DETERMINATION OF HEAT LOSS AND MICROCLIMATE PARAMETERS IN NON-PRODUCTION BUILDING

A. Yerokhin, H. Zatserklianyi

Astract: An application package for the analysis of heat loss and microclimate in a non-production building is proposed with the purpose of determining or clarifying the building norms and rules used in metrological studies, which accompany each stage of construction or reconstruction of buildings.

Keywords: application package, modeling, heat loss, microclimate, non-production building

At all stages of construction and modernization of residential, administrative and other buildings, metrological studies are conducted to check compliance with construction norms and rules.

With the rapid development of construction technologies and the appearance of new building materials, the approved norms and rules quickly become obsolete and need to be clarified.

Experimental setting of norms and rules requires a great deal of time and money. Therefore, it is expedient to use a computational experiment to solve this problem.

This work is devoted to the organization of computational experiment for determining the parameters of microclimate and heat loss in the building.

The modern non-production building is a complex functional and constructive system with a variety of thermal processes, which take place in it. Extensive experimental study of the thermal process in the building is quite laborious and costly, and often impossible, because no one has ever built buildings for experiments. Therefore, the only possible method of studying the thermal conditions of buildings is mathematical modeling.

Mathematical modeling of complex systems, which includes the thermal conditions of the building, is expedient to use for organizing a computational experiment.

Computational experiment is a method of studying objects or processes by constructing their mathematical model and subsequent numerical study of this model that allows „to play“ their behavior under different conditions [1, 2]. A program realizing a mathematical model can be considered as an analogue of the experimental setup used in the full-scale experiment.

The preparation of computation programs is one of the most laborious stages in the numerical experiment. From the point of view of program implementing, the following features of the computational experiment are essential: multimodeling and multivariance. Multimodeling means that during a computational experiment the mathematical model changes many times and it is refined; and multivariance means that the fulfilled refinements of the model can be combined in a variety of combinations.

These features reveal in fact that, firstly, the main work on carrying out the computation experiment does not aim at a single execution, but on frequent repetition of the described cycle. This to a certain extent is spreading for the program: the overwhelming part of the efforts of developers is not the creation of the first version of the calculation program, but its various modifications.

Secondly, the decisions (and along with them the corresponding programs) taken on the next cycle of the computation experiment are, as a rule, not discarded, but can be used subsequently for other calculations.

Thus, the main area of activity with programming the tasks of a computation experiment is not the creation of new ones, but the development of existing programs. This development is carried out, as a rule, not by replacing the existing modules with their more advanced versions, but by including new modules in the program fund, which reflect the various solutions that are adopted during the experiment.

Orientation in such a large and complex software farming is not easy. The main content of the work is not the writing of new program modules, although this always takes place, but the construction of calculation programs („research installations“) from modules, and the solution of a number of technical problems associated with the development and maintenance of the program fund. Performing such works based on standard software requires quite a lot of effort. Therefore, special system facilities are necessary. Such funds should provide for the storage, updating and modification of the program fund, as well as a simple, flexible and fast mechanism for assembling various program variants for specific calculations from individual fund modules. Such a tool is the application package.

The application package, focused on analyzing the thermal conditions of a non-production building consists of three main parts: a preprocessor, a calculator and a postprocessor. Each of these parts is independent and can be used as a separate program. Connection between programs is carried out by using standardized data streams (files).

The preprocessor, calculator and postprocessor are written in C# for the Linux operating system that allows to use a wide range of capabilities, which are provided by this operating system and the programming language.

The preprocessor is designed for visual input and editing of information, both geometric and thermophysical. The preprocessor sheath is built on the principle of a multidocument interface, which allows simultaneous editing of several data files.

The solver on primary data, obtained from the preprocessor, automatically generates an algorithm and a program of internal modules for solving a specific problem of analyzing the thermal process in a building or its element.

Modularization is realized by creation of the building design structure and the object-oriented approach to the analysis of the subject area.

The residential building is presented in the form of a tree, the lowest level of which is the elementary element (the components of the enclosing structure and the final parts of the steam-air mixture: the room, part of the staircase, part of the attic, part of the basement). Some elements in a particular house may be absent.

Such a structure allows to consider any building from the same positions, and interrelated and interdependent thermal conditions throughout the building to lead to the thermal regime in elementary elements, coordinating them with appropriate boundary conditions.

Today, there are several technologies for object-oriented development of application program systems, based on the construction and interpretation of computer models of these systems. We use the methodology OMT (Object Modeling Techniques) [3].

In OMT technology, the projected program system is modeled by three interrelated models:

• a functional model, in which the interaction of individual parts of the system (both data and control) is provided in the process of its operation;

• a dynamic model that describes the operation of individual parts of the system;

• an object model, which represents static, structural aspects of the system, mainly related to data.

The implementation of the functional model of heat loss analysis within the framework of object-oriented design in non-production buildings is as follows:

1. Determination of heat input due to radiation and convection. Transfer of results to the block for solving the problem of convective heat transfer.

2. Solution of the problem of convective heat transfer. Transfer of results to the block for solving the problem of heat loss due to thermal conductivity through the elements that supply the enclosing structure.

3. Determination of heat loss through the enclosing structure.

In the general case, the model of the heat process system in a non-production building consists of the following elements:

1. The external environment (EE);
2. Element of the enclosing structure (ES);
3. Steam air space (SAS);
4. Heating system (HS);
5. Lighting system (LS);
6. Solar radiation (SR);
7. People (P).

Such a methodology is adopted. Elementary elements of the system are typed, i.e. each elementary element of the system belongs to a certain type.

The state of the elementary element of the system is determined by the internal processes reflected in the submodel, and by the influences of other elementary elements of the system on it.

Each elementary element of the system receives information (parameters) about the state of those elementary elements of the system that have influence on it. Generalized information on the presence of such an influence for each elementary element of the system is supplied by a matrix of interelement effects. The names of the rows and columns in this matrix correspond to the names of the elementary elements of the system listed above. The value of the matrix element equal to 1 and placed in the row with the name of i and in the column with the name of k means that the elementary element named i is exposed to influence from the elementary element named k, and if this value is 0, then such influence is absent. This matrix is used to construct function interfaces that describe the behavior of elementary elements of the system.

Modeling the behavior of the system is performed on a certain time interval. The time step of the modeling is variable. In this case, the execution of each step includes 3 stages:

1. Each elementary element of the system gets information about the state of elementary elements that affect it at a given moment in time.

2. Each elementary element of the system calculates its state at the next moment in time taking into account the magnitude of the current modeling step (without passing to the next state). The calculations are performed on the basis of the corresponding submodel of the elementary element of the system.

3. Each elementary element of the system realizes (transfers itself into) the next state.

According to the ideology of object-oriented modeling, all the elementary elements of the model of the system are typed, i.e. belong to a certain class and each element of the system model is represented by an object of the corresponding class. In this case, the structure, properties and behavior of an object of a given class are uniquely determined by the description of this class. The class defines the information structure of the model element of the system and contains a set of functions (methods) that determine the evolution of its state. Hereat, the structure of interelement interactions is determined by the matrix of interelemental influences and is given in the corresponding classes in the form of lists of arguments of the functions-members of the class, which carry out the second stage of the next step in modeling the system.

In OMT technology, a dynamic model of a projected application package describes the operation of individual parts of this system. In our case, these are separate modules, which represent methods for analyzing thermal conductivity, convective heat exchange, and radiant radiation.

Elements of the enclosing structure (external walls, windows, doors, floors) are often multi-layered. Each layer has different thickness and different thermophysical characteristics. At what some layers can be thin enough. It is considered that a three-dimensional non-stationary process is observed in the element of the enclosing structure. A one-dimensional non-stationary process is a separate case. The method of calculation uses a modified finite element method. The modification concerns the separation of finite elements into homogeneous and heterogen.eous ones. Homogeneous elements have a regular geometric shape (rectangular parallelepiped) with guides, which are parallel to the coordinate axes, and a homogeneous composition, which in this case means constancy of the coefficient of thermal conductivity in the element. Heterogeneous elements are the other case. These are elements containing thin layers, within which the coefficient of thermal conductivity varies many times. Using the standard finite element method for such elements leads to physically incorrect solution profiles. Of course, such defects are corrected by the introduction of a detailed grid, but this can lead to an unacceptable increase in the size of the grid problem. Form functions for homogeneous elements are sought in the form of trilinear functions. For heterogeneous elements, if the thermal conductivity is piecewise constant and the required form function changes only in a direction that is perpendicular to the thin layer (the arrangement of the layer in the element can be arbitrary), the shape function is chosen piecewise linear and is found from conditions for reproducing the exact solution of the one-dimensional heat-conduction equation in the finite element. This way the selected kind of the form function in a heterogeneous element provides a correct description of the temperature field in the presence of a thin layer without using a detailed grid.

For the heat conduction problem in the elementary element under consideration, depending on the connection with other elements, the boundary conditions of the first, second, third and fourth kind can be present. This method is described in detail in [4].

Convective heat exchange is considered in the approximation of the non-stationary three-dimensional turbulent flow of a two-component vapor-air mixture in the presence of internal sources and sinks of energy in an elementary element. Modeling turbulence is performed on the basis of Menter’s SST-model [5]. For the numerical realization of the problem of convective heat and mass exchange, the finite volume method is used [6]. In this case, the equations of gas dynamics are written in integral form. The boundary conditions, depending on the connection with other elements, are the conditions on the permeable boundaries of the calculated region (input and output), boundary conditions on a solid impermeable wall, boundary conditions of symmetry, periodic boundary conditions. A special case of this method is a quasi-stationary model. This method is described in detail in [7].

The supply of heat from heating and lighting systems, solar radiation and people is modeled by punctual, surface and volumetric heat sources. At which the heat input due to convection and radiation is separately taken into account. The intensities of these sources are found using known techniques [8, 9].

These methods require some experimental data (materials of enclosing structures, their thermal and physical characteristics, type of heating device, the surface area of the device‘s heating, the type of lighting lamp, the angle of inclination of the glazing to the horizontal plane, and so on). The necessary experimental data are designed in the form of special tables and are an integral part of the application package.

The data obtained in a result of calculations are transferred to the postprocessor, which performs their further processing. The postprocessor allows to represent the properties of the process under investigation in the required form.

Interactive graphical representation of the program contains standard controls, which are typical for the multi-document interface, such as the main menu and dialog boxes.

Literature

[1] **Yershov A. P., Il‘in V. P.** Pakety programm - tekhnologiya resheniya prikladnykh zadach.- Novosibirsk: VTS SO AN SSSR, 1978.

[2] Pakety prikladnykh programm: programmnoye obespecheniye vychislitel‘nogo eksperimenta. Samarskiy A.A. (Red.) M.: Nauka. 1987. – 150 s.

[3] **Booch, G.** Object-oriented analysis and design [Text] / G. Booch // Addison-Wesley Publishing Company. - 2007. - 534 p.

[4] **Kutsenko O. S., Zatserklianyi H.A.** Modelyuvannya teploobminu cherez ogorodzhuval‘ni poverkhni budivli. Visnyk NTU «KhPI». 2012. № 42 (948).

[5] **Menter F.R.** Two Equation Eddy Viscosity Turbulence Models for Engineering Applications // AIAA J. - 1994. - 32, No. 8. - P. 1598-1605.

[6] **Anikeyev A.A., Molchanov A.M., Yanyshev D.S.** Osnovy vychislitel‘nogo teploobmena i gidrodinamiki. Uchebnoye posobiye.

[7] **Yerokhin A.L, Zatserklianyi H.A.** Informatsiyna tekhnologiya analizu konvektyvnogo teploobminu v prymishchenni budivli. Systemy obrobky informatsiyi. 2016. No. 9 (146), s. 187-192.

[8] **Weitzmann P.** Modeling building integrated heating and cooling systems / P. Weitzmann // Danmarks tekniske universitet. - 2004. - 239 p.

[9] **L‘vovskiy I.B.** Raschet postupleniya teploty solnechnoy radiatsii v pomeshcheniya [Tekst] / I.B. L‘vovskiy, B.V. Barkalov – M. – 1993. – S. 32.

Information about the authors:

Andriy L. Yerokhin
Doctor of Technical Sciences, Professor
Department of Software Engineering
Kharkiv National University of Radio Electronics
Science (Nauka) ave, 14, Kharkiv, Ukraine, 61166
e-mail: andriy.yerokhin@nure.ua
Contact phone: 38050-777-62-10

Heorhii A. Zatserklianyi
Post-graduate student
Department of Software Engineering
Kharkiv National University of Radio Electronics
Science (Nauka) ave, 14, Kharkiv, Ukraine, 61166
e-mail: george.zatserklyany@gmail.com
Contact phone: 38095-572-21-07