AIAA Aviation Forum and ASCEND co-located Conference Proceedings
29 July - 2 August 2024, Las Vegas, Nevada
AIAA AVIATION FORUM AND ASCEND 2024

10.2514/6.2024-4018

Solving Unsteady Potential Flow Problems in $O(n)$ Time

Ryan Anderson* and Andrew Ning†
Brigham Young University, Provo, UT, 84602

Design of vertical takeoff and landing aircraft is challenging in part due to significant aerodynamic interactions between rotors and wings. Computational models can aid in their design, but are computationally expensive. 3-D panel methods coupled with vortex particle wakes offer an attractive solution, but solving for the panel strengths scales poorly for large problems. Multigrid methods, such as Krylov subspace methods in conjunction with the fast multipole method (FMM), have been demonstrated to reduce the scaling to $O(N)$. We explore the performance and limitations of the Krylov-FMM method, traditional matrix-powered GMRES, LU decomposition, and a novel $O(N)$ multigrid approach to solving boundary element problems by combining FMM with Gauss-Seidel iterations. We also explore the effect of warm-starting these methods in unsteady aerodynamic simulation, leading to an order-of-magnitude decrease in cost.

Nomenclature

LU	=	lower-upper
FMM	=	Fast Multipole Method
P	=	FMM expansion order
θ	=	FMM multipole expansion threshold
l	=	FMM maximum leaf size
$\mathcal{F}$	=	matrix-free approximation of A generated using the FMM
GMRES	=	Generalized Minimum RESidual solver
MF-GMRES	=	Matrix-Free Generalized Minimum RESidual solver
FGS	=	Fast Gauss-Seidel solver
ϕ	=	velocity scalar potential
σ	=	panel strength
$\vec{v}$	=	velocity induced by a panel
$\hat{n}$	=	unit normal vector at the centroid of a panel
St	=	Strouhal number

I. Introduction

The design of electric vertical takeoff and landing (eVTOL) aircraft is challenging, in part due to the significant aerodynamic interactions that occur [1, 2] as well as a difficult transitional flight regime between hover and forward flight [3, 4] and in ground effect [5]. Computational models can aid the design process by providing performance estimates; however, designers must navigate the tradeoff between computational cost and physical accuracy. Traditional computational fluid dynamics can capture many of the relevant physics, but require significant computational cost. For example, Healy et al. demonstrated detached eddy simulation of two interacting rotors with 6 days of wall time on 512 processors [6].

Many efforts have been made to capture the relevant physics at reduced cost. The vortex particle method (VPM), for example, is a Lagrangian discretization of the Navier-Stokes equations. It requires no meshing of the fluid domain and has been shown to predict the unsteady wake behavior of rotorcraft [7], though long simulations are notoriously numerically unstable. Alvarez reformulated the VPM governing equations to solve the numerical stability problem, and found he could predict rotor performance at two orders of magnitude reduced cost when compared to mesh-based CFD of similar fidelity [8].

*Doctoral Candidate, Mechanical Engineering, AIAA Student Member
†Associate Professor, Mechanical Engineering, AIAA Associate Fellow

The cost of using VPM can be divided into two problems. The first problem is calculating the induced velocity and velocity gradient induced by all particles on all particles. This reduces to an n-body problem, which can be solved in $O(n)$ using the fast multipole method (FMM) [9]. The second problem lies in modeling the interaction of vortex particles with solid surfaces. The method of images has been used with VPM [10] at the cost of doubling the number of particles, but is unfortunately limited to simple geometries like flat ground or a sphere. Potential flow-based methods are popular for enforcing solid surface boundary conditions with VPM, with vortex lattice [7, 10], source panels [11], source-plus-dipole panels [12], and vortex sheets [13] all successfully demonstrated. However, these methods require the solution of a linear system to obtain the panel strengths, which naively scales as $O(n^3)$, and becomes the bottleneck cost for large problems [11]. This limits the size and fidelity of what can be modeled with VPM.

Large boundary element problems are not new, and many advances have been made to solve them in a scalable way. The problem reduces to the linear solve $Ax = b$. Solved naively, this results in $O(n^2)$ effort just to build and store matrix A. Multigrid methods rediscretize the problem at coarser levels, solving each level and interpolating to the finer level in an impressive $O(N)$ scheme at their best [14–17]. Matrix-free multigrid techniques have proven effective, with the FMM used to approximate the matrix-vector product. This has been used with a Galerkin solution approach for computational magnetics and electrostatics [18], as well as with matrix-free GMRES for several applications, including evaluating the acoustic field of a vibrating body [19], mesh-based large eddy simulation [20], and Stokes flow around red blood cells [21]. Various FMM-accelerated approaches have been used to solve electromagnetic scattering problems [22]. Jiang et al. used a hybrid Gauss-Seidel GMRES approach to solve for the electromagnetic scattering of a conducting body. They found it advantageous to break the boundary elements into several blocks, which they solved iteratively using Gauss-Seidel iterations. Each block was solved using GMRES. [23] Eibert solved the scattering problem using a "flexible" GMRES, capable of changing its preconditioner each iteration at the cost of double the memory. This allowed him to chain matrix-free GMRES solvers in a recursive fashion, with each inner solver set to lower accuracy (and lower cost) than its parent. He performed a Gauss-Seidel smoothing operation after each matrix-vector product to further improve convergence. [24]

In this work, we compare the performance four different approaches to solving for source panel strengths, with the motivation to develop an $O(N)$ solver that is suitable for unsteady aerodynamic simulation. A unique feature of this class of problems is that the solutions of two adjacent timesteps are very similar, leading to highly accurate initial guesses. We compare the performance of four solvers, including LU factorization, matrix-powered GMRES, matrix-free GMRES, and a novel multigrid approach coupling FMM with Gauss-Seidel iterations. Because we are interested in unsteady aerodynamic simulation, we explore the effect of warm-starting each solver, including a method of leveraging Taylor series extrapolation to improve the initial guess.

II. Methods

The $O(n)$ scaling achieved in this work is due to the FMM. We begin by briefly describing the FMM algorithm, and then each of the solvers we will consider. We discuss the theoretical implications of warm-starting each solver, and finally detail the test case we will use.

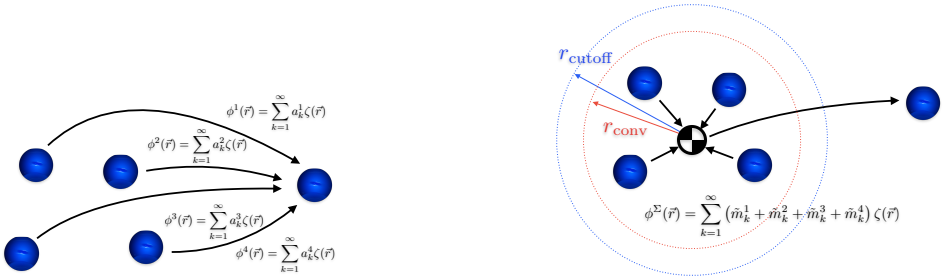
A. Fast Multipole Method

The Fast Multipole Method was developed in the 80's by Greengard and Rokhlin [9], and is capable of solving the N body problem in $O(N)$ time. It functions in 6 stages, which we will briefly outline here:

1) Multipole creation: Multipole expansions are generated to represent all bodies. This is analogous to defining a series expansion to represent the influence of each body.
2) Upward pass: Multipole expansions are translated and aggregated to the center of body clusters. Note that once two expansions have been translated to the same center, their coefficients may be simply added by superposition. These aggregated multipole expansions are further aggregated into larger clusters, recursively upward. Note that expansions of the highest level clusters represent the influence of many more bodies than the lower level clusters, leading to higher compression of the body influence; however, this comes at the cost of a higher convergence radius, meaning it can only be evaluated farther away for a fixed error tolerance. The aggregation of multipole expansions is illustrated in Fig. 1.
3) Horizontal pass: Multipole expansions at various levels are transformed into local expansions at clusters outside their convergence radius. Local expansions have the opposite convergence behavior of multipole expansions, meaning they only converge *inside* their convergence radius. The advantage of using local expansions is that the influence of many multipole expansions can be combined into a single local expansion before being evaluated,

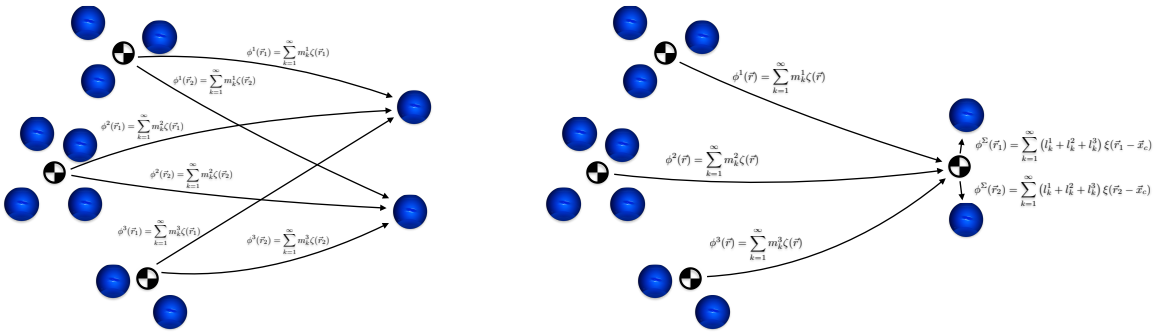
thus reducing the computational cost. This is illustrated in Fig. 2.

4) Downward pass: Local expansions are translated into all of their respective subclusters (already created during the second step). This results in the aggregation of all local expansions at the lowest level.
5) Evaluation: Local expansions are evaluated at all target bodies within their respective cluster.
6) Nearfield: Interactions between close bodies that couldn't be captured using multipole expansions are evaluated directly.



(a) **(b)**

Fig. 1 In Fig. 1a, the potential induced by a cluster of bodies is expressed as a sum of series expansions. In Fig. 1b, we obtain a multipole expansion representing the influence of the entire cluster by recentering each series expansion about a common center and summing them. However, the multipole expansion only converges outside a finite radius of convergence. It is common to define a cutoff radius, greater than, to provide some control over the error.



(a) **(b)**

Fig. 2 In Fig. 2a, three multipole expansions, too far away to be combined into a single multipole expansion, can be evaluated one at a time at a cluster of targets. However, Fig. 2b shows a smarter way to do this: many multipole expansions may be transformed into local expansions at a target cluster and summed, resulting in a single local expansion and reducing computational cost.

For the present work, we use 'FastMultipole.jl', an in-house FMM code developed in the Julia language to be made open-source soon*. Clustering is performed using an adaptive octree algorithm.

Finally, we note that three tuning parameters must be chosen to control the error and performance of the FMM. The first is the expansion order P, which determines how many terms to include in the expansions. Increasing P reduces the error, but increases the cost. Next is the the multipole threshold θ, which is used to determine the cutoff radius shown in Fig. 1b as

*https://github.com/byuflowlab

$$r_{\text{cutoff}} = r_{\text{conv}}/\theta \tag{1}$$

The multipole threshold is bounded by $0 < \theta < 1$ and controls how many multipole expansions. A higher value of θ typically results in lower computational cost but higher error. Finally, the leaf size l is the maximum number of bodies allowed in leaf-level clusters when forming the octree. The leaf size has little control over error, but has a significant effect on computational cost. It determines how many interactions are computed using expansions vs. direct calculations. Some benchmarking is used to determine these parameters in Sec III.

B. LU Decomposition

The first solution approach we consider is LU decomposition. This entails factorizing A into $A = LU$ where L is a lower-triangular matrix and U is upper-triangular. Then, the solution becomes a two-step process:

$$Ax = b \tag{2}$$
$$LUx = b \tag{3}$$
$$Ux = L^{-1}b \tag{4}$$
$$x = U^{-1}L^{-1}b \tag{5}$$

where L^{-1} and U^{-1} need not be obtained explicitly; rather, each step is solved using elimination. Note that the LU factorization itself costs $O(N^3)$, while the solution of x costs $O(N^2)$.

C. GMRES

The generalized minimum residual (GMRES) method for solving $Ax = b$ is a Krylov subspace method that uses Arnoldi factorization to form a Krylov subspace of dimension i at iteration i. For N unknowns, GMRES has guaranteed convergence after N iterations, as the Krylov subspace has grown to fill the entire linear space at that point. However, fewer iterations may be performed to reach an acceptable error tolerance for lower cost. The most expensive part is the matrix-vector product Ax, making the cost of each iteration $O(N^2)$ if the matrix A is formed explicitly. As the FMM can be used to approximate the matrix-vector product (see Sec. II.D.1), this can be done without explicitly forming the matrix A, reducing the time cost of each iteration to $O(N)$. In the present work, we use the Julia package Krylov.jl [25]. When A is formed explicitly, we will call this method GMRES. When the FMM is used in place of Ax, we will call this method matrix-free GMRES (MF-GMRES).

We remark that state-of-the-art methods employ preconditioners to accelerate convergence, and to avoid the convergence deterioration that occurs with greater problem discretization of GMRES [17]. Preconditioners are not considered in this work.

D. Fast Gauss-Seidel

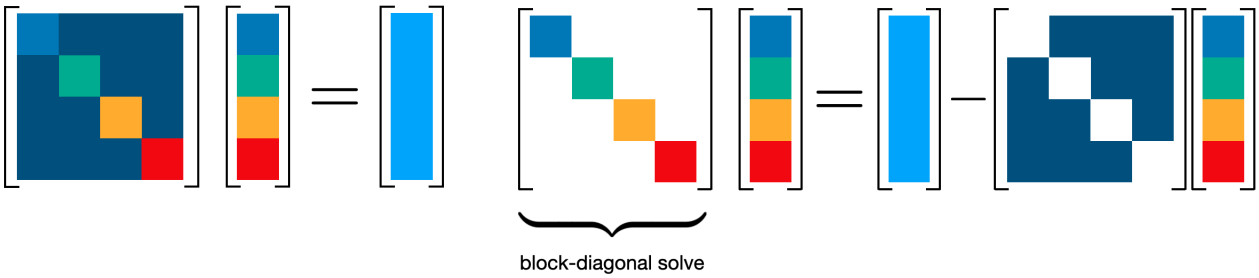
We now present a new way of solving a linear system,

$$Ax = b \tag{6}$$

which we call Fast Gauss Seidel (FGS) iterations. Before detailing the FGS algorithm, let us review block Gauss-Seidel iterations. In Fig 3a, we see how a matrix equation can be converted to a block-diagonal solve by subtracting the off-diagonal part to the right hand side. In Jacobi iterations, the entire right hand side is evaluated at once, and the resulting simplified equation is solved for the next iteration of unknowns. In Gauss-Seidel iterations, the right hand side is evaluated incrementally, using the most recently available value for each unknown as it is solved. Gauss-Seidel iterations tend to converge faster, as they provide a "coupling" between blocks.

1. Matrix Splitting with FMM

For a general boundary element problem, we can use the FMM to simulate the matrix-vector product Ax as follows. First, we note that each entry of A_{ij} represents the effect of body j on the boundary condition at body i. Thus, the product Ax is the combined influence of all bodies on the boundary condition at all bodies. We construct an operator



(a) (b)

Fig. 3 Fig. 3a is a visual representation of the linear system $Ax = b$ we desire to solve. Note that the colored blocks represent clusters of bodies. In Jacobi and Gauss-Seidel iterations, we subtract the farfield influence to the right-hand side, reducing the problem to a block diagonal solve as shown in Fig. 3b

$\mathcal{F}(x)$ to perform the following. First, approximate Ax by evaluating the net influence of all bodies of strength x on all bodies with an FMM call. Second, evaluate the resulting boundary condition on all bodies. Then, the linear system (Eq. 6) is approximated by

$$\mathcal{F}(x) = b \tag{7}$$

This approximation is depicted in Fig. 4. Note that $\mathcal{F}$ consists of 3 kinds of interactions:

1) farfield influence computed using multipole expansions, depicted by the dark blue background
2) nearfield influence of each leaf-level cluster on itself, depicted by the large single-colored blocks
3) nearfield influence of leaf-level clusters on each other, depicted by the small multi-colored blocks

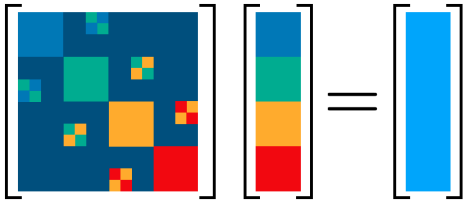


Fig. 4 Representation of Eq. 7. Colored blocks represent leaf-level clusters used in the FMM, multicolored blocks represent nearfield interactions, and the dark blue background represents farfield influence approximated by multipole expansions.

We can perform matrix splitting on Eq. 7, subtracting the influence of multipole expansions as well as nearfield leaf-level clusters on other leaf level clusters, as shown in Fig. 5. This only leaves the influence of each leaf-level cluster on itself on the left hand side. This is a block-diagonal solve, which is very inexpensive to evaluate. Note that we evaluate the farfield influence all at once at the beginning of each FGS iteration, but we evaluate the nearfield influences as they are needed using the recently solved strengths, reminiscent of Gauss-Seidel iterations.

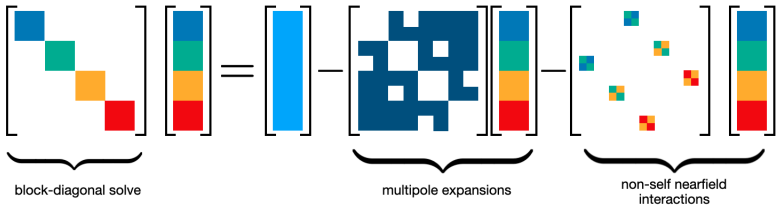


Fig. 5 In FGS iterations, we subtract the farfield and off-diagonal nearfield influences to the right-hand side, reducing the problem to a block-diagonal solve.

Note that the block-diagonals can be solved with any approach. Because they are typically small, and the operator $\mathcal{F}$ often does not change, we use an LU decomposition of each block, which we compute and store at the beginning of

each simulation to reduce cost. However, should large leaf sizes be desired, as could be the case if GPU is used, it may be preferable to employ another matrix-free method such as GMRES to solve the blocks.

2. Termination Criterion

As our termination criterion, we calculate the L^2 norm of the residual $\varepsilon = ||\mathcal{F}(x) - b||_2$. As this would require an additional FMM call and effectively double the cost of each iteration, we only evaluate the termination criterion if the L^∞ norm of change in boundary element strengths $||x_i - x_{i-1}||_\infty$ is less than the tolerance.

3. $O(N)$ Cost

The overall cost of a single FGS iteration C_{iter} then becomes

$$C_{\text{iter}} = C_{\text{block}} N_{\text{leaves}} + C_{\text{farfield}} + C_{\text{nearfield}} \tag{8}$$

where C_{block} is the cost of solving a single block, N_{leaves} is the number of leaf-level clusters, C_{farfield} is the cost of computing the multipole influence, and $C_{\text{nearfield}}$ is the cost of evaluating all nearfield interactions except that of each leaf-level cluster on itself. It is already known that $C_{\text{farfield}} + C_{\text{nearfield}} \leq O(N)$ [9]. Then assuming we require $N_{\text{iter}} \ll N$ iterations, and that there are a maximum of l bodies in each leaf-level cluster, and that $l \ll N$, this results in an overall cost of

$$\begin{aligned} C_{\text{FGS}} &= N_{\text{iter}} \left(C_{\text{block}} N_{\text{leaves}} + C_{\text{farfield}} + C_{\text{nearfield}} \right) && (9) \\ &= N_{\text{iter}} \left(N_O(l^2) O(N) + O(N) \right) && (10) \\ &= O(N) && (11) \end{aligned}$$

4. Slow Convergence and Robustness

Finally, we note that the rate of Gauss-Seidel iterations can be slow. A good preconditioner and a good initial guess can aid this weakness, and although we do not consider preconditioners here, warm-starting with a good initial guess is considered in the next section. Further, we employ a relaxation factor of 1.4 as used in successive over-relaxation (SOR) to speed up convergence.

The robustness of this approach may suffer for ill-conditioned problems, such as when panels of different surfaces end up very close to each other. This would benefit from a preconditioner, which we do not consider in this work.

E. Warm Start

In unsteady aerodynamic simulation, boundary element strengths at each timestep will be very similar to the timestep preceding. We explore the improved performance of the MF-GMRES and FGS solvers in the presence of a good initial guess, obtained by recycling the solution at the previous timestep. A better initial guess can be obtained by Taylor series approximation of the panel strengths of previous timesteps. We also benchmark the performance of an initial guess obtained using third-order backward finite difference approximation of the first derivative and second-order backward finite difference second derivatives.

F. Test Case

To test each solver, we use a source panel discretization of an untapered, untwisted wing of 1 m span with NACA 0012 airfoil.

A source panel of strength σ induces a potential at target point $\vec{x}_t$ equal to

$$\phi(\vec{x}_t) = \iint_{\text{panel}} \frac{\sigma}{4\pi |\vec{x}_t - \vec{x}|} dS \tag{12}$$

with its resulting velocity found by taking the gradient as

$$\vec{v} = -\nabla \phi \tag{13}$$

For simulating inviscid flow, the boundary condition at the centroid of each panel $\vec{x}_i$ with normal vector $\hat{n}_i$ is

$$\vec{v}(\vec{x}_i) \cdot \hat{n}_i = 0 \tag{14}$$

We obtain the multipole expansion coefficients for each panel using the $O(1)$ recursive method developed by Gumerov [26].

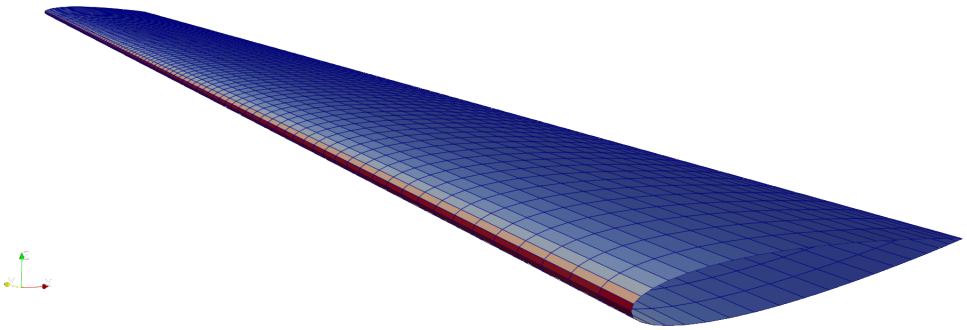


Fig. 6 Source panel wing used to test the performance of each solver.

To evaluate scaling, we benchmark each solver over a range of problem sizes by rediscretizing the same wing geometry with different numbers of panels and applying a freestream of 1.048 m/s at 10° angle of attack.

To evaluate the performance of unsteady problems with a good initial guess, we simulate sinusoidal heaving of the wing. We apply an x-component (streamwise) freestream of 1 m/s as well as a sinusoidal z-component (vertical) related to a desired Strouhal number ($St = \frac{fL}{U}$), where f is the frequency of oscillation, L is the amplitude of oscillation, and U is the ambient freestream velocity. We fix our timestep to $10^{-3}s$ and simulate for $St \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$.

All benchmarks are performed in serial on a 2022 MacBook Pro with M2 chip.

III. Results

A. FMM Parameter Tuning

Before performing any benchmarks, we must select FMM tuning parameters to optimize each solver's efficiency. To do this, we set all panel strengths to unity and performed a sweep of each tuning parameter in Table 1 by benchmarking the FMM and evaluating the error compared to a direct evaluation (no multipole expansions). A sample of the results for $P = 14$ is shown in Fig. 7. It is clear from this exercise the these tuning parameters can have a significant effect on the performance of the FMM. The optimal tuning parameters we found are reported in Table 2.

Table 1 FMM Tuning Parameter Sweep

parameter	range
P	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20\}$
θ	$\{0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7\}$
l	$\{5, 10, 15, 20, 25, 50, 100, 200, 300, 400\}$

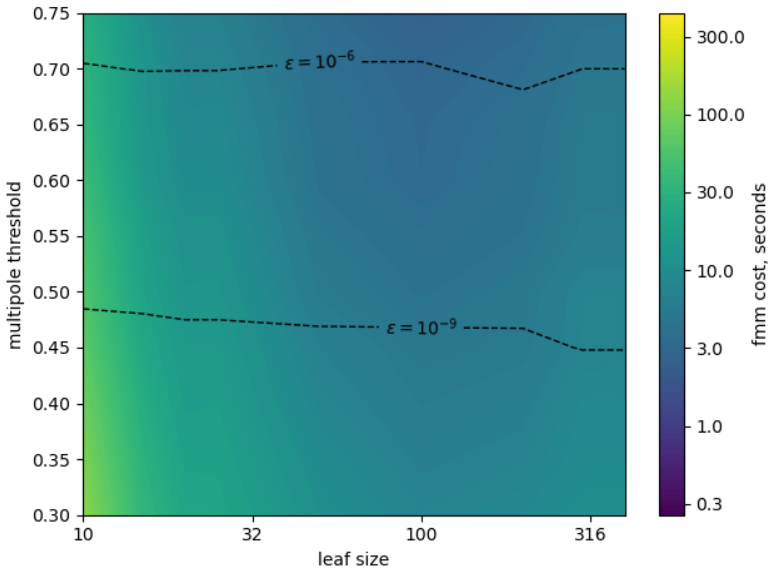


Fig. 7 Parameter sweep of the FMM tuning parameters for $P = 14$. Dotted lines denote error contours.

Table 2 FMM Optimal Parameters

ε	P^*	θ^*	l^*
10^{-3}	5	0.65	20
10^{-6}	7	0.4	30
10^{-9}	12	0.4	100

B. Cold Start Performance

First, we benchmark each solver to demonstrate the limitations of solver methods that explicitly build the matrix A. The cold-start benchmarks are shown for error tolerances of $\varepsilon \in \{10^{-3}, 10^{-6}, 10^{-9}\}$ in Figs. 8-10. The LU decomposition benchmarks are repeated in each figure, as it has no error control. First, we note the crossover size where the iterative methods become less expensive than LU decomposition and GMRES. This happens at roughly 7,000 panels for $\varepsilon = 10^{-3}$, 30,000 panels for $\varepsilon = 10^{-6}$, and 50,000 panels for $\varepsilon = 10^{-9}$. We remark that this crossover size would be much smaller in the presence of a good preconditioner, which we do not consider here. We also note a change in the $O(N^2)$ scaling of the LU decomposition and GMRES (slope of 2 on the log-log plot) to at least $O(N^3)$ (slope of 3) for the largest problem attempted. We hypothesize this to be the point at which the influence matrix no longer fits in RAM, representing a memory limit to these methods. Note also the $O(N)$ scaling of the MF-GMRES and FGS methods. Since the purpose of this work is to explore matrix-free methods that scale as $O(N)$, we will not consider LU decomposition or GMRES in the next section.

C. Warm Start Performance

In unsteady aerodynamic simulation, we can leverage the time continuity of the panel solution to warm start each subsequent solution to reduce its cost. The performance improvements of the warm-started solvers is illustrated in Fig. 11. On average, we see a roughly 20% reduction in cost for warm-starting, and roughly an order of magnitude reduction in cost with the Taylor-series extrapolated warm start. We see a slightly better performance for the smaller St, which is to be expected, as the panel strengths did not change as much. Also of note, cold-started MF-GMRES is more expensive for the higher St, suggesting it's performance depends on the right hand side (in this case, angle of attack), while cold-started FGS is relatively unaffected by St. This makes sense, as the inner iterations of FGS use the LU-factorization, whose cost does not depend on the right hand side.

We compare the residual behavior for cold-started, warm-started, and Taylor series extrapolated solvers in Fig. 12.

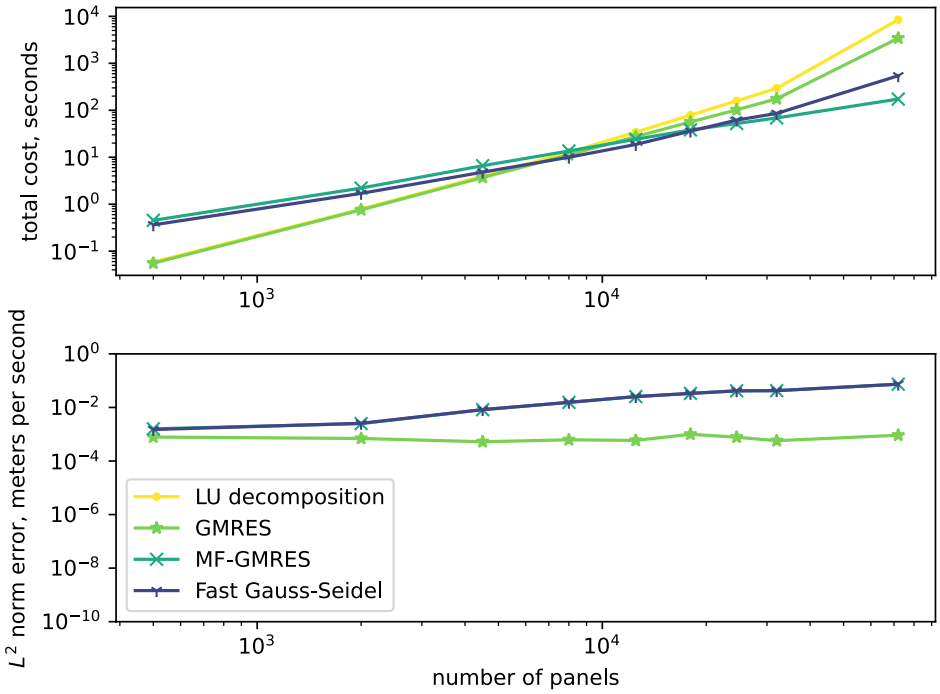


Fig. 8 Scaling comparison of the 4 solvers for a cold-start solve of a wing at 10° angle of attack with absolute error tolerance $\varepsilon = 10^{-3}$.

It is interesting to note that the convergence rate of cold-started MF-GMRES starts out faster than FGS, but flattens after a few iterations, and then proceeds at a rate slower than FGS until convergence. This contrasts with FGS, which converges at a relatively constant slope. Both solvers achieve very similar benchmarks for all three conditions, however.

IV. Conclusion

In this work, four solvers were compared with unsteady aerodynamics in mind. The $O(N^2)$ cost of forming the influence matrix of the first two, LU decomposition and GMRES where the matrix is explicitly formed, as well as the $O(N)$ cost of two matrix-free solvers is demonstrated. The first of the matrix-free solvers is GMRES with the matrix-vector product approximated by the fast multipole method (FMM). The second is a novel multigrid method combining FMM with Gauss-Seidel iterations. Without any preconditioner, we demonstrate an order of magnitude speedup by using Taylor series to extrapolate an initial guess for the two matrix-free solvers. Adding preconditioners to these solvers to improve their robustness is left for future work.

Acknowledgments

The paper was developed based on partial support through the Center for Autonomous Air Mobility and Sensing (CAAMS) under NSF award 2139551, and partial support from NASA under award No. 80NSSC21M0070.

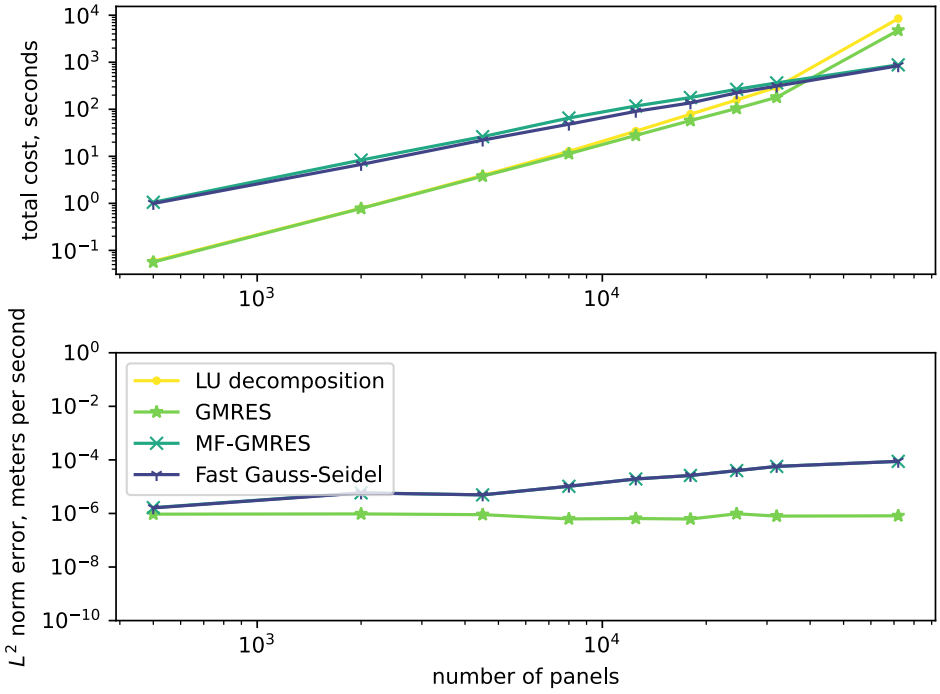


Fig. 9 Scaling comparison of the 4 solvers for a cold-start solve of a wing at 10° angle of attack with absolute error tolerance $\varepsilon = 10^{-6}$.

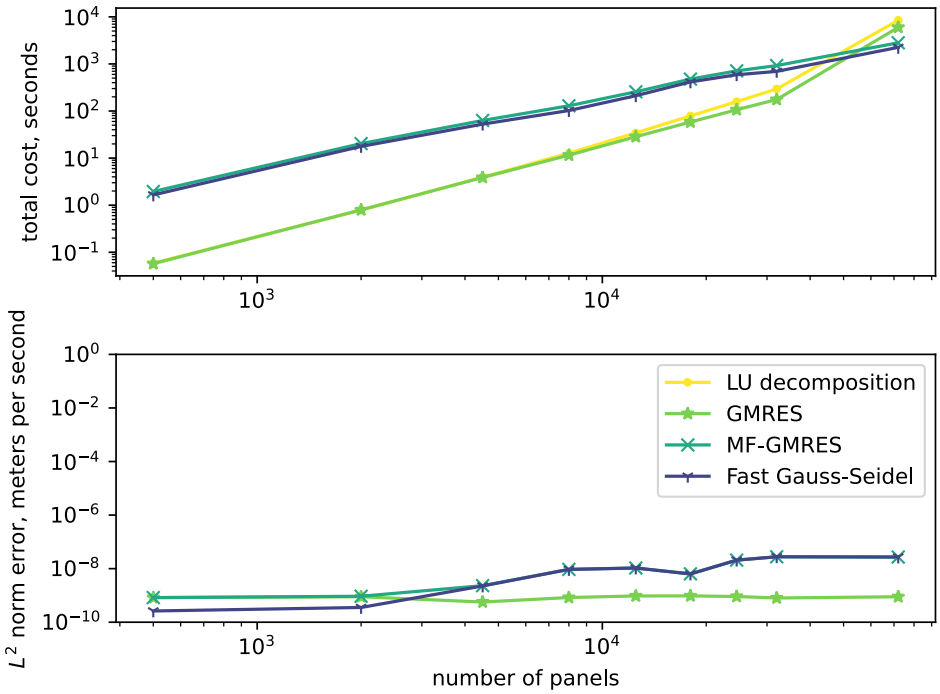


Fig. 10 Scaling comparison of the 4 solvers for a cold-start solve of a wing at 10° angle of attack with absolute error tolerance $\varepsilon = 10^{-9}$.

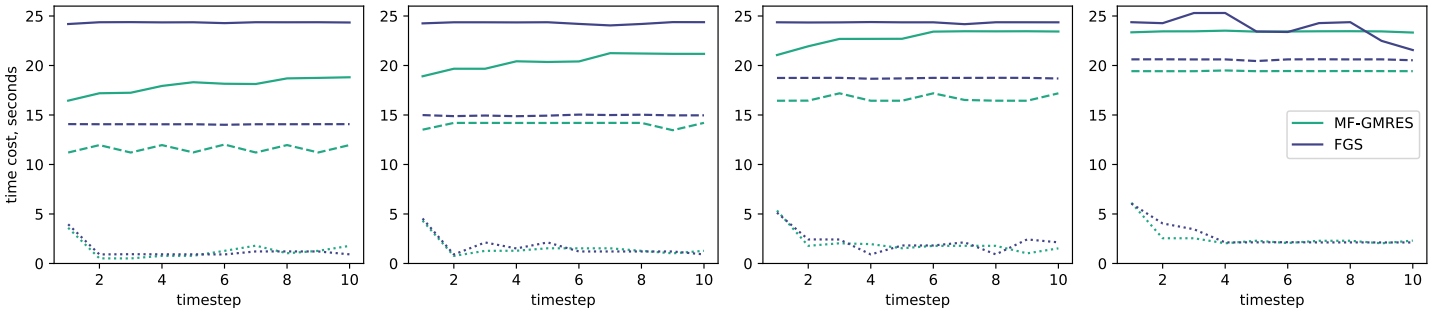


Fig. 11 Time cost of convergence of the matrix-free solvers for the first 10 timesteps of the wing heaving motion for 4 different Strouhal numbers, $St \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ from left to right. Solid lines represent cold-started solves, dashed lines represent warm-started solves, and dotted lines represent Taylor series-extrapolated warm-started solves.

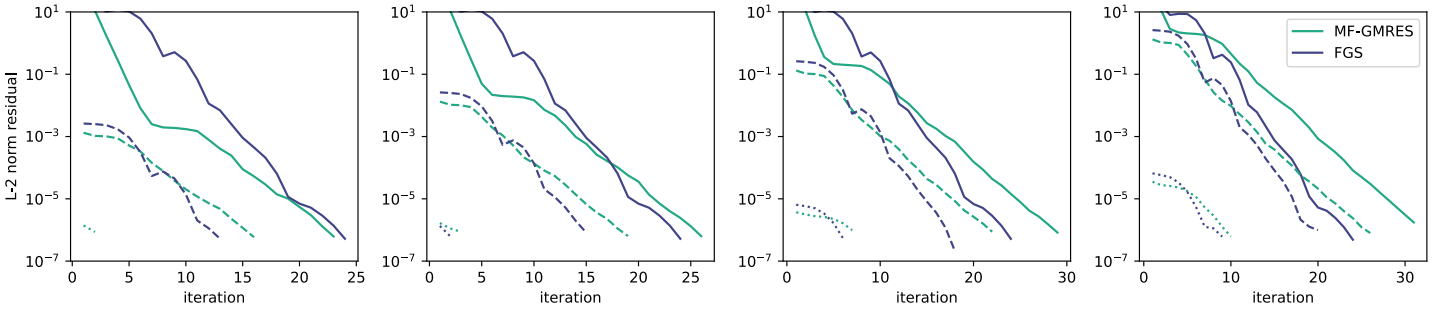


Fig. 12 Convergence of the residual of the matrix-free solvers for the second timestep of the wing heaving motion for 4 different Strouhal number, $St \in \{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ from left to right. Solid lines represent cold-started solves, dashed lines represent warm-started solves, and dotted lines represent Taylor series-extrapolated warm-started solves.

References

[1] Rothhaar, P. M., Murphy, P. C., Bacon, B. J., Gregory, I. M., Grauer, J. A., Busan, R. C., and Croom, M. A., "NASA Langley Distributed Propulsion VTOL Tilt-Wing Aircraft Testing, Modeling, Simulation, Control, and Flight Test Development," *14th AIAA aviation technology, integration, and operations conference*, 2014, p. 2999. https://doi.org/10.2514/6.2014-2999, URL http://arc.aiaa.org.

[2] Kim, H. D., Perry, A. T., and Ansell, P. J., "A review of distributed electric propulsion concepts for air vehicle technology," *2018 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*, IEEE, 2018, pp. 1–21.

[3] Leng, Y., Jardin, T., Bronz, M., and Moschetta, J.-M., "Experimental analysis of a blown-wing configuration during transition flight," *AIAA Scitech 2020 Forum*, 2020, p. 1983.

[4] Zanotti, A., Velo, A., Pepe, C., Savino, A., Grassi, D., and Riccobene, L., "Aerodynamic interaction between tandem propellers in eVTOL transition flight configurations," *Aerospace Science and Technology*, 2024, p. 109017.

[5] Dekker, H. N., Ragni, D., Baars, W. J., Scarano, F., and Tuinstra, M., "Aerodynamic Interactions of Side-by-Side Rotors in Ground Proximity," *AIAA Journal*, 2022, pp. 1–11.

[6] Healy, R., Misiorowski, M., and Gandhi, F., "A CFD-Based Examination of Rotor-Rotor Separation Effects on Interactional Aerodynamics for eVTOL Aircraft," *Journal of the American Helicopter Society*, Vol. 67, No. 1, 2022, pp. 1–12.

[7] Droandi, G., Syal, M., and Bower, G., "Tiltwing Multi-Rotor Aerodynamic Modeling in Hover, Transition and Cruise Flight Conditions," *AHS International 74th Annual Forum & Technology Display*, 2018.

[8] Alvarez, E. J., "Reformulated Vortex Particle Method and Meshless Large Eddy Simulation of Multirotor Aircraft," 2022.

[9] Greengard, L., and Rokhlin, V., "A fast algorithm for particle simulations," *Journal of computational physics*, Vol. 73, No. 2, 1987, pp. 325–348.

[10] Proulx-Cabana, V., Nguyen, M. T., Prothin, S., Michon, G., and Laurendeau, E., "A Hybrid Non-Linear Unsteady Vortex Lattice-Vortex Particle Method for Rotor Blades Aerodynamic Simulations," *Fluids*, Vol. 7, No. 2, 2022, p. 81.

[11] Anderson, R., and Ning, A., "A Coupled Source Panel, Actuator Line, and Viscous Vortex Particle Method in an O (n) Scheme," *AIAA SCITECH 2023 Forum*, 2023, p. 1381.

[12] Montagnani, D., Tugnoli, M., Fonte, F., Zanotti, A., Syal, M., and Droandi, G., "Mid-Fidelity Analysis of Unsteady Interactional Aerodynamics of Complex VTOL Configurations," *45th European Rotorcraft Forum (ERF 2019)*, 2019, pp. 1–11.

[13] Rajmohan, N., and He, C., "A VPM/CFD Coupling Methodology to Study Rotor/Ship Aerodynamic Interaction," *AIAA Modeling and Simulation Technologies Conference*, 2016, p. 1915.

[14] Stüben, K., et al., "An introduction to algebraic multigrid," *Multigrid*, 2001, pp. 413–532.

[15] Falgout, R. D., "An introduction to algebraic multigrid," Tech. rep., Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2006.

[16] Baker, A. H., Falgout, R. D., Gamblin, T., Kolev, T. V., Schulz, M., and Yang, U. M., "Scaling algebraic multigrid solvers: On the road to exascale," *Competence in High Performance Computing 2010: Proceedings of an International Conference on Competence in High Performance Computing, June 2010, Schloss Schwetzingen, Germany*, Springer, 2012, pp. 215–226.

[17] Ibeid, H., Yokota, R., Pestana, J., and Keyes, D., "Fast multipole preconditioners for sparse matrices arising from elliptic equations," *arXiv preprint arXiv:1308.3339*, 2013.

[18] Adelman, R., Gumerov, N. A., and Duraiswami, R., "FMM/GPU-accelerated boundary element method for computational magnetics and electrostatics," *IEEE Transactions on Magnetics*, Vol. 53, No. 12, 2017, pp. 1–11.

[19] Gunda, R., "Boundary element acoustics and the fast multipole method (FMM)," *Sound and Vibration*, Vol. 42, No. 3, 2008, p. 12.

[20] Jourdan de Araujo Jorge Filho, E., and Wang, Z. J., "A Matrix-free GMRES Algorithm on GPU Clusters for Implicit Large Eddy Simulation," *AIAA Scitech 2021 Forum*, 2021, p. 1837.

[21] Wang, T., Layton, S. K., and Barba, L. A., "Inexact GMRES iterations and relaxation strategies with fast-multipole boundary element method," *Advances in Computational Mathematics*, Vol. 48, No. 3, 2022, pp. 1–25.

[22] Aiello, G., Alfonzetti, S., Borzì, G., Dilettoso, E., and Salerno, N., "Solution of linear FEM-BEM systems for electrostatic field problems by means of GMRES," *International Symposium on Electric and Magnetic Fields*, 2006.

[23] Jiang, M., Hu, J., Zhao, R., Wei, X., and Nie, Z.-p., "Hybrid IE-DDM-MLFMA with Gauss-Seidel iterative technique for scattering from conducting body of translation," *The Applied Computational Electromagnetics Society Journal (ACES)*, 2015, pp. 148–156.

[24] Eibert, T. F., "Some scattering results computed by surface-integral-equation and hybrid finite-element-boundary-integral techniques, accelerated by the multilevel fast multipole method," *IEEE Antennas and Propagation Magazine*, Vol. 49, No. 2, 2007, pp. 61–69.

[25] Montoison, A., and Orban, D., "Krylov.jl: A Julia basket of hand-picked Krylov methods," *Journal of Open Source Software*, Vol. 8, No. 89, 2023, p. 5187. https://doi.org/10.21105/joss.05187.

[26] Gumerov, N. A., Kaneko, S., and Duraiswami, R., "Recursive computation of the multipole expansions of layer potential integrals over simplices for efficient fast multipole accelerated boundary elements," *Journal of Computational Physics*, Vol. 486, 2023, p. 112118.