Commun. Comput. Phys.
doi: 10.4208/cicp.150313.070513s

Vol. **15**, No. 4, pp. 944-958
April 2014

A Scalable Numerical Method for Simulating Flows Around High-Speed Train Under Crosswind Conditions

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

[1] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, P.R. China.*
[2] *Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA.*

Received 15 March 2013; Accepted (in revised version) 7 May 2013

Communicated by Peter Jimack

Available online 21 January 2014

Abstract. This paper presents a parallel Newton-Krylov-Schwarz method for the numerical simulation of unsteady flows at high Reynolds number around a high-speed train under crosswind. With a realistic train geometry, a realistic Reynolds number, and a realistic wind speed, this is a very challenging computational problem. Because of the limited parallel scalability, commercial CFD software is not suitable for supercomputers with a large number of processors. We develop a Newton-Krylov-Schwarz based fully implicit method, and the corresponding parallel software, for the 3D unsteady incompressible Navier-Stokes equations discretized with a stabilized finite element method on very fine unstructured meshes. We test the algorithm and software for flows passing a train modeled after China's high-speed train CRH380B, and we also compare our results with results obtained from commercial CFD software. Our algorithm shows very good parallel scalability on a supercomputer with over one thousand processors.

AMS subject classifications: 76D05, 76F65, 65M55, 65Y05

Key words: Three-dimensional unsteady incompressible flows, high-speed train, crosswind, full Navier-Stokes equations, Newton-Krylov-Schwarz algorithm, parallel computing.

1 Introduction

In recent years, high-speed trains with characteristics of high-speed, high-efficiency and low-energy consumption have been rapidly developed. When the wind is strong, trains

*Corresponding author. *Email addresses:* zz.yan@siat.ac.cn (Z. Yan), rl.chen@siat.ac.cn (R. Chen), yb.zhao@siat.ac.cn (Y. Zhao), cai@cs.colorado.edu (X.-C. Cai)

experience certain aerodynamic phenomena such as increased drag and lift, reduced stability and raised level of acoustic noise. Unfortunately, the increase of speed often enhances these negative effects, and some of which may result in accidents if they are not controlled correctly. Several train accidents caused by crosswind were reported in [6,22]. Crosswind is one of the main threats that impact passenger's comfort and/or vehicle's safety in high-speed train transportation. Therefore, it is both important and necessary to carefully study the aerodynamic performance of trains under crosswind conditions.

Experimental techniques and commercial software based numerical simulations are the two main tools in the train industry since analytical techniques are only applicable for a few limited cases. Experimental investigations are usually expensive and subject to the restrictions of design cycle, special boundary conditions and Reynolds numbers, and sometimes become not applicable or even impossible for high-speed trains. Numerical simulations are used more often and now become an important engineering tool, especially in the early design stages.

Historically, numerical simulations of crosswind around a train are inherited from a simple model, the Ahmed body [1], which includes most of the flow features in a real vehicle. Khier et al. [16] first use Reynolds averaged Navier-Stokes (RANS) equations combined with the $k-\epsilon$ turbulence model to study the flow structures around a simplified train under crosswind. In recent years, researchers have applied other techniques in solving this problem numerically. The majority of these simulations are to apply some kind of statistical techniques to reduce the computational cost. RANS [9], large eddy simulation (LES) [10,11] and detached eddy simulation (DES) [7,17] are the three main approaches. All of these approaches can successfully obtain the flow structures relatively accurately. However, with the speed of trains increasing, certain design problems neglected at low speed are raised, more detailed computations are needed for the operational safety of a high-speed train. In this paper, we solve the full Navier-Stokes equations numerically without any further simplifications, on a very fine mesh of size $10^{-3}m$ in order to capture the subtle features and obtain sufficiently accurate solutions. Due to the limitation of computer memory and processing speed, this method is usually considered as a research tool rather than an engineering tool. In the past few years, the full Navier-Stokes equations have been successfully used to the low Reynolds number cases, but high Reynolds number flows implies a wide instantaneous range of scales, and high computing cost. Thanks to the recent development of large scale parallel computers with a large number of processors, the computational capability is improved significantly. This makes it possible to use full Navier-Stokes for engineering problems like the aerodynamics of a high-speed train. Correspondingly, the parallel scalability becomes an important indicator of a good numerical algorithm.

In this work, we present a scalable parallel Newton-Krylov-Schwarz (NKS) method for the full unsteady incompressible Navier-Stokes equations describing the flows around high-speed train under crosswind. Generally speaking, the NKS method is suitable for solving large, sparse nonlinear systems and has been widely applied in various problems [4, 5, 13, 14, 20]. In our algorithm, a fully implicit backward difference scheme is

adopted for the temporal discretization and a stabilized finite element method for the spatial discretization on an unstructured tetrahedral mesh. The nonlinear systems are solved by using an inexact Newton method in which the Jacobian system is solved approximately rather than exactly at each time step. In each Newton step, a Krylov subspace method (GMRES) is used as a linear solver which is accelerated by an overlapping Schwarz preconditioner [19]. In addition, the pressure and velocity variables are not separated but coupled at each mesh point in the process of finite element analysis. The aim of this operation is to make it possible to construct point-block incomplete LU factorization which is more stable than the classical pointwise ILU.

We study our algorithm and software for a realistic train geometry with high Reynolds number. We also compare our methods with the commercial CFD software package ANSYS CFX. The numerical results show that our simulations provide more detailed information of the flow structures and our algorithm shows a good parallel scalability.

The paper is organized as follows. The high-speed train model and the computational domain are presented in Section 2. The governing equations, the finite element discretization and the Newton-Krylov-Schwarz algorithm are described in Section 3. In Section 4, some computational results are reported. The paper ends with some remarks in Section 5.

2 Physical models

In this section, we first describe the vehicle model and then discuss the computational domain and the gust model.

The train model is derived from China's new generation of high-speed train shown in Fig. 1, namely CRH380B, which is designed for passenger transportation. The operation speed is $300km/h$ and the maximum speed is up to $380km/h$ [21]. The geometry is symmetric and consists of four sets of wheels, a locomotive and a container. To simulate an actual train, we use the full scale parameters. The length of locomotive and container are $24.75m$ and $24.01m$, respectively, which make the total length of the model $L=48.76m$. The height of the platform from the ground is $0.915m$. The width and height of the train are $W=3.26m$ and $H=3.39m$, respectively.

The computational domain for the high-speed train subject to crosswind is shown in Fig. 1. This domain is chosen as in [16] and it is large enough to avoid the influence of the boundaries. The wind-inlet boundary (face $ABFE$) and the outflow boundary (face $DCGH$) are set to $10H$ and $13.3H$ far from the train, respectively. The distance between the inlet boundary (face $ABCD$) and the nose of the train is $13.3H$, the distance between the outflow boundary (face $FGHE$) and the end of the train is $26.6H$. The height of the computational domain (ground face $BCGF$ to top face $ADHE$) is set to $10H$.

We assume that the direction of the crosswind is perpendicular to the direction that the train travels. The impact of the crosswind is the result of the velocity of the wind and the train, so the effect of the crosswind can be strong even if the crosswind's velocity

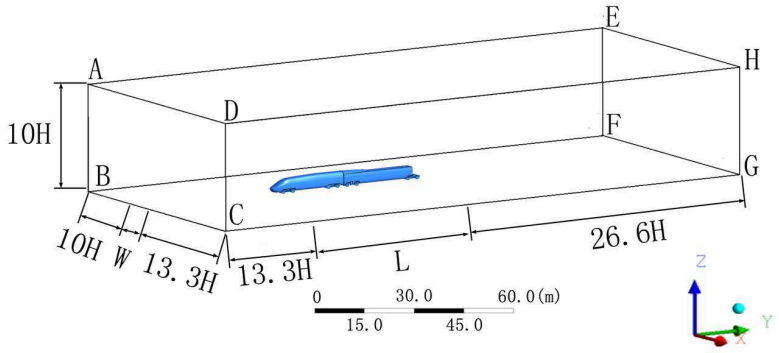


Figure 1: Dimensions of the model train and the computational domain.

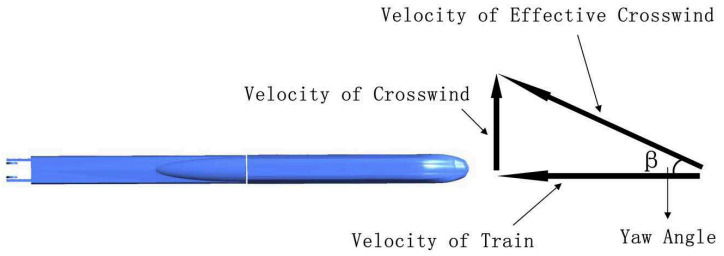


Figure 2: Effective crosswind.

is low. The yaw angle describes the direction between the effective crosswind and the direction at which the train travels, as Fig. 2 shows.

In practice, the wind velocity is stochastic and is usually influenced by the weather condition and the topography of the nature environment, therefore in this particular experiment, we set a few functions that vary in both time and space to describe the velocities of the wind and the train,

$$V_{wind} = -V_{wind}^{max} \frac{4(y-a)(y-b)}{(b-a)^2} t, \tag{2.1}$$

where $a=-44m$, $b=136m$ are the minimum and maximum y values of the computational domain.

$$V_{train} = -V_{train}^{max} \frac{4(x-c)(x-d)}{(d-c)^2} t, \tag{2.2}$$

where $c=-34.63m$, $d=45.63m$ are the minimum and maximum x values of the computational domain. In our simulation, the flows around the train are initialized with zero velocity and pressure everywhere.

3 Numerical methods

In this paper, the full 3D unsteady incompressible Navier-Stokes equations are used to describe the flows in the spatial domain $\Omega \subset R^3$,

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{3.1}$$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mathbf{f} + \mu\nabla^2\mathbf{u} \qquad \text{in } \Omega, \tag{3.2}$$

where $\mathbf{u}$ is the velocity representing the three components in the Cartesian coordinates x, y and z. ρ, p and μ are the density, pressure, dynamic viscosity, respectively. $\mathbf{f}$ is the body force. The governing equations (3.1) and (3.2) are equipped with boundary conditions on $\partial\Omega = \Gamma_{inlet} \cup \Gamma_{outlet} \cup \Gamma_{wall}$

$$\mathbf{u} = \mathbf{g} \qquad \text{on } \Gamma_{inlet}, \tag{3.3a}$$

$$\mathbf{u} = 0 \qquad \text{on } \Gamma_{wall}, \tag{3.3b}$$

$$\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p\mathbf{n} = 0 \qquad \text{on } \Gamma_{outlet}, \tag{3.3c}$$

and the initial condition

$$\mathbf{u} = \mathbf{u_0} \qquad \text{in } \Omega \quad \text{at } t = 0. \tag{3.4}$$

Here $\mathbf{g}$ represents the effective crosswind velocity.

To solve the set of governing equations, we first employ a P1-P1 stabilized finite element method introduced in [8] for the spatial discretization on the unstructured tetrahedral mesh and an implicit backward Euler finite difference formula for the temporal discretization. Because of the page limit, we don't discuss the details of the finite element discretization in this paper. Then, the governing equations are turned into the following system of nonlinear algebraic equations,

$$\frac{x^n - x^{n-1}}{\Delta t} = S(x^n), \tag{3.5}$$

where Δt is the time step size and $S(x^n)$ represents the system after the spatial discretization without the temporal term at the n^{th} time step. A large, sparse, nonlinear system

$$F^n(x^n) = 0, \tag{3.6}$$

needs to be solved at each time step. In addition, when defining the unknowns vector x^n of nodal values, the pressure and the velocity variables, are not ordered field-by-field like most approaches but element-by-element. The aim is to avoid the saddle point problem when solving the Jacobian system arising from the field-by-field ordering [12] because the saddle point problem significantly affects the convergence and the parallel performance.

In this work, we solve the nonlinear system (3.6) by a Newton-Krylov-Schwarz (NKS) algorithm. Roughly speaking, the NKS algorithm consists of three components, a nonlinear solver, a linear solver at each Newton step and an accelerator. The basic steps of the algorithm are listed below,

Step (a): Get an initial guess x_0^n using the solution of previous time step x^{n-1},

$$x_0^n = x^{n-1}. \tag{3.7}$$

Step (b): Obtain the Newton direction s_k^n by solving the following preconditioned Jacobian system inexactly with GMRES.

$$J_k^n (M_k^n)^{-1} M_k^n s_k^n = -F^n(x_k^n), \tag{3.8}$$

where J_k^n is the Jacobian of F^n at x_k^n, $(M_k^n)^{-1}$ is an additive Schwarz preconditioner.

Step (c): Find a step length λ_k^n using a cubic line search to update the approximation as follow,

$$x_{k+1}^n = x_k^n + \lambda_k^n s_k^n, \tag{3.9}$$

where x_k^n and x_{k+1}^n are the current and new approximation of x^n, respectively.

In Step (b), we solve the Jacobian system (3.8) inexactly by satisfying the condition

$$\|F(x_k^n) + J_k^n s_k^n\| \leq \eta_k \|F(x_k^n)\|, \tag{3.10}$$

where η_k is the forcing term used to control the accuracy of the solution.

Without preconditioning, GMRES in Step (b) usually converges slowly or even doesn't converge. In our algorithm, an overlapping restricted additive Schwarz (RAS) method [3] is employed as the preconditioner to accelerate the convergence. We first partition the computational domain Ω into n_p non-overlapping subdomains Ω_i, $i=1,2,\cdots,n_p$ where n_p is the number of processors. Then we get the overlapping subdomains Ω_i^δ by extending δ layers of mesh elements from the adjacent subdomains. Then each overlapping subdomain Ω_i^δ has its local Jacobian matrix J_i and the global-to-local restriction operator R_i^δ is defined to return all of the degree of freedom (DOF) belonging to Ω_i^δ from the global vector of unknowns. We define R_i^0 as the restriction operator for the non-overlapping subdomain Ω_i similar as R_i^δ. Then, the RAS preconditioner is defined as

$$M_{RAS} = \sum_{i=1}^{n_p} (R_i^0)^T B_i^{-1} R_i^\delta, \tag{3.11}$$

where B_i^{-1} is the local preconditioner. In this paper, The linear system corresponding to B_i^{-1} is solved by a point-block incomplete LU [18] with some levels of fill-in, which is more economical than LU as the local preconditioner.

4 Numerical experiments and discussions

In this section, a test case with $V_{wind} = 20m/s$ and $V_{train} = 100m/s$ is studied. The fluid material we adopt is air at 25°C with density $\rho = 1.185kg/m^3$ and dynamic viscosity $\mu =$

$1.831\times10^{-5}kg/ms$. The yaw angle $\beta\approx20.27^{\circ}$ and the Reynolds number $Re=2.238\times10^{7}$ is defined as

$$Re=\frac{\rho u_{\infty}D}{\mu}, \tag{4.1}$$

where u_{∞} is the freestream velocity which equals to the effective crosswind velocity, D is the characteristic length of the train and $D=H=3.39m$.

The computational domain is meshed by ANSYS ICEMCFD, a proprietary software package used for mesh generation. A mesh with about 14 million tetrahedral cells is generated for our test and the minimum volume of the mesh element is up to $3.9\times10^{-7}m^3$. The mesh is finer in the leeward side and near the train body where the flow field is of greater interest, as shown in Fig. 3.

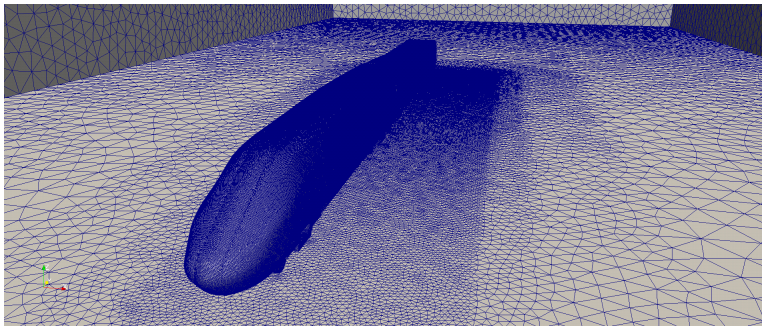


Figure 3: A view of the inside of the meshed domain.

ParMETIS [15] is used to partition the computational meshes into n_p subdomains for parallel computing. Fig. 4 is a schematic view of the partition with 12 subdomains. Note that the number of elements in the subdomains is more or less the same for the purpose of load balancing, but the size of subdomains can be very different. Our algorithm described in Section 3 is implemented using PETSc [2] of Argonne National laboratory.

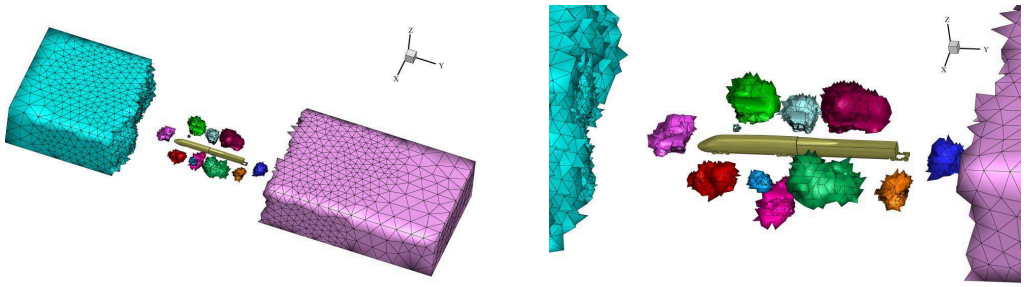


Figure 4: Schematic view of the partition of the computational mesh.

In this paper, the numerical tests are carried out on a DAWNING NEBULAE supercomputer with 640 nodes. Each node has a dual six-core Intel Xeon X5650@2.76GHz

processor and 24GB of memory. The numerical results obtained by our algorithm and ANSYS CFX are used for comparative analysis at first. Then, the parallel performance of our algorithm is investigated for solving these large-scale computational problems.

4.1 Comparative assessment of numerical results obtained by our algorithm and ANSYS CFX

We compare the NKS method and RANS with $k-\epsilon$ closure model by ANSYS CFX. The same geometry, computational mesh and boundary conditions are applied for these two simulations. Using an element-based finite volume approach, CFX applies the standard $k-\epsilon$ model to describe the turbulence behavior. A second-order backward Euler scheme is used for the transient term. The residual target of root mean square (RMS) convergence criteria is set to 1.0×10^{-6}. In the NKS algorithm, the computational domain is divided into subdomains and the number of subdomains is equal to the number of processors. The relative tolerances of linear and nonlinear convergence are set to 1.0×10^{-4} and 1.0×10^{-6}, respectively. For these two simulations, a constant time step Δt is set to $4.0\times10^{-3}s$ throughout the entire simulation, the solutions obtained at $t=1s$ are used for the comparison.

In order to present the results clearly, we create five cross sections at the location of $y=3m$ $(y/L=0.06)$, $15m$ $(y/L=0.31)$, $27m$ $(y/L=0.55)$, $39m$ $(y/L=0.80)$ and $51m$ $(y/L=1.05)$ as shown in Fig. 5.

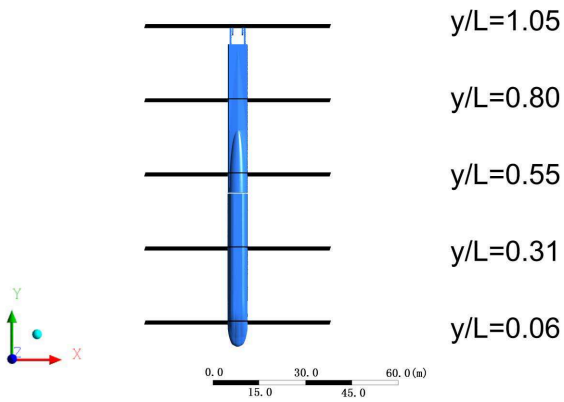
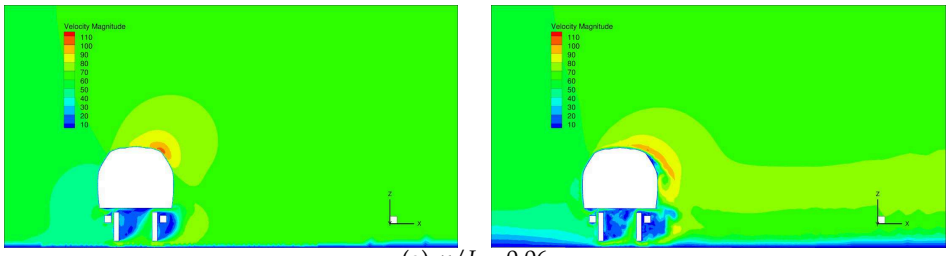
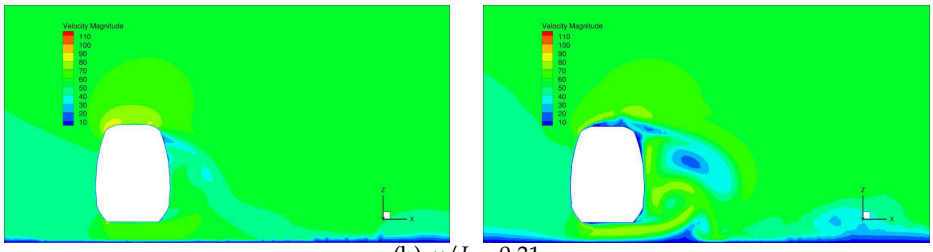


Figure 5: A view of the cross sections located on the y-axis.

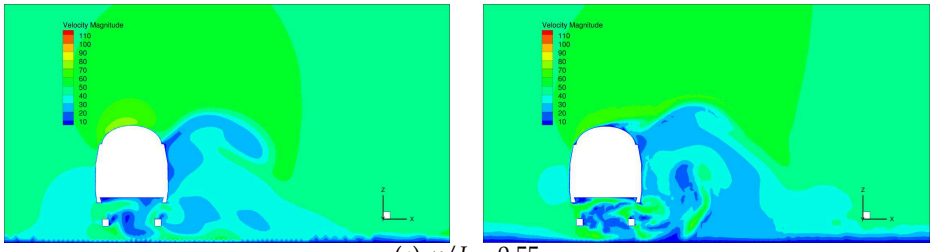
We first compare the velocity contours computed by CFX and our algorithm. As presented in Fig. 6, the velocity distribution is more or less the same, the higher value is located at the higher leeward edge of the leading nose, lower velocity is developed in the low leeward areas as well as the areas near the wheels. The contours also show that the distribution of the velocity field and the dominant vortices are similar. However, there are noticeable differences in smaller vortices and their locations. In Fig. 7, the dominant



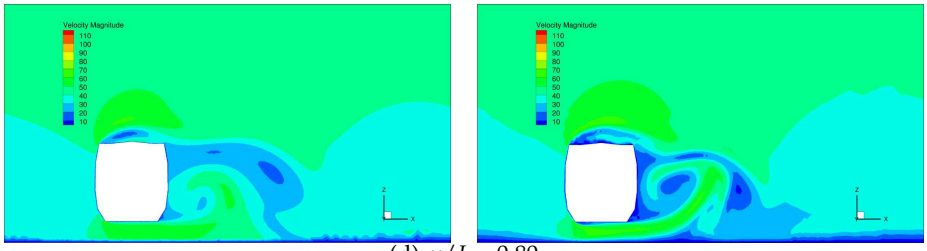
(a) $y/L=0.06$



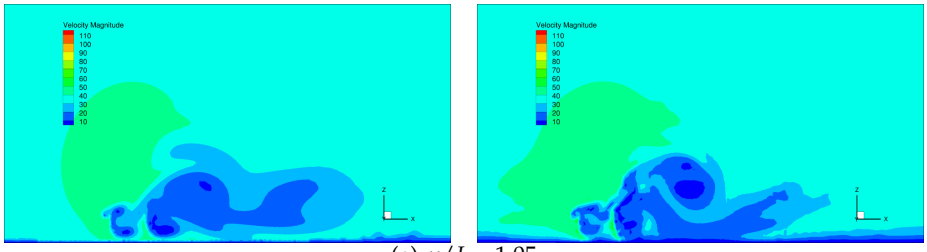
(b) $y/L=0.31$



(c) $y/L=0.55$

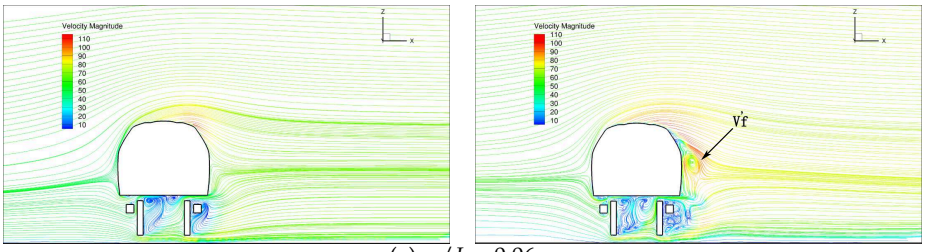


(d) $y/L=0.80$

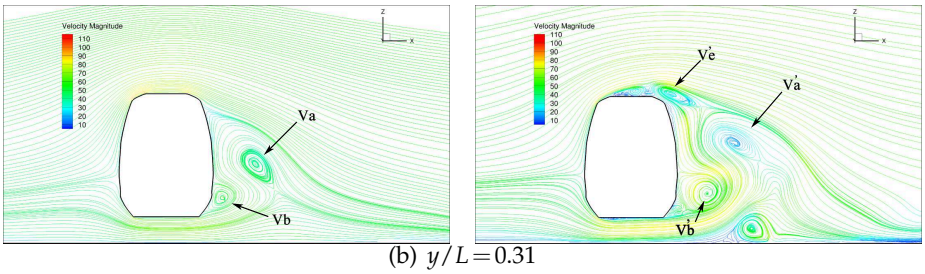


(e) $y/L=1.05$

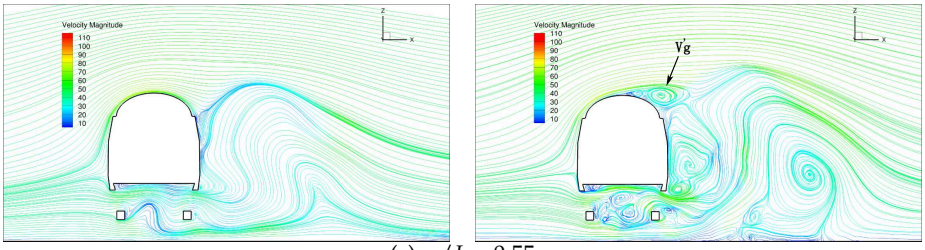
Figure 6: Velocity contours at different locations on the y-axis, $t=1s$, $\beta\approx 20.27^{\circ}$. Left: CFX; Right: NKS.



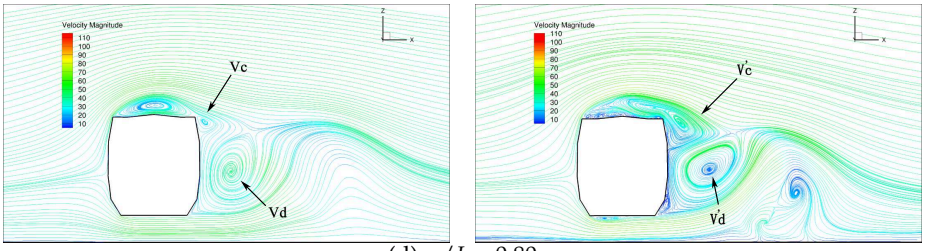
(a) $y/L=0.06$



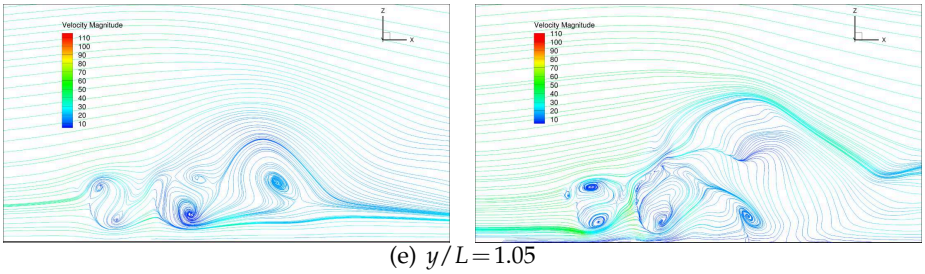
(b) $y/L=0.31$



(c) $y/L=0.55$



(d) $y/L=0.80$



(e) $y/L=1.05$

Figure 7: 2D streamlines at different locations on the y-axis, $t=1s$, $\beta\approx 20.27°$. Left: CFX; Right: NKS.

vortices V_a, V_b, V_c and V_d obtained by CFX can be found clearly in the numerical results of our algorithm, $V_a^{'}$, $V_b^{'}$, $V_c^{'}$ and $V_d^{'}$. Besides these main vortices, others, for example, $V_e^{'}$, $V_f^{'}$, $V_g^{'}$ and some small-scale vortices can be found in our results.

The differences of the flow structure can also be found by observing the surface streamline patterns on the leeward side of the train given in Fig. 8. The surface streamlines obtained by our algorithm are more turbulent than results from CFX. This is mainly due to the loss of flow information when the instantaneous quantity is decomposed in time-averaged and fluctuating quantities in the approach that CFX uses. The comparisons show that our simulation provides more details of the flows around the high-speed train under crosswind.

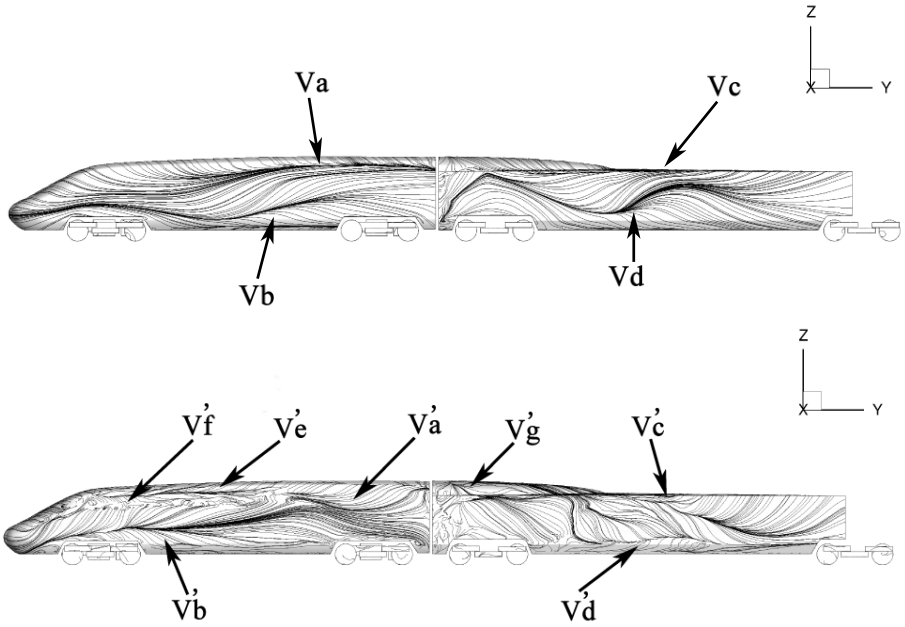


Figure 8: Surface streamline patterns on the leeward side for the train. Upper: CFX; Lower: NKS.

4.2 Parallel performance

In this subsection, we report the parallel performance of our algorithm. Two different meshes, one with about 7.2×10^6 elements (5.3×10^6 DOF) and the other with about 1.4×10^7 elements (1.0×10^7 DOF) are employed for the scalability test. A fixed time step $\Delta t = 4.0\times10^{-3}s$, a relative tolerance for the linear solver 1.0×10^{-4} and a relative tolerance for the nonlinear solver 1.0×10^{-12} are applied for all of the tests. Based on the first 20 time steps, Newton, GMRES and Time denote the average number of nonlinear iterations, GMRES iterations and the average walltime (in seconds) per time step, respectively.

Table 1 and Fig. 9 show the scalability of the computation with several different num-

Table 1: Parallel performance of the NKS algorithm. Here the level of ILU fill-ins $\ell=1$, the overlapping size for RAS $\delta=4$.

n_p	DOF=5.3×10^6			DOF=1.0×10^7		
	Newton	GMRES	Time	Newton	GMRES	Time
128	3.691	94.039	137.915	-	-	-
256	3.691	92.263	74.811	3.762	122.949	171.827
512	3.762	94.797	46.722	3.524	120.743	93.644
1024	3.143	87.182	32.701	3.619	123.092	62.427

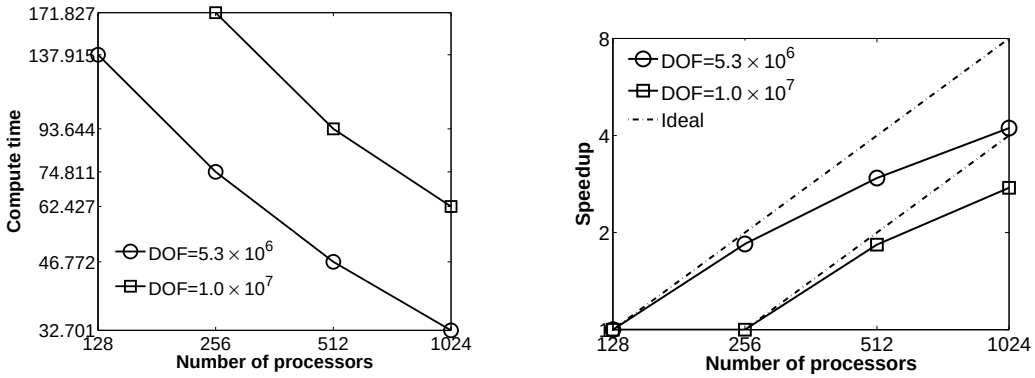


Figure 9: The average compute time per time step and the speedup (log-log scaled). Here the level of ILU fill-ins $\ell=1$, the overlapping size for RAS $\delta=4$.

ber of processors. We see that the number of nonlinear iterations per time step is quite small since the solution of the previous time step is used as the initial guess. The parallel results including the number of linear iterations and the total compute time show that our algorithm and software implementation are quite scalable.

In subdomain solvers, the level of fill-ins of ILU affects the performance of the NKS algorithm. Tables 2 and 3 show the comparison with different level of fill-ins ℓ. We see that larger ℓ helps in reducing the number of GMRES iterations but fewer level of ILU fill-ins $\ell=1$ provides better compute time.

We next present a comparison of additive Schwarz method with overlap δ changing from 0 to 4 using 1024 cores. Generally speaking, in the additive Schwarz method, the more overlap between subdomains, the faster the convergence is. However, the subproblems as well as the amount of communication become larger at the same time and result in the computing time increase. So, finding a good trade-off between the number of iterations and the computing time is important. The results shown in Table 4 verify this view. From this table, we observe that the number of GMRES iterations is decreased when large overlaps are used, but smaller overlap, $\delta=2$, offers the best timing results.

Table 2: Performance of NKS with respect to the level of ILU fill-ins ℓ. Here the overlapping size for RAS $\delta=4$, DOF$=5.3\times10^6$.

n_p	Newton			GMRES			Time		
	ILU(0)	ILU(1)	ILU(2)	ILU(0)	ILU(1)	ILU(2)	ILU(0)	ILU(1)	ILU(2)
128	3.810	3.691	3.476	172.750	94.039	57.233	151.039	137.915	170.280
256	3.857	3.619	3.190	168.605	92.263	55.015	84.147	74.811	82.465
512	3.875	3.762	3.286	166.988	94.797	57.203	52.792	46.772	50.091
1024	3.714	3.143	3.667	167.885	87.182	61.169	45.439	32.701	40.292

Table 3: Performance of NKS with respect to the level of ILU fill-ins ℓ. Here the overlapping size for RAS $\delta=4$, DOF$=1.0\times10^7$.

n_p	Newton			GMRES			Time		
	ILU(0)	ILU(1)	ILU(2)	ILU(0)	ILU(1)	ILU(2)	ILU(0)	ILU(1)	ILU(2)
256	3.810	3.762	3.714	254.600	122.949	84.603	373.825	171.827	223.134
512	3.810	3.524	3.810	250.512	120.743	86.838	366.144	93.644	124.798
1024	3.762	3.619	3.492	256.937	123.092	84.569	149.742	62.427	80.702

Table 4: Performance of NKS with respect to the overlapping size δ. Here the level of ILU fill-ins $\ell=2$.

RAS overlap δ	DOF=5.3×10^6			DOF=1.0×10^7		
	Newton	GMRES	Time	Newton	GMRES	Time
0	3.810	113.950	67.721	3.810	148.600	120.346
1	3.810	81.975	39.643	3.810	114.550	93.001
2	3.476	66.740	34.336	3.714	99.346	77.354
3	3.667	64.649	39.588	3.667	93.104	78.375
4	3.667	61.169	40.292	3.429	84.569	80.702

4.3 Influence of Reynolds number

When solving fluid dynamic problems like the external flows around the high-speed train under crosswind, the Reynolds number has a major impact on the complexity of the flows and the difficulty of the discrete system of equations. In this test, we consider Reynolds

Table 5: The robustness of the algorithm with respect to various values of Reynolds numbers. Here DOF $=1.0\times10^7$, $n_p=1024$.

Wind velocity	Train velocity	Reynolds number	Newton	GMRES	Time
0	1	2.194×10^5	2.857	78.517	65.100
0	5	1.097×10^6	2.857	78.233	70.583
0	10	2.194×10^6	2.905	78.672	61.992
10	20	4.906×10^6	2.905	78.656	70.277
30	100	2.291×10^7	3.476	85.616	82.522

numbers from 2.194×10^5 to 2.291×10^7 depending on the velocity of the crosswind and the speed of the train. The numerical results shown in Table 5 indicate the NKS algorithm works well for this range of Reynolds number.

5 Conclusions

Accurate simulation of external flows around a high-speed train under crosswind is a challenging computational problem. Finding a robust and scalable algorithm is a key issue for obtaining meaningful results in a timely matter for actual engineering applications. In this paper, we developed a parallel Newton-Krylov-Schwarz algorithm for the fully implicit solution of the full 3D unsteady incompressible Navier-Stokes equations. After the discretization, the nonlinear algebraic systems are solved by using an inexact Newton method in each time step. The GMRES method accelerated by an overlapping Schwarz preconditioner is used as the linear solver in each Newton step. We tested the algorithm for flows passing a high-speed train with realistic geometry and different Reynolds numbers. Based on the numerical results, we concluded that the algorithm and our software implementation are robust with a wide range of Reynolds numbers up to 2.291×10^7 and exhibit a good scalability with up to 1024 processors.

Acknowledgments

The research is supported in part by the Knowledge Innovation Program of the Chinese Academy of Sciences under KJCX2-EW-L01, and the International Cooperation Project of Guangdong province under 2011B050400037.

References

[1] S. R. Ahmed, G. Ramm, and G. Faltin, Some salient features of the time-averaged ground vehicle wake, SAE paper, 840300, 1984.
[2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and, H. Zhang, PETSc Users Manual, Technical Report, Argonne National Laboratory, 2012.
[3] X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput., 21 (1999), 792-797.
[4] R. Chen and X.-C. Cai, Parallel one-shot Lagrange-Newton-Krylov-Schwarz algorithms for shape optimization of steady incompressible flows, SIAM J. Sci. Comput., 34 (2012), 584-605.
[5] R. Chen, Y. 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., (to appear).
[6] B. Diedrichs, Studies of Two Aerodynamic Effects on High-Speed Trains: Crosswind Stability and Discomforting Car Body Vibrations Inside Tunnels, Ph.D. Thesis, Royal Institute of Technology (KTH), 2006.
[7] T. Favre, B. Diedrichs, and G. Efraimsson, Detached-Eddy simulations applied to unsteady crosswind aerodynamics of ground vehicles, Progress in Hybrid RANS-LES Modelling, Springer, 2010, 167-177.

[8] L. P. Franca and S. L. Frey, Stabilized finite element method: II. The incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg., 99 (1992), 209-233.
[9] E. Guilmineau, Computational study of flow around a simplified car body, J. Wind. Eng. Ind. Aerod., 96 (2008), 1207-1217.
[10] H. Hemida and C. Baker, Large-eddy simulation of the flow around a freight wagon subjected to a crosswind, Comput. Fluids, 39 (2010), 1944-1956.
[11] H. Hemida and S. Krajnovic, LES study of the impact of the wake structures on the aerodynamics of a simplified ICE2 train subjected to a side wind, presented at the 4th International Conference on Computational Fluid Dynamics, Ghent, Belgium, July 10-14, 2006.
[12] Q. Hu and J. Zou, Nonlinear inexact Uzawa algorithms for linear and nonlinear saddle-point problems, SIAM J. Optim., 16 (2006), 798-825.
[13] F.-N. Hwang and X.-C. Cai, A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations, J. Comput. Phys., 204 (2005), 666-691.
[14] 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 (2010), 199-208.
[15] G. Karypis, R. Aggarwal, K. Schloegel, V. Kumar, and S. Shekhar, ParMETIS home page, http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview.
[16] W. Khier, M. Breuer, and F. Durst, Flow structure around trains under side wind conditions: a numerical study, Comput. Fluids, 29 (2000), 179-195.
[17] S. Krajnovic, Numerical simulation of the flow around an ICE2 train under the influence of a wind gust, presented at the International Conference on Railway Engineering, Hong Kong, China, March 25-28, 2008.
[18] Y. Saad, Iterative Method for Sparse Linear Systems, SIAM, 2003.
[19] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge university press, Cambridge, 1996.
[20] Y. Wu and X.-C. Cai, A parallel two-level method for simulating blood flows in branching arteries with the resistive boundary condition, Comput. Fluids, 45 (2011), 92-102.
[21] G. W. Yang, D. L. Guo, S. B. Yao, and C. H. Liu, Aerodynamic design for China new high-speed trains, Sci. China Tech. Sci., 55 (2012), 1923-1928.
[22] D. Zhou, H. Q. Tian, and Z. J. Lu, Influence of the strong crosswind on aerodynamic performance of passenger train running on embankment, J. Traffic Transp. Eng., 7 (2007), 6-9.