DOI: 10.14529/jsfi170202

An Application of GPU Acceleration in CFD Simulation for Insect Flight

Yang Yao[1], *Khoon-Seng Yeo*[1]

The mobility and maneuverability of winged insects have been attracting attention, but the knowledge on the behavior of free-flying insects is still far from complete. This paper presents a computational study on the aerodynamics and kinematics of a free-flying model fruit-fly. An existing integrative computational fluid dynamics (CFD) framework was further developed using CUDA technology and adapted for the free flight simulation on heterogeneous clusters. The application of general-purpose computing on graphics processing units (GPGPU) significantly accelerated the insect flight simulation and made it less computational expensive to find out the steady state of the flight using CFD approach. A variety of free flight scenarios has been simulated using the present numerical approach, including hovering, fast rectilinear flight, and complex maneuvers. The vortical flow surrounding the model fly in steady flight was visualized and analyzed. The present results showed good consistency with previous studies.

Keywords: insect flight, free flight, general-purpose computing on graphics processing units (GPGPU), computational fluid dynamics (CFD), flapping-wing aerodynamics.

Introduction

Winged insects were the first animals to evolve flight locomotion. It is common to find them hovering, flying sideways, landing up-side down, and executing rapid changes in flight speed and direction. The capability of flight significantly contributes to insect diversity and abundance, as it has permitted access to various ecological resources and rapid escape from predators. The unparalleled mobility and maneuverability of winged insects have long captured the interest of researchers in the area of biomechanics and aerodynamics. It is believed that, by explaining the biomechanics of insect flight, scholars can yield insight to morphological evolution of insects and their ecological roles.

In the past decade, several unsteady aerodynamic phenomena have been discovered to be responsible for high agility and lift enhancement of flapping wing insect [12]. With the development of CFD methods, and the continuous improvement of computing technology, it has become feasible to study free flying insects and to visualize and analyze the complex unsteady aerodynamics using numerical approaches [9, 13]. The use of CFD provides a convenient approach to model various natural flapping wing flyers, since the geometries of the flyers and the environment the flyers encounter can be easily altered in a CFD model once it has been developed. Sun & Wang [14] adopted CFD simulation results in their stability analysis of flapping wing flight. Gao *et al.* [4] investigated the motion of a model fruit fly with six degrees of freedom based on CFD computation, and simulated passive dynamic motions of the flyer under small perturbation. They argued that an active control with sufficiently fast response is needed to maintain the stability of the flight under disturbance. Kolomenskiy *et al.* [6] simulated taking off flight of fruit-fly with two degrees of freedom using a pseudo-spectral method integrated with a flight dynamics solver.

However, the simulation of insects in free hovering and rectilinear flight and intricate maneuvering sequences remains highly challenging due to their dynamical complexities and high computational cost [17]. General-purpose computing on graphics processing units (GPGPU) is

[1] Dept. of Mechanical Engineering, National University of Singapore, Singapore

an emerging heterogeneous computing technique that performs massive parallelization on graphics processing unit (GPU). This technology is designed to achieve high float point operation rates and is suitable for acceleration of numerically intensive CFD computations. Here, we adopted the NVIDIA CUDA technology to accelerate the simulation of a free flying insect model and investigated its long-term behaviors in different flight scenarios.

1. Numerical Methods

1.1. Governing Equations and Numerical Discretization

The dynamics of the fluid driven by flying insects is governed by the incompressible non-dimensional Navier-Stokes (NS) equations, given here in an arbitrary Lagrangian-Eulerian (ALE) form:

$$\partial_t \mathbf{u} + (\mathbf{u} - \mathbf{u}^g) \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1b}$$

where $\mathbf{u}$ and p are the non-dimensional velocity and pressure fields respectively of the time-varying fluid domain , and $\mathbf{u}^g$ denotes the convection velocity of the computational node. The convection velocity $\mathbf{u}^g$ for meshfree nodes follow the movement of the boundary surfaces.

An implicit form projection method proposed in [3] is adopted in this work to advance the above governing equations in time. The discretized form of the momentum equations (1a) is written as:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{1}{2} \left\{ \left[-(\mathbf{u} - \mathbf{u}^g) \cdot \nabla \mathbf{u} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u} \right]^{n+1} + \left[-\nabla p - (\mathbf{u} - \mathbf{u}^g) \cdot \nabla \mathbf{u} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u} \right]^{n} \right\}, \tag{2a}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{2} \nabla p^{n+1}, \tag{2b}$$

where superscripts n and $n+1$ denote time level. The $\mathbf{u}^*$ is an approximation of the velocity field $\mathbf{u}^{n+1}$. Taking the divergence of (2b) yields the following pressure-Poisson equation, which allows us to obtain the new pressure field p^{n+1} from $\mathbf{u}^*$:

$$\nabla^2 p^{n+1} = \frac{2}{\Delta t} \nabla \cdot \mathbf{u}^*. \tag{3}$$

Boundary conditions given in [1], [16] and [17] have been implemented here to close the above discretized governing equations. The fractional step equations and boundary conditions are solved iteratively to advance the flow field to new time level. The pressure-Poisson equation is solved by a hybrid Jacobi-BiCGSTAB solver to attain the solution of the pressure field p^{n+1}.

1.2. SVD-GFD Scheme

Following the methodology developed in [3] and [15], the flow equations are solved on a hybrid Cartesian cum meshfree grid system, wherein the body and wings of the flyer, and their near fluid neighborhood are discretized by meshfree nodes. A set of computational mesh for a fruit fly flyer that used in this study is shown in fig. 1. The meshfree nodes around the boundaries convect with the motion of the body and wings. Generalized finite difference (GFD) scheme is adopted here to approximate derivatives involved in the solution on the meshfree nodes.

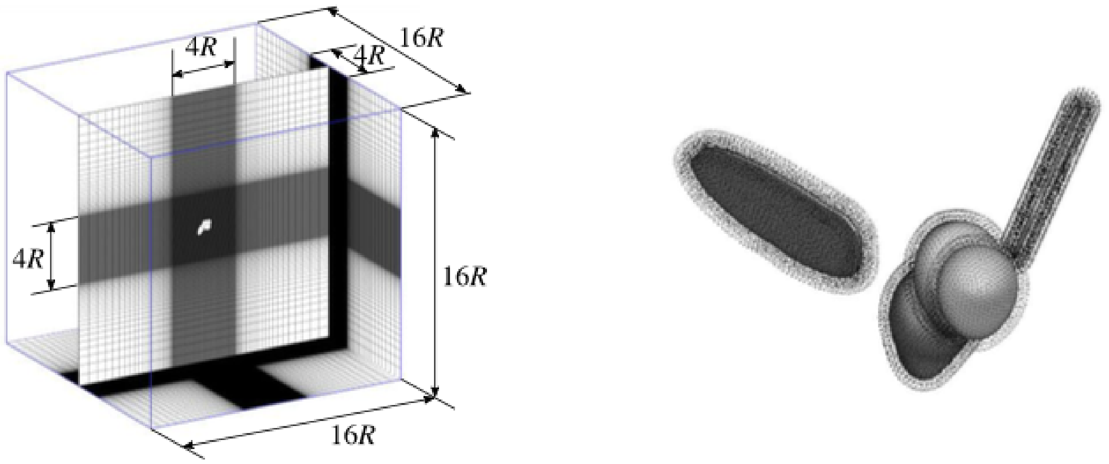


Figure 1. Schematic view of computational domain (R is the wing length of the flyer). Left: background grid; Right: meshfree nodes around insect model

The GFD method is based on the Taylor series expansion. The three dimensional Taylor series expansion of the function $f(\mathbf{x})$ at $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_1$ is given in terms of the derivatives at node $\mathbf{x}_0$ to order m by:

$$f(\mathbf{x}_1) = f(\mathbf{x}_0) + \sum_{1\leq j_1+j_2+j_3\leq m-1} \frac{\Delta x_1^{j_1}\Delta y_1^{j_2}\Delta z_1^{j_3}}{j_1!j_2!j_3!}\left[\partial_x^{j_1}\partial_y^{j_2}\partial_z^{j_3}\right] + \mathcal{O}(|\Delta\mathbf{x}_1|^m), \tag{4}$$

where ∂_x, ∂_y and ∂_z denote the partial derivatives with respect to the coordinate variables.

If the values of the function $f(\mathbf{x})$ are known at n nodes $\mathbf{x}_i = \mathbf{x}_0 + \Delta\mathbf{x}_i$ $(i = 1, 2, \cdots, n)$ in the vicinity of the central node $\mathbf{x}_0$, we can obtain n equations about the derivatives $[\partial_x^{j_1}\partial_y^{j_2}\partial_z^{j_3}f]$ at $\mathbf{x}_0$. By truncating the Taylor series after the third-order derivatives ($m = 4$), the equations can form following linear system:

$$\Delta\mathbf{f}_{n\times 1} = \mathbf{S}_{n\times 19}\partial\mathbf{f}_{19\times 1}, \tag{5}$$

where

$$\mathbf{f}_{n\times 1} = \begin{bmatrix} f_1 - f_0 & f_2 - f_0 & \cdots & f_n - f_0 \end{bmatrix},$$

$$\mathbf{S}_{n\times 19} = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 & 0.5\Delta x_1^2 & 0.5\Delta y_1^2 & 0.5\Delta z_1^2 & \cdots & \Delta x_1\Delta y_1\Delta z_1 \\ \Delta x_2 & \Delta y_2 & \Delta z_2 & 0.5\Delta x_2^2 & 0.5\Delta y_2^2 & 0.5\Delta z_2^2 & \cdots & \Delta x_2\Delta y_2\Delta z_2 \\ \vdots & & & & & & \ddots & \vdots \\ \Delta x_n & \Delta y_n & \Delta z_n & 0.5\Delta x_n^2 & 0.5\Delta y_n^2 & 0.5\Delta z_n^2 & \cdots & \Delta x_n\Delta y_n\Delta z_n \end{bmatrix},$$

$$\partial\mathbf{f}_{19\times 1} = \begin{bmatrix} \partial_x & \partial_y & \partial_z & \cdots & \partial_x\partial_y\partial_z \end{bmatrix}^T f|_{\mathbf{x}_0}.$$

Should the matrix $\mathbf{S}$ be a non-singular square matrix, an approximate solution of the derivatives $\partial\mathbf{f}_{19\times 1}$ at the central node may be determined by $\partial\mathbf{f}_{19\times 1} = [\mathbf{S}_{n\times 19}^{-1}]\Delta\mathbf{f}_{n\times 1}$ with formal accuracy of $\mathcal{O}(|\Delta\mathbf{x}|^3)$ and $\mathcal{O}(|\Delta\mathbf{x}|^2)$ for first order and second order derivatives, respectively. This involves finding the pseudo-inverse of the coefficient matrix $\mathbf{S}$. However, $\mathbf{S}$ tends to be ill-conditioned due to irregular arrangements of the neighboring nodes, especially when two or more nodes in the neighborhood are located very close to each other, making its inversion impossible to be implemented in practice.

The least-squares method and singular value decomposition (SVD) technique are classical approaches that give the best-fitting solution/approximation for over-determined systems. In the present 3D simulation, we implemented the SVD-GFD method presented by [1]. In the algorithm, a regularization of the solution by setting the very small singular values of **S** to zero is performed in order to avoid ill-conditioning. The system may become under-determined when the small singular values were omitted, but the SVD method ensures a solution with minimum L_2 error can be obtained. The SVD scheme with regularization has been found to be robust and accurate in the practice of 3D simulations for natural flyers and swimmers [11, 17, 18].

1.3. Nodal Selection

To minimize the approximation error caused by arbitrary node distribution in the meshfree cloud, we adopted a minimum of 40 supporting nodes in the computation of $\partial\mathbf{f}_{19\times1}$ for the central node and applied a novel nodal selection criterion to improve the configuration of the supporting nodes. In the current scheme, all neighboring nodes inside a predetermined support region are collected as candidates for supporting nodes and are sorted in distance order, from the closest to the furthest. A sphere with radius r_{inf} and a cone with angle α are assigned to each node including the central one as its zone of influence (fig. 2):

$$r_{inf} = \begin{cases} r_{inf0}, & \text{for central node,} \\ r_0 \sin\alpha, & \text{for nodes with } \bar{r}_i \le r_0, \\ \bar{r}_i \sin\alpha, & \text{for nodes with } \bar{r}_i > r_0, \end{cases} \tag{6}$$

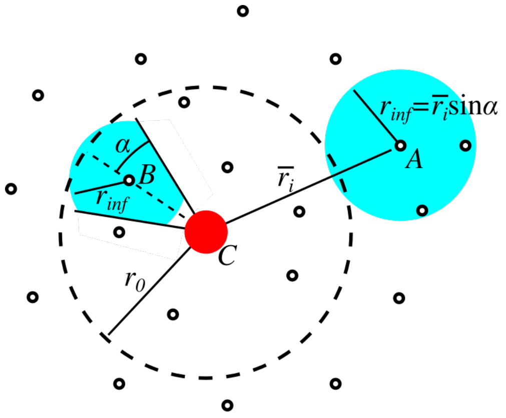


Figure 2. Schematic drawing showing region of influence of nodes A and B in the neighborhood of central node C

Moreover, it was pointed out that all the supporting nodes should be visible to the central node to ensure that the derivatives can be correctly discretized [11]. As defined in geometry, a supporting node is visible to the central node if the line segment that connects them does not intersect any boundary surface. A visibility check base on ray-casting algorithm is applied on the nodes close to a sharp edged object or a thin air foil like the insect wing, as their candidate nodes are probably selected across the boundary surface. As shown in fig. 3, the selected supporting nodes satisfy the visibility criterion for nodes near solid surface.

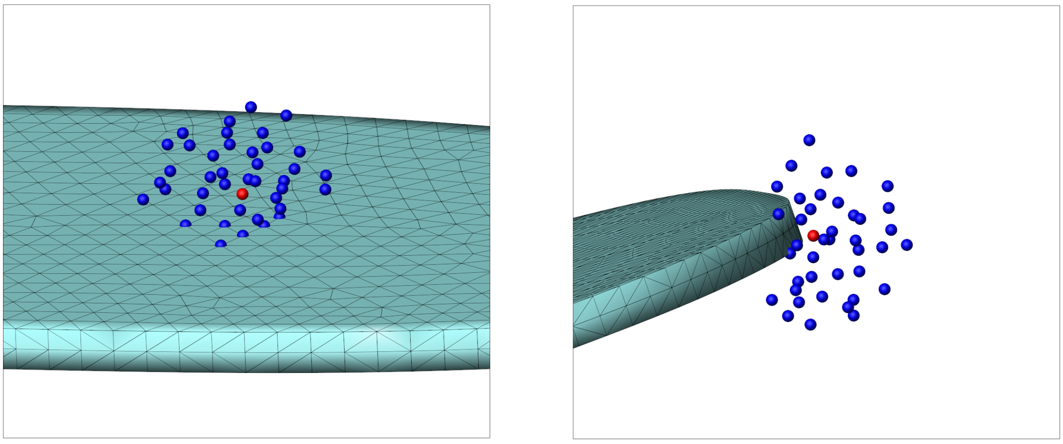


Figure 3. Resultant supporting nodes (blue) of central nodes (red) near solid surface

2. Implementation

The motion of the model insect in free flight is motivated by aerodynamic forces and is subject to Newton's law. With the rigid body assumption, a flapping wing flyer in free flight can be treated as a multi-body system that has six degrees of freedom (6-DoF). The wing mass of fruit fly (*D. melanogaster*) is about several micro grams and contributes to less than 0.5% of the total mass [2]. Hence, we here ignored the wing mass of the model fruit fly in the present study. Moreover, we supposed that the density of insect body is homogeneous, and computed the fly's inertia tensor and centre-of-mass (CoM) based on this homogeneous density distribution. The equations of motion for the 6-DoF flight are given by:

$$\begin{cases} m\dot{\mathbf{V}}_C(t) = m\mathbf{g} + \mathbf{F}_A(t) \\ \dot{\mathbf{X}}_C(t) = \mathbf{V}_C(t) \\ \dot{\mathbf{L}}_C(t) = \mathbf{M}_A(t) \\ \dot{\mathbf{\Theta}}_C(t) = \mathbf{K}(t) \cdot \boldsymbol{\omega}_C(t) \end{cases}, \tag{7}$$

where $\mathbf{V}_C$, $\mathbf{X}_C$, $\mathbf{L}_C$ and $\mathbf{\Theta}_C$ denote the linear velocity, body position, angular momentum and body orientation vector of the flyer at its body centre-of-mass (CoM) C in the global frame, respectively, $\mathbf{K}$ is a transformation matrix, $\boldsymbol{\omega}_C$ is the flyer's angular velocity determined by its angular momentum and moment of inertia, $\mathbf{g}$ is the gravity vector, and $\mathbf{F}_A$ and $\mathbf{M}_A$ are the aerodynamic forces and moments acting on the model fly respectively.

The CFD scheme presented in the last section was adopted here to resolve the flow field surround the flapping wing flyer, in order to model the aerodynamic forces from the pressure and viscous stress acting on the flyer surfaces. The surface configuration $\Gamma(\mathbf{G}(t)|\mathbf{V}_C, \mathbf{X}_C, \boldsymbol{\omega}_C, \mathbf{\Theta}_C)$ is time-dependent and comprise information about the geometry of the model fly (body and wings) $\mathbf{G}(t)$ in its body frame as well as the information on the state of motion $\boldsymbol{\xi} = [\mathbf{V}_C, \mathbf{X}_C, \boldsymbol{\omega}_C, \mathbf{\Theta}_C]^T$ of the flyer in the global frame. $\Gamma(t)$ determines the boundary condition in the CFD computation while is in turn solved from (7) using the resultant aerodynamic forces. Hence, the solution of $\Gamma(t)$ involves fluid-body interaction (FSI) and is essentially an implicit problem.

This FSI problem was solved through a fixed-point iteration on $\Gamma(t)$ using a predictor-corrector method based on the 4-step implicit Adams-Moulton scheme in the present study. The reason why multistep methods were selected is that evaluating aerodynamic forces by the CFD method in intermediate steps requires extra computing time and storage, and the time step size used in the CFD part is small enough to ensure stability of the selected multistep method. Moreover, the consistent time integration intervals allow the iterative projection method in CFD part to be embedded into the 6-DoF motion solver to achieve a tightly coupled fluid-body interaction simulation.

The calculation procedures are listed below and showed in fig. 4 in details:

1. Predicator step (P): at time step $n+1$, compute the current dynamic state of insect $\boldsymbol{\xi}^{n+1}$ from the previous dynamic state using the 2-step Adams-Bashforth scheme:

$$\boldsymbol{\xi}^{n+1,0} = \boldsymbol{\xi}^{n} + \Delta t \left(\frac{3}{2}\dot{\boldsymbol{\xi}}^{n} - \frac{1}{2}\dot{\boldsymbol{\xi}}^{n-1} \right); \tag{8}$$

2. Evaluator step (E): update the flow field state Φ from Φ^n to Φ^{n+1} using the CFD solver with boundary conditions given by $\boldsymbol{\xi}^{n+1}$, then evaluate $\dot{\boldsymbol{\xi}}^{n+1}$ from the latest available solution of Φ;
3. Corrector step (C): correct $\boldsymbol{\xi}^{n+1}$ using the 4-step implicit Adams-Moulton scheme:

$$\boldsymbol{\xi}^{n+1,j} = \boldsymbol{\xi}^{n} + \Delta t \left(\frac{3}{8}\dot{\boldsymbol{\xi}}^{n+1,j} + \frac{19}{24}\dot{\boldsymbol{\xi}}^{n} - \frac{5}{24}\dot{\boldsymbol{\xi}}^{n-1} + \frac{1}{24}\dot{\boldsymbol{\xi}}^{n-2} \right). \tag{9}$$

The above algorithm ran in iterative P(EC)^{k}E mode, with one predictor step and k corrector iterations, to solve the rigid body dynamics of insect flight.

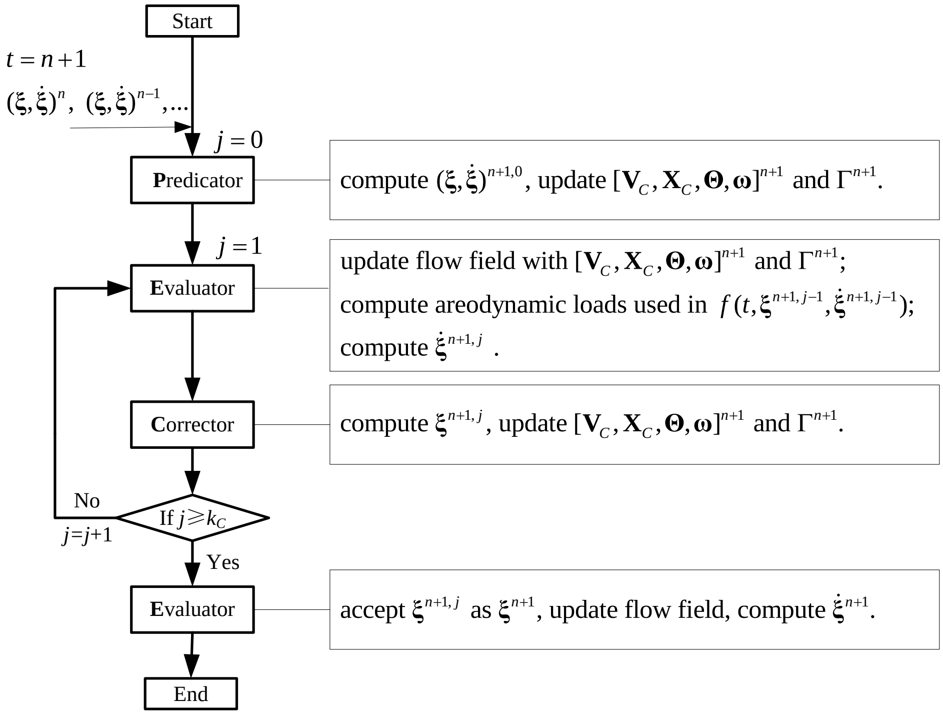

Start
$t = n+1$
$(\xi,\dot{\xi})^n,\ (\xi,\dot{\xi})^{n-1},...$
$j=0$
Predicator
compute $(\xi,\dot{\xi})^{n+1,0}$, update $[\mathbf{V}_C,\mathbf{X}_C,\boldsymbol{\Theta},\boldsymbol{\omega}]^{n+1}$ and Γ^{n+1}.
$j=1$
Evaluator
update flow field with $[\mathbf{V}_C,\mathbf{X}_C,\boldsymbol{\Theta},\boldsymbol{\omega}]^{n+1}$ and Γ^{n+1};
compute areodynamic loads used in $f(t,\xi^{n+1,j-1},\dot{\xi}^{n+1,j-1})$;
compute $\dot{\xi}^{n+1,j}$.
Corrector
compute $\xi^{n+1,j}$, update $[\mathbf{V}_C,\mathbf{X}_C,\boldsymbol{\Theta},\boldsymbol{\omega}]^{n+1}$ and Γ^{n+1}.
No
j=j+1
If $j \geqslant k_C$
Yes
Evaluator
accept $\xi^{n+1,j}$ as ξ^{n+1}, update flow field, compute $\dot{\xi}^{n+1}$.
End

Figure 4. Schematic of predicator corrector solution procedure in one time step

The CFD solution in the Evaluator step was orders of magnitude more numerically intensive than other calculations in the present in-house code. Hence, the in-house OpenMP parallelized code was re-programmed with CUDA technology to exploit the capability of modern heterogeneous clusters.

Parallelization of the CFD computations was straight forward due to semi-implicit nature of the projection method given by Equation (2a) and (2b), and the BiCGSTAB method for the pressure-Poisson equation (3). However, the SVD algorithm executed on each meshfree node required heavy numerical workload on individual one and was not yet re-programmed into reliable and efficient GPU code. However, the SVD algorithm executed on each meshfree node generated significant numerical workload that has still not yet been re-programmed into reliable and efficient GPU code. Hence, the SVD calculation of the CFD solver was performed on CPUs only in present simulations and was one aspect of the computation that could be further improved upon.

3. Results and Discussion

3.1. Benchmarking Cases

The parallel code was performed on the heterogeneous cluster of the National Supercomputing Centre of Singapore with the NVIDIA Tesla K40 accelerator installed. Additionally, an Intel Xeon E5 Workstation with the NVIDIA Tesla K20c accelerator was used for debugging and benchmarking purposes. Benchmarking computations were conducted on the Xeon E5 Workstation using three different mesh systems shown in tab. 1 (Mesh size: Set 1<Set 2<Set 3). From fig. 5a-c, it can be seen that the parallelization and the GPU acceleration greatly speedup the nodal search process and fractional step iteration in above CFD scheme. The GPU speedup increased with the size of the Cartesian mesh, because time used for data transfer between CPU and GPU became less important with increasing computation time. The drop of speedup in node search may be explained by the increasing overhead of memory operand in larger array. The time for the projection method calculation with GPU acceleration is about half of computation using only 12 threads of CPU parallelization. Considering the wall-time elapsed during one complete FSI iteration, the advantage of GPU acceleration is more impressive when the grid size is large (fig. 5d), due to the computational bottleneck caused by the SVD procedure executing on CPUs. In general, the current GPU-accelerated solver allowed us to complete the simulation of about 100 wingbeats of the free flying insect in forward flight within a week time. This significant reduction of computational time made possible by the using of GPUs allowed us to obtain the long-term quasi-steady state of the model insect in free forward flight.

Table 1. Details of benchmarking mesh sets

	Set 1	Set 2	Set 3
Cartesian grid	$125 \times 125 \times 245$	$161 \times 161 \times 161$	$221 \times 221 \times 221$
Meshfree nodes	40039	31863	31863

We adopted the experimental results in [10] to validate the CFD scheme presented in this paper. In the experiments of Muijres *et al.* [10], the forces on the insect wings were estimated from a scaled robot wing and normalized to the fly scale using $F^* = F/mg$ with a body mass of $m = 1.8$ mg. Compared to their unfiltered experimental data, our numerical results closely tracked the build-up and decrease of forces, and correctly captured major force peaks and troughs in the whole wingbeat (see fig. 6). The mean lift obtained in the present CFD simulation was 13.5% higher than the experimental results, while a 9.7% surplus was obtained on the mean drag. This relative error between experimental and numerical results agreed with the previous numerical studies [9, 11, 17], and it may caused by the oscillation/flutter of the robotic wing due to its imperfect rigidity and slips within the actuator mechanisms. This agreement indicates

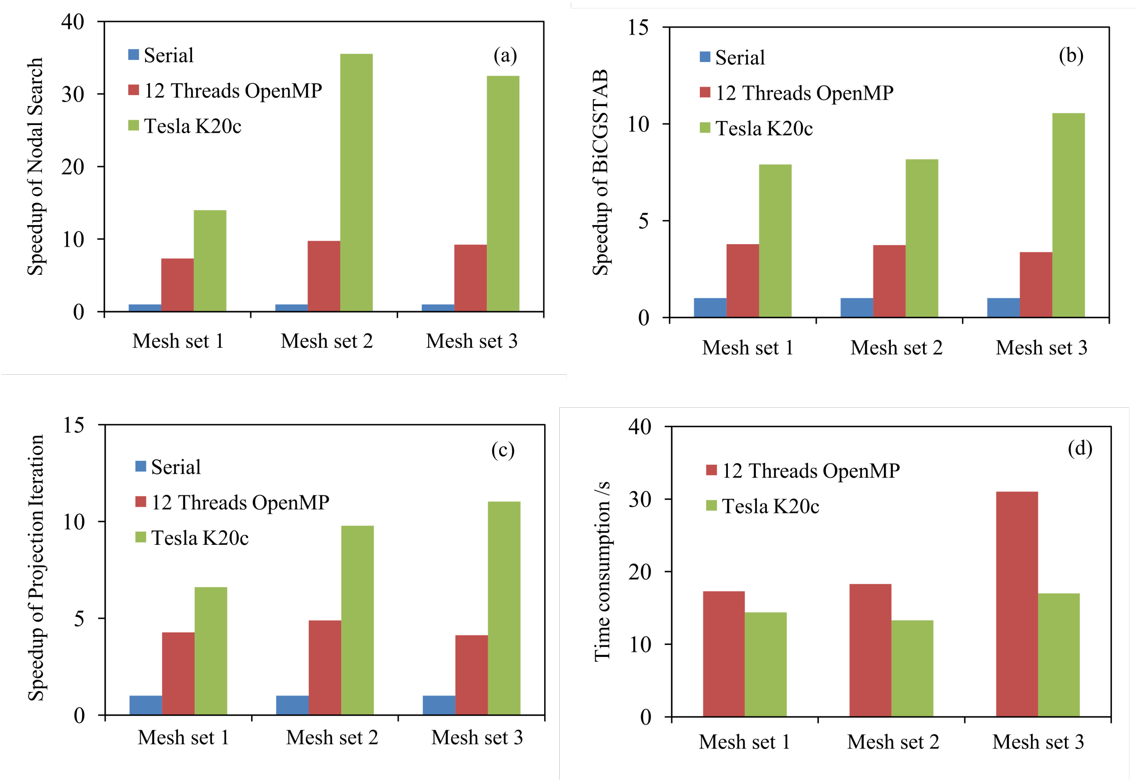
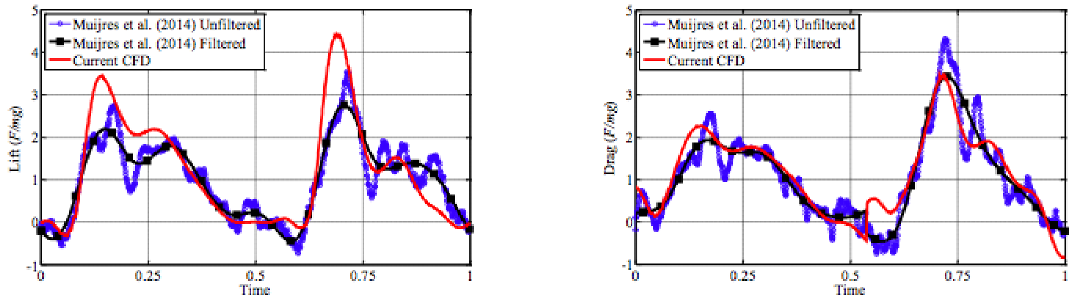


Figure 5. Performance of parallel code

that the present GPU-accelerated CFD solver can predict the force generation of insect wings with sufficient accuracy for our purpose.



(a) Lift force.

(b) Drag force.

Figure 6. Comparison of computational forces of a fruit fly wing executing natural wing motion at Re=115 with experimental results from [10]

3.2. Flight Simulations

The present numerical method resolves the full details of temporal dynamics of the flow field, and provides us a convenient way to visualize and quantify the 3D flow field produced by the flapping wings.

In hovering flight, the wings shed a copious amount of vorticity into the surrounding air. These took the forms of a leading-edge vortex (LEV), a wing-tip vortex (WTV) and a trailing-edge vortex (TEV) - identified by the regions on the wing where they are generated as shown fig. 7a. The vortices connected with each other to form a vortex ring (VR) on the wings

. The present results show that there existed a strong spanwise flow on the flapping wing in hovering flight. As suggested by previous studies [5, 7], the spanwise flow that may drain energy from the LEV, and thus limited its downstream chordwise development and prevented the wings from stalling. However, the increase of non-dimensional spanwise velocity, u^*_{span}, from the wing root to the middle wing-span also suggests that the spanwise flow may stretch the LEV along the wing and hence helped to prolong the attachment of LEV [8]. In fact, as u^*_{span} decreased in the distal section of the wing, the LEV soon detached from the wing surface. The spanwise flow may also confine the development of the TEV near the wing root which formed sheet-like vortices.

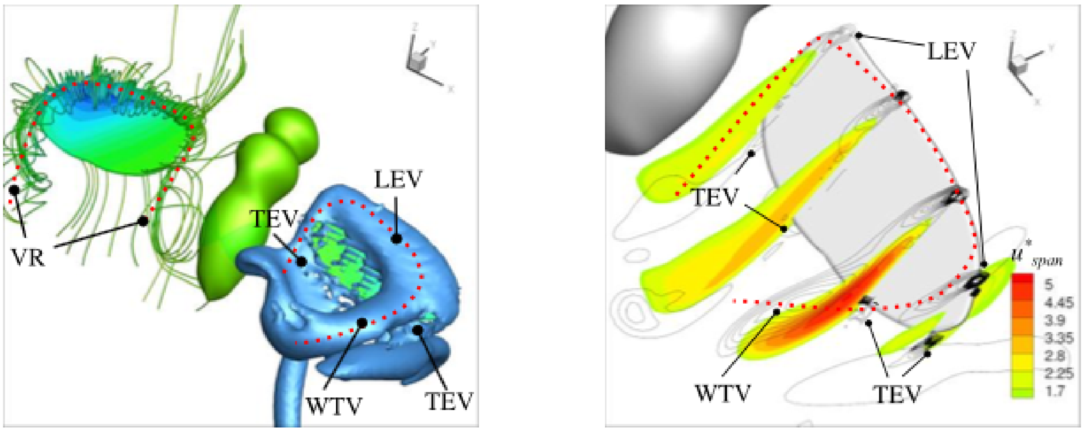
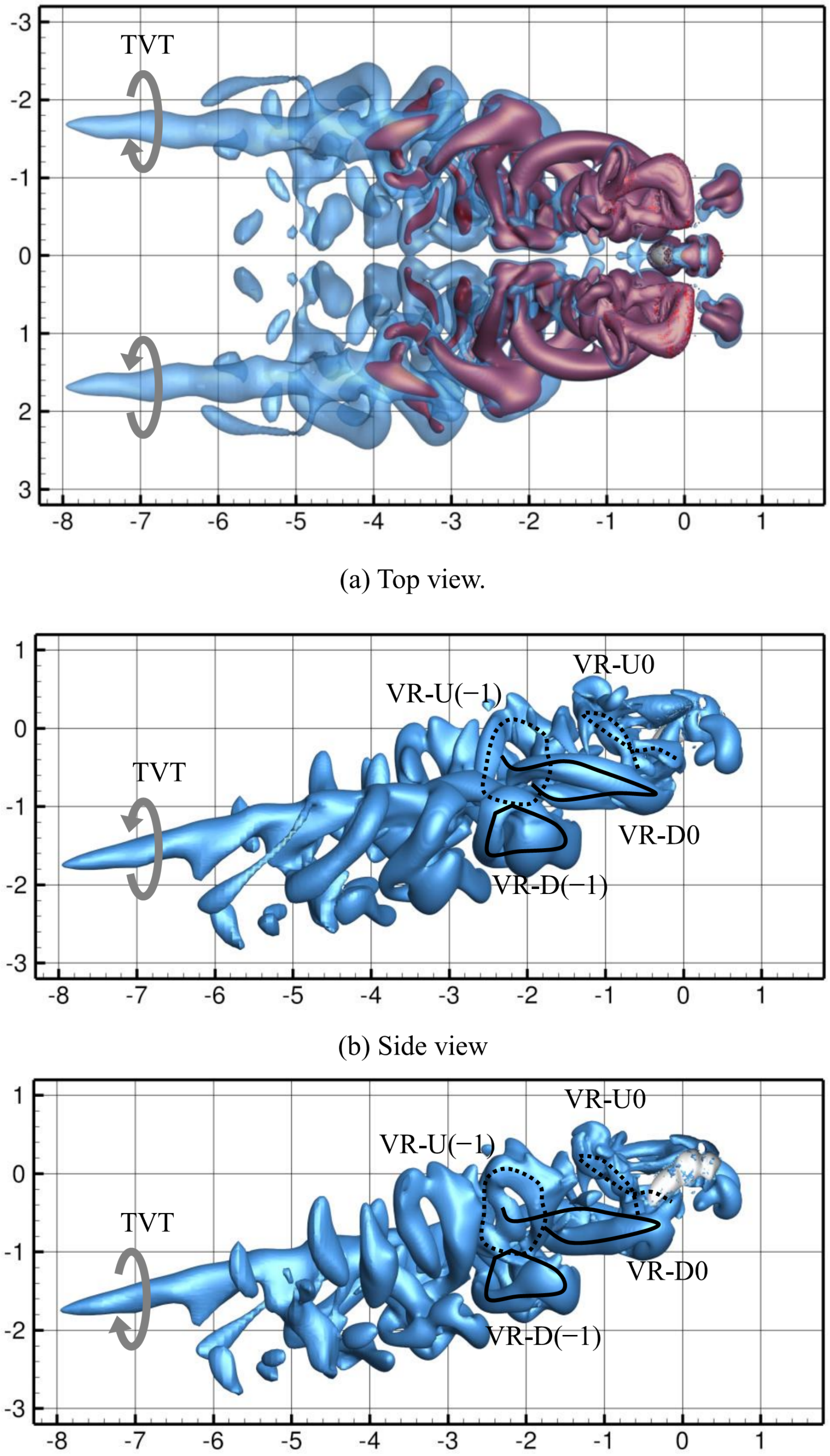


Figure 7. Vortices on the flapping wing obtained in hovering flight. Left: stream traces and iso-surface of $\lambda_2^* = -80$; Right: u^*_{span} contours and λ_2^* iso-lines

In forward flight, the wing stroke plane of the flyer is typically rotated forward to generate the requisite wing thrust, while the body is pitched forward to reduce aerodynamic drag. The quasi-steady posture of the flyer and the stroke angle are speed-dependant and ultimately governed by the balance of forces and moments acting on the whole flyer. Overall, the bulk of thrust was produced during the upstroke of a wingbeat, while the bulk of lift was generated by the downstroke. As shown in fig. 8, the vortex wake behind model fruit flyer in 60 cm/s flight - the near wake contains vortex rings shed by the up- and down-strokes, which eventually decayed to form a pair of travelling vortex tube (TVT) further downstream.

The shedding vortices become even more complex when the insect flyer is executing fast maneuvers, as the large body motion of the flyer interfered directly with the shed vortices in the wake. A simulation of fast banking flight has been carried out in this work. The insect was controlled to turn to its right side from steady hovering state in the banking flight. As shown in fig. 9, the wings of the rolling model fly collided with the stacked VRs and broke them down into disconnected vortex tubes. There were two highly stretched vortex tubes existing in the wake. During the fast rightward rolling, a strong vortex ring was created in the wake below the left wing (see fig. 9c) and the vortex moved towards the left side of the flyer (fig. 9b d). This vortex ring conveyed the lateral momentum induced by the rolling motion, and is noted as the rolling vortex. Thereafter, the downstroke WTV created on the right wing was greatly elongated to form a yawing vortex during the fast yawing phase, and remained visible in the flow field till the fifteenth wingbeat (fig. 8e f). The transient structure of the vortex wake soon decayed and evolved into the normal hovering wake after the flyer re-stabilized itself in a new hovering state.



(a) Top view.

(b) Side view

(c) Section view taken from the sagittal plane.

Figure 8. Shedding vortices (VR-Dn (—) and VR-Un ($\cdots$)) on 80 cm/s forward flight showing by λ_2^* iso-surfaces (light blue $\lambda_2^* = -0.18$, dark red $\lambda_2^* = -1.8$)

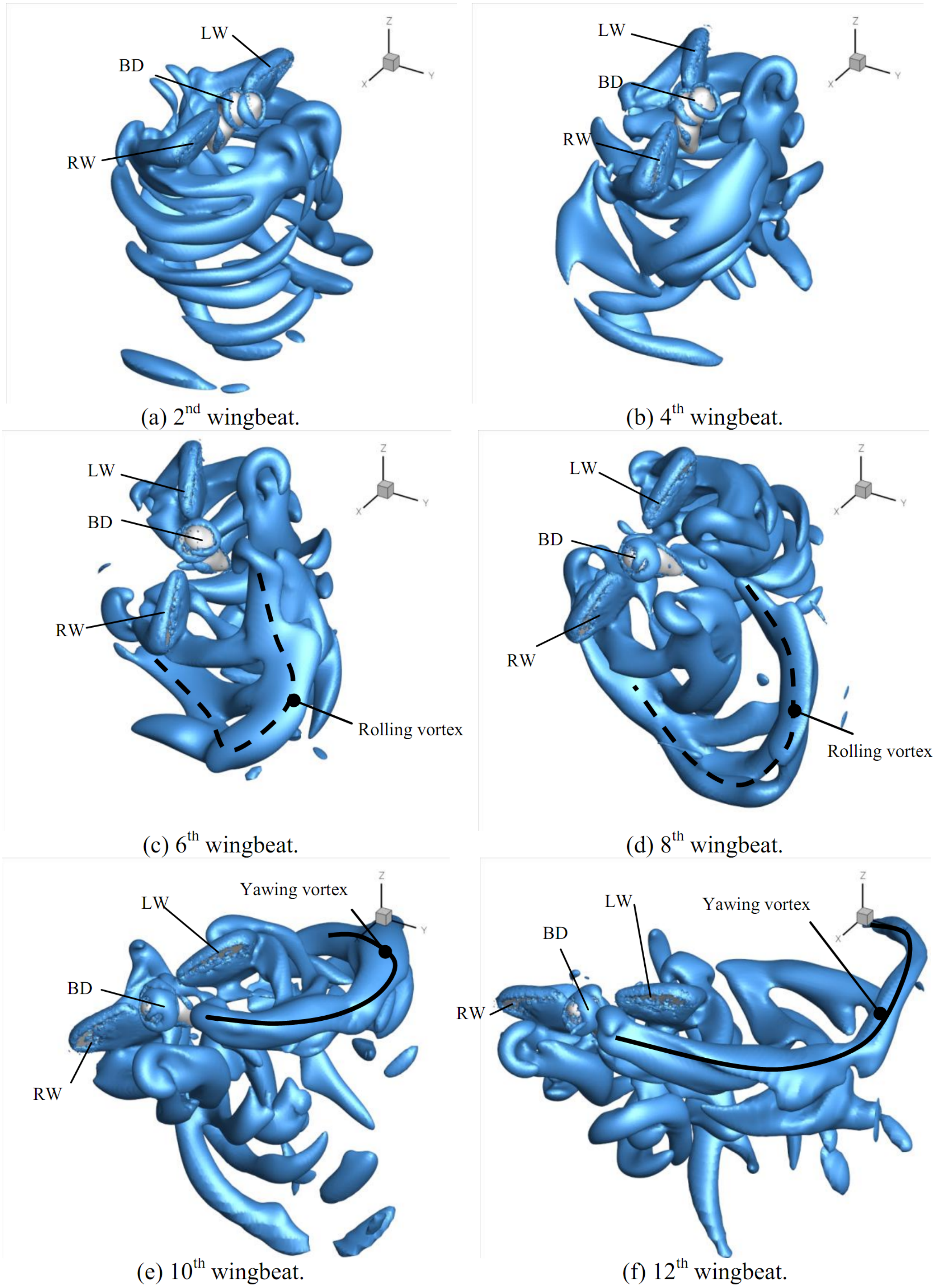


Figure 9. Development of vortex wake in banking flight (banking from steady hovering), showing by $\lambda_2^* = -1.8$ iso-surfaces obtained at the mid-downstroke. (BD: insect body; LW: left wing; RW: right wing)

Conclusions

This paper presents a numerical study on the flapping-wing free flight of a model fruit-lfy (*D. melanogaster*). The simulations were carried out with a coupled CFD/6-DoF motion solver on a 3D Cartesian-cum-meshfree grid by an SVD-GFD computational scheme. The study covered a variety of flight scenarios including hovering, forward flight and banking maneuvers. Unlike most previous insect flight studies, the model insect was allowed to move freely in all the six degrees of freedom. The computationally-intensive simulations were expedited by the application of CUDA-based GPUs, which greatly accelerated the speed of the CFD (Navier-Stokes) solver over that of the original CPU parallelized code. The gains in turnaround time were particularly significant and beneficial of highly prolonged simulations carried out to investigate the long-time quasi-steady performance of certain flight states, such as hovering and long-distance forward flights. The simulations allowed us to probe the complex aero-cum-body dynamics in flapping wing insect flight, and to study and refine control strategies and algorithms to achieve steady flight and complex aerial maneuvers.

The present computational simulations reveal the complex vortical wakes created by the wings of the model insect. The strong LEV, TEV, and WTV shed by the flapping wings dominated the highly complex near-wake of the flyer. An even more complex wake system enveloped the model insect in sharp maneuvering flights, where the wings may directly interfere with the shed flow structures. These flow structures are governed by the kinematics of the flapping wings and the motion of the flyer; and their analyses will help us to better understand the underlying physics. In level forward flight, the vortex wake in the rear of the model insect may decay into a pair of trailing vortices, similar to those frequently observed behind airplanes in flight.

The authors wish to thank the High Performance Computing Centre of National University of Singapore and National Supercomputing Centre of Singapore for providing their high performance computing facilities.

References

1. Ang, S.J., Yeo, K.S., Chew, C.S., Shu, C.: A singular-value decomposition (SVD)-based generalized finite difference (GFD) method for close-interaction moving boundary flow problems. International Journal for Numerical Methods in Engineering 76(12), 1892–1929 (2008), DOI:10.1002/nme.2398

2. Bergou, A.J., Ristroph, L., Guckenheimer, J., Cohen, I., Wang, Z.J.: Fruit Flies Modulate Passive Wing Pitching to Generate In-Flight Turns. Physical Review Letters 104(14) (Apr 2010), DOI:10.1103/physrevlett.104.148101

3. Chew, C., Yeo, K.S., Shu, C.: A generalized finite-difference (GFD) ALE scheme for incompressible flows around moving solid bodies on hybrid meshfree-cartesian grids. Journal of Computational Physics 218(2), 510–548 (2006), DOI:10.1016/j.jcp.2006.02.025

4. Gao, T., Lu, X.Y.: Insect normal hovering flight in ground effect. Physics of Fluids 20(8), 087101 (2008), DOI:10.1063/1.2958318

5. Jardin, T., Farcy, A., David, L.: Three-dimensional effects in hovering flapping flight. Journal of Fluid Mechanics 702, 102–125 (Jul 2012), DOI:10.1017/jfm.2012.163

6. Kolomenskiy, D., Maeda, M., Engels, T., Liu, H., Schneider, K., Nave, J.C.: Aerodynamic Ground Effect in Fruitfly Sized Insect Takeoff. PLOS ONE 11(3), e0152072 (Mar 2016), DOI:10.1371/journal.pone.0152072

7. Lentink, D., Dickinson, M.H.: Biofluiddynamic scaling of flapping, spinning and translating fins and wings. Journal of Experimental Biology 212(16), 2691–2704 (Aug 2009), DOI:10.1242/jeb.022251

8. Lim, T.T., Teo, C.J., Lua, K.B., Yeo, K.S.: On the prolong attachment of leading edge vortex on a flapping wing. Modern Physics Letters B 23(03), 357–360 (2009), DOI:10.1142/s0217984909018394

9. Liu, H.: Integrated modeling of insect flight: From morphology, kinematics to aerodynamics. Journal of Computational Physics 228(2), 439–459 (2009), DOI:10.1016/j.jcp.2008.09.020

10. Muijres, F.T., Elzinga, M.J., Melis, J.M., Dickinson, M.H.: Flies evade looming targets by executing rapid visually directed banked turns. Science 344(6180), 172–177 (2014), DOI:10.1126/science.1248955

11. Nguyen, T.T., Shyam Sundar, D., Yeo, K.S., Lim, T.T.: Modeling and analysis of insect-like flexible wings at low Reynolds number. Journal of Fluids and Structures 62, 294–317 (Apr 2016), DOI:10.1016/j.jfluidstructs.2016.01.012

12. Shyy, W., Aono, H., Kang, C.K., Liu, H.: An introduction to flapping wing aerodynamics. Cambridge aerospace series, Cambridge University Press, Cambridge; New York (2013), DOI:10.1017/cbo9781139583916.007

13. Shyy, W., Kang, C.K., Chirarattananon, P., Ravi, S., Liu, H.: Aerodynamics, sensing and control of insect-scale flapping-wing flight. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science 472(2186), 20150712 (2016), DOI:10.1098/rspa.2015.0712

14. Sun, M., Wang, J.K.: Flight stabilization control of a hovering model insect. Journal of Experimental Biology 210(15), 2714–2722 (Aug 2007), DOI:10.1242/jeb.004507

15. Wang, X.Y., Yeo, K.S., Chew, C.S., Khoo, B.C.: A SVD-GFD scheme for computing 3d incompressible viscous fluid flows. Computers & Fluids 37(6), 733–746 (2008), DOI:10.1016/j.compfluid.2007.07.022

16. Wang, X.Y., Yu, P., Yeo, K.S., Khoo, B.C.: SVDGFD scheme to simulate complex moving body problems in 3d space. Journal of Computational Physics 229(6), 2314–2338 (2010), DOI:10.1016/j.jcp.2009.11.037

17. Wu, D., Yeo, K.S., Lim, T.T.: A numerical study on the free hovering flight of a model insect at low reynolds number. Computers & Fluids 103, 234–261 (2014), DOI:10.1016/j.compfluid.2014.07.030

18. Yu, P., Yeo, K.S., Shyam Sundar, D., Ang, S.J.: A three-dimensional hybrid meshfree-Cartesian scheme for fluid-body interaction. International Journal for Numerical Methods in Engineering 88(4), 385–408 (Oct 2011), DOI:10.1002/nme.3182