Using AmgX to accelerate a PETSc-based Immersed Boundary Method code

Olivier Mesnard[a], Pi-Yueh Chuang[a], Lorena A. Barba[a,*]

[a] *Mechanical and Aerospace Engineering, The George Washington University, Washington, DC, 20052, United-States*

Abstract

Our open-source code `PetIBM`—an immersed boundary method with a fully discrete projection formulation—was written to take advantage of the `PETSc` library for solving the Poisson system. We have now added the capacity to accelerate the time to solution on CUDA-capable GPU devices using the Nvidia library `AmgX`. To provide access to `AmgX` solvers from a `PETSc`-based code, we developed a wrapper code that converts the data structures between the two libraries. This wrapper code could be useful to other `PETSc` applications that want to use GPUs via `AmgX`. Our application of interest is the three-dimensional flow around a flying-snake, to reveal the lift-enhancement mechanisms used by this unconventional glider. We are developing capability to study this problem in Microsoft Azure cloud services.

Keywords: Immersed Boundary Method, `PETSc`, `AmgX`, flying snake

1. PetIBM and AmgXWrapper

We have developed `PetIBM`[1] that implements using the `PETSc`[2] library, an immersed-boundary method (IBM) [**?**] in which the fully discrete algebraic system is solved via a projection method based on an approximate block-LU decomposition. The data structures and routines provided by the `PETSc` li-

*Corresponding author

Email address: `labarba@gwu.edu` (Lorena A. Barba)

[1] `PetIBM`: https://github.com/barbagroup/PetIBM

[2] `PETSc`: https://www.mcs.anl.gov/petsc

brary allowed us to rapidly develop a software that runs on distributed-memory architectures.

As expected with the projection method, the iterative Poisson solver is the bottleneck in our simulations. This is even worse for the IBM we use where the modified Poisson operator becomes larger and possesses more off-diagonal entries. To overcome this challenge, we use the Nvidia-library `AmgX`[3] to solve the iterative system on multiple CUDA-capable GPU devices. We have developed an `AmgX` wrapper[4] that provides the interface with the PETSc library and incorporated it in `PetIBM`. Past two-dimensional simulations with `PetIBM` showed a 21 times application speed-up in runtime from using our `AmgX` wrapper on one GPU node compared to using `PETsc` on a CPU node (see Figure 1).

The full codes, `PetIBM` and `AmgXWrapper`, are open-source, released under MIT license, and version-controlled on GitHub.

2. Flying snakes to the cloud

We aim to study the aerodynamics of the *Chrysopelea paradisi*, a species of snake with the amazing capability to glide through the air. Previous experimental work [?] and two-dimensional simulations [?] reported enhanced lift force on a snake gliding at a particular angle of attack of 35^o. Using `PetIBM` and `AmgXWrapper`, we now intend to understand the three-dimensional wake structures responsible for high gliding performances of the paradise tree snake.

Finally, we decided to use the public cloud Microsoft Azure to run our simulations so that we could compare the performances with our University HPC cluster.

References

[3] `AmgX`: https://developer.nvidia.com/amgx

[4] `AmgXWrapper`: https://github.com/barbagroup/AmgXWrapper

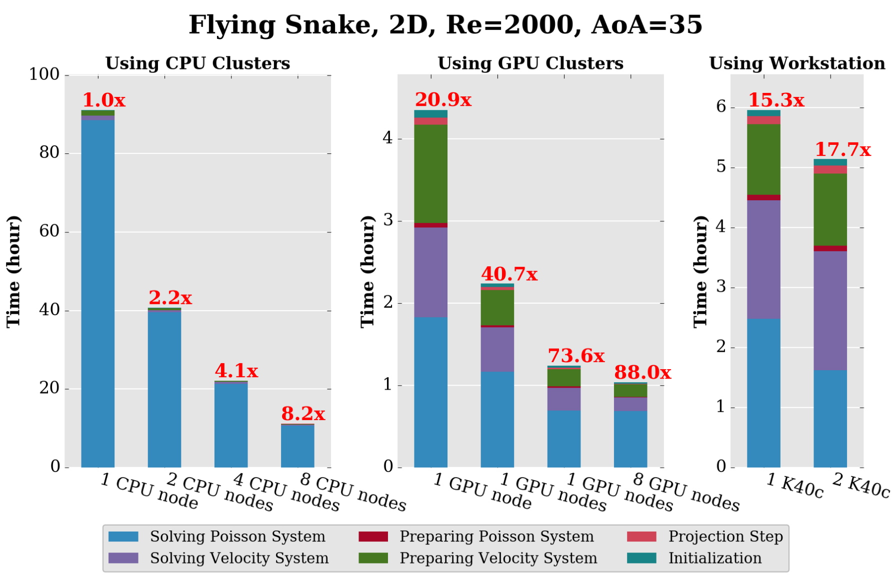

Figure 1: Runtimes for the flying-snake case, using `PetIBM` and `AmgX`.