Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

49. Asynchronous domain decomposition methods for solving continuous casting problem

E. Laitinen[1], A. Lapin[2], J. Pieskä[3]

1. Introduction. The general idea of the Schwarz alternating methods is to solve the boundary value problem restricted to each subdomain, using as the boundary conditions the function values of the approximative solution of the neighboring subdomains. One of the advantages of the additive Schwarz is that the solutions in the subdomains can be handled by the different processors of a parallel computer. However, due to the mutual waits among the processors when a synchronous method is applied, it leads to a substantial loss of computing time. To exploit the asynchronous parallel computing capacity of a multiprocessor system, we propose and study theoretically and numerically the asynchronous algorithms [1] for solving nonlinear finite-dimensional problem.

2. Continuous casting problem. A continuous casting problem can be stated mathematically as follows. Let $\Omega = \{0 < x_1 < L_{x_1}, 0 < x_2 < L_{x_2}\}$ be the rectangular domain with the boundary $\Gamma = \partial\Omega$ consisting of two parts: $\Gamma_1 = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = L_{x_2}\}$ and $\Gamma_2 = \{x \in \partial\Omega \setminus \Gamma_1\}$. We assume that the domain $\Omega \subset \mathbb{R}^2$ is occupied by a thermodynamically homogeneous and isotropic steel. We denote by $H(x,t)$ the enthalpy related to the unit mass and by $u(x,t)$ the temperature for $(x,t) \in \Omega\times]0,T[$. We have a constitutive law

$$H = H(u) = \rho \int_0^u c(\Theta)d\Theta + \rho L(1 - f_s(u)) \text{ in } \Omega\times]0,T[,$$

where ρ is density, $c(u)$ is specific heat, L is latent heat and $f_s(u)$ is solid fraction. For a steel casting process the graph $H(u)$ is an increasing function $\mathbb{R} \to \mathbb{R}$, involving nearly vertical segments, which correspond to a phase transition states, namely, for $u \in [T_L, T_S]$ where $0 < T_L < T_S$ are melting and solidification temperatures. When a copper casting problem is studied, the graph $H(u)$ has a vertical segment for $u = T_L = T_S$. We denote by $H(u),\ u \in \mathbb{R}$, a maximal monotone, generally multivalued, graph.

We also suppose, that the graph $H(u)$ is uniformly monotone: there exists a positive constant α such that

$$(\gamma_1 - \gamma_2, u_1 - u_2) \geq \alpha(u_1 - u_2, u_1 - u_2) \forall u_1, u_2 \forall \gamma_i \in H(u_i). \tag{2.1}$$

Now a continuous casting process can be described by a boundary-value problem, formally written in the following pointwise form: find $u(x,t)$ and $\gamma(x,t)$ such that

$$\text{(P)} \begin{cases} \dfrac{\partial\gamma}{\partial t} + v\dfrac{\partial\gamma}{\partial x_2} - \Delta u = 0 \text{ for } x \in \Omega, t > 0, \\ u = z(x_1,t) \text{ for } x \in \Gamma_1, t > 0, \\ \dfrac{\partial u}{\partial n} + au + b|u|^3 u = g, a \geq 0, b \geq 0, \text{ for } x \in \Gamma_2, t > 0, \\ \gamma = H_0(x) \text{ for } x \in \bar{\Omega}, t = 0, \\ \gamma(x,t) \in H(u(x,t)) \text{ for } x \in \Omega, t > 0. \end{cases}$$

Below we suppose, that the boundary temperature $z(x_1,t)$ at any point of Γ_1 and for all $t \geq 0$ does not coincide with the phase transition temperature $T_L = T_S$, in other words, the enthalpy function H has a single values at all these points. This corresponds to the physical

[1]University of Oulu, erkki.laitinen@oulu.fi
[2]Kazan State University, alapin@ksu.ru
[3]University of Oulu, jpieska@cc.oulu.fi

meaning of the problem, because the incoming material (points $x \in \Gamma_1 : x_2 = 0$) is in liquid state, while the outcoming material (points $x \in \Gamma_1 : x_2 = L_{\times_2}$) is in solid state. The existence and uniqueness of a weak solution for problem (P) are proved in [6].

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 the space variables a finite element method with the quadrature rules.

Let T_h be a triangulation of Ω in the rectangular elements δ of dimensions $h_1 \times h_2$ and $V_\mathrm{h} = \{u_\mathrm{h}(x) \in H^1(\Omega) : u_\mathrm{h}(x) \in Q_1 \text{ for all } \delta \in T_\mathrm{h}\}$, 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 $\delta \in T_\mathrm{h}$. 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 $V_\mathrm{h}^0 = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = 0 \text{ for all } x \in \Gamma_1\}$, $V_\mathrm{h}^\mathrm{z} = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = z_\mathrm{h} \text{ for all } x \in \Gamma_1\}$. Here z_h is the bilinear interpolation of z on the boundary Γ_1. For any continuous function $v(x)$ we define the quadrature formulas:

$$S_\delta(v) = \int_\delta \Pi_\mathrm{h} v dx, S_\Omega(v) = \sum_{\delta \in T_\mathrm{h}} S_\delta v,$$

$$S_{\partial\delta}(v) = \int_{\partial\delta} \Pi_\mathrm{h} v dx, S_{\Gamma_2}(v) = \sum_{\partial\delta \in T_\mathrm{h} \cap \bar{\Gamma}_2} S_{\partial\delta}(v);$$

$$E_\delta(v) = \int_\delta P_\mathrm{h} v dx, E_\Omega(v) = \sum_{\delta \in T_\mathrm{h}} E_\delta(v).$$

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

$$\begin{cases} S_\Omega(\partial_{\bar{\mathrm{t}}}\gamma_\mathrm{h}\eta_\mathrm{h}) + E_\Omega(v(t)\dfrac{\partial\gamma_\mathrm{h}}{\partial x_2}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) \\ +S_{\Gamma_2}((au_\mathrm{h} + b|u_\mathrm{h}|^3 u_\mathrm{h})\eta_\mathrm{h}) = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0, \\ \gamma_\mathrm{h}(x,t) \in H(u_\mathrm{h}(x,t)) \text{ for all mesh nodes } x. \end{cases} \tag{2.2}$$

When constructing the semi-implicit mesh scheme the term $\left(\frac{\partial}{\partial t} + v(t)\frac{\partial}{\partial x_2}\right)\gamma$ is approximate by using the characteristics of the first order differential operator (similar to [2], [3]). 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 this term by: $\left(\dfrac{\partial}{\partial t} + v(t)\dfrac{\partial}{\partial x_2}\right)\gamma \approx \dfrac{1}{\tau}\left(\gamma(x_1, x_2, t) - \gamma(x_1, \tilde{x}_2, t-\tau)\right)$. We denote $\tilde{\gamma}(x, t-\tau) = \gamma(x_1, \tilde{x}_2, t-\tau)$. If $\tilde{x}_2 < 0$ then we put $\tilde{\gamma}(x, t-\tau) = \gamma(x_1, 0, t-\tau)$. Note, that $\gamma(x_1, 0, t-\tau) = H(z(x_1, t-\tau))$ with single values $H(z(x_1, t-\tau))$ of H at these points, as it was mentioned above. In what follows we use the notation $d_{\bar{\mathrm{t}}}\gamma = \frac{1}{\tau}(\gamma(x,t) - \tilde{\gamma}(x, t-\tau))$ for the difference quotient in each mesh point on time level t.

Now, the semi-implicit finite difference scheme for problem (P) is: for all $t \in \omega_\tau$, $t > 0$, find $u_\mathrm{h} \in V_\mathrm{h}^\mathrm{z}$ and $\gamma_\mathrm{h} \in V_\mathrm{h}$ such that

$$\begin{cases} S_\Omega(d_{\bar{\mathrm{t}}}\gamma_\mathrm{h}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) + S_{\Gamma_2}((au_\mathrm{h} + b|u_\mathrm{h}|^3 u_\mathrm{h})\eta_\mathrm{h}) \\ = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0, \\ \gamma_\mathrm{h}(x,t) \in H(u_\mathrm{h}(x,t)) \text{ for all mesh nodes } x. \end{cases} \tag{2.3}$$

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

$$(Au, \eta) = S_\Omega(\nabla u_{\text{h}} \nabla \eta_{\text{h}}) + S_{\Gamma_2}(a u_{\text{h}} \eta_{\text{h}}),$$
$$(Bu, \eta) = S_\Omega(\frac{1}{\tau} u_{\text{h}} \eta_{\text{h}}) + E_\Omega(v(t) \frac{\partial u_{\text{h}}}{\partial x_2} \eta_{\text{h}}),$$
$$(Cu, \eta) = S_{\Gamma_2}(b|u_{\text{h}}|^3 u_{\text{h}} \eta_{\text{h}}).$$

Further we define a vector f: $(f, \eta) = S_{\Gamma_2}(g\eta_{\text{h}}) + S_\Omega(\frac{1}{\tau}\gamma(u_{\text{h}}(x, t-\tau))\eta_{\text{h}})$. Let now $\tilde{z}_{\text{h}}(x) \in V_{\text{h}}$ be the function which is equal to z_{h} on $\bar{\Gamma}_1$ and 0 for all nodes in $\Omega \cup \Gamma_2$, then f_0 is defined by the equality:

$$(f_0, \eta) = S_\Omega(\nabla \tilde{z}_{\text{h}}, \nabla \eta_{\text{h}}) + E_\Omega(v(t) \frac{\partial \Pi_{\text{h}}(H(\tilde{z}_{\text{h}}))}{\partial x_2} \eta_{\text{h}}) \text{ for all } \eta_{\text{h}} \in V_{\text{h}}^0.$$

(Here we use again the fact, that the graph $H(u)$ is single-valued for $u = \tilde{z}_{\text{h}}(x)$, when x is a mesh point). Finally we get $F = f - f_0$.

In these notations the algebraic form for the implicit mesh scheme (2.2) at fixed time level is:

$$Au + B\gamma + Cu = F, \gamma \in H(u). \tag{2.4}$$

If we set $(Bu, \eta) = S_\Omega(\frac{1}{\tau} u_{\text{h}} \eta_{\text{h}})$ and $(f, \eta) = S_{\Gamma_2}(g\eta_{\text{h}}) + S_\Omega(\frac{1}{\tau}\tilde{\gamma}_{\text{h}}\eta_{\text{h}})$, then the semi-implicit mesh scheme (2.3) has also the algebraic form (2.5).

$$Au + B\gamma + Cu = F, \ \ \gamma \in H(u). \tag{2.5}$$

The matrices A, and B and the operators C, and H have the following properties:

$$A \text{ and } B \text{ are } M - \text{matrices}, \tag{2.6}$$

$$A \text{ is weakly diagonally dominant in columns: } \sum_{j \neq i}^{N_0} |a_{\text{ji}}|/a_{\text{ii}} \leq 1 \forall i; \tag{2.7}$$

$$B \text{ is strictly diagonally dominant in columns: } \sum_{j \neq i}^{N_0} |b_{\text{ji}}|/b_{\text{ii}} \leq \beta < 1 \forall i; \tag{2.8}$$

(in fact, for the semi-implicit scheme matrix B is diagonal); the operators γ and C have the diagonal forms:

$$\gamma(u) = (\gamma(u_1), \gamma(u_2), ..., \gamma(u_{N_0}))^{\text{t}}, Cu = (c_1(u_1), c_2(u_2), ..., c_N(u_{N_0}))^{\text{t}}, \tag{2.9}$$

where c_{i} are continuous non-decreasing functions and $\gamma(.)$ is maximal monotone and uniformly monotone graph (see (2.1)). Note, that $\beta = \dfrac{\tau}{\tau + h_2}$ for the case of the implicit finite difference scheme, while $\beta = 0$ for the semi-implicit scheme.

Below we use the following notations: $u \gg 0 \Leftrightarrow u_{\text{i}} \geq 0 \quad \forall i, \quad A \gg 0 \Leftrightarrow a_{\text{ij}} \geq 0 \quad \forall i, j$. There exist a subsolution $(\underline{u}, \underline{\gamma})$:

$$A\underline{u} + B\underline{\gamma} + C\underline{u} \leq F, \ \underline{\gamma} \in H(\underline{u}), \tag{2.10}$$

and a supersolution $(\bar{u}, \bar{\gamma})$:

$$A\bar{u} + B\bar{\gamma} + C\bar{u} \geq F, \ \bar{\gamma} \in H(\bar{u}) \tag{2.11}$$

for form (2.4). Under above assumptions, the following theorem can be proved [4], [5], .

Theorem 2.1 *The implicit mesh scheme (2.2) and the semi-implicit mesh scheme (2.3) have unique solutions.*

3. Asynchronous algorithms. In this section we present the asynchronous additive Schwarz alternating algorithms.

Algorithm 1 (ASM1)

1. Divide the domain Ω into p overlapping subdomains and construct approximative subproblems in these subdomains.
2. Solve simultaneously the subproblems in the slave processors.
3. When the local stopping criterion in a slave processor is reached, send information about this to the master processor and keep calculating further.
4. When all slaves have finished the calculations, send the subsolutions to the master processor for updating the information for all slave processors.
5. If the accuracy is reached, then **STOP**, else goto 2.

Algorithm 2 (ASM2)

1. Divide the domain Ω into p overlapping subdomains and construct approximative subproblems in these subdomains.
2. Solve simultaneously the subproblems in the slave processors.
3. When the local stopping criterion in a slave processor is reached, send subsolution to the master processor and check if there is a new information from the neighboring subdomains. If yes, then update it and restart the calculations, otherwise keep calculating further.
4. When all slaves have finished the calculations, send the subsolutions to the master processor for updating the information for all slave processors.
5. If the accuracy is reached, then **STOP**, else goto 2.

In **Algorithm 1** we do not use the newest available information. This slows convergence. Although it is much faster to just send a signal to the master that the processor is ready than send the whole subsolution.

In **Algorithm 2** we send the subsolution to the master whenever there is an improvement. This increase the total calculation time. On the other hand we use the newest available information which decreases the calculation time.

Intuitively if there is a large load imbalance, i.e. if some processors have substantially more work than others, one can expect the asynchronous versions to converge faster than the synchronous one. It is also expected that ASM2 would be faster than ASM1.

4. Iterative methods. In this section we study the convergence of asynchronous iterative methods. For simplicity but without loss of generality we suppose that the domain Ω is decomposed into two overlapping subdomains Ω_1 and Ω_2, consisting of the elements of a triangulation T_h. We arrange the nodes of the mesh as follows. First, we enumerate the nodes lying in the non-overlapping part of the first subdomain, namely $x \in (\bar{\Omega}_1 \setminus \bar{\Gamma}_1) \setminus \overline{\Omega_1 \cap \Omega_2}$, then the nodes in the overlapping zone $x \in \overline{\Omega_1 \cap \Omega_2} \setminus \bar{\Gamma}_1$ and at last the nodes in the non-overlappping part of the second subdomain. A vector $u \in \mathbb{R}^{\mathrm{N}_0}$, $u \Leftrightarrow u_\mathrm{h}(x)$, takes the form $u = (u_{11}, u_{12}, u_{22})^\mathrm{t}$ with the subvectors u_ij corresponding to enumeration of the nodes.

This decomposition implies also the partitioning of the matrices and nonlinear operators: $A = (A_\mathrm{ij})_{\mathrm{ij}=1}^3$, $B = (B_\mathrm{ij})_{\mathrm{ij}=1}^3$, $C = \mathrm{diag}(C_1, C_2, C_3)$. Note, that $A_\mathrm{ij} \ll 0$, $B_\mathrm{ij} \ll 0$ for $i \neq j$ and the blocks A_{13}, A_{31}, B_{13}, B_{31} are equal to zero.

We use also the following notations:

$$A_0^1 = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, B_0^1 = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, A_1^1 = \mathrm{diag}(0, A_{23}),\ B_1^1 = \mathrm{diag}(0, B_{23});$$

$$A_0^2 = \begin{pmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{pmatrix}, B_0^2 = \begin{pmatrix} B_{22} & B_{23} \\ B_{32} & B_{33} \end{pmatrix}, A_1^2 = \mathrm{diag}(A_{21}, 0),\ B_1^2 = \mathrm{diag}(B_{21}, 0);$$

$$C^1 = \mathrm{diag}(C_1, C_2),\ C^2 = \mathrm{diag}(C_2, C_3).$$

Let further $u_1 = (u_{11}, u_{12})^{\mathrm{t}},\ u_2 = (u_{12}, u_{22})^{\mathrm{t}}$ and similar for all other vectors.

Then ASAM has the form (4.1), (4.2):

$$\begin{cases} A_0^1 v_1^{\mathrm{k}+1} + B_0^1 \eta_1^{\mathrm{k}+1} + C^1 v_1^{\mathrm{k}+1} = F_1 - A_1^1 u_2^{\mathrm{k}} - B_1^1 \gamma_2^{\mathrm{k}}; \eta_1^{\mathrm{k}+1} \in H(v_1^{\mathrm{k}+1}), \\ A_0^2 w_2^{\mathrm{k}+1} + B_0^2 \xi_2^{\mathrm{k}+1} + C^2 w_2^{\mathrm{k}+1} = F_2 - A_1^2 u_1^{\mathrm{k}} - B_1^2 \gamma_1^{\mathrm{k}}; \xi_2^{\mathrm{k}+1} \in H(w_2^{\mathrm{k}+1}), \end{cases} \tag{4.1}$$

$$\begin{cases} u_{11}^{\mathrm{k}+1} = v_{11}^{\mathrm{k}+1},\ u_{22}^{\mathrm{k}+1} = w_{22}^{\mathrm{k}+1},\ u_{12}^{\mathrm{k}+1} = \alpha v_{12}^{\mathrm{k}+1} + (1-\alpha) w_{12}^{\mathrm{k}+1}, \\ \gamma_{11}^{\mathrm{k}+1} = \eta_{11}^{\mathrm{k}+1},\ \gamma_{22}^{\mathrm{k}+1} = \xi_{22}^{\mathrm{k}+1},\ \gamma_{12}^{\mathrm{k}+1} = \alpha \eta_{12}^{\mathrm{k}+1} + (1-\alpha) \xi_{12}^{\mathrm{k}+1}, \end{cases} \tag{4.2}$$

with an initial guess (u^0, γ^0) and $\alpha \in (0, 1)$.

Let now every subproblem in (4.1) be solved by using a finite number of iterations of an inner iterative algorithm. Then we derive a two-stage additive Schwarz alternating method.

Let for $i = 1, 2\ A_0^{\mathrm{i}} = M_{\mathrm{i}} + N_{\mathrm{i}}, B_0^{\mathrm{i}} = K_{\mathrm{i}} + L_{\mathrm{i}}$ be regular splittings of A and B with $\mathrm{diag}(A_0^{\mathrm{i}}) \subseteq M_{\mathrm{i}},\ \mathrm{diag}(B_0^{\mathrm{i}}) \subseteq K_{\mathrm{i}}$ and $N_{\mathrm{i}} \ll 0, L_{\mathrm{i}} \ll 0$. Starting from the initial guess $z_{1,0} = u_1^{\mathrm{k}},\ z_{2,0} = u_2^{\mathrm{k}},\ \epsilon_{1,0} = \gamma_1^{\mathrm{k}},\ \epsilon_{2,0} = \gamma_2^{\mathrm{k}}$, we solve the subproblems in (4.1) by the iterative methods:

$$\begin{cases} M_1 z_{1,\mathrm{i}} + K_1 \epsilon_{1,\mathrm{i}} + C^1 z_{1,\mathrm{i}} = \varphi_1^{\mathrm{k}} - N_1 z_{1,\mathrm{i}-1} - L_1 \epsilon_{1,\mathrm{i}-1}, \\ \epsilon_{1,\mathrm{i}} \in H(z_{1,\mathrm{i}}),\ i = 1, \ldots, p_1, \end{cases} \tag{4.3}$$

$$\begin{cases} M_2 z_{2,\mathrm{i}} + K_2 \epsilon_{2,\mathrm{i}} + C^2 z_{2,\mathrm{i}} = \varphi_2^{\mathrm{k}} - N_1 z_{2,\mathrm{i}-1} - L_1 \epsilon_{2,\mathrm{i}-1}, \\ \epsilon_{2,\mathrm{i}} \in H(z_{2,\mathrm{i}}),\ i = 1, \ldots, p_2, \end{cases} \tag{4.4}$$

set $v_1^{\mathrm{k}+1} \equiv z_{1,\mathrm{p}_1},\ \eta_1^{\mathrm{k}+1} \equiv \epsilon_{1,\mathrm{p}_1};\ w_2^{\mathrm{k}+1} \equiv z_{2,\mathrm{p}_2},\ \xi_2^{\mathrm{k}+1} \equiv \epsilon_{2,\mathrm{p}_2}$ and then update the outer iterations using formulas (4.2).

Here $\varphi_1^{\mathrm{k}} = F_1 - A_1^1 u_2^{\mathrm{k}} - B_1^1 \gamma_2^{\mathrm{k}},\ \varphi_2^{\mathrm{k}} = F_2 - A_1^2 u_1^{\mathrm{k}} - B_1^2 \gamma_1^{\mathrm{k}}$ for method ASM1, when we calculate all subproblems by using inner iterative methods until we reach the desired accuracy in all subproblems and after that send the calculated $v_1^{\mathrm{k}+1},\ w_2^{\mathrm{k}+1},\ \eta_1^{\mathrm{k}+1},\ \xi_2^{\mathrm{k}+1}$ to the master processor to update the outer iterations to using formulas (4.2). On the other hand, for method ASM2 the formulas for φ_i^{k} are changed to $\varphi_1^{\mathrm{k}} = F_1 - A_1^1 w_2^{\mathrm{k}+1} - B_1^1 \xi_2^{\mathrm{k}+1}$ or to $\varphi_2^{\mathrm{k}} = F_2 - A_1^2 v_1^{\mathrm{k}+1} - B_1^2 \eta_1^{\mathrm{k}+1}$, depending on which of subproblems was solved faster.

Theorem 4.1 *Iterative method (4.3)– (4.4), (4.2) with an initial guess* $(u^0, \gamma^0) \in \langle (\underline{u}, \underline{\gamma}), (\bar{u}, \bar{\gamma}) \rangle$ *converges with geometric rate of convergence:*

$$||A^0(u^{\mathrm{k}+1} - u) + B^0(\gamma^{\mathrm{k}+1} - \gamma)||_1 \le q ||A^0(u^{\mathrm{k}} - u) + B^0(\gamma^{\mathrm{k}} - \gamma)||_1, \tag{4.5}$$

with $q = \dfrac{c_{\mathrm{AB}} + \alpha\beta}{c_{\mathrm{AB}} + \alpha} < 1,\ c_{\mathrm{AB}} = \max\limits_{1 \le i \le N_0} \dfrac{a_{\mathrm{ii}}}{b_{\mathrm{ii}}}$. *Here* $||v||_1 = \sum\limits_{i=1}^{N_0} |v_{\mathrm{i}}|$ *and* $c_{\mathrm{AB}} = \dfrac{2\tau(1 + h_2^2/h_1^2)}{h_2(\tau + h_2)}$ *for the implicit scheme, while* $c_{\mathrm{AB}} = \dfrac{2\tau(1 + h_2^2/h_1^2)}{h_2^2}$ *for the semi-implicit scheme. The parameter* α *is from equation (2.1) and* β *from (2.8).*

5. Numerical results. To validate theoretical results the following numerical example was considered. Let $\Omega =]0,1[\times]0,1[$ with the boundary Γ divided in two parts such that $\Gamma_{\mathrm{D}} = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = 1\}$ and $\Gamma_{\mathrm{N}} = \Gamma \setminus \Gamma_{\mathrm{D}}$, moreover let $T = 1$. Let us consider the case where the phase change temperature $u_{\mathrm{SL}} = 1$ and the latent heat $L = 1$ and the density $\rho = 1$. Let the velocity be $v(t) = \frac{1}{5}$. Our numerical example is

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

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

$$K(u) = \begin{cases} u & \text{if } u < u_{\mathrm{SL}}, \\ 2u-1 & \text{if } u \geq u_{\mathrm{SL}}, \end{cases} \quad \text{and } H(u) = \begin{cases} 2u & \text{if } u < u_{\mathrm{SL}}, \\ [2u_{SL}, 2u_{SL}+\rho L] & \text{if } u = u_{\mathrm{SL}}, \\ 6u - 4u_{SL} + \rho L & \text{if } u > u_{\mathrm{SL}}. \end{cases}$$

Furthermore the known right-hand side is

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

The stopping criterion of the outer iterations was the value of the L_2-norm of residual $\|r\|_{\mathrm{L}_2} = \|Au + B\gamma + \delta - f\|_{\mathrm{L}_2} \leq 10^{-3}$. We use through all the calculations the decomposition presented on the figure 5.1. The subdomain Ω_1 is roughly twice as big as other subdomains.

5.1. Implicit scheme. In our first test case we changed the number of grid points both in time and in space. We solved the problem by using the implicit scheme (2.2). The results can be seen in table 5.1. The over is the number of grid lines in the overlapping area. The inner iterations was performed till all of the processors have reached the desired accuracy $\|r\|_{\mathrm{L}_2} \leq 10^{-3}$. Due to this the number of inner iterations can be different for different processors. The synchronous Schwarz alternating method is denoted by (SASM)

Grid	over	ASM1 iterations	ASM1 T[s]	ASM2 iterations	ASM2 T[s]	SASM iterations	SASM T[s]
$65 \times 65 \times 128$	4	17	14.8	8	11.4	16	16.0
$129 \times 129 \times 256$	8	16	92.1	11	73.0	11	146
$257 \times 257 \times 512$	16	19	1184	17	1120	9	2776

Table 5.1: The number of outer iterations and calculation times in seconds for different grids for 4 processors; Implicit scheme.

5.2. Semi-Implicit scheme. We solve the same problem as for the implicit scheme to compare these methods against each other. The results can be seen in table 5.2.

6. Conclusions. Two mesh schemes with two different kind of discretizations for the convection term were considered, an implicit and a semi-implicit scheme. A model problem was solved by using both asynchronous methods ASM1 and ASM2. It can be seen from tables 5.1 and 5.2 that ASM2 takes fewer outer iterations than ASM1 and is thus the faster of the these two methods.

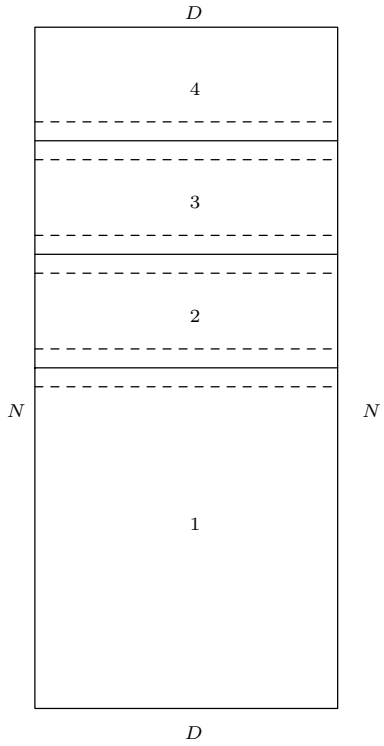

Figure 5.1: The decomposition used in model continuous casting problem.

Grid	over	ASM1 iterations	ASM1 T[s]	ASM2 iterations	ASM2 T[s]	SASM iterations	SASM T[s]
$65 \times 65 \times 128$	4	17	12.2	13	10.6	16	15.8
$129 \times 129 \times 256$	8	16	84.5	14	65.9	11	128
$257 \times 257 \times 512$	16	19	1171	17	1056	9	2528

Table 5.2: The number of outer iterations and calculation times in seconds for different grids for 4 processors: Semi-Implicit scheme.

Numerical results confirm the theoretical results. Our numerical results show that the calculation times of the asynchronous methods ASM1 and ASM2 are smaller than for the synchronous method SASM. In our opinion, ASM1 and ASM2 are faster for this kind of decomposition. We could also gain some advantage with asynchronous methods if the processors differ from each other.

REFERENCES

[1] Z.-Z. Bai. A class of asynchronous parallel iterations for the system of nonlinear algebraic equations. *Computers and Mathematics with Applications*, 39(7-8):81–94, 2000.

[2] 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.

[3] J. D. Jr. and T. F. Russel. Numerical methods for convection-dominated diffusion problem based on combining the method of characteristic with finite element or finite difference procedure. *Siam J. Numer Anal.*, 19:871–885, 1982.

[4] E. Laitinen, A. Lapin, and J. Pieska. Mesh approximation and iterative solution of the continuous casting problem. In P. Neittaanmaki, T. Tiihinen, and P. Tarvainen, editors, *ENUMATH 99 - Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications*, pages 601–617. World Scientific, Singapore, 1999.

[5] A. Lapin. Finite dimensional inclusions with several m-matricies and maximal monotone operators. Preprint, Department of Mathematical Sciences, University of Oulu, 2000.

[6] 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.