Mathematical Modelling and Analysis
Volume 13 Number 4, 2008, pages 471–482
Doi:10.3846/1392-6292.2008.13.471-482
 ISSN 1392-6292 print, ISSN 1648-3510 online

PARALLEL NUMERICAL ALGORITHMS FOR OPTIMIZATION OF ELECTRICAL CABLES

RAIM. ČIEGIS[1], REM. ČIEGIS[2], M. MEILŪNAS[1],
G. JANKEVIČIŪTĖ[1] and V. STARIKOVIČIUS[1]

[1] *Vilnius Gediminas Technical University*

Saulėtekio al. 11, LT-10223 Vilnius, Lithuania

E-mail: rc@fm.vgtu.lt; mm@fm.vgtu.lt; vs@sc.vgtu.lt

[2] *Vilnius University, Kaunas Faculty of Humanities*

Muitinės g. 8, LT-44280 Kaunas, Lithuania

E-mail: r.ciegis@evf.vdu.lt

Received September 1, 2008; published online December 1, 2008

Abstract. In this paper we propose new heuristic numerical algorithm for determination of the optimal wires diameters in electrical cables. Two multilevel parallel versions of the optimization algorithm are constructed. The first algorithm is based on master-slave technique and the second algorithm uses the data-parallel strategy. Multilevel structure of the algorithms gives a possibility to adapt them to parallel architecture, for example, cluster of multicore computers. Some results of numerical experiments are presented which agree well with theoretical analysis.

Key words: Parallel algorithms, optimization, finite volume method, electrical cables, mathematical modelling.

1 Introduction

In modern cars electrical and electronic equipment is of great importance. One of the main tasks for engineers is to determine optimal conductor cross-sections in bundles of electric cables in order to minimize the total weight of cables. A quantitative description of the thermo-electrical characteristics in the electrical cables can be obtained by using mathematical models based on parabolic partial differential equations. In our previous papers (see, [5, 11]) we have proposed efficient parallel numerical algorithms for simulation of temperature distribution in electrical cables and have solved an inverse problem for fitting the diffusion coefficient of the air-isolation material mixture to the experimental data. In this paper we solve the optimization problem and minimize the

total weight of cables.

The rest of the paper is organized as follows. In Section 2, the global optimization problem is formulated. The goal is to minimize the total mass of the metal in all wires under the constraint that the temperatures of the wires remain bounded by some given maximal value. Two different strategies of wires distribution, i.e. pessimistic and average, are considered. In the following subsections, first, the mathematical model of the temperature distribution in the bundle of electrical wires is described. Then, following [5], the Finite Volume (FV) discretization and sequential numerical solution algorithm are shortly presented. Next, the parallel version of the algorithm (proposed in [11]) is described. The parallel solution algorithm is based on the domain decomposition. Here we add the theoretical scalability analysis and study the performance of the algorithm on the hybrid parallel system used for computational experiments in this paper - cluster of multicore nodes. In Section 3 the optimization problem is solved. Since the formulated optimization problem is NP-hard [8] we restrict to heuristic algorithm which is based on a greedy type search method [1]. Parallel multilevel algorithms are developed using the master-slave and data decomposition paradigms. In the following subsections results of computational experiments are presented and the efficiency of the proposed parallel algorithms is investigated. The last example of this section gives the optimization results (distribution of wires diameters) for the bundle of electrical cables consisting of 32 wires. Some final conclusions are given in Section 4.

2 Problem Formulation

Let $G = \{d_k, \quad k = 1, \ldots, K\}$ be a set of feasible diameters of wires and $\mathcal{D} = (d_{k_1}, \ldots, d_{k_M})$, $d_{k_m} \in G$, $m = 1, \ldots, M$ - a set of diameters in a bundle from M cables. The critical scenarios of loads of electrical cables are defined by

$$S_j = \{\, I_j^m, \;\; m = 1, \ldots, M \,\}, \quad j = 1, \ldots, J,$$

where I_j^m defines the current applied to the m-th wire in the j-th scenario. These scenarios are selected from a set of regimes which are important for the design of a specific car.

The objective function W defines the total mass of metal in all wires. It is defined by the following formula

$$W(d_{k_1}, \ldots, d_{k_M}) = \frac{\rho\pi}{4} \sum_{m=1}^{M} d_{k_m}^2,$$

where ρ is the density of the metal.

Our aim is to solve the following optimization problem

$$\min_{d_{k_m} \in G,\ s.t.\ T \le T_{Max}} W(d_{k_1}, \ldots, d_{k_M}) = W(d_{k_1}^0, \ldots, d_{k_M}^0). \tag{2.1}$$

Here T_{Max} is the given critical (e.g., melting) temperature, T is the maximal

temperature of electrical wires with respect to all load scenarios

$$T = \max_{1\le j\le J} \max_{1\le m\le M} U_m(S_j),$$

and $U_m(S_j)$ denotes the simulated temperature of the m-th wire when S_j load scenario is used.

Temperatures $U_m(S_j)$ depend also on the packing of cables inside the bundle. This packing is practically not controlled in real bundles, therefore, in our optimization problem we consider two packing strategies to estimate the worst and average cases: *pessimistic* and *average*. With *pessimistic* strategy, for each scenario S_j, we define the distribution of wires $\Omega_P^j = \Omega_P(S_j)$ using the greedy packing heuristic. First, all wires are sorted according to their densities of loads

$$\eta^m = \left(\frac{4I_j^m}{\pi d_{k_m}^2}\right)^2, \quad m = 1,\dots,M.$$

Then the greedy packing algorithm takes wires with the highest density one by one and puts them to the center of a bundle of electrical cables as close as possible. Due to the maximum principle, we expect that such a distribution of cables leads to temperatures of wires close to maximum. For a given distribution of wires Ω_P^j, we find a discrete stationary solution

$$U(\Omega_P^j) = \{U_{ik},\ \ X_{ik} \in D_h(\Omega_p^j)\,\}$$

of the nonlinear heat conduction problem and define $U_m(S_j)$, $m = 1,\dots,M$. Here $D_h(\Omega_p^j)$ is a discrete grid defined on the domain of a cable bundle $D(\Omega_P^j)$. A short description of the mathematical model and the discrete finite-volume scheme for numerical solution of this problem will be presented in the next subsection.

With the *average* strategy, for each scenario S_j, we define a set Ω^j of random distributions of wires $\Omega^j = \{\Omega_l^j = \Omega_l(S_j),\ l = 1,\dots,L\}$ and compute $U_m(S_j)$ as an average of temperatures for different distributions of wires

$$U_m(S_j) = \frac{1}{L}\sum_{l=1}^{L} U_m(\Omega_l^j), \quad m = 1,\dots,M. \tag{2.2}$$

Thus, in order to compute a value of objective function W, we generate J or JL different distributions of wires and for each distribution we solve the given stationary heat conduction problem.

2.1 Mathematical model

For a given bundle of electrical cables we consider a domain $D \times (0, t_F]$, where $D = \{X = (x_1, x_2) :\ x_1^2 + x_2^2 \le R^2\}$, R is the outer radius of the bundle. Let $T(X,t)$ describe a distribution of the temperature in electrical cables. The mathematical model consists of the parabolic differential equation [5, 10]:

$$\rho(X)c(X,T)\frac{\partial T}{\partial t} = \sum_{i=1}^{2}\frac{\partial}{\partial x_i}\Big(k(X)\frac{\partial T}{\partial x_i}\Big) + f(X,T),\ \ (X,t) \in D \times (0,t_F], \tag{2.3}$$

subject to the initial and boundary conditions

$$T(X,0) = T_a, \quad X \in \bar{D} = D \cup \partial D, \tag{2.4}$$

$$k(X,T)\frac{\partial T}{\partial \eta} + \alpha_k(T)\big(T(X,t) - T_a\big) + \varepsilon\sigma\big(T^4 - T_a^4\big) = 0, \quad X \in \partial D. \tag{2.5}$$

The continuity conditions

$$[T(x,t)] = 0, \quad \Big[k\frac{\partial T}{\partial x_i}\Big] = 0$$

are specified at the points of discontinuity of coefficients.

In the model, $c(X,T)$ is the specific heat capacity, $\rho(X)$ is the density, $k(X)$ is the heat conductivity coefficient. The density of the energy source $f(X,T)$ is defined as $f = \big(I/A\big)^2\rho_0\big(1 + \alpha_\rho(T-20)\big)$, here I is the density of the current, A is a area of the cross-section of the cable, ρ_0 is the specific resistivity of the conductor, T_a is the temperature of the environment.

2.2 Finite volume discrete scheme

The computational grid $\bar{D}_h = D_h \cup \partial D_h$ is obtained by constructing the auxiliary grid $\widetilde{D}_h = \Omega_h \cap \bar{D}$, which is defined as intersection of the equidistant rectangular grid Ω_h with the computational domain $\bar{D}$ and deleting from it those nodes X_{ij}, for which both neighbours in some direction does not belong to $\widetilde{D}_h$. In $\bar{D}_h$ we define discrete functions $U_{ij}^n = U(x_{1i}, x_{2j}, t^n)$, $X_{ij} \in \bar{D}_h$, here $t^n = n\tau$ and τ is the discrete time step.

Integrating the differential equation over the control volume and approximating the obtained integrals with an individual quadrature for each term, the differential problem is discretized by the conservative scheme

$$S_{ij}\rho_{ij}c_{ij}(U_{ij}^n)\frac{U_{ij}^n - U_{ij}^{n-1}}{\tau} = \sum_{k=0}^{3}\delta_{ijk}J_{ijk}(U_{ij}^n)U_{ij}^n + S_{ij}f_{ij}(U_{ij}^n), \quad X_{ij} \in \bar{D}_h,$$

where $J_{ijk}^n(U_{ij}^n)U_{ij}^n$ is the heat flux through a surface of the control volume $e_{ijk} = e_k(X_{ij})$. More details on this approximation are given in [5].

The derived finite difference scheme defines a system of nonlinear equations. By using the predictor–corrector method it is approximated by the linear finite-difference scheme of the same order of accuracy. The obtained systems of linear equations are solved by using the BiCGSTAB iterative method with the Gauss-Seidel type preconditioner [15]. We note that solution of such systems is a challenging task, since the diffusion coefficient $k(X)$ is a discontinuous function and the jump of the coefficient is large. In such a situation the discrete values of the temperature in the metal regions start to be non-sensitive to outside changes of the temperature and iterations converge slowly. Extensive analysis of many iterative algorithms for solution of such problems is given in [2, 7, 13]. The discontinuous Galerkin method is used to approximate the PDE in [7], while in [2] the conservative finite difference scheme is proposed. Efficient solvers of such problems can be obtained by using multigrid algorithms, or multigrid solvers

can be used as efficient preconditioners for Krylov type iterative algorithms. Fore example, a multilevel additive Schwartz preconditioner is proposed in [7] and in [16] an interface preserving coarsening multigrid algorithm is proposed for elliptic problems with highly discontinuous coefficients.

In this work, we have investigated the dependence of the convergence rate of BiCGSTAB iterative method on the space and time discrete steps, which define the stiffness of the matrix, and on the values of the discontinuous coefficient $k(X)$. As a test problem we have computed a solution of the linearized discrete scheme at one time step. In the isolator and the air-isolator mixture regions the values of the diffusion coefficient were fixed to 0.17 and 0.13, respectively. The problem was solved for different values of $k(X) = k_0$ in the metal region, where $k_0 = 401$ corresponds to the copper. The outside radius of the electrical cable is fixed to $R = 0.004$ and the width of the isolation layer is equal to 0.00015. The domain is covered with the auxiliary grid of dimension $N \times N$.

In Table 1 we present the numbers of iterations required to reduce the initial error in the discrete maximum norm by factor 10^{-4}. First, the time step h_t is taken very large, corresponding to the solution of a stationary problem.

Table 1. Numbers of BiCGSTAB iterations for different values of N and k_0.

	$N = 100$	$N = 200$	$N = 400$	$N = 800$
$k_0 = 0.401$	70	112	209	385
$k_0 = 4.01$	75	137	270	457
$k_0 = 40.1$	80	157	299	600
$k_0 = 401$	99	183	371	714

When time step h_t is taken smaller, the convergence rate of the iterative algorithm is increased. In Table 2 the numbers of BiCGSTAB iterations are presented for $k_0 = 401$, $N = 200$, 400 and different values of h_t.

Table 2. Numbers of BiCGSTAB iterations for different values of time step h_t.

	$h_t = 100000$	$h_t = 100$	$h_t = 10$	$h_t = 1$
$N = 200$	183	154	191	64
$N = 400$	371	305	180	118

2.3 Parallel version of the discrete algorithm

In this subsection we give a short description of the parallel algorithm to solve the discrete problem obtained after approximation of the mathematical model by the finite volume scheme. More details can be find in our paper [11].

The parallel algorithm is based on the domain decomposition method. The discrete grid $\tilde{D}_h$ is distributed among p processors.

The sequential algorithm consists of two parts. First, the discretization operators are applied and the coefficients of the system of linear equations are computed for predictor and corrector steps of the linearization algorithm. Second, the obtained systems of linear equations are solved by the BiCGSTAB iterative method. In order to get a scalable parallel algorithm we implement both steps, i.e. the discretization and linear algebra steps, in parallel. It is well known that for such type of problems the main part of CPU time is spent in solving the systems of linear equations (see, e.g. [6, 14]).

The convergence rate of iterative methods depends essentially on the quality of the preconditioner. The Gauss-Seidel algorithm is sequential in its nature. For the parallel BiCGSTAB algorithm we use a Jacobi version of the Gauss-Seidel preconditioner. Such an implementation depends only on the local part of the matrix and no data communication is required. The convergence rate of the modified parallel BiCGSTAB solver can be worse than the convergence rate of the sequential iterative algorithm. Thus, in general, the complexity of the full parallel algorithm does not coincide with the complexity of the original sequential algorithm.

At the domain distribution step we solve the load balancing problem. First, since the computational complexity of the algorithm is proportional to the maximum number of grid points in local sub-domains, the aim is to divide the grid into subdomains with equal number of points. Second, due to the stencil of discretization, the computational domains of processes overlap. The information belonging to the overlapped regions should be exchanged between processes. This is done by the additional halo layers of so called ghost-cells. The time costs of such data exchanges are contributing to the additional costs of the parallel algorithm. Thus, a second goal of defining the optimal data mapping is to minimize the overlapping regions. It is well known that such global minimization problems belong to the class of NP-hard (*nondeterministic polynomial-time hard*) problems and thus we must restrict ourself to some heuristic algorithms. For data distribution we use the multilevel partitioning method from METIS software library [12]. This algorithm produces partitions of high quality and it has a linear algorithmic complexity.

Next, we estimate the complexity of the parallel algorithm. The matrices and right-hand side vectors are assembled element by element. This is done locally by each process. The time required to calculate all coefficients of the discrete problem is given by $W_{p,coeff} = c_1 n/p$, here n is the number of elements in the grid.

All ghost values of the vectors belonging to overlapping regions are exchanged between processes. The data communication can be done in parallel between different pairs of processes. Thus, we estimate the costs of data exchange operation as $W_{exch} = \alpha + \beta m$, where m is the number of items sent between two processes, α is the message startup time and β is the time required to send one element of data.

The sequential BiCGSTAB algorithm is modified to the parallel version in a way such that its convergence properties are not changed during the parallelization process. The only exception is due to the block Jacobi implementation of the Gauss-Seidel preconditioner $\mathbf{B}$. Then each process computes $\mathbf{B}^{-1}$ by using

only a local part of matrix **A**. Four different operations of the BiCGSTAB algorithm require different data communications between processes. Computation of vector *saxpy* (i.e. $\vec{y} = \alpha\vec{x} + \vec{y}$) operation is done in parallel and only local parts of vectors are used, the complexity of this operation is estimated by $W_{p,saxpy} = c_2 n/p$. Before computation of matrix-vector multiplication, vectors are updating information at ghost points, then all computations are done in parallel, thus the complexity of matrix-vector multiplication is given by

$$W_{p,mv} = \frac{c_3 n}{p} + 2(\alpha + \beta m(p)).$$

The computation of inner product or norm requires $W_{p,dot} = c_4 n/p + R(p)(\alpha + \beta)$. Here $R(p)$ defines the complexity of the global reduction operation and it depends on the given computer network. It is implemented by using MPI function `Allreduce`.

The computation of the preconditioner **B** and application of this preconditioner is done locally by each process without any communication operation, the cost of this step is given by $W_{p,D} = c_5 n/p$. Summing up all the estimates, we obtain the theoretical model of the complexity of the parallel algorithm:

$$W_p = K\big(c_6\frac{n}{p} + c_7(\alpha + \beta m(p))\big) \tag{2.6}$$
$$+ N(p)\big(c_8\frac{n}{p} + c_9 R(p)(\alpha + \beta) + c_{10}(\alpha + \beta m(p))\big),$$

where K is the number of time steps, and $N = N(p)$ is a total number of BiCGSTAB iterations. As was explained above N depends on p. The presented theoretical complexity model enables us to make a scalability analysis, similar to one presented in [6].

At the end of this subsection we present some results of computational experiments. Computations were performed on Vilkas cluster of computers at Vilnius Gediminas Technical University, consisting of nodes with Intel(R) Core(TM)2 Quad processor Q6600. Four processing cores are running at 2.4 GHz each and sharing 8 MB of L2 cache and a 1066 MHz Front Side Bus. Each of the four cores can complete up to four full instructions simultaneously.

The discrete problem was solved on 500×500 and 750×750 reference grids. In Table 3 for each number of nodes and cores in the node - $n_d \times n_c$ the values of the algorithmic speedup coefficient $S_p = T_1/T_p$ and efficiency $E_p = S_p/p$ are presented (here the total number of processes $p = n_d n_c$).

The presented results give us a possibility to make the scaling analysis the efficiency of the parallel algorithm. In the case when only one core per node is used, the efficiency is improved, when the size of the problem is increased. It can be recommended to use till two cores per one node in practical simulations. In the case of four cores per node data reading/writing operations start to be a bottle-neck of the parallel BiCGSTAB algorithm, since the discrete scheme defines very sparse matrix.

Table 3. Results of computational experiments on Vilkas cluster for different numbers and configurations of processes ($n_d \times n_c$).

	(2×1)	(1×2)	(4×1)	(2×2)	(1×4)	(8×1)	(8×2)
$S_p(500)$	1.96	1.60	3.68	3.43	1.78	5.45	6.81
$E_p(500)$	0.98	0.80	0.92	0.84	0.45	0.69	0.43
$S_p(750)$	1.98	1.50	3.94	3.27	1.65	7.25	11.49
$E_p(750)$	0.99	0.75	0.98	0.82	0.41	0.91	0.72

3 Optimization Algorithm

In general, optimization problem (2.1) is a NP-hard problem and exhaustive combinatorial search procedure is not realistic even for a small number of wires (see, [1, 8]). In order to find an approximate solution we propose the following heuristic, which is based on a greedy search algorithm (see, Fig. 1).

OptimalSetOfWires ()
begin
(1) $\mathcal{D}$ = SelectInitialSetOfWires();
(2) **while** (T $\le T_{Max}$) **do**
(3) SaveOptimalSetOfWires($\mathcal{D}$);
(4) oK = FindLighterSetOfWires($\mathcal{D}$, G);
(5) **if** (oK == 1) **do**
(6) **for** (j=0; j<J; j++) **do**
(7) Ω_j = GenerateSetOfDistributions(S_j, $\mathcal{D}$);
(8) T_j = ComputeTemperature(Ω_j);
end for do
(9) $T = \max_{0 \le j < J} T_j$;
else
(10) T = 2 T_{Max};
end if do
end while do
end OptimalSetOfWires

Figure 1. Algorithm for solving optimization problem (2.1) .

Using notations given in the definition of problem (2.1) we put

$$T_j = \max_{1<m<M} U_m(S_j), \quad j = 1, \ldots, J.$$

As was stated above, the pessimistic distributions of wires are generated by the greedy packing algorithm, which again is only an heuristic and it can not guarantee that we find the largest possible temperature T_j. Due to discrete version of the packing algorithm and due to the nonlinearity of the mathematical model it is possible to obtain situations, when the temperature T_j decreases

after testing a lighter set of wires. Therefore we propose a modification of the algorithm, when temperature of the wires is computed additionally on the grid Ω_j generated for the old optimal pessimistic distribution and the radius of one wire is changed according the selection step. Then T_j is equal to the largest value for two investigated distributions.

3.1 Parallel optimization algorithm

The proposed above optimization algorithm has at least three parallelization levels.

1. First, in step (6) all subproblems for different load scenarios are independent and can be solved in parallel.

2. The second parallelization level is obtained considering the solution of each of J subproblems. In the case of the pessimistic strategy in order to simulate the temperatures of cables two independent distributions of wires are generated. In the case of the average strategy (2.2) the number of generated independent distributions of wires is equal to L.

3. At the last parallelization level the temperature inside a bundle of electrical cables can be simulated by using the parallel version of the discrete algorithm given in Section 2.

Here we consider two parallel versions of the given optimization algorithm. The first one uses a very popular general approach to construct parallel algorithms: the master – slave paradigm. It is applied in heterogeneous environments of varying computational complexity of subproblems and/or non-uniform and non-dedicated parallel architecture (see, [3, 4, 9]). This approach introduces very naturally the dynamical load-balancing techniques.

For the given optimization algorithm we use the following master – slave template.

- The master constructs a pool of independent problems at Level 2 of the parallel optimization algorithm and sends these problems to slaves.

- Groups of slaves solve the obtained temperature simulation problems by using the parallel algorithm from Level 3.

The second version is based on the data distribution paradigm. Again, we construct a multilevel parallel algorithm. First, according to the scheme given above, all processes are divided into J groups. Each group of processes solves the heat conduction problem for one of the scenarios. In computational experiments, we have applied this strategy for a pessimistic variant of the wires distribution, therefore our parallel tool is restricted to the two level algorithm (i.e. only Levels 1 and 3 are implemented).

3.2 Results of computational experiments

As the first example, we consider the computation of optimal wire distribution in the case of average strategy, when the temperature of wires is averaged in J experiments. The master-slave tool is used to solve this optimization problem. In Table 4, the CPU time, the speedup coefficient S_p and the efficiency coefficient E_p of the parallel optimization algorithm are presented. The parameters of the problem are chosen as $J = 4$, $L = 9$ and the reference grid 250×250 is used for the approximation.

Table 4. Results of computational experiments using the master–slave tool.

	(2×1)	(1×4)	(5×1)	(2×4)	(5×2)	(10×1)	(10×2)
T_p	10717	11060	5353	6358	2627	2383	1250
S_p	2.0	1.94	4.0	3.37	8.15	8.98	17.14
E_p	0.666	0.484	0.80	0.42	0.82	0.90	0.86

It follows from the presented results, that the master-slave algorithm distributes the work in a load-balanced way. Since all subproblems are of very similar computational complexity the speedup is equal to the number of slaves when each slave gets the same number of jobs. The second conclusion is that only up to two cores of one node can be used efficiently for this type of solver (note, that the main part of CPU time is spend by BiCGSTAB solver and the matrix of the system is very sparse).

As the second example, we consider the computation of optimal wire distribution in the case of pessimistic strategy. The multilevel parallel algorithm based on the data decomposition method is used to solve this optimization problem. In Table 5, the CPU time, the speedup coefficient S_p and the efficiency coefficient E_p of the parallel optimization algorithm are presented. The finite volume scheme is constructed for the reference grid 250×250. The number of scenarios is $J = 4$, therefore all processes are divided into two or four groups, depending on the total number of processes. In Table 5 we define by $(n_d \times n_c)$ the number of nodes and cores per node used in computations.

Table 5. Results of computational experiments using the data-parallel optimization algorithm.

	(4×1)	(4×2)	(8×1)	(12×1)	(8×2)	(12×2)
T_p	900	506	579	557	393	371
S_p	4.0	7.11	6.21	6.46	9.16	9.7
E_p	1.0	0.89	0.78	0.54	0.57	0.41

In the last experiment we have optimized the diameters of the wires in a bundle of 32 electrical wires. The set of feasible diameters G has $K = 9$ different elements. We have restricted the search algorithm to 13 selection steps and

the temperature of the wires was simulated for 360 seconds. The pessimistic strategy was used to compute the maximum value of the temperature for a given set of wires. The total mass of the initial set of wires was 7.363 and the maximum temperature obtained in simulations was 99.1 degrees. After 13 steps of optimization the mass of wires was reduced to 6.186, while the temperature increased till 103.51 degrees.

4 Conclusions

In this work we have solved the optimization problem for determination of optimal diameters of wires in bundles of electrical cables. Several heuristics are proposed: greedy type search method and two packing strategies. Two parallel versions of optimization algorithm were developed: master-slave and data-parallel. In combination with parallel algorithm for the solution of heat conduction problem, we have obtained multilevel parallel algorithms, which can be adapted to different parallel computer architectures.

Acknowledgement

This work was partially supported by the Agency for International Science and Technology Development Programmes in Lithuania within the EUREKA Project E!3691 OPTCABLES and the Lithuanian State Science and Studies Foundation within the project on B-03/2008 "Global optimization of complex systems using high performance computing and GRID technologies".

References

[1] E. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization*. Princeton University Press, 2003.

[2] T. Aleinikova and R. Čiegis. On the iterative methods for nonlinear problems with discontinuous coefficients. *Informatica*, **4**(1):126–139, 1993.

[3] L. Baldo, L. Brenner, L. Gustavo Fernandes, P. Fernandes and A. Sales. Performance models for Master/Slave parallel programs. *Electronical Notes in Theoretical Computer Science*, **128**:101–121, 2005.

[4] M. Baravykaitė, R. Belevičius and R. Čiegis. One application of the parallelization tool of Master–Slave algorithms. *Informatica*, **13**:393–404, 2002.

[5] R. Čiegis, A. Ilgevičius, H. Liess, M. Meilūnas and O. Suboč. Numerical simulation of the heat conduction in electrical cables. *Math. Model. and Anal.*, **12**(4):425–439, 2007.

[6] R. Čiegis, O. Iliev and Z. Lakdawala. On parallel numerical algorithms for simulating industrial filtration problems. *Computational Methods in Applied Mathematics*, **7**(2), 2007.

[7] M. Dryja. On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients. *Comp. Methods in Applied Math.*, **3**(1):76–85, 2003.

[8] M. R. Garey and D. S. Johnson. *A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.

[9] A. Grama, G. Karypis, Vipin Kumar and A. Gupta. *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Addison Wesley, 2003.

[10] F. Incropera, P. DeWitt and P. David. *Introduction to Heat Transfer*. John Willey & Sons, New York, 1985.

[11] G. Jankevičiūtė and R. Čiegis. Parallel numerical solver for the simulation of the heat conduction in electrical cables. In R. Čiegis, D. Henty, B. Kagstrōm and J. Žilinskas(Eds.), *Parallel Linear Algebra and Optimization: Advances and Applications. Springer Optimization and Its Applications. ISBN: 978-0-387-09706-0*, volume 27, pp. 211–216, New-York, 2009. Springer.

[12] G. Karypis and V. Kumar. Parallel multilevel k-way partitioning scheme for irregular graphs. *SIAM Review*, **41**(2):278–300, 1999.

[13] A. A. Samarskii. *The Theory of Difference Schemes*. Marcel Dekker, Inc., New York–Basel, 2001.

[14] V. Starikovičius, R. Čiegis, O. Iliev and Z. Lakdawala. A parallel solver for the 3D simulation of flows through oil filters. In R. Čiegis, D. Henty, B. Kagstrōm and J. Žilinskas(Eds.), *Parallel Linear Algebra and Optimization: Advances and Applications. Springer Optimization and Its Applications. ISBN: 978-0-387-09706-0*, volume 27, pp. 185–196, New-York, 2009. Springer.

[15] H. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, **13**(3):631–644, 1992.

[16] W. Wan. Interface preserving coarsening multigrid for elliptic problems with highly discontinuous coefficients. *Numerical Linear Algebra with Appl.*, **7**(8):727–742, 2000.