STABILITY AND CONVERGENCE OF IMMERSED BOUNDARY COMPUTATIONS

John M. Stockie and Brian R. Wetton
Institute of Applied Mathematics, University of British Columbia
Vancouver, British Columbia, Canada
stockie@math.ubc.ca, wetton@math.ubc.ca

INTRODUCTION

The "Immersed Boundary Method" was developed by Peskin [1] to simulate the flow of blood through artificial heart valves. It has since been extended to three dimensions [2] and applied to various other physical situations, including swimming microorganisms [3, 4], amoeboid locomotion [5] and plasma simulations [6], to name a few. The main strengths of the method are its geometric flexibility and its ability to compute realistic qualitative results in situations where complex elastic interfaces or fibers interact with a surrounding fluid.

The heart muscle, or other immersed boundary, is modeled as a closely–interwoven mesh of elastic fibers which are immersed within an incompressible fluid. The fibers are neutrally buoyant and move with the local fluid velocity, while at the same time exerting on surrounding fluid particles an elastic force which depends on the stretched state of the fibers. The Immersed Boundary Method uses a mixed Eulerian–Lagrangian approach to discretize this problem. The fluid velocity and pressure are computed on a fixed rectangular grid, while the fibers are treated as a collection of moving points, linked to each other by elastic "springs."

The simplicity of the fiber representation allows moving, internal boundaries of practically any shape or configuration to be simulated. Furthermore, the fact that the underlying fluid grid is regularly–spaced means that fast fluid solvers can be applied to solve the equations of motion. On the other hand, the method is limited to first order accuracy in space due to the interpolation scheme that is used to transfer quantities between fluid and fiber grid points, though some recent work by Roma [7] uses adaptive gridding to overcome this limitation. Immersed boundary computations have also been demonstrated to suffer from a high degree of stiffness [8]. Even though a considerable amount of work has gone into developing improved schemes for coupling the fluid and fiber motion [3, 9, 8, 7], efficient implementations are forthcoming, and many computations are still being done with explicit schemes ([4], for example).

In this paper we present an analytical technique, based on Fourier mode analysis, which allows us to investigate the stability of the underlying equations of motion for immersed fibers. The results give insight into the behaviour of the solution and explain the high degree of stiffness inherent in immersed boundary computations. The technique is extended to the time–discrete problem, and various implicit schemes are compared in terms of the convergence rate of the iteration embedded in each time step, with the analytical results backed up by computations. The eventual aim of this work is to develop a more efficient implementation, using the insight gained from our analysis to deal more effectively with the stiffness inherent in immersed fibers.

IMMERSED FIBERS

In the remainder of this paper, we will consider a single, isolated fiber immersed in a two–dimensional fluid. Since a three–dimensional boundary is modelled as a mesh of immersed fibers, we expect that many important characteristics of the fibers will be captured in this two dimensional model. We begin by presenting the equations of motion for an immersed fiber in two dimensions and then outline the numerical scheme and several of its variants.

The Mathematical Formulation

Figure 1 depicts the model two–dimensional "heart," consisting of a fiber Γ immersed within a doubly–periodic fluid domain Ω. The fiber position is given by $\vec{x} = \vec{\mathcal{X}}(s,t)$, where s is a parametrization of the fiber in some reference configuration.

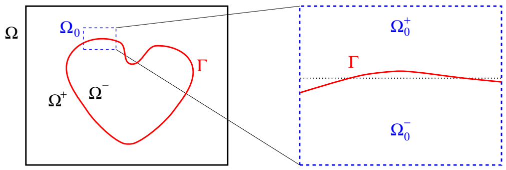


Figure 1: The 2D model "heart," consisting of a fiber Γ immersed in a unit square of fluid Ω. The stability analysis is performed on a subdomain Ω_0, on which the fiber is approximately flat.

The equations of motion for an incompressible fluid with velocity $\vec{u}(\vec{x},t)$, pressure $p(\vec{x},t)$, external force $\vec{F}(\vec{x},t)$, density ρ and viscosity μ are

$$\rho \frac{\partial \vec{u}}{\partial t} = -\rho \vec{u} \cdot \nabla \vec{u} + \mu \Delta \vec{u} - \nabla p + \vec{F}, \qquad (1a)$$

$$\nabla \cdot \vec{u} = 0. \qquad (1b)$$

The interaction between the fluid and fiber is a two–way process that can be summarized as follows:

- The fiber exerts a force on the surrounding fluid which is zero everywhere except on the fiber, and so can be written in terms of a convolution with a two–dimensional delta function $\delta^2(\vec{x})$:

  $$\vec{F} = \int_\Gamma \sigma \frac{\partial^2 \vec{\mathcal{X}}}{\partial s^2} \cdot \delta^2(\vec{x} - \vec{\mathcal{X}}(s,t))\, ds \qquad (1c)$$

  We have taken the force density on the fiber to be $\sigma(\partial^2 \vec{\mathcal{X}}/\partial s^2)$, which is analogous to successive fiber points being linked by linear springs with spring constant σ and resting length zero.

- The fiber points, in turn, move at the local fluid velocity:

  $$\begin{aligned} \frac{\partial \vec{\mathcal{X}}}{\partial t} &= \vec{u}(\vec{\mathcal{X}}(s,t),t) \\ &= \int_\Omega \vec{u}(\vec{x},t) \cdot \delta^2(\vec{x} - \vec{\mathcal{X}}(s,t))\, d\vec{x} \qquad (1d) \end{aligned}$$

  with the two forms of (1d) being equivalent.

Equations (1a)–(1d) are a coupled system of integro–partial differential equations, which we will refer to as the "*immersed fiber problem.*" The key features of these equations (which are exploited in the numerical method in the next section) are that the fiber affects the fluid only through the forcing term in the Navier–Stokes equations, and that the transfer between the fluid and fiber evolution equations is accomplished by the delta functions appearing in (1c) and (1d).

The Numerical Method

The Immersed Boundary Method arises naturally from the immersed fiber problem formulated in the previous section. We consider approximations at equally–spaced times $t_n = n\Delta t$, and divide the fluid domain $\Omega = [0,1]^2$ into a regular $N \times N$ lattice of points with spacing $h = \frac{1}{N}$. Then, discrete approximations of fluid quantities can be written $\vec{U}_{ij}^n \approx \vec{u}(ih, jh, n\Delta t)$ and $P_{ij}^n \approx p(ih, jh, n\Delta t)$, for $i,j = 0,1,\ldots,N-1$. Similarly, we discretize the fiber at a set of N_b moving points, with spacing $h_b = \frac{1}{N_b}$ and define $\vec{X}_k^n \approx \vec{\mathcal{X}}(kh_b, n\Delta t)$ for $k = 0,1,\ldots,N_b - 1$ (assuming the fiber parameter s takes on values between 0 and 1). A typical fluid/fiber grid is pictured in Figure 2, from which it is clear that fiber points need not coincide with the fluid mesh points. It is here that the role of the delta function as an *interpolating function* between the fluid and fiber grids becomes evident. We replace $\delta^2(\vec{x})$ in (1c) and (1d) by the discrete approximation $\delta_h^2(\vec{x}) = \delta_h(x) \cdot \delta_h(y)$, where

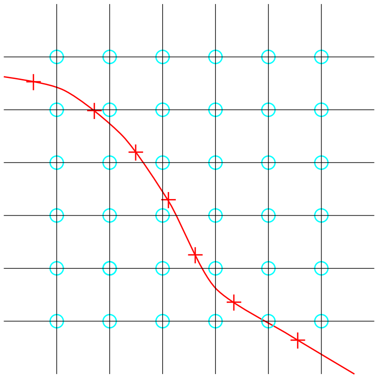


Figure 2: The fluid grid points ($\circ$) and moving fiber mesh points ($+$).

$$\delta_h(x) = \begin{cases} \frac{1}{2h}\left(1 + \cos\left(\frac{\pi x}{2h}\right)\right) & \text{if } x \le 2h, \\ 0 & \text{if } x > 2h, \end{cases} \qquad (2)$$

which is chosen according to [1] to satisfy a series of discrete compatibility conditions.

We are now in a position to formulate the discrete version of the problem as a fully implicit system of equations for the unknown solution values at time t_{n+1}. To be consistent with typical immersed boundary calculations which employ Chorin's split–step projection scheme [10], we use the operator $\mathcal{P}$ to represent a projection onto the space of divergence–free vector fields. Applying a Backward Euler time discretization to the problem, we obtain

$$\vec{F}_{ij}^{n+1} = \sum_k \sigma D_{h_b}^2 \vec{X}_k^{n+1} \cdot \delta_h^2(\vec{x}_{ij} - \vec{X}_k^{n+1}) h_b \tag{3a}$$

$$\begin{aligned}\rho \frac{\vec{U}_{ij}^{n+1} - \vec{U}_{ij}^n}{\Delta t} = \mathcal{P}\Big(&- \rho[\vec{U} \cdot \nabla_h \vec{U}]_{ij}^{n+1} \\ &+ \mu \Delta_h \vec{U}_{ij}^{n+1} + \vec{F}_{ij}^{n+1} \Big)\end{aligned} \tag{3b}$$

$$\frac{\vec{X}_k^{n+1} - \vec{X}_k^n}{\Delta t} = \sum_{i,j} \vec{U}_{ij}^{n+1} \delta_h^2(\vec{x}_{ij} - \vec{X}_k^{n+1}) h^2 \tag{3c}$$

In the Immersed Boundary Method, the convection and diffusion terms in (3b) are not actually handled in a fully implicit manner, but rather using an alternating–direction–implicit (ADI) step to get an intermediate velocity field, $\vec{U}^*$, which is then used in the projection step. Here, ∇_h and Δ_h are second order centered difference approximations to the divergence and Laplace operators, and $D_{h_b}^2 \vec{X}_k = \frac{1}{h_b^2}(\vec{X}_{k+1} - 2\vec{X}_k + \vec{X}_{k-1})$ replaces $\partial^2 \vec{\mathcal{X}} / \partial s^2$.

In practice, a scheme based on the fully implicit equations is extremely expensive since it requires a Newton solve for the coupled fluid–fiber equations at each time step. Tu & Peskin [8] recognized the necessity of handling the fiber force calculation implicitly, and several different semi–implicit schemes have since been proposed. Some of the major variants, which differ in whether the terms on the right hand sides of (3a)–(3c) are taken at time level n or $n + 1$, are listed below:

A. *Fully explicit:* All terms on the right hand sides of (3a)–(3c) are taken at time t_n. The stability requirements on the time step for this scheme are quite severe.

B. *"Approximate implicit":* This is the original scheme proposed by Peskin [1], in which the convection and diffusion terms in (3b) are computed using an ADI step to get an intermediate velocity $\vec{U}^*$. Only the force in (3b) is handled implicitly, resulting in a simple fixed point iteration for the fiber position. The fiber force can then be calculated using (3a), and the scheme proceeds as with the explicit method. There is considerable advantage to using this scheme because of its simplicity, but the time step restrictions are still severe.

C. *Semi–implicit:* The convection and diffusion terms in (3b) are computed by an ADI step, with the resulting scheme rewritten as an equation for $\vec{X}_k^{n+1}$ only. Mayo & Peskin [9] have proposed a fixed point iteration and a preconditioned conjugate gradient scheme for this version.

D. *Other variants:* Rather than a split–step projection scheme (which is known to suffer from pressure boundary layers), the velocity and pressure can be computed simultaneously using a Stokes solver. Another possibility is to reformulate the scheme as an iteration on the velocity rather than $\vec{X}^{n+1}$. We are currently investigating both of these alternatives.

It is important to note that the simple geometry of the fluid domain (a rectangular box with periodic boundary conditions) allows the fluid equations to be solved *very efficiently* (using a Fast Fourier Transform). Therefore, the central issue in developing an efficient scheme is an appropriate choice of coupling between the fluid and the fiber.

STABILITY AND CONVERGENCE ANALYSIS

Immersed boundary computations are known to require extremely small time steps for explicit and most semi–implicit schemes [8]. As mentioned in the previous section, much work has been done to develop better implicit schemes for coupling the fluid and fiber motion; however, there has been very little effort put in to *explaining why* immersed fibers are so hard to compute and investigating the convergence properties of the various schemes.

Stockie & Wetton [11] analyzed the stability of the equations of motion (1a)–(1d) using a modal analysis of the solution. They showed that in addition to Stokes' modes, the presence of an immersed fiber introduces several solution modes whose growth rates vary by several more orders of magnitude. Consequently, the fiber has some inherent stiffness, and computations based on the immersed fiber problem can be expected to be stiff. However, the analysis ignores smoothing effects from replacing the delta function by a discrete approximation. We extend these stability results in the next section to include smoothing effects.

Linear Stability

Consider a region such as Ω_0 pictured in Figure 1, where the fiber is not far from a horizontal equilibrium state, and introduce a small sinusoidal perturbation. We can then linearize the equations and take each dependent variable to be of the form

$$u(x,y,t) = e^{\lambda t + \imath \alpha x}\widehat{u}(y), \tag{4}$$

while replacing the delta function by the approximation $\delta_\epsilon(x)$ from (2) with smoothing radius ϵ. Using a symbolic algebra package such as *Maple* [12], we can solve the resulting system of integro–PDE's and derive a dispersion relation between λ and the wavenumber α and the other parameters in the problem (ϵ, σ, ν and ρ).

By restricting α to a finite range of integers $[0, N]$ (corresponding to the wavenumbers that can be resolved on an $N \times N$ grid), we can talk about the behaviour of discrete solutions in an idealized sense. The smoothing radius is chosen as $\epsilon = 2\pi\frac{2}{N}$, corresponding to a width of two "mesh points," with the extra factor of 2π needed to force the solution from (4) to have a period of 1. The sign of the quantity $Re(\lambda)$ indicates whether or not the immersed fiber problem is stable, and its magnitude gives an indication of the stiffness. The first column of Table 1 lists the analytical values for the largest growth rates, $Re(\lambda)$, for $N = 128$ and various values of the fiber stress, σ, that are used in typical calculations.

	Decay Rates $Re(\lambda)$		
σ	Smooth δ	Computed	Exact δ
1,000	–51.0	–39.2	–51.5
10,000	–81.3	–75.8	–83.9
100,000	–129.0	–119.4	–141.6
250,000	–152.9	–125.4	–175.6

Table 1: Analytical values of $Re(\lambda)$ for solution modes from the smoothed and exact delta–function problems along with the computed decay rates ($N = 128$).

We have implemented the Immersed Boundary Method and computed the rate of decay of the oscillations for a flat fiber given a small perturbation from equilibrium. This rate corresponds to the most slowly decaying modes ($\alpha = 1$), since the higher wavenumber modes decay most quickly. From the table, it is clear that the behaviour of the lowest wavenumber modes is captured quite accurately by the numerical scheme.

The final column in Table 1 presents the values of λ based on the results in [11] for the exact delta function (that is, without smoothing). It is clear that the introduction of smoothing has a considerable effect on the solution modes. By looking *all modes* (that is, by varying the wavenumber over the entire allowable range), we can determine how the smoothing of the delta function affects the stiffness of the problem. For $\sigma = 100,000$ and $\alpha \in [1, 128]$, we find for the smoothed problem that the growth rates lie in the range $-8.3 \times 10^4 \leq Re(\lambda) \leq -1.3 \times 10^2$, which indicates that the problem is quite stiff. By comparing to $-8.0 \times 10^5 \leq Re(\lambda) \leq -1.4 \times 10^2$ for the exact delta function from [11], we see that even though the problem is still stiff, it is much less severe for the smoothed problem and more in line with what is seen in computations.

Convergence of Time–Discrete Schemes

The modal analysis outlined in the previous section can be further extended to include the effects of time discretization in order to make conclusions about the convergence of various time–stepping schemes. We will consider two schemes, which are variants of the semi–implicit scheme C:

C1. A straightforward Crank–Nicholson–type splitting, with the terms on right hand sides of (3b) and (3c) averaged at times t_n and t_{n+1}.

C2. The preconditioned iteration proposed by Mayo & Peskin [9].

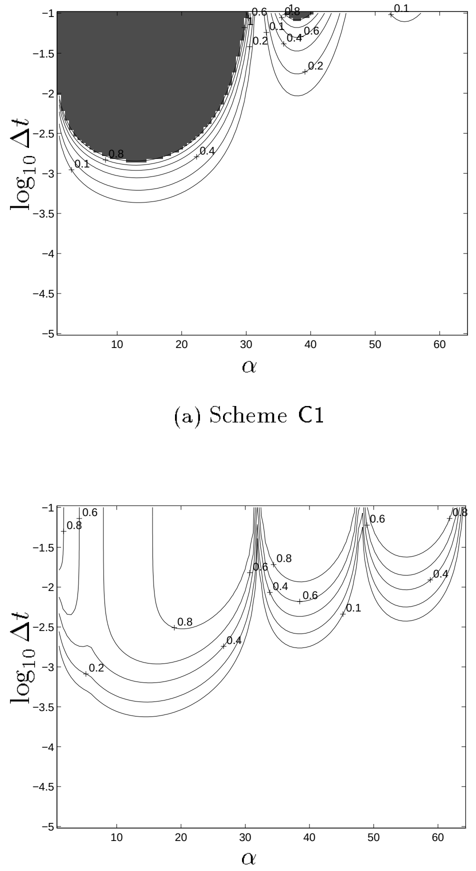
Both of these methods can be written (again with the aid of *Maple*) as a fixed point iterations on the interface position $\vec{X}^{n+1,k}$, with k denoting the iteration number:

$$A\vec{X}^{n+1,k+1} = B\vec{X}^{n+1,k} + R^n. \tag{5}$$

A and B are 2×2 matrices and R^n is composed of quantities from time level n, all depending on the parameters Δt, μ, ρ, α and σ.

If we define $\varrho_{\max}$ to be the largest eigenvalue of the matrix $B^{-1}A$, then the iteration converges if $|\varrho_{\max}| < 1$, and diverges otherwise. Plots of $|\varrho_{\max}|$ for both schemes are given in Figure 3, with parameter values $\mu = \rho = 1$, $\sigma = 10,000$ and $N = 64$ and over a range of Δt and $\alpha \in [1, 64]$. It is clear that the preconditioned scheme C2 converges for all plotted time steps, which agrees with the unconditional convergence result proven in [9].

On the other hand, scheme C1 converges only conditionally. To test the accuracy of the predicted stability region for this method, we performed some numerical experiments for which the results are summarized in Table 2. The stability requirement on



(a) Scheme C1

(b) Scheme C2

Figure 3: Convergence rate contours with the region of instability, $|\varrho_{\max}| \geq 1$, shaded ($\sigma = 10{,}000,\ N = 64$).

σ	Maximum Δt	
	Predicted	Computed
1,000	0.0050	0.0030
10,000	0.0014	0.0005
100,000	0.0004	0.0001
250,000	0.0002	0.00005

Table 2: Predicted and computed stability boundaries for Δt in scheme C1 ($N = 64$).

Δt in computations is *very sharp*, which matches with the steep contours in Figure 3(a) — that is, either the scheme diverges, or it converges within one or two iterations. The stability boundaries do not match exactly, but are within an order of magnitude of the predicted values.

We also performed several computations to verify the convergence rates for scheme C2 from Figure 3(b), and the results are summarized in Table 3. Here, the observed convergence rates match quite closely with the predicted values.

Δt	$\sigma = 10{,}000$		$\sigma = 100{,}000$	
	Pred.	Comp.	Pred.	Comp.
0.0001	0.02	0.01	0.16	0.24
0.0005	0.33	0.38	0.70	0.72
0.0010	0.57	0.60	0.80	0.90
0.0020	0.74	0.77	0.85	0.97
0.0050	0.84	0.96	0.89	0.99

Table 3: Predicted and computed convergence rates for scheme C2 ($N = 64$).

CONCLUSIONS AND FUTURE WORK

We have presented an analytical technique, based on Fourier mode analysis, which can be used to investigate the behaviour of immersed fibers. Predicted decay rates compare well with those observed in computations, and quantitative predictions can be made about the effects of smoothing on the numerical solution, due to the use of approximate delta functions. To our knowledge, this is the first effort at explaining why immersed boundary computations are so stiff.

The analytical tool can be extended to semi–discrete schemes (that is, discrete in time only), thereby allowing us to make qualitative comparisons between various iterative schemes for the immersed fiber problem. While a great deal of work has been done in developing new semi–implicit schemes, this is the first time that the stability and convergence of the various methods has been investigated analytically.

Based on the accuracy of predicted convergence rates, we expect that our modal analysis will prove to be a useful tool in evaluating *a priori* the effectiveness of other time–stepping schemes. With the insight we have gained about the behaviour of immersed fibers, we hope to develop an improved scheme that will deal more effectively with the stiffness intrinsic to immersed fibers.

REFERENCES

[1] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comp. Phys.*, 25:220–252, 1977.

[2] C. S. Peskin and D. M. McQueen. A three-dimensional computational model for blood flow in the heart. I. Immersed elastic fibers in a viscous incompressible fluid. *J. Comp. Phys.*, 81:372–405, 1989.

[3] L. J. Fauci and A. L. Fogelson. Truncated Newton methods and the modeling of complex immersed elastic structures. *Comm. Pure Appl. Math.*, 46:787–818, 1993.

[4] R. Dillon, L. J. Fauci, and D. Gaver III. Modeling biofilm processes using the immersed boundary method. *J. Comp. Phys.*, 129(1):57–73, 1996.

[5] D. C. Bottino. *An Immersed Boundary Model of Ameboid Deformation and Locomotion.* PhD thesis, Tulane University, 1996.

[6] G. Lapenta, F. Iinoya, and J. U. Brackbill. Particle-in-cell simulation of glow discharges in complex geometries. *IEEE Transactions on Plasma Science*, 23(4):769–779, Aug. 1995.

[7] A. M. Roma. *A Multilevel Self Adaptive Version of the Immersed Boundary Method.* PhD thesis, New York University, 1996.

[8] C. Tu and C. S. Peskin. Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods. *SIAM J. Sci. Stat. Comp.*, 13(6):1361–1376, 1992.

[9] A. A. Mayo and C. S. Peskin. An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. *Contemp. Math.*, 141:261–277, 1993.

[10] A. J. Chorin. Numerical solution of the Navier–Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[11] J. M. Stockie and B. T. R. Wetton. Stability analysis for the immersed fiber problem. *SIAM J. Appl. Math.*, 55(6):1577–1591, 1995.

[12] B. W. Char et al. *Maple V Language Reference Manual.* Springer-Verlag, 1991.