is the trilinear interpolation. This work is concerned with the study of the influence of the smoothing value (ν), number of mesh levels (L) and number of unknowns (N) on the CPU time, as well as the analysis of algorithm complexity.

Key words: multigrid, finite difference, Poisson 3D, solvers, parameters.

1. Introduction

The multigrid method [1] belongs to the group of iterative solvers and it is one of the most efficient and widespread methods to solve large systems of linear equations [2]. It consists of the transference of information between a refined grid, in which the numerical solution is desired, and coarse grids, in which numerical smoothers (or solvers) are more efficient. The transference is done through restriction and prolongation operators. The kind of information that is transferred among the grids defines the multigrid scheme, which can be CS (correction scheme, in which only the residual is transferred to the coarser grids) or FAS (full approximation scheme, in which both residue and solution are transferred to the coarser grids) [3, 4].

* To whom correspondence should be addressed

Throughout the years, the multigrid method has become closely intertwined with Computational Fluid Dynamics (CFD) and has been widely employed in major CFD codes. However, the full theoretical efficiency of the multigrid method has not yet been achieved for realistic engineering applications in CFD [5].

There are many parameters that can be selected more or less arbitrarily when using the multigrid method. Some of these parameters are the type of cycle, CS or FAS scheme, number of iterations in the smoother, number of grids, the smoother itself, and also restriction and prolongation schemes. Nevertheless, according to Ferziger and Peric [6], the ratio of the CPU time when using the best set of parameters is probably smaller than a factor of two. However, Trottenberg *et al.* [4] state that the choice of a parameter can significantly affect the efficiency of the multigrid. Pinto *et al.* [7] presented the optimum multigrid parameters for 1D advection-diffusion, whereas Burges, Rabi and De Lemos [8] studied the optimum parameters for a 2D advection-diffusion problem. Santiago and Marchi [9] carried out an analysis on parameters for a Navier 2D problem involving two equations. Oliveira *et al.* [10] investigated types of coarsening rates for heat diffusion problems and anisotropic grids by using Poisson's and Laplace's equation. Suero *et al.* [11] analyzed algebraic multigrid parameters for two-dimensional steady-state heat diffusion equations.

Decreasing the CPU time for solving a problem results in the reduction of project costs. By increasing efficiency, it is also possible, along with computation, to solve a problem in a more refined grid, with more nodes. That is, obtaining a numerical solution with less discretization error [12] improves the quality and reliability of the project.

The purpose of this article is to verify the effect of the parameters on the CPU time for a geometric multigrid method. The optimum value of a parameter is found when the solution is obtained in the lowest CPU time for fixed values of the other parameters. Such optimization involves the study of the multigrid components: the number of inner iterations of the solver (ν), the number of grid levels (L), and also the type of restriction. In the present study, the term "CPU time" is used instead of "work units" due to the reasons pointed by Trottenberg *et al.* [4] and Larsson *et al.* [13]. This work involves the three-dimensional heat linear diffusion governed by Poisson's equation with Dirichlet boundary conditions. The Finite Difference Method (FDM) [14] is utilized with central approximate scheme (CDS) and uniform grids.

This article is organized as follows: Section 2 presents the mathematical and numerical models, including details for the multigrid method. Section 3 contains the numerical results, and Section 4 presents the conclusion.

2. Mathematical and numerical models

2.1 Mathematical models and discretizations

One three-dimensional heat transfer problem in the unitary cube domain, with Dirichlet boundary conditions, is investigated in this work. It is governed by the following differential equation in the Cartesian coordinate system [15]

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = S \quad (2.1)$$

where x , y and z are the coordinate directions, T is the temperature, and S is a source term. The Poisson-type equation has a source term, defined by

$$S = -3\pi^2 \cdot \text{sen}(\pi \cdot x) \cdot \text{sen}(\pi \cdot y) \cdot \text{sen}(\pi \cdot z). \quad (2.2)$$

The FDM [16] is employed in the discretization of Eq.(2.1), with the use of the second-order central differencing scheme (CDS). The unitary cube domain is divided into N_x nodes in the x -direction, N_y nodes in

the y -direction and N_z nodes in the z -direction, using uniform grids in each direction, totalizing N nodes ($N = N_x N_y N_z$).

$$\frac{T_{i-1,j,k} - 2T_{i,j,k} + T_{i+1,j,k}}{h^2} + \frac{T_{i,j+1,k} - 2T_{i,j,k} + T_{i,j-1,k}}{h^2} + \frac{T_{i,j,k-1} - 2T_{i,j,k} + T_{i,j,k+1}}{h^2} = S, \quad (2.3)$$

where h is the distance between two different nodes of the grid in any direction. Equation (2.3) can be represented by one system of linear equations of the type

$$AT = b, \quad (2.4)$$

which must be solved. By means of the discretization procedures applied, it can be noticed that the coefficients matrix A in Eq.(2.4) is heptadiagonal, symmetric and positive-definite; T is the solution temperature vector and b is the independent vector.

2.2. Multigrid method and other details

The linear equation, Eq.(2.4), is solved by using the geometric multigrid method (GMG), as described by Wesseling [17], with the correction scheme (CS). Among the several multigrid cycling schemes available in the literature, the V-cycle was chosen because of its simplicity for programming implementation and for its smaller computational work for isotropic problems: W-cycles, for example, are roughly 50% more expensive than V-cycles [18].

Trilinear interpolation, injection operator and full weighting [4] were chosen for all the studied cases. The solver algorithms employed in this work are based on the lexicographical Gauss-Seidel method and red-black Gauss-Seidel method [19, 20]. The lexicographical and red-black orderings are present in Fig.1. In all numerical simulations, the number of grid levels (L) was taken in such way that $1 \leq L \leq L_{\max}$, where L_{\max} is the maximum number of different grids that can be employed in the multigrid cycle. Standard coarsening with ratio $r = 2$ is used.

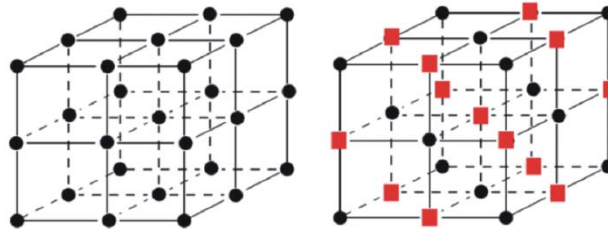


Fig.1. (a) Lexicographical and (b) red-black ordering. Adapted from Winfried and Christoph (2004).

Each V-cycle is repeated until the achievement of a given stop criterion, which is based on the non-dimensional l_2 -norm of the residual – the reference is the l_2 -norm of the initial guess – as found in [4]. The null value was taken as the initial guess for the whole domain. The admitted tolerance was equal to 10^{-10} for all the analyzed cases in this work.

The numerical codes were generated by using the Fortran Visual Studio 2008 with quadruple precision. The simulations were carried out on a microcomputer with a 2.66 GHz Intel Core 2 Duo processor, 8 GB of RAM and Windows xp 64 bits as the operating system.

3. Numeric results

With the purpose of analyzing the influence of different multigrid components on the CPU time in one heat diffusion problem, several numerical simulations were performed. The methodology employed consisted in, for a given component of interest, keeping the other ones with a fixed value and, by comparison, choosing the set of components that have shown the best performance. The numerical simulations belong to four categories: number of smoothing steps, type of restriction, number of grids and comparison among lexicographical and red-black Gauss-Seidel orderings.

3.1. Analysis of number of smoothing steps (ν)

The main purpose of this subsection is to establish the optimum number of smoothing steps ($\nu_{optimum}$), which provides the lowest CPU time (t_{cpu}) for a given set of components for isotropic grids, i.e., grids evenly spaced in all directions. In order to reduce the number of numerical simulations and to achieve t_{cpu} minimization for all the dependent variables, the number of levels is fixed equal to L_{max} , full weighting is chosen as the restriction operator and the lexicographical Gauss-Seidel (LEX-GS) and red-black Gauss-Seidel (RB-GS) as solvers.

Figure 2 shows the influence of the number of smoothings (ν) on CPU time for the 3D Poisson equation. For the lexicographical Gauss-Seidel solver, the number of smoothings that presented the lowest CPU time was $\nu_1 = \nu_2 = 3$, whereas for the red-black Gauss-Seidel solver it was $\nu_1 = \nu_2 = 2$. Notice that ν_1 and ν_2 are the numbers of pre- and post- smoothings, respectively.

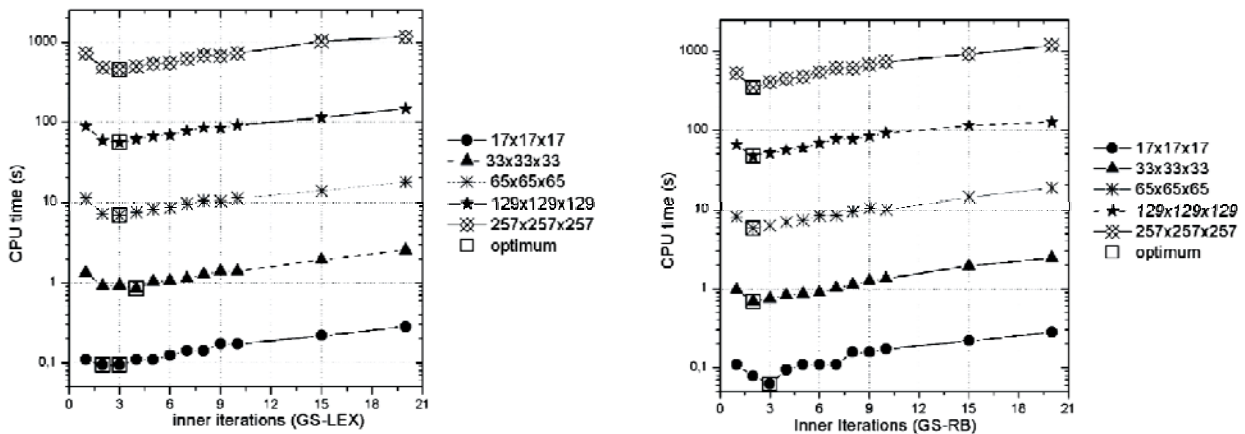


Fig.2. Analysis of number of smoothings for the lexicographical Gauss-Seidel and red-black Gauss-Seidel, both using full weighting restriction.

3.2. Analysis of type of restriction

Two types of restriction were considered: injection (INJ) and full weighting (FW). The injection restriction identifies and relates grid functions to coarse grid points and with their corresponding grid functions at fine grid points. The full weighting restriction is given by Eq.(3.1), as described by Wesseling [17]

$$\frac{1}{64} \begin{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}^{2h} & \begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix}^{2h} & \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}^{2h} \\ \hline \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h & \begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix}_h & \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_h \end{bmatrix}. \tag{3.1}$$

The influence of the number of smoothings (ν) on CPU time using the two types of restriction was tested using the lexicographical Gauss-Seidel solver. The red-black Gauss-Seidel solver was tested only with full weighting restriction as it diverges when used with injection restriction and lexicographical ordination. The optimum number of smoothings using injection restriction was $\nu_1 = \nu_2 = 5$, whereas for full weighting restriction it was $\nu_1 = \nu_2 = 3$. The lowest CPU time using the lexicographical Gauss-Seidel solver was obtained using injection restriction. For the $257 \times 257 \times 257$ grid, it was about one time faster than the full weighting restriction.

3.3. Grid levels (L)

Some authors, such as Suero *et al.* [11] and Gaspar [22], have also analyzed the number of grid levels for problems involving the multigrid method. In the previous subsections, the number of grids used for the multigrid was kept invariable and equal to L_{\max} . Since other components that influence the CPU time performance have been previously studied, in this subsection only the effect of L on the CPU time performance was evaluated.

Figure 3 summarizes the numerical results obtained of the effect of L on CPU time for the Poisson equation and red-black Gauss-Seidel solver. This solver was used because it results in a lower CPU time than the lexicographical Gauss Seidel. It is easily observed that the lowest CPU time is achieved when using L_{\max} . Similar results were achieved with the lexicographical Gauss-Seidel solver.

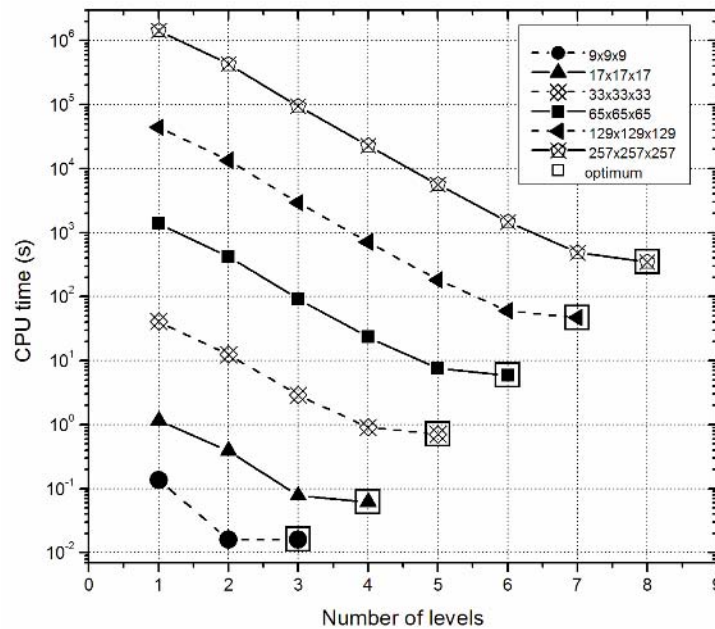


Fig.3. Number of levels versus the CPU time for the grids (from $N = 5 \times 5 \times 5$ to $N = 257 \times 257 \times 257$) for the Poisson equation and red-black Gauss-Seidel solver.

Results of the current work agree with those presented by Tannehill *et al.* [16]. These authors have investigated the existence of optimal components in multigrid method in a two-dimensional Laplace problem, using an isotropic 128×128 element grid, and from two to seven grids, and reported, regarding the CPU time, that the performance when using four or five grids was nearly the same as when using seven grids (which corresponds to L_{\max}). According to the numerical results of the current work, the recommended value of L is always L_{\max} .

3.4. Comparison between solvers

This subsection aims to compare how much CPU time is necessary to solve the 3D Poisson equation using the lexicographical Gauss-Seidel solver with injection restriction, and red-black Gauss-Seidel solver with weighting restriction. Such types of restriction were chosen as they result in a lower CPU time, as shown in subsection 3.2.

Figure 4 shows the CPU time obtained with lexicographical Gauss-Seidel and red-black Gauss-Seidel solvers for the 3D Poisson equation. The CPU time obtained with the red-black Gauss-Seidel solver is slightly lower than with the lexicographical Gauss-Seidel solver. For the $257 \times 257 \times 257$ grid, it is about 1.17 times faster.

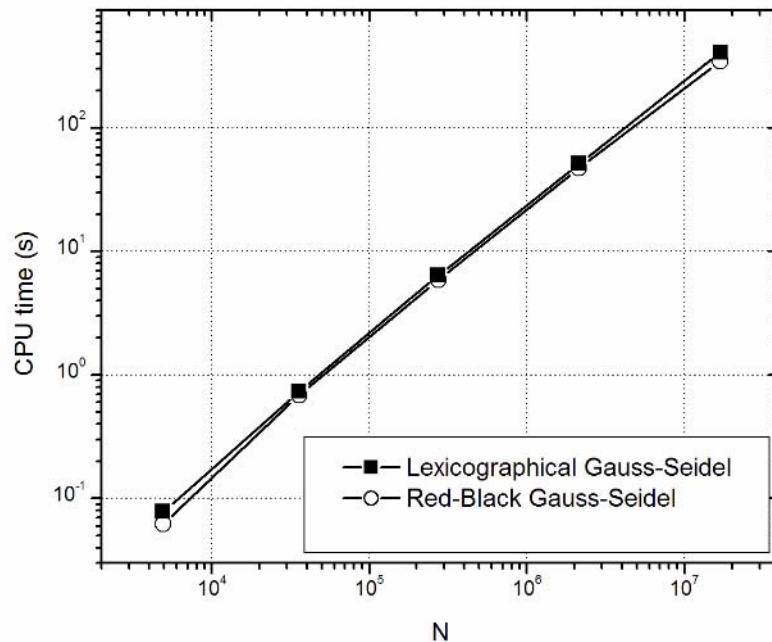


Fig.4. Comparison of the lexicographical Gauss-Seidel (injection restriction) and red-black Gauss-Seidel (full weighting restriction).

3.5. Complexity analysis

Exponent p , which was obtained using the least square method, was computed for the function $t_{cpu}(N) = cN^p$, where p is the order solver associated with the method used, c is the coefficient that depends on each method and N is the number of unknowns of the system. For the ideal multigrid $p = 1$, which means that the computational effort increases linearly with the size of grid [4, 18]. So, for a given hardware and compiler, as the value of p decreases, the efficiency of the algorithm increases.

Table 1 shows the coefficient for the geometric curve fitting for the multigrid method. The best result was obtained using the multigrid method with the red-black Gauss-Seidel solver and full weighting restriction.

The CPU time necessary to solve the 3D Poisson equation using the single grid and multigrid methods was assessed. The red-black Gauss-Seidel solver was used in both cases. Table 2 shows the coefficient for the geometric curve fitting for the single grid and multigrid methods, both using the red-black Gauss-Seidel solver, which is in agreement with the literature. For more refined grids, the coefficient p , on the multigrid method, tends to one.

Table 1. Coefficient for the geometric curve fitting.

Method	c	p
Multigrid Method (lexicographical Gauss-Seidel) full weighting restriction.	$1.5261E-05$	1.03558
Multigrid Method (lexicographical Gauss-Seidel) Injection Restriction.	$1.16102E-05$	1.04817
Multigrid Method (red-black Gauss-Seidel)	$9.53101E-06$	1.05397

Table 2. Coefficient for the geometric curve fitting.

Method	c	p
Multigrid Method (red-black Gauss-Seidel)	$9.53101E-06$	1.05397
Singlegrid Method (red-black Gauss-Seidel)	$1.60094E-06$	1.64497

4. Conclusion

This work aimed to assess the effect of multigrid components on CS grids. For the numerical analysis, three-dimensional heat diffusion problems governed by the Poisson equation, with Dirichlet boundary conditions, were employed and discretized with the Finite Difference Method and second order CDS approximations.

Based on the results of this work, these are the most interesting remarks:

- 1) The number of smoothings influences CPU time. This parameter depends on the solver and restriction type. It was obtained:
 - a) $\nu = 5$ as the optimum number for the lexicographical Gauss-Seidel solver with injection restriction;
 - b) $\nu = 3$ as the optimum number for the lexicographical Gauss-Seidel solver with full weighting restriction;
 - c) $\nu = 2$ as the optimum number for the red-black Gauss-Seidel solver with full weighting restriction;
- 2) The CPU time obtained with the red-black Gauss-Seidel solver is slightly faster than with the lexicographical Gauss-Seidel solver. For the $257 \times 257 \times 257$ grid, it is about 1.17 times faster.
- 3) The number of levels influences the CPU time. The lowest CPU time was obtained with L_{\max} , regardless of the solver and restriction type used.
- 4) The CPU time obtained with the multigrid method is lower than with the single grid method. The orders of complexity obtained from these two methods are 1.05397 and 1.64497 , respectively.

Acknowledgements

This work was partially supported by the Brazilian agencies: CNPq (National Council for Scientific and Technological Development - Brazil) and CAPES (Coordination for the Improvement of Higher

Education Personnel - Brazil). The first author acknowledges the financial support granted by the State University of Ponta Grossa.

Nomenclature

- A – coefficients matrix
- b – independent vector
- L – number of levels
- L_{\max} – the maximum number of different grids that can be employed in the multigrid cycle
- N_x, N_y, N_z – nodes in x -direction, y -direction and z -direction respectively
- r – ratio
- S – source term
- T – temperature
- t_{cpu} – CPU time
- x, y, z – coordinate directions
- v – number of smoothing steps
- v_1, v_2 – smoothings steps on the restriction and prolongation respectively
- $v_{optimum}$ – numbers of pre- and post- smoothings

References

- [1] Briggs W.L., Henson V.E. and McCormick S.F. (2000): *A Multigrid Tutorial*. second ed. – Philadelphia: SIAM.
- [2] Thekale A., Gradl T., Klamroth K. and Rde U. (2010): *Optimizing the number of multigrid cycles in the full multigrid algorithm*. – Numer. Linear Algebra Appl., No.17, pp.199–210.
- [3] Brandt A. (1977): *Multi-level adaptive solutions to boundary-value problems*. – Math. Comput. No.31, pp.333–390.
- [4] Trottenberg U., Oosterlee C. and Schller A. (2001): *Multigrid*. – San Diego: Academic Press.
- [5] Mohamed S.A. (2008): *Optimally efficient multigrid algorithm for incompressible Euler equations*. – Int. J. Numer. Methods Heat Fluid Flow, No.18, pp.783–804.
- [6] Ferziger J.H. and Peric M. (2002). *Computational Methods for Fluid Dynamics*. – 3 ed. - Berlin: Springer.
- [7] Pinto A.M., Santiago C.D. and Marchi C.H. (2005): *Effect of Parameters of a Multigrid Method on CPU Time for One-dimensional Problems*. – Proceedings of COBEM.
- [8] Rabi J.A. and De Lemos M.J.S. (2001): *Optimization of convergence acceleration in multigrid numerical solutions of conductive–convective problems*. – Appl. Math. Comput. No.124, pp.215-226.
- [9] Santiago C.D. and Marchi C.H. (2007): *Optimum Parameters of a Geometric Multigrid for a Two-Dimensional Problem of Two-Equations*. – Proceedings of COBEM.
- [10] Oliveira F., Pinto M.A.V., Marchi C.H. and Araki L.K. (2012): *Optimized Partial Semicoarsening Multigrid Algorithm, for Heat Diffusion Problems and Anisotropic Grids*. – Appl. Math. Modell. No.36, pp.4665-4676.
- [11] Suero R., Pinto M.A.V., Marchi C.H., Araki L.K. and Alves A.C. (2012): *Analysis of the algebraic Multigrid parameters for two-dimensional steady-state diffusion equations*. – Appl. Math. Modell., No.36, pp.2996-3006.
- [12] Roache P.J. (1998): *Fundamentals of Computational Fluid Dynamics*. – Albuquerque, USA: Hermosa Publishers.
- [13] Larsson J., Lien F.S. and Yee E. (2005): *Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic Grids*. – J. Comput. Phys. No.208, pp.368-383.
- [14] Golub G. H. and Ortega J.M. (1992): *Scientific Computing and Differential Equations: an Introduction to Numerical Methods*. – Academic Press, Inc..
- [15] Incropera F.P., DeWitt D.P., Bergman T.L. and Lavine A.S. (2007): *Fundamentals of Heat and Mass Transfer*. – Sixth ed. - John Wiley & Sons.

- [16] Tannehill J.C., Anderson D.A. and Pletcher R.H. (1997): *Computational Fluid Mechanics and Heat Transfer*. – Second ed. - Washington: Taylor & Francis.
- [17] Wesseling P.(1992): *An Introduction to Multigrid Methods*. – Philadelphia: John Wiley & Sons.
- [18] Hirsch C. (1988): *Numerical Computational of Internal and External Flows*. Vol. 1. – Chichester: John Wiley & Sons.
- [19] Parter S.V. (1988): *Estimates for Multigrid Methods Based on Red-Black Gauss-Seidel Smooth*. – Numer. Math., No.52, pp.701-723.
- [20] J. Zhang (1996): *Multigrid Acceleration Techniques and Applications to the Numerical Solution of Partial Differential Equations*. – Dissertation. Chongqing Univesity. China.
- [21] Gaspar F.J., Gracia J.L., Lisbona F.J. and Rodrigo C. (2009): *On geometric Multigrid methods for triangular grids three-coarsening strategy*. – Appl. Numer. Math. No.59, pp.1693-1708.
- [22] Winfried A. and Christoph F. (2004): *Iterative Solution of Large Linear Systems Arising in the 3-Dimensional Modelling of an Electric Field in Human Thigh*. – Technical Report. ANUM Preprint No. 12/04.

Received: April 27, 2017

Revised: June 20, 2017