A generalized Douglas ADI method for solving three-dimensional parabolic differential equations on multilayers

W. Dai and R. Nassar
Department of Mathematics and Statistics, Louisiana Tech University, Ruston, Louisiana, USA

1. Introduction

Parabolic differential equations in three dimensions and on multilayers often arise in engineering processes[1-5]. An example is x-ray lithography which is an important technology in micromanufacturing. The process is composed of a mask and a photoresist deposited on a substrate. A typical photoresist is the photosensitive polymethylmethacrylate (PMMA). The mask layer creates a desired pattern on the photoresist by selectively allowing the transition of irradiation from an x-ray beam. Prediction of the temperature distribution in three dimensions in the different layers (mask, photoresist and substrate) is essential for determining the effect of high flux x-ray exposure on distortions in the photoresist due to thermal expansion and on bonding between resist and substrate. A thorough understanding of the problem has been hampered by the difficulties involved in solving the differential equations describing temperature profiles in multilayers. These difficulties include the unknown value at the interface between layers, and the small spatial scale measured in microns. In applications of numerical methods for solving parabolic differential equations on multilayers, the common approach, to overcome the interface problem, is to apply the iteration method. As such, the unknown value at the interface between layers is replaced by the value at the previous time step, and is iterated until the solution is obtained. However, the iteration method in the three-dimensional case requires too much computational time. Furthermore, the small spatial scale results in a fine spatial grid size compared with the time increment. Thus, the mesh ratio is large, which causes many second-order accurate schemes, such as the Crank-Nicloson scheme and the Douglas ADI scheme, to converge slowly to the steady-state solution.

In this paper, we present three-dimensional numerical procedures for solving parabolic differential equations on multilayers. To avoid iteration at each time step, the ADI method is employed in these models. There are many ADI schemes for solving parabolic differential equations. A simple well known

International Journal for Numerical
Methods for Heat & Fluid Flow
Vol. 7 No. 7, 1997, pp. 659–674.

three-dimensional ADI scheme is the Douglas ADI scheme. It is unconditionally stable and second-order accurate. However, it is slow in convergence when used to simulate fast transient phenomena or for computations on fine spatial meshes. In this study, a generalized three-dimensional Douglas ADI scheme, which is suited for either simulating fast transient phenomena or for numerical computations on fine spatial meshes, is developed. Numerical models for multilayers that employ this scheme are formulated. To overcome the problem with the unknown value at the interface between layers, the generalized "divide and conquer" procedure for solving tridiagonal linear systems is applied. As such, the computational procedure is simple and efficient.

2. Generalized ADI scheme

Consider the three-dimensional parabolic equation

$$\frac{\partial U}{\partial t}=\mu\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}+\frac{\partial^2 U}{\partial z^2}\right)+f(x,y,z,t) \qquad (2.1)$$

where μ is the diffusivity coefficient. We let u_{ijk}^n denote the approximation to U $(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$, where Δx, Δy and Δz are the grid sizes in the x, y and z directions respectively, Δt is the time increment, $i = 0, \ldots, N_x$, $j = 0, \ldots, N_y$, and $k = 0, \ldots, N_z$. We use the centred-difference equation,

$$\frac{1}{\Delta x^2}\delta_x^2 u_{ijk}^n=\frac{1}{\Delta x^2}\left(u_{i+1jk}^n-2u_{ijk}^n+u_{i-1jk}^n\right),$$

to approximate $\frac{\partial^2 U}{\partial x^2}$, and so on. Then, the three-dimensional Douglas ADI scheme can be written as follows:

$$\left(1-\frac{r_x}{2}\delta_x^2\right)\left(u_{ijk}^{n+\frac{1}{3}}-u_{ijk}^n\right)=\left(r_x\delta_x^2+r_y\delta_y^2+r_z\delta_z^2\right)u_{ijk}^n+\Delta t\, f_{ijk}^{n+\frac{1}{2}}, \qquad (2.2a)$$

$$u_{ijk}^{n+\frac{2}{3}}-u_{ijk}^{n+\frac{1}{3}}=\frac{1}{2}r_y\delta_y^2\left(u_{ijk}^{n+\frac{2}{3}}-u_{ijk}^n\right), \qquad (2.2b)$$

$$u_{ijk}^{n+1}-u_{ijk}^{n+\frac{2}{3}}=\frac{1}{2}r_z\delta_z^2\left(u_{ijk}^{n+1}-u_{ijk}^n\right), \qquad (2.2c)$$

where $r_x = \frac{\mu\Delta t}{\Delta x^2}$, $r_y = \frac{\mu\Delta t}{\Delta y^2}$ and $r_z = \frac{\mu\Delta t}{\Delta z^2}$. The above scheme is unconditionally stable and second-order accurate. Based on the idea in Samarskii and Vabishchevich[6], we develop a generalized Douglas ADI scheme as follows:

$$\left(1-\frac{r_x+r_x\varepsilon}{2}\delta_x^2\right)\left(u_{ijk}^{n+\frac{1}{3}}-u_{ijk}^n\right)$$

$$=\left(r_x\delta_x^2+r_y\delta_y^2+r_z\delta_z^2\right)u_{ijk}^n+\left(6\varepsilon+2\varepsilon^3\right)\frac{r_x}{2}\frac{r_y}{2}\frac{r_z}{2}\delta_x^2\delta_y^2\delta_z^2u_{ijk}^n+\Delta t\, f_{ijk}^{n+\frac{1}{2}} \qquad (2.3a)$$

$$u_{ijk}^{n+\frac{2}{3}} - u_{ijk}^{n+\frac{1}{3}} = \frac{r_y + r_y \varepsilon}{2} \delta_y^2 (u_{ijk}^{n+\frac{2}{3}} - u_{ijk}^{n}), \tag{2.3b}$$

$$u_{ijk}^{n+1} - u_{ijk}^{n+\frac{2}{3}} = \frac{r_z + r_z \varepsilon}{2} \delta_z^2 (u_{ijk}^{n+1} - u_{ijk}^{n}), \tag{2.3c}$$

where ε is a small positive number. When ε = 0, the above scheme reduces to the Douglas ADI scheme.

For investigating the stability of the system, we assume r_x, = r_y = r_z = r, for convenience. Without the term $\Delta f_{ijk}^{n+\frac{1}{2}}$, we rewrite (2.3b) and (2.3c) as

$$u_{ijk}^{n+\frac{1}{3}} - u_{ijk}^{n} = (1 - \frac{r + r\varepsilon}{2} \delta_y^2)(u_{ijk}^{n+\frac{2}{3}} - u_{ijk}^{n}), \tag{2.4a}$$

$$u_{ijk}^{n+\frac{2}{3}} - u_{ijk}^{n} = (1 - \frac{r + r\varepsilon}{2} \delta_z^2)(u_{ijk}^{n+1} - u_{ijk}^{n}). \tag{2.4b}$$

Eliminating $u_{ijk}^{n+\frac{2}{3}} - u_{ijk}^{n}$, one obtains

$$u_{ijk}^{n+\frac{1}{3}} - u_{ijk}^{n} = (1 - \frac{r + r\varepsilon}{2} \delta_y^2)(1 - \frac{r + r\varepsilon}{2} \delta_z^2)(u_{ijk}^{n+1} - u_{ijk}^{n}). \tag{2.5}$$

Substituting (2.5) into (2.3a), we obtain an equivalent scheme to (2.3), namely,

$$(1 - \frac{r + r\varepsilon}{2} \delta_x^2)(1 - \frac{r + r\varepsilon}{2} \delta_y^2)(1 - \frac{r + r\varepsilon}{2} \delta_z^2)(u_{ijk}^{n+1} - u_{ijk}^{n})$$

$$= r(\delta_x^2 + \delta_y^2 + \delta_z^2)\, u_{ijk}^{n} + (6\varepsilon + 2\varepsilon^3)\left(\frac{r}{2}\right)^3 \delta_x^2 \delta_y^2 \delta_z^2 \, u_{ijk}^{n}. \tag{2.6}$$

In matrix form (2.6) may be expressed as

$$A\left(u^{n+1}\right) = B\left(u^{n}\right). \tag{2.7}$$

It can be seen that matrices A and B are symmetric. Hence, $H = A^{-1} B$ is a N × N symmetric matrix. As such,

$$\left(u^{n}\right) = H\left(u^{n-1}\right) = H^{n}\left(u^{0}\right). \tag{2.8}$$

If one expands (2.6) in a Fourier series, one obtains, by Parseval's equality,

$$\|u^{n}\|^2 = \sum_{l} \left| v_l^{n} \right|^2, \tag{2.9}$$

where $v_l^n$is the coefficient of the Fouier series. It can be seen that

$$v_l^{n} = G(r, \sigma_1, \sigma_2, \sigma_3, \varepsilon)\, v_l^{n-1} = \left(G(r, \sigma_1, \sigma_2, \sigma_3, \varepsilon)\right)^{n} v_l^{0}, \tag{2.10}$$

where the amplification factor is

$$G(r, \sigma_1, \sigma_2, \sigma_3, \varepsilon)$$

$$= 1 - \frac{4r(\beta_1 + \beta_1 + \beta_1) + 8r^3(6\varepsilon + 2\varepsilon^3)\beta_1\beta_2\beta_3}{\big(1 + 2r(1+\varepsilon)\beta_1\big)\big(1 + 2r(1+\varepsilon)\beta_2\big)\big(1 + 2r(1+\varepsilon)\beta_3\big)}. \qquad (2.11)$$

In equation (2.11), $\beta_1 = \sin^2\frac{\sigma 1h}{2}$, $\beta_2 = \sin^2\frac{\sigma 2h}{2}$ and $\beta_3 = \sin^2\frac{\sigma 3h}{2}$, $(0 < \beta_1, \beta_2, \beta_3 < 1)$, $\sigma_1 = i\pi$, $\sigma_2 = j\pi$ and $\sigma_3 = k\pi$; i, j, k = 1, …, N – 1. Therefore, the scheme (2.6) is stable if and only if $|G| < 1 + c\,\Delta t$, where c is a constant. When $0 \le \varepsilon < 1$, it is seen that

$$\begin{aligned}
&1 + 2r(1+\varepsilon)\beta_1 + 2r(1+\varepsilon)\beta_2 + 2r(1+\varepsilon)\beta_3 + 4r^2(1+\varepsilon)^2\beta_1\beta_2 \\
&+ 4r^2(1+\varepsilon)^2\beta_1\beta_3 + 4r^2(1+\varepsilon)^2\beta_2\beta_3 + 8r^3(1+\varepsilon)^3\beta_1\beta_2\beta_3 \\
&- 4r(\beta_1 + \beta_2 + \beta_3) - 8r^3(6\varepsilon + 2\varepsilon^3)\beta_1\beta_2\beta_3 \\
&= 1 - 2r(1-\varepsilon)\beta_1 - 2r(1-\varepsilon)\beta_2 - 2r(1-\varepsilon)\beta_3 + 4r^2(1+\varepsilon)^2\beta_1\beta_2 \\
&+ 4r^2(1+\varepsilon)^2\beta_1\beta_3 + 4r^2(1+\varepsilon)^2\beta_2\beta_3 + 8r^3(1-\varepsilon)^3\beta_1\beta_2\beta_3 \\
&< 1 + 2r(1+\varepsilon)\beta_1 + 2r(1+\varepsilon)\beta_2 + 2r(1+\varepsilon)\beta_3 + 4r^2(1+\varepsilon)^2\beta_1\beta_2 \\
&+ 4r^2(1+\varepsilon)^2\beta_1\beta_3 + 4r^2(1+\varepsilon)^2\beta_2\beta_3 + 8r^3(1+\varepsilon)^3\beta_1\beta_2\beta_3 \qquad (2.12)
\end{aligned}$$

and

$$\begin{aligned}
&-\Big\{1 + 2r(1+\varepsilon)\beta_1 + 2r(1+\varepsilon)\beta_2 + 2r(1+\varepsilon)\beta_3 + 4r^2(1+\varepsilon)^2\beta_1\beta_2 \\
&\qquad +4r^2(1+\varepsilon)^2\beta_1\beta_3 + 4r^2(1+\varepsilon)^2\beta_2\beta_3 + 8r^3(1+\varepsilon)^3\beta_1\beta_2\beta_3\Big\} \\
&< 1 - 2r(1-\varepsilon)\beta_1 - 2r(1-\varepsilon)\beta_2 - 2r(1-\varepsilon)\beta_3 + 4r^2(1+\varepsilon)^2\beta_1\beta_2 \\
&+ 4r^2(1+\varepsilon)^2\beta_1\beta_3 + 4r^2(1+\varepsilon)^2\beta_2\beta_3 + 8r^3(1-\varepsilon)^3\beta_1\beta_2\beta_3. \qquad (2.13)
\end{aligned}$$

From (2.12) and (2.13) it is seen that $-1 < G < 1$, hence, $|G| < 1$. Thus, the new ADI scheme is unconditionally stable. With $|G| < 1$, we have from equation (2.10)

$$|v_l^n| \le |v_l^0| \quad \text{and} \quad \lim_{n\to\infty}|v_l^n| = 0. \qquad (2.14)$$

Further, when r is large, implying that $\max(r\beta_1)$, $\max(r\beta_2)$ and $\max(r\beta_3)$ are also large, the dominating terms in (2.11) are $8r^3(1-\varepsilon)^3\beta_1\beta_2\beta_3$ and $8r^3(1+\varepsilon)^3\beta_1\beta_2\beta_3$. As such,

$$G \sim \frac{8r^3(1-\varepsilon)^3\beta_1\beta_2\beta_3}{8r^3(1+\varepsilon)^3\beta_1\beta_2\beta_3} = \frac{(1-\varepsilon)^3}{(1+\varepsilon)^3} < 1 \qquad \text{if } 0 < \varepsilon < 1, \tag{2.15}$$

while G ~ 1 if ε = 0. It is seen from equation (2.10) that $|v_i^n$ in the Douglas ADI scheme (ε = 0, G ~ 1) goes to zero very slowly as n → ∞. On the other hand, the generalized Douglas ADI scheme (ε > 0, G < 1) goes to zero much faster. A large mesh ratio r may come from a large μ or a fine spatial grid size compared with the time increment, implying that the Douglas ADI scheme is not well suited for either simulating a fast transient phenomenon or for computations on a fine spatial mesh.

It should be pointed out that there is an operator-splitting method called the θ-scheme, which is suited for simulating fast transient phenomena[7]. Its amplification factor is $G \sim \frac{1}{\sqrt{2}}$ when μ is large. However, it is not an ADI scheme because it does not split a finite difference algorithm into a sequence of one-dimensional operations. Hence, iteration must be used for obtaining the solution at each time step, which leads to problems in computing time.

To determine the accuracy of the generalized scheme, we replace u_{ijk} by $u(x_i, y_j, z_k)$ in (2.6) to obtain

$$\begin{aligned}
& u'\,u^{n+1} - \frac{r+r\varepsilon}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)u^{n+1} + \left(\frac{r+r\varepsilon}{2}\right)^2(\delta_x^2\delta_y^2 + \delta_x^2\delta_z^2 \\
&\quad + \delta_y^2\delta_z^2)\,u^{n+1} - \left(\frac{r+r\varepsilon}{2}\right)^3\delta_x^2\delta_y^2\delta_z^2\,u^{n+1} \\
= \; & u^n - \frac{r+r\varepsilon}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)u^n + \left(\frac{r+r\varepsilon}{2}\right)^2(\delta_x^2\delta_y^2 + \delta_x^2\delta_z^2 + \delta_y^2\delta_z^2)\,u^n \\
& - \left(\frac{r+r\varepsilon}{2}\right)^3\delta_x^2\delta_y^2\delta_z^2\,u^n + r\left(\delta_x^2+\delta_y^2+\delta_z^2\right)u^n \\
& + (6\varepsilon + 2\varepsilon^2)\left(\frac{r}{2}\right)^3\delta_x^2\delta_y^2\delta_z^2\,u^n \\
= \; & u^n + \frac{r-r\varepsilon}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)u^n + \left(\frac{r+r\varepsilon}{2}\right)^2(\delta_x^2\delta_y^2 + \delta_x^2\delta_z^2 \\
& + \delta_y^2\delta_z^2)\,u^n - \left(\frac{r-r\varepsilon}{2}\right)^3\delta_x^2\delta_y^2\delta_z^2\,u^n .
\end{aligned} \tag{2.16}$$

By Taylor series,

$$
\begin{aligned}
& u^{n+1} - \frac{\Delta t + \Delta t\varepsilon}{2}\left(\mu\Delta\, u^{n+1} + 0(h^2)\right) \\
& + \left(\mu\frac{\Delta t + \Delta t\varepsilon}{2}\right)^2\left(u^{n+1}_{xxyy} + u^{n+1}_{xxzz} + u^{n+1}_{yyzz} + 0(h^2)\right) \\
& - \left(\mu\frac{\Delta t + \Delta t\varepsilon}{2}\right)^3\left(u^{n+1}_{xxyyzz} + 0(h^2)\right) \\
= \quad & u^{n} + \frac{\Delta t - \Delta t\varepsilon}{2}\left(\mu\Delta\, u^{n} + 0(h^2)\right) \\
& + \left(\mu\frac{\Delta t + \Delta t\varepsilon}{2}\right)^2\left(u^{n}_{xxyy} + u^{n}_{xxzz} + u^{n}_{yyzz} + 0(h^2)\right) \\
& - \left(\mu\frac{\Delta t - \Delta t\varepsilon}{2}\right)^3\left(u^{n}_{xxyyzz} + 0(h^2)\right).
\end{aligned}
\tag{2.17}
$$

Therefore, at time $t = (n + \frac{1}{2})\Delta t$,

$$
\begin{aligned}
u_t^{n+\frac{1}{2}} &= \mu\,\Delta u^{n+\frac{1}{2}} + \Delta t\varepsilon\mu\Delta\, u_t^{n+\frac{1}{2}} + 0\left(h^2 + \Delta t^2 + \varepsilon\,\Delta t^2\right) \\
&= \mu\,\Delta u^{n+\frac{1}{2}} + 0\left(h^2 + \Delta t^2 + \Delta t\varepsilon + \varepsilon\,\Delta t^2\right).
\end{aligned}
\tag{2.18}
$$

which is first-order in accuracy. If ε is small, accuracy will be high. This raises the question as to how small ε should be chosen if both the amplification factor and accuracy are considered. To obtain a G value ($G \sim \frac{1}{\sqrt{2}} = 0.707$) similar to that of the θ-scheme when r is large, we choose ε = 0.05, which gives, from (2.15), G ~ 0.741. In practice, Δt is usually chosen not to be very small. Therefore, with ε = 0.05 the accuracy of the generalized Douglas ADI scheme is $0[h^2 + \Delta t^2 + 0.05\Delta t]$.

3. Generalized divide and conquer procedures

Consider solving a tridiagonal linear system

$$
\begin{aligned}
& -b_i\,x_{i-1} + a_i\,x_i - c_i\,x_{i+1} = d_i\,, \quad i = 1, \cdots, n, \\
& x_o = x_{n+1} = 0.
\end{aligned}
\tag{3.1}
$$

In matrix form equation (3.1) can be expressed as

$$\begin{bmatrix} a_1 & -c_1 & & & & & \\ -b_2 & a_2 & -c_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & -b_i & a_i & -c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -b_{n-1} & a_{n-1} & -c_{n-1} \\ & & & & & -b_n & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{3.2}$$

The usual approach for solving the above tridiagonal linear system is the Gaussian elimination technique. This approach results in a procedure called the “divide and conquer” procedure, shown as follows:

$$\beta_k = \frac{c_k}{a_k - b_k \beta_{k-1}}, \quad \beta_o = 0, \tag{3.3a}$$

$$\nu_k = \frac{d_k + b_k \nu_{k-1}}{a_k - b_k \beta_{k-1}}, \ \nu_o = 0; \ k = 1, 2, \ldots n, \tag{3.3b}$$

$$x_k = \beta_k x_{k+1} + \nu_k, \quad x_{n+1} = 0; \ k = n, \ldots, 1. \tag{3.3c}$$

In the above procedure, β_k, ν_k are calculated from k = 1 to k = n, while x_k is computed from k = n to k =1. A similar procedure that is opposite in direction can be expressed as

$$\tilde{\beta}_k = \frac{b_k}{a_k - c_k \tilde{\beta}_{k+1}}, \quad \tilde{\beta}_{n+1} = 0, \tag{3.4a}$$

$$\tilde{\nu}_k = \frac{d_k + c_k \tilde{\nu}_{k+1}}{a_k - c_k \tilde{\beta}_{k+1}}, \quad \tilde{\nu}_{k+1} = 0; \ k = n, \ldots, 1, \tag{3.4b}$$

$$x_k = \tilde{\beta}_k x_{k-1} + \tilde{\nu}_k, \quad x_o = 0; \ k = 1, 2, \ldots n. \tag{3.4c}$$

Further, if x_o and x_{n+1} are not zero, one has

$$\beta_k = \frac{c_k}{a_k - \beta_{k-1} b_k}, \ \beta_o = 0, \tag{3.5a}$$

$$\nu_k = \frac{d_k + b_k \nu_{k-1}}{a_k - b_k \beta_{k-1}}, \quad \nu_o = 0, \tag{3.5b}$$

$$\lambda_k = \frac{b_k \lambda_{k-1}}{a_k - \beta_{k-1} b_k}, \quad \lambda_o = 1; \ k = 1, \ldots, n, \tag{3.5c}$$

$$x_k = \beta_k x_{k+1} + \nu_k + \lambda_k x_0; \quad k = n,\dots, 1, \tag{3.5d}$$

and

$$\tilde{\beta}_k = \frac{b_k}{a_k - \tilde{\beta}_{k+1} c_k}, \quad \tilde{\beta}_{n+1} = 0, \tag{3.6a}$$

$$\tilde{\nu}_k = \frac{d_k + c_k \tilde{\nu}_{k+1}}{a_k - c_k \tilde{\beta}_{k+1}}, \quad \tilde{\nu}_{n+1} = 0, \tag{3.6b}$$

$$\tilde{\lambda}_k = \frac{c_k \tilde{\lambda}_{k+1}}{a_k - \tilde{\beta}_{k+1} c_k}, \quad \tilde{\lambda}_{n+1} = 1; \; k = n,\dots, 1; \tag{3.6c}$$

$$x_k = \tilde{\beta}_k x_{k-1} + \tilde{\nu}_k + \tilde{\lambda}_k x_{n+1}; \quad k = 1,\dots, n. \tag{3.6d}$$

Thus, combining procedures (3.3) and (3.4), one can develop a generalized "divide and conquer" procedure. To this end, let n = 2N + 1 for convenience and divide the system (3.1) into two subsystems, which consist of the first N equations and the last N equations, with the (N + 1)th equation denoting the interface. As such, the procedure can be described as follows:

Procedure 1

Step 1. Calculate $\left\{ \begin{array}{l} \beta_k, \nu_k, \; k = 1,\cdots, N \\ \\ \tilde{\beta}_k, \tilde{\nu}_k, \; k = 2N+1,\cdots, N+2 \end{array} \right\}.$

Step 2. Substitute $x_N = \beta_N x_{N+1} + \nu_N$ and $x_{N+2} = \tilde{\beta}_{N+2} x_{N+1} + \tilde{\nu}_{N+2}$ into $-b_{N+1} x_N + a_{N+1} x_{N+1} - c_{N+1} x_{N+2} = d_{N+1}$, then solve

$$x_{N+1} = \frac{d_{N+1} + b_{N+1} \nu_N + c_{N+1} \tilde{\nu}_{N+2}}{a_{N+1} - b_{N+1} \beta_N - c_{N+1} \tilde{\beta}_{N+2}}.$$

Step 3. Solve $\left\{ \begin{array}{l} x_k = \beta_k x_{k+1} + \nu_k, \; k = N,\cdots, 1 \\ x_k = \tilde{\beta}_k x_{k-1} + \tilde{\nu}_k, \; k = N+2,\cdots, 2N+1 \end{array} \right\}.$

Further, combining (3.3), (3.4), (3.5) and (3.6), one can develop a more complicated generalized "divide and conquer" procedure. To this end, let n = 3N + 2, for convenience. We divide the system (3.1) into three subsystems, which consist of the first N equations, the middle N equations and the last N equations. The (N + 1)th and the (2N + 2)th equations designate the interfacial equations. As such, the procedure can be described as follows:

Procedure 2

$$\text{Step 1. Calculate}\left\{\begin{array}{l}\beta_k^{(1)}, \nu_k^{(1)},\ k=1,\ldots,N,\ \text{for the first N equations by (3.3)}\\ \beta_k^{(2)}, \nu_k^{(2)}, \lambda_k^{(2)}\ k=1,\ldots,N.\ \text{for the middle N equations by (3.5)}\\ \tilde{\beta}_k^{(2)}, \tilde{\nu}_k^{(2)}, \tilde{\lambda}_k^{(2)}\ k=N,\ldots,1.\ \text{for the middle N equations by (3.6)}\\ \tilde{\beta}_k^{(3)}, \tilde{\nu}_k^{(3)},\ k=N,\ldots,1.\ \text{for the last N equations by (3.4)}\end{array}\right\}.$$

Step 2. Substitute $x_N = \beta_N^{(1)} x_{N+1} + \nu_N^{(1)}$, $x_{2N+1} = \beta_N^{(2)} x_{2N+2} + \nu_N^{(2)} + \lambda_N^{(2)} x_{N+1}$, $x_{N+2} = \tilde{\beta}_1^{(2)} x_{N+1} + \tilde{\nu}_1^{(2)} + \tilde{\lambda}_1^{(2)} x_{2N+2}$ and $x_{2N+3} = \tilde{\beta}_1^{(3)} x_{2N+2} + \tilde{\nu}_1^{(3)}$ into $\left\{\begin{array}{l}-b_{N+1} x_N + a_{N+1} x_{N+1} - c_{N+1} x_{N+2} = d_{N+1}\\ -b_{2N+2} x_{2N+1} + a_{2N+2} x_{2N+2} - c_{2N+2} x_{2N+3} = d_{2N+2}\end{array}\right\}$, then solve for x_{N+1} and x_{2N+2} in the following equations.

$$\begin{cases}A_1 x_{N+1} - C_1 x_{2N+2} = D_1\\ \\ -B_1 x_{N+1} + A_2 x_{2N+2} = D_2\end{cases}$$

where

$$\begin{cases}A_1 = a_{N+1} - b_{N+1} \beta_N^{(1)} - c_{N+1} \tilde{\beta}_1^{(2)}\\ C_1 = c_{N+1} \tilde{\lambda}_1^{(2)}\\ D_1 = d_{N+1} + b_{N+1} \nu_N^{(1)} + c_{N+1} \tilde{\nu}_1^{(2)}\end{cases}$$

and

$$\begin{cases}A_2 = a_{2N+2} - b_{2N+2} \beta_N^{(2)} - c_{2N+2} \tilde{\beta}_1^{(3)}\\ B_2 = b_{2N+2} \lambda_N^{(2)}\\ D_2 = d_{2N+2} + b_{2N+2} \lambda_N^{(2)} + c_{2N+2} \tilde{\nu}_1^{(3)}\end{cases}.$$

$$\text{Step 3. Solve}\left\{\begin{array}{l}x_k = \beta_k^{(1)} x_{k+1} + \nu_k^{(1)}\ ,\ k=N,\ldots\\ x_{N+k+1} = \beta_k^{(2)} x_{N+k+2} + \nu_k^{(2)} + \lambda_k^{(2)} x_{N+1}\ ,\ k=N,\ldots,1\\ x_{2N+k+2} = \tilde{\beta}_k^{(3)} x_{2N+k+1} + \tilde{\nu}_k^{(3)}\ ,\ k=1,\ldots,N\end{array}\right\}.$$

The idea of the above generalized “divide and conquer” procedures can be applied to a tridiagonal linear system which is divided into many subsystems. It should be pointed out that the generalized “divide and conquer” procedures are one type of the domain decomposition methods[8]. Therefore, they are characterized by a high inherent parallelism.

4. Numerical models for solving parabolic equations on multilayers

Consider a three-dimensional domain with three layers, as shown in Figure 1. Three-dimensional parabolic differential equations on three layers can be expressed as

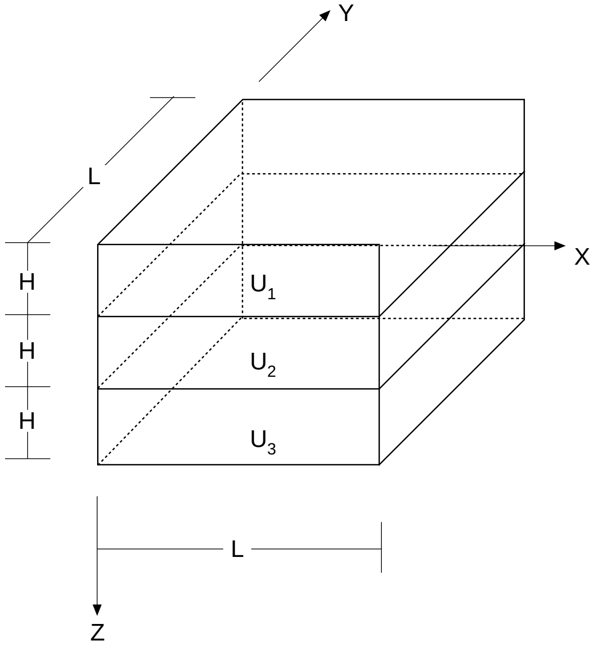

Figure 1. Three-dimensional domain with three layers for equations (4.1a, 1.1b and 4.1c)

$$\frac{\partial U_1}{\partial t} = \mu_1\left(\frac{\partial^2 U_1}{\partial x^2} + \frac{\partial^2 U_1}{\partial y^2} + \frac{\partial^2 U_1}{\partial z^2}\right) + f_1(x, y, z, t), \tag{4.1a}$$

$$\frac{\partial U_2}{\partial t} = \mu_2\left(\frac{\partial^2 U_2}{\partial x^2} + \frac{\partial^2 U_2}{\partial y^2} + \frac{\partial^2 U_2}{\partial z^2}\right) + f_2(x, y, z, t), \tag{4.1b}$$

$$\frac{\partial U_3}{\partial t} = \mu_3\left(\frac{\partial^2 U_3}{\partial x^2} + \frac{\partial^2 U_3}{\partial y^2} + \frac{\partial^2 U_3}{\partial z^2}\right) + f_3(x, y, z, t). \tag{4.1c}$$

We assume that the flux across the interface does not change. As such, at z = H,

$$-\mu_1 \frac{\partial U_1}{\partial z} = -\mu_2 \frac{\partial U_2}{\partial z}, \quad U_1 = U_2 \ , \tag{4.2a}$$

and at z = 2H,

$$-\mu_2 \frac{\partial U_2}{\partial z} = -\mu_3 \frac{\partial U_3}{\partial z}, \quad U_2 = U_3. \tag{4.2b}$$

To obtain the numerical solution in the three-dimensional case, we assume that there is a mesh grid of $N_x \times N_y \times N_z$ for each of the layers with the same grid size Δx , Δy and Δz, where $(N_x + 1)\Delta x = L$, $(N_y + 1)\Delta y = L$ and $(N_z + 1)\Delta z = H$. We employ the generalized Douglas ADI scheme in Section 2 to solve equations (4.1). As such,

$$\left(1-\frac{r_x+r_x\varepsilon_l}{2}\delta_x^2\right)\left[(u_l)_{ijk}^{n+\frac{1}{3}}-(u_l)_{ijk}^{n}\right]$$

$$=(r_x\delta_x^2+r_y\delta_y^2+r_z\delta_z^2)(u_l)_{ijk}^{n}+(6\varepsilon_l+2\varepsilon_l{}^3)\frac{r_x}{2}\frac{r_y}{2}\frac{r_z}{2}\delta_x^2\delta_y^2\delta_z^2(u_l)_{ijk}^{n} \tag{4.3a}$$

$$+\Delta t(f_l)_{ijk}^{n+\frac{1}{2}},$$

$$(u_l)_{ijk}^{n+\frac{2}{3}}-(u_l)_{ijk}^{n+\frac{1}{3}}=\frac{r_y+r_y\varepsilon_l}{2}\delta_y^2\left[(u_l)_{ijk}^{n+\frac{2}{3}}-(u_l)_{ijk}^{n}\right], \tag{4.3b}$$

$$(u_l)_{ijk}^{n+1}-(u_l)_{ijk}^{n+\frac{2}{3}}=\frac{r_z+r_z\varepsilon_l}{2}\delta_z^2\left[(u_l)_{ijk}^{n+1}-(u_l)_{ijk}^{n}\right], \tag{4.3c}$$

where l represents the (l)th layer (l = 1, 2, 3). For the interfacial equations, we let, at any time step n,

$$-\mu_1\left(\frac{(u_1)_{ijNz+1}^{n}-(u_1)_{ijNz}^{n}}{\Delta z}\right)=-\mu_2\left(\frac{(u_2)_{ij1}^{n}-(u_2)_{ij0}^{n}}{\Delta z}\right),\ (u_1)_{ijNz+1}^{n}=(u_2)_{ij0}^{n}, \tag{4.4a}$$

and

$$-\mu_2\left(\frac{(u_2)_{ijNz+1}^{n}-(u_2)_{ijNz}^{n}}{\Delta z}\right)=-\mu_3\left(\frac{(u_3)_{ij1}^{n}-(u_3)_{ij0}^{n}}{\Delta z}\right),\ (u_2)_{ijNz+1}^{n}=(u_3)_{ij0}^{n}. \tag{4.4b}$$

Hence, the computational procedure of the three-dimensional model is as follows:

The first step is to determine

$$\left\{(u_1)_{ijk}^{n+\frac{1}{3}}\right\},\left\{(u_2)_{ijk}^{n+\frac{1}{3}}\right\}\text{ and }\left\{(u_3)_{ijk}^{n+\frac{1}{3}}\right\}$$

by using equation (4.3a). To this end, we solve three tridiagonal linear systems to obtain

$$\left\{(u_1)_{ijk}^{n+\frac{1}{3}}\right\},\left\{(u_2)_{ijk}^{n+\frac{1}{3}}\right\}\text{ and }\left\{(u_3)_{ijk}^{n+\frac{1}{3}}\right\},$$

i = 1, …, N_x, j = 1, …, N_y, k = 1, …, N_z, independently, then, we substitute $(u_1)_{ijNz}^{n+\frac{1}{3}}$ and $(u_2)_{ij1}^{n+\frac{1}{3}}$ into equation (4.4a) to obtain $(u_1)_{ijNz+1}^{n+\frac{1}{3}}$, and $(u_1)_{ijNz+1}^{n+\frac{1}{3}}$ and $(u_2)_{ijo}^{n+\frac{1}{3}}$, and $(u_2)_{ijNz}^{n+\frac{1}{3}}$ and $(u_3)_{ij1}^{n+\frac{1}{3}}$ into equation (4.4b) to obtain $(u_2)_{ijNz+1}^{n+\frac{1}{3}}$ and $(u_3)_{ijo}^{n+\frac{1}{3}}$.

Similarly, the second step is to determine

$$\left\{(u_1)_{ijk}^{n+\frac{2}{3}}\right\},\left\{(u_2)_{ijk}^{n+\frac{2}{3}}\right\}\text{ and }\left\{(u_3)_{ijk}^{n+\frac{2}{3}}\right\}$$

by using equation (4.3b).

The third step is to determine

$$\left\{(u_1)_{ijk}^{n+1}\right\}, \left\{(u_2)_{ijk}^{n+1}\right\} \text{ and } \left\{(u_3)_{ijk}^{n+1}\right\}$$

by using equation (4.3c). As such, we express equations (4.3c), with l = 1, 2, 3, as three tridiagonal linear systems.

$$-b_k^1 (u_1)_{ijk-1}^{n+1} + a_k^1 (u_1)_{ijk}^{n+1} - c_k^1 (u_1)_{ijk+1}^{n+1} = d_{ijk}^1, \quad k = 1, \ldots, N_Z, \tag{4.5a}$$

$$-b_k^2 (u_2)_{ijk-1}^{n+1} + a_k^2 (u_2)_{ijk}^{n+1} - c_k^2 (u_2)_{ijk+1}^{n+1} = d_{ijk}^2, \quad k = 1, \ldots, N_Z, \tag{4.5b}$$

$$-b_k^3 (u_3)_{ijk-1}^{n+1} + a_k^3 (u_3)_{ijk}^{n+1} - c_k^3 (u_3)_{ijk+1}^{n+1} = d_{ijk}^3, \quad k = 1, \ldots, N_Z, \tag{4.5c}$$

where i = 1, …, N_x and j = 1, …, N_y. Since u_l (l = 1, 2, 3) at the (n + 1)Δt time step is unknown at the interface between layers, the above three tridiagonal linear systems cannot be solved. To overcome this difficulty, we apply the generalized "divide and conquer" Procedure 2, and calculate the coefficients listed in step 1 of Procedure 2, then, substitute the following four equations:

$$\begin{cases} (u_1)_{ijNz}^{n+1} = \beta_{Nz}^{(1)} (u_1)_{ijNz+1}^{n+1} + \nu_{Nz}^{(1)} \\ (u_2)_{ijNz}^{n+1} = \beta_{Nz}^{(2)} (u_2)_{ijNz+1}^{n+1} + \nu_{Nz}^{(2)} + \lambda_{Nz}^{(2)} (u_2)_{ijo}^{n+1} \\ (u_2)_{ij1}^{n+1} = \tilde{\beta}_1^{(2)} (u_2)_{ijo}^{n+1} + \tilde{\nu}_1^{(2)} + \tilde{\lambda}_1^{(2)} (u_2)_{ijNz+1}^{n+1} \\ (u_3)_{ij1}^{n+1} = \tilde{\beta}_1^{(3)} (u_3)_{ijo}^{n+1} + \tilde{\nu}_1^{(3)} \end{cases} \tag{4.6}$$

into the interfacial equations (4.4a) and (4.4b) to obtain $(u_1)_{ijNz+1}^{n+1}$, $(u_2)_{ijo}^{n+1}$, $(u_2)_{ijNz+1}^{n+1}$ and $(u_3)_{ijo}^{n+1}$. Finally, we solve for the rest of the unknowns in $\{(u_1)_{ijk}^{n+1}\}$, $\{(u_2)_{ijk}^{n+1}\}$ and $\{(u_3)_{ijk}^{n+1}\}$ by step 3 of Procedure 2.

The above iterations are continued until the steady state solution is obtained.

A similar numerical procedure for the two layer case can be obtained by Procedure 1 described in Section 3. It should be pointed out that the above procedure is one type of the domain decomposition methods for solving parabolic differential equations[9].

5. Numerical examples

To illustrate the advantage of the generalized Douglas ADI scheme, we first consider a three-dimensional parabolic differential equation.

$$\frac{\partial U}{\partial t} = \mu \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} \right), \qquad 0 < x, y, z < \pi, t > 0; \tag{5.1}$$

with its exact solution given as

$$U(x,y,z,t) = e^{-\mu t}(\sin(x)+\sin(y)+\sin(z)), \quad 0 \le x,y,z \le \pi, t > 0. \tag{5.2}$$

Initial and boundary conditions are obtained from the exact solution. From (5.2), it is seen that for large μ, the solution converges fast to the steady state solution (which is zero) as time increases. Hence, the equation in (5.1) has a fast transient phenomenon.

Equation (5.1) was solved by using the generalized ADI scheme (2.3). In our calculation, we first chose $\Delta x = \Delta y = \Delta z = h = \frac{\pi}{50}$ and $\Delta t = 0.01$. To obtain the steady state solution, the time iteration was continued until $\max|u| \le 10^{-3}$ was satisfied. The number of iterations for different μ values 10, 100 and 200 are compared with that of the Douglas ADI scheme (2.2), Table I.

ADI	n(μ = 10)	n(μ = 100)	n(μ = 200)
Douglas scheme	81	432	1,194
Scheme (2.3) (ε = 0.05)	81	50	40
Exact solution	81	9	5

Table I. Number of iterations, n (nΔt = t)

From Table I, it can be seen that the solution from the Douglas scheme converges very slowly when μ is large. Convergence of the numerical solutions from the generalized scheme was much faster than that from the Douglas scheme. This shows that the new scheme is suitable for simulating fast transient phenomena.

We now consider

$$\frac{\partial u}{\partial t} = \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right), \qquad 0 < x,y,z < 1, t > 0; \tag{5.3}$$

with its exact solution given as

$$u(x,y,z,t) = e^{-\mu\pi^2 t}(\sin(\pi x)+\sin(\pi y)+\sin(\pi z)),$$

$$0 \le x,y,z \le 1, t > 0. \tag{5.4}$$

We chose $\Delta t = 0.1$ with difference grid sizes, $h = 0.1$, 0.05 and 0.01. Again, to obtain the steady state solution (which is zero), the time iteration was continued until $\max|u| \le 10^{-3}$ was satisfied. The number of iterations for $\mu = 1.0$ is compared with that of the Douglas ADI scheme (2.2), Table II.

ADI	n(h = 0.1)	n(h = 0.05)	n(n = 0.01)
Douglas scheme	20	70	1,681
Scheme (2.3) (ε = 0.05)	16	31	59
Exact solution	9	9	9

Table II. Number of iterations, n (nΔt = t)

From Table II, it can be seen that the solution from the Douglas scheme converges very slowly when the spatial grid is fine. Convergence of the numerical solutions from the generalized scheme was much faster than that from the Douglas scheme. For h = 0.01, the Douglas scheme took 2,130 minutes of CPU time on the NCD workstation while the generalized Douglas scheme only took 60 minutes. This shows that the new scheme is suitable for computations on fine spatial meshes, which makes it applicable to micromanufacturing, such as the prediction of the temperature profile in x-ray lithography.

We now consider the three-dimensional heat conduction on three layers.

$$\frac{\partial U_1}{\partial t} = \mu_1 \left(\frac{\partial^2 U_1}{\partial x^2} + \frac{\partial^2 U_1}{\partial y^2} + \frac{\partial^2 U_1}{\partial z^2} \right), \tag{5.5a}$$

$$\frac{\partial U_2}{\partial t} = \mu_2 \left(\frac{\partial^2 U_2}{\partial x^2} + \frac{\partial^2 U_2}{\partial y^2} + \frac{\partial^2 U_2}{\partial z^2} \right), \tag{5.5b}$$

and

$$\frac{\partial U_3}{\partial t} = \mu_3 \left(\frac{\partial^2 U_3}{\partial x^2} + \frac{\partial^2 U_3}{\partial y^2} + \frac{\partial^2 U_3}{\partial z^2} \right). \tag{5.5c}$$

This situation may be encountered in x-ray lithography, used in micromanufacturing. The three layers are composed of a mask, a photoresist and a substrate. For this example, it is of interest to predict the temperature profile in each of the layers. Each layer is chosen to be of dimension 0.1(cm) × 0.1(cm) × 0.01(cm) as shown in Figure 2. The initial and boundary value

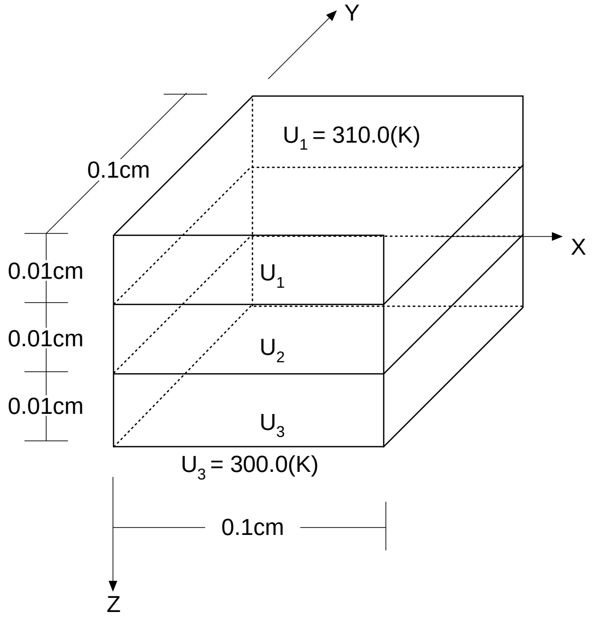

Figure 2.
Three-dimensional domain with three layers for the example of equation (5.5)

problem with homogeneous conditions (U_m = 300 (K), m = 1, 2, 3) except U_1 = 310.0 (K) at z = 0.0 is considered. Problem (5.5) with initial and boundary conditions is computed by using the numerical procedure described in Section 4. In this calculation, we chose $\Delta x = \Delta y = 0.002$ (cm) , $\Delta z = 0.0002$ (cm) and Δt = 0.01 (seconds). The contours of the temperature profile in the cross section at y = 0.05cm with various μ_1, μ_2 and μ_3 values when t = 1.0 (seconds) are plotted in Figures 3 and 4. The case $\mu_1 = \mu_2 = \mu_3 = 1.0$ (Wcm^2/J) with $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0.05$ is shown in Figure 3. When μ_3 = 10.0 (Wcm^2/J), it is seen that most of the heat in layer 3 was transferred to the outside, as shown in Figure 4.

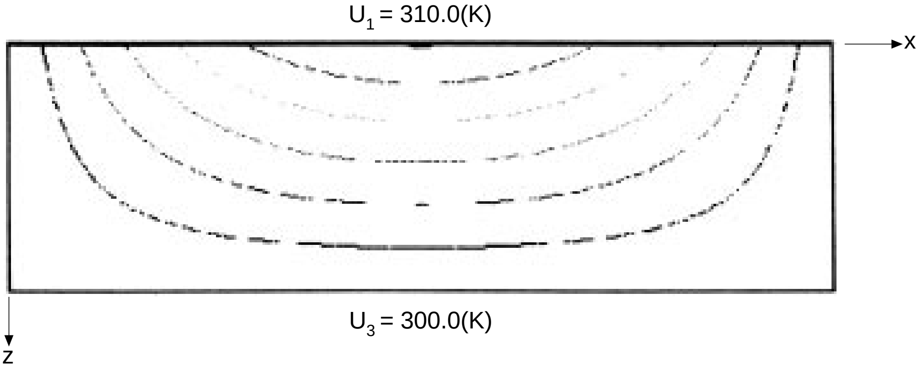

Figure 3. Contour of the temperature distribution in the cross section at y = 0.05cm when t = 1 second with $\mu_1 = \mu_2 = \mu_3 = 1.0$

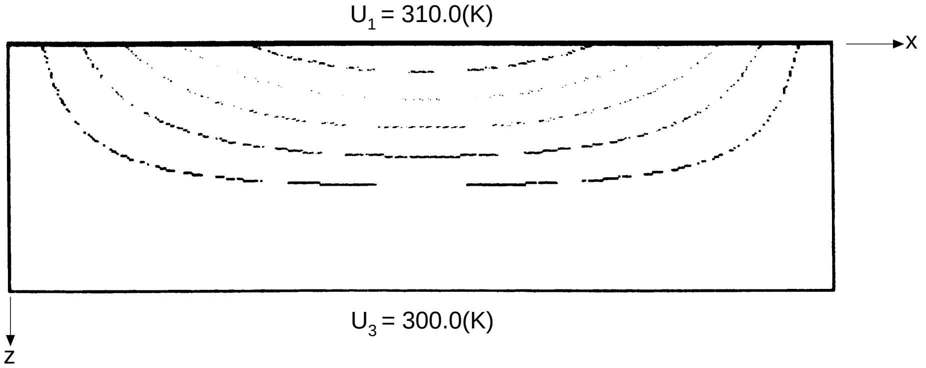

Figure 4. Contour of the temperature distribution in the cross section at y = 0.05cm when t = 1 second with $\mu_1 = \mu_2$ = 1.0 and μ_3 = 10.0

References

1. Ameel, T.A., Warrington, R.O., Yu, D. and Dahlbacka, G., “Thermal analysis of X-ray irradiated thick resists to determine induced structural deformations”, *Heat Transfer*, Vol. 4, 1994, pp. 313-8.
2. Cole, K.D. and McGahan, W.A., “Theory of multilayers heated by laser absorption”, *Journal of Heat Transfer*, Vol. 115, 1993, pp. 767-71.
3. Nassar, R. and Dai, W., “Mathematical modeling of micro engineering processes”, *Proceedings of Microsystems Technologies*, HARMST ’95, Germany, 1995.

4. Kant, R., “Laser-induced heating of a multilayered medium resting on a half-space: part I – stationary source”, *Journal of Applied Mechanics*, Vol. 55, 1988, pp. 93-7.

5. Madison, M.R. and McDaniel, T.W., “Temperature distributions produced in an N-layer by static or scanning laser or electron beam with application to magneto-optical media”, *J. Appl. Phys.*, Vol. 66, 1989, pp. 5738-48.

6. Samarskii, A.A. and Vabishchevich, P.N., “Regularized difference schemes”, *MMCE*, Vol. 1, 1994, pp. 44-50.

7. Dean, J. and Glowinski, R., “On some finite element methods for the numerical simulation of incompressible viscous flow”, in Gunzburger, M.D. and Nicolaides, R.A. (Eds), *Incompressible Computational Fluid Dynamics*, University Press, Cambridge, MA, 1993.

8. Ottega, J.M., *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, NY, 1988.

9. Chan, T.F., “Domain decomposition algorithms”, *Acta Numerica*, 1994, pp. 61-143.