

## FAST NUMERICAL METHODS FOR BERNOULLI FREE BOUNDARY PROBLEMS\*

CHRISTOPHER M. KUSTER<sup>†</sup>, PIERRE A. GREMAUD<sup>†</sup>, AND RACHID TOUZANI<sup>‡</sup>

**Abstract.** The numerical solution of the free boundary Bernoulli problem is addressed. An iterative method based on a level-set formulation and boundary element method is proposed. Issues related to the implementation, the accuracy, and the generality of the method are discussed. The efficiency of the approach is illustrated by numerical results.

**Key words.** Bernoulli, free boundary, level set, boundary elements

**AMS subject classifications.** 65N38, 35R35, 76B99

**DOI.** 10.1137/06065444X

**1. Introduction.** Bernoulli free boundary problems find their origin in the description of free surfaces for ideal fluids [12]. There are, however, numerous other applications leading to similar formulations; see, for instance, [11]. For concreteness, we focus on the exterior Bernoulli problem. Let  $\Omega$  be a bounded domain in  $\mathbb{R}^2$ . The exterior Bernoulli problem consists in seeking a bounded domain  $A \supset \Omega$  and a function  $u$  defined on  $\bar{A} \setminus \Omega$  such that

$$(1.1) \quad \Delta u = 0 \quad \text{in } A \setminus \bar{\Omega},$$

$$(1.2) \quad u = 1 \quad \text{on } \partial\Omega,$$

$$(1.3) \quad u = 0 \quad \text{on } \partial A,$$

$$(1.4) \quad \frac{\partial u}{\partial n} = \mu \quad \text{on } \partial A,$$

where  $\mu$  is given. In the previous example, one can think of  $u$  as a streamfunction and of  $\Omega$  as an obstacle. Taking into account (1.3), condition (1.4) can be written as  $|\nabla u| = |\mu|$  and corresponds, for fluid applications, to Bernoulli's principle; see, for instance, [9].

The above problem has been extensively studied; see [11] for general remarks. For a convex simply connected bounded domain  $\Omega$ , it is known that for any negative constant  $\mu < 0$ , the above problem admits a unique classical solution. Further, the free boundary  $\partial A$  has regularity  $\mathcal{C}^{2,\alpha}$ ; see [21, Theorem 1.1].<sup>1</sup> The convexity assumption is necessary for uniqueness, as counterexamples show (see [11, Example 13]). The study of the interior Bernoulli problem is more delicate, and not even convexity can ensure uniqueness.

---

\*Received by the editors March 16, 2006; accepted for publication (in revised form) October 26, 2006; published electronically March 30, 2007.

<http://www.siam.org/journals/sisc/29-2/65444.html>

<sup>†</sup>Department of Mathematics and Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC 27695-8205 (cmkuster@ncsu.edu, gremaud@ncsu.edu). The work of the first author was partially supported by the National Science Foundation (NSF) through grant DMS-0244488. The work of the second author was partially supported by the NSF through grants DMS-0204578, DMS-0244488, and DMS-0410561.

<sup>‡</sup>Laboratoire de Mathématiques, CNRS UMR 6620, Université Blaise Pascal (Clermont-Ferrand), 63177 Aubière Cedex, France (Rachid.Touzani@math.univ-bpclermont.fr).

<sup>1</sup>The result is, in fact, established in the more general case of the  $p$ -Laplacian in [21] and is generalized to nonconstant  $\mu$  in [22].

There are roughly two ways of tackling such problems numerically. First, a variational formulation may be considered and the corresponding cost function minimized [19, 23, 28]; this requires the calculations of shape gradients. Second, a fixed point-type approach can be set up where a sequence of elliptic problems is solved in a sequence of converging domains, those domains being obtained through some updating rule at each iteration [6, 11, 24]. The method studied in this paper falls in the latter category.

Two simple generic numerical tools are combined. First, the interface is represented through a level-set formulation as described in section 2. Second, the elliptic part of the problem is solved through the use of a boundary element method; see section 3 (see [14] for another example of a method involving those two tools in a different setting). The proposed method is thus conceptually simpler than the shape optimization approach. Since it requires, in principle, only the calculations of quantities being defined on the interface or close to it, it is potentially faster than the finite difference approaches of, for instance, [6, 11, 24] or more generally of immersed interface-type methods [27]. The feasibility of the algorithm and its complexity are investigated in section 4.

A general overview of the strategy is as follows: the potential problem is solved with one of the conditions on the free boundary omitted, and then the omitted condition is used to update the location of the free boundary. More specifically, given an initial domain  $A_0 \supset \bar{\Omega}$ , the simplest variant of this type consists in solving the sequence of problems

$$(1.5) \quad \Delta u_k = 0 \quad \text{in } A_k \setminus \bar{\Omega}, \quad k = 0, 1, 2, \dots,$$

$$(1.6) \quad u_k = 1 \quad \text{on } \partial\Omega,$$

$$(1.7) \quad \partial_n u_k = \mu \quad \text{on } \partial A_k.$$

For a given domain  $A_k$ , it is well known that problem (1.5)–(1.7) admits a unique solution; see, for instance, [15, Theorem 5.1]. The new domain  $A_{k+1}$  is found by moving  $\partial A_k$  in its normal direction so that  $u_k$  vanishes there. Let  $P_k \in \partial A_k$ ; to first order, we have

$$u_k(P_{k+1}) \approx u_k(P_k) + \mu d_k,$$

where  $P_{k+1} = P_k + n_k d_k$ ,  $n_k$  being the outer unit normal to  $\partial A_k$  at  $P_k$ . The new point  $P_{k+1}$ , or similarly the distance  $d_k$ , is determined by the requirement  $u_k(P_{k+1}) = 0$ , i.e.,  $d_k = \frac{-u_k(P_k)}{\mu}$ . The free boundary is thus updated according to<sup>2</sup>

$$(1.8) \quad \partial A_{k+1} = \partial A_k - \frac{u_k}{\mu} n_k.$$

**2. Level-set representation of the interface.** In the above iterative process, the interface  $\partial A_k$  has to be updated as long as the residual  $u_k|_{\partial A_k}$  is not (numerically) zero. Relation (1.8) indicates that the boundary should be moved in the normal direction by an amount proportional to the residual.

<sup>2</sup>The present iterative process can be modified by imposing the Neumann condition (1.7) on the “next” boundary  $\partial A_{k+1}$  instead of  $\partial A_k$  [11, 13]. However, in the formulation adopted here, those modifications are of little use, as they require information such as curvature which complicates the calculation of the “evolution” of the interface.

If the residual is considered as a normal speed, a time-dependent problem can be set up for the “evolution” (or correction) of the domain  $A_k$ . We denote the domain so generated  $A_k(t)$  with  $A_k(0) = A_k$ . Let  $F : \mathbb{R}^2 \rightarrow \mathbb{R}$  be an extension of the residual away from  $A_k$  (see subsection 2.3 below). We want the interface  $\Gamma(t) = \partial A_k(t)$  to be characterized by

$$\Gamma(0) = \partial A_k, \quad \Gamma(t) = \{(x(t), t); x(0) = x_0, x_0 \in \partial A_k\} \quad \text{for } t > 0,$$

where

$$\frac{dx}{dt} = Fn, \quad x(0) = x_0,$$

$n$  being the unit outer normal to  $\partial A_k(t)$ . A level-set approach, as pioneered in [30] (see also [29, 34]), consists in representing the interface  $\{\Gamma(t)\}_{t \geq 0}$  as the zero level set of family of level-set functions  $\{\phi(\cdot, t)\}_{t \geq 0}$  with the property

$$A_k(t) = \{x \in \mathbb{R}^2; \phi(x, t) < 0\}, \quad \Gamma(t) = \partial A_k(t) = \{x \in \mathbb{R}^2; \phi(x, t) = 0\},$$

where  $A_k(t)$  denotes the domain stemming from the evolution of  $A_k$  through the above process. By taking the time derivative of the relation  $\phi(x(t), t) = 0$ , the level-set equation is obtained:

$$(2.1) \quad \partial_t \phi + F|\nabla \phi| = 0,$$

$$(2.2) \quad \phi(\cdot, 0) = \phi_0,$$

where  $\phi_0$  is a level-set function corresponding to  $\partial A_k$ .

Several points related to the implementation of the above method now have to be considered. The fixed boundary  $\partial \Omega$  is approximated by a piecewise linear curve  $\partial \Omega_h$  with  $N$  elements. The size of the smallest element of  $\partial \Omega_h$  is denoted  $\Delta x$ . Let  $B \subset \mathbb{R}^2$  be a square domain of size  $M\Delta x \times M\Delta x$ , where  $M$  is chosen large enough so that  $B$  contains  $A$ . We associate with  $B$ , in a natural way, a uniform Cartesian mesh  $B_h$  of size  $\Delta x$ . In the following, the level-set functions are characterized by their nodal values on the mesh  $B_h$ . Two kinds of interpolation operators are considered on  $B_h$ . In the contouring step (subsection 2.1), a classical  $\mathcal{P}_1$  interpolation is used. In the projection step (subsection 2.2), a classical local  $\mathcal{Q}_2$  interpolation is considered.

**2.1. Contouring.** In many applications of the level-set method, the actual reconstruction of the interfaces is not needed. This is not the case here, as the elliptic problem (1.5)–(1.7) has to be solved in a family of successive domains defined by those interfaces. Let  $\phi$  be a given level-set function and  $I_{\mathcal{P}}\phi$  its  $\mathcal{P}_1$ -interpolant on  $B_h$ . By contouring, we mean the operation that associates with the nodal values of  $\phi$  the zero level set of  $I_{\mathcal{P}}\phi$ . More precisely, each square cell of the underlying Cartesian grid is divided into two triangular elements,<sup>3</sup> and on each of those triangles, the unique polynomial of degree 1 agreeing with the values of the level-set function at the vertices is considered. Contour lines are then constructed by connecting the zeros of  $I_{\mathcal{P}}\phi$  on the grid lines. This process is facilitated by two facts: first, each triangle contains at most one contour line passing through it, and, second, the gradient of  $I_{\mathcal{P}}\phi$  is piecewise constant on the grid elements, yielding an outward normal to  $A_k(t)$ . Note that both properties fail in general for the local  $\mathcal{Q}_2$  interpolant considered in the projection step below.

If  $I_{\mathcal{P}}\phi$  is uniformly equal to zero on a given triangle, then the problem is under-resolved ( $\Delta x$  is too large) and the algorithm fails.

---

<sup>3</sup>Any such decomposition is acceptable.

**2.2. Projection.** As mentioned above, some quantities such as the normal speed need to be extended away from the interface. The first step in this process consists again in an accurate reconstruction of the boundary. A loop through the mesh is done to determine which nodes are “close to” the interface. Here a node is close to the interface if it has a primary neighbor where  $\phi$  has opposite sign.

With each node  $x_i$ , we associate its eight closest neighbors and construct on this set of nine nodes the unique polynomial of degree 2 in each variable agreeing with the values of  $\phi$  there. This local  $\mathcal{Q}_2$  interpolant is denoted  $I_{\mathcal{Q},x_i}\phi$ . For each node  $x_i$  close to the interface, the closest point  $x^*$  to  $x_i$  on the zero level set of  $I_{\mathcal{Q},x_i}\phi$  is computed. The square of the Cartesian distance, i.e.,  $|x - x_i|^2$ , is minimized subject to the constraint  $I_{\mathcal{Q},x_i}\phi = 0$ . The Lagrangian for this problem is

$$(2.3) \quad \mathcal{L}(x, \lambda) = |x - x_i|^2 + \lambda I_{\mathcal{Q},x_i}\phi(x).$$

To find the point  $x^*$ , the system

$$(2.4) \quad \nabla_{x,\lambda}\mathcal{L}(x, \lambda) = 0$$

is solved by Newton’s method with Armijo line search [25]. This projection method was introduced and discussed in [17]. It roughly corresponds to a simpler version of the method proposed in [8] which is based on cubic splines; see [26] for more comments and comparisons. The above algorithm can fail if the interface is underresolved.

The projection step could potentially be used to reconstruct the interface, i.e., as another contouring step. However, while it is locally more accurate than the above contouring algorithm (and is thus ideal when used in conjunction with the extension step described below), it has several disadvantages as a contouring tool (local character, nonconstant elementwise gradient).

**2.3. Extension.** This step extends the speed  $F$  away from the interface, so that (2.1), (2.2) can be solved. Let  $\Gamma$  be the interface obtained from the contouring step. By construction,  $\Gamma$  is a closed (for the problems considered here), possibly multiconnected, piecewise linear (on the triangular mesh derived from  $B_h$ ) curve in  $B$ . The boundary nodes corresponding to  $\Gamma$  are denoted  $\{\xi_j\}$ ; i.e., the  $\xi_j$ ’s are the end points of the line segments that form  $\Gamma$ . Further, through the elliptic step (see section 3), the value of  $F$  at the center point of each linear segment of  $\Gamma$  is known; the center point nodes are denoted  $\{\bar{\xi}_j\}$ .

First, the values of  $F$  at the midpoint boundary nodes  $\{\bar{\xi}_j\}$  are extended to the set of Cartesian nodes  $\{x_i\}$  that form the vertices of the triangles containing the nodes  $\{\bar{\xi}_j\}$ .

More precisely, consider the node  $\bar{\xi}_j$  in Figure 1, left. This local extension step is based on the analysis of the “domain of influence” of the nodes  $\{\bar{\xi}_j\}$ . This domain of influence is taken here as the set of all the points in  $B$  whose orthogonal projection on the line containing the segment through  $\bar{\xi}_j$  belongs to that segment. If a node  $x_i$  belongs to the domain of influence of  $\bar{\xi}_j$ , then  $F$  is extended at  $x_i$  by the value  $F(\bar{\xi}_j)$  as in Figure 1, left. If instead the node  $x_i$  is in the domain of influence of more than one midpoint boundary node, as would be the case in the dark grey area in Figure 1, left, then the value at the midpoint of the closest segment to  $x_i$  is retained. Finally, if  $x_i$  does not belong to the domain of influence of any midpoint boundary node, as in the light grey area in Figure 1, right, then the value of  $F$  at  $x_i$  is taken as

$$F(x_j) = \frac{\beta}{\alpha + \beta} F_{j-1} + \frac{\alpha}{\alpha + \beta} F_j,$$

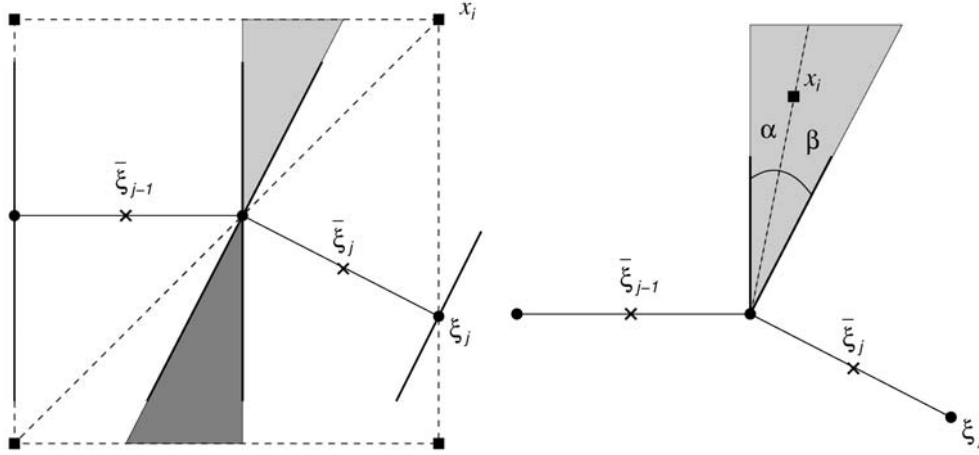


FIG. 1. Left: general view of the geometry involved in the local extension step and corresponding “domains of influence” (dark grey area: shock-like domain, light grey area: rarefaction-like domain); right: definition of the local angles  $\alpha$  and  $\beta$ .

where  $F_{j-1}$  and  $F_j$  are the values of  $F$  at the midpoint boundary nodes  $\bar{\xi}_{j-1}$  and  $\bar{\xi}_j$ , respectively, and where the angles  $\alpha$  and  $\beta$  are defined as in Figure 1, right. This way of defining the local extension of  $F$  is compatible with the global extension (2.7).

Second, a renormalized function  $\tilde{\phi}$  is initialized as a signed distance function at the same Cartesian nodes at which  $F$  has just been extended. The projection step is used to do this.

We emphasize that both of those local extension steps for  $F$  and  $\tilde{\phi}$  take place only on the nodes adjacent to the interface; the corresponding values are then used as starting points for the extension to the rest of the Cartesian nodes. This is accomplished using the fast marching method [2] to solve

$$(2.5) \quad |\nabla \tilde{\phi}| = 1 \quad \text{in } B,$$

$$(2.6) \quad \tilde{\phi} = 0 \quad \text{on } \Gamma,$$

$$(2.7) \quad \nabla \tilde{F} \cdot \nabla \tilde{\phi} = 0 \quad \text{in } B,$$

$$(2.8) \quad \tilde{F} = F \quad \text{on } \Gamma.$$

A fully upwind mixed first/second order discretization of the above equations is applied on the mesh  $B_h$ ; see [18, 35, 36] for more details.

**2.4. Updating the interface.** The interface is moved by updating the corresponding level-set function through (2.1), (2.2). More precisely, after the extension step, the level-set function  $\tilde{\phi}$  corresponding to the current interface  $\Gamma$  is a signed distance function, and, in particular,  $|\nabla \tilde{\phi}| = 1$ . Therefore, (2.1), (2.2) here reads

$$\partial_t \phi + \tilde{F} = 0,$$

$$\phi(\cdot, 0) = \tilde{\phi}.$$

The update is then trivially computed by taking one forward Euler step,

$$(2.9) \quad \phi_{\text{new}} = \tilde{\phi} - \Delta t \tilde{F},$$

where  $\Delta t = -\frac{1}{2\mu}$ . Note that this corresponds to half the optimal time step given by (1.8); taking the “optimal” value from (1.8) may lead to overshoots in the position of the interface and may, in fact, result in slowing down the convergence of the global iterative process.

**2.5. Initial interface.** An initial guess of the interface’s position,  $\partial A_0$ , needs to be provided. This can be done in an ad hoc way. In section 4,  $\partial A_0$  is taken as a curve of constant distance to  $\Omega$ .

**3. Boundary element method.** Consider again the problem (1.5)–(1.7). For the sake of simplicity, the subscript  $k$  is dropped in this section. We assume both  $\partial A$  and  $\partial\Omega$  to be simple closed curves and let  $\Gamma = \partial A \cup \partial\Omega$ . The region of interest  $A \setminus \bar{\Omega}$  being interior to  $\partial A$  and exterior to  $\partial\Omega$ ,  $\partial A$  is oriented counterclockwise, while  $\partial\Omega$  is clockwise.

Multiplying (1.5) by the fundamental solution  $G(x, y) := -\frac{1}{2\pi} \log |x - y|$  and integrating twice by parts leads to

$$(3.1) \quad u(x) = \int_{\Gamma} G(x, y) \frac{\partial u}{\partial n_y}(y) ds(y) - \int_{\Gamma} \frac{\partial G}{\partial n_y}(x, y) u(y) ds(y),$$

where  $n$  is the unit outer normal to  $A \setminus \bar{\Omega}$  at  $y$ . The above integral representation is valid for  $x \in A \setminus \bar{\Omega}$ . To treat the case  $x \in \Gamma$ , we define the linear operator  $\mathcal{L} : L^2(\Gamma) \rightarrow L^2(\Gamma)$  by

$$\mathcal{L}v(x) = \begin{cases} \frac{v(x)}{2} + \int_{\partial A} \frac{\partial G}{\partial n_y}(x, y) v(y) ds(y) - \int_{\partial\Omega} G(x, y) v(y) ds(y) & \text{for } x \in \partial A, \\ \int_{\partial A} \frac{\partial G}{\partial n_y}(x, y) v(y) ds(y) - \int_{\partial\Omega} G(x, y) v(y) ds(y) & \text{for } x \in \partial\Omega \end{cases}$$

and the function  $\mathcal{F} \in L^2(\Gamma)$  by

$$\mathcal{F}(x) = \begin{cases} \mu \int_{\partial A} G(x, y) ds(y) - \int_{\partial\Omega} \frac{\partial G}{\partial n_y}(x, y) ds(y) & \text{for } x \in \partial A, \\ \mu \int_{\partial A} G(x, y) ds(y) - \int_{\partial\Omega} \frac{\partial G}{\partial n_y}(x, y) ds(y) - \frac{1}{2} & \text{for } x \in \partial\Omega. \end{cases}$$

Taking into account the boundary conditions (1.6) and (1.7), it is then standard to check that if

$$w(x) = \begin{cases} u(x) & \text{for } x \in \partial A, \\ \frac{\partial u}{\partial n}(x) & \text{for } x \in \partial\Omega, \end{cases}$$

where  $u$  is the solution to (1.5)–(1.7), then

$$(3.2) \quad \mathcal{L}w(x) = \mathcal{F}(x) \quad \forall x \in \Gamma.$$

Problem (3.2) is discretized as follows. The interface  $\partial A = \partial A_h$  is obtained through contouring of a given level-set function, see section 2, and piecewise constant elements are considered. The function  $w$  solution to (3.2) is approximated by  $w_h$  such that

$$w_h(x) = w_e \quad \forall x \in e,$$

where  $e$  is an edge of either  $\partial A_h$  or  $\partial\Omega_h$  and where  $w_e$  is the constant value of  $w_h$  on that edge. Equation (3.2) is then collocated at the midpoints of the edges. In other words, for a generic piecewise constant function  $v_h$ , we define

$$\mathcal{L}_h v_h(\bar{\xi}_e) = \begin{cases} \frac{v_h(\bar{\xi}_e)}{2} + \int_{\partial A_h} \frac{\partial G}{\partial n_y}(\bar{\xi}_e, y) v_h(y) ds(y) - \int_{\partial\Omega_h} G(\bar{\xi}_e, y) v_h(y) ds(y) \\ \text{for } \bar{\xi}_e \in \partial A_h, \\ \int_{\partial A_h} \frac{\partial G}{\partial n_y}(\bar{\xi}_e, y) v_h(y) ds(y) - \int_{\partial\Omega_h} G(\bar{\xi}_e, y) v_h(y) ds(y) \\ \text{for } \bar{\xi}_e \in \partial\Omega_h, \end{cases}$$

where  $\bar{\xi}_e$  is the midpoint of the edge  $e$ . Similarly, we also have

$$\mathcal{F}_h(\bar{\xi}_e) = \begin{cases} \mu \int_{\partial A_h} G(\bar{\xi}_e, y) ds(y) - \int_{\partial\Omega_h} \frac{\partial G}{\partial n_y}(\bar{\xi}_e, y) ds(y) & \text{for } \bar{\xi}_e \in \partial A_h, \\ \mu \int_{\partial A_h} G(\bar{\xi}_e, y) ds(y) - \int_{\partial\Omega_h} \frac{\partial G}{\partial n_y}(\bar{\xi}_e, y) ds(y) - \frac{1}{2} & \text{for } \bar{\xi}_e \in \partial\Omega_h. \end{cases}$$

The approximate solution  $w_h$  is the solution to

$$(3.3) \quad \mathcal{L}_h w_h(\bar{\xi}_e) = \mathcal{F}_h(\bar{\xi}_e) \quad \forall \bar{\xi}_e \in \partial A_h \cup \partial\Omega_h.$$

The above integrals are computed exactly in the present implementation, in the sense of the Cauchy principal value where appropriate; see [4] for more details.

Both the integral equation (3.2) and the linear problem (3.3) are well conditioned with one exception: 0 is an eigenvalue of  $\mathcal{L}$  if and only if the transfinite diameter of  $A$  is 1 [20]. This can be avoided by appropriate rescaling if necessary. It can be verified that  $\mathcal{L}_h$  admits eigenvalues and singular values that are bounded independent of the mesh; see, e.g., [20] or [31] for explicit expressions of the eigenvalues in some specific cases. Further, in spite of the fact that the elements of  $\partial A_h$  are allowed to be arbitrarily small, the condition number of  $\mathcal{L}_h$  has been numerically verified to be of order  $N$ , which would correspond to the uniform mesh case [3]. The condition number of the matrices corresponding to the numerical tests of section 4 are on the order of 100. The resulting linear system is solved by GMRES [25, 33], which is consequently expected to perform well here even without preconditioning; see also [7] for a general convergence result for GMRES applied to integral equations. GMRES is restarted after 20 steps (i.e., the solver is GMRES(20)) and is stopped on small relative residuals; more precisely, the stopping criterion is

$$\|\mathcal{F}_h - \mathcal{L}_h w\|_2 \leq 10^{-10} \|\mathcal{F}_h\|_2,$$

where  $w$  denotes the current iterate. With the above parameters, GMRES has been observed to perform slightly better than other CG-like methods such as QMR and Bi-CGSTAB [25] on the test problems of section 4.

#### 4. Numerical results.

**4.1. Algorithm.** To solve the external Bernoulli problem, we use the following algorithm:

```

Input: a discretization of the boundary  $\partial\Omega$ ,  $\partial\Omega_h$ ,  $\mu$ 
Create the underlying Cartesian grid  $B_h$ 
Create a level set function  $\phi$  on  $B_h$  corresponding to  $\partial A_0$ 
 $k = 0$ 
 $R_{-1} = 10^{10}$  (initial residual)
loop
  Contour  $\phi$  (subsection 2.1) to find  $\partial A_k$ 
  Solve (3.3) to get  $u$ 
   $R_k = \max$  of  $|u|$  on  $\partial A_k$ 
  if  $(R_{k-1} - R_k)/R_k < 10^{-3}$  (small residual decrease) then
    STOP
  end if
  Set  $F = u$  on  $\partial A_k$ 
  Extend  $\phi$  and  $F$  (subsections 2.2 and 2.3)
  Move boundary (subsection 2.4)
   $k = k + 1$ 
end loop

```

Several remarks are in order.

- Profiling reveals that for reasonable meshes the relative costs of contouring and interface updating are very low (less than 1%), while most of the time is spent constructing and solving the linear system (3.3) (about 90%) and extending the solution (about 10%). Methods such as fast multipoles [5, 32] can be used to bring down the asymptotic complexity of the linear solver from  $\mathcal{O}(N^2)$  to  $\mathcal{O}(N)$ , where  $N$  is the number of elements of  $\partial\Omega_h$ . However, this results in increased efficiency only for very large values of  $N$ ; see [26, Appendix] for more details; see also [16] for a large-scale two-dimensional example. Fast summation methods should be considered for three-dimensional problems.
- Progressive mesh refinement can be considered; i.e., a coarse mesh solution can be used as the starting point. A strategy of this type is, for instance, used in [23] for a similar type of problem but for a different numerical approach.
- In solving (3.3), the “missing” condition (1.3) can be used when choosing the initial iterate for GMRES. This results in faster convergence (fewer GMRES iterates) as the algorithm progresses.
- The extension step through fast marching (subsection 2.3) is done in the whole computational domain  $B$ . The corresponding complexity is  $\mathcal{O}(M^2 \log M)$ , where  $M^2$  is the total number of nodes in the Cartesian grid  $B_h$ . A narrow band/computational tube implementation [2, 37] could be considered to speed up the algorithm. However, the global complexity of the problem would not change: if the width of the band is a constant multiple of  $\Delta x$ , say  $n\Delta x$ , then by (2.9),  $\Delta t$  should be reduced from  $\frac{-1}{2\mu}$  to a value less than  $\frac{n\Delta x}{|F|}$  since the band has to contain the boundary.
- Higher order boundary element methods can be used [4]. While second order convergence is observed in section 4, this is a result of the solution being constant on the outer free boundary. Piecewise linear elements should be used to ensure second order convergence for more general cases. To the authors’ knowledge, the present work is one of very few published results regarding the accuracy of a combined level-set boundary element method; see, for instance, [14].



TABLE 1

Convergence and complexity rates for Example 1 (radial case);  $N$  is number of elements on  $\partial\Omega_h$ ,  $L^\infty$  error refers to the Hausdorff distance between exact and computed boundaries,  $k$  is the number of nonlinear iterations (see section 4.1), Time is the runtime in seconds.

$N$	$\max  u $ on $\Gamma$	Rate	$L^\infty$ error	Rate	$k$	Time	Rate
25	1.64(−2)	—	1.53(−2)	—	6	0.71	—
50	5.07(−3)	1.7	4.40(−3)	1.8	8	4.1	2.5
100	1.30(−3)	2.0	1.12(−3)	2.0	10	19	2.2
200	3.34(−4)	2.0	2.85(−4)	2.0	11	76	2.0
400	8.18(−5)	2.0	7.10(−5)	2.0	12	372	2.1
800	1.92(−5)	2.1	1.74(−5)	2.0	14	1465	2.0

- The use of second order fast marching is not crucial for the accuracy of the method which is dictated by both the contouring and boundary element methods. It was observed that first order fast marching may lead to more iterations (domain updates), and, hence, due to the very little difference in cost per iteration, the use of second order fast marching is advocated.
- The numerical solutions are observed to be remarkably independent of the initial domain  $\partial A_0$  (for problems with unique solutions [11]). More iterations can be expected in case of poor initial guesses (for instance if the smoothness properties of  $\partial A_0$  are vastly different from those of  $\partial A$ ).

**4.2. Example 1.** Following [11], a quick look at the radial case is instructive. Let  $\Omega$  be the unit ball. We consider the problem (1.1)–(1.4) with  $\Omega$  as above and  $\mu = -2$ . The solution to (1.1)–(1.3) with  $A$  being the ball of radius  $R$  centered at the origin is

$$u(r) = -2R \log r + 1,$$

expressed in polar coordinates. An iterative process similar to the one above can then be considered. Taking (1.8) into account, the  $k$ th step of the algorithm reads

$$R_{k+1} = R_k - R_k \log R_k + \frac{1}{2}, \quad k = 1, 2, \dots$$

Therefore, in the fully radial case, the problem amounts to finding a fixed point to the function  $f(R) = R - R \log R + \frac{1}{2}$ . The function  $f$  has a unique fixed point  $\bar{R}$ , where

$$\bar{R} = \frac{1}{2W(\frac{1}{2})},$$

the function  $W$  being the Lambert W function<sup>4</sup> [10].

Table 1 shows second order convergence in the  $L^\infty$  norm. Full second order convergence is obtained even though the elliptic solver is based on a piecewise constant discretization. This is due to the fact that the exact solution is constant on both the inner and outer boundaries. Further, when measured with respect to runtime, the complexity is also second order.

The convergence history is instructive. Figure 2, left, displays the error (maximum of  $|u|$  on the free boundary) through the iterations. The behavior of the first iterates is governed by the geometry, see (2.9), and is only weakly dependent on the mesh size

<sup>4</sup>The Lambert W function is the inverse of  $W \mapsto z = W e^W$ .

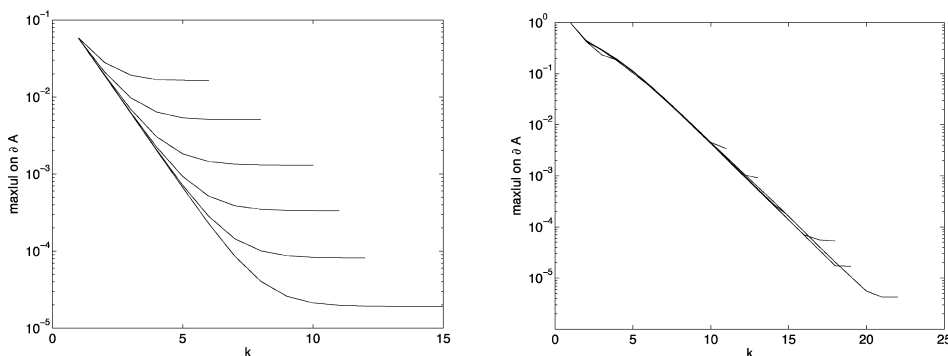


FIG. 2. Convergence history of the iterations: maximum value of  $|u|$  on the current interface as a function of the iteration number; left: Example 1, right: Example 2. The convergence curves are in the obvious order: from top to bottom,  $N = 25, 50, 100, 200, 400, 800$ .

$\Delta x$ . The later iterations during which the fine structure of the boundary is determined do depend on  $\Delta x$ . This explains the mesh dependency of the number of iterations observed in Table 1.

**4.3. Example 2.** We consider here the problem (1.1)–(1.4) with  $\Omega$  consisting of two disks of radius 1, one centered at  $(-2, 2)$  and one at  $(2, -2)$ ; further,  $\mu = -1/4$ . The initial boundary is taken as two circles of radius 1.1, one around each of the inner disks. Note that for this choice of  $\mu$ , the exact boundary is simply connected. A couple of iterates are displayed in Figure 3. One can note that after the first step the correct topology of the interface has already been achieved.

No exact solution is available for the present example. In Table 2, the maximum of  $u$  on the boundary is reported. By construction, this maximum should vanish for the converged solution. The complexity, as measured from the runtimes, is also reported. In both cases, the rates are about two.

Convergence history is displayed in Figure 2, right; a behavior similar as that of Example 1 is observed.

**5. Conclusion.** Solutions of the Bernoulli free boundary problem can be efficiently computed by the method presented here. Providing a Green's function is available, the method can be used to solve other free boundary problems. For instance, it can be applied with only minor modifications to the Prandtl–Batchelor problem (see [1] and the references therein), which consists in looking for a domain  $A$  which is now interior to the fixed domain  $\Omega$  such that for a given function  $\sigma$ ,

$$|\nabla u_A|^2 - |\nabla u_\Omega|^2 = \sigma \quad \text{on } \partial A,$$

where  $u_A$  and  $u_\Omega$  solve

$$\begin{cases} \Delta u_A = -\omega & \text{in } A, \\ u_A = 0 & \text{on } \partial A, \end{cases} \quad \begin{cases} \Delta u_\Omega = 0 & \text{in } \Omega, \\ u_\Omega = 0 & \text{on } \partial A, \\ u_\Omega = \mu & \text{on } \partial \Omega, \end{cases}$$

$\mu$  and  $\omega$  being positive constants.<sup>5</sup>

<sup>5</sup>To avoid the calculation of a bulk due to the inhomogeneity, a particular solution to  $\Delta u_0 = -\omega$  (for instance  $u_0 = -\frac{\omega}{4}(x^2 + y^2)$ ) can easily be subtracted from  $u_A$ . This restores homogeneity but renders the boundary condition nonconstant.

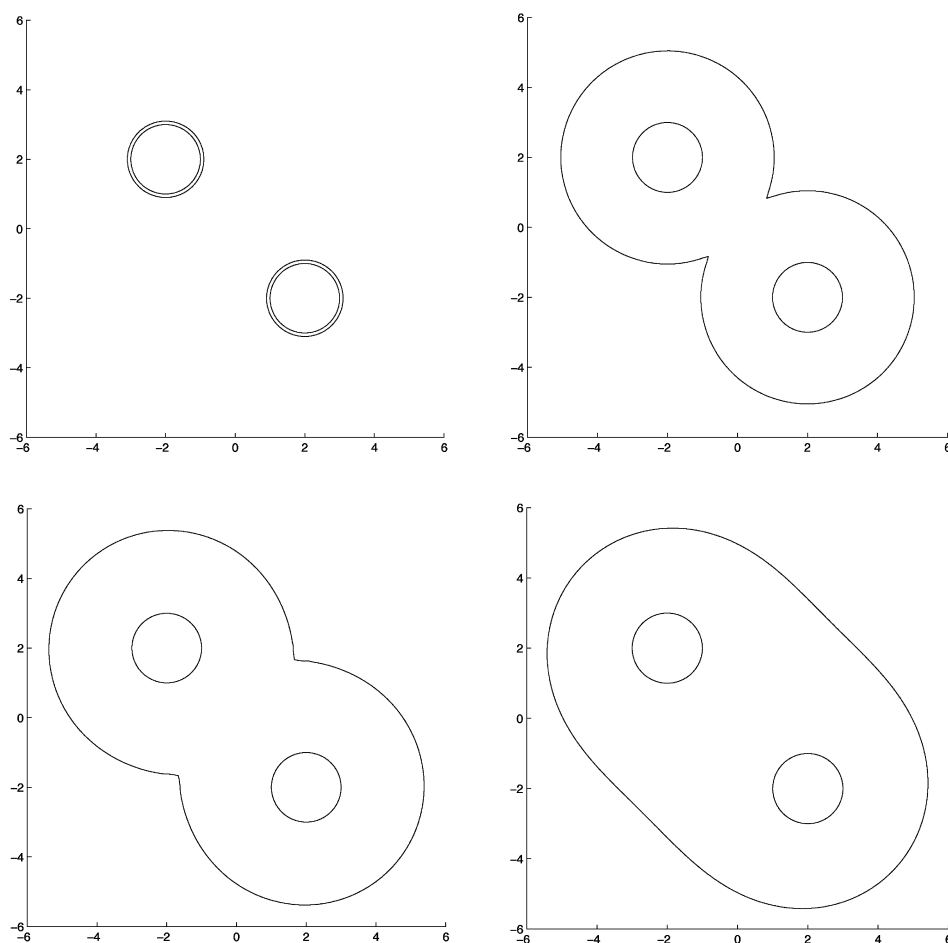


FIG. 3. Evolution of the interface for Example 2 at the initial step and after steps 1, 2, and 15 ( $N = 100$ ).

TABLE 2

Convergence and complexity rates for Example 2;  $N$  is number of elements on  $\partial\Omega_h$ ,  $k$  is the number of nonlinear iterations (see section 4.1), Time is the runtime in seconds.

$N$	$\max  u $ on $\Gamma$	Rate	$k$	Time	Rate
26	3.43(-3)	—	11	4.4	—
50	9.16(-4)	2.0	13	19	2.2
100	1.64(-4)	2.5	15	85	2.2
200	5.31(-5)	1.6	18	573	2.8
400	1.71(-5)	1.6	19	1678	1.6
800	4.25(-6)	2.0	21	7613	2.2

## REFERENCES

- [1] A. ACKER, *On the existence of convex solutions to a generalized Prandtl-Batchelor free boundary problem-II*, Z. Angew. Math. Phys., 53 (2002), pp. 438–485.
- [2] D. ADALSTEINSSON AND J. A. SETHIAN, *The fast construction of extension velocities in level set methods*, J. Comput. Phys., 148 (1999), pp. 2–22.

- [3] M. AINSWORTH, W. MCLEAN, AND T. TRAN, *The conditioning of boundary element equations on locally refined meshes and preconditioning by diagonal scaling*, SIAM J. Numer. Anal., 36 (1999), pp. 1901–1932.
- [4] K. E. ATKINSON, *The Numerical Solutions of Integral Equations of the Second Kind*, Cambridge Monogr. Appl. Comput. Math. 4, Cambridge University Press, Cambridge, UK, 1997.
- [5] R. BEATSON AND L. GREENGARD, *A short course on fast multipole methods*, in Wavelets, Multilevel Methods and Elliptic PDEs (Leicester, 1996), Numer. Math. Sci. Comput., Oxford University Press, New York, 1997, pp. 1–37.
- [6] F. BOUCHON, S. CLAIN, AND R. TOUZANI, *Numerical solution of the free boundary Bernoulli problem using a level set formulation*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 3934–3948.
- [7] S. L. CAMPBELL, I. C. F. IPSEN, C. T. KELLEY, C. D. MEYER, AND Z. Q. XUE, *Convergence estimates for solution of integral equations with GMRES*, J. Integral Equations Appl., 8 (1996), pp. 19–34.
- [8] D. L. CHOPP, *Some improvements of the fast marching method*, SIAM J. Sci. Comput., 23 (2001), pp. 230–244.
- [9] A. J. CHORIN AND J. E. MARSDEN, *A Mathematical Introduction to Fluid Mechanics*, Springer, New York, 1992.
- [10] R. M. CORLESS, G. H. GONNET, D. E. G. HARE, D. J. JEFFREY, AND D. E. KNUTH, *On the Lambert W function*, Adv. Comput. Math., 5 (1996), pp. 329–359.
- [11] M. FLUCHER AND M. RUMPF, *Bernoulli's free-boundary problem, qualitative theory and numerical approximation*, J. Reine Angew. Math., 486 (1997), pp. 165–204.
- [12] K. O. FRIEDRICHS, *Über ein Minimumproblem für Potentialströmungen mit freiem Rand*, Math. Ann., 109 (1934), pp. 60–82.
- [13] P. R. GARABEDIAN, *Calculation of axially symmetric cavities and jets*, Pacific J. Math., 6 (1956), pp. 611–684.
- [14] M. GARZON, D. ADALSTEINSSON, L. GRAY, AND J. A. SETHIAN, *A coupled level set-boundary integral method for moving boundary simulations*, Interfaces Free Bound., 7 (2005), pp. 277–302.
- [15] R. GONZÁLEZ AND R. KRESS, *On the treatment of a Dirichlet–Neumann mixed boundary value problem for harmonic functions by an integral equation method*, SIAM J. Math. Anal., 8 (1977), pp. 504–517.
- [16] A. GREENBAUM, L. GREENGARD, AND G. B. MCFADDEN, *Laplace's equation and the Dirichlet–Neumann map in multiply connected domains*, J. Comput. Phys., 105 (1993), pp. 267–278.
- [17] P. A. GREMAUD, C. M. KUSTER, AND Z. LI, *A study of numerical methods for the level set approach*, Appl. Numer. Math., to appear.
- [18] P. A. GREMAUD AND C. M. KUSTER, *Computational study of fast methods for the eikonal equation*, SIAM J. Sci. Comput., 27 (2006), pp. 1803–1816.
- [19] J. HASLINGER, T. KOZUBEK, K. KUNISCH, AND G. PEICHL, *Shape optimization and fictitious domain approach for solving free-boundary problems of Bernoulli type*, Comput. Optim. Appl., 26 (2003), pp. 231–251.
- [20] J. HAYES AND R. KELLNER, *The eigenvalue problem for a pair of coupled integral equations arising in the numerical solution of Laplace's equation*, SIAM J. Appl. Math., 22 (1972), pp. 503–513.
- [21] A. HENROT AND H. SHAHGHOLIAN, *Existence of classical solutions to a free boundary problem for the  $p$ -Laplace operator. I. The exterior convex case*, J. Reine Angew. Math., 521 (2000), pp. 85–97.
- [22] A. HENROT AND H. SHAHGHOLIAN, *The one phase free boundary problem for the  $p$ -Laplacian with non-constant Bernoulli boundary condition*, Trans. Amer. Math. Soc., 354 (2002), pp. 2399–2416.
- [23] K. ITO, K. KUNISCH, AND G. H. PEICHL, *Variational approach to shape derivatives for a class of Bernoulli problems*, J. Math. Anal. Appl., 314 (2006), pp. 126–149.
- [24] K. T. KÄRKKÄINEN AND T. TIIHONEN, *Free surfaces: Shape sensitivity analysis and numerical methods*, Internat. J. Numer. Methods Engrg., 44 (1999), pp. 1079–1098.
- [25] C. T. KELLEY, *Iterative Methods for Linear and Nonlinear Problems*, Frontiers Appl. Math. 16, SIAM, Philadelphia, 1995.
- [26] C. M. KUSTER, *Fast Numerical Methods for Evolving Interfaces*, Ph.D. thesis, Department of Mathematics, North Carolina State University, Raleigh, NC, 2006; <http://www.lib.ncsu.edu/theses/available/etd-04262006-083221/>.
- [27] Z. LI AND K. ITO, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, Frontiers Appl. Math. 33, SIAM, Philadelphia, 2006.
- [28] G. MEJAK, *Numerical solution of Bernoulli-type free boundary value problems by variable domain method*, Internat. J. Numer. Methods Engrg., 37 (1994), pp. 4219–4245.

- [29] S. J. OSHER AND R. P. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Appl. Math. Sci. 153, Springer, New York, 2002.
- [30] S. OSHER AND J. SETHIAN, *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 56 (1988), pp. 12–49.
- [31] G. J. RODIN AND O. STEINBACH, *Boundary element preconditioners for problems defined on slender domains*, SIAM J. Sci. Comput., 24 (2003), pp. 1450–1464.
- [32] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1983), pp. 187–207.
- [33] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [34] J. A. SETHIAN, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK, 1999.
- [35] J. A. SETHIAN AND A. VLADIMIRSKY, *Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes*, Proc. Natl. Acad. Sci. USA, 97 (2000), pp. 5699–5703.
- [36] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton-Jacobi equations: Theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325–363.
- [37] L. YATZIV, A. BARTESAGHI, AND G. SAPIRO,  *$O(N)$  implementation of the fast marching algorithm*, J. Comput. Phys., 212 (2006), pp. 393–399.