MSRI
Mathematical Sciences Research Institute

FREE BOUNDARY PROBLEMS IN FLUID MECHANICS

ANA MARIA SOANE* AND ROUBEN ROSTAMIAN

We consider a class of free boundary problems governed by the incompressible Navier-Stokes equations. Our objective is to develop robust numerical algorithms that capture the shape of the free boundary. In our approach, we combine a recent algorithm of Liu, Liu, and Pego for solving the Navier-Stokes equations with a prediction-correction technique for locating the free boundary. The Liu, Liu, and Pego algorithm calls for the rather uncommon C^1-compatible finite elements. A major part of our work is to develop software that implements their algorithm, then adapt it to handle free boundaries.

1. Numerical method for Navier-Stokes equations

Fluid flow is governed by the Navier-Stokes equations. For incompressible viscous fluid flow in a domain $\Omega \subset \mathbb{R}^N (N \geq 2)$, they take the following form:

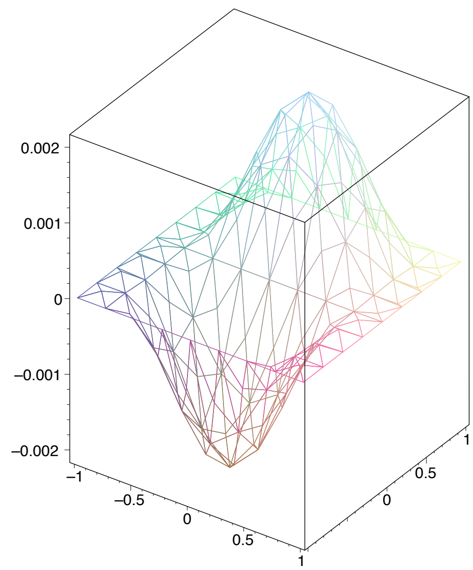
$$\mathbf{u}_t + (\nabla \mathbf{u})\mathbf{u} + \nabla p = \nu \Delta \mathbf{u} + \mathbf{f} \quad \text{in } \Omega \times (0, T], \tag{1}$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T], \tag{2}$$

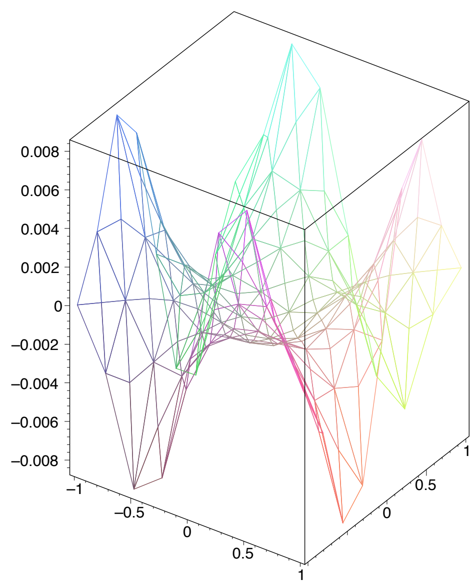
where $\mathbf{u}$ is the fluid velocity, p is the pressure, ν is the kinematic viscosity, and $\mathbf{f}$ is an external force per unit mass. The system (1)–(2) is completed with an initial condition for velocity and appropriate boundary conditions.

Some of the most popular numerical methods for the incompressible Navier-Stokes equations are splitting schemes that decouple the computation of the velocity and the pressure. In [4], Liu, Liu, and Pego have proposed such a scheme, based on a time-discretization explicit in pressure and convection terms. At each time step it is only required to solve a Poisson equation for the pressure and an elliptic boundary-value problem for the velocity. The corresponding fully discrete finite element method with C^1 elements for velocity and C^0 elements for pressure is unconditionally stable and does not require any compatibility conditions between the finite element spaces for velocity and pressure.

2000 *Mathematics Subject Classification.* 76D05, 76D27, 76M10.
Key words and phrases. Navier-Stokes equations, free boundary, finite elements.



(a) Error in x-velocity



(b) Error in pressure

FIGURE 1. Absolute error in velocity and pressure.

We have developed a finite element code that implements the algorithm in [4] in C. The algorithm requires the rather uncommon C^1-compatible finite elements; therefore, a major part of our work thus far has gone into implementing such elements. We use the triangular Argyris quintic polynomials for our elements. We have tested our finite element solver on several benchmark problems for Navier-Stokes solvers. We present here results for the problem (1)–(2) in $\Omega = (-1,1) \times (-1,1)$, with the source term $\mathbf{f}$ chosen such that the exact solution is

$$\mathbf{u}(x,y,t) = \begin{pmatrix} \cos t \cos^2(\pi x/2)\cos(\pi y/2)\sin(\pi y/2) \\ -\cos t \cos(\pi x/2)\sin(\pi x/2)\cos^2(\pi y/2) \end{pmatrix},$$
$$p(x,y,t) = \cos t \cos(\pi x/2)\sin(\pi y/2).$$

Figure 1 (page 234) shows the graphs of (a) absolute error in velocity and (b) absolute error in pressure at $t = 1$, using 256 elements and $\Delta t = 0.01$.

2. Numerical method for computing the free boundary

We are interested in computing flows with free boundaries. Our objective is to combine the algorithm in [4] with an interface tracking technique for computing the location of the free boundary. We present here the normal stress iterative method for solving a stationary free boundary problem governed by the Navier-Stokes equations, as described in [1] and [2]. The method works as follows. An initial guess for the shape of the free boundary is assigned and the flow within the domain bounded by that shape is computed after disregarding the continuity of normal component of the stress at the free boundary. Next, a new shape of the free boundary is computed that satisfies as closely as possible the relaxed boundary condition. The procedure is repeated until convergence is attained.

To illustrate, consider the steady-state motion of fluid in the domain

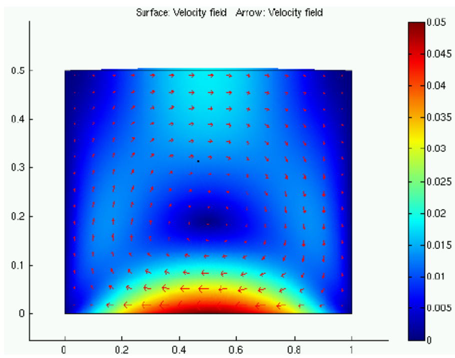
$$\Omega_\varphi = \{(x,y) \in \mathbb{R}^2 : 0 \le x \le L, 0 \le y \le \varphi(x)\},$$

where the curve $S : y = \varphi(x)$ is the free boundary to be determined. The fluid velocity on the part $\Gamma = \partial\Omega \setminus S$ of the boundary and the total volume of fluid V are prescribed. Thus, we have the following equations:

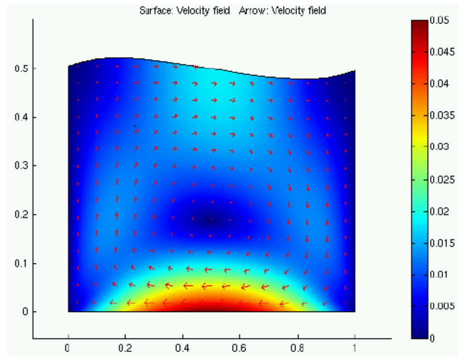
$$\begin{aligned} &\rho(\nabla\mathbf{u})\mathbf{u} = \mu\Delta\mathbf{u} - \nabla p - \rho g \begin{pmatrix} 0 \\ 1 \end{pmatrix} && \text{in } \Omega_\varphi, \\ &\operatorname{div}\mathbf{u} = 0 && \text{in } \Omega_\varphi, \\ &\mathbf{u} = \mathbf{u}_\Gamma && \text{on } \Gamma, \\ &\mathbf{u}\cdot\mathbf{n} = 0 && \text{on } S, \\ &T\mathbf{n} = \mathbf{0} && \text{on } S, \\ &\int_0^L \varphi(x)\,dx = V, \end{aligned} \tag{3}$$

where $\mathbf{n}$ is the unit outward normal on the boundary S, and T is the Cauchy stress tensor

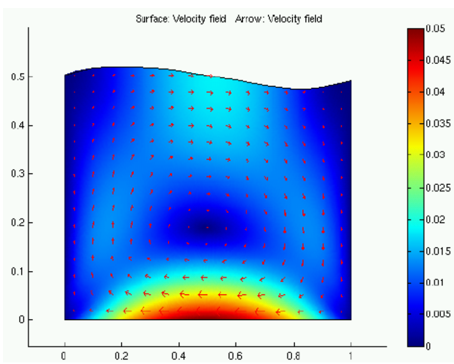
$$T = -pI + 2\mu D, \text{ where } D = \frac{1}{2}\mu\big[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\big].$$



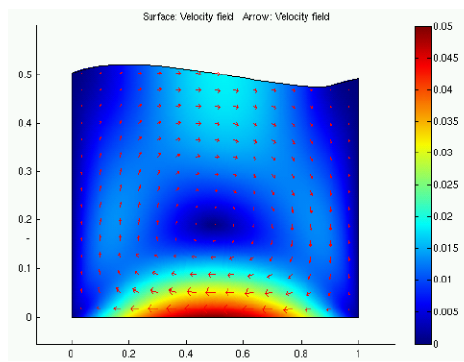
(a) Iteration 0.



(b) Iteration 1.



(c) Iteration 2.



(d) Iteration 3.

FIGURE 2. Computed free boundary.

Following [1], we introduce the following *auxiliary problem*: Given φ such that $\int_0^L \varphi(x)\,dx = V$, solve

$$\begin{aligned} &\rho(\nabla \mathbf{u})\mathbf{u} = \mu\Delta\mathbf{u} - \nabla p_0 && \text{in } \Omega_\varphi,\\ &\operatorname{div}\mathbf{u} = 0 && \text{in } \Omega_\varphi,\\ &\mathbf{u} = \mathbf{u}_\Gamma && \text{on } \Gamma,\\ &\mathbf{u}\cdot\mathbf{n} = 0 && \text{on } S,\\ &T\mathbf{n} \parallel \mathbf{n} && \text{on } S, \end{aligned} \tag{4}$$

where $T = -p_0 I + 2\mu D$. This is a standard boundary value problem for the Navier-Stokes equations. It can be shown that

$$\varphi(x) - \varphi^* = -\frac{1}{\rho g}\left\{\left[2\mu\frac{\partial\mathbf{u}}{\partial\mathbf{n}}\cdot\mathbf{n} - p_0\right]_S - \frac{1}{L}\int_0^L \left[2\mu\frac{\partial\mathbf{u}}{\partial\mathbf{n}}\cdot\mathbf{n} - p_0\right]_S dx\right\}, \tag{5}$$

where $\varphi^* = \frac{1}{L}$ is the height of the fluid when the fluid is static. Equation (5) forms the basis of an iterative search for the shape of the free boundary. As an initial guess, we pick $\varphi_0 = \varphi^*$. Then we generate a sequence of iterates φ_n as follows: Given φ_n, solve the auxiliary problem (4). Then evaluate the right hand side of (5) and compute φ_{n+1} from $\varphi_{n+1} = \varphi^* + rhs$. The iteration is guaranteed to converge if the motion is close enough to static.

We formulated and solved this problem using FEMLAB, with the iterative algorithm for computing the free boundary implemented in MATLAB. Numerical experiments show that for small Reynolds numbers, the method converges very quickly. However, even for moderate Reynolds numbers, convergence is not attained. Figure 2 (page 236) shows the computed free boundary for Reynolds number $Re = 1$. We expect that our solver will be able to handle a wider range of Reynolds numbers.

3. Future work

The major objective of our future work is to adapt the Navier-Stokes solver algorithm outlined in section 1 to handle free boundaries. We are also considering an alternative algorithm for the computation of the free boundary, based on the relaxation of the kinematic boundary condition on the free boundary [3]. Our ultimate goal is to analyze the symmetry-breaking in a stationary circular hydraulic jump.

References

[1] C. Cuvelier and R. M. S. M. Schulkes, *Some numerical methods for the computation of capillary free boundaries governed by the Navier-Stokes equations*, SIAM Rev. **32** (1990), no. 3, 355–423.

[2] C. Cuvelier, A. Segal, and A. A. van Steenhoven, *Finite Element Methods and Navier-Stokes Equations*. Mathematics and its Applications, 22. Dordrecht: D. Reidel Publishing Company, 1986.

[3] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*. 2nd rev. ed. Berlin: Springer-Verlag, 1999.

[4] J.-G. Liu, J. Liu, and R. L. Pego, *Stability and convergence of efficient Navier-Stokes solvers via a commutator estimate*. To appear in Communications on Pure and Applied Mathematics.

(Soane) Department of Mathematics and Statistics; University of Maryland, Baltimore County; 1000 Hilltop Circle; Baltimore, MD 21250 USA

E-mail address: asoane@math.umbc.edu

(Rostamian) Department of Mathematics and Statistics; University of Maryland, Baltimore County; Baltimore, MD 21250 USA