Department of Mathematics Journal Articles

Department of Mathematics

2010

Moving-boundary problems solved by adaptive radial basis functions

L. Vrankar

E.J. Kansa

L. Ling
Hong Kong Baptist University, lling@hkbu.edu.hk

F. Runovc

G. Turk

Citation

Vrankar, L., E.J. Kansa, L. Ling, F. Runovc, and G. Turk. "Moving-boundary problems solved by adaptive radial basis functions." *Computers & Fluids* 39.9 (2010): 1480-1490.

Elsevier Editorial System(tm) for Computers and Fluids
Manuscript Draft

Corresponding Author: Mr. Leopold Vrankar, Ph.D.

Corresponding Author's Institution: Slovenian Nuclear Safety Administration

First Author: Leopold Vrankar, PhD

Order of Authors: Leopold Vrankar, PhD; Edward J Kansa, PhD; Leevan Ling, PhD; Franc Runovc, PhD; Goran Turk, PhD

Abstract: Moving-boundary problems are often called the Stefan problems, with reference to the early work of J. Stefan, around 1890, when he was interested in the melting of the polar ice cap. Moving boundaries are also associated with time-dependent problems and the position of the boundary has to be determined as a function of time and space. Recent research on the numerical method has focused on the idea of using a meshless methodology for the numerical solution of partial differential equations (PDEs). The level set method (LSM) has become an attractive design tool for tracking, modelling and simulating the motion of free boundaries in fluid mechanics, combustion, computer animation and image processing. The objective of this paper is to present an alternative approach to the conventional level set methods for solving two-dimensional moving-boundary problems. In our case we will use the multiquadrics radial basis functions. The solution process is stabilized by an adaptive greedy algorithm.

Moving-Boundary Problems Solved by Adaptive Radial Basis Function

L.Vrankar[a,d,*], E. J. Kansa[b], L. Ling[c], F. Runovc[d], G. Turk[e]

[a]*Slovenian Nuclear Safety Administration, Železna cesta 16, Ljubljana, Slovenia*
[b]*Department of Mechanical and Aeronautical Engineering, University of California, USA*
[c]*Department of Mathematics, Hong Kong Baptist University, Hong Kong*
[d]*University of Ljubljana, Faculty of Natural Sciences and Engineering, Aškerčeva cesta 12, Ljubljana, Slovenia*
[e]*University of Ljubljana, Faculty of Civil and Geodetic Engineering, Jamova cesta 2, Ljubljana, Slovenia*

Abstract

Moving-boundary problems are often called the Stefan problems, with reference to the early work of J. Stefan, around 1890, when he was interested in the melting of the polar ice cap. Moving boundaries are also associated with time-dependent problems and the position of the boundary has to be determined as a function of time and space. Recent research on the numerical method has focused on the idea of using a meshless methodology for the numerical solution of partial differential equations (PDEs). The level set method (LSM) has become an attractive design tool for tracking, modelling and simulating the motion of free boundaries in fluid mechanics, combustion, computer animation and image processing. The objective of this paper is to present an alternative approach to the conventional level set methods for solving two-dimensional moving-boundary problems. In our case we will use the multiquadrics radial basis functions. The solution process is stabilized by an adaptive greedy algorithm.

Key words: moving-boundary problems, Stefan problems, level set method, multiquadric, partial differential equations, adaptive greedy algorithm

[*]Corresponding author: Email addresses: leopold.vrankar@gov.si

1. Introduction

Moving-boundary problems are often called the Stefan problems, with reference to the early work of J. Stefan, around 1890, when he was interested in the melting of the polar ice cap [1]. In this frame, we can include a large number of important physical processes involving heat conduction and materials undergoing a change of phase. One of these processes is the heat transfer problems involving phases change due to solidification or melting, that is important in many industrial applications such as the drilling of high ice-content soil, the storage of thermal energy and the safety studies of nuclear reactors.

Due to their wide range of applications, the phase change problems have drawn the considerable attention of mathematicians, engineers and scientists.

One common feature of phase change problems is that the location of the solid-liquid interface is not known *a priori* and must be determined during the course of the analysis. Mathematically, the motion of the interface is expressed implicitly in an equation for the conservation of thermal energy at the interface, the so-called Stefan conditions. This introduces a non-linear character to the system, which renders each problem somewhat uniquely. Moving boundaries are also associated with time-dependent problems and the position of the boundary has to be determined as a function of time and space.

Various numerical methods are known to solve the Stefan problems, e.g. front-tracking, front-fixing, and fixed-domain methods [2]. The finite-difference methods and finite-element techniques have been used extensively for numerical solution of moving boundary problems [3, 4].

During the past decade, increasing attention was given to the development of meshless methods using radial basis functions (RBFs) for the numerical solution of PDEs [5].

A level set method has become an attractive design tool for tracking, modeling and simulating the motion of free boundaries in fluid mechanics, combustion, computer animation and image processing [6–10]. In this paper, the RBF approaches are incorporated into the level set method to construct a more efficient approach. The RBFs, coupled with some adaptive techniques, are also used to prevent the occurrence of the flat level sets and to maintain the behaviour of the level set function at the front without reinitialization. The objective of this paper is to present an alternative approach to the conventional level set methods for solving two-dimensional moving-

boundary problems. An adaptive trial space selection algorithm is employed in the solution process.

2. Meshless radial basis functions (RBFs)

Recent research on the numerical method has focused on the idea of using a meshless methodology for the numerical solution of PDEs. One of the common characteristics of all mesh-free methods is their ability to construct functional approximation or interpolation entirely from information at a set of scattered nodes, among which there is no relationship or connectivity needed.

Three different approaches to develop meshless methods have been successfully proposed till now. The first one is based on the spirit of the finite element method and employs Petrov-Galerkin weak formulation [11]. The second one is of boundary element type [12]. The third approach employs the RBFs. The base of this approach is its employment of different interpolating functions to approximate solutions of differential equations. Kansa [13] introduced multiquadric functions to solve hyperbolic, parabolic and elliptic differential equations with collocation methods. One of the most powerful RBF methods is based on multiquadric basis functions (MQ), first used by R. L. Hardy [14].

A radial basis function is the function $\varphi_j(\mathbf{x}) := \varphi(||\mathbf{x} - \mathbf{x}_j||)$, which depends only on the distance between $\mathbf{x} \in \mathbb{R}^d$ and a fixed point $\mathbf{x}_j \in \mathbb{R}^d$. Here, φ_j is continuous and bounded on any bounded sub-domain $\Omega \subseteq \mathbb{R}^d$ whereas $\varphi : \mathbb{R}^d \to \mathbb{R}$. Let $r \geq 0$ denote the Euclidean distance between any pair of points in the domain Ω. The commonly used radial basis functions are: linear ($\varphi(r) = r$), cubic ($\varphi(r) = r^3$), thin-plate spline ($\varphi(r) = r^2 \log r$) and Gaussian ($\varphi(r) = e^{-\alpha^2 r}$). The most popular globally supported C^∞ RBFs are MQ ($\varphi(r) = c^\beta(1 + (r/c)^2)^\beta$) (MQ-prewavelet).

The parameter $c > 0$ is a shape parameter controlling the fitting of a smoothing surface to the data. It has big influence on the accuracy of the solution. For this reason, in almost all previous researches the shape parameter must be adjusted with the number of centers in order to produce equation systems that are sufficiently well conditioned to be solved with standard finite precision arithmetic. The optimal choice of the constant shape parameter is still an open question, and it is most often selected by trial and error approach. To overcome these problems with ill-conditioned matrix

many efforts have been made to find a new computational method that is capable of circumventing the ill-conditioning problems using linear solvers. In the literature the following methods are reported: *(a)* Using variable shape parameters [5, 13], *(b)* preconditioning the coefficient matrix, see Ling and Kansa [15, 16], *(c)* using domain decomposition methods in overlapping or non-overlapping schemes that decompose a very large ill-conditioned problem into many subproblems with better conditioning [13], *(d)* optimizing the center locations by the Greedy algorithm. Greedy algorithm chooses optimal c_j at location x_j from maxminizing residual errors, *(e)* using an improved numerical solver based on affine space decomposition [17], *(f)* using complex MQ shape parameters [18], etc. More information can be found in [19]. Combination of the above can be used as optimized solution procedure and accuracy.

2.1. Asymmetric radial basis function collocation methods for stationary problems

We briefly review the RBF asymmetric collocation scheme. We consider a PDE in the general form of

$$\mathrm{L}u(\mathbf{x}) = f(\mathbf{x}), \ \ \text{in} \ \ \Omega \subset \mathbb{R}^d, \tag{1}$$

$$\mathrm{B}u(\mathbf{x}) = g(\mathbf{x}), \ \ \text{on} \ \ \partial\Omega, \tag{2}$$

where u is the unknown solution, d denotes the dimension, $\partial\Omega$ is the boundary of the domain Ω, L is the differential operator on the interior, and B is the operator that specifies the boundary conditions of the Dirichlet, Neumann or mixed type. Both, f and g, are given functions with sufficient smoothness mapping $\mathbb{R}^d \mapsto \mathbb{R}$. Using the Kansa's asymmetric multiquadric collocation method, the unknown PDE solution u is approximated by RBFs in the form:

$$u \approx U(\mathbf{x}) = \sum_{j=1}^N \alpha_j \varphi_j(\mathbf{x}) + \sum_{l=1}^Q \gamma_l \nu_l(\mathbf{x}), \tag{3}$$

where φ can be any type of radial basis function, and $\nu_l, ..., \nu_M \in \prod_q^d$, are polynomials of degree m or less, $Q := \begin{pmatrix} q - 1 + d \\ d \end{pmatrix}$, see [20]. Let $(\mathbf{x}_j)_{j=1}^N$ be the N collocation points in $\Omega \bigcup \partial\Omega$. We assume the collocation points are arranged in such a way that the first N_I points are in Ω, whereas the last N_B points are on $\partial\Omega$. To evaluate or approximate the $N+Q$ unknown coefficients,

at least, $N + Q$ linearly independent equations are needed. Ensuring that $U(\mathbf{x})$ satisfies equations (1) and (2) at the collocation points results in a good approximation of the solution u. The first N equations are given by

$$\sum_{j=1}^{N} \alpha_j L\varphi_j(\mathbf{x}_i) + \sum_{l=1}^{Q} \gamma_l L\nu_l(\mathbf{x}) = f(\mathbf{x}_i), \quad \text{for} \ \ i = 1, ..., N_I, \tag{4}$$

and the others are given as

$$\sum_{j=1}^{N} \alpha_j B\varphi_j(\mathbf{x}_i) + \sum_{l=1}^{Q} \gamma_l B\nu_l(\mathbf{x}) = g(\mathbf{x}_i), \quad \text{for} \ \ i = N_I + 1, ..., N_I + N_B. \tag{5}$$

The last Q equations could be obtained by imposing some extra condition on $v(\cdot)$:

$$\sum_{j=1}^{N} \alpha_j \nu_k(\mathbf{x}_j) = 0, \quad k = 1, ..., Q. \tag{6}$$

The above procedure leads to the equivalent matrix form: $\mathbf{Ax} = \mathbf{b}$ or

$$\begin{bmatrix} W_L & v_L \\ W_B & v_B \\ v^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \gamma \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}, \tag{7}$$

where

$$W_L = L\varphi_j(\mathbf{x}_i), \quad \mathbf{x}_i \in X_I, \tag{8}$$
$$v_L = L\nu_l(\mathbf{x}_i), \quad \mathbf{x}_i \in X_I, \tag{9}$$
$$W_B = B\varphi_j(\mathbf{x}_i), \quad \mathbf{x}_i \in X_B, \tag{10}$$
$$v_B = B\nu_l(\mathbf{x}_i), \quad \mathbf{x}_i \in X_B, \tag{11}$$
$$v = \nu_k(\mathbf{x}_i), \quad k = 1, ..., Q. \tag{12}$$

The implementation of the RBF method for solving PDEs is straightforward and easy to implement. Since the RBF is differentiable, spatial derivatives are computed by simply differentiating the MQ-RBFs; time derivatives are computed by differentiating the time dependent expansions coefficients. The spatial and temporal partial derivatives, together, result in either a linear or nonlinear sets of ordinary differential equations.

3. The level set method (LSM)

The LSM is a numerical technique for tracking shapes and interfaces. The advantage of the LSM is that one can perform numerical computations involving curves and surfaces on fixed Cartesian grid. It is not necessary to parameterize the object [21]. The LSM makes it very easy to follow shapes that change topology (e.g., when a shape splits in two or develops holes).

3.1. Level set function and equation

In two dimensions, the LSM amounts to representing a closed curve $\Gamma \in \mathbb{R}^2$ in the plane as the zero level set of a two-dimensional auxiliary function $\Phi(\mathbf{x}, t) : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$, where $\mathbf{x}$ is a position of the interface, t is a moment in time. Therefore, the closed curve is presented as:

$$\Gamma = \{(\mathbf{x}, t)|\ \Phi(\mathbf{x}, t) = 0\}. \tag{13}$$

The function Φ is also called a level set function and is assumed to take positive values inside the region delimited by the curve Γ and negative values elsewhere [6, 7]. The level set function can be defined as a signed distance function to the interface. The moving interface is then captured at all time by locating the set of $\Gamma(t)$ for which Φ vanishes. The movement of the level set function can be described as the following Cauchy problem [22]:

$$\frac{\partial \Phi}{\partial t} + v^T \nabla \Phi = 0, \quad \Phi(\mathbf{x}, 0) = \Phi_0(\mathbf{x}), \tag{14}$$

where $\Phi_0(\mathbf{x})$ means the initial position of the interface and $v^T = [\nu_1, \nu_2]$ is the continuous field, which is a function of position $\mathbf{x}$.

This is a partial differential equation, and can be solved by using finite difference method on a Cartesian grid [6, 7]. Simple finite difference methods (FDM) fail quickly. The reason for this is that FDM includes upwinding. In other words, FDM introduces artificial numerical dispersion that can be very significant as compared to the physical dispersion [23].

Upwinding methods, such as the Godunov method, perform better; however the LSM does not guarantee the conservation of the volume and the shape of the level set in an advection field that does conserve the shape and size. Instead, the shape of the level set may get severely distorted and the level set may vanish over several time steps. Upwinding also introduces artificial viscosity in the solutions that must be considered. For this reason,

6

high-order finite difference schemes are generally developed, such as high-order essentially non-oscillatory (ENO) schemes [6, 7], and even then, the feasibility of long-time simulations is questionable.

In the last decade, the MQ RBF has also been used for modeling the LSMs [24]. Namely, the RBF solution has far less diffusion than the finite difference method that includes upwinding [23]. But time integration should not have diffusion. Due to above mentioned reasons we decided to incorporate RBFs into level set methods to construct a more efficient approach. At the initial time, all the time dependent variables should be specified over entire domain.

4. Level set equation construction with RBFs

The level set equation can be constructed in many ways. In our case it will be presented by the equation with exact time integration, and implicit discrete and Crank-Nicolson implicit schemes. The first one, an exact time marching scheme can be constructed as the initial value problem which can be considered equivalent to an interpolation problem. The second one is the classical time marching scheme which is one of the first order type schemes. The last one is special type of differentiating employed in eq. (14) which is called the Crank-Nicolson form and it is second order type shemes. In our case we stabilize the classical time marching schemes with the adaptive greedy algorithm.

4.1. Interpolation of the level set function

The starting point of the use of RBFs to solve partial differential equations is the interpolation problem. The MQ RBFs is used to interpolate the scalar level set function by using MQ basis centered at these *RBF centers*, $\Xi = \{\xi_1, \ldots, \xi_N\}$. The resulting RBF interpolant of the level set function can be written as:

$$\Phi(\mathbf{x}, t) = \sum_{j=1}^{N} \alpha_j(t) \varphi_j(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j(t) \varphi(\mathbf{x} - \xi_j), \qquad (15)$$

where $\alpha_j(t)$ is the weight of the radial basis function positioned at the j-th center. Knowing the initial data values $f_1, \ldots f_N \in \mathbb{R}$ at the data points $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_N\} \subset \Omega \subset \mathbb{R}^N$, the RBF interpolant (15) can be obtained by solving the system of N linear equations for N unknown coefficients:

$$\Phi(\mathbf{x}_i, 0) = f_i \quad i = 1, ..., N. \qquad (16)$$

7

With $\mathbf{x} = \mathbf{\Xi}$, the above equation can be re-written in matrix form as:

$$\mathbf{H}(\mathbf{x}, \mathbf{\Xi})\,\alpha(0) = \mathbf{f}, \tag{17}$$

where, for $\mathbf{x}_k \in \mathbf{x}$, $k = 1, \dots, N$,

$$\mathbf{H}(\mathbf{x}, \mathbf{\Xi}) = \begin{bmatrix} \varphi_1(x_1) & \cdots & \varphi_N(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_1(x_N) & \cdots & \varphi_N(x_N) \end{bmatrix} \in \mathbb{R}^{N \times N}, \tag{18}$$

$$\alpha(0) = [\alpha_1(0) \cdots \alpha_N(0)]^{\mathrm{T}}, \tag{19}$$

$$\mathbf{f} = [f_1 \cdots f_N]^{\mathrm{T}} \in \mathbb{R}^N. \tag{20}$$

The generalized expansion coefficients can be obtained by

$$\alpha(0) = \mathbf{H}(\mathbf{x}, \mathbf{\Xi})^{-1}\,\mathbf{f}. \tag{21}$$

The resulting RBF interpolant of the level set function can be re-written compactly as

$$\Phi(\mathbf{x}, 0) = \mathbf{H}(\mathbf{x}, \mathbf{\Xi})\,\alpha(0). \tag{22}$$

4.2. Exact time integration scheme of the level set equation

Equation (15) is time dependent, therefore we can further assume that all knots are fixed in space and the space and time are separable. The resulting RBF interpolant of the level set function (22) becomes time dependent as:

$$\Phi(\mathbf{x}, t) = \mathbf{H}(\mathbf{x}, \mathbf{\Xi})\,\alpha(t) =: \mathbf{H}\,\alpha(t). \tag{23}$$

At any arbitrary point $\mathbf{x} \in \Omega$, substituting equation (23) into (14) gives:

$$\mathbf{H}\frac{d\alpha}{dt} + v^T(\triangledown\mathbf{H})\alpha = 0, \tag{24}$$

where $\triangledown\mathbf{H} = \left[\frac{\partial\mathbf{H}}{\partial x}\frac{\partial\mathbf{H}}{\partial y}\right]^T$ with

$$\frac{\partial\mathbf{H}}{\partial x} = \left[\frac{\partial\varphi_1}{\partial x} \cdots \frac{\partial\varphi_N}{\partial x}\right]^{\mathrm{T}} \in \mathbb{R}^N, \tag{25}$$

$$\frac{\partial\mathbf{H}}{\partial y} = \left[\frac{\partial\varphi_1}{\partial y} \cdots \frac{\partial\varphi_N}{\partial y}\right]^{\mathrm{T}} \in \mathbb{R}^N. \tag{26}$$

The initial value problem can be considered equivalent to the interpolation problem since the expansion coefficients at the initial time are found as the solution of the interpolation problem. Therefore the preliminary starting point of the use of RBFs to solve PDEs is the interpolation problem that is equivalent to solving the initial value problem. The original equation (14) is thus converted into a time-dependent interpolation problem for the initial values of expansion coefficients and the propagation of the front is governed by the time dependent equation.

Using the present collocation method, the set of resulting ODEs can be compactly written as:

$$\frac{d\boldsymbol{\alpha}}{dt} + \mathbf{E}\boldsymbol{\alpha} = \mathbf{0}, \tag{27}$$

where $\mathbf{E} = \mathbf{H}^{-1}\nu_1\partial_x\mathbf{H} + \mathbf{H}^{-1}\nu_2\partial_y\mathbf{H}$.

The set of ODEs can be solved by several ODE solvers such as the first-order forward Euler's method and higher-order Runge-Kutta, Runge-Kutta-Fehlberg, Adams-Bashforth, or Adams-Moulten methods [25].

In our case the solution has the following form:

$$\boldsymbol{\alpha}(t + \Delta t) = \mathbf{expm}(-\mathbf{E}\Delta t)\boldsymbol{\alpha}(t - \Delta t), \tag{28}$$

where **expm** is a MATLAB exponential matrix function representing the series or a rational fraction:

$$\mathbf{expm}(-\mathbf{E}\Delta t) = \mathbf{I} - \mathbf{E}\Delta t + (\Delta t^2/2!)\mathbf{E} \mp \cdots. \tag{29}$$

4.3. Time marching schemes of the level set equation

In implicit discrete schemes, problem (14) is descretized with respect to both time and space variables. The discretization of the problem in time is then accomplished by a time-stepping scheme, followed by the spatial discretization based on the RBF method.

4.3.1. Implicit discrete scheme and MQ RBF method

We consider the implicit scheme of (14):

$$\frac{\Phi^{n+1} - \Phi^n}{\triangle t} + \nu_1\frac{\partial\Phi^{n+1}}{\partial x} + \nu_2\frac{\partial\Phi^{n+1}}{\partial y} = 0, \tag{30}$$

where $t_{n+1} = t_n + \triangle t$, Φ^{n+1} and Φ^n are the level set variable at time t_{n+1} and t_n.

The approximate solution is expressed as:

$$\Phi(\mathbf{x}, t_{n+1}) = \sum_{j=1}^{N} \alpha_j^{n+1} \varphi_j(\mathbf{x}), \tag{31}$$

where α_j^{n+1}, $j = 1, ..., N$, are the unknown coefficients to be determined and $\varphi_j(\mathbf{x}) = \sqrt{(x - x_j) + (y - y_j) + c^2}$ are Hardy's multiquadrics functions.

By substituting equation (31) into (30), we obtain:

$$\sum_{j=1}^{N} \left(\frac{\varphi_j}{\triangle t} + \nu_1 \frac{\partial \varphi_j(\mathbf{x}_i)}{\partial x} + \nu_2 \frac{\partial \varphi_j(\mathbf{x}_i)}{\partial y} \right) \alpha_j^{n+1} = \frac{\Phi^n(\mathbf{x}_j)}{\triangle t} \quad i = 1, ..., N. \tag{32}$$

4.3.2. Crank-Nicolson implicit and MQ RBF method

We consider the Crank-Nicolson implicit scheme of (14):

$$\frac{\Phi^{n+1} - \Phi^n}{\triangle t} + \frac{1}{2} \left(\nu_1 \frac{\partial \Phi^{n+1}}{\partial x} + \nu_2 \frac{\partial \Phi^{n+1}}{\partial y} + \nu_1 \frac{\partial \Phi^n}{\partial x} + \nu_2 \frac{\partial \Phi^n}{\partial y} \right) = 0, \tag{33}$$

where $t_{n+1} = t_n + \triangle t$, Φ^{n+1} and Φ^n are the level set variable at time t_{n+1} and t_n.

The approximate solution is expressed as:

$$\Phi(\mathbf{x}, t_{n+1}) = \sum_{j=1}^{N} \alpha_j^{n+1} \varphi_j(\mathbf{x}), \tag{34}$$

where α_j^{n+1}, $j = 1, ..., N$, are the unknown coefficients to be determined and $\varphi_j(\mathbf{x}) = \sqrt{(x - x_j) + (y - y_j) + c^2}$ are Hardy's multiquadrics functions.

By substituting equation (34) into (33), we obtain:

$$\sum_{j=1}^{N} \left(\frac{\varphi_j}{\triangle t} + \frac{1}{2} \left(\nu_1 \frac{\partial \varphi_j(\mathbf{x}_i)}{\partial x} + \nu_2 \frac{\partial \varphi_j(\mathbf{x}_i)}{\partial y} \right) \right) \alpha_j^{n+1} = \frac{\Phi^n(\mathbf{x}_j)}{\triangle t}$$

$$- \frac{1}{2} \left(\nu_1 \frac{\partial \Phi^n(\mathbf{x}_j)}{\partial x} + \nu_2 \frac{\partial \Phi^n(\mathbf{x}_j)}{\partial y} \right) \quad i = 1, ..., N. \tag{35}$$

5. Solution procedure with the Adaptive Algorithm

In the case of overdetermined system, the solution of the system can be obtained by the least-squares residual errors. MATLAB has several tools needed for computing a solution of the system of linear equations. In our case it is used the aid of the backslash operator \, which is the least-squares solution.

Otherwise, the usual method of solving PDEs with RBFs is similar to standard mesh based methods by constructing a uniform grid that is consecutively refined, yielding progressively more ill-conditioned systems of equations. Some authors who are accustomed to finite difference and finite element methods have claimed that RBFs could not be extended to complicated, multi-dimensional PDE problems in which many thousands of data centers are required. Fasshauer [26] found that adaptively refining the data centers in high gradient regions is more efficient than uniform h (grid density)–refinement, producing better conditioned systems. One possible way to reduce the ill-conditioning problems is to use the adaptive greedy algorithm.

It is well-known that the stability of the RBF method heavily depends on the *good* selection of its centers that is implicitly associated with the shape parameter. Finding optimal RBF centers is a difficult problem, but the problem is partially answered by a series of sub-optimal adaptive greedy algorithms [27, 28]. In this work, we couple the adaptive algorithm in [27] for selecting RBF centers. The idea is to specify the $N \times N$ matrix system by specifying the N RBF centers and N collocation points. Next, the adaptive algorithm is employed to select a proper subset of N_p RBF centers. The final approximation is given by the least-squares solution corresponding to the selected N_p RBF centers and N collocation points.

Considering the matrix system $\mathbf{H}\alpha = \mathbf{f}$, the adaptive algorithm runs iteratively by selecting collocation and RBF centers (row and column, respectively) one-by-one. The selected collocation points will be ignored as all N collocations points are used in the final least-squares process. For $k = 1, 2, \ldots$, the k-th stage solution α_k is one that only has k non-zero entries. A *new* collocation points is selected by locating the maximum-magnitude entries in the residual vector $\mathbf{r}_k = |\mathbf{H}\alpha_k - \mathbf{f}|$. To complete the row-column pair selection, the corresponding columns is selected based on the magnitude of the *dual residual*; interested reader are referred to the original article for implementation details. The selected pair is added to the previously selected pairs and the new approximation α_{k+1} is obtained by solving the sub-matrix

Figure 1: Solid body circulation: Shapes of the bubble at time $0, 0.5, 1, \ldots, 2$ obtained by 25×25 RBF centers.

Figure 2: Oriented flow: Shapes of the circular interface at time $0, 0.5, 1, \ldots, 2$ obtained by 25×25 RBF centers. (with or without the adaptive algorithm)

system associated with $k + 1$ collocation points and $k + 1$ RBF centers.

Before ending this section, we want to emphasize a few key features of the adaptive algorithm. The computational complexity is less than $O(N^3)$; the bound is tight only when we iterate all N steps. Secondly, the algorithm is matrix-free [29, 30] meaning that the $N \times N$ coefficient matrix needs not to be fully computed and stored. Hence, solving large-scale nonsparse RBF systems become possible [27]; also see [31] for more 3D computations.

6. Numerical results

6.1. Solid body circulation

To show good capability of the presented method, our first example consider the solid body circulation of a circular bubble (figure 1) of radius $r = 0.15$ centered at $(-0.1, 0.3)$ in a vortex flow with velocity field $(u, v) = (y, -x)$. The LMS and RBFs are also used to capture the moving interface with time, up to $T = 2$, in computational grid 25×25. The function Φ is then evaluated on a 49×49 grid in order to find zero contours. We choose a relative large shape parameter $c = 2$ to ensure smoothness of the bubbles over time. The initial time step is 0.01. Figure 1 shows the zero contours of the level set functions at different time during the rotation of the circle. A half cycle of circulation is performed.

6.2. Oriented flow

The next example is translation of circular interface in oriented flow (figure 2). The circular interface of radius $r = 0.15$, initially centered at $(0.3, 0.3)$ moving by the oriented flow in a cavity of size 1×1 with the velocity field (u, v) defined as follows:

$$u = 0.2(x + 0.5) \tag{36}$$
$$v = 0.2(y + 0.5). \tag{37}$$

12

Figure 3: Shear flow: Shapes of the bubble at time $0, 0.5, 1, \ldots, 4$ obtained by 25×25 RBF centers.

Figure 4: Shear flow: Shapes of the bubble at time $0, 0.5, 1, \ldots, 4$ obtained by 35×35 RBF centers.

For capturing moving circular interface in figure 2 LMS and RBFs are used. The calculation is performed with time, up to $T = 2$, in computational grid 25×25. The function Φ is evaluated on a 49×49 grid in order to find zero contours. We choose a relative large shape parameter $c = 2$ to ensure smoothness of the bubbles over time and the initial time step is 0.01. Only one figure is presented since equal results are obtained with or without the adaptive algorithm.

6.3. Shear flow

The following example was first presented in chapter XI of the book Advances in Meshless Methods [32]. We consider a circular bubble of radius $r = 0.15$, initially centered at $(0.5, 0.7)$ moving by a shear flow in a cavity of size 1×1 with the velocity field (u, v) defined as follows:

$$u = -\sin(\pi x)\cos(\pi y), \tag{38}$$

$$v = \cos(\pi x)\sin(\pi y). \tag{39}$$

In such a velocity field, the bubble is passively transported in the form of circulation and stretching. The LMS and RBFs are used to capture the moving interface with time, up to $T = 4$ with a time step $\triangle t = 0.01$, in computational grids 25×25 and 35×35. The function Φ is then evaluated on a 49×49 grid in order to find zero contours. We again choose the large shape parameter $c = 2$ to demonstrate stability over ill-conditioned linear systems. In Figure 3 and Figure 4, the shapes of the bubble at every $\triangle t = 0.5$ are shown. The shapes at $T = 4$ are highlighted with a thick solid boundary. The goal here is to demonstrate that by coupling the RBF method with adaptive technique the level set algorithm becomes stable even without reinitialization. In Figure 3, we can see that with or without the adaptive algorithm, the bubbles are more or less at the same location. However, the bubble obtained

13

Figure 5: Shear flow: Shapes of the bubble calculated with Crank-Nicolson implicit scheme, shape parameter is equal 2.

Figure 6: Shear flow: Shapes of the bubble calculated with the Crank-Nicolson implicit scheme, shape parameter is equal 3.

with the adaptive algorithm is more skewed; without adaptive algorithm, a more circular bubble is obtained. In both cases, we can see that some artifacts appear in the corners of the computational domain.

A good algorithm should allow users to refine the computational grids in a order to obtain better approximations. In Figure 4, the computational results on a 35×35 grid with and without the adaptive techniques are shown. Since the shape parameter is unchanged, it is well known that using more RBF centers will give rise to matrix systems that are more ill-conditioned compared to those in the 25×25 case. Without the adaptive algorithm, computational results with direct solver quickly fail; the zero contours after $T = 0.5$ are empty. On the other hand, the bubble obtained with the adaptive algorithm is, first, located at the same location and secondly more elongated. However, small artifact still presents near lower-left corner of the domain and reinitialization will be unavoidable if we want to carry on the computation to longer time.

Numerical examples presented till now were computed with the classical time marching schemes, but examples presented in Figure 5 and Figure 6 were computed only with the Crank-Nicolson implicit scheme.

The artifacts are also presented in Figure 5 although the shapes of the bubble were computed with the Crank-Nicolson implicit scheme which is more stable implicit scheme. In Figure 6, none artifacts are presented because it was used the shape parameter with the value 3.

Figure 7: Single-vortex flow: Shapes of the circular interface at time $T = 1$ obtained by 25×25 RBF centers with different shape parameters c and time stepping dt.

14

6.4. Single-vortex flow

The last example is circular interface in a single-vortex field which was first presented in [33]. The circular interface of radius $r = 0.15$, initially centered at $(0.5, 0.7)$ moving in a steady, non-uniform vorticity field in a cavity of size 1×1 with the velocity field (u, v) defined as follows:

$$u = -\sin^2(\pi x)\sin(2\pi y) \tag{40}$$

$$v = \sin^2(\pi x)\sin(2\pi x). \tag{41}$$

The LMS and RBFs are used to capture the moving interface with time, up to $T = 1$, in computational grid 25×25. In this example, we employ different shape parameters $c = 0.1, 0.5, 1.0$ and different time steps $dt = 0.005, 0.010, 0.020$. Obviously, smaller dt allows more stretching; at the same time, smaller c will have a similar effect. For example, for parameter-pair $[c, dt] = [0.5, 0.005]$ and $[0.5, 0.010]$ has more stretching than $[0.1, 0.020]$. Whereas $[0.1, 0.020]$ and $[0.5, 0.020]$ result in almost the same stretching. (see fig. 7)

7. Conclusions

This paper outlines our work done on an alternative approach to the conventional level set methods for solving two-dimensional moving-boundary problems. This approach is set up from MQ RBFs and the adaptive greedy algorithm. Our numerical examples suggest that the solution is more stable by employing the adaptive algorithm. The exact time integration scheme which is valid in an appropriate moving node frame is also presented, but it is not valid for most Eulerian formulations.

Four examples are presented: solid body circulation, oriented flow, shear flow (bubble starts circular but because of the velocity fields the bubble stretchs) and vortex field. The first two examples which include adaptive greedy algorithm or not, show almost the same results. The solution looks quite nice and it is a good start for demonstrating the usefulness of the proposed method. The third example shows the stability of the adaptive greedy algorithm (time period was $T = 4$). The last example is given to show that method is not sensitive to shape parameter c or that different c and time step dt will give similar solutions. In the case of small shape parameter c the results seem to be consistent. Large shape parameter c yields obviously different solution.

In the conventional level set methods, the level set equation is solved to evolve the interface using a capturing Eulerian approach. The solving procedure requires appropriate choice of the upwind schemes, reinitialization algorithms and extension velocity methods, which may require excessive amount of computational efforts. In our case we do not choose the reinitialization, because we try to control the smoothness of the moving boundary with appropriate choice of the type of MQ RBFs, shape parameters, time step, time marching schemes and greedy algorithm. The proposed alternative approach offers to use smaller computational grids, with no reinitialization in order to beat the upwind scheme. Therefore we can conclude that presented procedure gives enough means which offer much less computational work compared to classical reinitialization and also gives stable results.

Regarding to the time marching schemes it was found that the second order Crank-Nicolson implicit scheme does not improve the efficiency compared to the first order type schemes.

We are not claiming that reinitialization can be avoided by the proposed approach. Instead, the proposed approach is more stable therefore reinitialization can be postponed. In other words, with regular reinitialization, we

are expecting more accurate results if the RBF method is implemented with the adaptive algorithm than without it.

In the future work we can treat the discontinuity as a true discontinuity, subject to the normal Rankine-Hugoniot jump conditions or to use moving technique to locally represent the level set function.

8. Acknowledgement

References

[1] Šarler B. Stefan's work on solid–liquid phase changes. Engineering Analysis with Boundary Elements 1995; 16:83–92.

[2] Crank J. Free and moving boundary problems. Oxford Science Publications. Oxford: Clarendon Press. X, 425 p.; 1984.

[3] Javierre E, Vuik C, Vermolen FJ, van der Zwaag S. A comparison of numerical models for one-dimensional Stefan problems. J Comput Appl Math 2006; 192(2):445–59.

[4] Hill JM. One-dimensional Stefan problems: an introduction. Pitman Monographs and Surveys in Pure and Applied Mathematics, 31. Harlow, Essex: Longman Scientific and Technical; New York: John Wiley & Sons, Inc.; XVIII, 204 p.; 1987.

[5] Kansa EJ. Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics. I: Surface approximations and partial derivative estimates. Comput Math Appl 1990; 19(8-9):127–45.

[6] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces. Applied Mathematical Sciences. 153. New York, NY: Springer. xiv, 273 p.; 2003.

[7] Sethian JA. Level set methods and fast marching methods. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge Monographs on Applied and Computational Mathematics. 3. Cambridge: Cambridge University Press. xx, 378 p.; 1999.

[8] Cecil T, Qian JL, Osher S. Numerical methods for high dimensional hamilton-jacobi equations using radial basis functions. Journal of Computational Physics 2004; 196:327–47.

[9] Wang S, Wang MY. Radial basis functions and level set method for structural topology optimization. International Journal for numerical methods in Engineering 2006; 65:2060–90.

[10] Wang SY, Lim KM, Khoo BC, Wang MY. An extended level set method for shape and topology optimization. Journal of Computational Physics 2006; 221:395–421.

[11] Atluri SN, Shen S. The meshless local Petrov-Galerkin (MLPG) method. Encino, CA: Tech Science Press. viii, 429 p.; 2002.

[12] Gang Li, Aluru NR. Boundary cloud method: a combined scattered point/boundary integral approach for boundary-only analysis. Comput Methods Appl Mech Eng 2002; 191(21-22):2337–70.

[13] Kansa EJ. Multiquadrics – a scattered data approximation scheme with applications to computational fluid-dynamics. II: Solutions to parabolic, hyperbolic and elliptic partial differential equations. Comput Math Appl 1990; 19(8-9):147–61.

[14] Hardy RL. Multiquadric equations of topography and other irregular surfaces. J Geophys Res 1971; 176:1905–15.

[15] Ling L, Kansa EJ. A least-squares preconditioner for radial basis functions collocation methods. Adv Comput Math 2005; 23(1-2):31–54.

[16] Ling L, Kansa EJ. Preconditioning for radial basis functions with domain decomposition methods. Math Comput Modelling 2004; 40(13):1413–27.

[17] Ling L, Hon YC. Improved numerical solver for Kansa's method based on affine space decomposition. Eng Anal Bound Elem 2005; 29(12):1077–85.

[18] Fornberg B, Wright G. Stable computation of multiquadric interpolants for all values of the shape parameter. Comput Math Appl 2004; 48(5-6):853–67.

[19] Kansa EJ, Aldredge RC, Ling L. Numerical simulation of two-dimensional combustion using mesh-free methods. to appear in Eng Anal Bound Elem; 2009.

[20] Iske A. Characterization of conditionally positive definite functions for multivariable interpolation methods with radial basis functions. (Charakterisierung bedingt positiv definiter Funktionen für multivariate Interpolationsmethoden mit radialen Basisfunktionen.). PhD thesis, Göttingen: Math.-Naturwiss. FB, Univ. Göttingen, 56 p.; 1994.

[21] Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: Algorithms base on hamilton-jacobi formulations. J Computational Physics 1988; 79:12–49.

[22] Tsai R, Osher S. Level set methods and their applications in image science. Commun Math Sci 2003; 1(4):623–56.

[23] Vrankar L, Turk G, Runovc F. A comparison of the effectiveness of using the meshless method and the finite difference method in geostatistical analysis of transport modeling. Int J Comput Methods 2005; 2(2):149–66.

[24] Wang SY, Lim KM, Khoo BC, Wang MY. An extended level set method for shape and topology optimization. J Comput Phys 2007; 221(1):395–421.

[25] Greenberg MD. Advanced engineering mathematics. Englewood Cliffs, NJ: Prentice-Hall. xii, 946 p.; 1988.

[26] Fasshauer GE. Meshfree approximation methods with Matlab. With CD-ROM. Interdisciplinary Mathematical Sciences 6. Hackensack, NJ: World Scientific. xviii, 500 p.; 2007.

[27] Ling L, Schaback R. An improved subspace selection algorithm for meshless collocation methods. In review; 2008.

[28] Ling L, Schaback R. Stable and convergent unsymmetric meshless collocation methods. SIAM J Numer Anal 2008; 46(3):1097–1115.

[29] Lee CF, Ling L, Schaback R. On convergent numerical algorithms for unsymmetric collocation. Adv Comput Math 2009; 30(4):339–54.

[30] Ling L, Opfer R, Schaback R. Results on meshless collocation techniques. Eng Anal Bound Elem 2006; 30(4):247–53.

[31] Kwok TO, Ling L. On convergence of least-squares based Kansa's method for helmholtz equations. In review; 2008.

[32] Mai-Duy N, Mai-Cao L, Tran-Cong T. A new meshless RBF-based method for unsteady fluid flow analysis. In Sladek J, Sladek V (eds). Advances in meshless methods. Tech Science Press. xiii, 363 p.; 2006.

[33] Salih A, Ghosh Moulic S. Some numerical studies of interface advection properties of level set method. Sādhanā 2009; 34(2):271–98.
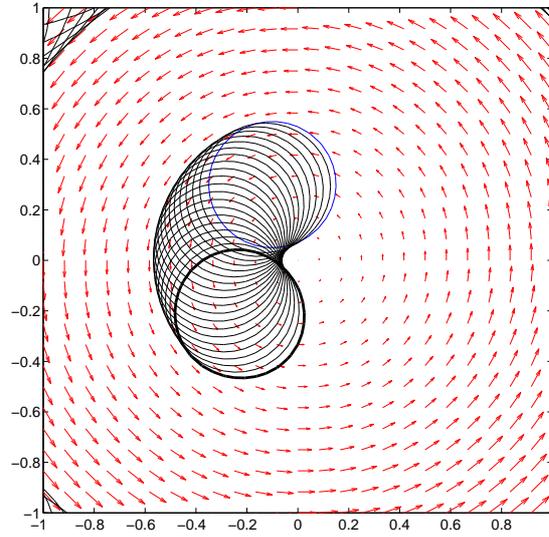
without the adaptive algorithm
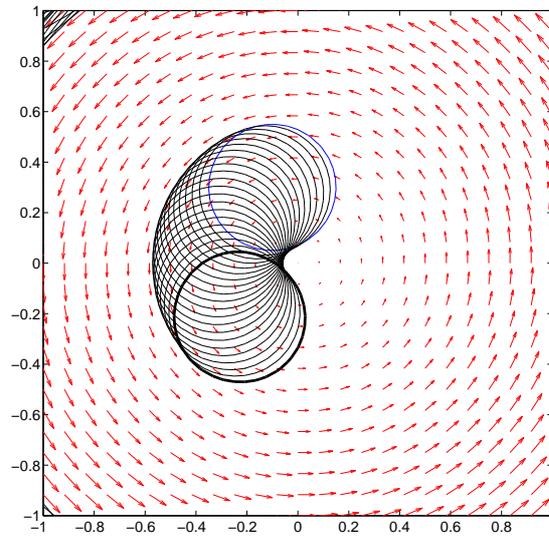
with the adaptive algorithm

Figure 1: Solid body circulation: Shapes of the bubble at time 0, 0.5, 1, ..., 2 obtained by 25×25 RBF centers.
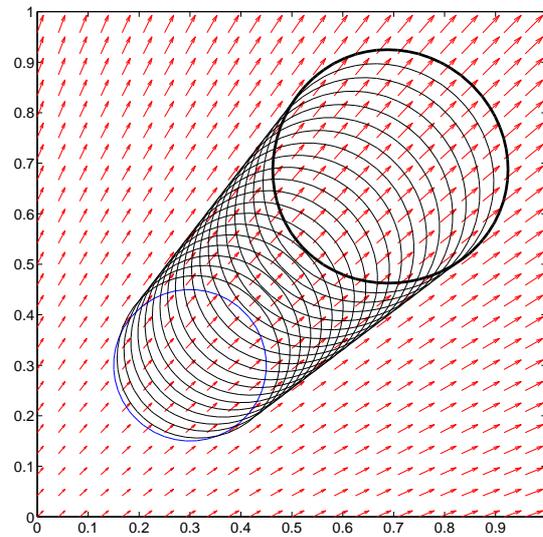
Figure 2: Oriented flow: Shapes of the circular interface at time $0, 0.5, 1, \ldots, 2$ obtained by 25×25 RBF centers. (with or without the adaptive algorithm)
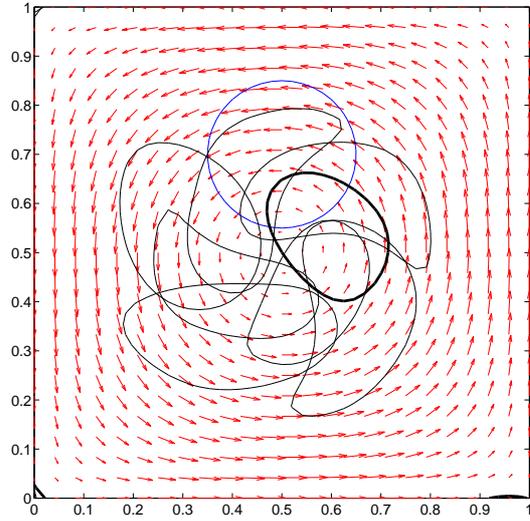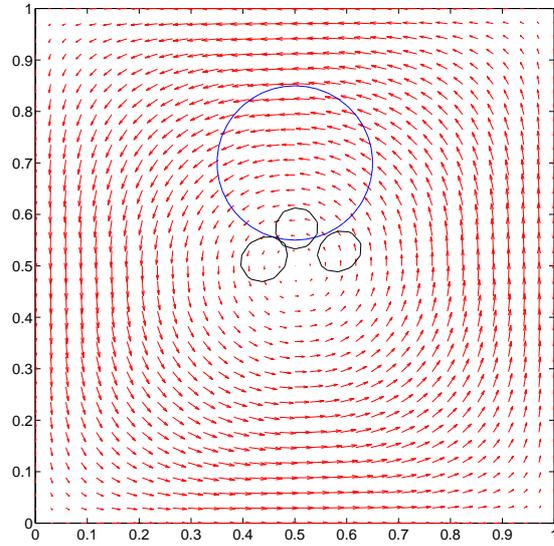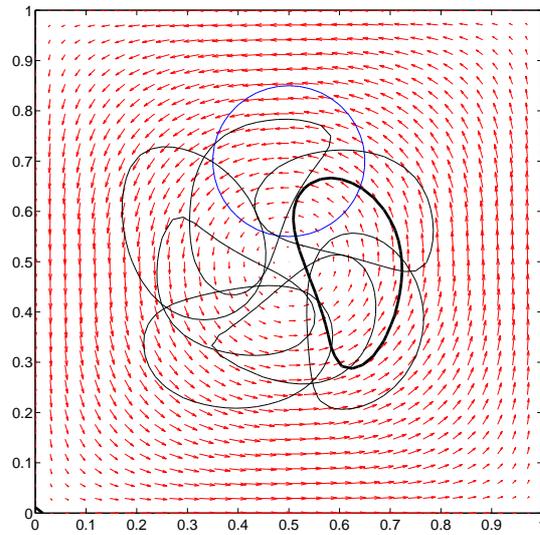
without the adaptive algorithm
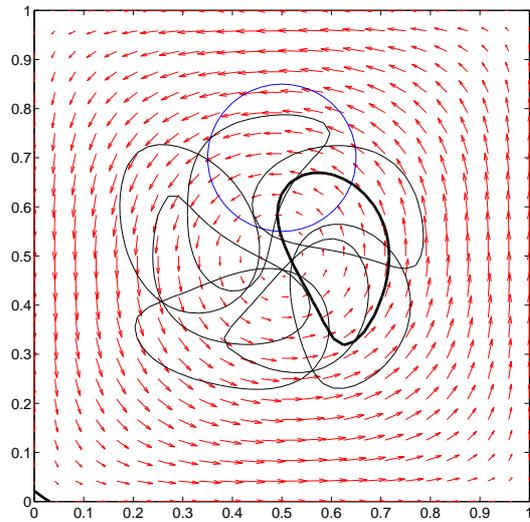
with the adaptive algorithm

Figure 3: Shear flow: Shapes of the bubble at time $0, 0.5, 1, \ldots, 4$ obtained by 25×25 RBF centers.
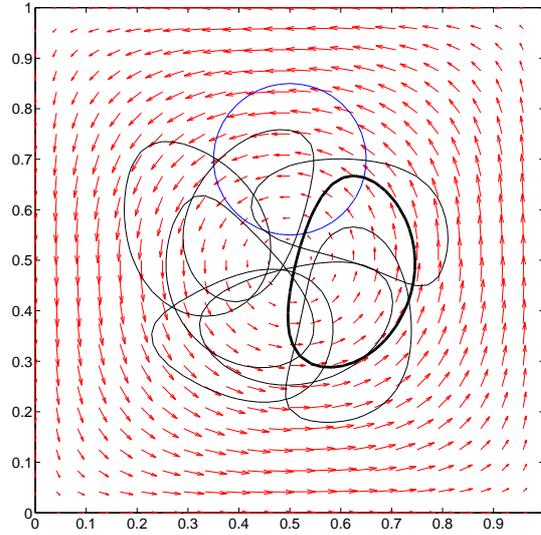
without the adaptive algorithm

with the adaptive algorithm

Figure 4: Shear flow: Shapes of the bubble at time $0, 0.5, 1, \ldots, 4$ obtained by 35×35 RBF centers.
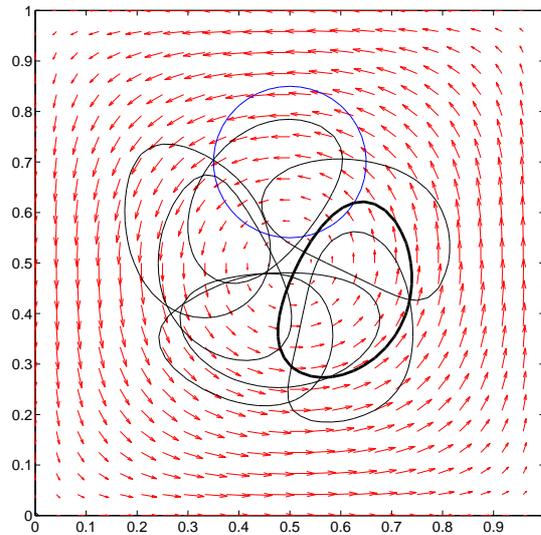
4
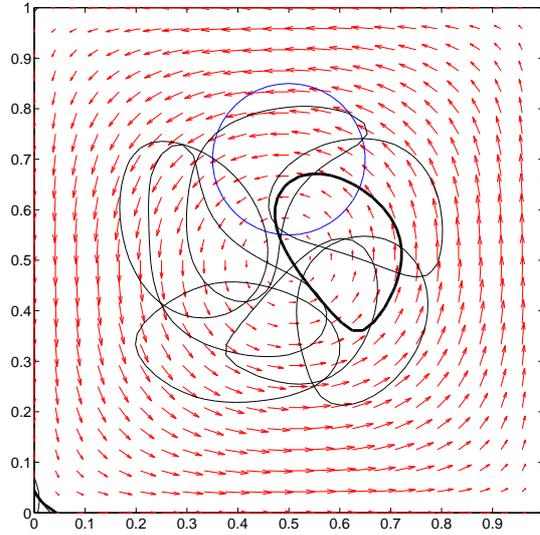
without the adaptive algorithm
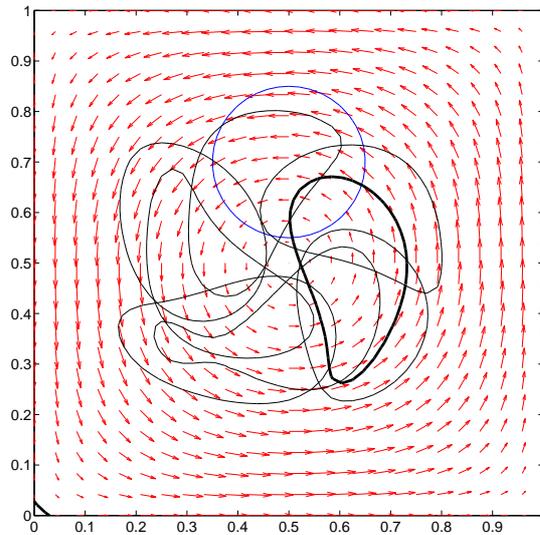
with the adaptive algorithm

Figure 5: Shear flow: Shapes of the bubble calculated with Crank-Nicolson implicit scheme, shape parameter is equal 2.

without the adaptive algorithm

with the adaptive algorithm

Figure 6: Shear flow: Shapes of the bubble calculated with the Crank-Nicolson implicit scheme, shape parameter is equal 3.
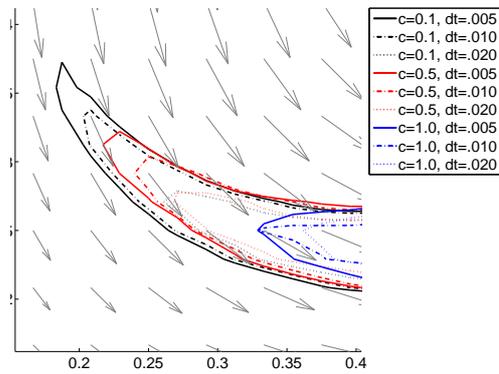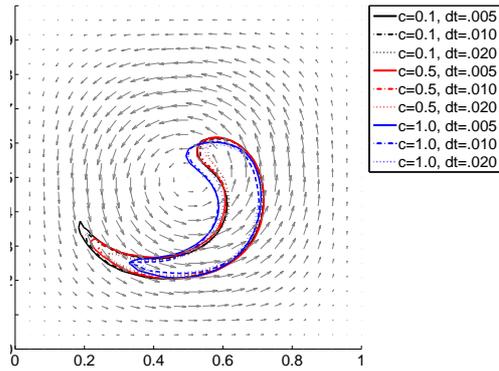
Figure 7: Single-vortex flow: Shapes of the circular interface at time $T = 1$ obtained by 25×25 RBF centers with different shape parameters c and time stepping dt.