ВЕСТНИК

ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2019
Т. 8, № 2

ISSN 2305-9052

СЕРИЯ

«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

Решением ВАК включен в Перечень научных изданий, в которых должны быть опубликованы результаты диссертаций на соискание ученых степеней кандидата и доктора наук

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет (национальный исследовательский университет)»

Тематика журнала:

- Вычислительная математика и численные методы
- Математическое программирование
- Распознавание образов
- Вычислительные методы линейной алгебры
- Решение обратных и некорректно поставленных задач
- Доказательные вычисления
- Численное решение дифференциальных и интегральных уравнений
- Исследование операций
- Теория игр
- Теория аппроксимации
- Информатика
- Искусственный интеллект и машинное обучение
- Системное программирование
- Перспективные многопроцессорные архитектуры
- Облачные вычисления
- Технология программирования
- Машинная графика
- Интернет-технологии
- Системы электронного обучения
- Технологии обработки баз данных и знаний
- Интеллектуальный анализ данных

BULLETIN
OF THE SOUTH URAL STATE UNIVERSITY
SERIES "COMPUTATIONAL MATHEMATICS AND SOFTWARE ENGINEERING"

2019
vol. 8, no. 2

ISSN 2305-9052

Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta.
Seriya "Vychislitel'naya Matematika i Informatika"

South Ural State University

The scope of the journal:

- Numerical analysis and methods
- Mathematical optimization
- Pattern recognition
- Numerical methods of linear algebra
- Reverse and ill-posed problems solution
- Computer-assisted proofs
- Numerical solutions of differential and integral equations
- Operations research
- Game theory
- Approximation theory
- Computer science
- Artificial intelligence and machine learning
- System software
- Advanced multiprocessor architectures
- Cloud computing
- Software engineering
- Computer graphics
- Internet technologies
- E-learning
- Database processing
- Data mining

Содержание

Contents

DOI: 10.14529/cmse190201

THE USE OF THE LINE-BY-LINE RECURRENT METHOD FOR SOLVING SYSTEMS OF DIFFERENCE ELLIPTIC EQUATIONS WITH NINE-DIAGONAL MATRICES

© 2020 A.A. Fomin[1], L.N. Fomina[2]

[1] *T.F. Gorbachev Kuzbass State Technical University*
(Vesennaya 28, Kemerovo, 650000 Russia),
[2] *Kemerovo State University*
(Krasnaya 6, Kemerovo, 650043 Russia)
E-mail: fomin_ aa@mail.ru, lubafomina@mail.ru
Received: 21.11.2018

The applying of the line-by-line recurrent method for solving systems of difference elliptic equations with nine-diagonal matrices is the subject of the article. Such matrices take place in the case of difference approximation of 2D differential problems of a higher order of accuracy on a regular grid covering the area under consideration. The technology of the so-called compensatory transform which allows replacing the initial nine-diagonal matrix of the system with the five-diagonal one is offered in the article, due to the fact that originally the line-by-line recurrent method was designed for solving systems of difference equations with a five-diagonal matrix. The efficiency of this technology is analyzed by comparing the solutions of the test boundary value problem in a unit square. The solutions are found both with the help of different implementations of the compensatory transform technology and by other modern highly efficient iterative methods for solving the systems of difference equations. The problem is solved on the sequence of grids from coarse (501×501) to fine (4001×4001) nodes. The accuracy of the solution convergence is determined by the relative norm of the residual, which is equal to 10^{-12} in the present work. It is shown that the line-by-line recurrent method retains its high efficiency over the entire range of the grids under consideration despite the use of the intermediate technology of the compensatory transform.

Keywords: grid method, system of difference elliptic equations, iterative method, convergence of solution.

FOR CITATION

Fomin A.A., Fomina L.N. The Use of Line-by-Line Recurrent Method for Solving Systems of Difference Elliptic Equations with Nine-Diagonal Matrices. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2020. vol. 8, no. 2. pp. 5–21. DOI: 10.14529/cmse190201.

Introduction

As is well known, many "standard" finite-difference approximation technologies for two-dimensional differential equations of problems of fluid dynamics and heat transfer on regular grids are based on five-point stencil. The Patankar scheme [1] can be mentioned here as an appropriate example of this kind of technologies. This technology yields an algorithm which strictly provides monotonicity and central-point dominance of the finite-difference five-point scheme. And as a result, a system of linear algebraic equations (SLAE) with a five-diagonal matrix of positive type arises [2]. The Patankar scheme has shown itself well in numerous studies. However, the need for higher-order accuracy schemes arises in a number of cases. For example, the five-point central-difference scheme provides a smaller error than the Patankar scheme, despite the fact that these schemes are formally of the same order of approximation. Unfortunately, the SLAE matrix obtained on the basis of the central-difference scheme can lose its positive type in certain conditions.

One can use the nine-point stencil (five points along each direction) to avoid this difficulty and increase the approximation order at the same time. In this connection a variety of higher-order

numerically stable schemes was developed using the nine-point stencil [3–8]. As a rule, the convective flux discretization is the focus of a higher-order difference approximation, as it was done in the previously cited works. While, the diffusion part of the transport equations is usually approximated by the standard five-point scheme of the second order accuracy. The works that use a higher-order difference approximation of diffusion flux are found much less often [9]. Difference approximation of differential equations with derivatives of higher order (up to the fourth one) is another origin of difference linear equations on five points along each independent coordinate [10]. In this case, it is evident that for 2D problems the difference stencil will consist of nine nodes.

It is remarkable that the absence or cursory mention of an iterative method for solving arising nine-diagonal SLAE is a common feature of the above works. Probably the researchers believe that the expansion of existing methods for solving systems with five-diagonal matrices to methods for solving systems with nine-diagonal ones is not an insuperable barrier. Indeed, the transition of such well-known methods as block successive over-relaxation (BSOR) or line-by-line method [1] from five-point to nine-point versions do not cause much difficulty. But on the other side, there are effective computational technologies for the SLAE solution, which don't have an easy expansion up to nine-point stencils (see, for example, [11, 12]).

A deferred correction procedure was developed to overcome this problem. Its idea is that a nine-diagonal matrix of the system of equations is replaced by a five-diagonal one which is obtained with the use of a difference approximation of lower order in a special way. After that one can use any available method for solution of SLAE with a five-diagonal matrix. Interestingly, the procedure occurs in two variants in the literature. The first variant is used when there is an independent difference approximation of the convective and diffusive terms of the transport equation [6]. So, it operates with convective terms of the equation on the stage of their difference approximation. The second variant of the procedure is applied to an already formed algebraic difference equation, regardless of how it was obtained and, therefore, it is more universal [13]. The article will consider the second version of the deferred correction procedure. It should be noted that the procedure has a weak point, namely: if there is no difference approximation of some terms of the equation of lower order on a reduced number of nodes, then it is not applicable. For example, such situation takes place when there are fourth-order derivatives in a differential equation that cannot be approximated using less than five nodes. Therefore, in this case one have to design other procedure to replace a nine-diagonal matrix of SLAE with the five-diagonal one, which does not have this shortcoming.

Recently, a highly efficient line-by-line recurrent method was developed and successfully used for solving systems of difference elliptic equations in some problems of computational fluid dynamics and heat transfer [14]. This method is applicable to systems of equations with five-diagonal matrices due to its design features in the case of two-dimensional problems [15]. However, attempts to modify this method for the case of systems of equations with nine-diagonal matrices faced great difficulties. Therefore, the way out is to use the line-by-line recurrent method in cooperation with an intermediate technology of matrices transformation like the deferred correction procedure, for example. In the light of the foregoing, the objective of the work is to develop a universal technology of matrix transformation and to investigate the effectiveness of the line-by-line recurrent method in solving systems of difference elliptic equations with nine-diagonal matrices.

The paper is organized as follows. The mathematical statement of a problem, namely: definition area, differential transport equation, boundary conditions, closing formulas — are described in Section 1. In the following Section 2, the details of the numerical technique including high order numerical discretization, deferred correction procedure, compensatory transform technology are given. The comparisons of solutions of the test problem obtained by different methods are presented in Section 3. The conclusions are drawn in the final Section.

1. Statement of problem

Let $\Omega = \{(x, y) : 0 \le x \le 1, 0 \le y \le 1\}$ be a unit square in Cartesian coordinates (Fig. 1) as the definition area of unknown $\Phi(x, y)$ which is governed by a differential transport equation.

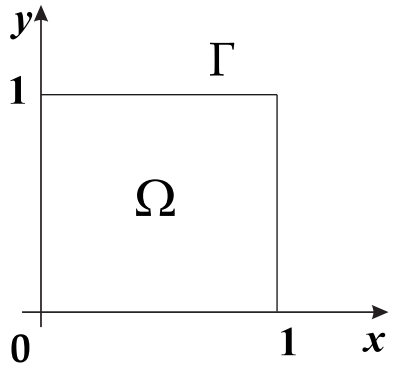


Fig. 1. Scheme of the problem area

In this case the formulation of a test 2D boundary value problem in Ω can be used as an origin for obtaining a system of difference elliptic equations with the sparse matrix with the help of difference approximation of the initial differential problem. The generalized steady-state convection-diffusion transport equation written for $\Phi(x, y)$ can be stated as [1]

$$U\frac{\partial \Phi}{\partial x} + V\frac{\partial \Phi}{\partial y} = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial \Phi}{\partial y}\right) - S, \tag{1}$$

where $U(x, y)$, $V(x, y)$ — flow velocity components, $\Gamma(x, y)$ — transfer coefficient, $S(x, y)$ — source. Dirichlet conditions take place on the area boundaries. Let velocity components and transfer coefficient be as follows:

$$U(x, y) = -3y^2 \arctan x, \quad V(x, y) = \frac{y^3}{1 + x^2}; \qquad \Gamma(x, y) = \exp(-l^2), \quad l^2 = x^2 + y^2.$$

It should be noted that the velocity field is solenoidal one. Lastly, let the solution of the test problem be the function $u(x, y) = \exp(-10l^2)\cos(8\pi l^2)$, then the substitution of u, U, V, and Γ in equation (1) makes it possible to define the expression for the source $S(x, y)$. It is not difficult to see that Dirichlet conditions at the area boundaries in the case of the $u(x, y)$ are written as:

$$\begin{aligned} &0 \le y \le 1: \quad \Phi(0, y) = \exp(-10y^2)\cos(8\pi y^2), \quad \Phi(1, y) = \exp(-10(1 + y^2))\cos(8\pi(1 + y^2)); \\ &0 \le x \le 1: \quad \Phi(x, 0) = \exp(-10x^2)\cos(8\pi x^2), \quad \Phi(x, 1) = \exp(-10(1 + x^2))\cos(8\pi(1 + x^2)). \end{aligned}$$

So, the test 2D boundary value problem is defined and one can begin to solve it numerically.

2. Numerical technique

2.1. High-resolution numerical discretization

The problem area is covered by a uniform orthogonal mesh which nodes one can separate into three groups: internal, near-boundary, and boundary nodes. The nine-point stencil with an internal central node is presented in Fig. 2a. The so-called SMART scheme [7] is used for higher-order approximation of (1) in internal mesh nodes. Briefly, the technology of the scheme

obtaining is as follows [16]. Integrating the equation (1) over the control volume (Fig. 2a) and using the divergence theorem for a Cartesian coordinate system allows getting the following discrete equation:

$$J_e - J_w + J_n - J_s = Q, \tag{2}$$

where J_e, J_w, J_n, J_s represent the total fluxes of unknown Φ across faces e, w, n, s of the control volume, and Q is the volume integral of the source term S. Each of the surface fluxes J contains convective and diffusive contributions. It is expressed, for example, for the face e, as follows:

$$J_e = \left(U_e \Phi_e - \Gamma_e \frac{\Phi_E - \Phi_P}{\delta x_e} \right) \Delta y, \tag{3}$$

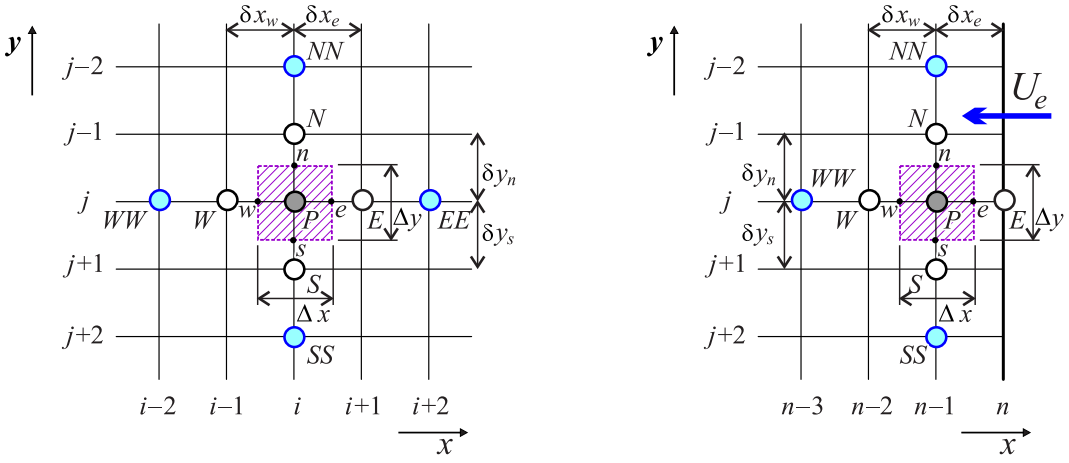
where

$$U_e = \frac{U_E + U_P}{2}, \quad \Gamma_e = 2\frac{\Gamma_E \Gamma_P}{\Gamma_E + \Gamma_P}, \quad \Phi_e = \begin{cases} \Phi_W + (\Phi_E - \Phi_W)\, f(\widetilde{\Phi}_P), & U_e \geq 0, \\ \Phi_{EE} + (\Phi_P - \Phi_{EE})\, f(\widetilde{\Phi}_E), & U_e < 0; \end{cases}$$

$$f(\widetilde{\Phi}) = \begin{cases} 3\widetilde{\Phi}, & 0 < \widetilde{\Phi} < 1/6, \\ 3/8 + 3/4\ \widetilde{\Phi}, & 1/6 \leq \widetilde{\Phi} \leq 5/6, \\ 1, & 5/6 < \widetilde{\Phi} < 1, \\ \widetilde{\Phi}, & \widetilde{\Phi} \text{ elsewhere.} \end{cases}$$

The tilde above unknown Φ denotes a so-called normalized variable which is defined as $\widetilde{\Phi} = (\Phi - \Phi_U)/(\Phi_D - \Phi_U)$. Here, subscripts U, D mean upflow and downflow nodes relative the central point, correspondingly. For example, for the face e the central node will be point P in the case $U_e \geq 0$ while conversely – point E if $U_e < 0$. So, $\widetilde{\Phi} = \widetilde{\Phi}_P = (\Phi_P - \Phi_W)/(\Phi_E - \Phi_W)$ for $U_e \geq 0$, and $\widetilde{\Phi} = \widetilde{\Phi}_E = (\Phi_E - \Phi_{EE})/(\Phi_W - \Phi_{EE})$ for $U_e < 0$.

The fluxes through the w, n and s faces can be found in a similar manner.



a) internal mesh node b) near-boundary mesh node

Fig. 2. Finite-difference stencil for higher-order discrete approximation

The eight-point stencil with a near-boundary central node is presented in Fig. 2b. For example, the right boundary of the problem area Ω is chosen. For this case the general approximation scheme described above can be applied if $U_e \geq 0$. In contrary, if $U_e < 0$ (as in the figure) it is necessary to follow special practices. In this context, it should be kept in mind

that a lower order approximation takes place as a result of the "windward rule" in any case. In literature the first order upwind scheme for near-boundary node is used as a rule [6]. But in the present work the more complex technology of Patankar scheme is adapted with a view to obtain a second order upwind difference scheme. For this case it is not difficult to get the approximation formula for Φ_e based on this methodology applying the profile of the fifth degree for unknown Φ, namely:

$$\Phi_e = \Phi_P + (\Phi_E - \Phi_P)\,\varphi(P_e), \tag{4}$$

where, in the context of the grid uniformity

$$\varphi(P_e) = \frac{\tilde{\Psi}(P_e)}{2\tilde{\Psi}(P_e/2)}, \qquad P_e = \frac{(\Gamma_E + \Gamma_E)(U_P + U_E)}{4\Gamma_P\Gamma_E}\delta x_e; \tag{5}$$

$$\tilde{\Psi}(P_e) = \begin{cases} -P_e, & P_e < -10, \\ (1 + 0,1P_e)^5 - P_e, & -10 \le P_e < 0, \\ (1 - 0,1P_e)^5, & 0 \le P_e \le 10, \\ 0, & P_e > 10. \end{cases} \tag{6}$$

Indeed, as is well known, the solution of the problem "convection and diffusion"

$$\frac{d}{dx}(\rho U\Phi) = \frac{d}{dx}\left(\Gamma\frac{d\Phi}{dx}\right)$$

for a domain $0 \le x \le L$ with boundary conditions: $\Phi = \Phi_0$ at $x = 0$, and $\Phi = \Phi_L$ at $x = L$ is

$$\Phi = \Phi_0 + (\Phi_L - \Phi_0)\frac{\exp(Px/L) - 1}{\exp(P) - 1} \tag{7}$$

on the assumption with Γ and ρU are constants [1]. Here P is a Peclet number defined by $P \equiv \rho UL/\Gamma$.

The value of Φ_e in (4) is calculated according to the solution profile (7), i. e. it is assumed that $\Phi_0 = \Phi_P$, $\Phi_L = \Phi_E$, $L = \delta x_e$, $x = \delta x_e/2$, $P = P_e$, and $\Phi = \Phi_e$ in the formula (7). So,

$$\Phi_e = \Phi_P + (\Phi_E - \Phi_P)\frac{\exp(P_e/2) - 1}{\exp(P_e) - 1} = \Phi_P + (\Phi_E - \Phi_P)\frac{1}{2}\frac{P_e}{\exp(P_e) - 1}\frac{\exp(P_e/2) - 1}{P_e/2}, \quad \text{or}$$

$$\Phi_e = \Phi_P + (\Phi_E - \Phi_P)\frac{\Psi(P_e)}{2\Psi(P_e/2)}, \tag{8}$$

where $\Psi(z) = z/(\exp(z) - 1)$. Because an exponential function is very expensive to compute, $\Psi(z)$ is approximated by Patankar's power-law scheme (see formulas (5.27) and (5.33) in [1]) which is represented by the complex formula (6) in the present work. In other words, $\Psi \approx \tilde{\Psi}$. As a result, it is easy to see that in this case the formulas (8) and (4) are almost identical taking into account the formula (5). What was required to show.

Finally, the trivial "approximation" takes place for the third group of the mesh (i. e. for the boundary lines of Ω) because of Dirichlet boundary conditions in the problem.

2.2. Deferred correction procedure

The deferred correction (DC) method is a simple and proven procedure that enables the use of high order approximation schemes in codes initially written for low order schemes. Let, in

general case, there be a difference scheme as a result of approximation of the original differential equation (1) on the nine-point stencil (Fig. 2a) of the following kind

$$a_P\Phi_P = a_E\Phi_E + a_W\Phi_W + a_N\Phi_N + a_S\Phi_S + a_{EE}\Phi_{EE} + a_{WW}\Phi_{WW} + a_{NN}\Phi_{NN} + a_{SS}\Phi_{SS} + b. \quad (9)$$

In turn, let the difference scheme of lower order approximation for the same equation (1) be as follows

$$a_P^L\Phi_P = a_E^L\Phi_E + a_W^L\Phi_W + a_N^L\Phi_N + a_S^L\Phi_S + b. \quad (10)$$

It is easy to see, adding to both sides of equation (9) the combination of $a_P^L\Phi_P - \sum\limits_{nb} a_{nb}^L\Phi_{nb}$, composed of the terms of equation (10), the DC procedure results in a five-point equation

$$a_P^L\Phi_P^{k+1} = \sum_{nb} a_{nb}^L\Phi_{nb}^{k+1} + \sum_{nb}(a_{nb}^L - a_{nb})\Phi_{nb}^k + \sum_{nnb} a_{nnb}\Phi_{nnb}^k + (a_P^L - a_P)\Phi_P^k + b, \quad (11)$$

where k is number of iteration, $nb = \{E, W, N, S\}, nnb = \{EE, WW, NN, SS\}$. It is clear that the solution of equation (11) tends to the solution of equation (9) with the convergence of the iterative process (i. e., with $\Phi^{k+1} \xrightarrow[k\to\infty]{} \Phi^k$). At the same time, one can use any previously created methods for solving SLAE with five-diagonal matrices to solve modified system on the base of equation (11).

Further in the article the DC procedure will be denoted as DSPt5, since the Patankar scheme with the profile of unknown Φ of the fifth degree is used to the lower order approximation.

2.3. Compensatory transform technology

As was mentioned above, the DC method is usable when a lower order approximation on a truncated five-point stencil takes place. Otherwise, one must apply other more general technology to transform a nine-diagonal matrix of SLAE to a five-diagonal one. Precisely that kind of a procedure, the so-called compensatory transform technology, is offered in the work. The major idea of the compensatory transform technology is to express the iterative increment of the sought-for solution in the "extreme" nodes of the stencil (in the Fig. 3 they are marked in cyan) through the increment in the "internal" nodes (white and black nodes of the stencil). So, the "extreme" nodes of the stencil are excluded and the matrix of the system of equations is transformed from nine-diagonal to five-diagonal one.

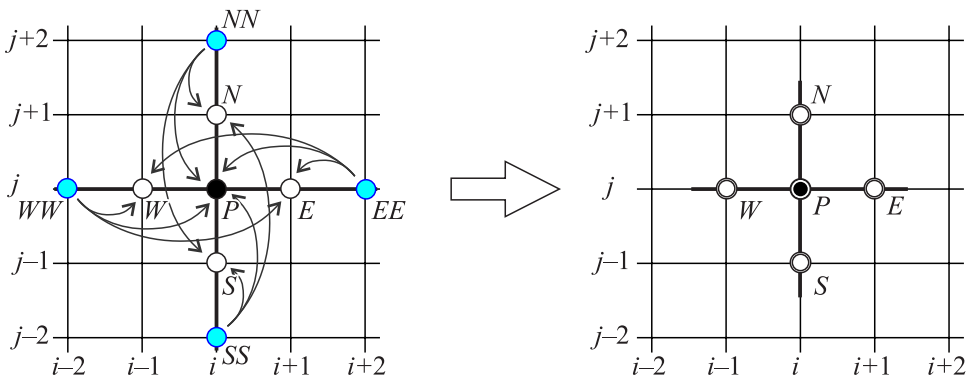


Fig. 3. The scheme of the compensatory transform of the nine-point stencil into five-point one

The transformation formula has the first or second order of accuracy depending on the number of the "internal" nodes of the difference stencil used in the expression. For example, in the case of uniform grid the formula of the first order accuracy for node EE is as follows

$$\Delta\Phi_{EE}^{k+1} = \theta\left(2\Delta\Phi_{E}^{k+1} - \Delta\Phi_{P}^{k+1}\right),$$

and the formula of the second order of accuracy for the same node is as follows

$$\Delta\Phi_{EE}^{k+1} = \theta\left[3\left(\Delta\Phi_{E}^{k+1} - \Delta\Phi_{P}^{k+1}\right) - \Delta\Phi_{W}^{k+1}\right].$$

Here $\Delta\Phi^{k+1} = \Phi^{k+1} - \Phi^{k}$ – is increment of the sought-for solution, θ is a parameter of compensation, which should be in the range $0 \le \theta \le 1$ [2]. It is easy to verify that the application of the above formulas will lead out to the following expressions for the transformed coefficient in the nearby point E

$$\bar{a}_E = a_E + 2\theta a_{EE},$$

$$\bar{a}_E = a_E + \theta\left(3a_{EE} + a_{WW}\right)$$

for the first and second order of accuracy respectively. Transformed coefficients for other nearby points W, N, S are written in a like manner. As a result, the transformed five-point difference equation is arrived as follows

$$\bar{a}_P\Phi_P^{k+1} = \sum_{nb}\bar{a}_{nb}\Phi_{nb}^{k+1} + \bar{b}, \tag{12}$$

where for the first order of accuracy

$$\bar{a}_P = a_P + \theta\left(a_{EE} + a_{WW} + a_{NN} + a_{SS}\right),$$
$$\bar{b} = b + a_{EE}\left[\Phi_{EE}^{k} - \theta\left(2\Phi_{E}^{k} - \Phi_{P}^{k}\right)\right] + a_{WW}\left[\Phi_{WW}^{k} - \theta\left(2\Phi_{W}^{k} - \Phi_{P}^{k}\right)\right] + \\ + a_{NN}\left[\Phi_{NN}^{k} - \theta\left(2\Phi_{N}^{k} - \Phi_{P}^{k}\right)\right] + a_{SS}\left[\Phi_{SS}^{k} - \theta\left(2\Phi_{S}^{k} - \Phi_{P}^{k}\right)\right],$$

and for the second order of accuracy, respectively

$$\bar{a}_P = a_P + 3\theta\left(a_{EE} + a_{WW} + a_{NN} + a_{SS}\right),$$
$$\bar{b} = b + a_{EE}\left\{\Phi_{EE}^{k} - \theta\left[3\left(\Phi_{E}^{k} - \Phi_{P}^{k}\right) + \Phi_{W}^{k}\right]\right\} + a_{WW}\left\{\Phi_{WW}^{k} - \theta\left[3\left(\Phi_{W}^{k} - \Phi_{P}^{k}\right) + \Phi_{E}^{k}\right]\right\} + \\ + a_{NN}\left\{\Phi_{NN}^{k} - \theta\left[3\left(\Phi_{N}^{k} - \Phi_{P}^{k}\right) + \Phi_{S}^{k}\right]\right\} + a_{SS}\left\{\Phi_{SS}^{k} - \theta\left[3\left(\Phi_{S}^{k} - \Phi_{P}^{k}\right) + \Phi_{N}^{k}\right]\right\}.$$

In the further, the compensatory transform technique of the first order of accuracy will be denoted as C1 and of the second order – as C2.

3. Computed results and discussion

3.1. Nomenclature of methods and the research strategy

In general, eight different methods are used to solve the problem formulated in the first section of the article. Nomenclature of methods (abbreviations and their expansions) is presented in Tab. 1. The solution of the problem is calculated with five uniform grids of different resolution: 501×501, 1001×1001, 2001×2001, 3001×3001, 4001×4001. Thus, the number of unknowns in generating SLAE varies from about 25×10^4 (coarse mesh) to 16×10^6 (fine mesh).

The research strategy is a comparative analysis of the characteristics of the convergence of the methods for solving SLAE (see Tab. 1) for each of the mesh partitions of the problem domain Ω.

Table 1

Iterative methods for SLAE solutions construct

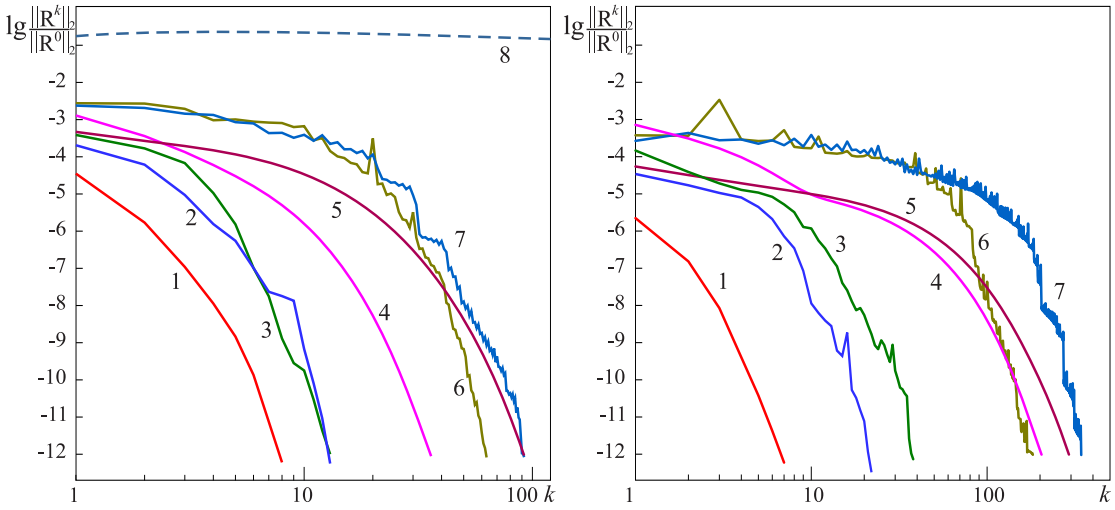
No.	Type of transform	Method	Abbreviation expansion
1	DCPt5	LR2sK	Deferred Correction Procedure with profile of the fifth degree polynomial + Line-by-Line Recurrent Method of the second order, accelerated in Krylov subspaces [19]
2	C2	LR2sK	Compensatory Transform Technology of the second order accuracy + Line-by-Line Recurrent Method of the second order, accelerated in Krylov subspaces
3	C1	LR1sK	Compensatory Transform Technology of the first order accuracy + Line-by-Line Recurrent Method of the first order, accelerated in Krylov subspaces [19]
4	C1	LR1	Compensatory Transform Technology of the first order accuracy + Line-by-Line Recurrent Method of the first order
5	C2	LR2	Compensatory Transform Technology of the second order accuracy + Line-by-Line Recurrent Method of the second order
6	–	BCGSt9 B	Bi-Conjugate Gradient Stabilized Method [17] for nine-diagonal matrix of SLAE with preconditioner on the base of explicit Buleev method [2, 18]
7	DCPt5	BCGSt B	Deferred Correction Procedure with profile of the fifth degree polynomial + Bi-Conjugate Gradient Stabilized Method for five-diagonal matrix of SLAE on the base of explicit Buleev method
8	–	BSOR9	Block Successive Over Relaxation Method [2] for nine-diagonal matrix of SLAE

The maximum effective value of the iteration parameter was selected for each method in each calculation, since all methods use the iteration parameters. In other words, an upper estimate of the effectiveness was made for each method. This approach made it possible to correctly identify the advantages of one methods in relation to others because all methods were placed in the same conditions.

3.2. Results: coarse and fine meshes

The most interesting for the analysis is the behaviour of the convergence curves which are the dependencies of the $\left\|R^k\right\|_2 / \left\|R^0\right\|_2$ value on the iteration number or the CPU time of the problem. Here $\left\|R^k\right\|_2$ is Euclidean norm of the residual error at the kth iteration. Such convergence curves as functions of the iteration number are plotted in Fig. 4 for the coarse and fine meshes. It is not difficult to see that accuracy of the solution convergence is 10^{-12}. The same value of accuracy is applied in all other results of the work.

Analysis of the curves in Fig. 4a allows the following conclusions. First, the classical block SOR method (curve 8) is not usable due to a huge number of iteration to converge the method – more than one thousand. Naturally, there is not enough place for such curve on the graph. Second, the combination DCPt5 + LR2sK (curve 1) is most effective both in reducing the initial residual



a) mesh 501×501 b) mesh 4001×4001

Fig. 4. Behavior of the convergence curves depending on the iteration number. Methods: 1 — DCPt5 + LR2sK, 2 — C2 + LR2sK, 3 — C1 + LR1sK, 4 — C1 + LR1, 5 — C2 + LR2, 6 — BCGSt9 B, 7 — DCPt5 + BCGSt B, 8 — BSOR9

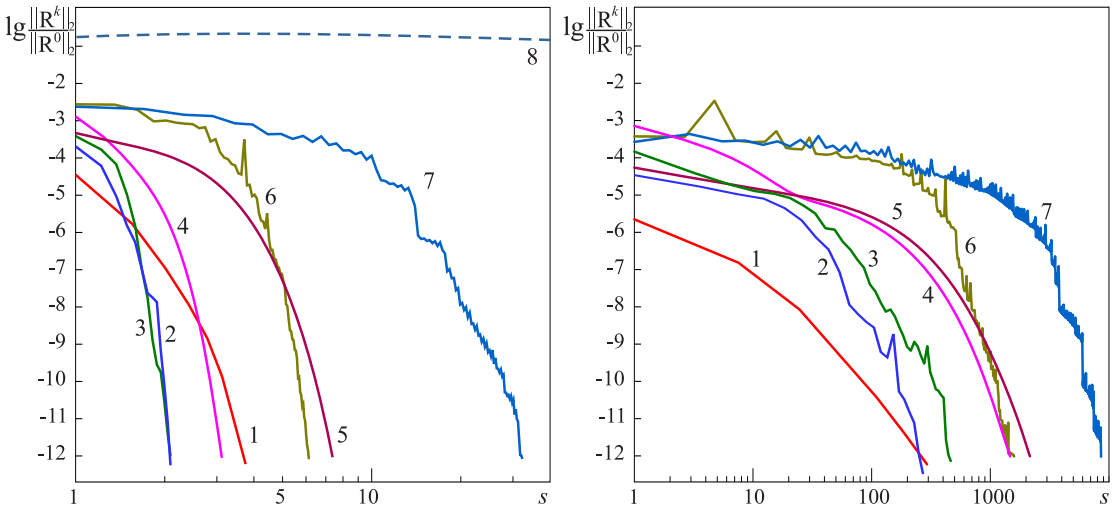
error on the first iteration and the total number of iterations for the method convergence. And finally, third, in whole, the versions with line-by-line algorithm are more powerful with respect to the variants of the bi-conjugate gradient method. As to calculations with the fine mesh (see Fig. 4b), here the results coincide qualitatively with the ones on the coarse grid, but, as a rule, the number of iterations for solution convergence is several times greater. The absence of the convergence curve of the BSOR9 method is explained by the lack of convergence of the method – the relative residual error was more than 5×10^{-8} after 3000 iterations.

It is obvious that different methods require different amounts of mathematical operations and, accordingly, different amounts of a CPU time to perform calculations of one iteration. Again, a researcher is ultimately interested in the time spent by a computer for working out a solution. Therefore, a comparison of computation times is also of research interest. Yet it is clear, that only the relative CPU times have a sense here. In other words, only the times of calculations performed on the same computer can be compared. Precisely such results in the form of convergence curves are shown in Fig. 5 for coarse and fine meshes. From now on, CPU time is presented in seconds.

It is easy to see that the combination of DCPt5 + LR2sK methods (curve 1) has not even got into the "top three winners". The reason is quite clear: recalculation of the right part of the system of linear equations by DC procedure at each iteration is a time-consuming activity. Owing to similar arguments the CPU time of the DCPt5 + LR2sK combination is almost equal to CPU time of the C2 + LR2sK one for the 4001×4001 mesh (see Fig. 5b). In all other respects, the relative behaviours of the curves in the graphs of Fig. 4 and Fig. 5 coincide qualitatively.

As is known, the average rate of convergence is one of the most indicative characteristics of the efficiency of an iterative method which is formulated as follows

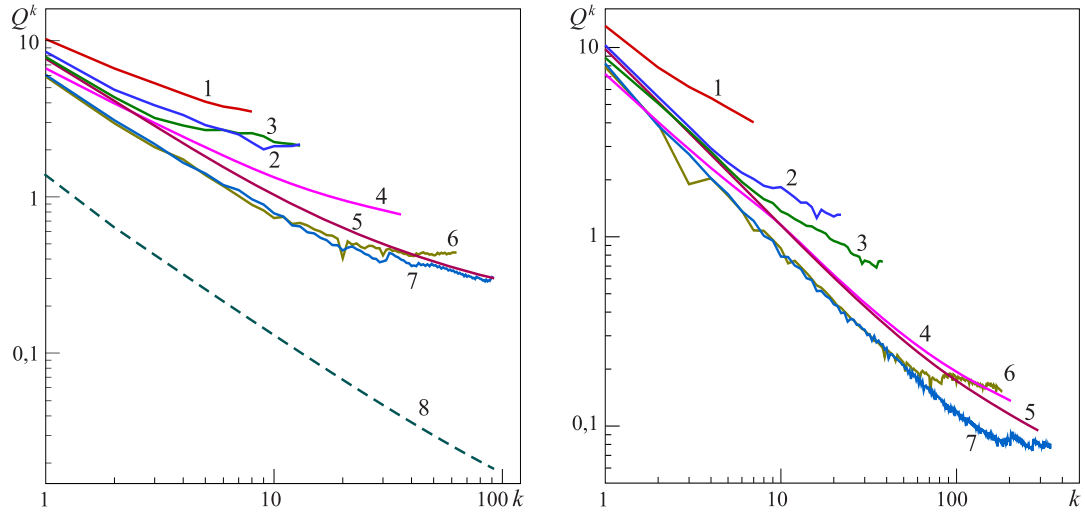
$$Q^k = \lim_{k \to \infty} \left(-\frac{1}{k} \ln \frac{\|Z^k\|_2}{\|Z^0\|_2} \right), \tag{13}$$



a) mesh 501×501 b) mesh 4001×4001

Fig. 5. Behavior of the convergence curves depending on CPU time. Methods: 1 — DCPt5 + LR2sK, 2 — C2 + LR2sK, 3 — C1 + LR1sK, 4 — C1 + LR1, 5 — C2 + LR2, 6 — BCGSt9 B, 7 — DCPt5 + BCGSt B, 8 — BSOR9

where $\|Z^k\|_2 = \sqrt{\sum_{ij}\left(\Phi_{ij}^k - u_{ij}\right)^2}$ — is Euclidean norm of the solution error, u — is analytical solution of the problem. The curves of the average convergence rate for the performed calculations with coarse and fine meshes are shown in Fig. 6. It goes without saying that the higher the curve the more efficient the method is. The low-lying curve 8 once again confirms the relative inefficiency of the classical method BSOR9. The productivities of the other methods are comparable with each other. And as expected from the previous graphs, the highest curve 1 corresponds to the most effective method – the combination DCPt5 + LR2sK. The almost direct behavior of the curves in the logarithmic system of coordinates indicates a power dependence of the average rate of convergence Q^k on the iteration number k.



a) mesh 501×501 b) mesh 4001×4001

Fig. 6. Behavior of the average convergence rate curves depending on iteration number. Methods: 1 — DCPt5 + LR2sK, 2 — C2 + LR2sK, 3 — C1 + LR1sK, 4 — C1 + LR1, 5 — C2 + LR2, 6 — BCGSt9 B, 7 — DCPt5 + BCGSt B, 8 — BSOR9

Some quantitative results of solving the problem on the coarse and fine grids with the use of the methods under consideration are presented in Tab. 2. The data in brackets for BSOR9 method for grid 4001×4001 emphasize the lack of the solution convergence with the required accuracy. Here $\left\|Z^k\right\|_\infty$ value is an infinity norm of the error at the moment of solution convergence. One

Table 2

The results of solving the problem by various methods with various meshes

Mesh	Type of transform	Method	θ	Number of iterations	CPU time, s	$\left\|Z^k\right\|_\infty$
501×501	–	BSOR9	1.9870	1 331	73.3	2.85E-05
	–	BCGSt9 B	0.99980	63	6.2	2.85E-05
	DCPt5	BCGSt B	0.999922	92	32.0	2.88E-05
	DCPt5	LR2sK	0.99999945	8	3.8	2.88E-05
	C1	LR1	0.999720	36	3.1	2.85E-05
	C2	LR2	0.99999350	92	7.4	2.85E-05
	C1	LR1sK	0.99930	13	2.1	2.85E-05
	C2	LR2sK	0.9999950	13	2.1	2.85E-05
4001×4001	–	BSOR9	1.9980	(3 000)	(29 833.2)	(4.13E-03)
	–	BCGSt9 B	0.9999979	182	1 655.2	4.46E-07
	DCPt5	BCGSt B	0.99999942	343	8 425.9	4.82E-06
	DCPt5	LR2sK	0.9999999995	7	288.0	4.81E-06
	C1	LR1	0.999972	204	1 448.7	4.42E-07
	C2	LR2	0.999999953	292	2 120.5	4.42E-07
	C1	LR1sK	0.999932	38	463.8	4.46E-07
	C2	LR2sK	0.999999920	22	279.2	4.45E-07

can see the norm is reduced by two orders of magnitude with a decrease in the value of the grid step by only an order of magnitude. It should also be noted that the number of iterations in this case increases by less than an order of magnitude, while the CPU time is increased by as much as two orders of magnitude on average. Special attention should be paid to the fact that 8 iterations were required for the method convergence on the 501×501 grid, and only 7 iterations — on the 4001×4001 grid, while using the combination DCPt5 + LR2sK. The explanation for this fact will be presented a little later.

3.3. Influence of the mesh resolution

The influence of the grid resolution on the number of iterations and the CPU time of the solution convergence is presented in Fig. 7. It is known that an increase in the dimension of SLAE (a decrease in the magnitude of the grid step), other things being equal, leads to an increase in the number of iterations [20]. Exactly such regularity takes place in Fig. 7a, except for curve 1 (the combination DCPt5 + LR2sK), which demonstrates the decrease in the number of iterations with increasing the grid dimensionality. The reason is that the original line-by-line recurrent method has a fundamental individuality: as the grid step decreases, the number of iterations required for convergence decreases [15]. This feature of the method has been manifested only in the combination DCPt5 + LR2sK. Yet other combinations with line-by-line recurrent method (curves 2–5) do not demonstrate the features because the presence

of the approximate compensatory transform technology in the combinations suppresses this fundamental individuality.

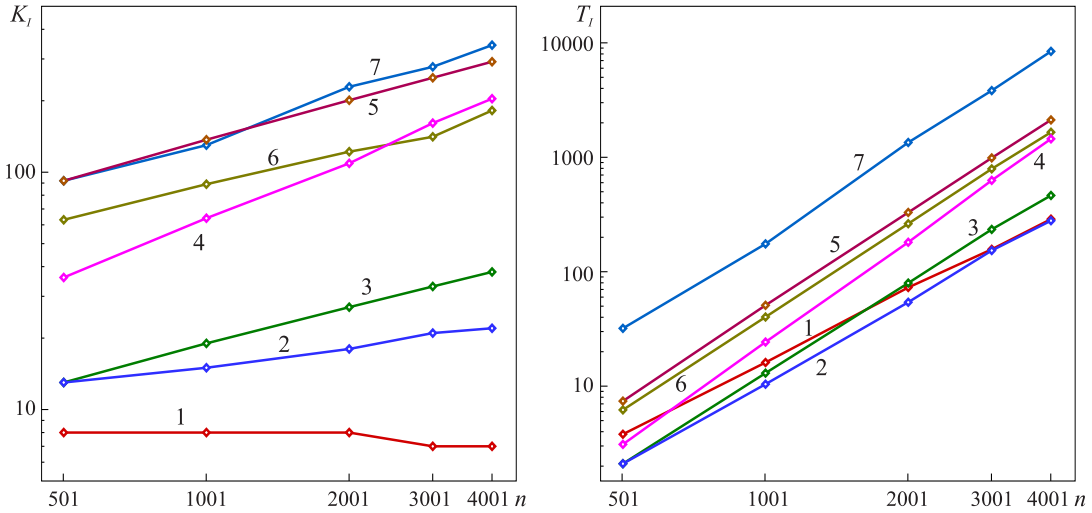


Fig. 7. The number of iterations (K_I) and CPU time (T_I) required for the convergence of the method, depending on the grid resolution. Methods: 1 — DCPt5 + LR2sK, 2 — C2 + LR2sK, 3 — C1 + LR1sK, 4 — C1 + LR1, 5 — C2 + LR2, 6 — BCGSt9 B, 7 — DCPt5 + BCGSt B

In the general case, line-by-line recurrent method of the second order LR2 is more efficient than the one of the first order LR1, regardless of whether this method was accelerated in Krylov subspaces or not, and the relationship of the curves 2 and 3 confirms this thesis. However, the locations of the curves 4 and 5 demonstrate the opposite. The reason is that additional approximate compensatory transformation technology decreases the method stability. It is necessary to lower the value of compensation parameter θ in relation to its optimum value to maintain stability. As an effect, the more parameter θ differs from its optimum, the slower the method carries out. And one has to make θ lower for LR2 than for LR1 due to lower stability of LR2, which in turn leads to a greater slowdown LR2 in relation to LR1.

And finally, it is not difficult to see that power dependences of the number of iterations K_I and CPU time T_I against a mesh resolution take place because graphs are almost direct in the logarithmic coordinates.

Conclusions

The technology of expanding the use of the line-by-line recurrent method on the case of SLAE with nine-diagonal matrices arising from the difference approximation of 2D boundary value problems of higher order was considered in the article. Approximation of the government differential equation have been carried out using the SMART scheme of the third order of accuracy. Also, approximation of the second order of accuracy in the near-boundary nodes using the Patankar scheme instead of classical upwind one of the first order of accuracy was proposed in the paper. Both the known deferred correction method and the original compensatory transform technology were used in the work to replace an initial nine-diagonal matrix of SLAE with a five-diagonal one. The comparative analysis of several modern methods and their combinations with algorithms for replacement of nine-diagonal matrices with five-diagonal ones to solve SLAE has been performed to reveal their efficiency in relation to each other.

Based on the conducted study, one can draw the following conclusions:

1. The compensatory transform technology does not require recalculation of the right part of the system of equations and, at least, is not inferior in efficiency to the deferred correction method.
2. The high efficiency of the line-by-line recurrent method is also conserved in the solving of systems of linear equations with the nine-diagonal matrix when considering two-dimensional boundary value problems.
3. The number of iterations and, accordingly, the CPU time required for the solution convergence has a power dependence against the grid resolution for all previously explored methods.

References

1. Patankar S.V. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation. New York, 1980. 197 p.
2. Il'in V.P. *Iterative Incomplete Factorization Methods*. Singapore: World Scientific Publishing Co., 1992. 300 p.
3. Leonard B.P. A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation. *Computer Methods in Applied Mechanics and Engineering*. 1979. vol. 19, no. 1. pp. 59–98. DOI: 10.1016/0045-7825(79)90034-3.
4. Gaskell P.H., Lau A.K.C. Curvature-Compensated Convective Transport: SMART, A New Boundedness – Preserving Transport Algorithm. *International Journal for Numerical Methods in Fluids*. 1988. vol. 8. pp. 617–641. DOI: 10.1002/fld.1650080602.
5. Leonard B.P. The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection. *Computer Methods in Applied Mechanics and Engineering*. 1991. vol. 88, no. 1. pp. 17–74. DOI: 10.1016/0045-7825(91)90232-U.
6. Darwish M.S. A New High-Resolution Scheme Based on the Normalized Variable Formulation. *Numerical Heat Transfer, Part B: Fundamentals*. 1993. vol. 24, iss. 3. pp. 353–371. DOI: 10.1080/10407799308955898.
7. Darwish M.S., Moukalled F. The Normalized Weighting Factor Method: a Novel Technique for Accelerating the Convergence of High-Resolution Convective Schemes. *Numerical Heat Transfer, Part B: Fundamentals*. 1996. vol. 30, iss. 2. pp. 217–237. DOI: 10.1080/10407799608915080.
8. Chirkov D.V., Chernyi S.G. Comparison of Accuracy and Convergence of Some TVD-Schemes. *Vychislitelnye tekhnologii* [Computational Technologies]. 2000. vol. 5, no. 5. pp. 86–107. (in Russian).
9. Sukhinov A.I., Chistakov A.E., Yakobovskii M.V. Accuracy of the Numerical Solution of the Equations of Diffusion–Convection Using the Difference Schemes of Second and Fourth Order Approximation Error. *Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya: "Vychislitelnaya matematika i informatika"* [Bulletin of the South Ural State University. Series:

Computational Mathematics and Software Engineering]. 2016. vol. 5, no. 1. pp. 47–62. (in Russian) DOI: 10.14529/cmse160105.

10. Prokudina L.A., Yaparova N.M., Vikhirev M.P Numerical Simulation of the Oscillations of the Elements of the Pipe with the Flow of an Incompressible Fluid. *Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya: "Vychislitelnaya matematika i informatika"* [Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering]. 2018. vol. 7, no. 3. pp. 55–64. (in Russian) DOI: 10.14529/cmse180304.

11. Schneider G.E., Zedan M. A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems. *Numerical Heat Transfer*. 1981. vol. 4, no. 1. pp. 1–19. DOI: 10.1080/01495728108961775.

12. Zverev V.G. Modified Line-by-Line Method for Difference Elliptic Equations. *Zhurnal vychislitelnoi matematiki i matematicheskoi fiziki* [Computational Mathematics and Mathematical Physics]. 1998. vol. 38, no. 9. pp. 1490–1498. (in Russian).

13. Sikovskii D.F. *Metody vychislitelnoi teplofiziki* [Computational Thermal Physics Methods]. Novosibirsk: NSU, 2013. 98 p. (in Russian).

14. Fomin A.A., Fomina L.N. On the Solution of Fluid Flow and Heat Transfer Problem in a 2D Channel with Backward–Facing Step. *Vestnik Samarskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: "Fiziko-matematicheskie nauki"* [Journal of Samara State Technical University. Series: Physical and Mathematical Sciences]. 2017. vol. 21, no. 2. pp. 362–375. DOI: 10.14498/vsgtu1545.

15. Fomin A.A., Fomina L.N. On the Convergence of the Implicit Iterative Line–by–Line Recurrence Method for Solving Difference Elliptical Equations. *Kompyuternye issledovaniya i modelirovanie* [Computer Research and Modeling]. 2017. vol. 9, no. 6. pp. 857–880. (in Russian) DOI: 10.20537/2076-7633-2017-9-6-857-880.

16. Darwish M.S., Moukalled F.H. Normalized Variable and Space Formulation Methodology for High-Resolution Schemes. *Numerical Heat Transfer, Part B: Fundamentals*. 1994. vol. 26, iss. 1. pp. 79–96. DOI: 10.1080/10407799408914918.

17. Van der Vorst H.A. BI-CGSTAB: a Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*. 1992. vol. 13, iss. 2. pp. 631–644. DOI: 10.1137/0913035.

18. Starchenko A.V. Comparative Analysis of Some Iterative Methods for the Numerical Solution of a Spatial Boundary Value Problem for Elliptic Equations. *Vestnik Tomskogo gosudarstvennogo universiteta. Byulleten operativnoi nauchnoi informatsii* [Bulletin of the Tomsk State University. The Bulletin of Operational Scientific Information]. Tomsk: TSU, 2003. no. 10. pp. 70–80. (in Russian).

19. Fomin A.A., Fomina L.N. Acceleration of the Line-by-Line Recurrent Method in Krylov Subspaces. *Vestnik Tomskogo gosudarstvennogo universiteta. Matematika i mekhanika* [Tomsk State University Journal of Mathematics and Mechanics]. 2011. no. 2. pp. 45–54. (in Russian).

20. Faddeeva V.N. *Computational Methods of Linear Algebra*. N.Y.: Dover Publications, 1959. 252 p.

УДК 519.63 DOI: 10.14529/cmse190201

ПРИМЕНЕНИЕ НЕЯВНОГО ИТЕРАЦИОННОГО ПОЛИНЕЙНОГО РЕКУРРЕНТНОГО МЕТОДА ПРИ РЕШЕНИИ СИСТЕМ РАЗНОСТНЫХ ЭЛЛИПТИЧЕСКИХ УРАВНЕНИЙ С ДЕВЯТИДИАГОНАЛЬНЫМИ МАТРИЦАМИ

© 2020 А.А. Фомин[1], Л.Н. Фомина[2]

[1]*Кузбасский государственный технический университет имени Т.Ф. Горбачева (650000 Кемерово, ул. Весенняя, д. 28),*

[2]*Кемеровский государственный университет (650043 Кемерово, ул. Красная, д. 6)*

E-mail: fomin_aa@mail.ru, lubafomina@mail.ru

Поступила в редакцию: 21.11.2018

В статье исследуется применение неявного итерационного полинейного рекуррентного метода для решения систем линейных разностных уравнений с девятидиагональными матрицами, которые возникают при разностной аппроксимации двумерных задач повышенного порядка точности на регулярном сеточном покрытии области решения. Поскольку изначально неявный итерационный полинейный рекуррентный метод разработан для решения систем уравнений с пятидиагональной матрицей, в работе предлагается технология так называемой компенсационной трансформации, позволяющая заменить исходную девятидиагональную матрицу системы уравнений на пятидиагональную. Эффективность подобного подхода анализируется путем сравнения параметров сходимости решения модельной краевой задачи в единичном квадрате как различными вариантами предложенного метода, так и другими современными высокоэффективными итерационными методами решения систем разностных уравнений. Задача решается на последовательности сеток от грубой в 501×501 узлов до подробной в 4001×4001 узлов. Точность сходимости решения определяется по относительной норме невязки, которая в настоящей работе равняется 10^{-12}. Показано, что несмотря на использование промежуточной технологии компенсационной трансформации, неявный итерационный полинейный рекуррентный метод сохраняет свои высокие скоростные и разрешающие способности во всем диапазоне сеточного разбиения области решения задачи.

Ключевые слова: метод сеток, система разностных эллиптических уравнений, итерационный метод, сходимость решения.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Fomin A. A., Fomina L. N. The Use of Line-by-Line Recurrent Method for Solving Systems of Difference Elliptic Equations with Nine-Diagonal Matrices // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 8, № 2. С. 5–21. DOI: 10.14529/cmse190201.

Литература

1. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости. М.: Энергоатомиздат, 1984. 152 с.
2. Ильин В.П. Методы неполной факторизации для решения алгебраических систем. М.: Физматлит, 1995. 288 с.
3. Leonard B.P. A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation // Computer Methods in Applied Mechanics and Engineering. 1979. Vol. 19, No. 1. P. 59–98. DOI: 10.1016/0045-7825(79)90034-3.

4. Gaskell P.H., Lau A.K.C. Curvature–Compensated Convective Transport: SMART, A New Boundedness – Preserving Transport Algorithm // International Journal for Numerical Methods in Fluids. 1988. Vol. 8. P. 617–641. DOI: 10.1002/fld.1650080602.

5. Leonard B.P. The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection // Computer Methods in Applied Mechanics and Engineering. 1991. Vol. 88, No. 1. P. 17–74. DOI: 10.1016/0045-7825(91)90232-U.

6. Darwish M.S. A New High-Resolution Scheme Based on the Normalized Variable Formulation // Numerical Heat Transfer, Part B: Fundamentals. 1993. Vol. 24, Iss. 3. P. 353–371. DOI: 10.1080/10407799308955898.

7. Darwish M.S., Moukalled F. The Normalized Weighting Factor Method: a Novel Technique for Accelerating the Convergence of High–Resolution Convective Schemes // Numerical Heat Transfer, Part B: Fundamentals. 1996. Vol. 30, Iss. 2. P. 217–237. DOI: 10.1080/10407799608915080.

8. Чирков Д.В., Черный С.Г. Сравнение точности и сходимости некоторых TVD–схем // Вычислительные технологии. 2000. Т. 5, № 5. С. 86–107.

9. Сухинов А.И., Чистяков А.Е., Якобовский М.В. Точность численного решения уравнения диффузии–конвекции на основе разностных схем второго и четвертого порядков погрешности аппроксимации // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2016. Т. 5, № 1. С. 47–62. DOI: 10.14529/cmse160105.

10. Прокудина Л.А., Япарова Н.М., Вихирев М.П. Численное моделирование колебаний элементов трубы с потоком несжимаемой жидкости // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2018. Т. 7, № 3. С. 55–64. DOI: 10.14529/cmse180304.

11. Schneider G.E., Zedan M. A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems // Numerical Heat Transfer. 1981. Vol. 4, No. 1. P. 1–19. DOI: 10.1080/01495728108961775.

12. Зверев В.Г. Модифицированный полинейный метод решения разностных эллиптических уравнений // Журнал вычислительной математики и математической физики. 1998. Т. 38, № 9. С. 1553–1562.

13. Сиковский Д.Ф. Методы вычислительной теплофизики: Учеб. пособие. Новосибирск: Новосибирский государственный университет, 2013. 98 с.

14. Fomin A.A., Fomina L.N. On the Solution of Fluid Flow and Heat Transfer Problem in a 2D Channel with Backward–Facing Step // Вестник Самарского государственного технического университета. Серия: Физико-математические науки. 2017. Т. 21, № 2. С. 362–375. DOI: 10.14498/vsgtu1545.

15. Фомин А.А., Фомина Л.Н. О сходимости неявного итерационного полинейного рекуррентного метода решения систем разностных эллиптических уравнений // Компьютерные исследования и моделирование. 2017. Т. 9, № 6. С. 857–880. DOI: 10.20537/2076-7633-2017-9-6-857-880.

16. Darwish M.S., Moukalled F.H. Normalized Variable and Space Formulation Methodology for High-Resolution Schemes // Numerical Heat Transfer, Part B: Fundamentals. 1994. Vol. 26, Iss. 1. P. 79–96. DOI: 10.1080/10407799408914918.

17. Van der Vorst H.A. BI-CGSTAB: a Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems // SIAM Journal on Scientific and Statistical Computing. 1992. Vol. 13, Iss. 2. P. 631–644. DOI: 10.1137/0913035.

18. Старченко А.В. Сравнительный анализ некоторых итерационных методов для численного решения пространственной краевой задачи для уравнений эллиптического типа // Вестник ТГУ. Бюллетень оперативной научной информации. Томск: ТГУ, 2003. № 10. С. 70–80.

19. Фомин А.А., Фомина Л.Н. Ускорение полинейного рекуррентного метода в подпространствах Крылова // Вестник Томского государственного университета. Математика и механика. 2011. № 2. С. 45–54.

20. Фаддеев Д.К., Фаддеева В.Н. Вычислительные методы линейной алгебры. М.: Физматгиз, 1963. 656 с.

Фомин Александр Аркадьевич, к.ф.-м.н., ст.н.с., отдел развития и международного сотрудничества, Кузбасский государственный технический университет имени Т.Ф. Горбачева (Кемерово, Российская Федерация)

Фомина Любовь Николаевна, к.ф.-м.н., доцент, кафедра ЮНЕСКО по информационным вычислительным технологиям, институт фундаментальных наук, Кемеровский государственный университет (Кемерово, Российская Федерация)

DOI: 10.14529/cmse190202

DEVELOPMENT OF A NUMERICAL METHOD FOR SOLVING THE INVERSE CAUCHY PROBLEM FOR THE HEAT EQUATION

© 2019 H.K. Al-Mahdawi
South Ural State University
(pr. Lenina 76, Chelyabinsk, 454080 Russia)
E-mail: hssnkd@gmail.com
Received: 04.06.2018

In this work, the initial temperature has been investigated in the Cauchy inverse problem for linear heat conduction equation that it depends on the given temperature at specification time. In this problem, the initial temperature distribution is unknown, but instead, there is a known temperature at the time, $t = T > 0$. The heat conduction problem can be formulated as Fredholm integral first kind equation. It is well known that this problem is an ill-posed problem and direct solution to this problem is unacceptable. An algorithm has been used to define a finite-dimensional operator for this problem also used the generalized discrepancy method to reduce the conditional extremum variation problem to unconditional extremum variation problem for the integral equation. The discretization of the integral equation has made it possible to reduce this problem to a system of linear algebraic equations. Then, Tikhonov's regularization inversion method has been used to find an approximation solution. Finally, the numerical computation example has been presented to verify the accuracy of the estimated solution.

Keywords: ill-posed problem, regularization, inverse problem, heat conduction.

FOR CITATION

Al-Mahdawi H.K. Development of a Numerical Method for Solving the Inverse Cauchy Problem for the Heat Equation. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2019. vol. 8, no. 2. pp. 22–31. DOI: 10.14529/cmse190202.

Introduction

The Cauchy inverse problem of heat equation is ill-posed in the sense that arbitrarily "small" change in the data can produce "large" errors in the solution. The problem can be defined in the sense of Jacques Hadamard, that a problem is well-posed if and only if the following properties hold [1].

- The solution exists, at least one solution exists (existence).
- The solution is unique, at most one solution exists (uniqueness).
- The solution depends continuously on the data (stability).

It is impossible to solve the ill-posedness problem by using classical numerical methods. It requires special techniques, e.g., regularization strategies. With the development of high-speed personal computers, it has become more convenient to use numerical techniques to solve heat transfer inverse problems. Theoretical concepts and computational implementation related to Cauchy inverse problem of heat equation have been discussed by many authors, and a lot of methods have been described [3–7].

In some of them, the author has been identified the heat flux at the front surface of a thick plate based on the measured temperature history at the plate back surface, which is insulated [3]. In [4] the author has been applied the numerical method involving the Laplace transform technique and the finite difference method in conjunction with the least-squares

scheme to an Inverse Heat Conduction Problems. The inversion model that simultaneously highlights the measurement errors and the inaccurate properties of the forward problem has been proposed in [5] to improve the inversion accuracy and robustness. With the assistance of the Tikhonov regularization method, a cost function is constructed to convert the original an Inverse Heat Conduction Problems into an optimization problem [5]. In other paper, a model has been developed to solve the inverse heat conduction problems for a triangular wall. The conjugate gradient method has been used with the finite element method to determine the two-dimensional variations of the temperatures and heat fluxes on the wall surface with time [6]. In [7] the Cauchy problem for the Laplace equation in a multiply connected region was solved by replaced the heat conduction problem to the Poisson equation and solve it in a simply connected region with an unknown source function different from zero in the adjoined region. The methods described in the [2] are used to solve a number of inverse problems in mathematical physics. The fundamentals of the optimal methods have been obtained for solving ill-posed problems, as well as ways to estimate accurate solution and accurate by order error estimates for these methods.

The main idea in this paper is to reconstruct the source function of the diffusion equation by using the algorithm which proposed in [8]. The corresponding inverse problem, by Fourier series expansion, has been represented as Fredholm equation of the first kind. Hence, the solution does not depend continuously on the data in conventional Banach spaces, so the solution unstable [1, 2, 8]. Therefore, this is an ill-posed problem. To get a well-posed problem Tikhonov Regularization will be using. The problem of selecting the best regularization parameter will be solved in this paper by using the residual principal method which described in [2].

All these steps will be implemented through the sections in this paper. Section 1 defines the direct problem for heat conduction problem as a linear partial differential equation and describes solution as an integral Fredholm equation of the first kind. Section 2 defines the inverse problem and give a discerption about the known data and operator. Section 3 considers the integral equation of the first kind and reduces it as a system of linear algebraic equations by implementing the algorithm in [8]. Then, in Section 4 the example has been presented to verify the accuracy of our estimated solution. Finally, the explanation of the suggested method has been summarized in the conclusion Section with suggested future work for solving the nonlinear backward heat problem.

1. Direct problem

The direct (forward) problem consists of passing heat conduction through a bar with the determined boundary condition and initial temperature condition. The mathematical formulation of this problem has been described by the following liner partial differential equation

$$\frac{\partial u(x,t)}{\partial t} = D\,\frac{\partial^2 u(x,t)}{\partial x^2}, 0 < x\ < 1\,, t\ \geq 0, \tag{1}$$

$$u(0,t) = 0, t\ \geq 0, \tag{2}$$

$$u(l,t) = 0, t\ \geq 0, \tag{3}$$

$$u(x,0) = u_0(x), 0 \leq x\ \leq 1, \tag{4}$$

where the $u(0,t)$ and $u(1,t)$ are boundary conditions, $u_0(x)$ initial condition it is representing the initial temperature. The (t) represents the time, (x) spatial variable and (D) denote the dispersion coefficient.

In the direct problem (1–4), the initial condition has been specified. For solving this type of problem there are many ways such as finite different method (FMD) and separation of variables. To formulate this problem as the Fredholm integral equation first kind, the Fourier series method by separation of variables has been used as follows:

$$u(x,t) = \sum_{n=1}^{\infty} a_n \, e^{-(n\pi)^2 t} \, sin(n\pi x), \tag{5}$$

$$u(x,0) = u_0(x) = \sum_{n=1}^{\infty} a_n \; sin(n\pi x), \tag{6}$$

$$\forall \; a_n = 2\int_0^1 u_0(x) \; sin(n\pi x)\, dx, \tag{7}$$

from (5) and (7) we get

$$u(x,t) = 2\int_0^1 \sum_{n=1}^{\infty} e^{-(n\pi)^2 t} \sin(n\pi x) \sin(n\pi y)\, u_0(y)\, dy. \tag{8}$$

The formula (8) is rewriting as integral equation first kind for some fixed $t = T$ as following:

$$u(x,t) = \int_0^1 K(x,y)\, u_0(y)\, dy \;, 0 \le x \le \; 1, \tag{9}$$

or we can write it as following:

$$Au(x) = \int_0^1 \; K(x,y)\, u_0(y)\, dy, \tag{10}$$

where $K(x,y) = \frac{2}{l}\sum_{n=1}^{\infty} e^{-(n\pi)^2 T} \sin(n\pi x) \sin(n\pi y)$ and $u_0(y)$ the initial function.

Where the kernel $K(x,y) \in C\,([0,1]\,X\,[0,1])$, $u_0(y) \in L_2[0,1]$ and $f(x) \in L_2[0,1]$. The kernel of the operator A is closed.

2. Inverse problem

The inverse problem, described as the initial temperature $u_0(y)$ is the unknown function inside integral. To estimate the unknown initial temperature the measurement temperature has been given at specific time T over the specified space interval $0 \le x \; \le 1$

$$u(x,T) = f(x), T > 0. \tag{11}$$

The measurement temperature includes some noise $f_\delta(x)$, where $\delta > 0$ defined as the range of error, $\|f_\delta(x) - f_0(x)\|_{L_2} \le \delta$. Additionally, the inverse operator A^{-1} is unbounded $\|A^{-1}\| = \infty$, it means the solution usually poor approximated even small value of δ. All this lead to the inverse heat conduction problem it ill-posed problem because the solution is not stable.

3. Computational scheme

We considered the following integral equation of the first kind. Our target reduces this problem to a system of linear algebraic equations

$$Au(x) = \int_0^1 K(x,y)\, u_0(y)\, dy = f(x), T > 0. \tag{12}$$

The kernel $K(x,y,T)$ is an infinite series and we cannot handle infinite sum and when n goes to ∞ the value $e^{-(n\pi)^2 T}$ become very small for simplicity, we finite the sum of series to 10 times

$$K(x,y) = 2\sum_{n=1}^{10} e^{-(n\pi)^2 T}\sin(n\pi x)\sin(n\pi y), T > 0. \quad (13)$$

Now introduce the operator C map $L_2[0,1]$ into $L_2(0,T]$

$$Cu(y) = \int_0^1 K(x,y)\,u(y)dy, \quad (14)$$

for the numerical solution of the equation (14) replace operator C by the finite-dimensional operator C_m, where $C \to C_m \; and \; C_m \sim A$.

Next step need to divide interval $[0,1]$ into m equal parts by points $x_i = \frac{i(1-0)}{m}$, $i = 0,1,\dots,m-1$ and $y_j = \frac{j(1-0)}{m}$, $j = 0,1,\dots,m-1$, the width for each interval $h = \Delta x = \Delta y = (x_{i+1} - x_i)$ and $x_i = y_j$.

Now introduce the kernel function

$$K(x,y,) = \overline{K}(x_i, y_j), \quad (15)$$

where $x_i \le x < x_{i+1}, y_j \le y < y_{j+1}$, $i = 0,1,\dots,m-1$ and $j = 0,1,\dots,m-1$

$$C_m u(y) = \int_0^1 \overline{K}(x_i, y_j)\,u(y)dy = f_\delta(x), \quad (16)$$

where $u(y) \to (u_j)$, $j = 0,1,\dots,m-1$. $u_j = u(y_j)$ and $f_\delta(x) \to (f_i^\delta)$, $i = 0,1,\dots,m-1$, $f_i^\delta = f_\delta(x_i)$.

Now introduces the finite-dimensional subspaces $Y_m \; and \; X_m$ of the space $L_2[0,1]$ consisting all functions on intervals $(x_i, x_{i+1}]$, $i = 0,1,\dots,m-1$, $(y_j, y_{j+1}]$, $j = 0,1,\dots,m-1$.

We denote by P_m and Q_m the metric projection operators from $L_2[0,1]$ onto Y_m and X_m subspaces respectively

$$C_m u(y_j) = \int_0^1 \overline{K}(x_i, y_j)\,u(y_j)dy = f_\delta(x_i). \quad (17)$$

By using the generalized discrepancy method which has been described in [8] for the approximate solution of equation (16). We will reduce the equation to the conditional extremum variation problem

$$\inf\{\|u(y)\|^2 : u(y) \in Y_m, \| C_m u(y) + {f_\delta}^m(x)\| \le \delta^2\}, \quad (18)$$

where ${f_\delta}^m(x) = Q_m[f_\delta(x)]$.

The variation problem (18) is reduced to unconditional extremum variation problem

$$\inf\{\|C_m u(y) - {f_\delta}^m(x)\|^2 - \alpha\,\|u(y)\|^2 : u(y) \in Y_m\}, \alpha > 0, \quad (19)$$

which is the version of the Tikhonov regularization method

$$C_m u_j = h\sum_{j=0}^{m-1} \overline{K}(x_i, y_j)u(y_j) = f_i, \quad (20)$$

where

$$u_j = P_m[u(y_j)] = h\int_{y_j}^{y_{j+1}} u(y), j = 0,1,\dots,m-1, \quad (21)$$

the form of operator Q_m implies that ${f_\delta}^m(x) = \{f_i : x_i \le x < x_{i+1}, i = 0,1,\dots,m-1\}$, where

$$f_i = Q_m[f_\delta(x_i)] = h\int_{x_i}^{x_{i+1}} f_\delta(x), i = 0,1,\dots,m-1. \quad (22)$$

From (19–22) and to give the approximate solution to $u(x)$, we can rewrite the problem as linear algebraic equations

$$C_m(u_j) = f_i,$$

$$h * \begin{bmatrix} \overline{K}(x_0, y_0) & \overline{K}(x_0, y_1) & \dots & \overline{K}(x_0, y_{m-1}) \\ \overline{K}(x_1, y_0) & \overline{K}(x_1, y_1) & \dots & \overline{K}(x_1, y_{m-1}) \\ \vdots & \vdots & \dots & \vdots \\ \overline{K}(x_{m-1}, y_0) & \overline{K}(x_{m-1}, y_1) & \dots & \overline{K}(x_{m-1}, y_{m-1}) \end{bmatrix} \begin{bmatrix} u(y_0) \\ u(y_2) \\ \vdots \\ u(y_{m-1}) \end{bmatrix} = \begin{bmatrix} f_\delta(x_0) \\ f_\delta(x_2) \\ \vdots \\ f_\delta(x_{m-1}) \end{bmatrix}. \quad (23)$$

The problem (23) is ill-posed in the sense that the inverse operator ${C_m}^{-1}$ of C_m exists but it is not continuous. The problem (23) has a unique solution when solving it directly will not give the right solution. Indeed, the linear operator C_m is ill-conditioned that any numerical attempt to directly solve (23) may fail.

In order to find the stable solution from the equation (23) which is described Tikhonov regularization method. The computation of the approximate solution $\mathrm{u}_{\alpha\delta}$ consists in solving the Euler equation

$$\mathrm{u}_\delta(\alpha) = (C_m^* C_m + \alpha I)^{-1} C_m^* {f_i}^\delta, \quad (24)$$

where C_m^* is the operator adjoin to the operator C_m and I is identity matrix and α is regularization parameter.

To determine the regularization parameter α in solution $\mathrm{u}_j(\alpha)$ we based on the residual principle method as described in [2]. The α should satisfy following equation

$$\left\| C_m \mathrm{u}_\delta(\alpha_s) - {f_i}^\delta \right\|_{L_2}^2 = \delta^2. \quad (25)$$

4. Numerical example

Considering the problem (1–4) we need to estimate the initial temperature $\mathrm{u}_0(\mathrm{x})$ from given function $\mathrm{u}(\mathrm{x}, \mathrm{T}) = \mathrm{f}(\mathrm{x})$ with known noise level δ where time (T=0.1 and T=0.15), (m=100) and D=1 for checking the approximation solution we will use the exact initial temperature $\mathrm{u}_0(\mathrm{x}) = \sin(\mathrm{x})$ in example 1 and $\mathrm{u}_0(\mathrm{x}) = 4\sin(3\pi\mathrm{x})$ in example 2. In each example, we can find $\mathrm{f}(\mathrm{x})$ function by using the equation (8) this called forward solution and add some noise to apply the inverse algorithm to estimate the initial function and check it with the exact initial temperature.

4.1. Example 1

Let the exact solution for the problems (1–4) be

$$\mathrm{u}_0(\mathrm{x}) = \sin(\mathrm{x}), 0 \le \mathrm{x} \le 1, \quad (26)$$

we consider two cases under different time (T=0.1 and T=0.15) as shown in Fig. 1.

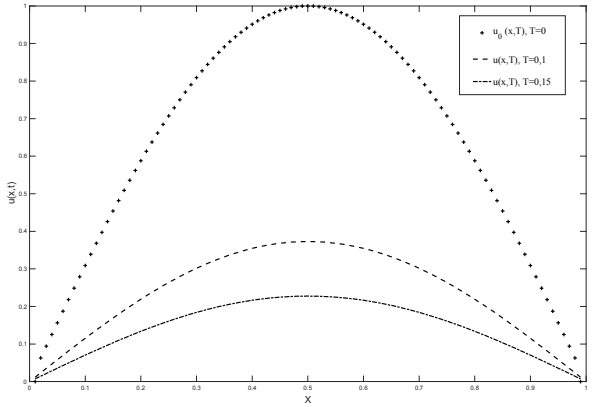
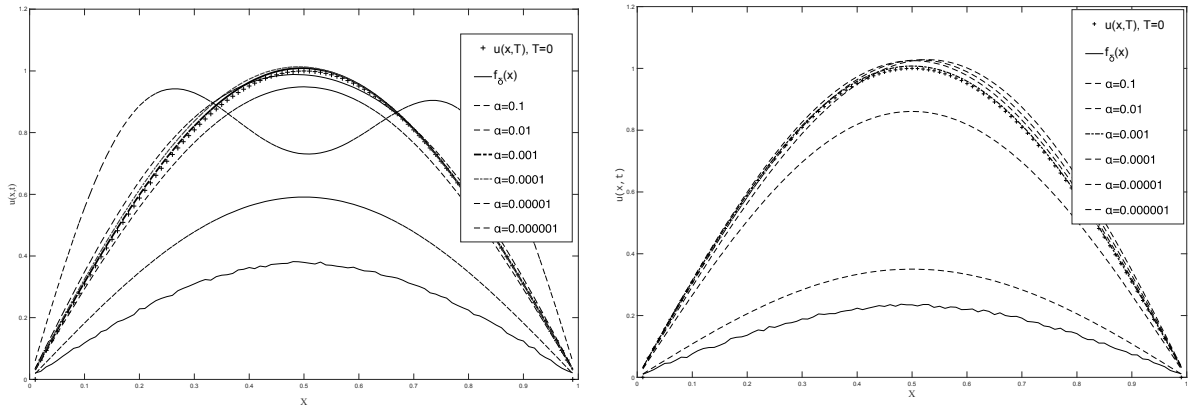


Fig. 1. Direct solution initial temperature $\mathrm{u}_0(\mathrm{x}) = \sin(\mathrm{x})$

We can add noise to each $u(x,T)$ for using them in the problem analysis. By using the equation (24) we can find estimated solutions with regularization parameters α. We can use the set of regularization parameters to obtain the best-estimated solution $\alpha_s = \{\alpha_1, \alpha_2, \dots, \alpha_s\}$, where $\alpha_1 = 10^{-1}$, $\alpha_1 = 10^{-2}$ and $\alpha_s = 10^{-s}$ as shown in Fig. 2.



a) T=0.1 and $\delta = 0.057$ b) T=0.15 and $\delta = 0.052$

Fig. 2. Inverse solution for the initial temperature $u_0(x) = \sin(x)$

The best regularization parameter α can be selected by using the residual principle method equation (25) as shown in Tab. 1.

Table 1

Best α residual principle method

α_s	$\left\|C_m u_\delta(\alpha_s) - f_i^\delta\right\|_{L_2}$, where T=0.1 and δ=0.057	$\left\|C_m u_\delta(\alpha_s) - f_i^\delta\right\|_{L_2}$, where T=0.15 and δ=0.052
$1 * 10^{-1}$	1.121952241	1.089270388
$1 * 10^{-2}$	0.183193635	0.269894863
$1 * 10^{-3}$	0.039406461	0.047722743
$1 * 10^{-4}$	0.034446107	0.036155186
$1 * 10^{-5}$	0.034377304	0.036009032
$1 * 10^{-6}$	0.034269028	0.036003374

4.2. Example 2

Let the exact solution for the problem (1–4) be

$$u_0(x) = 4\sin(3\pi x), 0 \le x \le 1, \tag{27}$$

we consider two cases under different time (T=0.01 and T=0.015) as shown in Fig. 3.

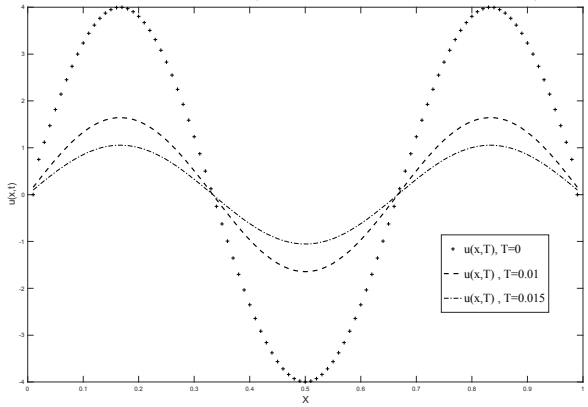
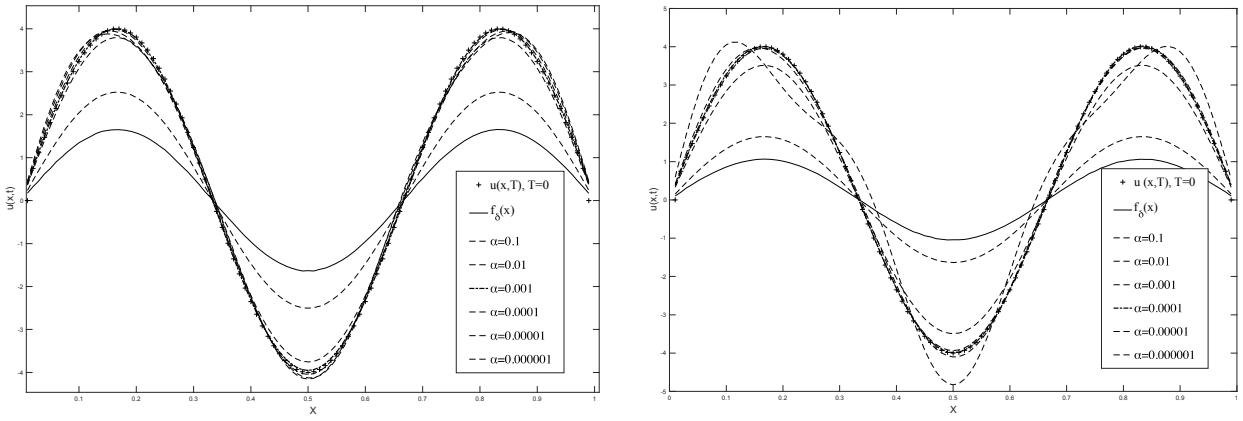


Fig. 3. Direct solution initial temperature $u_0(x) = 4\sin(3\pi x)$

In this example, we increase noise level δ and used the same set of the regularization parameters in the previous example. We obtained the best-estimated solutions as shown in Fig. 4.



a) T=0.01 and $\delta = 0.114$ b) T=0.015 and $\delta = 0.106$

Fig. 4. Inverse solution for the initial temperature $u_0(x) = 4\sin(3\pi x)$

The best regularization parameter α can be selected by using the residual principle method equation (25) as shown in Tab. 2.

Table 2

Best α residual principle method

α_s	$\left\|C_m u_\delta(\alpha_s) - f_i^\delta\right\|_{L_2}$, where T=0.01 and δ=0.114	$\left\|C_m u_\delta(\alpha_s) - f_i^\delta\right\|_{L_2}$, where T=0.015 and δ=0.106
$1 * 10^{-1}$	4.32936173855166	4.41171841520750
$1 * 10^{-2}$	0.653083706856141	0.941489551857426
$1 * 10^{-3}$	0.0905050094101809	0.119829324858028
$1 * 10^{-4}$	0.0587797017102758	0.0569163570726761
$1 * 10^{-5}$	0.0578365971019751	0.0557491308369363
$1 * 10^{-6}$	0.0578001716547491	0.0550439922968124

Conclusion

This work deals with the algorithm for solving the backward heat problem and some results have been collected. This problem is Cauchy ill-posed problem and special method need to solve such as problem. Fourier series method has been used by separation of variables for backward heat problem to represent the partial differential equation as Fredholm integral equation of the first kind. The numerical analyzes successfully apply to solve the inverse heat conducting problem by using discretization method to convert integral equation to a system of linear equations and using the Tikhonov's regularization method to estimate initial temperature and checking the estimated result with exact result. From the examples, we can note the algorithm was efficient to estimate the initial temperature depending on the given measurement temperature with the known noise level δ.

References

1. Kabanikhin S.I. Inverse and Ill-posed Problems: Theory and Applications. *Inverse and Ill-Posed Problems Series 55*. De Gruyter, 2012. 24 p.
2. Tanana V.P., Sidikova A.I. Optimal Methods for Solving Ill-posed Heat Conduction Problems. *Inverse and Ill-Posed Problems Series 62*. De Gruyter, 2018. 40 p.
3. Duda P. Solution of Inverse Heat Conduction Problem Using the Tikhonov Regularization Method. *Journal of Thermal Science*. 2017. vol. 26, no. 1. pp. 60–65. DOI: 10.1007/s11630-017-0910-2
4. Ahmadizadeh V., Soti Y., Pourgholi R., et al. Estimation of Heat Flux in One-dimensional Inverse Heat Conduction Problem. *International Mathematical Forum*. 2007. vol. 2, no. 10. pp. 455–464.
5. Mu H., Li J., Wang X., et al. Optimization Based Inversion Method for the Inverse Heat Conduction Problems. *IOP Conference Series: Earth and Environmental Science*. 2017. vol. 64, no. 1. pp. 9. DOI: 10.1088/1755-1315/64/1/012094
6. Zhu Y., Liu B., Jiang P., et al. Inverse Heat Conduction Problem for Estimating Heat Flux on a Triangular Wall. *Journal of Thermophys Heat Transf*. 2017. vol. 31, no. 1. pp. 205–210. DOI: 10.2514/1.T4877
7. Frąckowiak A., Botkin N.D., Ciałkowski M., et al. Iterative Algorithm for Solving the Inverse Heat Conduction Problems with the Unknown Source Function. Inverse Problems in Science and Engineering. 2015. vol. 23, no. 6. pp. 1056–1071. DOI: 10.1080/17415977.2014.986723
8. Tanana V.P., Sidikova A.I. On Estimating the Error of an Approximate Solution Caused by the Discretization of an Integral Equation of the First Kind. *Steklov Institute of Mathematics.* 2017. vol. 299, no. 1. pp. 217–224. DOI: 10.1134/S0081543817090231

УДК 519.642.3 DOI: 10.14529/cmse190202

РАЗРАБОТКА ЧИСЛЕННОГО МЕТОДА РЕШЕНИЯ ОБРАТНОЙ ЗАДАЧИ КОШИ ДЛЯ УРАВНЕНИЯ ТЕПЛОПРОВОДНОСТИ

© 2019 Х.К. Аль-Махдави
Южно-Уральский государственный университет
(454080 Челябинск, пр. им. В.И. Ленина, д. 76)
E-mail: hssnkd@gmail.com
Поступила в редакцию: 04.06.2018

В этой работе начальная температура была исследована в обратной задаче Коши для линейного уравнения теплопроводности, которая зависит от заданной температуры в заданное время с некоторыми шумовыми измерениями. В этой задаче начальное распределение температуры неизвестно, но вместо этого в то время известна температура, t=T $>$ 0. Задачу теплопроводности можно сформулировать так, как интегральное уравнение первого рода Фредгольма. Хорошо известно, что эта проблема является некорректной задачей, и прямое решение этой проблемы неприемлемо. Алгоритм, используемый для определения конечномерного оператора для этой задачи, также использовал метод обобщенной несоответствия для уменьшения условной проблемы вариации экстремума к безусловной проблеме изменения экстремума для интегрального уравнения. Дискретизация интегрального уравнения позволила свести эту задачу к системе линейных алгебраических уравнений. Тогда для решения аппроксимации использовался метод инверсии регуляризации Тихонова. Наконец, был представлен пример численного расчета для проверки точности оценочного решения.

Ключевые слова: некорректная задача, регуляризация, обратная задача, теплопроводность.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Al-Mahdawi H.K. Development of a Numerical Method for Solving the Inverse Cauchy Problem for the Heat Equation // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 22–31. DOI: 10.14529/cmse190202.

Литература

1. Kabanikhin S.I. Inverse and Ill-posed Problems: Theory and Applications. Inverse and Ill-Posed Problems Series 55. De Gruyter, 2012. 24 p.
2. Tanana V.P., Sidikova A.I. Optimal Methods for Solving Ill-posed Heat Conduction Problems. Inverse and Ill-Posed Problems Series 62. De Gruyter, 2018. 40 p.
3. Duda P. Solution of Inverse Heat Conduction Problem Using the Tikhonov Regularization Method // Journal of Thermal Science. 2017. Vol. 26, No. 1. P. 60–65. DOI: 10.1007/s11630-017-0910-2
4. Ahmadizadeh V., Soti Y., Pourgholi R., et al. Estimation of Heat Flux in One-dimensional Inverse Heat Conduction Problem // International Mathematical Forum. 2007. Vol. 2, No. 10. P. 455–464.
5. Mu H., Li J., Wang X., et al. Optimization Based Inversion Method for the Inverse Heat Conduction Problems // IOP Conference Series: Earth and Environmental Science. 2017. Vol. 64, No. 1. P. 9. DOI: 10.1088/1755-1315/64/1/012094
6. Zhu Y., Liu B., Jiang P., et al. Inverse Heat Conduction Problem for Estimating Heat Flux on a Triangular Wall // Journal of Thermophys Heat Transfer. 2017. Vol. 31, No. 1. P. 205–210. DOI: 10.2514/1.T4877

7. Frąckowiak A., Botkin N.D., Ciałkowski M., et al. Iterative Algorithm for Solving the Inverse Heat Conduction Problems with the Unknown Source Function // Inverse Problems in Science and Engineering. 2015. Vol. 23, No. 6. P. 1056–1071. DOI: 10.1080/17415977.2014.986723
8. Tanana V.P., Sidikova A.I. On Estimating the Error of an Approximate Solution Caused by the Discretization of an Integral Equation of the First Kind // Steklov Institute of Mathematics. 2017. Vol. 299, No. 1. P. 217–224. DOI: 10.1134/S0081543817090231

Аль-Махдави Хассан К. Ибрахим, аспирант, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

УДК 004.65, 004.421 **DOI: 10.14529/cmse190203**

ОБЗОР МЕТОДОВ ИНТЕГРАЦИИ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В СУБД

© 2019 М.Л. Цымблер

Южно-Уральский государственный университет
(454080 Челябинск, пр. им. В.И. Ленина, д. 76)
E-mail: mzym@susu.ru

Поступила в редакцию: 27.02.2019

Интеллектуальный анализ данных направлен на извлечение доступных для понимания знаний, необходимых для принятия решений в различных сферах человеческой деятельности. Феномен Больших данных является характерным признаком современного информационного общества. Процессы очистки и структурирования Больших данных приводят к образованию сверхбольших баз и хранилищ данных. Несмотря на появление большого количества NoSQL СУБД, основным инструментом управления базами данных по-прежнему остаются реляционные СУБД. Одним из перспективных направлений развития реляционных СУБД является внедрение в них средств интеллектуального анализа данных. Интеграция позволяет как избежать накладных расходов по экспорту анализируемых данных из хранилища и импорту результатов анализа обратно в хранилище, так и использовать при анализе данных системные сервисы, заложенные в архитектуре СУБД. В статье представлен обзор методов и подходов к решению задачи интеграции интеллектуального анализа данных в СУБД. Приводится классификация подходов к решению задачи интеграции интеллектуального анализа данных в СУБД. Представлены расширения языка баз данных SQL, обеспечивающие синтаксическую поддержку интеллектуального анализа данных в СУБД. Рассмотрены примеры реализации алгоритмов интеллектуального анализа данных на SQL и систем анализа данных в реляционных СУБД.

Ключевые слова: интеллектуальный анализ данных, реляционная СУБД, классификация, кластеризация, поиск шаблонов.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Цымблер М.Л. Обзор методов интеграции интеллектуального анализа данных в СУБД // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 32–62. DOI: 10.14529/cmse190203.

Введение

В настоящее время феномен *Больших данных (Big Data)* оказывает существенное влияние на технологии обработки данных [3, 19]. На сегодня имеется широкий спектр приложений (социальные сети, электронные библиотеки, геоинформационные системы и др.), в которых производятся неструктурированные данные, имеющие сверхбольшие объемы и высокую скорость прироста (от 1 Тб в день). Исследования аналитической компании IDC показывают, что мировой объем данных удваивается каждые два года и к 2020 г. достигнет 44 Зеттабайт (44 трлн. Гб) [80].

В современном информационном обществе, однако, критичными являются не объемы и скорость прироста данных, а наличие эффективных методов и алгоритмов интеллектуального анализа данных. Под *интеллектуальным анализом данных (Data Mining)* понимают совокупность алгоритмов, методов и программного обеспечения для обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия стратегически важных решений в различных сферах человеческой деятельности [25].

Процессы очистки и структурирования Больших данных приводят к образованию сверхбольших баз и хранилищ данных. Один из наиболее авторитетных ученых в области баз данных М. Стоунбрейкер указывает [75], что для решения проблем обработки сверхбольших данных необходимо использовать технологии систем управления базами данных (СУБД). Несмотря на появление большого количества NoSQL СУБД [22], СУБД на основе реляционной модели данных [21] по-прежнему остаются основным инструментом управления базами данных.

В 2016 г. в Бекманском отчете [6] ведущие мировые специалисты в области технологий обработки данных констатировали, что переход к умному обществу, управляемому данными, требует интегрированного и сквозного процесса от получения данных до извлечения из них полезных знаний.

Одним из перспективных направлений развития реляционных СУБД является внедрение в них средств интеллектуального анализа данных [54]. Размещение аналитических алгоритмов «рядом» с анализируемыми данными, имеющими сверхбольшие объемы, позволяет избежать существенных накладных расходов по экспорту анализируемых данных из хранилища и импорту результатов анализа обратно в хранилище [53]. Кроме того, интеллектуальный анализ данных внутри СУБД позволяет без дополнительных накладных расходов использовать системные сервисы, заложенные в архитектуре СУБД: отказоустойчивость, целостность и безопасность данных, исполнение запросов к данным на основе индексирования данных и управления буферным пулом и др.

Настоящая статья представляет собой обзор методов и подходов к решению задачи интеграции интеллектуального анализа данных в реляционные СУБД и организована следующим образом. В разделе 1 даны определения двух основных задач интеллектуального анализа данных: поиск шаблонов и кластеризация, — и кратко представлены основные алгоритмы их решения. В разделе 2 приводится классификация подходов к решению задачи интеграции интеллектуального анализа данных в СУБД. В разделе 3 рассмотрены методы реализации систем анализа данных в СУБД и представлены расширения языка баз данных SQL, обеспечивающие синтаксическую поддержку интеллектуального анализа данных в СУБД. Раздел 4 содержит примеры реализации алгоритмов интеллектуального анализа данных на SQL. Заключение резюмирует результаты, полученные в исследовании.

1. Основные задачи анализа данных и алгоритмы их решения

Поиск шаблонов и кластеризация являются одними из основных задач интеллектуального анализа данных [30]. Ниже приведены формальные определения указанных задач и кратко представлены алгоритмы их решения.

1.1. Поиск шаблонов

Поиск *шаблонов (pattern mining)* предполагает нахождение часто повторяющихся зависимостей в заданном наборе объектов и применяется в медицине (например, нахождение побочных эффектов лекарств), генной инженерии (поиск часто повторяющихся цепочек ДНК) и других предметных областях.

Общепринятой формой записи шаблона является *ассоциативное правило (association rule)*, которое формально определяется следующим образом. Пусть дано множество *объектов* $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$, любое непустое его подмножество называют *набором*.

Набор из k объектов ($1 \leqslant k \leqslant m$) называют *$k$-набором*. Пусть имеется множество *транзакций* (записей об операциях с объектами в данной предметной области) $\mathcal{D}$, в котором каждая транзакция представляет собой пару $(tid; I)$, где tid — уникальный идентификатор транзакции, $I \subseteq \mathcal{I}$ — набор. Ассоциативное правило представляет собой импликацию вида $A \to B$, где $A, B \subseteq \mathcal{I}$, $A \neq \emptyset, B \neq \emptyset$, $A \cap B = \emptyset$. Левая часть правила называется *антецедентом*, правая — *консеквентом*. В качестве мер полезности ассоциативных правил для аналитика применяются *поддержка (support)* и *достоверность (confidence)* правила, определяемые следующим образом:

$$\begin{aligned} support(A \to B) &:= P(A \cup B) \\ confidence(A \to B) &:= P(B \mid A). \end{aligned} \tag{1}$$

Устойчивым ассоциативным правилом (strong rule) называют правило, поддержка и достоверность которого не ниже наперед заданных пороговых значений *minsup* и *minconf* соответственно. Поиск устойчивых ассоциативных правил может быть разбит на две последовательно выполняемые задачи: поиск всех частых наборов и генерация устойчивых ассоциативных правил на основе найденных частых наборов [4].

Набор, имеющий поддержку не ниже $minsup$, называют *частым*, иначе набор называют *редким*. *Поддержкой набора* $I \subseteq \mathcal{I}$ является доля транзакций $\mathcal{D}$, содержащих данный набор:

$$support(I) = \frac{|\{T \in \mathcal{D} \mid I \subseteq T.I\}|}{|\mathcal{D}|}. \tag{2}$$

Множество всех частых k-наборов обозначают $\mathcal{L}_k$. Решением задачи *поиска частых наборов* будет множество $\mathcal{L} = \cup_{k=1}^{k_{max}} \mathcal{L}_k$, где k_{max} — максимальное количество объектов в частом наборе.

Классическим алгоритмом решения задачи поиска частых наборов является алгоритм *Apriori* [4]. Идея Apriori заключается в итеративной генерации множества *кандидатов* в частые наборы и последующем отборе кандидатов с подходящим значением поддержки. Итерация осуществляется по k, мощности наборов-кандидатов, начиная с 1. В алгоритме используется следующее свойство *антимонотонности поддержки* (принцип *a priori*), которое позволяет отсеивать заведомо редкие наборы: если k-набор является редким, то содержащий его $(k+1)$-набор также является редким. Узким местом Apriori является операция генерации и проверки наборов-кандидатов, поскольку при достаточно больших значениях k и малых значениях $minsup$ имеют место значительные накладные расходы на поддержку наборов-кандидатов и повторяющиеся операции сканирования множества транзакций и подсчета поддержки. Разработан ряд улучшений алгоритма Apriori, связанных с сокращением количества наборов-кандидатов, количества просматриваемых транзакций и количества операций сканирования: алгоритмы *AprioriTid* [4], *DHP* [65], *Partition* [70], *DIC* [14], *Eclat* [81] и др.

Альтернативой подходу с использованием кандидатов для решения задачи поиска частых наборов является алгоритм *FP-Growth* [32]. На первой фазе FP-Growth за две операции полного сканирования множества транзакций выполняется построение специальной структуры данных, *FP-дерева (FP tree, frequent pattern tree)*, которое в компактном виде хранит наборы и их поддержку. На второй фазе с помощью рекурсивного обхода построенного дерева осуществляется генерация частых наборов. FP-Growth предъявляет большие требования к объему необходимой оперативной памяти.

Улучшением данного подхода являются алгоритмы *AFOPT* [43], *OpportuneProject* [44] и др. В настоящее время отсутствует алгоритм поиска частых наборов, превосходящий все остальные для всех возможных вариаций множества транзакций и порогового значения поддержки $minsup$ [34].

1.2. Задача кластеризации

Задача *кластеризации (clustering)* заключается в разбиении множества объектов сходной структуры на заранее неизвестные группы (кластеры) в зависимости от похожести свойств объектов. Кластеризация применяется в широком спектре приложений: сегментирование медицинских и спутниковых изображений, анализ ДНК-микрочипов и текстов и др.

Формальное определение задачи кластеризации выглядит следующим образом. Пусть заданы конечные множества: $X = \{x_1, x_2, \ldots, x_n\}$, где $n > 1$ — множество объектов d-мерного метрического пространства, для которых задана функция расстояния $\rho(x_i, x_j)$, и $C = \{c_1, c_2, \ldots, c_k\}$, где $k \ll n$ — набор уникальных идентификаторов (номеров, имен, меток) кластеров.

Алгоритм (четкой) кластеризации определяется как функция $\alpha : X \to C$, которая каждому объекту назначает уникальный идентификатор кластера. Алгоритм кластеризации выполняет разбиение множества X на непересекающиеся непустые подмножества (*кластеры*) таким образом, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. *Алгоритм нечеткой кластеризации* позволяет одному и тому же объекту принадлежать одновременно всем кластерам, но с различной степенью принадлежности.

Алгоритм *разделительной (partitioning) кластеризации* предполагает начальное разбиение исходного множества объектов на кластеры (возможно, выполняемое случайным образом), при котором в каждом кластере имеется, по крайней мере, один объект, и каждый объект принадлежит в точности одному кластеру. Далее итеративно осуществляется перемещение объектов между кластерами с целью улучшить начальное разбиение (чтобы объекты из одного кластера были более «близкими», а из разных кластеров — более «далекими» друг другу).

В алгоритме *k-Means* [46] при улучшении разбиения каждый кластер представляется посредством среднего значения координат объектов в кластере. Для представления кластеров в разделительных алгоритмах могут использоваться также медиана или мода координат объектов (алгоритмы *k-Median* [28] и *k-Mode* [36] соответственно).

Алгоритмы *k-Medoids* и *PAM (Partitioning Around Medoids)* [39] в качестве представления каждого кластера используют тот объект подвергаемого кластеризации множества, который находится ближе остальных к центру кластера. Техника медоидов направлена на повышение устойчивости алгоритма к выбросам и шумам в данных (робастности).

Иерархическая кластеризация заключается в последовательном разбиении исходного множества объектов по уровням иерархии. *Агломеративный иерархический алгоритм* начинает работу в предположении, что каждый исходный объект образует отдельный кластер, и затем выполняет слияние близких друг к другу объектов или кластеров до тех пор, пока не будет получен единственный кластер или не будет выполнено условие завершения слияния. Примером агломеративного подхода является алгоритм *AGNES* [39].

Дивизимный иерархический алгоритм, напротив, стартует, предполагая, что все исходные объекты входят в один кластер, и затем итеративно выполняет его разбиение на менее мощные кластеры до тех пор, пока не будут получены кластеры-синглтоны или не будет выполнено условие завершения слияния. Дивизимный подход реализован в алгоритме *DIANA* [39].

Плотностная (density-based) кластеризация предполагает добавление объектов (называемых в контексте плотностных методов точками) в кластер до тех пор, пока плотность (количество) соседних точек не превысит некоторого наперед заданного значения порога концентрации. Плотностная кластеризация используется для нахождения аномалий и кластеров произвольной формы (в отличие от разделительных алгоритмов, которые приспособлены для нахождения кластеров сферической формы). Типичным представителем плотностной кластеризации является алгоритм *DBSCAN* [23], осуществляющий построение кластера как множества точек близкой плотности, которое имеет наибольшую мощность.

Решеточная (grid-based) кластеризация предполагает разбиение пространства исходных данных на конечное число ячеек, формирующих решеточную структуру, над которой выполняются операции, необходимые для кластеризации. Алгоритм *STING* [84] использует статистическую информацию, хранящуюся в прямоугольных ячейках решетки. Статистические данные о ячейках верхних уровней вычисляются на основе статистических данных о ячейках нижних уровней. Для кластеризации используются следующие статистические данные: количество точек в ячейке, минимальное, максимальное, среднее значение атрибутов и др.

2. Подходы к интеграции анализа данных в СУБД

Исследования в области интеграции интеллектуального анализа данных в реляционные системы баз данных начаты в конце XX в., практически одновременно с зарождением интеллектуального анализа данных как самостоятельной научной дисциплины. В работах Агравала (Agrawal) и Сараваджи (Sarawagi) [5, 68], где предложен термин «связывание» (coupling) интеллектуального анализа данных и СУБД. Хан (Han) предложил [30] различать следующие виды интеграции интеллектуального анализа данных в СУБД: слабое связывание, среднее связывание и сильное связывание.

При *слабом связывании (loose coupling)* система интеллектуального анализа данных отделена от СУБД и использует сервисы СУБД для экспорта исходных данных из хранилища и импорта результатов анализа обратно в хранилище данных. Данный подход использует большинство современных открытых систем для интеллектуального анализа данных: KNIME [9], Weka [24] и др.

При *среднем связывании (semitight coupling)* система интеллектуального анализа данных также отделена от СУБД, но применяет СУБД для реализации некоторых примитивных операций, часто используемых при подготовке данных для интеллектуального анализа. В качестве таких операций могут фигурировать индексирование, соединение отношений, построение гистограмм, статистические вычисления (поиск максимума и минимума, стандартного отклонения) и др. Помимо этого СУБД может обеспечивать хранение предварительно вычисленных и часто используемых промежуточных результатов интеллектуального анализа.

При *сильном связывании (tight coupling)* система интеллектуального анализа данных рассматривается как функциональная единица СУБД, которая обеспечивает выполнение запросов пользователя на анализ данных в базе данных, подобно тому как машина баз данных исполняет запросы SQL в приложениях OLTP (оперативной обработки транзакций). В этом случае функции интеллектуального анализа данных реализуются и оптимизируются на основе использования структур данных, схем индексирования и методов обработки запросов, встроенных в СУБД. Сильное связывание предпочтительно с точки зрения удобства прикладного программиста и конечного пользователя, но одновременно является наиболее трудоемким в реализации [30].

В рамках исследования подходов к реализации сильного связывания можно выделить следующие два основных направления работ: исследование методов создания систем анализа данных в СУБД и разработка методов реализации алгоритмов интеллектуального анализа данных на SQL.

Система интеллектуального анализа данных может быть реализована как *внедренная в СУБД подсистема*, которая поддерживает специальный язык аналитических запросов или расширяет SQL соответствующими конструкциями. Машина баз данных при этом модифицируется, чтобы осуществлять разбор, оптимизацию и выполнение запроса.

Реализация системы интеллектуального анализа данных возможна также в виде *медиатора (посредника)* между прикладным программистом баз данных и СУБД. Прикладному программисту предоставляется графический интерфейс или специализированный язык для формирования запросов интеллектуального анализа данных. Медиатор преобразует запросы интеллектуального анализа данных в набор запросов на SQL и/или вызовов хранимых процедур, которые затем исполняет СУБД.

Альтернативой для системного программиста, реализующего внедрение анализа данных в СУБД, является разработка *библиотеки хранимых процедур. Хранимая процедура (stored procedure)* представляет собой текст подпрограммы, компилируемый однократно и постоянно хранимый на сервере базы данных. Хранимая процедура похожа на подпрограммы языков высокого уровня (имеет параметры, локальные переменные и др.), но может возвращать результат запроса SQL. Такая подпрограмма может быть реализована на SQL или его процедурном расширении, либо на языке высокого уровня (эта возможность, как правило, поддерживается в современных СУБД). Подключая библиотеку к своему приложению базы данных, прикладной программист получает возможность выполнять интеллектуальный анализ данных, не выходя за рамки СУБД.

Реализация на SQL алгоритмов интеллектуального анализа данных предполагает, что исходные и промежуточные данные алгоритма, а также результаты его работы будут представлены в виде реляционных таблиц. Обработка указанных таблиц реализуется посредством запросов SQL, что обеспечивает потенциальную переносимость алгоритмов на другие реляционные СУБД. Однако, поскольку SQL является декларативным языком запросов, разработка в нем алгоритмов анализа данных сопряжена с определенными трудностями. Например, в SQL затруднена реализация структур данных в оперативной памяти (список, бинарное дерево, граф и др.) и агрегации данных по столбцам таблицы (штатные функции SQL `COUNT`, `MIN`, `MAX`, `AVG` выполняют только построчную агрегацию).

Одним из основных путей преодоления подобных проблем является разработка пользовательских функций, расширяющих штатные возможности SQL. *Пользовательская функция (user-defined function, UDF)* представляет собой хранимую на сервере баз данных

подпрограмму-функцию, вызов которой может быть включен в качестве выражения в оператор SQL, а результат вычисляется в рамках выполнения соответствующего запроса. Пользовательская функция допускает результат как скалярного, так и табличного типа. Реализация пользовательской функции может быть выполнена на SQL, процедурном расширении SQL либо на языке высокого уровня. Например, в работе [58] описан подход к реализации пользовательских функций, выполняющих агрегатные операции по столбцам реляционных таблиц.

Помимо расширения функциональности SQL, пользовательские функции могут в общем случае более эффективно, чем штатные средства СУБД, реализовать операции агрегации, математический вычисления и др. (за счет возможности уменьшить количество операций сканирования таблиц, переноса части вычислений в оперативную память и др.) [58, 59, 62]. Например, в работе [62] предложена реализация метода главных компонент в параллельной СУБД на основе пользовательских функций, использующих библиотеку параллельных подпрограмм Intel Math Kernel Library; в работе [60] предложен способ ускорения вычисления Байесовской модели для линейной регрессии на основе использования параллельных пользовательских функций. Следует также отметить, что обратной стороной подобного увеличения эффективности пользовательских функций является возможная потеря переносимости алгоритмов анализа данных в другие СУБД.

3. Методы разработки систем анализа данных в СУБД

В данном разделе рассмотрены заметные научные исследования в области разработки систем и библиотек анализа данных в СУБД, а также расширений языка баз данных SQL.

3.1. Системы и библиотеки анализа данных

```
select
  PredictAssociation([HealthMiningModel].[AssocLines], INCLUDE_STATISTICS, 3)
  from [HealthMiningModel]
    natural prediction join (
    select
      60 as [Age],
      TRUE as [isSmoker],
      'Pneumonia' as [Disease]) as [AssocLines]
```

Рис. 1. Пример запроса на языке DMX

В корпорации Microsoft разработаны стандарт OLE DB for Data Mining и язык запросов *DMX (Data Mining Extensions)* [78], используемые в ее продукте MS SQL Server Analysis Services. Стандарт специфицирует интерфейс программирования приложений (Application Programming Interface, API) интеллектуального анализа данных. Язык DMX имеет SQL-подобный синтаксис (операторы определения и манипулирования данными и др.), однако его операндами являются не реляционные отношения, а модели интеллектуального анализа данных. Под моделью интеллектуального анализа данных понимается сочетание самих данных, алгоритма интеллектуального анализа данных и коллекции значений параметров и фильтров, управляющих использованием и обработкой данных. Пример запроса на языке DMX показан на рис. 1.

```
DBMS_DATA_MINING.CREATE_MODEL (
  model_name           => 'credit_risk_model',
  function             => DBMS_DATA_MINING.classification,
  data_table_name      => 'credit_card_data',
  case_id_column_name => 'customer_id',
  target_column_name   => 'credit_risk',
  settings_table_name => 'credit_risk_model_settings');

select customer_name
from credit_card_data
where PREDICTION (credit_risk_model using *) = 'LOW'
  and customer_value = 'HIGH'
```

Рис. 2. Пример запроса на языке Oracle Data Mining

Подобный подход реализован также в коммерческой СУБД Oracle в виде модуля Oracle Data Mining [77]. На рис. 2 приведен пример создания модели классификации и запроса к ней.

Ванг (Wang) и др. разработали систему интеллектуального анализа данных *ATLAS* [83], которая поддерживает одноименный язык запросов, являющийся надстройкой над SQL. Язык ATLAS добавляет в SQL поддержку пользовательских функций и функций, возвращающих в качестве значения реляционную таблицу. На языке ATLAS реализованы алгоритм поиска шаблонов Apriori, алгоритм кластеризации DBSCAN [23] и классификация посредством деревьев решений.

```
-- Кластеризация k средних
kmeanspp(
  rel_source, -- имя таблицы с входными данными
  expr_point, -- имя колонки с данными
  k, -- количество искомых кластеров
  fn_dist, -- вид функции расстояния
  agg_centroid, -- вид агрегационной функции при расчете центроидов
  max_num_iterations, -- максимальное количество итераций
  min_frac_reassigned, -- минимальное количество переназначаемых объектов
  для останова вычислений
  seeding_sample_ratio) -- размер сэмпла данных для инициализации центроидов

select * from madlib.kmeanspp(
 'km_sample', 'points', 2, 'madlib.squared_dist_norm2',
 'madlib.avg', 20, 0.001);
```

Рис. 3. Пример функции библиотеки MADlib

Хеллерштейн (Hellerstein) и др. разработали библиотеку *MADlib* [33] с открытым исходным кодом для интеллектуального анализа данных в реляционных СУБД PostgreSQL и Greenplum. MADlib предоставляет богатый набор алгоритмов анализа данных (кластеризация, классификация, регрессия и др.), адаптированные для использования в реляционной СУБД и не требующие экспорта и импорта данных внешних аналитических приложений. В реализации MADlib используются пользовательские функции, написанные разработчиками на язык программирования Python, которые обеспечивают обращения к словарю базы данных и формирование корректной структуры таблиц с выходными

данными для заданных таблиц с входными данными. Пример интерфейса и вызова функции библиотеки MADlib приведен на рис. 3.

```
-- Объекты
create table Items (
  item integer primary key, -- Уникальный ИД объекта
  description varchar)     -- Описание объекта

-- Транзакции
create table D (
  tid integer,  -- Уникальный ИД транзакции
  item integer, -- Уникальный ИД объекта в транзакции
  pimary key (tid,item),
  foreign key item references Items (item))

-- Наборы
create table Sets (
  sid integer,  -- Уникальный ИД набора
  item integer, -- Уникальный ИД объекта в наборе
  pimary key (sid,item),
  foreign key item references Items (item))

-- Поддержка наборов
create table Support (
  sid integer pimary key, -- Уникальный ИД набора
  supp real)         -- Поддержка набора

-- Ассоциативные правила
create table Rules (
  rid integer,  -- Уникальный ИД правила
  sida integer, -- Уникальный ИД набора-антецедента
  sidc integer, -- Уникальный ИД набора-консеквента
  sid integer,  -- Уникальный ИД набора-объединения антецедента и консеквента
  supp, conf real -- Поддержка и достоверность правила
  foreign key sida references Sets (sid),
  foreign key sidc references Sets (sid),
  foreign key sid references Sets (sid))
```

Рис. 4. База данных для виртуальных представлений поиска шаблонов

В цикле работ [11–13] Блокилом (Blockeel), Гоэталсом (Goethals) и др. предложена система интеллектуального анализа данных в СУБД на основе виртуальных аналитических представлений. *Виртуальное аналитическое представление (virtual mining view)* создается как именованный запрос к таблицам базы данных и другим представлениям, который обеспечивает логическое хранение (в отличие от физического хранения таблиц базы данных) результатов интеллектуального анализа данных. При выполнении запроса пользователя к такому представлению в СУБД срабатывает системный триггер, который запускает алгоритм интеллектуального анализа данных. Далее СУБД материализует кортежи, запрошенные пользователем. Система поддерживает построение виртуальных аналитических представлений для поиска шаблонов и классификации с помощью деревьев решений. На рис. 4 приведена схема базы данных, используемой для построения виртуальных аналитических представлений поиска шаблонов. Данное исследование сконцентрировано, однако, на частичной или полной материализации запрошенных

пользователем результатов анализа и не затрагивает вопрос исполнения аналитических алгоритмов внутри СУБД.

В работе [61] Ордонезом и др. предложена облачная система интеллектуального анализа данных на основе реляционной СУБД. На локальной машине запускается реляционная СУБД, подключающаяся к облаку. База данных хранится и обрабатывается в облаке, а в локальную СУБД передаются только результаты анализа. Помимо возможностей обработки данных только на локальной машине или только в облаке, система поддерживает режим гибридного исполнения, когда выполняется распределение вычислительной нагрузки между облаком и локальной СУБД.

Ордонезом и др. также разработана система интеллектуального анализа данных, основанный на использовании реляционной СУБД и хранимых процедур [59]. Технологический цикл работы с системой выглядит следующим образом. Анализируемые данные, параметры аналитических алгоритмов и проч. хранится в в реляционных таблицах. Клиентское приложение соединяется с сервером СУБД по протоколу ODBC. С помощью графического интерфейса конечный пользователь специфицирует задачу интеллектуального анализа данных, ее параметры и таблицы исходных данных. Приложение запускает хранимую процедуру, которая, в свою очередь, выполняет генерацию необходимых SQL запросов. Вычислительно трудоемкие операции (например, вычисления, связанные с матрицами) выполняются с помощью предварительно созданных и откомпилированных соответствующих пользовательских функций. Графический интерфейс позволяет выполнять мониторинг выполнения аналитического алгоритма (время, количество итераций и др.) и последующую визуализацию результатов.

Махаян (Mahajan) и др. разработали систему анализа DAnA [47], которая выполняет автоматическое преобразование запросов на выполнение анализа данных в исходный код для выполнения на реконфигурируемых вычислительных системах FPGA. Реализация данного преобразования выполняется с помощью пользовательской функции на SQL, использующей язык Python. Система DAnA предполагает интеграцию в СУБД на основе специализированных аппаратных устройств, называемых *страйдерами (striders)*. Страйдер имеет прямой интерфейс доступа к буферному пулу СУБД и выполняет извлечение, очистку и обработку кортежей данных, которые затем передаются на ускоритель FPGA для параллельного исполнения аналитического алгоритма. Использование FPGA позволяет ускорить вычисления ценой, однако, потери переносимости разработанного решения, поскольку требует включения в состав системы специализированных аппаратных устройств (страйдеров).

В работе [2] Речкалов описал систему поиска шаблонов, реализованную в СУБД на основе предложенного языка XML-разметки алгоритмов поиска частых наборов, реализуемых на SQL. Разработана разметка алгоритмов ScanOnce [82] и SETM [35]. Используя разметку, система выполняет автоматическую генерацию хранимых процедур на языке SQL в зависимости от специфицированных пользователем таблиц исходных данных и параметров алгоритма. Среди полученных SQL реализаций система выбирает для исполнения наиболее эффективный, используя имеющуюся в составе современных СУБД команду `EXPLAIN`, которая позволяет получить стоимость (относительную оценку времени исполнения) запроса SQL без его фактического выполнения.

3.2. Языки запросов и расширения SQL для интеллектуального анализа данных

```
find association rules as HealthRuleSet
related to Salary, Age, isSmoker, Disease
from HealthDB
where Disease='Pneumonia' and Age>60
with support threshold=0.05
with confidence threshold=0.07
```

Рис. 5. Пример запроса на языке *DMQL*

Одним из первых языков интеллектуального анализа данных можно является язык *DMQL* [29], предложенный в 1996 г. Хан (Han) и др. DMQL предоставляет SQL-подобный синтаксис для записи запросов интеллектуального анализа данных. Примитивы DMQL позволяют определить данные, подлежащие анализу, решаемую задачу интеллектуального анализа (классификация, поиск ассоциативных правил и др.), семантические иерархии в анализируемых данных и пороговые значения параметров задачи (поддержка и др.). Пример запроса на языке *DMQL* приведен на рис. 5. Язык запросов DMQL был реализован в рамках системы анализа данных в СУБД DBMiner [29].

Язык DMQL позднее послужил основой для разработки целого ряда языков запросов интеллектуального анализа данных: язык для анализа временных данных *TQML* [20] Чена (X. Chen), 1998 г.; языки для анализа географических данных *GMQL* [31] Хана (Han), 1997 г. и *SDMQL* [49] Малербы (Malerba), 2004 г.; язык для анализа пространственно-временных данных *ST-DMQL* [15] Богорны (Bogorny), 2009 г.

```
mine rule HealthRuleSet as
  select distinct 1..n Disease as body,
  1..1 isSmoker as head
  from HealthDB
  where body.Disease='Pneumonia' and body.Age>60
  extracting rules with
  support: 0.1
  confidence: 0.3
```

Рис. 6. Пример запроса на языке MINE RULE

Meo (Meo) и др. в 1996 г. предложили SQL-подобный оператор *MINE RULE* [51], который предназначен для решения задачи поиска ассоциативных правил. Пример запроса с использованием оператора *MINE RULE* показан на рис. 6.

Позднее в 1999 г. в работе [37] Имилински (Imielinski) описал язык *MSQL*, представляющий собой расширение SQL для решения задачи поиска ассоциативных правил. В отличие от DMQL, язык MSQL предполагает не только нахождение ассоциативных правил, но и предоставляет возможность выборки результирующих правил. Соответствующие примеры запросов приведены на рис. 7.

Следует отметить, что описанные выше расширения языка баз данных не позволяют явно манипулировать полученными результатами интеллектуального анализа данных (подобно тому, как это обеспечивается в SQL). В этом смысле интересной является

```
GetRules(HealthDB)
  into HealthRuleSet R
  where R.Body in {(Disease=*), (Age=*), (Salary=*)}
  and R.Body has {(Disease='Pneumonia'), (Age>60)}
  and R.Consequent in {(isSmoker=*)}
  and Support>0.1
  and Confidence>0.7

SelectRules(HealthRuleSet)
  where Body has {(Disease='Pneumonia')}
  and {(Salary>0) and (Salary<=1000)}
  and Support>0.1
  and Confidence>0.7
```

Рис. 7. Пример запроса на языке MSQL

работа [17] Калдерса (Calders) и др., в которой предложены специализированные модель базы данных и алгебра для интеллектуального анализа данных.

```
select ST_Area(ST_Polygon(House.location))
from House
where House.household_income < 30000
cluster by House.location
```

Рис. 8. Пример запроса с оператором `CLUSTER BY`

Сан (Sun) и др. в работе [76] предложили расширить язык SQL оператором *`CLUSTER BY`* для кластеризации данных. Данная конструкция подразумевает выполнение группировки строк результата запроса в соответствии со специфицированным алгоритмом кластеризации, в отличие от стандартного оператора `GROUP BY`, который осуществляет группировку по точному совпадению значений в полях записей. На рис. 8 приведен пример использования указанного оператора в запросе, который задействует PostGIS [45], расширение СУБД PosgreSQL, обеспечивающее поддержку географических объектов. Силва (Silva) и др. в работе [73] предложили схожий по назначению оператор *`SIMILAR GROUP BY`*, реализованный авторами в СУБД PostgreSQL.

4. Реализация алгоритмов анализа данных на SQL

Реализация алгоритмов интеллектуального анализа данных в виде набора запросов SQL обеспечивает переносимость между различными СУБД. Далее приведен обзор работ, описывающих применение данного подхода для решения различных задач интеллектуального анализа данных.

4.1. Задача поиска шаблонов

Одной из первых SQL-реализаций задачи поиска частых наборов является алгоритм *SETM* [35], разработанный Хутсмой (Houtsma) и др. в 1995 г. В основе алгоритма лежит использование запросов SQL, выполняющих поиск частых наборов без использования принципа *a priori*, приведенных на рис. 9 (здесь и далее используются обозначения из раздела 1.1, определение таблицы D приведено на рис. 4). В 2000 г. Йосизава (Yoshizawa) и др. работе [85] предложили улучшение данного алгоритма, предполагающее использование

```
insert into L_k
  select d_1.item, ..., d_k.item, count(*)
  from L_{k-1} l, D d_1, ..., D d_k
  where d_1.tid= ... = d_k.tid and
    d_1.item = l.item_1 and
    ...
    d_{k-l}.item = l.item_{k-1} and
    d_k.item > d_{k-1}.item
  group by d_1.item, ..., d_k.item
  having count(*) >= :minsup
```

Рис. 9. Запросы SQL для поиска частых наборов в алгоритме SETM [35]

представлений вместо некоторых таблиц, и рефакторинг запросов на основе использования подзапросов.

```
insert into C_k
  select l_1.item_1, ..., l_1.item_{k-1}, l_2.item_{k-1}
  from L_{k-1} l_1, L_{k-1} l_2
  where l_1.item_1=l_2.item_1 and
    ...
    l_1.item_{k-2}=l_2.item_{k-2} and
    l_1.item_{k-1}<l_2.item_{k-1}
```

Рис. 10. Запросы SQL для генерации наборов-кандидатов в алгоритмах из работы [68]

В 1998 г. Сараваджи (Sarawagi) и др. [68] предложили алгоритмы *K-Way-Join*, *Three-Way-Join*, *Subquery* и *Two-Group-Bys*, которые основаны на классическом алгоритме Apriori [4] для оперативной памяти. SQL-реализация генерации наборов-кандидатов в указанных алгоритмах представлена на рис. 10.

```
insert into L_k
  select C_k.item_1, ..., C_k.item_k, count(*)
  from C_k, D d_1, ..., D d_k
  where d_1.item_1=C_k.item_1 and
    ...
    d_k.item_k=C_k.item_k and
    d_1.tid=d_2.tid and
    ...
    d_{k-1}.tid=d_k.tid
  group by C_k.item_1, ..., C_k.item_k
  having count(*) >= :minsup
```

Рис. 11. Запросы SQL для поиска частых наборов в алгоритме K-Way-Join [68]

Алгоритмы отличаются способом вычисления поддержки наборов-кандидатов. На рис. 11 представлен способ вычисления поддержки в алгоритме K-Way-Join. В алгоритме Three-Way-Join (см. рис. 12) для снижения количества затратных операций естественного соединения используется следующая модификация. В дополнение к полям (item_1, ..., item_k) в таблицу C_k добавляются три новых поля: (oid, id_1, id_2), где oid — уникальный идентификатор набора, а id_1 и id_2 — уникальные идентификаторы тех наборов из $\mathcal{L}_{k-1}$, которые использованы при создании данного набора. Кроме того, на k-м просмотре

базы транзакций создается дополнительная таблица T_k с полями (tid, oid), которая для каждого идентификатора транзакции tid хранит каждый идентификатор oid набора из C_k, входящего в транзакцию.

```
-- Создание дополнительной таблицы
insert into Tk
  select t1.tid, oid
  from Ck, Tk-1 t1, Tk-1 t2
  where t1.oid=Ck.id1 and t2.oid=Ck.id2 and t1.tid=t2.tid

-- Подсчет поддержки с помощью дополнительной таблицы
insert into Lk
  select Ck.oid, Ck.item1, ..., Ck.itemk, cnt
  from Ck, (
    select oid as cid, count(*) as cnt
    from Ck
    group by oid
    having count(*) >= :minsup)
  where Ck.oid=cid
```

Рис. 12. Запросы SQL для поиска частых наборов в алгоритме Three-Way-Join [68]

Томас (Thomas) и Чакраварти (Chakravarthy) в 1999 г. предложили алгоритм *Set-oriented Apriori* [79], в основе которого лежит идея сокращения вычислений при подсчете поддержки наборов за счет построения на каждом шаге подсчета дополнительной таблицы для хранения транзакций, содержащих соответствующее количество объектов. Эксперименты показали, что алгоритм Set-oriented Apriori показывает лучшую производительность, чем алгоритм Subquery.

Ранцау (Rantzau) в 2004 г. разработал алгоритм *Quiver* [66] для решения задачи поиска частых наборов, основанный на предложенной им с коллегами реляционной операции универсального квантования (universal quantification) [67]. Данная операция является аналогом операции деления в реляционной алгебре. Эксперименты на разработанном авторами исполнителе запросов, который поддерживает новую операцию, показывают, что алгоритм Quiver способен показать лучшую производительность, чем другие алгоритмы, реализованные на SQL. В то же время при выполнении на существующих коммерческих СУБД алгоритм Quiver заметно проигрывает в производительности другим алгоритмам, реализованным на SQL, поскольку в данных СУБД отсутствует эффективная реализация предложенной авторами операции.

В работе [82] Ванг (Wang) и др. описали алгоритм *ScanOnce* на языке PL/SQL СУБД Oracle, использующий курсоры. *Курсор* представляет собой указатель на область памяти, в которой хранится результат выполнения запроса, и позволяет осуществлять последовательное сканирование этого результата. В данном алгоритме организуется цикл по уникальным идентификаторам наборов, на каждом шаге которого создается курсор, указывающий на объекты, входящие в соответствующий набор. Далее осуществляется последовательное сканирование курсора и подсчет поддержки набора.

Алашкур (Alashqur) в 2010 г. разработал алгоритм *RDB-MINER* [7] поиска ассоциативных правил в реляционных таблицах. Алгоритм основан на классическом алгоритме Apriori и использовании динамически формируемых запросов SQL.

Алгоритм *Propad* [71], предложенный Шангом (Shang) и др. в 2004 г., представляет собой реализацию классического алгоритма FP-Growth [32] на SQL. Подобно классическому алгоритму, в данном алгоритме не выполняется затратная операция генерации наборов-кандидатов. Алгоритм Propad демонстрирует лучшую производительность, чем алгоритм K-Way-Join (подобно тому, как классический алгоритм FP-Growth является более производительным, чем классический алгоритм Apriori [4]). Позднее Сидло (Sidló) и Лукаш (Lukács) разработали алгоритм *FP-TDG* [72], который также представляет собой SQL-реализацию классического алгоритма FP-Growth [32].

4.2. Задача кластеризации

Ордонезом (Ordonez) в работах [55, 56] на языке SQL реализован разделительный алгоритм кластеризации k-Means [46]. Им же в сотрудничестве с коллегами в работах [48, 57] выполнена реализация на SQL алгоритма кластеризации EM [26]. В работе [1] Миниахметов и др. представили реализацию алгоритма нечеткой кластеризации данных Fuzzy C-Means [10] в СУБД PostgreSQL.

В указанных работах используется следующая техника индексного представления матричных данных. Пусть требуется организовать хранение матрицы кластеризуемых объектов $X \in \mathbb{R}^{n\times d}$. Естественным способом хранения будет таблица с заголовком $(x_1, x_2, \ldots, x_d)$, где каждое поле x_i имеет вещественный тип. Однако в этом случае отсутствует возможность применения агрегатных функций SQL для вычислений, выполняемых по столбцам таблицы (например, подсчет функции ρ расстояния между объектами), поскольку указанные функции осуществляют агрегацию только по строкам. Поэтому матрица X представляется в виде таблицы из $n{\cdot}d$ записей, которая имеет заголовок $(\underline{i,\ell}, val)$, где поля i и ℓ имеют целочисленный тип, а val — вещественный. В составном первичном ключе (i, ℓ) таблицы поле i указывает номер исходного объекта, поле ℓ — номер координаты этого объекта; поле val таблицы хранит значения координат объектов.

В работе [42] Лепиниоти (Lepinioti) разработал алгоритм иерархической кластеризации *Cobweb/IDX*. Реализация выполнена на языке PL/SQL для СУБД Oracle. Данный алгоритм является инкрементальным (поддерживает кластеризацию по мере появления новых данных).

В работе [64] Пан и др. представили подход к кластеризации вершин графа с помощью параллельной СУБД. Спроектирована реляционная база данных для хранения исходных и промежуточных данных алгоритма. Граф представляется в виде реляционной таблицы со списком ребер. Таблицы базы данных подвергаются горизонтальной фрагментации, полученные фрагменты распределяются по узлам вычислительного кластера. Каждый фрагмент обрабатывается отдельно экземпляром параллельной СУБД для получения таблицы вершин графа с метками кластеров. Обработка выполняется с помощью запросов SQL, реализующих стадии огрубления и восстановления графа в соответствии с алгоритмом кластеризации графа в оперативной памяти Кариписа—Кумара [38].

4.3. Другие задачи интеллектуального анализа данных

Помимо рассмотренных выше задач поиска шаблонов и кластеризации, SQL также используется для решения задачи классификации. Классификация похожа на задачу кластеризации в том, что ставит своей целью распределение по группам (классам) конечного числа объектов, имеющих сходную структуру в зависимости от похожести

их свойств. Отличие заключается в том, что в задаче классификации количество и семантика классов известны заранее. Одним из основных подходов к классификации является построение *дерева решений (decision tree)* [16]. Дерево решений представляет собой ориентированное дерево, в котором каждой внутренней вершине соответствует операция проверки значения указанного атрибута классифицируемых объектов, каждая дуга соответствует переходу к другой вершине в соответствии с результатом проверки, а каждому листу соответствует один из классов. Классификация на основе деревьев решений реализована в работах Саттлера и др. [69] и Ковальского и др. [41], а также Моертини (Moertini) и др. [52] для объектно-реляционных СУБД.

Несмотря на «нереляционную» природу графовых данных, внедрение интеллектуального анализа графов в реляционные СУБД является одним из актуальных направлений исследований. Падманабхан (Padmanabhan) и др. в работе [63] предложили подход к анализу структуры графов на основе использования SQL. Срихари (Srihari) и др. в работе [74] описали подход к поиску полного подграфа неориентированного графа, основанный на применении реляционной СУБД. Алгоритмы поиска часто встречающихся подграфов в графе, ориентированные на использование SQL, предложены Чакраварти (Chakravarthy) и др. и Ордонезом (Ordonez) и др. в работах [18] и [27] соответственно. Исследования, направленные на поиск циклов в графе с помощью реляционной СУБД, описаны Балачандраном (Balachandran) и др. в работе [8]. МакКафри (McCaffrey) разработал комбинированный подход к разбиению графов, использующий встраивание запросов SQL в реализацию алгоритма обработки графа на языке программирования высокого уровня [50].

Заключение

В настоящее время реляционные СУБД являются основным инструментом управления базами данных, несмотря на появление большого количества NoSQL СУБД. Одним из перспективных направлений развития реляционных СУБД является внедрение в них средств интеллектуального анализа данных. Интеллектуальный анализ данных направлен на извлечение доступных для понимания знаний, необходимых для принятия решений в различных сферах человеческой деятельности. Интеграция позволяет как избежать накладных расходов по экспорту анализируемых данных из хранилища и импорту результатов анализа обратно в хранилище, так и использовать при анализе данных системные сервисы, заложенные в архитектуре СУБД.

В статье представлен обзор методов и подходов к решению задачи интеграции интеллектуального анализа данных в реляционные СУБД. Наиболее предпочтительным с точки зрения удобства прикладного программиста и конечного пользователя является подход, предполагающий сильное связывание СУБД и технологий интеллектуального анализа данных. В этом случае система интеллектуального анализа данных рассматривается как функциональная единица СУБД, которая обеспечивает прозрачное выполнение запросов пользователя на анализ данных, хранимых в базе данных (подобно тому как машина баз данных исполняет запросы SQL в приложениях оперативной обработки транзакций). Функции интеллектуального анализа данных реализуются и оптимизируются на основе использования структур данных, схем индексирования и методов обработки запросов, встроенных в СУБД.

Рассмотрены системы интеллектуального анализа данных, которые реализуются как внедряемые в СУБД подсистемы, поддерживающие специальный язык запросов анализа данных или расширяющие SQL соответствующими конструкциями. Машина баз данных такой СУБД модифицируется, чтобы осуществлять разбор, оптимизацию и выполнение запросов анализа данных. Представлены расширения языка баз данных SQL, обеспечивающие синтаксическую поддержку интеллектуального анализа данных в СУБД.

Представлены системы интеллектуального анализа данных, реализуемые в виде медиатора (посредника) между прикладным программистом баз данных и СУБД. Прикладному программисту предоставляется графический интерфейс или специализированный язык для формирования запросов анализа данных. Медиатор преобразует запросы анализа данных в набор запросов на SQL и/или вызовов хранимых процедур, которые затем исполняет СУБД.

Приведены примеры внедрения анализа данных в СУБД на основе разработки библиотеки хранимых процедур. Хранимая процедура представляет собой текст подпрограммы, компилируемый однократно и постоянно хранимый на сервере базы данных. Хранимая процедура похожа на подпрограммы языков высокого уровня, но может возвращать результат запроса SQL. Подключая библиотеку к своему приложению базы данных, прикладной программист получает возможность выполнять интеллектуальный анализ данных, не выходя за рамки СУБД.

Дан краткий обзор основных основных задач интеллектуального анализа данных и алгоритмов их решения, рассмотрены примеры реализации указанных алгоритмов на SQL.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (грант № 17-07-00463), Правительства РФ в соответствии с Постановлением № 211 от 16.03.2013 (соглашение № 02.A03.21.0011) и Министерства образования и науки РФ (государственное задание 2.7905.2017/8.9).

Литература

1. Миниахметов Р.М., Цымблер М.Л. Интеграция алгоритма кластеризации Fuzzy c-Means в PostgreSQL // Вычислительные методы и программирование: Новые вычислительные технологии. 2012. Т. 13. С. 46–52.

2. Речкалов Т.В. Подход к интеграции интеллектуального анализа данных в реляционную СУБД на основе генерации текстов хранимых процедур // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2013. Т. 2, № 1. С. 114–121.

3. Agrawal R., Ailamaki A., Bernstein P.A. et al. The Claremont Report on Database Research // Commun. ACM. 2009. Vol. 52, No. 6. P. 56–65. DOI: 10.1145/1516046.1516062.

4. Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules in Large Databases // VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago de Chile, Chile. 1994. P. 487–499.

5. Agrawal R., Shim K. Developing Tightly-coupled Data Mining Applications on a Relational Database System // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. P. 287–290.

6. Abadi D., Agrawal R., Ailamaki A. et al. The Beckman Report on Database Research // Commun. ACM. 2016. Vol. 59, No. 2. P. 92–99. DOI: 10.1145/2845915.

7. Alashqur A. RDB-MINER: A SQL-Based Algorithm for Mining True Relational Databases // Journal of Software. 2010. Vol. 5, No. 9. P. 998–1005. DOI: 10.4304/jsw.5.9.998-1005.

8. Balachandran R., Padmanabhan S., Chakravarthy S. Enhanced DBSubdue: Supporting Subtle Aspects of Graph Mining Using a Relational Approach // Advances in Knowledge Discovery and Data Mining, 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9–12, 2006, Proceedings. 2006. P. 673–678. DOI: 10.1007/11731139_77.

9. Berthold M.R., Cebron N., Dill F. et al. KNIME - the Konstanz Information Miner: Version 2.0 and Beyond // SIGKDD Explorations. 2009. Vol. 11, No. 1. P. 26–31. DOI: 10.1145/1656274.1656280.

10. Bezdek J.C., Ehrlich R., Full W. FCM: The Fuzzy C-Means Clustering Algorithm // Computers and Geosciences. 1984. Vol. 10, No. 2. P. 191–203. DOI: 10.1016/0098-3004(84)90020-7.

11. Blockeel H., Calders T., Fromont E. et al. An Inductive Database Prototype Based on Virtual Mining Views // Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008. 2008. P. 1061–1064. DOI: 10.1145/1401890.1402019.

12. Blockeel H., Calders T., Fromont E. et al. An Inductive Database Prototype Based on Virtual Mining Views // Data Min. Knowl. Discov. 2012. Vol. 24, No. 1. P. 247–287. DOI: 10.1007/s10618-011-0229-7.

13. Blockeel H., Calders T., Fromont E. et al. Inductive Querying with Virtual Mining Views // Inductive Databases and Constraint-Based Data Mining. Ed. by S. Dzeroski, B. Goethals, P. Panov. Springer, 2010. P. 265–287. DOI: 10.1007/978-1-4419-7738-0_11.

14. Brin S., Motwani R., Ullman J.D., Tsur S. Dynamic Itemset Counting and Implication Rules for Market Basket Data // SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13–15, 1997, Tucson, Arizona, USA. 1997. P. 255–264. DOI: 10.1145/253260.253325.

15. Bogorny V., Kuijpers B., Alvares L.O. ST-DMQL: A Semantic Trajectory Data Mining Query Language // International Journal of Geographical Information Science. 2009. Vol. 23, No. 10. P. 1245–1276.

16. Breiman L., Friedman J., Olshen R., Stone C. Classification and Regression Trees. Wadsworth International Group, 1984.

17. Calders T., Lakshmanan L.V.S., Ng R.T., Paredaens J. Expressive Power of an Algebra for Data Mining // ACM Trans. Database Syst. 2006. Vol. 31, No. 4. P. 1169–1214. DOI: 10.1145/1189769.1189770.

18. Chakravarthy S., Pradhan S. DB-FSG: An SQL-based Approach for Frequent Subgraph Mining // Database and Expert Systems Applications, 19th International Conference, DEXA 2008, Turin, Italy, September 1–5, 2008. Proceedings. 2008. P. 684–692. DOI: 10.1007/978-3-540-85654-2_59.

19. Chaudhuri S. What Next?: a Half-dozen Data Management Research Goals for Big Data and the Cloud // Proceedings of the 31st ACM SIGMODSIGACT- SIGART Symposium on

Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20–24, 2012. 2012. P. 1–4. DOI: 10.1145/2213556.2213558.

20. Chen X., Petrounias I. Language Support for Temporal Data Mining // Principles of Data Mining and Knowledge Discovery, 2nd European Symposium, PKDD '98, Nantes, France, September 23–26, 1998, Proceedings. 1998. P. 282–290. DOI: 10.1007/BFb0094830.

21. Codd E.F. A Relational Model of Data for Large Shared Data Banks // Commun. ACM. 1970. Vol. 13, No. 6. P. 377–387. DOI: 10.1145/362384.362685.

22. Davoudian A., Chen L., Liu M. A Survey on NoSQL Stores // ACM Comput. Surv. 2018. Vol. 51, No. 2. P. 40:1–40:43. DOI: 10.1145/3158661.

23. Ester M., Kriegel H., Sander J., Xu X. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. P. 226–231.

24. Frank E., Hall M.A., Holmes G. et al. WEKA - A Machine Learning Workbench for Data Mining // The Data Mining and Knowledge Discovery Handbook. / Ed. by O. Maimon, L. Rokach. Springer, 2005. P. 1305–1314.

25. Frawley W.J., Piatetsky-Shapiro G., Matheus C.J. Knowledge Discovery in Databases: an Overview // Knowledge Discovery in Databases. AAAI/MIT Press, 1991. P. 1–30.

26. Dempster A., Laird N., Rubin D. Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm // Journal of The Royal Statistical Society. 1977. Vol. 39, No. 1. P. 1–38.

27. Garcia W., Ordonez C., Zhao K., Chen P. Efficient Algorithms Based on Relational Queries to Mine Frequent Graphs // Proceedings of the 3rd PhD Workshop on Information and Knowledge Management, PIKM 2010, Toronto, Ontario, Canada, October 30, 2010. 2010. P. 17–24. DOI: 10.1145/1871902.1871906.

28. Guha S., Mishra N., Motwani R., O'Callaghan L. Clustering Data Streams // Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12–14 November 2000, Redondo Beach, California, USA. 2000. P. 359–366. DOI: 10.1109/SFCS.2000.892124.

29. Han J., Fu Y., Wang W. et al. DBMiner: A System for Mining Knowledge in Large Relational Databases // Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. P. 250–255.

30. Han J., Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006. P. 743.

31. Han J., Koperski K., Stefanovic N. GeoMiner: A System Prototype for Spatial Data Mining // SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13–15, 1997, Tucson, Arizona, USA. 1997. P. 553–556. DOI: 10.1145/253260.253404.

32. Han J., Pei J., Yin Y. Mining Frequent Patterns without Candidate Generation // Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16–18, 2000, Dallas, Texas, USA. 2000. P. 1–12. DOI: 10.1145/342009.335372.

33. Hellerstein J.M., Re C., Schoppmann F. et al. The MADlib Analytics Library or MAD Skills, the SQL // PVLDB. 2012. Vol. 5, No. 12. P. 1700–1711.

34. HooshSadat M., Samuel H.W., Patel S., Zaiane O.R. Fastest Association Rule Mining Algorithm Predictor (FARM-AP) // Proceedings of the 4th International C* Conference on Computer Science and Software Engineering, C3S2E 2011, Montreal, Quebec, Canada, May 16–18, 2011. P. 43–50. DOI: 10.1145/1992896.1992902.

35. Houtsma M.A.W., Swami A.N. Set-Oriented Mining for Association Rules in Relational Databases // Proceedings of the 11th International Conference on Data Engineering, March 6–10, 1995, Taipei, Taiwan. 1995. P. 25–33. DOI: 10.1109/ICDE.1995.380413.

36. Huang Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values // Data Min. Knowl. Discov. 1998. Vol. 2, No. 3. P. 283–304. DOI: 10.1023/A:1009769707641.

37. Imielinski T., Virmani A. MSQL: A Query Language for Database Mining // Data Min. Knowl. Discov. 1999. Vol. 3, No. 4. P. 373–408. DOI: 10.1023/A:1009816913055.

38. Karypis G., Kumar V. Analysis of Multilevel Graph Partitioning // Proceedings of Supercomputing '95, San Diego, CA, USA, December 4–8, 1995. 1995. P. 29. DOI: 10.1145/224170.224229.

39. Kaufman L., Rousseeuw P.J. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley, 1990. DOI: 10.1002/9780470316801.

40. Krause C., Johannsen D., Deeb R. et al. An SQL-Based Query Language and Engine for Graph Pattern Matching // Graph Transformation - 9th International Conference, ICGT 2016, in Memory of Hartmut Ehrig, Held as Part of STAF 2016, Vienna, Austria, July 5–6, 2016, Proceedings. 2016. P. 153–169. DOI: 10.1007/978-3-319-40530-8_10.

41. Kowalski M., Stawicki S. SQL-based Heuristics for Selected KDD Tasks over Large Data Sets // Proceedings of the FedCSIS 2012, Federated Conference on Computer Science and Information Systems, Wroclaw, Poland, 9–12 September 2012. IEEE, 2012. P. 303–310.

42. Lepinioti K., McKearney S. Integrating Cobweb with a Relational Database // Proceedings of the International MultiConference of Engineers and Computer Scientists 2007, IMECS 2007, March 21–23, 2007, Hong Kong, China. 2007. P. 868–873.

43. Liu G., Lu H., Lou W. et al. Efficient Mining of Frequent Patterns Using Ascending Frequency Ordered Prefix-Tree // Data Min. Knowl. Discov. 2004. Vol. 9, No. 3. P. 249–274. DOI: 10.1023/B:DAMI.0000041128.59011.53.

44. Liu J., Pan Y., Wang K., Han J. Mining Frequent Item Sets by Opportunistic Projection // Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23–26, 2002, Edmonton, Alberta, Canada. 2002. P. 229–238. DOI: 10.1145/775047.775081.

45. Lizardo E.O., Davis C.A. A PostGIS Extension to Support Advanced Spatial Data Types and Integrity Constraints // Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2017, Redondo Beach, CA, USA, November 7–10, 2017. P. 33:1–33:10. DOI: 10.1145/3139958.3140020.

46. Lloyd S.P. Least Squares Quantization in PCM // IEEE Transactions on Information Theory. 1982. Vol. 28, No. 2. P. 129–136. DOI: 10.1109/TIT.1982.1056489.

47. Mahajan D., Kim J.K., Sacks J. et al. In-RDBMS Hardware Acceleration of Advanced Analytics // PVLDB. 2018. Vol. 11, No. 11. P. 1317–1331.

48. Matusevich D.S., Ordonez C. A Clustering Algorithm Merging MCMC and EM Methods Using SQL Queries // Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, BigMine 2014, New York City, USA, August 24, 2014. 2014. P. 61–76.

49. Malerba D., Appice A., Ceci M. A Data Mining Query Language for Knowledge Discovery in a Geographical Information System // Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries. 2004. P. 95–116. DOI: 10.1007/978-3-540-44497-8_5.

50. McCaffrey J.D. A Hybrid System for Analyzing Very Large Graphs // Ninth International Conference on Information Technology: New Generations, ITNG 2012, Las Vegas, Nevada, USA, April 16–18, 2012. 2012. P. 253–257. DOI: 10.1109/ITNG.2012.43.

51. Meo R., Psaila G., Ceri S. A New SQL-like Operator for Mining Association Rules // VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3–6, 1996, Mumbai (Bombay), India. 1996. P. 122–133.

52. Moertini V., Sitohang B., Santosa O.S. Searching Object-Relational

53. Ordonez C. Statistical Model Computation with UDFs // IEEE Trans. Knowl. Data Eng. 2010. Vol. 22, No. 12. P. 1752–1765. DOI: 10.1109/TKDE.2010.44.

54. Ordonez C. Can We Analyze Big Data Inside a DBMS? // Proceedings of the 16th International Workshop on Data Warehousing and OLAP, DOLAP 2013, San Francisco, CA, USA, October 28, 2013. 2013. P. 85–92. DOI: 10.1145/2513190.2513198.

55. Ordonez C. Programming the K-means Clustering Algorithm in SQL // Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22–25, 2004. 2004. P. 823–828. DOI:10.1145/1014052.1016921.

56. Ordonez C. Integrating K-Means Clustering with a Relational DBMS Using SQL // IEEE Trans. Knowl. Data Eng. 2006. Vol. 18, No. 2. P. 188–201. DOI: 10.1109/TKDE.2006.31.

57. Ordonez C., Cereghini P. SQLEM: Fast Clustering in SQL Using the EM Algorithm // Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16–18, 2000, Dallas, Texas, USA. 2000. P. 559–570. DOI: 10.1145/342009.335468.

58. Ordonez C., Chen Z. Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis // IEEE Trans. Knowl. Data Eng. 2012. Vol. 24, No. 4. P. 678–691. DOI: 10.1109/TKDE.2011.16.

59. Ordonez C., Garcia-Alvarado C. A Data Mining System Based on SQL Queries and UDFs for Relational Databases // Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24–28, 2011. 2011. P. 2521–2524. DOI: 10.1145/2063576.2064008.

60. Ordonez C., Garcia-Alvarado C., Baladandayuthapani V. Bayesian Variable Selection in Linear Regression in One Pass for Large Datasets // TKDD. 2014. Vol. 9, No. 1. P. 3:1–3:14. DOI: 10.1145/2629617.

61. Ordonez C., Garcia-Garcia J., Garcia-Alvarado C. et al. Data Mining Algorithms as a Service in the Cloud Exploiting Relational Database Systems // Proceedings of the ACM SIGMOD

International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22–27, 2013. 2013. P. 1001–1004. DOI: 10.1145/2463676.2465240.

62. Ordonez C., Mohanam N., Garcia-Alvarado C. PCA for Large Data Sets with Parallel Data Summarization // Distributed and Parallel Databases. 2014. Vol. 32, No. 3. P. 377–403. DOI: 10.1007/s10619-013-7134-6.

63. Padmanabhan S., Chakravarthy S. HDB-Subdue: A Scalable Approach to Graph Mining // Data Warehousing and Knowledge Discovery, 11th International Conference, DaWaK 2009, Linz, Austria, August 31 – September 2, 2009, Proceedings. 2009. P. 325–338. DOI: 10.1007/978-3-642-03730-6_26.

64. Pan C., Zymbler M. Very Large Graph Partitioning by Means of Parallel DBMS // Advances in Databases and Information Systems - 17th East European Conference, ADBIS 2013, Genoa, Italy, September 1–4, 2013. Proceedings. 2013. P. 388–399. DOI: 10.1007/978-3-642-40683-6_29.

65. Park J.S., Chen M., Yu P.S. An Effective Hash Based Algorithm for Mining Association Rules // Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22–25, 1995. 1995. P. 175–186. DOI: doi.org/10.1145/223784.223813.

66. Rantzau R. Frequent Itemset Discovery with SQL Using Universal Quantification // Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries. 2004. P. 194–213. DOI: 10.1007/978-3-540-44497-8_10.

67. Rantzau R., Shapiro L.D., Mitschang B., Wang Q. Algorithms and Applications for Universal Quantification in Relational Databases // Information Systems. 2003. Vol. 28, No. 1–2. P. 3–32. DOI: 10.1016/S0306-4379(02)00047-9.

68. Sarawagi S., Thomas S., Agrawal R. Integrating Mining with Relational Database Systems: Alternatives and Implications // SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2–4, 1998, Seattle, Washington, USA. 1998. P. 343–354. DOI: 10.1145/276304.276335.

69. Sattler K.-U., Dunemann O. SQL Database Primitives for Decision Tree Classifiers // Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5–10, 2001. ACM, 2001. P. 379–386. DOI: 10.1145/502585.502650.

70. Savasere A., Omiecinski E., Navathe S.B. An Efficient Algorithm for Mining Association Rules in Large Databases // VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11–15, 1995, Zurich, Switzerland. 1995. P. 432–444.

71. Shang X., Sattler K., Geist I. SQL Based Frequent Pattern Mining with FP-Growth // Applications of Declarative Programming and Knowledge Management, 15th International Conference on Applications of Declarative Programming and Knowledge Management, INAP 2004, and 18th Workshop on Logic Programming, WLP 2004, Potsdam, Germany, March 4–6, 2004, Revised Selected Papers. 2004. P. 32–46. DOI: 10.1007/11415763_3.

72. Sidlo C.I., Lukacs A. Shaping SQL-based Frequent Pattern Mining Algorithms // Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID 2005, Porto, Portugal, October 3, 2005, Revised Selected and Invited Papers. 2005. P. 188–201. DOI: 10.1007/11733492_11.

73. Silva Y.N., Aref W.G., Ali M.H. Similarity Group-By // Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29, 2009 – April 2, 2009, Shanghai, China. 2009. P. 904–915. DOI: 10.1109/ICDE.2009.113.

74. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQLBased Mining of Large Graphs on Relational Databases // Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21–24, 2010. Proceedings. Part II. 2010. P. 160–167. DOI: 10.1007/978-3-642-13672-6_16.

75. Stonebraker M., Madden S., Dubey P. Intel "Big Data" Science and Technology Center Vision and Execution Plan // SIGMOD Record. 2013. Vol. 42, No. 1. P. 44–49. DOI: 10.1145/2481528.2481537.

76. Sun P., Huang Y., Zhang C. Cluster-By: An Efficient Clustering Operator in Emergency Management Database Systems // Web-Age Information Management - WAIM 2013 International Workshops: HardBD, MDSP, BigEM, TMSN, LQPM, BDMS, Beidaihe, China, June 14–16, 2013. Proceedings. 2013. P. 152–164. DOI: 10.1007/978-3-642-39527-7_17.

77. Tamayo P., Berger C., Campos M.M., et al. Oracle Data Mining - Data Mining in the Database Environment // The Data Mining and Knowledge Discovery Handbook. Ed. by O. Maimon, L. Rokach. Springer, 2005. P. 1315–1329.

78. Tang Z., Maclennan J., Kim P.P. Building Data Mining Solutions with OLE DB for DM and XML for analysis // SIGMOD Record. 2005. Vol. 34, No. 2. P. 80–85. DOI: 10.1145/1083784.1083805.

79. Thomas S., Chakravarthy S. Performance Evaluation and Optimization of Join Queries for Association Rule Mining // Data Warehousing and Knowledge Discovery, 1st International Conference, DaWaK'99, Florence, Italy, August 30 – September 1, 1999, Proceedings. 1999. P. 241–250. DOI: 10.1007/3-540-48298-9_26.

80. Turner V., Gantz J., Reinsel D., et al. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. 2014. URL: http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm (дата обращения: 05.02.2019).

81. Zaki M.J. Scalable Algorithms for Association Mining // IEEE Trans. Knowl. Data Eng. 2000. Vol. 12, No. 3. P. 372–390. DOI: 10.1109/69.846291.

82. Wang F., Gordon J., Helian N. SQL Implementation of a ScanOnce Algorithm for Large Database Mining // Engineering Federated Information Systems, Proceedings of the 5th Workshop EFIS 2003, July 17–18 2003, Coventry, UK. 2003. P. 43–45.

83. Wang H., Zaniolo C., Luo C. ATLAS: A Small but Complete SQL Extension for Data Mining and Data Streams // VLDB. 2003. P. 1113–1116.

84. Wang W., Yang J., Muntz R.R. STING: A Statistical Information Grid Approach to Spatial Data Mining // VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25–29, 1997, Athens, Greece. 1997. P. 186–195.

85. Yoshizawa T., Pramudiono I., Kitsuregawa M. SQL Based Association Rule Mining Using Commercial RDBMS (IBM DB2 UBD EEE) // Data Warehousing and Knowledge Discovery, Second International Conference, DaWaK 2000, London, UK, September 4–6, 2000, Proceedings. 2000. P. 301–306. DOI: 10.1007/3-540-44466-1_30.

Цымблер Михаил Леонидович, к.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse190203

OVERVIEW OF METHODS FOR INTEGRATING DATA MINING INTO DBMS

South Ural State University (pr. Lenina 76, Chelyabinsk, 454080 Russia)
E-mail: mzym@susu.ru

Received: 27.02.2019

Data Mining is aimed to discovering understandable knowledge from data, which can be used for decision-making in various fields of human activity. The Big Data phenomenon is a characteristic feature of the modern information society. The processes of cleaning and structuring Big data lead to the formation of very large databases and data warehouses. Despite the emergence of a large number of NoSQL DBMSs, the main database management tool is still relational DBMS. Integration of Data Mining into relational DBMS is one of the promising directions of development of relational databases. Integration allows both to avoid the overhead of exporting the analyzed data from the repository and importing the analysis results back to the repository, as well as using system services embedded in the DBMS architecture for data analysis. The paper provides an overview of methods and approaches to solving the problem of integrating data mining in a DBMS. A classification of approaches to solving the problem of integrating data mining in a DBMS is given. The SQL database language extensions to provide syntactic support for data mining in a DBMS are introduced. Examples of the implementation of data mining algorithms for SQL and data analysis systems in relational databases are considered.

Keywords: data mining, relational DBMS, classification, clustering, pattern mining.

FOR CITATION

Zymbler M.L. Overview of Methods for Integrating Data Mining into DBMS. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2019. vol. 8, no. 2. pp. 32–62. (in Russian) DOI: 10.14529/cmse190203.

References

1. Miniakhmetov R.M., Zymbler M.L. Integration of Fuzzy c-Means Clustering algorithm with PostgreSQL database management system. *Vychislitel'nye Metody i Programmirovanie* [Numerical Methods and Programming]. 2012. vol. 13. pp. 46–52.

2. Rechkalov T.V. An Approach to Integration of Data Mining with Relational DBMS Based on Automatic SQL Code Generation. *Vestnik Yuzho-Uralskogo Gosudarstvennogo Universiteta. Seriya "Vychislitel'naya Matematika i Informatika"* [Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering]. 2013. vol. 2, no. 1. pp. 114–121.

3. Agrawal R., Ailamaki A., Bernstein P.A. et al. The Claremont Report on Database Research. *Commun. ACM.* 2009. vol. 52, no. 6. pp. 56–65. DOI: 10.1145/1516046.1516062.

4. Agrawal R., Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago de Chile, Chile. 1994. pp. 487–499.

5. Agrawal R., Shim K. Developing Tightly-coupled Data Mining Applications on a Relational Database System. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. pp. 287–290.

6. Abadi D., Agrawal R., Ailamaki A. et al. The Beckman Report on Database Research. *Commun. ACM*. 2016. vol. 59, no. 2. pp. 92–99. DOI: 10.1145/2845915.

7. Alashqur A. RDB-MINER: A SQL-Based Algorithm for Mining True Relational Databases. *Journal of Software*. 2010. vol. 5, no. 9. pp. 998–1005. DOI: 10.4304/jsw.5.9.998-1005.

8. Balachandran R., Padmanabhan S., Chakravarthy S. Enhanced DBSubdue: Supporting Subtle Aspects of Graph Mining Using a Relational Approach. Advances in Knowledge Discovery and Data Mining, 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9–12, 2006, Proceedings. 2006. pp. 673–678. DOI: 10.1007/11731139_77.

9. Berthold M.R., Cebron N., Dill F. et al. KNIME - the Konstanz Information Miner: Version 2.0 and Beyond. *SIGKDD Explorations*. 2009. vol. 11, no. 1. pp. 26–31. DOI: 10.1145/1656274.1656280.

10. Bezdek J.C., Ehrlich R., Full W. FCM: The Fuzzy C-Means Clustering Algorithm. *Computers and Geosciences*. 1984. vol. 10, no. 2. pp. 191–203. DOI: 10.1016/0098-3004(84)90020-7.

11. Blockeel H., Calders T., Fromont E. et al. An Inductive Database Prototype Based on Virtual Mining Views. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2008. 2008. pp. 1061–1064. DOI: 10.1145/1401890.1402019.

12. Blockeel H., Calders T., Fromont E. et al. An Inductive Database Prototype Based on Virtual Mining Views. *Data Min. Knowl. Discov.* 2012. vol. 24, no. 1. pp. 247–287. DOI: 10.1007/s10618-011-0229-7.

13. Blockeel H., Calders T., Fromont E. et al. Inductive Querying with Virtual Mining Views. Inductive Databases and Constraint-Based Data Mining. Ed. by S. Dzeroski, B. Goethals, P. Panov. Springer, 2010. pp. 265–287. DOI: 10.1007/978-1-4419-7738-0_11.

14. Brin S., Motwani R., Ullman J.D., Tsur S. Dynamic Itemset Counting and Implication Rules for Market Basket Data. SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13–15, 1997, Tucson, Arizona, USA. 1997. pp. 255–264. DOI: 10.1145/253260.253325.

15. Bogorny V., Kuijpers B., Alvares L.O. ST-DMQL: A Semantic Trajectory Data Mining Query Language. *International Journal of Geographical Information Science*. 2009. vol. 23, no. 10. pp. 1245–1276.

16. Breiman L., Friedman J., Olshen R., Stone C. Classification and Regression Trees. Wadsworth International Group, 1984.

17. Calders T., Lakshmanan L.V.S., Ng R.T., Paredaens J. Expressive Power of an Algebra for Data Mining. *ACM Trans. Database Syst.* 2006. vol. 31, no. 4. pp. 1169–1214. DOI: 10.1145/1189769.1189770.

18. Chakravarthy S., Pradhan S. DB-FSG: An SQL-based Approach for Frequent Subgraph Mining. Database and Expert Systems Applications, 19th International Conference, DEXA 2008, Turin, Italy, September 1–5, 2008. Proceedings. 2008. pp. 684–692. DOI: 10.1007/978-3-540-85654-2_59.

19. Chaudhuri S. What Next?: a Half-dozen Data Management Research Goals for Big Data and the Cloud. Proceedings of the 31st ACM SIGMODSIGACT- SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20–24, 2012. 2012. pp. 1–4. DOI: 10.1145/2213556.2213558.

20. Chen X., Petrounias I. Language Support for Temporal Data Mining. Principles of Data Mining and Knowledge Discovery, 2nd European Symposium, PKDD '98, Nantes, France, September 23–26, 1998, Proceedings. 1998. pp. 282–290. DOI: 10.1007/BFb0094830.

21. Codd E.F. A Relational Model of Data for Large Shared Data Banks. *Commun. ACM*. 1970. vol. 13, no. 6. pp. 377–387. DOI: 10.1145/362384.362685.

22. Davoudian A., Chen L., Liu M. A Survey on NoSQL Stores. ACM Comput. Surv. 2018. vol. 51, no. 2. pp. 40:1–40:43. DOI: 10.1145/3158661.

23. Ester M., Kriegel H., Sander J., Xu X. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. pp. 226–231.

24. Frank E., Hall M.A., Holmes G. et al. WEKA - A Machine Learning Workbench for Data Mining. The Data Mining and Knowledge Discovery Handbook. / Ed. by O. Maimon, L. Rokach. Springer, 2005. pp. 1305–1314.

25. Frawley W.J., Piatetsky-Shapiro G., Matheus C.J. Knowledge Discovery in Databases: an Overview. Knowledge Discovery in Databases. AAAI/MIT Press, 1991. pp. 1–30.

26. Dempster A., Laird N., Rubin D. Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm. *Journal of The Royal Statistical Society*. 1977. vol. 39, no. 1. pp. 1–38.

27. Garcia W., Ordonez C., Zhao K., Chen P. Efficient Algorithms Based on Relational Queries to Mine Frequent Graphs. Proceedings of the 3rd PhD Workshop on Information and Knowledge Management, PIKM 2010, Toronto, Ontario, Canada, October 30, 2010. 2010. pp. 17–24. DOI: 10.1145/1871902.1871906.

28. Guha S., Mishra N., Motwani R., O'Callaghan L. Clustering Data Streams. Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12–14 November 2000, Redondo Beach, California, USA. 2000. pp. 359–366. DOI: 10.1109/SFCS.2000.892124.

29. Han J., Fu Y., Wang W. et al. DBMiner: A System for Mining Knowledge in Large Relational Databases. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA. 1996. pp. 250–255.

30. Han J., Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006. pp. 743.

31. Han J., Koperski K., Stefanovic N. GeoMiner: A System Prototype for Spatial Data Mining. SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13–15, 1997, Tucson, Arizona, USA. 1997. pp. 553–556. DOI: 10.1145/253260.253404.

32. Han J., Pei J., Yin Y. Mining Frequent Patterns without Candidate Generation. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16–18, 2000, Dallas, Texas, USA. 2000. pp. 1–12. DOI: 10.1145/342009.335372.

33. Hellerstein J.M., Re C., Schoppmann F. et al. The MADlib Analytics Library or MAD Skills, the SQL. *PVLDB*. 2012. vol. 5, no. 12. pp. 1700–1711.

34. HooshSadat M., Samuel H.W., Patel S., Zaiane O.R. Fastest Association Rule Mining Algorithm Predictor (FARM-AP). Proceedings of the 4th International C* Conference on Computer Science and Software Engineering, C3S2E 2011, Montreal, Quebec, Canada, May 16–18, 2011. pp. 43–50. DOI: 10.1145/1992896.1992902.

35. Houtsma M.A.W., Swami A.N. Set-Oriented Mining for Association Rules in Relational Databases. Proceedings of the 11th International Conference on Data Engineering, March 6–10, 1995, Taipei, Taiwan. 1995. pp. 25–33. DOI: 10.1109/ICDE.1995.380413.

36. Huang Z. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Min. Knowl. Discov.* 1998. vol. 2, no. 3. pp. 283–304. DOI: 10.1023/A:1009769707641.

37. Imielinski T., Virmani A. MSQL: A Query Language for Database Mining. *Data Min. Knowl. Discov.* 1999. vol. 3, no. 4. pp. 373–408. DOI: 10.1023/A:1009816913055.

38. Karypis G., Kumar V. Analysis of Multilevel Graph Partitioning. Proceedings of Supercomputing '95, San Diego, CA, USA, December 4–8, 1995. 1995. pp. 29. DOI: 10.1145/224170.224229.

39. Kaufman L., Rousseeuw P.J. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley, 1990. DOI: 10.1002/9780470316801.

40. Krause C., Johannsen D., Deeb R. et al. An SQL-Based Query Language and Engine for Graph Pattern Matching. Graph Transformation - 9th International Conference, ICGT 2016, in Memory of Hartmut Ehrig, Held as Part of STAF 2016, Vienna, Austria, July 5–6, 2016, Proceedings. 2016. pp. 153–169. DOI: 10.1007/978-3-319-40530-8_10.

41. Kowalski M., Stawicki S. SQL-based Heuristics for Selected KDD Tasks over Large Data Sets. Proceedings of the FedCSIS 2012, Federated Conference on Computer Science and Information Systems, Wroclaw, Poland, 9–12 September 2012. IEEE, 2012. pp. 303–310.

42. Lepinioti K., McKearney S. Integrating Cobweb with a Relational Database. Proceedings of the International MultiConference of Engineers and Computer Scientists 2007, IMECS 2007, March 21–23, 2007, Hong Kong, China. 2007. pp. 868–873.

43. Liu G., Lu H., Lou W. et al. Efficient Mining of Frequent Patterns Using Ascending Frequency Ordered Prefix-Tree. *Data Min. Knowl. Discov.* 2004. vol. 9, no. 3. pp. 249–274. DOI: 10.1023/B:DAMI.0000041128.59011.53.

44. Liu J., Pan Y., Wang K., Han J. Mining Frequent Item Sets by Opportunistic Projection. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23–26, 2002, Edmonton, Alberta, Canada. 2002. pp. 229–238. DOI: 10.1145/775047.775081.

45. Lizardo E.O., Davis C.A. A PostGIS Extension to Support Advanced Spatial Data Types and Integrity Constraints. Proceedings of the 25th ACM SIGSPATIAL International Conference

on Advances in Geographic Information Systems, GIS 2017, Redondo Beach, CA, USA, November 7–10, 2017. pp. 33:1–33:10. DOI: 10.1145/3139958.3140020.

46. Lloyd S.P. Least Squares Quantization in PCM. IEEE Transactions on Information Theory. 1982. vol. 28, no. 2. pp. 129–136. DOI: 10.1109/TIT.1982.1056489.

47. Mahajan D., Kim J.K., Sacks J. et al. In-RDBMS Hardware Acceleration of Advanced Analytics. *PVLDB*. 2018. vol. 11, no. 11. pp. 1317–1331.

48. Matusevich D.S., Ordonez C. A Clustering Algorithm Merging MCMC and EM Methods Using SQL Queries. Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, BigMine 2014, New York City, USA, August 24, 2014. 2014. pp. 61–76.

49. Malerba D., Appice A., Ceci M. A Data Mining Query Language for Knowledge Discovery in a Geographical Information System. Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries. 2004. pp. 95–116. DOI: 10.1007/978-3-540-44497-8_5.

50. McCaffrey J.D. A Hybrid System for Analyzing Very Large Graphs. Ninth International Conference on Information Technology: New Generations, ITNG 2012, Las Vegas, Nevada, USA, April 16–18, 2012. 2012. pp. 253–257. DOI: 10.1109/ITNG.2012.43.

51. Meo R., Psaila G., Ceri S. A New SQL-like Operator for Mining Association Rules. VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases, September 3–6, 1996, Mumbai (Bombay), India. 1996. pp. 122–133.

52. Moertini V., Sitohang B., Santosa O.S. Searching Object-Relational DBMS Features for Improving Efficiency and Scalability of Decision Tree Algorithms. iiWAS'2006 - The 8th International Conference on Information Integration and Web-based Applications Services, December 4–6, 2006, Yogyakarta, Indonesia. 2006. pp. 323–330.

53. Ordonez C. Statistical Model Computation with UDFs. *IEEE Trans. Knowl. Data Eng.* 2010. vol. 22, no. 12. pp. 1752–1765. DOI: 10.1109/TKDE.2010.44.

54. Ordonez C. Can We Analyze Big Data Inside a DBMS?. Proceedings of the 16th International Workshop on Data Warehousing and OLAP, DOLAP 2013, San Francisco, CA, USA, October 28, 2013. 2013. pp. 85–92. DOI: 10.1145/2513190.2513198.

55. Ordonez C. Programming the K-means Clustering Algorithm in SQL. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22–25, 2004. 2004. pp. 823–828. DOI:10.1145/1014052.1016921.

56. Ordonez C. Integrating K-Means Clustering with a Relational DBMS Using SQL. *IEEE Trans. Knowl. Data Eng.* 2006. vol. 18, no. 2. pp. 188–201. DOI: 10.1109/TKDE.2006.31.

57. Ordonez C., Cereghini P. SQLEM: Fast Clustering in SQL Using the EM Algorithm. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16–18, 2000, Dallas, Texas, USA. 2000. pp. 559–570. DOI: 10.1145/342009.335468.

58. Ordonez C., Chen Z. Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis. *IEEE Trans. Knowl. Data Eng.* 2012. vol. 24, no. 4. pp. 678–691. DOI: 10.1109/TKDE.2011.16.

59. Ordonez C., Garcia-Alvarado C. A Data Mining System Based on SQL Queries and UDFs for Relational Databases. Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24–28, 2011. 2011. pp. 2521–2524. DOI: 10.1145/2063576.2064008.

60. Ordonez C., Garcia-Alvarado C., Baladandayuthapani V. Bayesian Variable Selection in Linear Regression in One Pass for Large Datasets. *TKDD*. 2014. vol. 9, no. 1. pp. 3:1–3:14. DOI: 10.1145/2629617.

61. Ordonez C., Garcia-Garcia J., Garcia-Alvarado C. et al. Data Mining Algorithms as a Service in the Cloud Exploiting Relational Database Systems. Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22–27, 2013. 2013. pp. 1001–1004. DOI: 10.1145/2463676.2465240.

62. Ordonez C., Mohanam N., Garcia-Alvarado C. PCA for Large Data Sets with Parallel Data Summarization. *Distributed and Parallel Databases*. 2014. vol. 32, no. 3. pp. 377–403. DOI: 10.1007/s10619-013-7134-6.

63. Padmanabhan S., Chakravarthy S. HDB-Subdue: A Scalable Approach to Graph Mining. Data Warehousing and Knowledge Discovery, 11th International Conference, DaWaK 2009, Linz, Austria, August 31 – September 2, 2009, Proceedings. 2009. pp. 325–338. DOI: 10.1007/978-3-642-03730-6_26.

64. Pan C., Zymbler M. Very Large Graph Partitioning by Means of Parallel DBMS. Advances in Databases and Information Systems - 17th East European Conference, ADBIS 2013, Genoa, Italy, September 1–4, 2013. Proceedings. 2013. pp. 388–399. DOI: 10.1007/978-3-642-40683-6_29.

65. Park J.S., Chen M., Yu P.S. An Effective Hash Based Algorithm for Mining Association Rules. Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22–25, 1995. 1995. pp. 175–186. DOI: doi.org/10.1145/223784.223813.

66. Rantzau R. Frequent Itemset Discovery with SQL Using Universal Quantification. Database Support for Data Mining Applications: Discovering Knowledge with Inductive Queries. 2004. pp. 194–213. DOI: 10.1007/978-3-540-44497-8_10.

67. Rantzau R., Shapiro L.D., Mitschang B., Wang Q. Algorithms and Applications for Universal Quantification in Relational Databases. *Information Systems*. 2003. vol. 28, no. 1–2. pp. 3–32. DOI: 10.1016/S0306-4379(02)00047-9.

68. Sarawagi S., Thomas S., Agrawal R. Integrating Mining with Relational Database Systems: Alternatives and Implications. SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2–4, 1998, Seattle, Washington, USA. 1998. pp. 343–354. DOI: 10.1145/276304.276335.

69. Sattler K.-U., Dunemann O. SQL Database Primitives for Decision Tree Classifiers. Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5–10, 2001. ACM, 2001. pp. 379–386. DOI: 10.1145/502585.502650.

70. Savasere A., Omiecinski E., Navathe S.B. An Efficient Algorithm for Mining Association Rules in Large Databases. VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11–15, 1995, Zurich, Switzerland. 1995. pp. 432–444.

71. Shang X., Sattler K., Geist I. SQL Based Frequent Pattern Mining with FP-Growth. Applications of Declarative Programming and Knowledge Management, 15th International Conference on Applications of Declarative Programming and Knowledge Management, INAP 2004, and 18th Workshop on Logic Programming, WLP 2004, Potsdam, Germany, March 4–6, 2004, Revised Selected Papers. 2004. pp. 32–46. DOI: 10.1007/11415763_3.

72. Sidlo C.I., Lukacs A. Shaping SQL-based Frequent Pattern Mining Algorithms. Knowledge Discovery in Inductive Databases, 4th International Workshop, KDID 2005, Porto, Portugal, October 3, 2005, Revised Selected and Invited Papers. 2005. pp. 188–201. DOI: 10.1007/11733492_11.

73. Silva Y.N., Aref W.G., Ali M.H. Similarity Group-By. Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29, 2009 – April 2, 2009, Shanghai, China. 2009. pp. 904–915. DOI: 10.1109/ICDE.2009.113.

74. Srihari S., Chandrashekar S., Parthasarathy S. A Framework for SQLBased Mining of Large Graphs on Relational Databases. Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21–24, 2010. Proceedings. Part II. 2010. pp. 160–167. DOI: 10.1007/978-3-642-13672-6_16.

75. Stonebraker M., Madden S., Dubey P. Intel "Big Data" Science and Technology Center Vision and Execution Plan. *SIGMOD Record*. 2013. vol. 42, no. 1. pp. 44–49. DOI: 10.1145/2481528.2481537.

76. Sun P., Huang Y., Zhang C. Cluster-By: An Efficient Clustering Operator in Emergency Management Database Systems. Web-Age Information Management - WAIM 2013 International Workshops: HardBD, MDSP, BigEM, TMSN, LQPM, BDMS, Beidaihe, China, June 14–16, 2013. Proceedings. 2013. pp. 152–164. DOI: 10.1007/978-3-642-39527-7_17.

77. Tamayo P., Berger C., Campos M.M., et al. Oracle Data Mining - Data Mining in the Database Environment. The Data Mining and Knowledge Discovery Handbook. Ed. by O. Maimon, L. Rokach. Springer, 2005. pp. 1315–1329.

78. Tang Z., Maclennan J., Kim P.P. Building Data Mining Solutions with OLE DB for DM and XML for analysis. *SIGMOD Record*. 2005. vol. 34, no. 2. pp. 80–85. DOI: 10.1145/1083784.1083805.

79. Thomas S., Chakravarthy S. Performance Evaluation and Optimization of Join Queries for Association Rule Mining. Data Warehousing and Knowledge Discovery, 1st International Conference, DaWaK'99, Florence, Italy, August 30 – September 1, 1999, Proceedings. 1999. pp. 241–250. DOI: 10.1007/3-540-48298-9_26.

80. Turner V., Gantz J., Reinsel D., et al. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. 2014. Available at: http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm (accessed: 05.02.2019).

81. Zaki M.J. Scalable Algorithms for Association Mining. *IEEE Trans. Knowl. Data Eng.* 2000. vol. 12, no. 3. pp. 372–390. DOI: 10.1109/69.846291.

82. Wang F., Gordon J., Helian N. SQL Implementation of a ScanOnce Algorithm for Large Database Mining. Engineering Federated Information Systems, Proceedings of the 5th Workshop EFIS 2003, July 17–18 2003, Coventry, UK. 2003. pp. 43–45.

83. Wang H., Zaniolo C., Luo C. ATLAS: A Small but Complete SQL Extension for Data Mining and Data Streams. VLDB. 2003. pp. 1113–1116.

84. Wang W., Yang J., Muntz R.R. STING: A Statistical Information Grid Approach to Spatial Data Mining. VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25–29, 1997, Athens, Greece. 1997. pp. 186–195.

85. Yoshizawa T., Pramudiono I., Kitsuregawa M. SQL Based Association Rule Mining Using Commercial RDBMS (IBM DB2 UBD EEE). Data Warehousing and Knowledge Discovery, Second International Conference, DaWaK 2000, London, UK, September 4–6, 2000, Proceedings. 2000. pp. 301–306. DOI: 10.1007/3-540-44466-1_30.

УДК 519.6 **DOI: 10.14529/cmse190204**

ПРИМЕНЕНИЕ МНОГОМЕРНОЙ КВАНТИЛЬНОЙ ФУНКЦИИ В ЗАДАЧЕ ПЕПТИД-БЕЛОК ДОКИНГА*

© 2019 С.В. Полуян[1], Н.М. Ершов[2]

[1]*Государственный университет «Дубна»*
(141982 Дубна, ул. Университетская, д. 19),
[2]*Московский государственный университет имени М.В. Ломоносова*
(119991 Москва, ул. Ленинские горы, д. 1)
E-mail: svpoluyan@gmail.com, ershov@cs.msu.ru

Поступила в редакцию: 03.11.2018

Настоящая работа посвящена исследованию применения стохастических эволюционных алгоритмов оптимизации к задаче пептид-белок докинга. В статье продемонстрированы основные положения, сводящие докинг к задаче непрерывной глобальной оптимизации. Представлены основные особенности рассматриваемой задачи и возникающие трудности применения эволюционных алгоритмов оптимизации. Предложен способ применения эволюционных алгоритмов, включающий использование эмпирической квантильной функции. Приведено краткое «рекурсивное» определение структуры многомерной квантильной функции с использованием одномерного квантильного преобразования. Представлен сеточный подход применения квантильной функции и указаны его недостатки. Предложен детерминированный алгоритм построения выборки, приведена схема его распараллеливания и получаемое ускорение. Для квантильной функции описана схема использования параллельных вычислений, включающая вычисления на графических ускорителях. Предложено несколько способов параллелизации с использованием выборки в явном виде. Продемонстрирована их производительность в зависимости от размера выборки. Представлены результаты докинга с использованием эволюционного алгоритма и его модификации с применением квантильной функции. Выполнено сравнение с актуальным методом докинга в рамках одного силового поля. Проведен анализ результатов вычислительных экспериментов.

Ключевые слова: глобальная оптимизация, эволюционные алгоритмы, эмпирическая квантильная функция, докинг.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Полуян С.В., Ершов Н.М. Применение многомерной квантильной функции в задаче пептид-белок докинга // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 63–75. DOI: 10.14529/cmse190204.

Введение

В задаче пептид-белок докинга необходимо найти оптимальное место связывания белка и пептида при взаимодействии друг с другом, а также соответствующую этой связи конформацию комплекса. Традиционными экспериментальными методами определения места связывания и соответствующих конформаций белка и пептида являются кристаллография, ядерный магнитный резонанс, а также другие техники [1]. Несмотря на их точность и эффективность, они требуют значительных лабораторных ресурсов и материальных затрат. Более того, пептид-белок комплекс сложнее кристаллизовать, чем отдельный белок. В то время как докинг менее затратный финансово, это лишь компьютерный метод предсказания структуры комплекса. В связи с этим вычислительные методы приобретают все большую популярность. Большинство из них на различных

*Статья рекомендована к публикации программным комитетом Международной конференции «Суперкомпьютерные дни в России – 2018».

этапах работы включает в себя разнообразные методы стохастической оптимизации [2, 3]. Одним из основных преимуществ использования стохастических методов оптимизации является возможность напрямую использовать различную статистическую информацию. Кроме того, использование методов оптимизации более привлекательно в вычислительном отношении, чем, например, применение методов молекулярной динамики.

В основе большинства подходов к докингу лежит термодинамическая гипотеза Анфинсена, основное утверждение которой следующее: оптимальное состояние комплекса уникально и находится в глобальном минимуме свободной энергии. Поэтому задача пептид-белок докинга может быть рассмотрена как задача глобальной оптимизации, в которой необходимо найти конформацию комплекса с минимальной энергией.

Статья организована следующим образом. В разделе 1 приводится постановка задачи и рассматриваются возникающие трудности применения эволюционных алгоритмов оптимизации. Раздел 2 посвящен эмпирической квантильной функции. Предложен сеточный подход применения квантильной функции, представлен способ построения выборки и описана параллельная реализация. В разделе 3 представлены результаты численных экспериментов. В разделе Заключение приводятся результаты выполненной работы и указываются направления дальнейших исследований.

1. Постановка задачи

В общем случае задачи пептид-белок докинга решаются комбинированными методами, включающими в себя несколько различных по структуре этапов и учитывающих разнообразную статистическую информацию. Такого рода комбинированные предсказания выходят за рамки текущего исследования. В большинстве случаев заключительным этапом является поиск в полноатомном разрешении оптимальной структуры комплекса в окрестности места связывания, так называемый прямой докинг. Именно на этом этапе используются стохастические методы оптимизации в сочетании с методами локальной оптимизации. Важно отметить, что применяемые на данном этапе методы оптимизации (как глобальной, так и локальной) обладают высокой степенью универсальности относительно решаемой задачи, т.е. структура и параметры алгоритмов, как правило, независимы от сложности целевой функции и соответствующего энергетического ландшафта. Примером, подчеркивающим указанную универсальность, может служить протокол докинга Rosetta FlexPepDock [2], структура и применяемые алгоритмы которого не зависят от фундаментально меняющегося состава стандартной скоринг-функции силового поля.

Необходимо отметить, что на заключительном этапе поиск оптимальной структуры комплекса ведется, как правило, с учетом структурных особенностей [1] предполагаемого места связывания. Здесь необходимо подчеркнуть специфику рассматриваемой задачи. В силу структурых особенностей пептиды обладают высокой гибкостью. Торсионные углы главной цепи каждого аминокислотного остатка пептида являются ротамерами. В связи с этим докинг даже простейших пептидов длиной 2-5 аминокислотных остатка в полноатомном разрешении представляет собой сложную (иногда невыполнимую) задачу даже для специально разработанных пакетов [1].

Поиск в окрестности места связывания довольно просто организовать с помощью методов сэмплирования. Однако, предлагаемые в настоящее время эвристические подходы к глобальной оптимизации, в частности, эволюционные алгоритмы, требуют непрерывного пространства поиска без ограничений, кроме границ поиска для каждого параметра.

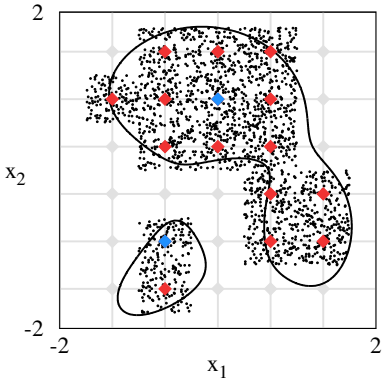
Возникает вопрос, каким образом не допустить значительного смещения пептида в окрестности области поиска и сохранить непрерывность пространства поиска? При этом избежать грубого подхода с использованием штрафных функций и сохранить условия для прямого применения эволюционных алгоритмов оптимизации. Ответом может послужить применение многомерного квантильного преобразования.

Настоящая работа посвящена исследованию применения эволюционных алгоритмов оптимизации к задаче пептид-белок докинга с использованием квантильного преобразования. При этом ставится задача удержания пептида в некоторой локальной окрестности области поиска.

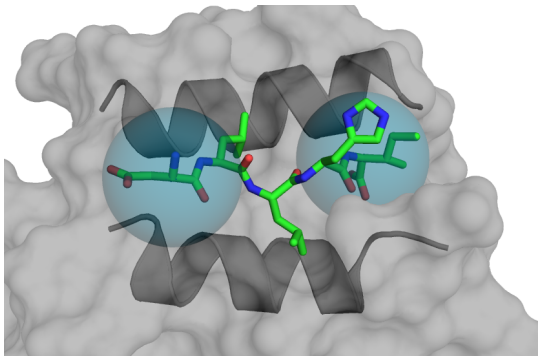
Задача оптимизации формулируется как задача минимизации энергии связывания (1), которая вычисляется как разница между энергией комплекса в связанном состоянии и энергией в свободном состоянии, т.е. когда белок и пептид друг с другом не взаимодействуют.

$$E_{\text{binding energy}} = E_{\text{complex}} - (E_{\text{protein}} + E_{\text{peptide}}). \tag{1}$$

Взаимодействие между пептидом и белком может быть описано целевой функцией. В численных экспериментах использовалось силовое поле Rosetta 3.8 [4]. Выбор силового поля обусловлен широкой распространенностью и ориентированностью к проблеме пептид-белок докинга. Детальное описание постановки задачи пептид-белок докинга, описание степеней свободы пептида и белка представлено в [5, 6]. В настоящей работе эксперименты проводились с комплексом 1JWG (код PDB) с линейным интерфейсом связывания [1, 7]. Комплекс представлен на рис. 1.



а) Сетка, два исходных узла, выборка и $2{\cdot}10^3$ распределенных точек



б) Стартовая позиция и границы смещения пептида комплекса 1JWG (код PDB)

Рис. 1. Покрытие непрерывной области поиска и рассматриваетмый пептид-белок комплекс

2. Многомерная эмпирическая квантильная функция

Определение многомерной эмпирической квантильной функции (или эмпирического квантильного преобразования) естественно выводится из определения эмпирической функции распределения. Впервые понятие многомерной квантильной функции введено в [8], однако, наиболее распространеное определение приведено в [9]. Здесь будет приведено краткое «рекурсивное» описание структуры квантильной функции.

Пусть дано вероятностное пространство и на нем определена случайная величина X. Функцией распределения случайной величины X назовем функцию $F_X : \mathbf{R} \to [0, 1]$,

задаваемой формулой $F_X(x) = \mathrm{P}(X \leqslant x)$. Для заданной функции квантильное преобразование $F_X^{-1} : [0,1] \to \mathbf{R}$ определяется следующей формулой:

$$F_X^{-1}(p) = \inf\{x \in \mathbf{R} : P(X \leqslant x) \geqslant p\}. \tag{2}$$

Определим эмпирическую функцию распределения следующим образом:

$$\hat{\mathrm{F}}_X(u) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(\mathbf{x}_{[i]} \leqslant u), \tag{3}$$

где $\mathbf{1}$ – индикаторная функция.

Важно отметить, что здесь и далее рассматриваются случайные величины, распределенные на сетке. Это связано с двумя факторами. Во первых, в процессе дальнейшего применения построенного преобразования важна только доступность той или иной области поиска, поэтому достаточно равномерного распределения. Во вторых, если значения будут распределены просто равномерно, а не по сетке, то в формуле (3) количество найденных элементов может стать равно нулю.

Процедура использования одномерного квантильного преобразования для непрерывного числа из отрезка $[0,1]$ выглядит следующим образом (см. рис. 2). Для заданной выборки *sample* на сетке *grid* и всех значений в узлах сетки выполняется процедура двоичного поиска необходимого узлового значения сетки. Вначале выбирается значение середины сетки, производится подсчет количества элементов в выборке меньше данного значения середины, которое затем делится на общее количество элементов в выборке. Аналогичные действия производятся для соседнего узла сетки. Затем производится шаг, аналогичный двоичному поиску: если непрерывное значение меньше полученного числа, то меняется верхняя граница поиска по сетке. В противном случае аналогично меняется нижняя граница. Однако, если непрерывное значение больше узлового значения и меньше соседнего значения, то процедура поиска нужного значения сетки заканчивается. Затем для поддержания непрерывности используется линейная интерполяция.

Используя определение многомерной эмпирической функции –

$$\hat{\mathrm{F}}_{X_1,X_2,\ldots,X_d}(u_1,u_2,\ldots,u_d) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(\mathbf{x}_{[i,1]} \leqslant u_1, \mathbf{x}_{[i,2]} \leqslant u_2, \ldots, \mathbf{x}_{[i,d]} \leqslant u_d), \tag{4}$$

где d – размерность, n – размер выборки, можно определить многомерную квантильную функцию $[0,1]^d \to \mathbf{R}^d$. Пусть F – d-мерная функция распределения и $X_1, \ldots, X_n$ – выборка. Используя одномерное квантильное преобразование (2) и выбрав вектор $z \in [0,1]^n$ можно определить рекурсивно квантильную функцию $Y = \tau_F^{-1}(z)$:

$$Y_1 = F_1^{-1}(z_1), \tag{5}$$

$$Y_k = F_{k|1,\ldots,k-1}^{-1}(z_k|Y_1,\ldots,Y_{k-1}), \quad 2 \leqslant k \leqslant d. \tag{6}$$

2.1. Построение выборки

Построение выборки для квантильной функции выглядит следующим образом. Пептид помещается в произвольную область поиска в окрестности места связывания. Поскольку структура интерфейса связывания известна, пептид располагается в приблизительно линейной структуре. Как указывалось выше, рассматривается только заключительный этап оптимизации, и, в общем случае, структура может быть произвольной. В координатах атома α-углерода первого и последнего аминокислотного остатка пептида создаются области поиска, которые будут определять границы степени свободы для положения

```
float getval(std::vector<float> &sample, std::vector<float> &grid, float val01)
{
    size_t count = grid.size() - 1, step, c1 = 0, c2 = 0;
    float f1, f2, n = sample.size();
    std::vector<float>::iterator first = grid.begin(), it;
    while(count > 0)
    {
        it = first;  step = count / 2;  std::advance(it, step);
        c1 = std::count_if(sample.begin(), sample.end(),
                        [&it](const float &v){ return v < *it;});
        c2 = std::count_if(sample.begin(), sample.end(),
                        [&it](const float &v){ return v < *(it + 1);});
        f1 = c1/n; f2 = c2/n;
        if(val01 > f1 && val01 < f2)
            break;
        if(f1 < val01)
        {
            first = ++it;
            count -= step + 1;
        }
        else
            count = step;
    }
    // Обработка исключительного случая равенства c1 и c2
    // доступна в открытом репозитории [13] сервиса GitHub
    return *it + (val01 - f1) * (*(it + 1) - *it) / (f2 - f1);
}
```

Рис. 2. Одномерное квантильное преобразование по сетке с линейной интерполяцией

выбранных атомов, которые зависят от оптимизируемых параметров. Поскольку положение первого α-углерода определяет смещение пептида относительно белка для отображения в сферу используется трехмерное квантильное преобразование, построенное по плотности распределения, которое уже использовалось в [5]. Положение пептида в определенной сферами области поиска зависит от параметров смещения пептида, угла и вектора поворота, а также торсионных углов главной цепи пептида. Для каждого параметра создается собственная сетка. Важно отметить, что границы каждого параметра переведены в диапазон $[0, 1]$. Параметры, определяющие смещение пептида и часть углов главной цепи пептида, уже находятся в диапазоне $[0, 1]$. Остальных параметры переводятся в этот диапазон и преобразуются в искомые с помощью линейной интерполяции. В итоге пространство поиска сведено в единичный гиперкуб.

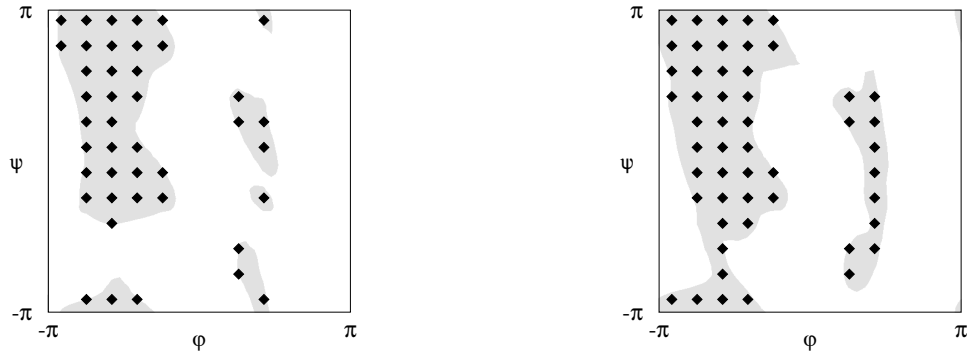
Для каждого параметра, принадлежащего гиперкубу, производится разбиение на независимое заданное число равных частей, которое будет определять узлы сетки. Для каждого параметра определяется ближайшее значение в полученной сетке. Если область поиска в ближайших узлах сетки не найдена, то происходит поиск в n-мерной окрестности фон Неймана или Мура.

Теперь можно приступить к процедуре построения выборки. Для этого используется n-мерный аналог алгоритма заливки Flood-fill [10] с использованием n-мерной окрестности

фон Неймана. Алгоритм Flood-fill в процессе своей работы может посетить одно и тоже значение узла в сетке несколько раз. В связи с тем, что смещение пептида и присвоение ему параметров требует вычислительных ресурсов, посещенные и вычисленные значения сетки добавляются в префиксное дерево. Поиск и проверка требуют значительно меньших ресурсов.

Важно отметить, что сам процесс построения выборки является обходом графа со структурой «решетка», но без определения самого графа. Выбор алгоритма обусловлен простотой реализации и исследованиями в [10], а также собственными экспериментами. Также необходимо отметить, что при обходе сетки невозможно использовать окрестность Мура для всех параметров в силу высокого количества соседних узлов. Например, в рассматриваемой задаче количество параметров в гиперкубе $n = 15$, для каждой точки окрестность фон Неймана $2n$ узлов, окрестность Мура 3^n узлов. Однако, алгоритм Flood-fill позволяет использовать окрестность Мура для части параметров.

На рис. 1 представлен результат построения выборки по двумерной сетке (5×6) с помощью реализованного алгоритма Flood-fill для произвольной области. Также на рис. 1, выражаясь в терминах компьютерной графики, отмечен «затравочный» узел и результат квантильного преобразования равномерно распределенных векторов $2{\cdot}10^3$. На рис. 3 показан результат покрытия параметров равномерной сеткой. Сами параметры кодируются в отрезке $[0, 1]$, на рис. 3 показаны получаемые в результате преобразования узлов сетки значения.



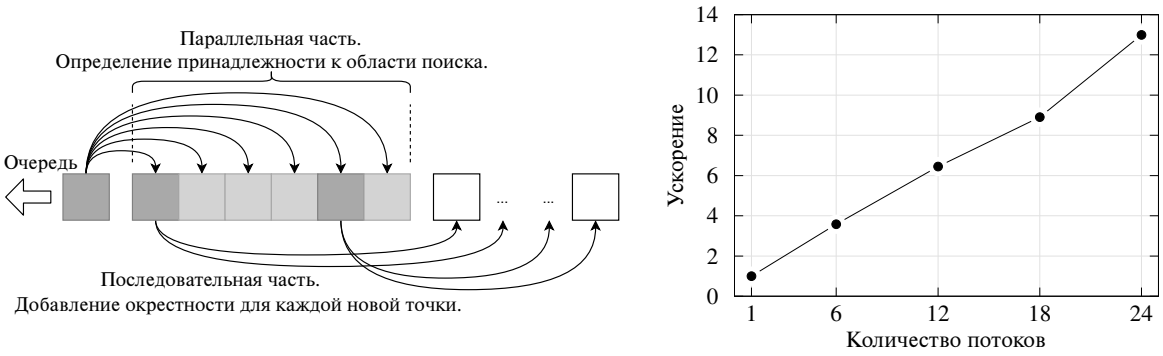
а) Лейцин б) Гистидин

Рис. 3. Покрытие распределений углов главной цепи пептида DLLHI комплекса 1JWG

Необходимо отметить, что в приведенных примерах явно указаны основные недостатки построения выборки по сетке и квантильного преобразования. Во первых, алгоритм подходит только для определения связанной области. В случае несвязанности области из других областей для алгоритма также нужен исходный узел. Во вторых, значительная часть области поиска может оказаться недоступной из-за большого шага сетки и использования окрестности фон Неймана. В третьих, часть области поиска, рядом с границей непрерывной области, недоступна. В четвертых, граница поиска может выйти за границы непрерывной области. Однако, последний недостаток не столь важен, так как незначительный выход за границы поиска несущественен.

В настоящей работе произведена параллельная реализация алгоритма построения выборки с применением технологии OpenMP. Произведено сравнение с последовательной версией алгоритма. На рис. 4 представлена схема распараллеливания и получаемое ускорение. Каждый узел сетки, который необходимо проверить на принадлежность к

области поиска, добавляется в очередь. Если текущий узел принадлежит области, в очередь добавляется его окрестность, за исключением уже посещенных узлов. В отличие от последовательной версии, где проверка и добавление в дерево с вычисленными значениями происходит на каждом шаге, в параллельной реализации некоторые узлы вычисляются несколько раз.



а) Добавление в очередь окрестности каждой принадлежащей области поиска точки

б) Производительность при использовании технологии OpenMP

Рис. 4. Используемая схема распараллеливания и ее производительность

2.2. Параллельная реализация

При использовании многомерной квантильной функции (5–6) для преобразования одного вектора $z \in [0,1]^d$ необходимо d раз пройти по всей выборке размера n и каждый раз отмечать удовлетворяющие всем предыдущим условиям значения, которые затем будут использоваться в одномерном квантильном преобразовании. Каждый элемент выборки – d-мерный вектор. Таким образом производится покоординатное сравнение вектора из выборки с двумя известными векторами (верхней и нижней границей). В случае невыполнения хотя бы одного условия при сравнении, т.е. непопадания координаты вектора в диапазон значений между границами, вектор из выборки не рассматривается.

На данном этапе довольно просто применить параллельные вычисления. Поскольку заранее неизвестно количество удовлетворяющих всем условиям векторов в текущей реализации одномерное квантильное преобразование выполняется на хосте.

В результате выполненной работы реализованы три различных способа выбора необходимых значений с использованием технологии OpenCL. Во всех трех случаях вся выборка располагается на графическом ускорителе. При этом используется тип данных $float$.

В первом случае используется простейшая схема распараллеливания. Ядро OpenCL запускается с максимальным возможным размером рабочей группы. Используя единственный глобальный номер потока в ядре происходит покоординатное сравнение с использованием цикла и условного оператора if. В случае невыполнения хотя бы одного условия происходит выход из ядра. Если все условия выполнены, последнее значение, для одномерного преобразования, записывается в возвращаемый с ускорителя на хост вектор длины n. При этом в параметры ядра передается только два необходимых для сравнения вектора. Данная реализация обозначена на рис. 6 как GPU 1.

Во втором случае используется дополнительный массив типа $char$ аналогичный выборке. В данном массиве сохраняются знаки покоординатного сравнения, которое

производится по формуле $(x_i - min_i)\cdot(max_i - x_i)$. При запуске ядра в параметрах используются рабочие группы 16×16. Поскольку рассматриваемая задача имеет размерность 15, массив расширен до размера рабочей группы. Следующее ядро выполняет процедуру покоординатного сложения каждой строки дополнительного массива, так называемый SumReduction. При этом используется локальный массив и обеспечивается когерентность запросов. Полученные значения суммы записываются в вектор длины n и копируются с ускорителя на хост. Искомый вектор найден, если сумма битов равна текущей размерности задачи. Данная реализация обозначена на рис. 6 как GPU 2. Также на рис. 5 представлена схема распараллеливания.

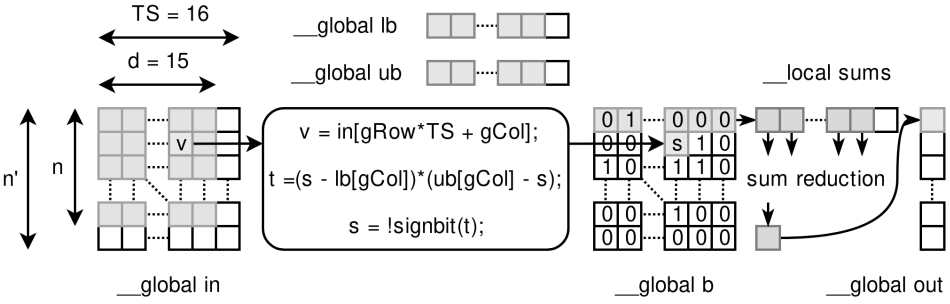
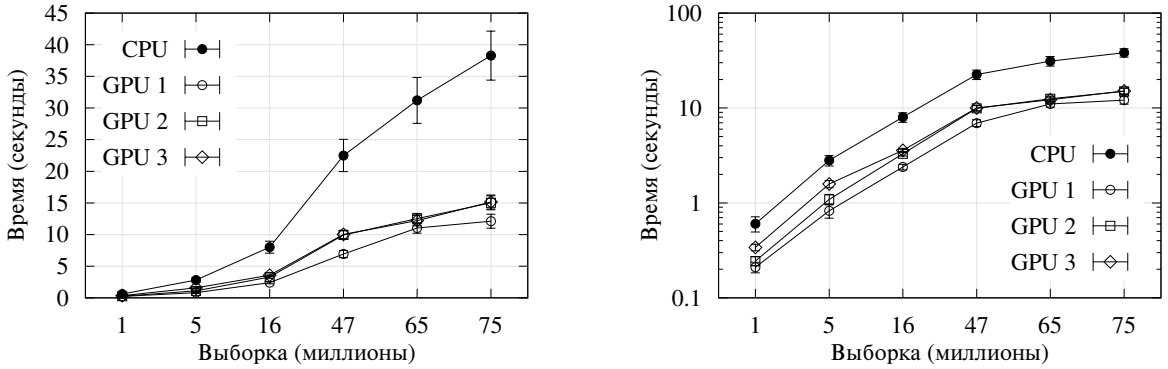


Рис. 5. Используемая схема распараллеливания GPU 2 с дополнительным массивом

В третьем случае исключается работа с дополнительным массивом. Ядро выполняет процедуру покоординатного сравнения аналогично GPU 2 и сразу же выполняет процедуру сложения полученных знаковых битов. При этом также используется локальный массив и обеспечивается когерентность запросов. Полученные значения суммы записываются в вектор длины n и копируются с ускорителя на хост. Данная реализация обозначена на рис. 6 как GPU 3.



а) Линейный масштаб б) Логарифмический масштаб

Рис. 6. Производительность применяемых схем параллельных вычислений

Результаты предложенных схем приведены на рис. 6. Для каждой выборки выполнено 10 запусков. Представлено среднее арифметическое время работы одного квантильного преобразования и среднеквадратическое отклонение. Несмотря на детерминированность получаемых выборок, предсказать их размер довольно трудно. Этим обусловлены представленные округленные размеры. В представленных результатах учитывается время полного квантильного преобразования, т.е. при каждом запуске произошло 15 проходов по

выборке и столько же раз с ускорителя на хост скопирован результирующий вектор. Также учитывается произведенное 15 раз одномерное квантильное преобразование.

Несмотря на полученное во всех случаях ускорение, оно остается приблизительно постоянным для каждой выборки. Максимально полученное ускорение дает первая, простейшая, реализация, со средним ускорением в $3,14$ раза. Худший результат со средним ускорением в $2,18$ раза дает версия с дополнительным массивом.

Причины плохой производительности следующие. Во первых, при каждом преобразовании 15 раз с ускорителя на хост копируется массив, равный размеру выборки. Во вторых, высокая скорость выполнения операции сравнения. В третьих, большое количество обращений к глобальной памяти ускорителя. В первой, простейшей реализации, меньше всего такого рода обращений, поскольку при невыполнении хотя бы одного условия происходит выход из функции. Этим обусловлено максимально полученное ускорение этой реализации.

Все вычисления выполнены на кластере ОИЯИ HybriLIT [11] с использованием одного ускорителя NVIDIA Tesla K40s. Необходимо отметить, что используемый вычислительный узел имеет три графических ускорителя. Используя дополнительные ресурсы можно несколько увеличить ускорение, разбив выборку на равные части.

3. Результаты численных экспериментов

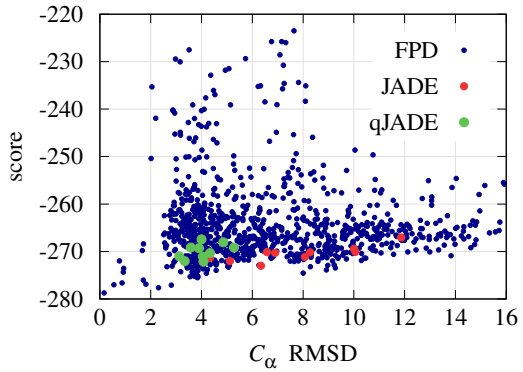
В настоящих экспериментах выполнялся докинг пептида DLLHI в белок 1JWG:B (см. рис. 1). Размерность задачи составила 54 параметра. Квантильное преобразование применялось к 15-ти параметрам, которые отвечают за положение пептида в определенной выше окрестности. Радиус сфер равен четырем ангстремам.

Сравнение адаптивной дифференциальной эволюции JADE [12] производилось с протоколом Rosetta FlexPepDock [2]. Условием применения данного протокола является присутствие пептида в радиусе пяти ангстрем от места связывания. Работа протокола включает в себя несколько различных этапов, заключительная стадия которого – алгоритм Монте-Карло с локальной оптимизацией. Выбор эволюционного алгоритма JADE обусловлен проведенными в [5, 6] исследованиями.

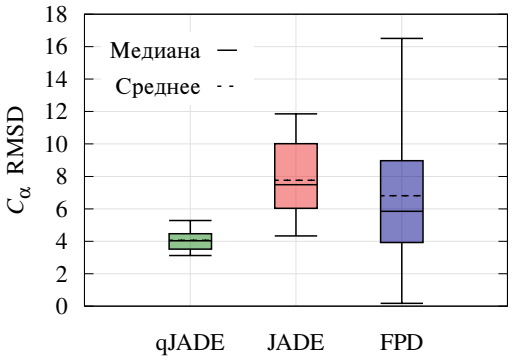
На рис. 7 представлены результаты использования алгоритма JADE, его модификации с использованием квантильной функции qJADE и протокола Rosetta FlexPepDock (FPD). Указано среднеквадратичное отклонение атомов главной цепи пептида относительно нативной структуры, которая прошла процедуру локальной оптимизации стандартными средствами пакета Rosetta. Нативное состояние комплекса имеет значение скоринг-функции приблизительно -280.

Начальная позиция пептида для эволюционных алгоритмов и FPD представляла собой перевернутый на 180 градусов относительно нативного состояния вдоль места связывания пептид. Приемлемым результатом докинга является субангстремное значение отклонения.

Результаты проведенных численных экспериментов показывают, что с поставленной задачей докинга справился только протокол FPD. Применением квантильного преобразования удалось сократить пространство поиска и добиться лучших результатов для эволюционного алгоритма JADE. Под сокращением пространства поиска имеется в виду не сокращение размерности, а удержание пептида в окрестности места связывания. На это указывает меньшее среднеквадратичное отклонение атомов пептида. Для худшего найденного значения результат изменился приблизительно в два раза. Однако в среднем



а) Среднеквадратичное отклонение атомов и значения скоринг-функции



б) Среднеквадратичное отклонение атомов главной цепи (диаграмма размаха)

Рис. 7. Результаты десяти независимых запусков JADE, qJADE и 10^3 запусков FlexPepDock

для лучших найденных значений энергия связывания практически аналогична. Это указывает и неспособность используемого эволюционного алгоритма оптимизации преодолеть сложный энергетический ландшафт.

В данном случае использовалась выборка размером приблизительно 75 миллионов. Построение такой выборки для приведенного пептида на одном узле заняло приблизительно 5 часов. Задаваемое количество вызовов целевой функции выбиралось из расчета сопоставимости времени вычислений.

Заключение

В результате выполненной работы проведена реализация многомерной эмпирической квантильной функции. Предложен сеточный подход к построению детерминированный выборки, произведена параллельная реализация и получено приемлемое ускорение. Произведена параллельная реализация квантильной функции.

На основании проведенных исследований можно заключить, что с помощью квантильной функции возможно свести задачу пептид-белок докинга в непрерывный единичный гиперкуб. При этом учитываются остальные параметры, которые также проходят процедуру преобразования [5]. Такая постановка задачи позволяет создать платформу для объективного сравнения различных алгоритмов глобальной оптимизации, таких как эволюционные, роевые, алгоритмы оценки распределений, алгоритмы со множественной оценкой и без оценки константы Липшица.

Недостатком использования квантильной функции является экспоненциальный рост выборки. Проведенные исследования показывают, что используемая размерность 15 параметров является верхней допустимой границей. Необходимо отметить, что во многих актуальных задачах пептид-белок докинга необходимо рассматривать пептиды длиной 10–15 аминокислотных остатков. Таким образом, размерность степеней свободы белка возрастает в 2–3 раза, что приводит к невозможности использования квантильной функции в текущей постановке.

Целью дальнейшей работы является расширение возможностей применения квантильной функции в задаче пептид-белок докинга. Смещение пептида определяет позицию первого α-углерода. В настоящей работе использовалось преобразование позиции пептида только в одну из двух ограничивающих сфер. Однако, довольно просто построить

отображение в две. Таким образом, можно рассматривать одновременно два потенциальных положения пептида в линейном интерфейсе связывания.

Важно отметить, что предложенный подход с применением квантильной функции может быть применен для широкого спектра задач со схожей формулировкой. Реализация аналога алгоритма заливки Flood-fill и реализация эмпирической квантильной функции доступны в открытом репозитории [13] сервиса GitHub.

Литература

1. Rentzsch R., Renard B.Y. Docking Small Peptides Remains a Great Challenge: An Assessment Using AutoDock Vina // Briefings in Bioinformatics. 2015. Vol. 16, No. 6. P. 1045–1056. DOI: 10.1093/bib/bbv008.

2. Raveh B., London N., et al. Rosetta FlexPepDock ab-initio: Simultaneous Folding, Docking and Refinement of Peptides onto Their Receptors // PLoS ONE. 2011. Vol. 6, No. 4. DOI: 10.1371/journal.pone.0018934.

3. Lopez-Camacho E., Garcia Godoy M.J., et al. Solving Molecular Flexible Docking Problems with Metaheuristics: A Comparative Study // Applied Soft Computing. 2015. DOI: 10.1016/j.asoc.2014.10.049.

4. Alford R.F., Leaver-Fay A., Jeliazkov R., et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. 2017. DOI: 10.1101/106054.

5. Полуян С.В., Ершов Н.М. Применение параллельных эволюционных алгоритмов оптимизации в задачах структурной биоинформатики // Вестник УГАТУ. 2017. Т. 21, № 4(78). С. 143–152.

6. Poluyan S., Ershov N. Parallel Evolutionary Optimization Algorithms for Peptide-Protein Docking // EPJ Web of Conferences. 2018. Vol. 173. P. 06010–06010. DOI: 10.1051/epjconf/201817306010.

7. Sellers M.S., Hurley M.M. XPairIt Docking Protocol for Peptide Docking and Analysis // Molecular Simulation. 2015. Vol. 42. P. 149–161. DOI: 10.1080/08927022.2015.1025267.

8. O'Brien G.L. The Comparison Method for Stochastic Processes // The Annals of Probability. 1975. Vol. 3, No. 1. P. 80–88. DOI: 10.1214/aop/1176996450.

9. Einmahl J.H.J., Mason D.M. Generalized Quantile Processes // The Annals of Statistics. 1992. Vol. 20, No. 2. P. 1062–1078. DOI: 10.1214/aos/1176348670.

10. Vučković V., Arizanović B., Le Blond S. Generalized N-way Iterative Scanline Fill Algorithm for Real-Time Applications // Journal of Real-Time Image Processing. 2017. DOI: 10.1007/s11554-017-0732-1.

11. Heterogeneous Platform HybriLIT. URL: http://hlit.jinr.ru/en/ (дата обращения: 03.11.2018).

12. Zhang J., Sanderson A. JADE: Adaptive Differential Evolution with Optional External Archive // IEEE Transactions on Evolutionary Computation. 2009. Vol. 13, No. 5. P. 945–958. DOI: 10.1109/TEVC.2009.2014613.

13. GitHub repositories. URL: https://github.com/poluyan (дата обращения: 03.11.2018).

Полуян Сергей Владимирович, аспирант, кафедра распределенных информационно-вычислительных систем, институт системного анализа и управления, Государственный университет «Дубна» (Дубна, Российская Федерация)

Ершов Николай Михайлович, к.ф.-м.н., с.н.с., кафедра автоматизации научных исследований, факультет вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (Москва, Российская Федерация)

DOI: 10.14529/cmse190204

USING MULTIVARIATE QUANTILE FUNCTION FOR PEPTIDE-PROTEIN DOCKING

© 2019 S.V. Poluyan[1], N.M. Ershov[2]

[1]*Dubna State University (Universitetskaya 19, Dubna, 141982 Russia),*

[2]*Lomonosov Moscow State Universisty (GSP-1, Leninskie Gory 1, Moscow, 119991 Russia)*

E-mail: svpoluyan@gmail.com, ershov@cs.msu.ru

Received: 03.11.2018

The paper presents an exploration of using evolutionary optimization algorithms in protein-peptide docking. The main assumptions that reduce docking to the continuous global optimization problem are described. Some special features of the given problem and the difficulties of using evolutionary algorithms are discussed. The paper provides a way of using evolutionary optimization algorithms based on using empirical quantile function. The multivariate quantile function structure is defined recursively using univariate quantile transform. The grid-based approach of using quantile function is presented. The disadvantages of this approach are indicated. The deterministic sampling algorithm is proposed. The used scheme of parallel sampling and the resulting speed-up are described. The GPU-accelerated approach for quantile function evaluation is presented. This paper provides multiple GPU-based ways which use a sample in explicit form. Their speed-up depending on sample size is shown. The paper presents the results of docking using an evolutionary algorithm and its quantile-function-based modification. The comparison with the relevant docking method within a particular force-field is made. The results of the experiments are analyzed.

Keywords: global optimization, evolutionary algorithms, empirical quantile function, docking.

FOR CITATION

Poluyan S.V., Ershov N.M. Using Multivariate Quantile Function for Peptide-Protein Docking. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2019. vol. 8, no. 2. pp. 63–75. (in Russian) DOI: 10.14529/cmse190204.

References

1. Rentzsch R., Renard B.Y. Docking Small Peptides Remains a Great Challenge: An Assessment Using AutoDock Vina. *Briefings in Bioinformatics.* 2015. vol. 16, no. 6. pp. 1045–1056. DOI: 10.1093/bib/bbv008.

2. Raveh B., London N., et al. Rosetta FlexPepDock ab-initio: Simultaneous Folding, Docking and Refinement of Peptides onto Their Receptors. *PLoS ONE.* 2011. vol. 6, no. 4. DOI: 10.1371/journal.pone.0018934.

3. Lopez-Camacho E., Garcia Godoy M.J., et al. Solving Molecular Flexible Docking Problems with Metaheuristics: A comparative study. *Applied Soft Computing.* 2015. DOI: 10.1016/j.asoc.2014.10.049.

4. Alford R.F., Leaver-Fay A., Jeliazkov R., et al. The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design. 2017. DOI: 10.1101/106054.

5. Poluyan S.V., Ershov N.M. Parallel Evolutionary Algorithms for Solving Optimization Problems in Structural Bioinformatics. *Vestnik Ufimskogo gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta* [Bulletin of the Ufa State Aviation Technical University]. 2017. vol. 21, no. 4(78). pp. 143–152. (in Russian)

6. Poluyan S., Ershov N. Parallel Evolutionary Optimization Algorithms For Peptide-Protein Docking. *EPJ Web of Conferences.* 2018. vol. 173. pp. 06010–06010. DOI: 10.1051/epjconf/201817306010.

7. Sellers M.S., Hurley M.M. XPairIt Docking Protocol for Peptide Docking and Analysis. *Molecular Simulation.* 2015. vol. 42. pp. 149–161. DOI: 10.1080/08927022.2015.1025267.

8. O'Brien G.L. The Comparison Method for Stochastic Processes. *The Annals of Probability.* 1975. vol. 3, no. 1. pp. 80–88. DOI: 10.1214/aop/1176996450.

9. Einmahl J.H.J., Mason D.M. Generalized Quantile Processes. *The Annals of Statistics.* 1992. vol. 20, no. 2. pp. 1062–1078. DOI: 10.1214/aos/1176348670.

10. Vučković V., Arizanović B., Le Blond S. Generalized N-way Iterative Scanline Fill Algorithm for Real-Time Applications. *Journal of Real-Time Image Processing.* 2017. DOI: 10.1007/s11554-017-0732-1.

11. Heterogeneous Platform HybriLIT. Available at: http://hlit.jinr.ru/en/ (accessed: 03.11.2018).

12. Zhang J., Sanderson A. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation.* 2009. vol. 13, no. 5. pp. 945–958. DOI: 10.1109/TEVC.2009.2014613.

13. GitHub repositories. Available at: https://github.com/poluyan (accessed: 03.11.2018).

УДК 004.052.3 DOI: 10.14529/cmse190205

КООРДИНИРОВАННОЕ СОХРАНЕНИЕ С ЖУРНАЛИРОВАНИЕМ ПЕРЕДАВАЕМЫХ ДАННЫХ И АСИНХРОННОЕ ВОССТАНОВЛЕНИЕ В СЛУЧАЕ ОТКАЗА*

© 2019 А.А. Бондаренко, П.А. Ляхов, М.В. Якобовский
Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047 Москва, Миусская пл., д. 4)
E-mail: bondaleksey@gmail.com, pavel.lyakhov@phystech.edu, lira@imamod.ru
Поступила в редакцию: 20.11.2018

Увеличивающийся рост числа компонент суперкомпьютеров приводит специалистов в области HPC к неблагоприятным оценкам для будущих суперкомпьютеров: диапазон среднего времени между отказами будет составлять от 1 часа до 9 часов. Данная оценка ставит под вопрос возможность проведения длительных расчетов на суперкомпьютерах. В работе предлагается метод восстановления после отказов, не требующий возврата большинства процессов к последней контрольной точке, что может позволить сократить накладные расходы для некоторых вычислительных алгоритмов. Стандартный метод обеспечения отказоустойчивости заключается в координированном сохранении, а в случае отказа осуществляется возврат всех процессов к последней контрольной точке. Предлагаемая стратегия заключается в координированном сохранении и журналировании передаваемых данных, а в случае отказа происходит асинхронное восстановление. При асинхронном восстановлении несколько запасных процессов проводят пересчет данных потерянных после отказа, а остальные процессы находятся в ожидании окончания процедуры восстановления потерянных данных. Разработаны параллельные программы, решающие задачу о распространении тепла в тонкой пластине. В данных программах отказы происходят после вызова функции raise (SIGKILL), а координированное или асинхронное восстановление осуществляется с помощью функционала ULFM. Для получения теоретических оценок накладных расходов предложен имитационный метод, моделирующий исполнение программы с отказами. В данном методе отказ может произойти во время расчетов, а также во время сохранения контрольных точек или в ходе восстановления. Проведено сравнение методов восстановления при разных значениях частоты отказов для задачи распространения тепла в тонкой пластине, в которой объем данных для журналирования незначителен. Сравнение показало, что применение асинхронного восстановления приводит к сокращению накладных расходов от 22 % до 40 % при теоретической оценке и от 13 % до 53 % в вычислительном эксперименте.

Ключевые слова: MPI, расширение ULFM, контрольные точки, координированное сохранение, асинхронное восстановление, отказоустойчивость.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Бондаренко А.А., Ляхов П.А., Якобовский М.В. Координированное сохранение с журналированием передаваемых данных и асинхронное восстановление в случае отказа // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 76–91. DOI: 10.14529/cmse190205.

Введение

В начале двухтысячных годов проходило масштабное наблюдение за отказами в вычислительных системах Лос-Аламоса. В нем было задействовано 22 кластера и около 5000

*Работа рекомендована программным комитетом международной конференции «Суперкомпьютерные дни в России 2018»

узлов. Результаты исследования показали, что за год в среднем в пересчете на один процессор приходится от 0,1 до 0,25 отказа в системе [1]. Наблюдения за использованием вычислительных систем Терафлопсного уровня показали, что время между отказами/прерываниями измеряется десятками часов; например, для некоторых машин эта величина была между 6,5 и 40 часами [2]. Работа с Blue Waters — одним из суперкомпьютеров Петафлопсного уровня [3] — показала, что в среднем каждые 4,2 часа происходили сбои, которые требовали локального исправления, а каждые 160 часов происходили сбои, требующие корректирования работы всего компьютера в целом. Современные суперкомпьютеры сложно устроены и включают в себя сотни тысяч, а некоторые даже миллионы ядер, десятки тысяч процессоров и тысячи узлов и, как правило, различные ускорители. Поэтому прогнозы специалистов для будущих суперкомпьютеров не утешительны, даже если удастся увеличивать время бесперебойной работы для составляющих компонентов, в целом для суперкомпьютеров среднее время между отказами будет между 1 и 9 часами [4].

Поэтому для проведения высокопроизводительных вычислений представляется важным решение следующей задачи: разработать принципы сохранения контрольных точек за время, меньшее характерной продолжительности безотказной работы системы, и алгоритмы, обеспечивающие, в случае отказа части оборудования, быстрое автоматическое возобновление расчета на работоспособной части вычислительного поля. В данной работе рассматривается модификация многоуровневого метода обеспечения отказоустойчивости, которая заключается в координированном сохранении и журналировании передаваемых данных, а в случае отказа происходит асинхронное восстановление. Предлагаемая стратегия позволяет сократить накладные расходы на обеспечение отказоустойчивости расчетов для некоторых вычислительных алгоритмов.

В первом разделе изложен многоуровневый метод обеспечения отказоустойчивости и предлагается модификация метода восстановления. Во втором разделе приводятся: особенности реализации параллельной программы, которая решает задачу о распределении тепла в тонкой пластине; имитационный метод, моделирующий исполнение программы с отказами, который позволяет получить теоретические оценки накладных расходов. В заключительном разделе приводятся результаты вычислительных экспериментов, а именно оценки накладных расходов для разных стратегий восстановления.

1. Стратегии сохранения и восстановления после отказов для многоуровневого метода обеспечения отказоустойчивости

Первый раздел посвящен методам обеспечения отказоустойчивых вычислений. Применение существующих средств сохранения контрольных точек на системном уровне, основанных на BLCR [5], показало ограниченность данного подхода для больших систем [6]. Наиболее перспективным считается подход, заключающийся в передаче пользователю функций по обработке отказа, так как пользователь сможет существенно сократить объем контрольных точек. Одним из таких методов является многоуровневый метод, который также позволяет применять разные стратегии отказоустойчивости в зависимости от информации об отказах в системе.

1.1. Многоуровневый метод обеспечения отказоустойчивости

Многоуровневый метод позволяет объединить разные алгоритмы [4, 6–10] обеспечения отказоустойчивости, в частности, разнообразные методы сохранения контрольных точек, что позволяет адаптироваться под различные типы отказов. Как правило, каждый уровень соответствует конкретному типу отказа и связан со стратегией хранения, которая позволяет провести восстановление после этого типа отказа.

При описании многоуровневого метода [11] принимаются следующие допущения:

– в системе могут происходить отказ от 1 до k уровней;
– наступление отказов разных уровней — независимые случайные величины;
– отказ j-го уровня имеет свою частоту отказа λ_j, свои накладные расходы на сохранение контрольной точки C_j и на восстановление после отказа R_j;
– отказ j-го уровня уничтожает все контрольные точки на нижних уровнях от 1 до j–1 и приводит к необходимости восстановления с уровня j или выше;
– восстановление после отказа j-го уровня должно восстановить данные контрольных точек всех нижних уровней;
– частота отказов убывает, а накладные расходы на сохранение и восстановление возрастают с ростом уровня, то есть $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$, $C_1 \leq C_2 \leq ... \leq C_k$ и $R_1 \leq R_2 \leq ... \leq R_k$.

Замечание 1. Ниже будет использоваться следующие выражения: «отказ j-го уровня», «сохранение j-го уровня», «восстановление j-го уровня». Под данными выражениями следует понимать, что в многоуровневом методе выбран набор отказов для каждого уровня, а также были определены стратегии сохранения и восстановления для соответствующего отказа, которые удовлетворяют допущениям описанным выше.

Замечание 2. Важной задачей применения многоуровневых методов является задача определения оптимальных периодов сохранения контрольных точек для каждого уровня. Один из подходов к решению данной проблемы заключается в составлении специальных шаблонов сохранения контрольных точек всех уровней на заданном участке времени и последующее тиражирование данного шаблона. В данной работе вопросы оптимизации периодов сохранения и составления шаблона не рассматриваются. Более подробно с данными вопросами можно ознакомиться в работах [12, 13].

Одним из примеров многоуровневых методов обеспечения отказоустойчивости служит трехуровневый метод, в котором авторы работы [12] выбрали следующие уровни отказов:

– отказ первого уровня — временные ошибки в памяти;
– отказ второго уровня — сбои в узлах, которые делают недоступными данные из оперативной памяти;
– отказ третьего уровня — сбои, приводящие к недоступности данных из оперативной памяти и данные из резервных копий соседних узлов.

Сохранение ключевых данных для первого уровня будет осуществляться в оперативную память, для второго уровня осуществляется дублирование данных некоторым соседям, для третьего уровня сохранение осуществляется в распределенную файловую систему.

1.2. Координированное сохранение и восстановление для многоуровневого метода

Координированное сохранение подразумевает сначала остановку расчетов и связанных с ними коммуникационных функций, и последующее копирование контрольных точек на выбранные устройства хранения. Каждый многоуровневый метод требует выбора своего набора стратегий сохранения данных исходя из параметров: наличие в вычислительной системе локальных (оперативная память, SSD, HDD) и глобальных средств хранения; скорости чтения/записи данных; объема контрольных точек; информации о частоте отказов. Поэтому выбор шаблона сохранения будет зависеть от применяемого вычислительного алгоритма и характеристик вычислительной системы.

На рис. 1 представлен шаблон сохранения контрольных точек для трех уровней, где по оси абсцисс время исполнения программы, по оси ординат уровни сохранения, черными квадратами отмечены моменты сохранения контрольных точек j-го уровня, P_j — периоды сохранения для соответствующего уровня. Для упрощения представления на рис. 1 время сохранения контрольных точек равно нулю.

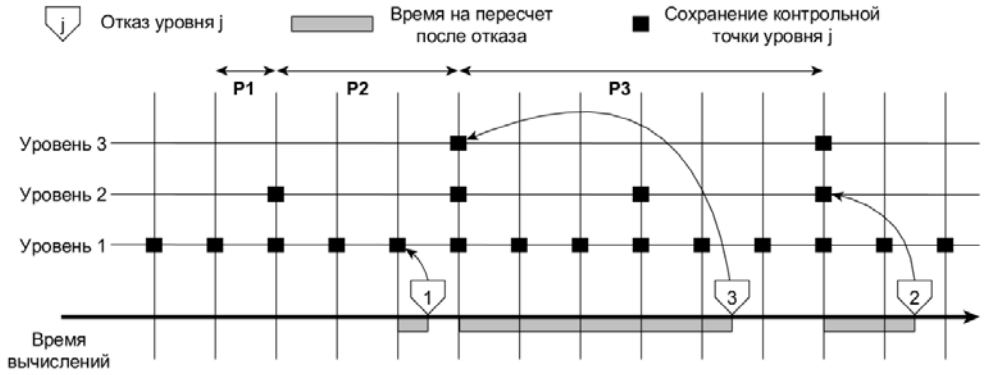


Рис. 1. Трехуровневый шаблон сохранения контрольных точек

В случае обнаружения отказа в системе необходимо осуществить запуск процедуры восстановления, которая описана на рис. 2. Первый шаг позволяет восстановить MPI среду для продолжения вызова коммуникационных функций. Если можно провести декомпозицию всех данных из последней контрольной точки на множество процессов избежавших отказа, то выполнять второй шаг не надо. Однако в данной работе предполагается, что такая декомпозиция не проводится и отказавшие процессы замещаются процессами, которые не принимали участие в расчете с самого начала, то есть были запасными. После второго шага часть процессов, принимавшая участие в расчете и избежавшая отказа, будет помнить состояние (расчетные данные) в момент отказа, но новый процесс, вновь введенный вместо отказавшего процесса, будет иметь состояние (расчетные данные) из последней контрольной точки. Для продолжения расчета необходимо осуществить синхронизацию состояний. Либо состояния расчетных процессов избежавших отказа вернуть к состояниям из последней контрольной точки, либо состояние нового процесса довести до состояния расчетных процессов избежавших отказа. Под координированным восстановлением понимают, возврат всех процессов к состояниям из последней контрольной точки, в этом и состоит третий шаг, описанный на рис. 2. Четвертый шаг связан с окончанием процедуры восстановления и возвратом к нормальному ходу вычислительного процесса.

Шаг 1. Перенастройка MPI среды для нормальной работы коммуникационных функций.
Шаг 2. Взамен отказавшим процессам для проведения расчетов вводятся резервные процессы (или новые процессы, если возможна генерация процессов), таким образом, что новые процессы имеют номера отказавших процессов.
Шаг 3. Теперь для возобновления работы все процессы должны осуществить чтение данных из последней контрольной точки.
Шаг 4. Осуществляется запуск продолжения расчетов.

Рис. 2. Алгоритм координированного восстановления после отказа

На рис. 1 приведены возможные отказы разных уровней, для которых стрелки указывают на контрольные точки, из которых необходимо будет производить координированное восстановление. Серым цветом на рис. 1 указано время потерянных вычислений, то есть время с последней контрольной точки восстановления до момента отказа. При координированном восстановлении, после возврата к контрольной точке, происходит повторный пересчет времени указанного серым цветом.

1.3. Координированное сохранение с журналированием и асинхронное восстановление для многоуровневого метода

Основная цель данной работы заключается в уменьшении накладных расходов расчетов, которые проходят на вычислительной системе с частыми отказами. Представляет интерес вопрос: «Как влияет синхронизация состояний процессов в процедуре восстановления на общий объем накладных расходов?» Было решено исследовать переход на асинхронное восстановление, когда синхронизация осуществляется за счет того, что состояние нового процесса доводят до состояния расчетных процессов избежавших отказа. Для этого предлагается использовать несколько запасных процессов, которые должны осуществить пересчет потерянного времени («серая область» на рис. 1).

Отметим, что с момента сохранения последней контрольной точки до наступления отказа, процессы в общем случае будут обмениваться информацией. Чтобы запасные процессы могли выполнить пересчет потерянных данных им будут нужна информация о том «когда», «какой процесс», «какие данные», предавал отказавшему процессу в период между последним сохранением до наступления отказа. Сохранение этой информации обозначим как журналирование (logging) передаваемых данных. Журналирование требует от каждого процесса сохранения: момента передачи, передаваемых данных и адресата передачи.

В рамках данной работы рассматривается асинхронное восстановление только для отказа первого уровня, что позволяет ограничить область хранения журнальных данных до периода сохранения контрольных точек на первом уровне. Таким образом, должно осуществляться координированное сохранение контрольных точек согласно выбранному шаблону, при этом каждый процесс обязан журналировать передаваемые им данные в течении P_1. На следующем периоде, журнальные данные с предыдущего периода не нужны и они могут быть удалены или перезаписаны.

Для реализации асинхронности в процедуре восстановления (рис. 2) необходимо заменить третий шаг (рис. 3).

Шаг 3. Процессы, избежавшие отказа, делятся на три группы.

- Первая группа – процессы, которые с момента последнего сохранения контрольной точки осуществляли передачу данных отказавшему процессу.
- Вторая группа – резервные процессы, которые будут осуществлять пересчет данных, потерянных в результате отказа.
- Третья группа – остальные процессы, которые будут ожидать окончание процедуры восстановления.

Процессы первой группы осуществляют передачу журнальных данных процессам из второй группы. Процессы второй группы осуществляют пересчет потерянных расчетов с последней контрольной точки до момента отказа и принимают необходимые журнальные данные.

Рис. 3. Шаг 3 алгоритма асинхронного восстановления после отказа

Отметим, что на пересчет выделяются резервные процессы в кратном размере, чтобы ускорить процесс восстановления по сравнению с восстановлением за счет одновременного возврата всех процессов к последней контрольной точке.

Замечание 3. Пусть $V_{жур}$ — объем передаваемых данных одним процессом за вычислительную итерацию, L — количество вычислительных итераций в течении P_1, то есть между сохранениями контрольных точек на первом уровне. Если произведение $V_{жур}{\cdot}L$ достаточно мало, чтобы поместиться в оперативную память вместе с другими данными для расчетов, то можно осуществлять журналирование целиком в оперативную память вычислительных узлов. Таким образом, время на журналирование несущественно и его можно считать равным нулю. Однако если $V_{жур}{\cdot}L$ имеет существенный объем, то потребуется разбить L на части для периодической записи журналов передаваемых данных на локальные устройства хранения (SSD/HDD), что приведет к необходимости оценивать соответствующие накладные расходы. В рамках данной работы время на журналирование считается равным нулю, а оценка накладных расходов на журналирование для различных вычислительных алгоритмов остается задачей для будущих работ.

2. Оценка накладных расходов

Для сравнения методов обеспечения отказоустойчивости, описанных в первом разделе, реализованы параллельные программы, решающие тестовую задачу. В программах отказ является случайной величиной, принадлежащий экспоненциальному распределению, который реализуются вызовом функции raise (SIGKILL). Обработка отказа (рис. 2) происходит с помощью функционала ULFM. Однако такой подход не позволяет получить большую выборку накладных расходов, поэтому в работе предлагается имитационный метод, моделирующий исполнение программы, для оценки накладных расходов. Особенность метода состоит в том, что отказ может произойти во время расчетов, а также во время сохранения контрольных точек или в ходе восстановления. Данный метод позволяет моделировать многократное исполнение прикладной программы со случайными отказами.

2.1. Программная реализация решения краевой задачи распределения тепла в тонкой прямоугольной пластине с отказами процессов

В данной работе для вычислительного эксперимента выбрана краевая задача распределения тепла в тонкой прямоугольной пластине. Для проведения вычислений применяется явная разностная схема и геометрический метод для параллельной реализации, так чтобы каждый процесс получал одинаковое количество узлов начальной сетки. Общий объем сетки составляет 256 миллионов узлов. Итерационный процесс вычисления сопровождается измерениями времени, которое играет ключевую роль для определения наступления события: сохранения контрольной точки в оперативную память или в распределенную файловую систему, а также наступление отказа. Вычисления продолжается до тех пор, пока не будет выполнен заранее запланированный объем расчетов ($T_з$). Общий алгоритм программ представлен на рис. 4. Наличие отказа в системе не мешает автономной работе каждого из процессов, а детекция отказа происходит только при выполнении коммуникационных функций, поэтому в данной работе запуск процедуры восстановления осуществляется только на третьем, пятом и седьмом шагах.

Выбраны следующие отказы: отказ первого уровня — незначительные сбои в системе, для которых достаточно данных, хранящихся в оперативной памяти соседнего процесса, отказ второго уровня — более сложные сбои, для восстановления после которых необхо димо хранить данные в распределенной файловой системе. Сохранение контрольных то-

Шаг 1. Каждый процесс проводит свои вычисления.
Шаг 2. Проверяется условие наступления отказа. В случае положительного ответа, один из процессов выполняет функцию raise(SIGKILL).
Шаг 3. Процессы производят обмен данными. Если при выполнении коммуникационных функций обнаружен отказ, то не отказавшие процессы выполняют процедуру восстановления.
Шаг 4. Проверяется условие наступления отказа. В случае положительного ответа, один из процессов выполняет функцию raise(SIGKILL).
Шаг 5. Происходит сохранение контрольных точек при выполнении соответствующих условий. Если в коммуникационных функциях обнаружен отказ, то не отказавшие процессы выполняют процедуру восстановления.
Шаг 6. Проверяется условие наступления отказа. В случае положительного ответа, один из процессов выполняет функцию raise(SIGKILL).
Шаг 7. Проверяется продолжение расчетов и происходит переход на шаг 1. Если при выполнении коммуникационных функций обнаружен отказ, то не отказавшие процессы выполняют процедуру восстановления.

Рис. 4. Алгоритм основной части программы с отказами

чек для первого уровня осуществляется в оперативную память соседнего процесса, а для второго уровня — в распределенную файловую систему.

Разработаны две программы с разными стратегиями обеспечения отказоустойчивости. Первая программа реализует координированное сохранение по шаблону для двух уровней, а в случае отказа восстановление происходит за счет координированного возврата всех процессов к последней контрольной точке соответствующего уровня. Вторая

программа реализует координированное сохранение по шаблону и журналирование передаваемых данных. В случае отказа первого уровня осуществляется асинхронное восстановление с двумя или пятью резервными процессами для пересчета, а в случае отказа второго уровня восстановление происходит за счет координированного возврата всех процессов к последней контрольной точке.

Для реализации многоуровневого метода необходимо определить ключевые параметры метода. Выбираются отказы и соответствующие алгоритмы сохранения и восстановления. Пусть известно среднее время между отказами (MTBF). Для выбранной вычислительной задачи надо определить набор ключевых данных, необходимых для осуществления восстановления в случае отказа. После этого определяются характеристики записи контрольных точек для каждого уровня C_i, а также значения времени восстановления системы и чтения этих данных с соответствующего носителя R_i. Далее необходимо определить параметры шаблона сохранения этих данных, а именно длину шаблона и количество сохранений контрольных точек разных уровней (W^{onm}, N_i^{onm}). Решение этой задачи описано в работе [12] (рис. 5).

```
void opt_par(double *lam, double *N^opt, double *C, int count, double &W^opt){
     double S_bot(0), S_top(0);
     for(int i=0;i<count;i++){
          N^opt [i] = sqrt(C(count-1)*lam(i)/lam(count-1)/C(i));
          S_top = S_top + N^opt (i)*C(i);
          S_bot = S_bot + lam(i)/N^opt(i);
     }
     W^opt = sqrt(2*S_top/S_bot);
}
```

Рис. 5. Процедура определения оптимальных параметров шаблона сохранения

Отказы рассматриваются как случайные величины, описываемые экспоненциальным распределением. Для реализации набора отказов в расчете используется массив времен, заполненный по алгоритму (рис. 6), который осуществляет генерацию случайных величин, следующих друг за другом и подчиняющихся экспоненциальному распределению. Когда время работы программы превышает время следующего отказа, то в системе фиксируется отказ и осуществляется запуск процедуры восстановления.

```
void generate_failures(float *Failures,int count, float lambda){
     double p(0),q(0);
     for(int i=0;i<count;i++){
          q=(double)rand()/(double)RAND_MAX;
          p=p-log(q)/lambda;
          Failures[i]=p;
     }
}
```

Рис. 6. Процедура генерации массива случайных чисел, следующих друг за другом и подчиняющихся экспоненциальному распределению

Стандартная версия MPI (MPI 3.1) не содержит средств обработки отказов при коммуникационных операциях. Для обнаружения отказа в системе, распространения информации об ошибке и последующем восстановлении среды MPI используются функции ULFM [14]: MPIX_Comm_agree(), MPIX_Comm_revoke(), MPIX_Comm_shrink(). Примеры реализации программ с ULFM и описанием его функционала можно найти в [14, 15].

2.2. Имитационный метод, моделирующий исполнение параллельной программы подверженной отказам

Имитационный метод, описывающий исполнение параллельной программы подверженной отказам, учитывает возможность наступления случайных отказов во время сохранения контрольных точек и во время восстановления расчетов, а также позволяет сравнивать различные техники обеспечения отказоустойчивости. Метод заключается в разделении времени исполнения программы на небольшие части величиной Δt и последовательном итерировании времени исполнения программы с шагом Δt. Такие события как «сохранение контрольной точки» или «наступление отказа» происходят, когда время работы программы не меньше времени наступления соответствующего события.

В основе предлагаемой имитационной модели лежат следующие положения.

– Программа моделирования должна оперировать с двумя переменными, описывающими время: $T_{\text{выч}}$ — время вычислений основной части кода, без учета накладных расходов на обеспечение отказоустойчивости, $T_{\text{тек}}$ — текущее время работы программы, включающее накладные расходы на обеспечение отказоустойчивости.
– Моделирование исполнения программы с отказами продолжается пока не будет выполнен весь заранее запланированный расчет $T_{\text{з}}$.
– По текущему времени работы $T_{\text{тек}}$ определяется наступление отказа в системе и сохранение контрольных точек.
– Ход выполнения программы осуществляется с некоторым шагом по времени Δt.

На рис. 7 представлен алгоритм имитационного метода описания исполнения параллельной программы подверженной отказам. Функция get time for saves() определяет, происходит ли на данной итерации запись контрольной точки на каком-либо уровне, и в соответствии с этим возвращает TFS — время на сохранение контрольных точек для текущей итерации. Функция get failure info () определяет FI — уровень и время отказа на данной итерации. Изменение переменных $T_{\text{выч}}$, $T_{\text{тек}}$ происходит с помощью функций get comp difference(), get cur difference().

```
while Tвыч<Tз do
     TFS = get time for saves (Tвыч,Δt,Ci)
     FI = get failure info(Tтек,Δt,TFS)
     Tвыч = Tвыч+get comp difference (Tвыч,Δt,FI)
     Tтек = Tтек+get cur difference (Tвыч,Tтек,Δt,TFS,FI,Ri)
end while
return Tтек
```

Рис. 7. Алгоритм имитационного моделирования программы с отказами

Приведем значения функций get comp difference(), get cur difference() при координированном восстановлении после отказа. Пусть шаг одной итерации по времени будет равен периоду сохранения контрольных точек на первом уровне ($\Delta t = P_1$). В табл. 1 приведены значения этих функций для ситуаций с отказами разных уровней представленных на рис. 8.

При отсутствии отказа на данной итерации, время основных вычислений ($T_{\text{выч}}$) должно увеличиться на значение шага $\Delta t = P_1$, а для текущего времени работы ($T_{\text{тек}}$) необходимо дополнительно учесть накладные расходы на сохранение, то есть

$$\text{get comp difference}() = \Delta t,\ \text{get cur difference}() = \Delta t + TFS.$$

В случае отказа на данной итерации, время основных вычислений ($T_{\text{выч}}$) должно вернуться к значению последней контрольной точки, то есть уменьшиться на значение $x{\cdot}\Delta t$, где x — число потерянных итераций. Текущее время работы ($T_{\text{тек}}$) должно увеличиться на время восстановление системы после отказа и на время вычислений сделанных в рамках этой итерации Y(«серое время» на рис. 8), то есть

get comp difference() = -$x{\cdot}\Delta t$, get cur difference() = R_i + Y.

Таблица 1

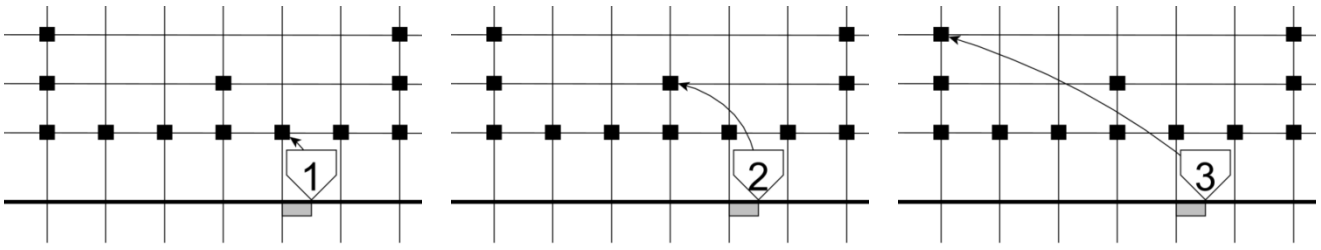
Значения функций при координированном восстановлении

	Нет отказа	Отказ первого уровня	Отказ второго уровня	Отказ третьего уровня
get comp difference()	Δt	0	-Δt	-4Δt
get cur difference()	Δt + *TFS*,	R_i +Y	R_i +Y	R_i +Y

Для получения предварительной оценки «максимального эффекта» от асинхронности восстановления в имитационном методе моделирования предполагается, что применение k процессов для пересчета позволит сократить время самого пересчета в k раз. Данное допущение позволяет получить верхнюю оценку сокращения накладных расходов обеспечения отказоустойчивости от применения асинхронного восстановления.

Замечание 4. В общем случае ускорение пересчета потерянных данных после отказа будет зависеть от многих факторов, а для некоторых вычислительных алгоритмов представлять отдельную задачу из-за сложности самого алгоритма или сложности декомпозиции данных, с которыми работал отказавший процесс.

В алгоритме имитационного метода при асинхронном восстановлении изменятся только значения функций get comp difference(), get cur difference(). Соответствующие примеры для ситуаций из рис. 8 представлены в табл. 2. Шаг одной итерации равен периоду сохранения контрольных точек на первом уровне ($\Delta t = P_1$).



а) отказ первого уровня б) отказ второго уровня в) отказ третьего уровня

Рис. 8. Примеры отказов разных уровней

При отсутствии отказа на данной итерации, расчетное время должно увеличиться на значение шага $\Delta t = P_1$, а в текущее время работы дополнительно надо учесть накладные расходы на сохранение, то есть

get comp difference() = Δt, get cur difference() = Δt + *TFS*.

В случае отказа на конкретной итерации необходимо завершить восстановление, поэтому время основных вычислений на данной итерации не будет увеличиваться. А для текущего времени работы должны быть учтены накладные расходы на восстановление после отказа (R_i) и время на синхронизацию состояний процессов (x·Δt/k+Y/k), так как

потерянным будет время от последней контрольной точки до момента наступления отказа (x·Δt+Y), а при пересчете будут участвовать k процессов.

get comp difference() = Δt, get cur difference() = R_i +(x·Δt+Y)/k.

Таблица 2

Значения функций при асинхронном восстановлении

	Нет отказа	Отказ первого уровня	Отказ второго уровня	Отказ третьего уровня
get comp difference()	Δt	0	0	0
get cur difference()	Δt + *TFS*,	R_i + Y/k	R_i + (Δt+Y)/k	R_i + (4Δt+Y)/k

В рамках данной работы были реализованы алгоритмы имитационного моделирования работы программы с координированным восстановлением и асинхронным восстановлением с двумя ($k = 2$) и пятью ($k = 5$) процессами принимающими участие в пересчете. В этих программах шаг итерации Δt не зависит от периодов сохранения и принимает значение 0,5 секунды.

3. Вычислительный эксперимент

В рамках вычислительного эксперимента были выбраны следующие пары величин среднего времени между отказами (MTBF) для отказа первого и второго уровня [$T_з$/2 10·$T_з$], [$T_з$/4 $T_з$], [$T_з$/10 $T_з$/2], [$T_з$/15 $T_з$/3], [$T_з$/20 $T_з$/4], где $T_з$ — время заранее запланированного счета $T_з$ = 3600 с. Для выбранной сетки время сохранения контрольных точек составляет 1 с и 6 с для первого и второго уровня соответственно, время восстановления после отказов первого и второго уровней равны 0,5 с и 4 с.

Для параллельных программ, реализующих вычислительный алгоритм расчета распределения тепла, было проведено 10 вычислительных экспериментов для каждого набора параметров, а для программ, реализующих имитационный метод моделирования, было проведено 10000 экспериментов.

В табл. 3 приведены параметры, описывающие полученные значения накладных расходов: $\bar{t}$ — среднее время накладных расходов, то есть среднее время работы программы с отказами минус время запланированного счета (3600 с); σ — среднее квадратическое отклонение накладных расходов; $\bar{n}_1$, $\bar{n}_2$ — средние значения числа отказов первого и второго уровней; ΔT — разность средних времен накладных расходов для программ с координированным восстановлением и с асинхронным восстановлением; % — отношение ΔT к среднему значению накладных расходов при координированном восстановлении выраженное в процентах; k — число резервных процессов, принимающих участие в пересчете потерянных данных из-за отказа.

Теоретические оценки средних значений накладных расходов, полученные имитационным методом, показывают, что применение асинхронного метода восстановления приводит к сокращению не менее чем на 22 % при двух процессах для пересчета, а при пяти процессах для пересчета не менее чем на 37 %.

Оценки средних значений накладных расходов, полученные в результате работы параллельных программ с отказами, показывают, что применение асинхронного ме-тода вос становления приводит к сокращению от 13 % до 47 % при двух процессах для пере-счета, а при пяти процессах для пересчета от 23 % до 53 %. Значения средних времен между отказами были выбраны в достаточно широком диапазоне, от редких отказов MTBF =

Таблица 3

Оценка накладных расходов для разных методов восстановления

		Координированное	Асинхронное, $k = 2$	Асинхронное, $k = 5$
MTBF = [1800 36000]	ИМ[1]	$\bar{t} = 187$; $\sigma = 130$ $\bar{n}_1 = 2{,}1$; $\bar{n}_2 = 0{,}1$	$\bar{t} = 142$; $\sigma = 72$ $\bar{n}_1 = 2{,}1$; $\bar{n}_2 = 0{,}1$ $\Delta T = 45$ (24 %)	$\bar{t} = 113$; $\sigma = 30$ $\bar{n}_1 = 2{,}1$; $\bar{n}_2 = 0{,}1$ $\Delta T = 74$ (40 %)
	ПП[2]	$\bar{t} = 225$ $\bar{n}_1 = 2$; $\bar{n}_2 = 0{,}1$	$\bar{t} = 119$ $\bar{n}_1 = 2{,}3$; $\bar{n}_2 = 0$ $\Delta T = 106$ (47 %)	$\bar{t} = 106$ $\bar{n}_1 = 1{,}4$; $\bar{n}_2 = 0{,}1$ $\Delta T = 119$ (53 %)
MTBF = [720 3600]	ИМ	$\bar{t} = 432$; $\sigma = 154$ $\bar{n}_1 = 5{,}6$; $\bar{n}_2 = 1{,}1$	$\bar{t} = 331$; $\sigma = 86$ $\bar{n}_1 = 5{,}5$; $\bar{n}_2 = 1{,}1$ $\Delta T = 101$ (23 %)	$\bar{t} = 271$; $\sigma = 44$ $\bar{n}_1 = 5{,}4$; $\bar{n}_2 = 1{,}1$ $\Delta T = 161$ (37 %)
	ПП	$\bar{t} = 438$ $\bar{n}_1 = 6{,}3$; $\bar{n}_2 = 1{,}6$	$\bar{t} = 328$ $\bar{n}_1 = 4{,}9$; $\bar{n}_2 = 0{,}8$ $\Delta T = 110$ (25 %)	$\bar{t} = 302$ $\bar{n}_1 = 6{,}3$; $\bar{n}_2 = 0{,}9$ $\Delta T = 136$ (31 %)
MTBF = [360 1800]	ИМ	$\bar{t} = 638$; $\sigma = 164$ $\bar{n}_1 = 11{,}8$; $\bar{n}_2 = 2{,}4$	$\bar{t} = 488$; $\sigma = 97$ $\bar{n}_1 = 11{,}4$; $\bar{n}_2 = 2{,}3$ $\Delta T = 150$ (24 %)	$\bar{t} = 400$; $\sigma = 53$ $\bar{n}_1 = 11{,}1$; $\bar{n}_2 = 2{,}2$ $\Delta T = 238$ (37 %)
	ПП	$\bar{t} = 630$ $\bar{n}_1 = 12{,}3$; $\bar{n}_2 = 2{,}8$	$\bar{t} = 545$ $\bar{n}_1 = 12{,}7$; $\bar{n}_2 = 2{,}8$ $\Delta T = 85$ (13 %)	$\bar{t} = 487$ $\bar{n}_1 = 11{,}2$; $\bar{n}_2 = 2{,}4$ $\Delta T = 143$ (23 %)
MTBF = [240 1800]	ИМ	$\bar{t} = 812$; $\sigma = 175$ $\bar{n}_1 = 18{,}5$; $\bar{n}_2 = 3{,}7$	$\bar{t} = 626$; $\sigma = 107$ $\bar{n}_1 = 17{,}6$; $\bar{n}_2 = 3{,}5$ $\Delta T = 186$ (23 %)	$\bar{t} = 513$; $\sigma = 60$ $\bar{n}_1 = 17{,}1$; $\bar{n}_2 = 3{,}4$ $\Delta T = 299$ (37 %)
	ПП	$\bar{t} = 794$ $\bar{n}_1 = 18{,}5$; $\bar{n}_2 = 3{,}5$	$\bar{t} = 664$ $\bar{n}_1 = 16{,}4$; $\bar{n}_2 = 2{,}6$ $\Delta T = 130$ (16 %)	$\bar{t} = 603$ $\bar{n}_1 = 18{,}3$; $\bar{n}_2 = 3{,}4$ $\Delta T = 191$ (24 %)
MTBF = [180 900]	ИМ	$\bar{t} = 955$; $\sigma = 184$ $\bar{n}_1 = 25{,}3$; $\bar{n}_2 = 5{,}1$	$\bar{t} = 744$; $\sigma = 116$ $\bar{n}_1 = 24{,}2$; $\bar{n}_2 = 4{,}9$ $\Delta T = 211$ (22 %)	$\bar{t} = 604$; $\sigma = 65$ $\bar{n}_1 = 23{,}3$; $\bar{n}_2 = 4{,}7$ $\Delta T = 351$ (37 %)
	ПП	$\bar{t} = 938$ $\bar{n}_1 = 26$; $\bar{n}_2 = 4{,}6$	$\bar{t} = 762$ $\bar{n}_1 = 23{,}7$; $\bar{n}_2 = 3{,}8$ $\Delta T = 176$ (19 %)	$\bar{t} = 715$ $\bar{n}_1 = 22{,}7$; $\bar{n}_2 = 4{,}3$ $\Delta T = 223$ (24 %)

$[T_з/2 \;\; 10{\cdot}T_з]$, до частых отказов MTBF = $[T_з/20 \;\; T_з/4]$. В вычислительных экспериментах число отказов в среднем доходило до 30 и более отказов за время работы программы. С увеличением частоты отказов явно растет объем накладных расходов и в случае самых частых отказов средние значения накладных расходов для стратегии координированного восстановления в среднем составляли около 26 % от времени заранее запланированного объема расчетов. Однако явно не прослеживается зависимость объема сокращений накладных расходов от частоты отказов при переходе от координированного к асинхронному восстановлению.

[1] ИМ — результаты полученные имитационным методом.

[2] ПП — результаты полученные параллельной программой.

Заключение

В данной работе рассматриваются два метода обеспечения отказоустойчивости расчетов при отказах в вычислительной системе. Первый метод (стандартный) заключается в координированном сохранении контрольных точек, а в случае отказа осуществляется координированный возврат всех процессов к последней контрольной точке. Второй метод основан на координированном сохранении и журналировании передаваемых данных, а в случае отказа происходит асинхронное восстановление. При асинхронном восстановлении несколько запасных процессов проводят пересчет данных, потерянных после отказа, а остальные процессы находятся в ожидании окончания процедуры восстановления потерянных данных.

В работе предложен имитационный метод, моделирующий исполнение параллельной программы подверженной отказам. Данный метод описывает исполнение параллельной программы, учитывая возможность наступления случайных отказов во время выполнения основного алгоритма, сохранения контрольных точек и восстановления расчетов, а также позволяет сравнивать различные методы обеспечения отказоустойчивости.

Для экспериментальной оценки накладных расходов были реализованы параллельные программы, вычисляющие распределение тепла в тонкой пластине с координированным и асинхронным восстановлением. Для теоретической оценки накладных расходов этих методов был использован имитационный метод. Результаты сравнения оценок накладных расходов позволяют говорить о целесообразности применения асинхронной стратегии восстановления для вычислительных алгоритмов с незначительным объемом данных для журналирования, однако возможность ее применения для других алгоритмов требует дальнейших исследований.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований в рамках научного проекта № 17-07-01604 а.

Литература

1. Schroeder B., Gibson G.A. Understanding Failures in Petascale Computers // Journal of Physics: Conference Series. 2007. Vol. 78, No. 1 P. 12–22. DOI: 10.1088/1742-6596/78/1/012022
2. Hsu C.-H., Feng W.-C. A Power-aware Run-time System for High-performance Computing // Proceedings of the 2005 ACM/IEEE Conference on Supercomputing (Seattle, WA, USA, November 12 – 18, 2005). IEEE, 2005. P. 1–9. DOI: 10.1109/sc.2005.3
3. Martino C.D., Kalbarczyk Z., Iyer R.K., Baccanico F., Fullop J., Kramer W. Lessons Learned from the Analysis of System Failures at Petascale: The Case of Blue Waters // 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (Atlanta, Georgia, USA, June 23 – 26, 2014). IEEE, 2014. P. 610–621. DOI: 10.1109/dsn.2014.62
4. Dongarra J., Herault T., Robert Y. Fault-tolerance Techniques for High-performance Computing. Springer, Cham, 2015. 320 p. DOI: 10.1007/978-3-319-20943-2
5. Berkeley Lab Checkpoint/Restart (BLCR) for LINUX URL: http://crd.lbl.gov/departments/computer-science/CLaSS/research/BLCR/ (дата обращения: 03.11.2018)

6. Cappello F., Geist A., Gropp W., Kale S., Kramer B., Snir M., Toward Exascale Resilience: 2014 Update // Supercomputing Frontiers and Innovations. 2014. Vol. 1, No. 1. P. 5–28. DOI: 10.14529/jsfi140101
7. Elnozahy E.N. M., Alvisi L., Wang Y.-M., Johnson D. B. A Survey of Rollback-recovery Protocols in Message-passing Systems // ACM Comput. Surv. 2002. Vol. 34, No. 3. P. 375–408. DOI: 10.1145/568522.568525
8. Bouteiller A., Herault T., Bosilca G., Du P., Dongarra J. Algorithm-based Fault Tolerance for Dense Matrix Factorizations, Multiple Failures and Accuracy // ACM Transactions on Parallel Computing. 2015. Vol. 1, No. 2. P. 1–28. DOI: 10.1145/2686892
9. Engelmann C., Vallee G.R., Naughton T., Scott S.L. Proactive Fault Tolerance Using Preemptive Migration // 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing (Weimar, Germany, February 18 – 20, 2009). IEEE, 2009. P. 252–257. DOI: 10.1109/PDP.2009.31.
10. Бондаренко А.А., Якобовский М.В. Обеспечение отказоустойчивости высокопроизводительных вычислений с помощью локальных контрольных точек // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2014. Т. 3, № 3. С. 20–36 DOI: 10.14529/cmse140302
11. Di S., Bouguerra M.S., Bautista-Gomez L., Cappello F. Optimization of Multi-level Checkpoint Model for Large Scale HPC Applications // 28th International Parallel and Distributed Processing Symposium (Phoenix, Arizona, USA, May 19 – 23, 2014). IEEE, 2014. P. 1181–1190. DOI: 10.1109/IPDPS.2014.122.
12. Benoit A., Cavelan A., Le Fèvre V., Robert Y., Sun H. Towards Optimal Multi-level Checkpointing // IEEE Transactions on Computers. 2016. Vol. 66, No. 7. P. 1212–1226. DOI: 10.1109/TC.2016.2643660.
13. Di S., Robert Y., Vivien F., Cappello F. Toward an Optimal Online Checkpoint Solution under a Two-level HPC Checkpoint Model // IEEE Transactions on Parallel and Distributed Systems. 2016. Vol. 28, No. 1. P. 244–259. DOI: 10.1109/TPDS.2016.2546248.
14. Fault Tolerance Research Hub URL: http://fault-tolerance.org/ (дата обращения: 03.11.2018)
15. Бондаренко А.А., Ляхов П.А., Якобовский М.В. Накладные расходы, связанные с обеспечением отказоустойчивых вычислений при многоуровневом координированном сохранении контрольных точек // Параллельные вычислительные технологии (ПаВТ'2017): Труды международной научной конференции (Казань, 3 – 7 апреля 2017 г.). Челябинск: Издательский центр ЮУрГУ, 2017. С. 262–270.

Бондаренко Алексей Алексеевич, к.ф.-м.н., старший научный сотрудник, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

Ляхов Павел Александрович, аспирант, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

Якобовский Михаил Владимирович, член-корреспондент РАН, д.ф.-м.н., профессор, заместитель директора по научной работе, Институт прикладной математики им. М.В. Келдыша РАН (Москва, Российская Федерация)

DOI: 10.14529/cmse190205

COORDINATED CHECKPOINTING WITH SENDER-BASED LOGGING AND ASYNCHRONOUS RECOVERY FROM FAILURE

© 2019 A.A. Bondarenko, P.A. Lyakhov, M.V. Yakobovskiy
Keldysh Institute of Applied Mathematics Russian Academy of Sciences
(sq. Miusskaya 4, Moscow, 125047 Russia)
E-mail: bondaleksey@gmail.com, pavel.lyakhov@phystech.edu, lira@imamod.ru
Received: 20.11.2018

The increasing growth in the number of components of supercomputers leads HPC specialists to unfavorable estimates for future supercomputers: "the range of the mean time between failures will be from 1 hour to 9 hours." This estimate leads to the problem of long calculations on supercomputers. In this paper, we propose a recovery method from failure which does not require rollback for all processes. This method can reduce overhead costs for some computational algorithms. The standard fault tolerance method consists of two phases: coordinated checkpointing and rollback of all processes to the last checkpoint in the case of a failure. The proposed method includes coordinated checkpointing with sender-based logging and asynchronous recovery when most processes wait and several processes recalculate the lost data. We developed parallel programs to solve the problem of heat transfer in the thin plate. In these programs, failures occur by calling the function raise(SIGKILL), and coordinated or asynchronous recovery is performed by ULFM functions. In order to obtain theoretical estimates of overhead costs, we propose a simulation model of program execution with failures. This model assumes that failures strike during computations, checkpointing and recovery. We made a comparison of recovery methods with different failure rates for the problem with a small amount of data for logging. The comparison showed that the use of asynchronous recovery results in a reduction of overhead costs by theoretical estimates from 22 % to 40 %, and by computational experiments from 13 % to 53 %.

Keywords: MPI, ULFM extension, coordinated checkpointing, asynchronous recovery, fault tolerance.

FOR CITATION

Bondarenko A.A., Lyakhov P.A., Yakobovskiy M.V. Coordinated Checkpointing with Sender-based Logging and Asynchronous Recovery from Failure. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2019. vol. 8, no. 2. pp. 76–91. (in Russian) DOI: 10.14529/cmse190205.

References

1. Schroeder B., Gibson G.A. Understanding Failures in Petascale Computers. *Journal of Physics: Conference Series.* 2007. vol. 78, no. 1. pp. 12–22. DOI: 10.1088/1742-6596/78/1/012022
2. Hsu C.-H., Feng W.-C. A Power-aware Run-time System for High-performance Computing. *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing (Seattle, WA, USA, November 12 – 18, 2005).* IEEE, 2005. pp. 1–9. DOI: 10.1109/sc.2005.3
3. Martino C.D., Kalbarczyk Z., Iyer R.K., Baccanico F., Fullop J., Kramer W. Lessons Learned from the Analysis of System Failures at Petascale: The Case of Blue Waters. *44th*

Annual IEEE/IFIP International Conference on Dependable Systems and Networks (Atlanta, Georgia, USA, June 23 – 26, 2014). IEEE, 2014. pp. 610–621. DOI: 10.1109/dsn.2014.62

4. Dongarra J., Herault T., Robert Y. *Fault-tolerance Techniques for High-performance Computing.* Springer, Cham, 2015. 320 p. DOI: 10.1007/978-3-319-20943-2
5. *Berkeley Lab Checkpoint/Restart (BLCR) for LINUX.* Available at: http://crd.lbl.gov/departments/computer-science/CLaSS/research/BLCR/ (accessed: 03.11.2018)
6. Cappello F., Geist A., Gropp W., Kale S., Kramer B., Snir M., Toward Exascale Resilience: 2014 Update. *Supercomputing Frontiers and Innovations.* 2014. vol. 1, no. 1. pp. 5–28. DOI: 10.14529/jsfi140101
7. Elnozahy E.N. M., Alvisi L., Wang Y.-M., Johnson D. B. A Survey of Rollback-recovery Protocols in Message-passing Systems. *ACM Computing Surveys.* 2002. vol. 34, no. 3. pp. 375–408. DOI: 10.1145/568522.568525
8. Bouteiller A., Herault T., Bosilca G., Du P., Dongarra J. Algorithm-based Fault Tolerance for Dense Matrix Factorizations, Multiple Failures and Accuracy. *ACM Transactions on Parallel Computing.* 2015. vol. 1, no. 2. pp. 1–28. DOI: 10.1145/2686892
9. Engelmann C., Vallee G.R., Naughton T., Scott S.L. Proactive Fault Tolerance Using Preemptive Migration. *17th Euromicro International Conference on Parallel, Distributed and Network-based Processing (Weimar, Germany, February 18 – 20, 2009).* IEEE, 2009. pp. 252–257. DOI: 10.1109/PDP.2009.31.
10. Bondarenko A.A., Yakobovskiy M.V. Fault Tolerance for HPC by Using Local Checkpoints. *Vestnik Yuzho-Uralskogo gosudarstvennogo universiteta. Seriya Vychislitel'naya matematika i informatika* [Bulletin of South Ural State University. Series: Computational Mathematics and Software Engineering]. 2014. vol. 3, no. 3. pp. 20–36. DOI: 10.14529/cmse140302 (in Russian)
11. Di S., Bouguerra M.S., Bautista-Gomez L., Cappello F. Optimization of Multi-level Checkpoint Model for Large Scale HPC Applications. *28th International Parallel and Distributed Processing Symposium (Phoenix, Arizona, USA, May 19 – 23, 2014).* IEEE, 2014. pp. 1181–1190. DOI: 10.1109/IPDPS.2014.122.
12. Benoit A., Cavelan A., Le Fèvre V., Robert Y., Sun H. Towards Optimal Multi-level Checkpointing. *IEEE Transactions on Computers.* 2016. vol. 66, no. 7. pp. 1212–1226. DOI: 10.1109/TC.2016.2643660.
13. Di S., Robert Y., Vivien F., Cappello F. Toward an Optimal Online Checkpoint Solution under a Two-level HPC Checkpoint Model. *IEEE Transactions on Parallel and Distributed Systems.* 2016. vol. 28, no. 1. pp. 244–259. DOI: 10.1109/TPDS.2016.2546248.
14. *Fault Tolerance Research Hub.* Available at: http://fault-tolerance.org/ (accessed: 03.11.2018)
15. Bondarenko A.A., Lyakhov P.A., Yakobovskiy M.V. The Overheads Associated with Multi-level Coordinated Checkpointing. *Parallelnye vychislitelnye tekhnologii (PaVT'2017): Trudy mezhdunarodnoj nauchnoj konferentsii (Kazan', 3 – 7 aprelya 2017)* [Parallel Computational Technologies (PCT'2017): Proceedings of the International Scientific Conference (Kazan, Russia, 3 – 7 April, 2017)]. Chelyabinsk, Publishing of the South Ural State University, 2017. pp. 262–270. (in Russian)

УДК 004.652.43, 004.421.4 DOI: 10.14529/cmse190206

ОБНОВЛЕНИЕ МНОГОТАБЛИЧНЫХ ПРЕДСТАВЛЕНИЙ НА ОСНОВЕ КОММУТАТИВНЫХ ПРЕОБРАЗОВАНИЙ БАЗЫ ДАННЫХ

© 2019 В.С. Зыкин[1], М.Л. Цымблер[2]

[1] *Омский государственный технический университет*
(644050 Омск, пр. Мира, д. 11),
[2] *Южно-Уральский государственный университет*
(454080 Челябинск, пр. им. В.И. Ленина, д. 76)
E-mail: vszykin@mail.ru, mzym@susu.ru

Поступила в редакцию: 26.09.2018

В современных технологиях реляционных баз данных механизм представлений (view) реализует внешний уровень архитектуры ANSI–SPARC, скрывая детали концептуальной структуры базы данных от конечных пользователей. Однако использование данного механизма сопряжено с необходимостью решения задачи корректного обновления представлений: СУБД должна обеспечить корректное выполнение операций вставки, удаления или обновления кортежа в представлении над соответствующими базовыми отношениями данного представления. Для решения указанной задачи в стандарте SQL вводится жесткое ограничение: модифицируемому кортежу представления может соответствовать только один кортеж в базовом отношении. Триггеры, реализующие обновление представлений, обладают рядом недостатков: необходимость создания триггера для каждого представления базы данных, непредсказуемый порядок запуска триггеров, относящихся к одному представлению и др. В статье рассматривается подход к решению данной задачи на основе применения коммутативных преобразований базы данных. При этом не накладывается ограничение единственности кортежа базового отношения, соответствующего обновляемому кортежу в представлении. Описан Сопроцессор СУБД, который размещается на клиентском компьютере и обеспечивает коммутативные преобразования в отношениях базы данных, хранимых на сервере. Сопроцессор выполняет формирование текста транзакции, реализующей коммутативные преобразования, и осуществляет запуск этой транзакции на сервере. Представлена реализация сопроцессора для свободной СУБД PostgreSQL. Проведены вычислительные эксперименты, подтверждающие эффективность предложенного подхода в приложениях классов OLAP и OLTP.

Ключевые слова: коммутативное преобразование, реляционная алгебра, многотабличное представление, обновление представлений, реляционная СУБД, триггер.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Зыкин В.С., Цымблер М.Л. Обновление многотабличных представлений на основе коммутативных преобразований базы данных // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 2. С. 92–106. DOI: 10.14529/cmse190206.

Введение

Современные системы баз данных строятся в соответствии с трехуровневой архитектурой ANSI—SPARC [11]. Внутренний (физический) уровень отвечает за физический способ организации данных. Промежуточный (концептуальный) уровень инкапсулирует реляционную схему базы данных. Внешний (пользовательский) уровень показывает, как выглядит база данных с точки зрения конечного пользователя и реализуется с помощью представлений. *Представление (view)* — это виртуальное (логическое) отношение базы данных, которое является синонимом запроса к хранимым отношениям базы данных. Механизм представлений позволяет скрывать детали концептуальной структуры базы данных от конечных пользователей. Однако

использование данного механизма сопряжено с необходимостью решения задачи корректного *обновления представлений*, которая заключается в следующем. Поскольку представление воспринимается конечным пользователем как хранимое отношение базы данных, то возможны операции вставки, удаления или обновления кортежа в представлении. СУБД должна обеспечить корректное выполнение указанных операций над соответствующими базовыми отношениями данного представления.

В стандартах языка баз данных SQL, начиная с первой версии [12], решение данной задачи основано на введении ограничений на структуру представления. В соответствии с этими ограничениями для каждого кортежа представления необходимо наличие соответствующего кортежа в базовом отношении, а для каждого обновляемого атрибута представления необходимо наличие соответствующего ему атрибута в базовом отношении. В стандарте SQL:1999 вводится концепция триггера, существенно расширяющая функциональные возможности СУБД [20]. *Триггер* представляет собой подпрограмму на процедурном расширении SQL, постоянно хранимую в виде исходного текста в базе данных, и ассоциированную со специфицированным событием в базе данных. При наступлении указанного события СУБД автоматически исполняет тело триггера. Триггеры могут быть использованы для обновления представлений: с событием обновления представления необходимо ассоциировать тело триггера, выполняющее надлежащее обновление базового отношения.

Однако по следующим причинам триггеры не являются идеальным решением рассматриваемой задачи. Триггеры, реализующие обновление представлений, требуется разрабатывать для каждой отдельной базы данных. Триггеры являются источником накладных расходов СУБД, поскольку требуют блокировки различных ресурсов, проверки наступления события триггера, поддержки временных таблиц (например, таблицы INSERTED и DELETED в СУБД MS SQL Server используются для проверки результатов изменений данных и установки условий срабатывания триггеров [8]) и др. Кроме того, при наличии нескольких триггеров, относящихся к одному отношению (представлению) и ассоциированных с одним событием, последовательность их запуска не детерминирована.

Научные исследования, посвященные задаче корректного обновления представлений, начаты с момента становления теории реляционных баз данных и продолжаются в настоящее время. Однако, как показывает обзор научных публикаций по данной тематике, универсальное решение пока не найдено.

В данной работе предлагается подход к обновлению представлений, основанный на аппарате коммутативных преобразований данных. Коммутативность операций преобразования представления и преобразования отношений понимается как соответствие начального и конечного состояний базы данных между различными преобразованиями. Обновление представления осуществляется с помощью сопроцессора коммутативных преобразований, который создает транзакцию, выполняющую коммутативные преобразования отношений базы данных.

Статья организована следующим образом. В разделе 1 приводится обзор работ по теме исследования. Раздел 2 кратко описывает аппарат коммутативных преобразований. Приводятся формулы реляционной алгебры для операций удаления, добавления и обновления кортежа в представлении. В разделе 3 описана архитектура и принципы реализации сопроцессора коммутативных преобразований. В разделе 4 представлены

результаты вычислительных экспериментов, исследующих эффективность предложенного подхода. В заключении резюмируются итоги выполненного исследования.

1. Обзор работ

Научные исследования, посвященные задаче корректного обновления представлений, начаты с момента становления теории реляционных баз данных и продолжаются в настоящее время.

Работу 1981 г. [4], в которой предложено понятие коммутативных преобразований и сформулированы условия коммутативности обновления представлений, можно назвать пионерской в рассматриваемой области исследований. Авторы, однако, не предлагают алгоритмов обновления представлений.

В статье 1982 г. Даял исследует задачу соответствия модификаций пользовательских представлений данных с соответствующими преобразованиями в исходной базе данных [6]. В первую очередь в данной работе была рассмотрена корректность модификаций, их свойства и условия существования. Однако, практической реализации обновления представлений не приводится.

В 1984 г. Мазунага в работе [18] предложил вероятностный подход к формированию правил преобразования данных. Вероятностный характер результата преобразований является следствием использования семантического подхода. При возникновении нештатных ситуаций пользователю предоставляется возможность вручную откорректировать операции модификации. Очевидно, что в общем случае такой подход не гарантирует корректность результата обновления.

В 1985 г. Келлер в работе [14] предложил подход к решению задачи корректного обновления многотабличных представлений, который требует наличия в представлении атрибутов первичных ключей. Между тем в современной практике проектирования реляционных баз данных используются суррогатные первичные ключи отношений (вместо одного из атрибутов, выбираемого из множества потенциальных ключей), реализуемые в СУБД с помощью автоинкрементного типа данных. Суррогатные ключи, таким образом, не имеют определенной семантики и лишены возможности изменения.

В 1987 г. была разработана первая версия стандарта ISO/IEC 9075 [12], описывающего язык баз данных SQL. В данном стандарте присутствуют ограничения на обновления представлений, согласно которым один кортеж в представлении должн соответствовать одному кортежу в базовом отношении.

В 1988 г. Готтлоб и др. в работе [9] предложили понятие согласованного представления (consistent view), которое обладает следующим свойством: если возможны операции обновления представления, тогда имеется однозначный набор операций обновления базы данных, дающий тот же результат. Рассмотрена задача трансляции набора операций обновления представлений в набор операций обновления базы данных. Однако, в данной работе для кортежа согласованного представления допускается только один соответствующий кортеж в базовом отношении.

В 1990 г. Лангерак [15] предложил понятие репрезентативного экземпляра представления, который формируется из множества атрибутов всех отношений базы данных. Представления, таким образом, являются проекцией репрезентативного экземпляра. В данной работе, однако, рассмотрен только частный случай схемы баз данных,

когда каждое отношение является подмножеством проекций репрезентативного экземпляра по атрибутам каждого отношения.

В 2003 г. Лечтенбёргер в статье [16] представил подход «постоянного дополнения» (constant complement approach), развивающий идеи работы [4]. Данный подход предоставляет пользователю возможность отменить произведенные обновления представлений, используя результаты последующих обновлений. Практическая реализация предложенного подхода, однако, не описана.

В современной работе [5] Бертосси и Салими рассматривают задачу обновления представлений применительно к подготовке данных для машинного обучения. Удаление кортежей из представления рассматривается как исключение зашумленных данных из обучающей выборки. Таким образом, авторы ограничиваются рассмотрением только одной операции удаления кортежа из представления.

Рассмотренные работы подразумевают наличие ограничения, которое требует соответствия одного кортежа представления одному кортежу в базовом отношении, и рассматривают частные случаи обработки многотабличных представлений. В следующих разделах данной статьи мы рассмотрим подход, позволяющий преодолеть указанное ограничение в общем случае на основе использования аппарата коммутативных преобразований.

2. Коммутативные преобразования реляционных отношений

В данном разделе кратко представлен аппарат коммутативных преобразований [3], который является основой предлагаемого подхода к решению проблемы корректного обновления многотабличных представлений.

2.1. Базовые обозначения и определения

Далее используются следующие стандартные обозначения реляционных операций [7]: $\pi_X(R)$ — *операция проекции* отношения R по множеству атрибутов X, $\sigma_F(R)$ — *операция выбора* над отношением R, F — логическое выражение на значениях атрибутов, $R_1 \bowtie R_2$ — операция *естественного соединения* отношений R_1 и R_2.

В соответствии со статьей [3] введем обозначение модели информационной системы: $\Omega := (M, D, O, P)$, где M — описание схемы данных, D — множество допустимых состояний базы данных, O — множество операций модификации представлений, P — множество ограничений (предикатов) на состояния представления данных.

Введем краткую запись для последовательности операций естественного соединения нескольких отношений $R_1, \ldots, R_m$:

$$R_1 \bowtie R_2 \bowtie \ldots \bowtie R_m := \bowtie_{i=1}^{m} R_i. \tag{1}$$

Определим *многотабличное представление данных* Q как результат запроса на выборку данных из отношений $R_1, \ldots, R_m$:

$$Q := \pi_{X_0}\Big(\sigma_F\big(\bowtie_{i=1}^{m} \pi_{X_i}(R_i)\big)\Big), \tag{2}$$

где X_0 — множества атрибутов, которые будут формировать заголовок результирующего отношения. X_i — подмножество атрибутов отношения R_i, которое будет использоваться для построения многотабличного представления. Множество атрибутов, задействованных в объекте $\mathcal{O}$, будем обозначать как $\langle\mathcal{O}\rangle$.

Далее определим, по каким атрибутам будет производиться операция проекции над отношениями. Пусть атрибут $A \in \langle R_i \rangle$, тогда $\forall i\ (1 \leqslant i \leqslant m) : A \in X_i$, если выполняется хотя бы одно из следующих условий:

1) $A \in X_0$;
2) $\exists R_\ell$: $A \in \langle R_\ell \rangle$, $\ell \neq i$;
3) $A \in \langle F \rangle$.

Для корректного выполнения операций коммутативных преобразований на схему базы данных M накладывается требование ее ацикличности. Совокупность отношений базы данных $R_1, R_2, \ldots, R_k$ является ацикличной, если отсутствуют упорядоченные транзитивные связи по внешним ключам между отношениями R_1 и R_2, между R_2 и R_3, $\ldots$, между R_{k-1} и R_k, а также связь между отношениями R_k и R_1 [2].

Пусть модель Ω будет *исходной*, а $\Omega' := (M', D', O', P')$ — *целевой*. Межмодельные преобразования подразумевают построение отображения $\Omega \Leftrightarrow \Omega'$. В данной работе показывается, каким образом отображение $\Omega \Rightarrow \Omega'$ может быть сведено к выполнению транзакции в базе данных.

Совокупность всех значений данных в базе данных, неизменных в течение некоторого промежутка времени, будем называть *состоянием базы данных*. Пусть при работе с моделью Ω' пользователь выполняет команду обновления представления f'_p, которая переводит модель Ω' из состояния $d'_i \in D'$ в состояние $d'_j \in D'$. Тогда необходимо выполнить преобразования, соответствующие переходу модели Ω из состояния $d_i \in D$ в состояние $d_j \in D$.

Преобразование представления данных, соответствующие отображению $\Omega \Leftrightarrow \Omega'$, будем считать *корректным*, если выполнены следующие *условия коммутативности*:

$$\begin{aligned} &d_i \xrightarrow{Q} d'_i \xrightarrow{f_p} d'_j, \\ &d_i \xrightarrow{Alg_p} d_j \xrightarrow{Q} d'_j, \end{aligned} \tag{3}$$

где Q — многотабличный запрос; Alg_p — алгоритм преобразования исходной базы данных, сопоставленный команде обновления f_p.

Таким образом, переход в состояние d'_j возможен двумя способами, но результат должен быть один и тот же. В работе [3] построены выражения реляционной алгебры для алгоритмов Alg_p, реализующих операции обновления многотабличных представлений.

Отношения $R_1, \ldots, R_m$ должны иметь упорядочение по внешним ключам: главные (ссылающиеся) отношения в этой последовательности стоят раньше, подчиненные (ссылаемые) — позже. Таким образом, существует частичный порядок, при котором имеется только одно отношение R_m, не имеющее подчиненных отношений. Далее это отношение будем называть *целевым*. Операции обновления данных должны быть реализованы только в целевом отношении R_m. Остальные отношения, используемые для формирования представления Q, назовем *отношениями-справочниками*.

2.2. Операции модификации многотабличного представления

В данном разделе приведена реализация реляционных операций обновления многотабличного представления на основе коммутативных преобразований данных. Операция обновления подразумевает одно из следующих действий, связанных с обновлением целевого отношения: удаление кортежа из Q, вставка кортежа в Q и

обновление кортежа в Q. Коммутативность операций удаления и вставки кортежа доказана в работе [3], суперпозиция указанных операций реализует операцию обновления кортежа.

Удаление кортежа

Допустим, что кортеж u удаляется из представления, соответствующего запросу Q (далее запрос и соответствующее ему отношение мы обозначаем как Q). Результат данной операции можно выразить функцией $Q' := DELETE(Q, u)$, где в качестве параметров фигурируют удаляемый кортеж u и многотабличное представление Q, а возвращаемым значением является представление Q':

$$DELETE(Q, u) := R_m \setminus \pi_{\langle R_m \rangle}(T_D), \tag{4}$$

где T_D — множество кортежей в целевом отношении R_m, совпадающих с кортежем u на атрибутах X_0. Множество T_D определяется следующим образом:

$$\begin{gathered} T_D := \pi_{X_0}\Big(\sigma_{c_{del}}\big(\bowtie_{i=1}^{m} \ \pi_{X_i}(R_i)\big)\Big), \\ c_{del} := F \wedge (X_0 = u), \end{gathered} \tag{5}$$

где запись $X_0 = u$ означает равенство значений атрибутов кортежа, удаляемого из представления, и соответствующих атрибутов в отношениях базы данных. Во время выполнения операции удаления кортежей в соответствии с семантикой предметной области формируется T_D — множество кортежей целевого отношения, соответствующих удаляемому кортежу в представлении.

Вставка кортежа

Пусть выполняется вставка кортежа u в представление Q. Представим результат данной операции в виде функции $Q' := INSERT(Q, u)$, где Q — многотабличное представление, u — новый кортеж, Q' — возвращаемое многотабличное представление со вставленным кортежем. При выполнении этой функции в целевое отношение R_m необходимо вставить множество кортежей T_I.

$$INSERT(Q, u) := R_m \cup T_I, \tag{6}$$

где множество кортежей T_I для вставки в целевое отношение определяется следующим образом:

$$\begin{aligned} T_I &:= \pi_{X_m}\big(\sigma_F(T_I' \bowtie u)\big), \\ T_I' &:= \pi_Y\Big(\sigma_{c_{ins}}\big(\bowtie_{i=1}^{m-1} \ \pi_{X_i}(R_i)\big)\Big), \\ Z &:= X_0 \cap \cup_{i=1}^{m-1} X_i, \\ Y &:= \Big(\langle R_m \rangle \cup \langle F \rangle \cup X_0\Big) \cap \cup_{i=1}^{m-1} X_i, \\ c_{ins} &:= F' \wedge (Z = u[Z]). \end{aligned} \tag{7}$$

Здесь F' представляет собой набор условий, основанный на множестве F, и не включающий в себя условия, накладываемые на целевое отношение: $\langle F' \rangle := \langle F \rangle \cap \cup_{i=1}^{m-1} X_i$ [19].

В процессе формирования множества кортежей T_I, формируется представление T'_I. Заголовок представления T'_I состоит из атрибутов всех отношений, за исключением целевого. Представление T'_I можно интерпретировать как множество всех кортежей базы данных, которые могут быть связаны с кортежем, добавляемым в представление Q («множество универсум»). Далее для формирования представления T_I выполняются следующие действия. Множество T'_I с помощью операции естественного соединения связывается с кортежем u, затем применяются следующие операции: выбор по условиям F и проекция по атрибутам целевого отношения.

Обновление кортежа

Пусть в представлении Q кортеж u заменяется на кортеж u', тогда обновление представления выражается функцией Q: $Q' := UPDATE(Q, u, u')$. Данную функцию можно представить в виде суперпозиции функций для операций удаления текущего кортежа и добавления нового кортежа:

$$UPDATE\big(Q, u, u'\big) := INSERT\Big(DELETE\big(Q, u\big), u'\Big). \tag{8}$$

3. Сопроцессор коммутативных преобразований

В данном разделе описан Сопроцессор СУБД, который обеспечивает коммутативные преобразования в базе данных при обновлении многотабличного представления.

3.1. Архитектура и методы реализации

Сопроцессор коммутативных преобразований (СКоП) — это программная система, которая обеспечивает корректное выполнение операций удаления, добавления и обновления кортежей в многотабличных представлениях базы данных на основе использования операций, выраженных формулами (4)–(8). Архитектура СКоП (см. рис. 1) предполагает следующие подсистемы: Коммутатор, Парсер и Локальный словарь базы данных.

Парсер — это подсистема, которая обеспечивает получение метаданных об отношениях, прямо и косвенно вовлеченных в запрос на обновление представления, из словаря базы данных, и их сохранение в Локальном словаре.

Локальный словарь обеспечивает хранение следующих основных метаданных: имена отношений и атрибутов каждого отношения, логические выражения, накладываемые на значения атрибутов и др. При инициализации клиентского приложения СКоП загружает данные Локального словаря в оперативную память. После обновления многотабличного представления в базе данных обновляется соответствующая информация в локальном словаре.

Коммутатор — подсистема, которая получает запрос пользователя на обновление представления, формирует текст транзакции, выполняющей обновление набора кортежей в целевом отношении базы данных, и затем запускает сформированный текст транзакции на сервере. Коммутатор возвращает клиентскому приложению отклик сервера (результат выполнения операции обновления или код ошибки). Для формирования текста транзакции Коммутатор использует данные из Локального словаря. Транзакция реализует операции реляционной алгебры, выраженные формулами (4)–(8).

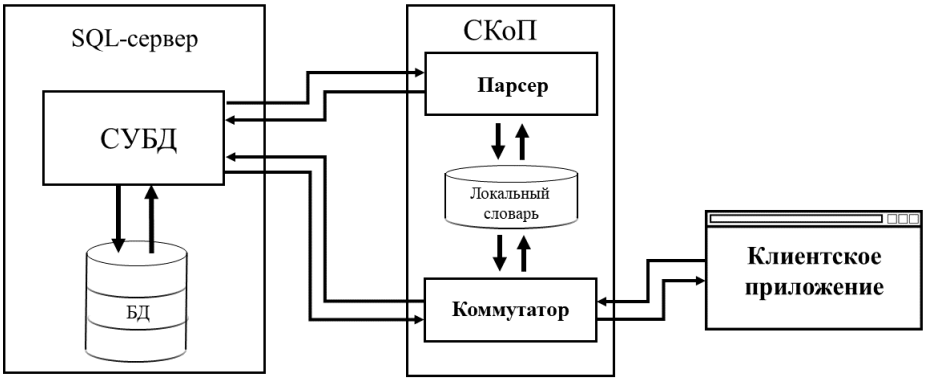


Рис. 1. Система баз данных с Сопроцессором коммутативных преобразований

3.2. Сопроцессор баз данных для PostgreSQL

Предложенная архитектура СКоП была реализована для свободной СУБД PostgreSQL [1], соответствующий Сопроцессор получил название pgCOPCT (PostgreSQL COProcessor Commutative Transformations). Сопроцессор разработан на программной платформе .NET Framework на языке программирования C# и представляет собой приложение ОС Windows. Реализация СКоП для другой реляционной СУБД потребует хотя и значительных, но относительно механических модификаций исходных текстов в соответствии с интерфейсами новой СУБД.

Работа сопроцессора pgCOPCT может быть кратко описана следующим образом (см. рис. 2). База данных под управлением СУБД PostgreSQL располагается на сервере. На клиентском компьютере устанавливается сопроцессор pgCOPCT и драйвер Npgsql для подключения к СУБД. В течение сеанса работы на стороне клиента в Локальном словаре сохраняются только данные, необходимые для обновления представлений. pgCOPCT состоит из следующих компонентов. Класс FormBuilder представляет собой конструктор запросов и обеспечивает формирование многотабличного представления. Класс DBConnector обеспечивает заполнение Локального словаря базы данных. Класс FormResult предоставляет оконный интерфейс редактирования многотабличных представлений. Класс DBEditor выполняет запуск сформированной транзакции на сервере.

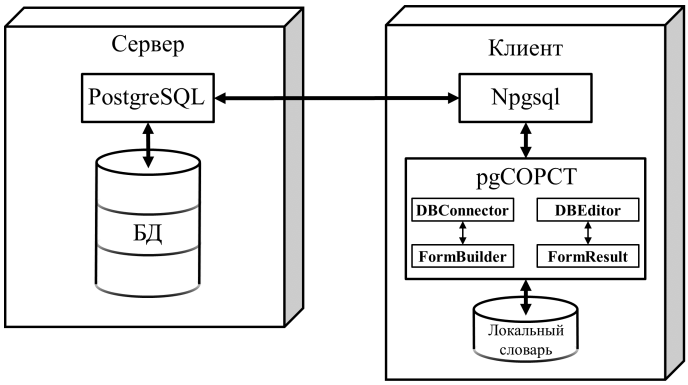


Рис. 2. Диаграмма развертывания сопроцессора pgCOPCT

4. Экспериментальное исследование

4.1. Цели и вычислительная среда экспериментов

В данном разделе описаны вычислительные эксперименты, исследующие эффективность предложенного подхода. *Эффективность* понимается как время выполнения операций обновления многотабличного представления с помощью сопроцессора коммутативных преобразований в сравнении с современными СУБД PostgreSQL, Oracle и MS SQL Server, которые для тех же целей используют триггеры.

Применение СКоП исследовалось для двух классов приложений: OLAP и OLTP. Приложения *OLAP (Online Analitical Processing, оперативный анализ данных)* связаны с выполнением сложных запросов на выборку данных из нескольких таблиц хранилища данных и использованием агрегационных функций. В связи с этим для приложений OLAP применение СКоП ограничивается случаем обновления хранилища данных, которому соответствует выполнение операций вставки новых кортежей в многотабличное представление. *Приложения OLTP (Online Transaction Processing, оперативная обработка транзакций)* подразумевают обработку коротких транзакций, которые выполняют вставку, удаление и обновление кортежей в базе данных в реальном времени. В соответствии с этим для приложений OLTP исследовалась эффективность СКоП при выполнении операций вставки, удаления и обновления кортежей в многотабличном представлении.

Эксперименты проводились с использованием синтетических баз данных, созданных в соответствии со спецификациями стандартных тестов консорциума TPC (Transaction Processing Council): TPC-H [10] для приложений класса OLAP и TPC-E [17] для приложений класса OLTP. При этом распределение кортежей в базе данных осуществлялось по правилу Зипфа «80-20» [7]: 80 % кортежей в целевом отношении соответствует 20 % кортежей в отношении-справочнике.

Аппаратная платформа экспериментов резюмирована ниже в таблице.

Таблица

Аппаратная платформа экспериментов

Характеристика	**Значение**
Процессор	Intel Core 2 Duo P7450 (2 ядра по 2,13 ГГц)
Оперативная память	4 Гб (DDR3-533)
Дисковая память	256 Гб (твердотельный накопитель OCZ)

4.2. Эффективность СКоП в приложениях класса OLAP

В стандартном тесте TPC-H [10] СУБД имитирует обработку заказов, используя базу данных, схема которой представлена на рис. 3. Базовыми отношениями многотабличного представления выбраны отношения CUSTOMER, ORDERS, LINEITEM и PARTSUPP. В качестве целевого отношения выбрана отношения LINEITEM. Многотабличное представление задействует все атрибуты целевого отношения, не являющиеся атрибутами внешнего ключа, а также атрибуты отношения CUSTOMER.Name, ORDERS.OrderPriority, PARTSUPP.Comment, к которым применено ограничение по атрибуту CUSTOMER.Address (отбор клиентов из специфицированного города). Варьируемым параметром экспериментов

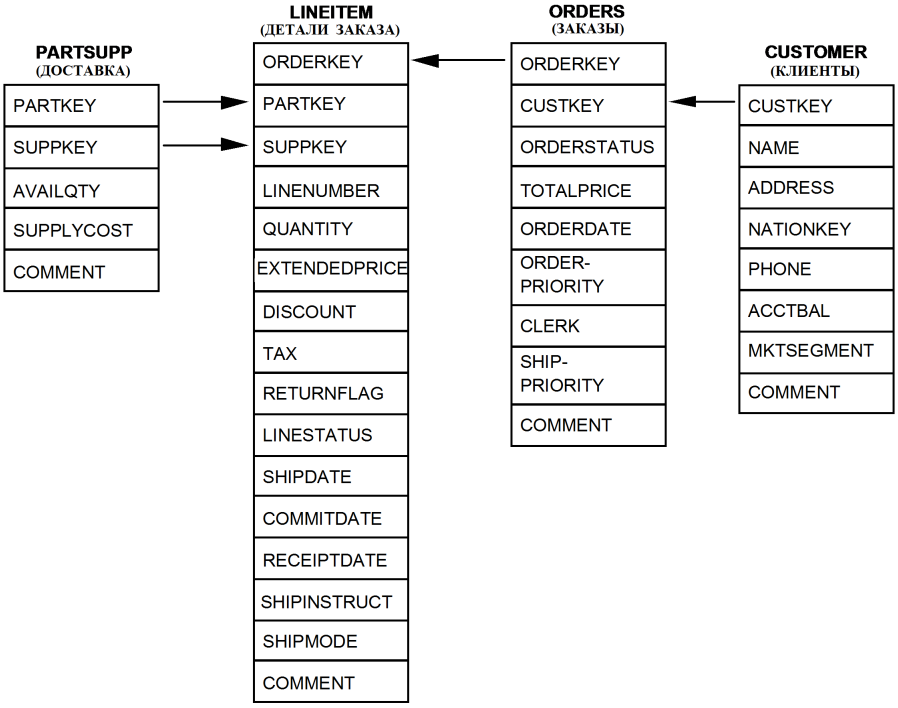


Рис. 3. Схема базы данных теста TPC-H

является количество кортежей в целевом отношении, соответствующих изменяемому кортежу в представлении.

Результаты экспериментов представлены на рис. 4. Можно видеть, что СУБД MS SQL Server, PostgreSQL и Oracle выполняют операции вставки данных в многотабличное представление с помощью СКоП быстрее, чем с помощью триггеров, на 10–35 %.

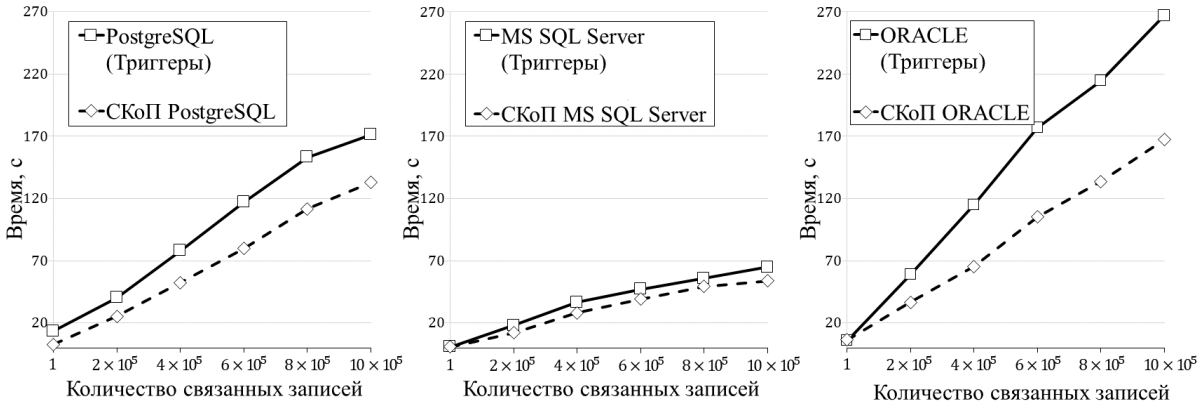


Рис. 4. Эффективность СКоП в приложениях класса OLAP

4.3. Эффективность СКоП в приложениях класса OLTP

В стандартном тесте TPC-E [17] СУБД имитирует торговлю на фондовой бирже и включает в себя следующие три сценария выполнения транзакций: обновление информации о сделке (Trade-Update), очистка информации об специфицированной сделке (Trade-Cleanup) и Market-Feed (протоколирование текущей рыночной активности), —

в каждом из которых выполняется модификация от четырех до шести отношений. Эксперименты проводились с использованием рассмотренного выше сопроцессора pgCOPCT.

Для каждого из указанных сценариев теста TPC-E было сформировано многотабличное представление, обновление которого выливается в выполнение транзакции, которая выполняет обновление отношений базы данных в соответствии со сценарием. Выполнение каждого сценария с использованием СКоП осуществлялось в режимах холодного и горячего запуска. Холодному запуску СКоП соответствует ситуация, когда Парсер сначала формирует Локальный словарь базы данных, а затем Коммутатор выполняет необходимые действия. Горячий запуск СКоП означает, что необходимость формирования Локального словаря базы данных отсутствует, и Коммутатор сразу выполняет необходимые действия.

Результаты экспериментов представлены на рис. 5. Можно видеть, что для каждого из рассмотренных сценариев в режиме горячего запуска pgCOPCT выполняет обновление многотабличного представления быстрее, чем СУБД PostgreSQL с помощью триггеров. Однако использование триггеров выгоднее в режиме холодного запуска. Таким образом, накладные расходы на поддержку Локального словаря базы данных являются необходимой платой за эффективность СКоП.

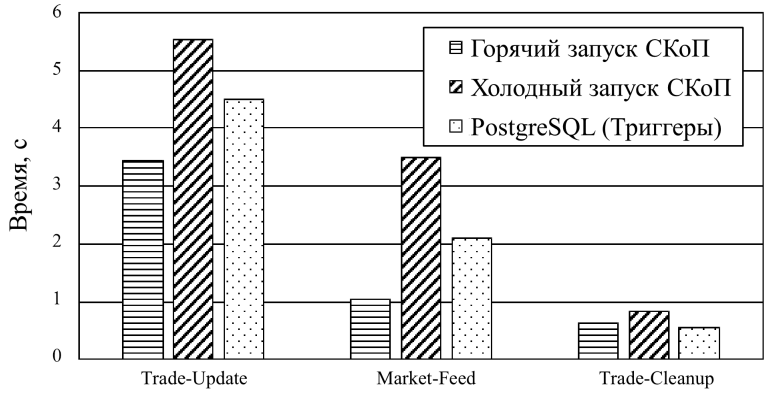


Рис. 5. Эффективность СКоП в приложениях класса OLTP

Заключение

В статье представлен подход к решению задачи корректного обновления многотабличных представлений (views) в реляционных базах данных на основе использования аппарата коммутативных преобразований.

Описан Сопроцессор СУБД, который размещается на клиентском компьютере и обеспечивает коммутативные преобразования в отношениях базы данных, хранимых на сервере. Сопроцессор состоит из трех подсистем: Парсер, Локальный словарь базы данных и Коммутатор. Парсер обеспечивает чтение метаданных словаря баз данных и их сохранение в Локальном словаре. Коммутатор, используя метаданные Локального словаря, формирует текст транзакции, реализующей коммутативные преобразования, и осуществляет запуск этой транзакции на сервере. Представлена реализация сопроцессора для свободной СУБД PostgreSQL.

Проведены вычислительные эксперименты, исследующие эффективность использования предложенного подхода в приложениях классов OLAP и OLTP с

использованием синтетических баз данных, специфицированных в стандартных тестах TPC-H и TPC-E соответственно. При обновлении многотабличных представлений Сопроцессор коммутативных преобразований показывает лучшее быстродействие, чем триггеры СУБД.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (государственное задание 2.7905.2017/8.9).

Литература

1. Зыкин В.С. Редактор многотабличного представления данных: свидетельство о государственной регистрации программ для ЭВМ – № 2018661249 от 04.09.2018; Правообладатель: Омский государственный технический университет.
2. Зыкин В.С. Ссылочная целостность данных в корпоративных информационных системах // Информатика и ее применения. 2015. Т. 9. № 3. С. 119–127.
3. Зыкин С.В., Зыкин В.С. Коммутативные преобразования в базе данных при редактировании многотабличных запросов // Информационные технологии. 2018. Т. 24, № 5. С. 330–338. DOI: 10.17587/it.24.330-338.
4. Bancilhon F., Spyratos N. Update Semantics of Relational Views // ACM Trans. Database Syst. 1981. Vol. 6, No. 4. P. 557–575. DOI: 10.1145/319628.319634.
5. Bertossi L., Salimi B. Causes for Query Answers from Databases: Datalog Abduction, View-updates, and Integrity Constraints // Int. J. Approx. Reason. 2017. Vol. 90. P. 226–252. DOI: 10.1016/j.ijar.2017.07.010.
6. Dayal U., Bernstein P.A. On the Correct Translation of Update Operations on Relational Views // ACM Trans. Database Syst. 1982. Vol. 7, No. 3. P. 381–416. DOI: 10.1145/319732.319740.
7. Garcia-Molina H., Ullman J.D., Widom J. Database System Implementation. Prentice Hall, 2000. 653 p.
8. Ghandeharizadeh S., Yap J. SQL Query to Trigger Translation: A Novel Transparent Consistency Technique for Cache Augmented SQL Systems // Proceedings of the 28th International Workshop on Database and Expert Systems Applications, DEXA 2017, August 28–31, 2017, Lyon, France. P. 37–41. DOI: 10.1109/DEXA.2017.24.
9. Gottlob G., Paolini P., Zicari R. Properties and Update Semantics of Consistent Views // ACM Trans. Database Syst. 1988. Vol. 13, No. 4. P. 486–524. DOI: 10.1145/49346.50068.
10. Hayamizu Y., Kawamichi R., Goda K., Kitsuregawa M. Benchmarking and Performance Analysis of Event Sequence Queries on Relational Database // Proceedings of the 10th TPC Technology Conference Performance Evaluation and Benchmarking for the Era of Artificial Intelligence, TPCTC, August 27–31, 2018, Rio de Janeiro, Brazil. P. 110–125. DOI: 10.1007/978-3-030-11404-6_9.
11. Interim Report: ANSI/X3/SPARC Study Group on Data Base Management Systems. FDT - Bulletin of ACM SIGMOD. 1975. Vol. 7, No. 2. P. 1–140.
12. ISO/IEC 9075:1987 Information Technology. Database Languages. SQL. Washington, 1987.
13. ISO/IEC 9075-11:2016 Information technology. Database languages. SQL. Part 11: Information and Definition Schemas (SQL/Schemata). Washington. 2016. 327 p.

14. Keller A. Algorithms for Translating View Updates to Database Updates for Views Involving Selections, Projections and Joins // Proceedings of the 4th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, PODS'85, March 25–27, 1985, Portland, USA. ACM, 1985. P. 154–163. DOI: 10.1145/325405.325423.

15. Langerak R. View Updates in Relational Databases with an Independent Scheme // ACM Trans. Database Syst. 1990. Vol. 15, No. 1. P. 40–66. DOI: 10.1145/77643.77645.

16. Lechtenbörger J. The Impact of the Constant Complement Approach Towards View Updating // Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'03, June 9–11, 2003, San Diego, CA, USA. ACM, 2003. P. 49–55. DOI: 10.1145/773153.773159.

17. Li Y., Levine C. Extending TPC-E to Measure Availability in Database Systems // Proceedings of the 10th Technology Conference Measurement and Characterization, August 29 – September 3, 2011, Seattle, WA, USA. P. 111–122. DOI: 10.1007/978-3-642-32627-1_8.

18. Masunaga Y. A Relational Database View Update Translation Mechanism // Proceedings of the 10th International Conference on Very Large Data Bases, VLDB'84, August 27–31, 1984, Singapore. P. 309–320.

19. Mosin S.V., Zykin S.V. Truth Space Method for Caching Database Queries // Моделирование и анализ информационных систем. 2015. Т. 22. № 2. С. 248–258.

20. Stonebraker M. Triggers and Inference In Database Systems. On Knowledge Base Management Systems (Islamorada). 1985. P. 297–314.

Зыкин Владимир Сергеевич, аспирант, старший преподаватель, кафедра прикладной математики и фундаментальной информатики, Омский государственный технический университет (Омск, Российская Федерация)

Цымблер Михаил Леонидович, к.ф.-м.н., доцент, кафедра системного программирования, Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск, Российская Федерация)

DOI: 10.14529/cmse190206

UPDATING OF MULTI-TABLE VIEWS BASED ON COMMUTATIVE DATABASE TRANSFORMATIONS

© 2019 V.S. Zykin[1], M.L. Zymbler[2]

[1] *Omsk State Technical University (pr. Mira 11, Omsk, 644050 Russia),*
[2] *South Ural University (pr. Lenina 76, Chelyabinsk, 454080 Russia)*
E-mail: vszykin@mail.ru, mzym@susu.ru

Received: 26.09.2018

In modern relational database technologies, views implement the external layer of the ANSI–SPARC architecture, which encapsulates details of the database conceptual structure from end-users. However, when using views, we need to solve the problem of correct view updating: DBMS must execute insertion, deletion, and updating tuples of the view while providing correct modifications of corresponding target relation(s) of this view. To solve this problem, the SQL standard introduces a strict restriction: only one tuple in the target relation can correspond to the modified tuple in the view. Also, triggers are not a satisfactory solution of this problem because

of necessity of such a trigger for each view of the database, and unpredictable sequence in execution of triggers that belong to the same view, etc. The paper presents an approach to solve the problem of correct view updating based on the commutative database transformations. This does not limit the tuple uniqueness in the target relation that corresponds to the updated tuple in the view. We describe the DBMS Coprocessor, which is deployed on the client computer and provides commutative transformations in the database relations stored on the server side. The coprocessor generates a transaction's script that implements commutative transformations and runs the transaction on the server. We present implementation of the Coprocessor for the PostgreSQL open-source DBMS. Experimental evaluation confirms the effectiveness of the proposed approach in OLAP and OLTP applications.

Keywords: commutative transformation, relational algebra, multi-table view, view updating, relational DBMS, trigger.

FOR CITATION

Zykin V.S., Zymbler M.L. Updating Multi-table Views Based on Commutative Database Transformations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2019. vol. 8, no. 2. pp. 92–106. (in Russian) DOI: 10.14529/cmse190206.

References

1. Zykin V.S. Multi-table Data View Editor: Certificate of State Registration of Computer Programs – No. 2018661249; registration date: 04.09.2018; Copyright holder: Omsk State Technical University.
2. Zykin V.S. Referential Integrity of Data in Corporate Information Systems. *Informatics and Applications.* 2015. vol. 9. no 3. pp. 119–127.
3. Zykin S.V., Zykin V.S. Commutative Conversion in the Database when Editing a Multitable Query. *Information Technologies.* 2018. vol. 24, no. 5. pp. 330–338. DOI: 10.17587/it.24.330-338.
4. Bancilhon F., Spyratos N. Update Semantics of Relational Views. *ACM Trans. Database Syst.* 1981. vol. 6, no. 4. pp. 557–575. DOI: 10.1145/319628.319634.
5. Bertossi L., Salimi B. Causes for Query Answers from Databases: Datalog Abduction, View-updates, and Integrity Constraints. *Int. J. Approx. Reason.* 2017. vol. 90. pp. 226–252. DOI: 10.1016/j.ijar.2017.07.010.
6. Dayal U., Bernstein P.A. On the Correct Translation of Update Operations on Relational Views. *ACM Trans. Database Syst.* 1982. vol. 7, no. 3. pp. 381–416. DOI: 10.1145/319732.319740.
7. Garcia-Molina H., Ullman J.D., Widom J. Database System Implementation. Prentice Hall, 2000. 653 p.
8. Ghandeharizadeh S., Yap J. SQL Query to Trigger Translation: A Novel Transparent Consistency Technique for Cache Augmented SQL Systems. Proceedings of the 28th International Workshop on Database and Expert Systems Applications, DEXA 2017, August 28–31, 2017, Lyon, France. pp. 37–41. DOI: 10.1109/DEXA.2017.24.
9. Gottlob G., Paolini P., Zicari R. Properties and Update Semantics of Consistent Views. *ACM Trans. Database Syst.* 1988. vol. 13, no. 4. p. 486–524. DOI: 10.1145/49346.50068.

10. Hayamizu Y., Kawamichi R., Goda K., Kitsuregawa M. Benchmarking and Performance Analysis of Event Sequence Queries on Relational Database. Proceedings of the 10th (TPC) Technology Conference Performance Evaluation and Benchmarking for the Era of Artificial Intelligence, TPCTC, August 27–31, 2018, Rio de Janeiro, Brazil. pp. 110–125. DOI: 10.1007/978-3-030-11404-6_9.

11. Interim Report: ANSI/X3/SPARC Study Group on Data Base Management Systems. FDT - Bulletin of ACM SIGMOD. 1975. vol. 7, no. 2. pp. 1–140.

12. ISO/IEC 9075:1987 Information technology. Database languages. SQL. Washington. 1987.

13. ISO/IEC 9075-11:2016 Information technology. Database languages. SQL. Part 11: Information and Definition Schemas (SQL/Schemata). Washington. 2016. 327 p.

14. Keller A. Algorithms for Translating View Updates to Database Updates for Views Involving Selections, Projections and Joins. Proceedings of the 4th ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, PODS'85, March 25–27, 1985, Portland, USA. ACM, 1985. pp. 154–163. DOI: 10.1145/325405.325423.

15. Langerak R. View Updates in Relational Databases with an Independent Scheme. *ACM Trans. Database Syst.* 1990. vol. 15, no. 1. pp. 40–66. DOI: 10.1145/77643.77645.

16. Lechtenbörger J. The Impact of the Constant Complement Approach Towards View Updating. Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'03, June 9–11, 2003, San Diego, CA, USA. ACM, 2003. pp. 49–55. DOI: 10.1145/773153.773159.

17. Li Y., Levine C. Extending TPC-E to Measure Availability in Database Systems. Proceedings of the 10th Technology Conference Measurement and Characterization, August 29 – September 3, 2011, Seattle, WA, USA. pp. 111–122. DOI: 10.1007/978-3-642-32627-1_8.

18. Masunaga Y. A Relational Database View Update Translation Mechanism. Proceedings of the 10th International Conference on Very Large Data Bases, VLDB'84, August 27–31, 1984, Singapore. pp. 309–320.

19. Mosin S.V., Zykin S.V. Truth Space Method for Caching Database Queries. *Modeling and Analysis of Information Systems*. 2015. vol. 22, no 2. pp. 248–258.

20. Stonebraker M. Triggers and Inference In Database Systems. On Knowledge Base Management Systems (Islamorada). 1985. pp. 297–314.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Научный журнал «Вестник ЮУрГУ. Серия «Вычислительная математика и информатика» основан в 2012 году.

Учредитель — Федеральное государственное автономное образовательное учреждение высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет).

Главный редактор — Л.Б. Соколинский.

Свидетельство о регистрации ПИ ФС77-57377 выдано 24 марта 2014 г. Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ; индексируется в библиографической базе данных РИНЦ. Журнал размещен в открытом доступе на Всероссийском математическом портале MathNet. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Решением Президиума Высшей аттестационной комиссии Министерства образования и науки Российской Федерации журнал включен в «Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук» по научным специальностям и соответствующим им отраслям науки: 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (физико-математические науки), 05.13.17 – Теоретические основы информатики (физико-математические науки).

Подписной индекс научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»: 10244, каталог «Пресса России». Периодичность выхода — 4 выпуска в год.

Адрес редакции, издателя: 454080, г. Челябинск, проспект Ленина, 76, Издательский центр ЮУрГУ, каб. 32.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Правила подготовки рукописей и пример оформления статей можно загрузить с сайта серии `http://vestnikvmi.susu.ru`. **Статьи, оформленные без соблюдения правил, к рассмотрению не принимаются.**
2. Адрес редакционной коллегии научного журнала «Вестник ЮУрГУ», серия «Вычислительная математика и информатика»:

   Россия 454080, г. Челябинск, пр. им. В.И. Ленина, 76, ЮУрГУ, кафедра СП,

   ответственному секретарю Цымблеру М.Л.
3. Адрес электронной почты редакции: `vestnikvmi@susu.ru`
4. **Плата с авторов за публикацию рукописей не взимается, и гонорары авторам не выплачиваются.**