A monolithic algorithm for the flow simulation of flexible flapping wings

Tao Yang1, Mingjun Wei1,2 ⋆, Kun Jia2 and James Chen3

Abstract
It has been a challenge to simulate flexible flapping wings or other three-dimensional problems involving strong fluid–structure interactions. Solving a unified fluid–solid system in a monolithic manner improves both numerical stability and efficiency. The current algorithm considered a three-dimensional extension of an earlier work which formulated two-dimensional fluid–structure interaction monolithically under a unified framework for both fluids and solids. As the approach is extended from a two-dimensional to a three-dimensional configuration, a cell division technique and the associated projection process become necessary and are illustrated here. Two benchmark cases, a floppy viscoelastic particle in shear flow and a flow passing a rigid sphere, are simulated for validation. Finally, the three-dimensional monolithic algorithm is applied to study a micro-air vehicle with flexible flapping wings in a forward flight at different angles of attack. The simulation shows the impact from the angle of attack on wing deformation, wake vortex structures, and the overall aerodynamic performance.

Keywords
Immersed boundary method, fluid–structure interaction, viscoelasticity, flexible wings, monolithic

Introduction
Fluid–structure interaction (FSI) is inevitable in the study of dynamic behaviors in nature and engineering applications. Due to its complexity involving both fluid flows and flexible structures, FSI is usually studied in a progressive manner: first one-way coupling, then two-way coupling but with rigid-body motion, and eventually two-way coupling with visco-elastic solid structures. One example is the century-long study of (flexible) flapping-wing flight mechanism involving strong FSI between air and wings. The basic mechanism of the thrust generation from flapping wings is described independently by Knoller1 and Betz2 more than a century ago. The difference in thrust generation from pure plunging and pitching was first studied by Garrick.3 More recently, there were studies4–7 considering “onsideri impact from moving structures to fluid flow. With the need to understand the impact of flexibility on flapping-wing aerodynamics in modern technologies,8–16 it becomes crucial to explore the physics of two-way fully coupled FSI problems computationally and experimentally.

Numerical simulation of a fully coupled FSI problem typically includes two separate sets of equations, algorithms and modularized computer codes, for fluid and solid, respectively. The coupling of the two parts is achieved by an iterative process of exchanging information at the fluid–solid interface.17–20 During the process, other algorithms may be involved for the projection and mapping between different coordinate systems for the computation of fluid flows and solid structures. Despite the convenience from separating the code development on fluid and solid solvers, this common practice faces challenges in computational efficiency and stability. The projection and exchange at the fluid–solid interface may potentially reduce the...
overall efficiency and may not reach the convergence with an iterative process at each time step. Physically, the response in fluids and solids are largely different in time scale, and the smaller time scale may limit the overall time to reach the convergence at the fluid–solid interface and therefore reduce the computational efficiency. An ideal remedy would be to have a unified description and a monolithic algorithm to solve the entire FSI problem simultaneously without iterations.

The presented monolithic approach is based on a unified fluid–solid equation in an Eulerian configuration for a combined domain of fluids, solids, and the interfaces.\textsuperscript{10,21,22} The immersed boundary method\textsuperscript{23–32} was implemented relying on surface-force and body-force terms to represent the fluid–structure boundaries and the structural dynamics for viscoelastic bodies with one simple discretization. On the other hand, to assure the accuracy in the computation of structural dynamics, a Lagrangian mesh is kept to track the location of the structure. In the current work, the elastic stress in solids, at the expense of a projection process between fluid and solid,\textsuperscript{10,21} gets more sophisticated for three-dimensional systems. When the situation requires a prescribed motion for certain solid structures (e.g. control gear in mechanical system, or skeleton system in a flying bird),\textsuperscript{10,31} the aforementioned formulation allows additional terms to define motion trajectory in a way similar to traditional immersed boundary method.\textsuperscript{26,30}

The rest of the paper introduces the basic idea and derivation of the unified governing equations for fluids and solids. Followed by the governing equations, a detailed description of various algorithms is presented to deal with numerical challenges in the implementation of the unified fluid–solid equation. The focus is to explain the theoretical derivation and computational treatment of elastic stress terms during the cell division and the projection between the Eulerian and Lagrangian coordinates. Two benchmark cases were demonstrated to validate the algorithm in three dimensions and the viscoelastic treatment in two dimensions. Finally, the current monolithic approach for FSI systems was applied to simulate the flow surrounding the flexible flapping wings of a micro-air vehicle (MAV), which involves all the fundamental features provided by the current algorithm: three-dimensional system, passive viscoelastic structure (i.e. the wing), actively prescribed motion (i.e. the control gear for flapping mechanism). In this study of a flapping-wing forward flight, different angles of attack are considered for the impact on wing deformation, flow structures, and the overall aerodynamic performance. A concise discussion concludes this work.

\textbf{Governing equations}

The whole region is divided into two parts: the fluid region \(\Omega_f\) and the solid region \(\Omega_s\). The solid region \(\Omega_s\) is further divided into two parts, the prescribed region \(\Omega_{sp}\) and the free passive region \(\Omega_{sf}\), as shown in Figure 1. In other words, \(\Omega = \Omega_f \cup \Omega_s\) and \(\Omega_s = \Omega_{sp} \cup \Omega_{sf}\). The whole region \(\Omega\) is formulated in an Eulerian coordinate, while the solid region \(\Omega_s\) is tracked in a Lagrangian manner. The combined governing equations for both incompressible solids and fluids can be expressed in a unified format under the Eulerian configuration.

\textbf{Eulerian description for fluids}

For an incompressible fluid, the non-dimensional continuity and momentum equations must be satisfied

\[ \nabla \cdot \mathbf{u}_f = 0 \]  \hspace{1cm} (1)

\[ \rho \frac{D\mathbf{u}_f}{Dt} = \nabla \cdot \mathbf{\sigma}_f \]  \hspace{1cm} (2)

where the Cauchy stress for incompressible fluids \(\mathbf{\sigma}_f\) includes pressure \(\rho\) and dissipative viscous stress \(\mathbf{\tau}_f\)

\[ \mathbf{\sigma}_f = -\rho \mathbf{I} + \mathbf{\tau}_f \]  \hspace{1cm} (3)

For a Newtonian fluid, the dissipative viscous stress tensor can be derived by employing Onsager reciprocal relations and Coleman–Noll procedure in Clausius–Duhem inequality as

\[ \mathbf{\tau}_f = \mu_f (\nabla \mathbf{u}_f + \nabla \mathbf{u}_f^T) \]  \hspace{1cm} (4)

By substituting equations (3) and (4) into equation (2), the celebrated Navier–Stokes equations for
incompressible viscous flows can be found
\[ \nabla \cdot \mathbf{u}_f = 0 \quad (5) \]
\[ \frac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f \cdot \nabla \mathbf{u}_f = -\nabla p_f + \frac{1}{Re_f} \nabla^2 \mathbf{u}_f \quad (6) \]

where \( \mathbf{u}_f \) is the velocity vector for fluid, \( t \) is time, \( p \) is pressure, and \( Re_f = \rho_f U_\infty L / \mu_f \) is the Reynolds number defined by density \( \rho_f \), free stream velocity \( U_\infty \), characteristic length \( L \), and dynamics viscosity \( \mu_f \).

**Eulerian description for solids**

Motions of solid structures are usually described in a Lagrangian coordinate, and it is uncommon that the motions are formulated in an Eulerian configuration. The balance law of linear momentum for solids in reference configuration is\(^{34,35} \)
\[ \rho \mathbf{u}_s \cdot \nabla \cdot \mathbf{\sigma}_s - \rho \mathbf{b} = 0 \quad (7) \]
where \( \mathbf{u}_s \) is the velocity field for solid, \( \mathbf{\sigma}_s \) is the Cauchy stress for solids, and \( \mathbf{b} \) is the body force density. In an Eulerian description, \( \mathbf{u}_s \) is expanded as the material derivative of velocity field.

For a viscoelastic solid, the constitutive equation can be found as a function of \( \mathbf{\sigma}_s = \mathbf{\sigma}_s(\mathbf{e}, \dot{\mathbf{e}}) \) through Coleman–Noll procedure subject to the axioms of admissibility and objectivity, where
\[ \mathbf{e} = \mathbf{F} \cdot \mathbf{F}^T - \mathbf{I} \quad (8) \]
is the Eulerian strain tensor with the deformation gradient tensor, \( \mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X} \), and
\[ \dot{\mathbf{e}} = \frac{1}{2} (\nabla \mathbf{u}_s + \nabla \mathbf{u}_s^T) \quad (9) \]
is the Eulerian strain rate tensor.\(^{34} \)

The obedience of objectivity allows the Cauchy stress tensor for solid to be expanded with the complete and irreducible set of generators in isotropic function representation theorem.\(^{33,34,36,37} \) With the approximation of incompressibility for solids, i.e. \( \det(\partial \mathbf{x} / \partial \mathbf{X}) = 1 \), the condition of unity in strain should be enforced by a Lagrange multiplier \( p_s \) as an undetermined pressure in the constitutive equations. The linear constitutive model for viscoelastic solids is then found as\(^{34,38,39} \)
\[ \mathbf{\sigma}_s = -p_s \mathbf{I} + \mathbf{\tau}_s + \mathbf{\sigma}_{\text{visc}} \]
\[ = -p_s \mathbf{I} + \mu_s \mathbf{e} + \mu_{fs} \dot{\mathbf{e}} \]
\[ = -p_s \mathbf{I} + \mu_s (\mathbf{F} \cdot \mathbf{F}^T - \mathbf{I}) + \frac{1}{2} \mu_{fs} (\nabla \mathbf{u}_s + \nabla \mathbf{u}_s^T) \quad (10) \]
where \( \mu_s \) is the shear modulus and \( \mu_{fs} \) is the viscosity of viscoelastic materials, and neo-Hookean model is applied for the shear stress term.

Finally, by employing the material derivative and substituting equation (10) into equation (7), we have
\[ \frac{\partial \mathbf{u}_s}{\partial t} + \mathbf{u}_s \cdot \nabla \mathbf{u}_s = -\nabla p_s + \frac{1}{Re_s} \nabla^2 \mathbf{u}_s + \nabla \cdot \mathbf{\tau}_s + \mathbf{b} \quad (11) \]
which is the governing equation for solid motion in an Eulerian description and has a similar format to the Navier–Stokes equations with \( Re_s = \rho_s U_\infty L / \mu_s \).

**Unified framework for solids and fluids**

By comparing equations (6) and (11) and assuming the same density and viscosity in both solids and fluids (i.e. \( \rho_f = \rho_s \) and \( \mu_f = \mu_s \)), the governing equation can be rewritten in a unified fashion as
\[ \frac{\partial \mathbf{u}_m}{\partial t} = -\nabla p_m - \mathbf{u}_m \cdot \nabla \mathbf{u}_m + \frac{1}{Re} \nabla^2 \mathbf{u}_m + \nabla \cdot (\chi_s \mathbf{\tau}_s) \quad (12) \]
where \( m \) indicates the material choice (\( s \) is for solids and \( f \) is for fluids)
\[ \chi_s = \begin{cases} 
1 & \text{in } \Omega_s \\
0 & \text{otherwise} 
\end{cases} \quad (13) \]
and the elastic stress term is calculated by using a neo-Hookean model
\[ \mathbf{\tau}_s = \mu_s (\mathbf{F} \cdot \mathbf{F}^T - \mathbf{I}) \quad (14) \]

The external force at the fluid–solid interface becomes an internal force and is removed from the unified formulation. Although the assumption of the same density provides convenience in the current work, it is possible to remove this constraint for more general applications.\(^{13} \)

**Implementation of prescribed motion**

The unified framework allows to solve the fluid flows and the solid structure simultaneously. Using a rubber ball in a driven cavity flow as an example,\(^{21} \) part of the computation domain is labeled as solid domain for the rubber ball, and the rest of the computational domain is labeled as fluid. By solving equation (12), motions of both the rubber ball and the surrounding fluid flow can be achieved. It should be noticed that the rubber ball only deforms under the influence of the surrounding fluid flow. In other words, the trajectories of each
solid points are passively controlled by the surrounding flow. However, in some applications (e.g. the motion of an actively flapping wing), an actively moving trajectory for the solid, i.e. $\Omega_{sp}$, is required. The Lagrangian cells used for actively deformed solids are tagged as control cells. The moving gaits of such control cells are defined in a way similar to the commonly used control cells used for actively deformed solids are tagged as control cells. The moving gaits of such control cells are defined in a way similar to the commonly used direct-forcing approach.\(^{24}\)

In the direct-forcing approach, an additional force term is added for the prescribed motion. To show the basic idea of a direct-force approach, we may lump all the right-hand side terms of equation (12) into RHS, and discretize the governing equation with an Euler forward method, without loss of generality

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = (\text{RHS})^n$$  \hspace{1cm} (15)

Then a body force term being confined in the prescribed region $\Omega_{sp}$

$$\mathbf{f} = \begin{cases} -(\text{RHS})^n + \frac{1}{\Delta t}(\mathbf{V} - \mathbf{u}^n) & \text{in } \Omega_{sp} \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (16)

can be added to the right-hand side of equation (15) with a prescribed velocity $\mathbf{V}$. The velocity $\mathbf{V}$ is used to prescribe the exact moving trajectories of the control cells in $\Omega_{sp}$.

The unified fluid–solid dynamical equation

Based on the unified governing equations for FSI and the trajectory control with a direct-forcing approach, the final momentum equations for a combined fluid–solid system and with active control mechanism are

$$\frac{\partial \mathbf{u}_m}{\partial t} = -\nabla p_m - \mathbf{u}_m \cdot \nabla \mathbf{u}_m + \frac{1}{Re} \nabla^2 \mathbf{u} + \nabla \cdot (\chi_s \mathbf{r}) + \chi_c \mathbf{f}$$  \hspace{1cm} (17a)

$$\nabla \cdot \mathbf{u} = 0$$  \hspace{1cm} (17b)

$$\det \left( \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \right) = 1$$  \hspace{1cm} (17c)

where $\chi_s$ is defined earlier in equation (13) and $\chi_c$ is defined similarly but with values only within the domain $\Omega_{sp}$. It should be noted that equation (17b) is enforced in the whole computation domain $\Omega$, while equation (17c) is only employed on the solid domain $\Omega_s$.

**Numerical implementation**

**Solving momentum equation**

The numerical implementation of equation (17) for FSI is straightforward by adopting the conventional simulation methods for incompressible Navier–Stokes equations (e.g. projection method\(^{40}\)). In this study, all the computation in an Eulerian framework is based on a staggered Cartesian mesh, where the velocity components $u$, $v$, and $w$ are evaluated at the surface center of Cartesian cells and the pressure $p$ is evaluated at the cell center. A third-order Runge–Kutta/Crank–Nicholson scheme\(^{21,41}\) was chosen for time advancement. The non-temporal term for fluids

$$\mathbf{R}_f = -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}$$

discretized by a central difference scheme on the staggered Cartesian mesh. The direct forcing term $\chi_c \mathbf{f}$ in equation (17a) is specified by a prescribed velocity (for “forc skeletoned identified in a Lagrangian fashion for solids.

In order to account for the prescribed motions, the velocity is then projected back to the Cartesian mesh points for the calculation in an Eulerian framework. As a result, the corresponding Cartesian mesh points associated with each Lagrangian points need to be located. When uniform mesh is used, the surrounding Cartesian points of each Lagrangian point $(x_s, y_s, z_s)$ are easily located by floor function and ceiling function.\(^{21,41}\)

**Elastic stress calculation and projection**

Although the unified formulation is used for both solid structures and fluid flows, a Lagrangian mesh is required to keep the sharp interface between fluids and solids. Therefore, there are two steps for the numerical implementation of the elastic term for solids, $\mathbf{S} = \nabla \cdot (\chi_s \mathbf{r})$, in $\Omega_{sp}$: (1) stress calculation (on Lagrangian mesh) and (2) force projection (to Eulerian mesh). The force projection is critical to keep the accuracy of fluid flow and the sharpness of fluid–solid interface in simulation.

**Stress calculation.** First, elastic stress is calculated on Lagrangian mesh points using the neo-Hookean model in equation (14). The stress field inside the Lagrangian cell is obtained by using interpolation coefficients, i.e. shape functions. However, the stress field is discontinuous at the interface between the neighboring cells. This discontinuity in stress field is smoothed by using Zienkiewicz–Zhu (ZZ) patch, which uses
polynomials to approximate the target variables through a least-squares approach. \(^42\)

**Force projection.** The force project method proposed by Zhao et al. \(^21,41\) is used to distribute the smoothed solid stress from the Lagrangian frame back to the background Eulerian mesh with the trilinear interpolation as

\[
f_{ijk} = -\frac{1}{h_x h_y h_z} \int_{V_s} \nabla \psi_{ijk} \cdot \tau_s \, dV
\]

(18)

where \(f_{ijk}\) is the projected force on the background mesh, indices \(i, j,\) and \(k\) are the Gauss points in the Lagrange mesh corresponding to an Eulerian point, and \(\psi_{ijk}\) is the trilinear interpolation coefficient.

**Gauss quadrature for force projection.** The solid stress field at an arbitrary point \((x, y, z)\) can be approximated by \(\tau_s = a_0 + a_x x + a_y y + a_z z\), where the coefficients \(a_0, a_x, a_y,\) and \(a_z\) were determined by solving a set of linear equations. Therefore, the targeted integration becomes a linear combination of several polynomials \(a_{l+m+n} x^l y^m z^n\) \((l + m + n \leq 4)\) over the cell volume \(V_s\). The Gauss quadrature technique is adopted to calculate the integration of polynomials. Green–Gauss theorem can then further convert the volume integration of polynomials into a surface integral over \(V_s\):

\[
\int_{V_s} \left( \sum (a_{l+m+n} x^l y^m z^n) \right) dV = \int_{\partial V_s} \left( \sum \left( \frac{1}{l+1} x^{l+1} y^m z^n \right) n_s \right) dS
\]

(19)

Then Gauss quadrature rule was used to calculate the integration

\[
\int_{\partial V_s} \left( \sum \left( \frac{1}{l+1} x^{l+1} y^m z^n \right) n_s \right) dS = \sum_{q} \left\{ A_i \sum_{iGq} \left[ \sum (a_{l+m+n} x^{l+1} y^m z^n) \right] w_{Gq} \right\}
\]

(20)

For a two-dimensional integration, \(M^2\) integration points are needed to obtain the exact integration of polynomial with a degree of \(2M - 1\). Since the highest degree in equation (20) is 4, at least \(M^2 = 2.5^2 \approx 7\) integration points are needed in using Gauss quadrature technique. Additionally, to accomplish the Gauss quadrature rule, the outward normal vector of the surface \(\mathbf{n}\) of the corresponding Lagrangian cell should be found. Such normal vector of surfaces can be determined by using three points, \(P_1, P_2, P_3\), on the surface and another point, \(P_4\), which belongs to the cell but not the surface. Figure 2 shows the sketch of the four points, and the desired normal vector is calculated by

\[
\mathbf{n} = -\frac{(\vec{P}_{21} \times \vec{P}_{31}) \cdot (\vec{P}_{41})}{|\vec{P}_{21} \times \vec{P}_{31}|} (\vec{P}_{21} \times \vec{P}_{31})
\]

(21)

where \(\vec{P}_{ij} = (x_i, y_i, z_i) - (x_j, y_j, z_j)\) defines a vector pointing from \(P_i\) to \(P_j\).

**Cell division technique.** The original work of Zhao et al. \(^21,41\) laid the foundation to maintain the accuracy in the projection of elastic stresses from the Lagrangian mesh to the background Eulerian mesh. In particular, they tracked the intersection points of two meshes in a two-dimensional domain and used Gauss quadrature for an accurate integration of stresses in the projection to the Cartesian mesh of an Eulerian framework. In an extension from the two-dimensional algorithm to the current three-dimensional configuration, there are obvious challenges in the formulations including more complicated Gauss quadrature rules as explained earlier. However, the most challenging (and complicated) modification comes from the change of both cells, the Lagrangian mesh cell for solids and the Eulerian mesh cell for the unified fluid–solid equation, from two-dimensional geometries to three-dimensional geometries. The entire process to deal with the intersection of three-dimensional cells in the projection of stresses including special treatment for reducing numerical errors, dubbed as cell division technique, is described below.

The cell division starts as the Lagrangian cells being cut into sub-elements by the two-dimensional faces of the Cartesian (Eulerian) cells. The surfaces of new sub-elements belong to either the original surfaces of the Lagrangian cells or the Cartesian surfaces cutting into. The boundaries of sub-surfaces are defined by the
intersection points of all cut surfaces which are computed and saved whenever there is a relative motion of the two meshes. Figure 3 demonstrates a case of a tetrahedron being divided into three sub-elements by two background Cartesian surfaces.

The basic rule for cell division to generate sub-elements is simple and it adds almost negligible computational overhead. However, some extreme cases may appear in the implementation and require special treatments. Figure 4 shows an example of such special cases. The tetrahedron $OABC$ represents a typical Lagrangian solid cell for the computation of elastic stresses. The cell is cut by Cartesian surfaces $X_i$, $X_{i+1}$, $Y_j$, and $Y_{j+1}$ of the background Eulerian framework. As the Lagrangian cells move, the relative locations of surfaces are changed and result in different divisions.

The special case here assumes that the cell surface $OAC$ coincide to the location of one Cartesian surface $Z_k$. In theory, its sub-surface $DEFGHJK$ from the cut stays on the surface $Z_k$. However, the sub-surface is described by all intersection points which are computed numerically as the tetrahedron moving through the background surfaces. The characteristics of numerical process inevitably lead to misalignment of these intersection points at the level of round-off error. The newly defined sub-surface will not be perfectly coincided to the location of $Z_k$ or even perfectly flat. A possible scenario is shown in Figure 5, where some points (in red) are behind the surface $Z_k$ and other points (in blue) are in front of $Z_k$. The misalignment essentially defines some concave cells and leads to the invalidation of derivation and a failed computation. It is not hard to notice that the above scenario and anything similar only happen when a vertex of the tetrahedron cell is

![Figure 3](image1.png)  
**Figure 3.** Cell division of a tetrahedron cell by two background Cartesian surfaces.

![Figure 4](image2.png)  
**Figure 4.** Demonstration of a tetrahedron $OABC$ being cut by the background Cartesian surfaces $X_i$, $X_{i+1}$, $Y_j$, and $Y_{j+1}$, where the surface $OAC$ is assumed to be parallel to a Cartesian surface $Z_k$, and the cut of surfaces creates a new sub-surface $DEFGHJK$.

![Figure 5](image3.png)  
**Figure 5.** A possible scenario for the sub-surface $DEFGHJK$ being misaligned with the background Cartesian surface $Z_k$: red points are behind $Z_k$ and blue points are in front of $Z_k$. 
close enough to the background Cartesian surface. So, one remedy is to add a check process for all vertices before the cell division and prevent the situation in an ad hoc manner. It is noted that this challenge of misalignment does not appear in previous studies of two-dimensional configurations. The reason is simply that two intersection points guarantee a straight line to define edges (of sub-elements) in a two-dimensional case but multiple intersection points will not numerically guarantee a flat surface to define sub-surfaces in a three-dimensional case.

**Benchmark studies**

Two benchmark cases were studied for the validation of different aspects in the development of the current algorithm: (1) a two-dimensional viscoelastic particle in shear flow is simulated to validate the treatment of solid in the unified formulation and its interaction with surrounding fluids; (2) the flow passing a rigid sphere, as a classical flow dynamics problem with embedded solid structure, to further validate the extension of the algorithm from two-dimensional to three-dimensional.

**Viscoelastic particle in shear flow**

To validate the accuracy of treating elastic stresses in the unified Eulerian framework and the associated fluid–solid interaction in the same framework, the algorithm is applied for the simulation of a two-dimensional floppy viscoelastic particle in shear flow (Figure 6). It is noted that the immersed boundary approach used here is fast and different from the algorithm in the original study of the same problem.

Typically, the inertia term is neglected for creeping-flow problems where the Reynolds number, $Re = \frac{\rho \gamma a^2}{\mu}$, based on shear rate $\gamma$ and particle radius $a$ is very small. The governing equation becomes

$$\frac{\partial \mathbf{u}_m}{\partial t} = -\nabla p_m + \frac{1}{Re} \nabla^2 \mathbf{u}_m + \nabla \cdot (\lambda_s \mathbf{\tau}_s)$$

where the stress $\mathbf{\tau}_s$ is based on the same neo-Hookean model in equation (14). Figure 6 shows the basic setup of the problem that a viscoelastic particle initially with circular shape is moving, deforming, and rotating under the interaction with the surrounding Couette flow. Three different types of motions, shown in Figure 7, are determined by the particle’s initial aspect ratio $\omega_0$ and capillary number $G = \frac{\gamma}{\mu \mu_s}$: tank-treading, swinging (trembling), and tumbling.

The current immersed boundary approach was able to simulate all different types of motions. The computational domain is $[-2,2] \times [-2,2]$ with the mesh of $200 \times 200$. For a more quantitative benchmark, parametric study was performed for different aspect ratio and capillary number to find the boundary in parametric space to separate the region of trembling motion and the region of tumbling motion. As shown in Figure 8, the boundary computed by the current approach matches well with the study of others using a different approach.

**Incompressible flow passing a rigid sphere**

An incompressible flow passing a stationary rigid sphere, as a classical benchmark case, has been investigated theoretically, numerically, and experimentally. This case was used to further validate the current approach of using a unified fluid–solid equation in a three-dimensional configuration which was not numerically guarantee a flat surface to define sub-surfaces in a three-dimensional case.

---

**Figure 6.** Sketch of a viscoelastic particle in a shear flow.

**Figure 7.** Different types of motion of viscoelastic particle in a shear flow: (a) tank-treading, (b) trembling, (c) tumbling.
not included in the first benchmark case. There are two reasons to consider only a rigid sphere in this validation: one is that the rigid sphere case is well studied and suitable for benchmark purpose, and the other is that the elastic formulation has already been validated in the first benchmark but the direct-forcing term (for rigid body and control gears) has not been validated yet.

In this study, the non-dimensional diameter of the sphere is $D = 1$, and the free stream velocity is $u_\infty = 1$. The dynamic viscosity of fluid $\mu_f$ is varied for different Reynolds number $Re = \rho u_\infty D/\mu_f$: $Re = 50, 100, 150, 200, 250, and 300$, which are the same as the numbers used by Johnson and Patel.$^{47}$ The computational domain is $[-10, 10] \times [-4, 4] \times [-4, 4]$ with the mesh of $500 \times 100 \times 100$. As shown in Figure 9, at low Reynolds numbers $Re = 50, 100, 150, and 200$, the downstream flow is axisymmetric and different wake patterns are compared quantitatively between the simulations by the current algorithm and the study of Johnson and Patel.$^{47}$ When Reynolds number increases to 250 and 300, the flow becomes asymmetric and a clear structure of hairpin vortex appears. Figure 10 compares the hairpin structure computed by Johnson and Patel$^{47}$ and the current approach at Reynolds number 300. A more quantitative comparison is given in Figure 11, where the drag coefficient

$$C_d = \frac{F_d}{\frac{1}{2} \rho U_\infty^2 D}$$

(23)

Figure 8. Regions of trembling motion and tumbling motion for a two-dimensional viscoelastic particle in a shear flow: a comparison of our results to the results from literature$^{45}$ using a different approach.

Figure 9. Comparison of the vorticity contours ($\omega_x$) showing different wake patterns for the flow passing a sphere: (left) the numerical simulation by Johnson and Patel$^{47}$ and (right) the current simulation using an immersed boundary approach with a unified formulation.
is computed for different Reynolds numbers. A good match is observed between our results and other numerical and experimental studies.47,48

**Application on the simulation of a MAV with flexible wings**

Established by the aforementioned benchmark cases, the current approach is ready for the simulation to study the aerodynamics of a MAV with flexible flapping wings, where the challenges of three-dimensionality, elastic structures, and direct-forcing controls all exist.

The computation domain is $-1 < x < 3$, $-1 < y < 1$, and $0 < z < 2$, with a mesh of $200 \times 100 \times 100$. A uniform staggered mesh is used in an Eulerian framework, and quadratic tetrahedron cells are used for a Lagrangian mesh.

The wing model used in this simulation is based on the measurements from a wing of a toy MAV (by Interactive Toy Concepts Ltd). Figure 12 compares the original toy model to the numerical model with flapping motion. The lengths in the figure are non-dimensionalized by the length of the wing leading edge. The green part of the numerical model moves passively and described by the elastic terms in the unified formulation, and the blue part of the model is prescribed by the direct-forcing terms either to control the flapping motion (the blue "skeleton line") or to fix the wing structure to the body (the blue patch).

The flapping of wing is driven by the rotation of the blue "skeleton" line described by the flapping angle $\theta$ (marked in Figure 12) and the angular velocity $\omega = \dot{\theta}$

$$\theta = \frac{a}{L} \sin(2.0K\pi t) + \theta_0 \quad (24a)$$

$$\omega = \frac{2.0aK\pi}{L} \cos(2.0K\pi t) \quad (24b)$$

where $a = 0.1$, $K = 1.0/\pi$, $L = 18/(19.5\pi)$, and $\theta_0 = 10.5^\circ$. The parameters were chosen to match with the realistic data of the toy as close as possible. In the study, two cases are simulated to understand the impact from two different angles of attack, $\alpha = 0^\circ$ and $\alpha = 14^\circ$.

As shown in Figure 13, the wing deformation of the case with angle of attack $\alpha = 0^\circ$ is larger in both the deformation amplitude and the size of area with obvious deformation. For more quantitative comparison, the deformation is painted on wing surfaces by contours measured by the distance between points on the deformed wing surface and their projection on an imaginary rigid wing surface under the same motion. In fact, this larger deformation of the flexible wing shown in the case of $\alpha = 0^\circ$ may help to limit the width of vortex structures shedding as shown in Figure 14. It is also shown in Figure 14 that higher value of streamwise vorticity (marked in red), $\omega_x \leq 1$, is being sustained further downstream in wake structures when the angle of attack is higher, as an indication of generally stronger shedding vortices in the case with higher angle of attack.

Figure 15 compares the overall aerodynamic performance of these two cases. It is not surprising to see the non-dimensional lift increases with the increase of...
angle of attack, with the averaged non-dimensional lift $F_L = 3.92 \times 10^{-2}$ for $\alpha = 0^\circ$ and $F_L = 4.58 \times 10^{-2}$ for $\alpha = 14^\circ$. The effect on drag is, however, more complicated. For the case of $\alpha = 14^\circ$, though the shedding vortex is stronger, the resulted thrust is actually smaller as the vortex structures are diverted sideward. As the result, the case of $\alpha = 0^\circ$ actually achieves a thrust with an averaged non-dimensional drag $F_D = -1.44 \times 10^{-2}$ (the negative sign indicates a thrust), while the case of $\alpha = 14^\circ$ stays in the drag region with an averaged non-dimensional drag $F_D = 7.52 \times 10^{-3}$ under current control parameters.
Conclusions

In this study, a strongly coupled three-dimensional algorithm for FSI is constructed through a monolithic immersed boundary method. The motions of flows and structures as well as its interaction are simulated under a unified Eulerian framework, while the location of structure is tracked accurately by a Lagrangian mesh. The interpretation and projection between the Eulerian and Lagrangian frameworks are calculated through a cell division technique and Gauss quadrature. It should be noted that the interpretation and projection between the Lagrangian and Eulerian frameworks are meant for the evaluation of solid elastic stress, and the time advancement for both fluid and solid is performed in an Eulerian framework (i.e. the unified governing equation). It is intrinsically a monolithic immersed boundary algorithm for fluid–solid interactions in an Eulerian description.

The monolithic FSI algorithm was validated by comparing the simulation results with other numerical simulations and experimental measurements in literature using two classical cases to cover two- and three-dimensional configurations and rigid and flexible structures: the flow passing a three-dimensional rigid stationary sphere and a two-dimensional floppy viscoelastic particle in a shear flow. The qualitative and
quantitative comparisons agree well with the existing literature in both cases.

The monolithic FSI algorithm was then used to study three-dimensional flexible flapping wings with different kinematics and parameters in a forward flight. The simulations at an angle of attack $\alpha = 0^\circ$ and $\alpha = 14^\circ$ showed the impact from the angle of attack to wing deformation, wake vortex structures, and eventually aerodynamic performances of a flexible flapping wing in forward flight. It is shown that the vortex street in wake tends to spread spanwise at larger angle of attack while the strength increases slightly. Further investigation shows that, at lower angle of attack, the larger deformation of flexible wings helps to align the vortex street downstream and reduce the wake spreading, and therefore increase the vortex-generated thrust. Overall, the case with angle of attack $\alpha = 0^\circ$ provides more thrust, and the case of larger angle of attack at $\alpha = 14^\circ$ provides more lift.

**Declaration of conflicting interests**
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**
The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: MW and TY gratefully appreciate the support by the Air Force Office of Scientific Research under grant number W911NF-08-2-0004. JC would like to acknowledge support for the research, authorship and/or publication of this article.

**ORCID iD**
Mingjun Wei https://orcid.org/0000-0001-7757-2355

**References**

1. Knoller R. Die gesetze des luftwiderstandes. *Flug- und Motortechnik (Wien)* 1909; 3: 1–7.
2. Betz A. Ein beitrag zur erklung des segelfluges. *Z Flugtechnik Motorluftschiffahrt* 1912; 3: 269–272.
3. Garrick IE. Propulsion of a flapping and oscillating airfoil. NACA-TR-567, 1936, National Advisory Committee for Aeronautics, Langley Aeronautical Lab.
4. Jones KD, Dohring CM and Platzer MF. Experimental and computational investigation of the Knoller–Betz effect. *AIAA J* 1998; 36: 1240–1246.
5. Anderson JM, Streitlien K, Barrett DS, et al. Oscillating foils of high propulsive efficiency. *J Fluid Mech* 1998; 360: 41–72.
6. Lewin GC and Haj-Hariri H. Modelling thrust generation of a two-dimensional heaving airfoil in a viscous flow. *J Fluid Mech* 2003; 492: 339–362.
7. Buchholz HJH and Smits AJ. The wake structure and thrust performance of a rigid low-aspect-ratio pitching panel. *J Fluid Mech* 2008; 603: 331–365.
8. Heathcote S and Gursul I. Flexible flapping airfoil propulsion at low Reynolds numbers. *AIAA J* 2007; 45: 1066–1079.
9. Ishihara D, Horie T and Denda M. A two-dimensional computational study on the fluid–structure interaction of wing pitch changes in dipteran flapping flight. *J Exp Biol* 2009; 212: 1–10.
10. Yang T, Wei M and Zhao H. Numerical study of flexible flapping wing propulsion. *AIAA J* 2010; 48: 2909–2915.
11. Xu M, Wei M, Yang T, et al. Nonlinear structural response in flexible flapping wings with different density ratio. In: 49th *AIAA aerospace sciences meeting*, Orlando, FL, AIAA Paper 2011–376, 4–7 January 2011.
12. Xu M, Wei M, Li C, et al. Adjoint-based optimization of flapping plates hinged with a trailing-edge flap. *Theor Appl Mech Lett* 2015; 5: 1–4.
13. Xu M, Wei M, Yang T, et al. An embedded boundary approach for the simulation of a flexible flapping wing at different density ratio. *Eur J Mech B Fluids* 2016; 55: 146–156.
14. Tian FB, Dai H, Luo H et al. Fluid16;j.eurom interaction involving large deformations: 3d simulations and applications to biological systems. *J Comput Phys* 2014; 258: 451–469.
15. Xu M and Wei M. Using adjoint-based optimization to study kinematics and deformation of flapping wings. *J Fluid Mech* 2016; 799: 56–99.
16. Xu M, Wei M, Li C, et al. Adjoint-based optimization for thrust performance of three-dimensional pitching-rolling plate. *AIAA J* 2018 (accessed 26 December 2018).
17. Donea J, Giuliani S and Hallaux J. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Comput Methods Appl Mech Eng* 1982; 33: 689–723.
18. Zhu Q. Numerical simulation of a flapping foil with chordwise or spanwise flexibility. *AIAA J* 2007; 45: 2448–2457.
19. Luo H, Mittal R, Zheng X, et al. An immersed-boundary method for lowed-bound interaction in biological systems with application to phonation. *J Comput Phys* 2008; 227: 9303–9332.
20. Xu L, Tian FB, Young J, et al. A novel geometry-adaptive Cartesian grid based immersed boundary-lattice Boltzmann method for fluid–structure interactions at moderate and high Reynolds numbers. *J Comput Phys* 2018; 375: 22–56.
21. Zhao H, Freund JB and Moser RD. A fixed-mesh method for incompressible flow-structure systems with finite solid deformations. *J Comput Phys* 2008; 227: 3114–3140.
22. Boffi D, Gastaldi L, Heltai L, et al. On the hyper-elastic formulation of the immersed boundary method. Comput Methods Appl Mech Eng 2008; 197: 2210–2231.
23. Peskin CS. Numerical analysis of blood flow in the heart. J Comput Phys 1977; 25: 220–252.
24. Mohd-Yusof J. Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. Annual Research Briefs – 1997, Center for Turbulence Research, 1997.
25. Lai MC and Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J Comput Phys 2000; 160: 705–719.
26. Fadlun EA, Verzicco R, Orlandi P, et al. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys 2000; 161: 35–60.
27. Silva ALE, Silveria-Neto A and Damasceno J. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. J Comput Phys 2003; 189: 351–370.
28. Tseng Y and Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. J Comput Phys 2003; 192: 593–623.
29. Dong H, Mittal R and Najjar F. Wake topology and hydrodynamic performance of low aspect-ratio flapping foils. J Fluid Mech 2006; 566: 309–343.
30. Mittal R, Dong H, Bozkurttas M, et al. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. J Comput Phys 2008; 227: 4825–4852.
31. Yang T, Wei M and Zhao H. Numerical study of flexible flapping wing propulsion. In: 48th AIAA aerospace sciences meeting and exhibit, Orlando, FL, AIAA paper 2010-0553, 4-7 January 2010.
32. Yin B and Luo H. Effect of wing inertia on hovering performance of flexible flapping wings. Phys Fluids 2010; 22: 111902.
33. Chen J. Extension of nonlinear Onsager theory of irreversibility. Acta Mech 2013; 224: 3153–3158.
34. Eringen AC. Mechanics of continua. Florida, USA: Krieger Publishing Company, 1980.
35. Gonzalez O and Stuart AM. A first course in continuum mechanics. Cambridge: Cambridge University Press, 2008.
36. Wang CC. A new representation theorem for isotropic functions. Part I, Part ii. Arch Ration Mech Anal 1969; 36: 166–223.
37. Wang CC. Corrigendum to representation for isotropic functions. Arch Ration Mech Anal 1970; 43: 392–395.
38. Macosko CW. Rheology: principles, measurements, and applications. Germany: Wiley-VCH, 1994.
39. Ogden RW. Nonlinear elastic deformations. USA: Courier Dover Publications, 1997.
40. Brown DL, Cortez R and Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. J Comput Phys 2001; 168: 464–499.
41. Zhao H. A fixed-mesh flow-structure solver for biological systems with large solid deformations. PhD Thesis, Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, Urbana, IL, 2006.
42. Zienkiewicz OC and Zhu JZ. The superconvergent patch recovery and a posteriori error estimates. Part 1: the recovery technique. Int J Numer Methods Eng 1992; 333: 1331–1364.
43. Yang T. Numerical study of flexible flapping wings. PhD Thesis, Department of Mechanical and Aerospace Engineering, New Mexico State University, Las Cruces, NM, 2012.
44. Gao T, Hu HH and Castaneda PP. Rheology of a suspension of elastic particles in a viscous shear flow. J Fluid Mech 2011; 687: 209–237.
45. Gao T, Hu HH and Castaneda PP. Shape dynamics and rheology of soft elastic particles in a shear flow. Physical Review Letters 2012; 108 (5): 058302.
46. Kim I and Pearlstein AJ. Stability of the flow past a sphere. J Fluid Mech 1990; 211: 73–93.
47. Johnson TA and Patel VC. Flow past a sphere up to a Reynolds number of 300. J Fluid Mech 1999; 378: 19–70.
48. Roos F and Willmarth WW. Some experimental results on sphere and disk drag. AIAA J 1971; 9: 285–291.
49. Szaltys P, Chrust M, Przadka A, et al. Nonlinear evolution of instabilities behind spheres and disks. J Fluids Struct 2012; 28: 483–487.