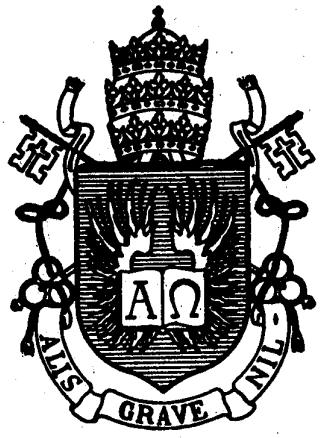

PUC

Series: Monografias em Ciência da Computação

Nº 11/79

A DIRECT METHOD FOR SOLVING TWO-DIMENSIONAL ONE PHASE STEFAN PROBLEM

by

Vitoriano Ruas

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 – CEP-22453

RIO DE JANEIRO – BRASIL

PUC/RJ - DEPARTAMENTO DE INFORMÁTICA

Series: Monografias em Ciência da Computação
Nº 11/79

Series Editor: Daniel A. Menascé August, 1979

A DIRECT METHOD FOR SOLVING TWO-DIMENSIONAL ONE PHASE STEFAN PROBLEM*

by

Vitoriano Ruas

* This Work has been partially sponsored by FINEP

AKNOWLEDGEMENTS

The author wishes to express his gratitude to Prof. H. Fujita of the Department of Mathematics of the University of Tokyo for extending a warm welcome to him and for giving valuable suggestions which were essential for the development of these results.

Thanks should be extended to the Japanese Ministry of Education (Monbushõ) and to the Brazilian National Research Council (CNPq) for the financial support.

This work has already been published in Memoirs of Numerical Mathematics, Tokyo, Japan, and is also being published in this series for wider dissemination.

RESUMO:

Um novo algoritmo para a solução do problema de Stefan é proposto, considerando-se o caso em que a região inicial ocupada por um meio condutor em uma dada fase é um domínio bi-dimensional estrelado. Obtém-se a evolução dessa região com o tempo, através da determinação direta de sua fronteira livre. Esta é aproximada passo a passo por um polígono, no interior do qual a equação do calor é resolvida com elementos finitos lineares por pedaços. Mostra-se, por meio de vários exemplos numéricos, que o método é estável e que se pode esperar convergência sob hipóteses adequadas de regularidade da fronteira e dos dados iniciais do problema.

Palavras-chave: Problema de Stefan unifásico, Linear por pedaços, Elementos finitos, Domínio estrelado bi-dimensional, Processo de triangulação automática, Fronteira livre, Equação do calor.

ABSTRACT:

A new algorithm is proposed for solving the Stefan problem for the case when the initial region occupied by the medium in a given phase is a starshaped two-dimensional domain. The evolution of this region as time increases is determined by plotting the free boundary directly. At each time step this is approximated by a polygon in the interior of which the heat equation is solved with piecewise linear infinite elements. Numerical experiments indicate that the method is stable and that the convergence is to be expected, under suitable assumptions on the regularity of both the boundary and initial data of the problem.

Keywords: One phase Stefan problem, piecewise linear, finite elements, starshaped two-dimensional domain, automatic triangulation process, free boundary, heat equation.

1. INTRODUCTION

We shall be concerned here with an algorithm for solving one-phase Stefan problems in two-dimension space. The basic tool for the algorithm is a method of automatic generation of triangular finite element meshes that the author initially proposed for solving boundary value problems defined on stationary starshaped domains [8]. In that work many technical details of its implementation are given, while regularity properties of the so generated mesh have been analysed in [9].

This automatic triangulation process is used here both for plotting the free boundary and for adjusting the mesh, so that the heat equation can be conveniently solved just in the domain occupied by one of the phases of a certain medium undergoing a change of phase. This is what we call a direct method of solution of a Stefan problem.

Actually, our algorithm generalizes the one proposed by Mori [6] for the one-dimensional case. In his work the free boundary is determined after each increase in time of a fixed value Δt, while the mesh containing a fixed number of intervals moves according to the new shape of the domain. As it was well remarked by Prof. Fujita [3], our triangulation process could be the appropiate means of doing the same thing in the two-dimensional case, provided that the other boundary of the initial domain is starshaped and that it remains starshaped as it evolves in time. Of cource, since the domain is in itself one of the unknowns of the problem, the latter condition cannot be satisfied a priori. However, one might expect that, in most practical cases, the domain becomes more and more regular as time increases, in the sence that it actually remains starshaped,

although rigorous proofs are not available yet.

Let us also say that Morí applied with success a slight generalization of his one phase algorithm to a two-dimensional strip domain [7]. Later Bonnerot & Jamet proposed an algorithm to direct solution of one phase two-dimensional problems, based on time space finite elements [5][1]. Although their method works well in many cases, as shown by their examples, they did not propose any solution to the problem of dealing with the spatial mesh as time increases. We believe this may cause either numerical inconvenience or difficulties of implementation for practical cases, unless a suitable solution of this problem is a part of the algorithm itself (see comments in Section 5).

Finally we note that a number of reasonable algorithms for indirect solution of the two-dimensional two phase problem are available at present. As a significant example of those we mention the work of Ciavaldini [1], that is based on a suitable transformation of the heat equation into a semilinear problem. This is solved on the fixed bounded domain occupied by the medium in both phases, and the position of the free boundary is then determined with the help of a function of the solution of the transformed equation. By introducing appropriate modifications, this approach can also be used for solving the pure one phase problem. However, in this case, although one can linearize the discrete problem it is necessary to perform the calculations for a domain much larger than the one occupied by the only phase undergoing changes of temperature. This can be impractical in the case where the domain evolves to very large regions compared to the initial one.

1. Actually, their algorithm gives aproximate problems very close to Morís, in the cases he considers.

2. The Continuous Problem

As we will comment in Section 5, with the eventual introduction of simple modifications, our algorithm can be applied to the solution of a very wide class of Stefan problems. However, we shall confine ourselves here to the solution of the special case of continuous problem defined on domains whose growth is not at all limited a priori, as described below. Our algorithm seems to attain the best of its efficiency compared to others particularly in the case where the domain may become much larger than the initial one.

In the sequel we introduce the formulation of the one phase Stefan problem we shall consider here, as a special case of the general multidimensional one treated by Friedman [2].

Let a certain medium exist in two different phases 1 and 2. Phase 1 occupies initially a certain bounded region Ω^0 of the x-y plane, whose boundary consists of two disjoint curves Γ^0 and Γ^*. Γ^0 is the interface separating both phases at time $t = 0$ and Γ^* is a fixed curve where heat sources are to be applied. We assume that Γ^* lies in the interior of Γ^0 and we denote its own interior by Ω^*.

We assume also that both $\Omega^0 \cup \bar{\Omega}^*$ and Ω^* are starshaped domains such that there exists one point $0 \in \Omega^*$ for which the segments joining it to any other point either of Ω^* or $\Omega^0 \cup \bar{\Omega}^*$ lie completely in those sets, respectively. An example of sets Ω^* and Ω^0 satisfying this assumption is illustrated in Figure 1.

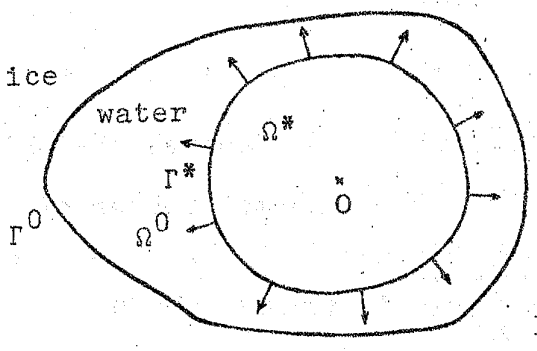

Figure 1

As time increases a change of phase 2 into phase 1 will occur in the medium lying in the region beyond Γ^0. So, the corresponding problem will be defined on the region $\mathbb{R}^2 - \Omega^*$.

As a practical example of this situation we have the problem of melting ice by heating some already melted region Ω^0 with a pipeline represented by Γ^* (Figure 1).

We shall denote by $\Omega(t)$ and $\Gamma(t)$ the domain occupied by the medium in phase 1 and its outer boundary respectively, at time t, $t \geq 0$. Clearly we have $\Omega(0) = \Omega^0$, $\Gamma(0) = \Gamma^0$ and that $\Gamma(t) \cup \Gamma^*$ is the boundary of $\Omega(t)$.

Let

$$\phi(x, y, t) = 0 \tag{1}$$

be the equation of the free boundary $\Gamma(t)$. We assume that ϕ is such that $\phi(x, y, t) < 0$ for $(x, y) \in \Omega(t) \cup \overline{\Omega}^*$ and $\phi(x, y, t) > 0$ for $(x, y) \notin \overline{\Omega(t) \cup \Omega^*}$. Now, supposing that the heating process takes place up to time T, for the temperature u of the medium in phase 1 we have the heat equation:

$$\frac{\partial u}{\partial t} = \Delta u \quad \text{in} \quad \Omega(t) \qquad \forall t \in [0, T] \tag{2}$$

with the initial condition:

$$u(x, y, 0) = u_0(x, y) \qquad \text{in} \quad \Omega^0 \tag{3}$$

and the boundary conditions:

$$u(\vec{x}, t) = g(\vec{x}, t) \tag{4}$$

or

$$\frac{\partial u}{\partial \nu}(\vec{x}, t) = g'(\vec{x}, t) \tag{4}'$$

(ν denotes the unit outer normal vector of Γ^*),

for $\vec{x} \in \Gamma^*$, $\forall t \in [0, T]$,

and

$$(5) \quad u(\vec{x}, t) = 0 \quad \text{for} \quad \vec{x} \in \Gamma(t), \quad \forall t \in [0, T].$$

Also for each time t, the so-called Stefan condition holds on $\Gamma(t)$

$$(6) \quad (\nabla\phi, \nabla u)\big|_{\vec{x}\in\Gamma(t)} = \kappa \frac{\partial \phi}{\partial t} .$$

where κ is a positive constant and $(\cdot, \cdot)$ denotes the scalar product in $\mathbb{R}^2$.

Thus the problem we want to solve is finding ϕ and u to satisfy (2) $\sim$ (6).

As we have already mentioned, the algorithm we shall employ for solving this problem, is constructed upon a process of automatic triangulation. This in turn is based on the representation of the free boundary by an equation in polar coordinates (ρ, θ). So it will be useful to take ϕ to be an expression of the form:

$$(7) \quad \phi = \rho - s(\theta, t).$$

In order to do so we must choose first of all a suitable origin of coordinates lying in Ω^*. According to the assumptions that we have made it is possible to choose such an origin so that both Γ^* and $\Gamma(t)$ can be represented in polar coordinates[2]. Thus it makes sense to write ϕ in form (7), and we have:

$$\nabla\phi = \left(\cos\theta + \frac{1}{s}\frac{\partial s}{\partial \theta}\sin\theta, \ \sin\theta - \frac{1}{s}\frac{\partial s}{\partial \theta}\cos\theta\right)$$

and

---

[2] Strictly speaking $\Omega^0 \cup \overline{\Omega}^*$ should be "non-singular" star-shaped. For the corresponding definition and other details see [9].

and

$$\frac{\partial \phi}{\partial t} = - \frac{\partial s}{\partial t} .$$

Thus (6) becomes:

$$(8) \qquad \left[\frac{\partial u}{\partial \rho} - \frac{1}{s}\frac{\partial s}{\partial \theta}\frac{\partial u}{\partial \sigma}\right]_{|(\rho,\theta)\in\Gamma(t)} = - \kappa \frac{\partial s}{\partial t}$$

where $\frac{\partial u}{\partial \rho}$ and $\frac{\partial u}{\partial \sigma}$ are, respectively, derivatives of u in two perpendicular directions (see Figure 2). The meaning of the first derivative is clear whereas the second one is simply $\frac{1}{\rho}\frac{\partial u}{\partial \theta}$.

On the other hand, according to (5), the tangential derivative $\frac{\partial u}{\partial \tau}$ on $\Gamma(t)$ must vanish.

So denoting by α the angle between the polar radius and by ν the outer normal on $\Gamma(t)$, we have:

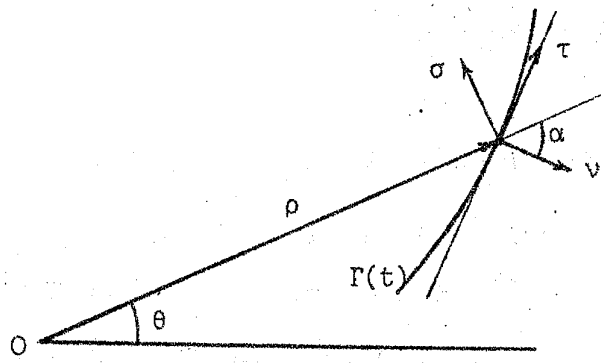

Figure 2

$$(9) \qquad \frac{\partial u}{\partial \rho}\sin\alpha + \frac{\partial u}{\partial \sigma}\cos\alpha = 0.$$

Using (8) and (9) and taking into account that

$$\tan\alpha = \frac{1}{s}\frac{\partial s}{\partial \theta} ,$$

we get:

$$(10) \qquad \left[1+\left(\frac{1}{s}\frac{\partial s}{\partial \theta}\right)^2\right]\frac{\partial u}{\partial \rho}\Big|_{\rho=s(\theta,t)} = -\kappa\frac{\partial s}{\partial t}.$$

(10) is the expression of the Stefan condition when u is given in polar coordinates.

3. The Algorithm

We shall now describe the algorithm that we propose for solving (2) $\sim$ (6) numerically.

First of all the discretization of (2) $\sim$ (5) is performed by standard methods applied to the corresponding variational formulation, namely:

$$(11) \qquad \int_{\Omega(t)} \frac{\partial u}{\partial t} v\,dx = -\int_{\Omega(t)} \nabla u \nabla v\,dx \quad \forall v \in V, \quad u \in \tilde{V}$$

if the Dirichlet-type boundary condition (4) holds, or

$$(11)' \qquad \int_{\Omega(t)} \frac{\partial u}{\partial t} v\,dx = -\int_{\Omega(t)} \nabla u \nabla v\,dx + \int_{\Gamma^*} g' v\,ds \quad \forall v \in V', \quad u \in V'$$

in case the Neumann-type boundary condition (4)' holds. Here V and V' are the subspaces of $H^1[\Omega(t)]$ of functions that vanish everywhere on $\Gamma(t) \cup \Gamma^*$ and on $\Gamma(t)$, respectively, and $\tilde{V}$ is the linear variety of V consisting of functions w such that $w = g$ on Γ^*.

For the discretization in space of (11) and (11)' we use a triangulation of $\Omega(t)$ on which we construct finite element spaces V_h or V_h' of piecewise linear continuous functions. Functions g and g' will be replaced by their piecewise linear interpolate g_h and g_h' which coincide with them at the nodes lying on Γ^*, respectively. Thus in case (4) holds, the approximate solution u_h will belong to $\tilde{V}_h$, discrete analogue of $\tilde{V}$, i.e., the linear variety of V_h such that $w_h \in \tilde{V}_h$

implies $w_h = g_h$ on the inner polygonal boundary approximating Γ^*. If (4)' holds, integration will be performed along this polygon instead of Γ^*.

For the discretization in time we employ standard schemes. This means that once obtained a linear system of ordinary differential equations after discretization in space:

$$M_h(t)\frac{\partial u_h(t)}{\partial t} + A_h(t)u_h(t) = 0 \ , \tag{12}$$

we start with $u_h^0 = u_{0_h}$, u_{0_h} being the V_h'-interpolate of u_0, and calculate $u_h^1, u_h^2, \ldots, u_h^n, \ldots$, approximations of $u_h(t)$ at increasing times values $t_1, t_2, \ldots, t_n, \ldots$ by

$$M_h^n(\tau_n)\frac{u_h^n - u_h^{n-1}}{t_n - t_{n-1}} + A_h^n(\tau_n)[wu_h^n + (1-w)u_h^{n-1}] = 0$$

where $\tau_n = wt_n + (1-w)t_{n-1}$, and $w \in [0,1]$ is the scheme parameter.

As a matter of fact, we take a constant increment of time Δt, so that $t_n = n\Delta t$, and approximate $\Omega(t_n)$, say by Ω_h^n.

Integrations are performed on a weighted domain given by $w\Omega_h^n + (1-w)\Omega_h^{n-1}$. Matrices M_h^n and A_h^n do not really depend on t, for the approximate domain Ω_h^n changes discretely. Thus the argument τ_n above only accounts for this interpolation.

Before describing how we determine Ω_h^n, we should give a short account of the triangulation method that we use.

Let $\rho = s^0(\theta)$ and $\rho = s^*(\theta)$ be the equations in polar coordinates of Γ^0 and Γ^*, respectively.

Now, given $\varepsilon > 0$ (possibly small), we choose integers m and p such that:

$$(13)\qquad \max_{\theta\in[0,2\pi]} \left|\frac{s^*(\theta)}{p} - \frac{s^0(\theta)-s^*(\theta)}{m}\right| \leq \varepsilon$$

and we define

$$M = 8(p+m).$$

Let the boundary of Ω^0 be approximated by polygons Γ_h^0 and Γ_h^* whose vertices are the intersections of Γ^0 and Γ^* with the lines $\theta = \theta_j$, $j = 0,1,\ldots,M-1$, and $\theta = \theta_i^*$, $i = 0,1,\ldots,8p-1$, respectively, where $\theta_j = j\beta$ and $\theta_i^* = i\beta^*$, β and β^* being given by

$$(14)\qquad \beta = \pi/4(p+m)$$

and

$$(14)'\qquad \beta^* = \pi/4p\ .$$

Let Ω_h^0 be the domain bounded by Γ_h^0 and Γ_h^*, and $\rho = s_h^0(\theta)$ be the equation of Γ_h^0 in polar coordinates.

Now the vertices $P_{k\ell}$ of the triangulation of Ω_h^0 are defined as follows:

$$(15)\qquad \begin{cases} P_{k\ell} = (\rho_{k\ell},\ \theta_{k\ell}) \\ \theta_{k\ell} = \dfrac{2\pi\ell}{k} \\ \rho_{k\ell} = (k-p)\dfrac{s_h^0(\theta_{k\ell}) - s^*(\theta_{k\ell})}{m} + s^*(\theta_{k\ell}) \\ k = p,\ p+1,\ \ldots,\ p+m;\quad \ell = 1,\ 2,\ \ldots,\ 8k\ . \end{cases}$$

An illustration of the so obtained triangulation is given in Figure 6 (see Section 4) for $p = 4$ and $m = 2$.[3] For other details see [9].

The meaning of ε is establishing a relation between p

---

[3] This triangulation is slightly different of the one we consider in [9]. The essential difference is due to the exclusion of the triangles lying in Ω^*.

and m so that the length of all the edges of the mesh remain of the same order as we refine the mesh. In the ideal case ϵ should be chosen to be at least $O(m^{-2})$. In this way we can immediately conclude that the spacial step size h is defined as $O(m^{-1})$ or equivalently as $O(p^{-1})$. But in most practical cases ϵ is just $O(h) = O(m^{-1})$ and still the above equivalence could be verified.

Notice that by this process, Γ_h^0 is a polygon of M vertices and that the domain is divided into 8 basic sectors of amplitude $\pi/4$, where the same kind of partition is performed. Also, in each sector we have $p+m$ boundary triangles, i.e., $p+m$ triangles adjacent to Γ_h^0 along an edge; between neighboring pairs of those boundary triangles we have $p+m-1$ interposed triangles, intersecting Γ_h^0 by one vertex.

Let us now introduce the process for determining the position of the free boundary Γ_h^n at the n-th time step, $n = 1, 2, \ldots$, and consequently Ω_h^n, the interior of $\Gamma_h^n \cup \Gamma_h^*$.

First we denote by s_j^0 the polar radii of Γ_h^0 in the directions θ_j, $j = 0, 1, \ldots, M-1$.

Now we assume that during the fixed time increment Δt, Γ_h^{n-1} moves to Γ_h^n in the way shown in Figure 3, which represents a part of the domain.

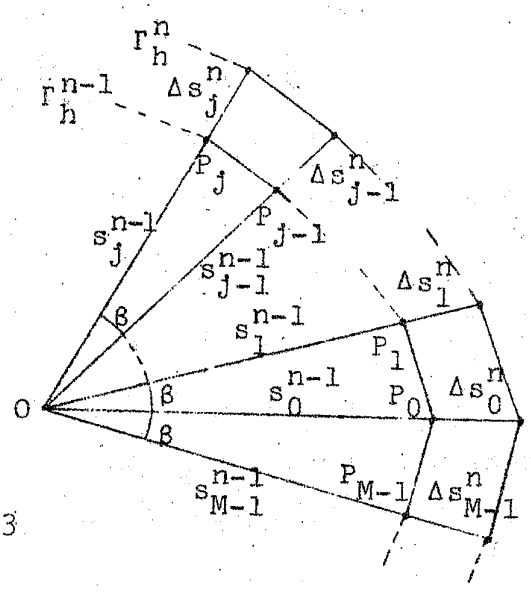

Figure 3

Δs_j^n being the displacement of the boundary in the direction θ_j, $j = 0,1,\ldots,M-1$, the polar radii s_j^n defining the new boundary Γ_h^n are given by

$$s_j^n = s_j^{n-1} + \Delta s_j^n, \quad n = 1,2,\ldots .$$

The increments Δs_j^n are calculated by discretizing the Stefan condition (10) as follows:

$$(16) \qquad -\kappa \frac{\Delta s_j^n}{\Delta t} = \frac{\partial u_h^{n-1}}{\partial \rho}(P_j)\left[1 + \left(\frac{s_{j+1}^{n-1} - s_{j-1}^{n-1}}{2\beta s_j^{n-1}}\right)^2\right]$$

where $s_M^{n-1} = s_0^{n-1}$, $s_{-1}^{n-1} = s_{M-1}^{n-1}$, β is given by (14) and P_j are the vertices of Γ_h^n, as shown in Figure 3. $\frac{\partial u_h^{n-1}}{\partial \rho}(P_j)$ is calculated in two different ways according to the position of P_j:

1. If $j = k(p+m)$, $k = 0,1,\ldots,7$ then it is simply calculated along the edge $\overline{Q_j P_j}$ lying on the line $\theta = \theta_j$ (see Fig. 4a).

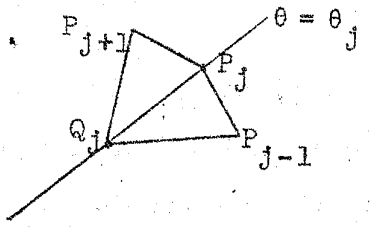

Fig. 4a

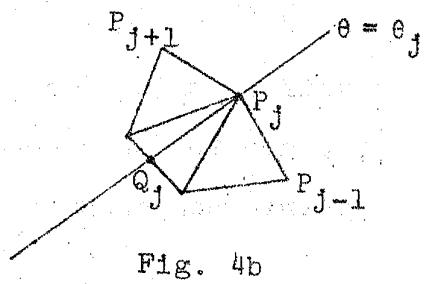

Fig. 4b

Figure 4

2. If $j \neq k(p+m)$, $k = 0,1,\ldots,7$, then $\frac{\partial u_h^{n-1}}{\partial \rho}(P_j)$ is calculated along the segment $\overline{Q_j P_j}$ (see Figure 4b) lying in the interposed triangle between the boundary triangles having P_j as a common vertex.

We note that due to the definition of our triangulation process, the latter operation always makes sense.

Finally, keeping the number of triangles constant, we simply define the new mesh by the automatic triangulation method (15) by replacing s_h^0 by s_h^n, for $n > 0$, $\rho = s_h^n(\theta)$ being the equation of the polygon Γ_h^n.

We should note that in our method each basis function φ_j depends on time, for the corresponding node P_i moves at each time step. Denoting by $N(h)$ the number of nodes and by $u_i(t)$ the coefficient of $u_h(t)$ associated with φ_i, that is, with node P_i, we have

$$\frac{\partial u_h}{\partial t} = \sum_{i=1}^{N(h)} \left(\frac{du_j(t)}{dt} \varphi_j + \frac{\partial \varphi_j}{\partial t} u_j \right).$$

Now recalling (12) we have:

$$A_h(t) = \{ \int_{\Omega(t)} [\nabla \varphi_i \nabla \varphi_j + \varphi_i \frac{\partial \varphi_j}{\partial t}] dx \}$$

and

$$M_h(t) = \{ \int_{\Omega(t)} \varphi_i \varphi_j dx \} .$$

The mass matrix $M_h(t)$ is obviously symmetric but $A_h(t)$ is non-symmetric due to the second term in the integrand. Actually the non-symmetric component of $A_h(t)$ is matrix $V_h(t)$,

$$V_h(t) = \{ \int_{\Omega(t)} \varphi_i \frac{\partial \varphi_j}{\partial t} dx \}$$

that is called by Mori the velocity matrix, since it accounts for the effect of displacement of the nodes with respect to time.

Another point that is worth a remark is the following. Since for this algorithm we do not increase the number of triangles, and since the edges of Γ_h^* have fixed length at every

time step, triangles closer to Γ_h^* will tend to elongate as n increases. Although from the accuracy viewpoint this may not be a desirable situation, as far as the rate of convergence is concerned, this violation of the classical angle condition for unbounded n does not yield any disadvantage. Indeed, as it has been proved by Jamet [4], the essential condition for maintaining the optimal rate of convergence of triangular finite element approximate solutions is not the lower boundedness of the angles, but the fact that no angle approaches $180°$ as the mesh is refined. Such a situation is not occurring here.

4. Numerical Results

We have tested the numerical viability of the algorithm proposed here which generally presented a good performance. Some singificant examples of the obtained results are given below. All the calculations were done in double precision on the HITAC/8800-8700 of the University of Tokyo Computer Center. For the solution of the systems of linear equations we used the Gaussian method for band matrices.

Example 1: For the given initial domain

$$\Omega_0 = \{(x, y)/1 < \rho < 2\}$$

we take as an exact solution the function

$$u = \frac{1}{2}(t + 2 - \frac{\rho^2}{t+2})$$

that satisfies a non-homogeneous equation of the form:

$$\frac{\partial u}{\partial t} - \Delta u = f \ .$$

The equation of the free boundary is $\rho = t + 2$. We preferred to choose (4)' as a boundary condition, so that we can compare

computed values of u at the stationary nodal points of the grid, i.e., those lying on Γ^*.

We calculate up to time $T = 1$ so that the final domain is

$$\Omega(T) = \{(x, y) / 1 < \rho < 3\}.$$

Due to the symmetry of the problem we have performed the calculations only for the sector $0 \leq \theta \leq \pi/4$.

We define $h = 1/m$ and we take $p = m$ (so, in (13) $\varepsilon = 0$). We also take $w = 1$ as the scheme parameter so that we obtain a usual implicit scheme. In this way we choose $\Delta t = h/5$.

Table 1: Computed values of u for $\rho = 1$.

θ	h	t = 0.1	t = 0.5	t = 1.0
0	1/2	0.8194	1.0528	1.3175
$\pi/8$		0.8111	1.0424	1.3068
0	1/4	0.8154	1.0487	1.3206
$\pi/8$		0.8114	1.0442	1.3161
0	1/8	0.8132	1.0483	1.3255
$\pi/8$		0.8117	1.0467	1.3237
0	1/16	0.8123	1.0488	1.3289
$\pi/8$		0.8118	1.0483	1.3283
Exact Value		0.8119	1.0500	1.3333

Table 2: Maximal absolute errors of the computed solution.

h	t = 0.1	t = 0.5	t = 1.0
1/2	0.0147	0.0661	0.1310
1/4	0.0070	0.0319	0.0649
1/8	0.0033	0.0156	0.0322
1/16	0.0016	0.0077	0.0160

Remarks: 1. Those maximal absolute errors occur on the computed free boundary where the computed solution vanishes.

2. From Table 2 one can see that the observed rate of convergence in the maximum-norm is one for each t.

Table 3: Position of the free boundary for $\theta = 0$ and computer time.

h	t = 0.1	t = 0.5	t = 1.0	Comp. Time
1/2	2.0875	2.4423	2.8807	4.019sec.
1/4	2.0940	2.4713	2.9396	5.963sec.
1/8	2.0971	2.4857	2.9696	25.284sec.
1/16	2.0986	2.4929	2.9847	254.324sec.
Exact Value	2.1000	2.5000	3.0000	

Remarks: 1. For other values of θ the computed values are nearly the same which means that the free boundary is stably plotted.

2. One can observe linear convergence for the position of the free boundary.

Example 2: We take an example similar to the preceding one. Only this time Γ^* is reduced to a point, namely the centre of the circular domain:

$$\Omega^0 = \{(x, y) / 0 < \rho < 1\}$$

The exact solution in this case is chosen to be:

$$u = \frac{1}{2}\left(t + 1 - \frac{\rho^2}{t+1}\right)$$

so that the equation of the free boundary is $\rho = t + 1$.

In this case it only makes sense choosing boundary condition (4).

Again we take $h = 1/m$, $w = 1$ and $h = \Delta t/5$. Obviously this time $p = 0$. We perform the calculations only in the sector $0 \leq \theta \leq \pi/4$, up to time $T = 1$, i.e., the final domain is given by

$$\Omega(T) = \{(x, y) / 0 < \rho < 2\}.$$

Table 4: Maximal absolute errors of the computed solution

h	t = 0.1	t = 0.5	t = 1.0
1/2	0.0299	0.1317	0.2516
1/4	0.0141	0.0643	0.1271
1/8	0.0067	0.0315	0.0635
1/16	0.0032	0.0156	0.0316

Remarks: 1. The basic convergence properties shown in the preceding example can be observed here also, although convergence itself seems to become slower.

2. For the position of the free boundary the preceding remark also applies.

Example 3. We take data symmetric with respect to the 4 directions $\theta = i\pi/4$, $i = 0, 1, 2, 3$ and we solve the following problem:

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u \text{ in } \Omega(t) \times [0, 1], \text{ plus Stefan condition on } \Gamma(t) \\ u(0, \vec{x}) = u_0 \text{ in } \Omega^0, \quad u_0 = 2 - \rho - \frac{\cos 4\theta}{10} \\ \Gamma^* \text{ is given by } s^*(\theta) = 1 \quad \forall \theta \\ \Gamma^0 \text{ is given by } s^0(\theta) = 2 - \frac{\cos 4\theta}{10} \\ \frac{\partial u}{\partial \nu}(t, \vec{x}) = 1 \text{ for } \vec{x} \in \Gamma^* \text{ and } t \in [0, 1] \\ u(t, \vec{x}) = 0 \text{ for } \vec{x} \in \Gamma(t) \text{ and } t \in [0, 1]. \end{cases}$$

We define $h = 1/m$ and we take $p = m$ (so that $\varepsilon < 0.1m^{-1}$).

We use again $w = 1$ but this time we take $\Delta t = h/2$.

For any quantity v, we denote by v_h its approximation for a certain value of h, and we define δv_h by:

$$\delta v_h = 10^4 | v_h - v_{2h} |.$$

Table 5: Position of the free boundary for $\theta = 0$.

$t \rightarrow$	0.25		0.50		0.75		1.00	
h	$s_h(0,t)$	δs_h	$s_h(0,t)$	δs_h	$s_h(0,t)$	δs_h	$s_h(0,t)$	δs_h
1/2	2.1500		2.3110		2.4407		2.5533	
1/4	2.1153	347	2.2585	525	2.3750	657	2.4768	765
1/8	2.0960	193	2.2303	282	2.3400	350	2.4366	402
1/16	2.0859	101	2.2155	148	2.3219	181	2.4158	208
1/32	2.0807	52	2.2080	75	2.3127	92	2.4053	105

Remark: Again the observed rate of convergence is one for the position of the free boundary.

Table 6: Computed values of u for $\rho = 1$ and $\theta = 0$.

$t \rightarrow$	0.25		0.50		0.75		1.00	
h	$u_h(1,0,t)$	δu_h	$u_h(1,0,t)$	δu_h	$u_h(1,0,t)$	δu_h	$u_h(1,0,t)$	δu_h
1/2	0.8745		0.8545		0.8522		0.8605	
1/4	0.8650	95	0.8424	121	0.8426	96	0.8545	60
1/8	0.8541	109	0.8279	145	0.8291	135	0.8427	118
1/16	0.8469	72	0.8183	56	0.8201	90	0.8346	81
1/32	0.8428	41	0.8130	53	0.8151	50	0.8301	45

Remarks: 1. It seems that the optimal rate $\Delta t/h$ for the purpose of economy is far less than 1/2. Indeed we observed that for the same value of h the

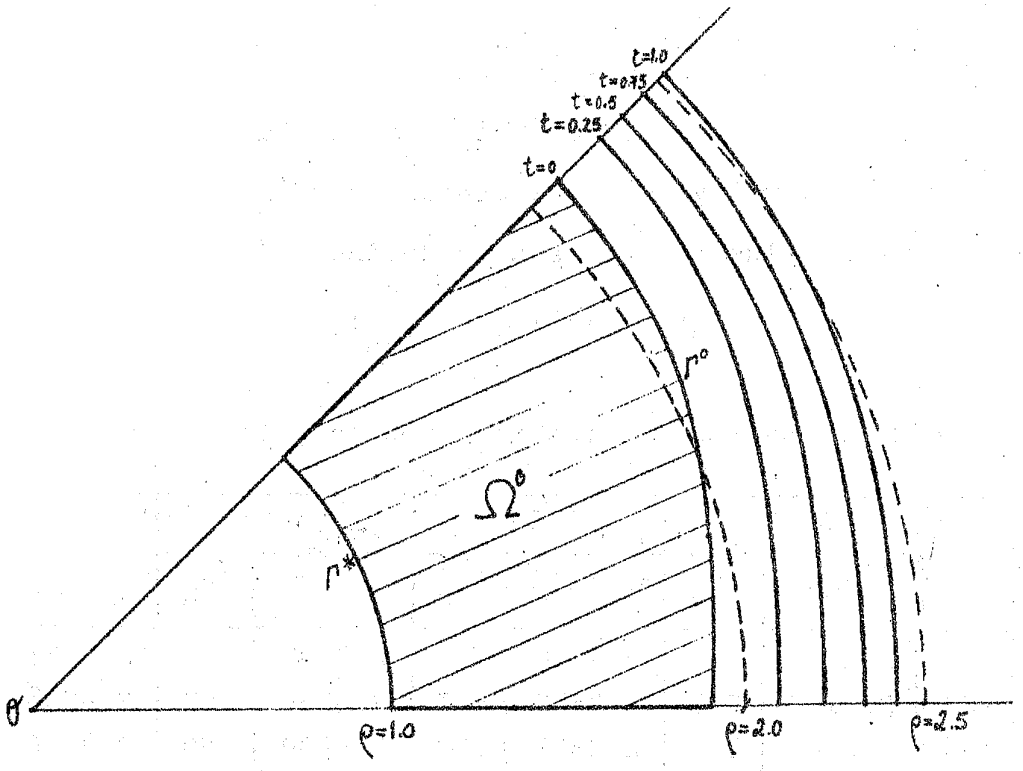

Figure 5

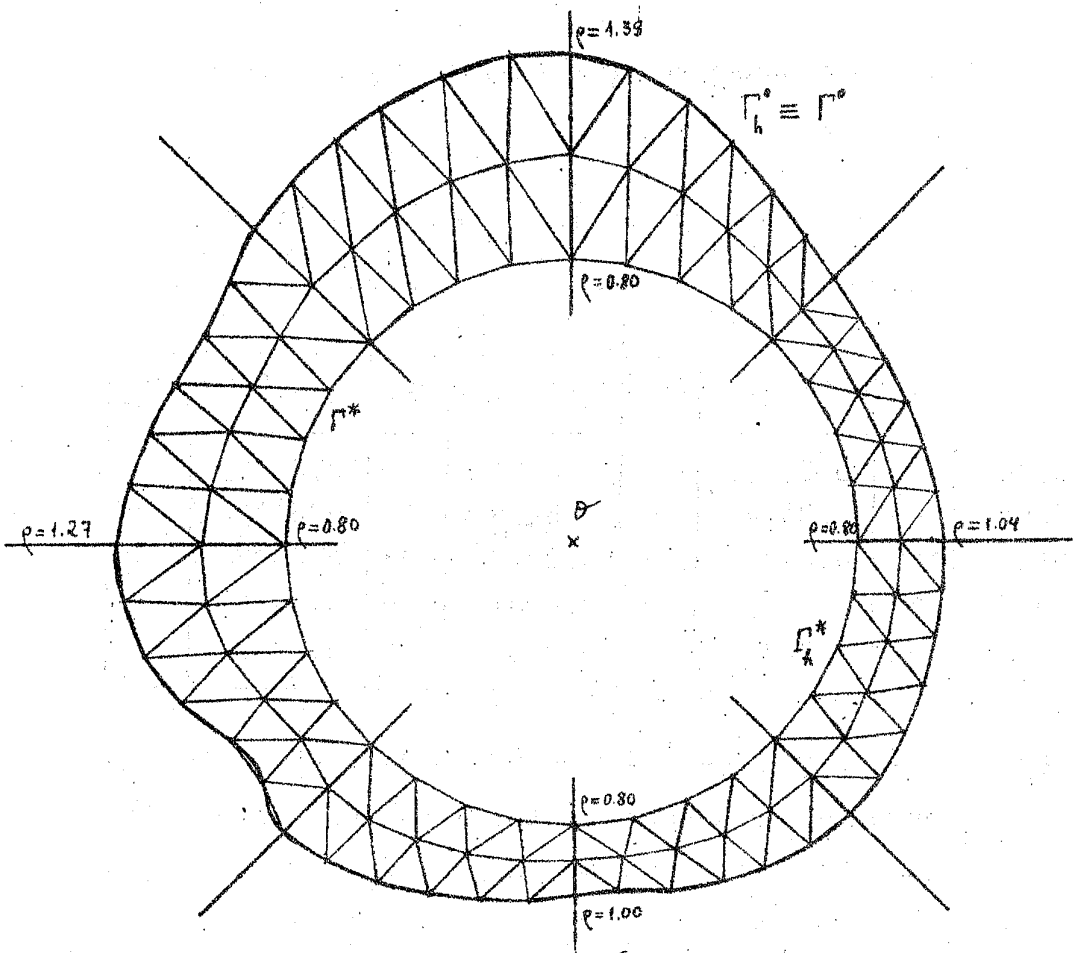

Figure 6

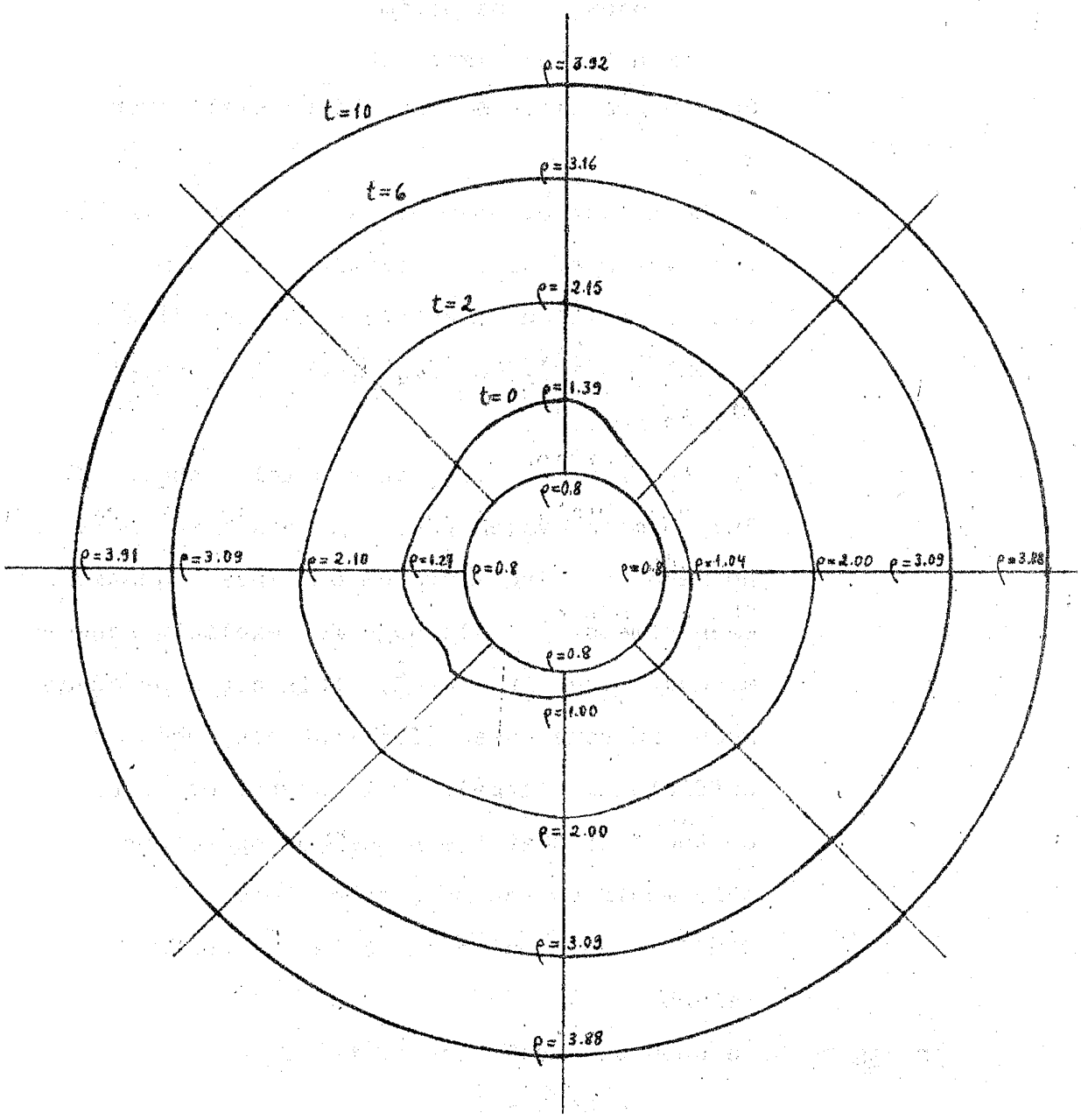
t=10
t=6
t=2
t=0
ρ=3.92
ρ=3.16
ρ=2.15
ρ=1.39
ρ=0.8
ρ=3.91
ρ=3.09
ρ=2.10
ρ=1.29
ρ=0.8
ρ=0.8
ρ=1.04
ρ=2.00
ρ=3.09
ρ=3.88
ρ=0.8
ρ=1.00
ρ=2.00
ρ=3.09
ρ=3.88

Figure 7

approximate values obtained with $\Delta t = h/5$ are much closer to the estimated solution (by extrapolation to the limit) than those in Table 5. So, in practice Δt should be small even for $w = 1$.

2. The number of operations necessary for solving the system of linear equations is $O(m^4)$ whereas the composition of matrices at each time step is only $O(m^2)$. So we suggest also the following procedure:

   Choose $w = 0$, i.e., an explicit scheme, with lumped mass system [6]. In this case there is no need to solve a system of linear equations at each time step. Although for explicit schemes we must take $\Delta t = O(h^2)$ this could be advantageous in some cases (for instance, when m is sufficiently large). This statement is based on the fact that the number of operations with this modified scheme becomes $O(m^4)$ instead of $O(m^5)$, and is also supported by the preceding remark.

In Figure 5 we show the evolution of the free boundary for increasing values of t and $h = 1/32$. The dotted lines show circles with center 0.

Example 4: We have tested our algorithm to a whole domain by solving a non-symmetric problem similar to the one of the preceding example. We have observed basically the same properties, although in this case we could not use too small mesh sizes because the computer time increases sharply. So we prefer

showing the results obtained for the following problem in which the domain evolves up to large values of T:

- Γ^0 is given by a set of 96 points equally spaced in θ and with polar radii ranging from 1.0 to 1.4, as shown in Figure 6. So we consider Γ^0 to be a polygon.
- Γ^* is defined by $\rho = 0.8$.
- u_0 is the function such that $\frac{\partial u_0}{\partial \rho} = 1 \quad \forall \vec{x} \in \Omega^0, \quad u_0|_{\Gamma^0} = 0.$
- $\frac{\partial u}{\partial \nu}(\vec{x}, t) = 1 \quad \forall t \in [0, T]$ and $\vec{x} \in \Gamma^*$.

The aspect of the initial triangulation is shown in Figure 6 for $p = 4$, $m = 2$. We actually calculate with $p = 8$ and $m = 4$ (so that $\epsilon \leq 0.05$). Defining $h = 0.8/p$ we choose $\Delta t = h$, i.e., $\Delta t = 0.1$. We calculate up to time $T = 10.0$, and we show in Figure 7 the evolution of the free boundary as time increases. In this way the computation lasts about 3 minutes.

5. Concluding Remarks

Although we did not consider explicitly other cases, it should be clear that every one-phase Stefan problem in which the domain evolves as a starshaped one can be treated in a similar way to that described in Sections 2 and 3. In particular we mention the case of bounded domains such as the region Ω shown in Figure 8. By introducing appropriate modifications, we can take into account the gradual transformation of edges of the approximate free boundary into edges of the fixed external boundary $\tilde{\Gamma}$, and then solve the problem similarly (see [5]).

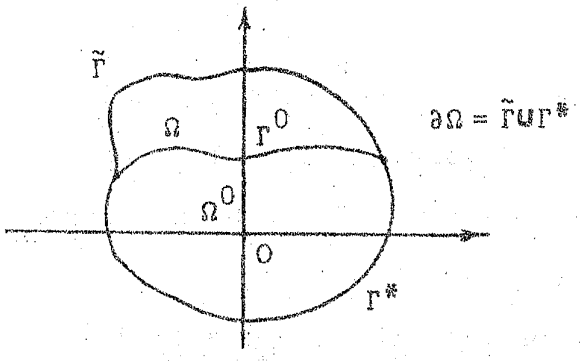

Figure 8

Unfortunately generalization of our algorithm to some cases such as that of the two phase problem, seems not to be so straightforward. However, as we have mentioned other algorithms based on the direct determination of the free boundary could be efficiently used in such cases, provided that good methods for adjusting the spacial mesh step by step are available. Indeed, one of the main features of our triangulation method is generating triangles whose angles remain reasonably bounded away from zero, or, in other words, approximately equal. In so doing the number of nodes of the mesh necessary to attain a given precision can be minimized and computer time saved.

So, generally speaking, we think that our work could be a starting point for research on the application of automatic discretization processes to the direct solution of two or three-dimensional free boundary problems. This is because we also believe that with this work we have helped to contradict the long-prevailing opinion, that this approach is inadequate to the numerical solution of such problems.

References

[1] Ciavaldini, J. F., Analyse Numérique d'un Problême de Stefan à Deux Phases par une Méthode d'Elément Finis, SIAM Journal on Numerical Analysis, Vol.12, 3 (1975), 464-487.

[2] Friedman, A., The Stefan Problem in Several Space Variables, Trans. Amer. Math. Soc., 132 (1968), 51-87.

[3] Fujita, H., Personal Communication, Department of Mathematics, University of Tokyo.

[4] Jamet, P., Estimations d'Erreur pour des Elêments Finis Droits Presque Dégénérés, R.A.I.R.O., Vol.10, 3 (1976), 43-61.

[5] Jamet, P. Bonnerot, N. Numerical Computation of the Free Boundary for the Two-Dimensional Problem by Space-Time Finite Elements, Journal of Computational Physics, Vol.25, 2 (1977), 163-181.

[6] Mori, M., Stability and Convergence of a Finite Element Method for Solving the Stefan Problem, Publ. RIMS, Kyoto Univ., 12 (1976), 539-563.

[7] Mori, M., A Finite Element Method for Solving Moving Boundary Problems, Preprints of IFIP Working Conference on Modelling of Environmental Systems, Tokyo, April 1976, 167-171.

[8] Ruas B. Santos, V., Sur l'Application de Quelques Elêments Finis non Conformes à la Résolution de Problêmes Biharmoniques, Thèse, Université Paris VI, April 1976.

[9] Ruas B. Santos, V., Automatic Generation of Triangular Finite Element Meshes (to appear in Computer and Mathematics with Applications).