Simulating Flows Passing a Wind Turbine with a Fully Implicit Domain Decomposition Method

Rongliang Chen[1], Zhengzheng Yan[1], Yubo Zhao[1], and Xiao-Chuan Cai[2]

Abstract
In this paper, we study a scalable overlapping domain decomposition method for solving the 3D unsteady incompressible Navier-Stokes equations and its application to the simulation of flows passing a full size wind turbine with realistic geometry and high Reynolds number. The algorithm features of a fully implicit finite element discretization on a moving unstructured mesh and a Newton-Krylov-Schwarz solver. We test the algorithm for a flow around a 5MW wind turbine with more than 8 million degrees of freedom on a supercomputer with up to 2048 processors.

1 Introduction

Wind power is an increasingly popular renewable energy. In the design process of the wind turbine blade, the accurate aerodynamic simulation is important. In the past, most of the wind turbine simulations were carried out with some low fidelity methods, such as the blade element momentum method [9]. Recently, with the rapid development of the supercomputers, high fidelity simulations based on 3D unsteady Navier-Stokes (N-S) equations become more popular. For example, Sorensen et al. studied the 3D wind turbine rotor using the Reynolds-Averaged Navier-Stokes (RANS) framework where a finite volume method and a semi-implicit method are used for the spatial and temporal discretization, respectively [18]. Bazilevs et al. investigated the aerodynamic of the NREL 5MW offshore baseline wind turbine rotor using large eddy simulation built with a deforming-spatial-domain/stabilized

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, P. R. China (rl.chen@siat.ac.cn, zz.yan@siat.ac.cn, yb.zhao@siat.ac.cn) ·
[2]Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA (cai@cs.colorado.edu)

space-time formulation [3, 10] and later extended the simulation to the full wind turbine including both the rotor and the tower [11]. Li et al. conducted dynamic overset CFD simulations for the NREL phase VI wind turbine using RANS and detached eddy models [15].

In this paper, we study a scalable parallel method based on the 3D unsteady incompressible N-S equations and its application to a NREL S-series wind turbine with realistic geometry and Reynolds number. In this simulation, the main challenges are: (1) the moving of the computation domain because of the rotation of the rotor; (2) the complex geometry; (3) the large computational meshes; and (4) the high nonlinearity resulting from high Reynolds number. To answer these challenges, an Arbitrary-Lagrange-Eulerian (ALE) method is used to handle the mesh movement, an unstructured tetrahedron mesh with a stabilized finite element method and a fully implicit backward difference scheme are employed to discretize the N-S equations [6, 19] and a parallel Newton-Krylov-Schwarz (NKS) method [4, 12] is used to solve the large sparse nonlinear system at each time step. In NKS, an inexact Newton method with analytic Jacobian is employed as the nonlinear solver, a Krylov subspace method is used as the linear Jacobian system solver in the Newton steps, and an overlapping domain decomposition method is used as a preconditioner to accelerate the convergence of the linear solver [5, 14]. For the rotor-only simulation, one can either fix the computation domain and apply a given velocity on the surface of the rotor, or let the domain move with the rotating rotor and apply a no-slip boundary condition on the rotor surface. We choose the latter one in this paper. We mainly focus on the solution method, including the robustness and parallel scalability.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the governing equations and their discretization. In Section 3, the Newton-Krylov-Schwarz algorithm is discussed, and some numerical results are presented in Section 4. In Section 5, we draw some conclusions.

2 Governing equations and a fully implicit discretization

We model the flow around the wind turbine using the 3D unsteady incompressible N-S equations. Since the computational domain moves during the simulation, a moving mesh method is introduced to handle the change of the flow domain. In this paper, we use the ALE method. Let $\mathbf{Y}$ be the ALE coordinate, $\mathbf{X}$ the Eulerian coordinate. Then the N-S equations read as [7]:

$$
\begin{aligned}
\rho\left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{Y}} + (\mathbf{u}-\omega)\cdot\nabla\mathbf{u}\right) + \nabla\cdot\sigma =&\ \mathbf{f} \ \text{ in } \ \Omega^t, \\
\nabla\cdot\mathbf{u} =&\ 0 \ \text{ in } \ \Omega^t, \\
\mathbf{u} =&\ \mathbf{g} \ \text{ on } \Gamma_{inlet} \\
\sigma\cdot\mathbf{n} =&\ \mathbf{0} \ \text{ on } \Gamma_{outlet}, \\
\mathbf{u} =&\ \mathbf{0} \ \text{ on } \Gamma_{wall}, \\
\mathbf{u} =&\ \mathbf{u_0} \ \text{ in } \ \Omega^t \ \text{ at } \ t=0,
\end{aligned} \tag{1}
$$

where Ω^t is the computational domain at time t and $\omega = \frac{\partial \mathbf{x}}{\partial t}$ is the velocity of the rotating flow domain which is equal to the rotor speed since we let the whole computation domain rotate with the rotor. $\sigma = -p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the Cauchy stress tensor. $\mathbf{u}$ and p are the velocity and pressure of the flow. ρ and μ are the density and viscosity of the fluid, respectively. $\mathbf{f}$ refers to the source term and $\mathbf{g}$ is a given function defined at the inlet boundary. $\mathbf{u_0}$ is a given initial condition which is zero in our test cases. Γ_{inlet}, Γ_{outlet} and Γ_{wall} refer to the inlet, outlet and wall boundaries, respectively.

A $P1-P1$ finite element method is used to discretize (1) on an unstructured tetrahedral mesh $\mathcal{T}^h = \{K\}$. Since this finite element method is not stable for the N-S equations because it does not satisfy the Ladyzenskaja-Babuska-Brezzi (LBB) condition, additional stabilization terms are needed in the formulation as described in [2]. We denote the finite element spaces of the trial and weighting functions for the velocity and pressure as $\mathcal{U}^h$, $\mathcal{U}^{0,h}$, and $\mathcal{P}^h$, respectively. Then the semi-discrete stabilized finite element formulation of (1) is given as follows: Find $\mathbf{u}^h \in \mathcal{U}^h$ and $p^h \in \mathcal{P}^h$, such that for any $\Phi^h \in \mathcal{U}^{0,h}$ and $\psi^h \in \mathcal{P}^h$,

$$
\mathbf{B}^h(\mathbf{u}^h, p^h; \Phi^h, \psi^h) - \mathbf{F}^h(\Phi^h, \psi^h) = \mathbf{0}, \tag{2}
$$

where $\mathbf{u}^h$, p^h are the nodal values of the velocity and pressure functions, φ^h and each of the three components of Φ^h are the basis functions which are piecewise continuous linear functions, and

$$
\begin{aligned}
\mathbf{B}^h(\mathbf{u}^h, p^h; \Phi^h, \psi^h) =&\ \rho\int_{\Omega^t} \left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\mathbf{Y}} \cdot \Phi^h d\Omega^t + \mu\int_{\Omega^t} \nabla\mathbf{u}^h : \nabla\Phi^h d\Omega^t \\
&+\rho\int_{\Omega^t} ((\mathbf{u}^h-\omega)\cdot\nabla)\mathbf{u}^h\cdot\Phi^h d\Omega^t - \int_{\Omega^t} p^h\nabla\cdot\Phi^h d\Omega^t \\
&+\int_{\Omega^t} (\nabla\cdot\mathbf{u}^h)\varphi^h d\Omega^t + \sum_{K\in\mathcal{T}} \left(\nabla\cdot\mathbf{u}^h,\ \tau_c\nabla\cdot\Phi^h\right)_K \\
+\sum_{K\in\mathcal{T}} &\left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\mathbf{Y}} + ((\mathbf{u}^h-\omega)\cdot\nabla)\mathbf{u}^h + \nabla p^h,\ \tau_m(\mathbf{u}^h\cdot\nabla\Phi^h + \nabla\varphi^h)\right)_K,
\end{aligned}
$$

$$
\mathbf{F}^h(\Phi^h, \psi^h) = \int_{\Omega^t} \mathbf{f}\cdot\Phi^h d\Omega^t + \sum_{K\in\mathcal{T}} \left(\mathbf{f},\ \tau_m(\mathbf{u}^h\cdot\nabla\Phi^h + \nabla\varphi^h)\right)_K.
$$

Here the parameters τ_c and τ_m are defined as in [2].

For the temporal discretization, we use an implicit backward finite difference formula with a fixed time step size Δt. In the implicit method, one needs to solve a nonlinear system at each time step (the n^{th} time step):

$$\mathbf{F}^n(\mathbf{U}^n) = \mathbf{0}, \tag{3}$$

to obtain the solution of the n^{th} time step $\mathbf{U}^n$, which consists of the nodal values of the velocity and pressure.

3 Monolithic Newton-Krylov-Schwarz algorithm

In most N-S solvers, such as the projection methods, the operator is split into the velocity component and pressure component, and the algorithm takes the form of a nonlinear Gauss-Seidel iteration with two large blocks. In the monolithic approach that we consider in this paper, the velocity and pressure variables associated with a grid point stay together throughout the computation. In this approach, the two critically important ingredients, namely the monolithic Schwarz preconditioner, and the robustness and scalability are realized with the point-block ILU based subdomain solver.

The nonlinear system (3) is solved by a Newton-Krylov-Schwarz method which reads as

- *Let* $\mathbf{U}^0$ *be the given initial condition and set* $n = 0$
- *For* $n = 1, 2, \cdots,$ *do*
  - *Using an initial guess* $\mathbf{U}_0^n = \mathbf{U}^{n-1}$ *and set* $k = 0$
  - *Move the computational domain* ($\Omega^{n-1} \rightarrow \Omega^n$) *and the mesh* $\mathcal{T}_h^n$ *(the coordinate of each mesh point at the current time step* $\mathbf{x}^n$ *is obtained by rotating the initial mesh* $\mathbf{x}^0$*):*

$$\mathbf{x}^n = \begin{bmatrix} cos(\omega n \Delta t) & -sin(\omega n \Delta t) & 0 \\ sin(\omega n \Delta t) & cos(\omega n \Delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}^0$$

  - *For* $k = 1, 2, \cdots,$ *until converges, do*
    - *Find* d_k^n *such that*

$$\| \nabla \mathbf{F}^n(\mathbf{U}_{k-1}^n)(\mathbf{M}_k^n)^{-1}(\mathbf{M}_k^n \mathbf{d}_k^n) + \mathbf{F}^n(\mathbf{U}_{k-1}^n) \| \leq \eta \| \mathbf{F}^n(\mathbf{U}_{k-1}^n) \| \tag{4}$$

    - *Set* $\mathbf{U}_k^n = \mathbf{U}_{k-1}^n + \tau_k^n \mathbf{d}_k^n$
  - *Set* $\mathbf{U}^n = \mathbf{U}_k^n$

Here $(\mathbf{M}_k^n)^{-1}$ is an additive Schwarz preconditioner to be defined shortly, ω is the angular speed of the rotor, and η is the relative tolerance for the linear solver. Not that, in the wind turbine simulation, we simply let the whole computational domain rotate at the same angular speed as the rotor, so the current mesh can be obtained by rotating the initial mesh and the connectivity of the mesh does not change. For simplicity, we ignore the scripts n and k for the rest of the paper.

In NKS, the most difficult and time-consuming step is the solution of the large, sparse, and nonsymmetric Jacobian system (4) by a preconditioned GMRES method. In the Jacobian solver, the most important component is the preconditioner, without which GMRES doesn't converge or converges very slowly, and a good preconditioner accelerates the convergence significantly. In this paper, we use an overlapping restricted additive Schwarz preconditioner introduced in [5]:

$$\mathbf{M}_{RAS} = \sum_{l=1}^{n_p} (R_l^0)^T \mathbf{J}_l^{-1} R_l^{\delta}, \tag{5}$$

where $\mathbf{J}_l$ is the local Jacobian matrix defined on the overlapping subdomain, n_p is the number of subdomains, R_l^{δ} and R_l^0 are the restriction operators from the whole domain to the overlapping and non-overlapping subdomain, respectively. In practice, we only need the application of $\mathbf{J}_l^{-1}$ to a given vector, which can be obtained by solving a subdomain linear system. Since $\mathbf{J}_l^{-1}$ is used as a preconditioner here, the subdomain linear system can be solved exactly or approximately by using LU factorization or incomplete LU factorization (ILU) in the point-block format [17].

4 Numerical experiments

In this section, we report some numerical experiments using the proposed algorithm. Our solver is implemented on top of the Portable Extensible Toolkit for Scientific computation (PETSc) [1]. The computations are carried out on the Dawning Nebulae supercomputer at the China National Supercomputer Center at Shenzhen. The geometry of the wind turbine is provided by GrabCAD[1] (we scale the size to that of a 5MW wind turbine) and meshed by ANSYS; see Figure 1 for details. The mesh partitions for the additive Schwarz preconditioner are obtained with ParMETIS [13]. The relative stopping conditions for the nonlinear and linear solvers are 10^{-12} and 10^{-6}, respectively.

In the experiments, we set the wind speed to be uniform at $15m/s$ and the rotor speed to be $22rpm$ (revolutions per minute). For the air flow, we set the kinematic viscosity $\mu = 1.831 \times 10^{-5} kg/(ms)$ and the density $\rho = 1.185 kg/m^3$. Figure 2 shows the simulation results: the velocity distribution and the isosurface of the flow at $t = 10.0s$, which are obtained on a mesh with about 1.1×10^7 elements and a fixed time step size $\Delta t = 0.01s$.

[1] `www.grabcad.com`

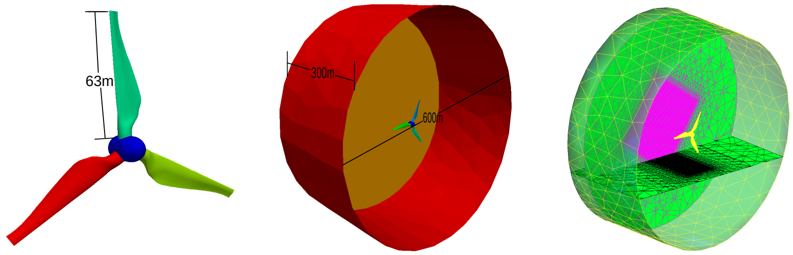

Fig. 1 A three-blade wind turbine with NREL S807 root region airfoil and NREL S806 tip region airfoil from GrabCAD (left), the computational domain (mid), and the computational mesh (right).

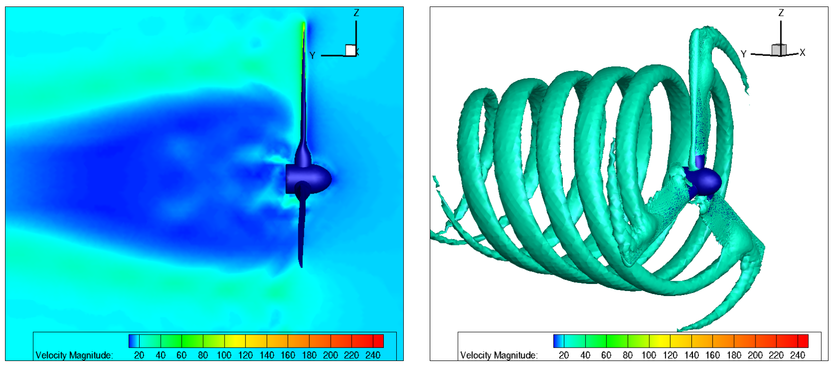

Fig. 2 The velocity distribution (left) and the isosurface (right) of the simulation

The parallel performance results are given in Table 1 for two different subdomain solvers ILU(2) and ILU(3) (here 2 and 3 refer to the fill-in levels of the point-block ILU factorization). With the increase of the number of processors (n_p) from 512 to 2048, the number of Newton iterations (Newton) changes a little, the number of GMRES iterations (GMRES) increases reasonably, and the compute time (Time) decreases. These results show that the algorithm scales well when n_p is around 1024 or less and the efficiency reduces with the increase of n_p, which is reasonable because we use a one-level method. The result also suggests that for large number of processors, in order to obtain a good scalability, multilevel methods are necessary.

5 Concluding remarks

A domain decomposition based fully implicit parallel algorithm for the numerical simulation of the flow around a wind turbine rotor was introduced and

Table 1 Parallel performance of the algorithm. Here the degrees of freedom (DOF) is about 8.4×10^6 and the overlapping size is 4. The "Time" refers to the average compute time in seconds at each time step.

n_p	ILU(3)			ILU(2)		
	Newton	GMRES	Time(s)	Newton	GMRES	Time(s)
512	3.0	51.72	127.3	3.0	64.02	75.0
1024	3.0	52.77	77.7	3.0	66.40	45.8
1536	3.1	53.94	67.5	3.0	67.60	35.6
2048	3.0	57.42	53.0	3.0	67.75	29.3

studied in this paper. The algorithm begins with a fully implicit discretization of the unsteady incompressible N-S equations on a moving unstructured mesh with a stabilized finite element method, then an inexact Newton method is employed to solve the large nonlinear system at each time step, and a preconditioned GMRES method is employed to solve the linear Jacobian system in each Newton step with a one-level restricted additive Schwarz preconditioner. We tested the algorithm for a flow around a 5MW wind turbine with more than 8 million degrees of freedom on a supercomputer with up to 2048 processors. The algorithm scales well when the number of processors is around 1024 or less. We plan to develop a multilevel version of the algorithm in order to obtain better scalability results when the number of processors is larger.

Acknowledgments

The research was supported in part by the Knowledge Innovation Program of the Chinese Academy of Sciences under KJCX2-EW-L01 and the Shenzhen Peacock Plan (China) under KQCX20130628112914303.

References

[1] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. *PETSc Users Manual*. Technical Report, Argonne National Laboratory, 2013.

[2] Y. Bazilevs, V. Calo, T. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Comput. Mech.*, 43: 3-37, 2008.

[3] Y. Bazilevs, M. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: geometry modeling and aerodynamics. *Int. J. Numer. Meth. Fluids*, 65: 207-235, 2011.

[4] X.-C. Cai, W. Gropp, D. Keyes, R. Melvin, and D. Young. Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation. *SIAM J. Sci. Comput.*, 19: 246-265, 1998.

[5] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21: 792-797, 1999.

[6] R. Chen, Q. Wu, Z. Yan, Y. Zhao, and X.-C. Cai. A parallel domain decomposition method for 3D unsteady incompressible flows at high Reynolds number. *J. Sci. Comput.*, 58: 275-289, 2014.

[7] F. Duarte, R. Gormaz, and S. Natesan. Arbitrary Lagrangian-Eulerian method for Navier-Stokes equations with moving boundaries. *Comput. Methods Appl. Mech. Engrg.*, 45: 4819-4836, 2004.

[8] S. Eisenstat and H. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM J. Sci. Comput.*, 17: 16-32, 1996.

[9] M. Hansen. *Aerodynamics of Wind Turbines, 2nd ed.* Erthscan, London. 2008.

[10] M. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Comput. Fluids*, 49: 93-100, 2011.

[11] M. Hsu and Y. Bazilevs. Fluid-structure interaction modeling of wind turbines: simulating the full machine. *Comput. Mech.*, 50: 821-833, 2012.

[12] F.-N. Hwang, C.-Y. Wu, and X.-C. Cai. Numerical simulation of three-dimensional blood flows using domain decomposition method on parallel computer. *J. CSME*, 31: 199-208, 2010.

[13] G. Karypis. METIS/ParMETIS webpage. University of Minnesota, `http://glaros.dtc.umn.edu/gkhome/views/metis`, 2013.

[14] A. Klawonn and L. Pavarino. Overlapping Schwarz methods for mixed linear elasticity and Stokes problems. *Comput. Methods Appl. Mech. Engrg.*, 165: 233-245, 1998.

[15] Y. Li, K. Paik, T. Xing, and P. Carrica. Dynamic overset CFD simulations of wind turbine aerodynamics. *Renew. Energy*, 37: 285-298, 2012.

[16] S. Owen and J. Shepherd. CUBIT project webpage. Sandia National Laboratories, `http://cubit.sandia.gov/`, 2013.

[17] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston. 1996.

[18] N. Sorensen, J. Michelsen, and S. Schreck. Navier-Stokes predictions of the NREL phase VI rotor in the NASA Ames 80 ft $\times$ 120 ft wind tunnel. *Wind Energ.*, 5: 151-169, 2002.

[19] Y. Wu and X.-C. Cai. A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation. *J. Comput. Phys.*, 258: 524-537, 2014.