SURVEYS

NUMERICAL SOLUTION OF HEAT AND MASS IMPULSE TRANSFER PROBLEMS*

W. Zwick

UDC 536.24

1. Introduction. Following the development of computational techniques, the numerical solution of the heat, mass, and momentum transfer equations has acquired much interest in many technical applications. In this connection, an important role is played by the theoretical and experimental justification of the validity region of specific numerical models, and their effectiveness is estimated for the study of various classes of problems.

The numerical solution of heat, mass, and momentum transfer problems can be separated into several stages. The mathematical model is formulated at the first stage. In this case the substantial physical features of the problem must be reflected properly, and its mathematical statement must be formulated correctly, guaranteeing existence and uniqueness of the solution, and stability of the equations. We note that stability of the equations of motion with respect to perturbations of original data is often not guaranteed at large Reynolds numbers.

At later stages this mathematical model is approximated by a system of algebraic equations, and a method of solving them is selected, which can be realized on available computers by means of standard or specially developed programming techniques. The numerical solution then obtained must be verified and interpreted correctly, following which it can be used.

Since many stationary problems can be solved by means of established methods, in what follows all problems are considered in the nonstationary statement in the time interval [0, T] and the spatial region Ω, having a boundary Γ. The time coordinate is denoted by $t \in [0, T/$. For the sake of simplicity it is assumed that the spatial region has two dimensions, while the radius-vector $x \in \Omega \cup \Gamma$ is given by the coordinates $x = (x_1, x_2)$. Besides, it is assumed that heat, mass, and momentum transfer is investigated in a Newtonian incompressible single-phase medium, the effect of electromagnetic forces and radiation is not considered, and only laminar flows are considered.

2. Basic Equations of Heat and Mass Impulse Transfer. The Navier–Stokes equation plays the main role in momentum transfer problems. For nonisothermal flows it is coupled with the heat transfer problems. For nonisothermal flows it is coupled with the heat transfer equation, and in the general case with the diffusion equation as well. Since the latter have an identical shape, without loss of generality the numerical methods can be illustrated on the Boussinesq system of equations, including the heat and momentum transfer equations.

Denoting the velocity by $v = (v_1, v_2)$, the temperature by θ, and the pressure by p, for the appropriately selected dimensionality the Boussinesq equations can be written in the following form:

$$\dot{\theta} - \varkappa\Delta\theta + v_l D_l \theta = f, \tag{1}$$

$$\dot{v}_k - \eta\Delta v_k + v_l D_l v_k + D_k p = \zeta_k \theta + g_k, \tag{2}$$

$$D_l v_l = 0, \tag{3}$$

where k acquires the values 1 and 2, and the summation must be carried out over the subscript ℓ from 1 to 2. Besides, we use here the notation $D_l = \partial/\partial x_l$, $\Delta = D_l D_l$, $\eta = 1/\mathrm{Re}$, $\varkappa = 1/(\mathrm{Re}\cdot\mathrm{Pr})$, $\zeta_1 = 0$, $\zeta_2 = \mathrm{Ri}$; where f and g are given sources of heat and external force. The dot over θ and v_k denotes partial differentiation with respect to time, while Re, Pr, and Ri are, respectively, the Reynolds, Prandtl, and Richardson numbers. The system includes the heat-transfer

*Translated from German into Russian by E. F. Nogotov.

Institute of Mechanics, Academy of Sciences of the German Democratic Republic, Berlin.
Translated from Inzhenerno-Fizicheskii Zhurnal, Vol. 49, No. 5, pp. 860-874, November, 1985. Original article submitted November 1, 1984.

0022-0841/85/4905-1364$09.50

equations (1), Navier—Stokes (for Ri = 0) (2), and continuity (3).

In numerical studies, Eqs. (2) and (3) are often transformed to the form

$$\dot{\omega} - \eta\Delta\omega + v_l D_l \omega = -\mathrm{Ri}\cdot D_1\theta + D_2 g_1 - D_1 g_2, \quad \omega = \Delta\Psi \tag{4}$$

or

$$\Delta\dot{\Psi} - \eta\Delta\Delta\Psi + v_l D_l \Delta\Psi = -\mathrm{Ri}\cdot D_1\theta + D_2 g_1 - D_1 g_2, \tag{5}$$

where Ψ is the stream function, $v_1 = D_2\Psi$; $v_2 = -D_1\Psi$, and $\omega = D_2 v_1 - D_1 v_2$ is the velocity vortex function. In this case, the continuity equation (3) is satisfied identically.

To calculate the pressure p it is necessary to supplement relationship (2). Applying it to the operator D_k, we obtain

$$D_k\dot{v}_k - \eta\Delta D_k v_k + v_l D_l D_k v_k + D_k v_l D_l v_k + \Delta p = \mathrm{Ri}\cdot D_2\theta + D_k g_k. \tag{6}$$

Here it is necessary to sum over the subscript k going from 1 to 2. Requiring, in addition, that the following condition be satisfied:

$$D_k v_l D_l v_k + \Delta p = \mathrm{Ri}\cdot D_2\theta + D_k g_k, \quad D_k v_k|_\Gamma = 0, \quad D_k v_k|_{t=0} = 0, \tag{7}$$

then, due to $D_k v_k - \eta\Delta D_k v_k + v_\ell D_\ell D_k v_k = 0$ in Ω in the whole region $\Omega \cup \Gamma$ for $t \geqq 0$, $D_k v_k = 0$. Therefore, the systems (2), (3) and (2), (7) become equivalent [1]. Furthermore, Eqs. (1), (2), and (7) are represented in the form of Boussinesq equations. The advantage of the latter is that a second-order equation is obtain for the pressure as well.

3. Initial and Boundary Conditions. The formulation of initial conditions is not difficult, since one can often start from a state of rest v = 0 at the moment of time t = 0, for which the temperature θ has a known distribution.

As boundary conditions for the heat-transport equations one can assign either the temperature (condition of the first kind), or its normal derivative (condition of the second kind), or a combination of these quantities (condition of the third kind). The velocity components are given for the equations mostly on the boundary. In this case, however, it is necessary to guarantee satisfaction of the vanishing condition of the integral of the normal velocity component along the boundary. In this case, the pressure p is uniquely determined accurately within a constant term. If the pressure values are given on some portion of the boundary Γ_p, then the boundary conditions for the velocity must vary in such a manner that the condition $D_k v_k|_{\Gamma p} = 0$ [1-3] be satisfied.

On the solid boundaries Γ_f the flow regions usually satisfy the adhesion condition $v|_{\Gamma f} = 0$. On the boundary portions where the liquid flow into the region or away from it, assigning boundary conditions for the velocity is more complicated and depends on the corresponding statement of the problem. Still more complicated is the statement of boundary conditions on the free surfaces and on the separation boundaries of phases on which boundary conditions are derived from the jump relationships.

The conditions at infinity in flow problems are usually obtained by asymptotic considerations. If symmetry lines are encountered in the flow region, then for computational regions one can consider only part of the original region, assigning on the symmetry lines vanishing conditions of the normal component and the normal derivative of the tangential velocity component, as well as the normal derivative of the temperature.

4. Approximation Region and Time Interval. For numerical solution of analytic problems with assigned initial and boundary conditions they must be approximated by algebraic equations. Truly, in isolated cases and when using Galerkin methods [4], for which the Fourier coefficients of the unknown functions are determined and closed contour integration is used, the approximation of the region Ω and its boundary Γ by discrete grid points is not necessary. In using difference methods and the finite element method, the original region is approximated by a system of discrete points. In this case, the smoothness of this approximation is substantially enhanced if the whole region is initially partitioned into elementary regions (elements), and then these elements are approximated by discrete points. The method of this partitioning depends substantially on the shape of the original region, and can be illustrated in detail on the example of a rectangle with sides parallel to the coordinate axes.

The rectangular region $\Omega\bigcup\Gamma$ is partitioned into a system of (I − 1) × (J − 1) rectangular elements R_{ij} (i = 1, ..., I − 1; j = 1, ..., J − 1), whose sides are also parallel to the coordinate axes. The vertices $x_{ij}=(x_{1i}, x_{2j})\in\Omega\bigcup\Gamma$ $(i=1, ..., I; j=1, ..., J)$ of these elements then form a system of $\Omega_h\bigcup\Gamma_h$ discrete points, by which is represented the region $\Omega_h\bigcup\Gamma_h$. The internal discrete points can be written in the form

$$\Omega_h=\{X_{ij}\,(i=2, \ldots, I-1; \quad j=2, \ldots, J-1)\},$$

and the boundary

$$\Gamma_h=\{X_{i1}\bigcup X_{iJ}\bigcup X_{1j}\bigcup X_{Ij} \quad (i=1, \ldots, I; \quad j=2, \ldots, J-1)\}.$$

In approximating boundary conditions of the second kind, an especially important role is played by the boundary band $\tilde{\Gamma}_h$:

$$\tilde{\Gamma}_h=\Gamma_h\bigcup\{X_{i2}\bigcup X_{iJ-1}\bigcup X_{2j}\bigcup X_{I-1j} \ (i=2, \ldots, I-1; j=3, \ldots, J-2)\},$$

which consists of points of the boundary Γ_h and internal adjacent boundary points.

For a uniform partitioning with a step H = (h_1, h_2) the points x_{ij} are represented in the form $x_{1i+1} = x_{1i} + h_1$, $x_{2j+1} = x_{2j} + h_2$. Here h_1 and h_2 are the magnitudes of the partitioning steps along the x_1 and x_2 axes. The point $X_{11} = (x_{11}, x_{21})$ is the left lower vertex of the region.

We also introduce a nonuniform partitioning with a step h_{1i} along x_1 and a step h_{2j} along x_2. The coordinates of the discrete points X_{ij} can be represented in the form

$$x_{1i+1}=x_{1i}+h_{1i}, \quad x_{2j+1}=x_{2j}+h_{2j} \ (i=1, \ldots, I-1; \ j=1, \ldots, \ J-1).$$

The same approach can also be used when the original region consists of several rectangular regions. If, however, the boundary Γ does not consist of segments parallel to the coordinate axes, partitioning of the region $\Omega\bigcup\Gamma$ into rectangular elements becomes complicated. In this case it is never possible to select a coordinate transformation for which the boundary Γ contains in the new coordinate system only segments parallel to the axes. A different, more general approach consists of the fact that the region $\Omega\bigcup\Gamma$ is decomposed into a finite set of (finite) elements E_i (i = 1, 2, ...) $(\Omega\bigcup\Gamma=\bigcup_i E_i)$ of arbitrary shape.

For regions having a polygon shape it is convenient to use elements of triangular shape. For a curvilinear edged region one introduces curvilinear triangular elements. For these partitions one can select the triangle vertices as the system of discrete points $\Omega_h\bigcup\Gamma_h=\{X_{ij}$ (i = 1, ..., i; j = 1, ..., J)} approximating the region $\Omega\bigcup\Gamma$. Similarly, one introduces the concept of the step magnitude H = (h_1, h_2) or, respectively, $H_{ij} = (h_{1i}, h_{2j})$.

For several problem statements it is meaningful to consider for the approximation region $\Omega^1\bigcup\Gamma^1$ not only constant, but also mobile grids [5] or a system of successive M grids $\Omega^m\bigcup\Gamma^m$ $(m=1, \ldots, M)$ [6]. In this case, $\Omega^1\bigcup\Gamma^1$ must be represented by a cruder grid, while $\Omega^{m+1}\bigcup\Gamma^{m+1}$ $(m=1, \ldots, M-1)$ – by a sequence of more refined ones. The sequence of grids can be selected by different methods. The simplest approach consists of that for a step value $H^{m+1}=(h_1^{m+1}, h_2^{m+1})$, or, respectively, $H_{ij}^{m+1}=(h_{1i}^{m+1}, h_{2j}^{m+1})$ for some grid $\Omega^{m+1}\bigcup\Gamma^{m+1}$ we obtain from the step $H^m=(h_1^m, h_2^m)$, or, respectively, $H_{ij}^m=(h_{1i}^m, h_{2j}^m)$, grids $\Omega^m\bigcup\Gamma^m$ by halving: $H^{m+1}=H^m/2$ or, respectively, $H_{ij}^{m+1}=H_{ij}^m/2$.

The time interval [0, T] of the (N + 1)-th discrete point $t_n\in[0, T]$ $(n=0, 1, ..., N; t_0=0$, t_N = T) is similarly approximated. For the uniform partitioning with step τ we have the ratio $t_{n+1} = t_n + \tau$ (n = 0, 1, ..., N − 1), while for the nonuniform partitioning with step τ_n we have $t_{n+1} = t_n + \tau_n$.

5. Approximation Functions. The simplest method of approximating the function $f(x)\,(x\in\Omega\bigcup\Gamma)$ by the grid functions $f_h(X_{ij})$ $(X_{ij}\in\Omega_h\bigcup\Gamma_h)$ is given by the equation

$$f_h(X_{ij})=f(X_{ij}). \tag{8}$$

A different possibility of representing the analytic function f by means of a finite number of values $\bar{f} = \bar{f}_{ij}$ consists of selecting for the latter weighted mean values of f or moments

$$\bar{f}_{ij}=(f, \ \psi_{ij})/(1, \ \psi_{ij}). \tag{9}$$

Here $(f, \psi) = \int\limits_{\Omega} f\psi dx$ and $\psi_{ij} = \psi_{ij}(x)$; $x \in \Omega \bigcup \Gamma$ is a known weight function, vanishing for all $x \notin U_{ij}$, where $U_{ij} \subseteq \Omega \bigcup \Gamma$ is a selected neighborhood of the_point X_{ij}. In isolated cases it can coincide with the region $\Omega \bigcup \Gamma$. These averaged values $\bar{f}_{ij}$ depend not only on the original function f, but also on the neighborhood U_{ij} and the weight function ψ_{ij}, and can differ strongly from the reference values $f(X_{ij})$.

Another approach to approximation by an analytic function f consists of approximating it by other functions $f_I = f_I(x)$, defined in terms of a finite set of coefficients $a_{\mu\nu}$ ($\mu = 1, \ldots, I$; $\nu = 1, \ldots, J$):

$$f_I = \sum_{\mu,\, \nu} a_{\mu\nu} \varphi_{\mu\nu}. \tag{10}$$

Here $\varphi_{\mu\nu} = \varphi_{\mu\nu}(x)$ is a system of known functions of coordinates, while the subscripts μ and ν imply summation from 1 to I and from 1 to J, respectively. The coefficients $a_{\mu\nu}$ are determined either from the system of equations

$$\sum_{\mu,\, \nu} a_{\mu\nu} \varphi_{\mu\nu}(X_{ij}) = f(X_{ij}), \quad i = 1, \ldots, I; \quad j = 1, \ldots, J, \tag{11}$$

or

$$\sum_{\mu,\, \nu} a_{\mu\nu} (\varphi_{\mu,\, \nu}, \ \psi_{ij}) = (f, \ \psi_{ij}), \quad i = 1, \ldots, I; \quad j = 1, \ldots, J. \tag{12}$$

Equation (10) is an interpolation polynomial, which in the case (11) has the same values at the points X_{ij}, while in the case (12) these are the same weighted mean values as for the function f. For unique determination of the coefficients $a_{\mu\nu}$ it is necessary and sufficient that the determinant of the matrix $\varphi_{\mu\nu}(x_{ij})$, or, respectively, of the matrix $(\varphi_{\mu\nu}, \psi_{ij})$, be nonvanishing.

If the functions $\varphi_{\mu\nu}$ and ψ_{ij} are orthonormal, then it follows from Eq. (12) that

$$a_{\mu\nu} = (f, \ \psi_{\mu\nu}). \tag{13}$$

If this scalar product (integral) is expressed by means of the quadrature equation

$$\int\limits_{\Omega} f dx = \sum_{i,\, j} \gamma_{ij} f(X_{ij}),$$

where γ_{ij} are selected coefficients of the corresponding problem, substituted into Eq. (13), then

$$a_{\mu\nu} = \sum_{i,\, j} \gamma_{ij} f(x_{ij}) \psi_{\mu\nu}(x_{ij}).$$

Substituting this expression further into Eq. (10), we obtain

$$f_I = \sum_{i,\, j} f(X_{ij}) \Phi_{ij}, \tag{14}$$

where $\Phi_{ij} = \Phi_{ij}(x) = \gamma_{ij} \sum\limits_{\mu,\, \nu} \varphi_{\mu\nu}(x) \psi_{\mu\nu}(X_{ij})$.

Expression (14) is an interpolation polynomial, in which the coefficients are the reference values of the original function f. If the coordinate and weight functions $\varphi_{\mu\nu}$ and $\psi_{\mu\nu}$ or the function Φ_{ij} are selected in such a manner that

$$\Phi_{ij}(x) \quad \begin{cases} 1 & \text{for} \quad x = X_{ij}, \\ 0 & \text{for} \quad x \in \Omega_h \bigcup \Gamma_h - X_{ij}, \end{cases}$$

we then obtain from Eq. (14) for f_1 at the point X_{ij} the same approximation as in Eq. (8).

Expression (14) plays the same important role in the finite element method, as does Eq. (8) in the finite difference method, or expression (10) in the Galerkin method. In using Eq. (10) as a system of coordinate functions one usually selects either systems of trigonometric functions or polynomials, which must be determined in the whole region Ω, and must be sufficiently smooth.

In using Eq. (14) one uses for Φ_{ij} predominantly piecewise-continuous low-order polynomials. In this case one often uses spline functions, nonvanishing only near X_{ij}; they are equal to unity at the point X_{ij}, while they vanish at all remaining grid points.

Choosing around the point X_{ij} some neighborhood U_{ij} consisting of four quadrilateral elements from E_1 to E_4 (see Fig. 1a), we then obtain, for example for the function Φ_{ij} of some internal grid site $X_{ij} \in \Omega_h$ the following expression:

$$\Phi_{ij}(x) \begin{cases} F_i^1(x) F_j^2(x), & x \in E_1, \\ F_j^2(x) F_i^3(x), & x \in E_2, \\ F_i^3(x) F_j^4(x), & x \in E_3, \\ F_i^1(x) F_j^4(x), & x \in E_4, \\ 0 & x \in \Omega \cup \Gamma - U_{ij}. \end{cases}$$

Here $F_i^1(x) = -(x_1 - x_{1i+1})/h_{1i}$; $F_j^2(x) = -(x_2 - x_{2j+1})/h_{2j}$; $F_i^3(x) = (x_1 - x_{1i-1})/h_{1i-1}$; $F_j^4(x) = (x_2 - x_{2j-1})/h_{2j-1}$.

These functions are also called shape functions. For these functions expression (14) acquires, for example in the element E_1, the following shape:

$$f_I(x) = f(X_{ij}) F_i^1(x) F_j^2(x) + f(X_{i+1j}) F_j^2(x) F_{i+1}^3(x) + f(X_{i+1\,j+1}) F_{i+1}^3(x) F_{j+1}^4(x) + f(X_{ij+1}) F_i^1(x) F_{j+1}^4(x).$$

The method indicated can also be extended to elements of triangular shape. We select a neighborhood U_{ij} of some point X_{ij}, consisting of 6 elements from E_1 to E_6 (see Fig. 1b). We then obtain the following expression, for example, for the function Φ_{ij} of some internal grid site X_{ij} and, consequently, of the corresponding triangular element:

$$\Phi_{ij}(x) = \begin{cases} F_i^1(x), & x \in E_1, \\ F_j^2(x), & x \in E_2, \\ F_i^3(x) - F_j^2(x), & x \in E_3, \\ F_i^3(x), & x \in E_4, \\ F_j^4(x), & x \in E_5, \\ F_j^4(x) - F_i^1(x), & x \in E_6, \\ 0 & x \in \Omega \cup \Gamma - U_{ij}. \end{cases}$$

For the thus selected shape function expression (14) acquires, for example, the following shape in the element E_1:

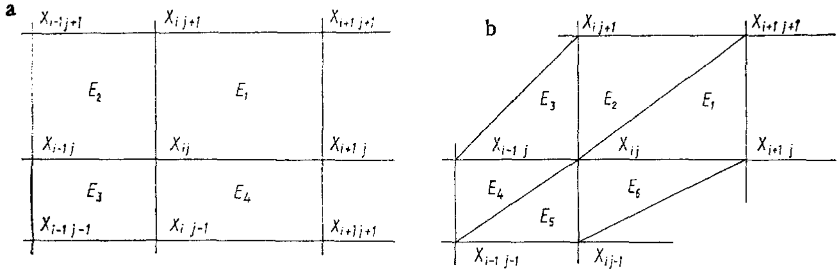

Fig. 1. Neighborhood $U_{ij} = E_1 \cup \ldots \cup E_4$ (a) and $U_{ij} = E_1 \cup \ldots \cup E_6$ (b) of the point X_{ij}.

$$f_I(x) = f(X_{ij}) F_i^1(x) + f(X_{i+1j}) [F_{i+1}^3(x) - F_j^2(x)] + f(X_{i+1\,j+1}) F_{j+1}^4(x).$$

In the other elements we obtain similar expressions for $f_I(x)$.

For the boundary sites $X_{ij} \in \Gamma_h$ we use the corresponding single-sided expressions for Φ_{ij}. This method can be generalized to arbitrarily located triangles, as well as curvilinear triangular elements [7]. In a more general approach and a choice of several grid points in the triangle one requires a higher degree of smoothness of the coordinate function. The choice of coordinate and weight functions is related to the specific features of the problem to be solved. This also equally refers to their smoothness properties, as well as to approximation equations and boundary conditions.

6. Approximation of Differential Equations. Similarly to the discussion above, we also approximate differential operators. We introduce the notation:

$$f = f(X_{ij},\ t_n), \quad \check{f} = f(X_{ij},\ t_{n-1}),$$
$$\overset{\pm 1}{f} = f(X_{i\pm 1,\,j},\ t_n), \quad \overset{\pm 2}{f} = f(X_{i,\ j\pm 1},\ t_n).$$

Then using the Taylor equation at the point X_{ij} and at the moment of time t_n, one can obtain the following approximations (app) [8, 9]:

$$\text{app}\,(\dot{f}) = f_{\bar{t}}, \quad \text{app}\,(D_l f) = f_{x_l}, \quad f_{\bar{x}_l} \text{ или } f_{\overset{\circ}{x}_l},$$
$$\text{app}\,(D_l^2 f) = f_{\bar{x}_l x_l},$$

where $f_{\bar{t}} = (f - \check{f})/\tau_{n-1}$; $f_{x_1} = (\overset{+1}{f} - f)/h_{1i}$; $f_{x_2} = (\overset{+2}{f} - f)/h_{2j}$; $f_{\bar{x}_1} = (f - \overset{-1}{f})/h_{1i-1}$; $f_{\bar{x}_2} = (f - \overset{-2}{f})/h_{2j-1}$; $f_{\overset{\circ}{x}_l} = 0.5\,(f_{x_l} + f_{\bar{x}_l})$; $f_{\bar{x}_1 x_1} = 2\,(f_{x_1} - f_{\bar{x}_1})/(h_{1i-1} + h_{1i})$; $f_{\bar{x}_2 x_2} = 2\,(f_{x_2} - f_{\bar{x}_2})/(h_{2j-1} + h_{2j})$.

The approximation errors are estimated by means of the values of grid parameters [8, 9]. If the differential operator D_ℓ is encountered as a cofactor in the product, one can use the expressions [2, 10]

$$\begin{aligned} \text{app}\,(v_l D_l w) &= 0.5\,(\overset{+l}{v_l} w_{x_l} + v_l w_{\bar{x}_l}); \\ \text{app}\,(v_l D_l w) &= 0.25\,[(\overset{+l}{v_l} + v_l)\, w_{x_l} + (v_l + \overset{-l}{v_l})\, w_{\bar{x}_l}] \end{aligned} \tag{15}$$

or

$$\text{app}\,(v_l D_l w) = 0.5\,[(v_l - |v_l|)\, w_{x_l} + (v_l + |v_l|)\, w_{\bar{x}_l}].$$

If the starting expressions, such as the convective terms $v_\ell D_\ell v_k$, are nonlinear, then, according to Eq. (15), the expressions approximating them will also be nonlinear. One can, however, also introduce approximations for which linearization is simultaneously realized, for example, app $(v_\ell D_\ell v_k)$ = app $(\check{v}_\ell D_\ell v_k)$.

Another possibility of avoiding nonlinearity consists of nonlinear expressions being approximated explicitly, for example, app $(v_\ell D_\ell v_k)$ = app $(\check{v}_\ell D_\ell \check{v}_k)$, or an iteration approximation is realized of nonlinear terms:

$$\text{app}\,(v_l D_l v_k) = \text{app}\,(\tilde{v}_l D_l v_k), \quad \text{app}\,(v_l D_l \tilde{v}_k), \quad \text{app}\,(\tilde{v}_l D_l \tilde{v}_k).$$

Here $\tilde{v}_\ell$, and correspondingly $\tilde{v}_k$, denote initial or approximate values, refined by the iteration.

In the numerical solution of nonstationary initial problems, the linearization method app $(v_\ell D_\ell v_k)$ = app $(\check{v}_\ell D_\ell v_k)$ seems to be more advantageous, since it leads to faster convergence of the solution than for explicit approximations, and to higher stability of the computational process in comparison with iteration methods.

If, however, the analytic functions are approximated by replacing their sums of the form (10) or (14), then the differential operators $D = (D_1, D_2, D_1^2, D_2^2)$ can be applied directly to the coordinate functions $\varphi_{\mu\nu}(x)$ or, respectively, to $\Phi_{\mu\nu}(x)$, and be considered at the point x_{ij}, more precisely

$$\mathrm{app}\,(Df)|_{X_{ij}} = D\,\mathrm{app}\,(f)|_{X_{ij}} = Df_I\,|_{X_{ij}} = \sum_{\mu,\nu} a_{\mu\nu} D\varphi_{\mu\nu}(X_{ij}) \text{ or } \sum_{\mu,\nu} f(X_{\mu\nu})\,D\Phi_{\mu\nu}(X_{ij}).$$

In this case, when these differential expressions must be approximated not at the discrete points X_{ij}, but expressed in terms of weighted means corresponding to Eq. (9):

$$\mathrm{app}\,(Df) = \overline{Df}_{ij} = (Df,\ \psi_{ij})/(1,\ \psi_{ij}),$$

while weight functions ψ_{ij} may be augmented so as to approximate the differential expressions. This approach within the Galerkin method and that of weighted residues (for example the finite element method), has recently acquired much interest.

7. Approximation of Problems with Initial and Boundary Conditions. The approximation of problems with boundary and initial conditions can be conducted by various methods. Within the difference approximation, for example, we construct the following approximation [2, 10] for the system (1)-(3):

$$\theta/\tau_{n-1} - \varkappa\theta_{\bar{x}_l x_l} + \mathrm{app}\,(\check{v}_l D_l \theta) = f + \check{\theta}/\tau_{n-1} \text{ in } \Omega_h, \tag{16}$$

$$v_k/\tau_{n-1} - \eta v_{k\bar{x}_l x_l} + \mathrm{app}\,(\check{v}_l D_l v_k) + p_{\bar{x}_k} = \zeta_k\theta + g_k + \check{v}_k/\tau_{n-1} \text{ in } \Omega_h, \tag{17}$$

$$v_{lx_l} = 0 \text{ in } \Omega_h \cup \Gamma_h' - X^*. \tag{18}$$

Here k = 1, 2; n = 1, ..., N, and Γ'_h denotes the points of the boundary Γ_h, in which must be determined the pressure value p, and X^* is an arbitrary point at $\Omega_h \cup \Gamma_h'$, in which Eq. (18) is not considered, since in the opposite case the system of equations will be linearly dependent [2].

Since by Eqs. (16)-(18) the pressure p is determined accurately only within a constant term, its value must be given at one arbitrary point $X^{**} \in \Omega_h \cup \Gamma'_h$. Without loss of generality one can select as X^{**} a point on the boundary Γ'_h and put $X^* = X^{**}$.

For the approximation app $(\check{v}_\ell D_\ell \theta)$ and, correspondingly, app $(\check{v}_\ell D_\ell v_k)$, one can use, for example, approximation (15). Due to this linear approximation of the nonlinear convective terms the system of equations (1)-(3) is decomposed into two parts, having only single-sided coupling. More precisely, initially Eq. (16) is solved and θ is determined, and then Eqs. (17) and (18) are used to determine v and p.

The initial and boundary conditions must also be approximated so as to solve these equations. So as not to have supplementary conditions for the points outside the grid $\Omega_h \cup \Gamma_h$, to approximate the derivatives appearing in the boundary conditions one often uses one-sided differences.

In approximating Eqs. (1)-(3) one can achieve higher levels of accuracy than in Eqs. (16)-(18) [10]. For this the terms having a large effect on the solution must be approximated with high accuracy. In the case of small κ and η values, a decisive role is played by the approximation of convective terms [11].

Similarly, by the difference approximations, Eqs. (4) and (5), we obtain

$$\begin{gathered}\omega/\tau_{n-1} - \eta\omega_{\bar{x}_l x_l} + \mathrm{app}\,(\check{v}_l D_l \omega) = -\mathrm{Ri}\theta_{\overset{\circ}{x}_1} + g_{1\overset{\circ}{x}_2} + g_{2\overset{\circ}{x}_1} + \check{\omega}/\tau_{n-1},\\ \omega = \Psi_{\bar{x}_k x_k},\end{gathered} \tag{19}$$

$$\Psi_{\bar{x}_k x_k}/\tau_{n-1} - \eta\Psi_{\bar{x}_k x_k \bar{x}_l x_l} + \mathrm{app}\,(\check{v}_l D_l \Psi_{\bar{x}_k x_k}) = -\mathrm{Ri}\,\theta_{\overset{\circ}{x}_1} + g_{1\overset{\circ}{x}_2} - g_{2\overset{\circ}{x}_1} + \check{\Psi}_{\bar{x}_k x_k}/\tau_{n-1}. \tag{20}$$

The approximation v = $(D_2\Psi, -D_1\Psi)$ can be realized in the form $v = (\Psi_{\overset{\circ}{x}_2}, -\Psi_{\overset{\circ}{x}_1})$ or $v = (D_2\Psi, -D_1\Psi)$. If on Γ_h one uses single-sided differences for calculating Ψ from the boundary values of v, then one can determine Ψ from a segment of the boundary $\tilde{\Gamma}_h$, consisting of the points $\Gamma_h \cup (\Omega_h \cap (\overset{\pm 1}{\Gamma}_h \cup \overset{\pm 2}{\Gamma}_h))$. To calculate ω on Ωh and Ψ on Ωh − $\tilde{\Gamma}$h, it is sufficient to consider the first equation (19) at the point $X_{ij} \in \Omega_h \text{--} \tilde{\Gamma}_h$ and the second equation (19) at the points $X_{ij} \in \Omega_h$ Equations (19) can be considered independently of each other. But then the required boundary values for ω can only be calculated within the iteration approximation, since ω is related to Φ by the second Eq. (19).

It is sufficient to consider Eq. (20) at the points $X_{ij} = \Omega_h - \tilde{\Gamma}_h$, and to determine Ψ on $\tilde{\Gamma}_h$ from the boundary conditions for v. Equations (19) and (20) have the advantage that the pressure p is eliminated in them and, therefore, they require substantially less computer time than Eqs. (17) and (18).

Similarly one can make approximations to Eq. (7):

$$\begin{gathered} p_{\overline{x}_k x_k} + \check{v}_{l \overset{0}{x}_k} v_{k \overset{0}{x}_l} = \mathrm{Ri}\, \theta_{\overset{0}{x}_2} + g_{k \overset{0}{x}_k} \text{ in } \Omega_h, \\ \operatorname{app}(D_k v_k) = 0 \ \text{ in } \Gamma_h. \end{gathered} \tag{21}$$

Here app ($D_k v_k$) on Γh is a one-sided difference approximation. Equation (21) can be solved simultaneously with (17) instead of (18). For this are needed boundary conditions for the pressure, which must be determined in such a manner that the operator app ($D_k v_k$) vanish at the boundary Γh. Therefore, in the system (17), (21) there is a connection between p and v for the boundary values of p as well. In this statement, the problem with explicit approximation of convective terms was solved numerically by means of the direct method [3].

In this case, starting from given boundary conditions for the pressure and Eq. (21), the pressure was determined, and from Eq. (17) was found the velocity, with boundary conditions for v. Since the velocity values thus obtained do not satisfy the second equation (21), corrections were calculated for the pressure at the boundary Γh, and the problem was solved at the second step.

Along with the difference approximations described here for the equations of heat transfer and of motion there exist many other approximations, obtained, for example, for different grids for the unknown functions and explicit approximations for the equations of motions [5], as well as in using splitting or iteration methods [12, 13]. The most effective among them are implicit approximations, leading to a system of algebraic equations with tridiagonal matrices. To shorten the computation time and decrease the number of iteration steps in the approximation equations one introduces iteration parameters, and methods of automatic selection of their optimal values have been developed [12]. Often used is a method in which, along with the approximation equations (2) and (3) one carries out their iterational self-consistency [14-16]. In this approach one usually starts from known initial or approximate values of the pressure $\tilde{p}$, which are substituted into Eq. (17), and v is determined from it. The $\tilde{p}$ value is then refined by means of the approximate relation

$$p = \tilde{p} - \sigma v_{l x_l}.$$

This refinement process of the p value is repeated until satisfactory accuracy is reached. Here σ is a small parameter, given in such a manner that the iteration process converges [14].

To enhance the stability of the computational process one can introduce in the approximation equations wider grid patterns or use the predictor-corrector method. A simple method, for which, in particular, there occurs smoothing of oscillations of the solution with respect to the time coordinate, consists of that among the solutions at the two latter time planes one constructs the weighted mean with weighting coefficient β (0 ≦ β ≦ 1): $\bar{W} = \beta W + (1 - \beta)\check{W}$, where W = (θ, v_1, v_2, p, Ψ, ω) [17].

Similar systems of algebraic equations have also been obtained for finite-element approximation equations, as well as for Galerkin approximations. In using these methods the derivative with respect to time is also replaced by a difference ratio, and the dependences of the unknown functions on spatial coordinates are approximated on the time planes t_n by representations of the form:

$$\mathrm{app}\,(\theta) = \sum_{\mu,\,\nu} z_{\mu\nu}\varphi_{\theta\mu\nu}, \quad \mathrm{app}\,(v_k) = \sum_{\mu,\,\nu} u_{k\mu\nu}\varphi_{k\mu\nu},$$

$$\mathrm{app}\,(p) = \sum_{\mu,\,\nu} q_{\mu\nu}\varphi_{p\mu\nu}$$

or

$$\mathrm{app}\,(\theta) = \sum_{\mu,\,\nu} \theta\,(X_{\mu\nu})\,\Phi_{\theta\mu\nu}, \quad \mathrm{app}\,(v_k) = \sum_{\mu,\,\nu} v_k\,(X_{\mu\nu})\,\Phi_{k\mu\nu},$$

$$\mathrm{app}\,(p) = \sum_{\mu,\,\nu} p\,(X_{\mu\nu})\,\Phi_{p\mu\nu}.$$

Here $z_{\mu\nu}$, $u_{k_{\mu\nu}}$, $q_{\mu\nu}$ are unknown coefficients, $\theta(X_{\mu\nu})$, $v_k(X_{\mu\nu})$, $p(X_{\mu\nu})$ are unknown reference values, and $\varphi_{\theta\mu\nu}$, $\varphi_{k\mu\nu}$, $\varphi_{p\mu\nu}$, $\Phi_{\theta\mu\nu}$, $\Phi_{k\mu\nu}$, $\Phi_{p\mu\nu}$ is a known system of coordinate functions, depending on x. After substituting these representations into the original equations (1)-(3), the latter are multiplied by appropriately selected weight functions $\psi_{\theta ij}$, $\psi_{k\lambda ij}$, and ψ_{pij}, and are integrated in the Ω region. In this case, λ acquires the values 1 and 2, the summation over k is carried out from 1 to 2, and i and j vary from 1 to I − 1 and J − 1, respectively. In this case the systems of algebraic equations for the unknown coefficients and, respectively for the reference values, are more complicated in comparison with those obtained in using finite-difference methods. As a rule, they have matrices with a wider band. To determine these matrices, the integrals must often be calculated numerically. In using Galerkin methods, the further problem is generated of determining the coordinate and weight functions occurring for complicated regions. The advantage of Galerkin methods consists of their higher accuracy and of their more convenient application in mathematical studies of the equations.

The advantage of the finite element method is its good adaptation to regions of complex shape, as well as to complicated transfer processes. Also, in individual special cases it can be simplified substantially. Thus, for example, if weight functions can be used possessing the property $\psi_{k\lambda ij} = 0$ on Γ and $D_{k_\psi}{}_{k\lambda ij} = 0$ on Ω, then due to $(D_k p,\ \psi_{k\lambda ij}) - (p,\ D_k\psi_{k\lambda ij}) = 0$ one can eliminate the pressure from the equations of motion.

8. Solution of Systems of Linear Algebraic Equations. As was shown in the preceding sections, there exist many methods of approximating the original equations of a system of linear algebraic equations $Ay = b$. Here $A = \{a_{\mu\nu}\}$ (μ, ν = 1, ..., K) is a square matrix of order K, and $y = \{y_\nu\}$ and $b = \{b_\mu\}$ are unknown and known vectors. The system of equations must be selected in such a manner that $a_{\mu\mu} \neq 0$. In this case it can be solved directly or by iterations. It must be noted that iteration methods usually require longer computation time, while direct methods require more memory. The shape of the system of algebraic equations depends substantially on the approximation method. Using explicit [5], partially implicit [14] difference schemes, or splitting methods [12, 13], we respectively obtain systems of equations, whose matrices have nonvanishing elements only in the main diagonal, or reduce to tridiagonal form. Such systems of equations can be solved by known direct methods. When the matrix A or part of it is obtained by five-point approximations of the Laplace operator, for direct solution of this system one can use the method of cyclic reduction [18], the method of fast Fourier transforms [19], or the stable itinerary method of error minimization [20]. Their advantages and disadvantages have been investigated in detail in [21].

For direct solution of an arbitrary system of linear algebraic equations one can use, in principle, the Gauss method. For economic matrix arrangement in a computer, however, it is ineffective. Therefore, more effective special methods of solution [7] have been developed for strip matrices. The LU-factorization method is well recommended for arbitrarily weakly filled matrices, in which case the problem reduces to sequential solution of two systems of equations with upper and lower triangular matrices.

Among the simplest iteration methods are the Jacobi method

$$a_{\mu\mu}\,(y_\mu - \tilde{y}_\mu) = b_\mu - \sum_{\nu=1}^{K} a_{\mu\nu}\tilde{y}_\nu$$

and the Gauss—Seidel method

$$\sum_{\nu=1}^{\mu} a_{\mu\nu}(y_\nu - \tilde{y}_\nu) = b_\mu - \sum_{\nu=1}^{K} a_{\mu\nu}\tilde{y}_\nu.$$

Here y_ν is the unknown solution, and $\tilde{y}_\nu$ is its known initial approximation.

After y_ν has been determined, it can be substituted for $\bar{y}_\nu$, and the calculation can be repeated until satisfactory accuracy has been reached [22]. For a suitable choice of the iteration parameter σ one can use a more effective point relaxation method [22]

$$a_{\mu\mu}(y_\mu - \tilde{y}_\mu) = \sigma\left(b_\mu - \sum_{\nu=1}^{K} a_{\mu\nu}\tilde{y}_\nu\right)$$

or a quickly convergent implicit relaxation method [23], for which a dominating submatrix $C = \{c_{\mu\nu}\}$ is separated from the algebraic equations, and the iteration process is constructed as follows:

$$\sum_{\nu=1}^{K} c_{\mu\nu}(y_\nu - \tilde{y}_\nu) = \sigma\left(b_\mu - \sum_{\nu=1}^{K} a_{\mu\nu}\tilde{y}_\nu\right).$$

In this case, C is selected in such a manner that it is easily inverted. At each iteration step one can then use direct solution methods. If C has a tridiagonal shape, one can usually apply a dismissal method, and if C is a Laplace operator, approximated by a five-point pattern, iteration methods are used.

A different, more general method of solving systems of algebraic equations on the basis of an implicit iteration process consists of an incomplete LU-factorization of the matrix A. In this case, C is represented in the form of a product of lower and upper triangular matrices in such a manner that the error $\|A-C\|$ be possibly small, and that the factors of the product C occupy lease computer memory space. The choice of the solution method depends on the structure and value of the original system of equations. For transport problems of heat, momentum, and mass in a two-dimensional system, and for a not too large number of equilibrium grid sites, it is recommended to use LU-factorization methods and fully implicit approximations of the original equations. These are well recommended both for heat-transfer equations [24], and for the Navier–Stokes [11] and Boussinesq [17] equations. In this case one uses special programming packages [25], containing a large number of subprograms which can be applied to different variants of the solution.

Since for the statements of nonstationary problems considered here all equations at all time layers always have an identical structure, it seemed possible to use for their solution a single program consisting of three parts. The first part leads to symbolic factorization, which for a large number of equations having identical structure can be satisfied only once. The second carries out the numerical factorization, and solves the system of equations. And, finally, the third part is a program for obtaining partial solutions, which is always used if the matrix is represented in the form of a product of lower and upper matrices.

Thus, if it is necessary to solve a large number of systems of equations, whose matrices remain invariant with only the right-hand sides changing, the first two parts of the program are used only for the first system. To decrease the required bulk of computer memory and enhance the stability of the calculation, there exists a program in the package which undertakes an advisable matrix redistribution ahead of its factorization.

To solve the heat-transfer equation (16) in a rectangular region on a 33 × 33 grid, one needs 15 sec on an ES1055 computer. If the matrix remains unchanged, but only the boundary conditions and the right-hand sides of Eq. (16) change, and the heat-conduction coefficient κ and velocity v are independent of time t, then to solve it at the second and each further time step one requires only 0.8 sec of computer time. If, however, the matrix elements vary from one time step to another, while the structure of the matrix is retained, then to calculate the matrix and the right-hand sides and solve the equations at the second and each further time step one requires 8 sec.

In solving the Navier–Stokes equation applied to flow in a rectangular band (a 25 × 25 grid), the Ψ-representation (20) was selected. The computation time at each time layer was 14.2 sec for the second and each subsequent time step. The total memory bulk of the program was 300 kbytes. Along with the solution of the Navier–Stokes equation in the Ψ-representation we also verified the ωΨ-representation (19) and the vp-representation (17) and (18). This verification has shown that the ωΨ-representation approximately agrees with the Ψ-representation both in computation time and in the bulk of memory required. For the numerical solution of the Navier–Stokes equation in the vp-representation the computation time and the bulk of memory required increased by approximately 4 times.

These methods were also used in solving the Boussinesq equation for numerical studies of free-convective flows in rectangular shape regions. In this case we used equations of motion in the Ψ-representation, and considered a 17 × 17 grid. For direct solution of the linearized Boussinesq equations at the second and each subsequent time layer, the computation machine time was 4.5 sec. The total length of the program was 162 kbytes.

To decrease the approximation error, for large Reynolds numbers it is required to use a very fine grid. This leads for a full LU-factorization matrix to the necessity of using a large memory, which sometimes exceeds the present computational possibilities. Due to the development of many-point methods, one can develop effective methods even for large systems of equations. It was also used for solving the Navier–Stokes and Boussinesq equations [6], and consists of the fact that in solving the equations on crude grids one uses direct methods, while on fine ones one uses iteration or smoothing methods, requiring less memory. These multipoint methods lead for a time layer t_n and grid space Ω^m to a system of algebraic equations of the shape $A^m y^m = b^m$, m = 1, ..., M. For the nonlinearized approximation of the original equations the matrix A^m also depends on y^m, i.e., $A^m = A^m(y)$. The many-point methods can be used in different shapes [6].

One of the solution methods of the nonlinear nonstationary systems of equations [26] consists of the fact that for some time layer t_n one starts from a known approximate solution $\bar{y}^M$ on the grid Ω^M, and calculates an initial solution $\tilde{y}^1 = I_M^1 \bar{y}^M$ on the grid Ω^1. Here one introduces the transition operator $I_{m_1}^{m_2}$ from the grid Ω^{m_1} to the grid Ω^{m_2}. This is an identification, prolongation, or restriction operator, respectively, for $m_1 = m_2$, $m_2 > m_1$ or $m_2 < m_1$. At the first stage of the n-th time layer y^M one can determine the extrapolation solution at the preceding time layer [26] and find the error $R^M = b^M - A^M(\bar{y})\bar{y}^M$ on Ω^M. By further restricting the step, one determines the error projection $R^m = I^m{}_M R^M$ on Ω^m (m = M − 1, ..., 1) and the approximate solution $\bar{y}^M$ is iteratively improved by solving the system of equations

$$A^m(y)\,y^m = H^m, \quad m = 1, \ldots, M, \tag{22}$$

where $H^m = A^m(\tilde{y})\,\tilde{y}^m + R^m$, $\tilde{y}^1 = I_M^1 \bar{y}^M$, $\tilde{y}^m = I_M^m \bar{y}^M + I_{m-1}^m (y^{m-1} - I_M^{m-1} \bar{y}^M)$, $m = 2, \ldots, M$.

The solution obtained for m = M is used following the new approximation $\bar{y}^M = y^M$, and the process is repeated until satisfactory accuracy (smoothness) is reached. Instead of the approximate solution obtained on a fine grid, one can also start from any solution obtained on a crude grid, and select different M values at separate time layers.

In the case of approximation (20), Eq. (22) for m = 1 was solved by the direct method by means of the LU-factorization matrix [27] $A_1(\tilde{y})y^1 = H^1$. On the fine grids Ω_m (m > 1) we used Jacobi and Gauss–Seidel relaxations, as well as an incomplete LU-factorization matrix with a damping factor σ, i.e., for some constant m > 1 we use for the smoothed solution y^m the following iteration process:

$$C^m(y^m - \tilde{\tilde{y}}^m) = \sigma\,[H^m - A^m(\tilde{\tilde{y}})\,\tilde{\tilde{y}}^m]. \tag{23}$$

At the first iteration step we took $\tilde{\tilde{y}}^m = \tilde{y}^m$, and at the further iteration steps $\tilde{\tilde{y}}^m$ was replaced by the solution y^m of Eq. (23).

In the case of Jacobi relaxation the nonvanishing elements of the matrix C^m coincided with the principal diagonal of the matrix $A^m(\tilde{y})$. In the case of Gauss–Seidel relaxation the lower triangular matrix and the principal diagonal of the matrix C^m coincided, respectively, with the lower triangular matrix and the principal diagonal of the matrix $A^m(\tilde{y})$ in the form of a product, whose lower and upper triangular matrices have a nonvanishing structure, as well as $A^m(\tilde{y})$ [6].This smoothing method was verified both with a constant parameter σ, approximately equal to unity, and with a dynamically controlled parameter σ. This parameter was controlled in such a manner that the norm of the new error equation (23') be smaller than the norm of the error obtained for $\tilde{y}^m$. In solving the Navier–Stokes equation (2) by the method indicated for a Reynolds number Re = 5000 in the case of flow in a closed rectangular band, we used three grids: 19 × 19, 35 × 35, and 67 × 67. The parameter σ was selected to be constant. In this case we observed in the transition from one grid to another the appearance of disturbances leading to instabilities of the computational process with an enhanced time step.

For dynamic selection of the iteration parameter σ we obtained for it relatively small values, leading to an insignificant change in the linearly interpolated solutions obtained for crude grids.

This behavior of the equations for the many-grid approximations is, obviously, related to their sensitivity to disturbances due to the selection of the approximation method of the Navier–Stokes equations and their boundary conditions. Since the boundary values for Ψ^m on $\Gamma^m - \Gamma^m$ are proportional to the value of the grid step for $\Psi^m = 0$ on Γ^m, a change in the latter leads to a change in them and, consequently, to a change of Ψ^m on Ω^m.

9. Conclusion. To solve special classes of stationary and nonstationary two-dimensional problems of momentum, heat, and mass transfer, we developed various packages of applied programs which, after the introduction of specific data, corresponding to a specific statement of the problem, and the development of their program provide its numerical solution. The use of fully implicit approximation methods and of direct solution of the linearized equations provides the possibility of developing effective numerical algorithms, making it possible to obtain the solution in a wide region of Reynolds and Rayleigh numbers.

These methods can also be used for numerical investigation of transport processes in turbulent flows and in rheological media. In principle, they can be generalized to solving three-dimensional problems. In this case, however, for a small value of the grid step the number of equations becomes very large, and high-power computers are needed to realize these methods.

LITERATURE CITED

1. L. Kleiser and U. Schumann, "Spectral simulations of the laminar-turbulent transition processes in plane Poiseuille flow," Proc. Symp. Spectral Methods Partial Diff. Eq., Aug. 1982, SIAM, Philadelphia (1984), pp. 141-163.
2. O. A. Ladyzhenskaya, Mathematical Theory of Viscous Incompressible Flow, Gordan and Breach (1969).
3. U. Schumann, "Direct finite difference Stokes solver," ZAMM, No. 64, 227-229 (1984).
4. C. A. J. Fletcher, Computational Galerkin Methods, Springer-Verlag, New York (1984).
5. A. A. Amsden and F. H. Harlow, "The SMAC method, a numerical technique for calculating incompressible fluid flow," Los Alamos Scientific Laboratory, Report LA 4370 (1970).
6. W. Hackbush, "Introduction to multigrid methods for numerical solution of boundary-value problems," Proc. Lecture Series, Hemisphere Publ. Co., Washington (1983).
7. D. Marsal, Numerical Solution of Partial Differential Equations [in German], Vol. 1, Wissenschaftsverlag, Zurich (1976).
8. A. A. Samarskii, Introduction to the Theory of Difference Schemes [in Russian], Nauka, Moscow (1971).
9. A. A. Samarskii and E. S. Nikolaev, Methods of Solving Grid Equations [in Russian], Nauka, Moscow (1978).
10. R. Peyret and T. D. Taylor, Computational Methods for Fluid Flow, Springer-Verlag, New York (1983).
11. W. Zwick, "Solution of the Navier–Stokes equation by means of a fully implicit iteration scheme," ZAMM, 64, 221-226 (1984).
12. W. Nadeborn, "Realization of a decoupling scheme with a dynamic iteration parameter handling," in: Proc. Int. Seminar-School, Mathematical Models in the Theory of Heat and Mass Transfer, Academy of Sciences, BSSR, Minsk (1982), pp. 93-105.
13. K. Perktold, "Solution of the nonstationary Navier–Stokes equation by the finite element method," Forschungszentrum Graz, Report No. 133 (1980).
14. R. Temam, Navier–Stokes Equations, North-Holland, Amsterdam (1977).
15. W. Zwick, "Calculation of laminary incompressible flows," in: Numerical Solution of Differential Equations [in German], Report R-Math-01/83, Academy of Sciences, DDR, Institute of Mathematics, Berlin (1983), pp. 93-102.

16. W. Zwick,"Calculation of nonlinear flows," in: Proc. Int. Seminar-School, Mathematical Methods in the Theory of Heat and Mass Transfer, Academy of Sciences, BSSR, Minsk (1982), pp. 3-12.

17. W. Zwick, Solution of the Boussinesq Equation by a Fully Implicit Iteration Scheme [in German], Academy of Sciences of the DDR (1983), pp. 223-238.

18. O. Buneman, "A compact noniterative Poisson solver," Report No. 294, Institute for Plasma Physics, Stanford (1969).

19. R. W. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis," Commun. Assoc. Comp. Mach., 12, 95 (1965).

20. R. V. Madala, "An efficient direct solver for separable and nonseparable elliptic equations," Mon. Weather Rev., 106, 1735 (1978).

21. D. L. Book, Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations, Springer-Verlag, New York (1981).

22. G. D. Smith, Numerical Solution of Partial Differential Equations [in German], Akademie-Verlag, Berlin (1971).

23. W. Zwick, "A relaxation method for solving balance equations," Report No. R-05/80, Academy of Sciences of the DDR, Berlin (1980), pp. 150-164.

24. W. Zwick, "Numerical solution of the heat-conduction equation," Academy of Sciences of the DDR (1981), pp. 251-263.

25. S.C. Eisenstat, M. C. Gursky, M. H. Schultz, and A. H. Sherman, Yale Sparse Matrix Package, II: The Nonsymmetric Codes, Research Report 114, Dept. Computer Science, Yale University.

26. P. J. van der Houwen and B. P. Sommeijer, "Analysis of Chebyshev relaxation in multigrid methods for nonlinear parabolic differential equations," ZAMM, 63, 193-201 (1983).

27. W. Zwick, "On a multigrid method for the numerical solution of Navier—Stokes equations," Second IUTAM Symp. Laminar-Turbulent Transition [in Russian], Novosibirsk, Springer-Verlag (1984).