

Increasing the efficiency of solving linear programming problems when solving systems of partial derivative differential equations using the control point method

*Aleksandr M. Kler and Nikolai O. Epishkin**

Melentiev Energy Systems Institute of Siberian Branch of Russian Academy of Sciences, 130 Lermontov str., Irkutsk, Russia

Abstract. The paper discusses the basic principles of the control point method, which reduces the solution of a linear system of partial differential equations to the solution of a linear programming (LP) problem. An analysis of various methods for solving LP problems is given and it is shown that the most suitable for the problems under consideration is the interior point method with unconditional sequential minimizations of the logarithmic penalty function. Moreover, with a large dimension of the LP problem, the main costs of computer time are associated with solving systems of linear algebraic equations (SLAE) that determine the direction of descent in Newton's method, used for unconditional minimization. A structural approach is proposed to the formation of optimized parameters of the LP problem and the organization of the solution of SLAE, in which the solution of SLAE of large dimension is reduced to the inversion of square matrices and the solution of SLAE of much smaller dimensions, which radically reduces the cost of computer time both for solving the SLAE and the LP problem.

1 Introduction

Calculations of stationary and non-stationary operating regimes of a number of elements of heat power plants (heat exchangers of various types, furnaces, pipelines, turbine stages, etc.) are reduced to solving systems of partial differential equations (SPDE). The main methods for solving such systems are the finite difference method (FDM) [1,2], the control volume method (CVM) [3,4] and the final element method (FEM) [5,6]. These methods are based on the formation and solution systems of linear algebraic equations (SLAE) of large dimension. When using them, difficulties arise with the solution if the initial state of the system under study is not known. Poor convergence observed for stiff SPDE. In these methods, the condition for a small deviation of the approximate solution of SPDE from its exact solution is the smallness of the characteristic geometric dimensions (mesh steps, maximum dimensions of control volumes and final elements). The most justified numerical

* Corresponding author: k0191@mail.ru

criterion for the sufficiency of such a deviation (the quality of the approximate solution) is the value of the maximum absolute residual at all considered (control) points of the computational domain. However, none of the mentioned methods uses this criterion. Taking into account the indicated shortcomings of FDM, CVM and FEM, a more effective, from our point of view, method for solving SPDE is proposed.

2 Control point method

The method developed at ESI SB RAS for solving systems of linear partial differential equations (control point method - CPM) [7-9] reduces the solution of this problem to a linear programming (LP) problem. As a result of solving the LP problem, coefficients of polynomials are selected that describe the dependences of variables on time and spatial coordinates sought from the system of differential equations. In this case, the entire region of change in time and spatial coordinates is divided into blocks and for each block polynomials with their own coefficients are found. Control points are located with sufficient density both in the internal parts of blocks and on their boundaries. At these points, residuals of the following types are determined.

1. Residuals characterizing the correspondence of polynomials to differential equations (differential equations are considered in implicit form). When substituting into differential equations the values of variables and their derivatives determined from polynomials at the coordinates of a specific control point, residuals are determined that characterize the proximity of the polynomials to the desired functions at the corresponding control point. These residuals are determined for all control points.
2. Residuals characterizing the correspondence of polynomials to the initial and boundary conditions. These are discrepancies between the known values of variables and/or their derivatives at some control points and the same values determined from polynomials. Such residuals are determined for those control points that lie on the boundaries of the region of change in time and spatial coordinates, for which the values of the sought variables and/or their derivatives are specified.
3. Residuals equal to the differences between the values of the variables and their derivatives included in the differential equations, determined at control points lying on the border of adjacent blocks using the corresponding polynomials of these blocks. Residuals are determined for all points lying on the boundaries of adjacent blocks.

If the coordinates of the control point and the coefficients of all polynomials are known, then it is easy to determine the numerical values of the desired variables and their derivatives at a given point. Substituting these values into differential equations specified in implicit form will allow us to determine residuals of the first type. Subtraction from known values of variables and/or their derivative values obtained from polynomials (for the corresponding control point) allows us to obtain residuals of the second type. Subtraction of the desired variables and their derivatives obtained using the polynomial of one adjacent block from the corresponding values obtained using the polynomial of another adjacent block (at control points lying on the border between adjacent blocks) will allow us to obtain residuals of the third type.

Since the absolute residuals have different dimensions, comparing them with each other is incorrect. Therefore, in CPM, relative residuals are used, obtained by dividing the absolute residuals by their maximum permissible values. The task is to minimize the maximum relative residual.

It should be noted that each residual in CPM is replaced by two constraints-inequalities of the form:

$$\begin{aligned} g_{jk}^+ &= z - \theta_{jk}^{rel}(d_k, x) \geq 0, \\ g_{jk}^- &= z + \theta_{jk}^{rel}(d_k, x) \geq 0, \\ j &= 1, \dots, M_k, \\ k &= 1, \dots, K, \end{aligned}$$

where $z > 0$ – is an auxiliary parameter, common for the constraints-inequality of all residuals of the problem,

θ_{jk}^{rel} – relative residual relating to the k -th control point and having number j in the list of residuals of the k -th control point,

x – vector of optimized parameters, including in the general case the coefficients of all polynomials of the problem,

d_k – control point coordinates,

M_k – number of residuals related to the k -th control point,

K – number of control points.

The relative residual θ_{jk}^{rel} is determined from the expression $\theta_{jk}^{rel} = \frac{\theta_{jk}^{abs}}{\varepsilon_{jk}^{max}}$,

where ε_{jk}^{max} – maximum permissible value of absolute residual module θ_{jk}^{abs} .

Constraint g_{jk}^+ prevents θ_{jk}^{rel} from becoming greater in absolute value than z if the residual is positive, and constraint g_{jk}^- prevents θ_{jk}^{rel} from becoming greater in absolute value than z if the residual is negative.

The value M_k depends on: the type of control point (internal; lies on the boundaries of changes in time and/or spatial coordinates; lies on the boundary between blocks); number of differential equations; the number of conditions taken into account at the boundary of changes in time and spatial coordinates; number of derivatives included in differential equations. In general, the linear programming problem of selecting coefficients of polynomials that provide a minimum of the maximum modulus of the relative residual will take the form:

$$\min_{z, x} Z$$

under conditions

$$\begin{aligned} g_{jk}^+ &= z - \theta_{jk}^{rel}(d_k, x) \geq 0, \\ g_{jk}^- &= z + \theta_{jk}^{rel}(d_k, x) \geq 0, \\ j &= 1, \dots, M_k, \\ k &= 1, \dots, K, \\ x - \underline{x} &\geq 0, \end{aligned}$$

where x – N -dimensional vector,

\underline{x} – a vector whose components are equal to the minimum values of the corresponding components of the vector x .

Moreover, the total number of constraints – inequalities

$$M = 2 \sum_{k=1}^K M_k .$$

Note that in real problems N and M are very large values. In this case, M is significantly larger than N .

In general, the LP problem being solved can be represented as follows:

$$\min \sum_{i=1}^N c_i x_i , \tag{1}$$

under conditions

$$\sum_{i=1}^N a_{ji} x_i + b_j \geq 0 , \tag{2}$$

$$j = 1, \dots, M ,$$

$$x_i \geq \underline{x}_i , \tag{3}$$

where x_i – i - th optimized parameter,

c_i – coefficient for the i -th optimized parameter in the objective function,

a_{ji} – coefficient in the j -th constraint-inequality, with the i -th parameter being optimized,

b_j – free term in the j -th constraint-inequality,

\underline{x}_i – minimum value of the optimized parameter x_i (can be negative).

3 Methods for solving the LP problem

To solve problem (1)-(3), the simplex method, as well as interior point methods, can be used. When using the simplex method, problem (1)-(3) is transformed into a canonical form with constraints-equality. In this case, M additional variables are introduced. The total number of optimized parameters will be $M + N$. In the simplex method, movement to the optimum is carried out along the vertices of the simplex. It is known that the number of operations required to solve a problem increases with the size of the problem, i.e. as $M + N$ grows exponentially [10]. In this regard, as the dimension of the problem increases, the efficiency of the simplex method drops sharply.

In interior point methods, the movement towards the optimum is carried out within the feasible region. Two groups of such methods can be distinguished:

- 1) methods based on fitting into the permissible ellipsoidal region (I. I. Dikin [11], N. Karmarkar [10]);
- 2) unconstrained sequential minimization method with logarithmic penalty function (A. Fiacco, G. Mc Cormick [12]).

To use the first group of methods, problem (1)-(3) needs to be transformed into canonical form with an increase in the number of optimized parameters to $M + N$. The second group of methods does not require such a transformation and its number of optimized parameters remains equal to N . It is known that for interior point methods the number of operations increases polynomially with the number of optimized parameters [10]. It should be noted that in the methods of the first group, at each iteration, a system of linear algebraic equations of dimension $M + N$ is solved. In the methods of the second group, when using the most efficient Newton method for unconditional minimization, a system of linear algebraic equations of dimension N is solved for each iteration. Since in the linear programming problems under consideration M is significantly larger than N , and the costs of solving SLAE constitute (if their dimension is large) the majority of the labor

costs for solving a linear programming problem, it is preferable to use methods of unconditional sequential minimization with a logarithmic penalty function.

In the method of unconditional sequential minimization with a logarithmic penalty function, the following are specified: $x_1^{beg1}, \dots, x_N^{beg1}$ – initial values of the optimized parameters at the 1st iteration in which all restrictions are satisfied as strict inequalities;

$R_1 > 0$, initial parameter value,

θ^{opt} – permissible deviation of the objective function from the optimal point.

For $k = 1, 2, \dots$ the problem is solved:

$$\min_{x_1, \dots, x_N} \left[L^k = \sum_{i=1}^N c_i x_i - R_k \sum_{j=1}^M \ln \left(\sum_{i=1}^N a_{ji} x_i + b_j \right) - R_k \sum_{i=1}^N \ln(x_i - \underline{x}_i) \right], \quad (4)$$

where k – is the iteration number (unconditional minimization).

Let us denote by $x_1^{*k}, \dots, x_N^{*k}$ the optimal values of the parameters at the k -th iteration.

The initial values of the optimized parameters at $k+1$ iteration is determined from the conditions $x_1^{beg k+1} = x_1^{*k}, \dots, x_N^{beg k+1} = x_N^{*k}$.

The value of the parameters at $k+1$ iteration is determined from the expression:

$$R_{k+1} = \frac{R_k}{K}, \text{ where } k > 1.$$

The process of unconditional sequential minimization stops at the k -th iteration if the condition is met:

$$\theta^{opt} \geq R_k M.$$

Experience has shown that the most effective method for solving problem (4) is Newton's method.

Let us consider the operation of Newton's method at the k -th iteration (unconditional minimization). Let us denote by t the step number of Newton's method. For $t=1$ the initial values of the optimized parameters are set from the conditions

$$x_1^t = x_1^{beg k}, \dots, x_N^t = x_N^{beg k}$$

1. For the current t at point x_1^t, \dots, x_N^t , the gradient vector $\nabla L(x^t)$ and the matrix of second derivatives $H(x^t)$ of the logarithmic penalty function are determined.

In this case, the i -th component of the gradient vector is determined from the expression

$$\nabla L_i(x^t) = c_i - R_k \sum_{l=1}^M \frac{a_{li}}{\left[\sum_{p=1}^N (a_{lp} x_p + b_l) - \frac{R_k}{(x_i - \underline{x}_i)} \right]},$$

and the element lying at the intersection of the i -th row and the j -th column of the matrix $H(x^t)$ is determined from the expressions:

$$H_{ij} = \frac{\partial^2 L(x^t)}{\partial x_i \partial x_j} = R_k \sum_{l=1}^M \frac{a_{li} a_{lj}}{\left[\sum_{p=1}^N (a_{lp} x_p + b_l) \right]^2} + \frac{R_k}{(x_i - \underline{x}_i)^2}, \text{ if } i = j, \quad (5)$$

$$H_{ij} = \frac{\partial^2 L(x^t)}{\partial x_i \partial x_j} = R_k \sum_{l=1}^M \frac{a_{li} a_{lj}}{\left[\sum_{p=1}^N (a_{lp} x_p + b_l) \right]^2}, \text{ if } i \neq j.$$

2. SLAE is being solved

$$H(x^t)(x - x^t) = H(x^t)S^t = \nabla L(x^t), \quad (6)$$

where $S^t = x - x^t$ – is the direction of descent.

3. A one-dimensional minimization of the logarithmic penalty function along the descent direction is carried out

$$\lambda^t = \arg \min_{\lambda} L(x^t + \lambda S^t)$$

and the point for $t+1$ step of Newton's method is determined

$$x^{t+1} = x^t + \lambda^t S^t.$$

If $\|\nabla L(x^1)\| \geq \|\nabla L(x^t)\| B$,

where $B \gg 1$, then the process of unconditional minimization at the t -th iteration stops and assumes $x_1^{*k} = x_1^t, \dots, x_N^{*k} = x_N^t$.

Since with increasing dimension of the SLAE the number of operations required to solve it increases approximately in proportion to the cube of its dimension, then for large LP problems almost all the time of solving them by the method of unconditional sequential minimizations with a logarithmic penalty function is spent on solving the SLAE (6). Therefore, it is important to find an approach to significantly reduce this time.

Analysis of expression (5) shows that at the intersection of the j -th row and the i -th column of the matrix H (at $i \neq j$) there will be a non-zero element if there is at least one inequality constraint for which both coefficients at the i -th and j -th variable are not equal to zero. If there is no such restriction, then at the intersection of the j -th row and the i -th column there will be a zero.

As stated earlier, at the boundaries between adjacent blocks, residuals of the third type are determined, which simultaneously depend on the coefficients of the polynomials (optimized parameters) of these blocks, which leads to the appearance of a large number of non-zero elements in the matrix H . As a result, it is almost impossible to identify groups of blocks for each of which would not have non-zero second derivatives with respect to the parameters of this group and the parameters of several other groups, which complicates the solution of the SLAE.

4 Structural approach to solving SLAE

To solve this problem, a special approach to organizing the space of parameters (time and coordinates) is proposed, which it is advisable to call structural. Its essence is as follows. The "simple" blocks discussed earlier are combined into large blocks or superblocks.

Boundaries are determined between superblocks (sets of points with a dimension one less than the dimension of the blocks). Polynomials are formed for the boundaries, which reflect the dependence of the sought functions on the parameters of the boundaries. In this case, it is possible to form “boundary” blocks for each of which have their own polynomials. Communications at the boundaries of superblocks are organized as follows. For a control point lying on the specified boundaries, two residuals are determined for each “joined” variable. One represents the difference between the calculated values at this point of a given variable determined by the polynomial for the boundary and the calculated values determined by the polynomial related to the first adjacent superblock, and the second represents the difference between the calculated values at this point of the same variable determined by the polynomial for the boundary and the calculated values determined by the polynomial related to the second adjacent superblock. We call the optimized coefficients of the boundary polynomials the parameters of the connection between superblocks. As a result, non-zero values of the second derivatives will be: 1) for the optimized parameters of the same superblock; 2) superblock parameter and communication parameter between superblocks; 3) two communication parameters. This leads to the structure of the matrix H as follows:

$$H = \begin{pmatrix} D_1^{in} & 0 & \bullet\bullet & 0 & D_1^{com} \\ 0 & D_2^{in} & \bullet\bullet & 0 & D_2^{com} \\ \bullet\bullet & \bullet\bullet & \bullet\bullet & \bullet\bullet & \bullet\bullet \\ \bullet\bullet & \bullet\bullet & \bullet\bullet & \bullet\bullet & \bullet\bullet \\ 0 & 0 & \bullet\bullet & D_{n_{bl}}^{in} & D_{n_{bl}}^{com} \\ (D_1^{com})^T & (D_2^{com})^T & \bullet\bullet & (D_{n_{bl}}^{com})^T & D^{com} \end{pmatrix}$$

where D_l^{in} – square matrix ($N^{in} * N^{in}$) with elements $\frac{\partial^2 L(x^t)}{\partial x_i \partial x_j}$, where i and j belong to

the set of numbers of internal optimized parameters of the l -th superblock.

D_k^{com} – matrix ($N^{in} * N^{com}$) with elements $\frac{\partial^2 L(x^t)}{\partial x_i \partial x_j}$, where i – belongs to the set of

numbers of the k -th block, j – belongs to the set of numbers of communication parameters.

D^{com} – square matrix ($N^{com} * N^{com}$) with elements $\frac{\partial^2 L}{\partial x_i \partial x_j}$, where i and j belong to the

set of communication parameter numbers.

$$S = \begin{pmatrix} S^1 \\ S^2 \\ \cdot \\ \cdot \\ \cdot \\ S^{n_{bl}} \\ S^{com} \end{pmatrix}, \quad \nabla L(x) = \begin{pmatrix} \nabla L(x)^1 \\ \nabla L(x)^2 \\ \cdot \\ \cdot \\ \cdot \\ \nabla L(x)^{n_{bl}} \\ \nabla L(x)^{com} \end{pmatrix}.$$

The SLAE solved in Newton’s method will include n_{bl} matrix-vector expressions of the form

$$D_l^{in} S^l + D_l^{com} S^{com} = \nabla L(x)^l, \quad l = 1, \dots, n_{bl}, \quad (7)$$

and one matrix-vector expression

$$\sum_{l=1}^{n_{bl}} (D_l^{in})^T S^l + D_l^{com} S^{com} = \nabla L(x)^{com}, \quad (8)$$

where S^l – direction of descent according to the internal optimized parameters of the l -th superblock, S^{com} – direction of descent according to communication parameters, $\nabla L(x)^l$ – components of the gradient vector of the logarithmic function according to the internal optimized parameters of the l -th superblock, $\nabla L(x)^{com}$ – also related to communication parameters.

The first n_{bl} expressions (7) can be resolved relative to S^l

$$S^l = (D_l^{in})^{-1} \nabla L(x)^l - (D_l^{in})^{-1} D_l^{com} S^{com}, \quad l = 1, \dots, n_{bl} \quad (9)$$

Substituting (9) into (8) we have

$$\left[D^{com} - \sum_{l=1}^{n_{bl}} (D_l^{com})^T (D_l^{in})^{-1} D_l^{in} \right] S^{in} = \left[\nabla L(x)^{in} - \sum_{l=1}^{n_{bl}} (D_l^{com})^T (D_l^{in})^{-1} \nabla L(x)^l \right]. \quad (10)$$

Expression (10) is a SLAE with dimension N^{com} . Thus, to solve the original SLAE of dimension $N^{com} n_{bl} + N^{com}$, it is necessary to invert n_{bl} square matrices of dimension N^{in} , solve one SLAE of dimension N^{com} and perform a certain number of operations of multiplication and addition of matrices whose dimension is much lower than the dimension of the original matrix H . In this case, matrix multiplication can be performed in parallel.

An example is a SLAE with a dimension of about 7000. Solving it using the best available solver required about 1 second of computer time on a computer with a 16-core processor.

Solving the same SLAE using the stated structural approach and the same computer took 0.04 seconds when selecting 12 superblocks and a set of communication parameters with a dimension of about 538.

5 Conclusion

It is shown that the most effective method for solving LP problems to which the problems of solving systems of linear differential equations are reduced within the framework of CPM is the interior point method with a logarithmic penalty function. With a large dimension of the LP problem, the main computer time is spent on solving the SLAE within the framework of Newton's method, used for unconditional minimization of the logarithmic penalty function. To radically reduce this expense, a structural approach is proposed, in which all optimized parameters of the LP problem (polynomial coefficients) are divided into internal optimized parameters of superblocks and optimized parameters of communication between superblocks that specify the coefficients of polynomials that describe the behavior of the desired functions at the boundaries between superblocks. As a result, the matrix of the second derivatives of the logarithmic penalty function acquires a structure that allows one to reduce the solution of the SLAE to matrix inversion and the solution of the SLAE of a much smaller dimension, which radically reduces the time for solving the original SLAE.

The study was supported by the Russian Science Foundation grant No. 22-29-01654,
<https://rscf.ru/project/22-29-21654/>

References

1. G. Smith, Numerical solution of partial differential equations: finite difference methods. Oxford university press, p. 321, 2004.
2. S.Kazem, D. Mehdi. "Application of finite difference method of lines on the heat equation." Numerical Methods for Partial Differential Equations 34.2: pp. 626-660, (2018).
3. S. Patankar, D. Spalding, Heat and mass transfer in boundary layers. Morgan-Grampian, 1968.
4. R. Eymard, Th. Gallouët, R. Herbin, Finite volume methods, volume 7 of Handbook of Numerical Analysis. Elsevier, 2000.
5. J. Zhu, O. Zienkiewicz, R. Taylor, The finite element method: its basis and fundamentals. Elsevier, 2005.
6. C. Johnson, Numerical solution of partial differential equations by the finite element method. Courier Corporation, 2012.
7. A. Kler, D. Apanovich, A. Maximov, An effective method for calculating the elements of thermal power plants, which are reduced to solving systems of partial differential equations, *E3S Web of Conferences*, 2020, **209**. - P. 03029. DOI: 10.1051/e3sconf/202020903029.
8. A.M. Kler, D.V. Apanovich, The method of dynamic calculations of elements of thermal power plants, which reduces the solution of systems of differential equations in partial derivatives to the solution of linear programming problems, *Information and mathematical technologies in science and management*, 2022, **3**(27), pp. 148-161. DOI:10.38028/ESI.2022.27.3.014.
9. A.M. Kler, D.V. Apanovich, Numerical simulation of the water hammer phenomenon taking into account the hydraulic friction of the pipeline by the control point method, *Analysis and Data Processing Systems*, 2022, **3** (87), pp. 59-74. DOI: 10.17212/2782-2001-2022-3-59-74.

10. N.K. Karmarkar, A new polynomial-time algorithm for linear programming, *combinatorica*, 4, 373-395 (1984) DOI: 10.1007/BF02579150
11. I. I. Dikin, Iterative solution of problems of linear and quadratic programming, *Dokl. Akad. Nauk SSSR*, 174:4, 747–748 (1967)
12. A.V. Fiacco, G.P. Mc Cormick, *Nonlinear programming: sequential unconstrained minimization techniques*. John Wiley, New York. (1968)