The mathematical simulation of the liquid transport in a multilayered nonwoven

Raimondas Čiegis
Institute of Mathematics and Informatics
Akademijos 4, 2600 Vilnius, Lithuania
e-mail:rc@fm.vtu.lt

Aivars Zemitis
AGTM
Department of Mathematics
University of Kaiserslautern
67663 Kaiserslautern, Germany
e-mail:zemitis@itwm.uni-kl.de

September 24, 1997

Abstract

In this report we treat an optimization task, which should make the choice of nonwoven for making diapers faster. A mathematical model for the liquid transport in nonwoven is developed. The main attention is focused on the handling of full and partially saturated zones, which leads to a parabolic-elliptic problem. Finite-difference schemes are proposed for numerical solving of the differential problem. Parallel algorithms are considered and results of numerical experiments are given.

Contents

1 Introduction

It is shown in Alt and Luckhaus [1] that mathematical models of the liquid transport in multilayered nonwoven leads to an elliptic-parabolic equation since this problem becomes elliptic in the region of saturation. In the parabolic region equation is of nonlinear degenerate parabolic type and it can lead to a nonlinear convection-dominated diffusion equation, at least in some parts of the domain. In our model the convection is due to the gravity. Effects of multilayered liquid transport are also included into our mathematical model.

This paper deals with numerical methods for solving problems described above. Finite-difference schemes are proposed for 1D and 2D problems and algorithms for the realization these schemes are investigated. Parallel numerical methods are given and results of numerical simulations are discussed.

The rest of the paper is organized as follows. At first we present the mathematical formulation of the problem in Section 2. Finite-difference schemes for 1D problem are investigated in Section 3. In Section 4 numerical methods for 2D problems are described. Parallel numerical algorithms are considered in Section 5.

2 Formulation of the Problem

We consider the non-stationary filtration problem

$$
\begin{aligned}
M_l(w_l)\frac{\partial w_l}{\partial t} &= \nabla(K_l(w_l)\nabla\Phi_l(w_l)) \quad &&\text{in} \quad \Omega_l \times (0,T], &&(2.1)\\
w_l(x,0) &= w_{l0}(x) &&\text{in} \quad \Omega_l, &&(2.2)\\
\Phi_1(w_1) &= q(t) &&\text{on} \quad \partial\Omega_D \times [0,T], &&(2.3)\\
K_l(w_l)\nabla\Phi_l(w_l) &= 0 &&\text{on} \quad \partial\Omega\backslash\partial\Omega_D \times [0,T],
\end{aligned}
$$

where $l = 1,2,\dots L$, L is the number of layers, $\Omega = \bigcup\limits_{l=1}^{L}\Omega_l$ is a rectangular bounded domain in $\mathbb{R}^2$ with a boundary $\partial\Omega$, and $\partial\Omega_D$ is a regular subset of $\partial\Omega$. Here w_l represents the saturation (or water content), $\Phi_l(w_l) = P_l(w_l)+x_1$ is the piesometric head for the lth layer, P_l represents the pressure. We also require the pressure and the flux to be continuous on the interior boundaries

$$
\begin{aligned}
K_l(w_l)\frac{\partial P_l(w_l)}{\partial x_1} &= K_{l+1}(w_{l+1})\frac{\partial P_{l+1}(w_{l+1})}{\partial x_1} \quad \text{on } \partial\Omega_l, &&(2.4)\\
P_l(w_l) &= P_{l+1}(w_{l+1}). &&(2.5)
\end{aligned}
$$

Functions $K(w), P(w)$ approximate the known experimental data or are obtained from literature. In our numerical experiments we have used the permeability function $K_l(w)$ defined by

$$K_l(w) = k_l(\frac{w}{m_l})^{n_{kl}}, \tag{2.6}$$

where m_l represents the porosity. The pressure $P_l(w)$ is defined as follows

$$P_l(w) = -p_l(1-(\frac{w}{m_l})^{n_{pl}}) \quad \text{for } w \geq w_{0l}, \tag{2.7}$$

$$
\begin{aligned}
P_l(w) &= -h_m + \frac{h_m + P_l(w_{0l})}{w_{0l}^2} w(2w_{0l} - w) \\
&\quad + \frac{P_l'(w_0)}{w_{0l}} w(w - w_{0l}) \quad \text{for } w < w_{0l},
\end{aligned}
$$

where p_l, w_{0l}, h_m are given parameters. We assume that initial data is such that

$$
0 \le w_{l0}(x) \le m_l, \quad l = 1, 2, \ldots, L.
$$

In general we will require functions $K(w)$ and $p(w)$ to satisfy the following conditions

$$
\begin{aligned}
K_l(w) &> 0, \quad \frac{dK_l(w)}{dw} \ge 0 \quad \text{for } w > 0, \\
K_l(0) &= 0, \\
\frac{dp_l(w)}{dw} &> 0 \qquad \text{for } \ w \ge 0.
\end{aligned}
$$

Then it is easily proved that problem (2.1) – (2.5) defines a monotone space operator, this guarantees the boundedness of the solution

$$
0 \le w_l(x,t) \le m_l \quad \text{in } \Omega_l \times [0,T]. \tag{2.8}
$$

We see that the mathematical model (2.1) – (2.5) is described by a degenerate nonlinear parabolic equation. It leads to the finite speed of propagation of disturbances to undisturbed region (see Lions [18], Alt and Luckhaus [1]). It is well known that non degenerate parabolic equations possess a smoothing property and even L_2 data yield a smooth solution. This property is not valid for degenerate parabolic equations.

The equation (2.1) becomes elliptic in the region of saturation, where $p > 0, w_l = m_l$:

$$
\nabla(k_l \nabla P_l) = 0. \tag{2.9}
$$

Hence in general the equation (2.1) (and (2.9)) is an elliptic-parabolic equation. We include into equation (2.1) both of these cases by defining

$$
\begin{aligned}
M_l(w_l) &= m_l \quad \text{for } 0 \le w_l \le m_l, \ p \le 0, \\
M_l(w_l) &= 0 \quad\ \ \text{for } p > 0, \ w_l = m_l.
\end{aligned}
$$

The aim of this paper is to construct efficient numerical algorithms for solving nonlinear degenerated elliptic-parabolic problems. The error analysis of finite-difference and Galerkin schemes for singular parabolic and elliptic-parabolic problems are well developed know. The study of convergence began with the papers by Jerome and Rose [14], Rose [25,26] and was improved by Nochetto [21,22]. The most complete results are given by Nochetto and Verdi [23].

Schemes which are proposed in our paper can be investigated by using these methods and the method of convergence analysis of nonlinear non-stationary parabolic problem given by Čiegis [5]. We note that in all such error estimates the approximation error is multiplied with the stability constant e^{CT} and this makes these estimates of little value for practical comparing of their accuracy. This fact is stressed by Johnson, Rannacher and Boman [15] where methods for aposteriori estimation of stability constants are proposed for the Navier-Stokes problem. Hence this subject deserves further investigation and application of the method of computational experiment is one of the most promising approaches. We deal with two dimensional problems but our goal is to propose such schemes which can be effectively used for three dimensional problems, as well.

3 One Dimensional Problem

In this section we shall consider finite-difference schemes for one dimensional case of problem (2.1) – (2.5). As was stated above problem (2.1) – (2.5) includes various phenomena, such as nonlinear degenerated diffusion equation, nonlinear convection-dominated diffusion equation and singularity of elliptic-parabolic problem. Numerical solving of all these problems requires special attention. We shall give such analysis in this section.

3.1 Nonlinear degenerated diffusion

We consider the problem

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x}(k(u)\frac{\partial \varphi(u)}{\partial x}) \quad \text{in} \quad (a,b)\times(0,T], \qquad (3.1)\\ u(x,0) &= u_0(x) \qquad\qquad \text{in } (a,b),\\ u(a,t) &= g_1(t),\ u(b,t)=g_2(t) \qquad \text{on } [0,T]. \end{aligned}$$

Here $k(u)$ is a nonnegative diffusion coefficient which vanishes for $u=0$, i.e., $k(0)=0, \varphi(u)$ is a given function with $\varphi'(u)>0$. Let assume that initial data is such that $0 \le u_0(x), g_1(t), g_2(t) \le U_M$. The maximum principle for parabolic equations implies that a solution of (3.1) also satisfies the same bounds

$$0 \le u(x,t) \le U_M \quad \text{in } (a,b)\times(0,T]. \tag{3.2}$$

We also note that the problem (3.1) defines a monotone operator, i.e., if we take two different initial functions $u_{01}(x) \ge u_{02}(x)$ on $[a,b]$, then the respective solutions $u_1(x,t)$ and $u_2(x,t)$ satisfy

$$u_1(x,t) \ge u_2(x,t) \quad \text{in } [a,b]\times[0,T]. \tag{3.3}$$

Though the last result is known, we present its simple proof, as this analysis will be useful for investigation of finite-difference schemes. First we rewrite the equation (3.1) in the form

$$\frac{\partial u}{\partial t} = \frac{\partial^2 \Phi(u)}{\partial x^2}, \tag{3.4}$$

where

$$\Phi(u) = \int_0^u k(\xi)\varphi'(\xi)d\xi. \tag{3.5}$$

Let define $H(\xi) = \Phi^{-1}(\xi)$ for real ξ. Then it follows from (3.4) that function $v = \Phi(u)$ is a solution of the equation

$$\frac{\partial H(v)}{\partial t} = \frac{\partial^2 v}{\partial x^2}. \tag{3.6}$$

We have from (3.5) and from the assumptions on functions $k(u), \varphi(u)$ that $\Phi'(u) \ge 0$. Hence initial data $v_{0i}(x) = \Phi(u_{0i})$ satisfy the same inequality

$$v_{01}(x) \ge v_{02}(x).$$

Let define function $Z(x,t) = v_1(x,t) - v_2(x,t)$, which is a solution of the following problem

$$\begin{aligned}
\frac{\partial}{\partial t}(H(v_1) - H(v_2)) &= \frac{\partial^2 Z}{\partial x^2} && \text{in} \quad (a,b)\times(0,T], \qquad (3.7)\\
Z(x,0) = Z_0(x) &\ge 0 && \text{in} \quad (a,b),\\
Z(a,t) = 0,\ Z(b,t) &= 0 && \text{on} \quad [0,T].
\end{aligned}$$

Since $H(v)$ is also a monotone nondecreasing function and

$$H(v_1) - H(v_2) = H'(\tilde v)Z,$$

we get that $Z(x,t) \ge 0$.

Both properties (3.2) and (3.3) are very important in describing a solution of problem (3.1) hence we would like that a discrete solution of finite-difference scheme also be positive, bounded by U_M and monotone. In order to preserve the monotonicity we restrict to fully implicit backward Euler finite-difference schemes.

There are many finite-difference and finite-element schemes proposed to approximate the degenerate parabolic equation (3.1) (see Samarskij [27], Socolovsky [28], Čiegis [4] and the citations contained therein).

We consider two finite-difference schemes. First we discretize the $x-t$ region by choosing difference grids. Let ω_h be a space grid in the interval (a,b)

$$\omega_h = \{x_i : x_i = a + ih, \quad i = 1, 2, \ldots, N-1\}, \; x_N = b,$$

and ω_τ be a time grid

$$\omega_\tau = \{t_n : t_n = n\tau, \; n = 1, 2, \ldots, K, \; t_K = T\}.$$

We denote a discrete function $U_i^n = U(x_i, t_n)$. For simplicity of notation we suppress the subscript i and superscript n if no confusion is possible.

The first scheme is obtained by the balance method (see Samarskij [27])

$$\begin{aligned} \partial_t U = \partial_{\bar{x}}(K_{i+0.5}(U)\partial_x \varphi(U)) \qquad & \text{in} \qquad \omega_h, \\ U^0 = u_0(x) \qquad & \text{in} \qquad \omega_h, \\ U_0^n = g_1(t_n), \quad U_N^n = g_2(t_n) \qquad & \text{in} \qquad \omega_\tau. \end{aligned} \tag{3.8}$$

Here $K_{i+0.5}(U) = 0.5(k(U_{i+1}) + k(U_i))$ and we use the following notation of discrete derivatives

$$\begin{aligned} \partial_t U^n &= (U^n - U^{n-1})/\tau, \\ \partial_x U_i &= (U_{i+1} - U_i)/h, \quad \partial_{\bar{x}} U_i = (U_i - U_{i-1})/h. \end{aligned}$$

The second finite-difference scheme is obtained by approximating the equation (3.4):

$$\partial_t U = \partial_{\bar{x}} \partial_x \Phi(U) \quad \text{in} \quad \omega_h. \tag{3.9}$$

Using the assumptions on functions k and φ and applying the discrete maximum principle we prove that solutions of both schemes (3.8) and (3.9) are nonnegative and bounded

$$0 \le U(x,t) \le U_M \quad \text{in} \;\; \bar{\omega}_h \times \omega_\tau.$$

Theorem 1. *The finite difference scheme (3.9) is a monotone scheme.*

Proof. We rewrite equation (3.9) in the form

$$\partial_t H(V) = \partial_{\bar{x}} \partial_x V, \tag{3.10}$$

where $V = \Phi(U)$. Let assume that we have two solutions of (3.10) with different initial data $v_{0i}(x)$ and $v_{02}(x)$, such that

$$v_{01}(x) \geq v_{02}(x) \quad \text{on} \quad \omega_h.$$

We prove that $Z(x,t) = V_1(x,t) - V_2(x,t) \geq 0$ by using the method of mathematical induction. Assume that

$$Z(x, t_j) \geq 0 \qquad \text{for } j = 0, 1, \ldots, n-1. \tag{3.11}$$

From (3.10) we have

$$\frac{H(V_1^n(x)) - H(V_2^n(x))}{\tau} = \partial_{\bar{x}} \partial_x Z^n + \frac{H(V_1^{n-1}(x)) - H(V_2^{n-1}(x))}{\tau}.$$

Applying the mean value theorem we get

$$H'(\tilde{V}^n) Z^n - \tau \partial_{\bar{x}} \partial_x Z^n = H'(\tilde{V}^{n-1}) Z^{n-1},$$

and the desired inequality $Z^n \geq 0$ follows from the discrete maximum principle and the assumption (3.11). □

We can not prove that difference scheme (3.8) is also unconditionally monotone. For this scheme function $Z(x,t) = U_1(x,t) - U_2(x,t)$ satisfies the following equation

$$\partial_t Z = \partial_{\bar{x}}(K(U_1)\partial_x \varphi(Z)) + \partial_{\bar{x}}(K'(\tilde{U})\partial_x(\varphi(U_2))Z).$$

The convective term on the right side of the equality for general data may violate the maximum principle.

We compute the discrete solution of nonlinear finite difference schemes (3.8), (3.9) using the Newton method. At each iteration we solve a system of linear equations with a tridiagonal matrix. The iterative method can be applied directly to the degenerated nonlinear discrete problem and no regularization of the problem is needed.

Some iterates may violate the positivity and boundedness of the solution, hence we include post-processing of the iterates $\overset{s}{U}$ of the Newton method

$$\begin{aligned} \overset{s}{U} &:= 0 \qquad \text{if} \quad \overset{s}{U} < 0, \\ \overset{s}{U} &:= U_M \qquad \text{if} \quad \overset{s}{U} > U_M. \end{aligned}$$

These solution limiters are absent after convergence of iterations.

The Newton method can be applied to the difference scheme (3.9) rewritten in the form

$$\partial_t H(V) = \partial_{\bar{x}} \partial_x V.$$

But since $(H')^{-1}(0) = 0$ for degenerated problems, some regularization of the problem is needed in this case. One interesting realization of iterative method with such regularization is proposed by Jager and Kačur [13], and Kačur et al. [17] . Analysis of the asymptotic behavior of a discrete solution near the wave front is given in Čiegis [4].

At the end of this section we present results of numerical experiments.

Example 1. We solve the problem (see also numerical experiments in Kačur et al. [17], Čiegis [4])

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u^m}{\partial x^2} \quad \text{in } \Omega \times (0, T).$$

A solution of this problem was given analytically by Barenblatt [2] as follows:

$$U(x,t) = (t+1)^{-\frac{1}{m+1}} \{[1 - \frac{m-1}{2m(m+1)} (\frac{x}{(t+1)^{\frac{1}{m+1}}})^2]_+\}^{\frac{1}{m-1}}.$$

Here, $a_+ = a$ for $a \geq 0$ and $a_+ = 0$ for $a < 0$. We have used the following initial and boundary conditions

$$\begin{aligned} u(x,0) &= U(x,0), \\ u(0,t) &= U(0,t), \quad u(b,t) = 0, \end{aligned}$$

and b was taken sufficiently large to contain the support of u for $t \leq T$. We also include results obtained by the algorithm S4 which is defined by Kačur [17]. We present results for the case $m = 6$.

Comparing results for finite difference schemes (3.8), (3.9) and algorithm S4 at $T = 5$ we obtain the following Table 1. The L_2 error norms are given, where τ and h are the time and space steps, respectively. In all experiments we have used $\tau = h$.

The obtained results show that monotone scheme (3.9) gives a slightly better accuracy than scheme (3.8). Difference scheme (3.9) and algorithm S4 have the same accuracy and are practically different only in iterative methods. The number of iterations was two for algorithm S4 and 2 – 4 iterates in average for the scheme (3.9).

We note that algorithm S4 is an explicit scheme for our model problem. We obtained the same results with only one iterate. Hence this scheme, as all

h	Scheme (3.8)	Scheme (3.9)	Algorithm S4
0.1	0.0454	0.0207	0.0203
0.05	0.0280	0.0121	0.0119
0.025	0.0154	0.0072	0.0070
0.0125	0.0065	0.0059	0.0058

Table 1: L_2 *error norms for the Example 1*

Method	τ	$h=\tau/2$	$h=\tau/4$	$h=\tau/8$
Algorithm S4	0.1	0.0245	0.05314	1.3091
	0.05	0.0128	0.6779	1.7664
Scheme (3.9)	0.1	0.0252	0.0133	0.0132
	0.05	0.0103	0.0071	0.0086

Table 2: L_2 *error norms for the Example 1 with different combinations of* τ *and* h

explicit schemes, requires time steps τ for which the front of the solution moves not more than to one space mesh point. In Table 2 we present the L_2 error norms at $T = 1$ for different combinations of τ and h.

3.2 Convection-dominated diffusion equation

There we consider finite-difference schemes used to approximate convection terms of the problem(2.1). Let assume that we solve a problem with L layers. Let $\overline{\Omega}_h = \overline{\Omega}_{1h} \cup \overline{\Omega}_{2h} \cup \ldots \cup \overline{\Omega}_{Lh}$ be a space grid, where Ω_{lh} is a uniform grid in Ω_l with space step $h = h_l$

$$\Omega_{lh} = \{x_i : x_i = a_l + ih_l,\ i = 1, 2, \ldots N_l - 1\},\ x_N = b_l.$$

The interior boundary points are included into $\overline{\Omega}_h$ twice, since $a_l = b_{l-1}$ for $l = 2, 3, \ldots L$. Hence the total number of grid $\overline{\Omega}_h$ points is $N_1 + N_2 + \cdots + N_L + L$.

We consider two approximations of the equation (2.1). The first scheme is obtained by using the upwind technique to handle the convective term

$$m_l \partial_t W_l = \partial_{\bar{x}} \partial_x \Phi_l(W_l) + \partial_x K_l(W_l) \quad \text{in} \quad \Omega_{lh}, \tag{3.12}$$

where

$$\Phi_l(W) = \int_0^W K_l(\xi) P_l'(\xi) d\xi.$$

The continuity conditions at interior boundary points are approximated by

$$P_l(W_l(x_{N_l})) = P_{l+1}(W_{l+1}(x_0)), \quad l = 1, 2, \ldots, L-1, \tag{3.13}$$

$$\begin{aligned} &\frac{1}{2}(h_l m_l \partial_t W_l(x_{N_l}) + h_{l+1} m_{l+1} \partial_t W_{l+1}(x_0)) \\ &= \partial_x \Phi_{l+1}(W_{l+1}(x_0)) - \partial_{\bar{x}} \Phi_l(W_l(x_{N_l})) \\ &+ h_{l+1} \partial_x K(W_{l+1}(x_0)). \end{aligned} \tag{3.14}$$

The second scheme is obtained by using a central differences approximation of the convective term

$$m_l \partial_t W_l = \partial_{\bar{x}} \partial_x \Phi_l(W_l) + \partial_{\overset{\circ}{x}} K(W_l) \quad \text{on} \quad \Omega_{lh}, \tag{3.15}$$

there

$$\partial_{\overset{\circ}{x}} W = (W(x_{i+1}) - W(x_{i-1}))/2h.$$

The continuity condition (2.4) is replaced by

$$\begin{aligned} &\frac{1}{2}(h_l m_l \partial_t W_l(x_{N_l}) + h_{l+1} m_{l+1} \partial_t W_{l+1}(x_0)) = \partial_x \Phi_{l+1}(W_{l+1}(x_0)) \\ &- \partial_{\bar{x}} \Phi_l(W_l(x_{N_l})) + \frac{1}{2}[K_{l+1}(W_{l+1}(x_1)) \\ &+ K_{l+1}(W_{l+1}(x_0)) - K_l(W_l(x_{N_l})) - K_l(W_l(x_{N_l-1}))]. \end{aligned} \tag{3.16}$$

It is well known (see, e.g., Thomas [29]), that the approximation error of (3.12) is of order $O(\tau+h)$, while the difference scheme (3.13) approximates the differential problem (2.1) on the uniform space grid with the order $O(\tau+h^2)$.

On the other hand a solution of the upwind scheme (3.12) satisfies the maximum principle, if we assume that $k'(u) > 0$ for $u > 0$. This condition is very important for multilayered liquid transport problems, as was stated above.

We compute the discrete solution of the nonlinear finite-difference scheme (3.12) (or (3.15)) with the Newton method. A system of linear equations

$$A\overset{s}{Z} = F, \quad \overset{s}{W} = \overset{s-1}{W} + \overset{s}{Z}$$

must be solved at each iteration, where A is a tridiagonal matrix except the equations at interior boundary points. In such a point x_{N_l} we have two values for our function: z_i and z_{i+1}. In an interior boundary point two successive equations can be written in the following form

$$c_i z_i - b_i z_{i+1} = f_i,$$

$$-d_{i+1} z_{i-1} - a_{i+1} z_i + c_{i+1} z_{i+1} - b_{i+1} z_{i+2} = f_{i+1}.$$

N	Scheme (3.12)	Scheme (3.15)
15	0.0195	0.0169
30	0.0116	0.0026
60	0.0064	0.0006
120	0.0035	0.00013

Table 3: *Numerical errors for upwind and central differences approximations of the convective term*

We must slightly modify the standard factorization method. After determination of the factorization coefficients $\alpha_{i-1}, \beta_{i-1}$ such that

$$z_{i-1} = \alpha_{i-1} z_i + \beta_{i-1},$$

we can rewrite the $(i+1)$th equation in tridiagonal form

$$a_{i+1} := a_{i+1} + \alpha_{i-1} d_{i+1}, \quad f_{i+1} := f_{i+1} + \beta_{i-1} d_{i+1}, \quad d_{i+1} = 0.$$

Only two additional multiplications and additions must be performed at each interior boundary in comparison with the standard factorization method.

Now we describe some numerical experiments concerning the convection-diffusion problem (2.1). Our goal is to evaluate the convergence orders for both schemes (3.12) and (3.15).

Example 2. Consider the problem (2.1) with Ω having one layer with parameters

$$\begin{array}{llll} b = 0.12, & k_1 = 0.01, & n_k = 3.5, & p_1 = 0.005, \\ n_p = 5, & m_1 = 0.91, & w_0 = 0.7, & h_m = 0.04. \end{array} \tag{3.17}$$

If we take the parameter $\tau = 0.001$ then the approximation error in time is negligible in comparison with the space approximation error. In Table 3 we present errors of the discrete solution at $T = 7, x = 0.04$ for different values of parameter N.

Example 3. Consider problem (2.1) with two layers, where the parameters of the first layer are given by

$$\begin{array}{llll} b = 0.06, & k_1 = 0.0001, & n_k = 3.5, & p_1 = 0.02, \\ n_p = 5, & m_1 = 0.91, & w_0 = 0.7, & h_m = 0.04, \end{array}$$

and the parameters of the second layer are given by (3.17).

The most important consequence of the obtained numerical results is that due to the nonstandard continuity condition (2.5) the approximation (3.16) proves to be nonconservative. We propose to use the conservative monotone continuity condition (3.14) instead of (3.16) for the difference scheme (3.15). Numerical experiments confirmed the efficiency of such a combined difference scheme (3.14) – (3.15).

3.3 Nonlinear elliptic-parabolic problem

Let us consider problem (2.1) with a region of saturation,

$$m_l \frac{\partial w_l}{\partial t} = \frac{\partial}{\partial x}(K_l(w_l)\frac{\partial P_l(w_l)}{\partial x}) + \frac{\partial}{\partial x}K_l(w_l) \quad \text{in} \quad \Omega_{Par}, \tag{3.18}$$

$$\frac{\partial}{\partial x}(K_l(m_l)\frac{\partial P_l}{\partial x}) = 0 \quad \text{in} \quad \Omega_{Ell}, \tag{3.19}$$

where Ω_{Ell} is the region of saturation and $\Omega_{Par} = \Omega \backslash \Omega_{Ell}$.

In order to construct a homogeneous finite difference scheme in Ω we first replace (3.18) and (3.19) with a new equation

$$M_l(u_l)\frac{\partial u_l}{\partial t} = \frac{\partial}{\partial x}(\tilde{K}_l(u_l)\frac{\partial P_l(u_l)}{\partial x}) + \frac{\partial}{\partial x}\tilde{K}_l(u_l) \quad \text{in} \quad \Omega, \tag{3.20}$$

where the new coefficients satisfy the conditions

$$\begin{aligned} M_l(u_l) &= \begin{cases} m_l & \text{for} \quad u_l \le m_l, \\ 0 & \text{for} \quad u_l > m_l; \end{cases} \\ \tilde{K}_l(u_l) &= \begin{cases} K_l(u_l) & \text{for} \quad u_l \le m_l, \\ K_l(m_l) & \text{for} \quad u > m_l. \end{cases} \end{aligned} \tag{3.21}$$

A smooth replacement of $K_l(u_l)$ near the corner can be considered. We use the same parameterization (2.7) of the pressure $P_l = P_l(u)$ in the region of saturation, i.e., when $u > m_l$. A physical water content $w_l(x,t)$ is defined by

$$w_l(x,t) = \min(u_l(x,t), m_l).$$

Then a homogeneous finite-difference scheme is obtained after the approximation of the generalized differential equation (3.20). For example we apply the upwind difference scheme for convective terms:

$$M_l(U_l)\partial_t U_l = \partial_{\bar{x}}\partial_x Q_l(U_l) + \partial_x \tilde{K}_l(U_l), \tag{3.22}$$

N	τ	$P_0 = 0$	$P_0 = 0.1$	$P_0 = 0.3$
60	0.1	3.09	3.36	3.65
60	0.01	2.24	2.21	2.23
120	0.05	3.07	3.18	3.54
120	0.02	2.92	2.78	3.30

Table 4: Numbers of iterates for solutions of elliptic-parabolic problem

where

$$Q_l(U) = \int_0^U \tilde{K}_l(\xi) P_l'(\xi) d\xi. \tag{3.23}$$

The continuity equations at interior boundary points are approximated analogously. The system of nonlinear equations (3.22) is solved with the Newton method. At each iteration we solve a system of linear equations with a matrix of the same structure as in previous section.

We made numerical experiments concerning the efficiency of the proposed method. The two-layered nonwoven model is used as an example of the elliptic-parabolic problem. The parameters of the material are the same as in Example 3. The obtained results are given in Table 4. The averaged number of iterates needed to fulfill the accuracy $\varepsilon = 10^{-3}$ is presented for different values of the pressure P_0 prescribed at the boundary Ω_D. For $P_0 > 0$ the elliptic region of full saturation exists. The problem (3.22) is solved for $0 < t \leq T := 10$.

We see from the results of the numerical experiments that the number of iterates depends weakly on the value of pressure prescribed at the boundary.

4 Two Dimensional Problem

In this section we consider finite difference schemes for the 2D problem (2.1). There are many papers dealing with the 2D nonlinear degenerated diffusion (see, e.g., Berger et al. [3], Socolovsky [28], Rose [25]) and nonlinear elliptic-parabolic problem (see, e.g., Nochetto and Verdi [23]). We note that in all of them discretization in time is done by one of the variants of the Euler method. The obtained nonlinear system is usually solved by the nonlinear Gauss-Seidel method (see Nochetto and Verdi [23], White [30], Zlamal [31]). In order to get convergence of iterates as fast as the linear Gauss-Seidel method for a positive definite matrix the convection terms must be approximated explicitly. An implicit approximation of convection terms and application of the Newton method leads to a system of linear equations with a non-symmetrical matrix. Hence linear algebra part of the realization of 2D difference schemes requires much more computations in comparison with the 1D finite-difference schemes proposed in Section 3. The situation becomes even worse for 3D problems, since the number of iterates increases for larger systems of linear equations. Our goal is to investigate finite-difference schemes constructed based on splitting methods. The idea of splitting for linear and nonlinear parabolic problems is well-known (see Samarskij [27], Marchouk [19]). A novelty of our work is that we propose splitting schemes for nonlinear elliptic-parabolic problems.

4.1 LOD scheme for the unsaturated problem

We start from the unsaturated problem (2.1). It is a nonlinear convection-degenerated diffusion parabolic problem in the whole region Ω. This case is used as a benchmark for the investigation of the efficiency of the splitting method algorithms. Let Ω_h be a space grid which is obtained as a product of two 1D space grids $\Omega_h = \omega_{x_1h} \times \omega_{x_2h}$. Finally we introduce some additional notation concerning the time discretization. We denote by

$$\partial_t U^{n+j/2} = (U^{n+j/2} - U^{n+(j-1)/2})/\tau$$

the discrete time derivative at the jth split time step. We also use the following notation for discrete space derivatives

$$\begin{aligned}
\partial_{x_1} U &= (U(x_1+h, x_2) - U(x_1, x_2))/h, \\
\partial_{\bar{x}_2} U &= (U(x_1, x_2) - U(x_1, x_2 - h))/h.
\end{aligned}$$

Then, the Locally One Dimensional (LOD) scheme is given by

$$m_l \partial_t W_l^{n+1/2} = \partial_{\bar{x}_1} \partial_{x_1} Q_l(W_l^{n+1/2}) + \partial_{x_1} K_l(W_l^{n+1/2}), \tag{4.1}$$

$$m_l \partial_t W_l^{n+1} = \partial_{\bar{x}_2} \partial_{x_2} Q_l(W_l^{n+1}). \tag{4.2}$$

It is easy to prove that

(P1) The approximation error of the LOD scheme is of order $O(\tau + h)$;

(P2) The solution of the LOD scheme satisfies the discrete maximum principle, i.e., $0 \le W^{n+j/2} \le U_M$ on Ω_h;

(P3) The LOD scheme is monotone.

N_{x_1}	N_{x_2}	t_{LOD}	t_{1D}
60	20	209 sec	7 sec
60	40	410 sec	7 sec
120	20	412 sec	14 sec
120	40	810 sec	14 sec

Table 5: *CPU times required to solve Example 4 using the LOD scheme*

All these results follow directly from the analysis analogous to that given in Section 3 for the 1D problem. We are now interested in the implementation of the LOD scheme. We compute the discrete solution of the LOD scheme by solving N_{x_1} 1D nonlinear convection-diffusion problems at the first split step and N_{x_2} 1D nonlinear diffusion problems at the second split step. Each subproblem is solved efficiently using the algorithm proposed in Section 3.

(P4) The algorithm can be extended straightforwardly to three dimensional multilayered nonwoven models.

At the end of this section we shall discuss the total amount of computations required in the implementation of LOD scheme. The ratio of floating-point operations to one grid node increases approximately twice in comparison with 1D finite-difference scheme.

Example 4. Consider the 2D two-layered nonwoven model problem. Both layers are rectangular and have parameters given in the Example 3. The layers are homogeneous in the x_2 direction. Let $\Omega = (0; 0.12) \times (0; 0.5), 0 < t \leq T := 100$. The Dirichlet boundary conditions are assigned on the interval $\Omega_D = [0.15, 0.35]$.

The discrete time step $\tau = 0.1$ is used in numerical experiments. Let us denote by t_{LOD} the CPU time in seconds required to solve this problem using the LOD scheme. Table 5 shows the results obtained with different numbers of grid points N_{x_1}, N_{x_2}. We also present the CPU time t_{1D} required to solve 1D discrete problem on the grid ω_h with N_{x_1} points.

4.2 ADI scheme for elliptic-parabolic problem

The method of Section 3.3 for the construction homogeneous finite-difference schemes for solving elliptic-parabolic problems can not be directly extended to the LOD scheme. The main difficulty is that the LOD scheme approximates the differential equation only in an *additive sense*. We have following approximation errors at each fractional time step (see Samarskij [27])

$$\begin{aligned}\Psi^{n+j/2} &= \overset{\circ}{\Psi}{}^{n+j/2} + \overset{*}{\Psi}{}^{n+j/2}, \quad j=1,2,\\ |\overset{\circ}{\Psi}{}^{n+j/2}| &= O(1), \quad |\overset{*}{\Psi}{}^{n+j/2}| = O(\tau+h^2),\end{aligned}$$

and additive approximation means that

$$\overset{\circ}{\Psi}{}^{n+1/2} + \overset{\circ}{\Psi}{}^{n+1} = 0.$$

Hence the LOD scheme can not be used for solving elliptic problems. We note that many classical approximations, e.g., the explicit Euler method, give iterative methods for solving elliptic problems.

There are economical difference schemes which also can be used as an iterative method for solving elliptic problems. We call a scheme *economical* if its implementation can be reduced to the solution of 1D discrete parabolic problems. In this section we consider the Alternating Direction Implicit (ADI) scheme (see Samarskij [27], Thomas [29])

$$\begin{aligned}2m_l\partial_t W_l^{n+1/2} &= \partial_{\bar{x}_1}\partial_{x_1}Q_l(W_l^{n+1/2}) + \partial_{x_1}K_l(W_l^{n+1/2}) \\ &\quad +\partial_{\bar{x}_2}\partial_{x_2}Q_l(W_l^n),\end{aligned} \tag{4.3}$$

$$\begin{aligned}2m_l\partial_t W_l^{n+1} &= \partial_{\bar{x}_1}\partial_{x_1}Q_l(W_l^{n+1/2}) + \partial_{x_1}K_l(W_l^{n+1/2}) \\ &\quad +\partial_{\bar{x}_2}\partial_{x_2}Q_l(W_l^{n+1}).\end{aligned} \tag{4.4}$$

At each inner boundary layer, we have slightly different discrete equations:

$$\begin{aligned}&\frac{1}{2}[h_l m_l \partial_t W_l^{n+j/2}(x_{1N_l}) + h_{l+1}m_{l+1}\partial_t W_{l+1}^{n+j/2}(x_{10})] \\ &= \partial_{x_1}Q_{l+1}(W_{l+1}^{n+1/2}(x_{10})) \\ &-\partial_{\bar{x}_1}Q_l(W_l^{n+1/2}(x_{1N_l})) + h_{l+1}\partial_{x_1}K(W_{l+1}^{n+1/2})(x_{10}) \\ &+\frac{1}{2}[h_{l+1}\partial_{\bar{x}_2}\partial_{x_2}Q_{l+1}(W_{l+1}^{n+j-1}) + h_l\partial_{x_2}Q_{l+1}(W_{l+1}^{n+j-1})],\end{aligned} \tag{4.5}$$

$$P_l(W_l^{n+j/2}(x_{1N_l})) - P_{l+1}(W_{l+1}^{n+j/2}(x_{1N_0})) = 0 \tag{4.6}$$

for $j=1,2$. In order to obtain an efficient implementation algorithm it is convenient to rewrite (4.3), (4.4) in equivalent form. Let

$$V_l^n = W_l^n + \frac{\tau}{2m_l}\partial_{\bar{x}_2}\partial_{x_2}Q_l(W_l^n).$$

The substitution of V_l^n into (4.3) leads to the equation

$$2m_l \frac{W_l^{n+1/2} - V_l^n}{\tau} = \partial_{\bar{x}_1}\partial_{x_1} Q_l(W_l^{n+1/2}) + \partial_{x_1} K_l(W_l^{n+1/2}).$$

It is easy to see that the same substitution is appropriate at the inner boundary points. Now let

$$V_l^{n+1/2} = W_l^{n+1/2} + \frac{\tau}{2m_l}(\partial_{\bar{x}_1}\partial_{x_1} Q_l(W_l^{n+1/2}) + \partial_{x_1} K_l(W_l^{n+1/2})). \tag{4.7}$$

Substitution of $V_l^{n+1/2}$ into (4.4) leads to the equation

$$2m_l \frac{W_l^{n+1} - V_l^{n+1/2}}{\tau} = \partial_{\bar{x}_2}\partial_{x_2} Q_l(W_l^{n+1}).$$

We need to investigate these equations at the inner boundary points. Since the equality

$$\frac{h_l}{h_l + h_{l+1}} + \frac{h_{l+1}}{h_l + h_{l+1}} = 1$$

holds, we get two equations

$$h_l[2m_l \frac{W_l^{n+1}(x_{1N_l}) - V_l^{n+1}(x_{1N_l})}{\tau} - \partial_{\bar{x}_2}\partial_{x_2} Q_l(W_l^{n+1}(x_{1N_l}))] \tag{4.8}$$
$$+h_{l+1}[2m_{l+1} \frac{W_{l+1}^{n+1}(x_{10}) - V_{l+1}^{n+1}(x_{10})}{\tau} - \partial_{\bar{x}_2}\partial_{x_2} Q_{l+1}(W_{l+1}^{n+1}(x_{10}))] = 0,$$
$$P_{l+1}(W_{l+1}^{n+1}(x_{10})) - P_l(W_l^{n+1}(x_{1N_l})) = 0. \tag{4.9}$$

We solve the system of nonlinear equations (4.8) – (4.9) using the Newton method. The corrections at each iterate

$$z_j = \overset{s+1}{W}_l(x_{1N_l}, x_{2j}) - \overset{s}{W}_l(x_{1N_l}, x_{2j}),$$
$$Z_j = \overset{s+1}{W}_{l+1}(x_{10}, x_{2j}) - \overset{s+1}{W}_{l+1}(x_{10}, x_{2j})$$

satisfy the following system of linear equations

$$h_l(-a_j z_{j-1} + c_j z_j - b_j z_{j+1} - f_j) \tag{4.10}$$
$$+h_{l+1}(-A_j Z_{j-1} + C_j Z_j - B_j Z_{j+1} - F_j) = 0,$$
$$D_j Z_j - d_j z_j = E_j, \quad j = 0, 1, \ldots, N_{x_2}. \tag{4.11}$$

Then, solving Z_j from (4.11) and substituting the obtained equalities into (4.10) we get a system of linear equations with tridiagonal matrix for the unknowns z_j.

Remark 1. The substitution formula (4.7) can be replaced with a computationally more simple one

$$V_l^{n+1/2} = W_l^{n+1/2} + W_l^{n+1/2} - V_l^n = 2W_l^{n+1/2} - V_l^n. \tag{4.12}$$

It is well known that the ADI scheme approximates the differential problem (2.1) with second order in time, i.e., $\Psi = O(\tau^2 + h^2)$, and it is unconditionally stable in the linear case (see, e.g., Thomas [29]). Due to the last property the ADI scheme is also used as an iterative solver of elliptic problems. This method is known in the literature as the *Alternating Direction Iterative* method.

Let us assume that there exists a region of saturation $\Omega_{Ell}(t)$. In order to construct a homogeneous finite-difference scheme in Ω, we follow the analysis given in Section 3 and replace equations (2.1) and (2.9) with the new equation

$$\begin{aligned} M_l(u_l)\frac{\partial u_l}{\partial t} &= \frac{\partial}{\partial x_1}(\tilde{K}_l(u_l)\frac{\partial P_l(u_l)}{\partial x_1}) + \frac{\partial}{\partial x_1}\tilde{K}_l(u_l) \\ &+\frac{\partial}{\partial x_2}(\tilde{K}_l(u_l)\frac{\partial P_l(u_l)}{\partial x_2}) \quad \text{on} \quad \Omega \times (0,T]. \end{aligned} \tag{4.13}$$

Then, we extend the ADI scheme to the generalized problem (4.13) in the following form:

$$\begin{aligned} m_l\frac{U_{l,s}^{n+1/2} - U_{l,s-1}^n}{2\tau} &= \partial_{\bar{x}_1}\partial_{x_1}Q_l(U_{l,s}^{n+1/2}) + \partial_{x_1}\tilde{K}_l(U_{l,s}^{n+1/2}) \\ &+\partial_{\bar{x}_2}\partial_{x_2}Q_l(U_{l,s-1}^n), \end{aligned} \tag{4.14}$$

$$\begin{aligned} m_l\frac{U_{l,s}^{n+1} - U_{l,s}^{n+1/2}}{2\tau} &= \partial_{\bar{x}_1}\partial_{x_1}Q_l(U_{l,s}^{n+1/2}) + \partial_{x_1}\tilde{K}_l(U_{l,s}^{n+1/2}) \\ &+\partial_{\bar{x}_2}\partial_{x_2}Q_l(U_{l,s}^{n+1}) \end{aligned} \tag{4.15}$$

for $s = 1, 2, \ldots,$ where

$$U_{l,s+1}^n = \begin{cases} U_l^n & \text{if } U_{l,s}^{n+1} \le m_l, \\ U_{l,s}^{n+1} & \text{if } U_{l,s}^{n+1} > m_l, \end{cases}$$

$U_{l,0}^n = U_l^n$. Equations (4.14), (4.15) define the *outer* iterative method (or *elliptic* iterations), while the Newton method for solving (4.14), (4.15) defines *inner* iterations (or *nonlinear* iterations).

In this section we have demonstrated the basic approach how to construct homogeneous economical finite-difference schemes for solving multidimensional elliptic-parabolic problems.

Let us mention two main limitations of the proposed method. Firstly, the ADI scheme is monotone only conditionally. Let λ_1 and λ_2 be eigenvalues of the discrete operators L_1 and L_2 which are used in the definition of splitting scheme (4.3), (4.4). Then the *stability function* of the ADI scheme is given by

$$q = \frac{(1+\frac{1}{2}\tau\lambda_1)(1+\frac{1}{2}\tau\lambda_2)}{(1-\frac{1}{2}\tau\lambda_1)(1-\frac{1}{2}\tau\lambda_2)}. \tag{4.16}$$

We see, that $|q| < 1$ unconditionally for $\lambda_1, \lambda_2 < 0$ but $q > 0$ only if

$$\tau \leq 2/\max_j |\lambda_j|.$$

Such a relation can be very restrictive for many problems, since $\lambda_j = O(h^{-2})$.

Remark 2. The behavior of the stability function (4.16) is quite similar to the behavior of the Crank-Nicolson scheme stability function

$$q = \frac{1+\frac{\tau}{2}(\lambda_1+\lambda_2)}{1-\frac{\tau}{2}(\lambda_1+\lambda_2)}.$$

Both stability functions differ only in terms of order $O(\tau^2)$.

Note that a possibility for the solution of the ADI scheme not to satisfy the discrete maximum principle follows directly from the substitution formula (4.12).

Secondly, the extension of an ADI scheme to the 3D problems is possible, but the scheme is only conditionally stable with $\tau = O(h^2)$.

4.3 Stability-correction scheme for elliptic-parabolic problem

Here we consider one more economical splitting scheme. It also approximates the differential equation in the classical sense and has stability properties similar to the backward Euler scheme.

We approximate the generalized elliptic-parabolic problem with the following Stability-Correction (SC) scheme:

$$m_l \frac{U_{l,s}^{n+1/2} - U_{l,s-1}^{n}}{\tau} = \partial_{\bar{x}_1}\partial_{x_1} Q_l(U_{l,s}^{n+1/2}) + \partial_{x_1}\tilde{K}_l(U_{l,s}^{n+1/2}) + \partial_{\bar{x}_2}\partial_{x_2} Q_l(U_{l,s-1}^{n}), \tag{4.17}$$

$$m_l \frac{U_{l,s}^{n+1} - U_{l,s}^{n+1/2}}{\tau} = \partial_{\bar{x}_2}\partial_{x_2} Q_l(U_{l,s}^{n+1}) - \partial_{\bar{x}_2}\partial_{x_2} Q_l(U_{l,s-1}^{n}). \tag{4.18}$$

N_{x_1}	N_{x_2}	t_{SC}
60	20	960 sec
60	40	2176 sec
120	40	1625 sec

Table 6: *CPU times for the SC scheme*

Inner boundary layer points are treated similar to the analysis of the ADI scheme.

In order to obtain the efficient implementation algorithm we use the substitution

$$V_l^n = U_l^n + \frac{\tau}{m_l}\partial_{\bar{x}_2}\partial_{x_2}Q_l(U_l^n)$$

in (4.17) and

$$V_l^{n+1/2} = U_l^{n+1/2} + U_l^n - V_l^n$$

in (4.18). It is well known that the approximation error of the SC scheme is of order $O(\tau + h^2)$. The stability function of the SC scheme is given by

$$q = \frac{1 + \tau\lambda_1\lambda_2}{(1-\tau\lambda_1)(1-\tau\lambda_2)}.$$

It is easy to see that $0 < q < 1$ holds unconditionally for $\lambda_j < 0$.

Remark 3. The extension of the SC scheme to 3D problems is obvious, since only an additional stability correction step must be added with respect to x_3 coordinate.

At the end of this section we describe some numerical results. Consider a 2D two-layered nonwoven model. We choose Ω and the parameters of the model as in Example 4. The Dirichlet data $P = 0.3$ is assigned on Ω_D, hence there exists the region of saturation $\{P > 0\}$. Elliptic iterates were computed with accuracy $2 \cdot 10^{-3}$. Table 6 shows the CPU times t_{SC} required to solve the given problem. These results can be compared with CPU times given in Table 5 for the LOD scheme which is used as a benchmark in this paper.

We see that the proposed difference scheme works efficiently for computing the numerical solution of elliptic-parabolic problems.

5 Parallel Numerical Algorithms

In this section we consider parallel version of the SC scheme. Our main goal is to focus on algorithms targeted for parallel computers with distributed memory or for clusters of workstations. Hence we will use message-passing models of parallel computation in our numerical experiments, in particular MPI and PVM models (Geits et al. [9], Gropp, Lusk and Skjellum [12]). We will give much attention to the investigation of parallel performance of our algorithms. It is well known that for parallel computers with distributed memory the most important characteristic is the ratio of computation to communication times. Let T_{comm} be the time which is needed to communicate n floating point numbers. It can be estimated by

$$T_{comm} = s + rn, \tag{5.1}$$

where s is the latency or startup time, r is the inverse rate of the communication (Freeman and Phillips [8]). We note that $s >> r$ for many parallel computers with distributed memory, e.g., clusters of workstations.

5.1 Parallel version of the SC algorithm

To parallelize the SC algorithm we must distribute the data across the processors. Let us assume that we have p processors. Let T_0 be the time required to solve a given problem with the best serial algorithm and T_p be the time required to solve the same problem with a parallel algorithm on p processors. Then, the *speedup* for p processors is usually defined as

$$S_p = \frac{T_0}{T_p}.$$

In order to obtain high speedups we must preserve a good computation and communication load balancing among processors. Here we assume that the processors are homogeneous. The work load balancing problem for solving 3D heat equation with the LOD scheme on non-homogeneous cluster of workstations is considered in Čiegis, Šimkevičius and Wasniewski [6].

Several approaches are possible for the decomposition of Ω_h (see Gropp, Luck and Skjellum [12]). In the first decomposition the computational domain is sliced into slabs and computations on each slab are handled by different processes. We choose the direction x_1 for such a decomposition.

As an alternative, a two-dimensional Cartesian decomposition of the computational domain Ω_h can be used.

At least three factors must be taken into account to select between these two alternatives:

- the possibility to get a good load balancing of computations;

- the minimization of data communication time;
- the *topology* of the computer.

The last factor mostly depends on the details of the underlying hardware. We do not investigate this problem. The extensive literature on this subject exists (see Gilbert, Schreiber [10], Mehrotra et al. [20]).

Let us consider the other two factors. Using formula (5.1) for the estimation of the communication time we get that data communication in 1D decomposition takes roughly

$$T_{1D,COMM} = 2(s + rN_{x_2}) \tag{5.2}$$

time. Let us assume that $p = m^2$ and let there be m processors in each of the directions. Then, with exception of the processes on edges of the domain Ω_h, the time of communications in one iterate in 2D decomposition is given by

$$T_{2D,COMM} = 4(s + r\frac{N}{\sqrt{p}}), N = N_{x_1} = N_{x_2}, \tag{5.3}$$

see Gropp, Luck and Skjellum [12]. Two important consequences follow from these estimates.

The first result is that the communication algorithm is serial in the 1D decomposition case and no speedup is obtained by enlarging the number of processors.

The second implication is that the communication startup time is twice larger in the 2D decomposition case. We note that the latency times for clusters of workstations are very large. Hence, the total time spent in sending a message is largely determined by the startup time unless the messages are very long, i.e., the number of grid points in one direction is large. In view of the goal to minimize the data communication time we give priority to the 1D decomposition.

Now we consider the load balancing of computations. The class of problems we are dealing with has non-uniformly distributed computational requirements. Firstly, more iterates are performed during the first splitting step in the x_1 direction in comparison with the second splitting step. Secondly, the number of iterates is not the same for different grid layers. Hence, again the 1D decomposition enables us to obtain a uniform workload distribution among all p processors during the first splitting step, which is the most work consuming.

Remark 4. 1D decomposition is also more convenient if a load balancing of computations must be obtained on non-homogeneous cluster of processors.

5.2 Parallel algorithm for solving equations (4.17)

In this section we consider algorithms for parallel implementation of the first splitting step of the SC scheme. In the case of 1D decomposition the second splitting step (4.18) constitutes N_1 tridiagonal systems distributed over p processors. The entire matrix of any tridiagonal system is stored in the same processor and no data transfer is needed.

The simplest parallel implementation of the SC scheme can be obtained by using *pipelining* (see Ortega [24], Demmel, Heath and Vorst [7]). The forward factorization sweep is started by one processor with only one system. As soon as this processor is finished with its part of forward factorization for the first system it sends the information to its neighbor and starts the forward factorization sweep for the second system, while the second processor starts working on its part of the first system. Thus after $p-1$ steps all processors will be active. The backward factorization sweep is similar. Hence the total amount of computation is not increased in comparison with the best serial algorithm. We note two main drawbacks of the algorithm. Firstly, the load balancing decreases with the increased number of processors. Secondly, such algorithm requires sending N_2 data messages from each processor during realization of one iterate. Hence it is not efficient on parallel computers with distributed memory due to high start-up times.

We use a modification of Wang's subtracting Gaussian elimination algorithm (see Johnson, Saad and Schultz [16], Golub and Loan [11]). The algorithm consists of four phases.

In the *first* phase a forward factorization sweep is performed with no interprocessor communication. In the kth processor the forward elimination is carried out on local equations in order to eliminate the sub-diagonal elements

$$U_i = \gamma_i U_{i_k} + \alpha_i U_{i+1} + \beta_i \quad i = i_k + 1, \dots, i_{k+1}. \tag{5.4}$$

The standard Gaussian elimination formula must be changed for equations at the inner boundary layer points. Let us denote these two equations

$$C_{j-1} U_{j-1} - B_{j-1} U_j = F_{j-1}, \tag{5.5}$$

$$-P_j U_{j-2} - A_j U_{j-1} + C_j U_j - B_j U_{j+1} = F_j. \tag{5.6}$$

Assume that the factorization (5.4) is carried out for $i \le j-2$. Then we get from (5.5)

$$\alpha_{j-1} = \frac{B_{j-1}}{C_{j-1}}, \quad \beta_{j-1} = \frac{F_{j-1}}{C_{j-1}}, \quad \gamma_{j-1} = 0.$$

Equation (5.6) is modified before the forward elimination step

$$-P_j \gamma_{j-2} U_{i_k} - A'_j U_{j-1} + C_j U_j - B_j U_{j+1} = F'_j,$$

where

$$A'_j = A_j + P_j \alpha_{j-2}, \quad F'_j = F_j + P_j \beta_{j-2}.$$

We obtain the following factorization coefficients

$$\alpha_j = \frac{B_j}{D}, \quad \beta_j = \frac{F'_j + A'_j \beta_{j-1}}{D},$$
$$\gamma_j = \frac{P_j \gamma_{j-2}}{D}, \quad D = C_j - A'_j \alpha_{j-1}.$$

Remark 5. The last processor carries out the forward elimination in the opposite direction. The factorization algorithm is simplified for the first and the last processors, since $\gamma_i = 0$.

In the *second* phase a backward elimination is carried out to eliminate the super-diagonal elements in rows $i_{k+1} - 1$ through $i_k + 1$ for processors $k = \{2, 3, \ldots, p-1\}$. These $p - 2$ processors compute in parallel and no interprocessor communication is required. After this phase the system of linear equations is given in the following form

$$U_i = \gamma_i U_{i_k} + \delta_i U_{i_{k+1}} + \varphi_i.$$

In the *third* phase the processors communicate the first and the last equations to the master processor. After implementation of the same backward factorization as in the second phase we get an independent tridiagonal system of $p - 1$ equations

$$U_{i_1} - B_1 U_{i_2} = F_1,$$
$$-A_j U_{i_{j-1}} + U_{i_j} - B_j U_{i_{j+1}} = F_j, \quad j = 2, \ldots, p-2,$$
$$-A_{p-1} U_{i_{p-2}} + U_{i_{p-1}} = F_{p-1}.$$

This system is solved using a standard Gaussian factorization algorithm and the vectors $\{U_{i_j}, j = 1, 2, \ldots, p-1\}, \{U_{i_j+1}, j = 1, 2, \ldots, p-1\}$ are broadcasted to all processors.

In the *last* phase all processors calculate their part of solution simultaneously with no communication between processors.

We note that after the implementation of these four stages each processor also has a solution corresponding to the first and the last equations of its neighbors, respectively. Hence a new iterate will start without additional communication between processors.

Remark 6. The total amount of computation of the parallel Gaussian elimination algorithm is increased twofold in comparison with the serial factorization algorithm. But this drawback is not very important since the factorization in the first splitting step takes less then 2% of the whole computation time.

5.3 Block-version of the parallel algorithm

Modern computers are classified as non-uniform memory access machines. In distributed memory machines or clusters of workstations each processor has its own local memory which presents the upper level of memory. The aggregate off – processor memory of all other processors presents the lower level of the memory hierarchy. High performance on such machines can be obtained by minimizing time-consuming accesses to the lower level of memory. A general approach to ensure this goal is to use block-partitioned algorithms. A well-known example is given by the LAPACK library.

In order to reduce the frequency of communication between processors, each phase of the parallel factorization algorithm is implemented on a block of M systems. It is typical for LAPACK routines that a larger block size results in greater load imbalance. There is no such dependence of load balance on the block size in our case. But a straightforward implementation of the block version of parallel factorization algorithm increases the total amount of computations, as the same number of iterates is implemented for all M systems. Therefore a tradeoff between increased work and communication startup costs can be controlled by varying the block size.

We use a more complex implementation of the block parallel factorization algorithm. The block size is defined to be N_2. A vector is used to store the information on systems which must be solved during the realization of nonlinear iteration phase.

5.4 Performance of the parallel algorithm

Here we will present some results of numerical experiments. Computations were done on a cluster of workstations which consisted of up to 5 processors. Table 7 gives computational time T_C and speed-up S_p as a function of the number of processors, while keeping the grid sizes fixed at $N_1 = 280, N_2 = 100$. The Example 4 was used as a test problem.

It is also interesting to investigate the scalability of the parallel factorization algorithm. Table 8 shows the speed-up as a function of the number of processors, while keeping the matrix size per processor fixed at $N_1/p = 50, N_2 = 100$. /clearpage

5.5 Conclusions

Numerical methods for the simulation of liquid transport in multilayered nonwovens are developed. The fully and partially saturated zones can be treated.

p	T_C	S_p	E_p
2	1594 sec	1.78	0.89
3	1128 sec	2.52	0.84
4	912 sec	3.12	0.78
5	812 sec	3.5	0.70

Table 7: *Computational time and speed-ups for a discrete problem of the fixed size.*

p	S_p	E_p
2	1.70	0.85
3	2.40	0.80
4	2.95	0.74
5	3.45	0.69

Table 8: *The speed-up obtained when the matrix size per processor is fixed.*

The corresponding parallel algorithms are discussed. These algorithms can be used as a base for design optimization of hygiene products.

References

[1] H.W. Alt, S. Luckhaus, *Quasilinear Elliptic-Parabolic Differential Equations*, Math. Z. , **183**(1983), 311 – 341.

[2] G.I. Barenblatt, *On certain non-stationary motions of liquid and gases in porous media*, Prikl. Mat. Mech., **16** (1952), 67 – 78.

[3] A.E. Berger, H. Brezis and J. Rogers, *A numerical method for solving the problem* $u_t - \Delta f(u) = 0$, RAIRO Numer. Anal., **13** (1979), 297 – 312.

[4] R. Čiegis, *Accuracy of difference schemes for quasilinear parabolic problems*, Lith. Math. J., **26** (1986), 88 – 93.

[5] Raim. Čiegis, Rem. Čiegis and M. Meilūnas, *On one general method for the investigation of difference schemes*, Lith. Math. J., **36** (1996), 281 – 302.

[6] R. Čiegis, J. Šimkevičius and J. Wasniewski, *Running finite difference schemes for 3D diffusion problems on parallel computers with distributed memory*, Informatica, **7** (1996), 295 – 310.

[7] J.W. Demmel, M.T. Heath and H.A. van der Vorst, *Parallel numerical linear algebra. Acta Numerica*, **7** (1993), 111 – 197.

[8] T.L. Freeman and C. Phillips, *Parallel Numerical Algorithms*, Prentice Hall, New York, London, Toronto, Sydney, Tokyo, Singapore, 1991.

[9] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM* 3.0 *User's Guide and Reference Manual*. Tennessee, 1993.

[10] J. Gilbert, R. Schreiber, *Optimal data placement for distributed memory architectures*, In Parallel Processing for Scientific Computing, SIAM, Mach, 1991, 462 – 471.

[11] G.H. Golub and Ch. F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, 1991.

[12] W. Gropp, E. Lusk and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, Cambridge, Massachusetts, London, England, 1994.

[13] W. Jager and J. Kačur, *Solution of porous medium type systems by linear approximation schemes*, Numer. Math., **60**(1991), 407 – 427.

[14] J.W. Jerome and M. Rose, *Error estimates for the multidimensional two-phase Stefan problem*, Math. Comp. , **39**(1982), 489 – 514.

[15] C. Johnson, R. Rannacher and M. Boman , *Numeric and hydrodynamic stability: toward error control in computational fluid dynamics*, SIAM J. Numer. Anal., **32**(1995), 1058 – 1079.

[16] S.L. Johnson, Y. Saad and M.H. Schultz, *Alternating direction methods on multiprocessors*, SIAM J. Sci. Stat. Comput., **8** (1987), 686 – 700.

[17] J. Kačur, A. Handlovičova and M. Kačurova, *Solution of nonlinear diffusion problems by linear approximation schemes*, SIAM J. Numer. Anal., **30** (1993), 1703 – 1722.

[18] J.L. Lions, *Quelques Methods de Resolution des Problems aux Limites non Lineaires*. Dunod, Paris, 1988.

[19] G.I. Marchouk, *Splitting Methods*, Nauka, Moscow, 1988 (in Russian).

[20] P. Mehrotra, J. Saltz, and R. Voigt, editors, *Unstructured Scientific Computation on Scalable Multiprocessors*, MIT Press, 1992.

[21] R.H. Nochetto, *A note on the approximation of free boundaries by finite element methods*, RAIRO Model. Math. Anal. Numer., **20**(1986), 355 – 368.

[22] R.H. Nochetto, *Error estimates for multidimensional singular parabolic problems*, Japan J. Appl. Math., **4**(1987), 111 – 138.

[23] R.H. Nochetto and C. Verdi, *Approximation of degenerate parabolic problems using numerical integration*, SIAM J. Numer. Anal., **25** (1988), 784 – 814.

[24] J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems.* Plenum Press, New York and London, 1988.

[25] M.E. Rose, *Numerical methods for flows through porous media.* I, Math. of Comput., **40**(1983), 435 – 467.

[26] M.E. Rose, *Numerical methods for flows through porous media.* II, Comput. Math. Appl., **20**(1980), 99 – 122.

[27] A. Samarskij, *Theorie der Differenzenverfahren*, Geest and Portig, Leipzig, 1984.

[28] E.A. Socolovsky, *Difference schemes for degenerate parabolic equations*, Math. Comput., **47** (1986), 411 – 420.

[29] J.W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods,* Springer-Verlag, New York, Berlin, Heidelberg, 1995. 88 – 93.

[30] R.E. White, *An enthalpy formulation of the Stefan problem*, SIAM J. Numer. Anal., **19**(1982), 1129 – 1157.

[31] M. Zlamal, *A finite element solution of the nonlinear heat equation*, RAIRO Numer. Anal., **14**(1980), 203 – 216.