APPLICATION OF OBJECT-ORIENTED FINITE ELEMENT PROGRAMMING FOR CASTINGS SOLIDIFICATION PROBLEMS

SCZYGIOL Norbert, SZWARC Grzegorz
Institute of Mechanics and Machine Design,
Technical University of Częstochowa
42-201 Częstochowa, ul. Dąbrowskiego 73, POLAND

ABSTRACT

In the work application of object-oriented technique for modelling of castings solidification has been presented. For numerical solution of the problem the FEM has been employed. Object constituting the program for castings solidification and some characteristic features and methods of the object-oriented technique have been discussed.

1. INTRODUCTION

In structural technology, computational programs are developed from the very beginning or, at least, with little usage of procedures derived in already existing programs. It results from this, that programs are mostly employed for solving specific tasks and therefore it is easier to formulate new procedures than to transform the existing ones. Object-oriented programming is a very up-to-date method of software development. The basic reason for rapid development of this technique, which is now taking place, is aiming at greater effectivity and clarity of computer program structure, with guarantee of higher data security at the same time [1,2]. Each of new defined data type is related to the procedures which are authorized to the change of their values only. Such a link between data type and procedures, which are operating on them, forms in the object-oriented programming the type of data called objects. Higher efficiency, in comparison to the structural programming, is effected by the so called inheritance. In this way we are able to create a hierarchy of objects which inherit successively after themselves attributes and methods (data and procedures). To the inheriting objects new attributes and methods may be added. There also exist a possibility to overlay the existing methods by the new ones. Moreover, the same names can be used for the similar acting methods occurring in different objects, what allows one to uniform the nomenclature.

As development and archivization of the new objects progress, new programs formulation is consisting in assembling them from the existing objects. Object-oriented technique is especially
useful in development of the complete software, but also in testing different models and solutions in the simulation of the specific phenomena, e.g. heat flow [3], solidification etc.

In this paper implementation of the object-oriented technique to solidification modelling is discussed. For numerical realization of the problem the finite element method has been assumed. Objects, which consist the program for solidification simulation, and also some specific features of the object-oriented technique utilized in this work, are presented and discussed.

2. SOLIDIFICATION MODEL

Heat flow in the solidifying casting is given by the following equation

$$\nabla \cdot (\lambda \nabla T) + Q = c_p \frac{\partial T}{\partial t},$$  \hspace{1cm} (1)

where $Q = L \cdot \frac{\partial f_s}{\partial t}$ is the so called source term, taking into consideration latent heat release, $L$ is the latent heat and $f_s$ is the solid fraction. Introducing enthalpy, defined as

$$H = \int_{T_{ref}}^{T} c_p(T) \, dT,$$  \hspace{1cm} (2)

where $T_{ref}$ is the reference temperature and calculating their derivative with respect to the temperature

$$\frac{dH}{dT} = c_p(T) = c^*(T),$$  \hspace{1cm} (3)

where $c^*$ is the so called effective heat capacity, the equation (1) may be transformed to the form

$$\nabla \cdot (\lambda \nabla T) = c^* \frac{\partial T}{\partial t}.$$  \hspace{1cm} (4)

2.1. Effective heat capacity approximation

Assuming that the latent heat is released uniformly over the whole solidification range of temperatures, the effective heat capacity is calculated from formulae [4, 5]

$$c^* = c_i \rho, \quad T < T_s$$

$$c^* = c_f \rho + \rho \frac{L}{T_L - T_s}, \quad T_s \leq T \leq T_L$$

$$c^* = c_f \rho, \quad T > T_L$$  \hspace{1cm} (5)

It may also be derived by applying directly the equation (3). Morgan et al. substitute the enthalpy derivative by backward difference representation [4,6]
\[ \frac{H^n - H^{n-1}}{T^n - T^{n-1}} \]

where \( n-1 \) and \( n \) denote the time levels. In some cases this substitution (Morgan's method) may lead to oscillations in solution, especially in the neighborhood of the boundaries of solidification range. In order to avoid this inconvenience, two different averaging expressions are often employed which approximate the effective heat capacity [4,6]:

a) expression proposed by Dei Giudice et al., in which direction cosines of the temperature gradient are taken into consideration additionally (Dei Giudice's method)

\[
\begin{align*}
\frac{\gamma}{\gamma_1} &= \frac{\left(\sum_{i=1}^{3} (\frac{\partial H}{\partial x})(\frac{\partial T}{\partial x}) + (\frac{\partial H}{\partial y})(\frac{\partial T}{\partial y}) + (\frac{\partial H}{\partial z})(\frac{\partial T}{\partial z})\right)}{\left[\left((\frac{\partial T}{\partial x})^2 + (\frac{\partial T}{\partial y})^2 + (\frac{\partial T}{\partial z})^2\right)\right]} \\
\end{align*}
\]

b) expression proposed by Lemmon, in which it is assumed that the temperature gradient is normal to the solidification interface (Lemmon's method)

\[
\begin{align*}
\frac{\gamma}{\gamma_1} &= \frac{\left(\sum_{i=1}^{3} (\frac{\partial H}{\partial x})^2 + (\frac{\partial H}{\partial y})^2 + (\frac{\partial H}{\partial z})^2\right)}{\left[\left((\frac{\partial T}{\partial x})^2 + (\frac{\partial T}{\partial y})^2 + (\frac{\partial T}{\partial z})^2\right)\right]} \\
\end{align*}
\]

3. SOLUTION BY THE FINITE ELEMENT METHOD

Applying the finite element method and making discretization of the space in equation (4), one obtains

\[ KT + MT = b, \]

where \( K \) is the conductivity matrix, \( M \) denotes the mass matrix, also called capacity matrix, \( T \) is the temperature vector and \( b \) is vector of the node sources, called also the boundary conditions vector. For the single element they are defined as follows

\[
\begin{align*}
K_{ij} &= \int_{\Omega} \lambda \nabla \varphi_i \nabla \varphi_j \, d\Omega, \\
M_{ij} &= \int_{\Omega} c^\ast \varphi_i \varphi_j \, d\Omega, \\
b_i &= \int_{\Gamma} \varphi_i \lambda \nabla T \, d\Gamma,
\end{align*}
\]

where \( \varphi \) is the interpolation function. Equation (9) ought to be time integrated. This integration may be performed by using one-step methods (analysis on two time levels) or multiple-step methods (analysis on three or more time levels) [7,8]. Applying the one-step method, the so called \( \Theta \)-method, we obtain
\[
(M^e + \Theta \Delta t K^e)T^{n+1} = (M^e - (1 - \Theta) \Delta t K^e)T^n + (1 - \Theta) \Delta t h^n + \Theta \Delta t b^{n+1}.
\]

From this equation, depending on the assumed parameter \( \Theta \) value, one of the possible integration schemes is derived, namely: explicit \((\Theta = 0)\), implicit \((\Theta = 1)\), Crank-Nicholson \((\Theta = 0.5)\) or Galerkin \((\Theta = \frac{1}{2})\). The most commonly used method of time integration is the implicit method, which assures stable solutions for arbitrary value of time-step.

4. HIERARCHY AND OBJECT DESCRIPTION

4.1. Object hierarchy

Computer program for simulation of the solidification process, which has been written utilizing object-oriented technique in Turbo Pascal language of programming (based on the model presented in point 2, and solution assumed in point 3) consists of eight basic objects (Fig. 1). Base object \textit{TFemBase} is making use of \textit{TArrays}, \textit{TMaterialInfSys} and \textit{TViewResults} objects. \textit{TArrays} object is responsible for creating dynamic arrays. It forms the arrays on dimensions corresponding to the size of the problem being solved. The size of each array may be greater than the size of the computer memory segment. \textit{TMaterialInfSys} object is the data base for material properties and supplies the base object with information on specific material properties. Information may be sent as individual number, as arrays (when material property is for example the function of temperature) or as formulae. \textit{TViewResults} object is used for graphic representation of calculation results. As it has attributes common with \textit{TVViewGrid} object (used for revising the finite element mesh and boundary conditions), it is its descendant (inheriting object).

\textit{TFemHeat} object inherits after \textit{TFemBase} object all attributes and methods which, in conjunction with its own attributes and methods allows one to perform numerical calculation in the field of the heat flow (without solidification).

Furthermore, \textit{TSolidification} object inherits after \textit{TFemHeat} object. It includes all attributes and methods necessary for performing numerical calculations of solidification.

The program which is denoted on Fig. 1 as \textit{Application} can use \textit{TFemHeat} object, when heat flow simulation is performed only, or \textit{TSolidification} object - when simulation of the solidification process is studied; but also inherits communication with user after \textit{TUserInterface}. Moreover, it also is making use of \textit{TVViewGrid} object.

4.2. Basic objects description

In object \textit{TFemBase} attributes and methods have been gathered which are common for almost all numerical problems solved by the finite element method. In the computer program this object can not exist independently, as it does not contain all the attributes and methods necessary for solving the specific problem. Therefore, it is an abstract object. Objects which inherit properties after \textit{TFemBase} have to complete it - to compose, as a whole, the complete solution.

\textit{TFemBase} object incorporates the following methods: data acquisition, verifying the width of the global matrix (and their optimization, consisting in change of the node numbers), calculation of the band dimensions (i.e. their width and length), reduction of band dimensions because of the given boundary conditions, solution of the set of equations, storing the results
of the performed solutions (of set of equations) and automatic dynamic memory release for other programs after performing the calculations.

![Object hierarchy diagram](image)

**Fig. 1. Object hierarchy (⇒ inheritance, ↔ message)**

*Rys. 1. Hierarchia obiektów (⇒ dziedziczenie, ↔ komunikat)*

*TFemHeat* object inherits all the properties after *TFemBase*. It supplements them on new attributes and methods which are characteristic for heat flow problems. *TFemHeat* includes the following methods: boundary conditions acquisition, calculation the matrix coefficients of the set of equations and vectors of the right-hand side of equations - for the chosen time integration method (Equation (11)), supplementation of coefficient matrix and right-hand side vector on the given boundary conditions, storing the results of calculation on the disc, and dynamic memory release occupied by the boundary condition matrices.

*TSolidification* object inherits all the properties of the *TFemHeat*. It supplements them on attributes associated directly with solidification process modelling, and on methods connected with the manner of numerical implementation of the solidification process. One of the many known schemes may be applied here: the temperature recovery method, the alternating phase truncation method [5], the fictitious heat-flow method [4] or other. Moreover, the *TSolidification* object consists of methods for effective heat capacity calculation (Formulae (5)-(8)), enthalpy (and its derivatives) and solid phase contribution, as well as of methods for dynamic memory release which is occupied by these quantities. Also, modification of the structure of the set of equation is taking place in this object; the structure which is inherited from *TFemHeat* object, applied here for temperature field determination both in casting mould and in the solidifying casting, what is realized by the virtual method mechanism. The storing method of the calculation results on the disc is supplemented on solid phase contribution. On the contrary, the method of boundary conditions acquisition is utilized in the unchanged form.
5. SUMMARY

In this work application of the object-oriented technique for modelling and analysis of the solidifying castings has been shown. The description, which has been presented, may be broadened on three-dimensional modelling of the casting solidification. The new type of variables, which are the objects, facilitates division the wide class of problems on the specific implementation tasks, which are common for many of them. Abstractive objects (built on this base), which can not function separately, may be applied repeatedly in computer software development for specific engineering problems. In this way high elasticity in software development is achieved, where activity consist in adjusting elements of the existing objects to the requirements of the actually investigated problem.

REFERENCES


STRESZCZENIE.

ZASTOSOWANIE TECHNIKI OBIEKTOWEJ DO MODELOWANIA KRZEPNIĘCIA ODLEWÓW METODĄ ELEMENTÓW SKOŃCZONYCH

Technika obiektowa jest nowoczesną metodą tworzenia oprogramowania. Jest ona szczególnie przydatna do tworzenia i testowania modeli oraz rozwiązań w symulacji zjawisk i procesów fizycznych. W pracy przedstawiono zastosowanie techniki obiektowej do modelowania procesów krzepnięcia odlewów. Zadanie rozwiązano numerycznie stosując metodę elementów skończonych. Omówiono obiekty składające się na program do symulacji krzepnięcia, a także pewne charakterystyczne cechy i metody techniki obiektowej.