An Application of GPU Acceleration in CFD Simulation for Insect Flight

Y. Yao, K. S. Yeo
(*Department of Mechanical Engineering, National University of Singapore*)

1 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 unparalleled mobility and maneuverability of winged insects have long captured the interest of researchers in the area of biomechanics and aerodynamics. In the past decade, several unsteady aerodynamic phenomena have been discovered to be responsible for high agility and lift enhancement of flapping insect [1], and have been visualized and analyzed using numerical approaches [2, 3]. However, due to prohibitive computational cost, long-term dynamics of insect flight was still lack of numerical investigation [4].

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

2 Numerical Methods

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:

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

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

where $\mathbf{u}$ and p are the non-dimensional velocity and pressure fields respectively of the time-varying fluid domain $\Omega(t)$, 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 Chew et al. [5] is adopted in this work to advance the above governing equations in time. The discretized form of the momentum equations (2) is written as:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{1}{2}\left\{\left[-(\mathbf{u} - \mathbf{u}^g) \cdot \nabla \mathbf{u} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}\right]^{n+1} + \left[-\nabla p - (\mathbf{u} - \mathbf{u}^g) \cdot \nabla \mathbf{u} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}\right]^{n}\right\} \tag{3}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{2} \nabla p^{n+1} \tag{4}$$

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 (4) yields the following pressure-Poisson equation, which allows us to obtain the new pressure field p^{n+1} from $\mathbf{u}^*$:

$$\nabla^2 p^{n+1} = \frac{2}{\Delta t} \nabla \cdot \mathbf{u}^* \tag{5}$$

Boundary conditions given in Ang et al. [6], Wang et al. [7] and Wu et al. [4] have been implemented here to close the above discretized governing equations. The fractional step equations and boundary conditions are iterated in simulation 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}.

Following the methodology developed by Chew et al. [5] and Wang et al. [8], 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. The meshfree nodes convect with the motion of the body and wings. Singular value decomposition (SVD) based generalized finite difference (GFD) scheme is adopted here to approximate derivatives involved in the solution with the second order accuracy. In this scheme, SVD technique is used to minimize the approximation error caused by arbitrary node distribution [6]. Figure 1 shows a set of computational mesh for a fruit fly flyer that used in this study.

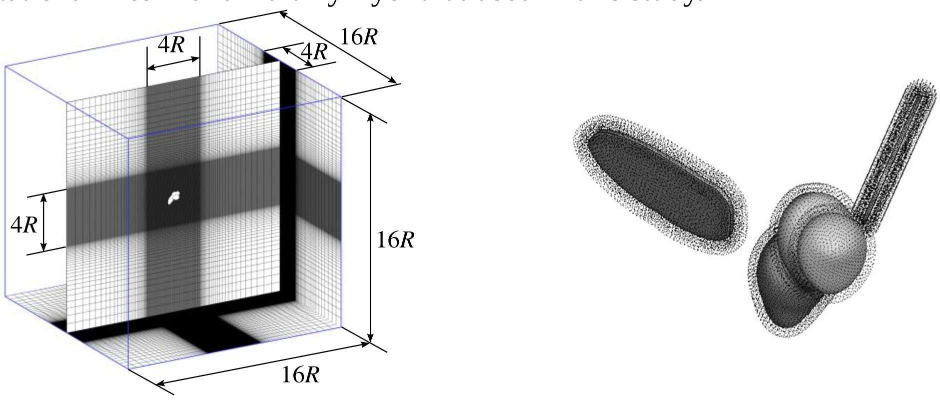


Figure. 1. Schematic view of computational domain (R is the wing length of the flyer). Left: background grid; Right: meshfree nodes around insect model.

3 Implementation

In the present study, the six-degrees of freedom (6-DoF) motion of insect flyer was simulated using a predictor-corrector method following the calculation procedures listed below:

1. Predicator step (P): at time step n+1, compute current dynamic state of insect ξ^{n+1} using previous flow field state Φ^n;

2. Evaluator step (E): update Φ^n to Φ^{n+1} using CFD solver with boundary conditions given by ξ^{n+1};

3. Corrector step (C): correct ξ^{n+1} using Φ^{n+1}.

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.

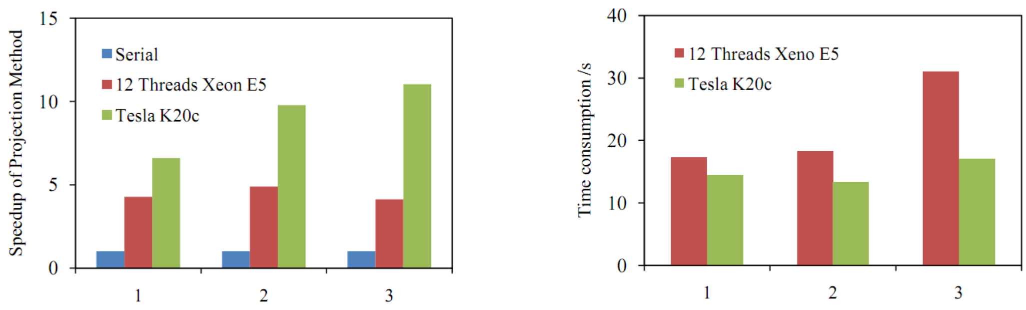
The CFD solver that used in the Evaluator step was constructed using the numerical methods described in Section 2. The CFD solution is orders of magnitude more numerically intensive than other calculations in present in-house code. Hence, the in-house OpenMP parallelized code was re-programmed with CUDA 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 (3) and (4), and the BiCGSTAB method for the pressure-Poisson equation (5). 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. Hence, the SVD calculation was performed on CPU in present simulations and was one bottleneck that limited the improvement of computational efficiency.

4 Results and Discussion

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.

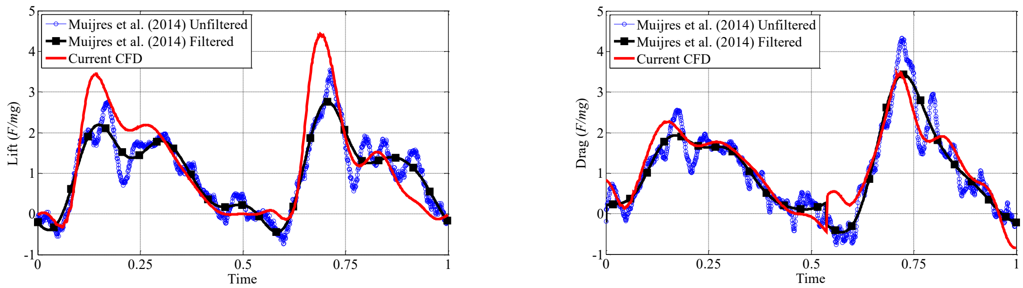
Benchmarking computations were conducted on the Xeon E5 Workstation using three different mesh systems (Mesh size: Set 3>Set 2>Set 1). From Figure 2a, it can be seen that the parallelization and the GPU acceleration greatly speedup the fractional step iteration in above CFD method. Even taking into account the SVD code executing on CPU only, the time difference is impressive, especially when the size of mesh is large (Figure 2b).



(a) Speedup of projection iterations. (b) Wall-time elapsed in one FSI iteration

Figure 2. Performance of parallel code.

We adopted the experimental results in Muijres et al. [9] to validate of the CFD methods presented in this paper. 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 downs in the whole wingbeat (see Figure 3). This agreement indicates that the present CFD code using the CUDA technology can predict the force generation of insect wings with sufficient accuracy.



(a) Lift force. (b) Drag force.

Figure 3. Comparison of computational forces of a fruit fly wing executing natural wing motion at Re=115 with experimental results from [9].

The GPU accelerated code allowed us to investigate long-term dynamics of insect flight which may cost prohibitive computational time on CPU workstations. We simulated the motion and surrounding flow field of free flying fruit fly in a series of flight scenarios, including hovering, rectilinear flight and maneuvers. The model fly can fly stably in a wide range of airspeed. The presence of oncoming flow exerted strong effect on the aerodynamics of the flapping wing flyer. The flyer can also swiftly alter its heading and flight direction using only subtle changes in wing kinematics. The vortical flow surrounding the model fly was systematically analyzed in all the cases (Figure 4). The present results showed good consistency with previous studies.

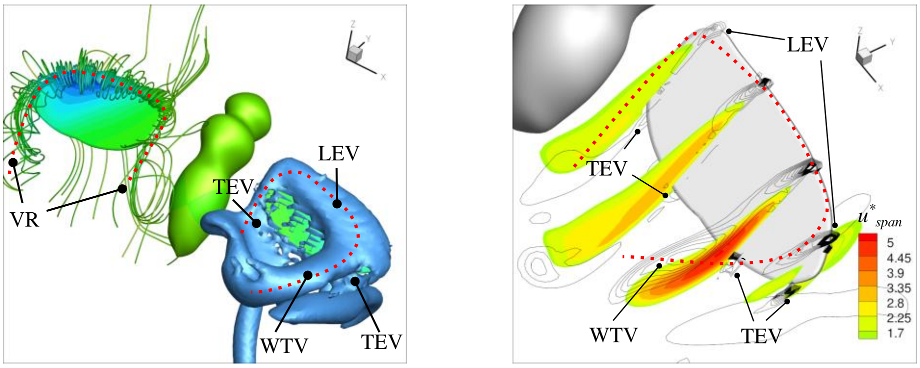


Figure 4. Vortices on the flapping wing obtained in hovering flight.
Left: stream traces and iso-surface of λ_2=−80; Right: u^*_{span} contours and λ_2 iso-lines.

5 Conclusions

This paper presents a numerical study on the free flight of a model insect. Adopting CUDA technology, the speed of the CFD solver was significantly accelerated as compared to the original CPU code. The results of simulation showed good agreement with the experimental data. This allows us to perform prolonged insect flight simulation with sufficient accuracy and acceptable computational time cost using the accelerated CFD solver. The results of prolonged simulation could further give insight into the long-term behavior of free flying insect.

References

1. Shyy, W., Aono, H., Kang, C. et al. An Introduction to Flapping Wing Aerodynamics, Cambridge University Press, New York, 2013.
2. Liu, H. Integrated modeling of insect flight: From morphology, kinematics to aerodynamics // J. Comput. Phys. 2009. Vol. 228. P. 439–459.
3. Shyy, W., Kang, C., Chirarattananon, P. et al. Aerodynamics, sensing and control of insect-scale flapping-wing flight // Proc. Roy. Soc. Ser. A. 2016. Vol. 472.
4. Wu, D., Yeo, K.S., Lim, T.T. A numerical study on the free hovering flight of a model insect at low Reynolds number // Comput. Fluids. 2014. Vol. 103. P. 234–261.
5. Chew, C.S., Yeo, K.S., Shu, C. A generalized finite-difference (GFD) ALE scheme for incompressible flows around moving solid bodies on hybrid meshfree–Cartesian grids // J. Comput. Phys. 2006. Vol. 218. P. 510–548.
6. Ang, S.J., Yeo, K.S., Chew, C.S. et al. A singular-value decomposition (SVD)-based generalized finite difference (GFD) method for close-interaction moving boundary flow problems // Int. J. Numer. Meth. Engng. 2008. Vol. 76. P. 1892–1929.
7. Wang, X.Y., Yeo, K.S., Chew, C.S. et al. A SVD-GFD scheme for computing 3D incompressible viscous fluid flows // Comput. Fluids. 2008. Vol. 37. P. 733–746.
8. Wang, X.Y., Yu, P., Yeo, K.S. et al. SVD–GFD scheme to simulate complex moving body problems in 3D space // J. Comput. Phys. 2010. Vol. 229. P. 2314–2338.
9. Muijres, F.T., Elzinga, M.J., Melis, J.M. et al. Flies evade looming targets by executing rapid visually directed banked turns // Sci. 2014. Vol. 344. P. 172–177.