PARALLEL NUMERICAL METHODS FOR THE
ELLIPTIC-PARABOLIC PROBLEM

RAIMONDAS ČIEGIS
Vilnius Gediminas Technical University,
Institute of Mathematics and Informatics
Akademijos 4, 2600 Vilnius , Lithuania
e-mail: rc@rasa.vtu.lt

AIVARS ZEMITIS
Department of Mathematics
University of Kaiserslauter , Germany

1 Finite-difference scheme

In this report we investigate numerical methods for solving problems which arise in mathematical modelling of nonwowens used for making diapers. We consider numerical methods for solving a 3D elliptic-parabolic equation, which is parabolic in the region Ω_p and it becomes elliptic in the region of saturation Ω_E. In the parabolic region the equation is of nonlinear degenerate parabolic type and it can lead to a nonlinear advection-dominated diffusion equation, at least in some parts of Ω_p [1,2]. Firstly we change the definition of some coefficients in such a way, that the new equation is parabolic in the whole region $\Omega = \Omega_p \cup \Omega_E$. Then we approximate this generalized problem with the following *stability – correction* scheme

$$\begin{aligned}
\frac{U_s^{n+1/3} - U_{s-1}^n}{\tau} &= \partial_{\bar{x}_1}\partial_{x_1} Q(U_s^{n+1/3}) + \partial_{x_1} K(U_s^{n+1/3}) \\
&\quad + \partial_{\bar{x}_2}\partial_{x_2} Q(U_{s-1}^n) + \partial_{\bar{x}_3}\partial_{x_3} Q(U_{s-1}^n), \\
\frac{U_s^{n+2/3} - U_s^{n+1/3}}{\tau} &= \partial_{\bar{x}_2}\partial_{x_2} Q(U_s^{n+2/3}) - \partial_{\bar{x}_2}\partial_{x_2} Q(U_{s-1}^n), \\
\frac{U_s^{n+1} - U_s^{n+2/3}}{\tau} &= \partial_{\bar{x}_3}\partial_{x_3} Q(U_s^{n+1}) - \partial_{\bar{x}_3}\partial_{x_3} Q(U_{s-1}^n),
\end{aligned} \tag{1}$$

where s defines the number of the elliptic (or *outer*) iterations

$$U_{s+1}^n = \begin{cases} U^n & \text{if } U_s^{n+1} \le m_l, \\ U_s^{n+1} & \text{if } U_s^{n+1} > m_l. \end{cases} \tag{2}$$

Here Q and K are given nonlinear functions, $\partial_{\bar{x}_i}$ and ∂_{x_i} denote the standard finite-difference approximations of derivatives. Equations (1) are defined on

the difference mesh $\Omega_h = \{(x_{1i}, x_{2j}, x_{3k}), 0 < i < N_1, 0 < j < n_2, 0 < k < N_3\}$. Boundary conditions are specified on $\partial\Omega_h$. The Newton method is used for solving (1) and it defines *inner* (or *nonlinear*) iterations.

We summarize the main properties of the algorithm (1):

P1 Each splitting step (1) constitutes $O(N^2)$ tridiagonal nonlinear systems. At the boundary layer points the two tridiagonal equations are replaced by the following two equations

$$C_{j-1}U_{j-1} - B_{j-1}U_j = F_{j-1}, \tag{3}$$
$$-P_jU_{j-2} - A_jU_{j-1} + CU_j - BU_{j+1} = F_j. \tag{4}$$

P2 A different number of Newton's iterations is needed in different parts of Ω_h.

P3 The solution of (1) has a finite speed of propagation of disturbances to undisturbed region due to the degenerate nonlinear parabolic equation.

2 Parallel Numerical Algorithm

We consider a parallel version of (1) scheme. Our main goal is to investigate algorithms targeted for parallel computers with distributed memory or for clusters of workstations. For such computers the most important characteristics are the ratio of computation to communication times and the latency (or start up) time.

Several approaches are possible for the decomposition of Ω_h. We propose to use the decomposition of the computational domain into 1D slabs and computations on each slab are handled by different processes. The x_1 direction is used for this decomposition.

We compare 1D decomposition with 3D decomposition (or *block* decomposition). The first implication is that the communication start-up time is three times larger in the 3D decomposition case. Next we consider the load balancing of computations. The SC algorithm (1) has a non-uniformly distributed computational requirements. Firstly, more iterates are performed during the first splitting step in comparison with the second and third splitting steps. Secondly, the number of iterates is not the same for different grid layers. Again 1D decomposition enables us to obtain a uniform workload distribution among all p processors. The computational domain Ω_h has moving boundaries in time. Hence a data redistubution must be implemented at

some specific time moments in order to get a better load balancing. 1D decomposition is also more convient if a load balancing of computations must be obtained on non-homogeneous clusters of workstations.

In the case of 1D decomposition the second and the third splitting steps (1) constitute $O(N^2)$ 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.

Now we consider the parallel implementation of the first splitting step of SC scheme. We use a modification of Wang's subtracting Gaussian elimination algorithm (see [3]).

The standard formula must be changed for equations (3) – (4) at the inner boundary layer points. In the first phase the kth processor performs a forward factorization sweep in order to eliminate the sub-diagonal elements

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

Then it follows from (3) that $\gamma_j = 0$. Equation (4) 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

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

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

References

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

[2] R. Čiegis, A. Zemitis, *Numerical Algorithms for Simulation of the Liquid Trasport in Maltilayered Fleece*, 15th IMACS World Congress, Berlin, august 1997, **2**, pp. 117 – 122.

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