Parallel Finite Difference Methods for phase change problems in materials.

CHR. A. SFYRAKIS
Mathematics Department, University of Athens,
Panepistimiopolis, 15784 Zographou, Greece
and
Institute of Applied and Computational Mathematics,FO.R.T.H.,
P.D. Box 1527, 71110 Heraklion, Greece
hammer@math.uoa.gr

Abstract: The great complexity of the problems in phase change materials to us to develop from a fast and methods to solve with parallel programming techniques. We consider the phase field model consisting of the system of p.d.e' s

$$q(\theta)\phi_t = \nabla \cdot (A(\theta)\nabla\phi) + f(\phi, u),$$
$$u_t = \Delta u + [p(\phi)]_t,$$

where $\phi = \phi(x, y, t)$ is the phase indicator function, $\theta = \arctan(\phi_y/\phi_x)$, $u = u(x, y, t)$ is the temperature, $q,\ p,$ and f are given scalar functions, and A is a 2×2 matrix of given functions of θ. This system describes the evolution of phase and temperature in a two-phase medium, and is posed for $t \geq 0$ on a rectangle in the x, y plane with appropriate boundary and initial conditions. We solve the system using two finite difference methods. The first method is based on the explicit Euler scheme for the first equation and the Crank-Nicolson-ADI method for the second. The other method uses for both equations the Crank-Nicolson-ADI scheme. We show results of relevant numerical experiments, compare the errors of the two methods, and compare their speed-up when we implement them using parallel processors. Also make comparisons between the methods as and for each method separately and draw conclusions depending on the number of nodes and the speed of execution method in one, two and four processors.

Key–Words: finite difference methods, simplified phase-field models, Parabolic system, explicit Euler scheme, Crank-Nicolson-ADI method, Error estimates, parallel implementation.

1 Introduction

The traditional method for the numerical solution of evolution equations modeling phase transition phenomena (e.g. solidification) is to discretize the partial differential equations (p.d.e's) that describe the mathematical model (e.g. the heat equation) in the, say, two domains where the material has different phase, and couple this discretization with appropriate interface conditions valid on the free boundary (interface) separating the two phases. This is quite complicated since it requires tracking the unknown interface and interpolating it on a given grid, or using space-time discretizations, for example adaptive grids that follow the interface.
As an alternative, one could use *phase-field models*, [1]. These models replace tracking and approximating the free interface with the introduction of another p.d.e., which is coupled with the energy (heat) equation in the whole domain and, in addition to the original unknown (the temperature), of a new unknown (the *phase*) which is equal to a characteristic value, e.g. 0 in the solid and 1 in the liquid phase, and changes abruptly in a small neighborhood of the interface. The solution of this system of two coupled nonlinear parabolic p.d.e's evolves from suitable initial conditions and its solution should exhibit a sharp moving front in the phase variable that defines the interface of the two phases of the medium, and, when appropriate, describe the development of the complex geometrical patterns (*dendrites*, regions where one phase locally penetrates into the other) that occur in realistic solidification problems. The accurate numerical solution of this coupled system of p.d.e' s requires fine spatial discretizations around the interface and small time steps and is quite time consuming even in two space dimensions. In this note we consider a specific phase field model in two space dimen-

sions, due to McFadden, Wheeler, Sekerka, Wang et al., [2]−[5]. The model consists of a system of two p.d.e's of the form

$$\begin{aligned} q(\theta)\phi_t &= \nabla\cdot(A(\theta)\nabla\phi) + f(\phi,u), \\ u_t &= \Delta u + [p(\phi)]_t, \end{aligned} \tag{1}$$

where $\phi = \phi(x,y,t)$ is the phase indicator function, $\theta = \arctan(\phi_y/\phi_x)$, and $u = u(x,y,t)$ is the temperature, all defined on a rectangle Ω in the x, y plane for $t \geq 0$. The functions q, f, p are given smooth scalar functions of their arguments and A is the anisotropy matrix given by

$$A(\theta) = \begin{pmatrix} r^2(\theta) & -r(\theta)r'(\theta) \\ r(\theta)r'(\theta) & r^2(\theta) \end{pmatrix}.$$

Here we take $r(\theta) = 1 + \delta_\gamma \cos(k\theta)$, where δ_γ is a constant measuring the anisotropy of the surface tension and $k > 1$ is an integer describing the direction of branching. We also have $q(\theta) = (1 + \delta_\gamma \cos(k\theta))/m(1 + \delta_\mu \cos(k\theta))$, where m is a constant and δ_μ a constant measuring the kinetic coefficient anisotropy. If $\delta_\gamma = \delta_\mu = 0$ the model is *isotropic*. If $\delta_\gamma = 0$ and $\delta_\mu \neq 0$ ($A(\theta) = I$), we will call the model *semi-anisotropic*. The system (1) is supplemented by given initial conditions $\phi(x,y,0) = \phi_0(x,y)$, $u(x,y,0) = u_0(x,y)$, $(x,y) \in \Omega$, and boundary conditions of Neumann or Dirichlet type for ϕ and u on the boundary $\partial\Omega$ of Ω for $t \geq 0$.

The system (1) has been solved numerically by Wang, [4], and Wang and Sekerka, [5], in the general, *anisotropic* case (when both δ_γ and δ_μ are nonzero) by an 'explicit-implicit' finite difference scheme that uses the explicit Euler method for advancing the phase-field over a temporal step in the first p.d.e. of (1), and then uses an ADI (Alternating Direction Implicit) scheme for the second p.d.e to solve for the temperature field. These two references contain many interesting numerical computations and measurements of the efficiency of the underlying numerical technique.

In [7]−[9] Rappaz and his collaborators have considered similar systems to (1), for which they have proved existence and uniqueness of weak solutions. They have also constructed and implemented fully discrete, adaptive finite element methods and used them to simulate dendritic growth in the anisotropic case.

In this note, we solve the system using two numerical methods. In Section 3, we consider a finite difference method based on the explicit Euler scheme for the first equation and the Crank-Nicolson-ADI method for the second. In Section 4 we solve the system by another finite difference method that employs for both equations the Crank-Nicolson-ADI method. We show, in Section 5, results of relevant numerical experiments, compare the errors of the two methods, and their speed-up when they are implemented on parallel processors. The results of this paper appeared initially in the author's Ph.D. thesis, [6].

2 The system.

On a square $\Omega = [\alpha,\beta] \times [\alpha,\beta]$ of the x, y plane, we consider the following generalization of (1) in the anisotropic case: For $(x,y,t) \in \Omega \times [0,T]$ we consider the system

$$\begin{aligned} q\phi_t &= \partial_x(a\partial_x\phi) + \partial_y(b\partial_y\phi) \\ &\quad -\partial_x(d\partial_y\phi) + \partial_y(d\partial_x\phi) + f, \\ u_t &= \Delta u + \partial_t p, \end{aligned} \tag{2}$$

where $q = q(\phi, \partial_x\phi, \partial_y\phi)$, $a = a(\phi, \partial_x\phi, \partial_y\phi)$, $b = b(\phi, \partial_x\phi, \partial_y\phi)$, $d = d(\phi, \partial_x\phi, \partial_y\phi)$, $f = f(\phi, u)$, $p = p(\phi)$ are given smooth functions of their indicated arguments. The system (2) is supplemented with initial conditions

$$\begin{aligned} \phi(x,y,0) &= \phi_0(x,y), \quad (x,y) \in \Omega, \\ u(x,y,0) &= u_0(x,y), \quad (x,y) \in \Omega, \end{aligned} \tag{3}$$

and, with homogeneous Dirichlet boundary conditions

$$\begin{aligned} \phi(x,y,t) &= 0, \quad (x,y,t) \in \partial\Omega \times [0,T], \\ u(x,y,t) &= 0, \quad (x,y,t) \in \partial\Omega \times [0,T], \end{aligned} \tag{4}$$

or Neumann boundary conditions

$$\begin{aligned} \frac{\partial\phi}{\partial n}(x,y,t) &= 0, \quad (x,y,t) \in \partial\Omega \times [0,T], \\ \frac{\partial u}{\partial n}(x,y,t) &= 0, \quad (x,y,t) \in \partial\Omega \times [0,T], \end{aligned} \tag{5}$$

where $\frac{\partial}{\partial n}$ denote the derivative in the direction of the outer normal on $\partial\Omega$. We will assume that the initial-boundary-value problems (2)-(4) or (2),(3),(5) have unique solutions, smooth enough for the purposes of the numerical approximation.

3 The explicit Euler-ADI method.

We consider the initial-boundary-value problem (2)-(4) with homogeneous Dirichlet boundary condition for simplicity in the notation. We discretize (2)-(4) as follows. Let $h = (\beta - \alpha)/(J + 1)$,

where J is positive integer, and $x_i = \alpha + ih$, $y_j = \alpha + jh$, $0 \leq i,j \leq J+1$. We define $\Omega_h := \{(x_i,y_j), i,j = 1,\dots,J\}$ and $\partial\Omega_h := \{(x_i,y_j), i = 0 \; or \; i = J+1 \; or \; j = 0 \; or \; j = J+1\}$. Let N be positive integer and $t^n = n\Delta t$, $n = 0,\dots,N$, where $\Delta t = T/N$, and define $t^{n+\frac{1}{2}} = t^n + \Delta t/2$, $x_{i\pm\frac{1}{2}} = x_i \pm h/2$, $y_{j\pm\frac{1}{2}} = y_j \pm h/2$, and
$S_h := \Big\{U = (U_{00},\dots,U_{J+1,J+1})^T \in \mathrm{I\!R}^{(J+2)\times(J+2)} : U_{ij} = 0 \; on \; \partial\Omega_h\Big\}$.

We approximate the solution of (2)-(4) by mesh functions $U^n, \Phi^n \in S_h$, using the explicit Euler-ADI scheme defined as follows:

$$\begin{array}{ll}
\Phi^0_{ij} = \phi_0(x_i,y_j),\; U^0_{ij} = u_0(x_i,y_j),(x_i,y_j) \in \Omega_h \bigcup \partial\Omega_h & \\
\text{For } \; n = 0,1,\dots,N-1: & \\
\text{(i)}\; q^n_{ij}\frac{\Phi^{n+1}_{ij}-\Phi^n_{ij}}{\Delta t} - (L^n_h\Phi)_{ij} = f^n_{ij}, & (x_i,y_j) \in \Omega_h \\
\Phi^{n+1}_{ij} = 0, & (x_i,y_j) \in \partial\Omega_h \\
\text{(ii)}\; \frac{2(U^{n+\frac{1}{2}}_{ij} - U^n_{ij})}{\Delta t} - \delta_{x\bar{x}}U^{n+\frac{1}{2}}_{ij} & \\
-\delta_{y\bar{y}}U^n_{ij} = \frac{p^{n+1}_{ij}-p^n_{ij}}{\Delta t}, & (x_i,y_j) \in \Omega_h \\
\text{(iii)}\frac{2(U^{n+1}_{ij} - U^{n+\frac{1}{2}}_{ij})}{\Delta t} - \delta_{x\bar{x}}U^{n+\frac{1}{2}}_{ij} & \\
-\delta_{y\bar{y}}U^{n+1}_{ij} = \frac{p^{n+1}_{ij}-p^n_{ij}}{\Delta t}, & (x_i,y_j) \in \Omega_h \\
U^{n+\frac{1}{2}}_{ij} = U^{n+1}_{ij} = 0, & (x_i,y_j) \in \partial\Omega_h
\end{array} \qquad (6)$$

Here, $U^{n+\frac{1}{2}} \in S_h$ is the intermediate approximation generated by the ADI scheme. In addition we denote:

$$(L^n_h v)_{i,j} := \begin{cases} \frac{1}{h^2}[\delta_x(a^n_{i-\frac{1}{2},j}\delta_{\bar{x}}v^n_{ij}) + \delta_y(b^n_{i,j-\frac{1}{2}}\delta_{\bar{y}}v^n_{ij}) & \\ \quad -\delta_x(d^n_{i,j}\delta_{\hat{y}}v^n_{ij}) + \delta_y(d^n_{i,j}\delta_{\hat{x}}v^n_{ij})], & \text{if } (x_i,y_j) \in \Omega_h, \\ \qquad 0, & \text{if } (x_i,y_j) \in \partial\Omega_h, \end{cases}$$

and $\delta_x v_{ij} = \frac{v_{i+1,j}-v_{i,j}}{h}$, $\delta_{\bar{x}} v_{ij} = \frac{v_{i,j}-v_{i-1,j}}{h}$, $\delta_y v_{ij} = \frac{v_{i,j+1}-v_{i,j}}{h}$, $\delta_{\bar{y}} v_{ij} = \frac{v_{i,j}-v_{i,j-1}}{h}$, $\delta_{x\bar{x}} v_{ij} = \frac{v_{i+1,j}-2v_{i,j}+v_{i-1,j}}{h^2}$, $\delta_{y\bar{y}} v_{ij} = \frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{h^2}$, $\Delta_h v_{ij} := \delta_{x\bar{x}}v_{ij} + \delta_{y\bar{y}}v_{ij}$.

Also we put

$$a^n_{i+\frac{1}{2},j} := a(\frac{\Phi^n_{i+1,j}+\Phi^n_{i,j}}{2}, \frac{\Phi^n_{i+1,j}-\Phi^n_{i,j}}{h}, \frac{(\Phi^n_{i+1,j+1}+\Phi^n_{i,j+1})-(\Phi^n_{i+1,j-1}+\Phi^n_{i,j-1})}{4h}),$$

$$a^n_{i-\frac{1}{2},j} := a(\frac{\Phi^n_{i,j}+\Phi^n_{i-1,j}}{2}, \frac{\Phi^n_{i,j}-\Phi^n_{i-1,j}}{h}, \frac{(\Phi^n_{i,j+1}+\Phi^n_{i-1,j+1})-(\Phi^n_{i,j-1}+\Phi^n_{i-1,j-1})}{4h}),$$

$$b^n_{i,j+\frac{1}{2}} := b(\frac{\Phi^n_{i,j+1}+\Phi^n_{i,j}}{2}, \frac{(\Phi^n_{i+1,j+1}+\Phi^n_{i+1,j})-(\Phi^n_{i-1,j+1}+\Phi^n_{i-1,j})}{4h}, \frac{\Phi^n_{i,j+1}-\Phi^n_{i,j}}{h}),$$

$$b^n_{i,j-\frac{1}{2}} := b(\frac{\Phi^n_{i,j}+\Phi^n_{i,j-1}}{2}, \frac{(\Phi^n_{i+1,j}+\Phi^n_{i+1,j-1})-(\Phi^n_{i-1,j}+\Phi^n_{i-1,j-1})}{4h}, \frac{\Phi^n_{i,j}-\Phi^n_{i,j-1}}{h}),$$

$$d^n_{i,j} := d(\Phi^n_{i,j}, \frac{\Phi^n_{i+1,j}-\Phi^n_{i-1,j}}{2h}, \frac{\Phi^n_{i,j+1}-\Phi^n_{i,j-1}}{2h}),$$

$$f^n_{ij} := f(\Phi^n_{ij}, U^n_{ij}),\; p^n_{ij} := p(\Phi^n_{ij}),$$

$$q^n_{ij} := q(\Phi^n_{ij}, \frac{\Phi^n_{i+1,j}-\Phi^n_{i-1,j}}{2h}, \frac{\Phi^n_{i,j+1}-\Phi^n_{i,j-1}}{2h}).$$

The implementation of the explicit Euler-ADI finite difference scheme (6) is straightforward: For $n \geq 1$, given Φ^n_{ij}, U^n_{ij}, we compute Φ^{n+1}_{ij} in stage (i). Then, we compute $U^{n+\frac{1}{2}}_{ij}$ in stage (ii) by solving for each j one $J \times J$ tridiagonal linear system, and then U^{n+1}_{ij} in stage (iii) by solving again for each i one $J \times J$ tridiagonal linear system. So, the total number of operations required to advance the solution by one time step is $O(J^2)$. The operations may be readily accelerated using S processors, by solving, in each one of the the stages (ii) and (iii), J/S tridiagonal systems of size $J \times J$ on each processor (assuming that J is a multiple of S).

Using the scheme (6) is very time consuming because of the stability condition $\Delta t = O(h^2)$, which is required by the explicit Euler discretization of the first p.d.e. of (2). Hence, in practice, we modify (6) as follows: We let $\Delta t = O(h)$ denote a (large) time step that is used in the ADI steps (6.ii) and (6.iii) and let $t^n = n\Delta t$, $0 \leq n \leq N$. Given Φ^n_{ij} and U^n_{ij}, we first compute Φ^{n+1}_{ij} using the explicit Euler scheme (6.i) repeatedly with a small time step Δt_{Euler}, where $\Delta t = M\Delta t_{Euler}$, $M = O(N)$, evaluating the coefficient functions q, a, d, and f at $t^\nu = t^n + \nu\Delta t_{Euler}$, for $\nu = 0,1,\dots,M$, Φ^ν_{ij}, and U^n_{ij} (for f). (We found that the accuracy of the scheme was increased when we used the extrapolated values $\frac{3}{2}\Phi^\nu_{ij} - \frac{1}{2}\Phi^{\nu-1}_{ij}$ in the aproximations the ϕ, $\partial_x\phi$, $\partial_y\phi$ variables in the coefficient b.)

Also, we solve the first equation with a different step than the second equation, which enables us to have a faster method than the Wang, [4], which uses the same step for both equations, while as seen in Table 1 the experimental order of convergence remains the same.

4 The ADI-ADI method.

Using the same notation as in Section 3, we now approximate the solution of (2)-(4) by mesh functions $U^n, \Phi^n \in S_h$ defined as follows:

$$
\begin{array}{lll}
& \Phi^0_{ij} = \phi_0(x_i, y_j), & (x_i, y_j) \in \Omega_h \bigcup \partial\Omega_h \\
& U^0_{ij} = u_0(x_i, y_j), & (x_i, y_j) \in \Omega_h \bigcup \partial\Omega_h \\
& \text{For } n = 0, 1, \ldots, N-1: & \\
\text{(i)} & \frac{2(\Phi^{n+\frac{1}{2}}_{ij} - \Phi^n_{ij})}{\Delta t} - \delta_x(a^n_{i-\frac{1}{2},j} \delta_{\bar{x}} \Phi^{n+\frac{1}{2}}_{ij}) & \\
& -\delta_y(b^n_{i,j-\frac{1}{2}} \delta_{\bar{y}} \Phi^n_{ij}) = f^n_{ij} + & \\
& +\delta_x(d^n_{i,j} \delta_{\bar{y}} \Phi^n_{ij}) - \delta_y(d^n_{i,j} \delta_{\bar{x}} \Phi^n_{ij}), & (x_i, y_j) \in \Omega_h \\
\text{(ii)} & \frac{2(\Phi^{n+1}_{ij} - \Phi^{n+\frac{1}{2}}_{ij})}{\Delta t} - \delta_x(a^n_{i-\frac{1}{2},j} \delta_{\bar{x}} \Phi^{n+\frac{1}{2}}_{ij}) & \\
& -\delta_y(b^n_{i,j-\frac{1}{2}} \delta_{\bar{y}} \Phi^{n+1}_{ij}) = f^{n+\frac{1}{2}}_{ij} & \\
& +\delta_x(d^{n+\frac{1}{2}}_{i,j} \delta_{\bar{y}} \Phi^{n+\frac{1}{2}}_{ij}) & \\
& -\delta_y(d^{n+\frac{1}{2}}_{i,j} \delta_{\bar{x}} \Phi^{n+\frac{1}{2}}_{ij}), & (x_i, y_j) \in \Omega_h \\
& \Phi^{n+\frac{1}{2}}_{ij} = \Phi^{n+1}_{ij} = 0 & (x_i, y_j) \in \partial\Omega_h \\
\text{(iii)} & \frac{2(U^{n+\frac{1}{2}}_{ij} - U^n_{ij})}{\Delta t} - \delta_{x\bar{x}} U^{n+\frac{1}{2}}_{ij} & \\
& -\delta_{y\bar{y}} U^n_{ij} = \frac{p^{n+1}_{ij} - p^n_{ij}}{\Delta t}, & (x_i, y_j) \in \Omega_h \\
\text{(iv)} & \frac{2(U^{n+1}_{ij} - U^{n+\frac{1}{2}}_{ij})}{\Delta t} - \delta_{x\bar{x}} U^{n+\frac{1}{2}}_{ij} & \\
& -\delta_{y\bar{y}} U^{n+1}_{ij} = \frac{p^{n+1}_{ij} - p^n_{ij}}{\Delta t}, & (x_i, y_j) \in \Omega_h \\
& U^{n+\frac{1}{2}}_{ij} = U^{n+1}_{ij} = 0, & (x_i, y_j) \in \partial\Omega_h
\end{array}
\quad (7)
$$

and

$$
\begin{array}{l}
a^n_{i+\frac{1}{2},j} := a(\frac{\hat{\Phi}^n_{i+1,j} + \hat{\Phi}^n_{i,j}}{2}, \\
\qquad \frac{\hat{\Phi}^n_{i+1,j} - \hat{\Phi}^n_{i,j}}{h}, \frac{(\hat{\Phi}^n_{i+1,j+1} + \hat{\Phi}^n_{i,j+1}) - (\hat{\Phi}^n_{i+1,j-1} + \hat{\Phi}^n_{i,j-1})}{4h}), \\
a^n_{i-\frac{1}{2},j} := a(\frac{\hat{\Phi}^n_{i,j} + \hat{\Phi}^n_{i-1,j}}{2}, \\
\qquad \frac{\hat{\Phi}^n_{i,j} - \hat{\Phi}^n_{i-1,j}}{h}, \frac{(\hat{\Phi}^n_{i,j+1} + \hat{\Phi}^n_{i-1,j+1}) - (\hat{\Phi}^n_{i,j-1} + \hat{\Phi}^n_{i-1,j-1})}{4h}), \\
b^n_{i,j+\frac{1}{2}} := b(\frac{\hat{\Phi}^n_{i,j+1} + \hat{\Phi}^n_{i,j}}{2}, \\
\qquad \frac{(\hat{\Phi}^n_{i+1,j+1} + \hat{\Phi}^n_{i+1,j}) - (\hat{\Phi}^n_{i-1,j+1} + \hat{\Phi}^n_{i-1,j})}{4h}, \frac{\hat{\Phi}^n_{i,j+1} - \hat{\Phi}^n_{i,j}}{h}), \\
b^n_{i,j-\frac{1}{2}} := b(\frac{\hat{\Phi}^n_{i,j} + \hat{\Phi}^n_{i,j-1}}{2}, \\
\qquad \frac{(\hat{\Phi}^n_{i+1,j} + \hat{\Phi}^n_{i+1,j-1}) - (\hat{\Phi}^n_{i-1,j} + \hat{\Phi}^n_{i-1,j-1})}{4h}, \frac{\hat{\Phi}^n_{i,j} - \hat{\Phi}^n_{i,j-1}}{h}), \\
d^n_{ij} := d(\hat{\Phi}^n_{ij}, \frac{\hat{\Phi}^n_{i+1,j} - \hat{\Phi}^n_{i-1,j}}{2h}, \\
\qquad \frac{\hat{\Phi}^n_{i,j+1} - \hat{\Phi}^n_{i,j-1}}{2h}), f^n_{ij} := f(\hat{\Phi}^n_{ij}, U^n_{ij}), \\
q^n_{ij} := q(\hat{\Phi}^n_{ij}, \frac{\hat{\Phi}^n_{i+1,j} - \hat{\Phi}^n_{i-1,j}}{2h}, \frac{\hat{\Phi}^n_{i,j+1} - \hat{\Phi}^n_{i,j-1}}{2h}),
\end{array}
$$

where $\hat{\Phi}^n_{ij} := \frac{3}{2}\Phi^n_{ij} - \frac{1}{2}\Phi^{n-1}_{ij}$, and for $n \geq 1$ $\hat{\Phi}^0_{ij} := \Phi^0_{ij}$.

The implementation of this ADI-ADI finite difference scheme is straightforward: For $n \geq 1$, given $\Phi^n_{ij}, U^n_{ij}, \Phi^{n-1}_{ij}, U^{n-1}_{ij}$, we compute $\Phi^{n+\frac{1}{2}}_{ij}$ in stage (i) by solving for each j one $J \times J$ tridiagonal linear system, and then Φ^{n+1}_{ij} in stage (ii) by solving again for each i one $J \times J$ tridiagonal linear system. A similar procedure is subsequently followed for the U equations in stages (iii) and (iv). So, the total number of operations required to advance the solution by one time step is $O(J^2)$. The operations may be readily accelerated using M processors, by solving, in each one of the 4 stages, J/S tridiagonal systems of size $J \times J$ on each processor (assuming that J is a multiple of S).

5 Numerical experiments

In the sequel, we will solve the system (2) under Neumann boundary condition on $\partial\Omega$. We first determine the experimental order of convergence of the two finite deference schemes. We took $\Omega = [0,1] \times [0,1]$, $T = 0.1$, $r(\theta) = 1 + \delta_\gamma \cos(4\theta)$, $\theta = \arctan(\phi_x \phi_y)$, (in order to avoid singularities at zeros of ϕ_x) $a = d = r^2(\theta)$, $b = -r(\theta)r'(\theta)$, $q = (1 + \delta_\gamma \cos(4\theta)/(m(1 + \delta_\mu \cos(4\theta)))$ $f = \phi(1-\phi)u/(1+0.25u)$, $p = \phi^3(10 - 15\phi + 6\phi^2)$, and constants $m = 1$, $\delta_\gamma = 0.03$, $\delta_\mu = 0.03$. We added a suitable nonhomogeneous term so that the solution of the associated initial-boundary-value problem with Neumann boundary conditions was $(\phi, u) = (e^{-t}\cos(x(x-1))\cos(y(y-1)), e^{-t}\cos(\pi x)\cos(\pi y))$.

J	$\|\cdot\|_\infty$		$\|\cdot\|_2$	
	errors	order	errors	order
50	$\phi: 1.906e-04$	$---$	$8.627e-05$	$---$
	$u: 7.856e-04$	$---$	$4.082e-04$	$---$
75	$\phi: 8.448e-05$	2.01	$3.650e-05$	2.12
	$u: 3.732e-04$	1.84	$1.915e-04$	1.87
112	$\phi: 3.926e-05$	1.89	$1.702e-05$	1.88
	$u: 1.607e-04$	2.08	$8.181e-05$	2.10
168	$\phi: 1.760e-05$	1.98	$7.408e-06$	2.05
	$u: 7.367e-05$	1.93	$3.727e-05$	1.94
252	$\phi: 8.125e-06$	1.95	$3.407e-06$	1.92
	$u: 3.161e-05$	2.09	$1.593e-05$	2.10
378	$\phi: 3.715e-06$	1.93	$1.510e-06$	2.01
	$u: 1.424e-05$	1.97	$7.161e-06$	1.97
567	$\phi: 1.740e-06$	1.87	$6.871e-07$	1.94
	$u: 6.285e-06$	2.02	$3.153e-06$	2.02

Table 1: Errors and order of convergence of the modified explicit Euler-ADI scheme.

Table 1 shows the errors and the experimental orders of convergence of the numerical solution computed by the Euler-ADI scheme (modified as described in the last paragraph of Section 3) at $T = 0.1$ in the discrete maximum norm $\|\cdot\|_\infty$ and also in the discrete ℓ_2 norm $\|\cdot\|_2$. We computed with $h = 1/(J+1)$, where $J = 50, 75, \ldots, 567$. The large time step Δt was taken equal to $0.01h$ (i.e. $N = 100(J+1)$), while the small time step Δt_{Euler} was computed with a factor of safety as $\Delta t_{Euler} = 10\Delta t/(J+1)$. (Hence

	$\|\cdot\|_\infty$		$\|\cdot\|_h$	
J	errors	order	errors	order
50	$\phi: 1.920e-04$	$--$	$8.664e-05$	$--$
	$u: 1.047e-04$	$--$	$4.762e-05$	$--$
75	$\phi: 8.708e-05$	1.95	$3.851e-05$	1.99
	$u: 4.725e-05$	1.96	$2.119e-05$	1.99
112	$\phi: 3.979e-05$	1.93	$1.731e-05$	1.97
	$u: 2.143e-05$	1.94	$9.513e-06$	1.97
168	$\phi: 1.806e-05$	1.94	$7.747e-06$	1.98
	$u: 9.622e-06$	1.97	$4.231e-06$	1.99
252	$\phi: 8.254e-06$	1.93	$3.486e-06$	1.96
	$u: 4.320e-06$	1.97	$1.883e-06$	1.99
378	$\phi: 3.814e-06$	1.90	$1.582e-06$	1.94
	$u: 1.944e-06$	1.96	$8.383e-07$	1.99
567	$\phi: 1.795e-06$	1.85	$7.300e-07$	1.90
	$u: 8.783e-07$	1.95	$3.736e-07$	1.99

Table 2: Errors and order of convergence of the modified ADI-ADI scheme.

$\Delta t_{Euler} = 0.1h^2$). For two consecutive runs with values $h_1,\ h_2$ of h, giving errors $e_1,\ e_2$, respectively, the rate of convergence was computed as $\log(e_2/e_1)/\log(h_2/h_1)$. It is evident that the orders of convergence are close to 2 for both variables ϕ and u.

In Table 2 we record the analogous errors and rate of convergence for the ADI-ADI scheme for the same problem, using $\Delta t = 0.01h$. Again, the orders of convergence are approximately equal to 2 for both variables.

Both finite difference schemes are quite time consuming. For example, in the above problem for $J = 378$, the Euler-ADI scheme required about 6.27 hrs while the ADI-ADI scheme about 36 mins. For this reason we tried implementing the se algorithms in arrays of parallel processors. The partitioning of the computational effect is done as follows: At every time step for both the explicit Euler and the ADI-ADI steps we create linear systems of the form

$$AX = B$$

where $A,\ X$ and B are $(J+2)\times(J+2)$ matrices. For the explicit Euler scheme A is diagonal, while in the ADI-ADI steps (for solving linear systems either in the x and the y-direction) A is tridiagonal. We partition the columns of B in $(J+2)/S$ groups of columns, where S is the number of processors. Hence, each processor for an Euler step computes a right hand side with $(J+2)/S$ columns and solves for the corresponding columns of X investing the diagonal A. For an ADI-ADI sub-step each processor computers a right-hand side with $(J+2)/S$ columns and solves $(J+2)/S$ tridiagonal systems of size $(J+2)\times(J+2)$ with the same matrix to compute the corresponding columns of X.

We use the parallel computer system Pegasus with 16 dual-processors (32 cpu's in total) at the Computer Center of the Dept. of Chemical Engineering, National Technical University, Athens, Greece (http://febui.chemeng.ntua.gr/pegasus.htm). Each mode consists of two Xeon processors running at 3GHz with 2GB of RAM. The nodes are interconnected with a Myrinet and a Gigabit Ethernet network. The Gigabit Ethernet is used for administering the system (NFS, monitoring, file transfer, etc.), while the Myrinet, which is considerably faster, is devoted to message-passing with MPI. The operating system is Rock Linux 4.1 (http://www.rocksclusters.org). The algorithms have been implemented in C using MPI for parallelization. Job submission is done with the batch-queuing system SGE (Sun Grid Engine).

In order to compare the performance of the two difference schemes on the chosen test problem, the order of magnitude of the errors of both schemes should be the same. For this purpose we ran again the modified explicit Euler-ADI scheme up to $T = 0.1$ using new $\Delta t = 0.005h$ and $\Delta t_{Euler} = 0.1h^2$. With the new steps, the modified explicit Euler-ADI scheme gave the errors of Table 3 which are quite close to those of the ADI-ADI scheme of Table 2. Table 5 shows the computing times (in seconds) required by the two schemes to achieve the error levels shown in Tables 2 and 3 on one, two and four cpu's. The data of Table 5 is shown in graphical form in Figure 1. Note that the processors exchange data at every time step and, since they reside on different computers, they do not share the same memory. We observe that the speed-up due to the increase of the number of processors is being slowed down due to the increase in communication time. For example, as J increases, the ratios of computing time achieved by the explicit Euler-ADI scheme on one cpu derived by the computing time on two cpu's approaches a value of about 1.66 while the same ratio of the time required by two cpu's over the time required by four cpu's approaches 1.16. The same ratios for the ADI-ADI method approach 1.64 and 1.23 respectively. Overall, for the larger value $J = 567$ the ADI-ADI scheme is about 16 times faster than the explicit Euler-ADI scheme, independently of the (same) number of cpu's.

Then we look in more detail some comparative results for the efficiency of parallel algorithms: (a) as altering the number of processors on the same method, and (b), changing the method the same

J	$\|\cdot\|_\infty$		$\|\cdot\|_2$	
	errors	order	errors	order
50	$\phi : 1.920e-04$	$---$	$8.755e-05$	$---$
	$u : 8.670e-05$	$---$	$4.505e-05$	$---$
75	$\phi : 8.692e-05$	1.95	$3.880e-05$	2.01
	$u : 3.904e-05$	1.97	$2.003e-05$	2.00
112	$\phi : 3.964e-05$	1.94	$1.737e-05$	1.98
	$u : 1.766e-05$	1.96	$8.985e-06$	1.98
168	$\phi : 1.794e-05$	1.95	$7.720e-06$	2.00
	$u : 7.894e-06$	1.99	$3.994e-06$	2.00
252	$\phi : 8.161e-06$	1.94	$3.439e-06$	1.99
	$u : 3.522e-06$	1.99	$1.775e-06$	2.00
378	$\phi : 3.747e-06$	1.92	$1.538e-06$	1.98
	$u : 1.570e-06$	1.99	$7.890e-07$	2.00
567	$\phi : 1.749e-06$	1.88	$6.945e-07$	1.96
	$u : 6.988e-07$	2.00	$3.507e-07$	2.00

Table 3: Errors and order of convergence of the modified explicit Euler-ADI scheme.

J	Euler-ADI Cpu 1	Euler-ADI Cpu 2	Euler-ADI Cpu 4	ADI-ADI Cpu 1	ADI-ADI Cpu 2	ADI-ADI Cpu 4
50	9	9	20	5	4	8
75	38	33	62	16	12	17
112	179	139	220	52	37	47
168	874	605	800	177	115	126
252	4384	2800	3181	609	378	369
378	22559	13907	13525	2155	1541	1286
567	111202	66821	57504	6934	4221	3435

Table 4: Computing time (in seconds) to obtain the results of Tables 2 and 3 on 1,2, and 4 cpu’s.

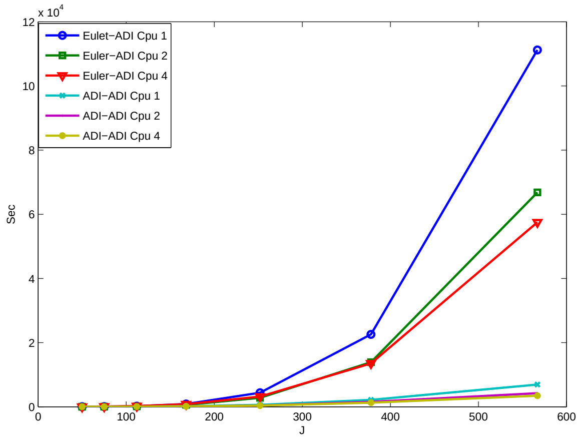


Figure 1: Graph of computing time versus J Data of Table 5.

number of processors.

Tables 5 and 6 shows the ratios of computation time depending on the choice of the number of processors on the method of Euler-ADI and ADI-ADI, respectively. Watching the column 1 and 3 of these tables show that when doubled the number of processors, the time for solving the problem is not halved. Indeed, for small size problems, the tables show that instead of gaining time, losing, due to the big time communications between processors. We note that in both tables the maximum ratio (when doubling the processor) is about 1.65. Table 7 shows the ratio of years of various direct methods of Euler-ADI for ADI-ADI for 1,2, and 4 processors. We note that for sufficiently small h the ADI-ADI method is about 16 times faster than the method Euler-ADI.

J	Cpu 1 / Cpu 2	Cpu 1 / Cpu 4	Cpu 2 / Cpu 4
50	1.00	0.45	0.45
75	1.15	0.61	0.53
112	1.29	0.81	0.63
168	1.44	1.09	0.76
252	1.57	1.38	0.88
378	1.62	1.67	1.03
567	1.66	1.93	1.16

Table 5: Comparison of time to process differences Euler-ADI.

J	Cpu 1 / Cpu 2	Cpu 1 / Cpu 4	Cpu 2 / Cpu 4
50	1.25	0.63	0.50
75	1.33	0.94	0.71
112	1.41	1.11	0.79
168	1.54	1.40	0.91
252	1.61	1.65	1.02
378	1.40	1.68	1.20
567	1.64	2.02	1.23

Table 6: Comparison of time to process differences ADI-ADI.

J	Cpu 1	Cpu 2	Cpu 4
50	1.80	2.25	2.50
75	2.38	2.75	3.65
112	3.44	3.76	4.68
168	4.94	5.26	6.35
252	7.20	7.41	8.62
378	10.47	9.02	10.52
567	16.04	15.83	16.74

Table 7: Reasons for years of litigation methods to Euler-ADI ADI-ADI as a function of J and the number of processors.

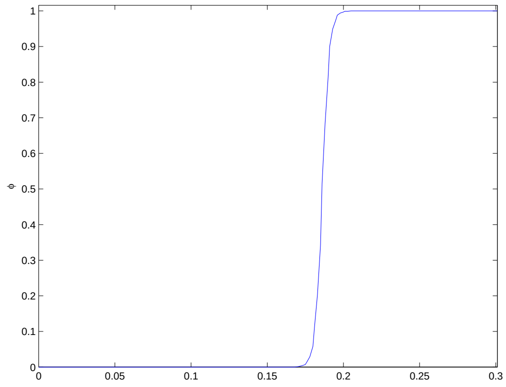
To simulate real anisotropic problems, one of the parameters involved in defining how fine the partition will choose us in the thickness of the interface e which usually ranges from $\frac{1}{200}$ to $\frac{1}{800}$. noticed that for the sake of accuracy the thickness should be at least four intervals. This forces us to choose the J from 600 to 2500. With this scheme we performed two numerical solidification experiments integrationg the system (1) (cf. Wang,[4]) on $\Omega = [0,1] \times [0,1]$, taking

$$\phi_0(x,y) = \frac{1}{2}\left[1 + \tanh\frac{\rho - R_0}{2\sqrt{2}\varepsilon}\right],$$

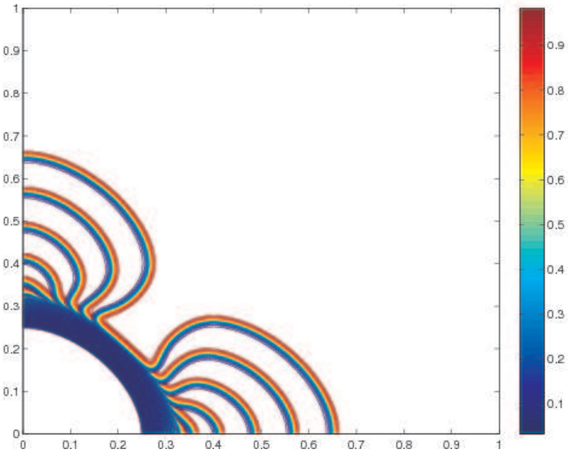
$$u_0(x,y) = \begin{cases} t_s & \rho < R_0 \\ \dfrac{\left[\ln\left(\frac{\rho}{R_0}\right) + t_s \ln\left(\frac{\rho}{R_{00}}\right)\right]}{\ln\left(\frac{R_0}{R_{00}}\right)} & R_0 \leq R_{00} \\ -1 & \rho \geq R_{00} \end{cases},$$

with $\rho = \sqrt{x^2+y^2}$, $q = 1/(m(1+\delta_\mu \cos(4\theta)))$, and $r(\theta) = 1 + \delta_\gamma \cos(4\theta)$, where $\theta = \arctan(\phi_y/\phi_x)$, $f = \frac{1}{\varepsilon^2}(\phi(1-\phi)(\phi - \frac{1}{2} + 30\varepsilon\alpha S \frac{u}{1+0.25u}\phi(1-\phi))$, $p = \phi^3(10 - 15\phi + 6\phi^2)/S$, S is the dimensionless supercooling of the melt, m is the ratio of the capillary to the kinetic length and ε is the ratio of the average interface thickness parameter. In the numerical experiments the values of the various physical parameters were $t_s = 0.01$, $R_0 = 0.1$, $R_{00} = 2R_0$, $S = 0.8$, $m = 0.1$, $\alpha = 70$, and $\varepsilon = 1/400$. All computations were done with the ADI-ADI scheme taking $h = 10^{-3}$, $\Delta t = 2 \cdot 10^{-5}$.

We first consider a semi-anisotropic case with $\delta_\gamma = 0.03$, $\delta_\mu = 0$. Figure 2.a shows the trace of the phase function ϕ on the diagonal $x = y$ of the square Ω at $t = 0.6$. Figure 2.b shows successive positions of the interface of ϕ starting from the almost circular contour at $t = 0.52$. (For $t < 0.52$ the evolution does not deviate much from the isotropic case in which the interface is circular.) The subsequent successive positions were plotted every 400 time steps. The last contour shown is at $t = 0.72$. The formation of a dendrite is quite clear.



(a)



(b)

Figure 2: Semi-anisotropic case with $\delta_\gamma = 0.03$, $\delta_\mu = 0$. (a):Trace of ϕ on $x = y$ at $t = 0.6$. (b): Evolution of the interface of ϕ. Shown are successive contours for $0.52 \leq t \leq 0.72$ every 400 time steps.

Figure 3 and Figure 4 shows a similar evolution in the fully anisotropic case with $\delta_\gamma = 0.03$, $\delta_\mu = 0.03$. There are slight differences, mainly in the speed of the evolving interface.

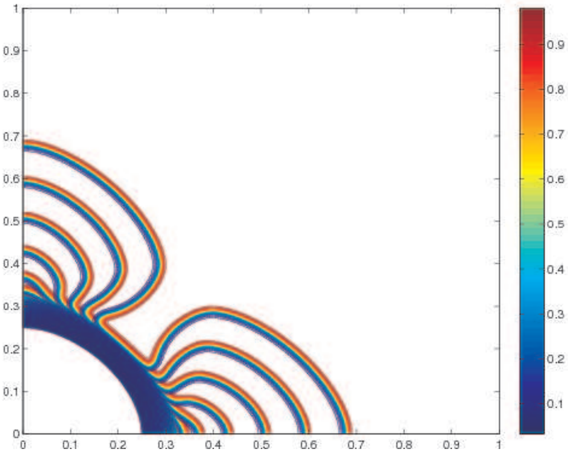
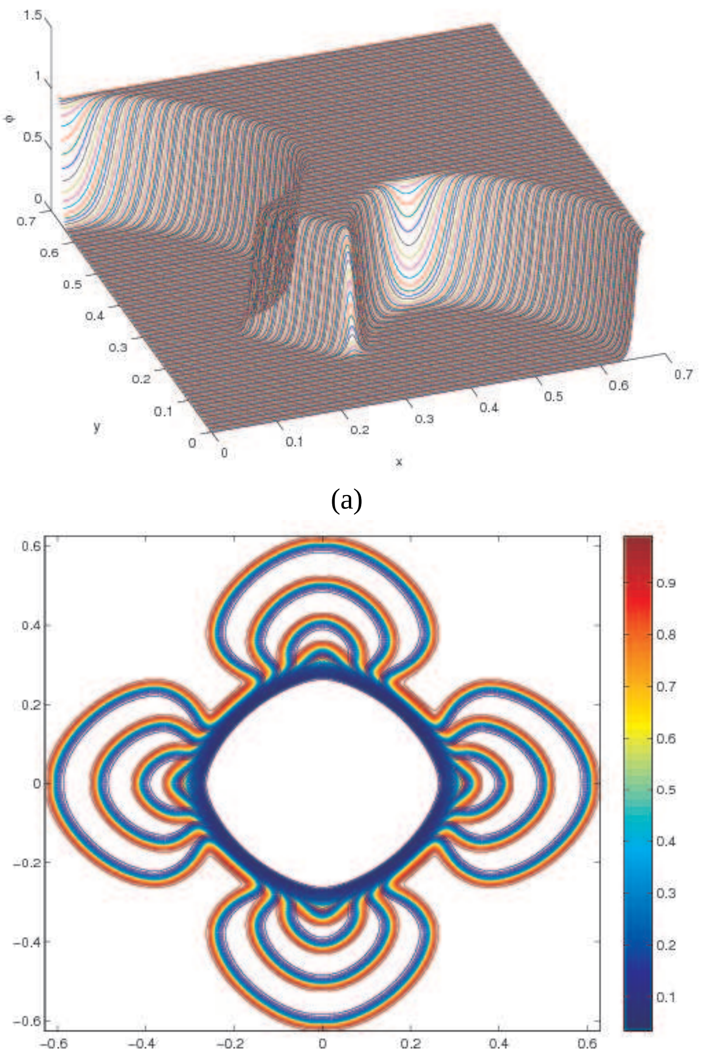


Figure 3: Anisotropic case with $\delta_\gamma = 0.03$, $\delta_\mu = 0.03$ evolution of the interface of ϕ. Shown are successive contours for $0.52 \leq t \leq 0.72$ every 400 time steps.



(a)

(b)

Figure 4: Anisotropic case with $\delta_\gamma = 0.03$, $\delta_\mu = 0.03$ (a):The surface of ϕ on 3D at $t = 0.6$. (b): Evolution of the interface of ϕ. Shown are successive contours for $0.52 \leq t \leq 0.72$ every 400 time steps.

Acknowledgements:
The author would like to thank Prof. V. Dougalis for suggesting the problem and Prof. A. Boudouvis for providing access to the cluster Pegasus, where the numerical experiments of this work were carried out.

References:

[1] J.S. Langer, “Models of pattern formation in first-order phase transitions”, in Directions in Condeased Matter Physics, ed. By G. Grinsteil and G. Mazenko, World Scientific, Singapore 1986, 164-186.

[2] S.L. Wang, R..F. Sekerka, A.A. Wheeler, B.T. Murray, S.R,. Coriell, RJ. Braun and G.B. McFadden, “Thermodynamically-consistent phase-field models for solidification”, Physica D. Vol. 69 (1993); 189-200.

[3] C.B. McFadden, A.A. Wheeler, R.J. Braun, S.R. Coriell and R.F. Sekerka, “Phase-field models for anisotropic interfaces”, Physical Review E, Vol. 48, (1993), 2016-2024.

[4] S. L.Wang, “Computation of dendritic growth at large supercoolings by using the phase field model”, Ph.D. thesis, Department of Physics, Carnegie Mellon University, Pittsburgh, 1995.

[5] S. L. Wang and R. F. Sekerka, “Algorithms for phase field computation of the dendritic operating state at large supercoolings”, Journal of Computational Physics, Vol. 127 (1996), 110-117.

[6] Chr. A. Sfyrakis, Ph.D. “Mathematical and numerical models for materials phase change problems”, Department of Mathematics, University of Athens, 2008 (In Greek).

[7] E. Burman and J. Rappaz, “Existence of solutions to an anisotropic phase field model”, Mathematical Methods in the Applied Sciences, Vol. 26 (2003), 1137-1160.

[8] E. Burman, M. Picasso. and J. Rappaz, “Analysis and computation of dendritic growth in binary alloys using a phase-field model”, Numerical mathematics and advanced applications, Springer, Berlin, 2004, 204-220.

[9] E. Burman, D. Kessler, J.Rappaz, “Convergence of the finite element method applied to an anisotropic phase-field model”, Annales Math. B. Pascal, Vol. 11 (2004), 67-94.

[10] J.E. Dendy, “Alternating direction methods for nonlinear time-depedent problems”, SIAM .I. Numer. Anal., Vol. 14 (1977), 313-326.

[11] Chr. A. Sfyrakis and V. A. Dougalis, “A fast numerical method for a simplified phase field model”, Mathematical Methods in Scattering Theory and Biomedical Engineering, ed. by D. I. Fotiadis and C. V. Massalas, World Scientific 2006, 208-215.

[12] Chr. A. Sfyrakis, “Accelerated Finite Difference Methods for a Simplified Phase Field Model”, Finite Differences - Finite Elements - Finite Volumes - Boundary Elements, by WSEAS Press, 2009

[13] Chr. A. Sfyrakis, “Finite difference methods for an anisotropic phase field model”, Recent Advances in Mathematics and Computers in Biology and Chemistry, by WSEAS Press, 2009, pg. 142-146