PAPER

Passive concentration dynamics incorporated into the library IB2d, a two-dimensional implementation of the immersed boundary method

To cite this article: Matea Santiago et al 2022 Bioinspir. Biomim. 17 036003

View the article online for updates and enhancements.

You may also like

- <u>Modal decompositions of the kinematics of</u> <u>Crevalle jack and the fluid–caudal fin</u> <u>interaction</u> Muhammad Saif Ullah Khalid, Junshi

Wang, Imran Akhtar et al.

- Bioinspired design and optimization for thin film wearable and building cooling systems
 Jonathan Grinham, Matthew J Hancock, Kitty Kumar et al.
- <u>School formation characteristics and</u> <u>stimuli based modeling of tetra fish</u> S R Rahman, I Sajjad, M M Mansoor et al.



This content was downloaded from IP address 69.242.232.75 on 17/04/2022 at 20:46

Bioinspiration & Biomimetics

CrossMark

RECEIVED 23 July 2021

REVISED 2 December 2021

ACCEPTED FOR PUBLICATION 13 January 2022

PUBLISHED 10 March 2022

Passive concentration dynamics incorporated into the library IB2d, a two-dimensional implementation of the immersed boundary method

Matea Santiago¹, Nicholas A Battista^{2,4}, Laura A Miller¹, and Shilpa Khatri^{3,*}

- ¹ Department of Mathematics, University of Arizona, PO Box 210089, Tucson, AZ 85721, United States of America
- ² Department of Mathematics and Statistics, The College of New Jersey, 2000 Pennington Rd., Ewing, NJ 08628, United States of America
 ³ Department of Applied Mathematics, University of California Merced, 5200 North Lake Rd., Merced, CA 95343, United States of America
- * Author to whom any correspondence should be addressed.
- ⁴ http://battistn.pages.tcnj.edu.

E-mail: malvarado27@ucmerced.edu, battistn@tcnj.edu, lauram9@math.arizona.edu and skhatri3@ucmerced.edu

Keywords: advection–diffusion, immersed boundary method, fluid-structure interaction, mathematical biology, biomechanics, biofluids Supplementary material for this article is available online

Abstract

PAPER

In this paper, we present an open-source software library that can be used to numerically simulate the advection and diffusion of a chemical concentration or heat density in a viscous fluid where a moving, elastic boundary drives the fluid and acts as a source or sink. The fully-coupled fluid-structure interaction problem of an elastic boundary in a viscous fluid is solved using Peskin's immersed boundary method. The addition or removal of the concentration or heat density from the boundary is solved using an immersed boundary-like approach in which the concentration is spread from the immersed boundary to the fluid using a regularized delta function. The concentration or density over time is then described by the advection-diffusion equation and numerically solved. This functionality has been added to our software library, IB2d, which provides an easy-to-use immersed boundary method in two dimensions with full implementations in MATLAB and Python. We provide four examples that illustrate the usefulness of the method. A simple rubber band that resists stretching and absorbs and releases a chemical concentration is simulated as a first example. Complete convergence results are presented for this benchmark case. Three more biological examples are presented: (1) an oscillating row of cylinders, representative of an idealized appendage used for filter-feeding or sniffing, (2) an oscillating plate in a background flow is considered to study the case of heat dissipation in a vibrating leaf, and (3) a simplified model of a pulsing soft coral where carbon dioxide is taken up and oxygen is released as a byproduct from the moving tentacles. This method is applicable to a broad range of problems in the life sciences, including chemical sensing by antennae, heat dissipation in plants and other structures, the advection-diffusion of morphogens during development, filter-feeding by marine organisms, and the release of waste products from organisms in flows.

1. Introduction

The interactions between fluid flow and chemical concentrations or heat densities play a significant role in many biological, chemical, and engineering systems. The flow relative to an organism can be due to interactions between a moving organism and the fluid, environmental flows moving past a tethered organism, or both [1]. These flows can bring nutrients or chemical cues to the organism or remove waste, and facilitate feeding processes [2–4], reproduction [5, 6], respiration [7], photosynthesis [8, 9], the dissipation of heat [10], and signaling [11]. *In silico* models of these processes have brought insight and knowledge into the field beyond what is possible via field and experimental work [12–15]. Although a broad body of computational work has been done to study fluid-structure interactions [9, 16-20], work that couples flow with concentration dynamics in these contexts is much more limited. Recently, the authors have developed a numerical method to couple concentration dynamics with a moving structure that drives the movement of fluid and acts as a source for the chemical concentration [21]. This method was inspired by a numerical method developed by Lai et al. [22] to simulate interfacial flows with insoluble surfactants. Here, we incorporate this novel methodology for chemical or thermal dynamics into the open-source code IB2d, a two-dimensional MATLAB and Python package that implements the immersed boundary (IB) method for solving fully-coupled fluid-structure interaction (FSI) problems.

The IB method, developed by Peskin [23–25], has been used extensively to study FSI problems [26, 27]. While improved upon numerous times [26, 28–34], it is still a leading numerical framework for studying problems in FSI due to its robustness, simplicity, and flexibility in modeling complex deformable structures, such as biological tissues [35, 36]. The IB method is well suited for applications with Reynolds numbers up to around $O(10^2)$ [37]. Part of the elegance of the IB method is its ability to solve fullycoupled FSI problems involving complicated timedependent geometries, all while using a fixed Cartesian mesh to describe the fluid flow. The immersed structure is composed of elastic fibers that govern the material properties of the structures. The immersed boundaries are then discretized on a curvilinear Lagrangian mesh. The IB moves at the local fluid velocity, enforcing the no-slip condition, and the IB applies a body force to the fluid. This coupling is handled using regularized delta functions [25]. There are many ways to model the material properties of the immersed structure, such as the use of virtual springs, beams, and target points, which allow for the desired characteristics under stretching or compression and bending [33, 35, 38], the effects of porosity [35, 39, 40] and poroelasticity [36], the action of muscles [37], and forces due to coagulation and aggregation [41, 42]. One should note that the limitations of the IB method are present in this work. Depending on the application, simulations can have non-physical pressure oscillations and non-conservation of mass. The user can refine the mesh to mitigate these issues.

The IB2d project has provided the scientific community an open-source software package that contains the aforementioned material property infrastructure options, along with over 80 built-in examples illustrating their usage [35–37]. It is written with two complete high-level programming implementations, MATLAB [43] and Python [44]. Both languages were chosen for familiarity and accessibility to students, scientists, and engineers. Beyond its infrastructure for research [45–51], a significant effort has been made to develop educational materials for its use as a teaching tool. Modules have already been designed to teach concepts in foundational physics (like damped pendulums [52]), numerical analysis (e.g. convergence tests [53], interpolation [54]), and the biomechanics of locomotion (e.g. idealized anguilliform swimming [54]). Furthermore, a complementary semiautomated meshing tool was created to ease the burden of manually discretizing complex geometries that describe the immersed structures [55].

In addition to fluid-structure interactions, there are many challenging problems in the life and physical sciences that require understanding chemical and thermal dynamics. Examples at a large scale include the modeling of pollutants in urban areas [56], chemicals in marine ecology [57, 58], and contaminants in hydrogeological systems [59]. In biofluids, studies of concentration dynamics can improve our understanding of the role of fluid dynamics in nutrient transport [60], feeding [2], and chemosensing [61]. For example, numerical simulations have been used to understand mass transport in coral colonies [14] and odor capture of terrestrial and marine crabs [62]. The advection-diffusion equations have been used to model heat transfer in biological tissues [63] and chemotaxis in bacteria [64]. A variety of numerical methods have been developed to study these advection-diffusion problems, including finite difference methods [65], finite element methods [66], and meshless methods [67, 68]. Additionally, there has been work done to couple diffusion through a permeable interface in an IB framework using a flux condition [69].

Coupling the advection and diffusion of a concentration or density with the movement of an elastic structure in a fluid that acts as a source or sink is an even more challenging problem. This is due to the fact that one must solve a system of nonlinear partial differential equations with a moving and deforming elastic boundary. The methodology outlined here was specifically developed to study the photosynthesis of symbiotic algae living in the tentacles of pulsing soft corals. These algae provide the corals with most of their energy, and the pulsing behavior of the coral is believed to remove oxygen and increase the uptake of carbon dioxide. These active dynamics are thought to enhance the rate of photosynthesis up to an order of magnitude [21, 70]. Photosynthesis in soft corals is an oxygen-limited process, and this method was initially applied to a problem that considers the advection and diffusion of oxygen with an IB acting as a source. In this method, a regularized delta function, similar to that used in the IB method [25], is used to spread the concentration from the IB source to the fluid. This paper extends this methodology to other application problems and shows how it can be used more generally. Such functionality has not yet been included in other IB software libraries and has not been commonly used.

Beyond the examples presented here, there are many other application problems where the numerical simulation of a FSI coupled to a chemical concentration or heat density, where the boundary is a source or sink, can provide insights into biological processes. Examples in the biological sciences include filter-feeding, sniffing or chemical sampling for nutrient access, predator awareness, and reproduction [62, 71-74], chemotaxis of bacteria [75, 76], photosynthesis in plants [77], and heat dissipation [78]. In the medical sciences, these tools can be used to study cellular dynamics that are often regulated by chemical concentrations near the boundary [79] or for understanding surfactants and the role they play in lung development [80]. These are just a few examples of how ubiquitous such processes are in the natural world around us. Clearly, the availability of an open-source software package that can model the advection and diffusion of a chemical or heat from a moving boundary would find immediate application to various research problems. Moreover, the creators of IB2d have prioritized ease of usage for researchers and students.

2. Mathematical model

This section introduces the two-dimensional IB method coupled with the modeling of a chemical or heat advecting with the flow and diffusing at a given rate that can be adsorbed or desorbed by the elastic IB. Note that the exact method outlined below is the version implemented in the IB2d software. All domain boundary conditions are set to be periodic in this software. For a full review of the IB method, see Peskin [25]. For more details of the IB implementation in IB2d, see [35, 36]. For a treatise of the novel methodology implemented here for the concentration or heat density, see [21]. Note here that we assume that the IB is neutrally buoyant unless otherwise stated. Additionally, the chemical concentration or heat density is also assumed to be neutrally buoyant.

2.1. Governing equations for the immersed boundary and fluid flow

The two-dimensional incompressible Navier–Stokes equations that describe the conservation of momentum and mass of a viscous fluid are solved for the fluid velocity, $\mathbf{u}(\mathbf{x}, t)$ and the pressure, $p(\mathbf{x}, t)$,

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} (\mathbf{x}, t) + (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) \right]$$

= $-\nabla p(\mathbf{x}, t) + \mu \Delta \mathbf{u}(\mathbf{x}, t) + \mathbf{f}(\mathbf{x}, t), \quad (1)$
 $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0, \quad (2)$

where $\mathbf{f}(\mathbf{x}, t)$ is the force per unit area applied to the fluid by the IB, and ρ and μ are the fluid density and dynamic viscosity, respectively. The independent

variables are the time t and the position \mathbf{x} . The variables \mathbf{u} , p, and \mathbf{f} are all defined in an Eulerian frame of reference on the fixed Cartesian mesh, \mathbf{x} .

The interaction equations, which handle the coupling between the Eulerian fluid grid and the curvilinear mesh describing the IB in a Lagrangian frame of reference, are given by the following two integral equations,

$$\mathbf{f}(\mathbf{x},t) = \int \mathbf{F}(s,t)\delta\left(\mathbf{x} - \mathbf{X}(s,t)\right) \mathrm{d}s,$$
(3)

$$\frac{\partial X}{\partial t}(s,t) = \mathbf{U}(s,t) = \int \mathbf{u}(\mathbf{x},t)\delta\left(\mathbf{x} - \mathbf{X}(s,t)\right) d\mathbf{x},$$
(4)

where $\mathbf{F}(s, t)$ is the force per unit length applied by the boundary to the fluid as a function of the material point labeled by the Lagrangian parameter, s, and time, t, $\mathbf{X}(s, t)$ is the Cartesian coordinates at time t of the material point labeled by s, and U(s, t) is the velocity of the boundary as a function of t and s. Each integral transformation uses the two-dimensional Dirac delta function kernel, $\delta (\mathbf{x} - \mathbf{X}(s, t))$, to transform between Lagrangian variables and Eulerian variables. Equation (3) applies the force density from the IB to the fluid through the external forcing term in equation (1), f(x, t). Equation (4) moves the IB at the local fluid velocity, $\mathbf{u}(\mathbf{x}, t)$. This enforces a noslip boundary condition. The manner in which the force density along the IB, $\mathbf{F}(s, t)$, is computed in the integrand of equation (3), is specific to the application and can include elastic forces along the boundary, penalty forces to prescribe the motion of the boundary, damping forces, and many more. Details of these forces and their derivations are given in [35, 36, 38]. The specific form that $\mathbf{F}(s, t)$ takes is described below in the corresponding section for that example.

2.2. Governing equations for the chemical or thermal dynamics

The advection-diffusion equation coupled with the immersed-boundary equations, equations (1)-(4), is used to model the chemical concentration or heat density,

$$\begin{aligned} \frac{\partial c}{\partial t}(\mathbf{x},t) + \mathbf{u}(\mathbf{x},t) \cdot \nabla c(\mathbf{x},t) \\ &= D\Delta c(\mathbf{x},t) + \int G(s,t)\delta\left(\mathbf{x} - \mathbf{X}(s,t)\right) \mathrm{d}s, \end{aligned}$$
(5)

where $c(\mathbf{x}, t)$ is the concentration or density in the Eulerian frame of reference and *D* is the diffusion coefficient of the concentration or the thermal conductivity in the fluid. The concentration or density, $c(\mathbf{x}, t)$, is solved for with periodic boundary conditions in the domain. The elastic boundary can either adsorb or desorb the chemical or heat, and this is

modeled in an immersed-boundary fashion via the last term in equation (5). The Dirac delta function, $\delta (\mathbf{x} - \mathbf{X}(s, t))$, is once again used to couple this adsorption or desorption model, G(s, t), defined in the Lagrangian frame of reference with the concentration or density solved on the Eulerian grid. This approach is based on the modeling by Chen and Lai for surfactants [81] and further developed by the authors for studying the photosynthesis of symbiotic algae on pulsing corals [21]. This model, equation (5), can also be used with other methods, including sharp interface methods.

The IB2d software allows for multiple concentrations and densities to be coupled with the IB simultaneously, each modeled using equation (5). Each concentration or density has its own corresponding diffusion coefficient and adsorption or desorption model. However, in the formulation given here, we present only one concentration or density for the sake of simplicity.

Additionally, the IB2d software allows for a userdefined function to be inputted for the model of adsorption or desorption of the chemical or heat by the elastic boundary, G(s, t). The framework allows for this user-defined function to use the concentration or density interpolated to the boundary,

$$C(s,t) = \int c(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))d\mathbf{x}.$$
 (6)

Here, we present two models for the adsorption or desorption of the chemical or heat by the elastic boundary. In the first model, the boundary adsorbs or desorbs a constant amount that is not dependent on how much chemical or heat is already present,

$$G(s,t) = \alpha, \tag{7}$$

where α can be positive or negative and is the amount of chemical or heat adsorbed or desorbed, respectively, per length of the boundary per unit of time. The second model is limiting, a more realistic model for most situations where the amount adsorbed or desorbed is dependent on how much chemical or heat is present locally,

$$G(s,t) = \beta C(s,t), \tag{8}$$

$$G(s,t) = \beta(C_{\infty} - C(s,t)).$$
(9)

In the examples below, equation (8) is the model for adsorption and equation (9) is for desorption, except for the coral example given in section 4.3, where further details about the models are provided. Here, β is a negative parameter in equation (8) and a positive parameter in equation (9) and may be considered as a velocity that multiplies the concentration or density in the Lagrangian frame of reference in two dimensions.

2.3. Nondimensional parameters

The governing equations presented in this section have been given in dimensional form, and the implementation in IB2d is also dimensional. It is useful when conducting simulations to consider the nondimensional parameters governing this system. Two nondimensional parameters are sufficient to describe this system with an additional parameter in the adsorption or desorption model (the nondimensional number corresponding to α or β). We will be using the Reynolds number, the ratio of inertial forces to viscous forces for the fluid flow, which is derived from the nondimensionalization of equations (1) and (2), and the Péclet number, the ratio of advective to diffusive forces, which is derived from the nondimensionalization of equation (5). One could also consider the Schmidt number instead, the ratio of the kinematic viscosity to the mass diffusivity. We choose to focus on the Péclet number, as in the examples presented in section 3, we will keep the Reynolds number fixed.

3. Numerical methods

3.1. Numerical methods for the governing equations

Equations (1)–(4) are numerically solved for the velocity and pressure of the fluid, the position of the boundary, and the force of the boundary acting on the fluid. We solve at time t^{n+1} using data from time t^n using the following steps [25]:

- (a) Compute the force density, Fⁿ_k on the discretized IB Lagrangian mesh, using the current boundary configuration, Xⁿ_k. This is dependent on the application, and details are provided below for the examples included here.
- (b) Use equation (3) to spread this boundary force, Fⁿ_k, from the Lagrangian boundary mesh to the Eulerian fluid grid, x_{ij}, solving for fⁿ_{ij}.
- (c) Solve the Navier–Stokes equations, equations (1) and (2), on the Eulerian grid using the fast Fourier transform. We are solving for uⁿ⁺¹_{ij} and pⁿ⁺¹_{ij} using uⁿ_{ij}, pⁿ_{ij}, and fⁿ_{ij}.
- (d) Update the material positions, $\mathbf{X}_{k}^{n+1} = \mathbf{X}_{k}^{n} + \Delta t \mathbf{U}_{k}^{n}$, using the local fluid velocities, \mathbf{U}_{k}^{n+1} , interpolated from \mathbf{u}_{ij}^{n+1} using equation (4).

A regularized and discretized delta function is used as the kernel in the interaction equations, equations (3) and (4). This makes the IB method relatively easy to implement and flexible since one does not have to consider if and how the Lagrangian boundary points coincide with the Eulerian fluid grid points. We use the regularized delta function given in [25],

$$\delta_h(\mathbf{x}) = \frac{1}{h^2} \phi\left(\frac{x}{h}\right) \phi\left(\frac{y}{h}\right), \qquad (10)$$

where $\mathbf{x} = (x, y)$ and $\phi(r)$ is defined as,

$$\phi(r) = \begin{cases} \frac{1}{8}(3-2|r|+\sqrt{1+4|r|-4r^2}), & 0 \leq |r| < 1\\ \frac{1}{8}(5-2|r|+\sqrt{-7+12|r|-4r^2}), 1 \leq |r| < 2\\ 0 & 2 \leq |r|. \end{cases}$$

Details of the discretization and the implementation of IB2d are given in the appendices of Battista *et al* 2015 [37] and Battista *et al* 2017 [35]. Further details of these steps of the IB can also be found in [25].

In these prior works, equation (3) was discretized for an open boundary in space at time t^n using the composite trapezoidal rule,

$$\int \mathbf{F}(s,t^{n})\delta\left(\mathbf{x}-\mathbf{X}(s,t^{n})\right)\mathrm{d}s \approx \frac{1}{2}\mathbf{F}_{1}^{n}\delta_{h}(\mathbf{x}_{ij}-\mathbf{X}_{1}^{n})\Delta s_{1}$$
$$+\frac{1}{2}\sum_{k=2}^{N-1}\mathbf{F}_{k}^{n}\delta_{h}(\mathbf{x}_{ij}-\mathbf{X}_{k}^{n})(\Delta s_{k-1}+\Delta s_{k})$$
$$+\frac{1}{2}\mathbf{F}_{N}^{n}\delta_{h}(\mathbf{x}_{ij}-\mathbf{X}_{N}^{n})\Delta s_{N-1}, \qquad (12)$$

where *N* is the number of points on the Lagrangian boundary mesh. In this formulation, $\Delta s_k = ||\mathbf{X}_{k+1} - \mathbf{X}_k||_2$ is a fixed Lagrangian spacing of the Lagrangian boundary at rest provided by the user. This is a good approximation when the boundary undergoes minimal stretching or compression. This discretization is modified appropriately in the case of a periodic boundary.

We have now modified the discretization of equation (3) in step 2 above to be more accurate when the boundary undergoes considerable stretching and compression and more consistent with the discretization of the integral in equation (5), described below,

$$\int \mathbf{F}(s, t^{n}) \delta\left(\mathbf{x} - \mathbf{X}(s, t^{n})\right) \mathrm{d}s \approx \frac{1}{2} \mathbf{F}_{1}^{n} \delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{1}^{n}) \Delta \tilde{s}_{1}^{n}$$

$$+ \frac{1}{2} \sum_{k=2}^{N-1} \mathbf{F}_{k}^{n} \delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{k}^{n}) (\Delta \tilde{s}_{k-1}^{n} + \Delta \tilde{s}_{k}^{n})$$

$$+ \frac{1}{2} \mathbf{F}_{N}^{n} \delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{N}^{n}) \Delta \tilde{s}_{N-1}^{n}, \qquad (13)$$

where $\Delta \tilde{s}_k^n = \|\mathbf{X}_{k+1}^n - \mathbf{X}_k^n\|_2$ is the mesh width for the Lagrangian boundary, updated in time. A convergence study was performed for varying grid resolutions with this discretization. This data is provided in appendix A for the rubber band example presented in section 3.3.

The focus of this work is the inclusion of the concentration and thermal dynamics in the IB2d library. Equation (5) is discretized using a forward Euler method in time. The velocity at time t^n , solved for using equations (1)–(4), is used to advance the chemical concentration or heat density on the same Eulerian grid as the fluid velocity and pressure, \mathbf{x}_{ij} . The diffusion term, $D\Delta c(\mathbf{x}, t)$, is discretized using standard centered differences. For the spatial discretization of the advection term, $\mathbf{u}^n(\mathbf{x}, t) \cdot \nabla c(\mathbf{x}, t)$, two options are provided for the user, either a first-order upwind method or a third-order weighted essentially non-oscillatory (WENO) scheme [82]. The option of the WENO scheme is provided to allow for more accuracy for the advection term in advection-dominated simulations.

For the last term that handles the adsorption and desorption of the chemical or heat, a similar method as given in equation (13) is used. At time t^n , for an open boundary this term is discretized as,

$$\int G(s, t^{n})\delta\left(\mathbf{x} - \mathbf{X}(s, t^{n})\right) \mathrm{d}s \approx \frac{1}{2}G_{1}^{n}\delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{1}^{n})\Delta\tilde{s}_{1}^{n}$$
$$+ \frac{1}{2}\sum_{k=2}^{N-1}G_{k}^{n}\delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{k}^{n})(\Delta\tilde{s}_{k-1}^{n} + \Delta\tilde{s}_{k}^{n})$$
$$+ \frac{1}{2}G_{N}^{n}\delta_{h}(\mathbf{x}_{ij} - \mathbf{X}_{N}^{n})\Delta\tilde{s}_{N-1}^{n}.$$
(14)

For the integral over the fluid grid, equation (6), a standard two-dimensional trapezoidal rule is used, as was used for equation (4).

3.2. Workflow

In Battista *et al.* [35], details of the IB workflow are provided for IB2d, along with additional capabilities of its library. The authors give details of the full structure of the library there. In this section, we focus on the workflow and how to implement the chemical or thermal dynamics in IB2d. IB2d can be downloaded at www.github.com/nickabattista/IB2d.

The numerical methods for the chemical concentration or heat density are implemented in the IBM_Blackbox folder, and as for the IB, the user should not need to modify these methods unless developing new numerical techniques. The user should rather make modifications in the example subdirectory they are working with, including adding a new problem or example. Several files are modified and/or added to incorporate the concentration or thermal dynamics.

The input2d file must be modified in the example subdirectory. This file is responsible for setting all parameters for the simulation. The necessary parameters needed to include chemical or thermal dynamics in the simulation are shown and described in figure 1. Note here that the number of concentration or densities being modeled is included and the corresponding diffusion coefficients are given in a vector. The number of elements of this vector must match the number of concentrations or densities being tracked.

Files including the initial conditions for the concentration or density must be included as well. Each file should be called *struct.concentration_X* if the immersed structure is called *struct* (i.e. there is already a *struct.vertex* file in the example subdirectory) where X is the concentration or density number. Each file



includes a matrix that is the initial concentration values on the Eulerian grid. This matrix should be $N_y \times N_x$, where N_x and N_y are the number of points on the Eulerian fluid grid in the *x*-direction and the *y*-direction, respectively, defined in the input2d file.

Additionally, the user-defined adsorption or desorption model is given in the file $give_me_source_model.m$ in the example directory. The inputs for the function are the array of concentrations or densities interpolated to the boundary and a parameter indicating which concentration or density is being updated. The output gives the adsorption or desorption model, G(s, t).

Finally, due to the new discretization implemented for equation (3) and the discretization implemented for equations (5), (13) and (14), respectively, a file including the connections between IB points must be included. This file should be called *struct.geo_connect*. This file includes a matrix, where each row gives the material point labels of the points on the Lagrangian mesh that are connected to one another.

3.3. Rubber band example

First, we present the standard IB example of a rubber band initially stretched relaxing to a circle, coupled with concentration dynamics. This example will serve to illustrate differences between different concentration dynamics functionality. Adjacent points along the rubber band are tethered to each other by springs. In this case, the force density applied by the Lagrangian boundary mesh to the fluid grid is defined as,

$$\mathbf{F}_{k}^{n} = k_{s}(\mathbf{X}_{k+1}^{n} - \mathbf{X}_{k}^{n}) - k_{s}(\mathbf{X}_{k}^{n} - \mathbf{X}_{k-1}^{n}), \qquad (15)$$

where k_s is the spring constant. The rubber band is initialized as an ellipse with a major axis of length 0.8 m aligned with the *y*-axis and a minor axis of length 0.4 m aligned with the *x*-axis centered in the domain. The fluid is initialized at rest in a 1 × 1 m² domain, and the flow develops from the motion of the elastic boundary. The density of the fluid is set to $\rho = 1 \text{ kg m}^{-3}$, and the viscosity of the fluid is set to $\mu = 0.01 \text{ N s m}^{-2}$.

Note that the fluid flow will stay the same across all examples presented in this section, only the concentration dynamics are varied. The fluid flow itself is not coupled to the concentration dynamics, i.e. there is only a one-way coupling. The spatial Eulerian grid for the velocity, pressure, and concentration is 512 × 512. Below we will notate this as $N_x = N_y = 512$ and refer to the spatial grid size as h = 1/512. The number of discretized points on the Lagrangian mesh is N = 1820. The spring constant, dependent on the resolutions of the Eulerian grid and Lagrangian mesh, is $k_s = 6.4 \times 10^6$ kg m s⁻². The time step is 3.125×10^{-5} s for most examples shown below except when the CFL condition for the concentration dynamics dictates otherwise, and such cases are noted. The physical and numerical parameters are provided in table 1. Snapshots of the resulting fluid flow and vorticity up to final time t = 2 s are presented in figure 2.

The Reynolds number in these examples is computed as

$$\operatorname{Re}_{\infty} = \frac{\rho L U_{\infty}}{\mu} \tag{16}$$

where the velocity scale, $U_{\infty} = 4.228 \text{ m s}^{-1}$, is the maximum velocity in space and time during the simulation, and the length scale, $L = \sqrt{(0.2)(0.4)}$ m is the radius of the equivalent circle. This maximum velocity was computed using data from a simulation saved every 0.01 s. This results in a Reynolds number $\text{Re}_{\infty} \approx 120$.

We now present several cases with varying chemical concentration models, solving equations (5) and (6), coupled with the fluid flow from the IB simulation presented in figure 2. In all cases presented, the elastic boundary is either adsorbing or desorbing a chemical. One could also consider these examples to be modeling some other quantity of interest, such as heat. In these examples, the advection term in equation (5) is handled by the first-order upwind scheme. Each simulation is presented until a final time of t = 2 s. Note that this is long enough that the effects of the periodic boundary conditions on the domain can be observed in some simulations.

The first model we present for the concentration dynamics is the constant model, equation (7), in figure 3. In this example, the diffusion coefficient is set to $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, and the amount of chemical desorbed from the elastic boundary is $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$. We observe that during the simulation, the chemical is emitted from the boundary and diffuses into the domain.

Table 1. Numerical and physical parameters for the rubber band examples.

Parameter	Value	Units
Domain size (D)	1.0 imes 1.0	m ²
Initial rubber band major axis	0.8	m
Initial rubber band minor axis	0.4	m
Equivalent radius of rubber band (L)	$\sqrt{(0.4)(0.2)}$	m
Fluid density (ρ)	1.0	$kg m^{-3}$
Fluid viscosity (μ)	0.01	$kg(m^{-1}s^{-1})$
Diffusivity (D)	0.001-0.1	$m^2 s^{-1}$
Maximum velocity (U_{∞})	4.23	${ m m~s^{-1}}$
Time step (d <i>t</i>)	$7.8125 imes 10^{-6} - 3.125 imes 10^{-5}$	s
Spatial grid size (<i>h</i>)	1/1024-1/64	m
Number of Lagrangian points (N)	249-3969	
Spring stiffness (k_s)	$6.1/(0.5h)^2$	$kg m s^{-2}$
Chemical desorbed in constant model (α)	0.1	$kg (m^{-1} s^{-1})$
Velocity of adsorption/desorption in limiting models (β)	0.1, -0.1	${ m m~s^{-1}}$
Maximum concentration in limiting model (C_{∞})	1.0	${\rm kg}~{\rm m}^{-2}$



Figure 2. Velocity and vorticity simulation of an initially stretched rubber band for the first two seconds. The vectors present the velocity field (m s^{-1}), and the color provides the vorticity (1/s). Note the time between panels is not equal. See supplemental movie 1 (https://stacks.iop.org/BB/17/036003/mmedia).



The Péclet number in these examples,

$$\mathrm{Pe}_{\infty} = \frac{LU_{\infty}}{D},\tag{17}$$

is computed using the same velocity scale and length scale as in the Reynolds number, $U_{\infty} = 4.228 \text{ m s}^{-1}$ and $L = \sqrt{(0.2)(0.4)}$ m. The diffusion coefficient varies among the simulations presented. In figure 3, the Péclet number is approximately $Pe_{\infty} \approx 120$. In figures 4 and 5, we present simulations where we have varied the diffusion coefficient and the amount of chemical desorbed from the boundary in comparison to the previous simulation to give a sense of these parameters in the concentration dynamics. In figure 4, the diffusion coefficient is now $D = 0.1 \text{ m}^2 \text{ s}^{-1}$, and the amount of chemical desorbed is the same as before, $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$. In this simulation, due to the CFL condition for the concentration dynamics, the time step is 7.8125×10^{-6} s. This smaller time step results in a very slightly different $U_{\infty} = 4.233$ m s⁻¹, resulting in a negligible difference in the Reynolds number. In this case, the Péclet number is approximately $Pe_{\infty} \approx 12$, which is an order of magnitude smaller. We observe that in this case, the chemical almost instantaneously diffuses into the entire domain, and we do not see a build-up around the boundary even though the chemical is being desorbed from the boundary, as in the simulation presented in figure 3.

In figure 5, the amount of chemical desorbed is modified to $\alpha = 0.2 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$, while the diffusion coefficient remains the same, $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, and therefore the Péclet number also remains the same, $\text{Pe}_{\infty} \approx 120$. Note that we have chosen to keep the concentration limits in the visualization the same as in the previous two simulations to allow for comparisons, from 0 to 1 kg m⁻². However, so much chemical is being desorbed that the build-up near the boundary allows for values larger than 1 kg m⁻². This simulation exemplifies how this model may not be realistic as usually a chemical is not desorbed without limitations. The next example presents a more realistic model in these types of situations.

It is more likely that the amount of chemical desorbed from a boundary will be dependent on the amount of that chemical present in the fluid locally, and there is a maximum amount of chemical concentration possible in the fluid, C_{∞} . Figure 6 presents a simulation with such a model for the chemical concentration, equation (9). In this simulation, the diffusion coefficient is set to $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, the velocity that multiplies the concentration in the Lagrangian frame is $\beta = 0.1 \text{ m s}^{-1}$, and the maximum possible concentration at any point in the fluid is $C_{\infty} = 1 \text{ kg m}^{-2}$. In comparison to the simulation in figure 3, the boundary desorbs less chemical as it is limited by the chemical already present in the fluid surrounding the boundary.

The last example in this section presents how adsorption is also included in the software package. In figure 7, we present an example of a rubber band adsorbing a chemical from the surrounding fluid. Here, the concentration was initialized in the domain as a constant, $c(\mathbf{x}, t) = 1$ kg m⁻², and the diffusion coefficient of the chemical in the fluid was set to $D = 0.01 \text{ m}^2 \text{ s}^{-1}$. This model used for the adsorption depends on the amount of chemical present locally, equation (8), where the velocity that multiplies the concentration in the Lagrangian frame was set to $\beta = -0.1 \text{ m s}^{-1}$. In this simulation, the chemical is adsorbed from the fluid locally by the boundary, and then more chemical diffuses towards the boundary. As a result, the depletion region widens as the simulation progresses.

3.4. Higher order advection scheme

If the Péclet number is large and the system is advection-dominated, a more accurate scheme for the advection may be necessary. For these cases, we implemented a third-order WENO scheme for the advection term in equation (5), as discussed in section 3.1. This WENO scheme takes the weighted average of all finite difference stencils for an upwind third-order approximation for the first derivatives of the concentration or density. The weights are chosen to be inversely proportional to the magnitude of the derivatives so that larger magnitude derivative approximations are given a smaller weight, preventing spurious oscillations from occurring. Details of this method can be found in [82]. In this section, we present results for this situation.

Figure 8 shows the results of the constant model, equation (7), using both the upwind and WENO advection schemes. The parameters are set such that the diffusion coefficient is $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ or 0.001 m² s⁻¹ and the amount of chemical desorbed from the elastic boundary is $\alpha = 0.1$ kg (m⁻¹ s⁻¹). For this simulation, the Péclet number is $Pe_{\infty} \approx$ 120 or 1200, respectively, corresponding to the two diffusion coefficients. Note that in these advectiondominated situations, we can observe the benefits of using a more accurate method for the advection term, even though it is not necessary. For this example, the other numerical and physical parameters are the same as in the above examples, provided in table 1. Note, however, that the grid resolution was set to $N_x = N_y = 1024$, and an appropriate time step of $dt = 1.5625 \times 10^{-5}$ s was chosen to ensure numerical stability. Furthermore, N and k_s were correspondingly modified to N = 3969 and $k_s = 6.1/(0.5h)^2$ kg m s⁻², where h = 1/1024. Note here that k_s increases in value with the increased refinement.

We present the total mass of the chemical in the domain and the error in the total mass versus time for both advection schemes. Details of how these are computed are provided in section 3.5. We observe that the total mass is conserved much better, up to approximately two orders of magnitude, when using the WENO scheme. Convergence results of these methods with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ and $\text{Pe}_{\infty} \approx 120$ are provided in section 3.5.

3.5. Convergence

In this section, we present convergence results of the chemical dynamics for the rubber band example. This example was selected for investigating convergence properties due to the Lagrangian structure undergoing large deformations. The associated convergence properties of the rubber band itself, the Lagrangian IB, and the underlying fluid flow, the Eulerian data, are provided in appendix A.













In this section we focus on the convergence of the concentration dynamics, for cases with both the constant desorption model, equation (7), and the limited desorption model, equation (9), with a diffusion coefficient of $D = 0.01 \text{ m}^2 \text{ s}^{-1}$. We calculate the L^2 and L^{∞} errors in the concentration between successive grid resolutions (see equations (A.1) and (A.2) in appendix A for more details) and also present

the total mass conservation properties. We present these convergence results for both advection schemes, upwind and WENO, discussed in section 3.1. In all the results shown here, we provide the Eulerian grid size, $N_x = N_y$ or $h = 1/N_x$. The number of discretized points on the Lagrangian mesh was chosen such that the spacing between nodes initially for the rubber band was 0.5*h*, and the time step was set to



Figure 8. Total mass and the error in the total mass (kg) when modeling desorption of a chemical concentration from the boundary using the constant model, equation (7), where *D* was either 0.01 or 0.001 m² s⁻¹ and $\alpha = 0.1$ kg (m⁻¹ s⁻¹). The Péclet number in this simulation is approximately 120 and 1200, respectively. In these simulations, the advection term in equation (5) is solved using both an upwind and a WENO method.

 $dt = 3.125 \times 10^{-5}$ s, although to maintain stability in the case of $N_x = N_y = 1024$, the timestep was changed to $dt = 1.5625 \times 10^{-5}$ s. Spring stiffnesses were also scaled with the resolution, $k_s = 6.1/(0.5h)^2$ kg m s⁻². The wall-clock time for the simulations at the coarsest refinement was about ten minutes and about ten days for the most refined simulations.

For cases of constant desorption, the total mass can be computed exactly via equations (5) and (7),

$$M(t) = \int_0^t \int_\Omega \alpha \, \delta \left(\mathbf{x} - \mathbf{X}(s, \tau) \right) \, \mathrm{d}s \, \mathrm{d}\tau$$
$$= \alpha \int_0^t L(\tau) \, \mathrm{d}\tau, \tag{18}$$

where there is no chemical in the domain initially, and L(t) is the length of the IB at time *t*. Equation (18) was computed numerically using the trapezoidal rule, where the length of the boundary was evaluated with a linear piecewise approximation using the location of the Lagrangian points at time *t*. The total mass in the domain during the simulation can be computed as

$$M_h^n = \int_{\Omega} c_h^n(\mathbf{x}) \mathrm{d}\mathbf{x} \approx \sum_{ij} c_{ij}^n h^2, \qquad (19)$$

and hence the total mass error at time t^n is

$$\left|M(t^{n})-M_{h}^{n}\right|.$$
 (20)

The convergence data corresponding to the constant desorption model with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ and $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$, presented in figure 3, is shown in figure 9 and is given in table 2. Over time the convergence rates of the L^2 and L^{∞} errors in the concentration are approximately the same for both the upwind and WENO schemes. They are found to be first-order, as expected. However, as discussed above in section 3.4, the WENO scheme better conserved the overall total mass in the domain, see figure 10. Similar trends were observed for other parameter values.

For the limited desorption model, we found similar results as in the constant desorption case. The L^2 and L^∞ errors in the chemical concentration were comparable, see figure 11. The WENO advection scheme has better mass conservation results than the upwind scheme, see figure 12. These results correspond to the case where $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, $\beta = 0.1 \text{ m s}^{-1}$, and $C_{\infty} = 1 \text{ kg m}^{-2}$, as in figure 6. Note that in these limited desorption simulations, the exact mass in the domain cannot be computed, as in equation (18) for the constant desorption case. Instead, the mass conservation error was computed by taking the difference between the total mass at a resolution and the next finest resolution, e.g. the error in the total mass for the case with resolution $N_x = N_y =$ 512 is $|M_{1/512}^n - M_{1/1\ 024}^n|$. Explicit convergence rates computed at t = 2.0 s are given in table 3. We observe first-order convergence for the chemical concentration, as expected. We observe that the upwind scheme results in non-monotonic trends in the total mass error and believe that this is due to both using consecutive resolutions to compute the error and needing finer resolutions to observe the trend. The WENO scheme better conserves the overall total mass in the domain and converges at a higher rate.

4. Examples

In this section, we present three examples where the modeling of chemical or thermal dynamics is useful in elucidating the biological processes occurring. These examples can be found in the IB2d directory Example_Concentration_Dynamics.

4.1. Flow between cylinders

Arrays of bristles, hairs, and other cylindrically shaped structures are often used in the natural world for



Figure 9. The (a) L^2 error and (b) L^{∞} error between successive grid resolutions ($N_x = N_y$) for the chemical concentration in the case of the constant desorption model, equation (7), for the rubber band example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ and $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$ over the course of the simulation. Both advection schemes, upwind (U) and WENO (W) are presented. (c) Convergence plot of the concentration errors for both advection schemes at time t = 2.0 s.

Table 2. The errors and the order of convergence for the chemical concentration for the IB2d upwind (top) and WENO (bottom) advection solvers in the case of the constant desorption model, equation (7), for the rubberband example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ and $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$. The errors were computed at time t = 2.0 s.

N_x	$h\left(1/N_{x} ight)$	$\ c_h - c_{h/2}\ _2$	Order	$\ c_h - c_{h/2}\ _{\infty}$	Order	$\left \int_{\Omega} c_h \mathrm{d}\underline{x} - \int_{\Omega} c \mathrm{d}\underline{x}\right $	Order
			IJr	wind			
128	7.8125×10^{-3}	1.83×10^{-2}	0.68	6.19×10^{-2}	0.50	8.47×10^{-3}	0.90
256	3.9062×10^{-3}	1.10×10^{-2}	0.72	4.14×10^{-2}	0.58	4.41×10^{-3}	0.94
512	1.9531×10^{-3}	6.11×10^{-3}	0.86	2.04×10^{-2}	1.02	2.26×10^{-3}	0.96
1024	$9.7655 imes10^{-4}$	—	_	—	_	$1.15 imes 10^{-3}$	0.97
			W	ENO			
128	7.8125×10^{-3}	$1.79 imes 10^{-2}$	0.69	5.70×10^{-2}	0.58	$6.78 imes 10^{-3}$	1.14
256	3.9062×10^{-3}	$1.09 imes 10^{-2}$	0.72	$3.85 imes 10^{-2}$	0.57	$3.03 imes10^{-3}$	1.16
512	$1.9531 imes 10^{-3}$	$5.99 imes 10^{-3}$	0.86	$1.88 imes 10^{-2}$	1.03	$1.42 imes 10^{-4}$	1.10
1024	$9.7655 imes 10^{-4}$	—	_	_	—	$7.03 imes10^{-5}$	1.01



Figure 10. The errors in the total mass in the case of the constant desorption model, equation (7), for the rubber band example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$ and $\alpha = 0.1 \text{ kg} (\text{m}^{-1} \text{ s}^{-1})$ for the (a) upwind and (b) WENO advection schemes, for a variety of grid resolutions ($N_x = N_y$) over the course of the simulation. (c) Convergence plot of the total mass errors for both advection schemes at time t = 2.0 s.

filter feeding, sniffing, and to alter boundary layers and, subsequently, exchange [62, 83–85]. At intermediate Reynolds numbers, the ratio of the gap between the cylinders to their diameter can alter the flow through the cylindrical arrays in a highly nonlinear way, and organisms have a variety of ways to take advantage of this effect [86–89]. These dynamics become particularly interesting when considering the associated Péclet number as small changes in velocity and spacing, which the animal can control, can greatly change the amount of chemical that is taken up or carried away.

In this example, a row of five cylinders moves back and forth in a fluid. The surface of the cylinders absorbs the chemical, simulating an uptake process such as feeding or sniffing. The physical parameters



Figure 11. The (a) L^2 error and (b) L^∞ error between successive grid resolutions ($N_x = N_y$) for the chemical concentration in the case of the limited desorption model, equation (9), for the rubber band example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, $\beta = 0.1 \text{ m} \text{ s}^{-1}$, and $C_\infty = 1 \text{ kg m}^{-2}$ over the course of the simulation. Both advection schemes, upwind (U) and WENO (W) are presented. (c) Convergence plot of the concentration errors for both advection schemes at time t = 2.0 s.



Figure 12. The errors in the total mass in the case of the limited desorption model, equation (9), for the rubber band example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, $\beta = 0.1 \text{ m} \text{ s}^{-1}$, and $C_{\infty} = 1 \text{ kg m}^{-2}$ for the (a) upwind and (b) WENO advection schemes, for a variety of grid resolutions ($N_x = N_y$) over the course of the simulation. (c) Convergence plot of the total mass errors for both advection schemes at time t = 2.0.

Table 3. The errors and the order of convergence for the chemical concentration for the IB2d upwind (top) and WENO (bottom) advection solvers in the case of the limited desorption model, equation (9), for the rubber band example with $D = 0.01 \text{ m}^2 \text{ s}^{-1}$, $\beta = 0.1 \text{ m} \text{ s}^{-1}$, and $C_{\infty} = 1 \text{ kg m}^{-2}$. The errors were computed at time t = 2.0 s.

N_x	$h\left(1/N_x\right)$	$\ c_h - c_{h/2}\ _2$	Order	$\ c_h - c_{h/2}\ _{\infty}$	Order	$\left \int_{\Omega} c_h \mathrm{d}\underline{x} - \int_{\Omega} c_{h/2} \mathrm{d}\underline{x}\right $	Order
Upwi	nd						
64	1.5625×10^{-2}	1.63×10^{-2}	1.86	$4.40 imes 10^{-2}$	2.15	$1.09 imes10^{-3}$	1.48
128	$7.8125 imes 10^{-3}$	$1.00 imes 10^{-2}$	0.71	$2.78 imes 10^{-2}$	0.66	$8.06 imes10^{-5}$	3.76
256	3.9062×10^{-3}	$5.96 imes 10^{-3}$	0.75	$1.83 imes 10^{-2}$	0.61	$3.17 imes10^{-4}$	-1.98
512	1.9531×10^{-3}	$3.28 imes 10^{-3}$	0.86	$8.78 imes10^{-3}$	1.06	$2.62 imes 10^{-4}$	0.27
WEN	0						
64	$1.5625 imes 10^{-2}$	$1.80 imes 10^{-2}$	1.83	$4.51 imes 10^{-2}$	2.24	$3.45 imes 10^{-3}$	1.43
128	$7.8125 imes 10^{-3}$	$1.05 imes 10^{-2}$	0.78	$2.69 imes 10^{-2}$	0.75	$1.03 imes 10^{-3}$	1.74
256	3.9062×10^{-3}	$6.09 imes 10^{-3}$	0.78	$1.76 imes 10^{-2}$	0.62	$2.08 imes10^{-4}$	2.31
512	1.9531×10^{-3}	3.32×10^{-3}	0.87	$8.46 imes 10^{-3}$	1.05	$4.96 imes 10^{-6}$	5.39

used are representative of a dynamically scaled physical model that is immersed in a highly viscous fluid, similar to experiments and simulations that have been performed for sniffing of marine blue crabs, *Callinectes sapidus*, in water, see the downstroke data in table 2 in [62]. In this simulation, the fluid domain is a 1.0 × 1.0 m² box with a density $\rho = 1000$ kg m⁻³ and a dynamic viscosity $\mu = 48$ kg (m⁻¹ s⁻¹). The cylinders of diameter 0.04 m move from position x = 0.05 m to x = 0.95 m in 0.5 s such that the average velocity of the row of cylinders is U = 1.8 m s⁻¹, making the cylinder based Reynolds number Re_c = 1.5. The diffusivity is set to $D = 7.2 \times 10^{-5}$ m² s⁻¹ such that the Péclet number is Pe = 1000, half of that as in [62]. We assume an initial thin chemical filament with a width ten times that of the diameter of the cylinders, see SI of [62], $c(\mathbf{x}, 0) = e^{-175(x-0.3)^2}$ kg m⁻².

The cylinders model the chemosensory hairs of the marine blue crabs that take up chemicals to sniff the environment. The prescribed motion of the hairs is accomplished through the use of target points. In this formulation, each Lagrangian point of the IB hair is tethered to an individual target point. When the target point position is dynamically updated during the simulation, it pulls the Lagrangian point with it. This action is implemented by applying a force proportional to the distance between the location of the Lagrangian point and its preferred position, in very much the same way as a virtual spring,

$$\mathbf{F}_{\text{target},k}^{n} = k_{\text{target}} \left(\mathbf{Y}_{k}^{n} - \mathbf{X}_{k}^{n} \right), \qquad (21)$$

where k_{target} is the stiffness coefficient and \mathbf{X}_k^n and \mathbf{Y}_k^n are the position of the *k*th Lagrangian point and the prescribed position of its target point at time t^n , respectively. Furthermore, to maintain the circular shape of the hairs throughout all of the motion, virtual springs and beams are used. Springs tether adjacent Lagrangian nodes within each hair as well as across each hair (circle), as in the case of the rubber band example, equation (15). These springs were given high spring stiffnesses in order to preserve the desired spacing between nodes.

Virtual non-invariant beams preserved the desired circular curvature of each hair. Non-invariant beams were used as the hairs did not rotate but only exhibited translational movement. The continuous deformation force of a beam was computed at time t^n ,

$$\mathbf{F}_{\text{beam}}^{n} = k_{\text{beam}} \frac{\partial^{4}}{\partial s^{4}} \left(\mathbf{X}^{n} - \mathbf{X}^{\text{con}} \right), \qquad (22)$$

where k_{beam} is the beam stiffness, \mathbf{X}^n is the position of a Lagrangian point along the hair at time t^n , and $\frac{\partial^4}{\partial s^4}(\mathbf{X}^{\text{con}})$ is its corresponding preferred curvature. Details regarding the 4th-order derivative discretization of equation (22) are provided in [36]. The force per unit length applied by the boundary to the fluid, **F**, is then the sum of these three forces from the target points, springs, and beams.

We use the model given in equation (8) with $\beta = -1 \text{ m s}^{-1}$ to model the uptake by the hairs. In this example, we choose to use the high-order advection scheme described in section 3.4 as the Péclet number is large. All physical and numerical parameters are provided in table 4.

In figure 13, we present the results of this simulation. Observe that the chemosensory hairs create flow that mixes the chemical filament as the animal takes up the chemical. In figure 13(b), we present the total mass of the chemical in the domain throughout the duration of the simulation. Notice that there are periods where the chemical is taken up when the hairs are moving through the filament and periods where much less chemical is taken up. The slight increase in the total mass at a few time points is within the numerical error of this simulation.

4.2. Flow past a plate

The example in this section is inspired by airflow over a broad leaf at low Reynolds numbers. It has long been established that many broad leaves reconfigure and reduce flutter in strong flows [90, 91]. There may be situations, however, in which flutter is desirable in light winds. Examples include the enhancement of canopy light penetration via the fluttering of poplar leaves [92] or for the purpose of convective and evaporative heat transfer [93, 94]. A particularly interesting case is the tulip poplar leaf that flutters more than a similarly shaped flexible cut out at low wind speeds while reducing flutter and drag at high wind speeds [95]. In the absence of passive movement, the shape of a stationary leaf itself has been shown to enhance heat dissipation [10, 96]. It may also be the case that some leaf shapes enhance passive movement through augmented vortex shedding, thereby increasing waste removal or nutrient uptake through increased mixing dynamics.

In this example, we model airflow and cooling of a leaf by the release of sensible heat [97] using a flexible flat plate. To enhance vortex formation over the plate for the purpose of this example, a cylinder is attached to the leading edge of the plate, which is similar to the petiole of a leaf. The petiole (cylinder) is held nearly rigid by target points. Springs and beams are also used to help further minimize geometric perturbations.

The plate (leaf) is given additional mass, and the upwind end of the plate is tethered using target points as in equation (21). Springs and beams are also placed between adjoining points to model the plate's flexibility. The additional mass is modeled using a formulation similar to target points, where a massive point is tethered to a Lagrangian point. If the force acting on the Lagrangian point is large enough, it will cause the Lagrangian point to drag the massive point along with it. Mathematically, this is described in the following way

$$\mathbf{F}_{\mathrm{mass}}^{n} = k_{\mathrm{mass}}(\mathbf{Y}_{k}^{n} - \mathbf{X}_{k}^{n})$$
(23)

$$M(k)\frac{\partial^2 \mathbf{Y}_k^n}{\partial t^2} = -\mathbf{F}_{\text{mass}}^n$$
(24)

where k_{mass} is a stiffness coefficient with $k_{\text{mass}} \gg 1$, M(k) is the mass density of the *k*th massive point, \mathbf{Y}_k^n , and \mathbf{X}_k^n and is the position of the *k*th Lagrangian point at time t^n .

The density and viscosity of the fluid are set to that of air [98], and a 25 cm s⁻¹ background flow is generated using an applied external force. Details of the mathematical methodology used to implement these forces for the IB plate can be found in [35].

The physical and numerical parameters for this example are given in table 5. The plate is 1 cm in length, and the fluid domain is a 30×5 cm² box. The plate is a source of the sensible heat with a flux

Table 4. Numerical and physical parameters for the example of flow between moving cylinders.

Parameter	Value	Units
Domain size (D)	1.0×1.0	m
Cylinder diameter (<i>d</i>)	0.04	m
Fluid density (ρ)	1000	$kg m^{-3}$
Fluid viscosity (μ)	48	$kg(m^{-1}s^{-1})$
Diffusivity (D)	$7.2 imes 10^{-5}$	$m^2 s^{-1}$
Average velocity (U)	1.8	${\rm m}~{\rm s}^{-1}$
Time step (dt)	$3.125 imes 10^{-6}$	s
Spatial grid size (<i>h</i>)	1/1024	m
Number of Lagrangian points per hair (N)	257	_
Spring stiffness (k_s)	$1 imes 10^{6}$	$kg m s^{-2}$
Target stiffness (k_{target})	$5 imes 10^6$	$kg m s^{-2}$
Beam stiffness (k_{beam})	$1.0 imes 10^9$	$kg m s^{-2}$
Velocity of adsorption in limiting models (β)	-1	$m s^{-1}$



of $\alpha = 204\,400$ g cm s⁻³ in the constant desorption model, equation (7). This value is based on parameters for modeling the sensible heat flux of a twodimensional leaf at 35 degrees Celsius in air that is at 31.5 degrees Celsius, with total leaf conductance of 2×10^{-4} mol (cm² s⁻¹) and specific heat of the moist air of 29.2 J (mol⁻¹ K⁻¹), see equation (10.5) and figure 10.3 in [97], resulting in a flux of $0.020\,44\,\mathrm{W}\,\mathrm{m}^{-2}$. We are only modeling the initial few seconds of the leaf releasing heat and not considering the temperature of the leaf or the air changing. The thermal diffusivity is set for air at 30 degrees Celsius [99]. The Reynolds number using the background flow, the length of the leaf, and the density and viscosity of the fluid is approximately $\text{Re} \approx 151$ and the Péclet number using the thermal diffusivity is approximately Pe \approx 109.

In figure 14, we present a time series of the simulation of the leaf flutter and the release of sensible heat. We observe that since the leaf and fluid start

at rest, it takes some time for the flutter to develop, three to four seconds. Initially, without much flow, the released heat builds up around the leaf, around time 1.5 s, but once there is significant flow, the heat is advected further away from the leaf.

4.3. Pulsing corals

The methodology presented in this work and incorporated into the IB2d software was initially developed by the authors to study pulsing soft corals [21]. Some species of sessile soft corals in the family Xeniidae pulse their tentacles. These corals live in colonies and are one of the few animals that have an energetically expensive behavior for a purpose other than locomotion. It is hypothesized that the pulsing behavior enhances the photosynthesis of their symbiotic algae, which provides the corals with most of their energy [70]. Fluid flow has also been shown to enhance photosynthesis in other marine benthic autotrophs such as stony corals, sea grass, and algae [8].

Parameter	Value	Units
Domain size (D)	30×5	cm ²
Plate length (<i>L</i>)	1.0	cm
Fluid density (ρ)	$1.145 imes 10^{-3}$	g cm ⁻³
Fluid viscosity (μ)	$1.895 imes 10^{-4}$	$g(cm^{-1} s^{-1})$
Inflow velocity (U)	25.0	${\rm cm}~{\rm s}^{-1}$
Thermal diffusivity (D)	0.23	$\mathrm{cm}^2 \mathrm{s}^{-1}$
Time step (dt)	$5 imes 10^{-5}$	s
Spatial grid size (<i>h</i>)	5/256	cm
Number of Lagrangian point (N)	103	_
Massive points (<i>m</i>)	$5 imes 10^{-4}$	g
Spring stiffness (k_{spring})	$2.5 imes10^4$	$g \text{ cm s}^{-2}$
Target stiffness (k_{target})	$2.5 imes10^4$	g cm s ⁻²
Beam stiffness (k_{beam})	$2.0 imes10^8$	g cm s ⁻²
Heat density desorbed in constant model (α)	204 400	$\rm g~cm~s^{-3}$



In this example, we simulate the motion of a pulsing soft coral polyp coupled with a model of the photosynthesis of the algae. During photosynthesis, the algae uptake carbon dioxide and release the byproduct, dissolved oxygen. We model the photosynthesis by tracking the concentration of oxygen, c_1 , and carbon dioxide, c_2 . The motion of the tentacles is prescribed using target points as in equation (21). The tentacles absorb the carbon dioxide according to the model $G(s, t) = -\beta C_2$ and desorb the oxygen according to the model $G(s, t) = \beta C_2$, where C_2 is the carbon dioxide present locally around the tentacles, given by equation (6).

The physical and numerical parameters for this example simulating a coral polyp pulsing in the ocean are given in table 6. The length of a coral tentacle is 0.407 cm and the polyp length, including the base, is 0.920 cm. The fluid domain is a $2.0 \times 5.0 \text{ cm}^2$ box with a density of $\rho = 1~{\rm g~cm^{-3}}$ and a dynamic viscosity of $\mu = 0.01$ g (cm⁻¹ s⁻¹). The period of the coral pulse is 1.89 s and is composed of contraction, expansion, and resting phases. The frequency-based Reynolds number is defined using the tentacle length, frequency of the pulsing, and viscosity and is $Re_f = 8.76$ and the frequency-based Péclet number is $Pe_f = 100$. In figure 15, we present the results of the simulation. The fluid started at rest with no oxygen in the domain and an initial carbon dioxide concentration of $c_2(\mathbf{x}, 0) = 10^{-6} \text{ mol cm}^{-2}$. The simulation was run until time t = 18.9 s, ten pulses of the coral. We present the results during the tenth pulse here. See supplemental movies 9 and 10

Table 6. Numerical and physical parameters for the example of pulsing corals.

Parameter	Value	Units
Domain size (D)	2.0×5.0	cm
Tentacle length (<i>L</i>)	0.407	cm
Fluid density (ρ)	1	g cm ⁻³
Fluid viscosity (μ)	0.01	$g(cm^{-1}s^{-1})$
Pulsing period (<i>T</i>)	1.89	S
Diffusivity of oxygen and carbon dioxide in water (<i>D</i>)	$8.75 imes10^{-4}$	$\mathrm{cm}^2 \mathrm{s}^{-1}$
Timestep (dt)	$2.5 imes10^{-4}$	S
Spatial grid size (<i>h</i>)	1/128	cm
Number of Lagrangian points (N)	235	_
Target stiffness (k_{target})	$8 imes 10^5$	$kg m s^{-2}$
Velocity of adsorbtion and desorption (β)	0.0215	${\rm cm}~{\rm s}^{-1}$



to see the simulation in its entirety, including all ten pulses. Observe that the fluid flow allows more carbon dioxide to reach the coral and therefore allows more photosynthesis to occur and more oxygen byproduct to be created. More results and analysis of this biological system were provided in [21].

5. Discussion and conclusion

This paper presents a new methodology to incorporate the advection and diffusion of heat or a chemical with a moving IB acting as a source or sink. Furthermore, the boundary is fully coupled to the fluid in that it moves at the local fluid velocity and applies a body force to the fluid. The methods are developed within the IB framework, a commonly used method for simulating fluid-structure interactions. The methodology supports solving for multiple concentrations, representing a heat distribution or chemical concentrations and userdefined source or sink models at the boundary. This new methodology was initially developed to study pulsing corals [21] and has been generalized here in the software library IB2d for application to diverse problems in biological fluid dynamics at intermediate scales. This tool should find broad use as many natural and engineered systems have fluid-structure interactions coupled with chemical or thermal dynamics and there are very few tools and software libraries available for the numerical simulations of these systems.

IB2d is a two-dimensional IB library designed to lower the steep learning curve necessary to conduct numerical simulations. Therefore, it is not only useful to the scientific community for research endeavors but can also be used in educational settings. Before this project, the package included over 80 IB examples that included multiple ways to describe the elastic boundary, including the use of springs, beams, tether points, masses, and porous interfaces. We have now included all of the examples described in this paper that model the advection and diffusion of heat and chemical concentrations. This will allow users to develop their own simulations with relative ease.

Future work consists of two different avenues: (1) extending the numerical methods to three dimensions and (2) improving the implementation and options in the two-dimensional methods within IB2d. Work is currently ongoing in both directions. The extension to three dimensions is mathematically straightforward, similar to the IB method. The authors plan to incorporate the method into the software library IBAMR [33]. In terms of method development, a major improvement would be the incorporation of implicit time stepping. The time step is already restricted in the IB due to the explicit methods used for the Navier-Stokes equations. By making the methodology implicit for equation (5), we would reduce the time step restrictions from two different ones to just the restriction from the fluid solver. This is especially relevant in the case of larger diffusion coefficients or smaller Péclet numbers as in the rubber band example with $Pe \approx 12$, see figure 4, where the restriction from the advection-diffusion equation required a smaller time step than it did for the fluid solver. Furthermore, we can extend the work here to take into account how the concentration interacts with the fluid more generally. This could be included in various forms, including through a reaction term in the bulk for a concentration or density and by having the fluid parameters depend on the amount of chemical or heat present. As the software library has more usage, we believe these extensions will be incorporated as they are needed for various applications.

Acknowledgments

M Santiago and S Khatri were supported by the National Science Foundation: PHY-1505061 and

DMS-1853608. The computational resources used by M Santiago and S Khatri were also funded by the National Science Foundation: ACI-1429783. M Santiago had further funding from the National Science Foundation training Grant: DMS-1840265 and the UC Merced graduate dean dissertation fellowship. Computational resources for N A Battista were provided by the National Science Foundation: OAC-1826915 and OAC-1828163. Funding for N A Battista was provided by the TCNJ Support of Scholarly Activity (SOSA) Grant, the TCNJ Department of Mathematics and Statistics, and the TCNJ School of Science. Funding for L A Miller was provided by the National Science Foundation: DMS-2111765 and CBET-2114309.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/nickabattista/IB2d.

Appendix A. Immersed boundary and fluid flow convergence for the rubber band example

As the force interpolation functionality was modified from the original IB2d implementation [35-37], we present convergence properties for both the Lagrangian IB and Eulerian fluid data for the rubber band example of section 3.3 and figure 2. The modifications to the force interpolation functionality were described in section 3.1. In the simulations presented herein, the Eulerian fluid grid resolution was varied and appropriate modifications were made to the Lagrangian rubber band structure based on each particular resolution. For each resolution, we selected a different number of nodes on the initial configuration of the stretched rubber band, the Lagrangian mesh, such that once in its circular their spacing satisfied 0.5*h*. state, Spring stiffnesses were also scaled appropriately, i.e. $k_{\rm s} = 6.1/(0.5h)^2$ kg m s⁻². The time step was also modified based on the resolution such that $dt = 3.125 \times 10^{-5}$ s for $N_x = 32, 64, 128, 256$ and 512, but was reduced to $dt = 1.5625 \times 10^{-5}$ s for $N_x = 1024$ to satisfy numerical stability requirements.

As the resolution increased, i.e. $N_x = N_y$ increased, the rubber band more accurately conserved its area over time, as shown in figure 16. The exact area is known based on the original area of the ellipse, $A = \pi ab = \pi (0.4)(0.8)$ m². The approximate area of the ellipse at each time point saved in the simulation was found via Green's theorem and







Table 7. Convergence properties of the rubber band, the immersed Lagrangian structure using the updated IB2d force spreading, equation (13). The errors were computed at time t = 2.0 s.

N _x	$h = 1/N_x$	$\left \operatorname{ecc}_{h} - \operatorname{ecc}_{h/2}\right $	Order	$ A_h - A $	Order
128	7.8125×10^{-3}	2.32×10^{-2}	2.17	$1.76 imes 10^{-2}$	0.68
256	3.9062×10^{-3}	8.96×10^{-3}	1.37	$1.01 imes 10^{-2}$	0.81
512	1.9531×10^{-3}	$2.18 imes 10^{-3}$	2.04	$5.44 imes 10^{-3}$	0.89
1024	$9.7655 imes 10^{-4}$	_	—	$2.82 imes 10^{-3}$	0.94

then using the trapezoidal rule to approximate the integral,

$$\begin{split} A_{h}^{n} &= \oint X^{n} \, \mathrm{d} Y \approx \left[\frac{1}{2} \sum_{k=1}^{N-1} \left[X_{k}^{n} + X_{k+1}^{n} \right] \left(Y_{k+1}^{n} - Y_{k}^{n} \right) \right. \\ &+ \frac{1}{2} \left[X_{N}^{n} + X_{1}^{n} \right] \left(Y_{N}^{n} - Y_{1}^{n} \right), \end{split}$$

where A_h^n is the approximate area at time t^n and $\mathbf{X}_k^n = (X_k^n, Y_k^n)$. The error in the area is computed

as the absolute value of the difference between the known exact area, A, and the successive approximations, A_h^n , at each time step, $|A - A_h^n|$.

Figure 16 illustrates the area conservation convergence properties, both over time (figure 16(a)) and as a convergence plot (figure 16(b)). The error in figure 16(b) was calculated at time t = 2.0 s. Moreover, varying grid resolutions also led to slightly perturbed dynamics in each case, as figure 17 illustrates.





Table 8. Convergence results of the fluid flow, specifically in the horizontal component of the velocity, $u \text{ (m s}^{-1})$, in IB2d using the updated force spreading functionality, as described in section 3.1, in the rubber band example. The errors were computed at time t = 2.0 s.

N _x	$h = 1/N_x$	$ u_h - u_{h/2} _2$	Order	$\ u_h-u_{h/2}\ _{\infty}$	Order
64 128	1.5625×10^{-2} 7.8125 × 10^{-3}	3.92×10^{-2} 2.00 × 10^{-2}	1.93	2.64×10^{-1} 1.90 × 10^{-1}	1.50
256	3.9062×10^{-3}	1.11×10^{-2}	0.85	1.00×10^{-1} 1.17×10^{-1}	0.70
512	1.9531×10^{-3}	$4.92 imes 10^{-3}$	1.17	$5.59 imes 10^{-2}$	1.06

Figure 17 provides the eccentricity of the rubber band over time for varying grid resolutions. The eccentricity is defined as,

$$\operatorname{ecc}_{h}^{n} = \sqrt{1 - \frac{\min\{r_{1}^{n}, r_{2}^{n}\}}{\max\{r_{1}^{n}, r_{2}^{n}\}}},$$

where r_1^n and r_2^n are the semi-major and semi-minor axes of the rubber band over time. As the rubber band oscillates over time, the major and minor axes of the ellipse switch. The error and convergence rates for both the area conservation and eccentricity are given in table 7, at time t = 2.0 s.

The L^2 and L^∞ errors in the Eulerian data were also computed. Since the exact fluid dynamics are not known, in order to calculate errors, the successive grid resolutions were compared and then the select metric was computed, i.e. for the quantity *f*,

$$\|f_h - f_{h/2}\|_2 \approx \sqrt{\sum_{ij} \left(\left(f_h\right)_{ij} - \left(f_{h/2}\right)_{ij} \right)^2 h^2}$$
 (A.1)

$$\|f_h - f_{h/2}\|_{\infty} = \max_{ij} \left| (f_h)_{ij} - (f_{h/2})_{ij} \right|.$$
 (A.2)

The higher resolved grids were subsampled, so that data for the same spatial locations were being compared in equations (A.1) and (A.2).

Figures 18(a) and (b) provide the L^2 and L^{∞} errors in the horizontal component of the velocity over time, *u*. The errors for the vertical component look very similar and are omitted. Figure 18(c) provides a convergence plot for both velocity components and table 8 provides the convergence rates in the L^2 and L^{∞} errors for the horizontal component of velocity. We observe first-order convergence, as expected.

ORCID iDs

Matea Santiago bhttps://orcid.org/0000-0002-4706-7240

Nicholas A Battista D https://orcid.org/0000-0003-2437-0383

Laura A Miller b https://orcid.org/0000-0003-3707-9798

Shilpa Khatri b https://orcid.org/0000-0002-0848-5925

References

- Vogel S 1994 Life in Moving Fluids: The Physical Biology of Flow—Revised and Expanded 2nd edn (Princeton, NJ: Princeton University Press) http://jstor.org/stable/j. ctvzsmfc6
- [2] Michelin S and Lauga E 2011 Optimal feeding is optimal swimming for all Péclet numbers *Phys. Fluids* 23 101901
- [3] Van Wassenbergh S and Aerts P 2008 Rapid pivot feeding in pipefish: flow effects on prey and evaluation of simple dynamic modelling via computational fluid dynamics *J. R. Soc. Interface* 5 1291–301
- [4] Santhanakrishnan A, Dollinger M, Hamlet C L, Colin S P and Miller L A 2012 Flow structure and transport characteristics of feeding and exchange currents generated by upside-down cassiopea jellyfish *J. Exp. Biol.* 215 2369–81
- [5] Jikeli J F et al Sperm navigation along helical paths in 3D chemoattractant landscapes *Nat. Commun.* 6 7985
- [6] Zimmer R K and Riffell J A 2011 Sperm chemotaxis, fluid shear, and the evolution of sexual reproduction *Proc. Natl Acad. Sci. USA* 108 13200–5

- [7] Grotberg J B 2001 Respiratory fluid mechanics and transport processes Annu. Rev. Biomed. Eng. 3 421–57
- [8] Mass T, Genin A, Shavit U, Grinstein M and Tchernov D 2010 Flow enhances photosynthesis in marine benthic autotrophs by increasing the efflux of oxygen from the organism to the water *Proc. Natl Acad. Sci.* 107 2527–31
- [9] Bitog J P, Lee I-B, Lee C-G, Kim K-S, Hwang H-S, Hong S-W, Seo I-H, Kwon K-S and Mostafa E 2011 Application of computational fluid dynamics for modeling and designing photobioreactors for microalgae production: a review *Comput. Electron. Agric.* 76 131–47
- [10] Vogel S 1970 Convective cooling at low airspeeds and the shapes of broad leaves J. Exp. Bot. 21 91–101
- [11] Seymour J R, Simó R, Ahmed T and Stocker R 2010 Chemoattraction to dimethylsulfoniopropionate throughout the marine microbial food web *Science* 329 342–5
- [12] Ghalambaz M, Jamesahar E, Ismael M A and Chamkha A J 2017 Fluid-structure interaction study of natural convection heat transfer over a flexible oscillating fin in a square cavity *Int. J. Therm. Sci.* 111 256–73
- [13] Al-Amiri A and Khanafer K 2011 Fluid-structure interaction analysis of mixed convection heat transfer in a lid-driven cavity with a flexible bottom wall *Int. J. Heat Mass Transfer* 54 3826–36
- [14] Hossain M M and Staples A E 2019 Passive vortical flows enhance mass transport in the interior of a coral colony *Phys. Fluids* **31** 061701
- [15] Shapiro O H, Fernandez V I, Garren M, Guasto J S, Debaillon-Vesque F P, Kramarsky-Winter E, Vardi A and Stocker R 2014 Vortical ciliary flows actively enhance mass transport in reef corals *Proc. Natl Acad. Sci. USA* 111 13391–6
- [16] Nielsen L T, Asadzadeh S S, Dölger J, Walther J H, Kiørboe T and Andersen A 2017 Hydrodynamics of microbial filter feeding *Proc. Natl Acad. Sci. USA* 114 9373–8
- [17] Adams M P et al 2016 Feedback between sediment and light for seagrass: where is it important? *Limnol. Oceanogr.* 61 1937–55
- [18] Crimaldi J P and Zimmer R K 2014 The physics of broadcast spawning in benthic invertebrates Annu. Rev. Mar. Sci. 6 141–65
- [19] Hamlet C, Miller L A, Rodriguez T and Santhanakrishnan A 2012 The fluid dynamics of feeding in the upside-down jellyfish *Natural Locomotion in Fluids and on Surfaces* (Berlin: Springer) pp 35–51
- [20] Fauci L J and McDonald A 1984 Sperm motility in the presence of boundaries *Bull. Math. Biol.* 57 679–99
- [21] Santiago M, Mitchell K A, Khatri S 2022 Numerical method for modeling photosynthesis of algae on pulsing soft corals *Phys. Rev. Fluids* (accepted)
- [22] Lai M-C, Tseng Y-H and Huang H 2008 An immersed boundary method for interfacial flows with insoluble surfactant J. Comput. Phys. 227 7279–93
- [23] Peskin C S 1972 Flow patterns around heart valves: a numerical method J. Comput. Phys. 10 252–71
- [24] Peskin C S 1977 Numerical analysis of blood flow in the heart J. Comput. Phys. 25 220–52
- [25] Peskin C S 2002 The immersed boundary method Acta Numer. 11 479–517
- [26] Mittal R and Iaccarino G 2005 Immersed boundary methods Annu. Rev. Fluid Mech. 37 239–61
- [27] Griffith B E and Patankar N A 2020 Immersed methods for fluid-structure interaction Annu. Rev. Fluid Mech. 52 421–48
- [28] Fauci L J and Fogelson A L 1993 Truncated Newton methods and the modeling of complex immersed elastic structures *Commun. Pure Appl. Math.* 46 787–818
- [29] Lai M-C and Peskin C S 2000 An immersed boundary method with formal second-order accuracy and reduced numerical viscosity J. Comput. Phys. 160 705–19

- [30] Cortez R and Minion M 2000 The blob projection method for immersed boundary problems J. Comput. Phys. 161 428–53
- [31] Griffith B E and Peskin C S 2005 On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems *J. Comput. Phys.* 208 75–105
- [32] Griffith B E, Hornung R D, McQueen D M and Peskin C S 2007 An adaptive, formally second order accurate version of the immersed boundary method *J. Comput. Phys.* 223 10–49
- [33] Griffith B E 2014 An adaptive and distributed-memory parallel implementation of the immersed boundary (ib) method https://github.com/IBAMR/IBAMR (online accessed 21 October 2014)
- [34] Griffith B E and Luo X 2017 Hybrid finite difference/finite element immersed boundary method Int. J. Numer. Methods Biomed. Eng. 33 e2888
- [35] Battista N A, Strickland W C and Miller L A 2017 IB2d: a Python and MATLAB implementation of the immersed boundary method *Bioinspiration Biomimetics* 12 036003
- [36] Battista N A, Strickland W C, Barrett A and Miller L A 2018 IB2dReloaded: a more powerful Python and MATLAB implementation of the immersed boundary method *Math. Methods Appl. Sci.* 41 8455–80
- [37] Battista N A, Baird A J and Miller L A 2015 A mathematical model and Matlab code for muscle-fluid-structure simulations *Integr. Comp. Biol.* 55 901–11
- [38] Eyre D J and Fogelson A L 1997 IBIS: a software system for immersed boundary and interface simulations http://math.utah.edu/IBIS/
- [39] Kim Y and Peskin C S 2006 2D parachute simulation by the immersed boundary method SIAM J. Sci. Comput. 28 2294–312
- [40] Stockie J M 2009 Modelling and simulation of porous immersed boundaries *Comput. Struct.* 87 701–9
- [41] Fogelson A L 1984 A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting J. Comput. Phys. 56 111–34
- [42] Fogelson A L and Guy R D 2008 Immersed-boundary-type models of intravascular platelet aggregation *Comput. Methods Appl. Mech. Eng.* 197 2087–104
- [43] 2015 MATLAB, Version 8.5.0 (R2015a) Natick, MAThe MathWorks https://mathworks.com/matlabcentral/ answers/780292-how-to-cite-matlab www.mathworks. com/products.html
- [44] Van Rossum G 2015 *Python, Version 3.5* https://python.org http://citebay.com/how-to-cite/python/
- [45] Taheri A 2018 Lagrangian coherent structure analysis of jellyfish swimming using immersed boundary FSI simulations J. Mech. Civil Eng. 15 69–74
- [46] Miles J G and Battista N A 2019 Naut your everyday jellyfish model: exploring how tentacles and oral arms impact locomotion *Fluids* 4 169
- [47] Pallasdies F, Goedeke S, Braun W and Memmesheimer R-M 2019 From single neurons to behavior in the jellyfish Aurelia aurita eLife 8 e50084
- [48] Lai X 2019 Modeling and numerical simulations of active and passive forces using immersed boundary method MS Thesis (Mathematical Sciences) Worcester Polytechnic Institute
- [49] Battista N A 2020 Diving into a simple anguilliform swimmer's sensitivity *Integr. Comp. Biol.* 60 1236–50
- [50] Chen S, Xue J, Hu J, Ding Q, Zhou L, Feng S, Cui Y, Lü S and Long M 2020 Flow field analyses of a porous membrane-separated, double-layered microfluidic chip for cell co-culture Acta Mech. Sin. 36 754–67
- [51] Malvar S and Carmo B S 2021 Wavelet analysis of a flexible filament kinematics: emulating *C. elegans* swimming behavior *J. Braz. Soc. Mech. Sci. Eng.* 43 207
- [52] Mongelli M and Battista N A 2020 A swing of beauty: pendulums, fluids, forces, and computers *Fluids* 5 48

- [53] Battista N A and Mizuhara M S 2019 Fluid-structure interaction for the classroom: speed, accuracy, convergence, and jellyfish! (arXiv:1902.07615)
- [54] Battista N A 2021 Fluid-structure interaction for the classroom: interpolation, hearts, and swimming! SIAM *Review* 63 1
- [55] Senter D M, Douglas D R, Strickland W C, Thomas S G, Talkington A M, Miller L A and Battista N A 2020 A semi-automated finite difference mesh creation method for use with immersed boundary software IB2d and IBAMR *Bioinspiration Biomimetics* 16 016008
- [56] Egan B A and Mahoney J R 1972 Numerical modeling of advection and diffusion of urban area source pollutants J. Appl. Meteorol. 11 312–22
- [57] Faugeras B and Maury O 2007 Modeling fish population movements: from an individual-based representation to an advection–diffusion equation *J. Theor. Biol.* 247 837–48
- [58] Largier J L 2003 Considerations in estimating larval dispersal distances from oceanographic data *Ecol. Appl.* 13 71–89
- [59] van der Lee J and De Windt L 2001 Present state and future directions of modeling of geochemistry in hydrogeological systems J. Contam. Hydrol. 47 265–82
- [60] Muñoz-García J, Neufeld Z and Torney C 2010 Nutrient exposure of chemotactic organisms in small-scale turbulent flows New J. Phys. 12 103043
- [61] Weissburg M 2000 The fluid dynamical context of chemosensory behavior *Biol. Bull.* 198 188–202
- [62] Waldrop L D, Miller L A and Khatri S 2016 A tale of two antennules: the performance of crab odour-capture organs in air and water J. R. Soc. Interface 13 20160615
- [63] Khaled A-R A and Vafai K 2003 The role of porous media in modeling flow and heat transfer in biological tissues *Int. J. Heat Mass Transfer* 46 4989–5003
- [64] Deleuze Y, Chiang C-Y, Thiriet M and Sheu T W H 2016 Numerical study of plume patterns in a chemotaxis-diffusion–convection coupling system *Comput. Fluids* 126 58–70
- [65] Wang K and Wang H 2011 A fast characteristic finite difference method for fractional advection–diffusion equations Adv. Water Resour. 34 810–6
- [66] Burman E and Ern A 2007 Continuous interior penalty hp-finite element methods for advection and advection–diffusion equations *Math. Comput.* 76 1119–41
- [67] Zhuang P, Gu Y T, Liu F, Turner I and Yarlagadda P K D V 2011 Time-dependent fractional advection–diffusion equations by an implicit MLS meshless method *Int. J. Numer. Methods Eng.* 88 1346–62
- [68] Liu G and Karamanlidis D 2003 Mesh free methods: moving beyond the finite element method Appl. Mech. Rev. 56 B17–8
- [69] Huang H, Sugiyama K and Takagi S 2009 An immersed boundary method for restricted diffusion with permeable interfaces J. Comput. Phys. 228 5317–22
- [70] Kremien M, Shavit U, Mass T and Genin A 2013 Benefit of pulsation in soft corals *Proc. Natl Acad. Sci.* 110 8978–83
- [71] Waldrop L D, Reidenbach M A and Koehl M A R 2015 Flexibility of crab chemosensory sensilla enables flicking antennules to sniff *Biol. Bull.* 229 185–98
- [72] Waldrop L D, He Y and Khatri S 2018 What can computational modeling tell us about the diversity of odor-capture structures in the pancrustacea? *J. Chem. Ecol.* 44 1084–100
- [73] Olson S D 2013 Fluid dynamic model of invertebrate sperm chemotactic motility with varying calcium inputs *J. Biomech.* 46 329–37
- [74] Kiselev A and Ryzhik L 2012 Biomixing by chemotaxis and enhancement of biological reactions *Commun. PDE* 37 298–318

- [75] Dillon R, Fauci L and Gaver D III 1995 A microscale model of bacterial swimming, chemotaxis and substrate transport *J. Theor. Biol.* 177 325–40
- [76] Stocker R and Seymour J R 2012 Ecology and physics of bacterial chemotaxis in the ocean *Microbiol. Mol. Biol. Rev.* 76 792–812
- Boulard T, Roy J-C, Pouillard J-B, Fatnassi H and Grisey A 2017 Modelling of micrometeorology, canopy transpiration and photosynthesis in a closed greenhouse using computational fluid dynamics *Biosyst. Eng.* 158 110–33
- [78] Dudley P N, Bonazza R, Jones T T, Wyneken J and Porter W P 2014 Leatherbacks swimming in silico: modeling and verifying their momentum and heat balance using computational fluid dynamics PLoS One 9 e110701
- [79] Zhang S, Guy R D, Lasheras J C and del Álamo J C 2017 Self-organized mechano-chemical dynamics in amoeboid locomotion of physarumfragments J. Phys. D: Appl. Phys. 50 204004
- [80] Nkadi P O, Merritt T A and Pillers D-A M 2009 An overview of pulmonary surfactant in the neonate: genetics, metabolism, and the role of surfactant in health and disease *Mol. Genet. Metabol.* 97 95–101
- [81] Chen K-Y and Lai M-C 2014 A conservative scheme for solving coupled surface-bulk convection-diffusion equations with an application to interfacial flows with soluble surfactant *J. Comput. Phys.* 257 1–18
- [82] Liu X-D, Osher S and Chan T 1994 Weighted essentially non-oscillatory schemes J. Comput. Phys. 115 200–12
- [83] Reidenbach M A, George N and Koehl M A R 2008 Antennule morphology and flicking kinematics facilitate odor sampling by the spiny lobster, panulirus argus *J. Exp. Biol.* 211 2849–58
- [84] Koehl M A R 1996 Small-scale fluid dynamics of olfactory antennae Mar. Freshw. Behav. Physiol. 27 127–41
- [85] Strickland C, Miller L, Santhanakrishnan A, Hamlet C, Battista N and Pasour V 2017 Three-dimensional low Reynolds number flows near biological filtering and protective layers *Fluids* 2 62
- [86] Koehl M A R 1995 Fluid flow through hair-bearing appendages: feeding, smelling, and swimming at low and intermediate Reynolds number Soc. Exp. Biol. Symp. 49 157–82
- [87] Cheer A Y L and Koehl M A R 1987 Paddles and rakes: fluid flow through bristled appendages of small organisms J. *Theor. Biol.* 129 17–39
- [88] Koehl M A R 2001 Transitions in function at low Reynolds number: hair-bearing animal appendages Math. Methods Appl. Sci. 24 1523–32
- [89] Koehl M A R 2004 Biomechanics of microscopic appendages: functional shifts caused by changes in speed J. Biomech. 37 789–95
- [90] Vogel S 2006 Drag reduction by leaf aquaplaning in hexastylis (aristolochiaceae) and other plant species in floods J. North Am. Benthol. Soc. 25 2–8
- [91] Vogel S 1989 Drag and reconfiguration of broad leaves in high winds J. Exp. Bot. 40 941–8
- [92] Roden J S and Pearcy R W 1993 Effect of leaf flutter on the light environment of poplars Oecologia 93 201-7
- [93] Thom A S 1968 The exchange of momentum, mass, and heat between an artificial leaf and the airflow in a wind-tunnel Q. J. R. Meterol. Soc. 94 44–55
- [94] Grant R H 1983 The scaling of flow in vegetative structures Boundary-Layer Meteorol. 27 171–84
- [95] Miller L A, Santhanakrishnan A, Jones S, Hamlet C, Mertens K and Zhu L 2012 Reconfiguration and the reduction of vortex-induced vibrations in broad leaves *J. Exp. Biol.* 215 2716–27

- [96] Vogel S 2009 Leaves in the lowest and highest winds: temperature, force and shape New Phytol. 183 13–26
- [97] Bonan G 2019 Chapter 10: Leaf Temperature and Energy Fluxes, Climate Change and Terrestrial Ecosystem Modeling (Cambridge: Cambridge University Press) pp 152–66
- [98] Engineers Edge, LLC 2000 Viscosity of air, dynamic and kinematic https://engineersedge.com/physics/viscosity_ of_air_dynamic_and_kinematic_14483.htm (accessed 25 June 2021)
- [99] Engineering Toolbox 2018 Air-thermal diffusivity https://engineeringtoolbox.com/air-thermal-diffusivity -d_2011.html (accessed 25 June 2021)