

SPATIALLY ADAPTIVE STOCHASTIC MULTIGRID METHODS FOR FLUCTUATING HYDRODYNAMICS

PAT PLUNKETT * AND PAUL J. ATZBERGER †

Abstract. The immersed boundary method is a numerical approach for simulating elastic structures which interact with a fluid flow. In many physical systems thermal fluctuations become significant at small scales and play a fundamental role. In this paper stochastic numerical methods are developed which extend the immersed boundary approach to account for thermal fluctuations by including appropriate stochastic forcing terms in the fluid equations. The stochastic numerical methods developed in this paper differ in three significant ways from prior work: (i) The new numerical methods allow for use of non-periodic non-uniform multilevel meshes, where prior methods were only applicable to uniform periodic meshes and relied heavily on the Fourier Transform. (ii) A new stochastic closure approximation is derived for the fast dynamics of the system to handle stiff features of the stochastic equations. (iii) Methods for the generation of stochastic fields with long-range covariance structure on multilevel meshes are developed having only linear computational complexity in the number of mesh cells. These advances in addition to allowing for improved accuracy and computational efficiency also allow for new physical phenomena to be studied with the stochastic immersed boundary method. To show how the methods can be used in practice, results for an interacting particle system and a polymer system are discussed which make particular use of non-periodic boundary conditions to capture in the hydrodynamic interactions the effects of walls and fixed inclusions in the fluid.

Key words. Immersed Boundary Method, Adaptive Numerical Methods, Multigrid, Stochastic Partial Differential Equations, Multiscale Methods.

Last update to manuscript PDF was on September 26, 2012; 4:37pm.

1. Introduction. The immersed boundary method is a numerical approach for handling immersed elastic structures which interact with a fluid flow [1]. In studying many physical systems at small length scales, such as interacting particles, polymers, or membranes, thermal fluctuations often play a fundamental role in the dynamics. In [?, ?] a general approach was discussed which extends the framework of the immersed boundary method to incorporate thermal fluctuations by using fluctuation-dissipation principles from statistical mechanics. To formulate stochastic numerical methods consistent with fluctuation-dissipation principles analysis must be carried out for each type of discretization of the hydrodynamic equations to appropriately handle the particular dissipative properties of the corresponding discrete operators.

Such an approach was taken for the special case of discretizations on a uniform periodic mesh in [?]. To obtain an efficient time integration method the Fourier Transform and Ito Calculus were used to derive analytic expressions to handle inherent stiffness in the stochastically forced equations [?]. For non-uniform multilevel meshes with potentially non-periodic boundary conditions the Fourier Transform can no longer be used directly.

In this paper we show how numerical methods can be formulated for non-periodic multilevel meshes. To obtain an efficient time integration method a new stochastic closure approximation is derived for the fast dynamics of the fluid-structure system. Stochastic numerical methods are then developed for the fluid-structure system which achieve a linear computational complexity in the number of mesh cells per time step. To achieve this complexity a method is developed which allows for the generation of stochastic fields, having potentially long-range covariance structure, on multilevel

*University of California, Department of Mathematics

†University of California, Department of Mathematics, Santa Barbara, CA 93106; e-mail: atzberg@math.ucsb.edu; phone: 805-893-3239; Work supported by NSF Grant DMS-0635535.

meshes with linear computational complexity. The presented methods permit the mesh to have selective resolution having more refined mesh cells on regions of interest, while permitting coarser mesh cells on regions of less interest, for example areas corresponding to the decay of large-scale hydrodynamic tails. The new methods also allow for general boundary conditions to be incorporated on regions of the mesh. To show how the methods can be used in practice simulation results for interacting particle and polymer systems are discussed which make particular use of non-periodic boundary conditions to capture in the hydrodynamic interactions the effect of walls and fixed inclusions in the fluid.

2. Immersed Boundary Method for Time Dependent Stokes Flows. For a fluid flow modeled by the time dependent Stokes equations the stochastic immersed boundary method is given by:

$$(2.1) \quad \rho \frac{\partial \mathbf{u}(\mathbf{x}, t)}{\partial t} = \mu \Delta \mathbf{u}(\mathbf{x}, t) - \nabla p(\mathbf{x}, t) + \mathbf{f}_S(\mathbf{x}, t) + \mathbf{f}_T(\mathbf{x}, t)$$

$$(2.2) \quad \nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$$

$$(2.3) \quad \frac{d\mathbf{X}^{[j]}}{dt} = \int \delta_a(\mathbf{y} - \mathbf{X}^{[j]}) \mathbf{u}(\mathbf{y}, t) dy$$

$$(2.4) \quad \mathbf{f}_S(\mathbf{x}, t) = \mathbf{F}^{[j]} \delta_a(\mathbf{x} - \mathbf{X}^{[j]})$$

The term p is the pressure, ρ is the fluid density, μ is the dynamic viscosity. The force density \mathbf{f}_S accounts for momentum transferred to the fluid by elastic deformations of immersed structures. The force density \mathbf{f}_T is a Gaussian random field δ -correlated in time which accounts for the thermal fluctuations of the fluid-structure system. Structures are modeled by M control points $\mathbf{X}^{[j]}$ along with a force interaction law. The force acting on the j^{th} control point is denoted by $\mathbf{F}^{[j]}(\{\mathbf{X}^{[j']}\})$. The structure dynamics are given by equation 2.3, which corresponds to advection of the control points with the local fluid velocity. The term $\delta_a(\mathbf{x})$ approximates the Dirac δ -function. In the immersed boundary method the δ_a functions are taken so that they integrate to one and vanish outside a disk of radius a . This gives a brief formulation of the equations of the immersed boundary method, for a more detailed discussion see [1].

3. Thermal Fluctuations. We now discuss the extension of the conventional immersed boundary method to account for thermal fluctuations. In [?] it was shown that in order for the model to be consistent with the principles of statistical mechanics only the fluid degrees of freedom should be stochastically forced. To account for thermal fluctuations of the system an appropriate choice must be made for the stochastic forcing \mathbf{f}_T . It will be assumed throughout that the stochastic field \mathbf{f}_T is Gaussian and δ -correlated in time. Consequently, the stochastic field is completely determined by its spatial mean and covariance. To avoid an overly technical discussion and to achieve our ultimate goal of formulating numerical methods, we shall discuss only the case in which the stochastic differential equations have been spatially discretized by finite differencing on a finite mesh. For this purpose, let L denote the finite difference approximation of the Laplacian Δ . The spatial covariance structure of the equilibrium fluctuations of the fluid can then be expressed in terms of the matrix $C = \langle \mathbf{u} \mathbf{u}^T \rangle$ and similarly the covariance of the stochastic forcing is given by $G = \langle \mathbf{f}_T \mathbf{f}_T^T \rangle$. In the notation, the composite vector consisting of all values on the mesh for the fluid velocity field and stochastic force field are denoted, respectively, by \mathbf{u} and \mathbf{f}_T . The $\langle \cdot \rangle$ denotes averaging with respect to the equilibrium probability distribution.

3.1. Fluctuation Dissipation Principle for Semi-Discretized Numerical Methods. We now derive general expressions for the covariance structure of the stochastic forcing which is required in order for the dynamics of a semi-discretized system to achieve a given covariance structure for the equilibrium fluctuations. Let $C_t = \langle \mathbf{u}(\mathbf{x}, t) \mathbf{u}^T(\mathbf{y}, t) \rangle$ denote the covariance of the fluctuations of the field at time t corresponding to the stochastic dynamics of the semi-discretized system

$$(3.1) \quad d\mathbf{u} = L\mathbf{u}dt + Qd\mathbf{B}(t)$$

and let $G = QQ^T$. From Ito's Lemma we have

$$(3.2) \quad dC_t = (LC_t + C_tL^T + G) dt$$

which for $dC_t = 0$ at steady-state requires

$$(3.3) \quad G = -(LC + CL^T).$$

This is referred to as the *fluctuation-dissipation principle* in statistical mechanics. The expression yields a relation between the equilibrium fluctuations of the system C , the dissipative mechanism of the dynamics L , and the stochastic forcing G . This gives a general expression for how a semi-discretized system should be stochastically forced to achieve a specified covariance for the equilibrium fluctuations.

For the stochastic immersed boundary method, the equilibrium fluctuations of the fluid should have Gibbs-Boltzmann statistics with probability density $\Psi(\mathbf{u}) = \exp(-E[\mathbf{u}]/k_B T)$. To make use of these relations an energy must be specified for the discrete system. Taking for the energy of the system the kinetic energy integrated over each mesh cell we have $E[\mathbf{u}] = \sum_{\mathbf{m}} \rho |\mathbf{u}_{\mathbf{m}}|^2 \Delta x_{\mathbf{m}}^d$, where $\mathbf{u}_{\mathbf{m}}$ is the fluid velocity and $\Delta x_{\mathbf{m}}^d$ is the volume of the mesh cell with index \mathbf{m} and where d is the spatial dimension of the physical system. Under the Gibbs-Boltzmann statistics the covariance matrix for the equilibrium fluctuations is given by

$$(3.4) \quad C_{\mathbf{m},\mathbf{m}} = (k_B T / \rho \Delta x_{\mathbf{m}}^3) I,$$

where $C_{\mathbf{m},\mathbf{m}}$ denotes the $d \times d$ submatrix of C along the diagonal and I denotes the identity matrix. From the Gibbs-Boltzmann statistics and the *fluctuation-dissipation principle* the Gaussian random field \mathbf{f}_T is determined for any semi-discretization of the fluid equations as $\langle \mathbf{f}_T \rangle = 0$ and $\langle \mathbf{f}_T(\mathbf{x}) \mathbf{f}_T(\mathbf{y}) \rangle = G = -(LC + CL^T)$.

4. Multilevel Meshes and Operators. The numerical methods will be formulated by discretizing the spatial derivatives of the equations 2.1, 2.3 on a non-uniform multilevel mesh. The MAC approach will be taken with values defined both at the mesh cell centers and at the mesh cell faces. In Figure ?? a typical multilevel mesh is depicted and in the zoomed in sub-figure the typical arrangement of the locations of the cell and face centered values.

We refer to faces which comprise the adjoining cells where the mesh changes from one resolution to another as the coarse-refined interface. On the coarse-refined interface there are two types of face centered values. The values associated with the refined mesh cells at the center of the refined cell faces and one value at the center of the coarse cell face. Throughout we shall work with operations on the mesh where the value at the center of the coarse cell is always taken to be the average of the values at the center of the refined mesh cellfaces.

We shall index the mesh cells by $\mathbf{m} = (m_1, m_2, m_3)$. The index of cell centered values will be index as $\mathbf{u}_{\mathbf{m}}$. The index of face centered values will be given by indices

of the form $\mathbf{m}' = (m_1 + d_1, m_2 + d_2, m_3 + d_3)$, where $d_i \in \{0, -\frac{1}{2}, \frac{1}{2}\}$ with only one index of i having $d_i \neq 0$. For example, the mesh cell with index \mathbf{m} with face centered value in the direction of the negative x-axis has index given by $(m_1 - \frac{1}{2}, m_2, m_3)$.

Following the MAC approach we shall define a gradient and divergence operator on the mesh using a combination of cell centered and face centered values (cite). On uniform regions of the mesh away from coarse-refined interfaces the divergence operator and gradient operator are defined as

$$(4.1) \quad (D\mathbf{u})_{\mathbf{m}} = \sum_{i \in \mathcal{I}} \frac{\mathbf{u}_{\mathbf{m}+\mathbf{h}_i}^{(i)} - \mathbf{u}_{\mathbf{m}-\mathbf{h}_i}^{(i)}}{\Delta x_{\mathbf{m}}}$$

$$(4.2) \quad (G\mathbf{u})_{\mathbf{m}+\mathbf{h}_i}^{(i)} = \frac{\mathbf{u}_{\mathbf{m}+\mathbf{q}_i}^{(i)} - \mathbf{u}_{\mathbf{m}-\mathbf{q}_i}^{(i)}}{\Delta x_{\mathbf{m}}}$$

where \mathbf{h}_i has component i equal to $\frac{1}{2}$ and all other components equal to zero, \mathbf{q}_i has component i equal to 1 and all other components equal to zero. The set \mathcal{I} consists of the indices corresponding to faces in the positive x-axis, y-axis, and z-axis directions only. At a coarse-refined interface the divergence operator is defined as above with the understood convention that the face centered value of the coarse mesh cell will be taken to be the average of the refined mesh cell face centered values. At a coarse-refined interface the gradient must be defined carefully to attain at least first order accuracy [?].

To avoid complications arising from all of the different cases, we shall only discuss the case when the coarse-refined interface occurs as shown in Figure ???. In this case the interface occurs for the coarse mesh cell along the face in the direction of the positive x-axis. In this case, the gradient at the faces of the coarse-refined interface is defined as

$$(4.3) \quad (G\mathbf{u})_B^{(1)} = \frac{\mathbf{u}_A^{(1)} - \frac{1}{2}(\mathbf{u}_B^{(1)} + \mathbf{u}_D^{(1)})}{\frac{3}{4}\Delta x_A}$$

$$(4.4) \quad (G\mathbf{u})_C^{(1)} = \frac{\mathbf{u}_A^{(1)} - \frac{1}{2}(\mathbf{u}_C^{(1)} + \mathbf{u}_E^{(1)})}{\frac{3}{4}\Delta x_A}$$

$$(4.5) \quad (G\mathbf{u})_A^{(1)} = \frac{1}{4} \left((G\mathbf{u})_B^{(1)} + (G\mathbf{u})_C^{(1)} + (G\mathbf{u})_D^{(1)} + (G\mathbf{u})_E^{(1)} \right)$$

with mesh cells D and E having respectively the same gradient values as B and C . The cases for the other faces involved in coarse-refined interfaces is defined similarly, see (cite).

These operators allow conveniently for a Laplacian to be defined for multilevel meshes. Letting \mathbf{u} denote the composite vector of all of the cell centered values we shall use

$$(4.6) \quad L\mathbf{u} = D G \mathbf{u}.$$

which we refer to as the ‘‘MAC Laplacian.’’ For uniform three dimensional meshes this can be shown to correspond to the discretization of the Laplacian with the standard seven-point stencil. We remark that if only central differences of the cell centered values were used to approximate the gradient and divergence on the mesh it can be shown that the corresponding Laplacian can have a non-trivial null space even

for non-periodic meshes, with each vector of the null space supported on a disjoint sub-mesh.

For the non-periodic case boundary conditions will also be required on the mesh. For the multilevel meshes the Dirichlet boundary conditions will be imposed by introducing boundary mesh cells which have a corresponding fixed cell centered value. For Neumann boundary conditions the faces in the direction of the boundary condition will have a corresponding fixed face centered value.

4.1. Multigrid Method on Multilevel Meshes. Central to the numerical methods we shall discuss for multilevel meshes is the multigrid approach which can be applied to solve linear systems of the form $A\mathbf{v} = \mathbf{b}$. The basic idea of the multigrid approach is to formulate an iterative method by using a hierarchy of meshes and associated linear systems by coarsening the original mesh. The multigrid method then uses standard iterative methods such as Gauss-Siedel iterations to relax toward a solution on each mesh in the hierarchy and combines the results to obtain an approximate solution on the original mesh. On an intuitive level, the use of multiple meshes at different levels of refinement accelerates the overall rate of convergence of the iterative method by allowing for information to be efficiently transmitted over large spatial scales of the mesh in some sense. A more precise discussion of the multigrid method can be found in (cite). We shall also discuss an interpretation of the multigrid method from the variational perspective in Section 5.4.2. The specific variant of multigrid that we shall use is referred to as “Fast Adaptive Composite Mesh Multigrid” which we shall abbreviate as FAC-multigrid. This is summarized in Algorithm 1.

A typical multigrid method consists of three components: a *restriction operator* to coarsen data from a more refined mesh to a coarser mesh of the hierarchy, a *prolongation operator* to interpolate data from a coarser mesh to a more refined mesh, and a *smoother operator* which iteratively relaxes an approximation toward a solution of the linear system on each mesh. We shall use for the prolongation operator in three dimensions the tri-linear interpolation. This allows for values on the coarse mesh at level r to be interpolated to a more refined mesh at level ℓ . We denote the prolongation by I_r^ℓ . For the restriction operator we shall use $I_\ell^r = (I_r^\ell)^T$, which has desirable variational properties as discussed in Section 5.4.2. For the smoother we shall use Gauss-Siedel iterations on the mesh which is given by

$$(4.7) \quad v_{\mathbf{m}}^{(\text{new})} = \frac{\mathbf{b}_{\mathbf{m}} - \sum_{\mathbf{k} > \mathbf{m}} A_{\mathbf{m},\mathbf{k}} v_{\mathbf{k}}^{(\text{old})} - \sum_{\mathbf{k} < \mathbf{m}} A_{\mathbf{m},\mathbf{k}} v_{\mathbf{k}}^{(\text{new})}}{A_{\mathbf{m},\mathbf{m}}}.$$

A multigrid iteration takes as input an initial guess for the solution \mathbf{v} and right-hand side \mathbf{b} of the linear system formulated for the mesh. The iteration then smooths the errors of the initial guess relative to the solution of the linear system by applying a few iterations of Gauss-Siedel. Rather than continuing on the refined mesh which usually slows down to a near stall after a few iterations, the residual $\mathbf{r} = A\tilde{\mathbf{v}} - \mathbf{b}$ is computed for the current approximate solution $\tilde{\mathbf{v}}$. The residual can be used to reformulate the problem in terms of the correction $\mathbf{e} = \mathbf{v} - \tilde{\mathbf{v}}$ for the approximation $\tilde{\mathbf{v}}$ relative to the exact solution \mathbf{v} . The exact correction satisfies the linear system $A\mathbf{e} = \mathbf{b} - A\tilde{\mathbf{v}} = \mathbf{r}$. It can be shown that for many problems when the refined mesh iterations “stall” the correction is “smooth” on the fine mesh. The restriction operator is then used to reformulate the problem on a coarser mesh by coarsening the residual values to the next coarsest mesh of the hierarchy and formulating the coarse mesh linear system $A^{(\ell-1)}\mathbf{q}^{(\ell-1)} = \mathbf{r}^{(\ell-1)}$, where $A^{(\ell-1)} = I_\ell^{\ell-1} A^{(\ell)} I_{\ell-1}^\ell$ and

$\mathbf{r}^{(\ell-1)} = I_{\ell}^{\ell-1} \mathbf{r}^{(\ell)}$. The smoother on the coarse mesh is then applied for a few iterations with initial guess $\mathbf{q} = 0$ to obtain an approximate solution $\tilde{\mathbf{q}}$ to the linear system. The multigrid idea can then be recursively applied to obtain a further collection of linear systems which correct solutions on the next coarsest mesh in the hierarchy. To obtain a solution on the original mesh the approximate solutions on each mesh must then somehow be combined.

This is accomplished for the most refined mesh using the approximate solution $\tilde{\mathbf{q}}$ found for the correction problem on the next coarsest mesh. This is applied to correct the solution on the refined mesh by interpolation to obtain $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + I_{\ell-1}^{\ell} \tilde{\mathbf{q}}$. A few applications of the smoother is then applied to the corrected solution $\tilde{\mathbf{v}}$ and the result returned. For the coarser meshes a similar correction procedure is used to transmit results down the mesh hierarchy until the most refined level is reached. This gives one iteration of the multigrid method.

For many problems this significantly accelerates convergence to the solution and only a few multigrid iterations are often required to obtain a solution having a residual comparable or smaller than the discretization error. An important feature of the multigrid method is that if the matrix A is sparse with only a constant number of non-zero entries per row, the iterations can be carried out with a computational complexity of order only M , where M is the number of mesh cells. For a more precise summary of the multigrid method see Algorithms 1, 2, and 3. A more detailed discussion of multigrid methods can be found in (cite).

Algorithm 1: $\mathbf{v} \leftarrow \text{FAC-Multigrid}(\mathbf{v}, \mathbf{b}, \nu, \mu_1, \mu_2)$.

Data: An initial guess \mathbf{v} , the right-hand-side \mathbf{b} , the number of smoother iterations (ν, μ_1, μ_2) .

Result: An approximate solution \mathbf{v} of the linear system $A\mathbf{w} = \mathbf{b}$.

Procedure:

1. Compute the residual of the initial guess $\mathbf{r} = A\mathbf{v} - \mathbf{b}$.
 2. Initialize the initial guess for the correction $\mathbf{q} \leftarrow 0$.
 3. Perform a full sweep of the mesh cells on the multilevel mesh
 $\mathbf{q} \leftarrow \text{Full-Sweep}(\mathbf{q}, \mathbf{r}, \nu, \mu_1, \mu_2)$.
 4. Correct the solution $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{q}$.
-

5. Numerical Methods for the Stochastic Immersed Boundary Method.

In the case of a uniform periodic mesh the stochastic immersed boundary method the matrices associated with discretization of the equations 2.1 was diagonalized by the Fourier transform and explicit analytic expressions were derived for the time integration of the Fourier modes. This allowed for time steps in the method which either directly or indirectly resolved contributions to the dynamics of the fluid and immersed structures over a wide range of time scales. For non-uniform or non-periodic meshes the Fourier Transform can no longer be directly applied to obtain such numerical methods. We show how a different approach can be taken to obtain stochastic numerical methods for multilevel meshes which are applicable over a wide range of time scales.

5.1. Summary of the Stochastic Numerical Methods. We shall consider two distinct regimes. The first corresponds to the regime where time steps are small relative to the time scales on which the fluid degrees of freedom relax. The second

Algorithm 2: $\mathbf{q}^{(\ell)} \leftarrow \text{Full-Sweep}(\mathbf{q}^{(\ell)}, \mathbf{r}^{(\ell)}, \nu, \mu_1, \mu_2)$.

Data: An initial guess $\mathbf{q}^{(\ell)}$, the right-hand-side $\mathbf{r}^{(\ell)}$, the number of smoother iterations (ν, μ_1, μ_2) .

Result: An approximate solution $\mathbf{q}^{(\ell)}$ of the linear system $A^{(\ell)}\mathbf{w} = \mathbf{r}^{(\ell)}$.

Procedure:

1. If the current mesh is at the level of refinement common to all meshes in the hierarchy then perform one V-Cycle on the mesh:
 $\mathbf{q}^{(\ell)} \leftarrow \text{V-Cycle}(\mathbf{q}^{(\ell)}, \mathbf{r}^{(\ell)}, \mu_1, \mu_2)$.
 2. Otherwise, coarsen the residual to the next mesh level $\mathbf{r}^{(\ell-1)} \leftarrow I_{\ell}^{\ell-1}\mathbf{r}^{(\ell)}$.
 3. Perform a full sweep of the cells of the multilevel mesh
 $\mathbf{q}^{(\ell-1)} \leftarrow \text{Full-Sweep}(\mathbf{q}^{(\ell-1)}, \mathbf{r}^{(\ell-1)}, \nu, \mu_1, \mu_2)$.
 4. Apply the correction to the solution on the current level: $\mathbf{q}^{(\ell)} \leftarrow I_{\ell-1}^{\ell}\mathbf{q}^{(\ell-1)}$.
 5. Apply the smoother with the initial guess $\mathbf{q}^{(\ell)}$ for ν iterations for the linear system $A^{(\ell)}\mathbf{w} = \mathbf{r}^{(\ell)}$.
-

Algorithm 3: $\mathbf{v}^{(\ell)} \leftarrow \text{V-Cycle}(\mathbf{v}^{(\ell)}, \mathbf{b}^{(\ell)})$.

Data: An initial guess $\mathbf{v}^{(\ell)}$, the right-hand-side $\mathbf{b}^{(\ell)}$, the number of smoother iterations (μ_1, μ_2) .

Result: An approximate solution $\mathbf{v}^{(\ell)}$ of the linear system $A^{(\ell)}\mathbf{w} = \mathbf{b}^{(\ell)}$.

Procedure:

1. Apply the smoother with the initial guess $\mathbf{v}^{(\ell)}$ for μ_1 iterations for the linear system $A^{(\ell)}\mathbf{w} = \mathbf{b}^{(\ell)}$.
 2. If the current mesh is at the coarsest level of refinement in the hierarchy then skip to step 5.
 3. Otherwise, perform a V-Cycle on the next coarsest mesh in the hierarchy.
Let $\mathbf{b}^{(\ell-1)} \leftarrow I_{\ell}^{\ell-1}(\mathbf{b}^{(\ell-1)} - A\mathbf{v}^{(\ell)})$, $\mathbf{v}^{(\ell-1)} \leftarrow 0$, then compute
 $\mathbf{v}^{(\ell-1)} \leftarrow \text{V-Cycle}(\mathbf{v}^{(\ell-1)}, \mathbf{b}^{(\ell-1)})$.
 4. Correct the solution on the current level $\mathbf{v}^{(\ell)} \leftarrow \mathbf{v}^{(\ell)} + I_{(\ell-1)}^{\ell}\mathbf{v}^{(\ell-1)}$.
 5. Apply the smoother with the initial guess $\mathbf{v}^{(\ell)}$ for μ_2 iterations for the linear system $A^{(\ell)}\mathbf{w} = \mathbf{b}^{(\ell)}$.
-

corresponds to the regime where the time step is large relative to the the time scales on which the fluid degrees of freedom relax, but small relative to the time scales on which the immersed elastic structures move appreciably. For each of the regimes distinct numerical methods will be formulated to approximate the stochastic immersed boundary equations on multilevel meshes. The first numerical method allows for the stochastic dynamics of both the fluid and immersed structures to be directly resolved in simulations. Such methods may be of interest for problems involving phenomena related to the relaxation of the fluid, such as the study of hydrodynamic memory effects. Directly resolving these dynamical features of the fluid, however, greatly restricts the time steps which can be taken. A second numerical method will be derived which only resolves directly the stochastic dynamics of the immersed structures, thereby allowing for much less restricted time steps. The method is based on a stochastic closure approximation derived to account statistically for the rapid fluctuations of the fluid over a time step. Each of the numerical methods are outlined

Algorithm 4: Stochastic IB Method (Time Dependent Stokes Flow)

Data: Numerical and physical parameters.

Result: Stochastic dynamics of the fluid-structure system.

Procedure:

1. Compute the forces acting on the immersed structures to obtain the force field \mathbf{f}_S .
 2. Generate the stochastic force field \mathbf{f}_T accounting for the thermal fluctuations of the fluid using 3.4.
 3. Perform the in-exact projection of \mathbf{u}^* using 5.4, which approximately imposes incompressibility of the fluid.
 4. Update the stochastic fluid flow \mathbf{u} .
 5. Update the configuration of the structures using the time integrated stochastic fluid flow \mathbf{u} .
 6. Return to step 1. to compute the next time step of the dynamics of the fluid and immersed structures.
-

Algorithm 5: Stochastic IB Method (Steady-State Stokes Flow)

Data: Numerical and physical parameters.

Result: Stochastic dynamics of the immersed structures.

Procedure:

1. Compute the forces acting on the immersed structures to obtain the force field \mathbf{f}_S .
 2. Compute \mathbf{U}_I the time integrated mean of the steady-state fluid flow.
 3. Compute \mathbf{u}_{II} the time integrated thermal fluctuations of the fluid flow.
 4. Perform the in-exact projection of $\mathbf{U} = \mathbf{U}_I + \mathbf{u}_{II}$ using 5.4, which approximately imposes incompressibility of the fluid.
 5. Update the configuration of the structures using the time integrated stochastic fluid flow \mathbf{U} .
 6. Return to step 1. to compute the next time step of the dynamics of the immersed structures.
-

in Algorithm 4 and 5. A more detailed discussion of their derivations is given in Section 5.2 and 5.3.

5.2. Regime I: Stochastic Numerical Methods for the Case of Time Dependent Stokes Flow. To integrate the stochastic dynamics of the fluid we shall use the following variant of the Euler-Maruyama method

$$(5.1) \quad \mathbf{u}^{*,(n+1)} = \mathbf{u}^{*,(n)} + \rho^{-1} L \mathbf{u}^{*,(n)} \Delta t + \rho^{-1} \tilde{\varphi}^T \mathbf{f}_S^{(n)} \Delta t + \rho^{-1} \mathbf{g} \sqrt{\Delta t}$$

$$(5.2) \quad \mathbf{u}^{(n+1)} = \tilde{\varphi} \mathbf{u}^{*,(n+1)}.$$

In the notation \mathbf{u}^* refers to the fluid velocity field obtained without fully accounting for the incompressibility of the fluid. The operator $\tilde{\varphi}$ refers to an "in-exact projection" which corresponds to imposing approximately the incompressibility constraint. In general, the approximate nature of the operator $\tilde{\varphi}$ poses difficulties for stochastic equations if we are interested in retaining in the numerical methods principles of statistical mechanics, such as fluctuation-dissipation. For the stochastic immersed

boundary method the scheme in equation 5.1 introduces the approximate projection appropriately to retain these features. We remark that an important aspect of the above method is that in contrast to other approaches cumulative errors from the approximate projection are only accumulated from the term involving the force density of the immersed structures. A further discussion motivating this method can be found in Appendix ??.

We shall use the specific in-exact projection operator $\tilde{\phi}$ which is obtained by approximately imposing the constraint that the discretized velocity field, when interpolated to the mesh cell faces, have zero divergence when applying D the discretized MAC divergence operator [?]. This is equivalent to solving the pressure equation

$$(5.3) \quad Lp = DI^{c \rightarrow f} \mathbf{u}$$

where $I^{c \rightarrow f}$ interpolates cell centered values to face centered values on the mesh and D is the discrete MAC divergence operator and L is the MAC Laplacian.

The in-exact projection operator is then defined in terms of the solution p by

$$(5.4) \quad \tilde{\phi} \mathbf{u} := \mathbf{u} - I^{f \rightarrow c} G p = \left(I - I^{f \rightarrow c} G L^{-1} D I^{c \rightarrow f} \right) \mathbf{u}$$

where $I^{f \rightarrow c}$ interpolates face centered values to cell centered values and G is the discrete MAC gradient operator. The projection is referred to as in-exact since the cell centered field only satisfies approximately the condition $DI^{c \rightarrow f} \mathbf{u} = 0$ to be divergence free when $\mathbf{u} = \tilde{\phi} \mathbf{u}^*$. As already mentioned, in contrast to the projection method the linear operator $\tilde{\phi}$ is not actually a true projection and in general $\tilde{\phi} \neq \tilde{\phi}^2$ may hold. To compute the in-exact projection of $\tilde{\phi}$ in practice requires that we solve equation 5.3 each time step. We discussed in Section 4.1 how the multigrid method could be used to solve on multilevel meshes such equations.

To account for the contributions of the thermal fluctuations of the fluid over the time step the Gaussian random field \mathbf{g} with mean zero and covariance G given in equation 5.6 must be generated each time step. The computational cost of the time step of the fluid will depend on the efficiency with which this field can be generated. In general this poses a significant challenge since utilizing widely used methods such as Cholesky factorization or Metropolis sampling to generate correlated random variables may become prohibitive for stochastic fields on a mesh. For example, for a three dimensional uniform mesh with $M = N^3$ cells the Cholesky factorization scales as $O(M^3) = O(N^9)$. In Section 5.2.1, we discuss an alternative approach which allows for the development of efficient methods to generate the stochastic fields on multilevel meshes.

To integrate the stochastic dynamics of the immersed structures over a time step we again use a variant of the Euler-Maruyama method

$$(5.5) \quad \mathbf{X}^{[j],(n+1)} = \mathbf{X}^{[j],(n)} + \sum_{\mathbf{m}} \delta_a \left(\mathbf{y}_{\mathbf{m}} - \mathbf{X}^{[j],(n)} \right) \tilde{\phi} \mathbf{u}_{\mathbf{m}}(s) \Delta x^3.$$

We remark that the spatial average is performed using δ_a which is an approximation of the Dirac δ -function, see Appendix A for one such choice. For a more general discussion of δ_a functions see [1].

5.2.1. Method to Generate the Stochastic Fields on Multilevel Meshes.

To numerically simulate the fluid flows requires that the stochastic forcing \mathbf{f}_T be efficiently generated to account for the thermal fluctuations of the fluid. This requires

each time step generating a Gaussian random variable \mathbf{g} on the mesh with mean 0 and covariance G . In the case of a uniform periodic mesh G can be diagonalized using the discrete Fourier transform. The stochastic forcing can then be obtained from explicit analytic expressions for the random variables in Fourier space. The stochastic forces is then generated by performing a Fast Inverse Fourier Transform. On non-uniform non-periodic meshes this approach can no longer be applied.

A common approach to generate correlated Gaussian random variables is to compute a factorization of the covariance matrix G of the form $G = QQ^T$, where Q is typically an upper triangular matrix. Such a factorization can be shown to exist for any symmetric positive semi-definite matrix. The random variable is then generated from $\mathbf{g} = Q\eta$, where η is the vector consisting of components which are independent standard Gaussian random variables. The factor Q is typically determined using the Cholesky factorization algorithm on G .

The efficiency of this approach depends crucially on two important issues: (i) The computational effort required to obtain the factor Q . (ii) The sparsity structure of Q which will determine the computational cost of the matrix-vector multiplications each time step. For $M = N^3$ mesh cells the cost of performing a Cholesky factorization is $O(M^3) = O(N^9)$. For a non-sparse full matrix factor Q the computational complexity for generating a variate each time step is on the order of $O(M^2) = O(N^6)$. These costs are prohibitive for meshes even with a moderate number of mesh cells N in each direction. All of this can of course be greatly improved if Q is sparse. For instance, in the case that Q has only a constant number of non-zero entries per row the complexity of matrix-vector multiplication per time step reduces to order $O(M) = O(N^3)$. Therefore, an approach is sought by which the factors Q can be determined efficiently for a multilevel mesh which are ideally as sparse as possible.

We remark that another issue is that when G is no longer positive definite, but only semi-definite as in the case of periodic meshes, the Cholesky factorization algorithm can no longer be directly applied. To avoid these issues, we shall take a different approach and show how a sparse factor Q (not necessarily lower triangular) can be obtained for the particular covariance matrix G given in equation 5.6.

Our factorization will be based on decomposing the random variable into two parts $\mathbf{g} = \mathbf{g}_1 + \mathbf{g}_2$. This allows for the covariance matrix to be expressed as $G = G_1 + G_2 + 2G_{(1,2)}$, where $G_k = \langle \mathbf{g}_k \mathbf{g}_k^T \rangle$, $k \in \{1, 2\}$, $G_{(1,2)} = \langle \mathbf{g}_1 \mathbf{g}_2^T \rangle$. We shall generate \mathbf{g} using two independent random sources. The first source \mathbf{g}_1 , can be thought of as corresponding to random fluxes at the cell faces. The second source \mathbf{g}_2 can be thought of as corresponding to random fluxes at the cell centers. Furthermore, the random variable \mathbf{g} can be extended to the mesh consisting of both the cell centered values and face centered values. The covariance matrix for the cell centers given in equation 5.6 can also be naturally embedded in the covariance matrix for this more extensive field. In particular, rows and columns corresponding to face centered values are taken to be zero. This description allows for the random sources to be described by one composite standard random Gaussian vector η with independent components at the cell centers and face centers. The algorithm to generate \mathbf{g} can then be described in terms of a factor Q as before.

We remark that if the two sources are taken to have mean zero and are independent then $G_{(1,2)} = 0$. As a consequence this allows for the factor Q to be decomposed as $Q = Q_1 + Q_2$. This follows since from $\mathbf{g}_1 = Q_1\eta$ and $\mathbf{g}_2 = Q_2\eta$ we have as a consequence of independence that $\langle \mathbf{g}_1 \mathbf{g}_2^T \rangle = Q_1 Q_2^T = 0$ and $Q_2 Q_1^T = 0$. This reduces the problem to finding a decomposition of $G = G_1 + G_2$ for which there are two sparse

matrices Q_1, Q_2 such that $Q_1 Q_1^T = G_1$ and $Q_2 Q_2^T = G_2$.

We now show how this decomposition can be used to obtain a sparse factor Q for multilevel meshes when the covariance arises from the MAC Laplacian and equilibrium fluctuations given in equation 5.4. To illustrate the approach we shall consider some typical rows of the covariance matrix corresponding to mesh cells at the coarse-refined interface, see Figure ?? . In the figure the coarse mesh cell has label A and the more refined neighboring mesh cells have labels B,C,D,E at the coarse-refined interface.

By the fluctuation-dissipation principle in Section 3.1 the covariance matrix corresponding to the MAC Laplacian is

$$(5.6) \quad G = -2LC.$$

The field value at the center of cell A has covariance with the other field values on the mesh given by the row entries $G_{A,A} = (8k_B T/3\Delta x^5) + (10k_B T/\Delta x^5)$, $G_{A,k} = -16k_B T/3\Delta x^5$ $k \in \{B, C, D, E\}$, and for the remaining unlabeled neighbors at the same level of refinement $G_{A,k} = -2k_B T/\Delta x^5$ $k \in \{\text{coarse-neigh.}\}$. Throughout we shall take $\Delta x = \Delta \mathbf{x}_A$ to be the mesh width of cell A . The field value at the center of cell B has covariance with the other field values given by $G_{B,B} = (64k_B T/3\Delta x^5) + (320k_B T/\Delta x^5)$, $G_{B,A} = -16/3\Delta x^5$, $G_{B,D} = 64k_B T/3\Delta x^5$ $G_{B,k} = -64k_B T/\Delta x^5$ $k \in \{\text{refined-neigh.}\}$. The mesh cells C, D, E have similar entries.

We now show the covariance structures which can be attained by taking the divergence of random fluxes J generated at each face. This gives at the cell center with index \mathbf{m} the value

$$g^{(\mathbf{m})} = (DJ)^{(\mathbf{m})} = \frac{(J_N^{(\mathbf{m})} - J_S^{(\mathbf{m})}) + (J_U^{(\mathbf{m})} - J_D^{(\mathbf{m})}) + (J_W^{(\mathbf{m})} - J_E^{(\mathbf{m})})}{\Delta x}$$

where $J_k^{(\mathbf{m})}$ denotes a random flux at face $k \in \{N, S, U, D, W, E\}$ where the indices correspond respectively to the north, south, up, down, west, and east directions. This gives the covariance on the mesh

$$\begin{aligned} \tilde{G}_{\mathbf{m},\mathbf{m}} &= \frac{(\sigma_N^{(\mathbf{m})})^2 + (\sigma_S^{(\mathbf{m})})^2 + (\sigma_U^{(\mathbf{m})})^2 + (\sigma_D^{(\mathbf{m})})^2 + (\sigma_W^{(\mathbf{m})})^2 + (\sigma_E^{(\mathbf{m})})^2}{\Delta x^2} \\ \tilde{G}_{\mathbf{m},\mathbf{k}} &= \frac{-\left(\sigma_{\mathbf{k}}^{(\mathbf{m})}\right)^2}{\Delta x^2} \text{ for } \mathbf{k} \in \{\text{neighbors}\} \end{aligned}$$

where $\langle J_k^{(\mathbf{m})} \rangle = 0$ and $\langle (J_k^{(\mathbf{m})})^2 \rangle = (\sigma_{\mathbf{k}}^{(\mathbf{m})})^2$.

We now discuss whether such an approach is adequate to produce to covariance $G = -2LC$. In the case of a uniform mesh, it can be shown this approach is sufficient to completely generate the covariance matrix G . More specifically, one takes $(\sigma_{\mathbf{k}}^{(\mathbf{m})})^2 = 2k_B T/\Delta x^3$. In contrast, for non-uniform meshes the random fluxes are not sufficient to generate the required covariance structure G as a consequence of features of the coarse-refined interfaces. However, the covariance structure obtained from such random fluxes can be used to generate partially the required covariances G . Consider the case in which the random fluxes are generated only at faces not involved in a coarse-refined interface. This then gives a covariance matrix \tilde{G} with entries given by equation 5.7, where zero variances are taken for the faces involved in the coarse-refined interfaces. This leaves the remaining covariance contributions $\Delta G = G - \tilde{G}$

where $\Delta G_{A,A} = 8k_B T/3\Delta x^5$, $\Delta G_{A,k} = -16k_B T/3\Delta x^5$, $k \in \{B, C, D, E\}$, $\Delta G_{B,A} = -16k_B T/3\Delta x^5$, $k \in \{B, C, D, E\}$, and $\Delta G_{B,D} = 64k_B T/3\Delta x^5$.

An important feature of ΔG is that each block of cells involved in a coarse-refined interface is decoupled from all other cells of the mesh. This allows each decoupled coarse-refined sub-system of mesh cells to be considered separately. Factoring out $c = 8k_B T/3\Delta x^5$ we can express $\Delta G = cM$. For the sub-system shown in Figure ??, this contributes the following terms to M ,

$$(5.7) \quad \tilde{M} = \begin{bmatrix} 1 & -2 & -2 & -2 & -2 \\ -2 & 8 & 0 & 8 & 0 \\ -2 & 0 & 8 & 0 & 8 \\ -2 & 8 & 0 & 8 & 0 \\ -2 & 0 & 8 & 0 & 8 \end{bmatrix}$$

with row and column ordering A, B, C, D, E . The eigenvalues of the matrix are $\lambda_0 = 0$, $\lambda_1 = 16$, $\lambda_2 = 17$, where λ_0 has multiplicity two. This shows that the matrix is positive semi-definite. Consequently, the additional covariance contributions to ΔG can be obtained by generating additional random variables independent of the random fluxes and independent for each coarse-refined sub-system. This can be done in practice by generating for each sub-system $\mathbf{h} = \sqrt{\lambda_1}\eta_1\mathbf{w}_1 + \sqrt{\lambda_2}\eta_2\mathbf{w}_2$, where η_1, η_2 are any two independent Gaussian random variables which by convention are taken to be the cell center sources of A and B . This uses the eigenvectors for the non-zero values $\mathbf{w}_1 = (1/2)[0, -1, 1, -1, 1]^T$ and $\mathbf{w}_2 = (1/\sqrt{17})[-1, 2, 2, 2, 2]^T$. A similar approach can be used to generate the contributions for the other cases of cell arrangements at the coarse-refined interfaces.

Algorithm 6: Generator for the Stochastic Forcing of the Fluid

Data: $G = -2LC$ covariance matrix for the stochastic force field.

Result: Stochastic random force field \mathbf{g} with mean zero and covariance G .

Procedure:

1. Generate independent random fluxes \mathbf{J} for all faces not involved in a coarse-refined interface with variances determined by G_1 .
 2. Compute $\mathbf{g}_1 = D\mathbf{J}$, by evaluating the MAC divergence of the random fluxes.
 3. Determine on the mesh the covariance structure G_2 for each decoupled sub-system at the coarse-refined interfaces.
 4. Generate the components of \mathbf{g}_2 by looping through each of the decoupled sub-systems of the mesh.
 5. Compute $\mathbf{g} = \mathbf{g}_1 + \mathbf{g}_2$.
-

To summarize, this gives the following procedure for generating the stochastic random field. Let Q_1 be the factor which gives $\mathbf{g}_1 = Q_1\eta$ corresponding to taking the divergence of the random fluxes generated at each face not involved in a coarse-refined interface. Let Q_2 be the factor which gives $\mathbf{g}_2 = Q_2\eta$ corresponding to generating the random variable for each decoupled sub-system of the coarse-refined interfaces from the cell centered random sources. The covariances obtained in each of these steps corresponds to $G_1 = Q_1Q_1^T = \tilde{G}$ and $G_2 = Q_2Q_2^T = \Delta G$. From independence of the two types of random sources it follows that $G_1 + G_2 = G$. For G this gives a sparse factor of the form $Q = Q_1 + Q_2$ which has only order one entries per row. The sparse factor can then be used to generate the random variable \mathbf{g} for the multilevel mesh with optimal linear computational complexity $O(M)$.

5.3. Regime II: Stochastic Numerical Methods for the Case of Steady-State Stokes Flow. We now discuss numerical methods for the case where the dynamics of the immersed structures are slow relative to the correlation time scale of the fluid velocity fluctuations. In this regime the stochastic equations of the fluid-structure system are numerically stiff and applying the numerical methods of Section 5.2 require small time steps making it computationally expensive to resolve the long-time dynamics of the immersed structures. We now discuss an approach for this regime by which the separation in time scales can be used to obtain a more effective method.

5.3.1. Stiffness of the Stochastic Differential Equations. In the stochastic dynamics of the fluid-structure system the time scale on which immersed structures diffuse a distance comparable to the length scale a appearing in δ_a is often much larger than the time scale associated with the temporal correlation of the spatially averaged fluid velocity determining the motion of the immersed structures. In the time integrators discussed in Section 5.2 this places severe constraints on the time steps which are permitted since the fluid fluctuations driven by the stochastic forcing must be explicitly resolved. In this section we discuss how more efficient numerical methods can be obtained by performing a temporal averaging of the stochastic fluctuations of the fluid to approximately account for their statistical contributions to the immersed structure dynamics over a time step.

While the spirit of our approach shares some similarities with the derivation of the Stokesian-Brownian dynamics method, which reduces the system to a hydrodynamic coupling tensor and stochastic forces acting only on the immersed particles, our approach differs by retaining the fluctuations of fluid velocity field on the entire mesh. While this may lead to a greater computation expense, one feature of this approach is that it potentially allows for a generalization to complex flows where only a constitutive law for the fluid need be numerically defined on the mesh, as opposed to computing in advance a hydrodynamic coupling matrix. For Newtonian fluids it can be shown that the effective hydrodynamic coupling and stochastic driving of immersed structures in the stochastic immersed boundary method yield the same dynamics as Stokesian-Brownian dynamics for the far-field interactions (cite). An important issue which allows for tractable numerical methods in our case are methods developed to generate the stochastic fields on the multilevel meshes with linear computational complexity in the mesh size $O(M)$.

To obtain effective equations for the particle dynamics we consider a Taylor expansion of the δ_a -function in \mathbf{x} to obtain:

$$(5.8) \quad \mathbf{X}^{[j]}(t) = \mathbf{X}^{[j]}(0) + \mathbf{I}_{X,j}(t)(1 + o(1))$$

$$(5.9) \quad \mathbf{I}_{X,j}(t) := \int \delta_a(\mathbf{x} - \mathbf{X}^{[j]}(0)) \tilde{\varphi} \mathbf{I}_v(\mathbf{x}, t) d\mathbf{x}$$

$$(5.10) \quad \mathbf{I}_v(\mathbf{x}, t) := \int_0^t \mathbf{u}(s) ds.$$

To numerically integrate the particle dynamics we shall use:

$$(5.11) \quad \mathbf{X}^{[j],n+1} = \mathbf{X}^{[j],n} + \int \delta_a(\mathbf{x} - \mathbf{X}^{[j],n}) \tilde{\varphi} \mathbf{I}_v(\mathbf{x}, \Delta t) d\mathbf{x}.$$

To close the equations so that they depend only on $\mathbf{X}^{[j]}$ and the instantaneous forces acting on the system, the steady-state statistics of $\mathbf{I}_v(\mathbf{x}, \Delta t)$ will be determined.

5.3.2. Stochastic Closure for the Fast Dynamics of the Fluid-Structure System. We now show how the statistics of $\mathbf{I}_V(t)$ can be computed to close the equations. Since $\mathbf{I}_V(t)$ is a Gaussian random field it is completely determined by its mean $\mu_V(t)$ and covariance $\Lambda_V(t)$. These can be expressed as

$$(5.12) \quad \langle \mathbf{I}_V(t) \rangle := \mu_V(t) = \int_0^t \langle \mathbf{u}(s) \rangle ds$$

$$(5.13) \quad \langle (\mathbf{I}_V(t) - \mu_V(t)) (\mathbf{I}_V(t) - \mu_V(t))^T \rangle := \Lambda_V(t) = \int_0^t \int_0^t \langle \mathbf{u}(r) \mathbf{u}(s)^T \rangle dr ds$$

$$(5.14) \quad - \Omega(t)$$

$$(5.15) \quad \Omega(t) := \int_0^t \int_0^t \langle \mathbf{I}_S(r) \mathbf{I}_S(s)^T \rangle dr ds.$$

By integrating equation 2.1 the fluid velocity at time t can be expressed as:

$$(5.16) \quad \mathbf{u}(t) = e^{tL} \mathbf{u}(0) + \mathbf{I}_S + \mathbf{I}_T$$

$$(5.17) \quad \mathbf{I}_S := \int_0^t e^{(t-s)L} \rho^{-1} \tilde{\zeta}^T \mathbf{f}_S(s) ds$$

$$(5.18) \quad \mathbf{I}_T := \int_0^t e^{(t-s)L} Q d\mathbf{B}(s)$$

where e^{tL} denotes the matrix exponential of the operator tL and $G = QQ^T$. It follows from the fact that $\langle \mathbf{u} \rangle = 0$ and from equations 5.17 and 5.18 that $\langle \mathbf{I}_S(t_1) \mathbf{I}_T(t_2)^T \rangle = 0$, $\langle \mathbf{I}_S(t_1) \mathbf{u}(0) \rangle = 0$, and $\langle \mathbf{I}_T(t_1) \mathbf{u}(0) \rangle = 0$. Furthermore, this shows that the mean of the time integrated velocity field is

$$(5.19) \quad \mu_V(t) = \int_0^t \mathbf{I}_S(s) ds.$$

The covariance can then be expressed as

$$(5.20) \quad \Lambda_V(t) = \int_0^t \int_0^t e^{rL} C e^{sL^T} dr ds + \int_0^t \int_0^t \langle \mathbf{I}_T(r) \mathbf{I}_T(s)^T \rangle dr ds.$$

From Ito's Isometry applied to (ref) and (ref) we have

$$(5.21) \quad \Lambda(t) = \Lambda_1(t) + \Lambda_2(t)$$

$$(5.22) \quad \Lambda_1(t) := \int_0^t \int_0^t e^{rL} C e^{sL^T} dr ds$$

$$(5.23) \quad \Lambda_2(t) := \int_0^t \int_0^t \int_0^{r \wedge s} e^{(r-w)L} G e^{(s-w)L^T} dw dr ds.$$

The first term can be integrated explicitly to obtain:

$$(5.24) \quad \Lambda_1(t) = L^{-1} (e^{tL} - \mathcal{I}) C L^{-T} (e^{tL^T} - \mathcal{I}).$$

The second term can be integrated by a more involved calculation to obtain, see Appendix (ref),

$$(5.25) \quad \Lambda_2(t) = -L^{-1} G L^{-T} \left(L^{-T} (e^{tL^T} - \mathcal{I}) - t\mathcal{I} \right)$$

$$(5.26) \quad -L^{-1}(L^{-1}(e^{tL} - \mathcal{I}) - t\mathcal{I})GL^{-T}$$

$$(5.27) \quad + \Lambda_{2,1}(t)L^{-T} + L^{-1}\Lambda_{2,1}(t)$$

$$(5.28) \quad \Lambda_{2,1}(t) := \int_0^t (t-q)e^{qL}Ge^{qL^T}dq.$$

The equations 5.19 and 5.20 give expressions for the mean $\mu_V(t)$ and covariance $\Lambda_V(t)$ of $\mathbf{I}_V(t)$. To use these expressions in practice requires they be numerically evaluated or approximated effectively.

5.4. Approximating the Mean and Covariance of the Temporally Averaged Fluctuations of the Fluid. The expressions above must be evaluated numerically or approximated to obtain a method to generate the corresponding stochastic fields. We now discuss how to approximate expressions for the mean and covariance in the regime in which the time scale of the particle dynamics is separated from the time scale of the fluid dynamics. This corresponds the regime in which $\tau_S \gg t \gg \tau_F$, where τ_F characterizes the relaxation time scale of the fluid modes on the length scale of the particles and τ_S is the length scale on which the particle positions change on the length scale of the particle size. In this regime we have that e^{tL} is small. Recall that the dissipative operator L is negative semi-definite.

Assuming the force changes on a time scale slower than or equal to τ_S we have from equation 5.17 the approximation:

$$(5.29) \quad \mathbf{I}_s(t) = -\rho^{-1}L^{-1}[\mathcal{I} - e^{tL}]\tilde{\phi}^T\mathbf{f}_s(0) + \dots$$

This yields an approximation of the mean of $\mathbf{I}_V(t)$. The covariance matrix $\Lambda(t)$ of the time integrated fluctuations of the fluid velocity can also be approximated in this regime. From (ref) it follows that, see Appendix (ref),

$$(5.30) \quad \Lambda_{2,1}(t) = tC + \dots$$

Substitution into (ref) gives the leading order term

$$(5.31) \quad \Lambda(t) = -t(L^{-1}C + CL^{-T}) + \dots$$

In the case when L is the "MAC Laplacian" and C is the equilibrium covariance given in 3.4 the matrix product is symmetric, $L^{-1}C = (L^{-1}C)^T$. This reduces the covariance to

$$(5.32) \quad \Lambda(t) = -2tL^{-1}C + \dots$$

Using the expressions above for the mean and covariance of the temporally averaged fluctuations of the fluid the numerical method to update the immersed structures can be expressed as

$$(5.33) \quad \mathbf{X}^{[j],(n+1)} = \mathbf{X}^{[j],(n)} + \sum_{\mathbf{m}} \delta_a(\mathbf{y}_{\mathbf{m}} - \mathbf{X}^{[j],(n)})\tilde{\phi}\mathbf{I}_{V,\mathbf{m}}^{(n)}\Delta x^3.$$

From equations (ref) and (ref) we see that the stochastic field for the contributions of the thermal fluctuations to the immersed structure dynamics over the time step can be naturally decomposed into a drift field and a fluctuating field. In particular, we can express this as $\mathbf{I}_V = \mathbf{u}_I + \mathbf{u}_{II}$ where $\mathbf{u}_I = -\rho^{-1}L^{-1}\tilde{\phi}^T\mathbf{f}_S\Delta t$ is a deterministic field over each time step obtained from equation (ref) and \mathbf{u}_{II} is a Gaussian random field with mean zero and covariance $G = -2\Delta tL^{-1}C$ obtained from equation (ref). To obtain a method which is useful in practice requires that each of the fields \mathbf{u}_I and \mathbf{u}_{II} be generated without too much computational expense. We now discuss how these fields can be efficiently computed for the stochastic immersed boundary method.

5.4.1. Computing \mathbf{u}_I : Solver for Steady-State Stokes Flow . The velocity field \mathbf{u}_I corresponds to the steady-state Stokes flow which is obtained when the force density \mathbf{f}_S acts on the fluid. To obtain this field the linear system $L\mathbf{u}_I = \rho^{-1}\tilde{\phi}^T\mathbf{f}_S$ is solved for \mathbf{u}_I by using the multigrid method described in Section 4.1. Carrying this out by using multigrid methods for the mesh has linear computational complexity of order $O(M)$, where M is the number of mesh cells.

5.4.2. Computing \mathbf{u}_{II} : Methods for Stochastic Fields for the Time Integrated Fluctuations of the Fluid. To numerically account for the contributions of the thermal fluctuations of the fluid in the dynamics of the immersed structures requires generation of a stochastic field each time step. For the time step spanning $[0, \Delta t]$ the field is given by $\mathbf{u}_{II} = \int_0^{\Delta t} \mathbf{f}_T(s)ds$, which has mean and covariance given in equation 5.19 and 5.20. We showed in Section 5.4 that this can be approximated by a Gaussian random field \mathbf{g} with mean zero and covariance $G = -2\Delta t L^{-1}C$.

In the case of uniform periodic meshes the Fourier transform can be used to diagonalize G and the field \mathbf{g} can be obtained by performing a Fast Inverse Fourier Transform on random variates generated in Fourier space. The computational complexity of this approach was of order $O(M \log(M))$, where $M = N^3$ was the number of mesh sites. For non-uniform or non-periodic meshes the Fourier transform is no longer directly applicable. We now show how a different approach can be taken to generate samples of the stochastic field \mathbf{g} on multilevel meshes, while still attaining a comparable computational efficiency as the approach used for uniform meshes.

Our approach will be based on modifying the multigrid method to produce random samples. To help motivate the methods it will be convenient to interpret the multigrid method from a variational perspective. A multigrid iteration for a symmetric positive definite matrix A corresponding to the linear system $A\mathbf{v} = \mathbf{b}$ can be interpreted as a minimization procedure for the energy $E(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T A\mathbf{v} - \mathbf{v}^T \mathbf{b}$. Using Gauss-Siedel as the smoother the update sweeps on the mesh are given by equations 4.7. where the indices \mathbf{m} are taken in some ordering. For each index \mathbf{m} the Gauss-Siedel update can be easily shown to correspond to solving for the $v_{\mathbf{m}}$ which minimizes the energy $E(\mathbf{v})$ holding all other component values of \mathbf{v} fixed. The sweep on the mesh then corresponds to a sequence of these site-wise constrained minimizations performed for the energy E . One interpretation the multigrid method is that the convergence to the minimizer can be accelerated by performing updates on collective degrees of freedom of the mesh instead of values at only one mesh site. These collective degrees of freedom in the multigrid method are obtained by considering the mesh sites at different levels of refinement and their influence through interpolation on collective sites of the finest mesh of the system.

Let I_ℓ^r be the prolongation operator which interpolates mesh values at refinement level ℓ to level r . Let the restriction operator which coarsens data from refinement level r to level ℓ be taken to be $I_r^\ell = (I_\ell^r)^T$. Let ℓ^* denote the the most refined level, which corresponds to the uncoarsened mesh. We can interpret values at individual mesh sites on the coarsened meshes as corresponding to collective degrees of freedom on the most refined mesh. These site values can be thought of as collective degrees of freedom in the sense that if we were to change a site value at refinement level ℓ this would influence multiple values on the uncoarsened mesh through the interpolation performed by the prolongation operator $I_\ell^{\ell^*}$.

The energy can be expressed in terms of the collective degrees of freedom obtained from the mesh at level ℓ by substituting the corresponding interpolation into the energy for the uncoarsened mesh. This gives $E(\mathbf{v}^{(\ell)}) := E(I_\ell^{\ell^*}\mathbf{v}^{(\ell)}) = (\mathbf{v}^{(\ell)})^T A^{(\ell)} (\mathbf{v}^{(\ell)}) -$

$(\mathbf{v}^{(\ell)})^T \mathbf{b}^{(\ell)}$, where $A^{(\ell)} := (I_{\ell}^{\ell*})^T A (I_{\ell}^{\ell*})$ and $\mathbf{b}^{(\ell)} := (I_{\ell}^{\ell*})^T \mathbf{b}$. The Gauss-Siedel updates on level ℓ then correspond to minimizing the energy E with respect to $\mathbf{v}_{\mathbf{m}}^{(\ell)}$ while holding the other components of $\mathbf{v}^{(\ell)}$ fixed. The multigrid iteration from the variational perspective then corresponds to approaching the minimizer of E by performing a sequence of constrained minimizations at different levels of refinement of the mesh. For a further discussion of multigrid methods, see (cite).

We now discuss how this perspective can be used to derive a Gibb's sampler for correlated random variables. The random variable \mathbf{g} with mean \mathbf{b} and covariance G has the probability density $\rho(\mathbf{g}) = \frac{1}{Z} \exp(-\frac{1}{2}\mathbf{g}^T G^{-1}\mathbf{g} + \mathbf{g}^T G^{-1}\mathbf{b})$, where Z is the normalization factor so that the density integrates to one. This can be related to the energy E above by letting $A = G^{-1}$, in which case $\rho(\mathbf{g}) = \frac{1}{Z} \exp(-E(\mathbf{g}))$.

A common approach to sampling a random variable with a distribution $\rho(\mathbf{g})$ is to construct a Markov-Chain that has $\rho(\mathbf{g})$ as the invariant distribution. The efficiency of the method in generating independent random variates is then determined by two factors. The first is the expense in computing each iteration of the Markov-Chain. The second is the number of samples which need to be generated to obtain variants having a correlation below some threshold. The latter is often characterized by the autocorrelation function of the sequence. To obtain a random sampler from the multigrid method we shall modify the Gauss-Siedel update by adding a random term as follows

$$(5.34) \quad v_{\mathbf{m}}^{(\text{new})} = \frac{\mathbf{b}_{\mathbf{m}} - \sum_{\mathbf{k} > \mathbf{m}} A_{\mathbf{m},\mathbf{k}} v_{\mathbf{k}}^{(\text{old})} - \sum_{\mathbf{k} < \mathbf{m}} A_{\mathbf{m},\mathbf{k}} v_{\mathbf{k}}^{(\text{new})}}{A_{\mathbf{m},\mathbf{m}}} + \left(\frac{1}{\sqrt{A_{\mathbf{m},\mathbf{m}}}} \right) \eta_{\mathbf{m}}$$

where $\eta_{\mathbf{m}}$ is a standard Gaussian random variable generated independently for each update.

This random update corresponds to generating $\mathbf{v}_{\mathbf{m}}$ according to the conditional probability distribution of $\rho(\mathbf{g})$ obtained when conditioning on the other components of \mathbf{v} . As a consequence, if the random variable \mathbf{v} has distribution $\rho(\mathbf{g})$ before the update the new random variable \mathbf{v}' obtained by the update also has distribution $\rho(\mathbf{g})$. This ensures that $\rho(\mathbf{g})$ is the invariant distribution of the Markov-Chain and transforms the multigrid iterator into a Gibb's sampler [?].

This approach was first proposed by Goodman and Sokal to generate variates for lattice gauge theories [?]. The efficiency with which nearly independent Gaussian random variates can be obtained is characterized by the autocovariance of the sequence of variates generated by the multigrid sampler. To obtain the autocovariance we can express the stochastic multigrid iteration as

$$\mathbf{v}^{(n+1)} = \mathcal{M}\mathbf{v}^{(n)} + Q\eta^{(n)}$$

where $\eta^{(n)}$ is a vector with components consisting of independent standard Gaussians generated independently each iteration, \mathcal{M} are matrices corresponding to the deterministic multigrid iterations, and Q is the matrix corresponding to the stochastic terms which modified the Gauss-Siedel updates.

Starting the iterations from statistical equilibrium with $\langle \mathbf{v}^{(0)} \rangle = A^{-1}\mathbf{b}$ and $\text{cov}(\mathbf{v}^{(0)}, \mathbf{v}^{(0)}) = A^{-1}$, the autocovariance of the generated sequence of samples is given by

$$\text{cov}(\mathbf{v}^{(n_0+\Delta n)}, \mathbf{v}^{(n_0)}) = \mathcal{M}^{\Delta n} A^{-1}.$$

From this it follows that the decay in Δn of the autocovariance is governed by the spectral radius of \mathcal{M} [?, ?]. This is the same factor which governs the decay of the error when the deterministic multigrid iterations are used to solve linear systems. Consequently, after only relatively few multigrid iterations the samples can be made to be correlated negligibly. For a more detailed discuss see [?, ?].

The efficiency of the algorithm also depends crucially on the expense with which iterations can be performed. An important issue is that the multigrid sampler performs iterations using the inverse of the covariance matrix $A = G^{-1}$. For the stochastic immersed boundary method the stochastic field has covariance $G = -2\Delta t L^{-1}C$ with inverse $G^{-1} = \frac{-1}{2\Delta t}C^{-1}L$ which is sparse with a constant number of non-zero entries per row. Consequently, the sampler based on the multigrid method produces nearly independent variates of the stochastic field \mathbf{g} on multilevel meshes with a computational complexity of only order $O(M)$.

Algorithm 7: Generator for Time Integrated Thermal Fluctuations of the Fluid

Data: $G = -2\Delta t L^{-1}C$ covariance matrix for the stochastic force field.

Result: Stochastic random force field \mathbf{g} with mean zero and covariance G .

Procedure:

1. Perform stochastic multigrid iterations on the mesh.
-

6. Applications.

6.1. Boltzmann Statistics of an Elastic Dimer.

6.2. Hydrodynamic Interactions of Two Interacting Particles. Show the effective hydro-dynamic interaction matrix and autocovariances achieved in fluctuations of two interacting particles (at fixed locations on the mesh).

6.3. Diffusivity of Particles in the Presence of Walls and Fixed Inclusions.

6.4. Fluctuations of a Wormlike Chain Polymer.

7. Conclusions.

8. Acknowledgements. The author P.J.A. acknowledges support from research grant NSF DMS-0635535 while at UCSB. The author would also like to thank Alexander Roma, Boyce Griffith, and Micheal Minion for helpful suggestions.

REFERENCES

- [1] C.S. PESKIN, *The immersed boundary method*, Acta Numerica, 11 (2002), pp. 479–517.

Appendix A. The Representation Function δ_a for Immersed Particles.

In the immersed boundary method, it is required that a function δ_a be specified to represent the elementary particles. The representation of this function is often derived from the following function ϕ which is known to have desirable numerical properties

[1]:

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

For three dimensional systems the function δ_a representing elementary particles of size a is

$$(A.2) \quad \delta_a(\mathbf{r}) = \frac{1}{a^3} \phi\left(\frac{\mathbf{r}^{(1)}}{a}\right) \phi\left(\frac{\mathbf{r}^{(2)}}{a}\right) \phi\left(\frac{\mathbf{r}^{(3)}}{a}\right),$$

where the superscript indicates the index of the vector component.

To maintain good numerical properties, the particles are restricted to sizes $a = n\Delta x$, where n is a positive integer. For a derivation and a detailed discussion of the properties of these functions see [1].

Appendix B. Periodic Meshes: Adjustments for Singular Mesh Operators. For periodic meshes the discrete Laplacian is singular with a null space spanned by the vector \mathbf{v} with identical components, $v_i = 1$ for all i . In the multigrid iterations this can cause problems both in the deterministic and stochastic case. In particular, the iterations exhibit drift which eventually degrades the computed solution through round-off errors or overflow. To obtain a non-singular matrix a rank-one matrix can be added to the discrete Laplacian to obtain $\tilde{L} = L - \mathbf{q}\mathbf{q}^T$. This introduces the eigenvalue $\lambda = \|\mathbf{q}\|^2$ for \tilde{L} . In order for \tilde{L} to be well-conditioned this eigenvalue should be comparable in magnitude to the eigenvalues of L . We take

$$(B.1) \quad \mathbf{q}_i = -\frac{2\pi}{\ell\sqrt{N}}$$

where ℓ is the length of the mesh box size, and N is the total number of mesh points. This gives an eigenvalue $\lambda = -4\pi/\ell^2$ which is comparable in magnitude to the eigenvalue associated with a uniform mesh and the Fourier mode with wavenumber $\mathbf{k} = (1, 0, 0)$. For non-uniform meshes this eigenvalue is also comparable in magnitude to those of L . The special form of the rank-one matrices can be used so that the added computational cost in the iterations is not great and the overall cost remains proportional to the total number of mesh points.

Appendix C. Approximate Projection of Stochastic Equations. (PJA: rewrite exposition of this section)

We shall now discuss a few issue related to the approximate projection of the stochastic equations of the immersed boundary method. In the schemes careful attention was paid to control discretization errors to preserve statistical mechanical features of the discrete dynamical system approximating the continuum system. Naively

applying approximate projection methods may disrupt these features of the numerical scheme. In this section we discuss an approach which introduces the approximate projection operations in a manner which preserves many of the statistical features of the unprojected scheme. In particular, we show that for the time dependent Stokes equations the projection can be introduced so that energy is preserved in the fluid-particle coupling. For the steady-state Stokes equations we show that the projection can be introduced so that detailed balance is preserved for the Gibbs-Boltzmann distribution.

Force density is modified to use instead

$$\tilde{\mathbf{f}} = \tilde{C}_\varphi^T \tilde{C}^{-1} \mathbf{f}.$$

This ensures that the hydrodynamic coupling and stochastic forcing of the system satisfy Detailed Balance for the Boltzmann distribution. This condition was found from analysis of whether this condition holds for the system. One could also in principle also try using φ in place of φ^T since this is expected to approx. as the mesh is refined the same continuum projection operator. The Detailed Balance condition would then hold approximately.

Appendix D. Figures.

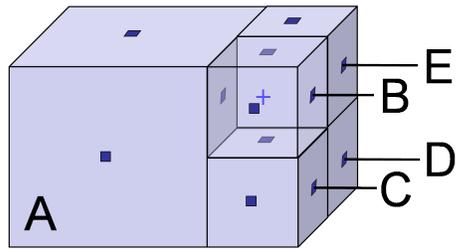


FIG. D.1. 3D Mesh