Proceedings of the Czech–Japanese Seminar in Applied Mathematics 2006
Czech Technical University in Prague, September 14-17, 2006
pp. 193–205

VECTOR FIELD VISUALIZATION BY MEANS OF ANISOTROPIC DIFFUSION

PAVEL STRACHOTA[1]

Abstract. We propose a method of vector field visualization based on noisy texture smearing. The smearing process is carried out by solving the Allen-Cahn equation with advection. We state the theorem of existence and uniqueness of the weak solution and derive the appropriate a priori estimate. The numerical algorithm for PDE solution is introduced together with an idea of its parallelization. Finally, some results are presented.

Key words. mathematical visualization, vector field, anisotropic diffusion, partial differential equations, weak solution, method of lines, parallelization.

AMS subject classifications. 35K60, 35K65, 65N06, 68U10

1. Introduction. Suppose we have a static vector field $\boldsymbol{v}$ defined in a rectangular domain $\Omega = (0, L^1) \times (0, L^2)$. Our goal is to make its stream lines emerge as smudges by means of smearing a noisy texture on Ω in the direction of the field. In addition to smearing, it is sometimes reasonable to make the texture move (*advect*) in the direction of the vector field.

For the above stated purposes, it is possible to use a diffusion parabolic PDE with the incorporated diffusion anisotropy and with the advection term. (see [18, 19, 4, 7]).

2. Formulation. Let $p : \mathcal{J} \times \Omega \mapsto \mathbb{R}$, $p = p(t, \boldsymbol{x})$ be the function of texture intensity at each point $\boldsymbol{x} \in \Omega$ and at time $t \in \mathcal{J}$ where $\mathcal{J} = [0, T]$ is a time interval. The initial boundary-value problem for the Allen-Cahn equation with advection (see [19]) reads as:

$$\xi^2 \frac{\partial p}{\partial t} + \xi^2 \boldsymbol{v} \cdot \nabla p = \xi^2 \Delta p + f_0(p) + c_0 F \xi, \tag{2.1}$$

$$\left.\frac{\partial p}{\partial n}\right|_{\partial\Omega} = 0, \tag{2.2}$$

$$p(0, \boldsymbol{x}) = I(\boldsymbol{x}), \tag{2.3}$$

where

$$f_0(p) = p(1-p)\left(p - \frac{1}{2}\right).$$

In the above problem, the term Δp is responsible for isotropic diffusion and the term $\boldsymbol{v} \cdot \nabla p$ causes texture advection. (see [4, 6, 8, 5]). The polynomial f_0 makes *nucleation* occur during the time. In this context, nucleation is a formation of areas where the value of p is near 0 or 1. As descibed for example in [19, 6, 7], the parameter ξ is proportional to the diffuse interface layer between such areas. We usually choose ξ such that it is small in comparison with the dimensions of Ω. The influence of

[1]Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague.

the parameter F can be explained in the context of the related problem for mean curvature flow

$$v_\Gamma = -\kappa_\Gamma + F. \tag{2.4}$$

For details, we refer the reader to [7, 4, 6, 19].

In the context of visualization, if $I : \Omega \mapsto \mathbb{R}$ represents the intensity of a noisy texture at each point, the solution p will reflect the gradual diffusion of the initial image I with increasing time. Both the state of p at some final time T and the entire solution evolution can be regarded as the result.

Introducing anisotropy. Our objective is to focus the diffusion mainly in the direction of the vector field. According to (2.1) with the term Δp written as $\nabla \cdot (\nabla p)$, the diffusion speed (i.e. the value of the derivative $\frac{\partial p}{\partial t}$) at a given point is proportional to the norm of the gradient of p. Therefore, we replace ∇p in $\nabla \cdot (\nabla p)$ by an expression which will have a large absolute value only when the gradient direction and the vector field direction (nearly) coincide.

Consider a norm Φ^0 defined on $\mathbb{R}^2$, which is called the *dual Finsler metric* in the context of the Finsler geometry (see [3, 4, 6]). We define its derivative with respect to the vector $\boldsymbol{\eta} = \begin{pmatrix} \eta^1 \\ \eta^2 \end{pmatrix} \in \mathbb{R}^2$ as the vector

$$\Phi^0_\eta(\boldsymbol{\eta}) = \begin{pmatrix} \partial_{\eta^1} \Phi^0(\boldsymbol{\eta}) \\ \partial_{\eta^2} \Phi^0(\boldsymbol{\eta}) \end{pmatrix}.$$

Next, we replace the gradient of p by the term $T^0(\nabla p)$, which represents a so-called Φ-gradient of p ($\nabla_\Phi p$) and is defined as

$$T^0(\nabla p) = \Phi^0(\nabla p)\Phi^0_\eta(\nabla p),$$

where $T^0 : \mathbb{R}^2 \mapsto \mathbb{R}^2$. Hence, the term Δp in (2.1) is replaced by $\nabla \cdot \big(T^0(\nabla p)\big)$.

If we choose $\Phi^0(\boldsymbol{\eta}) = |\boldsymbol{\eta}|$ then $T^0(\boldsymbol{\eta}) = \boldsymbol{\eta}$ and hence $\nabla \cdot \big(T^0(\nabla p)\big) = \Delta p$ which is the original isotropic case. In the text below, we consider

$$\Phi^0(\boldsymbol{\eta}) = \sqrt{\alpha \cdot (\tilde{\eta}^1)^2 + \beta \cdot (\tilde{\eta}^2)^2}. \tag{2.5}$$

In general, we have $\alpha = \alpha(\boldsymbol{v}(\boldsymbol{x}), \boldsymbol{x}) := \alpha(\boldsymbol{x})$, $\beta = \beta(\boldsymbol{x})$ and $\tilde{\eta}^1, \tilde{\eta}^2$ are the coordinates of the vector $\boldsymbol{\eta}$ in the orthonormal basis $(\frac{1}{v}\boldsymbol{v}, \frac{1}{v}\boldsymbol{v}^\perp)$. Choosing $\alpha \gg \beta$, the term $\Phi^0(\nabla p)$ (as well as the absolute value of the whole anisotropic term $\nabla \cdot \big(T^0(\nabla p)\big)$) will be much larger in case of $\nabla p \parallel \boldsymbol{v}$ than in case of $\nabla p \perp \boldsymbol{v}$.

Remark. The above described anisotropic model is a generalization of the *diffusion tensor* model (see [21, 13]), which is based on replacing Δp by the term

$$\nabla \cdot (\boldsymbol{D}(\nabla p)),$$

where $\boldsymbol{D}$ is a symmetric positive definite matrix. Indeed, it is easy to verify that defining

$$\Phi^0(\boldsymbol{\eta}) = \sqrt{\boldsymbol{\eta}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\eta}},$$

we obtain

$$T^0(\nabla p) = \boldsymbol{D}(\nabla p).$$

On the other hand, our special choice (2.5) can be expressed in terms of the diffusion tensor model. The corresponding tensor is such that it has the form

$$\boldsymbol{D} = \begin{pmatrix} \alpha & \\ & \beta \end{pmatrix},$$

expressed in the basis $(\frac{1}{v}\boldsymbol{v}, \frac{1}{v}\boldsymbol{v}^{\perp})$.

Anisotropic diffusion problem. We formulate the *anisotropic diffusion problem* for the PDE of Allen-Cahn type, which reads as follows:

$$\xi^2 \frac{\partial p}{\partial t} + \xi^2 \boldsymbol{v} \cdot \nabla p = \xi^2 \nabla \cdot T^0(\nabla p) + f_0(p) + c_0 F \xi \qquad \text{in } \mathcal{J} \times \Omega, \tag{2.6}$$

$$p|_{\partial\Omega} = 0 \qquad \text{on } \mathcal{J} \times \partial\Omega, \tag{2.7}$$

$$p|_{t=0} = I \qquad \text{in } \Omega. \tag{2.8}$$

Remark. For the sake of simplicity, we consider the homogeneous Dirichlet boundary condition instead of (2.2) in the problem analysis. However, there is a choice of various combinations of boundary conditions in the computational studies.

3. Mathematical analysis. We introduce the notations

$$(u, v) = \int_\Omega u(\boldsymbol{x}) v(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\,, \; u, v \in \mathrm{L}_2(\Omega),$$

$$(\nabla u, \nabla v) = \int_\Omega \nabla u(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\,, \; u, v \in \mathrm{H}_0^1(\Omega).$$

The norms induced by the above scalar products will be denoted by $\|u\|$, $\|\nabla u\|$ respectively.

The function $p \in \mathrm{L}_2(\mathcal{J}; \mathrm{H}_0^1(\Omega))$ is the *weak solution* of the anisotropic diffusion problem (2.6-2.8) if it satisfies

$$\xi^2 \frac{\mathrm{d}}{\mathrm{d}t}(p, q) + \xi^2 \left(\boldsymbol{v} \cdot \nabla p, q\right) + \xi^2 \left(T^0(\nabla p), \nabla q\right) = \left(f_0(p), q\right) + \left(c_0 F \xi, q\right), \qquad p(0) = I, \tag{3.1}$$

for each $q \in \mathcal{D}(\Omega)$, in the sense of $\mathcal{D}'(\mathcal{J})$.

Theorem. *If $I \in \mathrm{H}_0^1(\Omega)$ and for $\xi > 0$ fixed, there exists a unique weak solution p of the anisotropic diffusion problem (3.1) which satisfies*

$$p \in \mathrm{L}_2(\mathcal{J}; \mathrm{H}^2(\Omega) \cap \mathrm{H}_0^1(\Omega)),$$

$$\frac{\partial p}{\partial t} \in \mathrm{L}_2(\mathcal{J}; L_2(\Omega)).$$

Proof. To show existence, we use the Faedo-Galerkin method together with the method of compactness (see e.g. [10, 7, 5, 4]). We derive the a priori estimate for the problem (2.6-2.8).

As the essential step, we construct a sequence of approximations of the solutions of the original problem, using the orthonormal basis $\{v_i\}_{i\in\mathbb{N}}$ of $\mathrm{L}_2(\Omega)$ consisting of eigenfunctions of the operator $-\Delta$ coupled with the homogeneous Dirichlet boundary

conditions, together with the corresponding eigenvalues $\{\lambda_i\}_{i\in\mathbb{N}}$. Additionally, assume that

$$(\forall i \in \mathbb{N}) \left(v_i \in \mathrm{C}^2(\Omega) \cup \mathrm{C}^1(\bar{\Omega})\right).$$

Let

$$V_m = [v_1, v_2, ..., v_m]_\lambda$$

be a finite-dimensional subspace of $\mathrm{L}_2(\Omega)$ and let

$$\mathcal{P}_m : \mathrm{L}_2(\Omega) \mapsto V_m$$

denote the projector on V_m. We seek for an approximation of the solution of the problem (3.1)

$$p^m : [0, T) \mapsto V_m,$$

which solves the auxiliary problem $(\forall j \in \hat{m})$

$$\xi^2 \frac{\mathrm{d}}{\mathrm{d}t}(p^m, v_j) + \xi^2 \left(\boldsymbol{v} \cdot \nabla p^m, v_j\right) + \xi^2 (T^0(\nabla p^m), \nabla v_j) = (f_0(p^m), v_j) + (c_0 F \xi, v_j), \\ p^m(0) = \mathcal{P}_m p_{ini}. \tag{3.2}$$

We express the solution of (3.2) using the basis functions v_j as

$$p^m(t) = \sum_{j=1}^{m} \gamma_j^m(t) v_j.$$

After substituting this to (3.2), we obtain a system of ODEs for the unknown functions of time $\gamma_i^m(t)$. Using the method of compactness, it is possible to show strong convergence of the subsequence of p^m to the weak solution of (3.1), provided we have the boundedness of p^m and its derivatives in the appropriate norms. These properties follow from the a priori estimate. □

A priori estimate of the auxiliary solution. We multiply the equality (3.2) by $\mathrm{d}\gamma_j^m/\mathrm{d}t$ and sum over j. After rearrangement of the respective terms (see [18]), we obtain

$$\xi^2 \left\| \frac{\partial p^m}{\partial t} \right\|^2 + \frac{\xi^2}{2} \frac{\mathrm{d}}{\mathrm{d}t} \left(\Phi^0(\nabla p^m)^2, 1\right) + \frac{\mathrm{d}}{\mathrm{d}t} \left(w_0(p^m), 1\right) = \xi^2 \left(\frac{1}{\xi} c_0 F - \boldsymbol{v} \cdot \nabla p^m, \frac{\partial p^m}{\partial t} \right), \tag{3.3}$$

where $w_0' = f_0$. Assume that the vector field is bounded, i.e.

$$(\exists V > 0) \left(\forall \boldsymbol{x} \in \Omega\right) \left(\|\boldsymbol{v}(\boldsymbol{x})\| \leq V\right).$$

Due to the Schwarz inequality on $\mathbb{R}^2$ and further using the equivalence of the Euclidean norm and the Φ^0 norm, we can estimate

$$|\boldsymbol{v} \cdot \nabla p^m| \leq V \left|\nabla p^m\right| \leq C \Phi^0(\nabla p^m), \tag{3.4}$$

where $C > 0$. For the estimate of the right hand side, we use the Schwarz and then the Young inequality and get

$$\xi^2 \left(\frac{1}{\xi} c_0 F - \boldsymbol{v} \cdot \nabla p^m, \frac{\partial p^m}{\partial t} \right) \leq \frac{\xi^2}{2} \left(\left\| \frac{\partial p^m}{\partial t} \right\|^2 + \left(\left(\frac{1}{\xi} c_0 F - \boldsymbol{v} \cdot \nabla p^m \right)^2, 1 \right) \right)$$
$$\leq \frac{\xi^2}{2} \left\| \frac{\partial p^m}{\partial t} \right\|^2 + \frac{\xi^2}{2} \left(\left(\frac{1}{\xi} \left| c_0 F \right| + C \Phi^0(\nabla p^m) \right)^2, 1 \right).$$

In the last step, we use the triangle inequality and the relation (3.4). Let us denote $C_1 := \frac{1}{\xi C} \left| c_0 F \right|$. If we use the obvious inequality

$$(a+b)^2 \leq 2a^2 + 2b^2,$$

we can remove the squared sum on the right hand side, so that it is possible to continue in the upper estimates,

$$\xi^2 \left(\frac{1}{\xi} c_0 F - \boldsymbol{v} \cdot \nabla p^m, \frac{\partial p^m}{\partial t} \right) \leq \frac{\xi^2}{2} \left\| \frac{\partial p^m}{\partial t} \right\|^2 + 2C^2 \frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + C^2 \xi^2 \left(C_1^2, 1 \right).$$

After substituting to (3.3), we get

$$\frac{\xi^2}{2} \left\| \frac{\partial p^m}{\partial t} \right\|^2 + \frac{\xi^2}{2} \frac{\mathrm{d}}{\mathrm{d}t} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \frac{\mathrm{d}}{\mathrm{d}t} \left(w_0(p^m), 1 \right) \leq 2C^2 \frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right)$$
$$+ C^2 \xi^2 \left(C_1^2, 1 \right). \qquad (3.5)$$

Due to nonnegativity of the terms $\left\| \frac{\partial p^m}{\partial t} \right\|^2$ and $2C^2 \left(w_0(p^m), 1 \right)$, this yields

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right) \leq 2C^2 \left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right)$$
$$+ C^2 \xi^2 \left(C_1^2, 1 \right). \qquad (3.6)$$

We move the first term of the right hand side to the left and multiply the inequality by $\mathrm{e}^{-2C^2 t}$. Then

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right) \mathrm{e}^{-2C^2 t} \right) \leq C^2 \xi^2 \left(C_1^2, 1 \right) \mathrm{e}^{-2C^2 t}.$$

We integrate over $[0, t]$ and obtain

$$\left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right)(t)$$
$$\leq \left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right)(0) \mathrm{e}^{2C^2 t} + \frac{\xi^2}{2} \left(C_1^2, 1 \right) \underbrace{\left(\mathrm{e}^{2C^2 t} - 1 \right)}_{>0}. \qquad (3.7)$$

In the following, we return to the relation (3.5), where we estimate again the term on the right hand side as

$$2C^2 \frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) \leq 2C^2 \left(\frac{\xi^2}{2} \left(\Phi^0(\nabla p^m)^2, 1 \right) + \left(w_0(p^m), 1 \right) \right).$$

We integrate the resulting inequality with respect to t over the interval $[0,T]$ and acquire

$$\begin{aligned}
&\int_0^T \left(\frac{\xi^2}{2}\left\|\frac{\partial p^m}{\partial t}\right\|^2\right)(t)\mathrm{d}t + \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(T)\\
\leq\ &\left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(0)\\
&+2C^2\int_0^T \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(t)\mathrm{d}t + C^2\xi^2\left(C_1^2,1\right)T.
\end{aligned}$$

According to (3.7), it is possible to estimate the integral on the right as

$$\begin{aligned}
&2C^2\int_0^T \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)\mathrm{d}t\\
&\leq \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(T)\mathrm{e}^{C_F^2T} - \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(0)\\
&+\frac{\xi^2}{2}\left(C_1^2,1\right)\left(\mathrm{e}^{2C^2T}-1\right) - C^2\xi^2\left(C_1^2,1\right)T,
\end{aligned}$$

so that we come to the estimate

$$\begin{aligned}
&\int_0^T \left(\frac{\xi^2}{2}\left\|\frac{\partial p^m}{\partial t}\right\|^2\right)(t)\mathrm{d}t + \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(T)\\
&\leq \left(\frac{\xi^2}{2}\left(\Phi^0(\nabla p^m)^2,1\right)+\left(w_0(p^m),1\right)\right)(0)\mathrm{e}^{C^2T} + \frac{\xi^2}{2}\left(C_1^2,1\right)\left(\mathrm{e}^{2C^2T}-1\right). \qquad (3.8)
\end{aligned}$$

The inequality (3.8) implies that $\frac{\partial p^m}{\partial t}$ and (via the norm equivalence - see (3.4)) also ∇p^m are bounded in $\mathrm{L}_\infty\left(\mathcal{J},\mathrm{L}_2(\Omega)\right)$, independently of m. Thanks to the properties of the double well potential w_0, we also obtain the boundedness of p^m in $\mathrm{L}_\infty\left(\mathcal{J},\mathrm{L}_s(\Omega)\right)$ for any $1\leq s\leq 4$.

4. Numerical solution. For numerical solution, we use the *method of lines*. The spatial discretization is carried out by the finite difference method; for the temporal discretization, we employ a 4th-order Runge-Kutta solver with adaptive time stepping. First, let us introduce the notations

$$\boldsymbol{h}=(h^1,h^2),\ h^k:=\frac{L^k}{m^k},\ k\in\{1,2\},$$

$$\boldsymbol{x}_{i,j}=(x_i,y_j)=(i\cdot h^1, j\cdot h^2),\ \boldsymbol{v}_{i,j}=(v^1_{i,j},v^2_{i,j}):=\boldsymbol{v}(\boldsymbol{x}_{i,j}),\ u_{i,j}=u(\boldsymbol{x}_{i,j}), \qquad (4.1)$$

$$\omega_h=\left\{\left(ih^1,jh^2\right)\middle|\, i=1,...,m^1-1,\ j=1,...,m^2-1\right\},$$

$$\bar{\omega}_h=\left\{\left(ih^1,jh^2\right)\middle|\, i=0,...,m^1,\ j=0,...,m^2\right\},\ \gamma_h=\bar{\omega}_h-\omega_h,$$

$$\mathcal{H}_h = \left\{ w \middle| w : \bar{\omega}_h \to \mathbb{R} \right\}, \ \mathcal{P}_h u = u\big|_{\bar{\omega}_h}. \tag{4.2}$$

In the sense of (4.1), we introduce the following discrete substitutes for derivatives, gradient and divergence:

$$u_{\bar{x}^1,i,j} = \frac{u_{i,j} - u_{i-1,j}}{h^1} \quad , \quad u_{x^1,i,j} = \frac{u_{i+1,j} - u_{i,j}}{h^1},$$

$$u_{\bar{x}^2,i,j} = \frac{u_{i,j} - u_{i,j-1}}{h^2} \quad , \quad u_{x^2,i,j} = \frac{u_{i,j+1} - u_{i,j}}{h^2},$$

$$u_{\bar{x}^1 x^1,i,j} = (u_{\bar{x}^1})_{x^1,i,j} = \frac{1}{(h^1)^2} \left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right), \tag{4.3}$$

$$\bar{\nabla}_h u = (u_{\bar{x}^1}, u_{\bar{x}^2}) \quad , \quad \nabla_h u = (u_{x^1}, u_{x^2}) \quad , \quad \Delta_h u = u_{\bar{x}^1 x^1} + u_{\bar{x}^2 x^2},$$

$$\nabla_h \cdot \boldsymbol{V} = V^1_{x^1} + V^2_{x^2}, \ \bar{\nabla}_h \cdot \boldsymbol{V} = V^1_{\bar{x}^1} + V^2_{\bar{x}^2}, \ \boldsymbol{V} = (V^1, V^2).$$

Using the above definitions, we assemble the semi-discrete scheme of the problem (2.6-2.8) for the unknown grid function $p^h : \mathcal{J} \to \mathcal{H}_h$:

$$\xi^2 \frac{\mathrm{d}p^h}{\mathrm{d}t} + \xi^2 \mathcal{P}_h(\boldsymbol{v}) \cdot \nabla_h p^h = \xi^2 \nabla_h \cdot \left(T^0(\bar{\nabla}_h p^h) \right) + f_0(p^h) + c_0 F \xi \quad \text{in } \mathcal{J} \times \omega_h, \tag{4.4}$$

$$p^h|_{\gamma_h} = 0 \qquad \text{on } \mathcal{J} \times \gamma_h, \tag{4.5}$$

$$p^h(0) = \mathcal{P}_h I \qquad \text{in } \omega_h. \tag{4.6}$$

Interpolating the grid function p^h to Ω and using the method of compactness, it is possible to show (see [6]).

Theorem. *Let $I \in H^2(\Omega)$. Then the solution p^h of the semi-discrete scheme converges in $L_2(\mathcal{J}, L_2(\Omega))$ to the unique weak solution of the problem (2.6-2.8).*

Remark. The above stated convergence is understood in the sense

$$\lim_{h \to 0} \left\| p - \mathcal{Q} p^h \right\|_{\mathrm{L}_2(\mathcal{J}, \mathrm{L}_2(\Omega))} = 0,$$

where $\mathcal{Q}$ is the interpolation operator.

5. Parallelization. The numerical algorithm has been parallelized using MPI (Message Passing Interface, see [17]), the widely used message passing library designed for parallel computing on *distributed memory* systems. The idea of parallelization of the finite difference algorithm is to divide (decompose) the grid ω_h into blocks, each of those being handled by different process. Our choice was to compose a block of several rows of the grid, as shown in Fig. 5.1a.

For the sake of the following explanation, suppose the division into blocks in the manner „b“ and suppose we use Runge-Kutta temporal discretization. We solve a system of ODEs in the form $\dot{\boldsymbol{p}} = \boldsymbol{f}(t, \boldsymbol{p})$, where $\boldsymbol{p}$ represents a vector of unknown scalar functions of time, in our case $\boldsymbol{p} = p^h$. Due to the use of the 7-point scheme, for the calculation of the right hand side $\boldsymbol{f}(t, \boldsymbol{p})$ in the nodes at the boundary of the block (I, J), we need to know the values of $\boldsymbol{p}$ lying in up to three different adjacent blocks (again, see Fig. 5.1). Similar situation also occurs in the bottom left corner of the block (I, J).

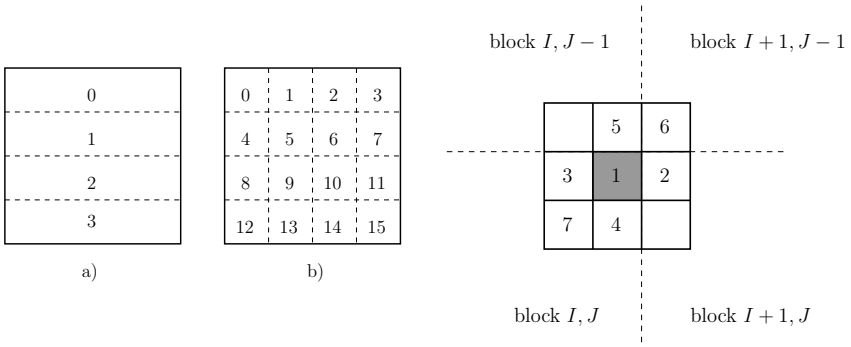

FIGURE 5.1. *Different methods of grid decomposition*

Configuration	Time [m:s]
1+1	6:12.77
2+1	3:49.73
3+1	3:05.42

Load balancing disabled

Configuration	Time [m:s]
1+1	3:42.75
2+1	2:44.04
3+1	1:04.89

Load balancing enabled

TABLE 5.1
Computation times on a given number of „fast“ + „slow“ machines.

Generally, each block lying in the interior of the grid must obtain values from 6 of 8 adjacent blocks. Passing the boundary values among blocks is called *synchronization*. During the coefficient calculation in the Runge-Kutta method, the previous coefficient appears in the argument of the right hand side. It is therefore necessary to perform synchronization before the calculation of each coefficient. Synchronization can easily take advantage of nonblocking (asynchronous) MPI communication (see [17, 18]).

Load balancing. If we divide the grid of $n \times n$ nodes into $N \times N$ blocks, the overall length of the block boundaries expressed by the number of nodes will be $2(N-1)n$. However, in case of the division into N^2 row blocks, the boundary length will be $\left(N^2-1\right)n$. Already with four processes ($N = 2$), we interchange one half more data. On the other hand, the row block division is easier to implement and moreover, it has the following advantages:

1. The number of processes can be arbitrary. (not necessarily a multiple of M or N).
2. The individual blocks may have different sizes.

The above properties allowed us to propose an interesting method of dynamic load balancing during the calculation. It is based on the changes of the block sizes, corresponding to the particular processes. For a given period, each process accumulates the wall time of its autonomous calculations (between synchronizations). The acquired time values are then converted to relative speeds of the processes. Afterwards, the master process calculates the new block sizes, proportional to the process speeds. We assume that with such block sizes, the idle times of the processes (waiting for synchronization) should be eliminated. Rearrangement of the blocks requires data to be redistributed among the blocks. The algorithm implementation tries to minimize the amount of data being sent and provides mechanisms to avoid meaningless rearrangements (when the changes to be made are negligible).

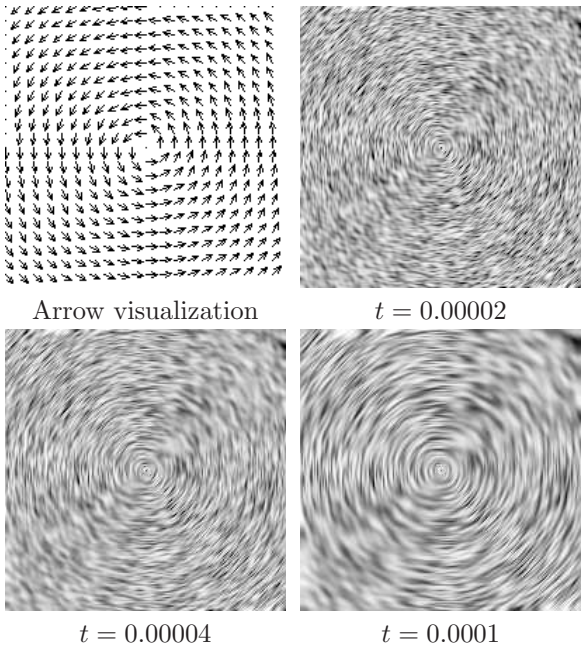

FIGURE 5.2. *Rotation.* $m = n = 200, \xi = 0.005, \gamma = 400, \sigma = 200, \kappa = 0.01$

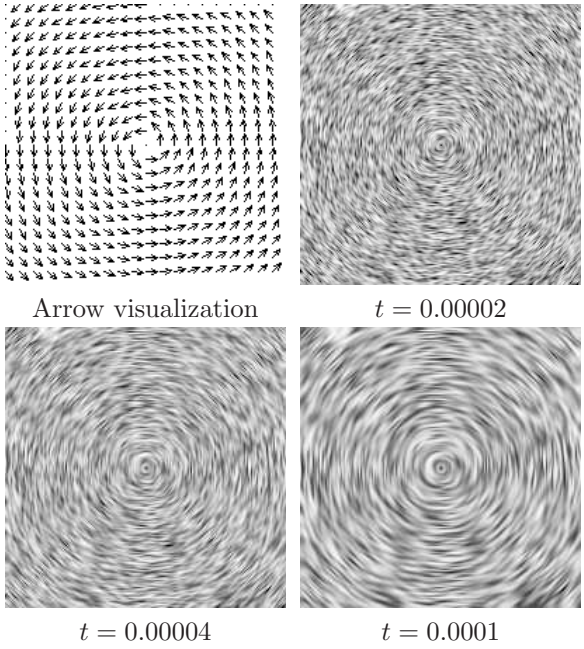

FIGURE 5.3. *Rotation. The same as in Fig. 5.2 calculated using the upwind scheme.*

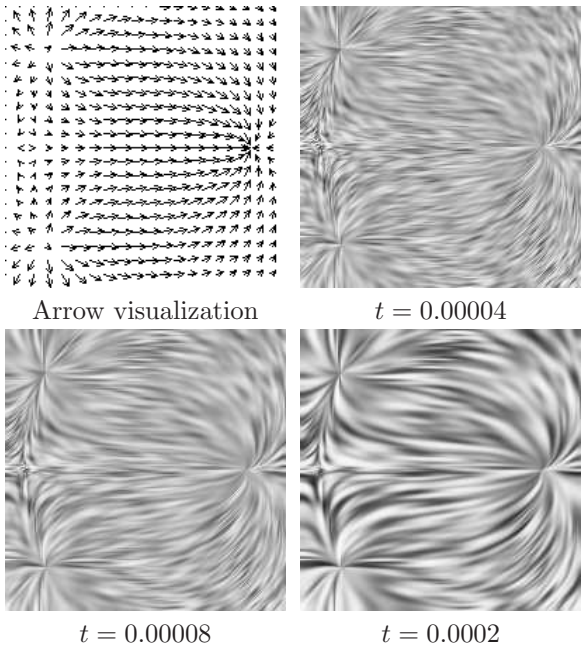

FIGURE 5.4. *3 el. charges.* $m = n = 200, \xi = 0.005, \gamma = 40, \sigma = 100, \kappa = 0.05$

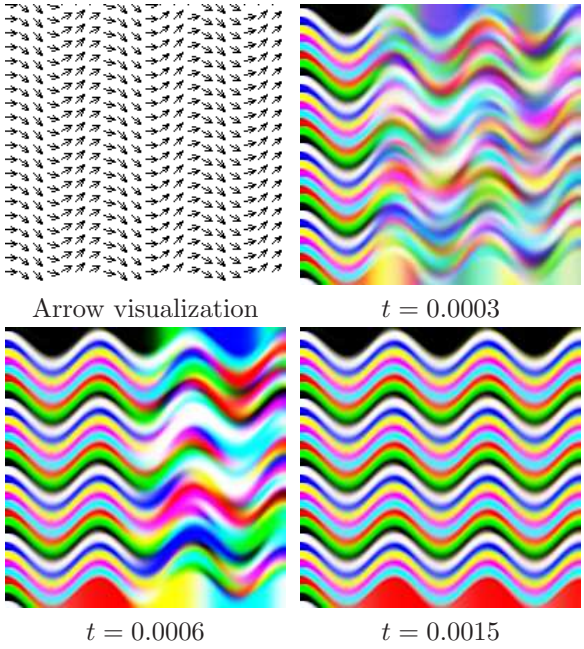

FIGURE 5.5. *Sine field.* $m = n = 200, \xi = 0.005, \gamma = 1000, \sigma = 50, \kappa = 0.1$

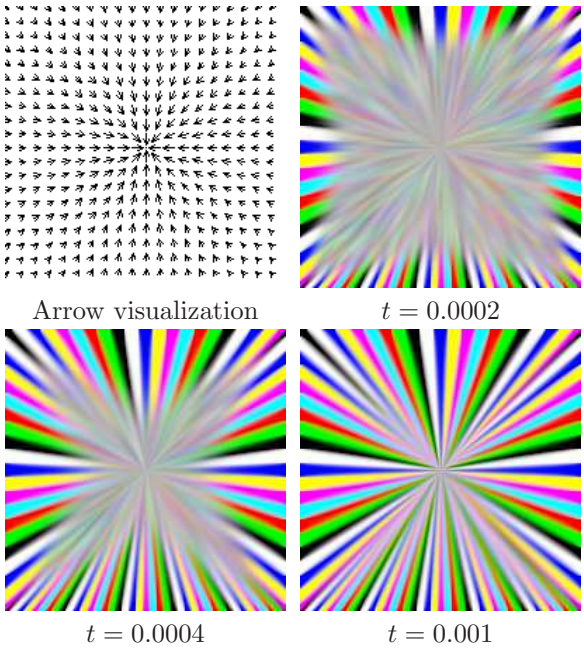

FIGURE 5.6. *Neg. el. charge.* $m = n = 200, \xi = 0.01, \gamma = 1000, \sigma = 50, \kappa = 0.1$

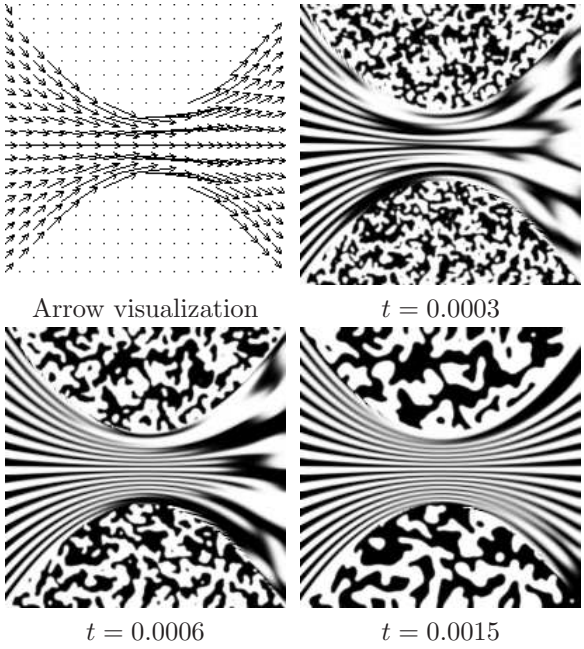

FIGURE 5.7. *A channel.* $m = n = 400, \xi = 0.003, \gamma = 600, \sigma = 50, \kappa = 0.1$

Thanks to the use of dynamic load balancing, we can employ machines with much different performance in the calculation. We can also load one or more of the machines with a different task without noticing a significant slowdown of the whole calculation. This is because the largest amount of work is always assigned to those machines where most performance is available. A sample demonstration of the load balancing benefits can be found in Table 5.1.

The proposed load balancing system is not suitable for advanced homogeneous cluster solutions controlled by resource managers. On such a system, all nodes utilized by the user application have the same performance and they are fully at its disposal for the whole program run time. No load balancing is therefore necessary.

6. Results. The results shown in figures have been obtained using the multiplicative factor γ at the advective term $\boldsymbol{v} \cdot \nabla p$ in (2.6) and using the choice

$$\alpha = \kappa\,(1 + \sigma v)\,,\ \beta = \kappa,$$

where $\kappa > 0$, $\sigma > 0$ (see Section 2). The first of the three pictures in a row always represents the vector field indicated by arrows, the following two pictures demonstrate the evolution of the image in time. All calculations were performed in $\Omega = (0,1) \times (0,1)$; the remaining parameters are given below figures. The boundary condition is always (2.2) except for Figs. 5.5, 5.6 and 5.7, where, at some part of the boundary, it has been replaced by the Dirichlet boundary condition in the form of a pattern.
Remark. In Fig. 5.2, it can be seen that in some directions, the anisotropy of the smearing is relatively poor. This issue is caused by the chosen numerical scheme. In the most recent work, it has been effectively eliminated by using a scheme of upwind type for the discretization of the term $\nabla \cdot T^0(\nabla p)$ in (2.6). The difference is shown in Fig. 5.3.

Acknowledgement. The work was partly supported by the project No. MSM 6840770010 "Applied Mathematics in Technical and Physical Sciences", and by the project No. LC06052 "Jindřich Nečas Center for Mathematical Modelling", both of the Ministry of Education, Youth and Sports of the Czech Republic.

REFERENCES

[1] B. Barney, *The OpenMP Tutorial*. `http://www.llnl.gov/computing/tutorials/openMP/`
[2] L. Bednárik, *Parallel algorithms for numerical solution of PDEs.* Research work, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2005. (in Czech)
[3] G. Bellettini, and M. Paolini, *Anisotropic motion by mean curvature in the context of Finsler geometry*. Hokkaido Math J. **25** (1996), 537–566.
[4] M. Beneš *Diffuse-interface treatment of the anisotropic mean-curvature flow*. Appl. Math. **48**, no. 6 (2003), 437–453.
[5] M. Beneš, *Mathematical analysis of phase-field equations with numerically efficient coupling terms*. Interfaces and Free Boundaries **3** (2001), 201–221.
[6] M. Beneš *Mathematical modelling of solidification in crystalline materials using nonlinear PDEs*. Habilitation thesis, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2001. (in Czech)
[7] M. Beneš, *On a Phase-Field Model with Advection*. In *Numerical Mathematics and Advanced Applications, ENUMATH 2003 (peer reviewed proceedings)*, pages 141–150, Springer Verlag, Berlin, 2004. eds. M. Feistauer, V Dolejší, P. Knobloch, K. Najzar, ISBN 3-540-21460-7.
[8] M. Beneš, V.Chalupecký, and K. Mikula, *Geometrical image segmentation by the Allen-Cahn equation*. App. Num. Math. **51** (2004), 187–205.
[9] U. Diewald, T. Preusser, and M. Rumpf *Anisotropic diffusion in vector field visualization on euclidean domains and surfaces.* Trans. Vis. and Comp. Graphics **6**, no. 2 (2000), 139-149.

[10] L. C. Evans, *Partial Differential Equations.* American Mathematical Society, Providence, 1998. ISBN 0-8218-0772-2.
[11] The LAM/MPI Team, *LAM/MPI User's Guide.* Tech. report, 2006.
[12] P. Perona, and J. Malik, *Scale space and edge detection using anisotropic diffusion.* IEEE Trans. Pattern Anal. Mach. Intell. **12** (1990), 629–639.
[13] M. Petřek, *Algorithm of mathematical visualization and their use in mathematical modeling.* Diploma thesis, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2004, (in Czech).
[14] J. A. Sethian, *Level Set Methods.* Cambridge University Press (1996).
[15] L. Schwartz, *Mathematical methods in Physics.* SNTL, Praha (1972), (in Czech).
[16] R. E. Showalter, *Hilbert Space Methods for Partial Differential Equations.* Electronic Journal of Diff. Eq., Monograph 01, 1994.
[17] M. Snir, S. Otto, S. Huss-Ledermann, D. Walker, and J. Dongarra, *MPI: The Complete Reference.* MIT Press (1995).
[18] P. Strachota, *Degenerate diffusion in mathematical visualization.* Investigation project, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2006 (in Czech).
[19] P. Strachota, *Degenerate diffusion in mathematical visualization.* Research work, Dept. of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, 2005. (in Czech)
[20] M. Šenkýř, J. Mikyška, and M. Beneš, Application of Parallel Computing Techniques for Problems of Degenerate Diffusion. In *Numerical Mathematics and Advanced Applications, ENUMATH 2003 (peer reviewed proceedings)*, pages 756–766, Springer Verlag, Berlin, 2004. eds. M. Feistauer, V Dolejší, P. Knobloch, K. Najzar, ISBN 3-540-21460-7.
[21] D. Tschumperlé, and R. Deriche, *Tensor Field Visualization with PDE's and Application to DT-MRI Fiber Visualization.* INRIA Sophia-Antipolis, Odyssée Lab, France (2004).
[22] M. Virius, *Programming in C++.* Czech Technical University in Prague, 2001. (in Czech)
[23] J. Žára, B. Beneš, J. Sochor, and P. Felkel, *Modern methods in computer graphics.* Computer Press (2004), (in Czech).