

---

# Predictor-Corrector Methods for Solving Continuous Casting Problem

J. Pieskä<sup>1</sup>, E. Laitinen<sup>1</sup>, and A. Lapin<sup>2</sup>

<sup>1</sup> University of Oulu, Department of Mathematical Sciences, Infotech of Oulu, Oulu, Finland, [erkki.laitinen@oulu.fi](mailto:erkki.laitinen@oulu.fi), [jali.pieska@oulu.fi](mailto:jali.pieska@oulu.fi)

<sup>2</sup> Kazan State University, Department of Computational Mathematics and Cybernetics Kazan, Russia, [alapin@ksu.ru](mailto:alapin@ksu.ru)

**Summary.** In this paper we present new numerical approach to solve the continuous casting problem. The main tool is to use IPEC method and DDM similar to Lapin and Pieska [2002] with multilevel domain decomposition. On the subdomains we use multidecomposition of the subdomains. The IPEC is used both in the whole calculation domain and inside the subdomains. The calculation algorithm is presented and numerically tested. Several conclusions are made and discussed.

## 1 Introduction

Theory of the so-called regional-additive schemes (splitting schemes with domain decomposition) for linear diffusion and convection-diffusion have been studied in Samarskii and Vabischevich [1996] and Vabischevich [1994]. The stability have been proved and error estimates have been deduced. For nonlinear problems like our their technique is not available.

Several new finite-difference schemes for a nonlinear convection-diffusion problem are constructed and numerically studied in Lapin and Pieska [2002]. These schemes are constructed on the basis of non-overlapping domain decomposition and predictor-corrector approach.

The paper of Lapin and Pieska [2002] was motivated by Dawson et al. [1991], Rivera et al. [2003], Rivera et al. [2001], where TL<sup>3</sup>, EP<sup>4</sup> and EPIC<sup>5</sup> methods have been studied and tested. The EPIC method was proved to be stable and scalable when solving on big number of processors. In the paper of Lapin and Pieska [2002] the scheme from Rivera et al. [2003], Rivera et al. [2001] was modified in such a way, that its implementation leads to IPEC<sup>6</sup> method.

---

<sup>3</sup> time lagging

<sup>4</sup> explicit predictor

<sup>5</sup> explicit predictor-implicit corrector

<sup>6</sup> implicit predictor-explicit corrector

The main idea of these kind of algorithms is first to solve the problem in artificial boundaries (predictor step). After the solution at the boundaries is known then it can be used as Dirichlet type boundary condition and the noncoupled subdomain problems can be solved parallel. The last step of these methods is to correct the solution at the artificial boundaries (corrector step).

The advantages of predictor-corrector methods (IPEC or EPIC) is that we reduce the amount of information send between processors. We need to send only once the subsolutions from slave processors to master processor. When we use Schwarz alternating methods with overlapping subdomains, the number of sending and receiving is much more bigger. The numerical experiences in Lapin and Pieska [2002] show that the speedup of IPEC method is linear. However, the stability and the rate of convergence for IPEC method is not known but asymptotically the rate of approximation is of the order  $O(\tau + h^2)$ .

The idea of multidecomposition method MDD is to use DDM with IPEC inside the subdomains. The subdomain is divided to smaller subdomains and then IPEC method is used to solve these smaller subproblems sequently. The main reason and motivation for this kind of method is to economise the number of processors. Nowadays the PC computers have multiprocessor cards but the number of processors in it are very small and limited. Our proposed algorithm gives a good and effective way to decrease calculation times in the case of only few processors.

## 2 Problem statement

The continuous casting problem can be mathematically formulated as follows. Let the rectangular domain  $\Omega \subset \mathbb{R}^2$ ,  $\Omega = (0, l_1) \times (0, l_2)$  be occupied by a thermodynamically homogeneous and isotropic steel. We denote by  $\bar{H}(x, t)$  the enthalpy and by  $T(x, t)$  the temperature for  $(x, t) \in \bar{\Omega} \times [0, t_f]$ . We have constitutive law  $\bar{H} = \bar{H}(T) = \rho \int_0^T c(\Theta) d\Theta + \rho L(1 - f_s(T))$ , where  $\rho$  is the density,  $c(T)$  is the specific heat,  $L$  is the latent heat and  $f_s(T)$  is the solid fraction at temperature  $T$ . The graph  $\bar{H}(T)$  is an increasing function  $\mathbb{R} \rightarrow \mathbb{R}$ , involving near vertical segment, which corresponds to a phase transition state, namely, for  $T \in [T_S, T_L]$ . In our numerical example in section 6 we choose for the enthalpy function phase change interval  $T_L - T_S = 0.02$ . This does not effect cruisial for convergence properties of our method. Further by  $k(T)$  we denote the thermal conductivity coefficient.

Using Kirchoff's transformation  $u = K(T) = \int_0^T k(\xi) d\xi$  a continuous casting process can be described by a boundary-value problem, formally written in the following pointwise form: find  $u(x, t)$  and  $H(x, t)$  such that

$$(P) \begin{cases} \frac{\partial H(u)}{\partial t} + v \frac{\partial H(u)}{\partial x_2} - \Delta u = 0, & \text{for } x \in \Omega, t > 0, \\ u = z(x, t) & \text{for } x \in \Gamma_D, t > 0, \\ \frac{\partial u}{\partial n} = g, & \text{for } x \in \Gamma_N, t > 0, H = H_0(x) & \text{for } x \in \bar{\Omega}, t = 0, \end{cases}$$

where  $v = const > 0$  is a casting speed in  $x_2$ -direction,  $\Gamma_D \cup \Gamma_N = \partial\Omega$  is the boundary of the domain, below  $\Gamma_D = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = l_2\}$ . The existence and uniqueness of a weak solution for problem (P) are proved in Rodrigues and Yi [1990].

### 3 Mesh approximation of continuous casting problem

We approximate problem (P) by an implicit finite difference scheme in time and finite element method in space. Let  $T_h$  be a partitioning of  $\Omega$  in the rectangular elements  $\delta$  of dimensions  $h_1 \times h_2$  and  $V_h = \{u_h(x) \in H^1(\Omega) : u_h(x) \in Q_1 \text{ for all } \delta \in T_h\}$ , where  $Q_1$  is the space of bilinear functions. By  $\Pi_h v(x)$  we denote the  $V_h$ -interpolant of a continuous function  $v(x)$ , i.e.  $\Pi_h v(x) \in V_h$  and coincides with  $v(x)$  in the mesh nodes (vertices of all  $\delta \in T_h$ ). We also use an interpolation operator  $P_h$ , which is defined as follows: for any continuous function  $v(x)$  the function  $P_h v(x)$  is piecewise linear in  $x_1$ , piecewise constant in  $x_2$  and on  $\delta = [x_1, x_1 + h_1] \times [x_2, x_2 + h_2]$  it coincides with  $v(x)$  at  $(x_1, x_2 + h_2)$  and  $(x_1 + h_1, x_2 + h_2)$ .

Let further  $V_h^0 = \{u_h(x) \in V_h : u_h(x) = 0 \text{ for all } x \in \Gamma_D\}$ ,  $V_h^z = \{u_h(x) \in V_h : u_h(x) = z_h \text{ for all } x \in \Gamma_D\}$ . Here  $z_h$  is the bilinear interpolation of  $z$  on the boundary  $\Gamma_D$ . For any continuous function  $v(x)$  we define the quadrature formulas:  $S_\delta(v) = \int_\delta \Pi_h v dx$ ,  $S_{\partial\delta}(v) = \int_{\partial\delta} \Pi_h v dx$ ,  $E_\delta(v) = \int_\delta P_h v dx$ ,  $S_\Omega(v) = \sum_{\delta \in T_h} S_\delta v$ ,  $S_{\Gamma_N}(v) = \sum_{\partial\delta \in T_h \cap \Gamma_N} S_{\partial\delta}(v)$ ,  $E_\Omega(v) = \sum_{\delta \in T_h} E_\delta(v)$ . Let also  $\omega_\tau = \{t_k = k\tau, 0 \leq k \leq M, M\tau = t_f\}$  be an uniform mesh in time on the segment  $[0, t_f]$  and  $\partial_t H = \frac{1}{\tau}(H(x, t) - H(x, t - \tau))$ . When constructing the characteristic mesh scheme we approximate the term  $(\frac{\partial}{\partial t} + v \frac{\partial}{\partial x_2})H$  by using the characteristics of the first order differential operator Chen [1991]. Namely, if  $(x_1, x_2, t)$  is the mesh point on the time level  $t$  we choose  $\tilde{x}_2 = x_2 - \int_{t-\tau}^t v(\xi) d\xi$  and approximate:  $(\frac{\partial}{\partial t} + v \frac{\partial}{\partial x_2})H \approx \frac{1}{\tau}(H(x_1, x_2, t) - \tilde{H}(x, t - \tau))$ , where we denote  $\tilde{H}(x, t - \tau) = H(x_1, \tilde{x}_2, t - \tau)$ . Near the boundary it can happen that  $\tilde{x}_2 < 0$ . In that case we put  $\tilde{H}(x, t - \tau) = H(x_1, 0, t - \tau)$ . In what follows we use the notation  $d_t H = \frac{1}{\tau}(H(x, t) - \tilde{H}(x, t - \tau))$  for the difference quotient in each mesh point on time level  $t$ .

Now, the characteristic finite difference scheme for problem (P) is: for all  $t \in \omega_\tau, t > 0$ , find  $u_h \in V_h^z$  and  $H_h \in V_h$  such that

$$S_\Omega(d_t H_h \eta_h) + S_\Omega(\nabla u_h \nabla \eta_h) = S_{\Gamma_N}(g \eta_h) \text{ for all } \eta_h \in V_h^0 \tag{1}$$

Let  $N_0 = \text{card } V_h^0$  and  $u \in \mathbb{R}^{N_0}$  be the vector of nodal values for  $u_h \in V_h^0$ . We use the writing  $u_h \Leftrightarrow u$  for this bijection. We define  $N_0 \times N_0$  matrices by the following relations: for all  $u, \eta \in \mathbb{R}^{N_0}$ ,  $u \Leftrightarrow u_h \in V_h^0$  and  $\eta \Leftrightarrow \eta_h \in V_h^0$ ,  $(\tilde{A}u, \eta) = S_\Omega(\nabla u_h \nabla \eta_h)$ ,  $(Mu, \eta) = S_\Omega(u_h \eta_h)$ ,  $A_0 = M^{-1}\tilde{A}$ . Let now  $\tilde{z}_h(x) \in V_h$  be the function, which is equal to  $z_h$  on  $\bar{\Gamma}_D$  and 0 for all nodes in  $\Omega \cup \Gamma_N$ . Then a right hand side vector  $f$  is defined by the equality  $(f, \eta) = S_{\Gamma_N}(g \eta_h) - S_\Omega(\nabla \tilde{z}_h, \nabla \eta_h) \quad \forall \eta \in \mathbb{R}^{N_0}, \eta \Leftrightarrow \eta_h \in V_h^0$ , and we

set  $F = M^{-1}f$ . In these notations the algebraic form for characteristic mesh scheme (1) becomes

$$d_{\bar{t}}H + A_0u = F, \tag{2}$$

It is easy to see, that  $A_0$  is the standard five-point finite difference approximation of Laplace operator,  $A_0u = -u_{x_1\bar{x}_1} - u_{x_2\bar{x}_2}$  for the internal mesh points with the notations  $u_{x_1} = h^{-1}(u(x_1 + h_1, x_2) - u(x_1, x_2))$ ,  $u_{\bar{x}_1} = h^{-1}(u(x_1, x_2) - u(x_1 - h_1, x_2))$ , and similarly for  $u_{x_2}$  and  $u_{\bar{x}_2}$ . Furthermore, let  $\bar{\omega}$  be the set of all grid points,  $\gamma_D = \bar{\Gamma}_D \cap \bar{\omega}$ ,  $\gamma_N = \bar{\Gamma}_N \cap \bar{\omega}$ ,  $\omega = \Omega \cap \bar{\omega}$ ,  $\gamma_N^- = \{x \in \gamma_N : x_1 = 0\}$ ,  $\gamma_N^+ = \{x \in \gamma_N : x_1 = l_1\}$ .

### 4 Domain decomposition by straight lines

In this section we present the IPEC algorithm of Lapin and Pieska [2002]. We restrict our discussion to the case of decomposition by unidirect straight lines. More variations and possibilities of decomposition is discussed and tested in Lapin and Pieska [2002].

Let the domain  $\Omega$  be decomposed into two subdomains  $\Omega_1$  and  $\Omega_2$  by a straight line  $S_y$  in  $x_2$ -direction, which is also a grid line. We denote by  $\delta_{S_y}$  the characteristic function of this line, i.e., the mesh function  $\delta_{S_y}(x) = 1$  for  $x \in S_y \cap \bar{\omega}$ , while  $\delta_{S_y}(x) = 0$  for other mesh points. Also, let  $\bar{\omega}_k$ ,  $k = 1, 2$  be the corresponding to the subdomains  $\bar{\Omega}_k$  sets of grid points,  $S_y$  being the common part of their boundaries.

Let  $A_2u = -\delta_{S_y}u_{x_1\bar{x}_1}$  and  $A_1 = A_0 - A_2$ ,

$$A_1u = \begin{cases} -(1 - \delta_{S_y})u_{x_1\bar{x}_1} - u_{x_2\bar{x}_2} & \text{for } x \in \omega, \\ -2h_1^{-1}u_{x_1} - u_{x_2\bar{x}_2} & \text{for } x \in \gamma_N^-, \\ 2h_1^{-1}u_{\bar{x}_1} - u_{x_2\bar{x}_2} & \text{for } x \in \gamma_N^+. \end{cases}$$

Now, instead of characteristic scheme (2) we consider the following scheme on the time level  $t_{n+1} = (n + 1)\tau$ :

$$\frac{1}{\tau}(H^{n+\frac{1}{2}} - \tilde{H}^n) + A_1u^{n+\frac{1}{2}} + A_2u^n = F, \tag{3}$$

$$\frac{\delta_{S_y}}{\tau}(H^{n+1} - \tilde{H}^n) + \frac{1 - \delta_{S_y}}{\tau}(H^{n+1} - H^{n+\frac{1}{2}}) + \delta_{S_y}A_1u^{n+\frac{1}{2}} + A_2u^{n+1} = \delta_{S_y}F, \tag{4}$$

Let us discuss the implementation of scheme (3),(4). In the points of  $S_y$  equation (3) has the form:

$$\frac{H^{n+\frac{1}{2}} - \tilde{H}^n}{\tau} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} - u_{x_1\bar{x}_1}^n = F, \tag{5}$$

i.e. in the points of  $S_y$  we have one-dimensional problem (5), that we solve first. After that the equation (3) is splitted in two non-coupled characteristic schemes in the subdomains:

$$\begin{cases} \frac{H^{n+\frac{1}{2}} - \tilde{H}^n}{\tau} - u_{x_1\bar{x}_1}^{n+\frac{1}{2}} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} = F, \text{ for } x \in \omega_1 \cup \omega_2, \\ \frac{H^{n+\frac{1}{2}} - \tilde{H}^n}{\tau} - \frac{2}{h_1} u_{x_1}^{n+\frac{1}{2}} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} = F, \text{ for } x \in \gamma_N^-, \\ \frac{H^{n+\frac{1}{2}} - \tilde{H}^n}{\tau} + \frac{2}{h_1} u_{\bar{x}_1}^{n+\frac{1}{2}} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} = F, \text{ for } x \in \gamma_N^+, \end{cases} \quad (6)$$

and these equations are accomplished by Dirichlet boundary conditions, given on  $\gamma_D$  and calculated from (5) on  $S_y$ . Finally we solve the system of the equations, corresponding to  $x \in S_y$ :  $\frac{H^{n+1} - \tilde{H}^n}{\tau} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} - u_{x_1\bar{x}_1}^{n+1} = F$ . As  $u^{n+1}(x) = u^{n+\frac{1}{2}}(x)$  for  $x \notin S_y$ , this system becomes

$$\begin{cases} \frac{H^{n+1} - \tilde{H}^n}{\tau} + 2 \frac{u^{n+1}(x_1, x_2)}{h_1^2} - u_{x_2\bar{x}_2}^{n+\frac{1}{2}} \\ - \frac{u^{n+\frac{1}{2}}(x_1 - h_1, x_2) + u^{n+\frac{1}{2}}(x_1 + h_1, x_2)}{h_1^2} = F, \text{ } x \in S_y. \end{cases} \quad (7)$$

Thus, the algorithm for the implementation of (3),(4) consists of 3 steps:

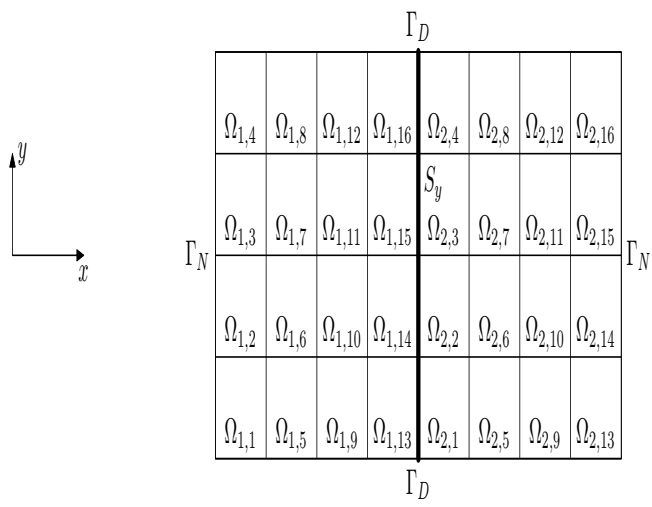
- 1) Predictor step: solving one-dimensional problem (5);
- 2) Main step: concurrent solving subproblems (6);
- 3) Corrector step: solving the system of scalar equations (7).

### 5 Multidecomposition method

The general idea of the multidecomposition is to divide the subdomain to smaller subdomains i.e. use two-level decomposition of the calculation domain. The division of the subdomains is presented in the figure 1. We use the notation  $\Omega_i = \cup_{j_i=1}^{p_i} \Omega_{i,j_i}$ . The use of high number of subdomains inside the subdomain may increase the error dramatically. To overcome this feature we introduce so called smoothing steps to our method. The calculation algorithm for characteristic mesh scheme (5)-(7) is presented below.

**Algorithm 1.**

1. Time step  $n$  perform on the main processor the predictor step (5) on  $S_y$ .
2. Send the values of  $u^{n+\frac{1}{2}}$  and  $H^{n+\frac{1}{2}}$  on  $S_y$  to the slave processors.
3. Concurrently on the slave processors perform the predictor step (5) on the artificial boundaries of the subdomains  $\Omega_{i,j_i}$ ,  $i = 1, 2$ ,  $j_1 = 1, \dots, p_1$ ,  $j_2 = 1, \dots, p_2$ .
4. Concurrently on the slave processors perform sequentially the main step (6) on the subdomains  $\Omega_{i,j_i}$ .
5. Concurrently on the slave processors perform the corrector step (7) on the artificial boundaries of the subdomains  $\Omega_{i,j_i}$ ,  $i = 1, 2$ ,  $j_1 = 1, \dots, p_1$ ,  $j_2 = 1, \dots, p_2$ .
6. On the slave processors perform the smoothing step i.e. few iterations of the MSOR-method over the whole subdomain  $\Omega_i$ .



**Fig. 1.** Used nonoverlapping domain decomposition and multidecomposition of the subdomains.

7. Send the subsolutions  $u^{n+1}$  and  $H^{n+1}$  to the main processor.
8. On the main processor perform the corrector step (7) on  $S_y$ .
9. On the main processor perform few iterations of the MSOR-method in the neighborhood of  $S_y$ .
10. Put  $n = n + 1$ , if the final time  $t_f$  is reached **STOP**, else **GOTO 1**.

*Remark 1.* On the step 3. we do not do the predictor step (5) on  $S_y$ .

*Remark 2.* On the steps 3.-6. we do the calculations concurrently. Each processor perform the steps asynchronously.

### 6 Numerical verification

Let  $\Omega = ]0, 1[ \times ]0, 1[$  with the boundary  $\Gamma$  divided in two parts  $\Gamma_D = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = 1\}$  and  $\Gamma_N = \Gamma \setminus \Gamma_D$ . Moreover, let  $t_f = 1$  and  $u_{SL} = 1$ . The phase change interval is  $[u_{SL} - \varepsilon, u_{SL} + \varepsilon]$ ,  $\varepsilon = 0.01$ , and the velocity is  $v(t) = \frac{1}{5}$ . Our numerical example is

$$\begin{aligned}
 \frac{\partial H}{\partial t} - \Delta K + v(t) \frac{\partial H}{\partial x_2} &= f(x; t) && \text{on } \Omega, \\
 u(x_1, x_2; t) &= (x_1 - \frac{1}{2})^2 - \frac{1}{2}e^{-4t} + \frac{5}{4} && \text{on } \Gamma_D, \\
 \frac{\partial u}{\partial n} &= 1 && \text{on } \Gamma_N, \\
 u(x_1, x_2; 0) &= (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 + \frac{1}{2} && \text{on } \Omega,
 \end{aligned}$$

where Kirchoff's temperature  $K(u)$  and enthalpy  $H(u)$  are according to their definition

$$K(u) = \begin{cases} u & \text{if } u < u_{SL} - \varepsilon, \\ \frac{3}{2}u - \frac{1-\varepsilon}{2} & \text{if } u \in [u_{SL} - \varepsilon, u_{SL} + \varepsilon], \\ 2u - 1 & \text{if } u > u_{SL} + \varepsilon, \end{cases}$$

and

$$H(u) = \begin{cases} 2u & \text{if } u < u_{SL} - \varepsilon, \\ \left(\frac{1+8\varepsilon}{2\varepsilon}\right)(u - 1) + \frac{5+4\varepsilon}{2} & \text{if } u \in [u_{SL} - \varepsilon, u_{SL} + \varepsilon], \\ 6u - 3 & \text{if } u > u_{SL} + \varepsilon. \end{cases}$$

The known right hand side  $f(x; t)$  is chosen such that our problem have the exact solution  $u(x_1, x_2; t) = (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 - \frac{1}{2}e^{-4t} + 1$ .

The stopping criterion of the calculations was the  $L_2$ -norm of residual  $\|r\|_{L_2(\Omega)} < 10^{-4}$ . We solve our problem by using different methods, Additive Schwarz alternating method (ASAM), Implicit Predictor-Explicit Corrector method (IPEC), Multidomain decomposition method (MDD) and sequential modified SOR method (SEQ). The results are presented in the table 1.

**Table 1.** Calculation times for ASAM, IPEC, MDD and sequential MSOR methods when the number of processors and calculation grid is changed. Number of inside subdomains,  $4 \times 4$ .

Grid	SEQ 1 proc.	ASAM 2 proc.	ASAM 4 proc.	ASAM 8 proc.	IPEC 2 proc.	IPEC 4 proc.	IPEC 8 proc.	MDD 2 proc.	MDD 4 proc.
$65 \times 65 \times 128$	8.67	7.01	4.76	3.91	3.76 s	2.33 s	1.64 s	5.4 s	4.4
$129 \times 129 \times 256$	112.9	77.6	59.3	33.9	47.3 s	25.1 s	14.1 s	25.9 s	14.6
$257 \times 257 \times 512$	1425	889	494	281	600 s	285 s	164 s	342 s	179

**Table 2.** Efficiencies for ASAM, IPEC, MDD and methods when the number of processors and calculation grid is changed. Number of inside subdomains,  $4 \times 4$ .

Grid	ASAM 2 proc.	ASAM 4 proc.	ASAM 8 proc.	IPEC 2 proc.	IPEC 4 proc.	IPEC 8 proc.	MDD 2 proc.	MDD 4 proc.
$65 \times 65 \times 128$	0.62	0.46	0.28	1.15	0.93	0.66	0.8	0.49
$129 \times 129 \times 256$	0.73	0.48	0.42	1.19	1.12	1	2.18	1.93
$257 \times 257 \times 512$	0.8	0.72	0.63	1.19	1.25	1.09	2.08	1.99

## 7 Conclusions

The numerical examples show that the multidecomposition method (MDD) is very effective numerical method when solving continuous casting problem. The idea to divide the subdomains to smaller subdomains seems to be very

good and profitable. The algebraic dimension of the subproblems inside the subdomains are very small and thus they are very quick to solve.

The introduced smoothing step allows us to use quite big number of subdomains. The accuracy of the different methods, MDD, ASAM and IPEC are the same. However, the smoothing step is economical to perform and calculation times for MDD are roughly half of the calculation times of the IPEC method.

The tables 1 and 2 show very clearly the advantages of the multidecomposition method over other methods. It is extremely quick and accuracy is the same than other methods. Implementation of MDD is straightforward and it do not need huge amount of processors to solve big and complicated problems.

## References

- Z. Chen. Numerical solutions of a two-phase continuous casting problem. In P. Neittaanmaki, editor, *Numerical Methods for Free Boundary Problem*, pages 103–121, Basel, 1991. International Series of Numerical Mathematics, Birkhuser.
- C. N. Dawson, Q. Du, and T. F. Dupont. A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Math. Comput.*, 57:63–71, 1991.
- A. V. Lapin and J. Pieska. On the parallel domain decomposition algorithms for time-dependent problems. *Lobachevskii Journal of Mathematics*, 10: 27–44, 2002.
- W. Rivera, J. Zhu, and D. Huddleston. An efficient parallel algorithm for solving unsteady nonlinear equations. In T. M. Pinkston, editor, *Proceedings of the International Conference on Parallel Processing Workshops*, pages 79–84. IEEE Computer Society, Los Alamitos, California, 2001.
- W. Rivera, J. Zhu, and D. Huddleston. An efficient parallel algorithm with application to computational fluid dynamics. *Computers and Mathematics with Applications*, 45:165–188, 2003.
- J. F. Rodrigues and F. Yi. On a two-phase continuous casting stefan problem with nonlinear flux. *Euro. of Applied Mathematics*, 1:259–278, 1990.
- A. Samarskii and P. Vabischevich. Factorized regional-additive schemes for convection-diffusion problems (in russian). Technical Report V. 346, Russian Academic of Sciences, 1996.
- P. Vabischevich. Parallel domain decomposition algorithms for time-dependent problems of mathematical physics. *Advances in Numerical Methods and Applications*, pages 293–299, 1994.