Lobachevskii Journal of Mathematics
Vol. 13, 2003, 67 – 80

E. Laitinen, A.V. Lapin, and J. Pieskä

NUMERICAL EXPERIMENTS WITH MULTILEVEL SUBDOMAIN DECOMPOSITION METHOD

Abstract. In this paper we present a new numerical approach to solve the continuous casting problem. The main tool is to use so-called IPEC method and DDM similar to [6] with multilevel domain decomposition. On the subdomains we use the multidecomposition of the subdomains. The IPEC is used both in the whole calculation domain and inside the subdomains. 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 [14],[15] and [16]. The stability have been proved and error estimates have been deduced. For the non-linear 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 [6]. These schemes are constructed on the basis of non-overlapping domain decomposition and predictor-corrector approach. (Note that the term "predictor-corrector domain decomposition method" was introduced by Quarteroni and Valli [13]).

The paper of Lapin and Pieskä [6] was motivated by [2], [9], [10], [11], where TL[1], EP[2] and EPIC[3] methods have been studied and tested. The EPIC method was proved to be stable and scalable when solving on a big number of processors. In the paper [6] the scheme from [10], [11] was modified in such a way, that its implementation leads to IPEC[4] method.

[1]time lagging
[2]explicit predictor
[3]explicit predictor-implicit corrector
[4]implicit predictor-explicit corrector

Key words and phrases. Stefan problem, continuous casting process, finite element method, predictor-corrector scheme, domain decomposition.
This work is supported by RFBR, project N 01-01-00068.

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 a Dirichlet-type boundary condition and the non-coupled subdomain problems can be solved in parallel. The last step of these methods is to correct the solution at the artificial boundaries (corrector step). The idea is similar to Schur's complement type methods but in our approach the calculation of the artificial boundaries between subdomains are very easy and cheap to perform. There is also no need to construct a good preconditioner for the interface problems.

The advantages of predictor-corrector methods (IPEC or EPIC) is that we reduce the amount of information sending between processors. We need to send only once the subsolutions from slave processors to master processor. When we use Schwarz alternating methods with the overlapping subdomains, the number of sending and receiving is much more bigger. The numerical experiences show that the speedup of IPEC method is very good comparing with Schwarz methods and calculation times are roughly half of the times of Schwarz methods. Moreover, the linear speedup is reached in the numerical tests of [6].

The idea of multidecomposition method MDD is to use DDM with IPEC inside the subdomains. The subdomain is divided into smaller subdomains and then IPEC method is used to solve these smaller subproblems sequently.

The rest of the article is organized as follows. In the section 2 we present the continuous casting problem. The section 3 deals with the mesh approximation of the problem. The used domain decomposition and calculation algorithm is presented in the section 4. The section 5 is devoted to numerical testing of the new algorithm. Finally, in the section 6 the conclusions are made.

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 related to the unit mass and by $T(x,t)$ the temperature for $(x,t) \in \bar{\Omega} \times [0, t_f]$. We have constitutive law

$$\bar{H} = \bar{H}(T) = \rho \int_0^T c(\Theta) d\Theta + \rho L(1 - f_{\mathrm{s}}(T)),$$

where ρ is the density, $c(T)$ is the specific heat, L is the latent heat and $f_{\mathrm{s}}(T)$ is the solid fraction at temperature T of the form

$$f_{\mathrm{s}}(T) = \begin{cases} 1 \text{ for } T < T_{\mathrm{S}}, \\ \dfrac{T_L - T}{T_L - T_S}, \text{ for } T_{\mathrm{S}} < T < T_{\mathrm{L}}, \\ 0 \text{ for } T > T_{\mathrm{L}}. \end{cases}$$

The graph $\bar{H}(T)$ is an increasing function $\mathbb{R} \to \mathbb{R}$, involving near vertical segment, which corresponds to a phase transition state, namely, for $T \in [T_{\mathrm{S}}, T_{\mathrm{L}}]$. In the case of piecewise constant specific heat $c(T)$, the enthalpy function is

piecewise linear of the form

$$\bar{H}(T) = \begin{cases} \bar{\alpha}_1 T + \bar{\beta}_1 \text{ for } T < T_{\mathrm{S}}, \\ \bar{\alpha}_2 T + \bar{\beta}_2, \text{ for } T_{\mathrm{S}} < T < T_{\mathrm{L}}, \\ \bar{\alpha}_3 T + \bar{\beta}_3 \text{ for } T > T_{\mathrm{L}}. \end{cases}$$

Further by $k(T)$ we denote the thermal conductivity coefficient, which is continuous and increasing in T.

A continuous casting process can be described by a boundary-value problem, formally written in the following pointwise form: find $T(x,t)$ and $\bar{H}(x,t)$ such that

$$\begin{cases} \dfrac{\partial \bar{H}}{\partial t} + v\dfrac{\partial \bar{H}}{\partial x_2} - \nabla \cdot (k(T)\nabla T) = 0 \text{ for } x \in \Omega, t > 0, \\ T = \bar{z}(x,t) \text{ for } x \in \Gamma_D, t > 0, \\ k(T)\dfrac{\partial T}{\partial n} = g, \text{ for } x \in \Gamma_N, t > 0, \\ \\ \bar{H} = \bar{H}_0(x) \text{ for } x \in \bar{\Omega}, t = 0, \end{cases}$$

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 $\Gamma_D = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = l_2\}$.

Using Kirchoff's transformation $u = K(T) = \displaystyle\int_0^T k(\xi)d\xi$ and the notation $H(u) = \bar{H}(T) = \bar{H}(K^{-1}(u))$, we can rewrite the continuous casting problem as

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

The existence and uniqueness of a weak solution for problem (P) are proved in [12].

In the case of piecewise constant specific heat $c(u)$ the enthalpy function takes the form

$$H(u) = \begin{cases} \alpha_1 u + \beta_1 \text{ for } u < u_{\mathrm{S}}, \\ \alpha_2 u + \beta_2, \text{ for } u_{\mathrm{S}} < u < u_{\mathrm{L}}, \\ \alpha_3 u + \beta_3 \text{ for } u > u_{\mathrm{L}}. \end{cases} \tag{1}$$

3. Mesh approximation of continuous casting problem

We approximate problem (P) by an implicit in time finite difference scheme and by a semi-implicit finite difference scheme, using for the approximation in space variables a finite element method with the quadrature rules.

Let Ξ_{h} be a partitioning of Ω in the rectangular elements δ of dimensions $h_1 \times h_2$ and $V_{\mathrm{h}} = \{u_{\mathrm{h}}(x) \in H^1(\Omega) : u_{\mathrm{h}}(x) \in Q_1 \text{ for all } \delta \in \Xi_{\mathrm{h}}\}$, where Q_1 is the space of bilinear functions. By $\Pi_{\mathrm{h}} v(x)$ we denote the V_{h}-interpolant of a continuous function $v(x)$, i.e. $\Pi_{\mathrm{h}} v(x) \in V_{\mathrm{h}}$ and coincides with $v(x)$ in the mesh nodes (vertices of all $\delta \in \Xi_{\mathrm{h}}$). We also use an interpolation operator

P_h, which is defined as follows: for any continuous function $v(x)$ the function $P_\mathrm{h}v(x)$ 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 $V_\mathrm{h}^0 = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = 0 \text{ for all } x \in \Gamma_D\}$, $V_\mathrm{h}^z = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = z_\mathrm{h} \text{ for all } x \in \Gamma_D\}$. Here z_h is the bilinear interpolation of z on the boundary Γ_D. For any continuous function $v(x)$ we define the quadrature formulas:

$$S_\delta(v) = \int_\delta \Pi_\mathrm{h} v dx, \quad S_{\partial\delta}(v) = \int_{\partial\delta} \Pi_\mathrm{h} v dx, \quad E_\delta(v) = \int_\delta P_\mathrm{h} v dx,$$

$$S_\Omega(v) = \sum_{\delta\in\Xi_\mathrm{h}} S_\delta v, \quad S_{\Gamma_2}(v) = \sum_{\partial\delta\in\Xi_\mathrm{h}\cap\bar{\Gamma}_2} S_{\partial\delta}(v), \quad E_\Omega(v) = \sum_{\delta\in\Xi_\mathrm{h}} E_\delta(v).$$

Let also $\omega_\tau = \{t_\mathrm{k} = k\tau, 0 \le k \le M, M\tau = t_f\}$ be an uniform mesh in time on the segment $[0, t_f]$ and $\partial_{\bar{\mathrm{t}}} H = \dfrac{1}{\tau}(H(x,t) - H(x,t-\tau))$. Then the implicit in time finite difference scheme with up-wind approximation of the convective term $v\,\partial H/\partial x_2$ can be written as follows: for all $t \in \omega_\tau$, $t > 0$, find $u_\mathrm{h} \in V_\mathrm{h}^z$ and $H_\mathrm{h} \in V_\mathrm{h}$ such, that

$$S_\Omega(\partial_{\bar{\mathrm{t}}} H_\mathrm{h}\eta_\mathrm{h}) + E_\Omega(v\frac{\partial H_\mathrm{h}}{\partial x_2}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0. \tag{2}$$

When constructing the characteristic mesh scheme we approximate the term $\left(\dfrac{\partial}{\partial t} + v\dfrac{\partial}{\partial x_2}\right) H$ by using the characteristics of the first order differential operator (similar to [1], [3]). Namely, if (x_1, x_2, t) is the mesh point on the time level t we choose $\tilde{x}_2 = x_2 - \displaystyle\int_{t-\tau}^{t} v(\xi)d\xi$ and approximate:

$$\left(\frac{\partial}{\partial t} + v\frac{\partial}{\partial x_2}\right) H \approx \frac{1}{\tau}\left(H(x_1, x_2, t) - \tilde{H}(x_1, x_2, t-\tau)\right),$$

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 $d_{\bar{\mathrm{t}}} H = \dfrac{1}{\tau}(H(x,t) - \tilde{H}(x, t-\tau))$ 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 $u_\mathrm{h} \in V_\mathrm{h}^z$ and $H_\mathrm{h} \in V_\mathrm{h}$ such that

$$S_\Omega(d_{\bar{\mathrm{t}}} H_\mathrm{h}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0 \tag{3}$$

Let $N_0 = \operatorname{card} V_\mathrm{h}^0$ and $u \in \mathbb{R}^{\mathrm{N}_0}$ be the vector of nodal values for $u_\mathrm{h} \in V_\mathrm{h}^0$. We use the writing $u_\mathrm{h} \Leftrightarrow u$ for this bijection. We define $N_0 \times N_0$ matrices by the following relations: for all $u, \eta \in \mathbb{R}^{\mathrm{N}_0}$, $u \Leftrightarrow u_\mathrm{h} \in V_\mathrm{h}^0$ and $\eta \Leftrightarrow \eta_\mathrm{h} \in V_\mathrm{h}^0$,

$$(\tilde{A}u, \eta) = S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}), \quad (Mu, \eta) = S_\Omega(u_\mathrm{h}\eta_\mathrm{h}), \quad (\tilde{C}u, \eta) = E_\Omega(v\frac{\partial u_\mathrm{h}}{\partial x_2}\eta_\mathrm{h}),$$

$$A_0 = M^{-1}\tilde{A}, \quad C = M^{-1}\tilde{C}.$$

Let now $\tilde{z}_{\rm h}(x) \in V_{\rm h}$ be the function, which is equal to $z_{\rm h}$ on $\bar{\Gamma}_D$ and 0 for all nodes in $\Omega \cup \Gamma_N$. Then a vector f is defined by the equality

$$(f, \eta) = S_{\Gamma_2}(g\eta_{\rm h}) - S_\Omega(\nabla \tilde{z}_{\rm h}, \nabla \eta_{\rm h}) \quad \forall \eta \in \mathbb{R}^{\rm N_0},\ \eta \Leftrightarrow \eta_{\rm h} \in V^0_{\rm h},$$

and we set $F = M^{-1} f$.

In these notations the algebraic form for implicit mesh scheme (2) at fixed time level is:

$$\partial_{\bar{t}} H + A_0 u + CH = F, \tag{4}$$

while characteristic mesh scheme (3) becomes

$$d_{\tilde{t}} H + A_0 u = F, \tag{5}$$

It is easy to see, that A_0 is the standard five-point finite difference approximation of Laplace operator, $A_0 u = -u_{x_1\bar{x}_1} - u_{x_2\bar{x}_2}$ 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 $u_{\bar{x}_2}$. For more detailed writing of the explicit form for $A_0 u$ let us introduce several sets of the grid points. Namely, let $\bar{\omega}$ be the set of all grid points, i.e. vertices of finite elements, $\gamma_D = \bar{\Gamma}_D \cap \bar{\omega}$, $\gamma_N = \Gamma_N \cap \bar{\omega}$, $\omega = \Omega \cap \bar{\omega}, \gamma_N^- = \{x \in \gamma_N : x_1 = 0\}$, $\gamma_N^+ = \{x \in \gamma_N : x_1 = l_1\}$.

Now $A_0 = A^1 + A^2$ with

$$A^1 u = \begin{cases} -u_{x_1\bar{x}_1} \text{ for } x \in \omega, \\ -2h_1^{-1} u_{x_1} \text{ for } x \in \gamma_N^-, \\ 2h_1^{-1} u_{\bar{x}_1} \text{ for } x \in \gamma_N^+, \end{cases}$$

$$A^2 u = \begin{cases} -u_{x_2\bar{x}_2} \text{ for } x \in \omega \cup \gamma_N, \\ 0 \text{ for } x \in \gamma_D. \end{cases}$$

The term CH in the implicit scheme corresponds to an up-wind approximation of the nonlinear convective term, $CH = H_{\bar{x}_2}$ for $x \in \omega \cup \gamma_N$.

4. Domain decomposition by straight lines

In this section we present the IPEC algorithm [6]. We restrict our discussion to the case of decomposition by unidirect straight lines. More variations and possibilities of decomposition are discussed and tested in [6]. Methods presented here and in the following section can be easily implemented and used with more complicated decompositions of calculation domain. Moreover, the generalization to 3-D case is straightforward. In that case the 1-D lines are replaced by 2-D planes.

Let the domain Ω be decomposed into two subdomains Ω_1 and Ω_2 by a straight line S_y in x_2-direction, which is also a grid line. We denote by δ_{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 $A_2 u = -\delta_{S_y} u_{x_1\bar{x}_1}$ and $A_1 = A_0 - A_2$,

$$A_1 u = \begin{cases} -(1 - \delta_{S_y}) u_{x_1\bar{x}_1} - u_{x_2\bar{x}_2} \text{ for } x \in \omega, \\ -2h_1^{-1} u_{x_1} - u_{x_2\bar{x}_2} \text{ for } x \in \gamma_N^-, \\ 2h_1^{-1} u_{\bar{x}_1} - u_{x_2\bar{x}_2} \text{ for } x \in \gamma_N^+. \end{cases}$$

Now, instead of implicit scheme (4) we consider the following scheme on the time level $t_{n+1} = (n+1)\tau$:

$$\frac{1}{\tau}(H^{n+1/2} - H^n) + A_1 u^{n+1/2} + A_2 u^n + C H^{n+1/2} = F, \tag{6}$$

$$\begin{aligned}\frac{\delta_{S_y}}{\tau}(H^{n+1} - H^n) + \frac{1-\delta_{S_y}}{\tau}(H^{n+1} - H^{n+1/2}) + \\ \delta_{S_y} A_1 u^{n+1/2} + A_2 u^{n+1} + \delta_{S_y} C H^{n+1} = \delta_{S_y} F.\end{aligned} \tag{7}$$

Similarly, characteristic scheme (5) is changed by

$$\frac{1}{\tau}(H^{n+1/2} - \tilde{H}^n) + A_1 u^{n+1/2} + A_2 u^n = F, \tag{8}$$

$$\frac{\delta_{S_y}}{\tau}(H^{n+1} - \tilde{H}^n) + \frac{1-\delta_{S_y}}{\tau}(H^{n+1} - H^{n+1/2}) + \delta_{S_y} A_1 u^{n+1/2} + A_2 u^{n+1} = \delta_{S_y} F. \tag{9}$$

Let us discuss the implementation of scheme (6),(7). In the points of S_y first equation of (6) has the form:

$$\frac{H^{n+1/2} - H^n}{\tau} - u^{n+1/2}_{x_2\bar{x}_2} - u^n_{x_1\bar{x}_1} + H^{n+1/2}_{\bar{x}_2} = F, \tag{10}$$

i.e. in the points of S_y we have one-dimensional problem (10), that we solve first. After, the rest part of the first equation in (6) is splitted into two non-coupled implicit schemes in the subdomains:

$$\begin{cases} \dfrac{H^{n+1/2} - H^n}{\tau} - u^{n+1/2}_{x_1\bar{x}_1} - u^{n+1/2}_{x_2\bar{x}_2} + H^{n+1/2}_{\bar{x}_2} = F, \text{ for } x \in \omega_1 \cup \omega_2, \\ \dfrac{H^{n+1/2} - H^n}{\tau} - \dfrac{2}{h_1} u^{n+1/2}_{x_1} - u^{n+1/2}_{x_2\bar{x}_2} + H^{n+1/2}_{\bar{x}_2} = F, \text{ for } x \in \gamma_N^-, \\ \dfrac{H^{n+1/2} - H^n}{\tau} + \dfrac{2}{h_1} u^{n+1/2}_{\bar{x}_1} - u^{n+1/2}_{x_2\bar{x}_2} + H^{n+1/2}_{\bar{x}_2} = F, \text{ for } x \in \gamma_N^+ \end{cases} \tag{11}$$

and these equations are accomplished by Dirichlet boundary conditions, given on γ_D and calculated from (10) on S_y.

Further, it is easy to check, that for the points $x \notin S_y$ the second equation (7) coincide with (11) and has the same Dirichlet boundary conditions on γ_D and on S_y, so, $u^{n+1}(x) = u^{n+1/2}(x)$ for $x \notin S_y$ and it makes no sense to solve these equations. It remains only to solve the system of the equations, corresponding to $x \in S_y$:

$$\frac{H^{n+1} - H^n}{\tau} - u^{n+1/2}_{x_2\bar{x}_2} - u^{n+1}_{x_1\bar{x}_1} + H^{n+1/2}_{\bar{x}_2} = F.$$

As $u^{n+1}(x) = u^{n+1/2}(x)$ for $x \notin S_y$, this system becomes

$$\begin{cases} \dfrac{H^{n+1} - H^n}{\tau} + H^{n+1/2}_{\bar{x}_2} + 2\dfrac{u^{n+1}(x_1, x_2)}{h_1^2} - u^{n+1/2}_{x_2\bar{x}_2} \\ -\dfrac{u^{n+1/2}(x_1 - h_1, x_2) + u^{n+1/2}(x_1 + h_1, x_2)}{h_1^2} = F, \ x \in S_y, \end{cases} \tag{12}$$

i.e. we get the system of scalar equations for $(u^{n+1}(x), H^{n+1}(x)),\ x \in S_y$.

Thus, the algorithm for the implementation of (6),(7) consists of 3 steps:

1): Predictor step: solving one-dimensional problem (10);
2): Main step: concurrent solving subproblems (11);
3): Corrector step: solving the system of scalar equations (12).

With the slight modifications the implementation of scheme (8),(9) is similar. Namely, in the points of S_y first equation of (8) has the form:

$$\frac{H^{n+1/2} - \tilde{H}^n}{\tau} - u^{n+1/2}_{x_2\bar{x}_2} - u^n_{x_1\bar{x}_1} = F, \tag{13}$$

i.e. in the points of S_y we have one-dimensional problem (13), that we solve first. After that the rest part of first equation in (8) is splitted into two non-coupled characteristic schemes in the subdomains:

$$\begin{cases} \dfrac{H^{n+1/2} - \tilde{H}^n}{\tau} - u^{n+1/2}_{x_1\bar{x}_1} - u^{n+1/2}_{x_2\bar{x}_2} = F, \text{ for } x \in \omega_1 \cup \omega_2, \\ \dfrac{H^{n+1/2} - \tilde{H}^n}{\tau} - \dfrac{2}{h_1} u^{n+1/2}_{x_1} - u^{n+1/2}_{x_2\bar{x}_2} = F, \text{ for } x \in \gamma_N^-, \\ \dfrac{H^{n+1/2} - \tilde{H}^n}{\tau} + \dfrac{2}{h_1} u^{n+1/2}_{\bar{x}_1} - u^{n+1/2}_{x_2\bar{x}_2} = F, \text{ for } x \in \gamma_N^+ \end{cases} \tag{14}$$

and these equations are accomplished by Dirichlet boundary conditions, given on γ_D and calculated from (13) on S_y.

Finally it remains to solve the system of the equations, corresponding to $x \in S_y$:

$$\frac{H^{n+1} - \tilde{H}^n}{\tau} - u^{n+1/2}_{x_2\bar{x}_2} - u^{n+1}_{x_1\bar{x}_1} = F.$$

As $u^{n+1}(x) = u^{n+1/2}(x)$ for $x \notin S_y$, this system becomes

$$\begin{cases} \dfrac{H^{n+1} - \tilde{H}^n}{\tau} + 2\dfrac{u^{n+1}(x_1, x_2)}{h_1^2} - u^{n+1/2}_{x_2\bar{x}_2} \\ -\dfrac{u^{n+1/2}(x_1 - h_1, x_2) + u^{n+1/2}(x_1 + h_1, x_2)}{h_1^2} = F,\ x \in S_y, \end{cases} \tag{15}$$

i.e. we get the system of scalar equations for $(u^{n+1}(x), H^{n+1}(x)),\ x \in S_y$.

Remark 1. Above we assumed that the calculation domain Ω is divided into two parts Ω_1 and Ω_2. This is not restrictive and we refer to [6] for more detailed discussion of the decomposition into several subdomains, also with corner and cross points. Even the case of curvilinear decomposition is studied in [6] and found to be stable and accurate under the natural assumptions for the mesh parameters.

Remark 2. We consider the 2D-case. The proposed methods have natural extensions to the 3D-case. We notice that in predictor and corrector steps the one-dimensional problems corresponding to the mesh line S_y are replaced by the two-dimensional problems corresponding to a plane S_{xy}.

4.1. **Multidecomposition method.** The general idea of the multidecomposition is to divide the subdomain to smaller subdomains i.e. we use two-level decomposition of the calculation domain. The main reason is to decrease the algebraic size of the problems and thus make the calculation times much smaller. The division of the subdomains is presented in the figure 1. The idea of using multilevel decomposition is not new but the technique presented in this paper gives a good and effective method when solving large complex time dependent problems.

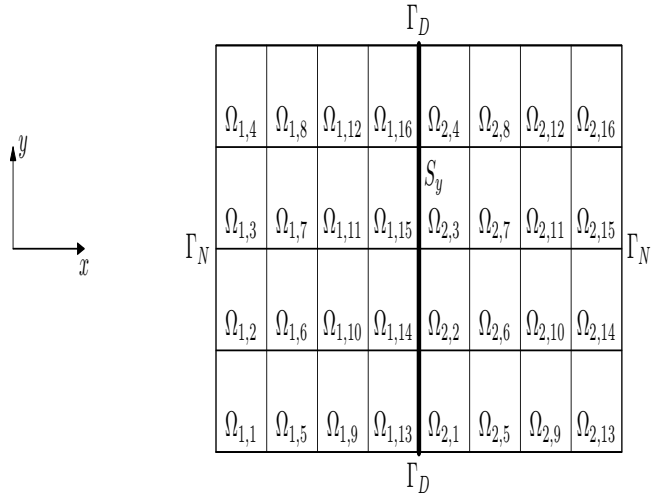

FIGURE 1. Used nonoverlapping domain decomposition and multidecomposition of the subdomains.

We use the notation $\Omega_i = \bigcup_{j_i=1}^{p_i} \Omega_{i,j_i}$. The calculation algorithm for characteristic mesh scheme (13)-(15) is presented below. The algorithm for implicit mesh scheme (10)-(12) is similar.

Algorithm 1. :

1.: On a time level n perform on the main processor the predictor step (13) on the artificial boundary S_y.

2.: Send the values of $u^{n+1/2}$ and $H^{n+1/2}$ on the boundary S_y to the slave processors.

3.: Concurrently on the slave processors perform the predictor step (13) on the artificial boundaries of the subdomains $\Omega_{i,j_i},\ i = 1, 2,\ j_1 = 1, ..., p_1,\ j_2 = 1, ..., p_2$.

4.: Concurrently on the slave processors perform sequentially the main step (14) on the subdomains Ω_{i,j_i}.

5.: Concurrently on the slave processors perform the corrector step (15) on the artificial boundaries of the subdomains $\Omega_{i,j_i},\ i = 1, 2,\ j_1 = 1, ..., p_1,\ j_2 = 1, ..., p_2$.

6.: Send the values of u^{n+1} and H^{n+1} from slave processors to the master processor.

7.: On the main processor perform the corrector step (15) on the artificial boundary S_y.
8.: Put $n = n + 1$, if the final time t_f is reached then **STOP**, else **GOTO 1.**

Remark 3. On the step **3.** of the algorithm 1 we do not do the predictor step (13) on the artificial boundary S_y.

Remark 4. On the steps **3.-5.** we do the calculations concurrently. However, we do not synchronize our calculation in such a way that all processors move from step to another at the same time. The step **6.** is synchronization point and it is performed at the same time.

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. Namely, our algorithm 1 is replaced by the following one.

Algorithm 2. :

1.: On the time step n perform on the main processor the predictor step (13) on the artificial boundary S_y.
2.: Send the values of $u^{n+1/2}$ and $H^{n+1/2}$ on the boundary S_y to the slave processors.
3.: Concurrently on the slave processors perform the predictor step (13) on the artificial boundaries of the subdomains $\Omega_{i,j_i},\ i = 1, 2,\ j_1 = 1, ..., p_1,\ j_2 = 1, ..., p_2$.
4.: Concurrently on the slave processors perform sequentially the main step (14) on the subdomains Ω_{i,j_i}.
5.: Concurrently on the slave processors perform the corrector step (15) on the artificial boundaries of the subdomains $\Omega_{i,j_i},\ i = 1, 2,\ j_1 = 1, ..., p_1,\ j_2 = 1, ..., p_2$.
6.: On the slave processors perform the smoothing step i.e. few iterations of the MSOR-method (modified SOR-method) over the whole subdomain Ω_i.
7.: Send the subsolutions u^{n+1} and H^{n+1} from slave processors to the main processor.
8.: On the main processor perform the corrector step (15) on the artificial boundary S_y.
9.: On the master 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 then **STOP**, else **GOTO 1.**

Remark 5. In the algorithm 2 we use the smoothing on steps **6.** and **9.** According to our knowledge the good amount of MSOR-iterations is less than 10. This amount ensures the reducing of the error and do only slightly increase the calculation time.

Remark 6. The step **9.** in the algorithm 2 is performed in the neighborhood of the artificial boundary S_y. The good width of the smoothing area is about 10 grid lines i.e from artificial boundary 5 grid lines to both directions.

Remark 7. In both algorithms 1 and 2 we use the predictor step **3.** and corrector step **5.** which require the decomposition along and against the convection. There are also cross points. In the article of Lapin and Pieskä [6] are discussed more carefully the implementation of such kind of decomposition.

Remark 8. The use of only two subdomains and processors is not restrictive. The usage of more processors and subdomains is straightforward. The numerical results presented in the section 5 verify this.

4.2. **Multidecomposition method with one processor.** The multidecomposition algorithms 1 and 2 are very effective and extremely quick in the case of many processors. We used the idea of multidecomposition for the situation where we have only one processor. Now we do not have artificial boundaries which decouples problem into subproblems. Anyway, we still have boundaries between subdomains inside the whole domain. The multidecomposition algorithm for one processor with characteristic mesh scheme read as follows.

Algorithm 3. :

1.: On every time level n perform the predictor step (13) on the artificial boundaries of the subdomains $\Omega_{1,j},\ j = 1, ..., p$.
2.: Perform sequentially the main step (14) on the subdomains $\Omega_{1,j}$.
3.: Perform the corrector step (15) on the artificial boundaries of the subdomains $\Omega_{1,j},\ j = 1, ..., p$.
4.: Perform the smoothing step i.e. few iterations of the MSOR-method over the whole calculation domain Ω.
5.: Put $n = n + 1$, if the final time t_f is reached then **STOP**, else **GOTO 1.**

5. Numerical experiments with one processor

Let $\Omega =]0,1[\times]0,1[$ with the boundary Γ divided in two parts such that $\Gamma_D = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = 1\}$ and $\Gamma_N = \Gamma \setminus \Gamma_D$, moreover let $t_f = 1$. Let us consider the case where the phase change temperature $u_{SL} = 1$ and the latent heat $L = 1$. Let the phase change interval be $[u_{SL} - \varepsilon, u_{SL} + \varepsilon]$, $\varepsilon = 0.01$, and the velocity is $v(t) = \frac{1}{5}$. Our numerical example is

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

where Kirchoff's temperature is according to it's 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 enthalpy

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

Furthermore, let the known right-hand side be

$$f(x;t) = \begin{cases} 4e^{-4t} + \frac{1}{5}(4x_2 - 2) - 4 & \text{if } u < u_{SL}, \\ 12e^{-4t} + \frac{1}{5}(12x_2 - 6) - 8 & \text{if } u > u_{SL}. \end{cases}$$

The exact solution of our problem is

$$u(x_1, x_2; t) = (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 - \frac{1}{2}e^{-4t} + 1.$$

The stopping criterion of the calculations was the L_2-norm of residual: $\|r\|_{L_2(\Omega)} < 10^{-4}$. We divide the domain Ω into non-overlapping subdomains Ω_i presented in the figure 1.

We tested algorithm 3 in two different cases. First, we fixed the number of grid points and changed the number of inner subdomains. The calculation grid was 129×129 in space and we took 256 time steps. The calculation times for different number of inner subdomains are presented in the table 1.

♯ of inside subdomains	Time [s]
1×1	112.9 s
2×2	76.2 s
3×3	66.1 s
4×4	56.9 s
8×8	39.9 s

TABLE 1. Calculation times in seconds for MDD when the grid size is fixed and number of inside subdomains are changed.

In the table 1 the 1×1 subdomain division means the calculation times for MSOR method for the whole calculation domain.

In the second test case we changed the calculation grid and number of time steps. We fixed the number of inner subdomains to be 4×4. For many calculation grid this decomposition is not the optimal one but it very clearly emphasizes the advantage of MDD. The results are in the table 2.

Grid	SEQ	MDD
$65 \times 65 \times 128$	8.67 s	4.75 s
$129 \times 129 \times 256$	112.9 s	56.9 s
$257 \times 257 \times 512$	1425 s	687 s

TABLE 2. Calculation times in seconds for sequential MSOR (SEQ) and MDD when the calculation grid is changed.

In the figure 2 we measured the time which different algorithms (MSOR and MDD) spend at different time levels. In this case we had the grid size

129×129. Normally for this kind of mesh we would take 256 times steps. Now we liked to test how these methods behave when the solution goes to steady-state situation. We took 2560 time steps i.e. 10 times more than usually to see the behavior. The MDD is much better comparing to MSOR in the beginning of the calculations when we have big changes in our simulation.

When time increases the calculation time drop very rapidly and both methods spend equal amount of time for each time step. This is good result for MDD because it is very effective for the case when we have big changes. Near steady-state situation it is as good as MSOR.

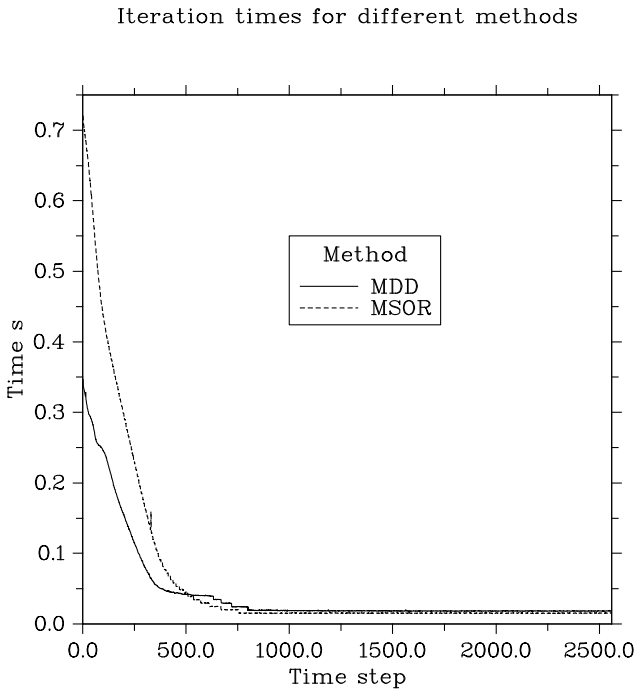

FIGURE 2. Time spend on each time level for different methods.

6. 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 dimensions of the subproblems inside the subdomains are very small and thus they are very quick to solve. The dimension is in many cases so small that even direct solvers could be effective.

The numerical results for one processor seem to be very promising. We do not need to have big parallel computers to achieve advantages of parallel computer. Only few processors are enough and in some cases even only one processor is good. The table 2 very clearly shows the advantages of MDD.

In that table the decomposition is clearly not optimal. The table 1 is good indicator to that. Anyhow, even with poor knowledge of the problem MDD method can be used. When the system is stable and big changes do not appear (like the number of grid points changes or time step changes) then MDD can be optimized good and numerical advantages comparing with MSOR method in the case of one processor and with additive Schwarz alternating method in the case of many processor becomes very dramatical.

References

[1] Z. Chen, *Numerical solutions of a two-phase continuous casting problem*, Numerical Methods for Free Boundary Problem, P. Neittaanmäki, eds. International Series of Numerical Mathematics 99., Birkhäuser, Basel., pp. 103-121, 1991.

[2] C. N. Dawson, Q. Du and T. F. Dupont, *A finite difference domain decomposition algorithm for numerical solution of the heat equation*, Mathematics of Computation, vol. 57, pp. 63-71, 1991.

[3] J. Jr. Douglas and T.F. Russel, *Numerical methods for convection-dominated diffusion problem based on combining the method of characteristic with finite element or finite difference procedures*, SIAM J. Numer. Anal., V. 19., pp. 871-885, 1982.

[4] E. Laitinen and A.V. Lapin, *Semi-Implicit Mesh Scheme and Splitting Iterative Methods for the Solution of Continuous Casting Problem*, Preprint, University of Oulu, Finland, 19p., 1999

[5] E. Laitinen, A.V Lapin and J. Pieskä, *Splitting iterative methods and parallel solution of variational inequalities*, Lobachevskii Journal of Mathematics, Vol. 8, p. 167-184, 2001. http://ljm.ksu.ru/vol8/latin.htm

[6] A.V Lapin and J. Pieskä, *On the parallel domain decomposition algorithms for time-dependent problems*, Lobachevskii Journal of Mathematics, Vol. 10, p. 27-44, 2002. http://ljm.ksu.ru/vol10/lpp.htm

[7] A.V. Lapin and D.O. Solovyev, *Splitting iterative methods for variational inequalities*, Preprint No 783, Center of Calcul., Novosibirsk, 24 p., 1988

[8] J.M Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press Inc. Orlando, Florida 1970.

[9] W. Rivera, J. Zhu and D. Huddleston, *An efficient parallel algorithm with application to computational fluid dynamics*, To be appear in Computers and Mathematics with Applications.

[10] W. Rivera and J. Zhu, *A scalable parallel domain decomposition algorithm for solving time dependent partial differential equations*, Proceedings of the International Conference on Parallel and Distributed Processing Technology and Applications, edited by H. R. Arabnia, CSREA Press, Athens GA, pp. 240-246, 1999.

[11] W. Rivera, J. Zhu and D. Huddleston, *An efficient parallel algorithm for solving unsteady nonlinear equations*, Proceedings of the International Conference on Parallel Processing Workshops, edited by T. M. Pinkston, IEEE Computer Society, Los Alamitos, California, pp. 79-84, 2001.

[12] J. F. Rodrigues and F. Yi, *On a two-phase continuous casting Stefan problem with nonlinear flux*, Euro J. App. Math., V. 1., pp. 259-278, 1990.

[13] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Clarendon Press Oxford, New York 1999.

[14] A.Samarskii, P.Vabischevich, *Finite difference schemes with operator multipliers*, Minsk, Institute of Mathematics of Belorussia, 442pp, 1998.

[15] A.Samarskii, P.Vabischevich, *Factorized regional-additive schemes for convection-diffusion problems (in Russian)*, Reports of Rusian Acad.Sciences (Mathematics), 1996, V.346, P/742-745.

[16] P.Vabischevich, *Parallel domain decomposition algorithms for time-dependent problems of mathematical physics* , Advances in Numerical Methods and applications. Singapore: World scientific, 1994, P.293-299.

University of Oulu, Finland.
Kazan State University, Russia.
E-mail address: alapin@ksu.ru
University of Oulu, Finland.

ReceivedOctober 15, 2003