Application of Parallel Computing Techniques for Problems of Degenerated Diffusion

Milan Šenkýř, Jiří Mikyška and Michal Beneš

Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Trojanova 13, 120 00 Prague, Czech Republic
Contact e-mails: *kiwi@kmlinux.fjfi.cvut.cz, mikyska@kmlinux.fjfi.cvut.cz, Michal.Benes@fjfi.cvut.cz*

1 Introduction

In this contribution, we discuss parallelization of the problem of curve dynamics in plane. Related PDEs are based on the levelset method introduced in [5], and on the phase-field method described in [1]. Numerical schemes use a finite-difference discretization in space and explicit time solvers. Parallel algorithms are designed for systems with distributed memory, and are based on the domain splitting. The achieved results indicate strength and efficiency of the described approach in case of such highly nonlinear problems.

2 Mean-curvature flow

We study the following motion law for closed planar curves denoted as Γ:

$$v_\Gamma = -g(\theta)\kappa_\Gamma + F, \tag{1}$$

in the direction of the Euclidean normal vector to Γ. Here, $\mathbf{n}_\Gamma$ denotes the normal vector to Γ, v_Γ the normal velocity, κ_Γ the mean curvature, F a forcing term, and g is a suitable positive 2π-periodic function of curve anisotropy, θ is the angle between $\mathbf{n}_\Gamma$ and a prescribed direction. We take $g(\theta) = \psi(\theta) + \psi''(\theta)$, where $\psi(\theta) = 1 + \zeta\cos(N_{fold}\,\theta)$, ζ is the anisotropy strength and N_{fold} a type of symmetry. The equation (1) in the form of the Gibbs-Thompson law is contained in the modified Stefan problem. For details, we refer the reader to [1]. In [4], we may find an application in noise filtering, edge detection and morphing of computer-processed image data.

Hamilton-Jacobi equation. Assume that the curve $\Gamma(t)$ is represented by a levelset of a function $P = P(t, x)$:

$$\Gamma(t) = \{x \in \mathbb{R}^2 \mid P(t, x) = const.\}.$$

We can express the quantities appearing in (1) by means of P:

$$\mathbf{n}_\Gamma = -\frac{\nabla P}{|\nabla P|}, \quad v_\Gamma = \frac{\partial_t P}{|\nabla P|}, \quad \kappa_\Gamma = div(\mathbf{n}_\Gamma).$$

Then, we can introduce the Hamilton-Jacobi equation (see [5, 3])

$$\frac{\partial P}{\partial t} = g(\theta)|\nabla P|\nabla \cdot (\frac{\nabla P}{|\nabla P|}) + |\nabla P|F. \tag{2}$$

Allen-Cahn equation. An extensive experience with non-linear reaction-diffusion equations led to the development of a phase-field approximation of (1) by the Allen-Cahn

equation [2], or by a modified Allen-Cahn equation [1]. The evolution of the levelset $\frac{1}{2}$ of its solution approximates the evolution of the manifold $\Gamma(t)$, as discussed in [1].

First, we denote a rectangular domain $\Omega = (0, L_1) \times (0, L_2) \subset \mathbb{R}^2$, $[x, y] \in \Omega$, the time variable $t \in (0, T)$. The problem for an unknown function $p = p(t, x, y)$ reads as follows

$$\xi \frac{\partial p}{\partial t} = g(\theta) \left(\xi \Delta p + \frac{1}{\xi} f_0(p) \right) + F(u) \xi |\nabla p|, \text{ in } (0, T) \times \Omega,$$
$$p|_{\partial\Omega} = 0 \text{ on } (0, T) \times \partial\Omega, \quad p|_{t=0} = p_{ini}(x) \text{ in } \Omega.$$

Here, $\xi > 0$ is a parameter related to the thickness of the interface layer (it is usually set to a value $<< 1$). The polynomial $f_0(p) = ap(1-p)(p-\frac{1}{2})$ with $a > 0$ is derived from the double-well potential w_0 as $w_0' = -f_0$. The function $F = F(x, y)$ is bounded. The function p_{ini} is an initial condition. We refer the reader to [1], for details concerning the equation and physical background of it.

Numerical schemes. We treat the PDE problems (2) and (3), both closely related to (1), by several numerical schemes implemented by means of parallelization tools for the systems with distributed memory. The problems are solved in a spatial domain $\Omega = (0, L_1) \times (0, L_2)$, which is discretized by a rectangular uniform grid with mesh sizes h_1, h_2 in directions x and y.

We introduce the following notations for a given function u:

$$h_1 = \frac{L_1}{N_1}, \; h_2 = \frac{L_2}{N_2}, \; u_{ij} = u(ih_1, jh_2),$$
$$\omega_{\mathbf{h}} = \{[ih_1, jh_2] \mid i = 1, \ldots, N_1 - 1; \; j = 1, \ldots, N_2 - 1\},$$
$$\bar{\omega}_{\mathbf{h}} = \{[ih_1, jh_2] \mid i = 0, \ldots, N_1; \; j = 0, \ldots, N_2\}, \; \gamma_{\mathbf{h}} = \bar{\omega}_{\mathbf{h}} - \omega_{\mathbf{h}},$$
$$u_{\bar{x},ij} = \frac{u_{ij} - u_{i-1,j}}{h_1}, \quad u_{x,ij} = \frac{u_{i+1,j} - u_{ij}}{h_1}, \quad u_{\mathring{x},ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h_1},$$
$$u_{\bar{y},ij} = \frac{u_{ij} - u_{i,j-1}}{h_2}, \quad u_{y,ij} = \frac{u_{i,j+1} - u_{ij}}{h_2}, \quad u_{\mathring{y},ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2h_2},$$
$$u_{\bar{x}x,ij} = \frac{1}{h_1^2} \left(u_{i+1,j} - 2u_{ij} + u_{i-1,j} \right), \quad u_{\bar{y}y,ij} = \frac{1}{h_2^2} \left(u_{i,j+1} - 2u_{ij} + u_{i,j-1} \right),$$
$$\bar{\nabla}_h u = [u_{\bar{x}}, u_{\bar{y}}], \quad \mathring{\nabla}_h \, u = [u_{\mathring{x}}, u_{\mathring{y}}], \quad \Delta_h u = u_{\bar{x}x} + u_{\bar{y}y}.$$

Direct discretization of the levelset equation. The curvature expressed in terms of second-order derivatives

$$\kappa_\Gamma = -\frac{\partial_{xx}^2 P \, (\partial_y P)^2 - 2 \, \partial_{xy}^2 P \, \partial_x P \, \partial_y P + \partial_{yy}^2 P \, (\partial_x P)^2}{((\partial_x P)^2 + (\partial_y P)^2)^{3/2}},$$

allows us to use central differences to approximate both first- and second-order derivatives. We then propose an explicit scheme in the following form (n is the time-level index, τ is the time step):

$$P_{ij}^{n+1} = P_{ij}^n + \tau \, g(\theta) \frac{P_{\bar{x}x} \, (P_{\mathring{y}})^2 - 2 \, P_{\mathring{x}\mathring{y}} \, P_{\mathring{x}} \, P_{\mathring{y}} + P_{\bar{y}y} \, (P_{\mathring{x}})^2}{(P_{\mathring{x}})^2 + (P_{\mathring{y}})^2} + \tau \sqrt{(P_{\mathring{x}})^2 + (P_{\mathring{y}})^2} F,$$

which is subject of a regularization when $(P_{\mathring{x}})^2 + (P_{\mathring{y}})^2 = 0$. The relationship of τ and h is given by a stability condition.

The equation (1) defines the motion law on $\Gamma(t)$ only. On the other hand, the function P is obtained from the equation (2) valid in Ω. In our work, we extend the forcing term F from (1) to (2) as it is. Other extensions (construction of extension velocities) are discussed, e.g. in [6].

Discretization of the regularized levelset equation. Let $\varepsilon > 0$ be a small regularization parameter. Instead of (2), we solve the following problem:

$$\frac{\partial P}{\partial t} = \sqrt{\varepsilon^2 + |\nabla P|^2}\ \mathrm{div}\left(\frac{\nabla P}{\sqrt{\varepsilon^2 + |\nabla P|^2}}\right) + F\sqrt{\varepsilon^2 + |\nabla P|^2},\ \text{in } (0,T)\times\Omega,$$

$$\left.\frac{\partial P}{\partial n}\right|_{\partial\Omega} = 0 \text{ on } (0,T)\times\partial\Omega, \quad P|_{t=0} = P_{ini}(x) \text{ in } \Omega.$$

It can be approximated using the following explicit nine-point-stencil finite-difference scheme:

$$P_{i,j}^{k+1} = P_{i,j}^{k} + \tau Q(\mathring{\nabla}_h\ P_{i,j}^{k})\left(F_{i,j}^{k} + \left(\frac{P_{\bar{x},i+1,j}^{k}}{h_1 Q(\nabla P_{i+\frac{1}{2},j}^{k})} - \frac{P_{\bar{x},ij}^{k}}{h_1 Q(\nabla P_{i-\frac{1}{2},j}^{k})} + \frac{P_{\bar{y},i,j+1}^{k}}{h_2 Q(\nabla P_{i,j+\frac{1}{2}}^{k})} - \frac{P_{\bar{y},ij}^{k}}{h_2 Q(\nabla P_{i,j-\frac{1}{2}}^{k})}\right)\right),$$

where $Q(u,v) = \sqrt{\varepsilon^2 + u^2 + v^2}$, $\nabla P_{i+\frac{1}{2},j}^{k} = [P_{x,i,j}^{k}, \frac{1}{2}(P_{\mathring{y},i+1,j}^{k} + P_{\mathring{y},i,j}^{k})]$, $\nabla P_{i-\frac{1}{2},j}^{k} = [P_{\bar{x},i,j}^{k}, \frac{1}{2}(P_{\mathring{y},i,j}^{k} + P_{\mathring{y},i-1,j}^{k})]$, $\nabla P_{i,j+\frac{1}{2}}^{k}$ and $\nabla P_{i,j-\frac{1}{2}}^{k}$ evaluated analogically. The scheme is only conditionally stable.

Discretization for the Allen-Cahn equation is derived by spatial finite differences. Nodal values then remain functions of time, for which we obtain a system of ODEs (the semi-discrete scheme) in the following form:

$$\xi^2\frac{\mathrm{d}p^h}{\mathrm{d}t} = \xi^2 g(\theta)\left(\Delta_h p^h + f_0(p^h)\right) + \xi^2|\nabla_h p^h|F \text{ on } \omega_{\mathbf{h}},$$
$$p^h\,|_{\gamma_{\mathbf{h}}} = 0, \quad p^h(0) = p_{ini}.$$

The equations are numerically solved by the Runge-Kutta-Mersn 4-th order method with adaptive time step. The scheme has been analyzed in [1] from the convergence viewpoint.

3 Parallelization techniques

The above described algorithms are parallelized by means of the Message Passing Library MPI both using Fortran 77/90 and C programming languages. Computations using MPI, version 1.1 were performed on the supercomputing systems IBM SP3 and Cray T3E at CINECA[1], IBM SP, IBM SP2 at the Czech Technical University in Prague, and computations using LAM MPI library[2] were performed on a local network of Linux PC workstations at the Czech Technical University in Prague. In both approaches described below, the computational task is performed by one or more processes, each of them running either on a separate processor (a hardware unit, or virtual unit in the emulated mode).

[1] Supercomputing Center of Italian Universities, Bologna

[2] LAM MPI, Local Area Multicomputer is an open source implementation of MPI standard, `http://www.lam-mpi.org`

Cartesian domain splitting is an approach where a rectangular domain Ω is decomposed into rectangular subdomains, each of them treated by one process. Boundaries of subdomains overlap by one grid line, on which they exchange data. The amount of communication between processes depends on the blocking strategy. We tested the *row-wise blocking strategy*, where the domain is decomposed row-wise. Each block interacts with neighbouring blocks during a timestep. The other tested strategy was the *chequerboard blocking*. In this case, each block communicates with maximum eight neighbours during a timestep.

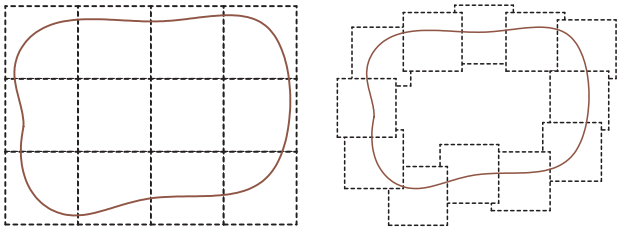

Fig. 1. Cartesian domain splitting (left), and narrow-band splitting (right).

Narrow-band technique introduced in [6] explores the fact that we are interested only in the evolution of the curve $\Gamma(t)$. It is therefore enough to follow the evolution of $P = P(t, x)$ in the vicinity of the levelset $\Gamma(t)$. The presented approach provides a significant speedup. On the other hand, it is less accurate and more difficult to implement, because it requires a reconstruction of the narrow band when Γ approaches its edge (the operation is called *reinitialization*). In our implementation, we cover the curve by overlapping squares of a constant width which are assigned to processes in an intuitive way (Fig. 1). For example, in case of 64 covering squares and 16 processes, the first four squares are computed by the first process, the second four squares by the second process etc. Consequently, the narrow band created by such squares is not of constant width. The processes exchange data for all nodes, where the squares overlap. The approach is easy to implement including processing of the grid by parts small enough to fit them in the fast cache memory of processors. For the purpose of algorithm evaluation, we define the following quantities:

$$\text{Speedup} = \frac{\text{run time in a single process}}{\text{run time in } n \text{ processes}}, \quad \text{Eff.} = \frac{\text{Speedup}}{\text{number of processes}}.$$

Speedup and efficiency of parallelization for the direct algorithm of the levelset equation - Study 1 (IBM SP). In this study, we consider a circle of the initial radius $R_0 = 1.35$ placed in a domain $(0, 4) \times (0, 4)$, which shrinks according to the law $v_\Gamma = -\kappa_\Gamma$ (see Figure 2a). The domain is discretized with the mesh size 0.02 in both directions, the time step is $\tau = 4 \cdot 10^{-5}$. Number of time steps is 22500, the computation stops right before the shrinking time $T = 0.9$ (see [1],). The code is parallelized by means of the domain splitting. The results achieved on the IBM SP system are shown in Table 1.

Study 2 (IBM SP). The above given problem (see Figure 2a) was recomputed using several choices of the mesh size and the time step. As it can be seen from Table 2, efficiency of parallelization depends on the size of data exchanged between processes (e.g., it is faster to send 200kB of data than twice 100kB, due to an initiation).

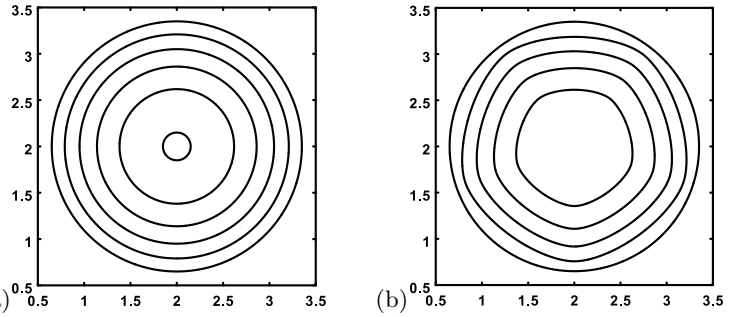

Fig. 2. (a) A circle in $(0,4) \times (0,4)$ shrinking from the initial radius $R_0 = 1.35$ to the radius $R_T = 0.15$ according to the isotropy law $v_\Gamma = -\kappa_\Gamma$. (b) An initial circle of the radius $R_0 = 3.0$ deforming itself according to the 5-folded anisotropy law described by Eq.(3), where $N_{fold} = 5$ and $\zeta = 0.025$.

Number of processes	Mesh nodes per process	CPU time per process	Mesh nodes total	Communication mesh nodes	Eff.
1	40000	908	40000	0	-
4	10000	258	40000	400 (1.0%)	88%
8	5000	149	40000	800 (2.0%)	76%
12	3333	113	40000	1000 (2.5%)	67%
16	2500	94	40000	1200 (3.0%)	60%

Table 1. The results of parallelization efficiency on IBM SP.

Mesh size	200 x 200	267 x 267	400 x 400	667 x 667
Time step	$4.0 \cdot 10^{-5}$	$2.3 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	$3.6 \cdot 10^{-6}$
Iterations	22500	40000	90000	250000

Mesh size \ Processes	1	4	8	12	16
200 x 200	908	258 (88%)	149 (76%)	113 (67%)	94 (67%)
267 x 267	2585	697 (93%)	392 (83%)	277 (78%)	231 (70%)
400 x 400	14171	3657 (97%)	1915 (93%)	1343 (88%)	1058 (84%)
667 x 667	98904	25574 (97%)	12889 (96%)	8740 (94%)	6775 (91%)

Table 2. Efficiency depends on the mesh size. Computation performed on the IBM SP system (CPU time per process and efficiency).

Study 3 (IBM SP and Linux network). In this case, the initial condition (a circle with the initial radius $R_0 = 1.35$) evolves according to (1) with $F = 0$, $N_{fold} = 5$ and $\zeta = 0.025$ as indicated in Figure 2b). With numerical parameters $h_1 = h_2 = 0.01$, $\tau = 1.1 \cdot 10^{-5}$ and 64286 time levels, it terminates at $t = 0.72$. The curve is covered by squares 35 points wide. Due to the curve shrinking, number of active nodes in the narrow band decreases from $\sim$ 33000 to 16000 as shown in Table 3. Compared to the domain splitting, efficiency is lower. This is caused by the fact, that the overlapping areas between processes are larger, and even the number of active nodes increases with the number of processes. On the other hand, the computation is faster, as only a part of the grid is active, and the absolute amount of exchanged data is smaller.

Number of processes	Min. no. of active nodes	Avg. no. of active nodes	Max. no. of active nodes	Communication mesh nodes (% of Avg)	CPU time per process (IBM SP)	CPU time per process (linux cluster)
1	16226	26390	33268	0	15219	2608
4	16366	26407	33252	560 (2.1206%)	5009 (76%)	884 (74%)
8	16383	26497	33368	1120 (4.2269%)	3260 (58%)	576 (57%)
12	16435	26517	33409	1680 (6.3356%)	2613 (49%)	456 (48%)
16	16581	26620	33536	2240 (8.4147%)	2365 (40%)	-

Table 3. Narrow-band approach on IBM SP and Linux network applied to an anisotropic circle shrinking.

Speedup and efficiency of parallelization for the regularized levelset equation - Study 4 (IBM SP3). We considered the test problem shown in Figure 3. For a given number of processes, it is possible to split the domain either into $NPROC$ rows, or into $NPROCX$ columns and $NPROCY$ rows in the checquerboard blocking ($NPROCX \times NPROCY = NPROC$). Unless $NPROC$ is a prime, there are several possibilities for selecting $NPROCX$ and $NPROCY$. Tables 4 and 5 present the runtimes, numbers of communication nodes, and efficiencies for both of the above mentioned blocking strategies attained on an IBM SP3 machine. It is clear from Table 5 that the checquerboard blocking is superior to the row-wise blocking for higher numbers of processes, which is due to a lower number of communicating nodes.

Number of processes	Mesh nodes per process	CPU time per process (s)	Mesh nodes total	Communication mesh nodes	Eff.
1	360000	2656	360000	0	-
4	90000	778	360000	1800(0.5000%)	85%
9	40000	307	360000	4800(1.3333%)	96%
16	22500	180	360000	9000(2.5000%)	92%
25	14400	145	360000	14400(4.0000%)	73%
36	10000	125	360000	21000(5.8333%)	57%

Table 4. Results of parallelization for the row-wise blocking.

Number of processes	Mesh nodes per process	CPU time per process (s)	Mesh nodes total	Communication mesh nodes	Eff.
1	360000	2656	360000	0	-
4	90000	684	360000	1201(0.3336%)	97%
9	40000	320	360000	2404(0.6677%)	92%
16	22500	171	360000	3609(1.0025%)	94%
25	14400	122	360000	4816(1.3377%)	87%
36	10000	85	360000	6025(1.6736%)	87%

Table 5. Results of parallelization for the chequerboard blocking.

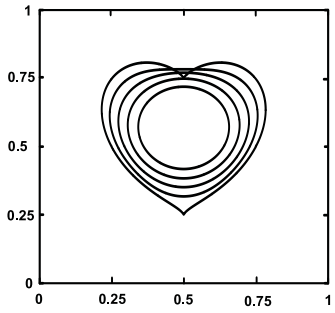

Fig. 3. Evolution of the cardioida curve is driven by the regularized levelset equation, solved in the unit square with $\varepsilon = 10^{-8}$, grid 600×600, $\tau = 10^{-6}$ and 20000 time levels.

Speedup and efficiency of parallelization for the Allen-Cahn equation - Study 5 (IBM SP3). In this computation, we studied the isotropic curve evolution starting at a four-folded pattern in a spatial domain $(0,2) \times (0,2)$. The curve approaches the circle of radius $R = 0.6$ according to the law $v_\Gamma = -\kappa_\Gamma + F(x)$ where the forcing F is a suitable radially symmetric and linear function. Other parameters are $\xi = 0.01$, $h_1 = h_2 = 0.00995$. The curve evolution is in Figure 4(a). The domain was divided into 1, 4, and 16 rectangular subdomains, and the computation was repeated with corresponding number of processes. The mesh size and the total number of mesh points remained the same, the number of mesh points per process decreased, the number of communication mesh points increased, both with increasing number of processes. Measurement results are in Table 6.

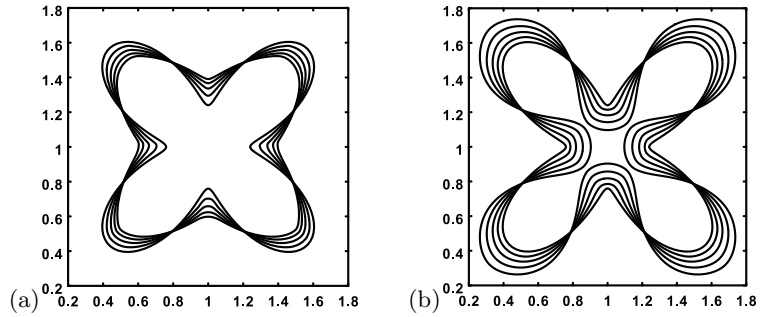

Fig. 4. (a) 4-folded initial curves in $(0,2) \times (0,2)$ approaches circle of radius $R = 0.6$ according to $v_\Gamma = -\kappa_\Gamma + F(x)$ for radially symmetric linear F; $\xi = 0.01$, $h_1 = h_2 = 0.00995$. (b) 4-folded initial curve in $(0,2) \times (0,2)$ shrinks inside and expands outside of the circle of radius $R = 0.6$ according to $v_\Gamma = -g(\theta)\kappa_\Gamma + F(x)$ for radially symmetric linear F, $g(\theta) = 1.0 - 0.8cos(4\theta - \pi/4)$; $\xi = 0.02$, $h_1 = h_2 = 0.00995$.

Study 6 (CRAY T3E). The computation, performed on CRAY T3E, studies the anisotropic curve evolution starting at a four-folded leaf-like curve placed in a spatial domain $(0, 0.4) \times (0, 0.4)$. The curve shrinks inside a circle of radius $R_0 = 0.1$, and expands outside of it thanks to a spatially dependent choice of F in the law $v_\Gamma = -g(\theta)\kappa_\Gamma + F$ $g(\theta) = 1.0 - 0.8cos(4\theta - \pi/4)$; $\xi = 0.02$, $h_1 = h_2 = 0.00995$. The curve evolution is in Figure 4(b). The domain was divided into 1, 4, 16, 25 and 64 rectangular subdomains,

Number of processes	Mesh elements per process	CPU time per process	Mesh elements total	Communication mesh elements	Eff.
1	40401	118.11	40401	0	-
4	10201	29.89	40401	401(0.9925%)	99%
16	2601	10.01	40401	1197(2.9628%)	74%

Table 6. Table of parameters for the use of IBM SP3 - Study 5.

and the computation was repeated with corresponding number of processes. Mesh size and total number of mesh points remained the same, number of mesh points per processes decreased, number of communication mesh points increased, both with increasing number of processes. Measurement results are in Table 7.

Number of processes	Mesh elements per process	CPU time per process	Mesh elements total	Communication mesh elements	Eff.
1	40401	37.55	40401	0	-
4	10201	9.65	40401	401(0.99%)	97%
16	2601	3.23	40401	1197(2.96%)	73%
25	1681	2.48	40401	1592(3.94%)	61%
64	625	2.20	40401	2765(6.84%)	27%

Table 7. Table of parameters for the use of CRAY T3E - Study 6.

Acknowledgment. The first author was partly supported by the project No. 159 of the Czech-Slovak Science Programme, the second author was partly supported by the project CTU 0309914 of the Czech Technical University in Prague, and the third author was partly supported by the project No. 201/01/0676 of the Grant Agency of Czech Republic. The authors gratefully acknowledge technical support of the Center for High Performance Computing at the Czech Technical University in Prague (Project No. 145), and technical support of the CINECA - High Performance Computing Centre, Bologna (Programme Minos).

References

1. M. Beneš, *Mathematical and computational aspects of solidification of pure substances*, Acta Mathematica Universitatis Comenianae **70, No. 1** (2001), 123–152.
2. C.M. Elliott, M. Paolini, and R. Schätzle, *Interface estimates for the fully anisotropic Allen-Cahn equation and anisotropic mean curvature flow*, Math. Models Methods Appl. Sci. **6** (1996), 1103–1118.
3. L.C. Evans and J. Spruck, *Motion of level sets by mean curvature I*, J. Diff. Geom. **33** (1991), 635–681.
4. V. Minárik and M. Beneš, *Numerical solution of degenerate parabolic equations of Hamilton-Jacobi type within the context of computer image processing*, ALGORITMY 2002, Proceedings of contributed papers and posters (Bratislava), 2002, pp. 162–170.
5. J.A. Sethian, *Level set methods*, Cambridge University Press, New York, 1996.
6. ———, *Evolution, implementation, and application of level set and fast marching methods for advancing fronts*, Journal of Computational Physics **169** (2001), 503–555.